Finite Volume Method For Charged Particle Transport In The Fokker-Planck Limit

by

J. Nauta

to obtain the degree of Bachelor of Science
at the Delft University of Technology,

Student number: 4025008

Contents

1 Introduction

1.1. Proton Radiation Therapy

Proton therapy is a type of radiation therapy which uses high energy protons to irradiate the tumor. A particle accelerater is used to create a beam of high energy protons to target the tumor. These protons are absorbed by the cancerous cells, destroying their DNA in the process. The cancer cells lack the mechanisms to repair DNA damage, making them particularly vulnerable to such damage and killing them off.

Unfortunately radiation also gets absorbed in the surrounding tissue. This damages the healthy tissue and increases the risk of new tumors and other complications. This is particularly a problem in sensitive areas such as the prostate or the brain. Proton radiation is more precise than other types of radiation therapy because of two properties of protons. Protons have a relatively high mass compared to other particles used in radiation therapy, reducing the scatter in the tissue. This keeps the proton beam narrow and focused on the tumor. Furthermore high energy protons have a very narrow Bragg peak. Very few protons get absorbed before this depth and almost none after it. By sending out proton beams at different energies the entire tumor can be irradiated with little absorbtion in the surrounding tissue. This is illustrated in figure 1.1. This also makes the proton beams susceptible to small errors in position. Numerical methods are needed to properly model the energy deposits in therapy [1].

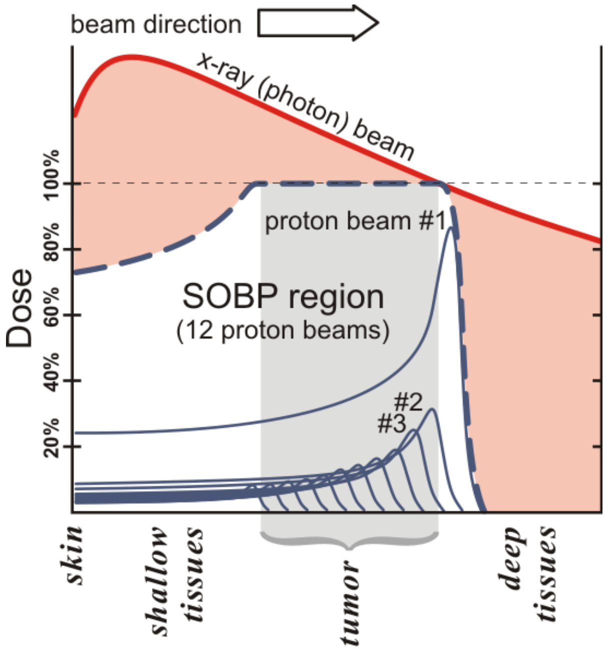

Figure 1.1: A comparison of the penetration depth of proton beams compared to x-rays. Most protons get absorbed in a very narrow region called the Bragg peak. By combining proton beams of different energies, these peaks can be combined into a region of finite size. After this region the proton absorption rate drops sharply compared to x-ray beams [1].

1.2. Numerical Methods In Proton Therapy

Unfortunately none of the currently available methods are suitable yet. One method uses a Monte Carle method to track the protons. This method is very accurate and flexible, but it is too costly for use in treatment. With the increase in computational power it is expected Monte Carlo methods will eventually become routine tools, but until such a time different methods are needed [2] [8].

Currently medical equipment uses a Fermi pencil-beam approximation. This method is fast, but inaccurate in non-homogenous tissues. Figure 1.2 shows a dose distribution calculated by these two systems. This figure shows that there is a high difference between the two systems around the brain stem. This inaccuracy could lead to future complications in the patient if the pencil-beam approximation is used.

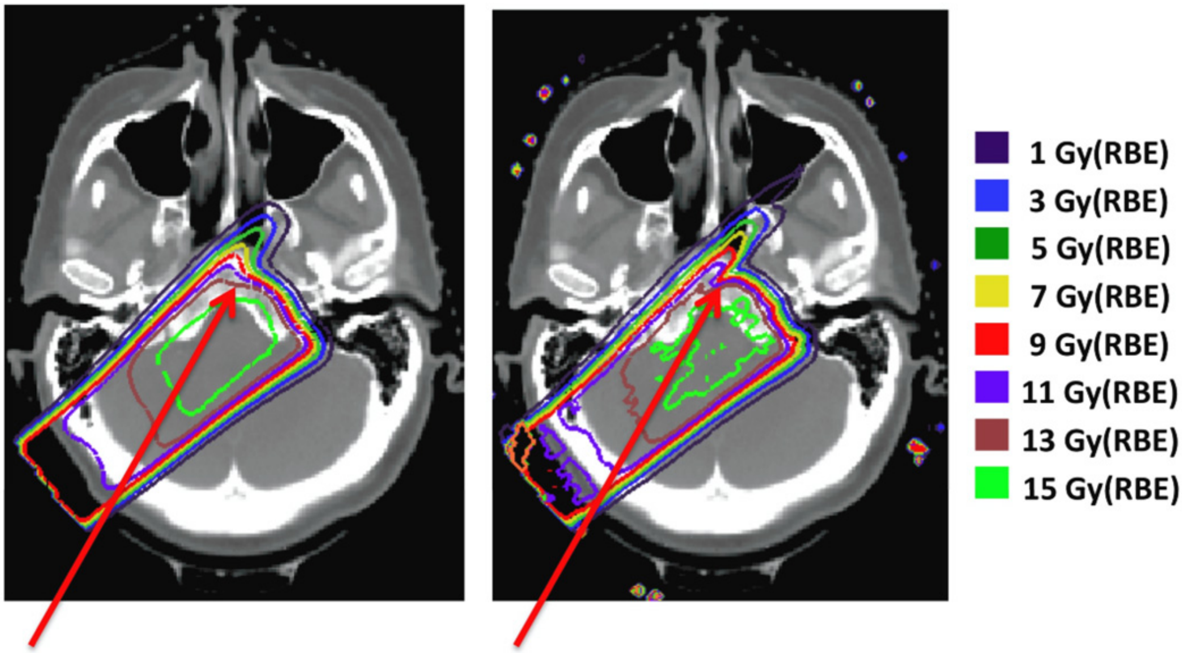

Figure 1.2: Axial view of dose distributions using two different systems. On the left is a commercially available system based on a pencil-beam algorithm (XiO (Computerized Medical Systems)) and on the right is a Monte Carlo system. The red arrow indicates an area where the discepency is large due to a density interface parallel to the proton beam [6].

1.3. The Transport Equation

In this research a mono-energetic, time-independent model of charged particle transport was considered. In such a model the angular particle flux $\phi = \phi(\mathbf{r}, \Omega)$ depends only on the spatial position $\mathbf{r}$ and the direct vector Ω. This angular flux can be described with the linear Boltzmann equation, which is given by

$$\Omega \cdot \frac{\partial}{\partial \mathbf{r}}\phi + \Sigma_{\mathbf{a}}\phi - \mathbf{Q}\phi = \mathbf{S} \tag{1.1}$$

where $\Omega \cdot \partial\phi/\partial\mathbf{r}$ is the streaming term, Σ_a is the macroscopic absorption cross section and S is the angular source. The scattering term $Q\phi$ can be written as

$$Q\phi(\mathbf{r}, \Omega) = \Sigma_S \left(\int_{S^2} \frac{1}{2\pi} p\left(\Omega' \cdot \Omega\right) \phi\left(\mathbf{r}, \Omega'\right) d\Omega' - \phi(\mathbf{r}, \Omega) \right) \tag{1.2}$$

where Σ_S is the macroscopic scattering cross section and p is a probability density function. This scattering term is dominated by Coulomb interactions between the charged particles and the nuclei. These interactions slightly deflect the particles, which results in the direction vector undergoing small deviations as the particle travels throught the medium.

Equation 1.2 can be simplified by taking the Fokker-Planck approximation. In the model the limit is taking of infinitely many particle-nucleus interactions, with each interaction being infinitesimaly small. This is chosen such that the total angular deviation per distance traveled is kept constant on average. This would mean that

$p\left(\Omega' \cdot \Omega\right) = 0$ for $\Omega' \cdot \Omega \neq \mathbf{1}$, allowing the integral to drop out of equation 1.2. Instead the scattering term is approximated as a spherical Laplace operator [5] [7]:

$$Q\phi \rightarrow \frac{\alpha}{2}\nabla_S^2\phi = \frac{\alpha}{2}\left(\frac{1}{\sin^2\psi}\frac{\partial^2\phi}{\partial^2\theta} + \frac{1}{\sin\psi}\frac{\partial}{\partial\psi}\left(\sin\psi\frac{\partial\phi}{\partial\psi}\right)\right) \tag{1.3}$$

Here θ and ψ are the polar and azimuthal angles respectively. This equation describes a diffusive process on the unit sphere. The diffusion constant α is also known as the macroscopic transport cross section. This is usually measured experimentally.
If we disregard the streaming term for now, this gives the mono-energetic Fokker-Planck equation

$$\Sigma_a\,\phi(\Omega) - \frac{\alpha}{2}\,\nabla_S^2\,\phi(\Omega) = S \tag{1.4}$$

This equation will be used to develop a numerical scheme for charged particle transport.

2
Method

2.1. Finite Volume Method

In this chapter a numerical method for modeling angular diffusion is derived. The Finite Volume Method (FVM) is chosen. This is a numerical method for approximating partial differential equations. The method is very flexible compared to the Finite Difference Method and simple compared to the Finite Elements Method (FEM). Furthermore this scheme automatically follows the conservation laws, making it highly suitable for conserved quantities.
First we consider the time-independent, mono-energetic Fokker-Planck equation.

$$\Sigma_a \, \phi(\Omega) - \frac{\alpha}{2} \nabla_S^2 \, \phi(\Omega) = S \tag{2.1}$$

In this equation Ω is the coordinate on the unit sphere and ∇_S^2 is the Laplace operator in spherical coordinates.
To apply the FVM we first divide the unit sphere in triangular patches. A 2D representation of these patches is shown in 2.1. We integrate the Fokker-Planck equation over one of these patches. Next we apply Gauss's divergence theorem to reduce the order of the differential equation.

$$\int \Sigma_a \, \phi d\Omega + \oint \frac{\alpha}{2} \, \hat{n} \cdot \vec{\nabla}\phi \; d\partial\Omega = \int S \; d\Omega \tag{2.2}$$

These integrals will each be approximated seperately. The two area integrals are approximated through a Gaussian quadrature. The differential will be approximated through a first order finite difference scheme evaluated between the centers of two neighboring patches. This will then be integrated along each of the three edges of the patch [3].

$$\int_{h_{i2}}^{h_{i1}} \frac{\alpha}{2} \, \hat{n} \cdot \vec{\nabla}\phi \; d\partial\Omega \approx \frac{\alpha}{2} \, \frac{\phi_j - \phi_i}{d_{\phi_i,\phi_j}} \cos\left(\beta\right) \; d_{h_1,h_2} \tag{2.3}$$

In this equation ϕ_i is the value of ϕ evaluated at the center of patch i, d_{ϕ_i,ϕ_j} is the distance between the centers of patches i and j along the surface of the sphere, d_{h_1,h_2} is the length of the boundary between the two patches along the surface of the sphere and β is the angle the arc between the two centers makes with the vector normal to the border between the two patches.
This approximation allows us to formulate a linear system of equations which can be solved through standard methods. For this research the LAPACK software package was used.

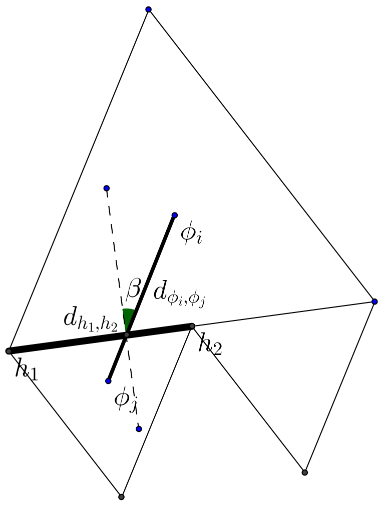

Figure 2.1: A 2D representation of the patches that constitute the mesh. Here ϕ_i is the center of patch i, d_{ϕ_i,ϕ_j} is the distance between the centers of patches i and j, h_1 and h_2 are the outsides of the edge between the two patches, d_{h_1,h_2} is the length of the edge between the two patches and β is the angle between the lines $\left(\phi_i,\phi_j\right)$ and (h_1,h_2).

2.2. FVM On Spherical Surface

To solve the equation the sphere is divided into spherical triangles. At the coarsest refinement level the sphere is divided into its eight octants. In each further refinement each spherical triangle is divided into four smaller spherical triangles. These triangles are chosen as the patches for the FVM with their centers of mass as the evaluation points. Such a mesh refined to 128 patches can be seen in figure 2.3
To evaluate the distance between two points on the unit sphere the inner product between these two points is taken.

$$\cos\left(d_{r_i,r_j}\right) = r_i \cdot r_j \tag{2.4}$$

This can be used to evaluate both the distance between the centers of two neighboring patches and the length of the boundary between the two patches.
To find the angle β we consider that $\cos\left(\beta\right) = \sin\left(\gamma\right)$ with γ being the angle the arc between the two centers makes with the border between two patches instead of the vector normal to the border. The angle between arcs is the same as the angle between the two planes spanned by the endpoints of the arcs and the center of the unit sphere, which is equal to the angle between the two vectors normal to these planes. The vector normal to a plane can be found by taking the cross product of any two independent vectors inside this plane. In this case the vectors between the center of the unit sphere and the endpoints of the arcs are chosen.

$$\begin{aligned} \cos\left(\beta\right) = \sin\left(\gamma\right) = \left\| \frac{\mathbf{n_f} \times \mathbf{n_g}}{\left\|\mathbf{n_f}\right\| \left\|\mathbf{n_g}\right\|} \right\| \\ \mathbf{n_f} = (\mathbf{h_1} - \mathbf{O}) \times (\mathbf{h_2} - \mathbf{O}) \\ \mathbf{n_g} = (\mathbf{r_i} - \mathbf{O}) \times \left(\mathbf{r_j} - \mathbf{O}\right) \end{aligned} \tag{2.5}$$

Here $\mathbf{r_i}$ and $\mathbf{r_j}$ are the centers of the two neighboring patches and $\mathbf{h_1}$ and $\mathbf{h_2}$ are the two endpoints of the border between the two patches.
With the values for β and d_{r_i,r_j} the linear system described in 2.3 can be constructed and the value of the flux can be estimated for each patch.

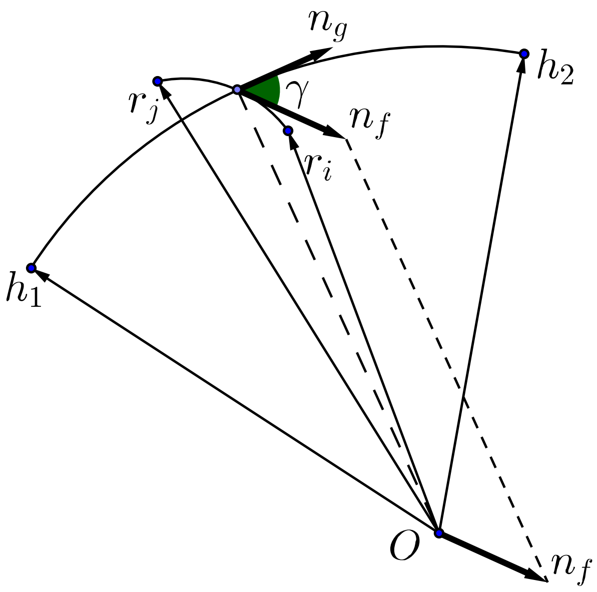

Figure 2.2: A figure showing the calculation for the value of β. Here r_i and r_j are the vectors from the center of the sphere to the centers of patches i and j, h_1 and h_2 are the vectors from the center of the sphere to the outsides of the edge between patches i and j and n_f is the cross product between h_1 and h_2, n_g is the cross product between r_i and r_j. $\sin(\gamma)$ can be calculated by taking the cross product of n_f and n_g and from γ the value of β can be calculated.

2.3. Error Estimates For Uniform Refinement

To determine the rate of convergence for the method only a single example needs to be considered. Any other solutions should follow the same rate. It is often simplest to use the eigenfunctions.

$$\phi(\Omega) = Y_{lm}(\Omega) \tag{2.6}$$

$$S(\Omega) = \left(\Sigma_a + \frac{\alpha}{2} l(l+1)\right)\phi(\Omega) \tag{2.7}$$

In these equations Y_{lm} is the real spherical harmonic with eigenvalues l and m. They are normalized such that $\langle Y_{lm}, Y_{l',m'}\rangle = \delta_{ll'}\delta mm'$.
The source has been chosen to obtain the following solution.

$$\phi(\Omega) = 0.7Y_{00} - 0.5Y_{11} + 0.1Y_{20} + 0.3Y_{32} \tag{2.8}$$

Furthermore the constants have been set to $\alpha = 25$ and $\Sigma_a = 10$.
The mesh is uniformly refined with the patch size defined as

$$h = \sqrt{\frac{4\pi}{N}} \tag{2.9}$$

with N being the total number of patches on the sphere. The error is defined as

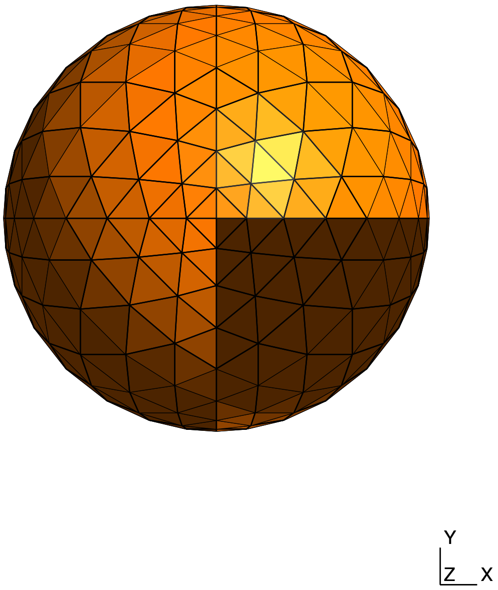

Figure 2.3: A sphere divided into a triangular mesh. The mesh consists of 512 patches.

$$\epsilon = \sqrt{\frac{\int (\phi - \phi_h)^2 \, d\Omega}{\int \phi^2 d\Omega}} \tag{2.10}$$

Here ϕ_h is the value of the flux in patch h estimated through the FVM and ϕ is the actual value. The integral is estimated with a quadrature method.

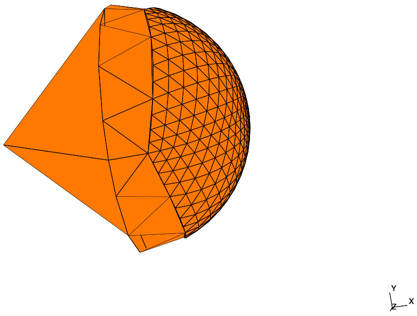

Figure 2.4: A sphere divided into a nonuniform mesh. The mesh is more refined closer to the top of the sphere. This mesh consists of 2324 patches.

2.4. Local Refinement

In reality the angular aspect of the proton beam is not similar to a spherical harmonic at all. The beam is very narrow with little defiation in direction. To simulate this the same error estimation was performed with a source which gave a localized Gaussian function as solution.

$$\phi\left(\Omega\right) = \exp\left(\frac{\Omega_x^2 + \Omega_y^2 + \left(\Omega_z - 1\right)^2}{4\sigma^2}\right) \tag{2.11}$$

$$S\left(\Omega\right) = \left(\Sigma_a + \frac{\alpha}{2}\left(\frac{\Omega_x^2}{4\sigma^4} - \frac{\Omega_x}{\sigma^2}\right)\right)\phi\left(\Omega\right) \tag{2.12}$$

This Gaussian solution is near zero on most of the unit sphere. Even the coarsest refinement level is able to compute the solution in this area with a very low relative error. The peak however does require a high refinement level to properly compute the solution. It is wasteful of resources to refine the entire sphere into such a high refinement level, so a local refinement algorithm should be used which only refines the patches where the solution value is high. Patches with near zero solutions are left unrefined.

Because we are working with a symmetric function the refinement was done in layers. Patches closer to the top of the sphere were allowed to refine more often than patches closer to the bottom, with patches on the bottom half not being refined at all. The boundaries of these layers were found through trial and error. A mesh made through this refinement scheme can be seen in 2.4

3
Results

3.1. Uniform Refinement

The results for the FVM on a uniform mesh can be found in table 3.1.

Table 3.1: Results of the FVM on a uniformly refined mesh with the exact solution given by 2.8. The patch size h is calculated according to 2.9 and the error ϵ is calculated according to 2.10.

N	h	ϵ
$8*4^0$	1.253*10^0	$4.459*10^{-1}$
$8*4^1$	6.267*10^{-1}	$3.052*10^{-1}$
$8*4^2$	3.133*10^{-1}	$1.442*10^{-1}$
$8*4^3$	1.567*10^{-1}	$7.420*10^{-2}$
$8*4^4$	7.833*10^{-2}	$3.857*10^{-2}$
$8*4^5$	3.917*10^{-2}	$2.178*10^{-2}$

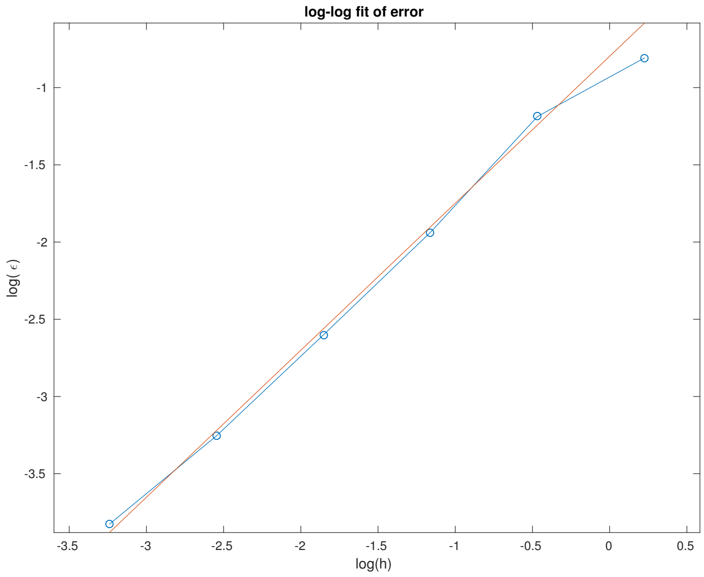

Figure 3.1: A log-log plot of the data points. The logarithm of the patch size h is on the x-axis and the logarithm of the error ϵ is on the y-axis. A linear fit has been performed on the datapoints to estimate an order of convergence of 0.951 ± 0.042. The first datapoint has been excluded from the fit due to its poor fitness.

Table 3.2: Results of the FVM on a locally refined mesh with the exact solution given by 2.11. The patch size h is calculated according to 2.9 and the error ϵ is calculated according to 2.10.

N	h	ϵ
8	1.253*10^{0}	$8.962 * 10^{-1}$
20	7.927*10^{-1}	$4.637 * 10^{-1}$
44	5.344*10^{-1}	$2.614 * 10^{-1}$
128	3.133*10^{-1}	$1.712 * 10^{-1}$
608	1.438*10^{-1}	$7.565 * 10^{-2}$
2324	7.353*10^{-2}	$4.745 * 10^{-2}$
9020	3.733*10^{-2}	$3.736 * 10^{-2}$

To estimate the order of convergence the logarithm was taken of the patch sizes and error values and a linear fit was performed on these values. The order of convergence is estimated by taking the slope of this linear fit. This fit can be found in figure 3.1. In this fit the first datapoint has been excluded because of its high defiation from the fitted linear function. From this fit an order of convergence of 0.951 ± 0.042 has been estimated. This estimate is close to the expected value of 1. The reason the estimated order of convergence is lower than the predicted one could be because the solution is insufficiently smooth on the patches. This is particularly visible on the largest patch size, where the order of convergence was low enough to be excluded from the fit. But even on the finer patches this effect should still be visible.

3.2. Local Refinement

The results for the FVM on a locally refined mesh can be found in table 3.2.

To estimate the order of convergence the logarithm was taken of the patch sizes and error values and a linear fit was performed on these values. The order of convergence is estimated by taking the slope of this linear fit. This fit can be found in figure 3.2 as the blue line.
From this fit an order of convergence of 0.908 ± 0.073 has been estimated. This estimate is close to, but a little less than the expected value of 1. The estimate is here a bit lower than in the uniformly refined case could be because of the different test function used. The Gaussian has a more complicated source than the eigenfunctions, which would add more to the error. It is also more local, which would further decrease the accuracy of lower refinements we have also seen in the case of the uniform refinement.
In figure 3.2 we can also see a fit of the Gaussian test function on a uniformly refined mesh as the green line. From the figure we can see that the locally refined mesh follows a similar trend as the uniformly refined mush, except the error is lower for the same number of patches.

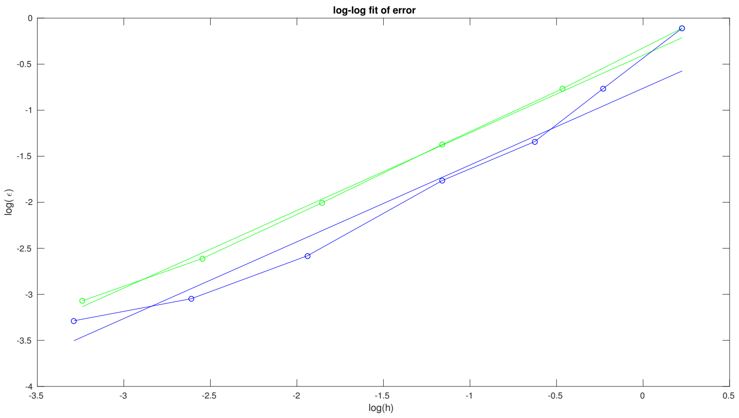

Figure 3.2: A log-log plot of the data points of the Gaussian test function on a uniformly refined mesh (green) and a locally refined mesh (blue). The logarithm of the patch size h is on the x-axis and the logarithm of the error ϵ is on the y-axis. The uniformly and locally refined mesh fits show a similar slope, with the locally refined mesh having a lower estimated error than the uniformly refined mesh at the same number of patches. A linear fit has been performed on the datapoints of the blue curve to estimate an order of convergence of 0.908 ± 0.073.

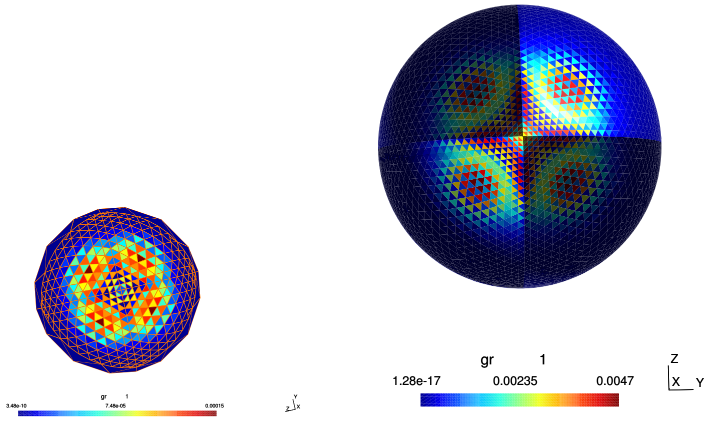

Figure 3.3: The top figure shows a heat map of the numerical error on a mesh that has been locally refined to 2324 patches. The red areas are areas that have high numerical error while the areas blue areas have low numerical error. There is a band of high error around the top of the sphere, where the derivative of the Gaussian would be highest.

The bottom figure shows the same function solved on a mesh refined up to 8192 patches. This figure shows high numerical errors in the same areas as the top figure. This suggests the error is inherent in solving the problem on this type of mesh rather than being a consequence of the local refinement.

4 Conclusion and Discussion

4.1. Conclusion

This research has focused on two separate parts. First the Finite Volume Method has been developed and tested on a uniformly refined mesh. Uniformly refined meshes are ordinarily used for FVM and provided a benchmark for the other test. Next this FVM scheme has been tested on a mesh which has been locally refined. In such a mesh the size of the patches, the angle the numerical flux makes through the edge of such a patch and the number of neighbors a patch has is different for each patch. These conditions are normally not present in a FVM scheme. If the FVM works properly in such a mesh, it is expected the rate of convergence will be higher than on a uniformly refined mesh.
The local refinement in this research was performed by splitting the mesh into layers, with each layer being allowed to refine a maximum number of times. The boundaries of these layers were found through trial and error.

The FVM works properly on a uniformly refined mesh refined up to 5 times. The convergence rate in this range has been estimated to be 0.951 ± 0.042. This is close to the expected value of 1. This estimated value is lower than the theoretical one because the function is insufficiently smooth on the patches. This is an inherent feature of this method and there is no obvious way to improve this. The convergence rate is expected to go to 1 as the refinement increases.

The FVM also converges almost linearly with local refinement. A convergence rate of 0.908 ± 0.073 has been estimated on this mesh. A uniformly refined mesh showed a similar convergence rate with the same test function. However the locally refined mesh showed a lower estimated error than the uniformly refined mesh at the same number of patches. This shows that the FVM also works properly on non-uniformly refined meshes and that this can be used to increase performance by leaving the mesh unrefined at uninteresting sections of the sphere.

4.2. Discussion

This FVM scheme appears to work properly as it is. However, the numerical error might be reduced further by increasing the order of the derivative approximations. This might become more complicated because the mesh is not rectangular but rather triangular. Another improvement to this FVM scheme might be obtained by using a more accurate method to approximate the numerical flux integrals. Currently the flux is taken to be constant at the edge between two patches. A higher accuracy might be obtained by instead using a quadrature scheme.
It might also be worth to investigate the h-box method. This method is supposed to increase the accuracy of the FVM on non-uniform meshes. The effort of implementing such a scheme was deemed to be outside the scope of this research [4].

The local refinement algorithm used could also be improved. For this research the sphere was split into layers with the boundaries of each layer being found through trial and error. A better refinement scheme might improve performance even further. That would also allow this FVM scheme to be tested with more complicated test functions.

The next step would be to incorporate this scheme into an upwind streaming scheme to emulate the total Boltzmann equation and test it against experimental values. At this point it should become more clear how this method holds up against different methods of proton stream modeling.

Bibliography

[1] M. van Vulpen B. W. Raaymakers, J. J. W. Lagendijk and J. Hartman. Mri guided proton therapy. Web page, 2017. URL `http://www.radiotherapie.nl/research/research-projects/mri-guided-radiotherapy/mri-guided-proton-therapy`. Recovered 2017.

[2] Harald Paganetti et al. Clinical implementation of full monte carlo dose calculation in proton beam therapy. *Physics in Medicine and Biology*, 53(17):4825, 2008.

[3] A. Segal J. van Kan and F. Vermolen. *Numerical Methods in Scientific Computing*. VSSD, 2008.

[4] R. J. Leveque M. J. Berger and L. G. Stern. Finite volume methods for irregular one-dimensional grids. *Proceedings of Symposia in Applied Mathematics*, 48:255–259, 1994.

[5] J. E. Morel. Fokker-planck calculations using standard discrete ordinates transport codes. *Nuclear Science and Engineering*, 79:340–356, June 1981.

[6] Harald Paganetti. Range uncertainties in proton therapy and the role of monte carlo simulations. *Physics in medicine and biology*, 57(11):R99117, June 2012b.

[7] Sander Bastiaan Uilkema. Proton therapy planning using the s n method with the fokker-planck approximation. *Master's thesis*, Delft University of Technology, October 2012.

[8] Kevin T. Murphy Xun Jia, Todd Pawlicki and Arno J. Mundt. Proton therapy dose calculations on gpu: advances and challenges. *Translational Cancer Research*, 1(3), 2012.