

Mathematical Optimization Methodology for Neutron Filters

Bachelor Thesis

Author:

Jos DE WIT
4007441
Applied Physics

Review committee:

Dr.ir. J.L. KLOOSTERMAN
Dr.ir. M. ROHDE

November 2012 - July 2013



Faculty of Applied Sciences
Nuclear Energy and Radiation Applications (NERA)

NERA-131-2013-007

Abstract

This thesis describes a mathematical optimization methodology for neutron filters. An important application for neutron filters is Boron Neutron Capture Therapy (BNCT). For this therapy, boron-10 (^{10}B) is brought in the tumor, it captures a neutron and falls apart in two small particles. This gives extensive damage to the tumor cell. The boron-capture reactions occur with thermal neutrons ($E < 1\text{eV}$). However neutrons are moderated in the tissue, so only neutrons with higher energy will reach a deeper tumor. Therefore epithermal neutrons ($1\text{eV} < E < 10\text{keV}$) are needed for this therapy. The fast neutron and gamma dose contributions should be minimized, because these damage the healthy tissue.

The goal of this thesis is to investigate if and how it is possible to tailor a neutron spectrum by mathematical optimization of the thicknesses of material layers in a filter arrangement.

In this thesis an optimization methodology for a neutron filter is developed and performed. The research contains two parts: an optimization on the neutron flux spectrum whereby the epithermal neutron flux is maximized and an optimization with a realistic model of a tumor and healthy tissue, in which the boron dose ratio between the tumor and the healthy tissue is maximized.

As neutron source a light water reactor core (3% enriched uranium) is modeled. Bi, Fe, Al and AlF are used as moderator materials in the filter. Cd and ^{10}B are used as thermal neutron absorbers and the last filter layer consists of Pb for gamma shielding. The simulated annealing method is used for the optimization. This is a probabilistic optimization algorithm inspired by the process of cooling down a metal into crystalline structure. The optimization is done in MATLAB. The neutron calculations are done with XSDRN, a deterministic code for 1D neutron transport calculations. The macroscopic cross sections for the calculation are generated with CSAS.

The optimization whereby the epithermal neutron flux is maximized gives a filter configuration for which the total neutron flux consists for 92% of epithermal neutron flux. The fast and thermal neutron fluxes are both 4% of the total neutron flux, while the total neutron flux is 2.2% of the flux before the filter. The optimization on the boron-dose ratio gives a filter with a 14% higher boron-dose ratio than for a situation without filter. However, other dose contributions (fast neutron and gamma dose) are not yet taken into account. The imposed limit on the fast neutron flux was reached, but at the expense of a low absolute dose rate in the tumor tissue.

Contents

Abstract	1
List of Symbols	6
1 Introduction	7
1.1 Introduction	7
1.2 Boron Neutron Capture Therapy	7
1.2.1 History and current status	8
1.2.2 Spectral tailoring for BNCT	8
1.2.3 Desired Neutron beam	9
1.3 Other applications	10
1.4 Scope of this thesis	10
1.5 Overview over the thesis	10
2 Theory	11
2.1 Introduction	11
2.2 Cross sections	11
2.2.1 Microscopic cross sections	11
2.2.2 Macroscopic cross section	12
2.2.3 Scattering	13
2.2.4 Absorption	14
2.3 Neutron transport	14

2.4	Neutron filter	15
2.4.1	Neutron source	15
2.4.2	Moderators	15
2.4.3	Absorption	16
2.4.4	Materials in literature	16
2.5	Optimization	16
2.5.1	Simulated annealing	17
3	Calculation methods	19
3.1	Introduction	19
3.2	Macroscopic cross section generation	19
3.3	1D1 neutron transport calculation with XSDRN	20
3.4	Optimization and control with MATLAB	20
3.4.1	Control of documentation codes	21
3.4.2	Simulated annealing optimization with MATLAB	22
3.4.3	Data analysis with MATLAB	24
4	Filter optimization on neutron flux spectrum	25
4.1	Introduction	25
4.2	Materials and CSAS calculation	25
4.3	Spectrum calculation with XSDRN	26
4.4	Optimization parameters	26
4.4.1	Goal function	26
4.4.2	Boundary conditions	27
4.4.3	Optimization settings	27
4.5	Optimization results	27
4.6	Spectrum analysis	30
4.7	Conclusions and discussion	30

5	Filter optimization on boron-dose ratio between tumor and tissue	32
5.1	Introduction	32
5.2	Materials and CSAS calculation	32
5.3	Spectrum calculation in XSDRN	33
5.4	Optimization parameters	33
5.4.1	Goal function	33
5.4.2	Boundary conditions	33
5.4.3	Optimization settings	34
5.5	Optimization results	34
5.6	Spectrum and reaction rate analysis	36
5.7	Reaction rate without filter	38
5.8	Reaction rate optimization without fast flux limit	39
5.9	Conclusions and discussion	41
6	Conclusions and recommendations	42
6.1	Introduction	42
6.2	Conclusions	42
6.3	Recommendations	43
6.3.1	Calculation method	43
6.3.2	Input data	43
6.3.3	Goal function	43
	Bibliography	45
	Appendix A	47
	Cross sections of filter materials and B10	47
	Appendix B	51
	MATLAB code of spectrum optimization	51

MATLAB code of dose ratio optimization	55
MATLAB code for the calculation without filter	60

List of Symbols

Symbol	Quantity	Units
A	atomic mass number	-
c	coefficient in goal function	-
D	dose	Gy
E	energy	eV
f	goal function (theory)	-
G	goal function (results)	-
I	(beam) intensity	$\text{cm}^{-2} \text{s}^{-1}$
M	atomic mass	g mol^{-1}
N	number of atoms	-
N_A	number of avogadro	mol^{-1}
n	angular neutron density	m^{-3}
p	chance	-
R	interaction rate	s^{-1}
\vec{r}	position in 3D	cm
R_0	term in goal function	s^{-1}
s	external neutron source	s^{-1}
t	time	s
T	temperature	K
\vec{v}	speed	m s^{-1}
x	position	cm
α	temperature decrement factor	-
β	variating constant	-
μ	cosine of the angle between neutron velocity and x -axis	-
ξ	average logarithmic energy loss	-
ρ_a	atom density	$\text{atoms barn}^{-1} \text{cm}^{-1}$
ρ_m	mass density	g cm^{-3}
Σ	macroscopic cross section	cm^{-1}
σ	microscopic cross section	cm^2 , barn
ϕ	neutron flux	$\text{cm}^{-2} \text{s}^{-1}$
$\hat{\Omega}$	direction neutron velocity	-

Chapter 1

Introduction

1.1 Introduction

This thesis describes an optimization methodology of a neutron filter. One of the main applications of neutron filters is the Boron Neutron Capture Therapy (BNCT). This chapter first gives a short overview on BNCT and the favorable neutron beam properties for that. After that some other applications are mentioned. This chapter ends with a more specific description of the scope of this thesis and an overview over the report.

1.2 Boron Neutron Capture Therapy

For the application of BNCT, a Boron-10 (^{10}B) compound is injected in the blood of the patient. With some chemical carriers it is possible to bring more ^{10}B to a tumor than to healthy tissue. When the tumor is irradiated with neutrons, ^{10}B atoms capture neutrons and fall apart in a lithium atom and a helium atom (α -particle).

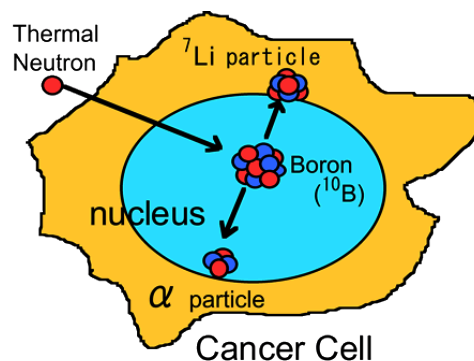


Figure 1.1: The (n, α) reaction in a cancer cell with BNCT

This (n, α) reaction gives a extensive damage to the DNA near the place where the reaction occurs. The travel distance of α -particles and Li nuclei is about a cell diameter, so there is

a high probability of damaging or even killing the cell. If the difference in ^{10}B concentration between tumor and healthy tissue is big enough, the amount of capture reactions in the tumor is significantly larger than in the healthy tissue. Therefore the tumor can get significantly more irradiation dose than the healthy tissue. The DNA of the tumor cells will be damaged and that results in a reduction of the tumor size. That is why this therapy can be very successful, especially for deeper tumors, tumors in neck region and for tumors which are dispersed. For those tumors surgical operations are not possible and selective irradiation from outside is difficult. BNCT gives the possibility to select the tumor cells from the inside through ^{10}B carriers. The main application of this therapy is on tumors in neck and head regions.

1.2.1 History and current status

Neutrons were first discovered by Chadwick in 1932[1]. Only three years later, in 1935, it was discovered that bombarding ^{10}B with slow neutrons lets it fall apart in a ^4He and a ^7Li particle[2]. In 1936 Locher published about the therapeutic possibilities of this reaction[3]. In the following years more research was done on this therapy. The first experiments were on mouse tumors. In 1950 the first trials on humans were done, without success. The patients died within 6 to 21 weeks, which was ordinary for such brain tumors at that time. From 1968 more successful trials were done in Japan[4]. This success resulted in more research and trials in US from 1994 and in Petten (Netherlands) from 1997.

Despite the simple concept and decades of research, BNCT is still in the experimental phase[5]. The main challenge of BNCT research is finding good ^{10}B carriers to bring it to the tumor and much less to the healthy tissue. Also standardization of results of clinical trials is necessary to improve the collaboration.

1.2.2 Spectral tailoring for BNCT

Several studies have been performed on which neutrons must be used for BNCT[6, 7, 8]. All those studies conclude that for irradiation epithermal neutrons ($1\text{eV} < E < 10\text{keV}$) must be used. Verbeke [11] conclude that even for shallow tumors epithermal neutrons are favorable. Even if the main tumor mass is shallow, the microscopic fingerlets spreading can reach greater depths. These fingerlets must get enough dose too. The shallow tumor will get enough thermal neutrons, because of the reflection of moderated thermal neutrons from the deeper tissue. Nievaart mentioned that there are two topics important for spectral tailoring for BNCT[4]:

1. Defining the source neutron energies of the BNCT treatment beam in order to obtain the maximum ^{10}B absorption reactions in the tumor and the minimum reactions in the healthy tissue. In addition, the location, direction and dimensions of the BNCT treatment beam need to be optimized for every individual tumor size and location.
2. Developing and constructing the neutron filter with the appropriate materials and obtain those beam properties starting with an available source.

This thesis focuses mainly on the second topic, but it will be combined with topic one.

There are many studies on filter design for BNCT[8, 9, 10, 11]. Most of them tried different filter configurations and compared them with each other. No mathematical optimizations were done to design the filter. The neutron flux calculations are mainly done with Monte Carlo codes. Azahra[10] achieved 72% epithermal neutrons in the final beam. The other cited studies do not mention a percentage of epithermal neutrons in the final beam, but compare the epithermal flux to dose parameters.

1.2.3 Desired Neutron beam

The probability of a neutron capture reaction in ^{10}B is highest for thermal neutrons (energy less than 0.5 eV). So it is desirable to get mostly thermal neutrons in the tumor. However irradiation with thermal neutrons is not effective, because (with deeper tumors) the neutrons will not reach the tumor. Human tissue contains a lot of light elements which let the neutrons slow down. Therefore, for effective irradiation, epithermal neutrons are desirable.

Besides thermal and epithermal neutrons, there are fast neutrons ($E > 10$ keV). These neutrons are not desirable for irradiation because they will induce recoil reactions, which will damage the healthy tissue before the tumor. Together with the neutrons and because of radiative neutron capture, there will also be gamma rays. These gamma rays give a non-selective dose to the tissue, because this dose is independent of the ^{10}B concentration. Thereby the gamma dose contribution to the tumor is the same as to the healthy tissue. The gamma radiation should therefore be minimized. In table 1.1 the different dose contributions are summarized.

Table 1.1: Overview of major dose components in BNCT.[4]

Physical dose name		Dose symbol	Reaction type	Reaction scheme	Remarks
Thermal	Boron dose	D_B	n, α	$^{10}\text{B} + {}^1_0\text{n} \rightarrow {}^7\text{Li} + {}^4\text{He}$	Biological effects in tumor and normal tissue are related to ^{10}B micro distribution
	Thermal neutron dose	D_p	n,p	$^{14}\text{N} + {}^1_0\text{n} \rightarrow {}^{14}\text{C} + {}^1_1\text{p}$	Induced proton 620 keV
	Induced gamma-ray dose	D_γ	n, γ	$^1\text{H} + {}^1_0\text{n} \rightarrow {}^2\text{H} + \gamma$	Induced γ -rays 2.2MeV
Fast	Fast neutron dose	D_n	n,n	$^1_0\text{H} + {}^1_0\text{n} \rightarrow {}^1_0\text{n} + {}^1_1\text{p} + {}^0_{-1}\text{e}$	Energy recoiling proton is on average half the neutron energy

1.3 Other applications

Besides BNCT, there are other applications of neutron beams. One important application is research. With neutron scattering it is possible to investigate the structure and other properties of materials. The scientific branches where scattering is used are, for example, material science, life science and chemistry. Tailoring the spectrum of neutron beams could be useful for improving this research as well.

Also neutron radiography is an interesting field of application. With neutrons other properties of materials are measured than with electromagnetic waves, such as X-rays. These are more industrial applications.

1.4 Scope of this thesis

This thesis will mainly focus on the second part of spectral tailoring mentioned in subsection 1.2.2. Topic of investigation is adjusting the neutron filter to get from an existing neutron source the right spectrum for irradiation. In this thesis it is investigated if and how it is possible to tailor the neutron spectrum by mathematical optimization of the thickness of material layers in a filter arrangement. Besides this neutron spectrum tailoring, it is investigated if it is possible to optimize a filter on the boron dose ratio between tumor and healthy tissue. For that optimization a realistic example of tissue and tumor containing boron-10 is considered. Neutron spectrum calculations are done with a deterministic 1D neutron transport code. For the optimization the probabilistic simulated annealing algorithm is used.

1.5 Overview over the thesis

After this chapter, the theory about neutrons and optimization is explained. In chapter 3 the calculation methods for neutron calculation and optimization are treated. After that the specific calculations and their results are given and briefly discussed. This is done in two chapters: the fourth chapter deals with the optimization on neutron flux spectrum and the fifth chapter treats the optimization on boron dose difference between the tumor and the healthy tissue. The thesis ends with the conclusions and recommendations. After that the bibliography and appendixes are given.

Chapter 2

Theory

2.1 Introduction

This section first explains the concept of neutron cross sections. After that the equation for neutron transport is explained. These sections are based on *Duderstadt and Hamilton*[12]. Subsequently the theory of neutron filtering is explained and some material properties are treated. Finally, attention is paid to some optimization theory.

2.2 Cross sections

2.2.1 Microscopic cross sections

When a neutron moves through a material there is a particular probability that it will interact with the nuclei of the material. To quantify this a thin layer of the material is taken into account. If the layer is sufficiently thin, no atoms are shielded by other atoms. In this case the interaction rate is proportional to the intensity of the neutron beam and the number of atoms in the layer. This gives the following equation:

$$R = \sigma IN \tag{2.1}$$

In which R is the interaction rate in $[\text{cm}^{-2} \cdot \text{s}^{-1}]$, I is the neutron beam intensity in $[\text{cm}^{-2} \cdot \text{s}^{-1}]$, N is the number of atoms in the layer in $[\text{cm}^{-2}]$ and σ is the constant of proportionality in $[\text{cm}^2]$. This constant σ is called the microscopic cross section. Since the nuclear radius is roughly 10^{-12} cm, the geometrical cross section is in the order of 10^{-24}cm^2 . The real microscopic cross section σ is often measured in units of this size, called barns. Due to quantum mechanical effects and resonances σ can be very different from the geometrical cross section.

There are basically two different ways of interaction between neutrons and nuclei: absorption and scattering. These can be subdivided as shown in figure 2.1. The scattering cross section contains two parts: elastic scattering and inelastic scattering. In the case of elastic scattering, the target nucleus remains in the ground state. After inelastic scattering the target nucleus is

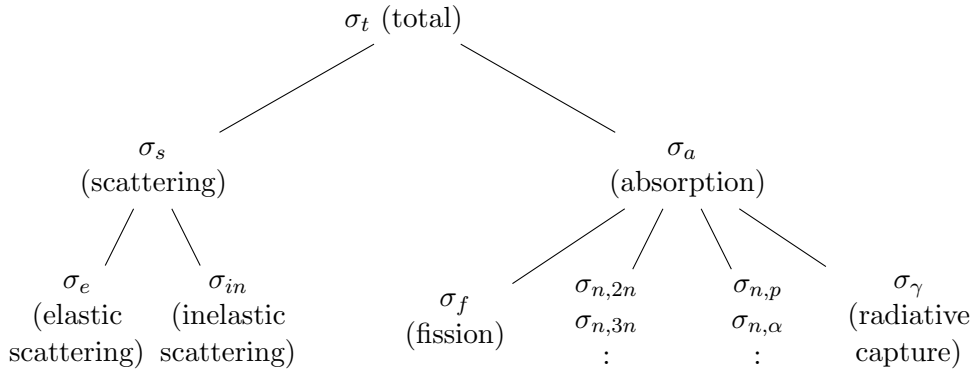


Figure 2.1: schematic representation of different cross sections

left in an excited state. This nucleus will relax by emitting a gamma ray. The sum of these two cross sections is the total scatter cross section σ_s . There are more possible types of absorption cross sections. The absorbed neutron can induce fission, or emission of gamma radiation, new neutrons or other particles. The total microscopic cross section is given by the sum of the different parts:

$$\sigma_t = \sigma_s + \sigma_a = \sigma_e + \sigma_{in} + \sigma_f + \sigma_{n,\alpha} + \sigma_{n,2n} + \sigma_\gamma + \dots \quad (2.2)$$

The value of cross sections strongly depend on the neutron energy (or speed).

2.2.2 Macroscopic cross section

So far are only thin layers taken into account. If the target layer is thicker, the deeper nuclei are shielded by other nuclei. The beam intensity will decrease further into the material. Figure 2.2 is a schematic view of this situation.

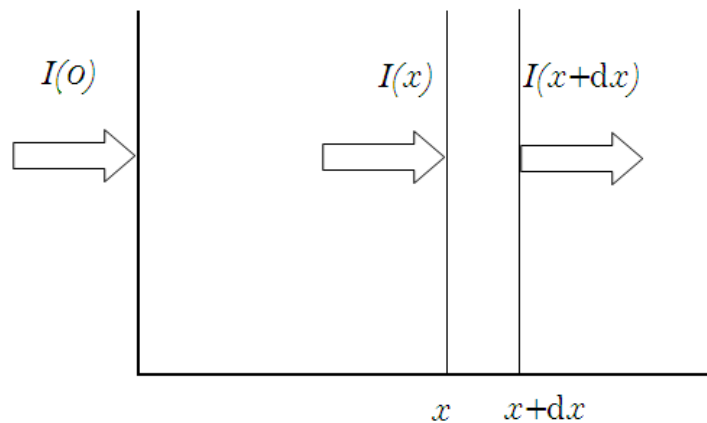


Figure 2.2: schematic view of a neutron beam through a material

For the thin layer with width dx , the microscopic cross section is valid. Any type of inter-

action removes a neutron from the initial neutron beam. The interaction rate R is therefore the same as the loss of intensity of the initial neutron beam in that thin layer. Then, for the intensity of the neutron beam the following equation applies:

$$-dI(x) = -[I(x+dx) - I(x)] = \sigma_t I N dx \quad (2.3)$$

In which $I(x)$ is the intensity as function of the dept and N is the number of nuclei per volume. Equation (2.3) can be transformed into the differential equation:

$$\frac{dI}{dx} = -N\sigma_t I(x) \quad (2.4)$$

If at $x = 0$ the beam intensity is I_0 , the solution of this equation for the intensity is:

$$I(x) = I_0 \exp(-N\sigma_t x) \quad (2.5)$$

The product $N\sigma_t$ is also known as the macroscopic cross section and is represented by the symbol Σ_t . This macroscopic cross section characterizes the probability of neutron interaction in a macroscopic piece of material. An other way of interpretation of this quantity is the fractional change in beam intensity occurring over a distance dx :

$$\Sigma_t = \frac{\left(\frac{-dI}{I}\right)}{dx} \quad (2.6)$$

Up to now, only the total macroscopic cross section is considered. The macroscopic cross sections can be subdivided in the way of figure 2.1. The absorption and scattering macroscopic cross sections are respectively defined as:

$$\Sigma_a \equiv N\sigma_a, \quad \Sigma_s \equiv N\sigma_s \quad (2.7)$$

The total macroscopic cross section is the sum of the partial cross sections. If there are different nuclei homogeneously distributed in a macroscopic piece of material, the macroscopic cross section of that material is defined by:

$$\Sigma = \sum_i N_i \sigma_i \quad (2.8)$$

Where N_i and σ_i are respectively the nuclide density and microscopic cross section of nuclide i .

2.2.3 Scattering

One of the two main parts of the total cross section is the scattering cross section. Scattering involves an individual neutron and a target nucleus. Only energy and momentum is exchanged between those particles. Inelastic scattering leaves the target nucleus in an excited state. This nucleus will often emit a photon when relaxing to the ground state. If the motion of the nucleus is negligible compared to that of the neutron, the neutron will transfer some of his energy to the nucleus and so it slows down.

For elastic scattering there is elastic resonance scattering. In that case, the neutron is absorbed in the nucleus after which a neutron is re-emitted and the nucleus remains in ground state. This resonance cross section is dependent on the quantum mechanical structure of the nucleus and is quite energy dependent.

The other elastic scattering is the potential scattering. For this type of scattering, the neutron doesn't penetrate into the nucleus, but interacts with the nucleus through the potential field of the nucleus. This cross section is not very dependent on energy and is approximately equal to the geometrical cross section.

For elastic scattering, kinetic energy and momentum are conserved. For elastic scattering with a nucleus of mass number A , the following equations apply:

$$\begin{aligned}\vec{v}_{n,before} &= \vec{v}_{n,after} + A\vec{v}_m \\ \frac{1}{2}(v_{n,before})^2 &= \frac{1}{2}(v_{n,after})^2 + \frac{1}{2}A(v_m)^2\end{aligned}\tag{2.9}$$

The index m stand for the nucleus used to slow down (moderate) the neutrons. The mass number A equals the number of neutrons and protons in the nucleus. With the assumption that the scattering direction is isotropic in the center-of-mass frame, the average energy loss in an elastic scattering collision is:

$$\frac{E_{n,before} - E_{n,after}}{E_{n,before}} = \frac{2A}{(1+A)^2}\tag{2.10}$$

The energy loss for elastic scattering is proportional to the initial energy. It follows from this equation that light nuclei are good moderators.

In nuclei, intended for moderation, neutrons can also be absorbed. The absorption of neutrons is usually undesirable for moderators. To take this effect into account, the moderating ratio or quality is defined as:

$$\text{Moderating ratio} \equiv \frac{\xi\Sigma_s}{\Sigma_a}\tag{2.11}$$

In which ξ is the average logarithmic energy loss of the neutrons and Σ_s en Σ_a are respectively the scattering and absorbing macroscopic cross section. For a nucleus with mass number A , ξ is defined as[12]:

$$\xi = 1 - \frac{(A-1)^2}{2A} \ln\left(\frac{A+1}{A-1}\right)\tag{2.12}$$

2.2.4 Absorption

The other main part of the cross section is the absorption cross section. With an absorption event, the neutron is captured in the nucleus and the nucleus is excited. The nucleus is often relaxed by emitting a photon, but also other emissions and fission are possible. The absorption cross section depends on the quantum mechanical structure of the nucleus. The probability of absorption is high when the sum of the neutron binding energy and kinetic energy is equal to an energy level of the nucleus. These resonances lead to narrow peaks in the absorption cross section.

2.3 Neutron transport

To describe neutron transport through a filter, the neutron transport equation need to be used. In chapter 4 of *Duderstadt and Hamilton*[12] this equation is extensively derived. For the 3D

case, the neutron transport equation is as follows:

$$\begin{aligned} \frac{\partial n}{\partial t} + v\hat{\Omega} \cdot \nabla n + v\Sigma_t n(\vec{r}, E, \hat{\Omega}, t) \\ = \int_{4\pi} d\hat{\Omega}' \int_0^\infty dE' v' \Sigma_s(E' \rightarrow E, \hat{\Omega}' \rightarrow \hat{\Omega}) n(\vec{r}, E', \hat{\Omega}', t) + s(\vec{r}, E, \hat{\Omega}, t) \end{aligned} \quad (2.13)$$

In this equation, n is the angular neutron density, v is the neutron speed and $\hat{\Omega}$ is the direction of the neutron velocity. The term $s(\vec{r}, E, \hat{\Omega}, t)$ stands for external source neutrons. If neutron transport is calculated in one direction, the one-dimensional form of the neutron transport equation can be used. It becomes:

$$\begin{aligned} \frac{1}{v} \frac{\partial \phi}{\partial t} + \mu \frac{\partial \phi}{\partial x} + \Sigma_t \phi(x, E, \mu, t) \\ = \int_{-1}^1 d\mu' \int_0^\infty dE' \Sigma_s(E' \rightarrow E, \mu' \rightarrow \mu) \phi(x, E', \mu', t) + s(x, E, \mu, t) \end{aligned} \quad (2.14)$$

In this equation, ϕ is the neutron flux and μ is the cosine of the angle of the neutron velocity with the x -direction. This angle reaches from 0 to π so μ reaches from -1 to 1.

2.4 Neutron filter

As earlier mentioned, neutrons interact with materials depending on neutron energy, nuclei density and the properties of the nuclei. This gives an opportunity to design a filter to tailor the spectrum of a neutron beam. This section deals with neutron sources and different types of materials and their role in a neutron filter arrangement.

2.4.1 Neutron source

There are different types of neutron sources. For a reasonable intense beam, only accelerator driven neutron sources and reactors are useful. The work in this thesis is based on a nuclear reactor as neutron source. When a reactor is in operation neutrons are continuously produced, moderated, absorbed and leaked. In a light water moderated reactor, this process gives a spectrum as shown in figure 2.3. The thermal neutrons (around 0.1 eV) are much present because of the strong moderating effect of the water. The fast neutrons (around 1 MeV) are created in the fission reactions.

2.4.2 Moderators

In many cases, the fast neutrons are not wanted in the neutron beam. Because the reactor spectrum contains many fast neutrons, the filter must slow down those fast neutrons. This can be done by scattering as described in paragraph 2.2.3. The materials used for slowing down the neutrons are called moderators. For BNCT the epithermal neutrons ($1\text{eV} < E < 10\text{keV}$) are favorable. Therefore the fast neutrons should be moderated and the scattering cross section for lower, epithermal energies should be low to prevent that the favorable epithermal neutrons become thermal. Absorption of fast neutrons is not desirable because then they can't contribute any more to the lower energy spectrum.

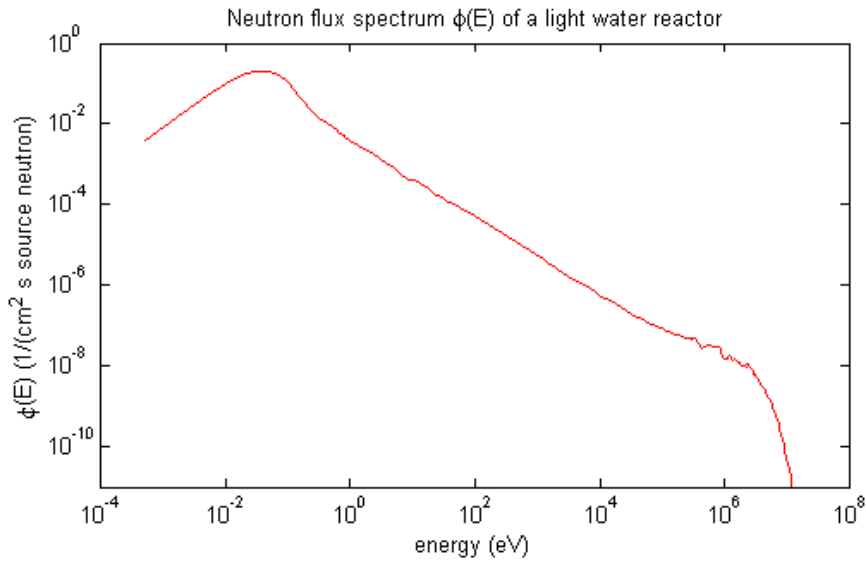


Figure 2.3: Typical neutron flux spectrum of a light water moderated reactor.

2.4.3 Absorption

For some applications, including BNCT, the thermal neutrons are not wanted in the neutron spectrum. Because these neutrons can not be accelerated to higher energies, they should be absorbed. The absorber materials should be positioned at the end of the filter, because in the moderation region there are also thermal neutrons produced. If the absorption is radiative capture, gamma radiation is produced. After the absorbers some gamma ray shielding should be provided, if gamma radiation is unwanted.

2.4.4 Materials in literature

In literature, many materials are studied for neutron filtering. Ross[9] used argon, aluminum, tin and cadmium. Liu[7] used also aluminum, cadmium and bismuth. Tracz[8], Bleuel[13] and Sakamoto[14] used aluminum and aluminumfluoride and some other materials. Rahmani[15] used magnesium, aluminum, iron, lead, carbon and nickel. Also BeO is used sometimes. From these materials Al, AlF, Fe, Ar, Mg, D₂O and BeO are mainly used as moderator. In Fe also (n,2n) reaction occur[16], which increase the neutron beam intensity. Cd is used as absorber and also B-10 is sometimes used as absorber. Bi and Pb are mainly used for gamma shielding.

2.5 Optimization

With a given configuration of a filter the neutron spectrum after the filter can be calculated. To find the best filter configuration, this configuration should be optimized. For optimization a goal function is needed, which gives a value to each configuration. This makes it possible to compare different solutions with each other and determine which is better. Using an optimization algorithm, the goal function can be maximized (or minimized). It is possible to

impose constraints on the solutions by determining boundary conditions. If these conditions are exceeded, the optimization will stop or the last optimization step will be undone.

There are mainly two types of optimization algorithms: deterministic and probabilistic. For deterministic optimization, the starting point and optimization input data determine the whole optimization path. Probabilistic optimization is a random process which can lead to an optimum. Probabilistic optimization is mainly used for complex problems. Neutron filters with several materials are very complex. For example Al, AlF and Fe which are all moderators with overlapping cross sections. The optimal layer thickness for each of them is quite dependent on the layer thickness of the others. This makes the optimization more complex. Also the strong non-linearity of successive neutron interactions leads to more complexity. There is also a danger to get stuck in a local maximum. Therefore an optimization method is needed which can handle those challenges. The simulated annealing algorithm is therefore chosen for this research.

2.5.1 Simulated annealing

The simulated annealing optimization algorithm is inspired by the process of cooling down and freezing of a metal into a crystalline structure with minimum energy. At high temperatures, the atoms can move easily with a slightly preference for places where the energy is low. Due to the high thermal energy the atoms can go through high energy barriers. When slowly cooling down, the atoms configure more and more into a minimum energy crystalline structure. With this algorithm a global optimum can be found with few danger to get stuck in a local maximum. At the starting temperature, there is globally searched for the area with the global optimum. When cooling down, the optimization focuses more locally in this area.

The simulated annealing starts with initial parameters and a goal function. From this point, the optimization parameters are randomly varied. With the new parameters the goal function is evaluated. Depending on how good the new goal function is in comparison with the old one and depending on the temperature, there is a probability that the new parameters are accepted. Even if the goal function becomes worse, they can be accepted. Assuming that is looked for a maximum, the probability that the new parameters j are accepted after the parameters i is:

$$p = \begin{cases} 1 & \text{if } f_j - f_i \geq 0 \\ e^{\frac{f_j - f_i}{T(t)}} & \text{if } f_j - f_i \leq 0 \end{cases} \quad (2.15)$$

In which f is the function to be maximized. The temperature decreases during the optimization process, therefore the probability of accepting a worse goal function also decreases.

The evolution of the temperature is important for the simulated annealing method. The temperature may not decrease too fast, because this gives a risk of being stuck in a local maximum, just as you get a bad crystal structure when you cool down some metal too fast. A relatively simple and common way is an exponential decreasing temperature. This is implemented by the rule[19]:

$$T_{t+1} = \alpha T_t \quad (2.16)$$

Here α is between 0 and 1. It is dependent on the system which is a good starting temperature. Busetty [19] says that the probability at T_0 with a typical value of Δf should be around 0.8. A schematic view of the simulated annealing algorithm is displayed in figure 2.4.

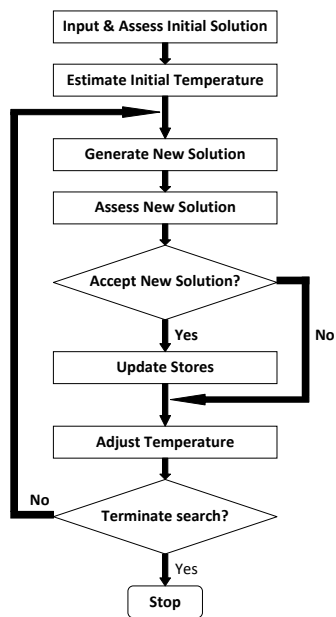


Figure 2.4: Block scheme of the simulating annealing optimization

Chapter 3

Calculation methods

3.1 Introduction

In this chapter, the calculation methods and codes used in this thesis are explained. The calculation methods can be divided in three main parts. First the CSAS code for generating macroscopic cross sections, after that the XSDRN code to calculate the neutron transport and last control and optimization with MATLAB. The optimization calculations are done on the computer cluster of TU Delft. For each calculation a core with a 2.2 GHz processor and about 4 GB random-access memory is used.

3.2 Macroscopic cross section generation

The Critical Safety Analysis Sequence with KENO V.a (CSAS5) in the SCALE code system was developed to calculate problem-dependent mixed cross sections for neutron multiplication factor (k_{eff}) and transport calculation. The cross section processing codes BONAMI, WORKER, CENTRM and PMC are used to create resonance, shielding and weighted cross sections. Figure 3.1 contains a schematic diagram of CSAS5 in.

For this thesis the sequence CSASI is used. CSASI calculates macroscopic cross sections for each of the processes and energy groups. For this thesis the 200 neutron/47 gamma library is used, which contains 200 neutron energy groups and 47 gamma energy groups. For many applications, gamma rays are an important factor to take into account. This 200N47G library gives that possibility.

In figure 3.2 an example of a CSAS input file is given. First the type 'CSASI' is specified. In the third line the energy group library is specified, in this case the 200N47G library. After that the nuclides are inserted. First the atomic symbol with the mass number and than the mixture number, the atom density in [atoms/(barn cm)] and last the temperature in Kelvin. Mixture 1 contains different atoms. Mixture 1 contains only one type of atoms. The next step is taking the nuclei together in a mixture and calculating the cross sections. The cross sections are written down in the file *jos_xs.out*. This library can be used by the next program for calculating the neutron flux spectrum.

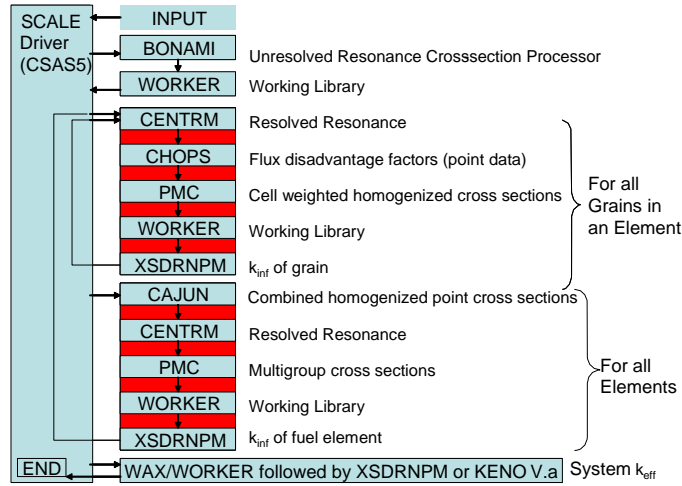


Figure 3.1: Schematic diagram of CSAS5 [20]

3.3 1Dl neutron transport calculation with XSDRN

XSDRNPM[17] is a one dimensional discrete-ordinates code for transport analysis. It is the latest in the series of the XSDRN family. XSDRNPM solves the 1D neutron transport equation 2.14 in a slab, cylindrical or spherical geometry. There is a lot of flexibility in the code to determine the spatial intervals, number of energy groups, the number of different materials and other parameters. There are several calculations possible, including k-calculations and fixed source calculations. The geometry, materials and their macroscopic cross sections from CSAS are given as input and XSDRNPM calculates the neutron (and gamma) flux spectrum iteratively. The spectrum can be normalized to the neutron source. In this thesis the neutron fission source is normalized to be 1. In this way the whole neutron flux is normalized to this setting.

The XSDRN code is a one dimensional deterministic one. The calculation time for the neutron flux is generally less than for probabilistic codes. For deterministic codes however, the geometry may not be very complicated. For this research a slab geometry is used.

3.4 Optimization and control with MATLAB

For this thesis, MATLAB is used to control CSAS and XSDRN and to perform the optimization. MATLAB is an environment for numerical computation, visualization and programming. Using MATLAB, data can be analyzed, algorithms developed and models created[18]. In the following subsections, first the control of the documentation codes from MATLAB is described. After that the implementation of the simulated annealing algorithm in MATLAB is described. At last, the data analysis with MATLAB is treated.

```

=csasi parm=centrm
JOS XS GENERATION
V7-200N47G
read comp
u-235      1 0 2.4467e-004 300 end
u-238      1 0 7.9110e-003 300 end
h-1        1 0 4.4558e-002 300 end
o-16       1 0 3.8590e-002 300 end
bi-209     2 0 2.8182e-002 300 end
end comp
read celldata
infhommedium 1 cellmix=3 end
infhommedium 2 cellmix=4 end
end celldata
end data
end
=shell
mv ft02f001 $RTNDIR/jos_xs.out
end

```

Figure 3.2: Example of a CSASI input file

3.4.1 Control of documentation codes

An example of a code to control with MATLAB is given below. After the setting of the temperature, which is used later in the code, a file is opened with the *fopen* command. In this case, the file is called *csasi* and the option *w* is given to clear an existing file with the same name or to make a new file. In the following lines, the file is filled with text with the *fprint* command. In the lines 10 to 14, the temperature is printed from the variable *T*. Finally the file is closed and with the *system* command the file *csasi* is executed. Figure 3.2 shows the file *csasi* which is created by this MATLAB script.

```

1
2     T=300; %Temperature (K)
3
4     fid=fopen('csasi','w');
5
6     fprintf(fid, '%s\n', '=csasi parm=centrm');
7     fprintf(fid, '%s\n', 'JOS XS GENERATION');
8     fprintf(fid, '%s\n', 'V7-200N47G');
9     fprintf(fid, '%s\n', 'read comp');
10    fprintf(fid, '%s %i %s\n', 'u-235      1 0 2.4467e-004', T, 'end');
11    fprintf(fid, '%s %i %s\n', 'u-238      1 0 7.9110e-003', T, 'end');
12    fprintf(fid, '%s %i %s\n', 'h-1        1 0 4.4558e-002', T, 'end');
13    fprintf(fid, '%s %i %s\n', 'o-16       1 0 3.8590e-002', T, 'end');
14    fprintf(fid, '%s %i %s\n', 'bi-209     2 0 2.8182e-002', T, 'end');
15    fprintf(fid, '%s\n', 'end comp');
16    fprintf(fid, '%s\n', 'read celldata');
17    fprintf(fid, '%s\n', 'infhommedium 1 cellmix=3 end');
18    fprintf(fid, '%s\n', 'infhommedium 2 cellmix=4 end');
19    fprintf(fid, '%s\n', 'end celldata');

```

```

20 fprintf(fid, '%s\n', 'end data');
21 fprintf(fid, '%s\n', 'end');
22 fprintf(fid, '%s\n', '=shell');
23 fprintf(fid, '%s\n', 'mv ft02f001 $RTNDR/jos_xs.out');
24 fprintf(fid, '%s\n', 'end');
25
26 fclose(fid);
27
28 system('batch6 -m csasi');

```

The control of XSDRNPM is done in the same way. It is also possible to execute (not change) files which are in the directory but aren't created by MATLAB. This will be used to let the XSDRNPM output be transformed before it can be read.

The XSDRNPM calculation creates a file containing the fluxes. This file must be transformed before it can be read by MATLAB. After that transformation, the flux data file is read by MATLAB and used to determine the next step in the optimization.

3.4.2 Simulated annealing optimization with MATLAB

In subsection 2.5.1 the theory of simulated annealing is briefly explained. The steps from figure 2.4 must be implemented in the MATLAB optimization code. Below an example of this implementation is given.

```

1 % simulated annealing with MATLAB
2
3 % determination of some parameters
4 no_grps=247; % number of energy groups
5 n_int=104; % total number of intervals
6 alpha=0.02; % part of goal function
7
8 % determination of first solution
9 d(1)=0; %starting point
10 d(2)=100; %cm thickness U+H2O
11 d(3)=20; %cm Fe
12 d(4)=10.4; %cm AlF3
13 d(5)=10; %cm Cd
14 d(6)=1; %cm Pb
15 for j=1:6
16 dist(j)=sum(d(1:j));
17 end
18
19 % calculate flux and goalfunction by first solution
20 run('csas_calc'); % calculate macroscopic cross sections
21 run('basic_calc'); % calculate some extra parameters such as grid, etc.
22 run('xsdrrn_calc'); % calculate neutron flux
23 run('calc_flux'); % read and process flux data
24 run('calc_goal'); % calculate goal function from fluxdata
25
26 % set some parameters for optimisation
27 m=500; % number of optimization steps
28 beta=0.08; % solution changing parameter
29 T=200; % starting temperature
30 g=0.99; % temperature multiplication factor

```

```

31 gamma_rand=2.5e-4;      % boundary value for gammarays
32 fast_rand=1e-3;        % boundary value for fast neutrons
33 results=zeros(9,m);    % empty matrix for results
34
35 % first result
36 results(:,1)=[goal; part; gamma; fast; zeros(2,1); d(3:6)'];
37
38 % optimization loop
39 for l=2:m
40     % variate solution randomly
41     var=rand(1,4).*2.*beta+(1-beta);
42     d(3:6)=d(3:6).*var;
43     for j=1:6
44         dist(j)=sum(d(1:j));
45     end
46     % calculate flux and determine goal function by new solution
47     run('xsdrn.calc');
48     run('calc_flux');
49     run('calc_goal');
50
51     % verify boundary conditions
52     if gamma>gamma_rand|fast>fast_rand
53         d(3:9)=d(3:9)./var;
54         results(:,l)=[results(1:5,l-1);0;2;d(3:9)'];
55     else
56         % decide if the new solution is accepted
57         if goal>results(1,l-1);
58             results(:,l)=[goal; ratio; part; gamma; fast; 1;0; d(3:9)'];
59         else
60             c=exp(-(results(1,l-1)-goal)/(0.02*T));
61             if c<rand
62                 d(3:9)=d(3:9)./var;
63                 results(:,l)=[results(1:5,l-1);c;1;d(3:9)'];
64             else
65                 results(:,l)=[goal; ratio; part; gamma; fast; c;-1;d(3:9)'];
66             end
67         end
68     end
69     % adjust temperature
70     T=g*T;
71 end
72
73 % saving results
74 dlmwrite('results.txt',results);

```

In the lines 3 to 6 some input parameters are set which will be used in the calculation. In lines 8-24 the first solution is determined and the flux and goal function for that solution is calculated. For that calculation two separate scripts are executed. The lines 26 to 36 are used to set some parameters for the optimization: number of steps, temperature, et cetera. Also the first solution is saved. In the lines 38 to 71 the optimization loop is written. First a new solution is generated and assessed, just as in figure 2.4. For this thesis the new solutions are generated by multiplication of the thickness of each layer with a random number. This number is taken from an uniform probability distribution between $1 - \beta$ and $1 + \beta$ ($0 < \beta < 1$) and is independent of the multiplication factor of the other layers. The solution is first tested against the boundary conditions. If these are exceeded, the new solution is rejected. If the boundary conditions aren't violated, the new solution is accepted or declined according to the probabilities of equation 2.15. The final step in the loop is to adjust the temperature. After the optimization

the results are saved in a text file. Note that the macroscopic cross sections are calculated only once in line 20. In this thesis, the composition of materials do not change.

For the optimization for this thesis, by far most time is consumed by the XSDRNPM calculations. Therefore it is less important to optimize the MATLAB code on calculation time.

3.4.3 Data analysis with MATLAB

After the optimization the data is analyzed with MATLAB. There is data from the optimization process which is saved at the end of that process and flux data from the optimized filter configuration. That flux data is calculated again with XSDRNPM for the final optimum filter configuration. This data contains fluxes for all the spatial intervals. Macroscopic cross sections are taken from the CSASI calculation and used to calculate interaction rates. All the data analysis in this thesis is done with MATLAB.

Chapter 4

Filter optimization on neutron flux spectrum

4.1 Introduction

In this chapter is first described which materials are used and how these are implemented in CSAS. After that the geometry and other parameters for the XSDRNPM calculation are explained. Next the goal function, boundary conditions and other optimization parameters are treated. This section ends with the results of the specific optimization process.

4.2 Materials and CSAS calculation

In CSAS the macroscopic cross sections of all materials are calculated. As mentioned in section 2.4.1, a light water reactor like HOR is used as neutron source. To implement this, a mix of uranium-238 (97%), uranium-235(3%) and water is used as the first material. This mix forms the neutron source. The filter is made of materials which are mentioned in literature. Section 2.4.4 give a short overview of them. After the source layer, bismuth, iron, aluminum and aluminum fluoride are implemented as moderating materials, with each having their own specific moderating properties. After the moderation, the thermal neutrons have to be filtered out. This is done by layers of cadmium and boron-10, which are both thermal neutron absorbers. Finally a layer of lead is added to shield the gamma flux beneath the boundary limit. The plots of the scattering and absorption cross sections of the used materials are displayed in appendix A. Figure 4.1 gives schematic view of the filter.

The nuclide density is calculated from the density at room temperature and the atomic mass of the material. The unit of the nuclide density is atoms/(barn cm). The following formula is used for calculating the right nuclide density:

$$\rho_a = \frac{\rho_m}{M} \cdot N_A \cdot 10^{-24} \quad (4.1)$$

In which ρ_a is the atom density in atoms/(barn cm), ρ_m is the mass density of the material in g cm⁻³, M is mass in grams per mole and N_A is the number of Avogadro, the amount of atoms

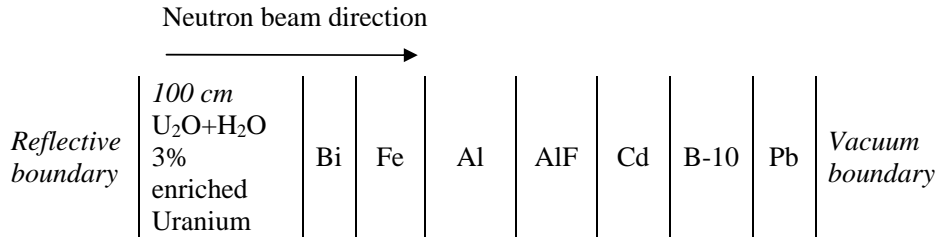


Figure 4.1: Schematic view of the neutron filter. The thickness of the material layers change during the optimization process.

per mole. The 10^{-24} is necessary to get the barns, used for σ . In the case of a material with two or more different elements, formula 4.1 is used to calculate the molecule density. For each atom this is multiplied by the number of that atom in the molecule.

4.3 Spectrum calculation with XSDRN

The neutron flux calculation is done by XSDRN. The macroscopic cross sections are loaded from the file, created in CSAS. For the flux calculation, a slab geometry is used, because a filter consist of multiple layers after each other. The numbers of materials and spatial intervals are determined in the code. In this case there are 8 materials. 64 spatial intervals are chosen: 8 for each zone with a material. It is tested if this are enough intervals by increasing the number of spatial intervals and compare the results with each other. This showed that 8 intervals per zone is enough. The left boundary is a reflective boundary. For the right boundary, at the end of the filter, a vacuum boundary is taken because the neutrons go out of the filter at that side. Figure 4.1 shows the filter geometry. XSDRN calculates one dimensional neutron transport, so only in one direction the materials and the flux variate.

In the input file, each of the zones is linked to a material. Further, the layer thicknesses are entered. In this case the first 100 cm is the mixture of uranium and water. The beginning and end of each layer is specified in the code. These positions are inserted by the MATLAB script and change during the optimization process. The starting thickness of each of the materials is displayed in the script shown in the appendix B.

4.4 Optimization parameters

4.4.1 Goal function

One of the most important things for optimization is defining a good goal function. As mentioned in the introduction 1.2.3, epithermal neutrons are desirable for BNCT. Fast and thermal neutrons are not desirable because these gives a non-selective dose. Therefore a possible goal function is:

$$G = \frac{\int_{1\text{eV}}^{10\text{keV}} \phi_n(E) dE}{\int_0^{\infty} \phi_n(E) dE} \quad (4.2)$$

In which $\phi_n(E)$ is the neutron flux as function of energy in the last spatial interval of the filter. The integral will be a sum over the energy groups. This goal function is equal to the fraction of the epithermal neutron flux from the total flux in the last spatial interval of the filter. If this goal function is maximized, the epithermal flux will be highest and the fast and thermal neutron fluxes will be low. However, this goal function gives the opportunity to let the absolute epithermal flux be very low, which is not desirable. It is possible to formulate a minimum flux as boundary condition to avoid that. Another possibility, which is used here, is to put a term in the goal function that makes a higher flux more desirable. In that case, the goal function become:

$$G = \frac{\int_{1\text{eV}}^{10\text{keV}} \phi_n(E) dE}{\int_0^\infty \phi_n(E) dE + c \cdot \int_{1\text{eV}}^{10\text{keV}} \phi_{n,b}(E) dE} \quad (4.3)$$

In this function $\phi_{n,b}(E)$ is the neutron flux as a function of energy just after the neutron source, the layer with uranium and water. c is a constant in the order of one hundredth. The advantage of this implementation is that the optimization path can go through an area with a lower neutron flux than desired. If a boundary condition is used, the optimization can get stuck when that value is reached. Every time that the boundary is exceeded, the optimization step is not accepted. That give the optimization script less freedom to find the optimum. And also above a given boundary, more neutron flux is favorable, even though the ratio will become a little bit worse.

4.4.2 Boundary conditions

As mentioned in section 1.2.3 the gamma radiation should be minimized. Therefore a boundary condition for the total gamma flux is implemented in the optimization. The gamma flux is calculated as the sum of the gamma flux over all the gamma energy groups. This maximum limit is set to $2.5 \cdot 10^{-5} \frac{\text{photons}}{\text{cm}^2 \cdot \text{s} \cdot \text{source neutron}}$.

4.4.3 Optimization settings

For the optimization some more parameters must be implemented. For the simulated annealing the temperature and temperature decrement coefficient α must be set. Their role in the optimization process is described in section 2.5.1. For the starting temperature the value 0.01 is chosen, this gives at the beginning a probability of about 0.8 to accept a worse solution. The coefficient α is chosen to be 0.99. The number of optimization steps and the constant β which is used for varying the solution (see section 3.4.2) are chosen to be 1000 and 0.08 respectively. The constant c in the goal function is chosen to be 0.02.

4.5 Optimization results

The optimization is executed and gives the following results for the goal function during optimization:

In the first few hundred optimization steps, the goal function grows fast. As expected from the simulated annealing method, also decreases in goal function take place in this part of the

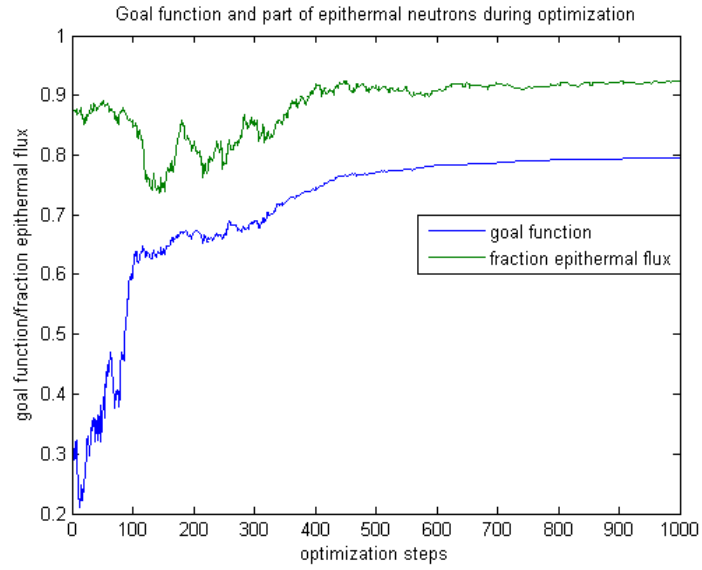


Figure 4.2: Goal function (equation 4.3) and fraction of epithermal neutron flux (equation 4.2) as function of optimization steps.

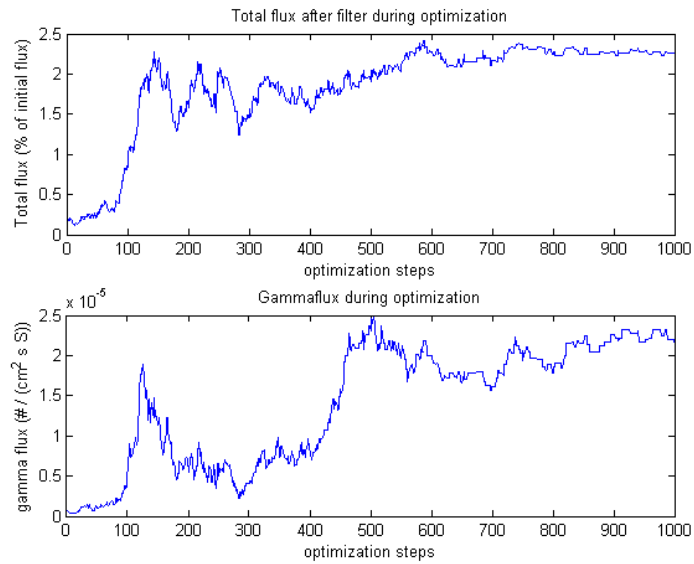


Figure 4.3: Final beam intensity as percentage of initial beam and the gamma flux as function of the optimization steps. In the unit of the gamma flux $\# = \text{photons}$ and $S = \text{fission source neutron}$.

optimization process, where is searched for the area with a global optimum. After 350 steps, the goal function smoothly increase and it ends up at 0.80. The final neutron flux consist of 92% epithermal neutron flux. The fraction of epithermal neutrons doesn't follow the goal function in the first 150 steps. In that part, the absolute flux increases while the fraction of epithermal flux decreases. However, at the end of the optimization, the ratio is highest and the absolute epithermal flux is large too. Figure 4.3 shows that at the end of the optimization the neutron flux is 2.2% of the flux just before the filter. The second goal function with an extra term in the denominator is successful: both the fraction of epithermal flux as the intensity are high and the optimization didn't stuck on a intensity boundary condition. The gamma flux only once reach the upper limit around 500 optimization steps. For the rest the gamma flux remains below the limit. The layer thickness of the materials in the filter evolved as shown in figure 4.4. It

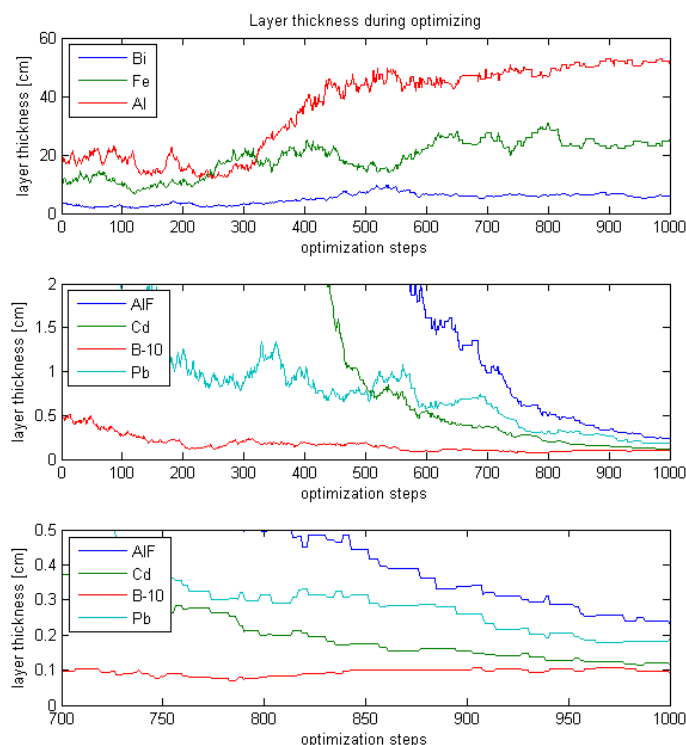


Figure 4.4: Thickness of the material layers in the filter as function of the optimization steps. The upper plot shows the layer thickness of Bi, Fe and Al. The other two plots shows the layer thickness of AlF, Cd, B-10 and Pb. The bottom plot is a magnification of the last part of the middle plot.

takes more optimization steps to let the layer thickness converge. The thick layers (Bi, Fe and Al) seem constant after 600 steps. The thinner materials become constant after 850 steps, but even then the relative change is quite large for AlF and Pb. One reason is that there are some materials who can replace each other because their properties are very similar. Some of those materials have a large macroscopic cross section and therefore a thin layer is sufficient. Some other materials, like AlF have less influence on the filter and if that layer becomes thin, it can be omitted.

It took 36 hours to execute this total optimization on the cluster.

4.6 Spectrum analysis

Figure 4.5 shows the neutron flux spectrum between each material in the optimized filter. The final neutron flux (after Pb) is quite uniformly distributed in the epithermal area. The thermal neutrons are filtered out quite well. In the fast neutron range, there is a narrow peak at 25 keV, but after that peak the fast neutron flux is low. The peak at 25 keV is caused by the moderation effect of the layers of Bi and Fe and the dip in the cross section of Fe for that energy.

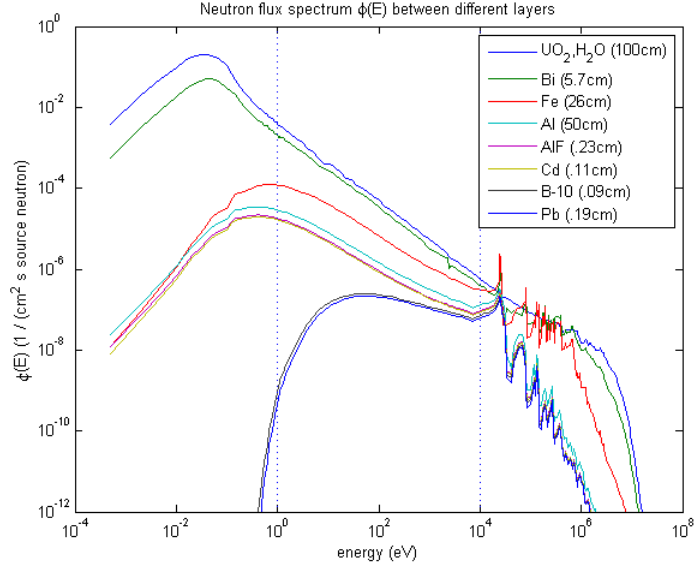


Figure 4.5: Neutron flux spectrum in the filter between each material layer. The legend gives the material behind which the flux is plotted and its layer thickness.

Figure 4.5 shows the influence of each material layer on the spectrum. The layer of iron strongly reduces the thermal neutron flux and moderates the very fast neutrons ($E > 10^6$ eV). Aluminum reduces the neutron flux in the whole fast neutron range. The thin layer of B-10 strongly reduces the thermal neutron flux. Some of the materials, especially cadmium have only very few influence on the spectrum.

4.7 Conclusions and discussion

From this section it becomes clear that mathematical optimization of a neutron filter is possible. A filter of different materials can be used to optimize a neutron flux spectrum. Behind the filter the spectrum contains 92% epithermal neutrons. This percentage is calculated as:

$$\frac{\sum_{\text{ep.gr.}} \phi_n(E)}{\sum_{\text{all gr.}} \phi_n(E)} \cdot 100\% \quad (4.4)$$

With in the numerator the sum of the fluxes over the epithermal energy groups and in the denominator the sum of the neutron fluxes over all energy groups. In the same way the part of the fast and thermal neutron flux is calculated. These are both 4%. This fraction of epithermal neutrons is more than Azahra[10] achieved. In the source spectrum, this fraction is 18%. The

total neutron flux is 2.2% of the flux before the filter. If this is high enough depends on the neutron source and the requirements for the beam. Depending on these requirements, the constant c can be changed, to get more or less flux.

In this optimization, there is not paid that much attention to choose a realistic gamma maximum limit. The chosen limit is only once reached and did not have much influence on the optimization. Another part to discuss is the difference between neutron flux and neutron current. For this calculation, the neutron flux spectrum is considered, but for a neutron beam the direction of the neutrons is also important. Lee[21] mentions the current-to-flux ratio as an beam quality parameter. From this calculation nothing can be said about this parameter.

Chapter 5

Filter optimization on boron-dose ratio between tumor and tissue

5.1 Introduction

This section describes the optimization of the boron-dose ratio between tumor and healthy tissue. As in the previous section, first the CSAS and XSDRN implementation is described. After that the optimization parameters are given and the section ends with the results of this optimization.

5.2 Materials and CSAS calculation

For the source and filter the same materials are used as in section 4.2. Behind the filter, a layer of skin, bone, tissue, tumor and tissue is modeled to imitate a brain tumor. The nuclei and their density are taken from table 4.1 of Verbeke[11]. In page 23 of [11], the concentrations of ^{10}B in tumor and tissue are given to be respectively $45.5 \mu\text{g/g}$ and $13 \mu\text{g/g}$. These values are used for the composition of skin, crane, tissue and tumor. To get the right units, equation 4.1 is used, in which ρ_m is the density of the tissue multiplied by the mass fraction of the nuclide. In the script *csas_calc.m*, shown in appendix B, the CSASI input is given. Figure shows 5.1 a schematic view of the filter with skin, crane, tissue and tumor.

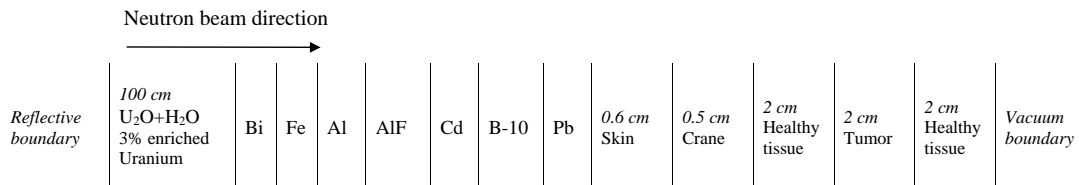


Figure 5.1: Schematic view of the neutron filter with an example of skin, crane, tissue and tumor. The thickness of the filter material layers change during the optimization process.

5.3 Spectrum calculation in XSDRN

For this dose optimization, a slab geometry is used as shown in figure 5.1. There are five more zones than in 4.3: skin, crane, tissue, tumor and tissue after the tumor. The skin and crane are 6 mm and 5 mm thick. The tissue before the tumor, the tumor itself and the tissue after the tumor are all 2 cm thick. The thickness of the filter material layers are changed during the optimization process. The starting thickness is displayed in the script *rr_optim.m*, shown in appendix B.

5.4 Optimization parameters

5.4.1 Goal function

For efficient BNCT treatment, the tumor dose must be as high as possible with the dose in the tissue around the tumor as little as possible. When ^{10}B is put into the tissue, the main fraction of the dose is from the (n, α) reactions. Therefore a possible goal function is the average reaction rate in the tumor divided by the average reaction rate in the surrounding tissue. That gives:

$$G = \frac{\int_{x_2}^{x_3} dx \int_0^\infty \Sigma_{(n,\alpha)} \phi_n(E, x) dE}{\frac{1}{2} \left(\int_{x_1}^{x_2} dx \int_0^\infty \Sigma_{(n,\alpha)} \phi_n(E, x) dE + \int_{x_3}^{x_4} dx \int_0^\infty \Sigma_{(n,\alpha)} \phi_n(E, x) dE \right)} \quad (5.1)$$

Here the tumor is between x_2 and x_3 and the healthy tissue between x_1 and x_2 and between x_3 and x_4 . The factor $\frac{1}{2}$ is to compensate for the double volume of the healthy tissue. The $\Sigma_{(n,\alpha)}$ is the (n, α) cross section, delivered by the ^{10}B . $\Sigma_{(n,\alpha)}$ is taken from CSASI calculation. If this goal function is maximized, the tumor-tissue Boron dose ratio is maximized. However, not only this ratio is important, but also the absolute dose. If that becomes too small, the irradiation time becomes too large. Therefore, as in section 4.4.1, some extra term is put in the denominator of the goal function. Then it becomes:

$$G = \frac{\int_{x_2}^{x_3} dx \int_0^\infty \Sigma_{(n,\alpha)} \phi_n(E, x) dE}{\frac{1}{2} \left(\int_{x_1}^{x_2} dx \int_0^\infty \Sigma_{(n,\alpha)} \phi_n(E, x) dE + \int_{x_3}^{x_4} dx \int_0^\infty \Sigma_{(n,\alpha)} \phi_n(E, x) dE \right) + R_0} \quad (5.2)$$

The constant R_0 is about one tenth to one fifth of the reaction rate in the healthy tissue. Then, the goal function is one fifth or one tenth less than the reaction rates. This will be sufficient large to get a high intensity and not too large to lose too much reaction rate ratio. The goal function will follow quite well the reaction rate ratio, but even the intensity will be high.

5.4.2 Boundary conditions

As with the spectrum optimization in the previous chapter, there is a limit for the total gamma flux at the end of the filter, just before the skin. This boundary limit is set to $5 \cdot 10^{-5} \frac{\text{photons}}{\text{cm}^2 \cdot \text{s} \cdot \text{source neutron}}$. An other important dose contribution is delivered by the fast neutrons,

which are partly moderated to thermal neutrons near the deep tumor. In this way they contribute to the reaction rate in the tumor. The the goal function will not suppress them, because the fast neutron dose, which occur especially near the skin, is not taken into account. This fast neutron dose is undesirable, so an upper limit of the fast neutron flux must be implemented as well. This boundary limit is set to $1 \cdot 10^{-3} \frac{1}{\text{cm}^2 \cdot \text{s} \cdot \text{source neutron}}$.

5.4.3 Optimization settings

Other parameters for the optimization must be set as well. The starting temperature and the temperature decrement constant α are set to 0.2 and 0.99 respectively. Further the solution varying constant β is set to 0.7 and the number of optimization steps to 1000. The constant R_0 in the goal function is set to $2 \cdot 10^{-6} \frac{1}{\text{cm}^2 \cdot \text{s} \cdot \text{source neutron}}$.

5.5 Optimization results

The optimization is executed and gives results for the goal function and reaction rate ratio as shown in figure 5.2. The ratio between the reaction rates in the tumor and the tissue increases

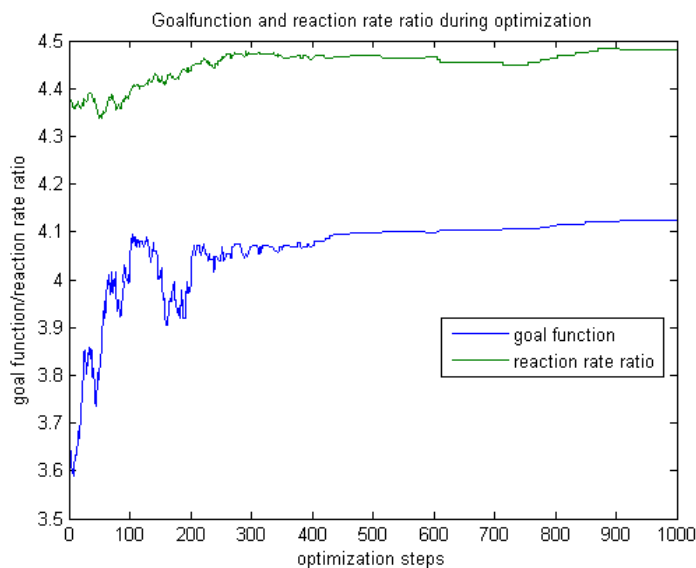


Figure 5.2: Goal function and the reaction rate ratio in tumor and tissue during the optimization process.

only from 4.38 at the beginning of the optimization to 4.48 at the end. The goal function increases from 3.7 to 4.1. The absolute reaction rate has thus increased during the optimization process. The increase of the intensity, which is shown in figure 5.3, supports that. During the first 250 optimization steps, the goal function fluctuates a lot, but after 250 steps it increases smoothly to the optimum. This was expected because of the simulated annealing method.

Figure 5.3 shows the intensity after the filter, the total gamma flux and the fast neutron flux during the optimization process. The neutron flux intensity, gamma flux and fast neutron flux

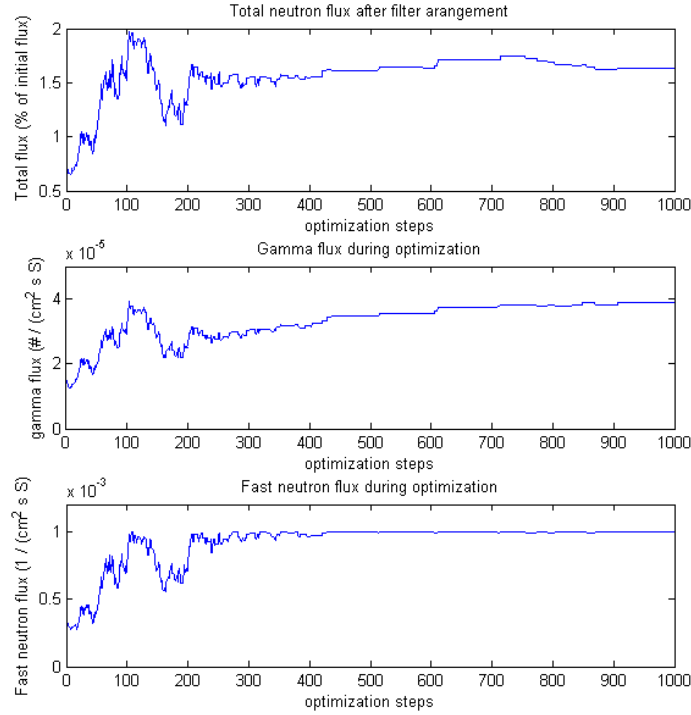


Figure 5.3: Beam intensity after the filter as percentage of initial intensity before the filter, the total gamma flux and the fast neutron flux during optimization. In the unit of the gamma and fast neutron flux $\#$ =photons and S =fission source neutron.

are strongly related to each other. They increase and decrease simultaneously. During the first 300 steps the neutron flux intensity, gamma flux and fast neutron flux fluctuates a lot. After 300 iterations they remain roughly constant (intensity, fast neutron flux) or increase smoothly (gamma flux). Increase of the gamma flux is not desirable, however it did not exceed the maximum limit. Note that the maximum limit for fast neutron flux at $1 \cdot 10^{-3} \frac{1}{\text{cm}^2 \cdot \text{s} \cdot \text{source neutron}}$ is reached. This could be a reason that the intensity and gamma flux not increase any further afterwards.

For more than 300 steps, the layer thicknesses (figure 5.4) of the materials are quite constant. Only bismuth thickness increases. After 400 steps, there are intervals up to 100 successive steps in which material layers do not change. This is related to the fast neutron maximum limit. When that limit is reached, many new solutions, with increasing goal function, let the fast neutron flux exceed that maximum limit. The temperature is low after 400 steps, so only the few changes with increasing goal function and not exceeding fast neutron flux can be accepted. This results in the long intervals without change. The layer of B-10 is thin because of the large absorption cross section of this material. The layer of lead becomes also very thin. This is related to the fact that the gamma flux limit is not reached. Therefore additional shielding is not needed to not exceed that boundary.

It took 50 hours to execute this optimization on the cluster.

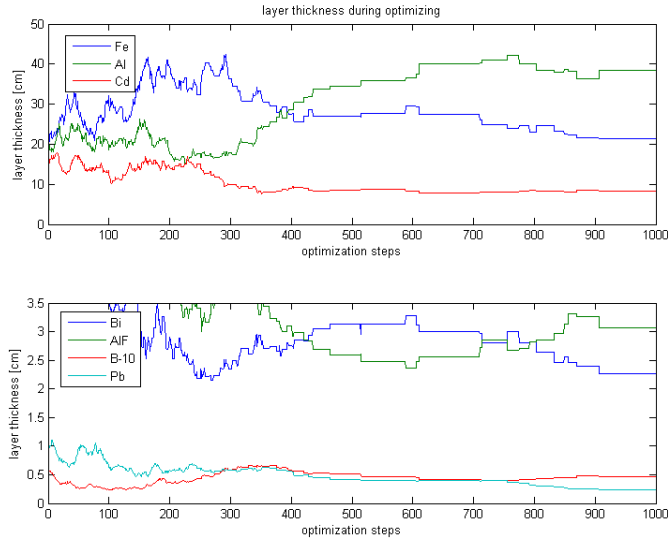


Figure 5.4: Material layer thickness during the optimization process.

5.6 Spectrum and reaction rate analysis

The final configuration of the filter gives the neutron flux spectra as shown in figure 5.5. The fast neutron flux has decreased a lot. The thermal neutron flux is large at the end of the filter compared to in the filter. After the layer of AlF (shown in figure 5.1), there are two thermal neutron absorbers: cadmium and boron-10. This increase of thermal neutrons is caused by layers of skin, crane and tissue. These layers contain many light nuclei which are good moderators. Some of the strongly moderated neutrons are reflected to the last layers of the filter and cause there a high thermal flux. In the tumor and tissue, the thermal neutron flux is very high while the epithermal and fast neutron fluxes are low. The high thermal flux in the tumor is beneficial for BNCT, because the capture cross section of ^{10}B is highest for thermal neutrons.

Figure 5.6 shows the boron-neutron capture reaction rate in the tissue before, in and behind the tumor. At the boundary between the tumor and the tissue, the ratio in reaction rate is around 3.5 because that is the difference in ^{10}B concentration. From the front to the back of the tumor, the reaction rate decreases about 30%. The ratio between the lowest reaction rate in the tumor and the highest in the healthy tissue is 2.4. Behind the tumor, the reaction rate is much lower than before.

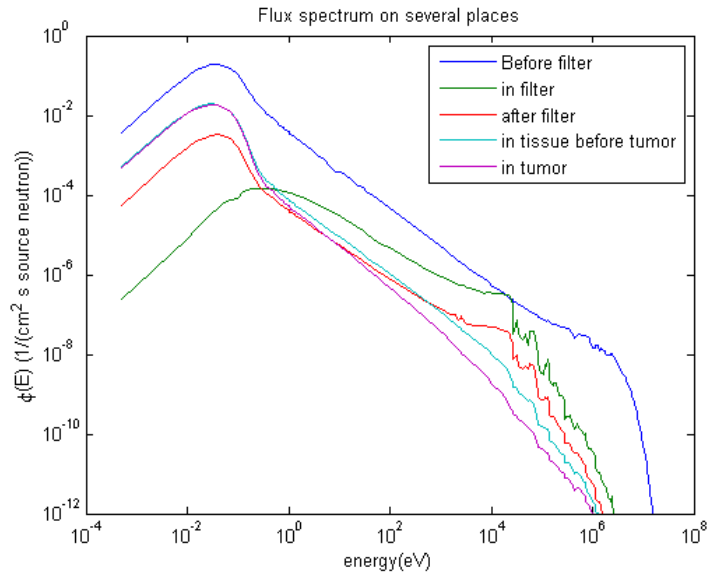


Figure 5.5: Neutron flux spectrum in different places. The position in the filter is after the layer of AlF (shown in figure 5.1). The flux in the tissue before the tumor and the flux in the tumor are average values over these regions.

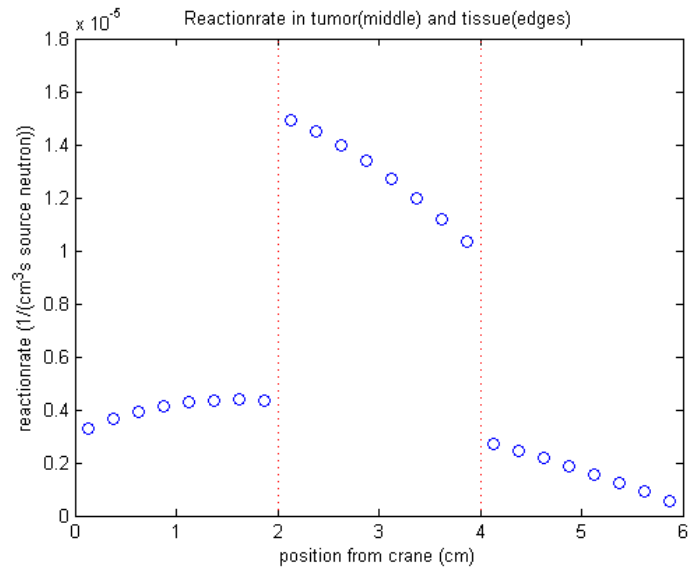


Figure 5.6: The boron-neutron capture reaction rate before, in and behind the tumor as a function of distance in the tissue. The tumor tissue is located between the dotted red lines.

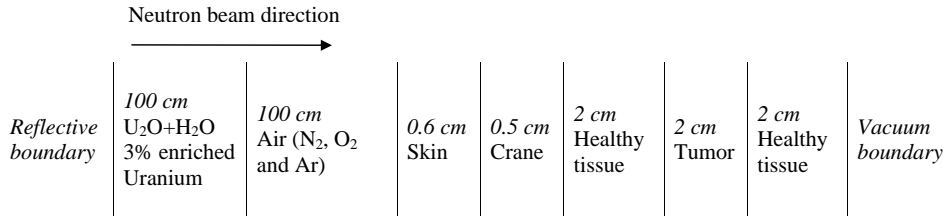


Figure 5.7: Schematic view of situation with a layer of air instead of a filter.

5.7 Reaction rate without filter

With the results of the optimization on reaction rate in the tissue and tumor, it is interesting to compare these results with a situation without filter. For this comparison, a spectrum and reaction rate calculation is done with a layer of air instead of the filter arrangement. For this layer nitrogen, oxygen and argon are taken, which together form more than 99.9% of air. This layer of air is set to 100 cm. The other parameters are the same as in the optimization. The air has 8 spatial intervals in XSDRN. Figure 5.7 gives a schematic view of this situation. In the section *MATLAB code for the calculation without filter* in appendix B, the scripts for this calculation are shown.

Figure 5.8 shows the boron-neutron reaction rate distribution in the tissue and the tumor. This distribution is only a little bit worse than the distribution in figure 5.6. The ratio of lowest

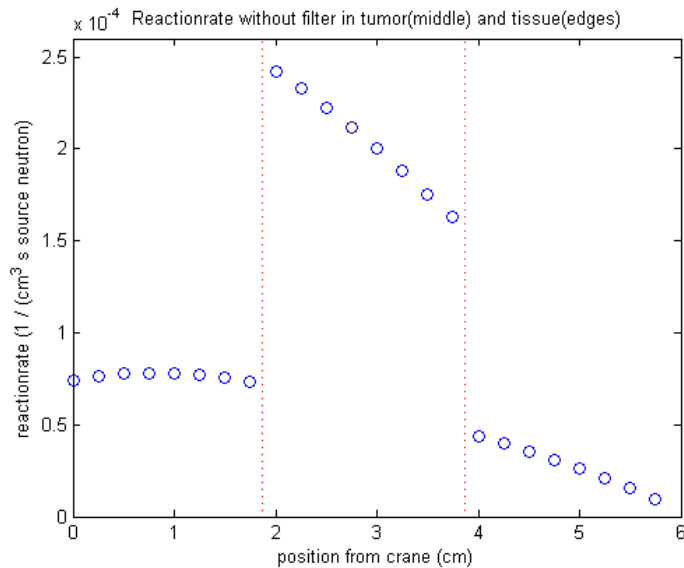


Figure 5.8: The boron-neutron capture reaction rate before, in and behind the tumor as a function of the distance in the tissue for the situation without neutron filter. Tumor tissue is located between the dotted red lines.

reaction rate in the tumor and highest in the healthy tissue is 2.1, a little bit less than the 2.4 with the filter. The integrated ratio, calculated according to equation 5.1 is 3.93, in comparison with the 4.48 for the optimized filter. So, the filter improves this ratio with 14%.

However, the fast neutron flux is also suppressed in the optimization. Figure 5.9 shows the neutron flux spectra for the situations with filter and without filter. In the last case, the fast

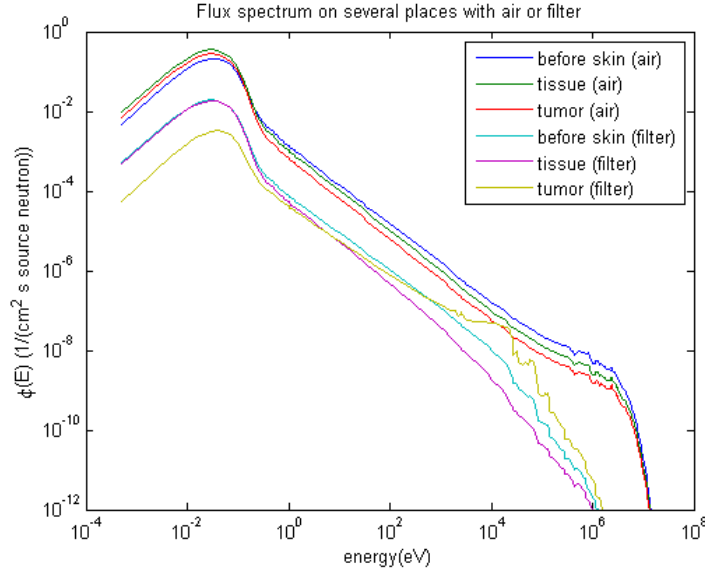


Figure 5.9: Neutron flux spectrum before the skin, in the tissue before the tumor and in the tumor, for the situation with 100 cm of air instead of a filter and with the filter. The flux spectra in the tissue and tumor are zone averaged.

neutron flux is very high, which causes a high fast neutron dose to the skin and further in the healthy tissue. Also the thermal neutron flux at the skin is much higher without filter. These neutrons damage the skin, but don't reach the tumor, therefore they are unwanted. The filter increases the ratio in reaction rate between tumor and tissue and decreases the fast neutron flux at the skin. This last effect is maybe more important, because the dose difference from reaction rate is mainly the result of the difference in ^{10}B concentration. If that difference is used for selective irradiation, the dose from other effects must be significantly lower.

5.8 Reaction rate optimization without fast flux limit

To reduce the unwanted fast neutron flux, a boundary condition is implemented. This boundary condition limits the fast neutron flux to an absolute maximum. Figure 5.3 shows a high correlation between the fast neutron flux and the total flux. Therefore, it is possible that an absolute fast neutron flux limit only decreases the total neutron flux, while the relative amount of fast neutrons does not decrease. To check this, an other optimization without that boundary condition is executed. For the rest the optimization is exactly the same as with fast neutron flux limit.

As expected, this optimization results in a reaction rate ratio which is higher than with boundary limit for fast neutrons. The ratio becomes 4.64 instead of 4.48. Figure 5.10 shows that the fast neutron flux becomes twice as large, while the total neutron flux increases with 35%. So the fraction of fast neutrons in the final flux increase with 50% if no boundary condition is implemented. Figure 5.11 shows the compositions of the beam with and without fast neutron maximum limit. With fast neutron limit, the fraction of fast neutrons in the final flux spectrum

becomes 43%, while without filter it becomes 63%. So, the fast neutron flux limit reduces the fast neutron flux with one third.

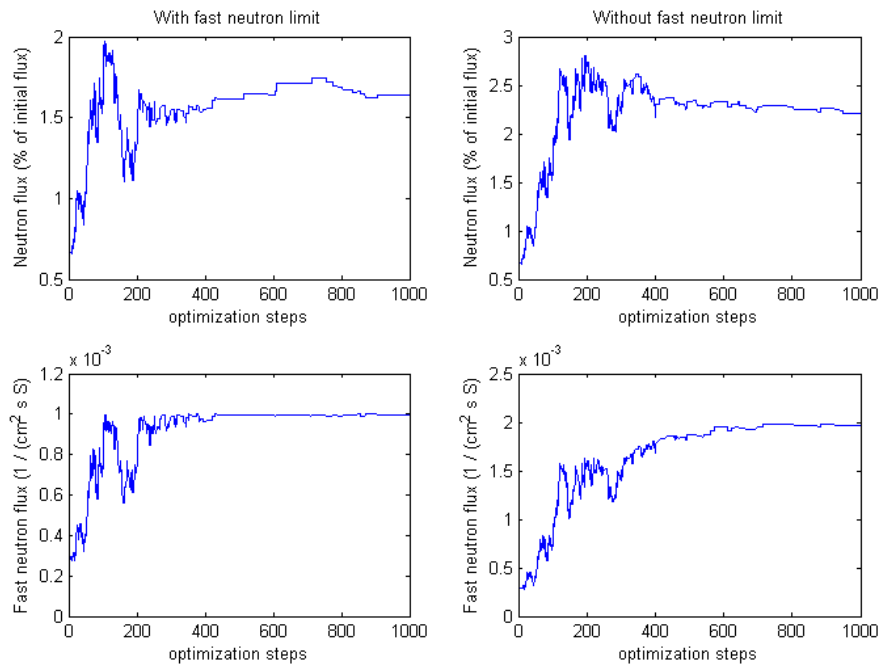


Figure 5.10: Relative flux intensity and fast neutron intensity with and without boundary condition for fast neutron flux. In the unit of the fast neutron flux S =fission source neutron.

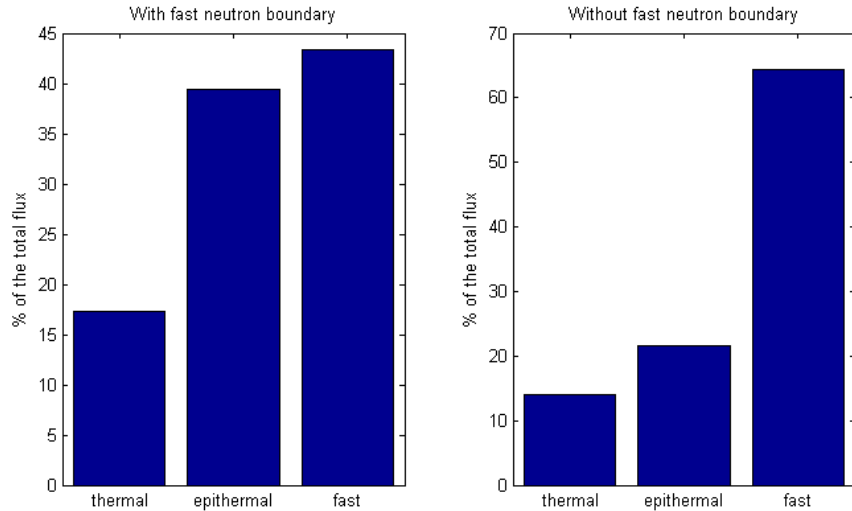


Figure 5.11: Percentage of thermal, epithermal and fast neutron flux at the end of the filter with and without boundary condition for the fast neutron flux.

5.9 Conclusions and discussion

Mathematical optimization on the reaction rate ratio between tissue and tumor is possible. The optimized filter gives a better ratio than the same beam with air instead of a filter. However, other dose effects than boron dose are not yet taken into account. Therefore it is still unclear whether this optimization results in the best dose distribution. BNCT is only successful if other dose contributions are significantly lower than the boron dose. The ratio in boron dose is mainly result of the difference in ^{10}B concentration. Therefore the neutron beam optimization can better concentrate on minimizing other dose contributions while retaining the neutrons which contribute to the boron dose.

The fraction of fast neutrons in the neutron flux behind the filter is reduced with one third by implementing a maximum limit for the fast neutron flux. However the fraction of fast neutrons in the final beam becomes with 43% much larger than the 4% which was achieved with the optimization on the neutron flux spectrum in the previous chapter. Also the absolute flux after the filter decreased with 35% because of the limit on fast neutrons.

For the calculations on reaction rate, a tumor and surrounding tissue is modeled as successive layers who only variate in one dimension. However, when a real tumor and surrounding tissue is taken into account, three dimensional effects will become important. Verbeke[11] compares three different neutron beam diameters with each other. The differences between them are significant, especially for deeper tumors. Therefore this one dimensional approximation is not very accurate for the calculation of the dose ratio between tumor and surrounding tissue.

Chapter 6

Conclusions and recommendations

6.1 Introduction

In this thesis is investigated if it is possible to tailor a neutron spectrum by mathematical optimization of the thicknesses of material layers in a filter arrangement. This study is done with deterministic, one dimensional neutron calculations and the simulated annealing optimization algorithm. As neutron source a light water reactor (3% enriched uranium) is modeled. The research contains two parts: an optimization on the neutron flux spectrum whereby the epithermal neutron flux ($1 \text{ eV} < E < 10 \text{ keV}$) is maximized and an optimization with a realistic example of a tumor and healthy tissue, in which the boron dose ratio between the tumor and the healthy tissue is maximized. This chapter gives the conclusions and some recommendations.

6.2 Conclusions

From the first part can be concluded that it is possible to tailor a neutron spectrum by mathematical optimization with a goal function which maximizes the epithermal neutron flux. The amount of epithermal flux in the total neutron flux increases from 18% after the source to 92% after the optimized filter. The fast and thermal neutron fluxes are both 4% of the total neutron flux, while the total neutron flux is 2.2% of the flux before the filter. This is a better result than Azahra[10] who achieved 72% epithermal neutron flux.

From the second part can be concluded that it is possible to optimize a filter configuration by mathematical optimization with a goal function which maximizes the boron-dose ratio between tumor and healthy tissue. With this optimized filter, a 14% higher boron dose ratio is achieved, than with a layer of air instead of the filter. However, the other important dose contributions (fast neutrons and gamma rays) are not enough limited by imposed restrictions. Therefore, it is not clear if the optimized configuration has the best dose distribution. The imposed limit on the fast neutron flux was reached, but at the expense of a low absolute dose rate in the tumor tissue.

6.3 Recommendations

This thesis has some open ends to do further research. Some recommendations are given below.

6.3.1 Calculation method

The first recommendations concern the calculation methods. In this thesis the neutron flux spectrum is considered. For many applications, including BNCT, the direction of the neutrons is also important. From XSDRN it is possible to get angular fluxes, which give the possibility to calculate the neutron current and the current-to-flux ratio. This current-to-flux ratio is an important beam parameter. Therefore, if further research is done to tailor the neutron spectrum, it would be good to pay attention to this. If tissue and tumor are implemented in the optimization, this current-to-flux ratio is less important because the flux distribution in the tissue is already calculated.

If further research is done on dose distribution, much improvement can be made when the calculations are done in three dimensions instead of one dimension. As discussed in part 5.9, the 3D effects are very important for the flux distribution in and around a tumor.

6.3.2 Input data

Other improvements can be made on the input data. For this research, more common materials are used for the filter. With a more thorough study, the materials and their position in the filter can be chosen better. Maybe the number of materials can be reduced to make the filter more simple.

Also optimization on the order of the materials in the filter is possible. For that optimization, the optimization algorithm must be executed for each sequence of materials. Changing the order of materials during the optimization process is not recommended, because that gives a too dramatic change in neutron flux. Such large changes are difficult to deal with for a mathematical optimization algorithm.

In this thesis little attention has been paid to find realistic gamma flux limits. This could be improved in further research. Because the gamma dose with BNCT increases with the irradiation time and the irradiation time decreases with increasing neutron beam intensity, the gamma maximum limit can be made relative to the epithermal neutron flux.

Beside the used nuclear reactor source, there are also accelerator driven neutron sources. These sources can also be considered in further research. They could have better properties than reactor sources.

6.3.3 Goal function

For this thesis, relative simple goal functions are used. The optimization can be improved a lot by choosing a better goal function. Especially in the case with a modeled tumor and surrounding

tissue. For that situation, it would be a good improvement to implement the important dose distributions in the goal function. Input from medical research is needed to get weight factors for these dose contributions and other requirements.

Bibliography

- [1] Chadwick J. *The existence of a neutron*. Proc Roy Soc London 136: 692–708, (1932).
- [2] Chadwick J, Goldhaber M. *Disintegration by slow neutrons*. Nature 135: 65, (1935).
- [3] Locher GL. *Biological effects and therapeutic possibilities of neutrons*. Am J Roentgenol 36: 1–13, (1936).
- [4] Nievaart, Victor Alexander. *Spectral Tailoring for Boron Neutron Capture Theory*. Delft University Press, (2007).
- [5] Barth, Rolf F., et al. *Current status of boron neutron capture therapy of high grade gliomas and recurrent head and neck cancer*. Radiation Oncology 7.1: 1-21, (2012).
- [6] K. W. Burn, L. Casalini, D. Mondini, E. Nava, G. Rosi, R. Tinti, *The Epithermal Neutron Beam for BNCT Under Construction at Tapiro: Physics* Journ. of Phys.: Conference Series 41, 187-194 (2006).
- [7] Y.-W.H. Liu, T. T. Huang, S. H. Jiang, H. M. Liu, *Renovation of Epithermal Neutron Beam for BNCT at THOR*. Appl. Rad. And Is. 61, 1039-1043 (2004).
- [8] G. Tracz, L. Dabkowski, D. Dworak, K. Pytel, U. Woznicka, *The Filter/ Moderator Arrangement-Optimization for the Boron Neutron Capture Therapy*. Rad. Prot. Dos. 110, 827-831 (2004).
- [9] D. Ross, G. Constantine, D.R. Weaver, T.D. Beynon, *Designing an epithermal neutron beam for boron neutron capture therapy for a DIDO type reactor using MCNP* Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 334.2: 596-606, (1993)
- [10] Azahra, M., A. Kamili, and H. Boukhal. *Monte Carlo calculation for the development of a BNCT neutron source (1ev–10keV) using MCNP code*. Cancer/Radiotherapie 12.5: 360-364, (2008)
- [11] Verbeke, Jérôme Maurice. *Development of High-Intensity D-D and D-T Neutron Sources and Neutron Filters for Medical and Industrial Applications*. University of California, Berkeley, (2000).
- [12] Duderstadt, James J. and Hamilton Louis J. *Nuclear Reactor analysis*. John Wiley & Sons, 1976.
- [13] Bleuel, D. L., R. J. Donahue, B. A. Ludewigt, and J. Vujic. *Designing accelerator-based epithermal neutron beams for boron neutron capture therapy*. Medical physics 25: 1725, (1998)

- [14] Sakamoto, S., W. S. Kiger III, and O. K. Harling. *Sensitivity studies of beam directionality, beam size, and neutron spectrum for a fission converter-based epithermal neutron beam for boron neutron capture therapy*. Medical physics 26: 1979, (1999)
- [15] Rahmani, Faezeh, and Majid Shahriari. *Beam shaping assembly optimization of Linac based BNCT and in-phantom depth dose distribution analysis of brain tumors for verification of a beam model*. Annals of Nuclear Energy 38.2: 404-409, (2011)
- [16] Fu, C. Y., and D. M. Hetrick. *Update of ENDF/BV Mod 3 iron: neutron-producing reaction cross sections and energy-angle correlations*. No. ORNL/TM-9964; ENDF-341. Oak Ridge National Lab., TN (USA), 1986.
- [17] Greene, N. M., and L. M. Petrie. *XSDRNPM: A One-Dimensional Discrete-Ordinates Code for Transport Analysis*. Oak Ridge National Laboratory, 2009.
- [18] <http://www.mathworks.nl/products/matlab/>, consulted on 30-5-2013
- [19] Buseti, Franco, *Simulated annealing overview*, <http://163.18.62.64/wisdom/Simulated%20annealing%20overview.pdf>, consulted on 30-5-2013
- [20] Goluoglu, S., Landers, N.F., Petrie, L.M. and Hollenbach, D.F. *CSAS5: Control Module for Enhanced Criticality Safety Analysis Sequences with KENO v.a*. Oak Ridge National Laboratory, 2009.
- [21] Lee, C. L., X-L. Zhou, R. J. Kudchadker, F. Harmon, and Y. D. Harker. *A Monte Carlo dosimetry-based evaluation of the Li (p, n) Be reaction near threshold for accelerator boron neutron capture therapy*. Medical physics 27: 192, (2000)

Appendix A

Cross sections of filter materials and boron-10

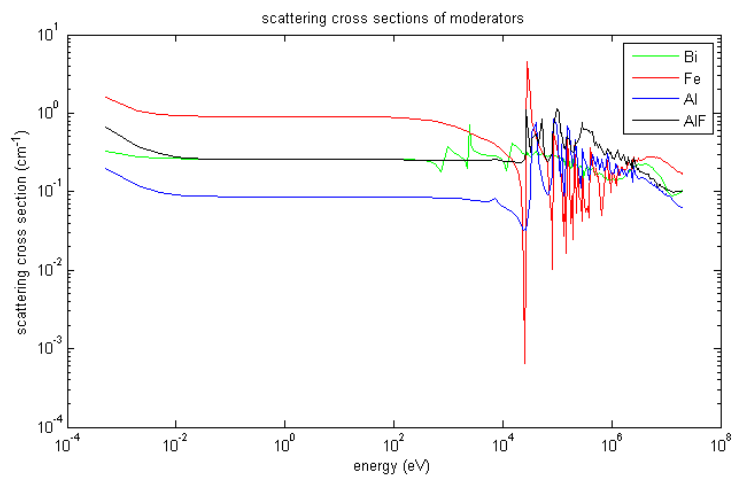


Figure 1: Scattering cross section of the moderator materials of the filter.

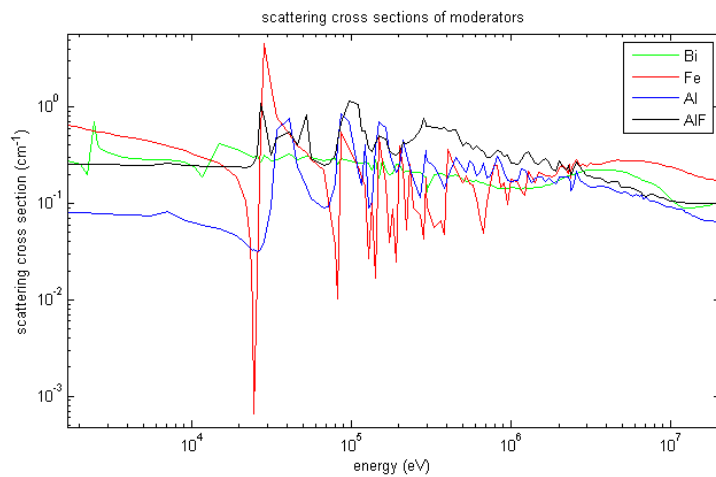


Figure 2: Scattering cross section of the moderator materials of the filter in the high energy area.

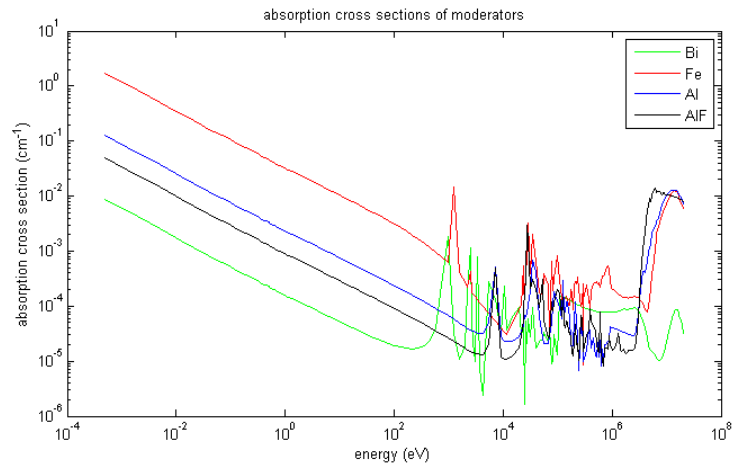


Figure 3: Absorption cross section of the moderator materials of the filter.

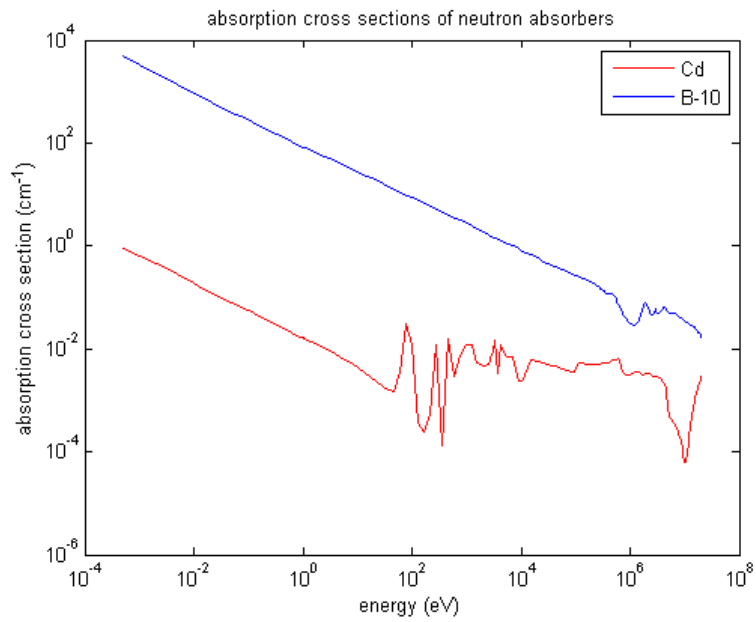


Figure 4: Absorption cross section of the thermal neutron absorption materials of the filter.

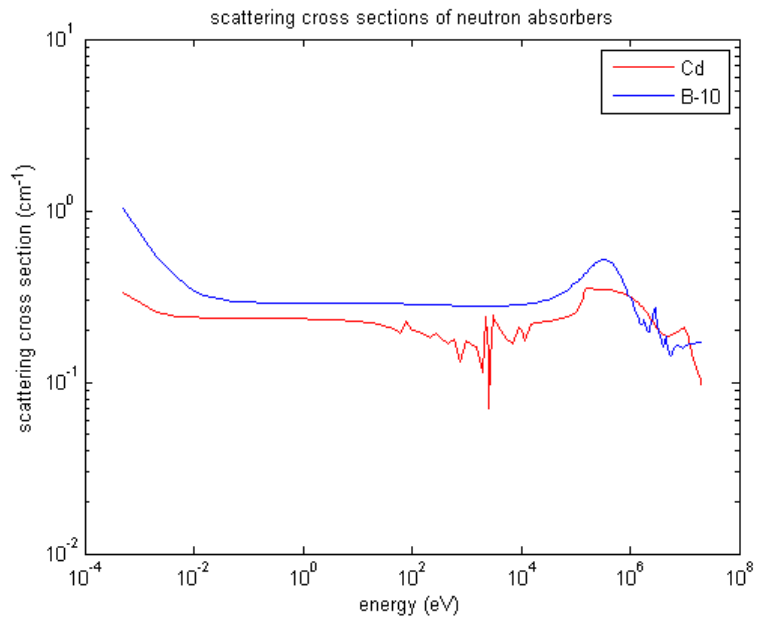


Figure 5: Scattering cross section of the thermal neutron absorption materials of the filter.

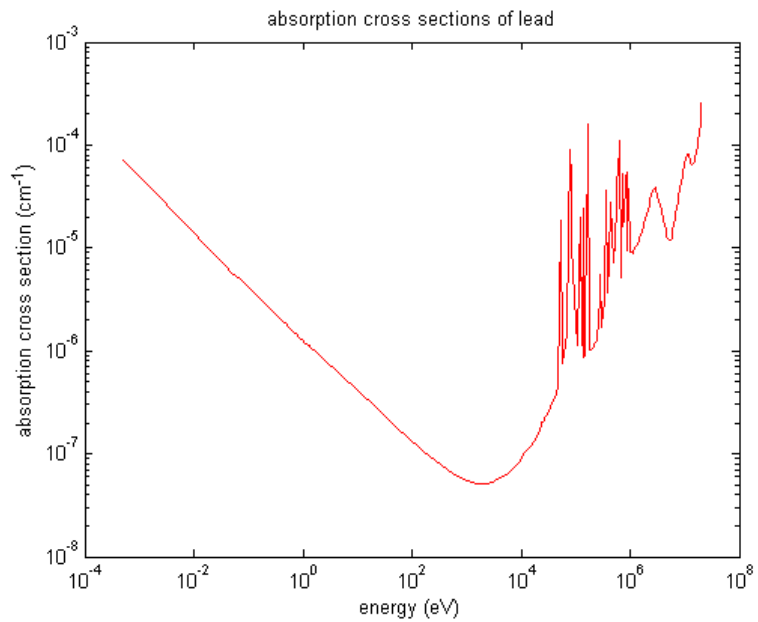


Figure 6: Absorption cross section of lead.

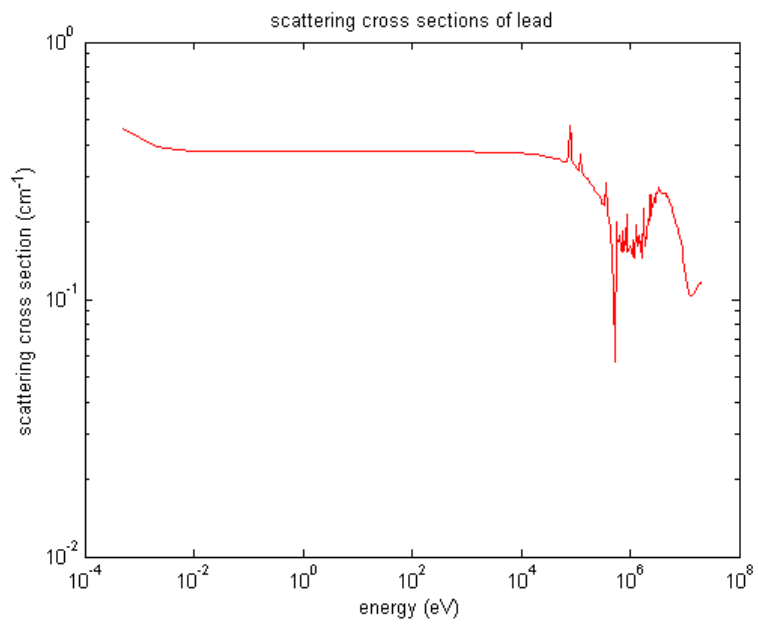


Figure 7: Scattering cross section of lead.

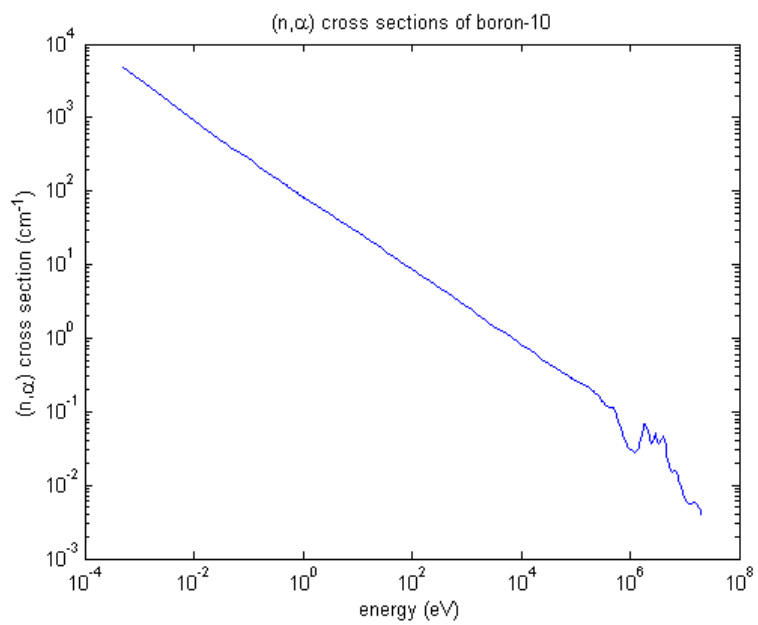


Figure 8: (n,α) cross section of boron-10.

Appendix B

MATLAB code for optimization on neutron flux spectrum

filter_optim.m

```
1 % spectrum optimization
2 no_grps=247;          % no of energy groups
3 n_int=64;            % number of intervals
4 alpha=0.02;
5 E_up=1e4;           % eV upper boundary epithermical area
6 E_low=1;            % eV lower boundary epithermical area
7
8 d(1)=0;             %startpunt
9 d(2)=100;          %cm dikte U+H2O
10 d(3)=3;           %cm Bi
11 d(4)=10;          %cm Fe
12 d(5)=20;          %cm Al
13 d(6)=20;          %cm AlF3
14 d(7)=10;          %cm Cd
15 d(8)=0.5;         %cm Bo-10
16 d(9)=5;           %cm Pb
17
18
19 for i=1:9
20     dist(i)=sum(d(1:i));
21 end
22
23 run('csas_calc2'); %csasi calculation
24 run('basic_calc'); %calculation of energy boundary positions
25 run('xsdrn_calc'); %xsdrn calculation
26 run('calc_flux'); %flux reading and processing
27 run('goal_func'); %calculation of goalfunction
28
29 m=1000;             %number of optimization iterations
30 beta=0.08;          %solution varying constant
31 T=0.1;              %starting temperature
32 alpha_T=0.99;       %temperature decrement
33 gamma_rand=2.5e-5; %gamma boundary condition
34 results=zeros(14,m);
35 results(:,1)=[goal; goal2; part; part2; gamma; zeros(2,1); d(3:9)'];
36
37 for l=2:m
38     % vary solution
39     var=rand(1,7).*2.*beta+(1-beta);
```

```

40     d(3:9)=d(3:9).*var;
41     for j=1:9
42         dist(j)=sum(d(1:j));
43     end
44     run('xsdrn_calc');
45     run('calc_flux');
46     run('goal_func');
47
48     if gamma>gamma_rand
49         d(3:9)=d(3:9)./var;
50         results(:,1)=[results(1:5,1-1);0;2;d(3:9)'];
51     else
52         if goal>results(1,1-1);
53             results(:,1)=[goal; goal2; part; part2; gamma; 1;0; d(3:9)'];
54         else
55             c=exp(-(results(1,1-1)-goal)/T);
56             if c<rand
57                 d(3:9)=d(3:9)./var;
58                 results(:,1)=[results(1:5,1-1);c;1;d(3:9)'];
59             else
60                 results(:,1)=[goal; goal2; part; part2; gamma; c;-1;d(3:9)'];
61             end
62         end
63     end
64     %adjust temperature
65     T=alpha_T*T;
66 end
67 % write optimization results
68 dlmwrite('spectrum_simu_ann2.txt', results);
69
70 % calculate fluxdata of final configuration
71 for j=1:9
72     dist(j)=sum(d(1:j));
73 end
74     run('xsdrn_calc');

```

csas_calc2.m

```

1     %CSASI calculation
2
3     T=300; %Temperature (K)
4
5     fid=fopen('csasi','w');
6
7     fprintf(fid, '%s\n', '=csasi parm=centrm');
8     fprintf(fid, '%s\n', 'JOS XS GENERATION');
9     fprintf(fid, '%s\n', 'V7-200N47G');
10    fprintf(fid, '%s\n', 'read comp');
11    fprintf(fid, '%s %i %s\n', 'u-235      1 0 2.4467e-004', T, 'end');
12    fprintf(fid, '%s %i %s\n', 'u-238      1 0 7.9110e-003', T, 'end');
13    fprintf(fid, '%s %i %s\n', 'h-1       1 0 4.4558e-002', T, 'end');
14    fprintf(fid, '%s %i %s\n', 'o-16      1 0 3.8590e-002', T, 'end');
15    fprintf(fid, '%s %i %s\n', 'bi-209    2 0 2.8182e-002', T, 'end');
16    fprintf(fid, '%s %i %s\n', 'fe-56     3 0 7.5268e-002', T, 'end');
17    fprintf(fid, '%s %i %s\n', 'al-27     4 0 6.0262e-002', T, 'end');
18    fprintf(fid, '%s %i %s\n', 'al-27     5 0 2.0653e-002', T, 'end');
19    fprintf(fid, '%s %i %s\n', 'f-19      5 0 6.1958e-002', T, 'end');

```

```

20 fprintf(fid, '%s %i %s\n', 'cd-112    6 0 4.6339e-002', T, 'end');
21 fprintf(fid, '%s %i %s\n', 'b-10     7 0 1.3258e-001', T, 'end');
22 fprintf(fid, '%s %i %s\n', 'pb-208   8 0 3.2958e-002', T, 'end');
23 fprintf(fid, '%s\n', 'end comp');
24 fprintf(fid, '%s\n', 'read celldata');
25 fprintf(fid, '%s\n', 'infhommedium 1 cellmix=9 end');
26 fprintf(fid, '%s\n', 'infhommedium 2 cellmix=10 end');
27 fprintf(fid, '%s\n', 'infhommedium 3 cellmix=11 end');
28 fprintf(fid, '%s\n', 'infhommedium 4 cellmix=12 end');
29 fprintf(fid, '%s\n', 'infhommedium 5 cellmix=13 end');
30 fprintf(fid, '%s\n', 'infhommedium 6 cellmix=14 end');
31 fprintf(fid, '%s\n', 'infhommedium 7 cellmix=15 end');
32 fprintf(fid, '%s\n', 'infhommedium 8 cellmix=16 end');
33 fprintf(fid, '%s\n', 'end celldata');
34 fprintf(fid, '%s\n', 'end data');
35 fprintf(fid, '%s\n', 'end');
36 fprintf(fid, '%s\n', '=shell');
37 fprintf(fid, '%s\n', 'mv ft02f001 $RTNDIR/jos_xs.out');
38 fprintf(fid, '%s\n', 'end');
39
40 fclose(fid);
41
42 system('batch6 -m csasi');

```

basic_calc.m

```

1 % basic calculations to do once
2
3 load('energy_gr.n.txt');
4 load('energy_gr.g.txt');
5
6 i=1;
7 a=1e6;
8
9 while a > 1e5
10     i=i+1;
11     a=energy_gr.n(i,1);
12     i_max=i;
13 end
14 while a > 1e3
15     i=i+1;
16     a=energy_gr.n(i,1);
17     i_min=i-1;
18 end

```

xsdrrn_calc.m

```

1
2
3 % xsdrn calculation
4
5 n_int2=n_int/8-1;
6 n_int3=n_int/8;

```

```

7
8   fid=fopen('xsdrn','w');
9
10  fprintf(fid, '%s\n', '=shell');
11  fprintf(fid, '%s\n', 'ln -fs $RTNDIR/jos_xs.out ft04f001');
12  fprintf(fid, '%s\n', 'end');
13  fprintf(fid, '%s\n', '=xsdrn');
14  fprintf(fid, '%s\n', '0$$ a3 4 a5 3 a7 17 e');
15  fprintf(fid, '%s %i %i %s %i %i %s\n', '1$$ 1', 8, n_int, '1 0', ...
16  8, 8, '16 5 1 a11 250 250 a14 0 e');
17  fprintf(fid, '%s\n', '3$$ 1 e');
18  fprintf(fid, '%s %i %s\n', '4$$ 0', no_grps, '3 e t');
19  fprintf(fid, '%s\n', '13$$ 1 2 3 4 5 6 7 8');
20  fprintf(fid, '%s\n', '14$$ 9 10 11 12 13 14 15 16');
21  fprintf(fid, '%s\n', '15** f1');
22  fprintf(fid, '%s\n', 't');
23  fprintf(fid, '%s\n', '33## f1 t');
24  fprintf(fid, '%s %i%s %d %i%s %d\n %i%s %d %i%s %d\n %i%s %d\n %i%s %d ...
    %i%s %d\n %i%s %d %d %s\n', ...
25  '35**', n_int2, 'i', dist(1), n_int2, 'i', dist(2), n_int2, 'i', ...
    dist(3), n_int2, 'i', ...
26  dist(4), n_int2, 'i', dist(5), n_int2, 'i', dist(6), n_int2, 'i', ...
    dist(7), n_int2, 'i', ...
27  dist(8), dist(9), 'e');
28  fprintf(fid, '%s %i%s %i%s %i%s %i%s %i%s %i%s %i%s\n', '36$$', ...
    n_int3, 'r1', ...
29  n_int3, 'r2', n_int3, 'r3', n_int3, 'r4', n_int3, 'r5', n_int3, ...
    'r6', n_int3, 'r7', n_int3, 'r8 e');
30  fprintf(fid, '%s\n', '39$$ 1 2 3 4 5 6 7 8 e');
31  fprintf(fid, '%s %i%s %i %s\n', '51$$', no_grps-2, 'i 1', no_grps, 'e 5t');
32  fprintf(fid, '%s\n', 'end');
33
34  fprintf(fid, '%s\n', '=shell');
35  fprintf(fid, '%s\n', 'cp ft17f001 $RTNDIR/flux_scal_xsdrn');
36  fprintf(fid, 'end');
37
38  fclose(fid);
39
40  system('batch6 -m xsdrn');
41  system('sh run3');

```

calc_flux.m

```

1  % load and configurate fluxdata
2  fluxdata = load('fort.23');
3  flux = zeros(n_int, no_grps);
4  for i=1:n_int
5      for gr=1:no_grps
6          flux(i,gr) = fluxdata((i-1)*(no_grps)+gr);
7      end
8  end
9
10 % normalizing fluxdata
11 norm_flux_n=zeros(n_int,200);
12 norm_eflux_n=zeros(n_int,200);
13 for i=1:n_int
14 norm_flux_n(i,:) = flux(i,1:200) ./ energy_gr_n(1:200,2);

```

```

15 norm_eflux_n(i,:)=norm_flux_n(i,:).*energy_gr_n(1:200,1)';
16 end
17 norm_flux_g=zeros(n_int,47);
18 int_g=zeros(n_int,47);
19 for i=1:n_int
20 norm_flux_g(i,:)=flux(i,201:247)'./energy_gr_g(1:47,2);
21 int_g(i,:)=norm_flux_g(i,:)'.*(energy_gr_g(1:47,2)-(energy_gr_g(1:47,2)./2));
22 end

```

goal_func.m

```

1 % Goal function
2
3 e_bound=[energy_gr_n(i_min,1), energy_gr_n(i_max,1)];
4 goal=sum(flux(n_int,i_max:i_min))/(sum(flux(n_int,1:200))+alpha ...
5     *sum(flux(n_int/8,i_max:i_min)));
6 goal2=sum(flux(n_int,i_max:i_min))/sum(flux(n_int,1:200));
7 part=sum(flux(n_int,1:200))/sum(flux(n_int/8,1:200));
8 part2=sum(flux(n_int,i_max:i_min))/sum(flux(n_int/8,1:200));
9
10 gamma=sum(flux(n_int,201:247));

```

Matlab code for optimization on boron dose ratio

rr_optim.m

```

1 % Optimization on reaction rate ratio
2
3 no_grps=247; % number of energy groups
4 n_int=104; % number of spatial intervals
5 R_0=0.2e-5; % constant in goalfunction
6
7 d(1)=0; %starting point
8 d(2)=100; %cm thickness U+H2O
9 d(3)=6; %cm Bi
10 d(4)=20; %cm Fe
11 d(5)=20.5; %cm Al
12 d(6)=10; %cm AlF3
13 d(7)=15; %cm Cd
14 d(8)=0.6; %cm Bo-10
15 d(9)=1; %cm Pb
16 d(10)=0.6; %cm skin
17 d(11)=0.5; %cm bone
18 d(12)=2; %cm brain
19 d(13)=2; %cm tumor
20 d(14)=2; %cm tissue after tumor
21
22 for i=1:14
23     dist(i)=sum(d(1:i));
24 end
25

```



```

26 run('csas_calc'); % macroscopic cross sections with CSASI
27 run('data.load'); % load energy groups and macroscopic n,alpha cross section
28 run('xsdrn_calc'); % flux spectrum calculation with XSDRNPM
29 run('calc_flux'); % read and normalize flux from xsdrn
30 run('rr_goal'); % calculate goal function and data for boundary conditions
31
32 m=1000; % number of optimization steps
33 beta=0.07; % solution variation constant
34 T=0.200; % starting temperature
35 alpha.T=0.99; % temperature decrement constant
36 gamma.rand=5e-5; % gamma rays boundary
37 fast.rand=1e-3; % fast neutrons boundary
38 results=zeros(15,m); % create empty results matrix
39 results(:,1)=[goal; ratio; part; gamma; fast; T; zeros(2,1); d(3:9)'];
40
41 for l=2:m
42 %vary solution
43 var=rand(1,7).*2.*beta+(1-beta);
44 d(3:9)=d(3:9).*var;
45 for j=1:14
46 dist(j)=sum(d(1:j));
47 end
48 % calculate flux and goal to the new solution
49 run('xsdrn_calc'); % flux spectrum calculation with XSDRNPM
50 run('calc_flux');
51 run('rr_goal');
52
53 % verify boundary conditions
54 if gamma>gamma.rand|fast>fast.rand
55 d(3:9)=d(3:9)./var;
56 results(:,l)=[results(1:5,l-1);T ;0;2;d(3:9)'];
57 else
58 % decide if new solution is accepted
59 if goal>results(1,l-1);
60 results(:,l)=[goal; ratio; part; gamma; fast; T; 1;0; d(3:9)'];
61 else
62 c=exp(-(results(1,l-1)-goal)/(T));
63 if c<rand
64 d(3:9)=d(3:9)./var;
65 results(:,l)=[results(1:5,l-1);T;c;1;d(3:9)'];
66 else
67 results(:,l)=[goal; ratio; part; gamma; fast; T; c;-1;d(3:9)'];
68 end
69 end
70 end
71 % adjust temperature
72 T=T*alpha.T;
73 end
74
75 %write optimization results
76 dlmwrite('rr_results.txt',results);
77
78 %calculate fluxdata of final configuration
79 for j=1:14
80 dist(j)=sum(d(1:j));
81 end
82 run('xsdrn_calc');

```

csas_calc.m

```

1      %CSASI calculation
2
3      T=300; %Temperature (K)
4
5      fid=fopen('csasi','w');
6
7      fprintf(fid, '%s\n', '=csasi parm=centrm');
8      fprintf(fid, '%s\n', 'JOS XS GENERATION');
9      fprintf(fid, '%s\n', 'V7-200N47G');
10     fprintf(fid, '%s\n', 'read comp');
11     fprintf(fid, '%s %i %s\n', 'u-235      1 0 2.4467e-004', T, 'end');
12     fprintf(fid, '%s %i %s\n', 'u-238      1 0 7.9110e-003', T, 'end');
13     fprintf(fid, '%s %i %s\n', 'h-1       1 0 4.4558e-002', T, 'end');
14     fprintf(fid, '%s %i %s\n', 'o-16     1 0 3.8590e-002', T, 'end');
15     fprintf(fid, '%s %i %s\n', 'bi-209   2 0 2.8182e-002', T, 'end');
16     fprintf(fid, '%s %i %s\n', 'fe-56    3 0 7.5268e-002', T, 'end');
17     fprintf(fid, '%s %i %s\n', 'al-27    4 0 6.0262e-002', T, 'end');
18     fprintf(fid, '%s %i %s\n', 'al-27    5 0 2.0653e-002', T, 'end');
19     fprintf(fid, '%s %i %s\n', 'f-19     5 0 6.1958e-002', T, 'end');
20     fprintf(fid, '%s %i %s\n', 'cd-112   6 0 4.6339e-002', T, 'end');
21     fprintf(fid, '%s %i %s\n', 'b-10     7 0 1.3258e-001', T, 'end');
22     fprintf(fid, '%s %i %s\n', 'pb-208   8 0 3.2958e-002', T, 'end');
23     fprintf(fid, '%s %i %s\n', 'h-1      9 0 6.6410e-002', T, 'end');
24     fprintf(fid, '%s %i %s\n', 'c        9 0 1.2740e-002', T, 'end');
25     fprintf(fid, '%s %i %s\n', 'n        9 0 1.2375e-003', T, 'end');
26     fprintf(fid, '%s %i %s\n', 'o-16     9 0 2.5365e-002', T, 'end');
27     fprintf(fid, '%s %i %s\n', 'cl-35    9 0 3.8170e-005', T, 'end');
28     fprintf(fid, '%s %i %s\n', 'b-10     9 0 8.3766e-007', T, 'end');
29     fprintf(fid, '%s %i %s\n', 'h-1     10 0 4.7996e-002', T, 'end');
30     fprintf(fid, '%s %i %s\n', 'c       10 0 1.7066e-002', T, 'end');
31     fprintf(fid, '%s %i %s\n', 'n       10 0 2.7620e-003', T, 'end');
32     fprintf(fid, '%s %i %s\n', 'o-16    10 0 2.6288e-002', T, 'end');
33     fprintf(fid, '%s %i %s\n', 'na-23   10 0 4.2172e-005', T, 'end');
34     fprintf(fid, '%s %i %s\n', 'p-31    10 0 2.5290e-003', T, 'end');
35     fprintf(fid, '%s %i %s\n', 'cl-35   10 0 3.8170e-005', T, 'end');
36     fprintf(fid, '%s %i %s\n', 'ca-40   10 0 4.2454e-003', T, 'end');
37     fprintf(fid, '%s %i %s\n', 'mg-24   10 0 7.9765e-005', T, 'end');
38     fprintf(fid, '%s %i %s\n', 's-32    10 0 9.0724e-005', T, 'end');
39     fprintf(fid, '%s %i %s\n', 'b-10    10 0 1.2604e-006', T, 'end');
40     fprintf(fid, '%s %i %s\n', 'h-1     11 0 6.6100e-002', T, 'end');
41     fprintf(fid, '%s %i %s\n', 'c       11 0 7.3230e-003', T, 'end');
42     fprintf(fid, '%s %i %s\n', 'n       11 0 8.2825e-004', T, 'end');
43     fprintf(fid, '%s %i %s\n', 'o-16    11 0 2.8600e-002', T, 'end');
44     fprintf(fid, '%s %i %s\n', 'na-23   11 0 3.8395e-005', T, 'end');
45     fprintf(fid, '%s %i %s\n', 'p-31    11 0 7.9400e-005', T, 'end');
46     fprintf(fid, '%s %i %s\n', 'cl-35   11 0 2.4900e-005', T, 'end');
47     fprintf(fid, '%s %i %s\n', 'k-39    11 0 6.2889e-005', T, 'end');
48     fprintf(fid, '%s %i %s\n', 'b-10    11 0 8.1965e-007', T, 'end');
49     fprintf(fid, '%s %i %s\n', 'h-1     12 0 6.6100e-002', T, 'end');
50     fprintf(fid, '%s %i %s\n', 'c       12 0 7.3230e-003', T, 'end');
51     fprintf(fid, '%s %i %s\n', 'n       12 0 8.2825e-004', T, 'end');
52     fprintf(fid, '%s %i %s\n', 'o-16    12 0 2.8600e-002', T, 'end');
53     fprintf(fid, '%s %i %s\n', 'na-23   12 0 3.8395e-005', T, 'end');
54     fprintf(fid, '%s %i %s\n', 'p-31    12 0 7.9400e-005', T, 'end');
55     fprintf(fid, '%s %i %s\n', 'cl-35   12 0 2.4900e-005', T, 'end');
56     fprintf(fid, '%s %i %s\n', 'k-39    12 0 6.2889e-005', T, 'end');
57     fprintf(fid, '%s %i %s\n', 'b-10    12 0 2.8688e-006', T, 'end');

```

```

58     fprintf(fid, '%s\n', 'end comp');
59     fprintf(fid, '%s\n', 'read celldata');
60     fprintf(fid, '%s\n', 'infhommedium 1 cellmix=13 end');
61     fprintf(fid, '%s\n', 'infhommedium 2 cellmix=14 end');
62     fprintf(fid, '%s\n', 'infhommedium 3 cellmix=15 end');
63     fprintf(fid, '%s\n', 'infhommedium 4 cellmix=16 end');
64     fprintf(fid, '%s\n', 'infhommedium 5 cellmix=17 end');
65     fprintf(fid, '%s\n', 'infhommedium 6 cellmix=18 end');
66     fprintf(fid, '%s\n', 'infhommedium 7 cellmix=19 end');
67     fprintf(fid, '%s\n', 'infhommedium 8 cellmix=20 end');
68     fprintf(fid, '%s\n', 'infhommedium 9 cellmix=21 end');
69     fprintf(fid, '%s\n', 'infhommedium 10 cellmix=22 end');
70     fprintf(fid, '%s\n', 'infhommedium 11 cellmix=23 end');
71     fprintf(fid, '%s\n', 'infhommedium 12 cellmix=24 end');
72     fprintf(fid, '%s\n', 'end celldata');
73     fprintf(fid, '%s\n', 'end data');
74     fprintf(fid, '%s\n', 'end');
75     fprintf(fid, '%s\n', '=shell');
76     fprintf(fid, '%s\n', 'mv ft02f001 $RTNDIR/jos-xs.out');
77     fprintf(fid, '%s\n', 'end');
78
79     fclose(fid);
80
81     system('batch6 -m csasi');

```

data_load.m

```

1  %load data and calculate fast neutron boundary
2  load('energy_gr.n.txt');
3  load('energy_gr.g.txt');
4  load('nalpha_cross.txt')
5
6  i=1;
7  a=1e6;
8  while a > 1e4
9      i=i+1;
10     a=energy_gr.n(i,1);
11     i_max=i;
12 end
13
14 while a > 1
15     i=i+1;
16     a=energy_gr.n(i,1);
17     i_min=i;
18 end

```

xsdrn_calc.m

```

1  % xsdrn calculation
2
3  n_int2=n_int/13-1;
4  n_int3=n_int/13;
5

```

```

6     fid=fopen('xsdrn','w');
7
8     fprintf(fid, '%s\n', '=shell');
9     fprintf(fid, '%s\n', 'ln -fs $RTNDIR/jos_xs.out ft04f001');
10    fprintf(fid, '%s\n', 'end');
11    fprintf(fid, '%s\n', 'xsdrn');
12    fprintf(fid, '%s\n', '0$$ a3 4 a5 3 a7 17 e');
13    fprintf(fid, '%s %i %i %s %i %i %s\n', '1$$ 1', 13, n_int, '1 0', ...
14    13, 13, '16 5 1 a11 250 250 a14 0 e');
15    fprintf(fid, '%s\n', '3$$ 1 e');
16    fprintf(fid, '%s %i %s\n', '4$$ 0', no_grps, '3 e t');
17    fprintf(fid, '%s\n', '13$$ 1 2 3 4 5 6 7 8 9 10 11 12 13');
18    fprintf(fid, '%s\n', '14$$ 13 14 15 16 17 18 19 20 21 22 23 24 23');
19    fprintf(fid, '%s\n', '15** fl');
20    fprintf(fid, '%s\n', 't');
21    fprintf(fid, '%s\n', '33## fl t');
22    fprintf(fid, '%s %i %s %d %i %s %d\n %i %s %d %i %s %d\n %i %s %d %i %s %d\n ...
    %i %s %d %i %s %d\n %i %s %d %i %s %d\n %i %s %d %i %s %d\n %i %s %d %d ...
    %s\n', ...
23    '35**', n_int2, 'i', dist(1), n_int2, 'i', dist(2), n_int2, 'i', ...
24    dist(3), n_int2, 'i', dist(4), n_int2, 'i', ...
25    dist(5), n_int2, 'i', dist(6), n_int2, 'i', ...
26    dist(7), n_int2, 'i', dist(8), n_int2, 'i', ...
27    dist(9), n_int2, 'i', dist(10), n_int2, 'i', ...
28    dist(11), n_int2, 'i', dist(12), n_int2, 'i', ...
29    dist(13), dist(14), 'e');
30    fprintf(fid, '%s %i %s %i %s %i %s %i %s %i %s %i %s %i %s %i %s ...
    %i %s %i %s\n', ...
31    '36$$', n_int3, 'r1', ...
32    n_int3, 'r2', n_int3, 'r3', n_int3, 'r4', n_int3, 'r5', n_int3, ...
    'r6', ...
33    n_int3, 'r7', n_int3, 'r8', n_int3, 'r9', n_int3, 'r10', n_int3, ...
34    'r11', n_int3, 'r12', n_int3, 'r13 e');
35    fprintf(fid, '%s\n', '39$$ 1 2 3 4 5 6 7 8 9 10 11 12 13 e');
36    fprintf(fid, '%s %i %s %i %s\n', '51$$', no_grps-2, 'i 1', no_grps, 'e 5t');
37    fprintf(fid, '%s\n', 'end');
38
39    fprintf(fid, '%s\n', '=shell');
40    fprintf(fid, '%s\n', 'cp ft17f001 $RTNDIR/flux_scal_xsdrn');
41    fprintf(fid, 'end');
42
43    fclose(fid);
44
45    system('batch6 -m xsdrn');
46    system('sh run3');

```

calc_flux.m

```

1 % calculate and normalize fluxdata from XSDRN
2
3 fluxdata = load('fort.21');
4 flux = zeros(n_int, no_grps);
5
6 for i=1:n_int
7     for gr=1:no_grps
8         flux(i,gr) = fluxdata((i-1)*(no_grps)+gr);
9     end

```

```

10 end
11
12 % normalize flux
13 % neutron flux
14 norm_flux_n=zeros(n_int,200);
15 norm_eflux_n=zeros(n_int,200);
16
17 for i=1:n_int
18 norm_flux_n(i,:)=flux(i,1:200)'./energy_gr_n(1:200,2);
19 norm_eflux_n(i,:)=norm_flux_n(i,:).*energy_gr_n(1:200,1)';
20 end
21
22 % gamma flux
23 norm_flux_g=zeros(n_int,47);
24 int_g=zeros(n_int,47);
25
26 for i=1:n_int
27 norm_flux_g(i,:)=flux(i,201:247)'./energy_gr_g(1:47,2);
28 int_g(i,:)=norm_flux_g(i,:)'.*(energy_gr_g(1:47,2)-(energy_gr_g(1:47, 2)./2));
29 end

```

rr_goal.m

```

1 %goal function and other parameters
2
3 ratio=sum(flux(89:96,1:200)*nalpha_cross(:,3))/...
4     (0.5*(sum(flux(81:88,1:200)*nalpha_cross(:,2))+ ...
5         sum(flux(97:104,1:200)*nalpha_cross(:,2))));
6 goal=sum(flux(89:96,1:200)*nalpha_cross(:,3))/...
7     (0.5*(sum(flux(81:88,1:200)*nalpha_cross(:,2))+ ...
8         sum(flux(97:104,1:200)*nalpha_cross(:,2)))+R_0);
9
10 part=sum(flux(n_int*8/13,1:200))/sum(flux(n_int/13,1:200));
11 fast=sum(flux(65,1:i_max));
12
13 gamma=sum(flux(n_int*8/13,201:247));

```

Matlab code for the calculation without filter

without_filter.m

```

1 % reaction rate without filter
2
3 no_grps=247; % number of energy groups
4 n_int=56;    % number of spatial intervals
5
6
7 d(1)=0;      %starting point
8 d(2)=100;   %cm U+H2O
9 d(3)=100;   %cm air
10 d(4)=0.6;   %cm skin
11 d(5)=0.5;   %cm bone

```

```

12 d(6)=2;          %cm brain
13 d(7)=2;          %cm tumor
14 d(8)=2;          %cm tissue after tumor
15
16 for i=1:8
17     dist(i)=sum(d(1:i));
18 end
19
20 run('csas_calc4');
21 run('xsdrn_calc4');

```

csas_calc4.m

```

1     %CSASI calculation
2
3     T=300; %Temperature (K)
4
5     fid=fopen('csasi','w');
6
7     fprintf(fid, '%s\n', '=csasi parm=centrm');
8     fprintf(fid, '%s\n', 'JOS XS GENERATION');
9     fprintf(fid, '%s\n', 'V7-200N47G');
10    fprintf(fid, '%s\n', 'read comp');
11    fprintf(fid, '%s %i %s\n', 'u-235    1 0 2.4467e-004', T, 'end');
12    fprintf(fid, '%s %i %s\n', 'u-238    1 0 7.9110e-003', T, 'end');
13    fprintf(fid, '%s %i %s\n', 'h-1      1 0 4.4558e-002', T, 'end');
14    fprintf(fid, '%s %i %s\n', 'o-16     1 0 3.8590e-002', T, 'end');
15    fprintf(fid, '%s %i %s\n', 'o-16     2 0 1.0122e-005', T, 'end');
16    fprintf(fid, '%s %i %s\n', 'n        2 0 3.7735e-005', T, 'end');
17    fprintf(fid, '%s %i %s\n', 'ar-40    2 0 2.2569e-007', T, 'end');
18    fprintf(fid, '%s %i %s\n', 'h-1      3 0 6.6410e-002', T, 'end');
19    fprintf(fid, '%s %i %s\n', 'c        3 0 1.2740e-002', T, 'end');
20    fprintf(fid, '%s %i %s\n', 'n        3 0 1.2375e-003', T, 'end');
21    fprintf(fid, '%s %i %s\n', 'o-16     3 0 2.5365e-002', T, 'end');
22    fprintf(fid, '%s %i %s\n', 'cl-35    3 0 3.8170e-005', T, 'end');
23    fprintf(fid, '%s %i %s\n', 'b-10     3 0 8.3766e-007', T, 'end');
24    fprintf(fid, '%s %i %s\n', 'h-1      4 0 4.7996e-002', T, 'end');
25    fprintf(fid, '%s %i %s\n', 'c        4 0 1.7066e-002', T, 'end');
26    fprintf(fid, '%s %i %s\n', 'n        4 0 2.7620e-003', T, 'end');
27    fprintf(fid, '%s %i %s\n', 'o-16     4 0 2.6288e-002', T, 'end');
28    fprintf(fid, '%s %i %s\n', 'na-23    4 0 4.2172e-005', T, 'end');
29    fprintf(fid, '%s %i %s\n', 'p-31     4 0 2.5290e-003', T, 'end');
30    fprintf(fid, '%s %i %s\n', 'cl-35    4 0 3.8170e-005', T, 'end');
31    fprintf(fid, '%s %i %s\n', 'ca-40    4 0 4.2454e-003', T, 'end');
32    fprintf(fid, '%s %i %s\n', 'mg-24    4 0 7.9765e-005', T, 'end');
33    fprintf(fid, '%s %i %s\n', 's-32     4 0 9.0724e-005', T, 'end');
34    fprintf(fid, '%s %i %s\n', 'b-10     4 0 1.2604e-006', T, 'end');
35    fprintf(fid, '%s %i %s\n', 'h-1      5 0 6.6100e-002', T, 'end');
36    fprintf(fid, '%s %i %s\n', 'c        5 0 7.3230e-003', T, 'end');
37    fprintf(fid, '%s %i %s\n', 'n        5 0 8.2825e-004', T, 'end');
38    fprintf(fid, '%s %i %s\n', 'o-16     5 0 2.8600e-002', T, 'end');
39    fprintf(fid, '%s %i %s\n', 'na-23    5 0 3.8395e-005', T, 'end');
40    fprintf(fid, '%s %i %s\n', 'p-31     5 0 7.9400e-005', T, 'end');
41    fprintf(fid, '%s %i %s\n', 'cl-35    5 0 2.4900e-005', T, 'end');
42    fprintf(fid, '%s %i %s\n', 'k-39     5 0 6.2889e-005', T, 'end');
43    fprintf(fid, '%s %i %s\n', 'b-10     5 0 8.1965e-007', T, 'end');
44    fprintf(fid, '%s %i %s\n', 'h-1      6 0 6.6100e-002', T, 'end');

```

```

45     fprintf(fid, '%s %i %s\n', 'c          6 0 7.3230e-003', T, 'end');
46     fprintf(fid, '%s %i %s\n', 'n          6 0 8.2825e-004', T, 'end');
47     fprintf(fid, '%s %i %s\n', 'o-16      6 0 2.8600e-002', T, 'end');
48     fprintf(fid, '%s %i %s\n', 'na-23     6 0 3.8395e-005', T, 'end');
49     fprintf(fid, '%s %i %s\n', 'p-31      6 0 7.9400e-005', T, 'end');
50     fprintf(fid, '%s %i %s\n', 'cl-35     6 0 2.4900e-005', T, 'end');
51     fprintf(fid, '%s %i %s\n', 'k-39      6 0 6.2889e-005', T, 'end');
52     fprintf(fid, '%s %i %s\n', 'b-10      6 0 2.8688e-006', T, 'end');
53     fprintf(fid, '%s\n', 'end comp');
54     fprintf(fid, '%s\n', 'read celldata');
55     fprintf(fid, '%s\n', 'inhommedium 1 cellmix=7 end');
56     fprintf(fid, '%s\n', 'inhommedium 2 cellmix=8 end');
57     fprintf(fid, '%s\n', 'inhommedium 3 cellmix=9 end');
58     fprintf(fid, '%s\n', 'inhommedium 4 cellmix=10 end');
59     fprintf(fid, '%s\n', 'inhommedium 5 cellmix=11 end');
60     fprintf(fid, '%s\n', 'inhommedium 6 cellmix=12 end');
61     fprintf(fid, '%s\n', 'end celldata');
62     fprintf(fid, '%s\n', 'end data');
63     fprintf(fid, '%s\n', 'end');
64     fprintf(fid, '%s\n', '=shell');
65     fprintf(fid, '%s\n', 'mv ft02f001 $RTNDIR/jos_xs.out');
66     fprintf(fid, '%s\n', 'end');
67
68     fclose(fid);
69
70     system('batch6 -m csasi');

```

xsdrrn_calc4.m

```

1
2
3     % xsdrrn calculation
4
5     n_int2=n_int/7-1;
6     n_int3=n_int/7;
7
8     fid=fopen('xsdrrn','w');
9
10    fprintf(fid, '%s\n', '=shell');
11    fprintf(fid, '%s\n', 'ln -fs $RTNDIR/jos_xs.out ft04f001');
12    fprintf(fid, '%s\n', 'end');
13    fprintf(fid, '%s\n', '=xsdrrn');
14    fprintf(fid, '%s\n', '0$$$ a3 4 a5 3 a7 17 e');
15    fprintf(fid, '%s %i %i %s %i %i %s\n', '1$$$ 1', 7, n_int, '1 0', ...
16    7, 7, '16 5 1 a11 250 250 a14 0 e');
17    fprintf(fid, '%s\n', '3$$$ 1 e');
18    fprintf(fid, '%s %i %s\n', '4$$$ 0', no_grps, '3 e t');
19    fprintf(fid, '%s\n', '13$$$ 1 2 3 4 5 6 7 ');
20    fprintf(fid, '%s\n', '14$$$ 7 8 9 10 11 12 11 ');
21    fprintf(fid, '%s\n', '15** f1');
22    fprintf(fid, '%s\n', 't');
23    fprintf(fid, '%s\n', '33## f1 t');
24    fprintf(fid, '%s %i %s %d %i %s %d\n %i %s %d %i %s %d\n %i %s %d %i %s %d\n ...
25    %i %s %d %d %s\n', ...
26    '35**', n_int2, 'i', dist(1), n_int2, 'i', dist(2), n_int2, 'i', ...
27    dist(3), n_int2, 'i', dist(4), n_int2, 'i', ...
    dist(5), n_int2, 'i', dist(6), n_int2, 'i', ...

```

```

28     dist(7), dist(8), 'e');
29 fprintf(fid, '%s %i%s %i%s %i%s %i%s %i%s %i%s\n', ...
30     '36$$', n_int3, 'r1', ...
31     n_int3, 'r2', n_int3, 'r3', n_int3, 'r4', n_int3, 'r5', n_int3, ...
32     'r6', ...
33     n_int3, 'r7 e');
34 fprintf(fid, '%s\n', '39$$ 1 2 3 4 5 6 7 e');
35 fprintf(fid, '%s %i%s %i %s\n', '51$$', no_grps-2, 'i 1', no_grps, 'e 5t');
36 fprintf(fid, '%s\n', 'end');
37 fprintf(fid, '%s\n', '=shell');
38 fprintf(fid, '%s\n', 'cp ft17f001 $RTNDIR/flux_scal.xsdrn');
39 fprintf(fid, 'end');
40
41 fclose(fid);
42
43 system('batch6 -m xsdrn');
44 system('sh run3');

```