Delft, University of Technology

bachelor's thesis

Higher order basis functions for the discontinuous Galerkin finite element method

Author:
Emiel Hartsema

Supervisor:
Dr. Ir. Danny Lathouwers

July 19, 2018

TU Delft, Faculty of Applied Sciences, BSc program Applied Physics

Abstract

Proton therapy is a relatively new method of treating cancer, with this method more accurate treatment plans have to be made. Phantom-DG is a program designed to make treatment plans more quickly than with traditional methods with the use of the discontinuous Galerkin finite element method. This thesis explores the possibility of increasing the currently existing linear basis set to quadratic basis sets for angular diffusion. This would increase the order of convergence for the solver and is potentially favorable in terms of computing time or accuracy. The basis sets are defined on the sphere and on the octahedron, where the functions on the sphere are spherical harmonics and the functions on the octahedron are 2d polynomials on the surface of the octahedron, and are projected to the unit sphere. The basis set quadratic on the sphere encounters significant numerical errors. The direct cause of these errors is unknown. The most likely cause are errors introduced when solving linear systems of equations to calculate the coefficient matrix C. The set quadratic on the octahedron does not encounter these errors and does converge, however its rate of convergence is low enough to deem it unsuitable for practical use. A resolution for this problem is unknown. If all these problems were fixed one could expect a order of convergence for both quadratic basis sets of 3. If the rate of convergence cannot be increased using extended basis sets is not advantageous.

Contents

List of Symbols

Symbol	Meaning
φ	Angular Flux
S	Angular source
Ω	Unit vector
Σ_a	Macroscopic absorption cross section
α	Macroscopic transport cross section
∇	Euclidean gradient operator
∇_s	Spherical gradient operator
Δ_s	Spherical Laplace operator
L_{FP}	Fokker-Planck operator
$I^{n\times n}$	$n \times n$ Identity matrix
ϕ_i	i-th basis function
$\{\phi\}$	Span of functions ϕ_i
P_i	i-th vertex on a spherical patch
δ_{ij}	Kronecker delta function
V_i	i-th vertex on the octahedron
ξ	Local coordinate vectors on a flat triangle
$\|\|z\|\|_n$	L_n Norm of z
z	Direction vector with $\|\|z\|\|_1 = 1$
Y_{lm}	Spherical harmonic with order l and degree m
$\langle\cdot,\cdot\rangle$	Standard inner product on local patch
φ_h	numerical approximation of φ

1 Introduction

1.1 Background

In the treatment of cancer, radiotherapy is an often used method. With this method a beam is used to deposit energy in the body, the deposited energy per kilogram body mass is called dose. This dose can be made sufficiently strong to damage and kill cells. By focusing the beam on a specific spot treatment planners can choose where to deposit the most dose. Traditionally this kind of therapy is done with gamma radiation. The disadvantage of using gamma radiation is a relatively high entrance and exit dose, as shown in the figure below.

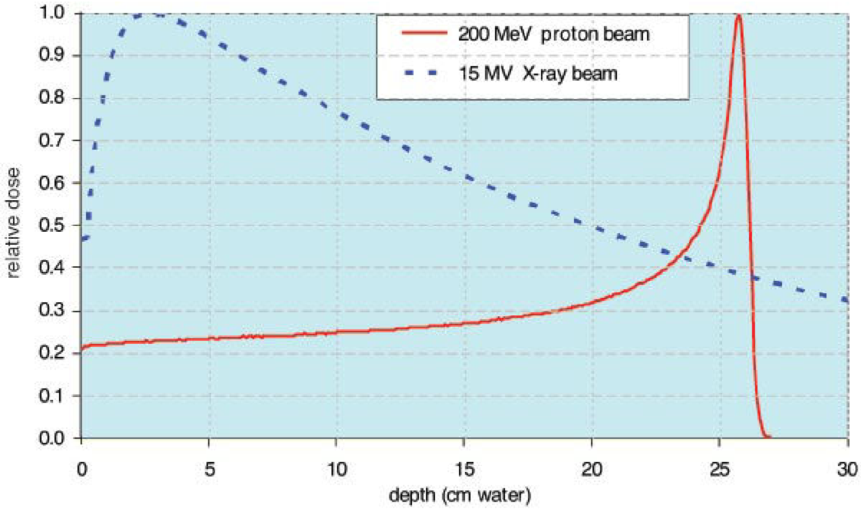


Figure 1: Dose deposition of gamma therapy and proton therapy

In recent years it has become more feasible to use protons to treat cancers. Protons have a much lower entrance dose and the exit dose is almost non-existent. The enhanced precision of proton therapy requires enhanced accuracy of the treatment plan.

The primary method to calculate the treatment plan is by Monte Carlo simulation. This type of simulation is reliable but slow. In this type of simulation every variable is discretized and the computer calculates all possible outcomes with random inputs to give a final dose distribution. The section of Reactor Physics and Nuclear Materials (RPNM) of the department of Radiation Science and Technology (RST) is working on a deterministic particle transport code to calculate treatment plans more quickly and with sufficient accuracy. In a deterministic code the solution is not calculated with random inputs or random processes, it works by numerically integrating integrals and numerically approximation the solution. Deterministic code will always generate the same output for a certain input.

1.2 Setup of this thesis

The next sections briefly explains some important concepts and gives an introduction to the particle transport equations and the method used to solve these equations numerically. This is followed a chapter explaining the mathematics behind two different types of basis functions. Chapter 4 presents the mathematics required to calculate the shape of the basis functions, and chapter 5 explains how the errors are quantified. The results of this thesis are presented in chapter 6, this is followed by a discussion and conclusions in chapters 7 and 8. Appendix A lists different configuration of nodal point and corresponding coefficient matrices, and appendix B contains visual representations of the basis functions.

2 Introduction to proton transport

2.1 Important quantities

Macroscopic absorption cross section

To introduce the concept of macroscopic absorption cross section, the microscopic cross section has to be introduced first.
Assume a beam of indecent protons to a thin slice of material as shown below

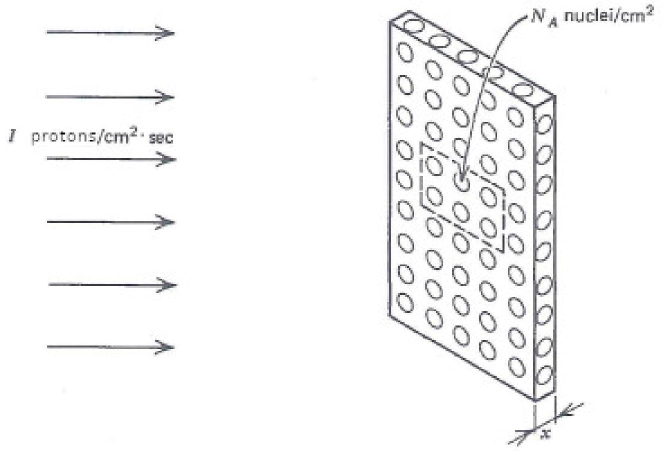


Figure 2: Schematic representation of a proton beam indecent to a thin sheet of target nuclei [3]

The microscopic cross section (σ) can be visualized as the area of a cross section of a sphere around the nucleus where a proton passing by will interact with it. The sphere of influence is often much larger than geometrical cross section of the nucleus itself. To find the microscopic absorption cross section the cross section is multiplied with a factor that is defined as follows: What is the probability the proton is absorbed on entry of the sphere of influence? In other words what is the probability the interaction is an absorption. With this procedure all partial microscopic cross sections can be defined, for example transport cross section[1] and scattering cross section.

To define the macroscopic cross section, assume a block of arbitrary thickness. At a depth x there is a certain intensity $I(x)$, this is lower than the original intensity because some protons have already scattered or have been absorbed. As shown above the chance a proton will interact with a single nucleus when traveling through a unit area is given by the microscopic cross section. The amount of protons scattering in a slice with thickness dx

[1] The transport cross section is a combination of scattering and absorption and quantifies how difficult it is for a proton to pass through.

is

$$R = I(x)\sigma N dx = -[I(x) - I(x + dx)] = -dI(x) \tag{1}$$

Where R is the amount of protons interacting in the slice with thickness dx every second and N the amount of nuclei per unit area. Rearranging the equation gives

$$\frac{dI(x)}{dx} = -\sigma N I(x) = -\Sigma I(x) \tag{2}$$

Σ is the macroscopic cross section. If one is only interested in one type of interaction, the macroscopic absorption cross section is defined with the microscopic absorption cross section $\Sigma_a = \sigma_a N$. This procedure is equivalent for all partial macroscopic cross sections

Angular flux

Assume a small cube of material with sides dr around r, in this cube there are a certain number of protons, this number can be expressed in the terms of the proton density, Ndr^3. These protons can have different energies or travel in any direction. The protons can also be separated into proton densities by their energy and direction of motion, this gives rise to a different density, n. Where $n\ dr^3 dEd\hat{\mathbf{\Omega}}$ is the amount of protons in a small cube with sides dr around r, with a energy dE around E and traveling in direction $d\hat{\mathbf{\Omega}}$ around $\hat{\mathbf{\Omega}}$. The Angular flux φ is defined as the proton density multiplied by the velocity of the protons.

$$\varphi = vn(\mathbf{r}, E, \hat{\mathbf{\Omega}}) \tag{3}$$

The physical interpretation of this quantity can be thought of as the amount of protons traveling through a plane.

2.2 Transport equations

Proton transport is governed by the transport equation, as shown in [3]

$$\mathbf{\Omega}\frac{\partial}{\partial r}\varphi + \Sigma_a\varphi - Q\varphi = S \tag{4}$$

φ is the angular flux, S is the source term, $\mathbf{\Omega}\frac{\partial}{\partial r}\varphi$ is the spatial streaming part of the equation, Σ_a the macroscopic absorption cross section and Q is a collection of all the scattering terms. The terms in the equation represent the three possible outcomes for a proton traveling in a medium. The spatial streaming term represents the proton moving though without interaction and allows for spatial translation of the proton. The second term represents the absorption of a proton in the medium. The third term represents scattering and allows for translation in angular space.

Charged particles, for example protons, have a very short mean free path (mfp). Because of this short mfp the scattering term $Q\varphi$ in the transport equation can be approximated with the Fokker Planck approximation [6].

$$Q\varphi \to \frac{\alpha}{2}\Delta_s\varphi \equiv L_{FP}\varphi \tag{5}$$

In this equation α is the macroscopic transport cross section and Δ_s is the spherical Laplacian on the unit sphere[2]. The macroscopic transport cross section and the spherical Laplacian define the Fokker-Planck operator. The Fokker Planck approximation takes the limit where the amount of interactions go to infinity, and the scatter angle goes to zero. The particles will behave like a diffusing gas as described by Fick's law, $\dot{n} = D\Delta n$. The Fokker-Planck approximation is essentially Fick's second law in angular space.

The Laplacian used in the Fokker-Planck equation has to take into account the two dimensional geometry of the space it is embedded, namely the surface of the unit sphere. To make sure the Laplacian is calculated correctly the gradient vector of any function on the unit sphere has to be parallel to the surface. This is achieved by removing the radial component from the euclidean gradient.

$$\nabla_s = (I^{3\times3} - \mathbf{\Omega}\mathbf{\Omega}^T)\nabla \tag{6}$$

with $\Delta_s = \nabla_s \cdot \nabla_s$. ∇ is the standard gradient with respect to the Cartesian coordinates $\mathbf{\Omega}$.

To solve the particle transport equation, the problem is split in two parts, a spatial and angular problem. The domain is meshed in to voxels, each voxel containing an angular mesh. The angular problem is solved for each angular mesh. In this thesis, the spatial problem will not be discussed. Therefore a test case is made with only one voxel, because there are no neighboring voxels for a particle to move to the spatial streaming term vanishes from the equation and (4) reduces to

$$\Sigma_a\varphi - L_{FP}\varphi = S \tag{7}$$

This equation is called the angular diffusion equation in the Fokker-Planck limit.

2.3 Galerkin method

Equation (7) is solved using the discontinuous Galerkin finite element method (DGFEM) on a unit sphere which is meshed into angular elements. The Galerkin method is based on a discretization of the solution space V into V_n, with $V_n \subset V$. The Galerkin method aims to find $u_n \in V_n$ where u_n is the projection of the analytical solution u on the discrete solution space V_n

[2] Not to be confused with a Laplacian in spherical coordinates

[4]. The discrete solution space V_n is spanned by a set of basis functions. By increasing the set of basis function to a higher order, the discrete solution space V_n is increased and solutions can be found more accurate.
For DGFEM the problem has to be stated as a weak formulation

$$B(u,v) = F(v) \qquad \forall v \in V \tag{8}$$

The solution $u \in V$, for all test functions $v \in V$ with bilinear operator B, where V is the solution space.
Writing equation (4) with Fokker-Planck in bilinear form gives

$$\begin{aligned} B(u,v) \quad = \quad & \int_{\mathbb{S}^{\neq}} \frac{\alpha}{2} \nabla_s u \cdot \nabla_s v - \sum_{F \in \mathbb{F}_h} \int_F \Big([v]\{\frac{\alpha}{2}\nabla_s u\} \cdot \mathbf{n}_F + [u]\{\frac{\alpha}{2}\nabla_s v\} \cdot \mathbf{n}_F \Big) \\ & + \sum_{F \in \mathbb{F}_h} \int_F \frac{\alpha}{2} \frac{\eta}{h_F} [u][v] + \int_{\mathbb{S}^{\neq}} \Sigma_a uv \end{aligned} \tag{9}$$

The terms of this equation are explained in [5]. The specific meaning of the terms in this equation are not important for this thesis.
In equation (9) the terms in the integrals are all constants, the value of u or v or the spherical gradient of u or v. u_n and v_n are a projections of u and v on the discrete solution space, therefore they can be written as a linear combinations of the basis functions. If the constants are known, only the function values and gradients of the basis functions have to be calculated to evaluate equation (9).

2.4 Goal of this thesis

This thesis investigates the possibility of extending the linear basis functions currently in the program to quadratic basis functions, as identified by Aldo Hennink in his Master Thesis [5], and subsequent paper [6].

3 Basis functions

If a set of functions spans a solution space, a linear combination of these functions will span the same space, given the linear combination does not project to a lower dimension. This means in general the basis functions can be written as a linear combination of a certain set of spanning functions.

$$\phi = Cb \tag{10}$$

Where ϕ contains the basis functions ϕ_i, b contains the spanning functions which span the solution space V_n and C is a coefficient matrix. The columns of C have to be independent to guarantee $C : \mathbb{R}^n \rightarrow \mathbb{R}^n$.
The next sections will describe two different sets of spanning functions b. The calculation of C is discussed in the next chapter.

3.1 Spherical harmonic functions

The basis functions have to be defined on the unit sphere. Therefore logical choice for these functions are spherical harmonics. To avoid the use of trigonometric functions, $\mathbf{\Omega}$ is expressed in Cartesian coordinates with the constraint $||\mathbf{\Omega}||_2 = 1$.

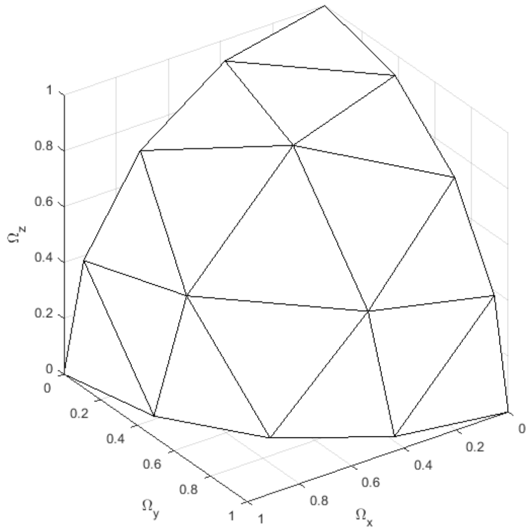


Figure 3: One octant of a level 2 uniform refined sphere

Spherical harmonics in Cartesian representation up to order 3 are listed in [2].

This thesis only works with sets of spherical harmonics up to order 2 but could theoretically be extended to arbitrary order.

$$b = \begin{bmatrix} 1 \\ \Omega_x \\ \Omega_y \\ \Omega_z \\ \Omega_x\Omega_y \\ \Omega_x\Omega_z \\ \Omega_y\Omega_z \\ 2\Omega_z^2 - \Omega_x^2 - \Omega_y^2 \\ \Omega_x^2 - \Omega_y^2 \end{bmatrix}$$

Basis functions linear in Omega only use the first 4 functions from the above array as spanning functions, the full set is referred to as quadratic in Omega. The spanning functions themselves cannot be used as basis functions (i.e. $C = I^{n\times n}$). If the patches become very small these functions will become nearly linear dependent making them very susceptible to numerical error as shown in [8].
The spherical gradient can easily be calculated because the basis functions are direct functions of omega.

$$\nabla_s \phi = (I^{3\times 3} - \mathbf{\Omega}\mathbf{\Omega}^T)\frac{\partial b}{\partial \Omega} C^T \tag{11}$$

with gradient $\frac{\partial b}{\partial \Omega} = \sum_i \frac{\partial b}{\partial \Omega_i} e_i$

3.2 Octahedron functions

The basis functions can also be defined on the octahedron (L_1-sphere). The basis functions are defined on the flat triangles, and projected to the unit sphere along straight lines intersecting the origin.
A flat triangle is defined by three vertices, V_1, V_2, V_3. A triangle can be spanned by two vectors and one support.

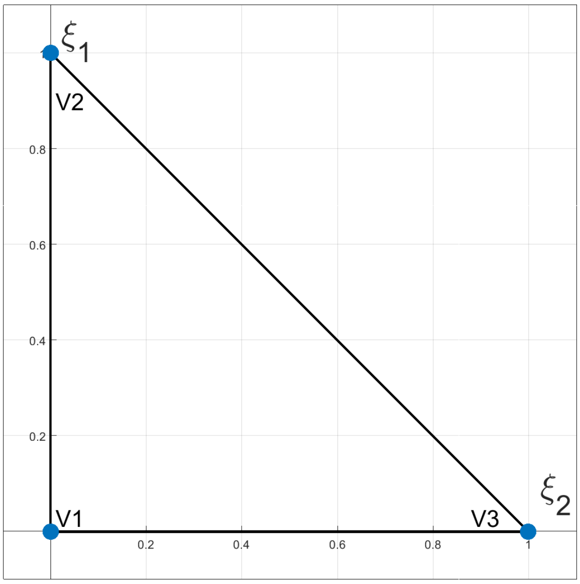


Figure 4: A flat triangle spanned by ξ_1 and ξ_2 supported on $V1$

Any point z in the triangle can be found as a function of $\xi = [\xi_1, \xi_2]^T$

$$\begin{aligned} z &= V_1 + (V_2 - V_1)\xi_1 + (V_3 - V_1)\xi_2 \\ &= V_1 + D\xi \end{aligned} \tag{12}$$

Where $D = [(V_2 - V_1), (V_3 - V_1)]$, and $\xi = \{\xi_1 \geq 0, \xi_2 \geq 0, \xi_1 + \xi_2 \leq 1\}$. To find ξ given V_1, V_2, V_3 one could think to invert D and multiply with one of the vertices to find its corresponding set ξ. This is not regularly possible because D is not a square matrix, it is however possible to find a matrix E such that $ED = I$, this is called a left inverse[3].

$$E = (D^T D)^{-1} D^T \tag{13}$$

Combining equation (12) with (13)

$$\xi = E(z - V_1) \tag{14}$$

3.3 Octahedron basis functions

The set of functions b linear on the octahedron is defined as follows:

$$b = \begin{bmatrix} 1 - \xi_1 - \xi_2 \\ \xi_1 \\ \xi_2 \end{bmatrix} = C^0 \begin{bmatrix} 1 \\ \xi_1 \\ \xi_2 \end{bmatrix}$$

And the set of functions b quadratic on the octahedron is given by [7].

[3] Note: the left inverse cannot be seen as an inverse matrix because, $ED \neq DE$

$$b = \begin{bmatrix} (2\xi_1 - 1)\xi_1 \\ (2\xi_2 - 1)\xi_2 \\ (2(1-\xi_1-\xi_2)-1)(1-\xi_1-\xi_2) \\ 4\xi_1\xi_2 \\ 4\xi_2(1-\xi_1-\xi_2) \\ 4\xi_1(1-\xi_1-\xi_2) \end{bmatrix} = C^0 \begin{bmatrix} 1 \\ \xi_1 \\ \xi_2 \\ \xi_1^2 \\ \xi_2^2 \\ \xi_1\xi_2 \end{bmatrix}$$

The seemingly arbitrary choice to set the structure functions b to a linear combination of a simple set of structure functions will be explained in section 4.

The spherical gradient is more difficult to calculate for functions on the octahedron with respect to spherical harmonic functions. This is because the basis set is not defined in terms of Ω, therefore a Jacobian matrix has to be developed.

$$\begin{aligned} \nabla_s\phi &= (I - \Omega\Omega^T)\frac{\partial\phi}{\partial\Omega} \\ &= (I - \Omega\Omega^T)\frac{\partial z}{\partial\Omega}\frac{\partial\xi}{\partial z}\frac{\partial b}{\partial\xi}C^T \\ &= \frac{\partial z}{\partial\Omega}\frac{\partial\xi}{\partial z}\frac{\partial b}{\partial\xi}C^T \end{aligned} \tag{15}$$

The last equation in (15) is derived in (20). The above set of equations states: if a basis function is defined in terms of z, and z can be expressed in terms of Ω, its spherical gradient is equal to the Euclidean gradient with respect to Ω.

The differentials have to be calculated separately.

The calculation of $\frac{\partial z}{\partial\Omega}$ starts with the relations between Ω and z

$$\Omega = \frac{1}{||z||_2}z \qquad z = \frac{1}{||\Omega||_1}\Omega \tag{16}$$

from these relations can be derived $||\Omega||_1^{-1} = ||z||_2$. $\frac{\partial z}{\partial\Omega}$ can be calculated from the second equality in (16)

$$\begin{aligned} \frac{\partial z}{\partial\Omega} &= \frac{\partial}{\partial\Omega}\left(\frac{1}{||\Omega||_1}\right)\Omega^T + \frac{1}{||\Omega||_1}\frac{\partial\Omega}{\partial\Omega} \\ &= \frac{\partial||\Omega||_1}{\partial\Omega}\frac{\partial||\Omega||_1^{-1}}{\partial||\Omega||_1}\Omega^T + \frac{1}{||\Omega||_1}I^{3\times3} \\ &= \text{sign}(\Omega)\frac{-1}{||\Omega||_1^2}\Omega^T + \frac{1}{||\Omega||_1}I^{3\times3} \\ &= ||z||_2(I^{3\times3} - \text{sign}(z)z^T) \end{aligned} \tag{17}$$

The function sign(z), works component-wise and defined as follows

$$\text{sign}(\mathbf{v}) = \sum_i \text{sign}(v_i)e_i \tag{18}$$

Where sign(n) for a real number n is defined as

$$sign(n) \begin{cases} -1 & n < 0 \\ 1 & n \geq 0 \end{cases} \tag{19}$$

the equality $\text{sign}(\Omega) = \text{sign}(z)$ follows from the fact that an angular element is made from refining the octahedron, therefore every element is fully contained in one octant of the octahedron.

The last equation in (15) is because

$$\begin{aligned} \Omega^T \frac{\partial z}{\partial \Omega} &= \Omega^T ||z||_2 (I^{3\times3} - \text{sign}(z) z^T) \\ &= z^T ||z||_2^2 (I^{3\times3} - \text{sign}(z) z^T) \\ &= ||z||_2^2 (z^T - z^T \text{sign}(z) z^T) \\ &= 0 \end{aligned} \tag{20}$$

with $z^T \text{sign}(z) = ||z||_1 = 1$.

$\frac{\partial \xi}{\partial z}$ can be found by taking the partial derivative with respect to z from equation (14)

$$\frac{\partial \xi}{\partial z} = E^T \tag{21}$$

$\frac{\partial b}{\partial \xi}$ can easily be found by taking the gradient of b with respect to ξ_1 and ξ_2

4 Calculating the coefficient matrix

The specific choice of C does theoretically not matter, however the problem is solved numerically therefore there are some additional conditions. To check if the basis functions behave well in numerical processes, the mass matrix is introduced, after which two methods to calculate C are explained.

4.1 Mass matrix

If u is the analytical solution, u_n is the projection of u on the span of the basis functions $\{\phi\}$. u_n can be expressed as a linear combination of the basis functions.

$$u_n = \sum_i \alpha_i \phi_i \tag{22}$$

applying Fourier's trick gives

$$\langle u, \phi_j \rangle = \langle u_n, \phi_j \rangle = \sum_i \alpha_i \langle \phi_i, \phi_j \rangle \tag{23}$$

The inner product is the integral over the local spherical triangle. The quantity of interest are the coefficients of α as they can directly be used to calculate u_n with (22). Equation (23) can be written as a matrix vector product $\vec{u_\phi} = M\vec{\alpha}$ with the coefficients of M given by

$$M_{ij} = \langle \phi_i, \phi_j \rangle \tag{24}$$

M is called the Mass matrix. To accurately calculate α, the mass matrix has to be well conditioned. If $\kappa(M)$ is the condition number of the mass matrix, and $e(\vec{u_\phi})$ the relative error introduced in $\vec{u_\phi}$, than the relative error in α is given as [1]

$$e(\alpha) \leq \kappa(M) e(\vec{u_\phi}) \tag{25}$$

If the condition number of M is very large, a small error in $\vec{u_\phi}$ can lead to a large error in $\vec{\alpha}$. When there is a large error in α the calculation of u_n is inaccurate.

4.2 Nodal points

With the nodal point method, C can be calculated such that a reasonably conditioned mass matrix M is expected. The idea is to pick as many points P_i as there are spanning functions and label the points accordingly. The basis functions are defined as a linear combination of the spanning functions and have to be zero in all point except one. On the octahedron the location of the nodal points are known.

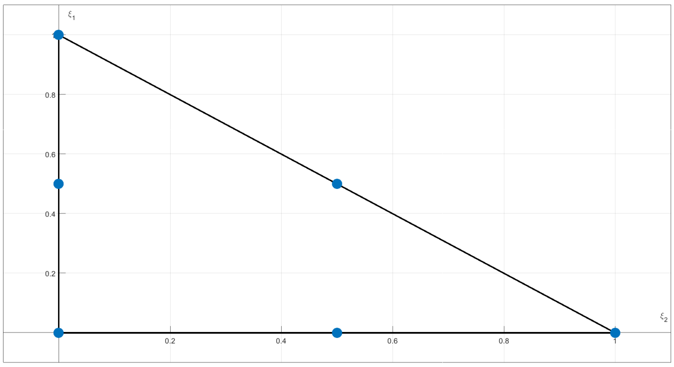


Figure 5: Nodal points for functions on the octahedron

For the functions defined in Omega the nodal points are not known. One of the sets used in this thesis is shown below, other sets of nodal points are listed in appendix A.

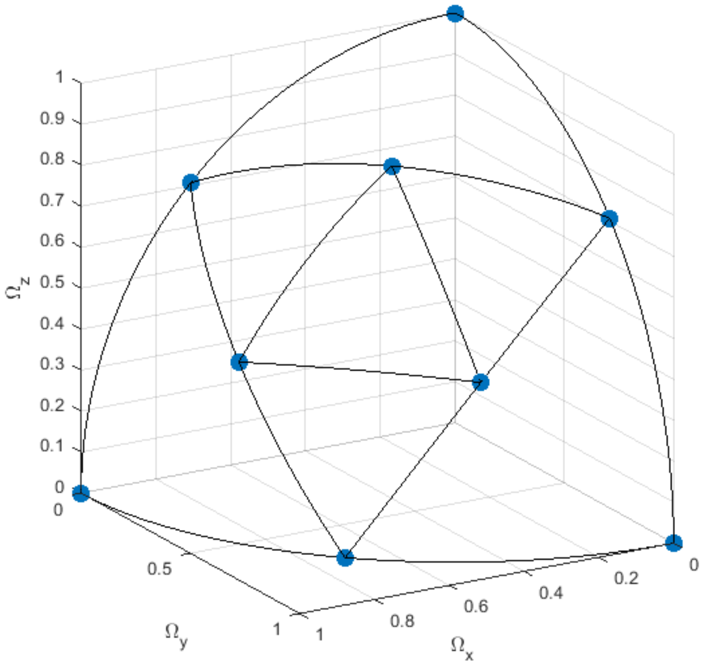


Figure 6: Nodal points for functions quadratic in Omega

$$\phi_i(P_j) = \delta_{ij} \tag{26}$$

Writing equation (26) gives the following matrix-matrix product

$$C[P_1, P_2, ..., P_n] = I^{n\times n} \tag{27}$$

The vectors P_i are defined on the basis b.
The positions of the points P define the matrix C according to

$$C = [P_1, P_2, ..., P_n]^{-1} \tag{28}$$

The resulting functions are shown in appendix B.

4.3 Orthogonalization

The orthogonalization method is specifically designed to construct a matrix C such that the corresponding mass matrix has a low condition number, in contrast to the nodal points where a low condition number is expected but not guaranteed. The orthogonalization is computationally more expensive to calculate and may not be required for sufficient accuracy.

The matrix will be well conditioned if the basis functions are orthogonal.

$$\langle \phi_i, \phi_j \rangle = A_{ij}\delta_{ij} \tag{29}$$

applying equation (10)

$$C_{il}\langle b_l, b_m \rangle C^T_{mj} = A_{ij}\delta_{ij} \tag{30}$$

This equation can be rewritten as a matrix multiplication

$$CM^0C^T = I \cdot \mathrm{diag}(A) \tag{31}$$

Where M^0 is a mass matrix with the coefficients $M^0_{lm} = \langle b_l, b_m \rangle$, and diag($A$) a vector containing the coefficients on the main diagonal of A

$$\mathrm{diag}(A) = \sum_i A_{ii}e_i \tag{32}$$

M^0 is symmetric positive definite (SPD) as shown below, $\forall w \neq 0$

$$\begin{aligned} w^T M^0 w &= \sum_i \sum_j w_i \langle b_i, b_j \rangle w_j \\ &= \sum_i \sum_j \langle w_i b_i, w_j b_j \rangle \\ &= \langle \sum_i w_i b_i, \sum_j w_j b_j \rangle > 0 \end{aligned}$$

Because M^0 is SPD, it can be written as a Cholesky decomposition, $A^{-1}M^0 = LL^T$. Substituting in equation (31) gives

$$CL\ L^TC^T = I \tag{33}$$

This equation is satisfied if the coefficient matrix equals $C = L^{-1}$.

If the matrix M^0 is not well conditioned, the calculation of ϕ is not very accurate. The process can be repeated by recalculating $M^{(n)} = \langle \phi_l^{(n)}, \phi_m^{(n)} \rangle$, where the superscript denotes $\phi^{(n)}$ is the result from the nth iteration. By calculating $M^{(n)}$ with a set of functions that is not the set of spanning functions b, the calculated matrix will be a correction on matrix used to get the set of functions $\{\phi^{(n)}\}$

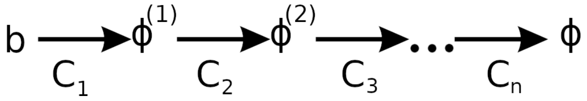


Figure 7: schematic representation of iterative orthogonalization

The new matrix $M^{(n)}$ is now better conditioned providing a more accurate calculation of ϕ. The coefficient matrix is as defined in equation (10) can be found by multiplying

$$C = C_n \cdots C_3 \cdot C_2 \cdot C_1 \tag{34}$$

The resulting basis functions are shown in appendix B.

5 Benchmarking

5.1 Benchmarking the solution

To test the accuracy of the code, the program is run with a source term such that the analytical solution is known. The simplest case of such a solution is when the solution φ is a (linear combinations of) spherical harmonic(s). This is because the spherical harmonics are eigenfunctions of the spherical Laplacian

$$\Delta_s Y_{lm} = -l(l+1)Y_{lm} \tag{35}$$

If φ is a spherical harmonic Y_{lm}, equation (7) reduces to

$$\left(\Sigma_a + \frac{\alpha}{2} l(l+1)\right) Y_{lm} = S \tag{36}$$

To determine the accuracy of the solution, the flux error is defined as

$$e_f \equiv \sqrt{\frac{\langle \varphi_h - \varphi, \varphi_h - \varphi \rangle}{\langle \varphi, \varphi \rangle}} \tag{37}$$

where φ is the true solution, and φ_h the numerical solution.

When the solution is not a spherical harmonic, the Laplacian has to be calculated explicitly by hand. For octahedron functions this is not very difficult. According to equation (15) the spherical gradient of an octahedron function is identical to the Euclidean gradient, therefore the spherical Laplacian is identical to the Euclidean Laplacian with respect to Ω.

5.2 Benchmarking intermediate solutions

In the code there are a lot of tests built in to spot a mistake during the execution of the code. Some of them will be discussed here.

5.2.1 Test function values and derivatives

The basis functions are very important in the code. To test if the values are correctly calculated the following tests are implemented:

In the code there is a section which calculates the function values and derivatives of the basis functions for certain values of $\mathbf{\Omega}$. The test generates a semi random vector $\mathbf{\Omega}_1$ on the sphere and a small deviation $\delta\mathbf{\Omega}$, with $\delta\mathbf{\Omega}$ such that $\mathbf{\Omega}_2 = \mathbf{\Omega}_1 + \delta\mathbf{\Omega}$ is on the sphere.
The vector $\mathbf{\Omega}_1$ is not truly random[4] because $\mathbf{\Omega}_1$ has to be on the angular element. To calculate a random $\mathbf{\Omega}_1$ on the angular element, a random linear combination of the vector v_1, v_2, v_3 is normalized to the unit sphere, where

[4] ignoring computer pseudo-randomness

the vectors v_1, v_2, v_3 are the vertices of the angular element. The value for $\mathbf{\Omega}_2$ is also restricted to the angular element
To test if the function values and gradients are correctly calculated the gradient in direction of $\hat{\delta\mathbf{\Omega}}$ is approximated with finite difference and compared against the analytical value.

$$f|_{\mathbf{\Omega_2}} - f|_{\mathbf{\Omega_1}} \approx \delta\mathbf{\Omega} \cdot \nabla f|_{\mathbf{\Omega_1}} \tag{38}$$

The right hand side of the equation is the slope in direction of $\hat{\delta\Omega}$ with step size $||\delta\Omega||_2$. The approximate error of this equation can be determined by Taylor expansion

$$f(x+h) = f(x) + f'(x)h + \frac{f''(x)}{2}h^2 + \mathcal{O}(h^3) \tag{39}$$

When $\mathcal{O}(h^3)$ is assumed negligible, the approximate error is define as

$$error = \frac{f''(x)}{2}h^2 \approx \frac{f'(x+h) - f'(x)}{2}h \tag{40}$$

translating the error in terms of gradients gives

$$error = \frac{\delta\mathbf{\Omega} \cdot \nabla f|_{\mathbf{\Omega_2}} - \delta\mathbf{\Omega} \cdot \nabla f|_{\mathbf{\Omega_1}}}{2} \propto ||\delta\Omega||_2^2 \tag{41}$$

This process is iterated over 100 times for each angular element. To make sure the edges and corners behave well, the vector $\mathbf{\Omega_1}$ is sometimes not placed randomly. On every angular element the vector $\mathbf{\Omega_1}$ is placed in every corner ones and placed on each edge randomly 10 times.
If an error in equation (38) is too large an error message is displayed.

The second test is to test if the spherical gradient is perpendicular to the vector $\mathbf{\Omega_1}$.

$$\mathbf{\Omega_1} \cdot \nabla f|_{\mathbf{\Omega_1}} = 0 \tag{42}$$

If an error in equation (42) is too large an error message is displayed.

5.2.2 Test angular quadrature set

To calculate integrals with the computer, the program uses a special type of gauss quadrature. This quadrature is specifically designed to solve polynomials on spherical surfaces. The quadrature sets cannot easily be extended. To test if the quadrature set is sufficient to calculate the integrals the following test is implemented.

The mass matrix is calculated with two different quadrature sets. The difference of the two matrices should be zero if both quadrature sets are sufficient. The code actually calculating the solution uses the best quadrature set available, and is in this test compared against an inferior one. Therefore if this test fails, it is only an indication of a possible error. if the test fails a warning message is displayed.

5.2.3 Test solution of a linear system of equations

In the program there are a lot systems of equations in the form of $Ax = b$. To test the accuracy of these solutions, tests calculating $Ax - b = 0$ are scattered throughout the code. If the error in one of these tests is out of bounds an error messages and location is displayed.

5.2.4 Test to check variable restrictions

Some quantities have restrictions on them dictated by underlying mathematics. For example $\xi = [\xi_1, \xi_2]^T$ carries the restriction $\{\xi_1 \geq 0, \xi_2 \geq 0, \xi_1 + \xi_2 \leq 1\}$. If such a restriction is violated an error message is displayed.

5.2.5 Test to check subroutine errors

Some subroutines in Fortran have error detection built in. For example: the Cholesky decomposition returns an error if the matrix is not symmetric positive definite. If an error is found an error message is displayed.

6 Results

6.1 Extended solution space

The first property to check is the extension of the solution space as explained in section 2.3.

The program is configured to solve equation (7) with only the macroscopic absorption cross section term on the left hand side.

$$\Sigma_a \varphi = S$$

The solution is a constant factor difference from the source, therefore it can always be found by the program in a single step.
Figure 8 shows the calculated flux error for different orders of angular solutions. In this graph all angular solutions are defined on the unit sphere, therefore only basis sets defined in Omega are shown.

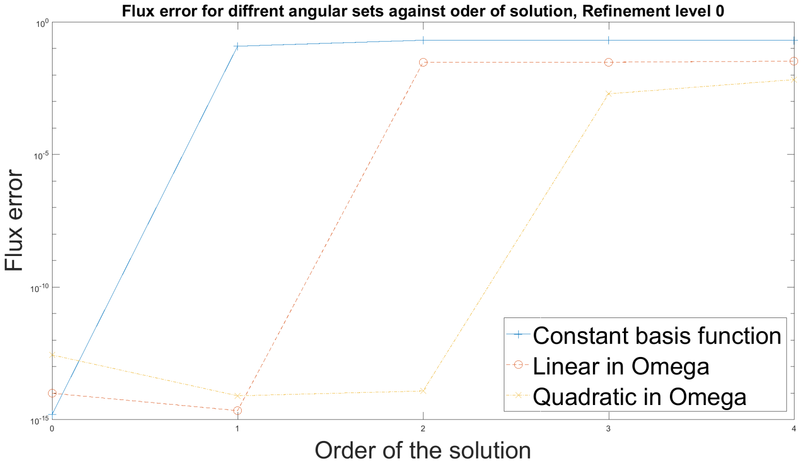


Figure 8: Flux error of different angular sets defined on the unit sphere against the order of the solution

The angular solutions used are shown below and labeled by their highest order:

0: $\varphi = \frac{1}{4\pi}$

1: $\varphi = 5 + \Omega_x + 2\Omega_y$

2: $\varphi = 5 + \Omega_x + 2\Omega_y + (3\Omega_x^2 - 1)$

3: $\varphi = 5 + \Omega_x + 2\Omega_y + (3\Omega_x^2 - 1) + \Omega_x\Omega_y\Omega_z$

4: $\varphi = 5 + \Omega_x + 2\Omega_y + (3\Omega_x^2 - 1) + \Omega_x\Omega_y\Omega_z + \Omega_x\Omega_y(\Omega_x^2 - \Omega_y^2)$

Figure 8 shows the constant basis set can only exactly represent solutions of order 0. The set linear in Ω can exactly represent solution up to order 1, and the set quadratic in Ω can represent solutions up to order 2 exactly.

The procedure above can be repeated for octahedron functions by first projecting all the omega in the set of angular solutions above, to the octahedron with equation (16) and recalculate the flux errors with the basis sets defined on the octahedron.

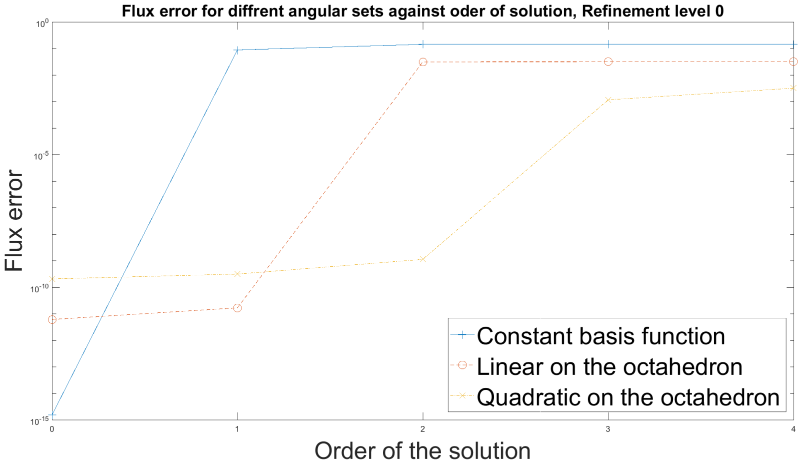


Figure 9: Flux error of different angular sets defined on the octahedron against the order of the solution

In the graph, the same pattern shows as with the functions on the sphere.

6.2 Convergence due to mesh refinement

The following test shows how the basis sets behave under mesh refinement. This test still uses equation (7) without the Fokker-Planck operator. The angular solution used is of order 3 and defined on the unit sphere. This solutions chosen on the unit sphere because calculating the Laplacian is mathematically easier, and the order is chosen to make sure even the set quadratic in Omega cannot represent this solution exactly.

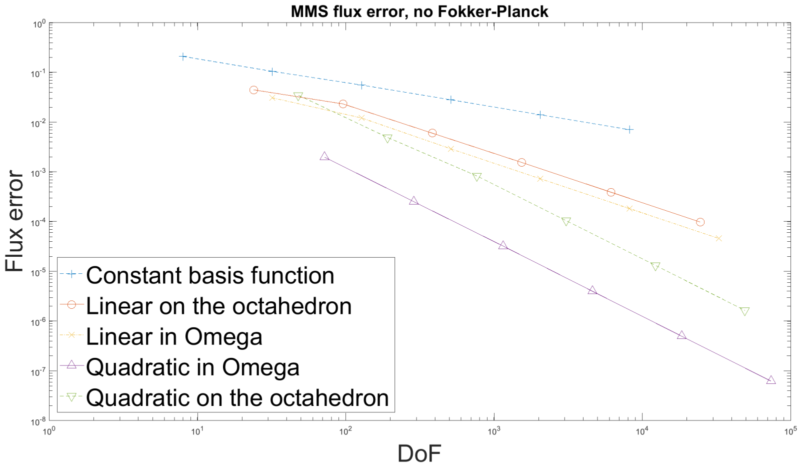


Figure 10: Convergence of angular sets under refinement without angular diffusion

Figure 10 shows nice convergence. This is no surprise because the program only has to project the analytical solution to the solution space provided by the basis set. This figure also shows the increased order of convergence for the bigger basis sets.
The order of convergence is defined as $\epsilon \backsim h^n$, where ϵ represents the flux error, and h a characteristic length and n the order of convergence. In one refinement step, every triangle is cut into 4, therefore the characteristic length is divided in half. When the triangle is refined the number of degrees of freedom is increased by a factor of 4. The number of degrees of freedom is related to the characteristic length by $h \backsim (DoF)^{-\frac{1}{2}}$. The order of convergence can be expressed in terms of DoF as follows

$$\epsilon \backsim (DoF)^{-\frac{1}{2}n} \tag{43}$$

The order of convergence is of a basis set is -2 times the angle in figure 10

Basis set	angle in figure 10	order of convergence
Constant	-0.5	1
Linear in Omega	-1	2
Linear on the octahedron	-1	2
Quadratic in Omega	-1.5	3
Quadratic on the octahedron	-1.5	3

6.3 Convergence due to mesh refinement with Fokker-Planck

When using all terms in equation (7) the constant basis set is known to not work and is therefore left out.

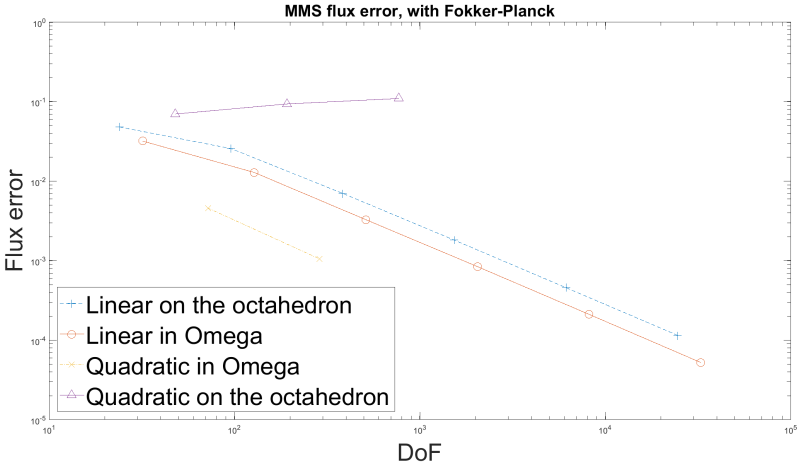


Figure 11: Convergence of angular sets under refinement with angular diffusion

The basis sets quadratic in Omega and quadratic on the octahedron could not be plotted for refinement levels higher than 1 and 2 respectively because the solver did not converge and therefore could not calculate the solution.
The basis set quadratic in Omega shows a predictable pattern for the two data points collected, however the basis set quadratic on the octahedron does not show a convergence pattern. The cause of this is unknown

7 Discussion

The main problem is the absence of convergence for the basis sets quadratic in Omega and quadratic on the octahedron. The direct cause for this absence is not known, but there is strong evidence for one possible cause for the basis set quadratic in Omega.
One of the most important calculations for calculating the basis set quadratic in Omega, is the calculation of the coefficient matrix C. When comparing a typical matrix C calculated with the basis set linear in Omega to a matrix C as calculated with the basis set quadratic in Omega, a pattern emerges.

Table 1: Typical coefficient matrix C for basis set linear in Omega, for refinement level 5

-622.16271	-619.16512	-621.16271	1862.49054
-15.31629	-35.54905	5.08836	45.77698
-25.46425	-66.10160	-45.81971	137.38556
622.66276	622.66276	622.66276	-1866.98828

Table 2: Typical coefficient matrix C for basis set quadratic in Omega, for refinement level 5

1772509	-828800	2521392	160364	-786096	-50034
1873372	-849016	2671544	199006	-807153	-60165
6271125	-2997634	8888304	728156	-2832977	-231493
-22434343	10659865	-31825413	-2541259	10084169	798154
-13451123	6328397	-19110043	-1407214	5994003	441577
-13652244	6368616	-19409546	-1484387	6037861	461977
24653222	-11897207	34916034	2755170	-11236281	-878648
-10491515	5274460	-14768003	-1473798	4954782	478597
25458998	-12058682	36115731.	3063961	-11408309	-959964.

The coefficient matrix for the basis set quadratic in Omega, only shows 5 columns to prevent it from running of the page. The matrix for the basis set quadratic in Omega contains much higher values, combining these high values with the finite precision of the computer cause errors build up. This could cause the solver not to converge. The nodal point method has no way to make the coefficients of the matrix smaller, except for finding a different configuration of points to use to generate the matrix. Orthogonalization does not lead to a matrix with much lower values in the C matrix. Coefficient matrices for different nodal point configurations are listed in appendix A.

Appendix A also lists two coefficient matrices for the basis set linear in Omega. The coefficients for a matrix with the basis set linear in Omega is expected to contain lower values because there are less points in the same area. To accommodate this effect the coefficient matrix of the set linear in Omega, for two refinement level higher is also shown. In this set the average distance between nodes is comparable to the basis set quadratic in Omega. The set linear in Omega still has lower coefficients than the set quadratic in Omega.

8 Conclusions

From the results, it can be seen the discretization is executed correctly and the solution space has been extended accordingly. Problems start to arise when the angular diffusion is added to the equation. The basis set linear in Omega and linear on the octahedron behave well with the angular diffusion enabled, but the extended quadratic basis functions do not work well with angular diffusion. For the basis set quadratic in Omega, the most likely cause are big values in the coefficient matrix C. These big values combined with the finite precision of the computer gives rise to a build up of errors in the function values and the gradients of the basis functions. These errors probably cause the solver not to converge. The basis set quadratic on the octahedron does not have this problem since the coefficient matrix C is defined as the identity matrix as calculated with the nodal point method. This basis set does seem to converge, but the rate of convergence is slow enough to deem it practically unusable. If the quadratic basis sets can be made to work, an order of convergence of 3 can be expected.

8.1 Future Work

In the future someone could try to find a method of creating the coefficient matrix C with lower coefficients if that is possible. To properly implement these basis sets the rate of convergence needs to increase. Solving this probably requires more knowledge of the sweep and in depth knowledge of the Galerkin method. This seems a lot of work for limited results, as the sets linear in Omega and linear on the octahedron work well. If the speed of convergence cannot be increased it is not worth to investigate further as it is too slow.

References

[1] E. Cheney and D. Kincaid. *Numerical Mathematics and Computing*. International student edition. Cengage Learning, 2007.

[2] C.D.H. Chisholm. *Group theoretical techniques in quantum chemistry*. Theoretical chemistry : a series of monographs. Academic Press, 1976.

[3] J.J. Duderstadt and L.J. Hamilton. *Nuclear Reactor Analysis*. Wiley, 1976.

[4] Ralf Hartmann. Numerical analysis of higher order discontinuous Galerkin finite element methods, 2008.

[5] Aldo Hennink. A discontinuous Galerkin method for charged particle transport in the Fokker-Planck limit. Master's thesis, TU Delft, the Netherlands, 2015.

[6] Aldo Hennink and Danny Lathouwers. A discontinuous Galerkin method for the mono-energetic Fokker-Planck equation based on a spherical interior penalty formulation. *Journal of Computational and Applied Mathematics*, 330:253 – 267, 2018.

[7] Heiner Igel. Lectures on computational seismology. Department of Earth and Environmental Sciences, Ludwig-Maximilians-University, Munich, July 2010.

[8] József Kópházi and Danny Lathouwers. A space-angle DGFEM approach for the Boltzmann radiation transport equation with local angular refinement. *Journal of Computational Physics*, 297:637 – 668, 2015.

Appendix A: Nodal points and coefficient matrices

This Appendix contains several sets of nodal points and a coefficient matrix C that has been generated with these nodal points. The coefficient matrices have been made with refinement level 3, unless otherwise specified. All set of nodal points contain the same three points, on $\mathbf{\Omega} = [1, 0, 0]^T, [0, 1, 0]^T, [0, 0, 1]^T$. These node will be referred to as cardinal nodes.

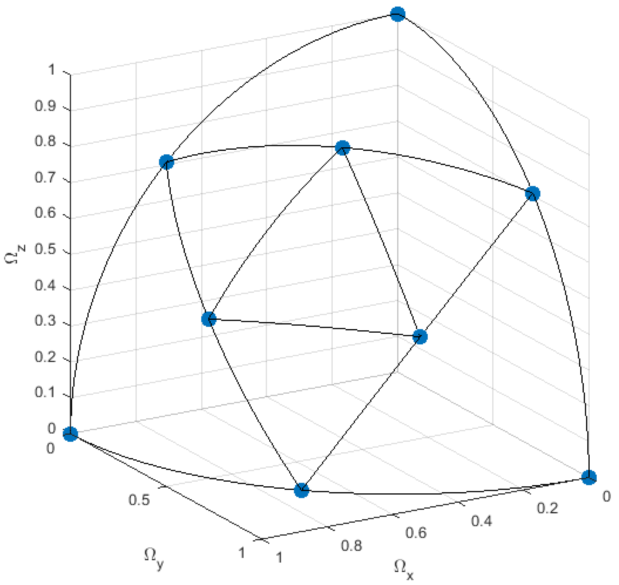


Figure 12: Nodal points for functions quadratic in Omega

This set of nodal points is constructed by starting with the three cardinal nodes. The next three nodes bisect the great circle segments connecting the cardinal nodes. The last three bisect the great circle segments connecting the previous nodes.

Table 3: Coefficient matrix corresponding to the above set of nodal points

18123	2866	13283	-23622	1391	-2501	-11569	2862	-1551
6528	383	4514	-8694	179	-339	-4016	1113	-513
7949	1034	5063	-10763	441	-936	-4570	1435	-522
-47817	-6269	-32568	63790	-2915	5542	29090	-8190	3505
-67662	-9861	-48656	88907	-4445	8796	42793	-10989	5583
-45094	-4995.	-31535	59803	-2357	4429	27876	-7582	3660
76750	13833	54402	-101038	6280	-12298	-48017	12538	-5953
65616	8716	50102	-84799	4124	-7700	-43360	10034	-6221
-14391	-5708	-14605	16415	-2699	5006	11773	-1221	2012

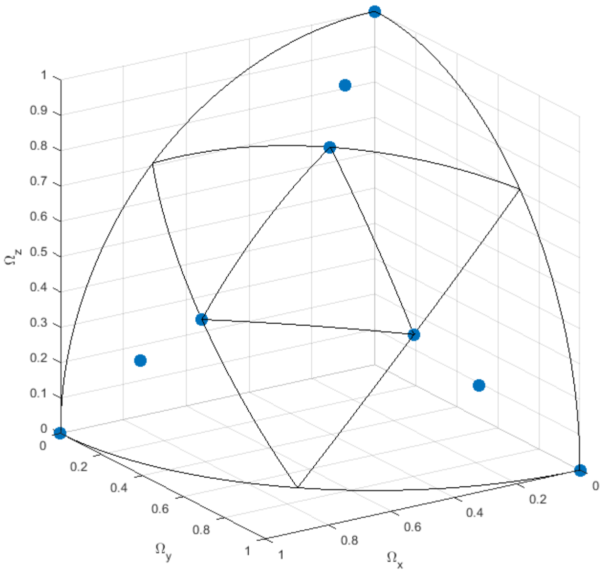


Figure 13: Nodal points for functions quadratic in Omega

This set is constructed by making the previous set and moving the nodes as shown in the figure to the position where they bisect the great circle segment connecting a cardinal node to its nearest neighbor.

Table 4: Coefficient matrix corresponding to the above set of nodal points

11181	6535	6535	-14346	2129	-5780	-5780	1643	0
5725	1401	2619	-8122	500	-1373	-2424	1205	-226
5725	2619	1401	-8122	500	-2424	-1373	1205	226
-58891	-35398	-35398	75225	-11013	31504	31504	-8462	-0
-26585	-4856	-12117	38263	-2182	5124	11097	-5858	1584
-26585	-12117	-4856	38263	-2182	11097	5124	-5858	-1584
15583	-2776	6885	-24089	374	1544	-6192	4226	-2260
15583	6885	-2776	-24089	374	-6192	1544	4226	2260
58264	37707	37707.	-72983	11498	-33502	-33502	7675	0

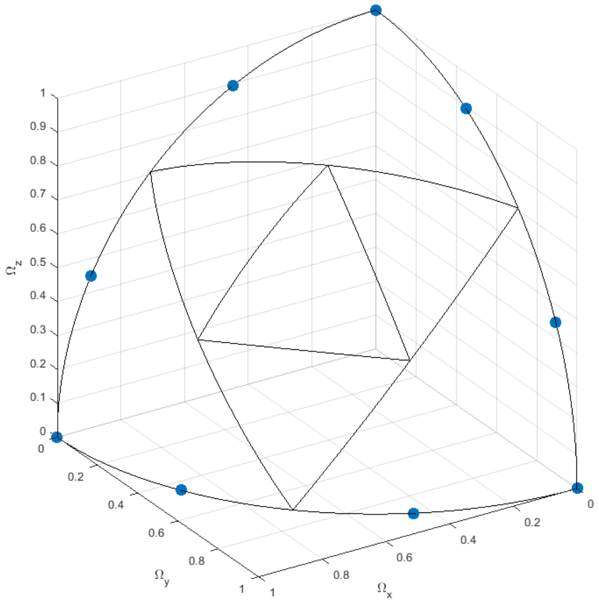


Figure 14: Nodal points for functions quadratic in Omega

This set is constructed by adding nodes at location where they trisect the great circle segment connecting cardinal nodes.
For this set of nodal points the coefficient matrix could not be calculated as some coefficients became infinitely large.

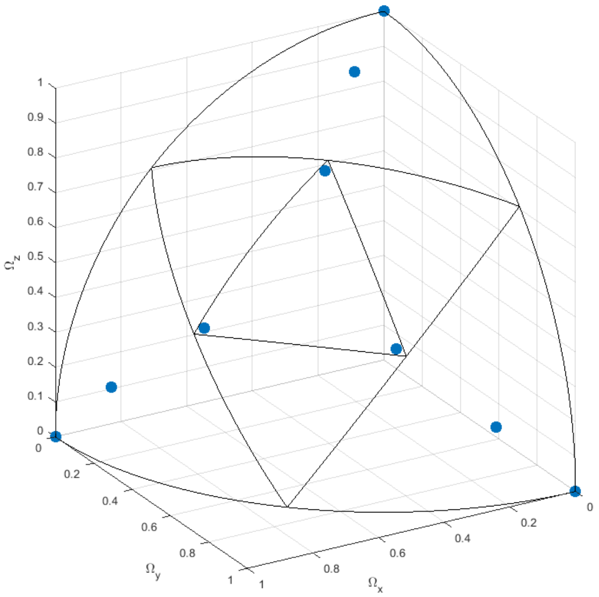


Figure 15: Nodal points for functions quadratic in Omega

This set is constructed by adding nodes at locations where they trisect the great circle segment connecting cardinal nodes to a virtual node in the middle. This virtual node is the same distance from all the cardinal nodes.

Table 5: Coefficient matrix corresponding to the above set of nodal points

8256	11065	1236	-5696	1005	-5224	-748	-494	2369
16881	25415	73	-5028	476	-5372	867	-1987	6455
8602	11922	3564	-3901	3320	-3496	-1219	-718	2658
-20884	-27659	-3150	15379	-2383	14114	2258	1189	-5739
9669	11537	1526	-10509	674	-9449	-2071	-188	1833
-46670	-70650	322	13414	-1056	14638	-3080	5601	-18065
35367	54493	-1949	-8533	-400	-10035	3771	-4537	14237
-21913	-30222	-10100	10022	-9385	8783	3633	1725	-6761
10692	14098	8476	-5150	7749	-3959	-3412	-591	3013

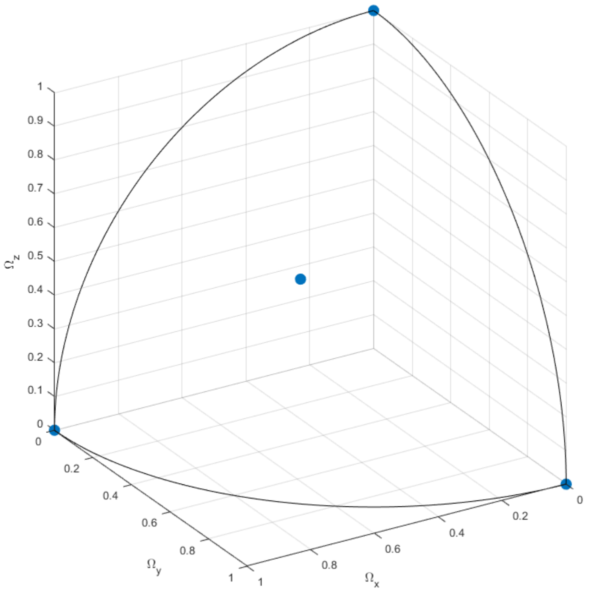


Figure 16: Nodal points for functions linear in Omega

This set is constructed by making a node with equal distance to each cardinal node.

Table 6: Coefficient matrix corresponding to the above set of nodal points

0.92009	15.16263	35.62795	38.04747
9.29174	18.72381	32.04576	38.04747
3.67784	23.44498	30.09387	38.04747
-13.88967	-57.33141	-97.76758	-113.14240

Table 7: Coefficient matrix corresponding to the above set of nodal points, made with refinement level 5

-257.94597	-556.24610	-69.64681	616.88085
-277.22339	-549.33876	-47.45611	616.88085
-241.07101	-567.52872	-35.25173	616.88085
776.24037	1673.11358	152.35464	-1849.64255

Appendix B: Basis functions

Nodal point method

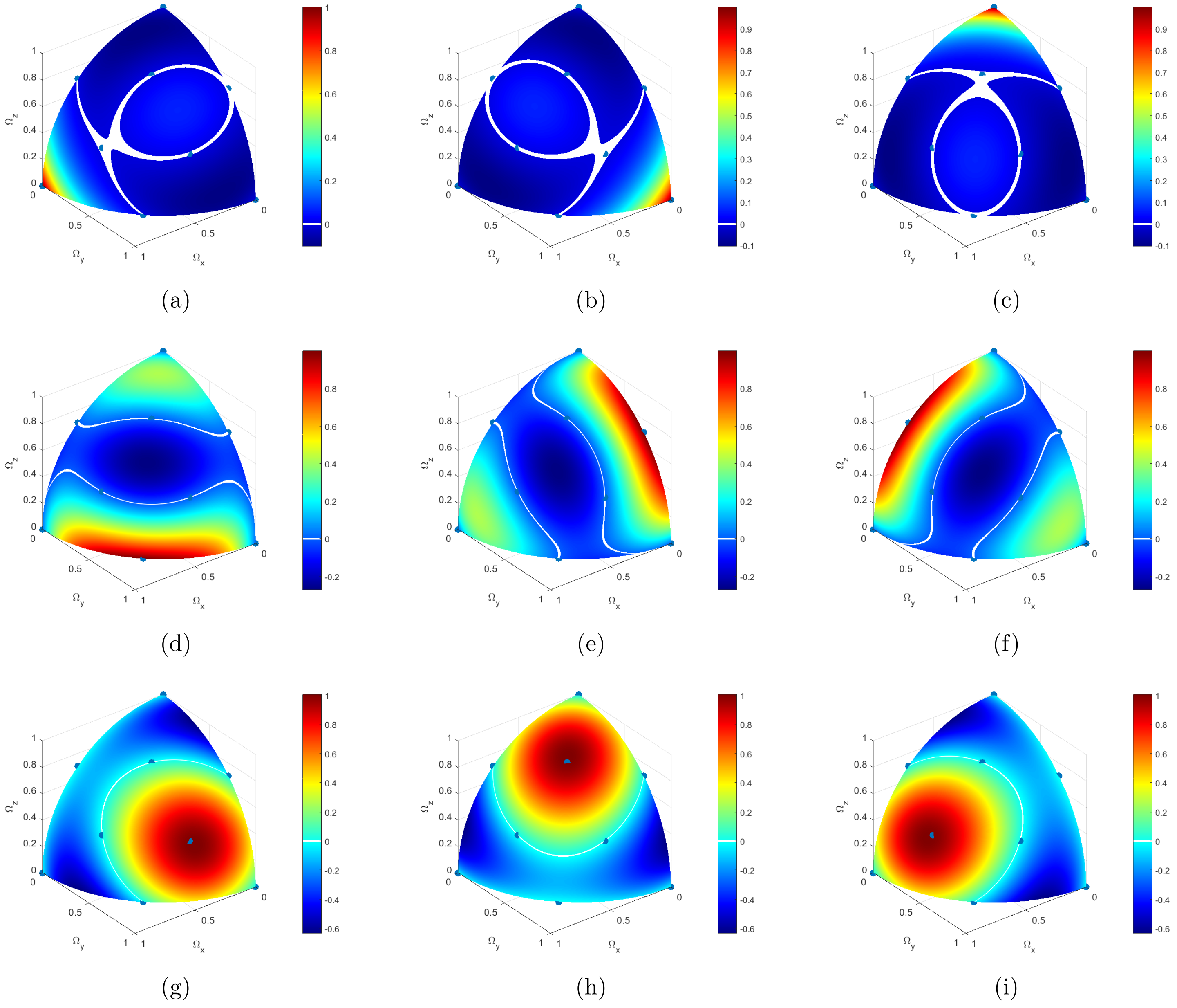
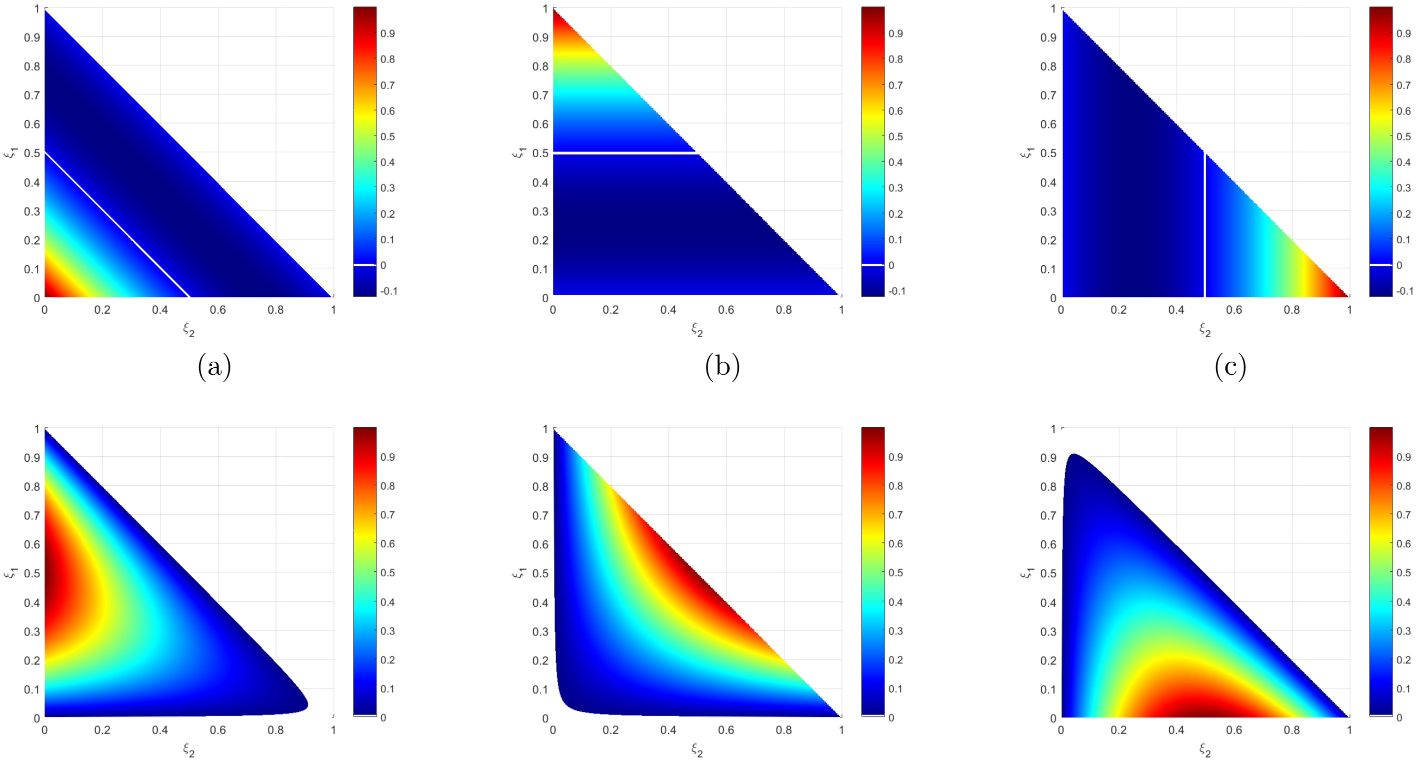


Figure 17: Basis functions in Omega

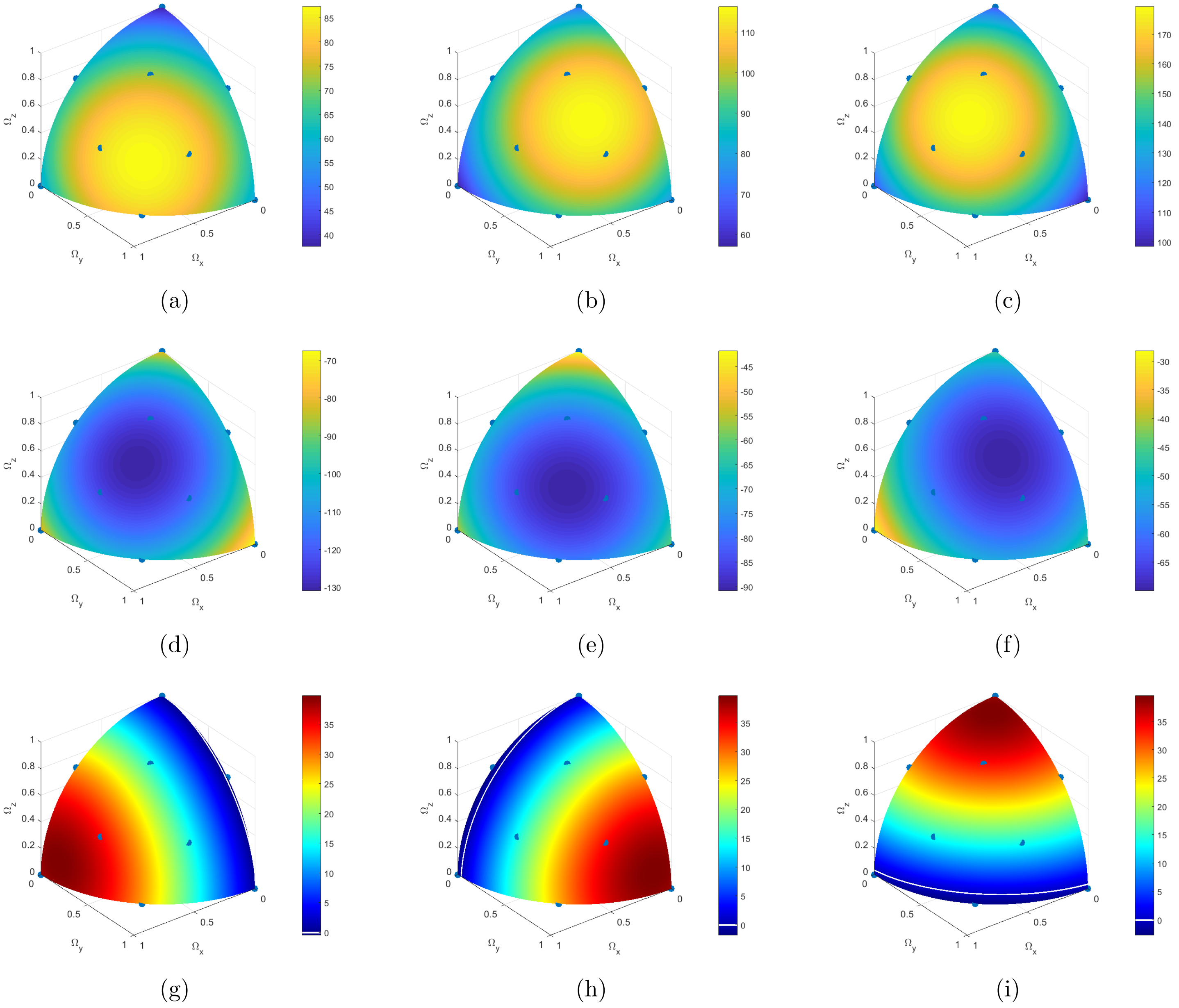


(a) (b) (c)

(d) (e) (f)

Figure 18: Basis functions on the octahedron

Orthogonalization

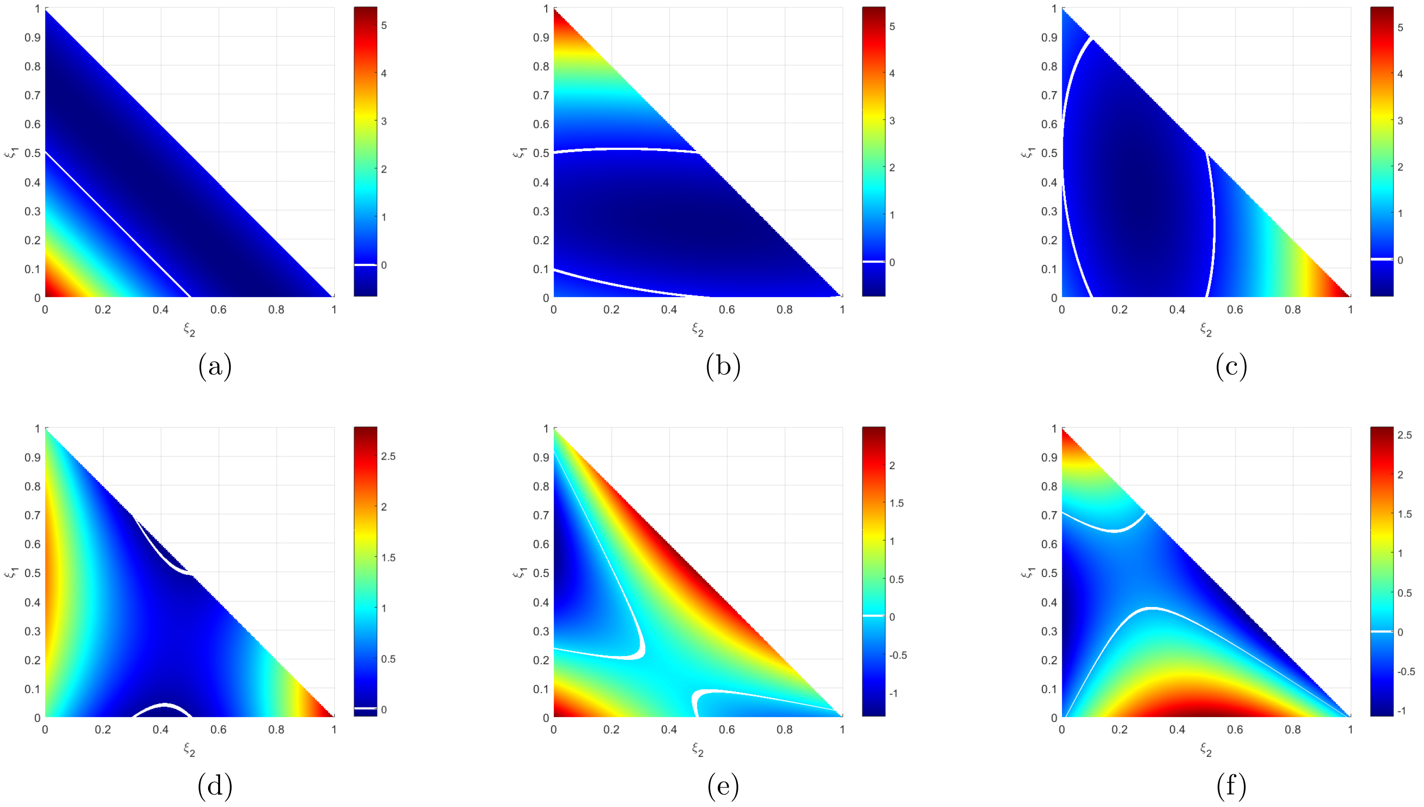


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 19: Orthogonalized basis functions in Omega



(a) (b) (c)

(d) (e) (f)

Figure 20: Orthogonalized basis functions on the octahedron