The University of Nottingham
UNITED KINGDOM · CHINA · MALAYSIA

Bennison, Tom (2014) Adaptive discontinuous Galerkin methods for the neutron transport equation. PhD thesis, University of Nottingham.

Access from the University of Nottingham repository:
http://eprints.nottingham.ac.uk/28944/1/corrected_thesis_main.pdf

Copyright and reuse:

The Nottingham ePrints service makes this work by researchers of the University of Nottingham available open access under the following conditions.

- Copyright and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners.
- To the extent reasonable and practicable the material made available in Nottingham ePrints has been checked for eligibility before being made available.
- Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.
- Quotations or similar reproductions must be sufficiently acknowledged.

Please see our full end user licence at:
http://eprints.nottingham.ac.uk/end_user_agreement.pdf

A note on versions:

The version presented here may differ from the published version or from the version of record. If you wish to cite this item you are advised to consult the publisher's version. Please see the repository url above for details on accessing the published version and note that access may require a subscription.

For more information, please contact eprints@nottingham.ac.uk

Adaptive Discontinuous Galerkin Methods for the Neutron Transport Equation

Thomas A. J. Bennison

Thesis submitted to The University of Nottingham
for the degree of Doctor of Philosophy

July 2014

Abstract

In this thesis we study the neutron transport (Boltzmann transport equation) which is used to model the movement of neutrons inside a nuclear reactor. More specifically we consider the mono-energetic, time independent neutron transport equation.
The neutron transport equation has predominantly been solved numerically by employing low order discretisation methods, particularly in the case of the angular domain. We proceed by surveying the advantages and disadvantages of common numerical methods developed for the numerical solution of the neutron transport equation before explaining our choice of using a discontinuous Galerkin (DG) discretisation for both the spatial and angular domain.
The bulk of the thesis describes an arbitrary order in both angle and space solver for the neutron transport equation. We discuss some implementation issues, including the use of an ordered solver to facilitate the solution of the linear systems resulting from the discretisation. The resulting solver is benchmarked using both source and critical eigenvalue computations. In the pseudo three–dimensional case we employ our solver for the computation of the critical eigenvalue for three industrial benchmark problems. We then employ the Dual Weighted Residual (DWR) approach to adaptivity to derive and implement error indicators for both two–dimensional and pseudo three–dimensional neutron transport source problems. Finally, we present some preliminary results on the use of a DWR indicator for the eigenvalue problem.

Acknowledgements

Firstly my thanks go to my supervisors; Paul Houston, Andrew Cliffe and Edward Hall. Paul, thank you for many stimulating meetings where we discussed the difficulties of the neutron transport equation and for your personal support throughout my studies. Andrew, I am very grateful that despite having more important things to worry about you made time to answer some of my most esoteric questions. Ed, thank you for your many hours discussing the pleasures of FORTRAN with me and all the help and advice that you gave me - I am proud to now count you as a close friend. Without the input of my supervisors this thesis would not have come in to existence and I will always feel privileged to have worked with you.

Thank you also to David Baker and Fynn Scheben who were also studying the neutron transport equation; I had many productive discussions with both of you. I would also like to acknowledge the support of fellow PhD students in the SCA group (Joe Collis, Scott Congreve, Steven Murphy and Nathan Sime) and Phil Ingrey for keeping me sane with lots of "geeky" conversations about technology and other random topics during the many coffee breaks during my PhD. Thank you also to all members of the Maths Cake Club for keeping me supplied with cake during the write up. I would also like to acknowledge Dave Parkin for dealing with what seemed like many server related questions and all the administrative staff in the department for making the department a nice place to work.

I would also like to thank my Mum and Sister for encouraging me to apply for a PhD in the first place.

Finally, my greatest thanks go to my wife Kirsty, whose love has been an endless encouragement. She has supported me always and put up with being ignored while I think about some coding right through to the end.

Contents

CHAPTER 1

Introduction

1.1 Background

As stated in a recent report by Richard Lester and Robert Rosner [103] "Many countries around the world are taking a fresh look at nuclear power". This remains true, even after the incident at the Japanese Fukushima Daiichi plant in March 2011. In addition to the long established nations with a stake in nuclear power, for example; the UK, France and the USA where there is a requirement to replace ageing reactors, developing countries such as China and India are also aiming to add to their generating capacity.

Energy use is growing, and the decisions governments make with regard to the energy supply are incredibly complex. To understand why there is a renewal of interest in nuclear power it is useful to consider the main factors; climate change, security of supply and economics. Currently, climate change is an important factor in government decisions in addition to ensuring a security of supply in a world that seems increasingly turbulent. Since the world's population is increasing and large developing countries such as China and India are embracing a more energy intensive 'Western' lifestyle it is increasingly difficult to balance these competing factors.

Coal, the traditional fuel used for electricity generation, whilst remaining fairly cheap is still, in terms of $C0_2$ and other waste gases, the dirtiest way to produce power, despite many improvements in generation technology and treatment of the waste gases. Oil and gas are running out, and also subject to large price spikes due to the the political instability of many oil and gas producing countries in the Middle East. The development of fracking techniques will go some way to reducing the reliance on fossil fuels from the Middle East by opening up new sources of domestic gas, however, the safety of this extraction technique remains to be proven. All European governments however,

have signed up to increase the percentage of renewable energy production when producing power to 20% by 2020 (the so called 20:20:20 target) and so there is a further reason to focus on renewable sources of energy. Renewable technologies such as solar power, wind turbines and tidal power are all too unpredictable to be used to provide a reliable base load for the national grid. In contrast to this, nuclear power provides a reliable and safe source of power that falls in to the renewable category; there have been very few nuclear accidents since the first commercial nuclear power plant began operation in the UK in 1956. In addition, the chief raw material for nuclear power generation, uranium, is now known to be available in large quantities and is mined in more stable areas of the world [164]. The cost of Uranium also contributes a relatively small percentage to the total cost of nuclear power generation; this means there is an inbuilt safeguard against price fluctuations.

All of these factors have led the UK government to investigate new nuclear power stations as a way to provide low carbon energy, stating in [63] "[the UK Governement] believes it is in the public interest that new nuclear power stations should have a role to play in this country's future energy mix alongside other low carbon sources".

Despite all the above scientific, economic and environmental considerations in favour of an increased number of nuclear power generators it is also important to gain public acceptance of nuclear power. In the UK, even after the Fukushima incident, 42% of the British public support the building of new nuclear power stations, see Figure 1.1, compiled from data collected by IpsosMori on behalf of the Nuclear Industry Association (NIA). The impact of public opinion can clearly be seen in Germany's recent commitment to decommission all nuclear power plants, just after a publicised U turn on that policy.

Modern reactors are incredibly complicated, (see Figure 1.2 for a diagram of a nuclear reactor core) and to gain public acceptance of nuclear power we need to be able to prove that they can be operated reliably and safely over a timescale of several decades.

Mathematically we model the processes occurring inside a nuclear reactor's core by the Neutron Transport Equation (also referred to as the Boltzmann Transport Equation).

Figure 1.1: IpsosMori survey data.

1.2 The Neutron Transport Equation

When modelling the behaviour of neutrons inside a reactor, we first have to decide on an appropriate characterisation of the neutron. Since a neutron is an atomic particle, one option is to consider a quantum mechanical model; we could also consider it as a relativistic particle since its mass varies over time when its speed approaches that of light. Fortunately (since if this wasn't possible we would have to consider a fully relativistic model) when a neutron is moving and not colliding with anything else it can be modelled as a ballistic particle; in this case we can couple this ballistic behaviour with some quantum mechanics to characterise the collisions with other nuclei or neutrons.

The neutron transport equation is an integro–differential equation posed in seven dimensions; 3 spatial dimensions x, y, z, 2 angular dimensions φ, θ, time t and energy E. To state the neutron transport equation we first define the neutron density $N(\boldsymbol{x}, \boldsymbol{\mu}, E, t)$ to be the number of neutrons per unit volume at position $\boldsymbol{x} \in \mathbb{R}^3$ that move at time t in direction $\boldsymbol{\mu} \in \mathbb{S}^2 := \{\boldsymbol{\mu} \in \mathbb{R}^3 : \|\boldsymbol{\mu}\|_2 = 1\}$ with energy E. Generally, the physical quantity that is used in the expression of the neutron transport equation is the angular flux $\psi(\boldsymbol{x}, \boldsymbol{\mu}, E, t)$ which is obtained from the neutron density in the following fashion

$$\psi(\boldsymbol{x}, \boldsymbol{\mu}, E, t) = v(E)N(\boldsymbol{x}, \boldsymbol{\mu}, E, t),$$

where v denotes the speed of the neutron under consideration.

Remark 1. *It is convenient to define $\boldsymbol{\mu}$ using the spherical polar coordinate system as*

$$\boldsymbol{\mu} = (\sin\theta\cos\varphi, \sin\theta\sin\varphi, \cos\theta)^\top, \qquad \theta \in [0, \pi], \quad \varphi \in [0, 2\pi),$$

where θ is the polar angle and φ the azimuthal angle. It can be seen that this satisfies the requirement $\|\boldsymbol{\mu}\|_2 = 1$.

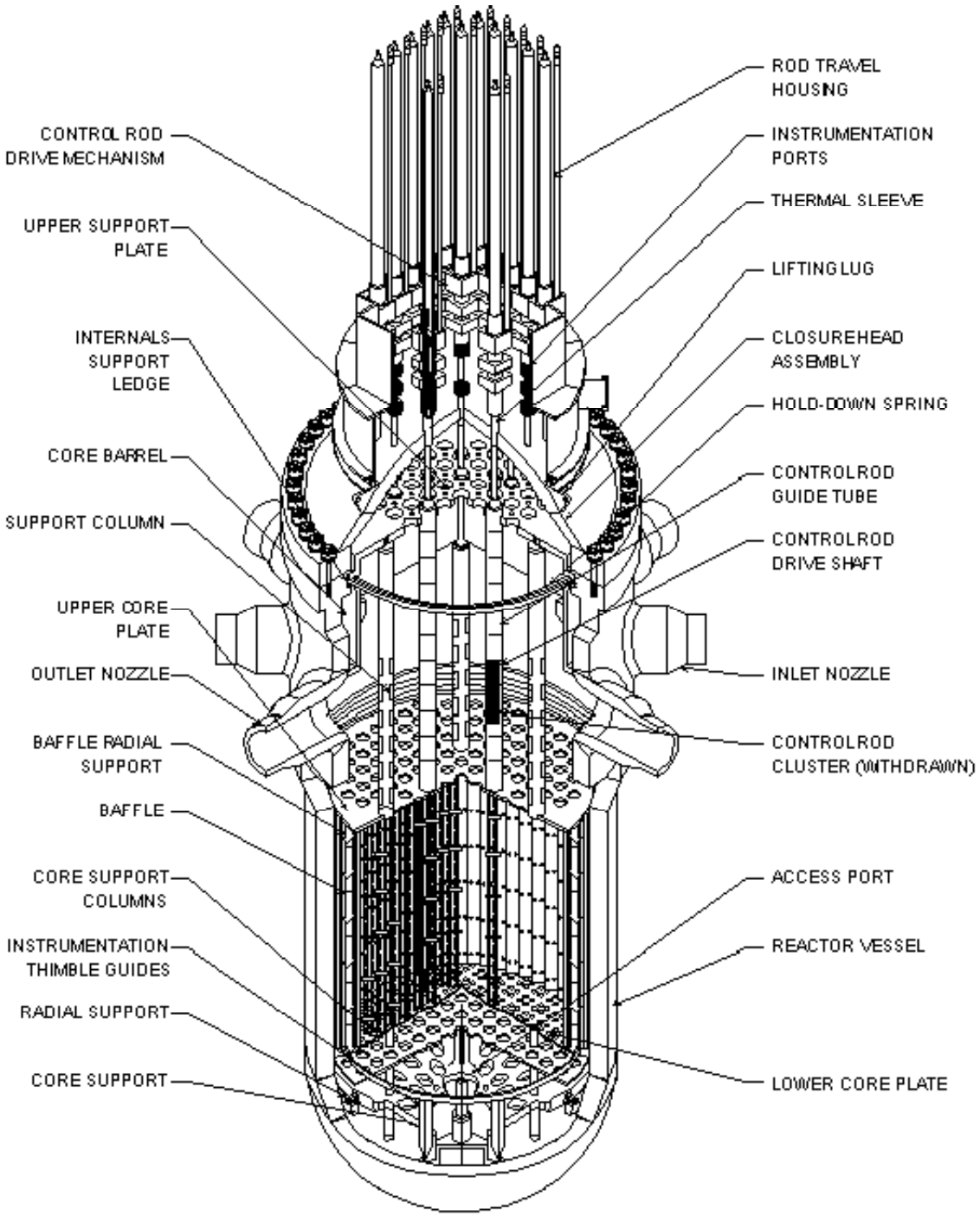


Figure 1.2: Pressurised Water Reactor Core

In modelling the transport of neutrons through a material we make the following assumptions.

(a) **Neutron as a Point Particle.**

Taking the neutron as a point particle means that we consider it as a particle which can be completely described by its position and velocity; thereby, we ne-

glect any effects due to polarization. Polarization can occur since each neutron has a spin and magnetic moment associated with it; this polarization affects the subsequent scattering of the neutron. Transport theory which takes into account polarization has in fact been developed, see for example, [35]. In principle there are cases where the effects on neutron transport could be large (for example, fast neutrons diffusing in helium), though in most practical situations these effects are quite minimal.

(b) **The Expected Value.**
The neutron transport equation expresses the expected value of the neutron density, but does not take into account fluctuations from the mean. In a nuclear power reactor, these fluctuations are in general small, in comparison with the average neutron density. Furthermore, fluctuations have no effect on the average neutron density and so the transport equation is valid for the average neutron density no matter how large the fluctuations may be. During the startup phase of a nuclear reaction, however, these deviations from the expected value are large and cannot be neglected; in this case alternative models for the neutron density must be employed. Possible approaches in this situation include stochastic models that allow for rare events such as large deviations, [34].

(c) **Neglect Neutron-Neutron Collisions.**
We neglect neutron-neutron collisions as the chance of them occuring is much smaller than the chance of neutron-nuclei collisions. For example, even in a reactor operating at a high neutron flux of 10^{16} neutrons per cm^2 per second, the neutron density is less than 10^{11} neutrons per cm^3, whereas the density of nuclei is of the order 10^{22} nuclei per cm^3. Due to this difference in density, neutron-neutron collisions will occur much less frequently than neutron-nuclei collisions. Neglecting neutron-neutron collisions ensures that the neutron transport equation is linear.

(d) **Prompt Fission Products.**
We assume in this thesis (indeed in much of the literature) that all neutrons produced by nuclear fission are released immediately. These are known as prompt neutrons. In practice, some neutrons could be released up to several minutes after a fission interaction and are known as delayed neutrons. The omission of delayed neutrons is not significant.

With these assumptions we can derive the Neutron Transport equation shown below. A full derivation is given in Appendix A; briefly we consider a neutron balance for a

region in the full 7 dimensional space, leading to the essential form of the equation.

$$\begin{aligned}\text{Change in the number of neutrons w.r.t. time} = &+\text{gain from scatters in}\\ &+\text{gain from fissions in}\\ &+\text{gain from any external source}\\ &-\text{losses from streaming}\\ &-\text{losses from interactions}\end{aligned}$$

Mathematically the full time–dependent form of the neutron transport equation is expressed as:

$$\begin{aligned}\frac{1}{v}\frac{\partial}{\partial t}\psi(\boldsymbol{x},\boldsymbol{\mu},E,t) = &\int_0^\infty\int_0^{4\pi}\Sigma_s(\boldsymbol{x},\boldsymbol{\mu}',E',\boldsymbol{\mu},E,t)\psi(\boldsymbol{x},\boldsymbol{\mu}',E',t)\,\mathrm{d}\boldsymbol{\mu}'\,\mathrm{d}E'\\ &+\frac{\chi(E)}{4\pi}\int_0^\infty\int_0^{4\pi}\nu(E')\Sigma_f(\boldsymbol{x},E',t)\psi(\boldsymbol{x},\boldsymbol{\mu}',E',t)\,\mathrm{d}\boldsymbol{\mu}'\,\mathrm{d}E'\\ &+Q(\boldsymbol{x},\boldsymbol{\mu},E,t)\\ &-\boldsymbol{\mu}\cdot\nabla_{\boldsymbol{x}}\psi(\boldsymbol{x},\boldsymbol{\mu},E,t)\\ &-\Sigma_t(\boldsymbol{x},\boldsymbol{\mu},E,t)\psi(\boldsymbol{x},\boldsymbol{\mu},E,t),\end{aligned}\tag{1.1}$$

where

$\boldsymbol{x} = (x, y, z)^\top \in \Omega \subset \mathbb{R}^3$	Position in 3 dimensional space.
$\boldsymbol{\mu} \in \mathbb{S}^2 = \{\boldsymbol{\mu} \in \mathbb{R}^3 : \|\|\boldsymbol{\mu}\|\|_2 = 1\}$	Direction of neutron travel.
$E \in \mathbb{R}^+$	Energy.
$t \in \mathbb{R}^+$	Time.
$\psi(\boldsymbol{x}, \boldsymbol{\mu}, E, t) \in \mathbb{R}$	Neutron flux at position $\boldsymbol{x}$ in direction $\boldsymbol{\mu}$ at time t, with energy E
$\nabla_{\boldsymbol{x}}\psi(\boldsymbol{x}, \boldsymbol{\mu}, E, t)$	Spatial Gradient of the neutron flux at position $\boldsymbol{x}$ in direction $\boldsymbol{\mu}$ at time t with energy E.
$\Sigma_t(\boldsymbol{x}, \boldsymbol{\mu}, E, t) \in \mathbb{R}^+$	Total cross section at position $\boldsymbol{x}$ for energy E at time t
$\Sigma_s(\boldsymbol{x}, \boldsymbol{\mu}', E', \boldsymbol{\mu}, E, t) \in \mathbb{R}^+$	Scatter cross section at position $\boldsymbol{x}$ from energy E' and direction $\boldsymbol{\mu}'$ to energy E and direction $\boldsymbol{\mu}$ at time t, respectively.
$\Sigma_f(\boldsymbol{x}, E', t) \in \mathbb{R}^+$	Fission cross section at position $\boldsymbol{x}$ for energy E' at time t
$\nu(E') \in \mathbb{R}^+$	Average number of neutrons produced per fission at energy E'.
$\chi(E) \in \mathbb{R}^+$	Fission neutron distribution.
$Q(\boldsymbol{x}, \boldsymbol{\mu}, E, t) \in \mathbb{R}^+$	All non–fission sources of neutrons with energy E and direction $\boldsymbol{\mu}$ at position $\boldsymbol{x}$ and time t.

We shall now briefly describe the terms appearing in equation (1.1).

Since a neutron is scattered out of a particular energy E' and direction $\boldsymbol{\mu}'$ it must necessarily be scattered into another energy E and change direction to $\boldsymbol{\mu}$: the first term on the right hand side of (1.1) represents this contribution to the total number of neutrons. The scatter cross section $\Sigma_s(\boldsymbol{x}, \boldsymbol{\mu}', E, \boldsymbol{\mu}, E, t)$ loosely represents the probability of a scatter occurring from energy E' and direction $\boldsymbol{\mu}'$ to energy E and direction $\boldsymbol{\mu}$, respectively. The second term on the right hand side of (1.1) represents the amount of neutrons that are gained from fission events. The fission cross section $\Sigma_f(\boldsymbol{x}, E', t)$ is independent of direction, since it is irrelevant what direction a nucleus that hits a nuclei is moving in when causing a fission. However, it is dependent on the energy E of the incident neutron since the speed of the neutron v affects the probability of a fission occurring,

as well as having an influence on the number of neutrons produced per fission; this is denoted by ν.
The term $Q(\boldsymbol{x}, \boldsymbol{\mu}, E, t)$ represents all non-fission sources, for example, neutrons produced by radioactive decay and any external sources.
Finally, the last two terms on the right hand side of the integro–differential equation (1.1) describe the loss of neutrons from the volume. The term $\boldsymbol{\mu} \cdot \nabla_x \psi$ is the loss due to neutrons streaming out of the volume and the $\Sigma_t \psi$ term accounts for the loss of neutrons that occurs when a neutron is scattered to a different energy and direction or absorbed by a nucleus. This is a loss because after a collision the neutron will not be moving in the same direction $\boldsymbol{\mu}$ or with the same energy E as before the collision and so must be removed from the expression for the angular flux $\psi(\boldsymbol{x}, \boldsymbol{\mu}, E, t)$.
In industrial applications the scalar flux is normally of more importance than a detailed knowledge of the angular flux. The scalar flux ϕ, is defined to be the integral over angle of the angular flux, resulting in a quantity with spatial, energy and time dependence, i.e.,

$$\phi(\boldsymbol{x}, E, t) = \int_0^{4\pi} \psi(\boldsymbol{x}, \boldsymbol{\mu}, E, t) \, \mathrm{d}\boldsymbol{\mu}.$$

1.2.1 Cross Sections

A reactor core contains many different materials that neutrons travel through, the properties of the material are described by the cross terms $\Sigma_t, \Sigma_s, \Sigma_f$ and ν in (1.1). These are known as the macroscopic cross sections of a material. The cross sections model the three possibilities, scatter, capture and fission that can occur when a neutron collides with a nucleus:

- A neutron travelling at some energy may impact the nucleus at an angle and be scattered or deflected away, in general with a different energy and at a different angle.
- A neutron may be captured by a nucleus, increasing the mass number of the nucleus.
- A neutron that has been captured by a nucleus may destabilise the nucleus, causing it to split into smaller fragments. This releases additional neutrons that then go on to have interactions themselves with different nuclei. The process of fission releases a lot of energy; this is what drives a nuclear reactor.

The capture and fission interactions are often combined and termed absorption.
The various cross sections for a nuclear fuel exhibit wide oscillations over the whole

energy range and so careful modelling is required to ensure that numerical codes give meaningful results. Cross sections are normally obtained by experiment and stored in nuclear data libraries such as JEFF [132] and JENDL [137] so that they don't have to be repeated.

To be more precise, the cross sections appearing in (1.1) are in fact the macroscopic cross sections and are derived from the microscopic cross sections, see [145] for further details. The corresponding microscopic cross sections, σ_t, σ_s and σ_f are properties of a single isotope of one particular nucleus; physically they can be thought of as the 'characteristic area' presented to a neutron incident to that nucleus. Microscopic cross sections are usually measured in barns (= 10^{-24}cm^2). Let

$$\sigma_{ij}(E) = \text{microscopic cross section for an interaction } i \text{ at energy } E \text{ between nuclide } j \text{ and a neutron,}$$

where i is one of s (scatter), c (capture) or f (fission).
We can now work out the macroscopic cross sections for a material made up of various nuclides (J of them) with volume fractions $v_j, j = 1, \ldots, J$ using the following formulae

$$\Sigma_i(x, E, t) = \sum_{j=1}^{J} v_j N_j(\boldsymbol{x}, t)\sigma_{ij}(E), \qquad i = s, c, f,$$

where $N_j(\boldsymbol{x}, t)$ is the nuclear atomic density for a nuclei of species j at a point $\boldsymbol{x}$ and time t. The nuclear atomic density of a nuclide j is related to the density ρ and atomic number A by Avogadro's number N_0 ($N_0 = 6.022 \times 10^{23}$), hence $N_j = N_0(\rho/A)$.

Remark 2. *As an example, consider a material composed of* 50% *carbon and* 50% ^{239}*Plutonium, using data tabulated in [145]. Then the macroscopic fission cross section can be calculated to be:*

$$\begin{aligned}\Sigma_f(\boldsymbol{x}, E, t) &= \sum_{j=1}^{J} v_j N_j(\boldsymbol{x}, t)\sigma_{ij}(E) \\ &= 0.5(1.6/12.011)(6.022 \times 10^{23})(0) \\ &+ 0.5 \times (19.74/239.0)(6.022E^{23})(698 \times 10^{-24}) \\ &= 17.359.\end{aligned}$$

1.2.2 Common Boundary Conditions

For a problem to be well–posed, in addition to the governing neutron transport equation and knowledge of all the material cross sections throughout the domain we must

also provide some suitable boundary conditions. There are many possible choices in the literature, although here we only concern ourselves with the two that we use later on in the thesis for the benchmarks we employ to validate our numerical method.

Vacuum Boundary Conditions

When designing a nuclear reactor we wish to ensure that neutrons do not pass from the reactor core to the surrounding area. This requirement can be modelled mathematically with vacuum boundary conditions. These are probably the easiest type of boundary conditions to impose since they state that the angular flux on the boundary $\Gamma := \partial\Omega$ of the spatial domain is zero for all incoming directions. Mathematically this can be written as

$$\psi(\boldsymbol{x}, \boldsymbol{\mu}, E, t) = 0 \quad \text{for} \quad \boldsymbol{x} \in \Gamma_{\boldsymbol{\mu}}^{-} = \{\boldsymbol{x} \in \Gamma : \boldsymbol{\mu} \cdot \boldsymbol{n}(\boldsymbol{x}) < 0\}, \tag{1.2}$$

for a point $\boldsymbol{x} \in \Gamma$. Here, $\boldsymbol{n}(\boldsymbol{x})$ denotes the unit outward normal vector to the boundary at the point $\boldsymbol{x} \in \Gamma$. The subscript $\boldsymbol{\mu}$ on the inflow boundary $\Gamma_{\boldsymbol{\mu}}$ is used to highlight the dependence of $\Gamma_{\boldsymbol{\mu}}^{-}$ on the direction $\boldsymbol{\mu}$.

External Boundary Sources

These are used when we have an artificial source on the boundary, meaning that that the angular flux entering the domain is pre–described, these are mainly used later in the thesis in a validation approach to verify the correctness of the numerical method. Mathematically these are expressed as

$$\psi(\boldsymbol{x}, \boldsymbol{\mu}, E, t) = g_0(\boldsymbol{x}, \boldsymbol{\mu}, E, t) \forall t \geq 0 \quad \text{for} \quad \boldsymbol{x} \in \Gamma_{\boldsymbol{\mu}}^{-}$$

Remark 3. *It is worth noting that, in industrial calculations despite not being considered here, reflective boundary conditions are commonly used. There is often a great degree of spatial symmetry present in the domain in the case of a reactor core, for example. Because of this, reflective boundary conditions can be used to reduce the amount of computational effort required. For some industrial benchmarks, such as the C5G7 benchmark [106] this is necessary as the domain is so complex. A quarter of the core comprising two* UO_2 *and two MOX fuel assemblies, where there are three different levels of enrinchment in the MOX assemblies is shown in Figure 1.3 When you consider that each of these four fuel assemblies is composed of a* 17 *by* 17 *grid of pin cells it is clear that significant savings can be made by using reflective boundary conditions.*

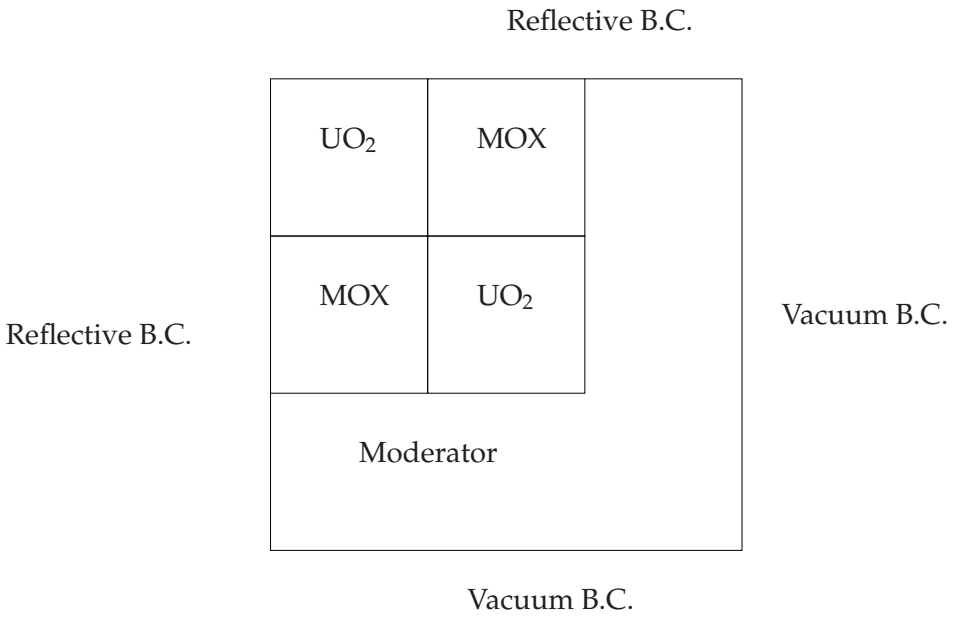


Figure 1.3: Configuration of the quarter core for the C5G7 benchmark problem.

1.3 Simplified Forms of the Neutron Transport Equarion

The full neutron transport equation is an integro–differential equation of 7 dimensions; in general it is not possible to obtain a solution to this analytically and is very time consuming to solve numerically, even with modern multi core computers. Due to the limited power of computers in the early days of the study of the neutron transport equation many simplified forms have been considered. We briefly state some of these in this section, as they will appear in the next chapter.

1.3.1 The Steady State Neutron Transport Equation

Since nuclear reactors are designed to operate for long periods of time they spend a large portion of that in the same 'operational phase' and so there is little change in the angular flux with respect to time. A more interesting question is to determine the flux at a particular point in space and angle; this motivates the study of the steady state neutron transport equation where there is no change in the angular flux over time. consider

$$\begin{aligned}\boldsymbol{\mu}\cdot\nabla_{\boldsymbol{x}}\psi(\boldsymbol{x},\boldsymbol{\mu},E)+\sigma_t(\boldsymbol{x},\boldsymbol{\mu},E,t)\psi(\boldsymbol{x},\boldsymbol{\mu},E) &= \int_0^{\infty}\int_0^{4\pi}\sigma_s(\boldsymbol{x},\boldsymbol{\mu}',E',\boldsymbol{\mu},E)\psi(\boldsymbol{x},\boldsymbol{\mu}',E')\,\mathrm{d}\boldsymbol{\mu}'\,\mathrm{d}E'\\ &\quad+\frac{\chi(E)}{4\pi}\int_0^{\infty}\int_0^{4\pi}\nu(E')\sigma_f(\boldsymbol{x},E')\psi(\boldsymbol{x},\boldsymbol{\mu}',E')\,\mathrm{d}\boldsymbol{\mu}'\,\mathrm{d}E'\\ &\quad+Q(\boldsymbol{x},\boldsymbol{\mu},E).\end{aligned} \tag{1.3}$$

Remark 4. *As stated above this equation is valid for modelling most operational phases of a nuclear reactor, however a different model must be used during the start up phases.*

Remark 5. *The integral over energy appearing in* (1.3) *is normally converted into a proper integral by considering energies that range from* 0 *to some appropriate* $E_{\max}$.

1.3.2 The Mono energetic Neutron Transport Equation

An additional common simplification is to remove the energy dependence in (1.3). Despite neutrons varying wildly in energy this simplification remains well used and is typically employed for a couple of reasons. Firstly, numerical methods developed for the mono energetic (sometimes also referred to as one speed) transport equation can be extended to model the varying energies of neutrons by multi group methods (see [145]) where neutron fluxes are considered in group wise averaged energy bands. Secondly, in certain restrictive geometries it is possible to solve the mono energetic equation analytically, from these solutions benchmarks can be developed, such as those in [74] for the verification of numerical solvers.
To derive the mono energetic neutron transport equation from (1.3) we make the following assumptions.

- There is no energy loss when neutrons undergo a scattering interaction.
- All fission neutrons are born at the same characteristic energy, and so the angular flux only contains neutrons of this energy.
- Any boundary and external sources must be mono energetic.

With these assumptions we can state the mono energetic steady state neutron transport equation as follows:

$$\begin{aligned}\boldsymbol{\mu}\cdot\nabla_x\psi(\boldsymbol{x},\boldsymbol{\mu})+\sigma_t(\boldsymbol{x},\boldsymbol{\mu})\psi(\boldsymbol{x},\boldsymbol{\mu}) = &\int_0^{4\pi}\sigma_s(\boldsymbol{x},\boldsymbol{\mu}',\boldsymbol{\mu})\psi(\boldsymbol{x},\boldsymbol{\mu}')\,\mathrm{d}\boldsymbol{\mu}'\\ &+\frac{\nu\sigma_f(\boldsymbol{x})}{4\pi}\int_0^{4\pi}\psi(\boldsymbol{x},\boldsymbol{\mu}')\,\mathrm{d}\boldsymbol{\mu}'+Q(\boldsymbol{x},\boldsymbol{\mu}).\end{aligned}$$

Assuming isotropic scattering (which in fact we shall do throughout this thesis), the above can be further simplified to

$$\begin{aligned}\boldsymbol{\mu}\cdot\nabla_x\psi(\boldsymbol{x},\boldsymbol{\mu})+\sigma_t(\boldsymbol{x},\boldsymbol{\mu})\psi(\boldsymbol{x},\boldsymbol{\mu}) = &\frac{\sigma_s(\boldsymbol{x})\nu\sigma_f(\boldsymbol{x})}{4\pi}\int_0^{4\pi}\psi(\boldsymbol{x},\boldsymbol{\mu}')\,\mathrm{d}\boldsymbol{\mu}'\\ &+Q(\boldsymbol{x},\boldsymbol{\mu}).\end{aligned}\tag{1.4}$$

1.3.3 Reducing Spatial Dimensionality

The last simplification that is commonly made is the restriction of the three dimensional problem in space to one or two spatial dimensions.

One Dimensional Neutron Transport

For completeness we present the one dimensional neutron transport equation as presented in [34], we do this as the first papers where many of the numerical methods described in the next chapter were derived using this formulation.
Here we reduce the spatial variable $\boldsymbol{x}$ to the first component $_1$ only and assume that there is no variation in the x_2 or x_3 directions. Because of this restriction this model is usually referred to as the 'infinite slab' model where the slab has arbitrary width. For the angular component we consider the angle between the normal to the slab and the trajectory of the neutron, it is usual to use the cosine, $\eta = \cos\theta$ of this angle. The infinite slab neutron transport equation is as follows:

$$\eta \frac{\partial}{\partial x_1}\psi(x_1,\eta) + \sigma_t(x_1)\psi(x_1,\eta) = \frac{\sigma_s(x_1)+\nu\sigma_f(x_1)}{2}\int_{-1}^{1}\psi(x_1,\eta')\,\mathrm{d}\eta' \tag{1.5}$$

Two Dimensional Neutron Transport

The two dimensional neutron transport equation will be the starting point for much of our work.
For this we restrict our spatial domain to two variables, x_1, x_2, this also means that we only need to consider one angular variable φ, the azimuthal angle. This is because all neutrons are confined to moving on a plane. In this case;

$$\boldsymbol{\mu} = (\cos\varphi, \sin\varphi)^\top$$

With this definition we have the two dimensional mono energetic steady state neutron transport equation:

$$\boldsymbol{\mu}\cdot\nabla_x\psi(\boldsymbol{x},\varphi) + \Sigma_t\psi(\boldsymbol{x},\varphi) = \frac{1}{2\pi}\int_0^{2\pi}(\Sigma_s+\nu\Sigma_f)\psi(\boldsymbol{x},\varphi')\,\mathrm{d}\varphi' + Q(\boldsymbol{x},\varphi) \quad \text{in } \Omega\times I \tag{1.6}$$

Remark 6. *Note the integration in angle is over the interval* $[0,2\pi]$ *now as we are integrating over the surface of a circle (it's edge) instead of the surface of a sphere.*

The Pseudo 3D Neutron Transport Equation

The two dimensional mono energetic steady state transport equation can be extended simply to model a situation where we assume an infinite homogeneous domain in the z direction. This shall be derived in a later chapter, but for completeness the resulting problem as stated in [15]is shown below.

$$\boldsymbol{\mu} \cdot \nabla_x \psi(\boldsymbol{x}, \boldsymbol{\mu}) + \sigma_t \psi(\boldsymbol{x}, \boldsymbol{\mu}) = \frac{1}{4\pi} \int_D (\sigma_s + \nu\sigma_f)\psi(\boldsymbol{x}, \boldsymbol{\mu}') \left(1 - |\boldsymbol{\mu}'|^2\right)^{-1/2} \, \mathrm{d}\boldsymbol{\mu}' \tag{1.7}$$

$$+ Q(\boldsymbol{x}, \boldsymbol{\mu}) \quad \text{in } \Omega \times D, \tag{1.8}$$

$$\psi(\boldsymbol{x}, \boldsymbol{\mu}) = g(\boldsymbol{x}, \boldsymbol{\mu}) \quad \text{on } \Gamma^-_{\boldsymbol{\mu}}. \tag{1.9}$$

In the above Ω remains the spatial domain but now the angular domain D is the unit disc, $D := \{\boldsymbol{\mu} \in \mathbb{R}^2 : |\boldsymbol{\mu}| \leq 1\}$. Instead of $\boldsymbol{\mu} = (\cos(\varphi), \sin(\varphi))^\top$ which we had in Equation (1.6) we instead have the following advective direction

$$\boldsymbol{\mu} = \begin{pmatrix} \sin(\theta)\cos(\varphi) \\ \sin(\theta)\cos(\varphi) \end{pmatrix}$$

where φ is the azimuthal angle such that $0 \leq \varphi \leq 2\pi$ and θ is the polar angle we have just introduced such that $0 \leq \theta \leq \frac{\pi}{2}$.

1.3.4 The Underlying Eigenvalue Problem

Whilst descriptive knowledge of the angular flux and scalar flux is important, in the industrial setting computation of a particular critical eigenvalue is significantly more important. The operational safety of a nuclear reactor can be determined with reference to this eigenvalue. Here we describe the underlying eigenvalue problem that gives rise to this critical eigenvalue. In subsequent chapters we will consider the numerical determination of the critical eigenvalue.

To derive the underlying eigenvalue problem we return to the full neutron transport equation as defined in (1.1). Firstly we consider the steady-state equation considered previously; then splitting up the integral on the right hand side of (1.1) into separate scattering and fission components and taking the scattering integral onto the left hand side yields

$$\begin{aligned} \boldsymbol{\mu} \cdot \nabla_x \psi(\boldsymbol{x}, \boldsymbol{\mu}, E) + \Sigma_t(\boldsymbol{x}, \boldsymbol{\mu})\psi(\boldsymbol{x}, \boldsymbol{\mu}, E) & \\ - \int_0^\infty \int_{4\pi} \Sigma_s(\boldsymbol{x}, \boldsymbol{\mu}', E', \boldsymbol{\mu}, E)\psi(\boldsymbol{x}, \boldsymbol{\mu}', E') \, \mathrm{d}\boldsymbol{\mu}' \, \mathrm{d}E' & \\ = \frac{\chi(E)}{4\pi} \int_0^\infty \int_0^{4\pi} \nu(E')\Sigma_f(\boldsymbol{x}, E')\psi(\boldsymbol{x}, \boldsymbol{\mu}', E') \, \mathrm{d}\boldsymbol{\mu}' \, \mathrm{d}E' & \end{aligned}$$

Note, also, that in the above we have neglected the presence of any external sources $Q(\boldsymbol{x}, \boldsymbol{\mu}, E)$; this is a reasonable assumption as in practice reactors are designed to prevent leakage. Defining the transport, scattering and fission operators, respectively, by

$$\begin{aligned}
\mathrm{T}\psi &= \boldsymbol{\mu} \cdot \nabla_x \psi(\boldsymbol{x}, \boldsymbol{\mu}, E) + \sigma_t(\boldsymbol{x}, \boldsymbol{\mu})\psi(\boldsymbol{x}, \boldsymbol{\mu}, E), \\
\mathrm{S}\psi &= \int_0^\infty \int_0^{4\pi} \sigma_s(\boldsymbol{x}, \boldsymbol{\mu}', E', \boldsymbol{\mu}, E)\psi(\boldsymbol{x}, \boldsymbol{\mu}', E') \, \mathrm{d}\boldsymbol{\mu}' \, \mathrm{d}E', \\
\mathrm{F}\psi &= \frac{\chi(E)}{4\pi} \int_0^\infty \int_0^{4\pi} \nu(E')\sigma_f(\boldsymbol{x}, E')\psi(\boldsymbol{x}, \boldsymbol{\mu}', E') \, \mathrm{d}\boldsymbol{\mu}' \, \mathrm{d}E'
\end{aligned}$$

the neutron transport equation can be rewritten in the following operator form

$$(\mathrm{T} - \mathrm{S})\psi = \mathrm{F}\psi. \tag{1.10}$$

The solution ψ to this equation exists if there is an exact balance between the loss and gain of neutrons (as described above). A nuclear reactor satisfying this condition is known as critical. In most cases such a solution will not exist, and we seek some kind of measurement of 'how close' to critical the reactor is. To this end, we transform the problem into a generalised eigenvalue problem by introducing an eigenvalue λ in front of the fission integral; this corresponds to an artificial up-scaling or down-scaling of the fission source. The eigenvalue problem is then defined as follows: find an eigenvalue-eigenfunction pair (λ, ψ) such that

$$(\mathrm{T} - \mathrm{S})\psi = \lambda \mathrm{F}\psi, \tag{1.11}$$

where $\lambda = \frac{1}{k_{eff}}$. Here, k_{eff} denotes the so–called k-effective value; this is the average number of neutrons entering the system per neutron leaving the system.
There are of course multiple pairs (ψ, λ) that satisfy this equation, but we seek the smallest such λ and consequently the highest value of k_{eff}, as this will determine the overall criticality of the system. In particular, we note that

- $\lambda < 1$: The reaction is supercritical. This means that more neutrons are being produced than being removed. Physically this means that the system is becoming more reactive; a situation that nuclear reactor operators wish to avoid as it could lead to a reactor meltdown.

- $\lambda = 1$: The reaction is critical. In this case, for every neutron that leaves the system exactly one enters. In terms of a nuclear reactor, this means that the chain reaction is sustainable over a long period of time.

- $\lambda > 1$: The reaction is subcritical. In this case more neutrons are leaving the system (through absorption, etc) than entering the system and so the system is becoming less reactive. A reaction of this type typically does not lead to a sustained chain reaction and so cannot be used for power generation.

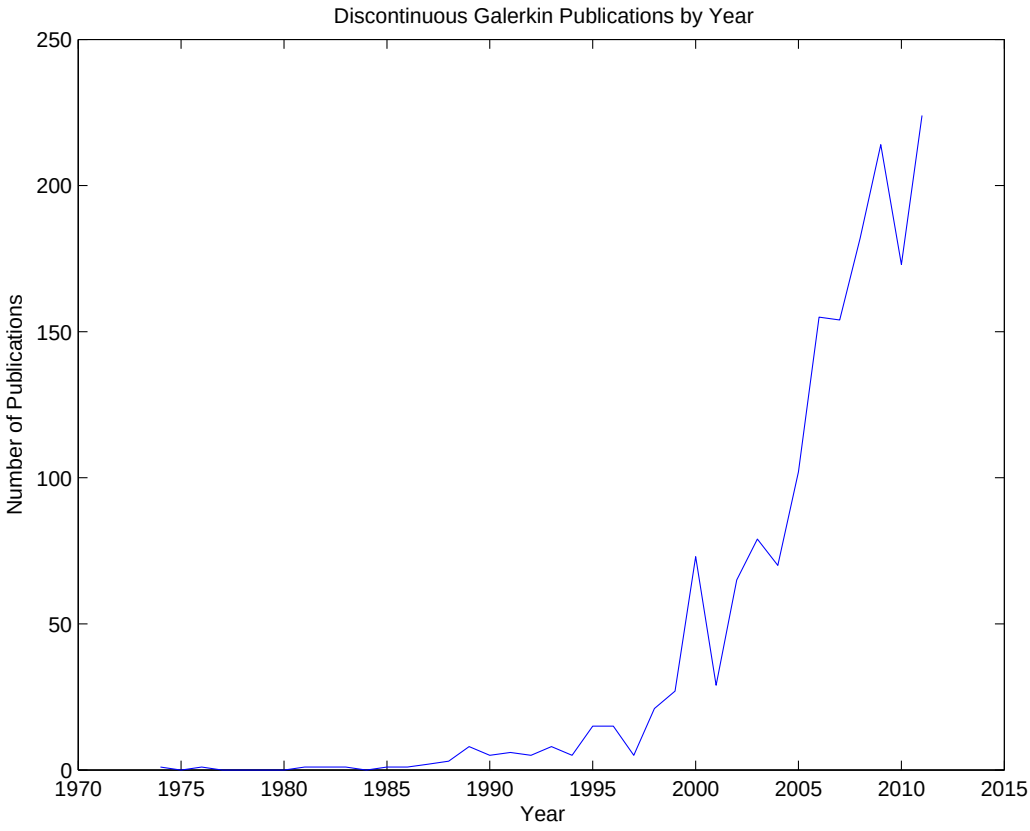


Figure 1.4

1.4 Discontinuous Galerkin Methods

As we shall see in the next chapter there are many numerical methods available for solving the neutron transport equation. The focus of this thesis will be on using a high order discontinuous Galerkin (DG) discretisation in both angle and space. Despite being first introduced in 1973 as a potential method for the spatial discretisation of the neutron transport equation [127] and the growth of research in DG, cf Figure 1.4, there has been very little work on applying DG methods to the neutron transport equation. Notable exceptions are considered in the following chapter.

1.4.1 The History of Discontinuous Galerkin Methods

Discontinuous Galerkin (DG) methods are a comparatively recent invention and in this section we provide a brief history of their application to various problems. We refer the reader to the excellent papers of Arnold, Brezzi, Cockburn and Marini [14] and Cockburn, Karniadakis and Shu [54] for further details of some aspects. These discretisation methods can be thought of as a generalisation of the idea of weak imposition of

boundary conditions [108] and can be considered as a generalisation of finite volume methods. (In fact, the analysis of the two has recently been explicitly combined with the so called $P_N P_M$ schemes of Dumbser [67, 68] for space–time discretisations.)

As already mentioned discontinuous Galerkin methods were first explicitly proposed in 1973 in relation to the neutron transport equation by Reed and Hill [127]. In their paper, Reed and Hill employ a discrete ordinates discretisation (see Section 2.11) and then propose two discretisations in space: one continuous and one discontinuous for the advection problem

$$\begin{aligned} \nabla \cdot (\boldsymbol{\mu}\psi) + \Sigma_t \psi &= f \quad \text{in } \Omega, \\ \psi(x, \boldsymbol{\mu}) &= 0 \qquad \text{on } \Gamma^-_{\boldsymbol{\mu}}, \end{aligned} \tag{1.12}$$

where Ω is the spatial domain and, as before, $\Gamma^-_{\boldsymbol{\mu}}$ denotes the inflow portion of the boundary Γ of the domain Ω. The advective direction $\boldsymbol{\mu}$ is considered to be one particular direction from the set of ordinates used to discretise the angular domain of the neutron transport equation. The standard DG method may then be constructed as follows: first a triangulation $\mathcal{T}_h = \{\kappa\}$ of the domain Ω is constructed. Then for all $\kappa \in \mathcal{T}_h$, $\psi_h|_\kappa$ is the unique solution of the following problem: find $\psi_h|_\kappa \in \mathcal{P}_p(\kappa)$ such that

$$-\int_\kappa \psi_h(\boldsymbol{\mu} \cdot \nabla v_h)\, \mathrm{d}\boldsymbol{x} + \int_{\partial\kappa} \hat{h} v_h \, \mathrm{d}s + \Sigma_t \int_\kappa \psi_h v_h \, \mathrm{d}\boldsymbol{x} = \int_\kappa f v h \, \mathrm{d}\boldsymbol{x} \quad \forall\, v_h \in \mathcal{P}_p(\kappa).$$

Here, $\mathcal{P}_p(\kappa)$ is the space of polynomials of degree at most p on element κ, $\partial\kappa$ is the boundary of κ and $\hat{h}$ denotes an appropriate numerical flux function which handles the numerical jumps in solution values across element boundaries. In this case a suitable choice of numerical flux function would be,

$$\hat{h}(\boldsymbol{x}) = \boldsymbol{\mu} \cdot \boldsymbol{n}_\kappa(\boldsymbol{x}) \lim_{s \searrow 0} \psi_h(\boldsymbol{x} - s\boldsymbol{\mu}),$$

where $\boldsymbol{n}_\kappa(\boldsymbol{x})$ is the unit outward normal to $\partial\kappa$ at the point $\boldsymbol{x}$. Using this definition of the numerical flux it is possible (as pointed out in [100]) to compute the numerical solution ψ_h element by element if the elements have been ordered in the characteristic direction $\boldsymbol{\mu}$; this is because $\lim_{s\searrow 0} \psi_h(\boldsymbol{x} - s\boldsymbol{\mu})$ is simply the value of ψ_h upwind in the characteristic direction.

In their paper Reed and Hill perform numerical experiments that demonstrate the efficiency of the method; the first analysis was published in the following year by LeSaint and Raviart [100]. In this paper they first study a discontinuous Galerkin method applied to an ordinary differential equation before analysing the DG method for the neutron transport equation (1.12) in the $L_2(\Omega)$ norm. Under the condition that $f \in L_2(\Omega)$

existence and uniqueness of the discrete solution ψ_h is established; moreover they show that the convergence order of the scheme is $O(h^p)$ where h is the mesh function $h := max\{h_\kappa\}$, h_κ denoting the diameter of the element κ. The rate of $O(h^p)$ is known to be suboptimal; for a Cartesian grid with tensor product polynomial basis functions LeSaint and Raviart derive a higher rate of convergence, namely $O(h^{p+1})$. The sub-optimal rate of LeSaint and Raviart was improved in 1986 by Johnson and Pitkäranta [93] who showed a rate of convergence of $O(h^{p+\frac{1}{2}})$ in the L_2 -norm for arbitrary order polynomials on locally regular triangulations, namely

$$\|\psi - \psi_h\|_{L_2(\Omega)} \leq Ch^{p+\frac{1}{2}}\|u\|_{H^{p+1}(\Omega)}$$

Analogous results in L_k were also derived, though these were only valid for $p = 1$ and on uniform or piecewise uniform triangulations. As for the case of rectangular meshes, Richter [129] also obtained the bound,

$$\|\psi - \psi_h\|_{L_2(\Omega)} \leq Ch^{p+1}\|u\|_{H^{p+\frac{1}{2}}(\Omega)},$$

on structured triangular meshes by utilising exact representations of u_h on the boundaries of elements. The above bound only holds on semiuniform meshes, where the edges of the triangular elements are bounded away from the edges of the domain. It was not known at the time whether the rate due to Johnson and Pitkäranta [93] was the best possible without assuming further structure on the triangulation; however, in 1991 Peterson [124] numerically showed that this was indeed the case. Peterson also provides a proof that for a piecewise constant approximation and inflow boundary data $g(\boldsymbol{x}) = x$, the point wise error is $O(h^{\frac{1}{2}})$ (since, in this case $p = 0$) for a particular triangular mesh. All of the papers mentioned above assume that the exact solution ψ is smooth in Ω; some work has been carried out towards proving error estimates in situations where this assumption does not hold. For example, in 1993, Lin and Zhou [107] showed convergence to the weak solution assuming that the analytical solution ψ belongs to $H^{\frac{1}{2}}(\Omega)$.

It is also possible to attain convergence to the weak solution by increasing the polynomial degree p used in the approximation, as well as reducing the mesh size h. In 1996, Bey and Oden [36] introduced the hp–version discontinuous Galerkin method (developing earlier work on the hp–finite element method, such as [24]) for hyperbolic conservation laws. They developed a priori and a posteriori error bounds on quadrilateral meshes. The a priori bounds possess the property that for a fixed p and as $h \to 0$ they reduce to those derived in [93]. The corresponding a posteriori error estimate led to the development of the adaptive parallelization strategies discussed in [37]. It is

worth noting that the first a posteriori error estimates for the discontinuous Galerkin method were established in 1990 by Strouboulis and Oden [146] where time dependent problems are considered.

The paper of Houston, Schwab and Süli [88] extended the results of [36] to stabilised hp–finite element methods, including both the streamline diffusion and stabilised DG methods. On quadrilateral meshes their error bounds depended explicitly on the elemental solution regularity which allowed exponential convergence to be established, as the polynomial degree $p \to \infty$. These results were extended, using a different analysis to the non–stabilised DG method by the same authors in [89]. For solutions in L_2 but which are locally smoother, Cockburn, Luskin, Shu and Suli [55] have shown that a post processing of the numerical solution of a transient hyperbolic equation can recover the order $O(h^{p+\frac{1}{2}})$ in Ω_0 where $\Omega_0 \subset \Omega$ with the solution smooth in Ω. They also show that this post processing can improve the rate of convergence in the cases where the analytical solution ψ is smooth to $O(h^{2p+1})$ over the established rate $O(h^{p+\frac{1}{2}})$ cf.[93].

The extension of the original DG method to nonlinear scalar hyperbolic conservation laws was first attempted in 1982 by Chavent and Solzano [49]. They considered a system of the form

$$u_t + \sum_{i=1}^{d} (\boldsymbol{f}_i(u))_{\boldsymbol{x}_i} = 0.$$

As before, multiplying by a test function and integrating formally by parts we obtain the following weak formulation: find u such that

$$\int_\kappa u_t v \, \mathrm{d}\boldsymbol{x} - \sum_{i=1}^{d} \int_\kappa \boldsymbol{f}_i(u) \frac{\partial v}{\partial \boldsymbol{x}_i} \, \mathrm{d}\boldsymbol{x} + \sum_{i=1}^{d} \int_{\partial\kappa} \boldsymbol{f}_i(u) \boldsymbol{n}_\kappa v \, \mathrm{d}s = 0,$$

for all v. Then the approximate solution u_h is defined as the solution of the following problem: find $u|_\kappa \in \mathcal{P}_p(\kappa)$ such that

$$\int_\kappa (u_h)_t v \, \mathrm{d}\boldsymbol{x} - \sum_{i=1}^{d} \int_\kappa \boldsymbol{f}_i(u_h) \frac{\partial v}{\partial \boldsymbol{x}_i} \, \mathrm{d}\boldsymbol{x} + \sum_{i=1}^{d} \int_{\partial\kappa} \hat{h} v \, \mathrm{d}s = 0. \quad \forall v \in \mathcal{P}_p(\kappa)$$

where as before $\hat{h}$ is some numerical flux function used to handle the jumps in the numerical solution u_h across element boundaries. Chavent and Solzano use the above discretisation in space with linear polynomials and employ a forward Euler method to step forward in time. The use of the forward Euler method resulted in an unstable method unless a very restrictive time step, of the order $\sqrt{h}$ was used. In 1989 this instability was overcome in [48] by using techniques borrowed from the finite volume community and introducing a projection based slope limiter. The resulting scheme

was proven to be total variation diminishing in the means (TVDM) and total variation bounded (TVB); however, this leads to only a first order method in both space and time.

In 1988 Shu [138] (also see [139, 140]) introduced an explicit TVD second order Runge-Kutta type time discretisation. This was used by Cockburn and Shu in their 1991 article [52] in conjunction with a piecewise linear DG discretisation in space with an improved slope limiter which maintains formal accuracy of the scheme at extremal points. The numerical results for this method show second order convergence in both space and time and effective resolution of sharp shocks. This approach was generalised to higher order schemes in [51] by employing a DG discretisation in space of order p and a TVD RK method in time with order $p+1$ to obtain an overall RKDG method of order $O(h^{p+1})$. Extending the above methods to the multidimensional case was undertaken for scalar problems in [53] and extended to systems in 1998 in the paper [56]. Time discretisations for DG methods is an active area of research; the local nature of the DG method suggest the possibility of local time stepping, where the time step varies on different spatial elements. Building on the work of Toro, Milligan and Nejad [153] and Toro and Titarev [152] in developing the arbitrary high order derivative (ADER) schemes, Kaser and Dumbser presented an arbitrary high order in both time and space method for the solution of elastic problems [96]. In 1997 this was extended by Dumbser, Kaser and Toro [69] to a method incorporating p adaptivity and local time stepping. The above developments have led to many applications of DG methods for hyperbolic systems, for example; Maxwell's Equations, gas dynamics, acoustics and elasticity, see [84] and the references therein for further details.

Richter performed a direct extension of the original DG method to linear convection diffusion equations in 1992 [130] which he showed possessed an order of convergence of $O(h^{p+\frac{1}{2}})$ when convection was dominant. Since then many other methods have been proposed for solving problems of this type. By rewriting second order operators into systems of first order equations the classic DG approach was extended to second order operators by Bassi and Rebay in 1997 [30]. This has since led to the development of local discontinuous Galerkin (LDG) methods which posses $O(h^p)$ convergence with respect to the L_∞–norm, which is sharp in this setting, see, for example, Cockburn and Shu in [57]. LDG methods can be seen as an extension of the purely hyperbolic RKDG methods. A comparison of three different approaches to discretising a diffusion equation is contained in [165] where the LDG approach is shown to be stable.

Independent of the development of DG methods for hyperbolic problems, discontinu-

ous Galerkin finite element methods were also developed for elliptic problems, though they were not recognised as DG methods at the time. In 1963 Lions (see [108] for a recent reprint of the original paper) considered solving elliptic problems with very rough boundary data. He considered solving, for example,

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega \\ u &= g \quad \text{on } \partial\Omega \end{aligned}$$

where f is taken to be in $L_2(\Omega)$ and g in $H^{-\frac{1}{2}}(\partial\Omega)$. Lions introduced the regularisation $g = u + \gamma^{-1}\frac{\partial u}{\partial \boldsymbol{n}}$ for γ a large positive parameter and considered the following weak form of the regularised problem: find $u \in H^1(\Omega)$ such that

$$\int_\Omega \nabla u \cdot \nabla v \, \mathrm{d}\boldsymbol{x} + \int_{\partial\Omega} \gamma(u-g)v \, \mathrm{d}s = \int_\Omega fv \, \mathrm{d}\boldsymbol{x} \quad \forall\, v \in H^1(\Omega)$$

Lions showed that for any $\gamma > 0$ there exists a unique solution to the above problem and also that as $\gamma \to \infty$ the solution u converges to the solution w of the original problem. In 1973 Babuška [18] applied this method to a finite element method for the first time, proving an order of convergence of $O(h^{\frac{2p+1}{3}})$ for a degree p approximation. This suboptimal rate is due to the lack of consistency in the formulation. Nitsche, in 1971 [117] considered the alternative consistent formulation: find $u \in H^1(\Omega)$ such that

$$\int_\Omega \nabla u \cdot \nabla v \, \mathrm{d}\boldsymbol{x} - \int_{\partial\Omega} \frac{\partial u}{\partial \boldsymbol{n}} v \, \mathrm{d}s - \int_{\partial\Omega} \frac{\partial v}{\partial \boldsymbol{n}} u \, \mathrm{d}s + \int_{\partial\Omega} \gamma(u-g)v \, \mathrm{d}s = \int_\Omega fv \, \mathrm{d}\boldsymbol{x},$$

The second term on the left hand side arises naturally when you apply Green's identity and ensures the consistency of the method; the third time ensures the method is symmetric; the last term guarantees stability. Stability is guaranteed as the last term penalises how far the trace of the numerical solution on the boundary is from the dirichlet data g. In this paper Nitsche showed that if $\gamma \sim \frac{\eta}{h}$ for a sufficiently large constant η and h being the mesh function as before, then the numerical solution converges to the analytical solution with an optimal order with respect to the H^1 and L_2 norms.

The interior penalty methods are a natural extension of the above approach based on weakly imposing continuity across element boundaries, leading to the use of spaces of piecewise discontinuous polynomials in the finite dimensional approximation spaces. In 1973 Babuška and Zlàmal [25] used the formulation of Lions to weakly impose C^1 continuity for fourth order problems. The normal derivatives of C^0 elements were penalised by Douglas and Dupont [64] with the aim of enforcing an idea of continuity, in some sense between the spaces C^0 and C^1 with an application to second order elliptic and parabolic problems. Nitsche's method was generalised to second order elliptic problems in 1977 by Baker [28] where the biharmonic equation was studied. Wheeler

[160] also studied second order elliptic problems in 1978. The interior penalty methods for linear and nonlinear elliptic and parabolic problems were rigorously analysed by Arnold in his thesis of 1979 [10] and the paper [11].

The similarity of the interior penalty methods and for example the flux formulation of Bassi and Rebay [30] has resulted in work which attempts to unify the analysis of various methods in a consistent way. To this end, Brezzi, Manzini, Marini, Pietra and Russo have considered a different formulation of the original method of Bassi and Rebay which was more amenable to mathematical analysis in their paper [40]. The original IP methods were usually presented in their primal form such as in [147], whereas the methods inspired by finite volume techniques and derived for the original *DG* method of Reed and Hill are often presented in terms of suitably chosen numerical fluxes as in [30]. Arnold, Brezzi, Cockburn and Marini analysed all of the elliptic DG methods in a unified framework and present a detailed overview of their similarities, differences and properties in [12] and [14].

1.4.2 The Advantages of Discontinuous Galerkin Methods

DG methods possess a number of key advantages over conventional finite element methods (FEMs). Firstly DG methods are are locally conservative; this is pertinent to the application considered in this thesis. This is a desirable property for a numerical method to possess when applied to problems that model the transport of some material, for example, the transport of neutrons throughout the domain as in our case. For problems containing interior or boundary layers it is known that conventional continuous FEMs suffer difficulties; namely, the growth of non-physical oscillations, when used on meshes too coarse to resolve the layer. Due to this difficulty, stabilised finite element methods such as the streamline upwind Petrov Galerkin (SUPG) method have been developed which add artificial diffusion in the streamwise direction to damp these oscillations. DG methods do not require this extra stabilisation as the discontinuous nature of the method allows the oscillations to be damped by natural numerical dissipation. An example comparing a DG method with a non stabilised continuous FEM for a simple one dimensional problem with a boundary layer is shown in [79]. The standard FEM exhibits global spurious oscillations, whereas, for the DG method these oscillations are localised around the boundary layer.

As explained in the reference by Solin, Segeth and Dolezel [142] conforming CG methods use finite element spaces that satisfy the underlying continuity properties of the

physical problem, for example the spaces $H(\mathrm{div})$ and $H(\mathrm{curl})$. When locally refining the mesh, hanging nodes (element vertices that lie on the face of another element) can be created; for a conforming discretisation inter–element projection operators must be applied to maintain the underlying continuity requirements. We could also subdivide the neighbouring elements, however, this subdivision may create new hanging nodes and this procedure will have to be repeated until no hanging nodes remain. With discontinuous Galerkin methods, due to the weak enforcement of continuity we admit multiple hanging nodes on a face without causing any computational difficulties. Similarly, a DG method naturally permits the use of different polynomial degrees on neighbouring elements; this is harder to perform for a CGFEM method and normally the minimal polynomial rule is applied.

Additionally, without the need for inter-element continuity, there is less communication between elements resulting in sparser matrices than those obtained from a conventional FEM. The major disadvantage with DG methods is the increased number of degrees of freedom when compared to the conventional finite element methods. This can be mitigated somewhat by utilising parallel algorithms, however to be competitive for large problems further work is required. As a consequence of this, there has been a lot of research into efficient linear solvers for matrices arising from DG discretisations of partial differential equations. The increased sparsity and block structure of the linear systems resulting from a DG discretisation goes someway to providing a faster solution procedure than for a similar order FEM, however, for large real world problems a solution using a direct linear solver is unfeasible, due to both the amount of memory required and the time taken to obtain a solution. This leads to the use of iterative solvers such as the conjugate gradient (CG) method or the Generalised Minimal Residual (GMRES) method. The convergence of these solvers is strongly dependent on the condition number $\kappa(A)$ of the underlying system. For a Poisson problem with a linear system arising from a DG discretisation, as shown in [8] the condition number satisfies $\kappa(A) \lesssim \alpha p^4 h^{-2}$ for a globally quasi uniform triangulation and polynomial approximation orders. This has led to interest in robust preconditioners that lower the condition number of the system before the application of an iterative solver. The approach of multi level preconditioning using domain decomposition has received significant attention recently see for instance [72] and [39] where two level additive Schwarz methods are employed for the solution of elliptic problems where the domain Ω has been split into a collection of overlapping domains. Recent work undertaken by Antonietti, and Ayuso [5–7] concerns the use of Schwarz type pre–conditioners on non overlapping subdomains. They demonstrate that it is possible to achieve uniform scalability

(meaning that the number of iterations needed to compute a solution is uniform as the mesh is refined) for a DG discretisation, this is in contrast to CGFEM discretisations where the subdomains must overlap. The disjoint nature of the sub domains is desirable as in parallel implementation this reduces the need for communication between processor nodes. To achieve uniform scalability a suitable coarse solution must be computed on coarse mesh where the ratio of the mesh size H on the coarse mesh and h on the fine mesh is kept constant. They obtain spectral bounds of order H/h for the preconditioned system. There has also been some work on developing multi–grid solvers for problems that have been discretised by the DG method, see for example, [115] where successive coarser grids have not only a larger mesh size but a lower degree of approximating polynomial. As an aside it is worth noting that a similar solution method has been employed for a conforming finite element discretisation with hierarchical basis functions [116].

One further advantage is the ease with which orthogonal basis functions can be constructed, including the Dubiner orthogonal basis as detailed in [66] and employed in AptoFEM, FENICS [97] and the solvers of Warburton [158]. Orthogonal bases are advantageous as they lead to diagonal mass matrices, for time dependent schemes the local nature of the mass matrix combined with the fact that diagonal matrices are trivial to invert can lead to explicit semi-discrete schemes [84].

1.5 Thesis Outline

This thesis is structured as follows. Chapter 2 provides a survey of the numerical methods that have been used to solve the neutron transport equation over the last 60 or so years.

In Chapter 3 we first briefly discuss a discontinuous Galerkin approximation for the advection equation before deriving and developing the discretisation of the two dimensional steady state neutron transport equation. We then extend this to a pseudo three dimensional problem that can be used in conjunction with industrial benchmarks to validate the code produced.

Chapter 4 contains a discussion of various implementational aspects of the two solvers discussed in the previous chapter, including the development of an ordered solver, and a neutron transport specific mesh for the angular dimension in the pseudo three dimensional case.

Following the previous two chapters, Chapter 5 contains numerical results for the two dimensional and pseudo three dimensional solvers. In addition to presenting results

from some test problems derived for benchmarking the source solvers we also present results for various eigenvalue test problems.
In Chapter 6 we introduce the concept of dual weighted residual (DWR) error estimation by applying it to an advection problem. We then apply it to the source and eigenvalue neutron transport problems. We complete the thesis in Chapter 6 by summarising our results and drawing conclusions, before discussing the numerous future areas of research that could follow on from this work.

CHAPTER 2

Numerical Methods for Solving The Neutron Transport Equation

Closed form analytical solutions to equation (1.1) only exist in certain restricted, unrealistic cases and so, for industrial calculations, numerical methods are essential. Many factors contribute to the neutron transport problem being hard to solve numerically. Firstly ψ, the solution to equation (1.1), is a function of three spatial dimensions $\boldsymbol{x}$, angular variables $\boldsymbol{\mu}$, energy E and the time t - making a total of seven variables. In addition to the challenge posed by the high dimensionality, the neutrons being modelled can range in energy from a fraction of an eV to up to 10 MeV. The geometry of the spatial domain Ω also contributes to the challenge of obtaining accurate numerical solutions. When modelling a nuclear reactor there are often many regions of the domain with differing properties, geometries with curved boundaries and for example, many small regions of the domain that need to be resolved accurately. Because of these difficulties various methods have been proposed and implemented during the last 60 or so years. Each of these make different approximations to mitigate the difficulties described above and so have varying degrees of success depending on the situation being modelled. In this chapter we provide a brief overview of the literature concerning the numerical solution of the neutron transport equation and provide a description of the common methods employed. We first look at a common simplification to the transport equation before considering two approaches to discretising the angular component. We then consider two techniques that have been used to discretise the spatial component of the angular flux before briefly discussing a stochastic approach that is often used in practice to obtain benchmark solutions. We conclude by justifying the choices made in discretisation methods for the work contained in this thesis.

2.1 Diffusion Approximations

Due to the difficulties discussed above, when the neutron transport equation was first studied researchers looked for a simpler model to understand first. This led to the development of the diffusion approximations which we discuss here.

Chapter 3 of Stacey's Nuclear Reactor Physics book [145] provides a good introduction to the theory of neutron diffusion approximate solutions to the neutron transport equation. In the 1940s Eddington showed that the transport equation could be recast as a single diffusion equation for the scalar flux, provided that the angular flux possessed certain properties. These properties and their applicability to modern reactor cores are outlined in the following bullet points:

- Assume there is a linear spatial variation of the neutron distribution. This is satisfied a few mean free paths away from the boundary of large homogenous media with relatively uniform source distributions.
- Assume isotropic scattering. This condition is satisfied for scattering from nuclei with a large atomic mass.
- Finally, assume that absorption is much less likely than scattering. This condition is satisfied for most of the structural and moderating materials in a reactor, but not for the control rods and fuel elements.

2.1.1 One Speed Diffusion Theory

In order to outline the theory of the diffusion approximation, below we present the one speed diffusion approximation following the derivation outlined in [145].

Referring to Figure 2.1 consider a point $\boldsymbol{x} \in \mathbb{R}^3$ elevated in the half space somewhere above a small area ΔA located about the origin. A differential volume element $\mathrm{d}\boldsymbol{x}$ is placed about the point $\boldsymbol{x}$. We can calculate the fraction of isotropically scattered neutrons leaving the volume element $\mathrm{d}\boldsymbol{x}$ in the direction of ΔA to be $-(\boldsymbol{x}/|\boldsymbol{x}|) \cdot \Delta A/4\pi r^2$, where $|\boldsymbol{x}|$ is the distance from $\boldsymbol{x}$ to the origin. However, not all of these scattered neutrons will actually reach ΔA since some can be scattered again or absorbed during their journey. Thereby, we can derive an expression for the differential current of neutrons passing downward through ΔA which had their last scattering collision in the small

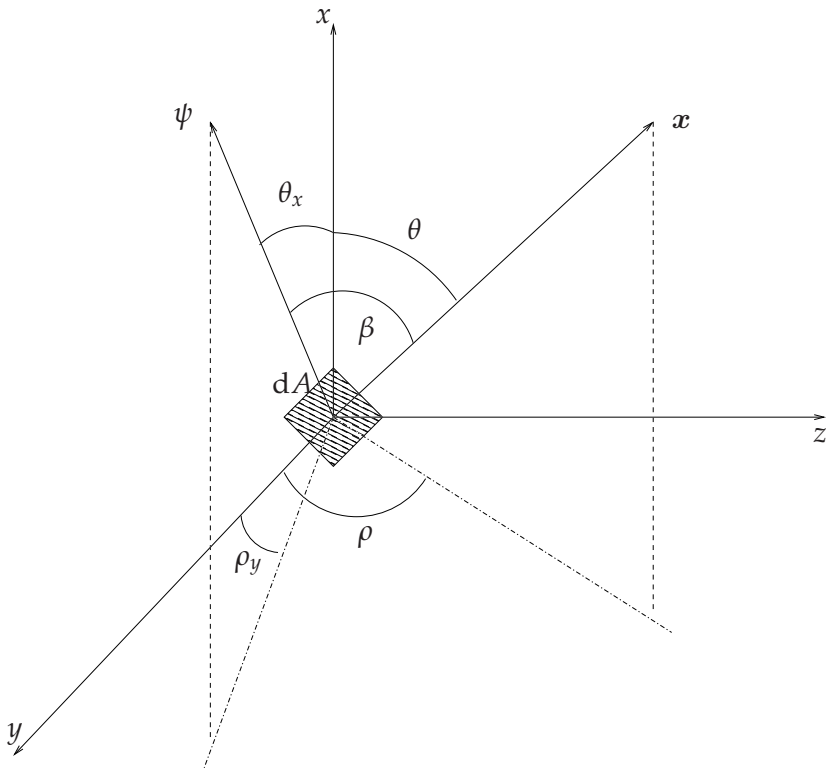


Figure 2.1: The Coordinate System for the Diffusion Derivation.

volume $\mathrm{d}\boldsymbol{x}$ surrounding $\boldsymbol{x}$:

$$j_-(0:|\boldsymbol{x}|,\mu,\rho)\ \mathrm{d}r\ \mathrm{d}A = \frac{\mu e^{-\Sigma_t|\boldsymbol{x}|}\Sigma_s\phi(|\boldsymbol{x}|,\mu,\rho)\ \mathrm{d}\boldsymbol{x}\ \mathrm{d}A}{4\pi|\boldsymbol{x}|^2},$$

where ρ is the azimuthal angle, μ is the cosine of the polar angle and ϕ is the scalar flux as defined previously. Integrating over the upper half plane $x > 0$ we obtain an expression for the total current passing downward through ΔA ,i.e.,

$$j_0(0)\ \mathrm{d}A = \Sigma_s \int_0^\infty \int_0^{2\pi} \int_0^1 \mu e^{-\Sigma_t|\boldsymbol{x}|}\phi(|\boldsymbol{x}|,\mu,\rho)\frac{\mathrm{d}A}{4\pi}\ \mathrm{d}\mu\ \mathrm{d}\rho. \tag{2.1}$$

Expanding the scalar flux ϕ in a Taylor series about the origin gives

$$\phi(\boldsymbol{r}) = \phi(0) + \boldsymbol{r}\cdot\nabla\phi(0) + \frac{1}{2}[r^2\nabla^2\phi(0)] + \cdots.$$

We assume that the flux varies sufficiently slowly in space for us to only retain the first two terms, that is we assume we can write $\phi(\boldsymbol{r}) \approx \phi(0) + \boldsymbol{r}\cdot\nabla\phi(0)$. Using the trigonometric identity $\cos(\beta) = \cos(\theta_x)\cos(\theta) + \sin(\theta_x)\sin(\theta)\cos(\rho_y - \rho)$ and also the approximation that absorption is small relative to scattering, so that $\Sigma_t \sim \Sigma_s$, equation (2.1) can be integrated to obtain the diffusion theory expression for the partial down-

ward density, namely,

$$\begin{aligned} j_0(0) &= \frac{1}{4}\phi(0) + \frac{1}{6\Sigma_s}|\nabla\phi(0)|\cos(\theta_x) \\ &= \frac{1}{4}\phi(0) + \frac{1}{6\Sigma_s}\frac{\mathrm{d}\phi(0)}{\mathrm{d}x} \\ &\equiv \frac{1}{4}\phi(0) + \frac{1}{2}D\frac{\mathrm{d}\phi(0)}{\mathrm{d}x}. \end{aligned}$$

In the above D is known as the diffusion coefficient. Similarly, we can derive the following expression for the partial upward current density,

$$j_=(0) = \frac{1}{4}\phi(0) - \frac{1}{2}D\frac{\mathrm{d}\phi(0)}{\mathrm{d}x}.$$

These two expressions can then be combined to give the diffusion theory expression for the net current at the origin (with the convention that postive sign is upwards),

$$\begin{aligned} J_x(0) &= j_+(0) - j_-(0) \\ &= \frac{1}{3\Sigma_s}\frac{\mathrm{d}\phi(0)}{\mathrm{d}x} \\ &= -D\frac{\mathrm{d}\phi(0)}{\mathrm{d}x}. \end{aligned}$$

We now perform similar derivations in the $x-y$ and $x-z$ planes, then combining the three equations we obtain the three dimensional generalisation,

$$\boldsymbol{J}(0) = -D\nabla\phi(0).$$

This equation is known as Fick's law and governs the diffusion of many other quantities, not just neutrons.

In deriving this equation we have also assumed that the scattering is isotropic; it is possible to derive a more realistic diffusion approximation that takes into account anisotropic scattering. In this latter case the diffusion coefficient will be given by

$$D = \frac{1}{3(\Sigma_t - \overline{\mu_o}\Sigma_s)},$$

with $\overline{\mu_o}$ being the average cosine of the scattering angle. This is approximately $\overline{\mu_o} \approx \frac{2}{3m}$, where m is the atomic mass of the scattering nuclei.

Solutions to the diffusion problem are well documented and so this offers a fast numerical solution to the transport problem. This method, however, has relatively low accuracy for real world applications due to the applicability of the assumptions outlined above. In addition, the approximation also becomes significantly less accurate

near boundaries of the domain since the angular flux is not isotropic within the vicinity of a boundary. The solution will also break down when solving in a space containing voids due to the diffusion coefficients possessing a term involving the factor $1/\sigma_t$, cf. [43] .

Some of these problems can be mitigated by using more accurate transport theory to improve the estimates provided by diffusion theory. For example, the many small elements inside a nuclear reactor can be replaced by a homogenized mixture with averaged cross sections and diffusion coefficients. Control rods which are highly absorbing can be represented by effective diffusion theory cross sections which replicate the transport theory absorption rates. Despite these improvements to diffusion theory the difficulties described above have meant that these techniques have in general, been superseded by other alternative methods in recent years.

2.2 Spherical Harmonic Techniques

The first of the techniques that are used to discretise the angular variables in the neutron transport equation are the spherical harmonic techniques. These are more commonly known as the P_N–methods and are another class of solution methods that first appeared in the 1940s [111, 112]. The name is derived from the functions used to build the approximation.

Spherical harmonic functions are used to approximate the angular variable in the transport equation; these are the angular portion of the solution to the homogeneous Laplace's equation $\nabla^2 u = 0$ when solved in spherical coordinates. The spherical harmonics form a set of complete orthonormal functions and so can be used as a basis in which to expand the angular variables (θ, φ) where θ is the polar angle and ρ is the azimuthal angle in spherical coordinates. They can be defined in the complex plane, but there is an equivalent definition for real valued spherical harmonics ([43]). Indeed, in this latter setting the spherical harmonic basis is defined to be the set $\{Y^c_{l,m}, Y^s_{l,m}\}$ for $l = 0, 1, \ldots, \infty$ and $m = 0, 1, \ldots, l$, where $Y^c_{l,m}$ and $Y^s_{l,m}$ denote the real valued sine and cosine spherical harmonics, respectively, i.e.,

$$Y^c_{l,m} = \cos(m\rho) P^m_l(\mu),$$
$$Y^s_{l,m} = \sin(m\rho) P^m_l(\mu),$$

where P^m_l are the associated Legendre polynomials. These are the canonical solutions of the general Legendre equation, and are most easily defined in terms of derivatives

of the usual Legendre polynomials,

$$P_l^m(x) = (-1)^m(1-x^2)^{m/2}\frac{\mathrm{d}^m}{\mathrm{d}x^m}(P_l(x)).$$

Restricting the expansion to only include all harmonic functions up to order N gives rise to the P_N–approximation. It can be shown that the approximation converges to the analytical solution in the limit as $N \to \infty$.
The spherical harmonic method is fairly accurate in most cases and since it is rotationally invariant ray effects (unphysical oscillations in the angular distribution) are guaranteed not to form in the scalar flux solution unlike when using the Discrete Ordinates method which is described in the next section. For scattering dominated problems and where the angular flux is isotropic, low order expansions are generally sufficient to yield accurate approximations [43].

Regardless of the dimensionality of the problem, obtaining a spherical harmonic approximation to the angular variable in the neutron transport equation proceeds in the same way:

1) Expansion of the angular flux into an infinite expansion in terms of spherical harmonic functions.

2) Substitution of the above expansion into the neutron transport equation.

3) Use of the addition theorem, orthogonality conditions and the recurrence relation associated with Legendre polynomials to form an infinite system of equations in space to be solved.

4) Keep the first $N+1$ equations to leave a system of $N+1$ equations with $N+2$ unknowns.

5) The spatial derivative of the highest order spherical harmonic (i.e. the $N+1$ term) is set to zero, leading to a solvable system of equations in space.

We now illustrate this approach in the case of one dimensional plane geometry.

2.2.1 The One Dimensional Spherical Harmonic Expansion

To illustrate the method we consider the following one dimensional, steady state, monoenergetic transport equation in plane geometry, i.e.,

$$\mu\frac{\partial\psi(x,\mu)}{\partial x} + \Sigma_t\psi(x,\mu) = \frac{\Sigma_s + \nu\Sigma_f}{2}\int_{-1}^{1}\psi(x,\mu')\,\mathrm{d}\mu', \tag{2.2}$$

where μ is the cosine of the angle between the direction of neutron travel and the x axis. Here it is assumed that Σ_t and $\Sigma_s + \nu\Sigma_f$ are constant.
Since ψ is a function of x and μ only, $\psi(x, \cdot)$ can be expanded in a convergent series of Legendre polynomials. The set of Legendre polynomials are defined by the following expressions

$$P_0(\mu) = 1, \tag{2.3}$$

$$P_1(\mu) = \mu, \tag{2.4}$$

$$(2n+1)\mu P_n(\mu) = (n+1)P_{n+1}(\mu) + nP_{n-1}(\mu) \quad \text{for } n \geq 1, \tag{2.5}$$

and they satisfy the following orthogonality condition on the interval $[-1, 1]$,

$$\int_{-1}^{1} P_m(\mu)P_n(\mu)\,\mathrm{d}\mu = \frac{2}{2n+1}\delta_{mn},$$

where δ_{mn} is the Kronecker delta, i.e., $\delta_{mn} = 1$ if $m = n$ and $\delta_{mn} = 0$, otherwise.
We now expand $\psi(x, \cdot)$ as a series of Legendre polynomials in the following fashion

$$\psi(x, \mu) = \sum_{n=0}^{\infty} \frac{2n+1}{4\pi}\psi_n(x)P_n(\mu), \tag{2.6}$$

where the expansion coefficients ψ_n are given by

$$\psi_n(x) = 2\pi \int_{-1}^{1} \psi(x, \mu)P_n(\mu)\,\mathrm{d}\mu. \tag{2.7}$$

We substitute this expansion into (2.2) to obtain, on multiplication by 4π,

$$\mu \sum_{n=0}^{\infty}(2n+1)\frac{\partial\psi_n(x)}{\partial x}P_n(\mu) + \Sigma_t \sum_{n=0}^{\infty}(2n+1)P_n(\mu)\psi_n(x) = (\Sigma_s + \nu\Sigma_f)\psi_0(x). \tag{2.8}$$

So that we can use the recurrence relation stated above we remove the terms corresponding to $n = 0$ from the summations leading to,

$$\mu\frac{\partial\psi_0(x)}{\partial x} + \sum_{n=1}^{\infty}(2n+1)\frac{\partial\psi_n(x)}{\partial x}P_n(\mu) + \Sigma_t\psi_0(x) + \Sigma_t \sum_{n=0}^{\infty}(2n+1)P_n(\mu)\psi_n(x) = (\Sigma_s + \nu\Sigma_f)\psi_0(x)$$

We can now use the recurrence relation (2.5) in the first summation on the left to give,

$$\mu\frac{\partial\psi_0(x)}{\partial x} + \sum_{n=1}^{\infty}\left((n+1)P_{n+1}(\mu) + nP_n(\mu)\right)\frac{\partial\psi_n(x)}{\partial x} + \Sigma_t\psi_0(x) + \Sigma_t \sum_{n=0}^{\infty}(2n+1)P_n(\mu)\psi_n(x) = (\Sigma_s + \nu\Sigma_f)\psi_0(x)$$

Finally, for all $m \in \mathbb{R}$ multiplying the above by $\frac{1}{2}(2m+1)P_m(\mu)$, integrating over the range $-1 \leq \mu \leq 1$, and using the orthogonality property it is found that the following system of equations must be satisfied,

$$\begin{cases} \dfrac{\partial \psi_1(x)}{\partial x} + \Sigma_t \psi_0(x) = (\Sigma_s + \nu\Sigma_f)\psi_0(x) \\ (m+1)\dfrac{\partial \psi_{m+1}(x)}{\partial x} + m\dfrac{\partial \psi_{m-1}(x)}{\partial x} + (2m+1)\Sigma_t \psi_m(x) = (2m+1)(\Sigma_s + \nu\Sigma_f)\delta_{0m}\psi_m(x), \end{cases} \tag{2.9}$$

for $m = 0, 1, 2, \ldots$.

Here, is an infinite set of equations in an infinite number of unknowns. We now introduce an approximation by truncating this set of equations so that we only consider the first $M+1$ of them (that is $m = 1, 2, \ldots, M$). However these involve $M+2$ unknowns, so to make the problem solvable we must remove one of the unknowns.To this end assume

$$\frac{\mathrm{d}\psi_{M+1}(x)}{\mathrm{d}x} = 0.$$

Note that

$$\psi_{M+1}(x) = 2\pi \int_{-1}^{1} \psi(x, \mu) P_{M+1}(\mu)\, \mathrm{d}\mu$$

and by the definition $P_{M+1}(\mu)$ changes sign $M+1$ times in the interval $[-1, 1]$ and so for large M will oscillate rapidly. This leads us to believe that ψ_{M+1} will be very small for large M and so justifies our assumption [34]. In making this assumption we have obtained the P_N approximation to the neutron transport problem.

Now we must solve this system of $N+1$ differential equations. For $x \neq 0$ we obtain a set of homogeneous first order constant coefficient differential equations from equations (2.9). To this end, we seek solutions of the form,

$$\psi_m(x) = g_m e^{v x \Sigma_t}, \qquad m = 0, 1, \ldots, M,$$

Substituting this into (2.9) we have,

$$\begin{cases} v g_1 + \Sigma_t g_0 = 0 \\ v((m+1)g_{m+1} + n g_{m-1}) + (2m+1)(\Sigma_t - (\Sigma_s + \nu\Sigma_f)\delta_{0m})g_m = 0, \end{cases}$$

since $\frac{\mathrm{d} g_m e^{v x \Sigma_t}}{\mathrm{d}x} = v\Sigma_t g_m e^{v x \Sigma_t}$.

For these equations to be compatible it is necessary for the determinant of the matrix of coefficients to vanish; this is achieved by manipulating the value of v. It can be seen that g_m depends on v, so if v_j are all the values of v that make the determinant of the coefficient matrix equal to zero then the general solution can be expressed as

$$\psi_m(x) = \sum_j A_j g_m(v_j) e^{v_j x \Sigma_t}, \qquad m = 0, 1, \ldots, M. \tag{2.10}$$

In the above the A_j are arbitrary constants that depend on the boundary conditions of the problem.

It is expected that the P_N–approximation will be quite accurate for large N. As stated in [34] it is possible to easily obtain an indication of the error made in applying the P_N–approximation in this 1D case: the P_N equations would be exact for a problem in which the source term is modified by the addition of the term,

$$\frac{N+1}{4\pi}\frac{\mathrm{d}\psi_n(x)}{\mathrm{d}x}P_N(\mu).$$

Thereby, the error could be approximated as arising from a source of the above form. In practice, however, to get an idea of the error made it is better to compare the P_N approximation with an analytical solution for a test case where this is known.

Physical Interpretation of the Legendre Expansion of the Flux

The first two terms in the Legendre expansion of the angular flux given in (2.6) have physical meaning. For $m = 0$ we have by definition $P_0(\mu) = 1$, and so equation (2.7) implies that $\psi_0(x)$ is the total flux at the point x.
Similarly, for $m = 1$ we have that $P_1(\mu) = \mu$ and so

$$\psi_1(x) = 2\pi \int_{-1}^{1} \mu\psi(x,\mu)\ \mathrm{d}\mu,$$

which is the current $J(x)$ at x in the direction μ.
It is possible to expand the flux in a different set of orthogonal polynomials, and in some cases this may make it easier to fit boundary conditions, however this physical meaning of the first two terms in the expansion would not remain.

2.2.2 Simplified Spherical Harmonic Methods

The use of Spherical Harmonic methods for practical problems has been hampered by the fact that the multidimensional P_N sets of equations are expensive to solve, as well as the fact that they also grow rapidly in the order of $(N+1)^2$ due to the two variables associated with each basis function. There are other problems associated with the method [43], including sharp gradients in the scalar flux being smoothed out by low order expansions and a difficulty in approximating vacuum boundary conditions because of the continuous variation in the approximated flux.

These difficulties in solving the multidimensional P_N equations led Gelbard [75] in the early 1960s to consider expanding the 1–dimensional spherical harmonic equations

(which are simple and grow at a rate of $(N+1)$ as opposed to $(N+1)^2$) into the multi-dimensional setting. He did this by replacing the spatial derivatives with Laplacian and divergence operators, leading to the Simplified Spherical Harmonic (SP_N) approximation. Despite this method performing well for the initial test problems tried by Gelbard, his derivation was ad-hoc and so interest in this method remained relatively small until the 1980s. In 1981, Lemanska [102] observed that using the simplified third order approximation led to a dramatic improvement in accuracy compared to when diffusion theory is applied. Smith in his paper of 1986 [141] independently introduced the SP_N equations and showed them to produce superior results to a diffusion approximation when applied to two dimensional lattice and core applications. However, he also states that: "Attempts to derive these equations directly from the multidimensional transport equation have been unsuccessful." .
The SP_N method was put on a more sound theoretical footing in 1992 when Larson, McGhee and Morel [99] showed that the SP_N expansion was a high order approximation to the transport equation in cases where the P_1 expansion was the leading term.

2.3 Discrete Ordinates Methods

Discrete ordinates methods (also known as S_N–methods) are another method for discretising the angular variable and were proposed by Wick in 1943 [161]. They were subsequently developed further in the 1950s by Carlson and Chandrasekhar, see [47]. Carlson, whilst working at Los Alamos is credited with much of their development during the 1960s.
In Discrete Ordinates the angular distribution of the neutron flux is evaluated in a discrete number of directions (the ordinates) and quadrature rules are used to replace the scattering and fission neutron source integrals with respect to the angle with summations over the ordinates.

2.3.1 An Illustration of the Method

For ease of explanation we consider the 1D version (2.2) of the neutron transport equation, as we did for the consideration of the spherical harmonic methods.
We discretise the continuous variable μ which in this case varies in the range
$-1 \leq \mu \leq 1$ into a set of discrete directions (these are the ordinates in the name of the method) $\{\mu_i\}_{i=1}^{N}$, we then approximate the integral of the angular flux using an

appropriate quadrature rule to obtain the approximation,

$$\int_{-1}^{1} \psi(x,\mu')\,\mathrm{d}\mu' \approx \sum_{i=1}^{N} \omega_i \psi(x,\mu_i).$$

This yields a set of coupled differential equations that can be solved using any appropriate technique, for example, Finite Differences, when coupled with boundary conditions. Namely, we have

$$\mu_j \frac{\mathrm{d}\psi(x,\mu_j)}{\mathrm{d}x} + \Sigma_t \psi(x,\mu_j) = \frac{\Sigma_s + \nu\Sigma_f}{2} \sum_{i=1}^{N} \omega_i \psi(x,\mu_i) \qquad j = 1,2,\ldots,N. \tag{2.11}$$

The choice of ordinates and weights is highly influential on the accuracy of the underlying method for a given N; they should nevertheless satisfy the following simple conditions:

(a) The weights are normalised so that

$$\sum_{i=1}^{N} \omega_i = 2.$$

(b) It is usually sensible to choose quadrature weights and ordinates that are symmetric about $\mu = 0$. This is achieved by selecting

$$\mu_i = \mu_{N+1-i},$$
$$\omega_i = \omega_{N+1-i}.$$

We do this so that we have equal resolution in the backward and forward fluxes.

(c) The quadrature weights should all satisfy $\omega_i > 0\ i = 1,2,\ldots N$.

(d) We would like the approximation to the integral to be exact in cases where $\psi(x,\cdot)$ is a low order polynomial in μ. This implies the following conditions

$$\sum_{i=1}^{N} \omega_i \mu_i^n = \frac{2}{n+1} \qquad \text{for } n \text{ even},$$
$$\sum_{i=1}^{N} \omega_i \mu_i^n = 0 \qquad \text{for } n \text{ odd}.$$

In fact, the condition for n odd is guaranteed by property (b).

2.3.2 The Relation to P_N Methods

It can be shown that a discrete ordinates method using two ordinates is equivalent to a P_1–approximation. From (2.7) we have that the integrals involved in the approximation are

$$\psi_n(x) = 2\pi \int_{-1}^{1} \psi(x,\mu) P_n(\mu)\,\mathrm{d}\mu.$$

In the discrete ordinates method these are approximated using the quadrature rule, so we have

$$\tilde{\psi}_n(x) = 2\pi \sum_{i=1}^{N} \omega_i \psi(x, \mu_i) P_N(\mu_i),$$

where $\tilde{\psi}_n(x)$ denotes the approximation to $\psi_n(x)$ made by using the quadrature rule. To obtain a set of equations satisfied by $\tilde{\psi}_n(x)$ we multiply (2.11) by $2\pi(2n+1)\omega_j P_n(\mu_j)$ and sum over j. Now using the recurrence relation for the Legendre polynomials (2.5) and the final requirement placed on the quadrature rule outlined above we obtain the following set of equations,

$$(n+1)\frac{\mathrm{d}\tilde{\psi}_{n+1}}{\mathrm{d}x} + n\frac{\mathrm{d}\tilde{\psi}_{n-1}}{\mathrm{d}x} + (2n+1)(\Sigma_t - (\Sigma_s + \nu\Sigma_f)\delta_{0n})\tilde{\psi}_n = 0 \tag{2.12}$$

for $n = 0, 1, 2, \ldots, N-1$.

This is exactly the same set of equations as derived for the spherical harmonics technique in equation (2.9). There is however one important difference: when using spherical harmonics, to obtain a finite set of equations we set $\frac{\mathrm{d}\psi_{N+1}(x)}{\mathrm{d}x} = 0$. We cannot simply do this for the system of equations (2.12) since $\frac{\mathrm{d}\tilde{\psi}_N}{\mathrm{d}x}$ is already determined as it has been defined by the quantities $\psi(x, \mu)$ using the quadrature rule. Instead we choose the ordinates $\{\mu_i\}$ to be the N roots of the nth Legendre polynomial $P_N(\mu)$; this ensures that $\frac{\mathrm{d}\psi_{N+1}(x)}{\mathrm{d}x}$ is automatically zero. Thereby, the set of equations (2.12) is identical to those of the truncated Spherical Harmonic expansion.

2.3.3 Advantages and Disadvantages of the S_N discretisation

Discrete ordinates methods are widely used for calculating the neutron flux distribution in a nuclear reactor core when diffusion approximations lack sufficient accuracy. However, the choice of ordinates and quadrature weights are important, and affect the accuracy and convergence properties. In addition, the spatial discretisation that is chosen to be used in conjunction with the S_N method plays an important role in determining the overall effectiveness of the method. Suppose a finite difference method has been used in one dimensional space with a distance Δ between points, then if Δ is large in comparison with $|2\mu_i|/\Sigma_t$ for any direction $\mu_i, i = 1, \ldots N$ then it is possible for the angular flux to take unphysical negative values. Consequently, if you increase the number of angular ordinates then the spatial mesh must be refined also as some of the ordinates will become increasingly close to zero. This leads to a condition on the relative discretisation sizes somewhat similar to the Courant-Friedrichs-Lewy (CFL) condition. Lathrop developed differencing schemes that avoided this problem by ensuring positive values of the flux [101].

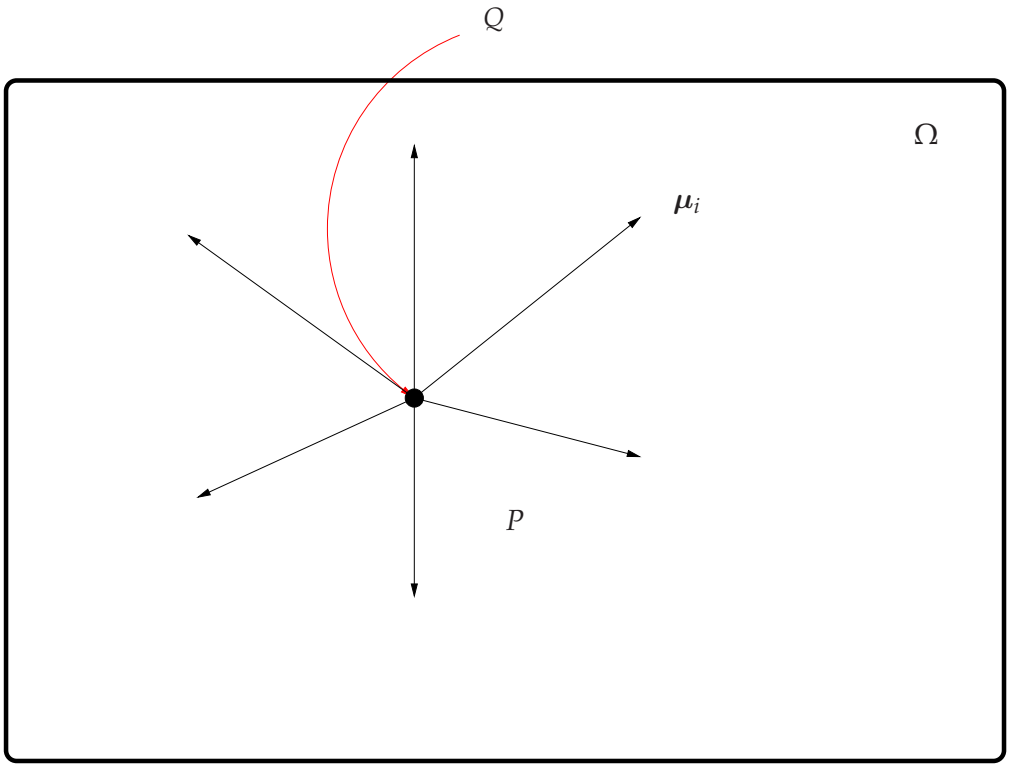


Figure 2.2: With a source placed at point Q and ordinates as shown, no absorption will take place in the direction towards the point P, leading to large oscillations in the angular flux.

For problems involving localised neutron sources and very little scattering (and so high absorption rates), the well documented problem of unphysical oscillations in the angular distribution, known as ray effects can occur [41, 46]. This happens because the solution is only calculated in discrete directions. Looking at Figure 2.2 at the point P there will no absorption as this only occurs in the directions μ_i. To obtain a good approximation to the flux at the point P the discrete ordinates approximation relies on their being enough scattering collisions to re distribute the neutron distribution. For low scattering materials this does not happen and so large oscillations between the fluxes at different directions can occur. One simple remedy is to add more ordinates, but this clearly increases the computational cost. An alternative approach is to perform a semi-analytical calculation of a first collision source to be used in a subsequent discrete ordinates calculation resulting in a conversion from the discrete ordinates equations to spherical harmonic equations which do not exhibit ray effects. It is also possible to design quadrature schemes particular to the neutron transport equation [1] in an attempt to eliminate ray effects.

2.4 Method of Characteristics

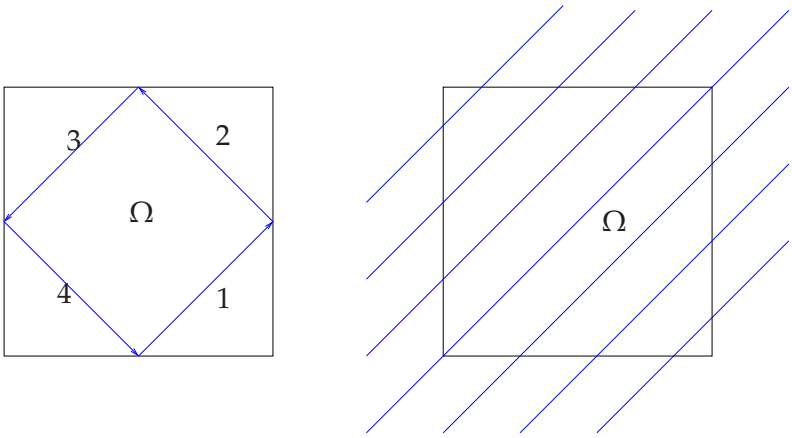
In conjunction with discretisations of the angular dimension spatial discretisations must be considered; one of the most popular of these is the method of characteristics. The method of characteristics (MoC) was first proposed by Riemann in his article of 1859 "Über die Fortpflanzung ebener Luftwallen von endlicher Schwingungsweite" (which can be found in German in the collection [131]).This is a general method applicable to many partial differential equations (PDEs) with a hyperbolic part. In this method, one seeks to change the coordinate system in which the PDE is defined to another where the PDE reduces to an ordinary differential equation (ODE). In discretising this method and applying it to the neutron transport problem one finds the characteristic directions of this PDE are, conveniently straight lines (since $\boldsymbol{\mu}$ does not vary spatially). These straight lines, referred to as tracks, can then be shifted through the domain. This enables one to build up a solution across the entire domain by solving the ODEs along these tracks.

The method was first applied to neutron transport in the 1950s, but it was not until the early 1970s that it was applied to more realistic geometries within the technical report [17]. The first large computer code that made use of the method of characteristics was CACTUS developed in 1980 at Winfrith, Dorset; this code is still being developed and is now one of the main packages included in Serco's commercial WIMS software.

In the original CACTUS implementation a 2D region is covered with a regular grid and then tracking lines begin from a point on the boundary at various angles. When these tracking lines hit another boundary they are reflected; a process which continues until the tracking lines meet their respective starting points again. This results in the region being evenly covered by tracking lines. However, certain conditions on the location of the starting point and the angle chosen have to be met to ensure this constraint is satisfied. Each tracking line approximates the flux at a certain angle, then the values of the scalar flux density are stored for each of the discretised values of energy at each mesh point. This results in a system with complete spatial, angular and energy discretisation. It should be noted that Serco's software has been further developed since 1980 so that it can now also handle 3D domains.

This traditional implementation of ray tracing for the method of characteristics is restricted to relatively simple geometries because tracking lines must return to their starting point. Problems also occur in 3D; in particular, some regions may not be adequately covered by tracking lines, whilst others are covered too densely. To overcome this difficulty Serco have now implemented methods that use so-called "once-through" tracking, where tracking lines are not reflected back through the domain. The difference

between these types of tracking procedures is depicted in Figure 2.3 below and discussed further in [27, 151]..



(a) Reflected tracking. (b) Once through tracking.

Figure 2.3: Reflected tracking (a) and Once-through tracking (b)

2.4.1 The Method of Characteristics in 2D

As an example of the application of the method of characteristics, consider (1.6) which for clarity we reproduce here

$$\boldsymbol{\mu} \cdot \nabla_x \psi(\boldsymbol{x}, \varphi) + \Sigma_t \psi(\boldsymbol{x}, \varphi) = \frac{\Sigma_s + \nu\Sigma_f}{2\pi} \int_0^{2\pi} \psi(\boldsymbol{x}, \varphi') \, \mathrm{d}\varphi'. \tag{2.13}$$

To solve this using a characteristics based approach, first, we perform a discrete ordinates discretisation in angle. Thereby, the scalar flux is calculated by a weighted sum of the angular flux evaluated at specific discrete ordinate directions, i.e.,

$$\begin{aligned} \phi(\boldsymbol{x}) &= \int_0^{2\pi} \psi(\boldsymbol{x}, \varphi') \, \mathrm{d}\varphi' \\ &\approx \sum_{k=1}^{N} \omega_k \psi_k(\boldsymbol{x}, \varphi_k). \end{aligned}$$

Here, N is the number of directions, φ_k are the discrete ordinate directions and ω_k are the weights. If the directions are linearly spaced then $\omega_k = \frac{2\pi}{N}$. The weights ω_k must sum to 2π since we would like the approximation to be exact for constant functions. For each k, ψ_k is the solution to an advection problem; thereby, numerically approximating the neutron transport equation reduces to the problem of solving a sequence of coupled advection problems. Indeed, for each $k = 1, \ldots, N$, we have that

$$\boldsymbol{\mu}_k \cdot \nabla_x \psi_k(\boldsymbol{x}, \varphi_k) + \Sigma_t \psi_k(\boldsymbol{x}, \varphi_k) = \frac{\Sigma_s + \nu\Sigma_f}{2\pi} = \sum_{k=1}^{N} \omega_k \psi_k(\boldsymbol{x}, \varphi_k), \tag{2.14}$$

where $\boldsymbol{\mu}_k = (\cos(\varphi_k), \sin(\varphi_k))^\top$. To solve systems of equations such as (2.14) we can use a fixed source iteration technique. This will be discussed in more detail in Chapter 4, but briefly, this approach uses the scalar flux calculated at the nth level of iteration as the source term on the RHS when solving for the angular flux at the $(n+1)$st iteration as shown below

$$\boldsymbol{\mu}_k \cdot \nabla_x \psi_k^{n+1}(\boldsymbol{x}, \theta_k) + \Sigma_t \psi_k^{n+1}(\boldsymbol{x}, \theta_k) = \frac{\Sigma_s + \nu\Sigma_f}{2\pi} \sum_{k=1}^{P} \omega_k \psi_k^n(\boldsymbol{x}, \theta_k). \tag{2.15}$$

For the first iteration (i.e., $n = 0$) the RHS is taken to be any function that satisfies the boundary conditions. This iteration is then continued until the scalar flux ϕ has converged with respect to some pre-set tolerance.

The method of characteristics can then be used to solve each advection problem; these take the following form

$$\boldsymbol{\mu}_k \cdot \nabla_x \psi_k(\boldsymbol{x}, \theta_k) + \Sigma_t \psi_k(\boldsymbol{x}, \theta_k) = f(\boldsymbol{x}) \qquad \text{in } \Omega, \tag{2.16}$$

$$\psi_k(\boldsymbol{x}) = g(\boldsymbol{x}) \qquad \text{on } \partial^-\Omega, \tag{2.17}$$

where as before $\partial^-\Omega$ denotes the inflow boundary of the domain Ω. The inflow boundary is defined by $\partial^-\Omega = \{x \in \partial\Omega : \boldsymbol{\mu} \cdot \boldsymbol{n}(\boldsymbol{x}) < 0\}$, where $\boldsymbol{n}(\boldsymbol{x})$ is the unit outward normal at the point $\boldsymbol{x}$.

First we split the domain Ω into a mesh $\mathcal{T}_h$ composed of elements of diameter no greater than h and define the following function space associated with the mesh,

$$S_h = \{v \in L_2(\Omega) : v|_k \in P_0(\kappa)\ \forall \kappa \in \mathcal{T}_h\},$$

where $P_0(\kappa)$ denotes the space of constant functions over κ.

As stated before, in this case the characteristics are straight lines and so we place a number of parallel lines across the domain which correspond to the characteristics propagating in the direction of the advection φ_i. Each characteristic crosses a number of elements and the flux on each element is the sum of the contributions from each characteristic crossing that element. For this reason the track spacing is determined in such a manner that at least one track crosses each element.

Along these characteristics, each advection problem of the form (2.16) reduces to the following ODE

$$\frac{\mathrm{d}\psi_k(s)}{\mathrm{d}s} + \Sigma_t \psi_k(s) = f(s), \tag{2.18}$$

where s is a variable which measures length along the characteristic. The ODE (2.18) can be solved using the integrating factor method by writing,

$$\frac{\mathrm{d}}{\mathrm{d}s}(\psi_k e^{\Sigma_t s}) = e^{\Sigma_t s} f(s).$$

Hence,

$$\psi_k(t_2) = e^{-\Sigma_t t_2}\left(\int_{t_1}^{t_2} e^{\Sigma_t r} f(r)\ \mathrm{d}\ r + \psi_k(t_1)e^{\Sigma_t t_1}\right),$$

where t_1 is the location of the inflow point and t_2 is the location of the outfow point. For every intersection of an element by a characteristic, the location of the inflow point t_1 is either known from a previous outflow point of a neighbouring element or the boundary conditions, and then t_2 can easily be calculated using the equation of the straight line.

So far this solution along the characteristic is exact; the approximation to the integral by ,for example, the trapezium rule gives

$$\int_{t_1}^{t_2} e^{\Sigma_t r} f(r)\ \mathrm{d}\ r \approx \frac{(t_2 - t_1)(e^{\Sigma_t t_1} f(t_1) + e^{\Sigma_t t_2} f(t_2))}{2}.$$

Since the trapezium rule is second order accurate the first order property of the approximation for ψ should not be degraded.

Once we know $\psi_k(t_1)$ and $\psi_k(t_2)$ we can calculate the average angular flux $\overline{\psi_k}$ over the current track/element intersection using the following approximation

$$\overline{\psi_k} = \frac{\psi_k(t_1) + \psi_k(t_2)}{2}.$$

We then add this onto the already calculated angular flux in the element κ. Once we have solved for all tracks we obtain a weighted average of the flux on an element κ:

$$\psi_h|_\kappa = \frac{\sum_{i=1}^{\rho_\kappa} \overline{\psi_{i,\kappa}} l_{i,\kappa}}{\sum_{i=1}^{\rho_\kappa} l_{i,\kappa}},$$

where ρ_κ is the number of characteristics which cross the element κ and $l_{i,\kappa}$ the length of the intersection between track i and element κ. The appearance of the subscript i in the average angular flux $\overline{\psi_k}$ describes the explicit dependence of the average angular flux on the track i.

This methodology is then repeated until the flux converges to some pre–set tolerance.

2.4.2 Short Characteristic Methods

The method of short characteristics has been introduced by Takeuchi [133, 149, 150] in an attempt to mitigate some of the drawbacks of the Method of Characteristics. This approach was developed further by Dedner and Vollmöller [62] into a high order method for the radiative transport equation, and has since been analysed by Baker in his recent PhD thesis [27]. We will briefly describe the methodology and main result from [27] in this section. One large disadvantage of the original method of characteristics is that to resolve fine detail in a particular region of space you must have small elements with

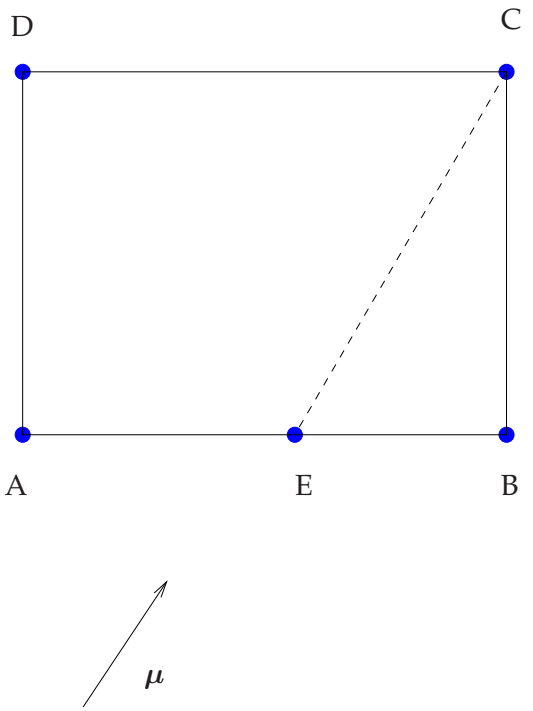


Figure 2.4: Spatial element for the short characteristics method

densely packed characteristics in that region. However, since these characteristics must extend across the whole domain you greatly increase the cost of the overall computation. In the method of short characteristics the characteristics are defined locally on an element instead of globally, and so elements in regions of interest can have characteristics that are packed more densely than in regions of the spatial domain that are not of interest. In Takeuchi's method the unknowns to be determined using the method of short characteristics were the values of the angular flux at the vertex points of a mesh imposed on the spatial domain. Suppose the element shown in Figure 2.4 is one such element. Flux is incident on this element in the direction $\boldsymbol{\mu}$ and so faces AB and AD are inflow faces; we therefore have values for the angular flux at the vertices A, B and D. We wish to find the angular flux at vertex C. To do this we track back from point D in the direction $-\boldsymbol{\mu}$ to a point E on the inflow face AB and then solve the characteristic ODE along this line, where the initial condition at E has been provided by a linear interpolation of the angular flux values at A and B. This approach produced a bilinear representation of the angular flux and was generalised by Dedner and Vollmöller [62] to give rise to a high order spatial approximation.

Baker [27] undertook a rigorous analysis of both the standard method of (long) characteristics and method of short characteristics and showed convergence results for the application of characteristic methods in space with a discrete ordinates discretisation in angle.

2.5 Finite Element Methods

Finite Element Methods (FEMs) are popular for solving partial differential equations on complex domains because of their flexibility. In a finite element method, the problem is recast in a weak formulation where we seek a solution ψ in some function space W. This weak formulation is then discretised by partitioning the domain into a mesh of elements (which in two dimensions are normally triangles or quadrilaterals) and considering a finite dimensional subspace $W_h \subset W$ of polynomials of degree p. A standard FEM can then be summarised by writing the discretised weak form as: find $\psi_h \in W_h$ such that

$$A(\psi_h, v_h) = l(v_h) \qquad \forall v_h \in \overline{V_h},$$

where V_h is the space in which the solution lies, $\overline{V}$ is the test space (note that V_h is not necessarily the same space of functions as $\overline{V}$). Here $A : V_h \times V_h \to \mathbb{R}$ is a bilinear form and $l : V_h \to \mathbb{R}$ is a linear functional Conforming Finite Element Methods are discretisation methods which were developed out of the work of Courant [58]; in this paper from 1943, Courant presents a variational method for the solution of certain vibrational problems. The method was subsequently rediscovered by structural engineers [166] who named it the Finite Element Method (FEM). For the neutron transport equation finite element methods were traditionally used to discretise the spatial dimensions. Conforming FEMs enforce certain continuity constraints between elements and in the 1980s these were applied to neutron transport in conjunction with a spherical harmonic discretisation of the angular domain by Oliveira [60], this led to the development of the EVENT multi dimensional finite element-spherical harmonics radiation transport code as described in [61]. Lewis, Carrico and Palmiotti have also developed a hybrid continuous finite element method for the spatial discretisation of the neutron transport equation (where the angular dimension is also discretised with a spherical harmonics approach) for the code VARIANT [105, 123] which can handle cartesian and hexagonal 2D and 3D geometries. In [157] a finite element discretisation utilising high order conforming finite elements in the library deal.II is described for the diffusion approximation to the transport equation.

It is well known that conforming FE methods can exhibit unphysical oscillations (as discussed in the previous chapter), because of this in their papers [121, 122] Pain et al. advocate the use of the Streamline Upwind Petrov-Galerkin (SUPG) method as developed by Brooks and Hughes in 1982 [42] for the solution of the neutron transport equation. This method introduces an artificial diffusion term in the stream wise direction to damp unphysical oscillations in the vicinity of sharp flux gradients. It is well

known that even these methods do not guarantee the absence of oscillations near large gradients of the solution and that the only way to reliably suppress these oscillations is to introduce some type of non-linear dissipation within the numerical method. This is achieved by adding a discontinuity capturing term to the usual SUPG formulation that attempts to resolve sharp gradients in the solution field by adjusting the amount of dissipation in the space-time streamline direction.

As previously mentioned conforming finite element methods enforce continuity requirements across inter-element boundaries; in contrasts Discontinuous Galerkin (DG) methods do not possess this restriction. As discussed in Section 1.4.1 DG methods were in fact first introduced as a proposed spatial discretisation method for the neutron transport equation in 1973 by Reed and Hill in the technical report [127]. Here a spatial DG discretisation was used in conjunction with a discrete ordinates angular discretisation. In this article they conclude that, despite the increase in computational time due to the increased number of variables to be solved for, the DG approach offer some clear advantages. The DG method is seen to be much more stable, not exhibiting as pronounced unphysical oscillations. The authors also note that a solution acceleration technique known as coarse mesh rebalance [126] works in conjunction with the DG method, whereas it can lead to a divergent scheme when used with a CG discretisation. The first analysis of the method of Reed and Hill was undertaken by LeSaint and Raviart in 1974 [100] where they considered only the spatial discretisation. In addition, they discussed an element by element solution scheme for the resulting linear equations. Subsequently in 1983, Johnson and Pitkäranta published their paper [92] in which they prove an error estimate for a fully discrete method for the numerical solution of a 2D model problem from neutron transport theory. Their work considers a discrete ordinates discretisation for the angular variable and uses DG in the space variable. The problem considered in [92] corresponds to one where the physical region is a thin plate as defined in equation (1.6) but without the presence of the term σ_t and with $(\sigma_s + \nu\sigma_f)/2\pi$ constant. They then recast this equation in operator form.

$$(I - \lambda T)\phi = Tf,$$

where

$$T = \int_S T_{\boldsymbol{\mu}} \, \mathrm{d}\boldsymbol{\mu}$$

and for any $\boldsymbol{\mu} \in S$, $T_{\boldsymbol{\mu}}$ is the solution operator for the problem: given $g \in L_2(\Omega)$, find w such that

$$\boldsymbol{\mu} \cdot \nabla w + w = g \qquad \text{in } \Omega, \tag{2.19}$$

$$w = 0 \qquad \text{on } \Gamma^-_{\boldsymbol{\mu}}, \tag{2.20}$$

(this means that $w = T_{\boldsymbol{\mu}} g$ if w satisfies (2.19) and (2.20)). Finally, ϕ is the scalar flux, defined as follows:

$$\phi = \int_S \psi(x, \boldsymbol{\mu}) \, \mathrm{d}\boldsymbol{\mu}.$$

For an angular discretisation consisting of N discrete ordinates and a spatial discretisation with mesh width h, denoting the error due to quadrature and the error due to the spatial discretisation by e_N and e_N^h respectively, the authors prove an error bound of the form

$$||\phi - \phi_N^h|| \leq C(||e_N|| + ||e_N^h||),$$

where ϕ_N^h denotes the approximate solution and C is a positive constant independent of N and h. This bound can be interpreted as showing a rate of convergence of $O(N^{-1})$ with respect to the angular discrete ordinate discretisation and rate $O(h^{1/2})$ with respect to the mesh width h of the spatial discretisation. Using realistic regularity estimates and supposing that f is sufficiently smooth this can be slightly improved to $N^{-3/2+\varepsilon}$ and $h^{1-\varepsilon}$, respectively. The analysis of Johnson and Pitkäranta has been extended in many papers by Asadzadeh [15, 16].

Radiative transfer theory is often used to describe the radiation fields around stars and other inter-stellar objects and is an area of research that is closely related to neutron transport theory since their governing equations are very similar. As a consequence research in this field should also be considered for its applicability to neutron transport. Kanschat has, since his PhD thesis [95] been working (among other areas) in the area of applying DG methods to radiative transfer problems. In their article, Führer and Kanschat [73] consider the following problem:

$$\boldsymbol{\mu} \cdot \nabla \psi(x, \boldsymbol{\mu}) + (\kappa + \lambda)\psi(x, \boldsymbol{\mu}) = \frac{\lambda}{2\pi} \int_S \psi(x, \theta') \, \mathrm{d}\boldsymbol{\mu}' + \kappa f(x), \quad x \in (\Omega \times S),$$

where κ and λ are constants, Ω, S and $\boldsymbol{\mu}$ are as defined for the two dimensional mono energetic steady state neutron transport equation.Together with homogeneous boundary conditions on the inflow boundary,

$$\psi(x, \mu) = 0 \qquad \text{on } x \in \Gamma^-_{\boldsymbol{\mu}},$$

In [73] a discretisation which uses finite element methods in both angular and spatial discretisations has been developed. In this work they use a DG(p) (i.e. a discontinuous

Galerkin discretisation using polynomials of degree p) for the angular discretisation and the streamline diffusion method with an additional small artificial diffusion term for the spatial discretisation. In this article when using a DG(0) discretisation in angle and using suitable dual problems, the authors derive an a posteriori error estimate for the error in computing ϕ. Moreover, they present numerical experiments which indicate the error estimator is reliable in that it can be used to guide refinement of the spatial mesh when used on regular and quadrilateral meshes. Adaptivity is important since the computational time would become excessively high if only global refinement of the mesh was employed. Despite this success, the authors emphasise that the indicator they present may over-estimate the true error on grids that are not regular. Führer and Kanschat [73] also note that using a DG(0) (i.e. piecewise constants) discretisation in angle is essentially the same as performing a discrete ordinates discretisation, which enables existing, well developed iteration and parallelisation techniques to be used in the solution of the discrete problem.
Such techniques are described in more detail in an article by Richling et al. [128] where the three dimensional radiative transfer problem is considered. Described in this paper is the use of an iterative solver that exploits the structure of the matrix resulting from the discretisation. They also perform a number of numerical tests which produce results that strongly favour the use of finite element methods in solving this problem.

It is worth noting that despite the original paper of Reed and Hill considering higher order DG discretisations, since then there has been minimal work on applying high order polynomial DG approximations to the neutron transport problem. One notable exception is the work of Wang and collaborators [125, 155, 156]. They consider a high order DG spatial discretisation in conjunction with a discrete ordinates angular discretisation for the solution of the multi group neutron transport equations. In [155] two error indicators, one projection based and the other jump based, are proposed for the adaptive refinement of the spatial mesh. Wang points out that these indicators do not perform well for goal oriented calculations, indeed for a boundary flux calculation the solution using adaptively refined meshes is less accurate than that obtained when using uniform refined meshes with less degrees of freedom. Issues concerning parallelisation and the acceleration of the solution process using discrete synthetic acceleration (DSA) are also discussed.

2.6 Monte Carlo Methods

All of the discretisation methods that we have discussed previously are deterministic in nature, it is also possible to use stochastic methods for the solution of neutron transport problems. Indeed, historically benchmark results used to compare solvers have been obtained using Monte Carlo methods.

It is possible to"follow" an individual neutron through a domain, using statistical distributions to model interaction rates with other neutrons; the information generated for each neutron is known as a trial. Monte Carlo Methods exploit a large number of trials and the name is generally credited to Stanislaw Ulam, Enrico Fermi, John von Neumann and Nicholas Metropolis while working at Los Alamos in the 1940s.

Spanier and Gelbard's book [144] provide a good introduction to Monte Carlo methods as applied to Neutron Transport. Monte Carlo methods are stochastic methods and as such will not lead to a fixed deterministic outcome but, if enough simulations are run, very accurate modelling of the problem can be achieved with results to the accuracy required by nuclear engineers obtained. Essentially, the idea is to simulate a large number of particles (significantly less than the total number though) and follow them through time and space. Each particle has values that determine the position, energy and direction at time t associated with it. These values are then used to determine the set of values at time $t + \Delta t$ for each particle, using probability density functions. Each neutron of energy E will have associated probabilities of scattering, absorption, causing fission and continuation (which sum to unity) that are dependent on E. In the instance of scatter and fission, additional probabilities will determine the paths and energies of the emerging neutrons. An entire set of so called 'histories' can be generated by a random number generator which produces uniformly distributed values between 0 and 1. This procedure is carried out as many times as necessary to produce realistic solutions.

The Monte Carlo method is often used in practical applications as it can be used to model complex geometries. However, the method is very computationally intensive, due to the large number of particles that must be simulated to generate reliable results. It also converges slowly, more precisely for N realisations, the error is proportional to $1/N^{1/2}$, so to halve the error requires 4 times as many realisations to be generated.

2.6.1 Variance Reduction

Many techniques have been developed to try and reduce the computational load and increase the efficiency of the Monte Carlo method and several fit into a class known

as"variance reduction" [145]. In general the first moment of a function $h(x)$ is defined as the following integral

$$\langle h \rangle = \int_a^b h(x)p(x)\,\mathrm{d}x,$$

where $a \leq x \leq b$ and $p(x)$ is the probability density function of the random variable x. The first moment is know as the expectation of the function $h(x)$. The second moment is defined to be

$$\langle h^2 \rangle = \int_a^b h^2(x)p(x)\,\mathrm{d}x.$$

Using these two moments we can define the variance of $h(x)$ to be

$$\begin{aligned} V(h) &= \int_a^b (h(x) - \langle h \rangle)^2 p(x)\,\mathrm{d}x \\ &= \langle h^2 \rangle - \langle h \rangle^2. \end{aligned}$$

Finally the standard deviation is defined to be the square root of the variance, i.e., $\mathrm{std}(h) = \sqrt{V(h)}$.

Suppose that we choose N random values of the variable x from the cumulative distribution function (defined to be $F(x) = \int_a^x p(x')\,\mathrm{d}x'$) we can estimate the mean value $\langle h \rangle$ to be

$$\bar{h} = \frac{1}{N}\sum_{i=1}^{N} h(x_n)$$

The error in the estimate above can be bounded using the central limit theorem for large N. This states that if many estimates $\overline{h}$ of $\langle h \rangle$ are obtained in the above way where each estimate involves N trials then the variable $\overline{h}$ is normally distributed about $\langle h \rangle$ to terms of accuracy $O(1/N^{1/2})$.

Usually the first and second moments of $h(x)$ are unknown, but the statistical data can be used to calculate the following approximations:

$$\langle \overline{h} \rangle = \frac{1}{N}\sum_{i=1}^{N} \langle h \rangle = \langle h \rangle$$

and

$$\langle \overline{h}^2 \rangle = \frac{1}{N}\left[N\langle h^2 \rangle + N(N-1)\langle h \rangle^2\right] = \frac{\langle h^2 \rangle}{N} - \frac{N_1}{N}\langle h \rangle^2$$

Using these we can approximate the variance in the statistical estimate of $\bar{h}$ by

$$\begin{aligned} V(\overline{h}) &= \frac{1}{N}\left(\langle h^2 \rangle - \langle h \rangle^2\right) \\ &= \frac{V(h)}{N} \\ &\approx \frac{1}{N-1}\left(\overline{h^2} - \overline{h}^2\right). \end{aligned}$$

We can also approximate the mean squared fractional error associated with the statistical estimate of $\bar{h}$ to be

$$\varepsilon^2 = \frac{1}{N}\left(\frac{\langle h^2\rangle}{\langle h\rangle^2} - 1\right) \approx \frac{1}{N-1}\left(\frac{\overline{h^2}}{\overline{h}^2} - 1\right). \tag{2.21}$$

To increase the confidence in the Monte Carlo calculation it is important to reduce the mean squared error; from (2.21) it is easily seen that one way of doing this is to run more histories. However, this increases the computational cost, hence alternative methods have been introduced to reduce this error.

One common approach is known as importance sampling: if a priori we know that we are more interested in some regions of the domain than others we can use importance sampling [41] to resolve more detail in the region of high interest. In regions of high importance each neutron can be split into two neutrons, each having half the weight of the original neutron. This allows a more detailed investigation of the region than would be possible without doing this. Similarly, in regions of low importance we can remove neutrons, by giving each neutron a fifty-fifty chance of survival, with the proviso that if it survives its weight is doubled.

Another method to reduce the computational load needed is to implement Woodcock tracking [41, 163]. Woodcock tracking is implemented in Serco's MONK code. If the domain is composed of subdomains of different materials with vastly different properties then every particle's track must be divided into sections with the distances between section boundaries known. This is very expensive to implement in practice, and so Woodcock introduced the idea of a "hole region" in which the complex geometry is contained. In this "hole" region all materials are given the same total cross section and so a particle can track through as if it was in a uniform medium ignoring the complex geometry. The value chosen for the total cross section σ_t is chosen to be the largest Σ_t for any material in the region. For the remaining materials (which have a lower Σ_t) there is the addition of a new cross section denoted byΣ_0, which is the probability of a particle scattering forwards with no change in energy and so moving on as if no collision occurred. Using "hole" regions reduces the difficulty in programming the method to solve the transport equation on complex regions. It is possible that routines implementing "hole regions" may be slower than normal regions [41], but this isn't always the case.

2.6.2 The Method of Perturbations

In neutron transport problems it is often required to understand the effect that a small perturbation to the system has on the solution, for example, the change caused by mov-

ing a control rod in or out of the system. When employing conventional Monte Carlo methods it would be necessary to completely re-calculate the solution with the new perturbation. In addition to this taking time, it may also not allow the user to see the change due to the perturbation because of the inherent uncertainties in the Monte Carlo method. The Method of Perturbations has been investigated for many years [78, 81] and seeks to circumvent this problem by solving a simple problem on a homogeneous geometry using a deterministic method, before introducing a perturbation to the system and applying Monte Carlo to calculate a new solution. Classical perturbation theory generally ignores higher order terms, an alternative method of perturbations has been developed by Serco Assurance (now AMEC) and is incorporated into their MAX module which is part of the WIMS software [90]. By performing a sequence of perturbations a solution to a "real life" complex geometry problem can be obtained, using the original deterministic solution as a starting point. In doing this there is also the advantage that the affect of each perturbation can be analysed separately.
To illustrate this method we follow the approach of [134] and consider the following rearrangement of equation (1.10),

$$T\psi = S\psi + \lambda F\psi. \tag{2.22}$$

We assume that we have a solution to the unperturbed problem given by the pair (ψ_0, λ_0), that is we have the solution to the following problem:

$$T_0\psi_0 = S_0\psi_0 + \lambda_0 F_0\psi_0. \tag{2.23}$$

We now introduce a perturbation to the system and define the following new operators,

$$\begin{aligned} T &= T_0 + \Delta T, \\ S &= S_0 + \Delta S, \\ F &= F_0 + \Delta F. \end{aligned}$$

These perturbations will yield a new solution pair (ψ, λ) such that $\psi = \psi_0 + \Delta\psi$ and $\lambda = \lambda_0 + \Delta\lambda$ and so we therefore have the equation,

$$T(\psi_0 + \Delta\psi) = S(\psi_0 + \Delta\psi) + (\lambda_0 + \Delta\lambda)F(\psi_0 + \Delta\psi). \tag{2.24}$$

We now rearrange (using the full expressions for T, S and F) to collect terms involving $\Delta\psi$ on the left hand side and all other terms on the right hand side

$$\begin{aligned} [(T_0 + \Delta T) - (S_0 + \Delta S) - (\lambda_0 F_0 + \lambda_0\Delta F + \Delta\lambda F_0 + \Delta\lambda\Delta F)]\Delta\psi \\ = [(S_0 + \Delta S) - (T_0 + \Delta T) + \lambda_0(F_0 + \Delta F)]\psi_0 \\ + \Delta\lambda(f_0 + \Delta F)\psi_0. \end{aligned}$$

Using the definitions for the perturbed operators and equation (2.23) we obtain the following

$$(T - S - (\lambda_0 + \Delta\lambda)F)\Delta\psi = (\Delta S + \lambda_0 \Delta F - \Delta T)\psi_0 + \Delta\lambda F\psi_0. \tag{2.25}$$

To proceed further and obtain an expression for $\Delta\lambda$ we require the adjoint (or dual) solution (ψ_0^*, λ_0) of the unperturbed problem

$$T_0^*\psi_0^* = S_0^*\psi_0^* + \lambda_0 F_0^*\psi_0^*.$$

We now multiply (2.25) by the adjoint angular flux and integrate over all the spatial, angular and energy domains to obtain (on re-expanding the left hand side and rearranging),

$$\begin{aligned}\int_0^{4\pi}\int_0^{\infty}\int_\Omega \psi_0^*(T_0 - S_0 - \lambda_0 F_0)\Delta\psi \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}E\,\mathrm{d}\boldsymbol{\mu} = \\ \int_0^{4\pi}\int_0^{\infty}\int_\Omega \psi_0^*(\Delta S + \lambda_0\Delta F - \Delta T)(\psi_0 + \Delta\psi)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}E\,\mathrm{d}\boldsymbol{\mu} \\ + \Delta\lambda\int_0^{4\pi}\int_0^{\infty}\int_\Omega \psi_0^* F(\psi_0 + \Delta\psi)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}E\,\mathrm{d}\boldsymbol{\mu},\end{aligned}$$

where Ω is the spatial domain. Due to the nature of the adjoint problem, the left hand side of the above is zero and so we can rearrange the above equation to obtain an expression for the perturbation in the eigenvalue

$$\Delta\lambda = \frac{\int_0^{4\pi}\int_0^{\infty}\int_\Omega \psi_0^*(-\Delta S - \lambda_0\Delta F + \Delta T)(\psi_0 + \Delta\psi)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}E\,\mathrm{d}\boldsymbol{\mu}}{\int_0^{4\pi}\int_0^{\infty}\int_\Omega \psi_0^* F(\psi_0 + \Delta\psi)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}E\,\mathrm{d}\boldsymbol{\mu}}. \tag{2.26}$$

This together with (2.25) can be used to define an iterative scheme to "build up to" the final solution, since if we substitute the expression for $\Delta\lambda$ back into (2.25) we have the following fixed-source equation

$$\begin{aligned}(T - S - (\lambda_0 + \Delta\lambda)F)\Delta\psi = (\Delta S + \lambda_0\Delta F - \Delta T)\psi_0 \\ + \frac{\int_0^{4\pi}\int_0^{\infty}\int_\Omega \psi_0^*(-\Delta S - \lambda_0\Delta F + \Delta T)(\psi_0 + \Delta\psi)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}E\,\mathrm{d}\boldsymbol{\mu}}{\int_0^{4\pi}\int_0^{\infty}\int_\Omega \psi_0^* F(\psi_0 + \Delta\psi)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}E\,\mathrm{d}\boldsymbol{\mu}} F\psi_0.\end{aligned}$$

By relating the method discussed above to the method of (shifted) inverse iteration (an approach for computing eigenvalues and eigenvaectors of a matrix or operator [91], which we shall see later) Scheben shows a linear rate of convergence for the method of perturbations.

2.7 A Comparison of the Methods Discussed

In this concluding section we briefly compare the numerical methods discussed and explain the choice of using a discontinuous Galerkin method for both the angular and

spatial discretisation.

Stochastic Monte Carlo methods are often used to generate benchmark eigenvalue computations but as the focus of this work is adaptive deterministic discretisations they shall not be considered here.
Diffusion methods have the advantage that they produce solutions quickly (this would have been more important when computers were in their infancy), however due to the assumptions on the scalar flux that are required when re-casting as a diffusion problem they have limited applicability. In particular a straight forward diffusion approximation is not suitable for a reactor core eigenvalue computation.
To evaluate the remaining methods discussed in this chapter it is useful to consider angular and spatial discretisations separately.

2.7.1 Angular Discretisation of the Neutron Transport Equation

Historically the most popular angular discretisation was the discrete ordinates method due to it's conceptual simplicity. It can be seen, however, that careful choice of ordinates is important to ensure the accuracy and convergence of the method if this discretisation is chosen. The choice of spatial discretisation used in conjunction with a discrete ordinates method is also important. The primary disadvantage with the discrete ordinates method is the possible presence of ray effects (un physical oscillations) which occur due to the ordinates only sampling specific angular directions. Despite work on methods to mitigate these ray effects they remain a problem for industrial benchmark computations and here the use of spherical harmonic methods as the angular discretisation of choice possess advantages. Spherical harmonic methods do not suffer from ray effects, though in their purest sense they are computationally expensive for three dimensional benchmark problems. Due to this difficulty present when solving the multidimensional S_N equations the simplified spherical harmonic method were proposed, however, despite the work of Larson et. al. [99] the theoretical footing of these methods is still limited.
As mentioned, in his work on the radiative transfer problem [95] Kanschat employed a discontinuous Galerkin method for the angular discretisation, limiting the order of discretisation to piecewise constants. As explained earlier this can be shown to be equivalent to a discrete ordinates (since a one point angular quadrature rule can be used for each angular element) and so ray effects could be present in the numerical solution. A higher order discontinuous Galerkin method in the angular domain shouldn't possess these problems, yet still enable parallelisation in the angular domain. For this reason,

and because we are aiming to develop an *a–posteriori* adaptive method the work in this thesis utilises an arbitrary order discontinuous Galerkin method in angle.

2.7.2 Spatial Discretisation of the Neutron Transport Equation

Finite element methods are eminently suited to the spatial discretisation of the neutron transport equation since they can be of arbitrary order and are also able to handle complicated domains with curved boundaries. Higher order methods have been shown to be beneficial over low order methods since they converge at a higher rate.
The method of characteristics is popular method, though since it is of low order and the implementations often limit the geometries that can be considered the method of short characteristics (as studied in Baker [27] and others) has been developed. One shortcoming of this method is that rigorous analysis is hard.

Because of the desire to implement a high order method with adaptivity the analysis techniques available when using a variational method were attractive and so for this work a finite element discretisation of the spatial domain was proposed. A discontinuous Galerkin method was favoured over a conforming finite element method because of the relative ease of implementing adaptive algorithms. The use of discontinuous Galerkin methods in both angle and space will allow for a unified computational treatment of both discretisations in addition to affording a unified theoretical analysis of the method.

CHAPTER 3

DG Methods for Neutron Transport

In this chapter we derive discontinuous Galerkin discretisations of the neutron transport equation. We begin by discretising a linear advection problem and providing a brief survey of associated *a priori* convergence results. We then turn our attention to the two dimensional mono energetic steady state neutron transport equation, firstly discretising the spatial variable using a DG method and employing a discrete ordinates approximation in angle, before detailing an arbitrary order discontinuous Galerkin discretisation of both the angular and spatial domains. Finally, we discuss some modifications to the problem and discretisation to enable us to consider problems in a specific restriction of the full three dimensional angular space.

3.1 The Linear Advection Problem

Let Ω be a bounded open polyhedral domain in $\mathbb{R}^d$ with smooth boundary Γ. In this section we consider the following linear advection reaction equation

$$\left.\begin{aligned} \mathcal{L}_a u \equiv \boldsymbol{b}\cdot\nabla_x u + \Sigma_t u = f \qquad & \text{in } \Omega \\ u = g \qquad & \text{on } \Gamma^-, \end{aligned}\right\} \tag{3.1}$$

where $f \in L_2(\Omega)$, $\Sigma_t \in L_\infty(\Omega)$ are real valued functions and $\boldsymbol{b} = \{b_i\}_{i=1}^d$ is vector function whose entries b_i are Lipschitz continuous real valued functions on the closure of Ω. Writing $\boldsymbol{n}(\boldsymbol{x})$ to denote the unit outward normal vector to Γ at a point $\boldsymbol{x} \in \Gamma$, on introducing the so called Fichera function $\boldsymbol{b}\cdot\boldsymbol{n}$ (as used in [120]), we can define the following splitting of Γ:

$$\Gamma^- = \{\boldsymbol{x} \in \Gamma : \boldsymbol{b}(\boldsymbol{x})\cdot\boldsymbol{n}(x) < 0\}, \tag{3.2}$$

$$\Gamma^+ = \{\boldsymbol{x} \in \Gamma : \boldsymbol{b}(\boldsymbol{x})\cdot\boldsymbol{n}(x) \geq 0\}. \tag{3.3}$$

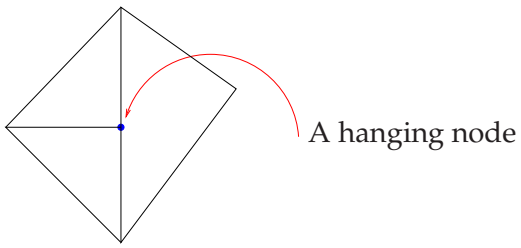


Figure 3.1: A mesh patch with one hanging node on a face

with $\Gamma = \Gamma^- \cup \Gamma^+$. We refer to Γ^- as the inflow boundary and Γ^+ as the outflow boundary.

Now we let $\mathcal{T}$ denote a subdivision of Ω into disjoint open element domains κ such that

$$\bar{\Omega} = \bigcup_{\kappa \in \mathcal{T}} \bar{\kappa},$$

where $\mathcal{T}$ is at most 1-irregular, i.e., we allow at most one hanging node on each face of an element $\kappa \in \mathcal{T}$ as shown in Figure 3.1. By h_κ we denote the diameter of the element $\kappa \in \mathcal{T}$. We assume that the family of subdivisions $\mathcal{T}$ is shape regular (see [65] for example for further details), meaning that there exists some number $d > 0$ such that every $\kappa \in \mathcal{T}$ contains a circle of radius ρ_κ with

$$\rho_\kappa \geq \frac{h_\kappa / 2}{d}.$$

We assume that each element κ can be expressed as a mapping of a fixed reference element $\hat{\kappa}$, that is $\kappa = F_\kappa(\hat{\kappa})$, where F_κ is smooth and bijective, for each $\kappa \in \mathcal{T}$. For simplicity we will now restrict ourselves to the situation $d = 2$, that is our domain Ω is a portion of $\mathbb{R}^2$. The reference element $\hat{\kappa}$, in this case can be either the open unit triangle

$$\hat{\kappa}_T = \{\hat{\boldsymbol{x}} = (\hat{x_1}, \hat{x_2}) \in \mathbb{R}^2 : 0 < \hat{x_1} + \hat{x_2} < 1, \hat{x_i} > 0,\ i = 1, 2\},$$

or the open hypercube $\hat{\kappa}_C = (-1, 1)^2$ in $\mathbb{R}^2$.
On $\hat{\kappa}$ we define the following spaces of polynomials of degree $p \geq 0$, with the convention that α is a multi-index:

$$\mathcal{P}_p = \text{span}\{\hat{\boldsymbol{x}}^\alpha : 0 < |\alpha| \leq p\},$$
$$\mathcal{Q}_p = \text{span}\{\hat{\boldsymbol{x}}^\alpha : 0 \leq \alpha_i \leq p, 1 \leq i \leq 2\}.$$

For each $\kappa \in \mathcal{T}$ we denote by $p_\kappa \geq 0$ the polynomial approximation degree employed on κ. Forming the vectors $\boldsymbol{p} = \{p_\kappa : \kappa \in \mathcal{T}\}$ and $\boldsymbol{F}_\Omega = \{F_\kappa : \kappa \in \mathcal{T}\}$, we introduce the discontinuous hp-finite element space

$$V_{h,p} = S^{\boldsymbol{p}}(\Omega, \mathcal{T}, \boldsymbol{F}_\Omega) = \{u \in L_2(\Omega) : u|_\kappa \circ F_\kappa \in \mathcal{S}_{p_\kappa},\ \kappa \in \mathcal{T}\}, \tag{3.4}$$

where

$$\mathcal{S}_{p_\kappa} = \begin{cases} \mathcal{P}_{p_\kappa}(\hat{\kappa}) & \text{if } F_\kappa^{-1}(\kappa) = \hat{\kappa}_T, \\ \mathcal{Q}_{p_\kappa}(\hat{\kappa}) & \text{if } F_\kappa^{-1}(\kappa) = \hat{\kappa}_C. \end{cases}$$

Remark 7. *The extra degrees of freedom present in the space $\mathcal{Q}_p$ are required to maintain the optimality of the approximation properties on a general quadrilateral mesh, see [13] for details.*

If the polynomial degree is uniform, i.e., $p_\kappa = p$ for all $\kappa \in \mathcal{T}_h$ then we simply write $S^p(\Omega, \mathcal{T}, \boldsymbol{F}_\Omega)$. For any element $\kappa \in \mathcal{T}$, we let $\partial\kappa$ be the union of 1–dimensional open faces of κ. Similarly, for any $\boldsymbol{x} \in \partial\kappa$, $\boldsymbol{n}_\kappa(\boldsymbol{x})$ denotes the unit outward normal vector to $\partial\kappa$ at $\boldsymbol{x}$. Using these definitions, we can define, the inflow and outflow parts of $\partial\kappa$, respectively, by

$$\partial^-\kappa = \{x \in \partial\kappa : \boldsymbol{b}(\boldsymbol{x}) \cdot \boldsymbol{n}_\kappa(\boldsymbol{x}) < 0\},$$
$$\partial^+\kappa = \{x \in \partial\kappa : \boldsymbol{b}(\boldsymbol{x}) \cdot \boldsymbol{n}_\kappa(\boldsymbol{x}) \geq 0\}.$$

where $\boldsymbol{n}_\kappa(\boldsymbol{x})$ now denotes the unit outward normal vector to $\partial\kappa$ at $\boldsymbol{x} \in \partial\kappa$. For any $\kappa \in \mathcal{T}$ we denote the interior trace of $v \in H^1(\kappa)$ by v_κ^+. For an element $\kappa \in \mathcal{T}$ such that $\partial^-\kappa \setminus \Gamma$ is non empty then for every $\boldsymbol{x} \in \partial^-\kappa \setminus \Gamma$ there exists a unique element $\kappa' \in \mathcal{T}$ such that $\boldsymbol{x} \in \partial^+_{\boldsymbol{\mu}}\kappa'_\Omega$. Then, assuming that $v \in H^1(\kappa)$ for all elements $\kappa \in \mathcal{T}$, we can also define the outer trace v^- of v on $\partial^-_{\boldsymbol{\mu}}\kappa \setminus \Gamma_-$ relative to κ as the interior trace v^+ relative to the element κ'. This is shown in Figure 3.2.

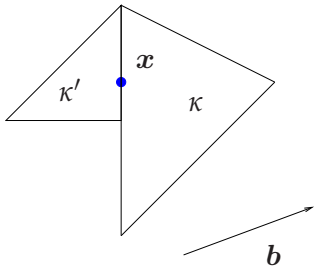


Figure 3.2: A point $\boldsymbol{x}$ such that $\boldsymbol{x} \in \partial^-\kappa_\Omega$ and $\boldsymbol{x} \in \partial^+\kappa'_\Omega$

3.1.1 *hp* DGFEM Discretisation

To define the *hp* discontinuous Galerkin finite element discretisation we consider an alternative form of the advection equation (3.1), namely the conservative form.

$$\left.\begin{aligned} \boldsymbol{b} \cdot \nabla_x u + \Sigma_t u &= f \qquad \text{in } \Omega \\ u &= g \qquad \text{on } \Gamma^-, \end{aligned}\right\} \tag{3.5}$$

We seek a solution $u \in V$ where V is a suitable function space.To this end, multiplying (3.5) by a smooth test function v and integrating over an arbitrary element κ, utilising integration by parts to give

$$-\int_\kappa (\boldsymbol{b}u)\cdot\nabla v \,\mathrm{d}\boldsymbol{x} + \int_{\partial\kappa} (\boldsymbol{b}u^+)\cdot\boldsymbol{n}_\kappa v^+ \,\mathrm{d}s = 0.$$

We can weakly enforce inter element continuity and boundary conditions with the use of a suitable numerical flux function $\mathcal{H}(\cdot,\cdot,\boldsymbol{n}_\kappa)$ to obtain,

$$-\int_\kappa (\boldsymbol{b}u)\cdot\nabla v \,\mathrm{d}\boldsymbol{x} + \int_{\partial\kappa} \mathcal{H}(u^+,u^-,\boldsymbol{n}_\kappa)\boldsymbol{n}_\kappa v^+ \,\mathrm{d}s = \int_\kappa fv \,\mathrm{d}\boldsymbol{x},$$

finally, summing over all elements $\kappa \in \mathcal{T}$ we obtain the following continuous weak formulation: find $u \in V$ such that

$$\sum_{\kappa\in\mathcal{T}}\left(-\int_\kappa (\boldsymbol{b}u)\cdot\nabla v \,\mathrm{d}\boldsymbol{x} + \int_{\partial\kappa} \mathcal{H}(u^+,u^-,\boldsymbol{n}_\kappa)\boldsymbol{n}_\kappa v^+ \,\mathrm{d}s\right) = \sum_{\kappa\in\mathcal{T}}\int_\kappa fv \,\mathrm{d}\boldsymbol{x},$$

for all $v \in V$. to obtain the DG discretisation we simply restrict to the finite dimensional space $V_{h,p}$ and replace u with its discontinuous Galerkin approximation u_{DG} and v with v_h. This leads us to our discretised weak formulation to be solved: find $u_{DG} \in V_{h,p}$ such that,

$$\sum_{\kappa\in\mathcal{T}}\left(-\int_\kappa (\boldsymbol{b}u_{DG})\cdot\nabla v_h \,\mathrm{d}\boldsymbol{x} + \int_{\partial\kappa} \mathcal{H}(u^+_{DG},u^-_{DG},\boldsymbol{n}_\kappa)v^+_h \,\mathrm{d}s\right) = \sum_{\kappa\in\mathcal{T}}\int_\kappa fv_h \,\mathrm{d}\boldsymbol{x}, \tag{3.6}$$

for all $v_h \in V_{h,p}$.

Remark 8. *Note that all integrals in the above are elemental and we have not performed an integration by parts globally*

The numerical flux function $\mathcal{H}(\cdot,\cdot,\cdot)$ handles the jumps in solution values across element boundaries. There are many choices of numerical flux function, see for example [104], however any chosen should possess the following properties:

- **Consistency**
  For each $\kappa \in \mathcal{T}$ we have that

  $$\mathcal{H}(v,v,\boldsymbol{n}_\kappa|_{\partial\kappa} = (\boldsymbol{b}v)\cdot\boldsymbol{n}_\kappa \quad \forall\,\kappa\in\mathcal{T}$$

- **Conservation**
  For any two neighbouring elements κ and κ' from the partition $\mathcal{T}$, at each point $\boldsymbol{x} \in \partial\kappa \cap \partial\kappa' \neq \emptyset$, noting that $\boldsymbol{n}_{\kappa'} = -\boldsymbol{n}_\kappa$ we have that

  $$\mathcal{H}(v,w,\boldsymbol{n}_\kappa) = -\mathcal{H}(w,v,-\boldsymbol{n}_\kappa)$$

The simplest choice would be the upwind numerical flux

$$\mathcal{H}(u_{DG}^+, u_{DG}^-, \boldsymbol{n}_\kappa) = \boldsymbol{b}\cdot\boldsymbol{n}_\kappa \lim_{s\to 0^+} uh(x - s\boldsymbol{b}) \quad \text{for} \kappa \in \mathcal{T},$$

however, to be consistent with later chapters we choose the Lax–Friedrichs flux

$$\mathcal{H}(u_{DG}^+, u_{DG}^-, \boldsymbol{n}_\kappa) = \frac{1}{2}(\boldsymbol{b}\cdot\boldsymbol{n} + |\boldsymbol{b}\cdot\boldsymbol{n}|)u_{DG}^+ + \frac{1}{2}(\boldsymbol{b}\cdot\boldsymbol{n} - |\boldsymbol{b}\cdot\boldsymbol{n}|)u_{DG}^-. \tag{3.7}$$

This flux needs to be augmented with a definition on the boundary, since for element faces coincident with the boundary u_{DG}^- has no meaning. Instead, the second argument is replaced by u_Γ, defined in the following way

$$u_\Gamma(u_{DG}^+) = u_{DG}^+ \quad \text{for} \boldsymbol{x} \in \Gamma^+$$
$$u_\Gamma(u_{DG}^+) = g \quad \text{for} \boldsymbol{x} \in \Gamma^-$$

Note, that for linear problems the upwind numerical flux and the Lax–Friedrichs flux are equivalent.

3.1.2 Error Analysis of the *hp*–DGFEM for the Advection Equation

In their paper of 1996 [36], K. Bey and J.T. Oden introduce *hp*–discontinuous Galerkin methods and establish *a priori* and *a posteriori* error bounds on quadrilateral meshes. For fixed p and as $h \to 0$ the results derived in this paper reduce to the classical optimal order estimates of Johnson et al. [93, 94] in a mesh dependent norm.

The results of [36] were generalised by P. Houston, Ch. Schwab and E. Süli [88], where they established a unified framework for the analysis of the streamline diffusion finite element method and the stabilized discontinuous Galerkin finite element methods. These error estimates are sharp as $h \to 0$ and $p \to \infty$. In deriving these estimates for the case of quadrilateral meshes, one dimensional approximation results established in [135] are employed to derive tensor product approximation results on quadrilaterals. In addition, when quadrilateral meshes are used the estimates depend explicitly on the elemental regularity of the solution; this allows exponential convergence rates with respect to the polynomial order p to be deduced for solutions that are piecewise analytic. Whilst not shown in [88], the authors remark that analogous results can be obtained for simplicial meshes using the approximation results stated in [24]; however, these bounds are no longer explicit respect to the local regularity.

In [89] P. Houston, Ch. Schwab and E. Süli extended the analysis of [88] to general

advection-diffusion-reaction problems without the addition of streamline diffusion stabilisation. Indeed, employing alternative approximation results, optimal order error bounds in both h and p are derived for the hyperbolic advection reaction equation.

Define the mesh dependent DG norm $|||\cdot|||_{DG}$ by

$$|||w|||_{DG} = \sum_{\kappa\in\mathcal{T}} \left(\|c_0 w\|^2_{L_2(\kappa)} + \frac{1}{2}\|w^+\|^2_{\partial^-\kappa\cap\Gamma^-} + \frac{1}{2}\|w^+ - w^-\|^2_{\partial^-\kappa\setminus\Gamma} + \frac{1}{2}\|w^+\|^2_{\partial^+\kappa\cap\Gamma} \right). \tag{3.8}$$

With Π_p denoting the orthogonal L_2 projector into the finite element space, such that

$$(u - \Pi_p u, v) = 0 \qquad \forall v \in S^{\boldsymbol{p}}(\Omega, \mathcal{T}, \boldsymbol{F})$$

the error in the approximation $u - u_{DG}$ can be expressed as

$$u - u_{DG} = (u - \Pi_p u) + (\Pi_p u - u_{DG}) = \eta + \xi. \tag{3.9}$$

With this decomposition [88] established the following *a priori* bound.

Theorem 3.1.1. *Let $\Omega \subset \mathbb{R}^d$ be a bounded polyhedral domain and let $\mathcal{T} = \{\kappa\}$ be a shape regular subdivision of Ω into d-parallelepipeds κ with diameter h_κ. Let $u_{DG} \in S^{\boldsymbol{p}}(\Omega, \mathcal{T}, \boldsymbol{F})$ be the DG approximation to u and suppose that $u|_\kappa \in H^{k_\kappa}$ for each $\kappa \in \mathcal{T}$ for integers $k_\kappa \geq 1$. Then, assuming a suitable positivity condition holds and that $\boldsymbol{b}\cdot\nabla v_h \in S^{\boldsymbol{p}}(\Omega, \mathcal{T}, \boldsymbol{F})\ \ \forall v_h \in S^{\boldsymbol{p}}(\Omega, \mathcal{T}, \boldsymbol{F})$ the following error bound holds:*

$$\begin{aligned} |||u - u_{DG}|||_{DG} &\leq |||\xi|||_{DG} + |||\eta|||_{DG} \\ &\leq \left(C \sum_{\kappa\in\mathcal{T}} h_\kappa^{2s_\kappa - 1} \left(\beta_\kappa \Phi_1^2(p_\kappa, s_\kappa) + \gamma_\kappa h_\kappa \Phi_2^2(p_\kappa, s_\kappa) \right) |u|^2_{H^{s_\kappa}(\kappa)} \right)^{\frac{1}{2}} \end{aligned}$$

for integers s_κ, $1 \leq s_\kappa \leq \min(p_\kappa + 1, k_\kappa)$, and $p_\kappa \geq 0$. In the above C, β_κ and γ_κ are positive constants. The constant C depends only on the dimension d and shape regularity of $\mathcal{T}$. The functions $\Phi_1(p,s)$ and $\Phi_2(p,s)$ are defined as follows:

$$\begin{aligned} \Phi_1(p,s) &= (2p+1)^{-\frac{1}{2}} \left[G_p(2-s,s)^{\frac{1}{2}} + G_p(3-s,1+s)^{\frac{1}{2}} \right] \\ &\qquad + G_p(2-s,2_s)^{\frac{1}{4}} G_p(3-s,1+s)^{\frac{1}{4}} + G_p(2-s,2+s)^{\frac{1}{2}}, \\ \Phi_2(p,s) &= G_p(2-s,2+s)^{\frac{1}{2}}, \end{aligned}$$

where $G_p(s,t) = \Gamma(p+s)/\Gamma(p+t)$, with Γ signifying the gamma function.

In the case of uniform polynomial orders $p_\kappa = p \geq 0$, $h = \max_{\kappa\in\mathcal{T}} h_\kappa$ and $s_\kappa = s$, $1 \leq s \leq \min(p+1, k)$, $k \geq 1$ the above error bound reduces to

$$|||u - u_{DG}|||_{DG} \leq C(h/(p+1))^{s-\frac{1}{2}} |u|_{s,\mathcal{T}} \tag{3.10}$$

Table 3.1: Results for the advection discretisation with $p = 0$

Mesh No	No Dofs	$\|u - u_{DG}\|_{L_2(\Omega)}$	Order	$J(u) - J(u_{DG})$	Order
1	4	2.522607E-01		-4.246733E-02	
2	16	1.374662E-01	0.88	-2.616338E-02	0.70
3	64	7.336309E-02	0.91	-1.482783E-02	0.82
4	256	3.833225E-02	0.94	-7.976484E-03	0.89
5	1024	1.970891E-02	0.96	-4.155732E-03	0.94
6	4096	1.002805E-02	0.97	-2.125016E-03	0.97

Table 3.2: Results for the advection discretisation with $p = 1$

Mesh No	No Dofs	$\|u - u_{DG}\|_{L_2(\Omega)}$	Order	$J(u) - J(u_{DG})$	Order
1	16	2.269749E-02		-3.092564E-03	
2	64	5.994330E-03	1.92	-4.130726E-04	2.90
3	256	1.539235E-03	1.96	-5.368232E-05	2.94
4	1024	3.899229E-04	1.98	-6.854942E-06	2.97
5	4096	9.812421E-05	1.99	-8.667161E-07	2.98
6	16384	2.461198E-05	2.00	-1.089987E-07	2.99

3.1.3 Numerical Examples

Consider the advection problem (3.5) with a discontinuous Galerkin discretisation given by (3.6). Prescribing the solution $u = \exp(x - y)$ we can consider the error in the L_2–norm, $\|u - u_{DG}\|_{L_2(\Omega)}$. We let $\Omega = [0, 1] \times [0, 1]$ with boundary conditions on the inflow boundary Γ^- given by the analytical solution u. We also consider the linear mean value functional

$$J(u) = \int_\Omega \omega u \, \mathrm{d}\boldsymbol{x},$$

with weighting $\omega = 1$. The results of performing uniform refinement of the mesh (we start with a mesh of 4 quadrilateral elements) with polynomial degrees $p = 0, 1, 2$ are given in Tables 3.1, 3.2 and 3.3 respectively. For a quadrilateral mesh of this type the mesh size $h = \sqrt{2}/N$, $N = 2, 4, 8, 16, 32, 64$ and the order of convergence can then be computed as

$$\text{order} = \frac{\log(\|u - u^{n+1}_{DG}\|_{L_2(\Omega)}) - \log(\|u - u^{n}_{DG}\|_{L_2(\Omega)})}{\log(h(n+1)) - \log(h(n))}, \quad n = 1, \ldots 5.$$

We note that once we are in the asymptotic regime we observe the expected orders of

Table 3.3: Results for the advection discretisation with $p = 2$

Mesh No	No Dofs	$\|u - u_{DG}\|_{L_2(\Omega)}$	Order	$J(u) - J(u_{DG})$	Order
1	36	9.566985E-04		2.704227E-06	
2	144	1.237145E-04	2.95	1.075672E-07	4.65
3	576	1.567134E-05	2.98	3.726066E-09	4.85
4	2304	1.970045E-06	2.99	1.230567E-10	4.92
5	9216	2.468914E-07	3.00	3.979483E-12	4.95
6	36864	3.089926E-08	3.00	1.350031E-13	4.88

convergence, namely $O(h^{p+1})$ for the L_2–norm and $O(h^{2(p+1)-1})$ for the error made in approximating the functional. We shall return to these results later on in this thesis.

3.2 2D Mono Energetic Steady State Neutron Transport

Before extending the discretisation of the advection–reaction problem considered previously to the discretisation of the neutron transport equation we briefly consider the existence and uniqueness of solutions to the neutron transport equation before discretising the neutron transport equation.

3.2.1 Existence and Uniqueness of a Solution for the Neutron Transport Equation

The book [59] by Dautray and Lions includes an extensive discussion into the existence and uniqueness of solutions to the neutron transport problem in both the stationary and time dependent cases. The result presented here assumes that the appropriate boundary condition g, and the material data, i.e., the cross sections do not depend on the angle φ. In addition, this result is also only valid for a subcritical source problem. Defining the operator $\mathcal{L}$, known as the transport operator in the following fashion

$$\mathcal{L}\psi := -\boldsymbol{\mu} \cdot \nabla_x \psi(\boldsymbol{x}, \varphi) - \Sigma_t \psi(\boldsymbol{x}, \varphi) + \frac{1}{2\pi} \int_0^{2\pi} (\Sigma_s + \nu\Sigma_f)\psi(\boldsymbol{x}, \varphi')\, \mathrm{d}\varphi',$$

problem (1.6) with appropriate boundary conditions can be written as

$$\left.\begin{aligned} -\mathcal{L}\psi(\boldsymbol{x}, \theta) &= Q(\boldsymbol{x}, \theta) \qquad \text{in} \Omega \times I, \\ \psi(\boldsymbol{x}, \theta) &= g(\boldsymbol{x}, \theta) \qquad \text{on} \Gamma_{\boldsymbol{\mu}}^{-}. \end{aligned}\right\} \tag{3.11}$$

where $\Gamma_{\boldsymbol{\mu}}^{-}$ is the inflow boundary with respect to the direction $\boldsymbol{\mu}$. A suitable choice of function space in which to look for a solution must be made, we wish to investigate

existence and uniqueness in the space $L^p(\Omega \times I)$ This operator is naturally defined in the space $L^1(\Omega \times I)$ since we know that the scalar flux is defined to be

$$\phi = \int_0^{2\pi} \psi(\boldsymbol{x}, \theta') \, \mathrm{d}\theta'.$$

In the above the space $L^1(\Omega \times I)$ is defined to be the space of real-valued measurable functions $f \in \Omega \times I$ such that

$$||f||_{L^1(\Omega \times I)} \equiv \int_{\Omega \times I} |f(\boldsymbol{x}, \theta)| \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\theta < \infty$$

However, we take the result from [59] for the case when $p = \infty$. The domain of the operator can be expressed as the set $D(\mathcal{L}) = \{\psi : \psi \in L^1(\Omega \times I), \mathcal{L}\psi \in L^1(\Omega \times I)\}$. We now recall the result of Proposition 7, given on page 245 of [59].

Proposition Consider the following problem: find ψ satisfying (3.11) under the following assumptions,

- $\Sigma_t \in L^\infty(\Omega \times I), \Sigma_t > \sigma_0 \geq 0,$
- $\int_0^{2\pi} \frac{\Sigma_s + \nu\Sigma_f}{2\pi} \, \mathrm{d}\theta \leq \beta\sigma_t, \qquad 0 \leq \beta < 1,$
- $Q \in L^\infty(\Omega \times I),$
- g is a positive function such that $g \in L^\infty(\Gamma_{\boldsymbol{\mu}}^-)$.

Then the problem (3.11) has a solution in $L^\infty(\Omega \times I)$; this solution is unique and satisfies

$$\|\psi\|_\infty \leq \sup(\|g\|_\infty, \alpha\|Q\|_\alpha),$$

where $\alpha > 0$ is a constant.

It should be noted that analogous results hold for the spaces $L^p(\Omega \times I)$ with $p \in [1, \infty)$ and can be obtained by considering a lifting of the boundary conditions which reduces the problem to the following

$$\left.\begin{aligned} -\mathcal{L}\psi(\boldsymbol{x}, \theta) &= Q(\boldsymbol{x}, \theta), \qquad \text{in } \Omega \times I, \\ \psi(\boldsymbol{x}, \theta) &= 0 \qquad \text{on } \Gamma_{\boldsymbol{\mu}}^-. \end{aligned}\right\} \tag{3.12}$$

The proof is then given in [59], Theorem 4, page 241.

3.2.2 A Discrete Ordinates Approach

We can extend the advection reaction solver described in the last section into a numerical method for the Neutron Transport equation by supplementing the discretisation of the spatial domain with an appropriate discretisation of the angular domain. The simplest angular discretisation method is the discrete ordinates method as described in Section 2.3.

We consider the two dimensional mono energetic steady state equation, which for convenience we reproduce here.

$$\left.\begin{aligned} \boldsymbol{\mu}\cdot\nabla_x\psi(\boldsymbol{x},\varphi)+\Sigma_t\psi(\boldsymbol{x},\varphi) &= \frac{1}{2\pi}\int_0^{2\pi}(\Sigma_s+\nu\Sigma_f)\psi(\boldsymbol{x},\varphi')\,\mathrm{d}\varphi'+Q(\boldsymbol{x},\varphi),\\ \psi(\boldsymbol{x},\varphi) &= g(\boldsymbol{x},\varphi) \qquad \text{on } \Gamma^-_{\boldsymbol{\mu}}, \end{aligned}\right\} \tag{3.13}$$

with $\boldsymbol{\mu} = (\cos(\varphi),\sin(\varphi))$. As noted in the one dimensional illustration in Section 2.3, the discrete ordinates approximation is essentially the application of a quadrature rule to compute the integral over angle of the angular flux to arrive at the scalar flux, i.e.,

$$\int_0^{2\pi}\psi(\boldsymbol{x},\varphi)\,\mathrm{d}\varphi \approx \sum_{i=1}^{N}\omega_i\psi(\boldsymbol{x},\varphi_i)$$

resulting in N directions (or ordinates) $\mathcal{O} = \{\boldsymbol{\mu}(\varphi_1),\boldsymbol{\mu}(\varphi_2),\ldots,\boldsymbol{\mu}(\varphi_N)\}$ with associated positive weights ω_i.

For consistency with the error analysis presented in [92] we require that our choice of ordinates and weights possess certain properties (see [27]). Firstly, we wish to ensure stability of the discrete ordinates scheme by placing a limit on the non uniformity of the distribution of the discrete ordinates. To this end, we consider a splitting of the tensor product of the set of ordinates with itself, denoted by $\mathcal{O}^2$. Given an $\varepsilon > 0$, we can define the following splitting:

$$\sum_{(\mu,\nu)\in\mathcal{O}^2} = \sum_{(\mu,\nu)\in I'_\varepsilon} + \sum_{(\mu,\nu)\in I''_\varepsilon}$$

where

$$\begin{aligned} I'_\varepsilon &= \left\{(\mu,\nu)\in\mathcal{O}^2 : \min\left(\gamma(\mu,\nu),\gamma(\mu,d^k),\gamma(\nu,d^k)\right)\geq\varepsilon, \quad k=1,\ldots,M\right\},\\ I''_\varepsilon &= \left\{(\mu,\nu)\in\mathcal{O}^2 : (\mu,\nu)\notin I'_\varepsilon\right\}, \end{aligned}$$

and $\gamma(\mu,\nu) = \sin(d(\mu,\nu))$ with $d(\mu,\nu)$ being the smallest angle between μ and ν. The values d^k are defined to be the directions of the M sides (in our case 4) of Ω, the spatial

domain. The following property ensures the stability of the discrete ordinates scheme

$$\sum_{\varepsilon}'' \omega_\mu \omega_\nu \to 0 \qquad \text{as} \quad \max(1/N, \varepsilon) \to 0. \tag{3.14}$$

Johnson and Pitkäranta also required the following approximation property to be satisfied for their results to hold (these results will be summarised at the end of this chapter), namely,

$$\left| \int_I u(\boldsymbol{\mu}) \, \mathrm{d}\boldsymbol{\mu} - \sum_{\mu \in \mathcal{Q}} u(\mu)\omega_\mu \right| \leq \frac{C}{N} \int_0^{2\pi} \left| \frac{\mathrm{d}u}{\mathrm{d}\varphi} \right| \, \mathrm{d}\varphi, \tag{3.15}$$

where C is a positive constant, independent of the number of ordinates N and $\boldsymbol{\mu} = (\cos(\varphi), \sin(\varphi))$.

Here we select one of the simplest possible choices that satisfies properties (3.14) and (3.15): namely, the uniformly spaced ordinates,

$$\mathcal{O} = \left\{ \left(\cos \frac{2\pi i}{N}, \sin \frac{2\pi i}{N} \right) \right\}_{i=1}^{N},$$

resulting in a uniform weighting of $\omega_i = 2\pi/N$. This choice of ordinates results in splitting the range of $\varphi \in [0, 2\pi)$ into n equal segments with the ordinates using the midpoint of these segments.

Remark 9. *This is the simplest choice of ordinates, in fact better results may be obtained by weighting the ordinates close to* 0 *and* 2π*; however, care must be taken to ensure the stability property is still satisfied.*

To solve the discrete ordinates problem, we now discretise the spatial problem, which for a fixed direction $\boldsymbol{\mu}_i = \boldsymbol{\mu}(\varphi_i)$, is an advection reaction equation of the form,

$$\left.\begin{aligned} \boldsymbol{\mu}_i \cdot \nabla_x \psi(\boldsymbol{x}, \varphi_i) + \Sigma_t \psi(\boldsymbol{x}, \varphi_i) = Q + \frac{\Sigma_s + \nu\Sigma_f}{2\pi} \sum_{j=1}^{N} \omega_j \psi(\boldsymbol{x}, \varphi_j) \qquad & \text{in } \Omega, \\ u = g \qquad \text{on } \Gamma^-, & \end{aligned}\right\} \tag{3.16}$$

for $i = 1, \ldots, N$.

To this end, we use the spatial finite element space defined previously and apply the DG discretisation (3.6) of the advection problem given in Section 3.1.1 for each fixed ordinate $\boldsymbol{\mu}_i$ to obtain the fully discrete problem: for each $i = 1, \ldots N$ find $\psi_h \in V_{h,p}$ such that

$$A_{NT}(\psi_h, v_h) = l_{NT}(v_h) \quad \forall\, v_h \in V_{h,p} \tag{3.17}$$

with the bilinear form $A_{NT} : V_{h,p} \times V_{h,p} \to \mathbb{R}$ given by

$$\sum_{\kappa \in \mathcal{T}} \left(- \int_\kappa (\boldsymbol{\mu}_i \psi_h) \cdot \nabla v_h \, \mathrm{d}\boldsymbol{x} + \int_{\partial\kappa} \mathcal{H}(\psi_h^+, \psi_h^-, \boldsymbol{n}_\kappa) v_h^+ \, \mathrm{d}s \right)$$

and linear functional $l_{NT} : V_h \to \mathbb{R}$,

$$l_{NT}(v_h) = \sum_{\kappa \in \mathcal{T}} \left(\int_\kappa Q v_h \, \mathrm{d}\boldsymbol{x} + \int_\kappa \frac{\Sigma_s + \Sigma_f}{2\pi} \left(\sum_{j=1}^{N} \omega_j \psi_h(\boldsymbol{x}, \varphi_j) \right) \, \mathrm{d}\boldsymbol{x} \right)$$

Note, in the above, the addition of the coupling term which is being approximated by the use of the discrete ordinates quadrature. This means that we are not just solving N discrete equations, but in fact we have to use an iterative procedure to attain converge to the exact scalar flux, the simplest of which is the industry standard iterative procedure, source iteration (SI) [2]; this will be discussed further in Chapter 4. Informally we take some initial guess for the solution on each ordinate $\boldsymbol{\mu}_i$ and compute an initial approximation to the coupling integral (an approximation to the scalar flux) appearing in the neutron transport equation. We then solve the N discretised equations corresponding to each $\boldsymbol{\mu}_i$ and update our numerical approximation of the scalar flux. We then repeat this procedure until the differences between successive approximations to the scalar flux are within some pre described tolerance.

Some numerical results for the combination of discrete ordinates and DG in space are given in Chapter 5.

3.2.3 Full DG Discretisation

In the previous section we developed a Discontinuous Galerkin discretisation of the two dimensional mono energetic steady state neutron transport equation, where we employed a discrete ordinates method in the angular domain.
The low order of the angular approximation of the discrete ordinates approach severely restricts the accuracy and convergence rates of the resulting method; in Chapter 5 we will see that as we increase the polynomial order in space, even if we recover higher order convergence at first, eventually the angular error will dominate and we will be constrained to a first order scheme. To rectify this issue, we seek to develop a solver that is arbitrary order in angle, as well as in space. That is a solver which is $DG(p)$ in space and $DG(q)$ in angle, for p,q arbitrary non negative integers, for the neutron transport problem

$$\left.\begin{aligned} \boldsymbol{\mu} \cdot \nabla_x \psi(\boldsymbol{x}, \varphi) + \Sigma_t \psi(\boldsymbol{x}, \varphi) &= \frac{1}{2\pi} \int_0^{2\pi} (\Sigma_s + \nu\Sigma_f) \psi(\boldsymbol{x}, \varphi') \, \mathrm{d}\varphi' + Q(\boldsymbol{x}, \varphi), \\ \psi(\boldsymbol{x}, \varphi) &= g(\boldsymbol{x}, \varphi) \qquad \text{on } \Gamma_{\boldsymbol{\mu}}^-, \end{aligned}\right\} \tag{3.18}$$

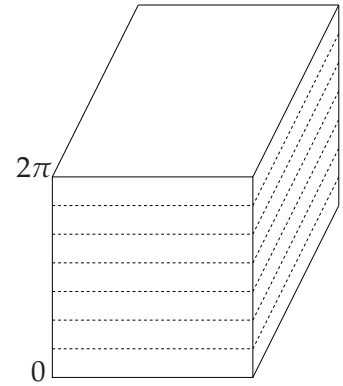


Figure 3.3: The domain $\Omega \times I$ as a collection of space - angle slabs.

where $\Gamma^-_{\boldsymbol{\mu}}$ denotes the dependence on the advective direction $\boldsymbol{\mu}$ of the inflow portion of the boundary Γ as before. To discretise this integro–differential equation, we consider splitting the combined domain $\Omega \times I$ into space–angle slabs as shown in Figure 3.3. In order to rigorously define the DG method, for both the spatial and angular components we have to define an appropriate partition of the respective domain into a mesh, and provide a definition of the finite element space in this case. To this end, we define a finite element space on the spatial and angular domains separately; the finite element space defined on the full domain will be constructed as the tensor product of these two spaces. The definitions presented for the spatial domain are analogous to those presented previously, however, for completeness we repeat them here; note the presence of a subscript Ω on the spatial definitions here to make it explicit which quantities belong to the spatial domain.

We define a partition of the spatial domain $\mathcal{T}_\Omega = \{\kappa_\Omega\}$ in to a shape regular partition of the domain Ω into open element domains κ_Ω which are at most 1-irregular, as in Section 3.1

The mesh function h_Ω is a piecewise constant function with $h_\Omega(\boldsymbol{x}) = h_{\kappa_\Omega} = \operatorname{diam}(\kappa_\Omega)$ when $\boldsymbol{x} \in \kappa_\Omega$. We assume that each element κ_Ω can be expressed as a mapping of a fixed reference element $\hat{\kappa}$, that is $\kappa_\Omega = F_{\kappa_\Omega}(\hat{\kappa})$, where F_{κ_Ω} is smooth and bijective, for each $\kappa_\Omega \in \mathcal{T}_\Omega$. The reference element $\hat{\kappa}$, in this case can be either the open unit triangle

$$\hat{\kappa}_T = \{\hat{\boldsymbol{x}} = (\hat{x_1}, \hat{x_2}) \in \mathbb{R}^2 : 0 < \hat{x_1} + \hat{x_2} < 1, \hat{x_i} > 0,\ i = 1, 2\},$$

or the open hypercube $\hat{\kappa}_C = (-1, 1)^2$ in $\mathbb{R}^2$.

On $\hat{\kappa}$ we define the following spaces of polynomials of degree $p \geq 0$, with the convention that α is a multi-index:

$$\mathcal{P}_p = \operatorname{span}\{\hat{\boldsymbol{x}}^\alpha : 0 < |\alpha| \leq p\},$$
$$\mathcal{Q}_p = \operatorname{span}\{\hat{\boldsymbol{x}}^\alpha : 0 \leq \alpha_i \leq p, 1 \leq i \leq 2\}.$$

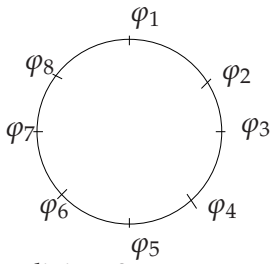


Figure 3.4: The angular domain split into 8, not necessarily uniform, angular elements

For each $\kappa_\Omega \in \mathcal{T}_\Omega$ we denote by $p_{\kappa_\Omega} \geq 0$ the polynomial approximation degree employed on κ_Ω. Forming the vectors $\boldsymbol{p} = \{p_{\kappa_\Omega} : \kappa_\Omega \in \mathcal{T}_\Omega\}$ and $\boldsymbol{F}_\Omega = \{F_{\kappa_\Omega} : \kappa_\Omega \in \mathcal{T}_\Omega\}$, we introduce the discontinuous hp-finite element space

$$S^{\boldsymbol{p}}(\Omega, \mathcal{T}_\Omega, \boldsymbol{F}_\Omega) = \{u \in L_2(\Omega) : u|_{\kappa_\Omega} \circ F_{\kappa_\Omega} \in \mathcal{S}_{p_{\kappa_\Omega}},\ \kappa_\Omega \in \mathcal{T}_\Omega\}, \tag{3.19}$$

where

$$\mathcal{S}_{p_\kappa} = \begin{cases} \mathcal{P}_{p_\kappa}(\hat{\kappa}) & \text{if } F_\kappa^{-1}(\kappa) = \hat{\kappa}_T, \\ \mathcal{Q}_{p_\kappa}(\hat{\kappa}) & \text{if } F_\kappa^{-1}(\kappa) = \hat{\kappa}_C. \end{cases}$$

If the polynomial degree is uniform, i.e., $p_{\kappa_\Omega} = p$ for all $\kappa \in \mathcal{T}_h$ then we simply write $S^p(\Omega, \mathcal{T}_\Omega, \boldsymbol{F}_\Omega)$. For any element $\kappa_\Omega \in \mathcal{T}_\Omega$, we let $\partial\kappa_\Omega$ be the union of 1–dimensional open faces of κ_Ω. Similarly, for any $\boldsymbol{x} \in \partial\kappa_\Omega$, $\boldsymbol{n}_{\kappa_\Omega}(\boldsymbol{x})$ denotes the unit outward normal vector to $\partial\kappa_\Omega$ at $\boldsymbol{x}$. Using these defintions, we can define, the inflow and outflow parts of $\partial\kappa_\Omega$, respectively, by

$$\partial_{\boldsymbol{\mu}}^- \kappa_\Omega = \{x \in \partial\kappa_\Omega : \boldsymbol{\mu} \cdot \boldsymbol{n}(\boldsymbol{x}) < 0\},$$
$$\partial_{\boldsymbol{\mu}}^+ \kappa_\Omega = \{x \in \partial\kappa_\Omega : \boldsymbol{\mu} \cdot \boldsymbol{n}(\boldsymbol{x}) \geq 0\}.$$

Note that the inflow portion of an element boundary will be different for each angle due to the dependence on $\boldsymbol{\mu}$.

Remark 10. *In the above the subscript $\boldsymbol{\mu}$ denotes the explicit dependence on the direction $\boldsymbol{\mu}$ of the inflow and outflow boundaries of an element κ. This means that on different angular elements we obtain different matrix sparsity patterns and we shall see that in practice when we approximate integrals over faces with a quadrature rule it is possible for a face to be classed as both inflow and outflow.*

We now consider the angular domain. As before we need to mesh the circumference of the unit circle. An example of such a mesh is shown in Figure 3.4; here, the angular domain is split into 8 angular elements, not necessarily uniform. To define the angular finite element space we proceed in the same manner as for the spatial finite element space.

We define $\mathcal{T}_I = \{\kappa_I\}$ to be a partition of the angular domain I into open intervals κ_I. The mesh function corresponding to this partition is denoted by h_I; h_I is a piecewise constant function such that $h_I|_{\kappa_I} = h_{\kappa_I} = \text{diam}(\kappa_I)$. As before, each element of the angular mesh κ_I can be expressed as a mapping of a fixed reference interval $J = [-1, 1]$, so that $\kappa_I = F_{\kappa_I}(J)$. Here, F_{κ_I} is a smooth and bijective function for each $\kappa_I \in \mathcal{T}_I$.
On J we define the space of polynomials of degree $q \geq 0$, i.e.,

$$\mathcal{P}_q = \text{span}\{y^\beta : 0 \leq \beta \leq q\}.$$

For each $\kappa_I \in \mathcal{T}_I$ we assign an integer $q_{\kappa_I} \geq 0$; forming the vectors $\boldsymbol{q} = \{q_{\kappa_I} : \kappa_I \in \mathcal{T}_I\}$ and $\boldsymbol{F}_I = \{F_{\kappa_I} : \kappa_I \in \mathcal{T}_I\}$, we introduce the discontinuous hp-finite element space

$$S^{\boldsymbol{q}}(\mathcal{I}, \mathcal{T}_I, \boldsymbol{F}_I) = \left\{ w \in L_2(I) : w|_{\kappa_I} \circ F_{\kappa_I} \in \mathcal{P}_{q_{\kappa_I}},\ \ \kappa_I \in \mathcal{T}_I \right\}.$$

If the polynomial degree is uniform, i.e., $q_{\kappa_I} = q$ for all $\kappa_I \in \mathcal{T}_I$, then as before, we simply write $S^q(\Omega, \mathcal{T}_h, \boldsymbol{F}_I)$.

To define the full discontinuous hp-finite element space we tensor product the spatial discontinuous hp-finite element space with the angular discontinuous hp-finite element space, giving

$$V^{h_\Omega, h_I} = S^{\boldsymbol{p}}(\Omega, \mathcal{T}_\Omega, \boldsymbol{F}_\Omega) \times S^{\boldsymbol{q}}(I, \mathcal{T}_I, \boldsymbol{F}_I). \tag{3.20}$$

Then all $v \in V^{h_\Omega, h_I}$ can be expressed as $v = v_\Omega v_I$ such that $v_\Omega \in S^{\boldsymbol{p}}(\Omega, \mathcal{T}_\Omega, \boldsymbol{F}_\Omega)$ and $v_I \in S^{\boldsymbol{q}}(I, \mathcal{T}_I, \boldsymbol{F}_I)$.

The Two Dimensional DG Discretisation

First we make that observation that $\boldsymbol{\mu}$ is spatially independent; i.e., we can write

$$\boldsymbol{\mu} \cdot \nabla_{\boldsymbol{x}} \psi(\boldsymbol{x}, \varphi) = \nabla_{\boldsymbol{x}} \cdot (\boldsymbol{\mu} \psi(\boldsymbol{x}, \varphi)).$$

Thereby we arrive at the following equivalent formulation for the integro–differential equation

$$\nabla_{\boldsymbol{x}} \cdot (\boldsymbol{\mu} \psi(\boldsymbol{x}, \varphi)) + \Sigma_t \psi(\boldsymbol{x}, \varphi) = \frac{\Sigma_s + \nu \Sigma_f}{2\pi} \int_0^{2\pi} \psi(\boldsymbol{x}, \varphi')\, \mathrm{d}\varphi' + Q(\boldsymbol{x}, \varphi) \qquad \text{in } \Omega \times I \tag{3.21}$$

To solve (3.21) numerically we shall discretise in both space and angle using a Discontinuous Galerkin method; the resulting system of equations will be solved based on employing an iterative scheme.

To this end, we multiply (3.21) by a smooth test function and integrate over space then angle, where we have used integration by parts in the spatial domain to give the following

$$\begin{aligned}\sum_{\kappa_I\in\mathcal{T}_I}\sum_{\kappa_\Omega\in\mathcal{T}_\Omega}\Big[\int_{\kappa_I}\int_{\kappa_\Omega}\{-\psi\boldsymbol{\mu}\cdot\nabla_{\boldsymbol{x}}v+\Sigma_t\psi v\}\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\varphi+\int_{\kappa_I}\int_{\partial\kappa_\Omega}(\boldsymbol{\mu}\psi\cdot\boldsymbol{n}_{\kappa_\Omega})v\,\mathrm{d}s\,\mathrm{d}\varphi\Big]\\ =\sum_{\kappa_I\in\mathcal{T}_I}\sum_{\kappa_\Omega\in\mathcal{T}_\Omega}\Big[\int_{\kappa_I}\int_{\kappa_\Omega}\frac{\Sigma_s+\nu\Sigma_f}{2\pi}\int_0^{2\pi}\psi(\boldsymbol{x},\varphi')v\,\mathrm{d}\varphi'\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\varphi\Big]\\ +\sum_{\kappa_I\in\mathcal{T}_I}\sum_{\kappa_\Omega\in\mathcal{T}_\Omega}\Big[\int_{\kappa_I}\int_{\kappa_\Omega}Q(\boldsymbol{x},\varphi)v\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\varphi\Big].\end{aligned}\tag{3.22}$$

To discretise (3.22) we replace ψ by the Discontinuous Galerkin finite element approximation ψ_h and the test function v by v_h, where both ψ_h and v_h belong to the finite element space V^{h_Ω,h_θ} as defined above. As before we introduce the Lax–Friedrichs flux:

$$\mathcal{H}(\psi_h^+,\psi_h^-,\boldsymbol{n})=\frac{1}{2}\left(\boldsymbol{\mu}\cdot\boldsymbol{n}_{\kappa_\Omega}+|\boldsymbol{\mu}\cdot\boldsymbol{n}_{\kappa_\Omega}|\right)\psi_h^+ +\frac{1}{2}\left(\boldsymbol{\mu}\cdot\boldsymbol{n}_{\kappa_\Omega}-|\boldsymbol{\mu}\cdot\boldsymbol{n}_{\kappa_\Omega}|\right)\psi_h^-,$$

to deal with the elemental boundary integrals. The Lax–Friedrichs flux with the arguments given above can be used on elemental faces that are in the interior of the spatial domain. For elemental faces coincident with the domain boundary the concept of ψ^- does not make sense since there is only an element on one side of the face. Instead, the second argument is replaced by ψ_{Γ_μ}, which is defined as follows

$$\begin{aligned}\psi_{\Gamma_\mu}(\psi_h^+)&=\psi_h^+ \quad \text{for } \boldsymbol{x}\in\Gamma_\mu^+,\\ \psi_{\Gamma_\mu}(\psi_h^+)&=g \quad \text{for } \boldsymbol{x}\in\Gamma_\mu^-.\end{aligned}$$

It should be noted that this numerical flux implicitly depends on the angle φ since $\boldsymbol{\mu}$ is a function of φ; similarly, the inflow and outflow domain boundaries depend on the direction $\boldsymbol{\mu}$. Using this numerical flux function we arrive at the following DG scheme: find $\psi_h\in V^{h_\Omega,h_I}$ such that

$$\begin{aligned}\sum_{\kappa_I\in\mathcal{T}_I}\sum_{\kappa_\Omega\in\mathcal{T}_\Omega}\Big[\int_{\kappa_I}\int_{\kappa_\Omega}\{-\psi_h\boldsymbol{\mu}\cdot\nabla_{\boldsymbol{x}}v_h+\Sigma_t\psi_h v_h\}\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\theta+\int_{\kappa_I}\int_{\partial\kappa_\Omega\setminus\Gamma}\mathcal{H}(\psi_h^+,\psi_h^-,\boldsymbol{n})v_h^+\,\mathrm{d}s\,\mathrm{d}\theta\\ +\int_{\kappa_I}\int_{\partial\kappa_\Omega\cap\Gamma}\mathcal{H}(\psi_h^+,\psi_\Gamma^{\boldsymbol{\mu}}(\psi_h^+),\boldsymbol{n})v_h^+\,\mathrm{d}s\,\mathrm{d}\theta\Big]\\ =\sum_{\kappa_I\in\mathcal{T}_I}\sum_{\kappa_\Omega\in\mathcal{T}_\Omega}\Big[\int_{\kappa_I}\int_{\kappa_\Omega}\frac{\Sigma_s+\nu\Sigma_f}{2\pi}\int_0^{2\pi}\psi_h(\boldsymbol{x},\theta')v_h\,\mathrm{d}\theta'\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\theta\Big]\\ +\sum_{\kappa_I\in\mathcal{T}_I}\sum_{\kappa_\Omega\in\mathcal{T}_\Omega}\Big[\int_{\kappa_I}\int_{\kappa_\Omega}Q(\boldsymbol{x},\theta)v_h\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\theta\Big]\quad\forall v_h\in V^{h_\Omega,h_I}.\end{aligned}\tag{3.23}$$

Here, the subscript h appearing in ψ_h and v_h denotes the pairing $h=(h_\Omega,h_I)$.
Note that the left hand side of (3.23) is an advection–reaction problem. This infinite

(there are infinite possible values for $\boldsymbol{\mu}$) set of advection–reaction equations is coupled together by the integral operator on the right hand side. The resulting set of linear equations arising from the Discontinuous Galerkin discretisation will be solved by employing the source iteration that was employed for the discrete ordinates discretisation.

3.3 The "Pseudo-3D" Neutron Transport Discretisation

So far we have considered the model problem posed in [92] among others. This problem however, has limited applicability in the real world: in the full form of the neutron transport equation there are three spatial dimensions. We now develop the discretisation of a problem that can be applied to real world industrial benchmarks.

3.3.1 Derivation of the Pseudo 3D Neutron Transport Problem

We now consider the spatial domain considered previously as a slice through a three dimensional domain that has been constructed subject to some conditions. The first of these conditions is that when the full domain is cut across the xy plane at any point z_i, then the resulting two dimensional plane must have the same dimensions and properties as that obtained by slicing at a different point z_j. This means that the physical plane will look the same regardless of where the slice has been taken, and so the spatial domain of our discretisation remains two dimensional; this leads to a significant computational saving when compared to solving the full three dimensional problem. A further condition is that the three dimensional domain extends to infinity in both directions in x_3. These approximations are valid for many reactor physics computations, such as determining criticality, as there is little variation in the x_3 spatial component in these situations and so a reactor core geometry can be safely assumed to be constant and infinite in x_3. These two conditions ensure that the angular flux on any infinitesimally thin slice of the three dimensional domain is the same as that on any other.

As we are now considering three dimensional problems, modelling the direction of a neutron's travel becomes harder since it is no longer constrained to a planar surface. Because of this, in addition to the azimuthal angle, φ, we introduce the polar angle, θ.

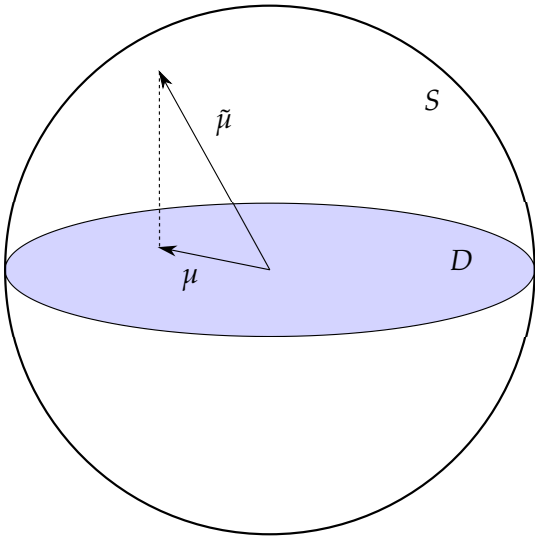


Figure 3.5: Projection of unit vector $\tilde{\mu} \in S$ onto a vector $\mu \in D$.

Below, we pose the new model problem, as stated in [15]:

$$\boldsymbol{\mu} \cdot \nabla_x \psi(\boldsymbol{x}, \boldsymbol{\mu}) + \sigma_t \psi(\boldsymbol{x}, \boldsymbol{\mu}) = \frac{1}{4\pi} \int_D (\sigma_s + \nu\sigma_f)\psi(\boldsymbol{x}, \boldsymbol{\mu}') \left(1 - |\boldsymbol{\mu}'|^2\right)^{-1/2} \, \mathrm{d}\boldsymbol{\mu}'$$
$$+ Q(\boldsymbol{x}, \boldsymbol{\mu}) \quad \text{in } \Omega \times D, \tag{3.24}$$
$$\psi(\boldsymbol{x}, \boldsymbol{\mu}) = g(\boldsymbol{x}, \boldsymbol{\mu}) \quad \text{on } \Gamma^-_{\boldsymbol{\mu}}. \tag{3.25}$$

In the above Ω remains the spatial domain but now the angular domain D is the unit disc, $D := \{\boldsymbol{\mu} \in \mathbb{R}^2 : |\boldsymbol{\mu}| \leq 1\}$. Instead of $\boldsymbol{\mu} = (\cos(\varphi), \sin(\varphi))^\top$ which we had in equation (1.6) we now have the following advective direction

$$\boldsymbol{\mu} = \begin{pmatrix} \sin(\theta)\cos(\varphi) \\ \sin(\theta)\sin(\varphi) \end{pmatrix},$$

where φ is the azimuthal angle, such that $0 \leq \varphi \leq 2\pi$ and θ is the polar angle, such that $0 \leq \theta \leq \frac{\pi}{2}$ (since we consider only the top hemisphere). The angular domain is the unit disc, since this is the projection of all points on the unit sphere down on to $\mathbb{R}^2$ as shown in Figure 3.5.

One further difference between our original problem and the 'pseudo 3D' problem we are now investigating is the addition of the term $\left(1 - |\boldsymbol{\mu}'|^2\right)^{-1/2}$ in the coupling integral on the right hand side of equation (3.24). Following the explanation in [15] it can be seen that this is due solely to the geometry of the problem being considered.

Working in a spherical coordinate system we have that the area element of the unit sphere is given by

$$\mathrm{d}A = \sin(\theta)\, \mathrm{d}\theta\, \mathrm{d}\varphi, \qquad \text{for } 0 \leq \varphi \leq 2\pi \text{ and } \quad 0 \leq \theta \leq \pi.$$

Now using the definition of $\boldsymbol{\mu}$ previously stated we have

$$\begin{aligned}|\boldsymbol{\mu}| &= \sqrt{\left(\sin(\theta)\cos(\varphi)\right)^2 + \left(\sin(\theta)\sin(\varphi)\right)^2} \\ &= \sqrt{\sin^2(\theta)\cos^2(\varphi) + \sin^2(\theta)\sin^2(\varphi)} \\ &= \sqrt{\sin^2(\theta)\left(\cos^2(\varphi) + \sin^2(\varphi)\right)} \\ &= \sin(\theta).\end{aligned}$$

As a consequence of this

$$\frac{\mathrm{d}\,|\boldsymbol{\mu}|}{\mathrm{d}\theta} = \cos(\theta) \quad \Rightarrow \quad \mathrm{d}\theta = \frac{\mathrm{d}|\boldsymbol{\mu}|}{\cos(\theta)}.$$

Hence, by way of a standard trigonometric identity,

$$\begin{aligned}\sin(\theta)\,\mathrm{d}\theta &= \frac{\sin(\theta)}{\cos(\theta)}\mathrm{d}|\boldsymbol{\mu}| \\ &= \frac{|\boldsymbol{\mu}|}{\cos(\theta)}\mathrm{d}|\boldsymbol{\mu}| \\ &= \frac{|\boldsymbol{\mu}|}{\left(1-|\boldsymbol{\mu}|^2\right)^{1/2}}\mathrm{d}|\boldsymbol{\mu}|.\end{aligned}$$

Now using this, together with the fact (obtained by converting from cartesian to polar coordinates) that,

$$\begin{aligned}\mathrm{d}\boldsymbol{\mu} &= \mathrm{d}\mu_1\mathrm{d}\mu_2 \\ &= |\boldsymbol{\mu}|\mathrm{d}|\boldsymbol{\mu}|\mathrm{d}\varphi,\end{aligned}$$

we can deduce,

$$\begin{aligned}\mathrm{d}A &= \sin(\theta)\mathrm{d}\theta\mathrm{d}\varphi \\ &= \frac{|\boldsymbol{\mu}|}{\left(1-|\boldsymbol{\mu}|^2\right)^{1/2}}\mathrm{d}|\boldsymbol{\mu}|\mathrm{d}\varphi \\ &= \frac{1}{\left(1-|\boldsymbol{\mu}|^2\right)^{1/2}}\mathrm{d}\boldsymbol{\mu}\end{aligned}$$

The derivation of the scheme proceeds in the same manner as for the two dimensional case; the only changes being the addition of the weighting $\left(1-|\boldsymbol{\mu}'|^2\right)^{-1/2}$ to the computation of the scalar flux and coupling term, and the different definition of the advective direction $\boldsymbol{\mu}$. The mesh in the angular domain is now a triangulation of the unit disc as opposed to a discretisation of the surface of the unit circle; this is described in detail in Section 4.2.1

We, therefore, have the following DG scheme for the pseudo 3D problem: find $\psi_h \in V^{h_\Omega,h_I}$ such that

$$\begin{aligned}
&\sum_{\kappa_D\in\mathcal{T}_D}\sum_{\kappa_\Omega\in\mathcal{T}_\Omega}\Bigg[\int_{\kappa_D}\int_{\kappa_\Omega}\{-\psi_h\boldsymbol{\mu}\cdot\nabla_{\boldsymbol{x}}v_h+\Sigma_t\psi_h v_h\}\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\theta\mathrm{d}\varphi\\
&\quad+\int_{\kappa_D}\int_{\partial\kappa_\Omega\setminus\Gamma}\mathcal{H}(\psi_h^+,\psi_h^-,\boldsymbol{n})v_h^+\,\mathrm{d}s\,\mathrm{d}\theta\,\mathrm{d}\varphi\\
&\quad+\int_{\kappa_D}\int_{\partial\kappa_\Omega\cap\Gamma}\mathcal{H}(\psi_h^+,\psi_\Gamma^{\boldsymbol{\mu}}(\psi_h^+),\boldsymbol{n})v_h^+\,\mathrm{d}s\,\mathrm{d}\theta\mathrm{d}\varphi\Bigg]\\
&\quad=\sum_{\kappa_D\in\mathcal{T}_D}\sum_{\kappa_\Omega\in\mathcal{T}_\Omega}\left[\int_{\kappa_D}\int_{\kappa_\Omega}\left(\frac{\Sigma_s(\boldsymbol{x})+\nu(\boldsymbol{x})\Sigma_f(\boldsymbol{x})}{2\pi}\int_0^{2\pi}\int_0^{\pi/2}\psi_h(\boldsymbol{x},\theta',\varphi')\frac{1}{\cos(\theta)}\,\mathrm{d}\theta'\,\mathrm{d}\varphi'\right)\mathrm{d}\boldsymbol{x}\,\mathrm{d}\theta\,\mathrm{d}\varphi\right]\\
&\quad+\sum_{\kappa_D\in\mathcal{T}_D}\sum_{\kappa_\Omega\in\mathcal{T}_\Omega}\left[\int_{\kappa_D}\int_{\kappa_\Omega}Q(\boldsymbol{x},\theta,\varphi)v_h\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\theta\,\mathrm{d}\varphi\right]\quad\forall v_h\in V^{h_\Omega,h_I}.
\end{aligned}\tag{3.26}$$

In the next chapter we discuss some implementation issues that are pertinent to this pseudo 3D discretisation.

3.4 *A priori* Results for the Neutron Transport Equation

We end this chapter by briefly summarising the *a priori* results for finite element discretisations of the neutron transport equation that are in the literature. One of the relevant articles in this field is the paper by Johnson and Pitkäranta [92] in which they consider a discretisation of the two dimensional mono energetic steady state neutron transport equation. The authors derive error bounds on the scalar flux, employing results contained in [93]. Here, to discretise the problem they employ a discrete ordinates discretisation in angle with a discontinuous Galerkin method in space. For convenience, we state the two dimensional problem in the notation that is used

$$\begin{aligned}
&\boldsymbol{\mu}\cdot u(x,\boldsymbol{\mu}))+u(x,\boldsymbol{\mu})=\lambda\int_S u(x,\boldsymbol{\mu}')\,\mathrm{d}\boldsymbol{\mu}'+f(x)\quad(x,\mu)\in\Omega\times S,\\
&u(x,\mu)=0\quad\text{on }\Gamma_\mu^-,
\end{aligned}\tag{3.27}$$

where Ω is a bounded convex polyhedral domain in $\mathbb{R}^2$, S is the boundary of the unit disc $S=\{\mu\in\mathbb{R}^2:|\mu|=1\}$ and Γ_μ^- is the inflow boundary of Ω with respect to the direction μ defined in the usual way.

Defining the scalar flux $U(x)=\int_S u(x,\mu)\,\mathrm{d}\mu$ then with the use of a solution operator T_μ for the advection problem, by integrating over S they reformulate the neutron transport problem as an integral equation in U

$$(I-\lambda T)U=Tf.\tag{3.28}$$

In the above $T : L_2(\Omega) \to H^1(\Omega)$ is a Fredholm equation of the second form. With a discrete ordinates quadrature (satisfying the conditions discussed previously) and a DG spatial discretisation the error in the scalar flux is given by the following decomposition

$$\begin{aligned}(I - \lambda T_N^h)(U - U_N^h) &= (T - T_N)(\lambda U + f) + (T_N - T_N^h)(\lambda U + f) \\ &= e_N + e_N^h,\end{aligned}$$

where U_N^h is the full discrete approximation to the scalar flux U, T_N is the semi discrete equivalent of the operator T (where an angular discretisation only has been performed) and T_N^h is the fully discretised operator. The final error bound given in the paper is the following

$$\|U - U_N^h\| \leq C(N^{-1} + h^{1/2})(\|U\|_1 + \|f\|_1), \tag{3.29}$$

where $\|\cdot\|$ denotes the L_2 norm and $\|\cdot\|_1$ is the norm in the Sobolev space $H^1(\Omega)$. This bound indicates that the error decays with order $1/2$ with respect to the spatial discretisation, which is suboptimal. Using realistic regularity assumptions this can be improved to $O(1-\varepsilon)$ in space and $O(3/2+\varepsilon)$ in angle, for any $\varepsilon > 0$.

Defining a quadrature scheme for the angular variable in this setting is more involved, with the ordinates defined to be

$$\mathcal{O} = \{\mu_{kj} \in D : \mu_{kj} = r_k(\cos\varphi_j, \sin\varphi_j), \quad k = 1, \ldots, N, \quad j = 1, \ldots, M\}$$

where $\varphi_j = \frac{2\pi j}{M}$ and r_k are the zeros of the Chebyshev polynomials which are orthogonal polynomials with respect to the distribution $\mathrm{d}\alpha(r) = (1-r^2)^{-1/2}$ on the interval $[0,1]$ (see [148], the classical reference for orthogonal polynomials for further details). This leads to a quadrature rule with MN points, there is also an assumption that $M \sim N$.

With this discretisation of the angular domain and a discontinuous Galerkin method (with linear approximation) discretising the spatial domain, performing an analysis analogous to that of [92], making suitable modifications to the definition of the operator T appearing in (3.28) to include the weighting $(1 - |\mu'|^2)^{-1/2})$, Asadzadeh obtains the estimate for $U, f \in H^1(\Omega)$.

$$\|U - U_N^h\| \leq C\left(\frac{1}{M} + \frac{1}{N} + h^{1/2}\right)(\lambda\|U\|_1 + \|f\|_1) \tag{3.30}$$

Again this bound is sub optimal in h.

In 1998 M. Asadzadeh returned to the neutron transport equation [16] with improvements to the bounds given in [15]. The analysis of Johnson and Pitkäranta is again broadly followed, but with the addition of the application of some Sobolev and Besov (interpolation spaces between two Sobolev spaces) space embedding relations. As in the previous paper, a discrete ordinates discretisation is used for the angular dimension and a DG discretisation with linear polynomials used for the spatial discretisation. For the analysis to hold the following compatibility conditions must be satisfied between h the maximum diameter of an element $\kappa \in \mathcal{T}$, where $\mathcal{T}$ is a triangulation of the spatial domain and n:

$$h^{-1}(n) \sim \sqrt{n} = \sqrt{MN} \quad \text{and} \quad M \sim N. \tag{3.31}$$

If this condition is not satisfied then the contributions of 'bad directions' to the spatial error will not be of the desired order of $\sim h$. The discrete ordinates are split into many 'good directions' and a few 'bad directions' so that each split part contributes the same order of convergence. The 'bad directions' are those such that μ is small, or those directions which are closely aligned to the directions of the sides of the spatial domain Ω.

Assuming that the constant $\lambda \notin \sigma(T)$, where $\sigma(T)$ is the spectrum of the linear operator T as used in [15], U being the scalar flux, and U_n^h being the full discrete approximation to U the following error bound is proved

$$\|U - U_n^h\| \leq C|\log h|h^{1-\varepsilon'}\|g\|_{H^{3/2-\varepsilon}(\Omega)}$$

where C is a constant, $g \in H^{3/2-\varepsilon}(\Omega)$ and $\varepsilon', \varepsilon$ are small satisfying $0 < \varepsilon < \varepsilon'$. The above error estimate only holds supposing that h is sufficiently small. This is an optimal estimate with respect to h, [93].

CHAPTER 4

Implementation Aspects

Here we discuss some issues arising in the implementation of the neutron transport solvers discussed in the previous chapter. Some aspects are general and applicable to both the two dimensional solver and the pseudo 3D solver; we treat these general issues first.

4.1 Implementation Issues for Both Solvers

Firstly we will develop an understanding of the matrix structure of the linear at techniques used to solve the linear systems resulting from the full DG discretisation considered in the last chapter before examining solution techniques. We consider the use of an outer source iteration to incorporate the coupling across angular elements due to the integral present on the right hand side of (1.6) before considering a novel solution strategy to solve the linear systems on each angular element. Finally we look at a method to solve the critical eigenvalue problem.

4.1.1 Developing the Discretisation

To further examine the scheme, recall that both the numerical solution ψ_h and the test function v_h belong to the space V^{h_Ω,h_I}. As V^{h_Ω,h_I} is constructed as the tensor product of the two spaces $S^{\boldsymbol{p}}(\Omega,\mathcal{T}_\Omega,\boldsymbol{F}_\Omega)$ and $S^{\boldsymbol{q}}(I,\mathcal{T}_I,\boldsymbol{F}_I)$, the functions which span V^{h_Ω,h_I} are comprised of products of the basis functions which span these separate spaces. To this end, let

$$S^{\boldsymbol{p}}(\Omega,\mathcal{T}_\Omega,\boldsymbol{F}_\Omega) = \operatorname{span}\{\zeta_i(\boldsymbol{x})\}, \quad i = 1,\ldots,n,$$
$$S^{\boldsymbol{q}}(I,\mathcal{T}_I,\boldsymbol{F}_I) = \operatorname{span}\{\xi_k(\varphi)\}, \quad k = 1,\ldots,m.$$

Then the full *DG* space can be expressed in the following fashion,

$$V^{h_\Omega,h_\theta} = \text{span}\{\zeta_i(\boldsymbol{x})\xi_k(\varphi)\} \qquad i = 1,\dots,n,\ k = 1,\dots,m. \tag{4.1}$$

Therefore, any function $w_h \in V^{h_\Omega,h_I}$ can be expressed as a linear combination of basis functions, i.e.,

$$w_h = \sum_{j=1}^{n}\sum_{l=1}^{m} W_{j,l}\zeta_j(\boldsymbol{x})\xi_l(\varphi),$$

where $W_{j,l}$ is the corresponding degree of freedom.

The angular basis functions $\xi_k(\varphi)$ are defined so that each of them only have support on a particular angular element. As each angular element κ_I is a real interval, the $\xi_k(\varphi)$ appearing in (4.1) can be chosen to be mapped orthogonal Legendre polynomials (as defined on the reference element $I = [-1,1]$) to κ_I. The first few Legendre polynomials on the interval $[-1,1]$ are as follows:

$$\begin{aligned} l_0(y) &= 1, \\ l_1(y) &= y, \\ l_2(y) &= \frac{3}{2}y^2 - \frac{1}{2}, \\ l_3(y) &= \frac{5}{2}y^3 - \frac{3}{2}y. \end{aligned}$$

These polynomials are orthogonal on the reference interval $[-1,1]$ and since the mapping to the angular element κ_I is linear, their mappings will be orthogonal on κ_I. For consistency of indexing with the spatial basis functions in the discretisation that follows, ξ_k will represent the $(k-1)$th Legendre polynomial after it has been mapped to the present angular element κ_I.
On each angular element we let n denote the number of spatial degrees of freedom (that is the number of spatial basis functions). We also let

$$m = \sum_{\kappa_I} m_{\kappa_I},$$

where $m_{\kappa_I} = q_{\kappa_I} + 1$ denotes the local number of angular degrees of freedom on the angular element κ_I. Using these definitions, we make the observation that on a particular angular element κ_I, we can express the numerical approximation ψ_h to the angular flux ψ on κ_I in terms of basis functions in the following way

$$\psi_h|_{\kappa_I} = \sum_{j=1}^{n}\sum_{l=1}^{m_{\kappa_I}} \Psi_{j,l}^{\kappa_I}\zeta_j(\boldsymbol{x})\xi_l(\varphi).$$

Note that the superscript κ_I may later be omitted from $\Psi_{j,l}^{\kappa_I}$, when considering an expansion of the solution restricted to a particular angular element. For example, If we choose

a piecewise linear approximation in angle on an element κ_I, then any $\psi_h \in V^{h_\Omega, h_I}$ can be expressed as

$$\psi_h|\kappa_I = \sum_{j=1}^{n} \Psi_{j,1}\zeta_j(\boldsymbol{x})\xi_1(\varphi) + \sum_{j=1}^{n} \Psi_{j,2}\zeta_j(\boldsymbol{x})\xi_2(\varphi).$$

For later use, given two angular basis functions of degree k and l, respectively, we define the scaled direction $\bar{\boldsymbol{\mu}}_{k,l}$ by,

$$\bar{\boldsymbol{\mu}}_{k,l} = \int_{\kappa_I} \xi_k(\varphi)\xi_l(\varphi)\boldsymbol{\mu}\,\mathrm{d}\varphi. \tag{4.2}$$

The discretisation (3.23) results in a linear system to be solved for the coefficients $\{\Psi_{j,l}^{\kappa_I}\}, j = 1, \ldots, n,\ l = 1, \ldots, m_{\kappa_I}$ of the angular flux on each angular element κ_I. With this in mind we solve a linear system on each angular element, then using these solutions, compute the scalar flux which couples the angular elements together. The change in scalar flux between these outer iterations will be used as a measure of convergence of the source iteration.

We have the following system to investigate:

$$\begin{aligned}
&\sum_{\kappa_I\in\mathcal{T}_I} \underbrace{\sum_{\kappa_\Omega\in\mathcal{T}_\Omega}\left[\int_{\kappa_I}\int_{\kappa_\Omega} -\psi_h\boldsymbol{\mu}\cdot\nabla_{\boldsymbol{x}} v_h\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\varphi\right]}_{\text{Term 1}} + \sum_{\kappa_I\in\mathcal{T}_I} \underbrace{\sum_{\kappa_\Omega\in\mathcal{T}_\Omega}\left[\int_{\kappa_I}\int_{\kappa_\Omega} \sigma_t\psi_h v_h\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\varphi\right]}_{\text{Term 2}} \\
&\quad + \sum_{\kappa_I\in\mathcal{T}_I} \underbrace{\sum_{\kappa_\Omega\in\mathcal{T}_\Omega}\left[\int_{\kappa_I}\left(\int_{\partial\kappa_\Omega\setminus\Gamma} \mathcal{H}(\psi_h^+,\psi_h^-,\boldsymbol{n})v_h^+\,\mathrm{d}s\,\mathrm{d}\varphi + \int_{\partial\kappa_\Omega\cap\Gamma} \mathcal{H}(\psi_h^+,\psi_\Gamma^{\boldsymbol{\mu}}(\psi_h^+),\boldsymbol{n})v_h^+\,\mathrm{d}s\,\mathrm{d}\varphi\right)\right]}_{\text{Term3}} \\
&\quad = \sum_{\kappa_I\in\mathcal{T}_I} \underbrace{\sum_{\kappa_\Omega\in\mathcal{T}\Omega}\left[\int_{\kappa_I}\int_{\kappa_\Omega} \frac{\Sigma_s+\nu\Sigma_f}{2\pi}\int_0^{2\pi}\psi_h(\boldsymbol{x},\varphi')v_h\,\mathrm{d}\varphi'\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\varphi\right]}_{\text{Term 4}} \\
&\quad + \sum_{\kappa_I\in\mathcal{T}_I} \underbrace{\sum_{\kappa_\Omega\in\mathcal{T}_\Omega}\left[\int_{\kappa_I}\int_{\kappa_\Omega} F(\boldsymbol{x},\theta)v_h\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\varphi\right]}_{\text{Term 5}}.
\end{aligned} \tag{4.3}$$

Remark 11. *The underbraces above do not include the sum over angular elements since when the scalar flux is known each angular element may be considered as an independent linear system.*

After considering the scalar flux, we will consider each term appearing above separately, making use of the expression of the angular flux as the weighted sum of the aforementioned spatial and angular basis functions.

By definition, the scalar flux for a problem with two spatial and one angular dimension is given by

$$\phi(\boldsymbol{x}) = \int_0^{2\pi} \psi(\boldsymbol{x}, \varphi') \, \mathrm{d}\varphi'.$$

We let $\phi_h(\boldsymbol{x})$ denoting our discretised scalar flux, it can be evaluated from the discretised angular flux $\psi_h(\boldsymbol{x}, \varphi)$ in the following manner

$$\begin{aligned}\phi_h(\boldsymbol{x}) &= \sum_{\kappa_I \in \mathcal{T}_I} \int_{\kappa_I} \sum_{j=1}^{n} \sum_{l=1}^{m_{\kappa_I}} \Psi_{j,l}^{\kappa_I} \zeta_j(\boldsymbol{x}) \xi_l(\varphi') \, \mathrm{d}\varphi' \\ &= \sum_{\kappa_I \in \mathcal{T}_I} \sum_{j=1}^{n} \sum_{l=1}^{m_{\kappa_I}} \Psi_{j,l}^{\kappa_I} \zeta_j(\boldsymbol{x}) \int_{\kappa_I} \xi_l(\varphi') \, \mathrm{d}\varphi'.\end{aligned}$$

We make the following observation

$$\int_{\kappa_I} \xi_l(\varphi) \, \mathrm{d}\varphi = \int_{-1}^{1} l_l(y) \times \mathrm{Jac}(F_{\kappa_I}^{-1}(\kappa_I)) \, \mathrm{d}y = \frac{|\kappa_I|}{2} \int_{-1}^{1} l_l(y) \, \mathrm{d}y,$$

and use the property of Legendre polynomials,

$$\int_{-1}^{1} l_i(y) \, \mathrm{d}y = \begin{cases} 2, & i = 0, \\ 0, & \text{otherwise}, \end{cases}$$

to write the integral of the angular basis function as

$$\int_{\kappa_I} \xi_l(\theta) \, \mathrm{d}\theta = \begin{cases} |\kappa_I|, & \text{if } l = 0, \\ 0, & \text{otherwise.} \end{cases}$$

Hence, the scalar flux can be written as

$$\phi_h(\boldsymbol{x}) = \sum_{j=1}^{n} \left(\sum_{\kappa_I \in \mathcal{T}_I} \Psi_{j,0}^{\kappa_I} |\kappa_I| \right) \zeta_j(\boldsymbol{x}),$$

where $\sum_{\kappa_I \in \mathcal{T}_I} \Psi_{j,0}^{\kappa_I} |\kappa_I|$ is the coefficient of the jth spatial basis function in the expansion of the discrete scalar flux.

Figure 4.1 illustrates the relative storage required for the solution vectors of the angular flux and scalar flux. Each black rectangle is a vector containing coefficients corresponding to all spatial degrees of freedom.

Remark 12. *We note that this means that information corresponding to the higher order angular modes does not contribute to the scalar flux.*

Term 1:

Here, we are interested in expanding the first term of the discretisation, namely,

$$\sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \left[\int_{\kappa_I} \int_{\kappa_\Omega} -\psi_h \boldsymbol{\mu} \cdot \nabla_{\boldsymbol{x}} v_h \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\varphi \right].$$

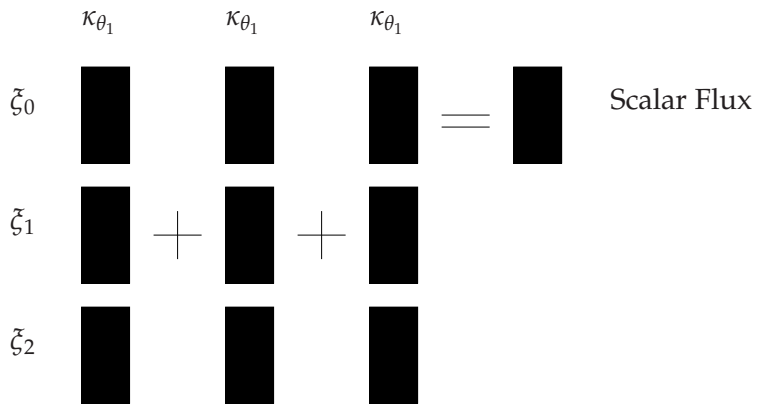


Figure 4.1: Combining angular flux on 3 angular elements to give the scalar flux.

As stated before, since we are working in the two dimensional case, $\boldsymbol{\mu} = (\cos(\varphi), \sin(\varphi))^\top$, and using the expansion of ψ_h in terms of the basis functions of the space V^{h_Ω, h_θ}, together with letting a test function be represented as $v_h = \zeta_i(\boldsymbol{x})\xi_k(\varphi)$, we obtain the following

$$\begin{aligned}
&\sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \int_{\kappa_I} \int_{\kappa_\Omega} -\psi_h \boldsymbol{\mu} \cdot \nabla v_h \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\varphi \\
&\qquad = \sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \left[\int_{\kappa_I} \int_{\kappa_\Omega} - \sum_{j=1}^{n} \sum_{l=1}^{m} \Psi_{j,l}^{\kappa_I} \zeta_j(\boldsymbol{x}) \xi_l(\varphi) \begin{pmatrix} \cos(\varphi) \\ \sin(\varphi) \end{pmatrix} \cdot \nabla_x (\zeta_i(\boldsymbol{x}) \xi_k(\varphi)) \mathrm{d}\boldsymbol{x} \, \mathrm{d}\varphi. \right.
\end{aligned}$$

Now since the $\nabla_{\boldsymbol{x}}$ operator acts spatially we can move the term $\xi_k(\varphi)$ outside of the gradient operator, and combine $\xi_l(\varphi)$, $\xi_k(\varphi)$ and $\boldsymbol{\mu}$ together to give

$$\begin{aligned}
&\sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \left[\int_{\kappa_I} \int_{\kappa_\Omega} -\psi_h \boldsymbol{\mu} \cdot \nabla v_h \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\varphi \right] \\
&\qquad = \sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \sum_{j=1}^{n} \sum_{l=1}^{m} \int_{\kappa_\Omega} \Psi_{j,l}^{\kappa_I} \zeta_j(x) u_j(x) \left(\int_{\kappa_I} \begin{pmatrix} \cos(\varphi) \\ \sin(\varphi) \end{pmatrix} \xi_l(\varphi) \xi_k(\varphi) \, \mathrm{d}\varphi \right) \cdot \nabla_x \zeta_i(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \\
&\qquad = \sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \sum_{j=1}^{n} \sum_{l=1}^{m} \int_{\kappa_\Omega} \Psi_{j,l}^{\kappa_I} \zeta_j(x) \left(\int_{\kappa_I} \begin{pmatrix} \cos(\varphi) \\ \sin(\varphi) \end{pmatrix} \xi_l(\varphi) \xi_k(\varphi) \, \mathrm{d}\varphi \right) \cdot \begin{pmatrix} \frac{\partial}{\partial x} \zeta_i(\boldsymbol{x}) \\ \frac{\partial}{\partial y} \zeta_i(\boldsymbol{x}) \end{pmatrix} \, \mathrm{d}\boldsymbol{x}
\end{aligned}$$

Now, we make use of the definition for $\bar{\boldsymbol{\mu}}_{k,l}$ given in (4.2), indeed we can write

$$\begin{aligned}
&\sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \sum_{j=1}^{n} \sum_{l=1}^{m} \int_{\kappa_\Omega} \Psi_{j,l}^{\kappa_I} \zeta_j(\boldsymbol{x}) \left(\int_{\kappa_I} \begin{pmatrix} \cos(\varphi) \\ \sin(\varphi) \end{pmatrix} \xi_l(\varphi) \xi_k(\varphi) \, \mathrm{d}\varphi \right) \cdot \begin{pmatrix} \frac{\partial}{\partial x} \zeta_i(\boldsymbol{x}) \\ \frac{\partial}{\partial y} \zeta_i(\boldsymbol{x}) \end{pmatrix} \, \mathrm{d}\boldsymbol{x} \\
&\qquad = \sum_{j=1}^{n} \sum_{l=1}^{m} \left[\sum_{\kappa_\Omega} \int_{\kappa_\Omega} \zeta_j(\boldsymbol{x}) \left(\bar{\boldsymbol{\mu}}_{k,l} \cdot \nabla_x \zeta_i(\boldsymbol{x}) \right) \, \mathrm{d}x \right] \Psi_{j,l}^{\kappa_I}, \;\; j = 1, \ldots, n, \; k = 1, \ldots m_{\kappa_I}.
\end{aligned}$$

Finally, for later we define

$$A_{i,j}^{k,l} = \sum_{\kappa_\Omega} \int_{\kappa_\Omega} \zeta_j(\boldsymbol{x}) \left(\bar{\boldsymbol{\mu}}_{k,l} \cdot \nabla_x \zeta_i(x) \right) \, \mathrm{d}\boldsymbol{x}. \tag{4.4}$$

Term 2:

In this section we expand and simplify the matrices arising from the discretisation of the reaction term

$$\sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \left[\int_{\kappa_I} \int_{\kappa_\Omega} \Sigma_t \psi_h v_h \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\varphi \right].$$

Note that Σ_t is a function of $\boldsymbol{x}$, i.e., $\Sigma_t = \Sigma_t(\boldsymbol{x})$. So using the same expansion of the angular flux and representation of a test function v_h, we have

$$\begin{aligned}
&\sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \left[\int_{\kappa_I} \int_{\kappa_\Omega} \Sigma_t \psi_h v_h \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\varphi \right] \\
&\quad = \sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \sum_{j=1}^{n} \sum_{l=1}^{m} \left[\int_{\kappa_I} \int_{\kappa_\Omega} \Sigma_t(\boldsymbol{x}) \Psi_{j,l}^{\kappa_I} \zeta_j(\boldsymbol{x}) \xi_l(\varphi) \zeta_i(\boldsymbol{x}) \xi_k(\varphi) \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\varphi \right] \\
&\quad = \sum_{j=1}^{n} \sum_{l=1}^{m} \left[\sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \int_{\kappa_\Omega} \Sigma_t(\boldsymbol{x}) \Psi_{j,l}^{\kappa_I} \zeta_j(\boldsymbol{x}) \zeta_i(\boldsymbol{x}) \left(\int_{\kappa_I} \xi_l(\varphi) \xi_k(\varphi) \, \mathrm{d}\varphi \right) \mathrm{d}\boldsymbol{x} \right].
\end{aligned}$$

By the orthogonality of the scaled Legendre polynomials on the angular element κ_I, we have

$$\int_{\kappa_I} \xi_l(\varphi) \xi_k(\varphi) \, \mathrm{d}\theta = \begin{cases} \dfrac{2|\kappa_I|}{2l+1} & \text{if } l = k, \\ 0 & \text{otherwise.} \end{cases} \tag{4.5}$$

Thus, on each angular element, term 2 only appears when the angular degrees of freedom k and l are equal, i.e., in the large diagonal blocks of the matrix corresponding to that angular element.

Thereby, we arrive at

$$\begin{aligned}
&\sum_{j=1}^{n} \sum_{l=1}^{m} \left[\sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \int_{\kappa_\Omega} \Sigma_t(\boldsymbol{x}) \Psi_{j,l}^{\kappa_I} \zeta_j(\boldsymbol{x}) \zeta_i(\boldsymbol{x}) \left(\int_{\kappa_I} \xi_l(\varphi) \xi_k(\varphi) \, \mathrm{d}\varphi \right) \mathrm{d}\boldsymbol{x} \right] \\
&\qquad = \sum_{j=1}^{n} \sum_{l=1}^{m} \left[\sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \frac{2|\kappa_I|}{2l+1} \int_{\kappa_\Omega} \Sigma_t(\boldsymbol{x}) \zeta_j(\boldsymbol{x}) \zeta_i(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \right] \Psi_{j,l}^{\kappa_I}, \quad j = 1, \ldots n.
\end{aligned}$$

We define

$$B_{i,j}^{k,k} := \sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \frac{2|\kappa_I|}{2l+1} \int_{\kappa_\Omega} \Sigma_t(\boldsymbol{x}) \zeta_j(\boldsymbol{x}) \zeta_i(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}.$$

Term 3:

This is the term that deals with the discontinuities in the solution across elemental boundaries. For simplicity, in calculations we decompose the integral around the boundary of a spatial element into the sum of the integrals along each face of the element. For

clarity we only show this for the first part of Term 3; the part that corresponds to interior element boundaries. Hence, we obtain

$$\sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \int_{\kappa_I} \int_{\partial\kappa_\Omega} \mathcal{H}(\psi_h^+, \psi_h^-, \boldsymbol{n}_{\kappa_\Omega})\, v_h \,\mathrm{d}s = \sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \int_{\kappa_I} \sum_{e \in \mathcal{E}_{\kappa_\Omega}} \int_e \mathcal{H}(\psi_h^+, \psi_h^-, \boldsymbol{n}_{\kappa_\Omega})\, v_h \,\mathrm{d}s$$
$$= \sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \sum_{e \in \mathcal{E}_{\kappa_\Omega}} \int_{\kappa_I} \int_e \mathcal{H}(\psi_h^+, \psi_h^-, \boldsymbol{n}_{\kappa_\Omega})\, v_h \,\mathrm{d}s,$$

where $\mathcal{E}_{\kappa_\Omega}$ is the set of faces of the element κ_Ω.
Since $\boldsymbol{\mu}(\varphi) = (\cos(\varphi), \sin(\varphi))^\top$, $\boldsymbol{\mu} \cdot \boldsymbol{n}_{\kappa_\Omega}$ (recall that the numerical flux depends on this quantity) will have a dependence on φ (as will ψ_h and v_h). This dependence implies that on any given face of a spatial element, for a particular angular element, $\boldsymbol{\mu} \cdot \boldsymbol{n}_{\kappa_\Omega}$ may not be exclusively greater than or less than zero. As a consequence, for a combination of a general spatial mesh and a general angular mesh we cannot make one categorisation of spatial faces into inflow or outflow edges that holds for the integration over angle.
Instead, we approximate the volume integral over the angular element κ_I by a quadrature rule with nq_{κ_I} quadrature points φ_r and associated weights ω_r:

$$\sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \sum_{e \in \mathcal{E}_{\kappa_\Omega}} \int_{\kappa_I} \int_e \mathcal{H}(\psi_h^+, \psi_h^-, \boldsymbol{n}_{\kappa_\Omega})\, v_h \,\mathrm{d}s$$
$$\approx \sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \sum_{e \in \mathcal{E}_{\kappa_\Omega}} \sum_{r=1}^{nq_{\kappa_I}} \omega_r \int_e \mathcal{H}(\psi_h^+, \psi_h^-, \boldsymbol{n}_{\kappa_\Omega}, \varphi_r)\, v_h(s, \varphi_r) \,\mathrm{d}s,$$

where $\mathcal{H}(\psi_h^+, \psi_h^-, \boldsymbol{n}_{\kappa_\Omega}, \varphi_r)$ denotes the numerical flux $\mathcal{H}(\psi_h^+, \psi_h^-, \boldsymbol{n}_{\kappa_\Omega})$ evaluated at the angular quadrature point φ_r. For clarity of presentation (and also in practice), we consider looping over all faces in the spatial mesh and consider at first the expression

$$\omega_r \int_e \mathcal{H}(\psi_h^+, \psi_h^-, \boldsymbol{n}_{\kappa_\Omega}, \varphi_r)\, v_h(s, \varphi_r) \,\mathrm{d}s. \tag{4.6}$$

This represents the evaluation of the integral over the spatial face e for one quadrature point φ_r in the angular quadrature. When evaluating (4.6), for each angular quadrature point φ_r, we evaluate $(\boldsymbol{\mu} \cdot \boldsymbol{n}_{\kappa_\Omega})$ and depending on whether it is positive or negative we perform different operations. We then sum over the quadrature points on each face before summing over the faces to perform the integral over angle and then summing over all spatial elements to define the matrix and right hand–side vector entries. We consider each term of the flux seperately.

Interior Element Faces For every quadrature point θ_r and its associated quadrature weight ω_r, we evaluate $(\boldsymbol{\mu}(\varphi_r) \cdot \boldsymbol{n}_{\kappa_\Omega}(s))$, where s is a point on the face. There are two possibilities: we shall consider first the case where

$(\boldsymbol{\mu}(\varphi_r) \cdot \boldsymbol{n}_{\kappa_\Omega}(s)) < 0.$

When $(\boldsymbol{\mu}(\varphi_r) \cdot \boldsymbol{n}_{\kappa_\Omega}(s)) < 0$, we have

$$\mathcal{H}(\psi_h^+, \psi_h^-, \boldsymbol{n}_{\kappa_\Omega}, \varphi_r) = -|\boldsymbol{\mu} \cdot \boldsymbol{n}_{\kappa_\Omega}|\psi_h^-.$$

Since $\psi_h^- = \sum_{j=1}^{n} \sum_{l=1}^{m_{\kappa_I}} \Psi_{j,l}^- \zeta_j(s)\xi_l(\varphi)$ and $v_h = \zeta_i(x)\xi_k(\varphi)$ our expression for a face integral becomes

$$\begin{aligned}
\omega_r \int_e \mathcal{H}(\psi_h^+, \psi_h^-, \boldsymbol{n}_{\kappa_\Omega}, \varphi_r)\, v_h(s, \varphi_r)\, \mathrm{d}s & \\
&= \omega_r \int_e -|\boldsymbol{\mu}(\varphi_r) \cdot \boldsymbol{n}_{\kappa_\Omega}(s)|\psi^- v_h(s, \varphi_r)\, \mathrm{d}s \\
&= \omega_r \int_e -|\boldsymbol{\mu}(\varphi_r) \cdot \boldsymbol{n}_{\kappa_\Omega}(s)| \sum_{j=1}^{n} \sum_{l=1}^{m} \Psi_{j,l}^- \zeta_j(s)\xi_l(\varphi_r)\zeta_i(s)\xi_k(\varphi_r)\, \mathrm{d}s \\
&= \sum_{j=1}^{n} \sum_{l=1}^{m} \omega_r \xi_l(\varphi_r)\xi_k(\varphi_r) \int_e -|\boldsymbol{\mu}(\varphi_r) \cdot \boldsymbol{n}_{\kappa_\Omega}(s)|\Psi_{j,l}^- \zeta_j(s)\zeta_i(s)\, \mathrm{d}s,
\end{aligned}$$

for $i = 1, \ldots, n,\ k = 1, \ldots, m_{\kappa_I}$. Now, with t indexing the set of all quadrature points φ_r such that $(\boldsymbol{\mu}(\theta_r) \cdot \boldsymbol{n}_{\kappa_\Omega}(x)) < 0$, we can sum over the edges, over quadrature points and then over spatial elements to define the matrix entries

$$C_{i,j}^{k,l} := \sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \sum_{t} \omega_t \xi_l(\varphi_t)\xi_k(\varphi_t) \sum_{e \in \mathcal{E}_{\kappa_\Omega}} \int_e -|\boldsymbol{\mu}(\varphi_r) \cdot \boldsymbol{n}_{\kappa_\Omega}(s)|\zeta_j(s)\zeta_i(s)\, \mathrm{d}s.$$

Here, the angular degrees of freedom k and l index matrix blocks for the particular angular element κ_I, κ_Ω runs over the nested blocks and the spatial indices i and j run over entries in nested blocks. The structure of the matrix will be discussed in more detail in Section 4.1.1.

Similarly, if $\boldsymbol{\mu}(\varphi_r) \cdot \boldsymbol{n}_{\kappa_\Omega}(s)) > 0$, we have

$$\mathcal{H}(\psi_h^+, \psi_h^-, \boldsymbol{n}_{\kappa_\Omega}, \varphi_r) = |\boldsymbol{\mu} \cdot \boldsymbol{n}_{\kappa_\Omega}|\psi_h^+,$$

and consequently, following the same steps as above we have

$$\begin{aligned}
\omega_r \int_e \mathcal{H}(\psi_h(s, \varphi_r), \boldsymbol{n}_{\kappa_\Omega}, \varphi_r)\, v_h(s, \varphi_r)\, \mathrm{d}s & \\
&= \sum_{j=1}^{n} \sum_{l=1}^{m} \omega_r \xi_l(\varphi_r)\xi_k(\varphi_r) \int_e |\boldsymbol{\mu}(\varphi_r) \cdot \boldsymbol{n}_{\kappa_\Omega}(s)|\Psi_{j,l}^+ \zeta_j(s)\zeta_i(s)\, \mathrm{d}s\ .
\end{aligned}$$

So, with an abuse of notation and with u indexing the set of all quadrature points φ_r such that $(\boldsymbol{\mu}(\varphi_r) \cdot \boldsymbol{n}_{\kappa_\Omega}(x)) > 0$, we have the following expression for the matrix entries

$$C_{i,j}^{k,l} := \sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \sum_{u} \omega_u \xi_l(\varphi_u)\xi_k(\varphi_u) \sum_{e \in \mathcal{E}_{\kappa_\Omega}} \int_e -|\boldsymbol{\mu}(\varphi_r) \cdot \boldsymbol{n}_{\kappa_\Omega}(s)|\zeta_j(s)\zeta_i(s)\, \mathrm{d}s.$$

Outflow Domain Boundaries Coincident with an Elemental Face

In this case we know that the face e is a face on the spatial domain boundary and that $(\boldsymbol{\mu}(\varphi_r) \cdot \boldsymbol{n}_{\kappa_\Omega}(s)) > 0$. Since on an outflow boundary we do not have any information regarding the angular flux, we make use of the following expansion of the interior trace of the numerical solution

$$\psi_h^+ = \sum_{j=1}^{n} \sum_{l=1}^{m_{\kappa_I}} \Psi_{j,l}^+ \zeta_j(s) \xi_l(\varphi).$$

Using this expression and that $v_h = \zeta_i(s)\xi_k(\varphi)$, we write

$$\begin{aligned}\omega_r \int_e \mathcal{H}(\psi_h^+, \psi_h^-, \boldsymbol{n}_{\kappa_\Omega}, \varphi_r)\, v_h(s, \varphi_r)\, \mathrm{d}s &= \omega_r \int_e (\boldsymbol{\mu}(\varphi_r) \cdot \boldsymbol{n}_{\kappa_\Omega}(s)) \sum_{j=1}^{n} \sum_{l=1}^{m} \Psi_{j,l}^+ \zeta_j(s) \xi_l(\varphi_r) \zeta_i(s) \xi_k(\varphi_r) \mathrm{d}s \\ &= \sum_{j=1}^{n} \sum_{l=1}^{m} \omega_r \xi_l(\theta_r) \xi_k(\theta_r) \int_e (\boldsymbol{\mu}(\varphi_r) \cdot \boldsymbol{n}_{\kappa_\Omega}(x)) \Psi_{j,l}^+ \zeta_j(s) \zeta_i(s) \mathrm{d}s\,,\end{aligned}$$

for $i = 1, \ldots, n,\ k = 1, \ldots, m_{\kappa_I}$.

We define the matrix entry $D_{i,j}^{k,l}$, where u indexes the set of all quadrature points φ_r such that $(\boldsymbol{\mu}(\varphi_r) \cdot \boldsymbol{n}_{\kappa_\Omega}(s)) > 0$, by

$$D_{i,j}^{k,l} := \sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \sum_{u} \omega_t \xi_l(\varphi_u) \xi_k(\varphi_u) \sum_{e \in \mathcal{E}_{\kappa_\Omega}} \int_e (\boldsymbol{\mu}(\varphi_u) \cdot \boldsymbol{n}_{\kappa_\Omega}(s)) \Psi_{j,l}^+ \zeta_j(s) \zeta_i(s) \mathrm{d}s.$$

Inflow Domain Boundaries Coincident with an Elemental Face

In this case, at a quadrature point φ_r, we necessarily have that $(\boldsymbol{\mu}(\varphi_r) \cdot \boldsymbol{n}_{\kappa_\Omega}(s)) < 0$ and that the face e is therefore a face on the inflow spatial domain boundary $\Gamma_{\boldsymbol{\mu}}^-$.

From the formulation of the problem, we know that the flux entering the element at this point is given by $g(s, \theta_r)$, hence the present term appears in the right hand–side vector of the linear system. Thus, we have

$$\begin{aligned}\omega_r \int_e \mathcal{H}(\psi_h(s, \varphi_r), \boldsymbol{n}_{\kappa_\Omega}, \varphi_r)\, v_h(s, \varphi_r)\, \mathrm{d}s &= \omega_r \int_e (\boldsymbol{\mu}(\varphi_r) \cdot \boldsymbol{n}_{\kappa_\Omega}(s)) g(s, \varphi_r)\, v_h(s, \varphi_r)\, \mathrm{d}s \\ &= \omega_r \int_e (\boldsymbol{\mu}(\varphi_r) \cdot \boldsymbol{n}_{\kappa_\Omega}(s)) g(\boldsymbol{x}, \varphi_r)\, \zeta_i(s) \xi_k(\varphi_r)\, \mathrm{d}s,\end{aligned}$$

on replacing v_h with the usual expansion in terms of basis functions.

We extract such a term for each l for every angular degree of freedom k on each angular element when $(\boldsymbol{\mu}(\varphi_r) \cdot \boldsymbol{n}_{\kappa_\Omega}(x)) < 0$, thus the corresponding vector entry for a given angular element will be a sum over l. Hence, we define, with t again indexing the set of all quadrature points φ_r such that $(\boldsymbol{\mu}(\varphi_r) \cdot \boldsymbol{n}_{\kappa_\Omega}(x)) < 0$

$$E_i^k := \sum_{l=1}^{m} \sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \sum_{t} \omega_t \sum_{e \in \mathcal{E}_{\kappa_\Omega}} \int_e (\boldsymbol{\mu}(\varphi_t) \cdot \boldsymbol{n}_{\kappa_\Omega}(s)) g(\boldsymbol{x}, \varphi_t)\, \zeta_i(s) \xi_k(\varphi_t)\, \mathrm{d}s.$$

Term 4:

Since by definition $\int_0^{2\pi} \psi_h(\boldsymbol{x}, \varphi')$ is the discretised scalar flux, Term 4 can be re-written as

$$\sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \left[\int_{\kappa_I} \int_{\kappa_\Omega} \frac{\Sigma_s + \nu\Sigma_f}{2\pi} \int_0^{2\pi} \psi_h(\boldsymbol{x}, \varphi') v_h \, \mathrm{d}\varphi' \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\varphi \right] = \sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \left[\int_{\kappa_I} \int_{\kappa_\Omega} \frac{\Sigma_s(\boldsymbol{x}) + \nu(\boldsymbol{x})\Sigma_f(\boldsymbol{x})}{2\pi} \phi_h v_h \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\varphi \right].$$

Now, expanding ϕ_h in terms of spatial basis functions i.e.,

$$\phi_h = \sum_{j=1}^{n} \Phi_j \zeta_j(\boldsymbol{x}),$$

and setting $v_h = \zeta_i(\boldsymbol{x})\xi_k(\varphi)$ $i = 1, \ldots, n$, $k = 1, \ldots, m_{\kappa_I}$, we have,

$$\begin{aligned}
&\sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \left[\int_{\kappa_I} \int_{\kappa_\Omega} \frac{\Sigma_s(\boldsymbol{x}) + \nu(\boldsymbol{x})\Sigma_f(\boldsymbol{x})}{2\pi} \phi_h(\boldsymbol{x}) v_h(\boldsymbol{x}, \varphi) \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\varphi \right] \\
&\quad = \sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \int_{\kappa_I} \int_{\kappa_\Omega} \frac{\Sigma_s(\boldsymbol{x}) + \nu(\boldsymbol{x})\Sigma_f(\boldsymbol{x})}{2\pi} \sum_{j=1}^{n} \Phi_j \zeta_j(\boldsymbol{x}) \zeta_i(\boldsymbol{x}) \xi_k(\varphi) \mathrm{d}\boldsymbol{x} \, \mathrm{d}\varphi \\
&\quad = \sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \int_{\kappa_\Omega} \frac{\Sigma_s(\boldsymbol{x}) + \nu(\boldsymbol{x})\Sigma_f(\boldsymbol{x})}{2\pi} \left(\sum_{j=1}^{n} \Phi_j \zeta_j(\boldsymbol{x}) \right) \zeta_i(\boldsymbol{x}) \left(\int_{\kappa_I} \xi_k(\varphi) \, \mathrm{d}\varphi \right) \mathrm{d}\boldsymbol{x} \, .
\end{aligned}$$

Since $\int_{\kappa_I} \xi_k(\varphi) \mathrm{d}\varphi = 0$ unless $k = 0$, when it equals $|\kappa_I|$, we have that

$$\begin{aligned}
&\sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \left[\int_{\kappa_I} \int_{\kappa_\Omega} \frac{\Sigma_s(\boldsymbol{x}) + \nu(\boldsymbol{x})\Sigma_f(\boldsymbol{x})}{2\pi} \phi_h v_h \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\varphi \right] \\
&\quad = |\kappa_I| \sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \int_{\kappa_\Omega} \frac{\Sigma_s(\boldsymbol{x}) + \nu(\boldsymbol{x})\Sigma_f(\boldsymbol{x})}{2\pi} \left(\sum_{j=1}^{n} \Phi_j \zeta_j(\boldsymbol{x}) \right) \zeta_i(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \; i = 1, \ldots, n
\end{aligned}$$

Finally, we define

$$R_i^0 = |\kappa_I| \sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \int_{\kappa_\Omega} \frac{\Sigma_s(\boldsymbol{x}) + \nu(\boldsymbol{x})\Sigma_f(\boldsymbol{x})}{2\pi} \left(\sum_{j=1}^{n} \Phi_j \zeta_j(\boldsymbol{x}) \right) \zeta_i(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}.$$

Term 5:

Here, we consider the final term appearing in the discretisation, that is the term arising due to the presence of an external forcing function $Q(\boldsymbol{x}, \varphi)$, which importantly is a function of both $\boldsymbol{x}$ and φ. This term is given by

$$\sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \left[\int_{\kappa_I} \int_{\kappa_\Omega} Q(\boldsymbol{x}, \varphi) v_h \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\varphi \right].$$

We need to test for all possible test functions v_h, and in doing so (on substituting our usual expression for a test function) we obtain every entry of the right hand side vector F, indexed by angular degree of freedom k, and spatial degree of freedom i, i.e.,

$$Q_i^k = \sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \int_{\kappa_I} \int_{\kappa_\Omega} F(\boldsymbol{x}, \varphi) \zeta_i(x) \xi_k(\varphi) \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\varphi.$$

Unfortunately this has to be evaluated as a true double integral since the integrand includes the function $Q(\boldsymbol{x}, \varphi)$ so we cannot take advantage of any orthogonality properties of the underlying basis functions.

Assembling the Linear System

First, we introduce the following definitions:

$$
\begin{aligned}
Q^k &= (Q_1^l, Q_2^l, \ldots, Q_n^l)^\top, \\
R^0 &= (U_k^l, U_2^k, \ldots, U_n^k)^\top, \\
E^k &= (E_k^l, E_2^k, \ldots, E_n^k)^\top, \\
A^{k,l} &= \begin{pmatrix} A_{1,1}^{k,l} & A_{1,2}^{k,l} & \cdots & A_{1,n-1}^{k,l} & A_{1,n}^{k,l} \\ A_{2,1}^{k,l} & \ddots & \ddots & \ddots & A_{2,n}^{k,l} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ A_{n,1}^{k,l} & A_{n,2}^{k,l} & \cdots & A_{n,n-1}^{k,l} & A_{n,n}^{k,l} \end{pmatrix}, \\
B^{k,k} &= \begin{pmatrix} B_{1,1}^{k,k} & B_{1,2}^{k,k} & \cdots & B_{1,n-1}^{k,k} & B_{1,n}^{k,k} \\ B_{2,1}^{k,k} & \ddots & \ddots & \ddots & B_{2,n}^{k,k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ B_{n,1}^{k,k} & B_{n,2}^{k,k} & \cdots & B_{n,n-1}^{k,k} & B_{n,n}^{k,k} \end{pmatrix}, \\
C^{k,l} &= \begin{pmatrix} C_{1,1}^{k,l} & C_{1,2}^{k,l} & \cdots & C_{1,n-1}^{k,l} & C_{1,n}^{k,l} \\ C_{2,1}^{k,l} & \ddots & \ddots & \ddots & C_{2,n}^{k,l} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ C_{n,1}^{k,l} & C_{n,2}^{k,l} & \cdots & C_{n,n-1}^{k,l} & C_{n,n}^{k,l} \end{pmatrix}, \\
D^{k,l} &= \begin{pmatrix} D_{1,1}^{k,l} & D_{1,2}^{k,l} & \cdots & D_{1,n-1}^{k,l} & D_{1,n}^{k,l} \\ D_{2,1}^{k,l} & \ddots & \ddots & \ddots & D_{2,n}^{k,l} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ D_{n,1}^{k,l} & D_{n,2}^{k,l} & \cdots & D_{n,n-1}^{k,l} & D_{n,n}^{k,l} \end{pmatrix}.
\end{aligned}
$$

Using these definitions a block linear system is formed; this is best illustrated with a concrete example. Consider a quadratic approximation in angle; there is, therefore, (in the two dimensional problem) 3 angular degrees of freedom. We then have the

following linear system to solve:

$$A \begin{pmatrix} U^0 \\ U^1 \\ U^2 \end{pmatrix} = \begin{pmatrix} Q^0 + E^0 + R^0 \\ Q^1 + E^1 \\ Q^2 + E^2 \end{pmatrix},$$

where the matrix A is given by

$$A = \begin{pmatrix} D^0 & O^{0,1} & O^{0,2} \\ O^{1,0} & D^1 & O^{1,2} \\ O^{2,0} & O^{2,1} & D^2 \end{pmatrix},$$

with diagonal blocks given by

$$D^k = A^{k,k} + B^{k,k} + C^{k,k} + D^{k,k},$$

and the off diagonal blocks

$$O^{k,l} = A^{k,l} + C^{k,l} + D^{k,l}.$$

These matrices quickly become large, especially as we increase the approximation degree in angle; we shall discuss efficient methods to solve systems of this form in the next chapter.

Remark 13. *It is worth noting that a discontinuous Galerkin discretisation with piecewise constants in angle (that is, $q = 0$) can be shown to be equivalent to a discrete ordinates discretisation in angle.*

4.1.2 Outer Source Iteration

As briefly mentioned in the previous chapter we solve the coupled neutron transport equations with a method known as Source Iteration (SI). We shall now discuss this in more detail. Consider the continuous problem,

$$\boldsymbol{\mu} \cdot \nabla_x \psi(\boldsymbol{x}, \varphi) + \Sigma_t \psi(\boldsymbol{x}, \varphi) = \frac{1}{2\pi} \int_0^{2\pi} (\Sigma_s + \nu\Sigma_f)\psi(\boldsymbol{x}, \varphi') \, \mathrm{d}\varphi',$$

with no external sources and assuming vacuum boundary conditions. Defining the operators

$$T(\psi(\boldsymbol{x}, \varphi) = (\boldsymbol{\mu} \cdot \nabla_x + \Sigma_t)(\psi(\boldsymbol{x}, \varphi))$$
$$R(\psi(\boldsymbol{x}, \varphi) = \frac{1}{2\pi} \int_0^{2\pi} (\Sigma_s + \nu\Sigma_f)(\psi(\boldsymbol{x}, \varphi') \, \mathrm{d}\varphi'$$

the continuous problem can be expressed as

$$T\psi = S\psi.$$

Mathematically the SI scheme is defined to b

$$T\psi^l = S\psi^{l-1}, \quad l \geq 1, \tag{4.7}$$

where ψ^0 is some initial guess. A suitable choice is to pick $\psi^0 = 0$. In this case there is a physical interpretation of the lth estimate to the angular flux ψ^l; it is the the angular flux due to neutrons that have scattered at most $l-1$ times [2]. Assuming that $\Sigma_t > \Sigma_s$ everywhere in the domain this iteration is guaranteed to converge, although the speed of convergence will depend on the relative sizes of the two material cross sections. Adams and Larsen [2] provide a convergence analysis of the source iteration technique using Fourier modes and note that the rate of convergence for Source iteration is heavily dependent on the spatial discretisation method used but only weakly dependent on the angular discretisation used. Indeed, we observe no difference in the rate of convergence observed when using a discrete ordinates or DG angular discretisation.

We now consider the use of this iteration technique in practice. As the source iteration is independent of our discretisation procedure, for simplicity we recall the discrete ordinates in angle, DG in space (DO-DG) discretisation: for each $i = 1, \dots N$ find $\psi_h \in V_{h,p}$ such that

$$A_{NT}(\psi_h, v_h) = l_{NT}(v_h) \quad \forall\, v_h \in V_{h,p} \tag{4.8}$$

with the bilinear form $A_{NT} : V_{h,p} \times V_{h,p} \to \mathbb{R}$ given by

$$\sum_{\kappa \in \mathcal{T}} \left(- \int_\kappa (\boldsymbol{\mu}_i \psi_h^l) \cdot \nabla v_h \, \mathrm{d}\boldsymbol{x} + \int_{\partial\kappa} \mathcal{H}(\psi_h^{l,+}, \psi_h^{l,-}, \boldsymbol{n}_\kappa) v_h^+ \, \mathrm{ds} \right)$$

and linear functional $l_{NT} : V_h \to \mathbb{R}$,

$$l_{NT}(v_h) = \sum_{\kappa \in \mathcal{T}} \left(\int_\kappa Q v_h \, \mathrm{d}\boldsymbol{x} + \int_\kappa \frac{\Sigma_s + \Sigma_f}{2\pi} \left(\sum_{j=1}^{N} \omega_j \psi_h^{l-1}(\boldsymbol{x}, \varphi_j) \right) \, \mathrm{d}\boldsymbol{x} \right)$$

Let ψ_h^l denote the lth iterate of the source iteration. To start the SI procedure we have to choose an initial ψ_h^0 for each ordinate $\boldsymbol{\mu}_i$, the most natural would be $\psi_h^0 = 0$, since in general we have no *a priori* information about the angular flux. Then using ψ_h^0 in the right hand side of the discretisation we compute the solution ψ_h^1 for every ordinate $\boldsymbol{\mu}_i$ and then update the scalar flux, thus updating the right hand side of the discretisation. This process is repeated until successive approximations to the scalar flux are within some tolerance.

Despite using this iterative procedure, it is possible to consider the full linear system that would be obtained if you solve all angles at the same time. The full linear system

considered in this thesis comprises the stiffness matrix corresponding to each angular element together with terms corresponding to the quadrature rule with which $\phi(x)$ is calculated from $\psi(x, \varphi)$. This will be a blocked system with $(\text{number of angles})^2$ blocks. The diagonal blocks will contain the stiffness matrices corresponding to the application of the operator T on each angle; the off diagonal blocks will contain the coupling terms. When we consider this situation, we note that the SI scheme is a Jacobi iteration. We note that it is possible to consider other splittings of the full discretisation matrix and hence obtain other iterative schemes; however the Jacobi iteration is by far the easiest to implement and is an industry standard approach which for all practical problems converges in a reasonable number of iterations.

If the source iteration fails to converge (which has not been observed this for any of the problems considered in this thesis) then other iterative solution methods could be employed. One possible choice is the Generalised Minimal Residual (GMRES) method would be a suitable choice as this does not require the strict conditions for convergence that a Jacobi method requires. In the literature, multi grid [110, 118, 119] methods have been applied, however, not to our knowledge, for general discretisations on unstructured grids. Techniques to accelerate the convergence of the source iteration have been considered; coarse mesh rebalancing is also a common choice of acceleration technique for discrete ordinate solvers. For discontinuous Galerkin discretisations, some work has been performed on applying diffusion synthetic acceleration (DSA) [155]. Diffusion synthetic acceleration is essentially a form of preconditioner which is commonly used to accelerate convergence of the source iteration when materials are highly diffusive (i.e. they have a high scattering cross section). Here a diffusion solver is employed to accelerate the transport solver.

Remark 14. *We note that the size of the coefficient $(\sigma_s + \nu\sigma_f)/2\pi$, appearing in the coupling term of the neutron transport equation, significantly affects the number of outer iterations that are required. This due to the increase in size of the coupling terms in the full matrix of the system. As this factor is increased the diagonal dominance of the diagonal blocks that stem from each angular element. Some numerical experiments which show this will be presented in Chapter 5. Based on these experiments the increase in the number of outer iterations only becomes significant when $(\Sigma_s + \nu\Sigma_f)/2\pi$ becomes unphysically large, although we do see slower convergence in one particular eigenvalue benchmark we consider in Chapter 5.*

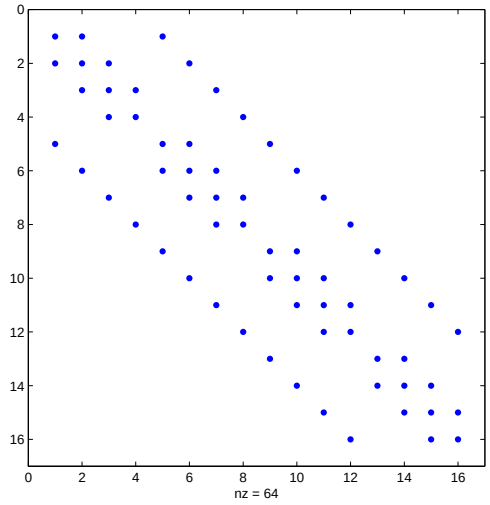


Figure 4.2: Matrix for one angular element corresponding to $p = 0$, $q = 0$ with 16 spatial elements.

4.1.3 The Solution on Each Angular Element

To implement our discretisation with an outer source iteration we need to solve a large linear system on each angular element.

In Section 4.1.1, we described the block linear structure of the linear systems on each angular element, taking advantage of the orthogonality of the basis functions we have used. It is, however, instructive to visualise the sparsity patterns of the matrices in the linear systems that we must solve on each angular element. For a discretisation where the degree of approximation in both angle and space is zero and we have 16 spatial elements the maximum sparsity pattern will be as shown in Figure 4.2 This is a much smaller matrix than would be solved in practice, as the system only contains 16 degrees of freedom with potentially 64 non–zero entries. Increasing the spatial degree of approximation p by 1, increases the number of degrees of freedom on each spatial element to 4 (when using quadrilateral elements in 2D with a $\mathcal{Q}_p$ basis) and so the total number of degrees of freedom grows to 64 (16 elements with 4 degrees of freedom each), however the overall structure of the matrix remains as shown in Figure 4.2. If, however, we wished to increase the degree of angular approximation to $q = 1$ as well, we end up with already a much larger matrix. For the two dimensional problem we obtain a matrix comprised of 4 large blocks, each of which has the same structure as shown in Figure 4.2. The sparsity pattern for a $p = q = 1$ approximation on 16 spatial elements is shown in Figure 4.3; here we have 128 degrees of freedom with potentially

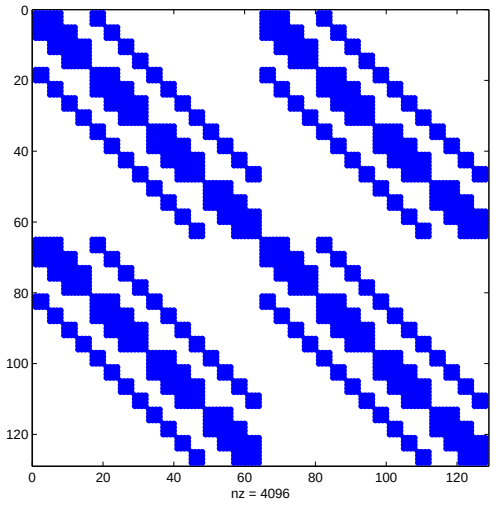


Figure 4.3: Matrix for one angular element corresponding to $p = 1$, $q = 1$ with 16 spatial elements.

4096 non zero entries. If we were considering the pseudo 3D discretisation, because of the increase in number of angular degrees of freedom for each element in moving from a one dimensional angular element to a two dimensional triangular angular element the full matrix is in fact larger comprising 9 large blocks arranged in a 3×3 grid.
These matrices could of course be solved with an available package such as MUMPS [4] or SPARSKIT, however when implementing the discretisations discussed in the previous chapter it became obvious that large memory use was a limiting factor in the accuracy that could be obtained when solving the neutron transport equation. For discretisations with very fine meshes in both space and angle, storing (even when using a sparse form of storage) and solving the linear system becomes prohibitive; for example in some of the finest discretisations used to obtain the results from the test cases considered in the next chapter over 90 gigabytes of RAM would be required when using MUMPS. When MUMPS can be used, it is very fast, and in fact can't be beaten by the ordered solver we ascribe here, however of large problems the use of an ordered solver becomes critical. In collaboration with S. Murphy a low memory solver for advection equations where the use of a re-odering strategy for the spatial elements eliminates the need to store the full linear system has been developed. Here we discuss the ordered solver that has been implemented for the neutron transport equation; further details will be presented S. Murphy's PhD thesis [114].

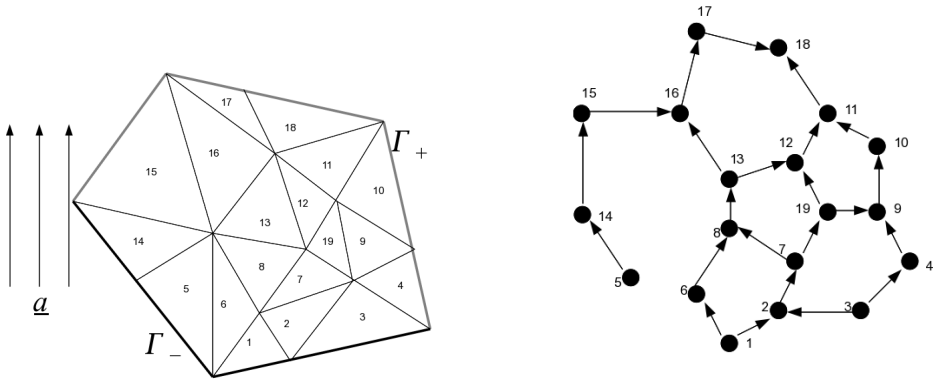


Figure 4.4: A general mesh over some domain with its associated directed graph.

Element Ordering for the Advection Equation

The idea of making use of an ordering of the spatial elements in the solution of a neutron transport discretisation appears to have first been suggested, in a fairly informal way, by LeSaint and Raviart in their paper of 1974 [100]. As mentioned above, for each iterate of the outer source Iteration, we need to solve an advection type equation. In this section, we summarise some results from [114] and discuss the ordering procedure for the linear solves required on each angular element.

Firstly, we remind ourselves of some basic concepts from graph theory [38]. A directed graph $G = (V, E)$, is composed of a set of vertices V, and a set of ordered pairs of vertices called edges, $(m, n) = e \in E$ where $m, n \in V$. A cycle is a set C

$$C = \{(n_1, n_2), (n_2, n_3), ..., (n_N, n_1)\} \subset E.$$

A graph is said to be acyclic if there does not exist such a subset.
We now consider how to find an ordering of the elements given an advective direction. The goal is to order the elements in such a way that the discretisation matrix becomes lower triangular and so is amenable to being solved by block back substitution without the need to store, in memory, the whole matrix. Any spatial finite element mesh $\mathcal{T}_\Omega$ can be represented as a directed graph G. Each element κ_Ω in the mesh $\mathcal{T}_\Omega$ is represented by a vertex V and then every elemental face (that is non characteristic) corresponds to an edge E in G. The direction in which the edge points (that is which vertex it points from and to) depends on the direction of the flow across that face. An example of an irregular mesh and its associated directed graph is shown in Figure 4.4.
In [114], S. Murphy proves the following result

Lemma 4.1.1. *Consider the advection problem*

$$\boldsymbol{\mu} \cdot \nabla u + bu = f$$

with constant $\boldsymbol{\mu}$ and inflow boundary condition $u = g$ on Γ_-. Then, there exists an element ordering such that the discontinuous Galerkin method will yield a block lower triangular stiffness matrix for any mesh $\mathcal{T}_\Omega$, if for all $\kappa \in \mathcal{T}_\Omega$, the inflow boundary $\partial_- \kappa$ is a connected set.

The proof can be applied for all meshes with convex elements. For the ordering to yield a lower triangular linear system, the graph representing the mesh must be acyclic. In the case of a fixed $\boldsymbol{\mu}$, this will be the case; however, we have previously seen that integrating over the angular element with respect to φ (φ and θ in the pseudo 3D case) introduces a slight difficulty.
Depending on the structure of the spatial and angular meshes it is possible, when integrating over the angular element, for $\boldsymbol{\mu} \cdot \boldsymbol{n}$, where $\boldsymbol{n}$ is a unit outward normal to a particular face, to take both positive and negative values on that face. This is due to the approximation to the angular integral we make by using a quadrature rule. At each angular quadrature point, θ_r, we compute $\boldsymbol{\mu}(\theta_r) \cdot \boldsymbol{n}(\boldsymbol{x})$ which may be greater than or less than zero for differing θ_r; in this case some of the faces can be characterised as both inflow and outflow faces. This introduces a cycle into the graph and so the resulting matrix will not be lower triangular, since there may also be blocks above the diagonal. Instead it can be described as a block lower triangular matrix where not all blocks are the same size (see the next sub section for an example of this). However, it is still possible to find an ordering, we consider two separate algorithms, below.

For the acyclic case, that is when all faces in the spatial mesh are uniquely inflow or outflow, we may use the Topological Sort algorithm. This works by gradually removing vertices which have no incoming edges from the graph, placing them in the ordering and then deleting the outgoing edges associated with the vertex. This process continues until all that remains is the empty graph. The Topological sort algorithm takes $O(|V| + |E|)$ operations since it considers each vertex and edge once. Below is the pseudo code for this algorithm:

Algorithm 4.1.2. (Topological Sort) *For a directed acyclic graph $G = (V, E)$, the following pseudo-code will output on ordered list L, which gives a, possibly not unique, topological ordering. Let n, m be vertices in the graph, then the algorithm is as follows:*

$L \leftarrow$ *list to contain ordered vertices*
$S \leftarrow$ *set of nodes with no incoming edges*
while $S \neq \phi$ ***do***
  $S \leftarrow S \setminus n$
  $L \leftarrow L + n$
  for $m \in V$ such that $(n, m) \in E$ ***do***

$E \leftarrow E \setminus (n, m)$
if $\nexists\, l \in V$ such that $(l, m) \in E$ ***then***
$S \leftarrow S + m$
end if
end for
end while
output(Proposed order: 'L')

If instead we have the situation where there are cycles in the graph then we can use Tarjan's algorithm to find an ordering of the strongly connected components of the graph. This is more complicated than the topological sort algorithm and requires more memory but it does work at a comparable speed. To describe the pseudo code of this algorithm we first describe a function `strongconnect`. This function takes as input a vertex V and performs a depth first search of the graph finding all successors from the node V and records all strongly connected components of that subgraph. In the following pseudo code the `index` variable is the depth first search node number counter, `S` is the node stack; this will start being empty and is then filled with nodes as the algorithm reaches them but aren't yet committed to a strongly connected component. Once a connected component is found the algorithm pops (removes from the stack) the root node and all nodes connected to it. Locating the root node is done in the function `strongconnect`. Each node is given a depth search index `v.index` which numbers the nodes consecutively in the order in which they are discovered. Each node is also assigned a value `V.lowlink` that equals the index of some node reachable from V. This will always be less than `V.index` or the same as `V.index` if no other node is reachable from V. Hence V is the root of a strongly connected component if `V.index` = `V.lowlink`.

Algorithm 4.1.3. (Function strongconnect(V)) *Given a node V the following pseudo code outputs a strongly connected component.*

v.index $\leftarrow$ *index*
v.lowlink $\leftarrow$ *index*
S.push(v)
for $\forall (v, w) \in E$ ***do***
if *W.index is undefined* ***then***
strongconnect(v)
v.lowlink $\leftarrow \min(v.lowlink, w.index)$
end if
if $w \in S$ ***then***

```
        v.lowlink ← min(v.lowlink, w.idex)
      end if
    end for
    if v.lowlink = v.index then
      Start a new strongly connected component
      repeat
        w ← S.pop()
        Add w to current strongly connected component
      until w = v
      Output the current strongly connected component.
    end if
```

Using the above function we can now describe the pseudo code for Tarjan's algorithm.

Algorithm 4.1.4. (Tarjan's Algorithm) *Given a graph $G(V, E)$ this output a set of strongly connected components.*

```
Input( Graph G = (V, E)
index ← 0
S ← empty stack
for all v ∈ V do
  if v.index is undefined then
    call strongconnect(v)
  end if
end for
output(A list of the strongly connected components of G)
```

Direct method

In this section, we aim to tackle the blocked matrix directly, re–order it so that it can be blocked into a block lower triangular matrix before solving it by block backward substitution. For an illustration of the method, consider the stiffness matrix corresponding to a single angular element when a linear approximation is taken in angle $q = 1$, for the case with just four spatial elements. If the matrix is blocked, first according to the angular degrees of freedom, then spatial degrees of freedom (ordered by spatial element) the matrix will look like one of the following two cases.

Case 1: First, consider the case where there are no spatial element boundaries for which $\boldsymbol{\mu} \cdot \boldsymbol{n}_{\kappa_\Omega}$ takes both positive and negative values on the present angular element. In this

case, each of the large matrix blocks indexed by the angular degrees of freedom can be ordered to be lower triangular; this results in a matrix of the form M, shown below, to solve on every angular element.

$$M = \left(\begin{array}{cccc|cccc} A_{1,1} & & & & B_{1,1} & & & \\ A_{2,1} & A_{2,2} & & & B_{2,1} & B_{2,2} & & \\ A_{3,1} & A_{3,2} & A_{3,3} & & B_{3,1} & B_{3,2} & B_{3,3} & \\ A_{4,1} & A_{4,2} & A_{4,3} & A_{4,4} & B_{4,1} & B_{4,2} & B_{4,3} & B_{4,4} \\ \hline C_{1,1} & & & & D_{1,1} & & & \\ C_{2,1} & C_{2,2} & & & D_{2,1} & D_{2,2} & & \\ C_{3,1} & C_{3,2} & C_{3,3} & & D_{3,1} & D_{3,2} & D_{3,3} & \\ C_{4,1} & C_{4,2} & C_{4,3} & C_{4,4} & D_{4,1} & D_{4,2} & D_{4,3} & D_{4,4} \end{array}\right). \tag{4.9}$$

The linear system corresponding to this matrix has the form

$$M \left(\begin{array}{c} U_{AC1} \\ U_{AC2} \\ U_{AC3} \\ U_{AC4} \\ \hline U_{BD1} \\ U_{BD2} \\ U_{BD3} \\ U_{BD4} \end{array}\right) = \left(\begin{array}{c} rhs_{AC1} \\ rhs_{AC2} \\ rhs_{AC3} \\ rhs_{AC4} \\ \hline rhs_{BD1} \\ rhs_{BD2} \\ rhs_{BD3} \\ rhs_{BD4} \end{array}\right). \tag{4.10}$$

We can then in turn, re-order the block structure according to first the spatial elements, then the angular degrees of freedom, then spatial degrees of freedom to get the following block triangular system

$$M_{\text{reordered}} \left(\begin{array}{c} U_{AC1} \\ U_{BD1} \\ U_{AC2} \\ U_{BD2} \\ \hline U_{AC3} \\ U_{BD3} \\ U_{AC4} \\ U_{BD4} \end{array}\right) = \left(\begin{array}{c} rhs_{AC1} \\ rhs_{BD1} \\ rhs_{AC2} \\ rhs_{BD2} \\ \hline rhs_{AC3} \\ rhs_{BD3} \\ rhs_{AC4} \\ rhs_{BD4} \end{array}\right), \tag{4.11}$$

where the matrix M given below is block lower triangular (where the new blocks are highlighted by the light grey lines);

$$M_{\text{reordered}} = \left(\begin{array}{cc|cc|cc|cc} A_{1,1} & B_{1,1} & & & & & & \\ C_{1,1} & D_{1,1} & & & & & & \\ \hline A_{2,1} & B_{2,1} & A_{2,2} & B_{2,2} & & & & \\ C_{2,1} & D_{2,1} & C_{2,2} & D_{2,2} & & & & \\ \hline A_{3,1} & B_{3,1} & A_{3,2} & B_{3,2} & A_{3,3} & B_{3,3} & & \\ C_{3,1} & D_{3,1} & C_{3,2} & D_{3,2} & C_{3,3} & D_{3,3} & & \\ \hline A_{4,1} & B_{4,1} & A_{4,2} & B_{4,2} & A_{4,3} & B_{4,3} & A_{4,4} & B_{4,4} \\ C_{4,1} & D_{4,1} & C_{4,2} & D_{4,2} & C_{4,3} & D_{4,3} & C_{4,4} & D_{4,4} \end{array}\right). \quad (4.12)$$

When the system is written in this form it is clear that we may progress through the linear system first solving for U_{AC1} and U_{BD1} and subtracting the product of the face matrices $A_{2,1}$, $B_{2,1}$, $C_{2,1}$ and $D_{2,1}$ with these solution values from the right hand side. This will allow for the solution for the values of U_{AC2} and U_{BD2} and a back substitution for the next set of face matrices. This procedure may then be repeated until the full solution is determined.

Case 2:

Second, consider the case where there is a spatial element boundary for which $\boldsymbol{\mu} \cdot \boldsymbol{n}_{\kappa_\Omega}$ takes both positive and negative values on the present angular element. In the following example, this happens between the second and third spatial elements.

$$M = \left(\begin{array}{ccccc|cccc} A_{1,1} & & & & & B_{1,1} & & & \\ A_{2,1} & A_{2,2} & A_{2,3} & & & B_{2,1} & B_{2,2} & B_{2,3} & \\ A_{3,1} & A_{3,2} & A_{3,3} & & & B_{3,1} & B_{3,2} & B_{3,3} & \\ A_{4,1} & A_{4,2} & A_{4,3} & A_{4,4} & & B_{4,1} & B_{4,2} & B_{4,3} & B_{4,4} \\ \hline C_{1,1} & & & & & D_{1,1} & & & \\ C_{2,1} & C_{2,2} & C_{2,3} & & & D_{2,1} & D_{2,2} & D_{2,3} & \\ C_{3,1} & C_{3,2} & C_{3,3} & & & D_{3,1} & D_{3,2} & D_{3,3} & \\ C_{4,1} & C_{4,2} & C_{4,3} & C_{4,4} & & D_{4,1} & D_{4,2} & D_{4,3} & D_{4,4} \end{array}\right). \quad (4.13)$$

Again, we reorder the linear system according to first the spatial elements, then the angular degrees of freedom, then spatial degrees of freedom to obtain the following

matrix

$$M_{\text{reordered}} = \left(\begin{array}{cc|cccc|cc} A_{1,1} & B_{1,1} & & & & & & \\ C_{1,1} & D_{1,1} & & & & & & \\ \hline A_{2,1} & B_{2,1} & A_{2,2} & B_{2,2} & A_{2,3} & B_{2,3} & & \\ C_{2,1} & D_{2,1} & C_{2,2} & D_{2,2} & C_{2,3} & D_{2,3} & & \\ A_{3,1} & B_{3,1} & A_{3,2} & B_{3,2} & A_{3,3} & B_{3,3} & & \\ C_{3,1} & D_{3,1} & C_{3,2} & D_{3,2} & C_{3,3} & D_{3,3} & & \\ \hline A_{4,1} & B_{4,1} & A_{4,2} & B_{4,2} & A_{4,3} & B_{4,3} & A_{4,4} & B_{4,4} \\ C_{4,1} & D_{4,1} & C_{4,2} & D_{4,2} & C_{4,3} & D_{4,3} & C_{4,4} & D_{4,4} \end{array}\right). \quad (4.14)$$

The same block backward substitution may be implemented, except this time the linear solve for the solution values U_{AC2}, U_{BD2}, U_{AC3} and U_{BD3} must be performed as a single solve, i.e.,

$$\begin{pmatrix} U_{AC2} \\ U_{BD2} \\ U_{AC3} \\ U_{BD3} \end{pmatrix} = \begin{pmatrix} A_{2,2} & B_{2,2} & A_{2,3} & B_{2,3} \\ C_{2,2} & D_{2,2} & C_{2,3} & D_{2,3} \\ A_{3,2} & B_{3,2} & A_{3,3} & B_{3,3} \\ C_{3,2} & D_{3,2} & C_{3,3} & D_{3,3} \end{pmatrix}^{-1} \left[\begin{pmatrix} rhs_{AC2} \\ rhs_{BD2} \\ rhs_{AC3} \\ rhs_{BD3} \end{pmatrix} - \begin{pmatrix} A_{2,1} & B_{2,1} \\ C_{2,1} & D_{2,1} \\ A_{3,1} & B_{3,1} \\ C_{3,1} & D_{3,1} \end{pmatrix} \begin{pmatrix} U_{AC1} \\ U_{BD1} \end{pmatrix} \right] \quad (4.15)$$

The problem with this strategy is the possibility of having to perform extremely large solves if several spatial elements are coupled together by these two way element boundaries. For example, if the number of spatial elements coupled is N, each with order of polynomial approximation p, with order of approximation in angle q, then the size of the system to be solved will be $N(q+1)(p+1)^2$ by $N(q+1)(p+1)^2$. So if $p = q = 5$ we have a $216N$ by $216N$ linear system to solve. We could use MUMPS to solve these inner block systems once the size is over some prescribed tolerance.

4.1.4 Approximating the Eigenvalue Problem

The critical eigenvalue problem as introduced in Section 1.3.4 is of great interest practically; because of this we wish to extend our solver to the solution of the underlying eigenvalue problems.

To this end, we consider the two dimensional eigenvalue problem; in this setting the

transport, scatter and fission operators are as follows:

$$\begin{aligned} T\psi &:= \boldsymbol{\mu} \cdot \nabla_x \psi + \Sigma_t \psi, \\ S\psi &:= \int_I \frac{\Sigma_s}{2\pi} \psi \, \mathrm{d}\varphi, \\ F\psi &:= \int_I \frac{\nu\Sigma_f}{2\pi} \psi \mathrm{d}\varphi. \end{aligned}$$

Then recall that the generalised eigenvalue problem can be expressed as

$$(T - S)\psi = \lambda F \psi, \tag{4.16}$$

where $\lambda = 1/k_{\text{eff}}$.

Since we are solving this in a variational setting, the above operators are replaced by their variational formulations $\tilde{T}, \tilde{S}$ and $\tilde{F}$:

$$\begin{aligned} \tilde{T}(\psi, v) &= \int_\Omega \int_I (\boldsymbol{\mu} \cdot \nabla_x \psi + \Sigma_t \psi) \, v \, \mathrm{d}\varphi \mathrm{d}\boldsymbol{x}, \\ \tilde{S}(\psi, v) &= \int_\Omega \int_I \left(\int_I \frac{\Sigma_s}{2\pi} \psi(\boldsymbol{x}, \varphi') \, \mathrm{d}\varphi' \right) v \, \mathrm{d}\varphi \, \mathrm{d}\boldsymbol{x}, \\ \tilde{F}(\psi, v) &= \int_\Omega \int_I \left(\int_I \frac{\nu\Sigma_f}{2\pi} \psi(\boldsymbol{x}, \varphi') \, \mathrm{d}\varphi' \right) v \, \mathrm{d}\varphi \, \mathrm{d}\boldsymbol{x}, \end{aligned}$$

where $v \in V^{h_\Omega, h_I}$ and V^{h_Ω, h_I} is as defined in (3.20). This leads to the generalised eigenvalue problem

$$(\tilde{T} - \tilde{S})(\psi, v) = \lambda \tilde{F}(\psi, v). \tag{4.17}$$

To solve these we need to be able to provide the action of the operator $\tilde{F}$. As an alternative we transform (4.17) into a standard eigenvalue problem by performing the following manipulations. Starting with the following,

$$\tilde{T}(\psi, v) = \tilde{S}(\psi, v) + \lambda \tilde{F}(\psi, v),$$

moving $\tilde{S}$ to the other side leads to

$$(\tilde{T} - \tilde{S})(\psi, v) = \lambda \tilde{F}(\psi, v),$$

now multiplying by $\tilde{T}^{-1}$

$$(I - \tilde{T}^{-1}\tilde{S})(\psi, v) = \lambda \tilde{T}^{-1}\tilde{F}(\psi, v),$$

re arranging

$$k_{\text{eff}}(I - \tilde{T}^{-1}\tilde{S})(\psi, v) = \tilde{T}^{-1}\tilde{F}(\psi, v)$$

and finally applying the inverse of the operator $(I - \tilde{T}^{-1}\tilde{S})$ to both sides, we obtain the standard eigenvalue problem

$$k_{\text{eff}}(\psi, v) = (I - \tilde{T}^{-1}\tilde{S})^{-1}\tilde{T}^{-1}\tilde{F}(\psi, v),$$

where we have used the fact that the operator $(I - \tilde{T}^{-1}\tilde{S})$ is always non-singular [45, 71]. As this is now a standard eigenvalue problem we can apply the power method [162] as long as we can apply the operator $(I - \tilde{T}^{-1}\tilde{S})^{-1}\tilde{T}^{-1}\tilde{F}$. To this end, we make use of the Neumann sum expansion

$$(I - \tilde{T}^{-1}\tilde{S})^{-1} = \sum_{n=0}^{\infty}(\tilde{T}^{-1}\tilde{S})^n,$$

leading to the approximation

$$(I - \tilde{T}^{-1}\tilde{S})^{-1}\tilde{T}^{-1}\tilde{F} = \tilde{T}^{-1}\tilde{F}\psi + \tilde{T}^{-1}\tilde{S}\tilde{T}^{-1}\tilde{F}\psi + (\tilde{T}^{-1}\tilde{S})^2\tilde{T}^{-1}\tilde{F}\psi + \cdots + (\tilde{T}^{-1}\tilde{S})^n\tilde{T}^{-1}\tilde{F}\psi$$

where we continue adding terms until the difference between successive terms is below some user prescribed tolerance.

After discretisation, to compute the above Neumann sum, we start with our numerical eigenfunction ψ_h^n from the previous step of the power method. We first wish to compute the action of $\tilde{T}^{-1}\tilde{F}$ on ψ_h^n. To do this we form the following linear system, using our DG discretisation

$$\tilde{T}u_0 = \tilde{F}\psi_h^n,$$

solving this gives us $u_0 = \tilde{T}^{-1}\tilde{F}\psi_h^n$ which is the first term of the Neumann expansion. The second term in the Neumann expansion $\tilde{T}^{-1}\tilde{S}\tilde{T}^{-1}\tilde{F}\psi_h^n$, To this end we solve the linear system

$$\tilde{T}u_1 = \tilde{S}u_0,$$

which results in

$$u_1 = \tilde{T}^{-1}\tilde{S}u_0 = \tilde{T}^{-1}\tilde{S}\tilde{T}^{-1}\tilde{F}\psi_h^n$$

which is the second term in the expansion. This approach can be continued until we compute the u_m such that the difference between u_m and u_{m-1} is less than some prescribed tolerance in a suitable norm. The Neumann sum is then computed as follows

$$(I - \tilde{T}^{-1}\tilde{S})^{-1}\tilde{T}^{-1}\tilde{F}\psi_h^n = u_0 + u_1 + u_2 + \ldots u_n$$

Remark 15. *We remark that in this setting the eigenalue package ARPACK cannot be used since we do not explicitly know the action of the operator $\tilde{F}$ appearing on the right hand side of the generalised eigenvalue problem.*

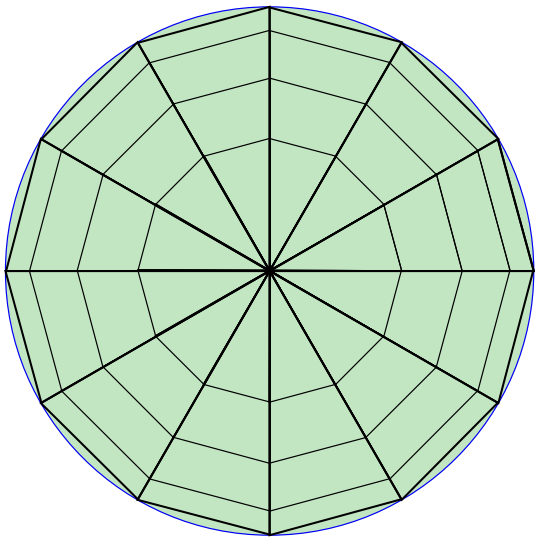


Figure 4.5: A graded mesh of the angular domain D of the pseudo 3d problem.

4.2 Implementation Issues Specific to the Pseudo 3D solver

We now discuss some implementation issues that are specific to the discretisation for the pseudo 3D problem.

4.2.1 Angular Mesh Generation

Fo the pseudo 3D problem the angular domain is the unit disc; thereby it is necessary to partition this into elements. Noting that when $|\mu| = 1$, which occurs on the boundary of the disc the following

$$\mathrm{d}A = \frac{1}{(1 - |\boldsymbol{\mu}|^2)^{1/2}} \mathrm{d}\boldsymbol{\mu},$$

is undefined. This leads to a singularity in the computation of the scalar flux, so we cannot use a quadrature that contains the nodes of an element as a quadrature point. This is easily solved as we use tensor product Gaussian quadrature where all quadrature points are in the interior of an element.

When using Gaussian quadrature on a mesh composed of irregular triangles as generated by the mesh generator Triangle, however, the singularity on the boundary still results in significant errors in the quadrature. Because of this we seek a mesh which is graded towards the boundary, such as that shown in Figure 4.5. To achieve such a mesh we construct a splitting of the unit disc such that the area of each element with respect to a certain weighting is constant.

Firstly we split the disc into N_a sectors, each of area $2\pi/N_a$; this gives us the radial

lines as shown in Figure 4.5. We now must determine the graded splitting of the radii. Let $\boldsymbol{\mu}$ be denoted by r for convenience; then we have that the area element on the unit sphere in spherical polar coordinates is

$$\mathrm{d}A = \frac{r}{(1-r^2)^{1/2}}\,\mathrm{d}r\,\mathrm{d}\varphi.$$

Hence the area of the unit sphere can be calculated as

$$\mathrm{Area}(S^2) = 2\int_0^{2\pi}\int_0^1 \frac{r}{(1-r^2)^{1/2}}\,\mathrm{d}r\,\mathrm{d}\varphi,$$

To find the radial splitting we perform an iterative procedure which divides the unit disc evenly with respect to the weighting found in the integrand above. Given two radii r_1 and r_2 (we start with $A_1 = 0$ and $A_2 = 1$) we find the radius r_m such that $r_1 \leq r_m \leq r_2$ such that r_m splits the weighted integral between r_1 and r_2 evenly in to two areas. That is, we seek r_m such that;

$$\int_{r_1}^{r_m} \frac{r}{(1-r^2)^{1/2}}\,\mathrm{d}r = \int_{r_m}^{r_2} \frac{r}{(1-r^2)^{1/2}}\,\mathrm{d}r,$$

that is

$$r_m = \sqrt{1-\left(\frac{\sqrt{1-r_1^2}+\sqrt{1+r_2^2}}{2}\right)^2}.$$

We can then perform the same procedure but on the intervals (r_1, r_m) and (r_m, r_2) to further split the domain. This procedure can be repeated until we have generated a mesh of sufficient resolution.

Remark 16. *As an example, suppose we wish to split each of the N_a sectors into 4 elements as shown in Figure 4.5, we start with $r_1 = 0$ and $r_2 = 1$. Performing the above approach gives $r_m = \sqrt{\frac{3}{4}} \approx 0.8660$. Then using the interval $(0, \sqrt{\frac{3}{4}})$ we obtain $r_m = \sqrt{\frac{7}{4}} \approx 0.6614$ and on the interval $(\sqrt{\frac{3}{4}}, 1)$ we have $r_m = \sqrt{1+\frac{1}{4}\left(\frac{\sqrt{3}}{2}-1\right)} \approx 0.9831$. This gives us a vector of 5 radial coordinates that can be used to define the angular elements, namely,*

$$r_i = \begin{pmatrix} 0 \\ 0.6614 \\ 0.8660 \\ 0.9831 \\ 1 \end{pmatrix}.$$

It can be seen that this has the desired affect of generating a mesh that is graded towards the boundary. To supplement this improvement of the angular mesh, we use AptoFEM to curve the edges of the boundary elements so that we also improve the approximation of the unit disc in this manner.

Computation of the Scalar Flux

For the pseudo 3D problem we have to make a slight modification to the procedure outlined before for computing the scalar flux since we have to include the scaling arising due to the change of coordinates from the surface of the unit sphere to the unit disc in two dimensions. More precisely, we have to compute the following,

$$\phi(\boldsymbol{x}) = \int_D \psi(\boldsymbol{x}, \varphi, \theta)(1 - |\boldsymbol{\mu}|^2)^{1/2} \, \mathrm{d}\boldsymbol{\mu}',$$

where D is the unit disc. As noted before, this scaling can be expressed in terms of θ in the following way,

$$\begin{aligned}
\phi(\boldsymbol{x}) &= \int_D \psi(\boldsymbol{x}, \varphi, \theta)(1 - |\boldsymbol{\mu}|^2)^{1/2} \, \mathrm{d}\boldsymbol{\mu}' \\
&= \int_D \psi(\boldsymbol{x}, \varphi, \theta) \frac{1}{\sqrt{1 - |\boldsymbol{\mu}|^2}} \, \mathrm{d}\theta \, \mathrm{d}\varphi \\
&= \int_D \psi(\boldsymbol{x}, \varphi, \theta) \frac{1}{\sin(\theta)} \, \mathrm{d}\theta \, \mathrm{d}\varphi.
\end{aligned}$$

As before we can expand the angular flux $\psi(\boldsymbol{x}, \varphi, \theta)$ on a particular angular element in terms of basis functions,

$$\psi(\boldsymbol{x}, \varphi, \theta) = \sum_{j=1}^{n} \sum_{l=1}^{m_{\kappa_I}} \Psi_{j,l}^{\kappa_I} \zeta_j(\boldsymbol{x}) \xi_l(\theta, \varphi).$$

Using the above and decomposing the integral over the unit disc into the summation of integrals over angular elements we have

$$\begin{aligned}
\phi(\boldsymbol{x}) &= \int_D \psi(\boldsymbol{x}, \varphi, \theta) \frac{1}{\sin(\theta)} \, \mathrm{d}\theta \, \mathrm{d}\varphi \\
&= \sum_{\kappa_I} \int_{\kappa_I} \sum_{j=1}^{n} \sum_{l=1}^{m_{\kappa_I}} \Psi_{j,l}^{\kappa_I} \zeta_j(\boldsymbol{x}) \xi_l(\theta, \varphi) \frac{1}{\sin(\theta)} \, \mathrm{d}\theta \, \mathrm{d}\varphi \\
&= \sum_{j=1}^{n} \left(\sum_{\kappa_I} \int_{\kappa_I} \sum_{l=1}^{m_{\kappa_I}} \Psi_{j,l}^{\kappa_I} \zeta_j(\boldsymbol{x}) \xi_l(\theta, \varphi) \frac{1}{\sin(\theta)} \, \mathrm{d}\theta \, \mathrm{d}\varphi \right) \\
&= \sum_{j=1}^{n} \left(\sum_{\kappa_I} \sum_{l=1}^{m_{\kappa_I}} \Psi_{j,l}^{\kappa_I} \underbrace{\left[\int_{\kappa_I} \xi_l(\theta, \varphi) \frac{1}{\sin(\theta)} \, \mathrm{d}\theta \, \mathrm{d}\varphi \right]}_{\text{scaling}} \right) \zeta_j(\boldsymbol{x}).
\end{aligned}$$

Now, on a particular angular element the term denoted by 'scaling' is a constant for each angular degree of freedom. However, unlike the two dimensional problem where the angular domain was the circumference of the unit disc every term here is potentially non-zero due to the scaling, and not just the term corresponding to $\xi_1(\theta, \varphi)$.
This makes the computation of the scalar flux more involved than in the two dimensional steady state mono energetic case.

Remark 17. *Notice, that this means that information from the higher order modes now contributes to the scalar flux.*

4.2.2 Computation of $\boldsymbol{\mu}$

We note that the quadrature points for the angular domain are given in cartesian coordinates, from these, we need to compute $\boldsymbol{\mu}$, φ and θ. Let the two dimensional angular quadrature point be denoted by $qk_{\text{ang}} = (x_{\text{ang}}, y_{\text{ang}})$. This is obtained by defining a Gaussian quadrature on every element of the angular mesh and needs to be manipulated in order to compute $\boldsymbol{\mu}$. Recalling that φ represents the azimuthal angle we can easily compute $\varphi = \arctan_{[0,2\pi]}(y_{\text{ang}}/x_{\text{ang}})$ where $\arctan_{[0,2\pi]}$ is the arc tangent function such that the range is $[0, 2\pi]$ and not $[-\pi, \pi]$. As well as computing φ we need to compute θ, or at least $\sin(\theta)$ - this can also be performed in a simple manner since $|\boldsymbol{\mu}| = \sqrt{x_{\text{ang}}^2 + y_{\text{ang}}^2} = \sin(\theta)$. Finally, we note that $\sin(\theta)\cos(\varphi) = x_{and}$ and that $\sin(\theta)\sin(\varphi) = y_{ang}$, hence we have $\boldsymbol{\mu} = qk_{ang}$.

CHAPTER 5

Numerical Experiments With Uniform Refinement

In this chapter we present numerical experiments for the two dimensional mono energetic steady state source problem using first a discrete ordinates discretisation in angle with an arbitrary order discontinuous Galerkin method in space. Then we present the results from the solver employing a DG discretisation in both space and angle for both the two dimensional mono energetic steady state problem and some critical eigenvalue computation benchmarks. Finally, we provide analogous results for the pseudo 3D source problem and some industrial eigenvalue benchmarks.

5.1 Two Dimensional Discrete Ordinates Discretisation

To demonstrate the convergence of the discrete ordinates discretisation in the angular domain when used in conjunction with a discontinuous Galerkin spatial discretisation we consider solving the two dimensional monoenergetic steady state neutron transport equation (1.6). Recall that we have the discretised problem (which we solve using Source Iteration): for each $i = 1, \dots N$ find $\psi_h \in V_{h,p}$ such that

$$A_{NT}(\psi_h, v_h) = l_{NT}(v_h) \quad \forall\, v_h \in V_{h,p} \tag{5.1}$$

with the bilinear form $A_{NT} : V_{h,p} \times V_{h,p} \to \mathbb{R}$ given by

$$\sum_{\kappa \in \mathcal{T}} \left(- \int_\kappa (\boldsymbol{\mu}_i \psi_h) \cdot \nabla v_h \, \mathrm{d}\boldsymbol{x} + \int_{\partial\kappa} \mathcal{H}(\psi_h^+, \psi_h^-, \boldsymbol{n}_\kappa) v_h^+ \, \mathrm{d}s \right)$$

and linear functional $l_{NT} : V_h \to \mathbb{R}$,

$$l_{NT}(v_h) = \sum_{\kappa \in \mathcal{T}} \left(\int_\kappa Q v_h \, \mathrm{d}\boldsymbol{x} + \int_\kappa \frac{\Sigma_s + \Sigma_f}{2\pi} \left(\sum_{j=1}^{N} \omega_j \psi_h(\boldsymbol{x}, \varphi_j) \right) \, \mathrm{d}\boldsymbol{x} \right)$$

Table 5.1: Convergence of a Discrete ordinates angular discretisation with a DG(0) discretisation in space.

Ordinates	No Dofs	$\|\phi_h - \phi\|_{L_2(\Omega)}$	Order	$\|\psi_h - \psi\|_{L_2(\Omega\times I)}$	Order
8	32	2.531528E-01		5.135307E-01	
16	256	9.961423E-02	1.35	2.468900E-01	1.06
32	2048	4.680659E-02	1.09	1.167889E-01	1.08
64	16384	2.535137E-02	0.88	5.650621E-02	1.05
128	131072	1.359664E-02	0.90	2.786843E-02	1.02
256	1048576	7.083038E-03	0.94	1.388158E-02	1.01
512	8388608	3.619629E-03	0.97	6.943199E-03	1.00

For the results presented here we assume that $\psi = \sin(\varphi)(x\cos(y) + y\sin(x))$, leading to a scalar flux $\phi = 0$. We take the material properties to be $\Sigma_t = 0.32640$, $\Sigma_f = 0.081600$, $\nu = 2.84$ and $\Sigma_s = 0.225216$. Defining the following norms,

$$\|\phi_h - \phi\|_{L_2(\Omega)} = \left(\int_\Omega (\phi_h - \phi)^2 \,\mathrm{d}\boldsymbol{x}\right)^{\frac{1}{2}} \tag{5.2}$$

and

$$\|\psi_h - \psi\|_{L_2(\Omega\times I)} = \left(\int_\Omega \int_I (\psi_h - \psi)^2 \,\mathrm{d}\varphi \,\mathrm{d}\boldsymbol{x}\right)^{\frac{1}{2}}, \tag{5.3}$$

which we refer to as the L_2–norm of the scalar and angular fluxes respectively, we can investigate the convergence rate of the discrete ordinates discretisation proposed. We also define the number of degrees of freedom, no dofs, to be

$$\text{no dofs} = \text{number of ordinates} \times \text{number of spatial dofs}$$

Results for the discrete ordinates discretisation with a DG(0) (that is, a piecewise polynomial approximation in space) discretisation in space are presented in Table 5.1, similarly for a linear approximation in Table 5.2, quadratic approximation in Table 5.3 and finally a cubic approximation in Table 5.4. These can be seen graphically in Figure 5.1

As we predicted, the rate of convergence is limited by the discrete ordinates scheme, and the best rate we can obtain is $O(h^2)$ convergence of the scalar flux; the observed convergence rate of the angular flux is in fact only $O(h^{\frac{3}{2}})$. We also notice, that in fact increasing the polynomial degree in space is detrimental in this case, giving slightly larger errors for $p = 2$ and $p = 3$ than for $p = 1$. These results highlight the potential benefit a higher order discretisation in angle could bring, as we shall see in the next section.

Table 5.2: Convergence of a Discrete ordinates angular discretisation with a DG(1) discretisation in space.

Ordinates	No Dofs	$\|\phi_h - \phi\|_{L_2(\Omega)}$	Order	$\|\psi_h - \psi\|_{L_2(\Omega\times I)}$	Order
8	128	3.663290E-01		3.531264E-01	
16	1024	1.067422E-01	1.78	1.430795E-01	1.30
32	8192	2.828434E-02	1.92	5.317593E-02	1.43
64	65536	7.253903E-03	1.96	1.920836E-02	1.47
128	524288	1.835293E-03	1.98	6.859081E-03	1.49
256	4194304	4.614856E-04	1.99	2.436692E-03	1.49
512	33554432	1.157000E-04	2.00	8.635260E-04	1.50

Table 5.3: Convergence of a Discrete ordinates angular discretisation with a DG(2) discretisation in space.

Ordinates	No Dofs	$\|\phi_h - \phi\|_{L_2(\Omega)}$	Order	$\|\psi_h - \psi\|_{L_2(\Omega\times I)}$	Order
8	288	3.666868E-01		3.554966E-01	
16	2304	1.066189E-01	1.78	1.433719E-01	1.31
32	18432	2.824920E-02	1.92	5.322589E-02	1.43
64	147456	7.244953E-03	1.96	1.921748E-02	1.47
128	1179648	1.833026E-03	1.98	6.860736E-03	1.49
256	9437184	4.609141E-04	1.99	2.436989E-03	1.49

Table 5.4: Convergence of a Discrete ordinates angular discretisation with a DG(3) discretisation in space.

Ordinates	No Dofs	$\|\phi_h - \phi\|_{L_2(\Omega)}$	Order	$\|\psi_h - \psi\|_{L_2(\Omega\times I)}$	Order
8	512	3.668677E-01		3.557192E-01	
16	4096	1.066351E-01	1.78	1.434039E-01	1.31
32	32768	2.825164E-02	1.92	5.323270E-02	1.43
64	262144	7.245283E-03	1.96	1.921878E-02	1.47
128	2097152	1.833069E-03	1.98	6.860972E-03	1.49
256	16777216	4.609195E-04	1.99	2.437031E-03	1.49

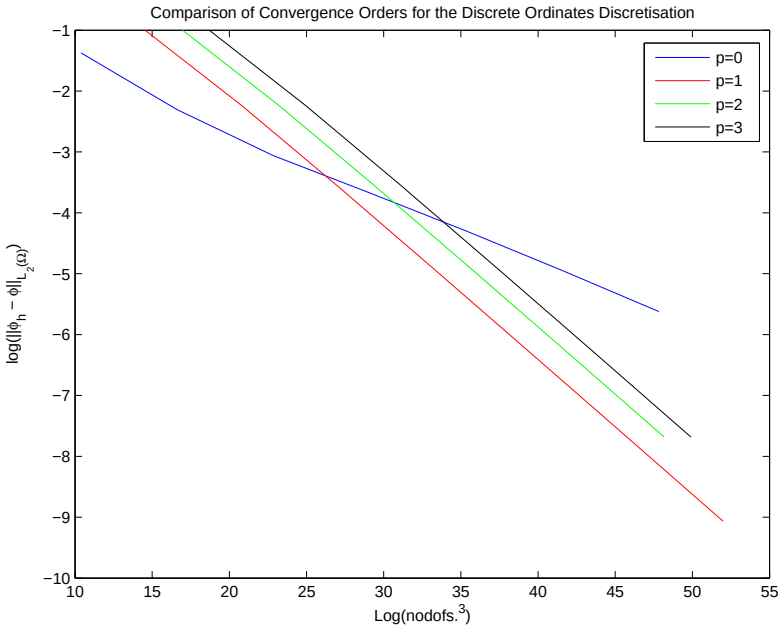


Figure 5.1: Relative convergence of the DO-DG discretisations for differing polynomials degree in space.

Table 5.5: Cross sections used for the two dimensional source problem tests

Cross section	Value
Σ_s	0.225216
Σ_t	0.32640
Σ_f	0.081600
ν	2.84

5.2 Two Dimensional Full DG Discretisation

5.2.1 The Source Problem

In all of these source problems we are considering the solution of the two dimensional monoenergetic steady state neutron transport equation (1.6) which for convenience we reproduce below.

$$\boldsymbol{\mu} \cdot \nabla_x \psi(\boldsymbol{x}, \varphi) + \Sigma_t \psi(\boldsymbol{x}, \varphi) = \frac{1}{2\pi} \int_0^{2\pi} (\Sigma_s + \nu\Sigma_f)\psi(\boldsymbol{x}, \varphi')\, \mathrm{d}\varphi' + Q(\boldsymbol{x}, \varphi) \quad \text{in } \Omega \times I \tag{5.4}$$

with boundary conditions determined by the analytic solution we prescribe. The following results are for the full DG discretisation, i.e., DG in space and angle. In addition, all of these results use the same physical cross sections, these are taken from the LA7 benchmark that we will come across later and are given in Table 5.5.

Test Case 1

Let the angular flux

$$\psi = xy \sin(\pi x) \sin(\pi y)\theta,$$

with associated scalar flux

$$\phi = 2\pi^2 xy \sin(\pi x) \sin(\pi y).$$

Using the cross sections contained in Table 5.5 we wish to investigate convergence of the full discontinuous Galerkin discretisation. We shall again examine convergence with respect to the scalar flux and the angular flux using norms (5.2) and (5.3). In addition, we will also investigate the convergence rate of a linear functional, as we did for the advection problem in Chapter 3. We shall, again, consider the mean value of the angular flux,

$$J(u) = \int_\Omega \int_I \omega\psi \, \mathrm{d}\varphi \, \mathrm{d}\boldsymbol{x},$$

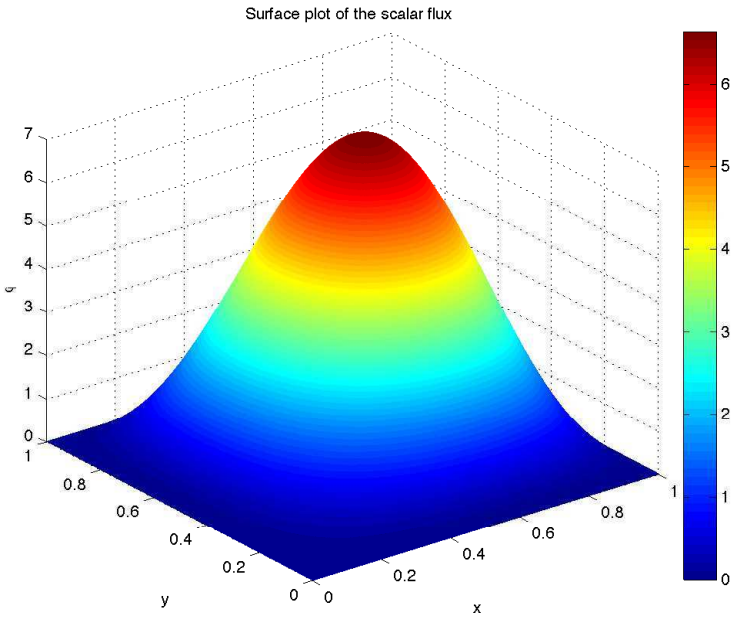


Figure 5.2: The analytical scalar flux for Test Case 1.

here with weight $\omega = xy^2\varphi^2$. In this case, the analytical value of the functional can be computed in Mathematica to be $J(u) = 9.205251234021861$. For reference a surface plot of the scalar flux is shown as Figure 5.2. Comparing Figure 5.2 with a pseudocolour plot of the converged numerical scalar flux, as shown in Figure 5.3 it can be seen that qualitatively the code has converged to the right solution. With reference to Tables 5.6, 5.7, 5.8 and Table 5.9, we can examine the convergence of the considered method in more detail. The full discontinuous Galerkin discretisation of the neutron transport equation does not suffer from a restriction on the convergence rate as the method utilising a discrete ordinates discretisation did. Indeed, we see optimal orders achieved; the order of convergence for $p = q = 0$ is markedly worse when being measured by the L_2–norm of the angular flux than for any other approximation order. We can also investigate the convergence of the functional $J(\psi)$ as given in Table 5.10 For the linear approximation it is seen that we approach the expected asymptotic rate of $O(h^{2(p+1)-1}) = O(h^3)$. The behaviour in the case of piecewise constants is trickier to explain, we believe that we have not reached the asymptotic regime in this case.

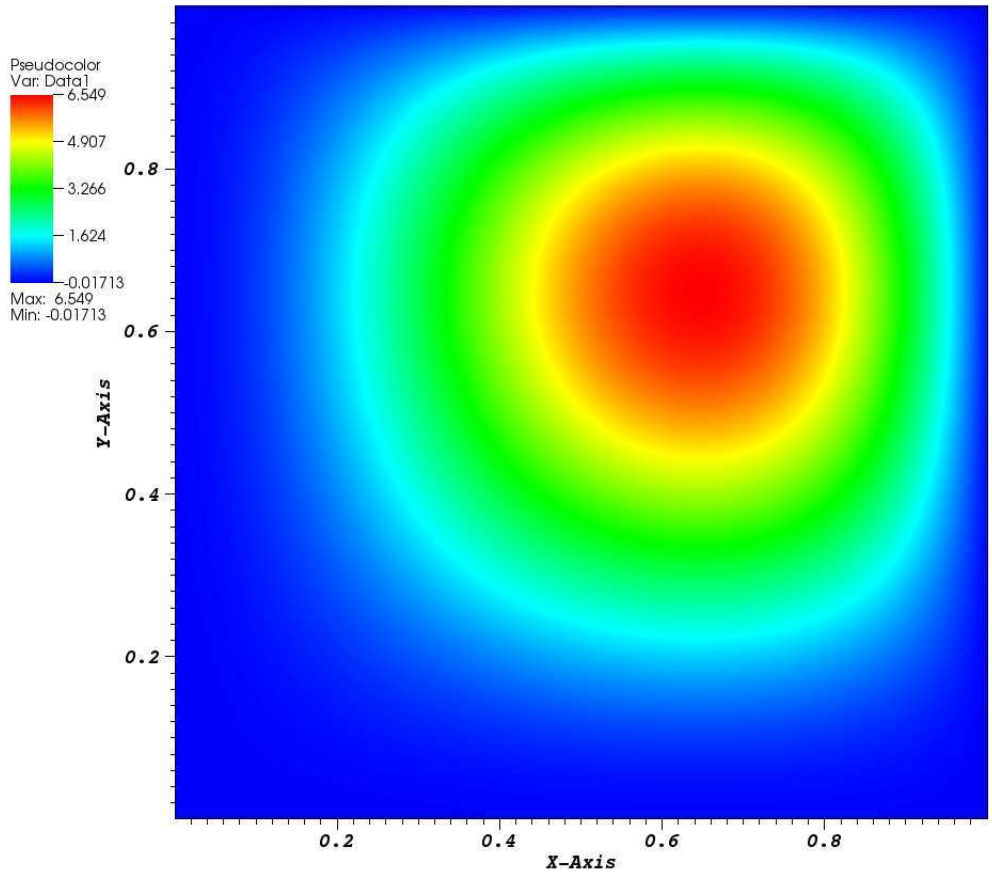


Figure 5.3: The computed scalar flux for Test Case 1.

Table 5.6: Convergence of a DG-DG discretisation with polynomial degree $p = q = 0$ for the first test problem.

No Angle Eles	No Dofs	$\|\phi_h - \phi\|_{L_2(\Omega)}$	Order	$\|\psi_h - \psi\|_{L_2(\Omega\times I)}$	Order
2	8	2.392107145		1.148978465	
4	64	1.332316108	0.84	0.851275391	0.43
8	512	6.90E-01	0.95	4.80E-01	0.83
16	4096	3.50E-01	0.98	2.59E-01	0.89
32	32768	1.78E-01	0.98	1.37E-01	0.92
64	262144	8.99E-02	0.99	7.12E-02	0.95
128	2097152	4.52E-02	0.99	4.52E-02	0.66

Table 5.7: Convergence of a DG-DG discretisation with polynomial degree $p = q = 1$ for the first test problem.

No Angle Eles	No Dofs	$\|\phi_h - \phi\|_{L_2(\Omega)}$	Order	$\|\psi_h - \psi\|_{L_2(\Omega\times I)}$	Order
2	64	7.500834E-01		4.687597E-01	
4	512	1.770892E-01	2.08	1.181036E-01	1.99
8	4096	4.421813E-02	2.00	2.982916E-02	1.99
16	32768	1.104667E-02	2.00	7.495364E-03	1.99
32	262144	2.762745E-03	2.00	1.881478E-03	1.99
64	2097152	6.909504E-04	2.00	4.715694E-04	2.00
128	16777216	1.727789E-04	2.00	1.180650E-04	2.00

Table 5.8: Convergence of a DG-DG discretisation with polynomial degree $p = q = 2$ for the first test problem.

No Angle Eles	No Dofs	$\|\phi_h - \phi\|_{L_2(\Omega)}$	Order	$\|\psi_h - \psi\|_{L_2(\Omega\times I)}$	Order
2	216	8.364283E-02		5.114040E-02	
4	1728	1.185519E-02	2.82	7.814308E-03	2.71
8	13824	1.515062E-03	2.97	1.004594E-03	2.96
16	110592	1.904219E-04	2.99	1.260287E-04	2.99
32	884736	2.384504E-05	3.00	1.574566E-05	3.00
64	7077888	2.982771E-06	3.00	1.966753E-06	3.00

Table 5.9: Convergence of a DG-DG discretisation with polynomial degree $p = q = 3$ for the first test problem.

No Angle Eles	No Dofs	$\|\phi_h - \phi\|_{L_2(\Omega)}$	Order	$\|\psi_h - \psi\|_{L_2(\Omega\times I)}$	Order
2	512	1.498632E-02		9.325590E-03	
4	4096	9.171288E-04	4.03	5.898273E-04	3.98
8	32768	5.724592E-05	4.00	3.692886E-05	4.00
16	262144	3.579405E-06	4.00	2.311638E-06	4.00
32	2097152	2.239206E-07	4.00	1.446503E-07	4.00

Table 5.10: Convergence of the linear functional for the first test problem, employing a DG-DG method.

	$p = 0$		$p = 1$	
No Angle Eles	$J(\psi) - J(\psi_h)$	Order	$J(\psi) - J(\psi_h)$	Order
2	5.011543014		1.639331E-01	
4	1.004046327	2.32	1.607784E-02	3.35
8	0.218373463	2.20	2.457070E-03	2.71
16	3.99E-02	2.45	3.294583E-04	2.90
32	4.64E-03	3.10	4.359673E-05	2.92
64	-1.25E-03	1.90	5.640226E-06	2.95
128	-1.48E-03		7.087954E-07	2.99

The Effect of varying Σ_s

As mentioned in Remark 14 the value of the scattering cross section Σ_s can have a dramatic influence on the convergence of the outer Source Iteration. For the problems considered in this thesis, we typically observe that the number of source iterations is around 25. We consider the solution of (5.4) where we prescribe $\psi(\boldsymbol{x}, \varphi) = x\cos(y) + y\sin(x)$, which gives a scalar flux $\phi(\boldsymbol{x}) = 2\pi(x\cos(y) + y\sin(x))$, and source term,

$$\begin{aligned} Q(\boldsymbol{x}, \varphi) &= \cos(\varphi)(y\cos(x) + \cos(y)) + \sin(\varphi)(\sin(x) - x\sin(y)) \\ &\quad + \Sigma_t(x\cos(y) + y\sin(x)) - \frac{(\Sigma_s + \nu\Sigma_f)}{2\pi} 2\pi(x\cos(y) + y\sin(x)) \end{aligned}$$

For the material cross sections we let $\Sigma_t = 0.32640$, $\Sigma_f = 0.081600$, $\nu = 2.84$ and let Σ_s vary. The effect of varying Σ_s is shown in Table 5.11 and graphically in Figure 5.4.

Table 5.11: How Σ_s effects the convergence of Source Iteration

Σ_s	Number of Source Iterations	Σ_s	Number of Source Iterations
0.225216	20	1.45	101
0.3	22	1.5	112
0.5	28	1.55	126
0.75	37	1.6	143
1.0	50	1.65	164
1.05	53	1.75	193
1.1	57	1.8	233
1.25	71	1.85	392
1.3	77	1.9	589
1.35	84	1.95	1183
1.4	92	2.00	Does not converge

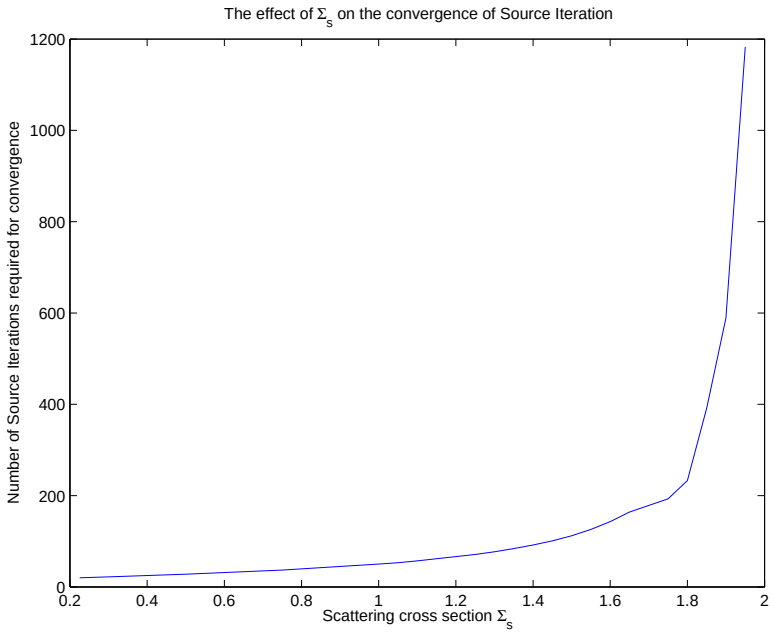


Figure 5.4: Graphical representation of the effect that Σ_s has on the convergence of the source iteration

For $\Sigma_s = 2$ the source iteration does not converge. This is what we would expect as in this case since,

$$(\Sigma_s + \nu\Sigma_f)/2\pi = 0.355193 > \Sigma_t = 0.32640$$

and so the iteration is not guaranteed to converge. Despite that large increase in the number of iterations for small increases in the scattering cross section, in practice we see this is not an issue for the problems considered in this thesis. If we wished to extend the code to shielding computations it would be an issue, as then the materials used for the shields will be optically thick.

5.2.2 The Critical Eigenvalue Benchmarks

It is significantly quicker to solve a neutron transport eigenvalue problem in the two dimensional case than in the pseudo 3D case, so ideally to validate the DG-DG method further we could perform some two dimensional critical eigenvalue calculations. Unfortunately there are no suitable industrial benchmarks for the two dimensional problem as considered in this thesis. Because of this some two dimensional eigenvalue benchmarks were developed in conjunction with Baker [26] and the full DG discretisation can be benchmarked against the results (which we have to two decimal places) obtained by the characteristics code of Baker [27]. This section presents results of three of these available benchmarks.

Test Case 1

Here we consider the spatial domain to be the square $(x, y) \in [0, 4] \times [0, 4]$ and the following coefficients $\Sigma_t = 0.75$, $\Sigma_f = 0.25$, $\Sigma_s = 0.25$ and $\nu = 2.0$. From David Baker's work we have a reference eigenvalue of $k_{eff}0.60$ to 2 significant figures which was computed with a discrete ordinates long characteristics code with a 32×32 spatial grid and 32 angular ordinates.
The results of the discontinuous Galerkin code with a piecewise constant discretisation in both the angular and spatial domains are given in Table 5.12. These tables contain the numerical eigenvalue obtained along with the number of iterations required by the power method to find the dominant eigenvalue. Note that the following identity for the total number of degrees of freedom (dofs) holds

$$\text{Total number of dofs} = \text{No angles} \times \text{No spatial dofs} \times \text{No angular dofs per angle}$$

We note the slow convergence of the $p = q = 0$ approximation with much better results for higher order discretisations. The higher order results suggest that the eigenvalue is 0.5957 to 4 decimal places. A plot of the converged dominant eigenfunction scalar flux is shown in Figure 5.5. We observe the expected gradual increase in flux from the vacuum boundary conditions to a peak in the centre of the domain. It can also be confirmed that the expected behaviour is observed if you increase Σ_f and/or ν; namely an increase in the k_{eff} eigenvalue. For example, keeping all other cross sections the same but increasing Σ_f to 0.45 results in an eigenvalue $k_{eff} = 1.05939780$, meaning that in this case the system is supercritical.

Table 5.12: Eigenvalues for 2D Eigenvalue Test Case 1 with $p = 0$, $q = 0$.

No Angles	Spatial Dofs	Total Dofs	Eigenvalue	Power Iterations
2	4	8	0.63157894	2
4	16	64	0.55894925	27
8	64	512	0.57925861	27
16	256	4096	0.58708930	26
32	1024	32768	0.59114517	26
64	4096	262144	0.59333141	25
128	16384	2097152	0.59448243	25
256	65536	16777216	0.59507500	25

Table 5.13: Eigenvalues for 2D Eigenvalue Test Case 1 with $p = 1$, $q = 1$.

No Angles	Spatial Dofs	Total Dofs	Eigenvalue	Power Iterations
2	16	64	0.58577820	32
4	64	512	0.59414683	25
8	256	4096	0.59559685	25
16	1024	32768	0.59566428	25
32	4096	262144	0.59567763	25
64	16384	2097152	0.59567971	25

Table 5.14: Eigenvalues for 2D Eigenvalue Test Case 1 with $p = 2$, $q = 2$.

No Angles	Spatial Dofs	Total Dofs	Eigenvalue	Power Iterations
2	36	216	0.59915072	31
4	144	1728	0.59584716	24
8	576	13824	0.59567954	25
16	2304	110592	0.59568004	25
32	9216	884736	0.59568006	25

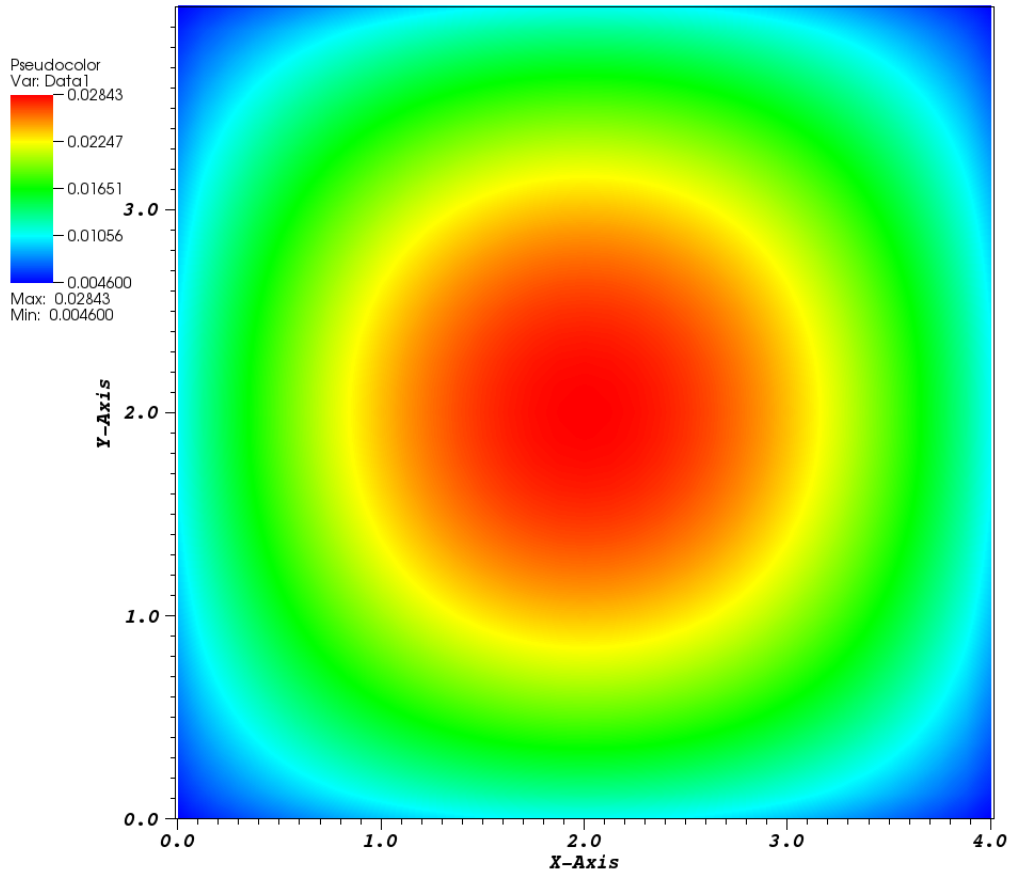


Figure 5.5: Converged Scalar Flux for Test Case 1

Table 5.15: Cross Sections for Test Case 2 and Test Case 3

Cross Section	Material 1	Material 2
σ_t	0.75	0.9
σ_s	0.25	0.2
σ_f	0.25	0.7
ν	2	4

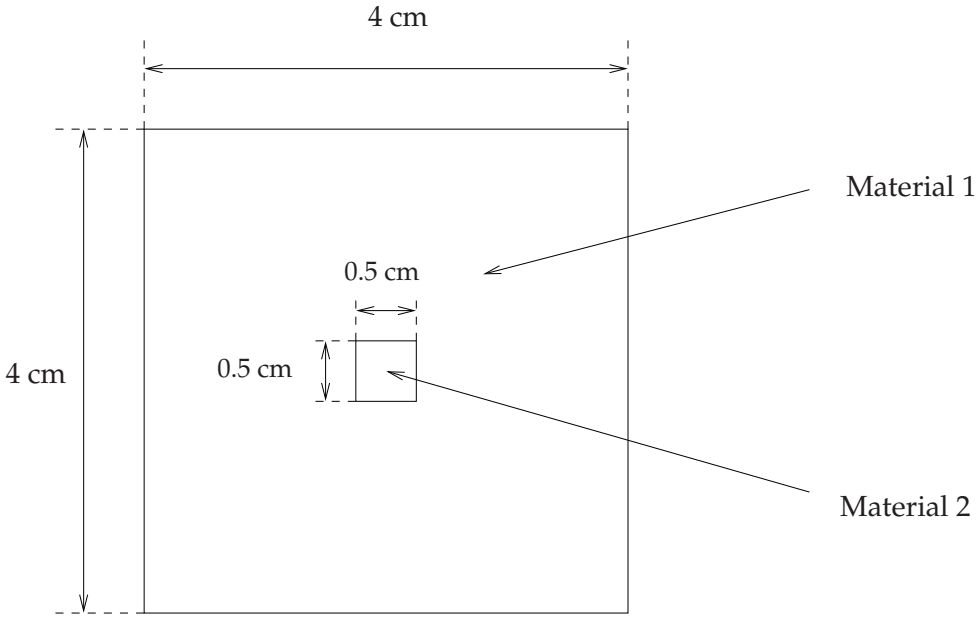


Figure 5.6: Domain for 2D Eigenvalue Test Case 2

Test Case 2

Test Case 1 considered previously is unrealistic in that the spatial domain is a single piece of homogeneous material. In applications it is more reasonable to consider an assembly made up of more than one material.

The simplest case is to consider a region of fissionable material surrounded by some form of moderator. Mathematically this corresponds to assigning different cross sections to various parts of the domain. Material 1 has the same cross sections as the homogeneous material considered in the above test case, however material 2, placed in the centre of the domain has a larger fission cross section. The spatial domain for Test Case 2 is shown in Figure 5.6 and the cross sections used are given in Table 5.15.

As Material 2 has a larger fission cross section and more neutrons released per fission we would expect a higher scalar flux in the region of material 2 and a higher k effective eigenvalue. Baker provides a reference value of 0.83 for the k effective eigenvalue. Again the high order codes all converge to this eigenvalue (albeit more runs of

Table 5.16: Eigenvalues for 2D Eigenvalue Test Case 2 with $p = 0, q = 0$.

No Angles	Spatial Dofs	Total Dofs	Eigenvalue	Power Iterations
2	4	8	0.6315789	2
4	16	64	0.8942461	24
8	64	512	0.7085404	33
16	256	4096	0.8849276	36
32	1024	32768	0.8538860	35
64	4096	262144	0.8405874	34
128	16384	2097152	0.8348132	33
256	65536	16777216	0.83219784	32

Table 5.17: Eigenvalues for 2D Eigenvalue Test Case 2 with $p = 1, q = 1$.

No Angles	Spatial Dofs	Total Dofs	Eigenvalue	Power Iterations
2	16	64	1.45289548	19
4	64	512	0.92966520	31
8	256	4096	0.65955428	38
16	1024	32768	0.82780334	33
32	4096	262144	0.82948193	31
64	16384	2097152	0.82974432	29

the $p = q = 0$ case would be desirable), and suggest that a more accurate value for the eigenvalue would be 0.82987. A plot of the converged dominant eigenfunction scalar

Table 5.18: Eigenvalues for 2D Eigenvalue Test Case 2 with $p = 2, q = 2$.

No Angles	Spatial Dofs	Total Dofs	Eigenvalue	Power Iterations
2	36	216	1.12083029	28
4	144	1728	0.72547460	43
8	576	13824	0.82556466	34
16	2304	110592	0.82978657	34
32	9216	884736	0.82978756	32

flux in Figure 5.7 clearly shows the expected localisation around material 2. We note that the number of power iterations does not appear to be as stable upon refinement as for Test Case 1.

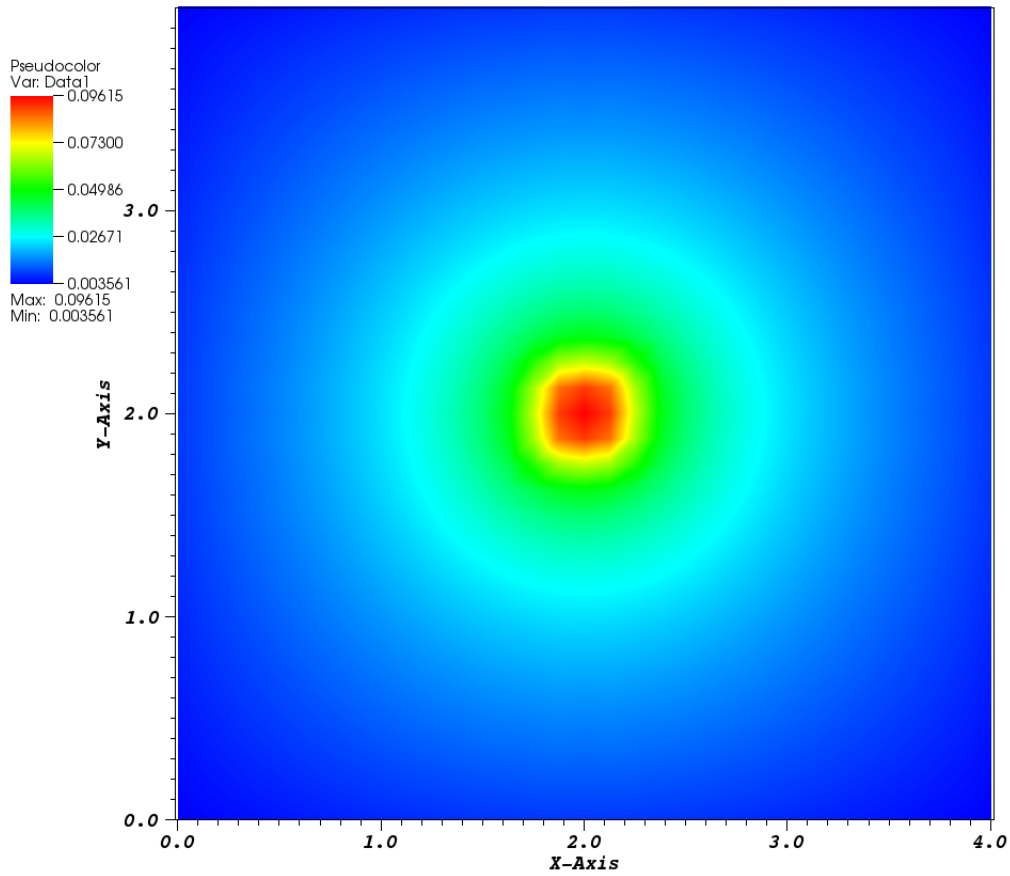


Figure 5.7: Converged Scalar Flux for Test Case 2

Test Case 3

As a final test case we consider a more complicated spatial domain; one that is intended to represent an assembly of four fuel rods surrounded by a moderator. The fuel rods are made up of material 2 and the moderator material 1, the cross sections of which are those appearing in Table 5.15. The spatial domain is shown in Figure 5.8

Baker provides a reference eigenvalue correct to two significant figures of 0.92. The results of the DG discretisation for varying polynomial degrees are shown in Tables 5.19, 5.20 and 5.21.

The converged dominant eigenfunction scalar flux is in some sense more interesting than the previous cases, showing high flux in the region of the fuel rods, with low flux elsewhere. We also provide a surface plot of the scalar flux in this case, shown in Figure 5.10.

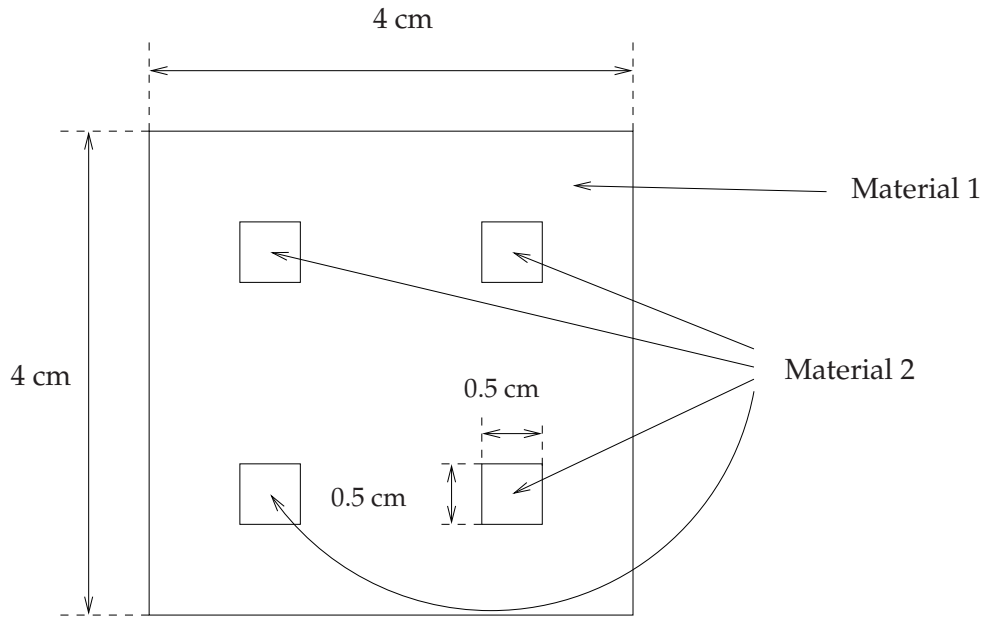


Figure 5.8: Domain for 2D Eigenvalue Test Case 3

Table 5.19: Errors for 2D Eigenvalue Test Case 3 with $p = 0$, $q = 0$.

No Angles	Spatial Dofs	Total Dofs	Eigenvalue	Power Iterations
2	4	8	0.6315789	2
4	16	64	1.1412657	29
8	64	512	0.8017940	25
16	256	4096	0.9828623	26
32	1024	32768	0.9507339	25
64	4096	262144	0.9357632	24
128	16384	2097152	0.9288944	24
256	65536	16777216	0.9256823	23

Table 5.20: Errors for 2D Eigenvalue Test Case 3 with $p = 1$, $q = 1$.

No Angles	Spatial Dofs	Total Dofs	Eigenvalue	Power Iterations
2	16	64	1.02773597	19
4	64	512	1.03878004	31
8	256	4096	0.75871414	38
16	1024	32768	0.92204520	33
32	4096	262144	0.92236552	31
64	16384	2097152	0.92260986	29

Table 5.21: Errors for 2D Eigenvalue Test Case 3 with $p = 2, q = 2$.

No Angles	Spatial Dofs	Total Dofs	Eigenvalue	Power Iterations
2	36	216	0.59915072	31
4	144	1728	0.79323043	33
8	576	13824	0.92027850	25
16	2304	110592	0.92250089	22
32	9216	884736	0.92265734	22

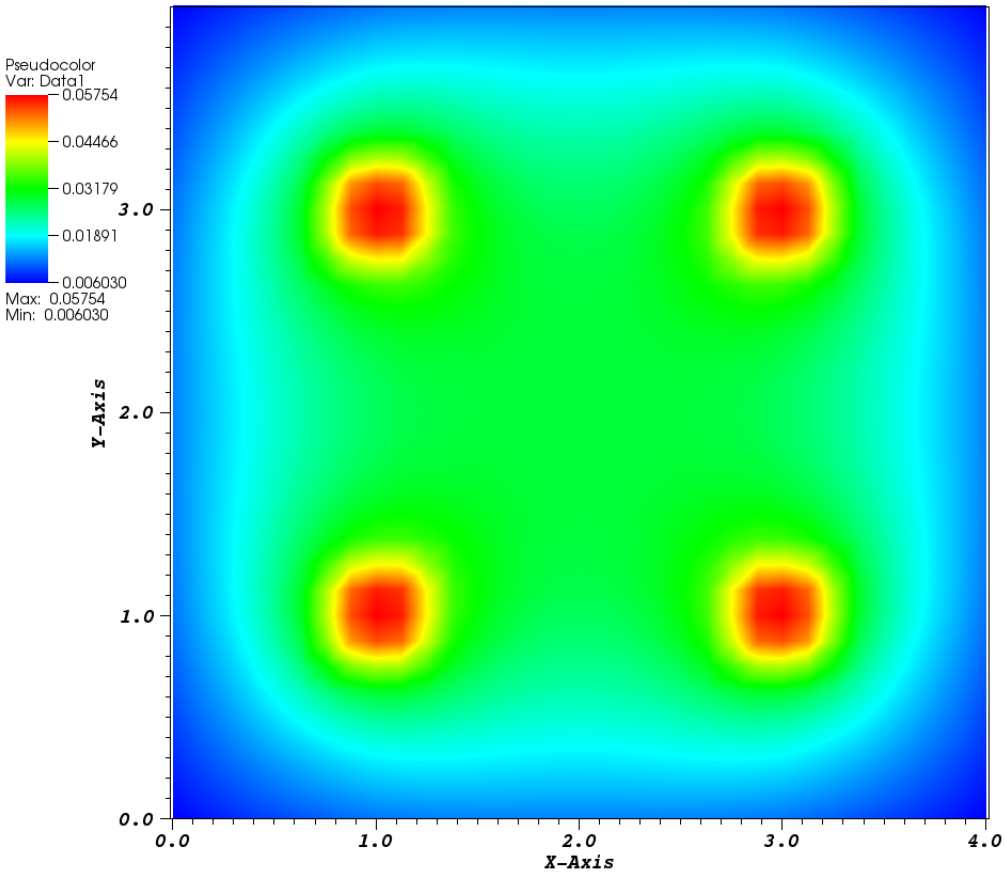


Figure 5.9: Converged Scalar Flux for Test Case 3

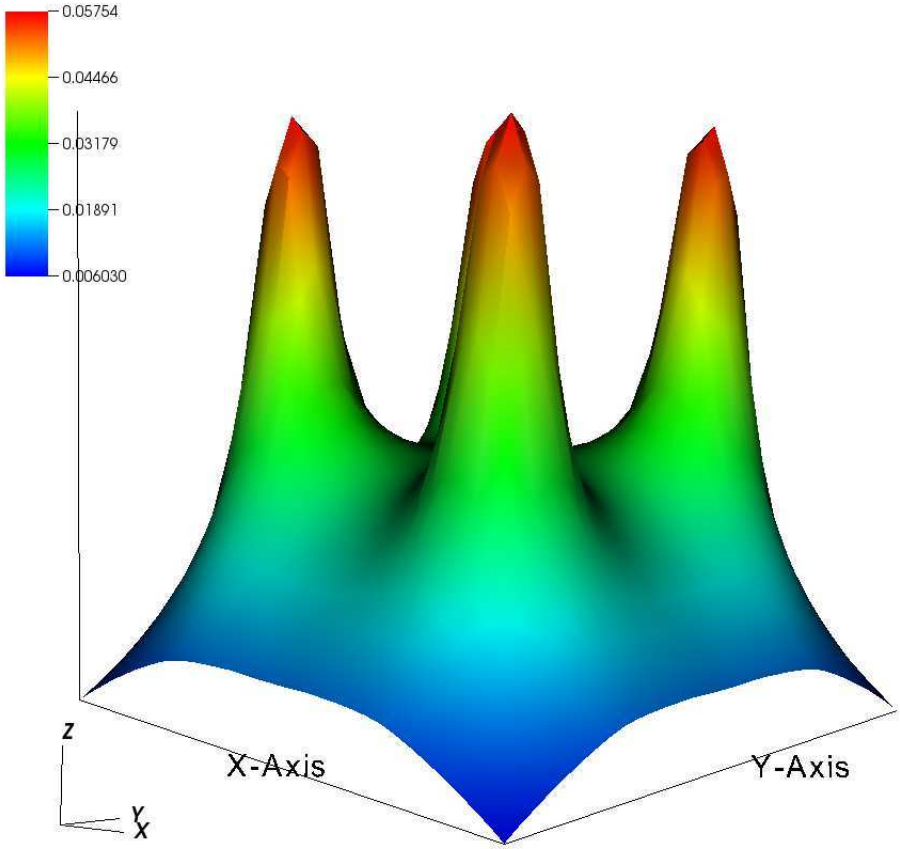


Figure 5.10: Converged Scalar Flux for Test Case 3

5.3 Pseudo 3D Full DG Discretisation

Here we give a numerical method for the pseudo 3D problem, showing that with the angular mesh discussed in Chapter 4 we can obtain optimal convergence rates for the scalar flux and a mean value functional, however, it appears that the convergence of the angular flux is suboptimal in that we don't observe the $O(h^{p+1})$ we would expect. We then introduce three industrial critical eigenvalue benchmarks from the literature that can be tackled using our code.

5.3.1 The Source Problem

Recall the pseudo 3D discontinuous Galerkin discretisation: find $\psi_h \in V^{h_\Omega, h_I}$ such that

$$\begin{aligned}
&\sum_{\kappa_D \in \mathcal{T}_D} \sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \Bigg[\int_{\kappa_D} \int_{\kappa_\Omega} \{-\psi_h \boldsymbol{\mu} \cdot \nabla_{\boldsymbol{x}} v_h + \Sigma_t \psi_h v_h\} \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\theta \mathrm{d}\varphi \\
&\quad + \int_{\kappa_D} \int_{\partial\kappa_\Omega \setminus \Gamma} \mathcal{H}(\psi_h^+, \psi_h^-, \boldsymbol{n}) v_h^+ \, \mathrm{d}s \, \mathrm{d}\theta \, \mathrm{d}\varphi \\
&\quad + \int_{\kappa_D} \int_{\partial\kappa_\Omega \cap \Gamma} \mathcal{H}(\psi_h^+, \psi_\Gamma^{\boldsymbol{\mu}}(\psi_h^+), \boldsymbol{n}) v_h^+ \, \mathrm{d}s \, \mathrm{d}\theta \mathrm{d}\varphi \Bigg] \\
&\quad = \sum_{\kappa_D \in \mathcal{T}_D} \sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \left[\int_{\kappa_D} \int_{\kappa_\Omega} \left(\frac{\Sigma_s(\boldsymbol{x}) + \nu(\boldsymbol{x}) \Sigma_f(\boldsymbol{x})}{2\pi} \int_0^{2\pi} \int_0^{\pi/2} \psi_h(\boldsymbol{x}, \theta', \varphi') \frac{1}{\cos(\theta)} \, \mathrm{d}\theta' \, \mathrm{d}\varphi' \right) \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\theta \, \mathrm{d}\varphi \right] \\
&\quad + \sum_{\kappa_D \in \mathcal{T}_D} \sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \left[\int_{\kappa_D} \int_{\kappa_\Omega} Q(\boldsymbol{x}, \theta, \varphi) v_h \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\theta \, \mathrm{d}\varphi \right] \quad \forall v_h \in V^{h_\Omega, h_I}.
\end{aligned} \tag{5.5}$$

We have already discussed the difficulty posed by the $\frac{1}{\cos(\theta)}$ term appearing in the definition of the scalar flux. In practice this manifests itself in sub optimal convergence rates for the angular flux as we shall now demonstrate. Indeed it appears that the angular flux is now limited to the same rate as for the discrete ordinates discretisation; however the rate of convergence of the scalar flux is not limited as was the case for the discrete ordinates discretisation of the angular dimension.

Let the angular flux be given by,

$$\psi = xy \sin(\pi x) \sin(\pi y) \sin(\varphi) \cos(\theta),$$

this leads to a scalar flux of $\phi = 0$. Using the material cross sections given in Table 5.5 and defining the source $Q(\boldsymbol{x}, \theta, \psi)$ to be

$$\begin{aligned}
Q = &\sin(\theta) \cos(\varphi) (y \cos(\theta) (\pi x \cos(\pi x) + \sin(\pi x)) \sin(\pi y) \sin(\varphi)) \\
&+ \sin(\theta) \sin(\varphi) (x \cos(\theta) \sin(\pi x) (\pi y \cos(\pi y) + \sin(\pi y)) \sin(\varphi)) \\
&+ 0.32640 xy \cos(\theta) \sin(\pi x) \sin(\pi y) \sin(\varphi),
\end{aligned}$$

Table 5.22: Convergence of a pseudo 3D discretisation with polynomial degree $p = q = 0$

No Angle Eles	No Dofs	$\|\phi_h - \phi\|_{L_2(\Omega)}$	Order	$\|\psi_h - \psi\|_{L_2(\Omega\times I)}$	Order
8	32	1.643170E-01		1.39E-01	
32	512	1.144210E-01	0.5221	9.32E-02	0.5728
128	8192	6.272210E-02	0.8763	5.26E-03	4.1479
512	131072	3.262879E-02	0.9428	2.82E-02	2.4214
2048	2097152	1.665750E-02	0.97	1.47E-02	0.9381
8192	33554432	8.421851E-03	0.984	7.55E-03	0.9621

Table 5.23: Convergence of a pseudo 3D discretisation with polynomial degree $p = q = 1$

No Angle Eles	No Dofs	$\|\phi_h - \phi\|_{L_2(\Omega)}$	Order	$\|\psi_h - \psi\|_{L_2(\Omega\times I)}$	Order
8	512	5.907741E-02		6.130789E-02	
32	8192	1.510575E-02	1.9675	1.926798E-02	1.6699
128	131072	3.764944E-03	2.0044	6.134180E-03	1.6513
512	2097152	9.388661E-04	2.0036	2.028414E-03	1.5965
2048	33554432	2.341207E-04	2.0037	6.899622E-04	1.5558

we have a well posed problem with solution ψ. As we wish to also investigate the convergence of functionals of the solution, we consider the mean value functional

$$J(u) = \int_\Omega \int_D \omega\psi \, \mathrm{d}\theta \, \mathrm{d}\varphi \, \mathrm{d}\boldsymbol{x},$$

with weighting in this case given by

$$\begin{aligned}\omega = &-\sin(\theta)\cos(\varphi)(y\cos(\theta)(\pi x\cos(\pi x)+\sin(\pi x))\sin(\pi y)\sin(\varphi))\\ &-\sin(\theta)\sin(\varphi)(x\cos(\theta)\sin(\pi x)(\pi y\cos(\pi y)+\sin(\pi y))\sin(\varphi))\\ &+0.32640xy\cos(\theta)\sin(\pi x)\sin(\pi y)\sin(\varphi),\end{aligned}$$

giving an analytical value $J(u) = 0.00682789$.

The convergence results for $p = q = 0$ are given in Table 5.22, similarly for $p = q = 1$, these are in Table 5.23. Finally the results for $p = q = 2$ are contained within Table Convergence of the linear functional is also shown in Table 5.25

Table 5.24: Convergence of a pseudo 3D discretisation with polynomial degree $p = q = 2$

No Angle Eles	No Dofs	$\|\phi_h - \phi\|_{L_2(\Omega)}$	Order	$\|\psi_h - \psi\|_{L_2(\Omega\times I)}$	Order
8	2592	6.669943E-03		1.727332E-02	
32	41472	9.423997E-04	2.8233	5.143702E-03	1.7477
128	663552	1.206784E-04	2.9652	1.736031E-03	1.567
512	10616832	1.509855E-05	2.9987	6.087938E-04	1.5118

Table 5.25: Convergence of the linear functional for the pseudo 3D problem.

	$p = 0$		$p = 1$	
No Angle Eles	$J(\psi) - J(\psi_h)$	Order	$J(\psi) - J(\psi_h)$	Order
8	1.806410E-02		7.270140E-03	
32	1.845629E-02		1.013368E-03	2.84
128	1.144982E-02	0.07	1.307158E-04	2.95
512	6.181905E-03	0.89	1.647571E-05	2.99
2048	3.192689E-03	0.95	2.061906E-06	3.00
8192	1.620306E-03	0.98		

5.3.2 Industrial Critical Eigenvalue Benchmarks

Here we present three benchmark problems to solve from the neutron transport literature. Due to the geometries involved, the elements used in the spatial domains now have to be triangles as opposed to quadrilaterals which we have used thus far. To generate the meshes required we use Shewchuk's mesh generator Triangle [136]. When generating meshes with triangle, one can specify a "maximum element area", this determines the size of the elements generated, and so we will tabulate these with our results. The first two benchmarks considered are posed on circular spatial geometries. When generating meshes for these we approximate the circle by a many sided polygon. Hence, in addition to decreasing an elements' size, we also improve our approximation to the circle. To this end, when we reduce the maximum area of an element by four we also increase the number of polygonal sides by a factor of two.

LA7 Benchmark

The first benchmark we consider comes from an article by Sood, Forster and Parsons of the Los Alamos National Laboratory [143]. In this article they collect together many

Table 5.26: Converged Eigenvalues for LA7 Benchmark

		$p=0$		$p=1$	
Max Ele Area	No faces	Eigenvalue	Power Iterations	Eigenvalue	Power Iterations
$1/4^0$	50	0.894405	26	0.928050	27
$1/4^1$	100	0.941936	18	0.964291	21
$1/4^2$	200	0.969204	20	0.982128	21

benchmarks from the literature, we shall concern ourselves with two of them. The first is known as the LA7 benchmark (due to the numbering scheme used in the paper). The goal, here, is to compute the *k*-effective eigenvalue k_{eff} as accurately as possible.

The LA7 benchmark is derived from an older article [159] and models an infinite cylinder of Plutonium, Pu-239 surrounded by vacuum boundary conditions. The spatial domain is a circle of radius $r = 4.279960cm$; for this radius $k_{eff} = 1$. Some results are shown in Table 5.26; these are promising as for both polynomial degrees the eigenvalue appears to be converging to the expected k_{eff}, however more runs are required to properly observe this. It would also be desirable to investigate higher degree polynomials as $p = 1$ appears to perform better than $p = 0$. In the table the maximum element area is that as given to Triangle, no faces describes how many faces the polygon we use to approximate the spatial domain has and power iterations reports how many iterations of the power method were required to converge to the eigenvalue in this case.

Remark 18. *We remark that we have computed eigenvalues for this benchmark with a discrete ordinates DG scheme and we converge to the eigenvalue well. The accuracy of the eigenvalue at each refinement step is in this case in line with those of Baker [27]. This confirms our observations about the difficulty quadrature is causing in the pseudo 3D setting and in fact improved results may be obtained if we consider the full three dimensional discretisation.*

LA9 Benchmark

The LA9 benchmark is more complicated than the *LA7* benchmark since it is composed of two materials. There is the same cylinder of Pu-239 used in the LA7 case, but now it is surrounded by an reflector material, in this case water. This reflector means that there is a good chance that neutrons leaving the Pu-239 will be reflected back in, to cause further fissions. This leads to a smaller critical radius of Pu-239 being required. The cross sections for the LA7 and LA9 benchmarks are shown in Table 5.27 The critical thickness of Plutonium 239 in this case is 3.397610cm, this is then surrounded by a

Table 5.27: Material cross sections for the LA7 and LA9 benchmarks

	$Pu-239$	H_20
Σ_t	0.32640	0.32640
Σ_s	0.225216	0.293760
Σ_f	0.081600	-
ν	2.84	-

Table 5.28: Converged Eigenvalues for LA9 Benchmark

		$p=0$		$p=1$	
Max Ele Area	No faces	Eigenvalue	Power Iterations	Eigenvalue	Power Iterations
$1/4^0$	50	0.863938	22	0.919426	24
$1/4^1$	100	0.922172	20	0.955621	20
$1/4^2$	200	0.960348	19		

water region of thickness 3.063725 giving a combined radius for the whole assembly of 6.461335. A typical triangle generated mesh for the LA9 benchmark that respects the material boundaries is shown in Figure 5.11. Some eigenvalue approximations are shown in Table 5.28 and the scalar flux for the dominant eigenvalue is shown in Figure 5.12 Note that in this case, our eigenvalue approximations start further away from unity than they did for the LA7 benchmark.

Maire and Talay Benchmark

The Maire and Talay benchmark is a more complicated benchmark that is more representative of a practical nuclear assembly. This was considered by Maire and Talay in 2006 [109]. The assembly is composed of of a rectangular domain, comprising five materials, two of which are fissile, as shown in Figure 5.13 with cross sections as given in Table 5.29.

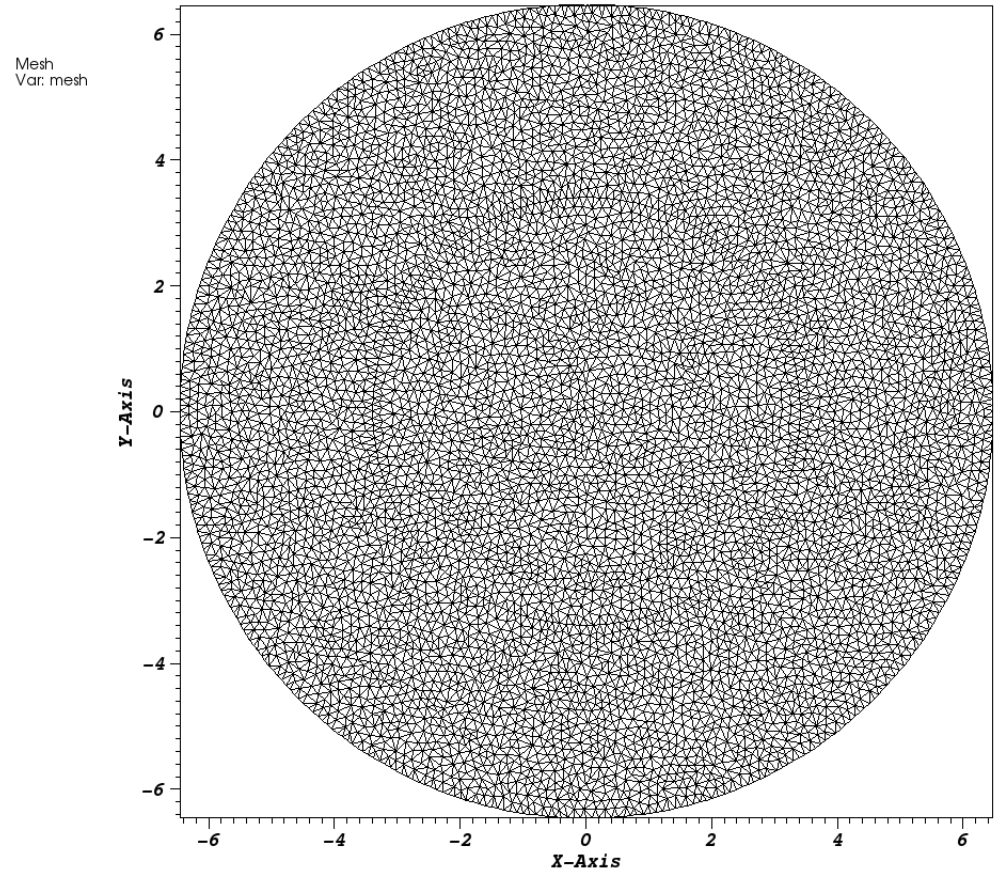


Figure 5.11: A mesh (generated by Triangle) of the spatial domain for the LA9 benchmark.

Table 5.29: Material cross sections for the LA7 and LA9 benchmarks

Material	1	2	3	4	5
Σ_t	6.0E-1	4.8E-1	7.0E-1	6.5E-1	9.0E-1
Σ_s	5.3E-1	2.0E-1	6.6E-1	5.0E-1	8.9E-1
$\nu\Sigma_f$	7.9E-2	-	4.3E-2	-	-

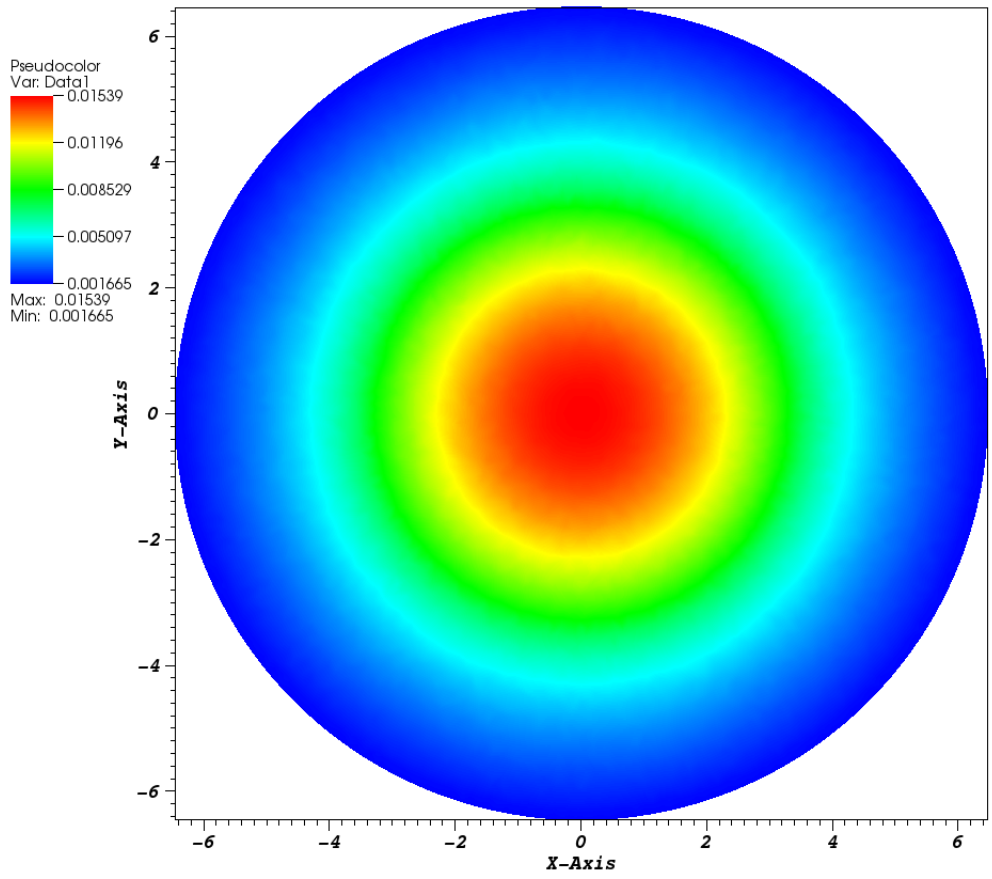


Figure 5.12: A mesh (generated by Triangle) of the spatial domain for the LA9 benchmark.

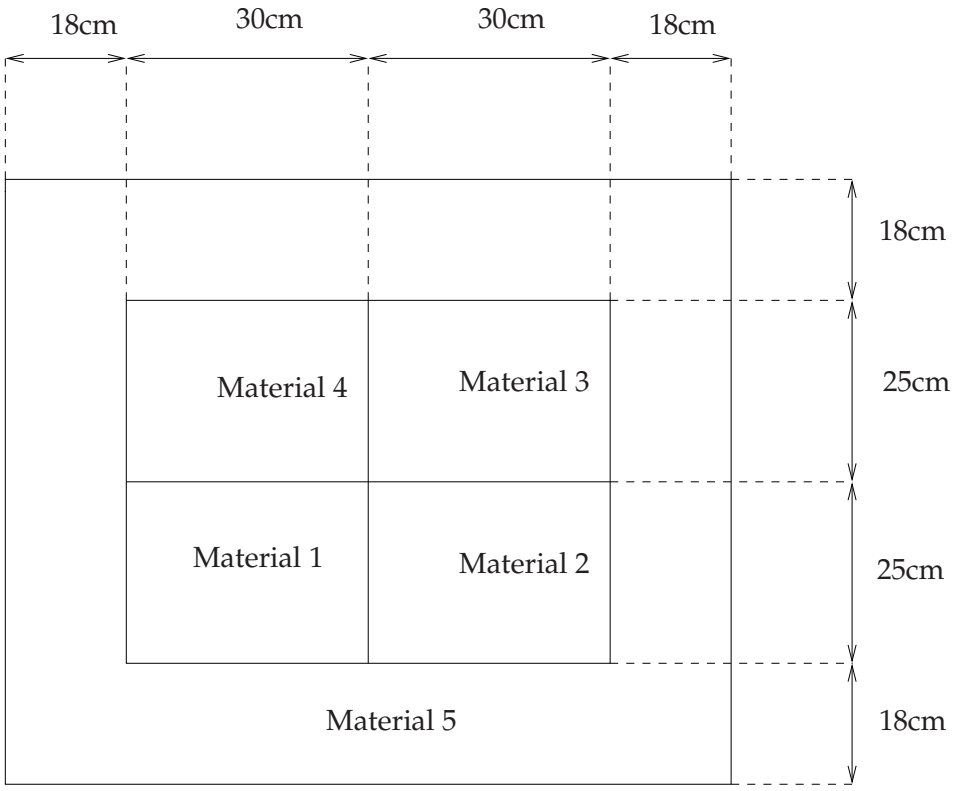


Figure 5.13: The spatial domain for the Maire and Talay benchmark problem

Table 5.30: Converged Eigenvalues for Maire and Talay Benchmark

	$p = 0$		$p = 1$	
Max Ele Area	Eigenvalue	Power Iterations	Eigenvalue	Power Iterations
$200/4^0$	0.563351	62		
$200/4^1$	0.692246	58		
$200/4^2$	0.806739	60		
$200/4^4$	0.890736	55		

Looking at Table 5.30 we observe slower convergence to unity than for the LA7 and LA9 benchmarks, also note the large number of power iterations that are required to converge to the eigenvalue in this case. This slower convergence was also noted by Baker [27] with his discrete ordinates in angle, characteristics in space scheme and indicates that an investigation into whether a method other than the power method would produce better results for this benchmark.

It is possible to obtain further results for these benchmarks by refining both the the angular and spatial domains. However, at present, these take a considerably long time to produce and so are not included here. Optimisation of the code is an important future task to enable us to solve industrial three dimensional benchmarks to a specified accuracy and will be discussed further in the conclusion.

CHAPTER 6

Adaptivity And *A Posteriori* Error Estimation

As we have seen in the previous chapter, obtaining numerical solutions of sufficient accuracy for both the neutron transport source problem and the critical eigenvalue problem is difficult at best, as a consequence we need to investigate the idea of adaptive mesh refinement. In this section we introduce the idea of adaptivity, before specifically investigating dual weighted residual (DWR) error estimation an its use for adaptivity. DWR error estimation is useful when we are interested in computing functionals of the solution to a problem. We first apply this technique to the advection reaction equation studied in Chapter 3. We then investigate DWR error estimation for the neutron transport source problem before considering the DWR approach in the context of eigenvalue problems; first for the eigenvalues of the Laplace operator and then for the neutron transport critical eigenvalue problem.

6.1 Introduction to Adaptivity

As seen in Chapter 1, the convergence of any finite element methods is dependent on the mesh width and also the polynomial degree. It is possible to globally refine the mesh (which we term uniform refinement, as used for the numerical results contained in the last chapter) or globally increase the polynomial degree in order to obtain h-convergence. Unfortunately this is generally inefficient, i.e., it is very expensive in terms of solve time and the number of degrees of freedom to obtain a solution to a pre specified tolerance. This is because the regions in the domain where a smaller mesh is required to resolve features such as singularities or boundary layers are likely to be a small fraction of the total domain. In practice, the size of problem that can be solved

is bounded by the time it takes and the amount of memory available on the computer that is being used, this has led to the development of adaptive techniques.

Conceptually, let us consider the abstract case where we approximate the solution of the following model problem, where A is some differential operator

$$A\psi = f,$$

with some appropriate boundary conditions. Performing a finite element discretisation we obtain the following weak formulation: Find ψ_{DG} such that

$$A_{DG}(\psi_{DG}, v) = l_{DG}(v) \quad \forall\, v \in V_h,$$

where V_h is a finite dimensional space on a mesh $\mathcal{T}_h = \{\kappa\}_{i=1}^{N}$, A_{DG} is a bilinear form and l_{DG} is a linear functional. We wish to design the finite dimensional space V_h in the most efficient way so that in some norm $\|\cdot\|$ the following condition is satisfied

$$\|\psi - \psi_{DG}\| \leq \text{Tol},$$

where Tol is some user defined tolerance. The aim is to adaptively refine the space as we solve the problem; to do this we require some kind of indicator function that indicates the relative error for different regions in the domain. Any *a priori* estimates that have been used to indicate the rate of convergence will be useless in this case as they will involve explicitly the analytical solution ψ and so are not computable. As a consequence, we seek an *a posteriori* error estimate such that the right hand side of the estimate is computable using only the approximate solution ψ_h and the data f of the associated problem. With this indicator, we can then design an adaptive algorithm in the style of Verfürth [154]:

1. Design an initial mesh.
2. Solve the discrete system.
3. Calculate an *a posteriori* error estimate.
4. If the error estimate is less than a prescribed accuracy Tol then stop, else go to 5.
5. Perform adaptive mesh refinement and go to 2.

Early adaptive algorithms used somewhat *ad hoc* criterions to guide refinement, such as the size of the gradient of the physical quantity of interest. There is intuitive reasoning behind this (regions of sharp gradient are likely to need more resolving to be accurately

represented), however it may not lead to an optimal mesh design. In this setting optimal means the most economical for achieving a certain tolerance or the most accurate for a given maximum nuber of degrees of freedom. In 1978, Babuška and Rheinboldt published some very influential papers ([22] , [23]) where they introduced the idea of performing *a posteriori* error anlaysis with respect to a natural energy norm $\|\cdot\|_E$ that is induced by the differential operator of the problem being investigated. When performing error analysis with respect to an energy norm one can obtain error estimates of the form

$$\|u - u_{DG}\|_{DG} \leq C\mathcal{E}(u, u_{DG}, R(u_h))$$

where $R(u_h) = f - Au_h$ is the finite element residual which is computable, and $\mathcal{E}(\cdot,\cdot,\cdot)$ just denotes some formula involving the residual which will normally be some form of norm (as there is then meaning to the relative size of the error). This norm will then be split up into N elemental contributions (where we have N elements in out partition of the mesh $\mathcal{T}$) that can be used to indicate regions with high and low error.

$$\mathcal{E}(u, u_{DG}, R(u_h)) \leq \sum_{\kappa \in \mathcal{T}} \eta_\kappa$$

Since the original papers of Babuška and Rheinboldt, this approach has been developed in many varied settings.

With some suitable error estimate we must then decide where to apply refinement and which type of refinement this is to be. The decision on where to refine a mesh is normally called a "marking strategy'" and two possible choices are:

- **Fixed Fraction** At each refinement step we order the elements according to the size of the absolute value of the local error indicators η_κ. Then a fixed fraction of the elements κ with the largest $|\eta_k|$ are refined and a fixed fraction of elements κ with the smallest $|\eta_k|$ are derefined. As an example, we could choose a refinement percentage of 20% and a derifinement percentage of 10%.

- **Equidistribution** Here the idea is to equilibrate the local error indicators by refining or derefining the elements κ according to the criterion

  $$\eta_k \approx \frac{\text{Tol}}{\sqrt{N}},$$

  where N denotes the number of elements in the current mesh.

Once we have selected elements for refinement we can then choose which refinement type to employ from:

- *h*–**refinement** Elements that have been marked for refinement we split into daughter elements. For elements marked for derefinement we combine neighbouring elements.
- *p*–**refinement** We increase the degree of approximation on elements that are marked for refinement and lower the degree on elements that are marked for derefinement.
- *r*–**refinement** The nodes of the mesh are moved in such a way as to reduce the approximation error.

Remark 19. *It is also possible to regenerate the whole mesh with a suitable mesh generator targeting the local parts where the error indicators were large, and increasing the resolution in those regions.*

As noted in Section 1.4.2 performing adaptive refinement is easier for a DG discretisation than for a conforming finite element discretisation. To perform *h*–refinement on a simplicial element, to prevent interior angles becoming small, one performs what is known as red refinement and split the element into 4 daughter elements as shown in Figure 6.1. As can be seen, this potentially leads to hanging nodes, which for a conforming discretisation can be problematic. To remove the hanging node a green refinement can be made, see Figure 6.1. This green refinement is temporary in the sense that if an element that has been green refined is selected for refinement at the next refinement step then the green refinement is removed before performing a red refinement step. When we employ a DG discretisation we can have an arbitrary number of hanging nodes (though, based on experience, we limit the number of hanging nodes per face to one) and so green refinement is not necessary. Performing *p*–refinement is again relatively easy with a DG discretisation as there is no need to match up approximations of differing polynomial orders across element boundaries, as you would need to for a conforming method. In 1987 Babuška & Suri [24] proved that for a function $u|_\Omega \in H^k(\Omega)$ where Ω is a domain and Π_{hp} is a projection onto a finite element space, the following approximation result holds

$$\|u - \Pi_{hp}u\|_{H^s(\Omega)} \leq C\,\frac{h^{\min(p+1,k)-s}}{p^{k-s}}\,\|u\|_{H^k(\Omega)}, \qquad 0 \leq s \leq \min(p+1,k),$$

where $\Pi_{hp}u$ denotes an interpolant of u. Schwab derived the following error bound for a real analytic function u on Ω in 1998 [135],

$$\|u - \Pi_{hp}u\|_{H^s(\Omega)} \leq C(u)\,h^{p+1-s}\,\mathrm{e}^{-bp}, \quad b > 0.$$

Here u_h is the finite element approximation which has been computed on a partition of Ω into quadrilaterals. This result implies that for a smooth solution exponential

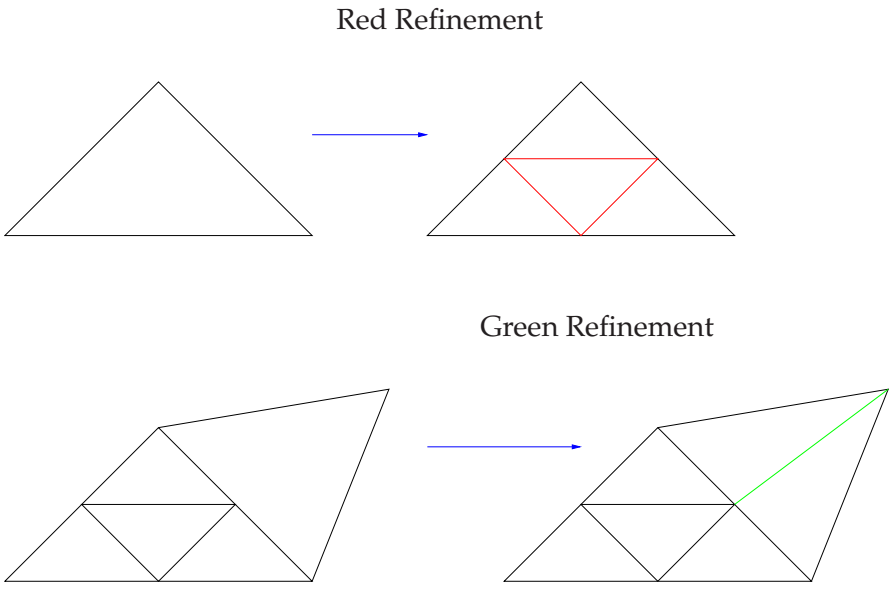


Figure 6.1: Mesh Refinement Strategies

convergence can be obtained when refining the polynomial degree. This implication could not be made using the result of Babuška & Suri since the constant C depended on the regularity of the solution, and so increased with k; hence it is not possible to take to the limit and obtain exponential convergence. The above result suggests that a combination of local h and p–refinement, so called hp–refinement strategies will lead to the "best" rates of convergence. Indeed, in practice it is possible to recover exponential convergence even for solutions with singularities by utilising a hp–adaptation stategy.The use of an hp–refinement algorithm introduces a further complexity in that once an element has been selected for refinement we must choose whether to perform h–refinement or p–refinement. There are various strategies for this, see the unpublished paper [113] for a comparehensive survey. All the methods attempt to estimate the smoothness [86] of the approximate solution u_{DG}; if u_{DG} is assessed to be smooth we perform p–refinement, else we perform h–refinement.
In this work we consider solely h–refinement, with hp–refinement being an obvious extension which will be discussed in the final chapter.

6.2 Dual Weighted Residual (DWR) adaptivity

Performing adaptive refinement with respect to some norm of the error will lead to achieving a certain accuracy in that norm with fewer degrees of freedom than uniform refinement, however we are often interested in approximating a functional of the numerical solution u_{DG}. We often refer to this functional as "the goal" or target functional. This desire has led to the development of *goal oriented* error estimates.

To this end, it is possible to employ duality arguments, similar to the common "Aubin - Nitsche duality trick" used in deriving a priori error estimates. Babuška (in his work with Miller) [19–21] is again credited with introducing the idea of using duality arguments in *a posteriori* error estimation. The survey paper by Eriksson, Estep, Hansbo and Johnson [70] provides a good introduction to this area. Becker and Rannacher [31–33] developed this approach into the Dual Weighted Residual (DWR) method which we describe in this section. Examples shown in [29] show the superiority of this approach over using more heuristic error indicators, resulting in optimal meshes with less refinement in areas of the domain where it is not necessary.

For completeness we introduce DWR error estimation in the setting of nonlinear problems see for example [33, 82, 83], despite both the advection problem and the neutron transport problem being linear; we will then restrict to the linear case.

Suppose that $\Omega \subset \mathbb{R}^d$ is an open bounded domain and that $\boldsymbol{N}$ is a nonlinear differential operator such that with suitable boundary conditions we can consider consider the nonlinear problem

$$\boldsymbol{N}u = 0 \quad \text{in } \Omega$$

Discretising the problem leads to the following nonlinear problem: find $u_{DG} \in V_{h,p}$ such that,

$$\mathcal{N}(u_{DG}, v_h) = 0 \quad \text{for all} v_h \in V_{h,p},$$

where $V_{h,p}$ is a discrete function space such that $V_{h,p} \subset V$ with V is a suitable space containing the analytical solution u. the space $V_{h,p}$ is defined on a mesh $\mathcal{T}$ of elements κ which is a partition of the domain Ω. The semi linear form $\mathcal{N} : V \times V \to \mathbb{R}$ is nonlinear in the first argument and linear in the second argument. We assume that $\mathcal{N}(u, v) = 0$ for all $v \in V$ and so the discretisation is consistent. We note that Galerkin orthogonality holds, i.e.,

$$\mathcal{N}(u, v_h) - \mathcal{N}(u_{DG}, v_h) = 0 \quad \text{for all } v_h \in V_{h,p}.$$

We define the mean value linearization of $\mathcal{N}(\cdot, v)$ in the standard way, namely,

$$\begin{aligned}\mathcal{M}(u, u_{DG}; u - u_{DG}, v) &= \mathcal{N}(u, v) - \mathcal{N}(u_{DG}, v), \\ &= \int_0^1 \mathcal{N}'[\theta u + (1-\theta)u_{DG}](u - u_{DG})\, v \, \mathrm{d}\theta,\end{aligned}$$

where $\mathcal{N}'[w](\cdot, v)$ denotes the Fréchet derivative evaluated at some $w \in V$. Now, suppose we are interested in the (possibly) nonlinear functional (which is differentiable),

we can define the mean value linearization of $J(\cdot)$ analogously

$$\begin{aligned}\bar{J}(u, u_h; u - u_h) &= J(u) - J(u_{DG}), \\ &= \int_0^1 J'[\theta u + (1-\theta)u_{DG}](u - u_{DG})\,\mathrm{d}\theta.\end{aligned}$$

We can now introduce the following formal dual problem: find $z \in V$ such that

$$\mathcal{M}(u, u_{DG}; w, z) = \bar{J}(u, u_{DG}; w) \quad \text{for all } w \in V.$$

Here we assume that the dual problem is well posed, this is dependent on both the form $\mathcal{N}(\cdot,\cdot)$ and the functional $J(\cdot)$. Finally, with $w = u - u_{DG}$ we can derive the following error representation formula

$$\begin{aligned}J(u) - J(u_{DG}) &= \bar{J}(u, u_{DG}; u - u_{DG}) \\ &= \mathcal{M}(u, u_{DG}; u - u_{DG}, z) \\ &= \mathcal{M}(u, u_{DG}; u - u_{DG}, z - z_h) \\ &= -\mathcal{N}(u_{DG}, z - z_h) \quad \text{for all}/z_h \in V_{h,p},\end{aligned}$$

where we have exploited consistency and Galerkin orthogonality. Defining $\mathcal{R}(u_{DG}, z - z_h) = -\mathcal{N}(u_{DG}, z - z_h)$ we have the following error representation formula

$$J(u) - J(u_{DG}) = \mathcal{R}(u_{DG}, z - z_h) \tag{6.1}$$

where the error representation formula comprises residuals of the primal numerical solution u_{DG} multiplied by weightings involving the dual solution z.

The derivation of the DWR error representation formula is more straight forward than in the nonlinear case since there is no need to perform a linearization. Let $\mathcal{L}$ be a linear differential operator, we can then consider the solution $u \in V$ to the following abstract linear problem,

$$\mathcal{L}u = f \quad \text{in } \Omega, \tag{6.2}$$

$$u = g \quad \text{on } \Gamma, \tag{6.3}$$

where $f \in L_2(\Omega)$, $g \in L_2(\Gamma)$ and Ω and Γ are as defined previously. Discretising the above, as usual we obtain the variational problem: find $u_{DG} \in V_{h,p}$ such that

$$A_{DG}(u_{DG}, v_h) = l(v_h) \quad \forall\, v_h \in V_{h,p}, \tag{6.4}$$

where $V_{h,p} \subset V$ is a finite element space on the triangulation $\mathcal{T}_{h,p}$ of elements of maximum diameter h such that the piecewise approximation polynomials are of degree p.

$A_{DG}(\cdot,\cdot) : V \times V \to \mathbb{R}$ is a bilinear form and $l(\cdot) : V \to \mathbb{R}$ is a linear functional. We also assume that the analytical solution u also satisfies (6.4); that is the discretisation is consistent such that

$$A_{DG}(u, v) = l(v) \quad \forall\, v \in V.$$

Galerkin orthogonality holds for the discretisation and so,

$$\begin{aligned} A_{DG}(u, v_h) - A_{DG}(u_{DG}, v_h) &= A_{DG}(u - u_{DG}, v_h) \\ &= 0 \end{aligned}$$

We suppose that the physical quantity we are interested in can be expressed as a functional $J(\cdot)$ of the solution and introduce the following adjoint or dual problem: Find $z \in V$ such that

$$A_{DG}(w, z) = J(w) \quad \forall\, w \in V, \tag{6.5}$$

where we assume the solution z exists. Under the assumption that (6.5) is well posed we can, following the same steps as in the nonlinear case, establish the following result for the error in the approximation of our target functional.

$$\begin{aligned} J(u) - J(u_{DG}) &= J(u - u_{DG}) \quad \text{by linearity} \\ &= A_{DG}(u - u_{DG}, z) \\ &= A_{DG}(u - u_{DG}, z - z_h) \quad \text{by Galerkin orthogonality} \\ &= l(z - z_h) - A_{DG}(u_{DG}, z - z_h) \\ &\equiv \mathcal{R}(u_{DG}, z - z_h) \end{aligned} \tag{6.6}$$

for all $z_h \in V_{h,p}$. We have equality at the moment, however, decomposing the above into a sum over all elements $\kappa \in \mathcal{T}_h$ and applying the triangle inequality we obtain the following weighted a posteriori error bound with local error indicators η_k,

$$|J(u) - J(u_h)| \leq \sum_{\kappa \in \mathcal{T}_h} |\eta_k| = \sum_{\kappa \in \mathcal{T}_h} |\mathcal{R}(u_{DG}, z - z_h)|_\kappa|. \tag{6.7}$$

The local error indicators η_κ involve the multiplication of finite element residuals depending on the primal solution u_{DG} with local weighting terms involving the difference between the dual solution z and its interpolant z_h and provide useful information concerning the global transport of error. These weights can be thought of as a representation of the sensitivity in the functional $J(\cdot)$ with respect to variations of the local element residuals.

Remark 20. *We note that, due to Galerkin orthogonality, for a DG method we can formally set the arbitrary function $z_h \in V_{h,p}$ to 0.*

The dual solution in (6.7) is not usually known analytically and in practice must be approximated numerically. In addition, due to the Galerkin orthogonality property the dual solution cannot be computed on the same finite dimensional subspace $V_{h,p}$ as the primal solution u_{DG} as this would lead to an error representation formula that is identically zero. Hence, the discrete dual solution $\tilde{z}_h$ which we use to approximate z in either error representation formula (6.1) or (6.6) must come from an enhanced space $\tilde{V}_{hp}$. In practice we use the approach as used in [76, 77, 82] among others and increase the degree of polynomial approximation from that used when computing the primal solution and compute the dual solution $\tilde{z}_h$ on the same spatial mesh $\mathcal{T}_h$.

To be able to compare error estimation techniques it is customary to refer to the (reciprocal) effectivity indicator as defined in [29]:

$$I_{eff} = \left| \frac{\sum_{\kappa \in \mathcal{T}_h} \eta_\kappa}{J(u) - J(u_h)} \right|$$

Ideally, this effectivity indicator will be (close to) one as it measures how much the error representation formula over estimates the true error.

Remark 21. *The above is known as a Type I error bound. It is possible to remove the unknown dual solution from the error representation formula and derive a Type II error bound. This is done by bounding norms of the dual solution z by bounds of the data for the dual problem by utilising the well posedness properties of the dual problem. However, even though we would now not have to compute an approximation to the dual problem, thus saving time, it has been shown (in [87]) that this may lead to uneconomical meshes as the rate of convergence of the error bound may be inferior to the rate of convergence of $J(u) - J(u_h)$.*

6.2.1 DWR for the Advection Reaction Problem

We consider again the advection reaction problem introduced in Section 3.1

$$\left.\begin{aligned} \mathcal{L}_a u = \boldsymbol{b} \cdot \nabla_x u + \Sigma_t u = f \qquad & \text{in } \Omega \\ u = g \qquad & \text{on } \Gamma^-, \end{aligned}\right\} \tag{6.8}$$

where Ω is the domain with smooth boundary Γ, $f \in L_2(\Omega)$, $\Sigma_t \in \mathbb{R}$ and the entries b_i of the vector $\boldsymbol{b}$ are also real constants. With the same triangulation and finite element space as before we have the discretisation: Find $u_{DG} \in S^{\boldsymbol{p}}(\Omega, \mathcal{T}, \boldsymbol{F})$ such that

$$A(u_{DG}, v_h) = l(v_h) \qquad \forall v_h \in S^{\boldsymbol{p}}(\Omega, \mathcal{T}, \boldsymbol{F}), \tag{6.9}$$

where the bilinear form and linear functional are as defined in Section 3.1.

In the DWR framework we consider two choices of functional $J(\cdot)$ for the above problem.

- **Mean Value:** For a weight function $\omega \in L_2(\Omega)$ we obtain the weighted mean value is given by
$$J(u) = \int_\Omega \omega u \, \mathrm{d}\boldsymbol{x}$$
In this case the dual solution z is the solution of the following dual problem: find z such that
$$\begin{aligned} -\boldsymbol{b}\nabla_{\boldsymbol{x}} z + \Sigma_t z &= \omega \quad \text{in } \Omega \\ z &= 0 \quad \text{on } \Gamma^+ \end{aligned}$$
The well posedness of this dual problem is established in [85].

- **Point value:** Assuming that u is a continuous function in the neighbourhood of some point $\boldsymbol{x}_0 \in \Omega$ we can consider the evaluation of the point value
$$J(u) = u(\boldsymbol{x}_0)$$
with corresponding dual problem: find z such that
$$\begin{aligned} -\boldsymbol{b}\nabla_{\boldsymbol{x}} z + \Sigma_t z &= \delta_{\boldsymbol{x}_0} \quad \text{in } \Omega \\ z &= 0 \quad \text{on } \Gamma^+, \end{aligned}$$
where, formally, $\delta_{\boldsymbol{x}_0}$ is a δ distribution centred at the point $\boldsymbol{x}_0$.

Remark 22. *Note that the boundary conditions of the dual problem are defined on* Γ^+ *since the direction* $\boldsymbol{b}$ *is the reverse of that in the primal problem.*

We now show some numerical results where our goal is efficient computation of a mean value functional of the solution.

We consider the advection problem (6.8) where we prescribe the solution u to be $u = 10y\tanh(100(x - \frac{1}{2}))$ which possesses a steep change of sign along the line $x = \frac{1}{2}$ as shown in Figure 6.2. We let the vector $\boldsymbol{b} = 1$, $\Sigma_t = 1$ and f be defined appropriately. With weighting $\omega = 1$ the mean value $J(u) = 0$. For comparison, we first provide some uniform refinement results, see Table 6.1 With these established we can now compare adaptive h–refinement and adaptive hp–refinement with global refinement. The results for h–refinement and hp–refinement are in Tables 6.2 and 6.3 respectively. Comparing the two strategies we can see that both h and hp–refinement are preferable to uniform

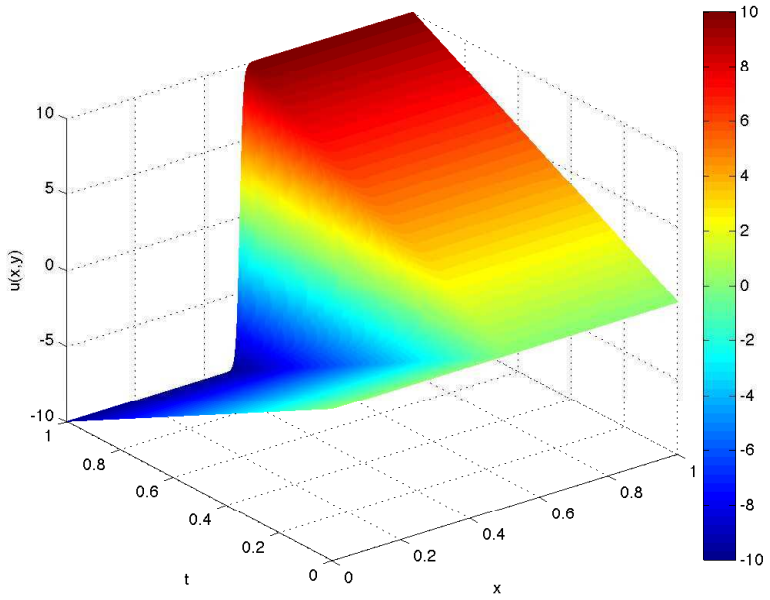


Figure 6.2: The solution $u = 10y\tanh(100(x - \frac{1}{2}))$

refinement in that they reduce the error by a greater degree for fewer degrees of freedom, as shown in Figure 6.3. In fact, it can be seen that the hp algorithm proceeds as the h algorithm until the last two meshes. We note here, that we would ideally have effectivities that are closer to one, starting the refinement algorithm on a finer initial mesh should achieve this. Some of the meshes generated by the h–refinement algorithm are shown in Figure 6.4, as expected, refinement is targeted along the line $x = \frac{1}{2}$.

Table 6.1: Uniform results for the advection problem with $u = 10y\tanh(100(x - \frac{1}{2}))$

Mesh No	Number of dofs	$J(u) - J(u_{DG})$	Error Estimate	Effectivity Index
1	16	2.460939E+00	1.246649E-01	0.05
2	64	2.185174E+00	9.966457E-01	0.46
3	256	6.299934E-01	9.014820E-01	1.43
4	1024	-2.853505E-01	-2.005073E-01	0.70
5	4096	-2.420000E-02	-3.160000E-02	1.31
6	16384	2.270000E-03	2.340000E-03	1.03
7	65536	9.120000E-06	9.550000E-06	1.05
8	262144	2.140000E-08	2.140000E-08	1.00

Table 6.2: Adaptive h–refinement results for the advection problem with $u = 10y\tanh(100(x - \frac{1}{2}))$

Mesh No	Number of dofs	$J(u) - J(u_{DG})$	Error Estimate	Effectivity Index
1	16	2.460939E+00	1.246649E-01	0.05
2	28	2.366181E+00	3.652978E-01	0.15
3	64	1.937479E+00	7.871037E-01	0.41
4	124	1.033200E+00	9.281184E-01	0.90
5	208	2.815161E-01	3.015645E-01	1.07
6	364	-1.267146E-01	-4.640000E-02	0.37
7	628	-5.290000E-02	-5.020000E-02	0.95
8	1096	-5.920000E-03	-8.190000E-03	1.38
9	1900	9.010000E-04	7.950000E-04	0.88
10	3496	5.100000E-04	5.240000E-04	1.03

Table 6.3: Adaptive hp–refinement results for the advection problem with $u = 10y\tanh(100(x-\frac{1}{2}))$

Mesh No	Number of dofs	$J(u)-J(u_{DG})$	Error Estimate	Effectivity Index
1	16	2.460939E+00	1.246649E-01	0.05
2	28	2.366181E+00	3.652978E-01	0.15
3	64	1.937479E+00	7.871037E-01	0.41
4	124	1.033200E+00	9.281184E-01	0.90
5	208	2.815161E-01	3.015645E-01	1.07
6	364	-1.267146E-01	-4.640000E-02	0.37
7	628	-5.290000E-02	-5.020000E-02	0.95
8	1096	-5.920000E-03	-8.190000E-03	1.38
9	1912	1.380000E-03	1.410000E-03	1.02
10	3700	3.900000E-05	3.990000E-05	1.02

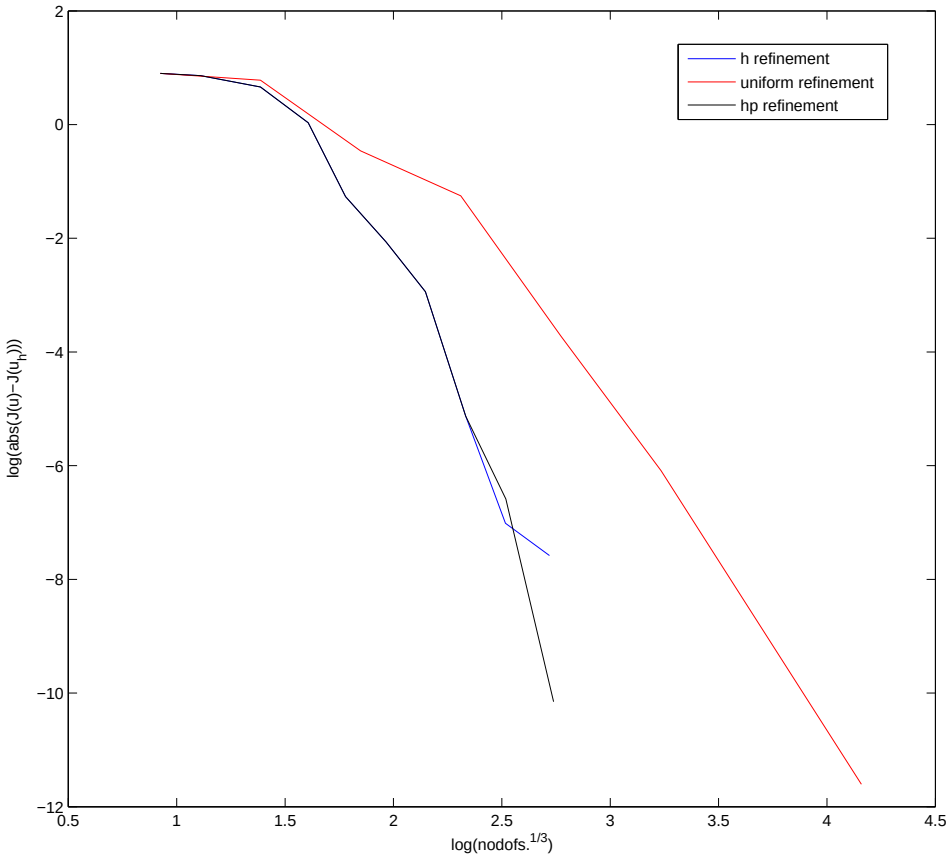


Figure 6.3: Comparison of refinement strategies for the advection problem.

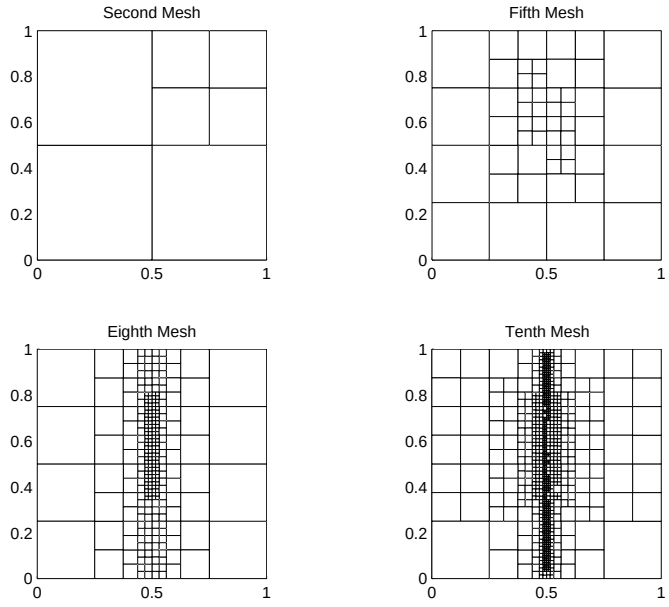


Figure 6.4: A sequence of adaptively refined meshes for the advection problem

6.3 DWR for the Two Dimensional Neutron Transport Source Problem

Like in Chapter 5 we are considering the solution of the two dimensional mono energetic steady state neutron transport equation (1.6) which for convenience we reproduce below.

$$\boldsymbol{\mu} \cdot \nabla_x \psi(\boldsymbol{x}, \varphi) + \Sigma_t \psi(\boldsymbol{x}, \varphi) = \frac{1}{2\pi} \int_0^{2\pi} (\Sigma_s + \nu\Sigma_f)\psi(\boldsymbol{x}, \varphi') \, \mathrm{d}\varphi' + Q(\boldsymbol{x}, \varphi) \quad \text{in } \Omega \times I \tag{6.10}$$

with boundary conditions determined by the analytic solution we prescribe. The following results are for the full DG discretisation, i.e., DG in space and angle.

As for the advection problem we can obtain an error representation formula with local error indicators $\eta_{\kappa_\Omega,\kappa_I}$

$$J(u) - J(u_h) = \sum_{\kappa \in \mathcal{T}_{\Omega,I}} \eta_{\kappa_\Omega,\kappa_I},$$

where $J(\cdot)$ is some linear functional and $\mathcal{T}_{\Omega,I} = \mathcal{T}_\Omega \times \mathcal{T}_I$. As we have the same spatial triangulation on each angular element, conceptually we obtain M elemental error indicators where M is the number of angular elements multiplied by the number of spatial elements. From these indicators, it remains to decide, which elements in each mesh require refining. We propose two possible approaches:

- **Method 1** We either mark $\eta_{\kappa_\Omega,\kappa_I}$ such that

  $$\eta_{\kappa_\Omega,\kappa_I} > \frac{\mathrm{Tol}}{M}$$

  for some user defined tolerance Tol or rank the $\eta_{\kappa_\Omega,\kappa_I}$ according to their size and select a percentage of the largest to mark for refinement. We then refine both κ_Ω and κ_I.

- **Method 2** We mark $\kappa_\Omega \in \mathcal{T}_\Omega$ such that

  $$\sum_{\kappa_I \in \mathcal{T}_I} \eta_{\kappa_\Omega,\kappa_I} > \frac{\mathrm{Tol}_\Omega}{N_\Omega}$$

  for some user defined tolerance Tol_Ω with N_Ω being the number of elements in the spatial mesh. Similarly, we mark the angular elements $\kappa_I \in \mathcal{T}_I$ such that

  $$\sum_{\kappa_\Omega \in \mathcal{T}_\Omega} \eta_{\kappa_\Omega,\kappa_I} > \frac{\mathrm{Tol}_I}{N_I}$$

  for some tolerance Tol_I and N_I being the number of angular elements. Alternatively, we could perform the summation over spatial and angular elements and

then apply the fixed fraction approach to the two vectors of indicators that we obtain.

We find that in practice, the second method performs better; method one introduces too much refinement. We also use the fixed fraction procedure for the results presented here.

Remark 23. *Another possiblity would be to proceed as is often done for space–time problems and define projection operators from $L_2(\Omega)$ into the finite element space associated with the angular domain and into the finite element space associated with the spatial domain. We can then consider a splitting of the error indicator, and use this splitting to guide refinement.*

Now we have established a method to decide where to perform refinement we present some numerical experiments.

6.3.1 Adaptive Test Problem 1

We consider applying the adaptive strategy discussed above to the two dimensional mono energetic steady state neutron transport equation (6.10). Here we prescribe the smooth solution $u = \exp(x - y)\theta$ and our goal is to compute the weighted mean value

$$\int_\Omega \int_I \omega\psi \, \mathrm{d}\varphi \, \mathrm{d}\boldsymbol{x},$$

where the weighting, ω, is a steep Gaussian in the spatial domain centred around $(0.3, 0.3)$. in particular $\omega = \exp(-50(x - 0.3)^2 - 50(y - 0.3)^2)$ which leads to an analytical value $J(u) = 1.2491685936435003162$. Defining the source $Q(\boldsymbol{x}, \varphi)$ appropriately and using the material cross sections used for the source problems in the previous chapter our problem is well posed.

We perform h–refinement using method 2 and polynomial degree $p = 0$. We set the refinement and derefinement percentages for the spatial mesh to be 25% and 10% respectively. For the angular mesh, 25% of elements will be selected for refinement and we do not permit derefinement. The numerical results are shown in Table 6.4. The sequence of 12 meshes that are obtained by this method are given in Figures 6.5 - 6.7. The figures show refinement around the centre of the Gaussian, however there appears to be a fair amount of pollution. Despite this pollution, the adaptive algorithm compares well with global refinement as shown in Figure 6.8. Examining Figure 6.8 in more detail we observe that at first the adaptive algorithm performs a lot better than uniform refinement, however, the rate at which the error in the functional decays soon becomes broadly similar to that when uniform refinement is applied. This is in contrast to the

Table 6.4: h–adaptive results for Adaptive Test Problem 1 with $p = q = 0$

No Angles	No spatial dofs	No total dofs	$J(\psi) - J(\psi_h)$	Error Estimate	Effectivity
8	64	512	-4.564190E-02	-4.550123E-02	1.00
10	112	1120	-2.258667E-02	-2.250947E-02	1.00
12	190	2280	-1.539253E-02	-1.537331E-02	1.00
15	316	4740	-1.412665E-02	-1.412109E-02	1.00
18	538	9684	-9.699873E-03	-9.696472E-03	1.00
22	904	19888	-7.718203E-03	-7.715121E-03	1.00
27	1549	41823	-6.532702E-03	-6.531553E-03	1.00
33	2614	86262	-4.782968E-03	-4.783006E-03	1.00
41	4462	182942	-3.779713E-03	-3.779587E-03	1.00
51	7543	384693	-2.830361E-03	-2.830405E-03	1.00
63	12784	805392	-2.332112E-03	-2.332131E-03	1.00
78	22141	1726998	-1.715023E-03	-1.715019E-03	1.00

results shown in Figure 6.3 for the advection problem. Looking at the meshes shown in Figures 6.5 - 6.7 it can be seen that after the initial targeting of refinement around the point $(0.3, 0.3)$ as you would expect we then see broadly uniform refinement of the mesh with the refinement a few steps behind what you would see if uniform refinement had been employed.

This pollution and probable over refinement requires more investigation. The refinement and derefinement percentages have been varied and you observe the same affect regardless of what these are. This leads us to believe that the differing advective direction on each angular element is leading to an over refinement. Separate spatial meshes for each angular element, whilst incurring additional storage would prevent this from happening.

Despite the over refinement the error indicator performs well, accurately predicting the error made in the numerical approximation of the functional of interest. This leads to effectivity indices of 1 as shown in Table 6.4.

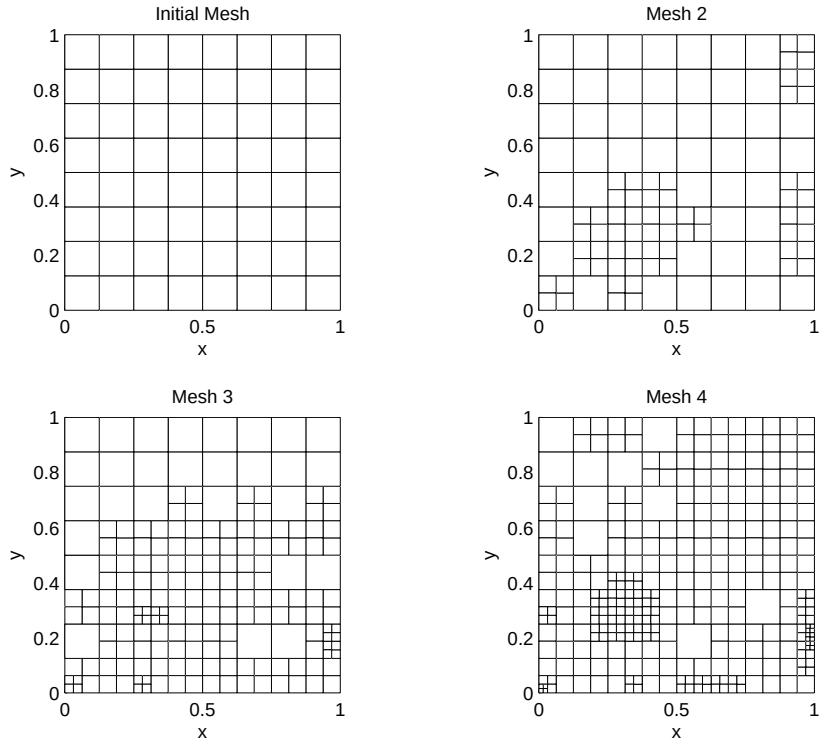


Figure 6.5: The first 4 spatial meshes from adaptive refinement for Adaptive Test Problem 1

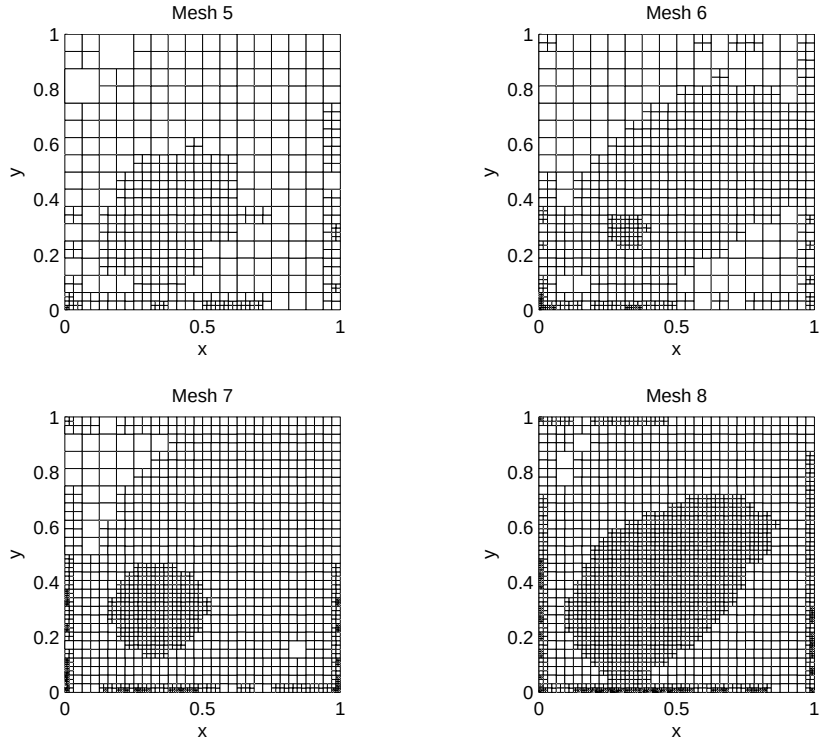


Figure 6.6: Meshes 8 - 12 from adaptive refinement for Adaptive Test Problem 1

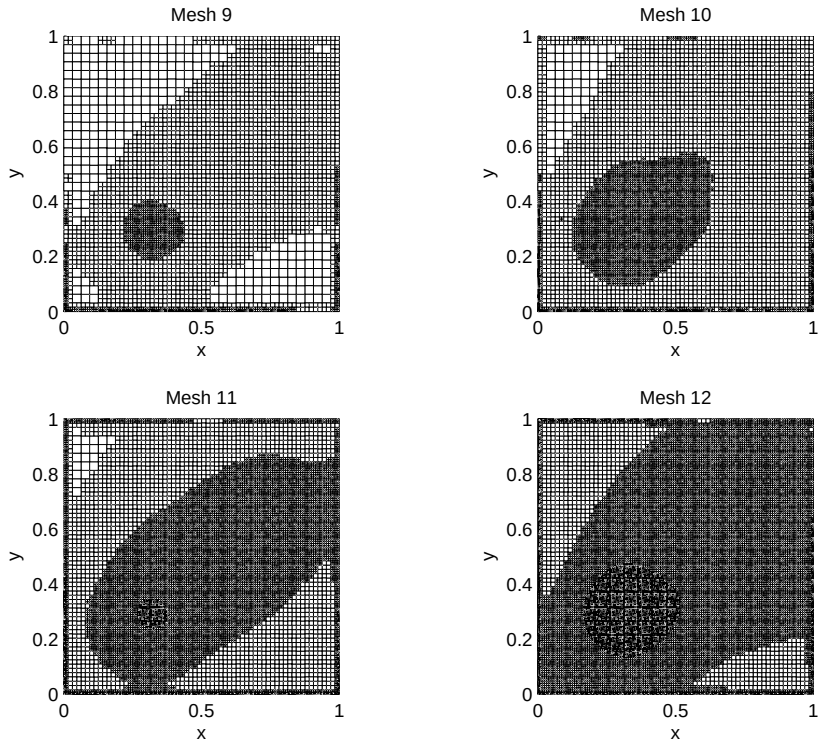


Figure 6.7: The final 4 meshes from adaptive refinement for Adaptive Test Problem 1

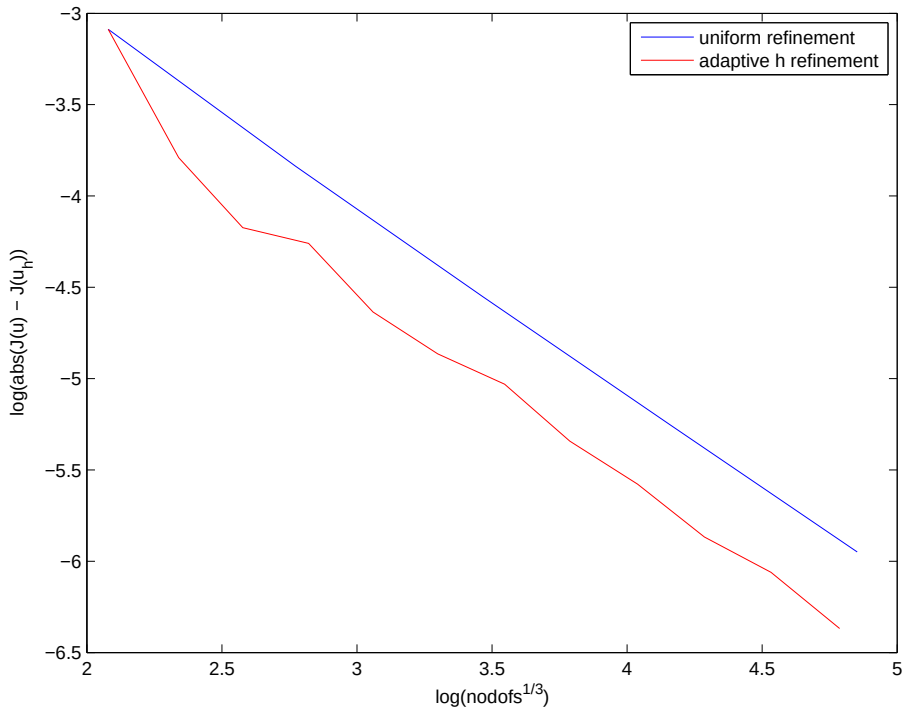


Figure 6.8: Comparison of adaptive and uniform refinement for Adaptive Test Problem 1.

Table 6.5: h–adaptive results for Adaptive Test Problem 2 with $p = q = 0$

No Angles	No spatial dofs	No total dofs	$J(\psi) - J(\psi_h)$	Error Estimate	Effectivity
8	64	512	1.316075E+00	-7.747917E-01	-0.59
10	112	1120	-1.187983E-01	-6.056625E-01	5.10
12	196	2352	-7.098559E-01	-7.890739E-02	0.11
15	313	4695	-1.437588E+00	-8.170352E-01	0.57
18	526	9468	-7.454808E-01	-2.726251E-01	0.37
22	913	20086	-1.434524E-01	-1.148321E-01	0.80
27	1522	41094	-8.483920E-02	-8.828744E-02	1.04
33	2554	84282	-5.203825E-02	-5.110238E-02	0.98
41	4378	179498	-4.295347E-02	-4.314575E-02	1.00
51	7630	389130	-3.657667E-02	-3.658347E-02	1.00
63	12712	800856	-2.939368E-02	-2.940166E-02	1.00
78	21661	1689558	-1.867230E-02	-1.867768E-02	1.00

6.3.2 Adaptive Test Problem 2

Wishing to investigate the effectiveness of our adaptive algorithm in the case of solution which has the presence of sharp gradients we investigate another test problem,

Here, we prescribe that the angular flux ψ is given by

$$\psi(\boldsymbol{x}, \varphi) = 5(1 + \varphi) \tanh\left(100\left(x - \frac{1}{2}\right)\right) \tanh\left(100\left(y - \frac{1}{2}\right)\right)$$

with scalar flux

$$\begin{aligned}\phi(\boldsymbol{x}) = 10\pi \tanh\left(100\left(x - \frac{1}{2}\right)\right) \tanh\left(100\left(y - \frac{1}{2}\right)\right) \\ + 10\pi^2 \tanh\left(100\left(x - \frac{1}{2}\right)\right) \tanh\left(100\left(y - \frac{1}{2}\right)\right).\end{aligned}$$

The source term $Q(\boldsymbol{x}, \varphi)$ is defined appropriately, and we use the same cross sections as in Chapter 5. Using a mean value functional with weighting $\omega = 1$, that is our functional $J(u)$ is of the form,

$$J(\psi) = \int_\Omega \int_I \psi(\boldsymbol{x}, \varphi) \, \mathrm{d}\varphi \mathrm{d}\boldsymbol{x},$$

with analytical value $J(\psi) = 0$. The results are shown in Table 6.5 As can be seen in the table we obtain effectivity indices of 1 on the last four meshes. It is interesting to note that the sign of the error estimator doesn't reflect the sign of the error made in the

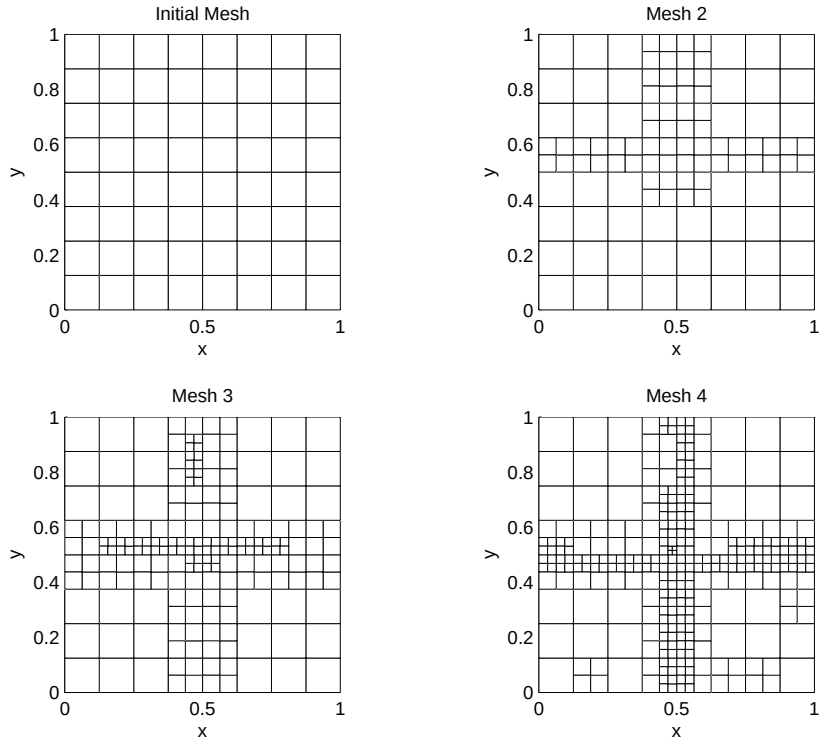


Figure 6.9: The first 4 spatial meshes from adaptive refinement for Adaptive Test Problem 2

approximation of the functional. With a weighting of $\omega = 1$ for the mean value we would expect to see the algorithm refine around the features of the domain, which in this case are the steep gradients along the lines $x = 1/2$ and $y = 1/2$. The sequence of meshes obtained is shown in Figures 6.9 - 6.11 These plots show that the adaptive algorithm has guided refinement in the regions of sharp gradient as expected, however there seems to be a large amount of refinement elsewhere; we believe this is dues to the percentages (25%) of elements selected for refinement and the mesh smoothers that have been used. As the algorithm guides refinement of the angular mesh too, it is instructive to visualise the angular meshes generated. For the two dimensional problem these are just partitions of the interval $[0, 2\pi]$ and so we visualise all angular meshes on one plot, Figure 6.12. We see that the refinement of the angular mesh is broadly uniform apart form slightly more refinement towards the 2π end. This behaviour is potentially due to the fact that the values of the angular component of ψ at 0 and 2π are not equal. It may, however, be solely an artefact of the algorithm, 25% of angular elements will be refined at each step - even if the indicators on all elements are very close to each other - the 25% that are fractionally larger will be selected for refinement. As $1 + \varphi$ is smooth, we wouldn't expect much else than uniform refinement.

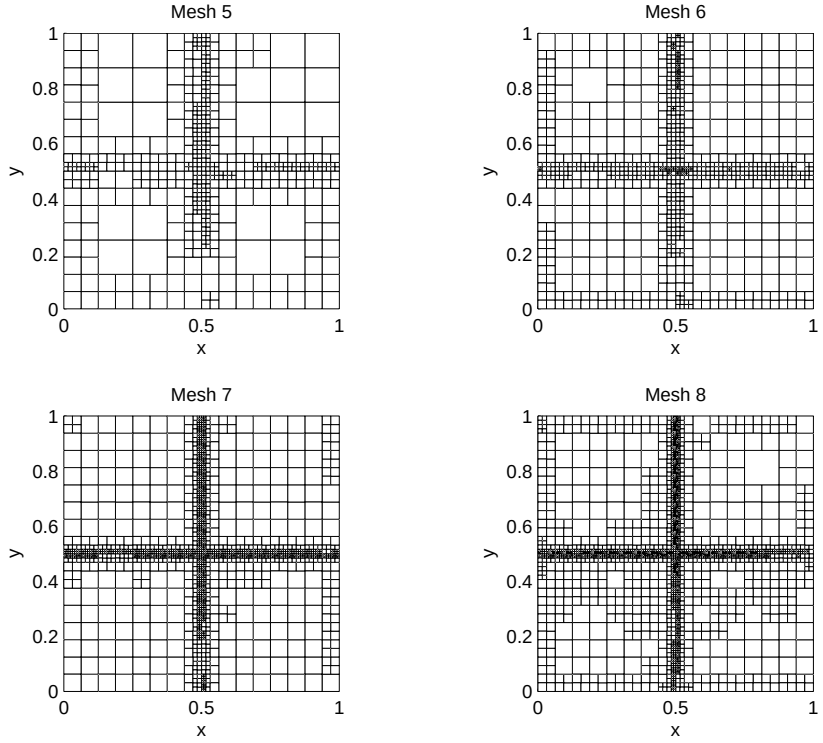


Figure 6.10: Meshes 8 - 12 from adaptive refinement for Adaptive Test Problem 2

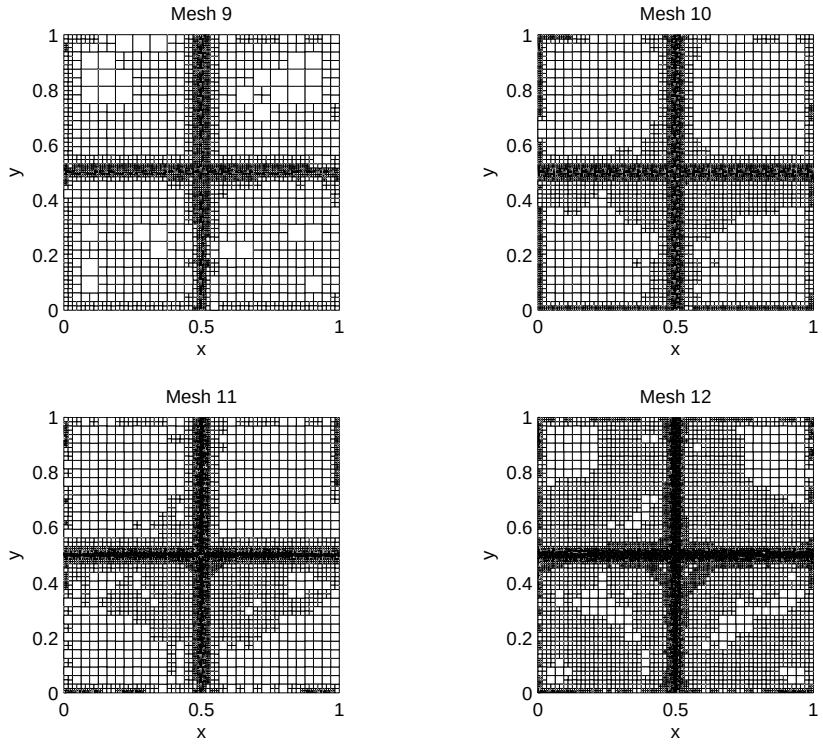


Figure 6.11: The final 4 meshes from adaptive refinement for Adaptive Test Problem 2

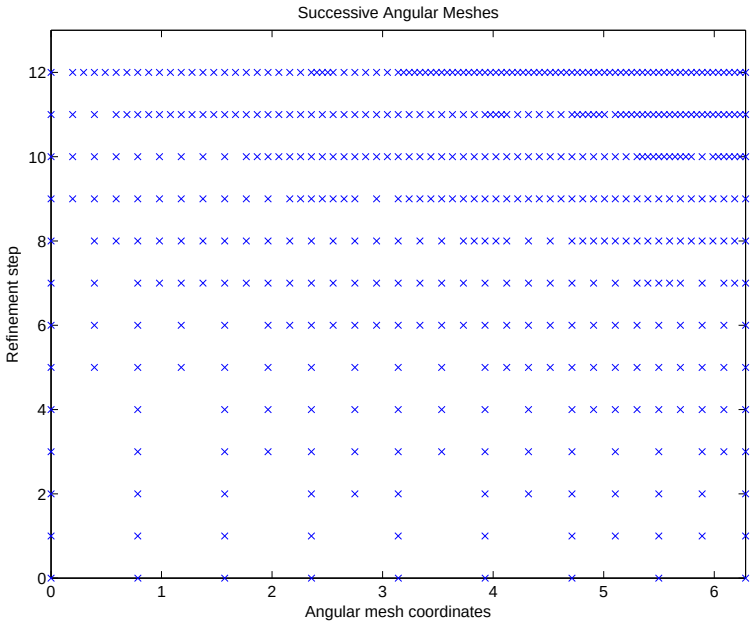


Figure 6.12: Pictorial representation of the angular mesh refinement for Test Problem 2

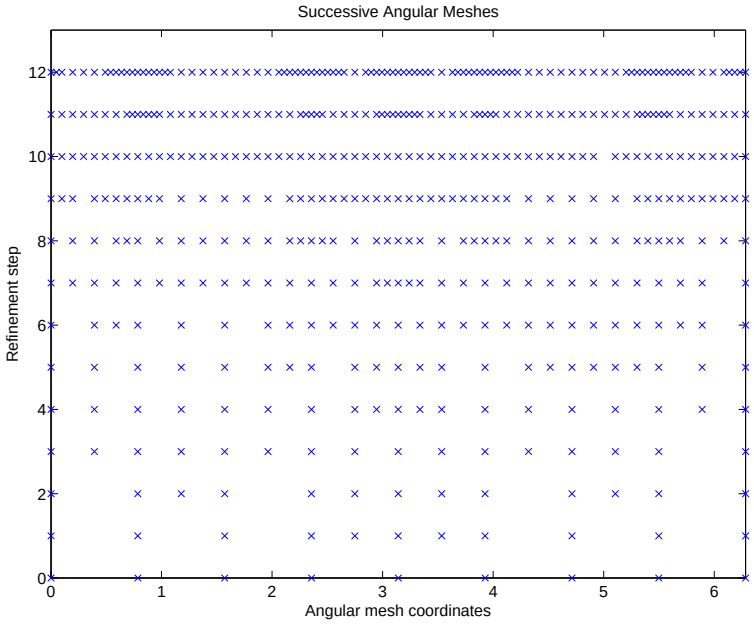


Figure 6.13: Pictorial representation of the angular mesh refinement for Test Problem 3

Adaptive Test Problem 3

Both the previous test problems have had smooth angular component; this has led to the adaptive algorithm refining, more or less, uniformly in the angular domain. To investigate how the algorithm deals with situations when this is not the case, we consider prescribing the following solution,

$$\psi = 5\tanh(10(\varphi - \pi)\tanh\left(100\left(x - \frac{1}{2}\right)\right)\tanh\left(100\left(y - \frac{1}{2}\right)\right)$$

In addition to the sharp gradients in the solution's spatial components we also now have a sharp gradient centered around $\varphi = \pi$ in the angular variable. Because of this sharp gradient we would hope to see increased refinement in the centre of the angular domain. Keeping everything else the same as in the previous test problem, we obtain the sequence of angular meshes shown in Figure 6.13 Comparing the two angular mesh plots from adaptive test problem 2 and adaptive test problem 3 it can be seen that mesh refinement is focussed in the middle of the domain for problem 3 as we expected. The spatial meshes should be refined in the same fashion as the last problem, indeed we see this to be the case; the last four spatial meshes are plotted in Figure 6.14. For completeness the numerical results for this problem are given in Table 6.6.

Table 6.6: h–adaptive results for Adaptive Test Problem 3 with $p = q = 0$

No Angles	No spatial dofs	No total dofs	$J(\psi) - J(\psi_h)$	Error Estimate	Effectivity
8	64	512	2.139047E-02	7.518609E-03	0.35
10	112	1120	8.783555E-01	3.922821E-02	0.04
12	190	2280	4.752898E-02	4.562332E-02	0.96
15	304	4560	2.947588E-01	4.413628E-02	0.15
18	493	8874	-1.383348E-01	-1.937770E-02	0.14
22	832	18304	-9.260863E-03	3.371327E-03	-0.36
27	1447	39069	6.203725E-03	5.714888E-03	0.92
33	2446	80718	3.027399E-05	-2.336395E-04	-7.72
41	4207	172487	5.266135E-03	5.278894E-03	1.00
51	7360	375360	2.216210E-03	2.215768E-03	1.00
63	12403	781389	2.243352E-03	2.243001E-03	1.00
78	21016	1639248	5.547802E-04	5.555121E-04	1.00

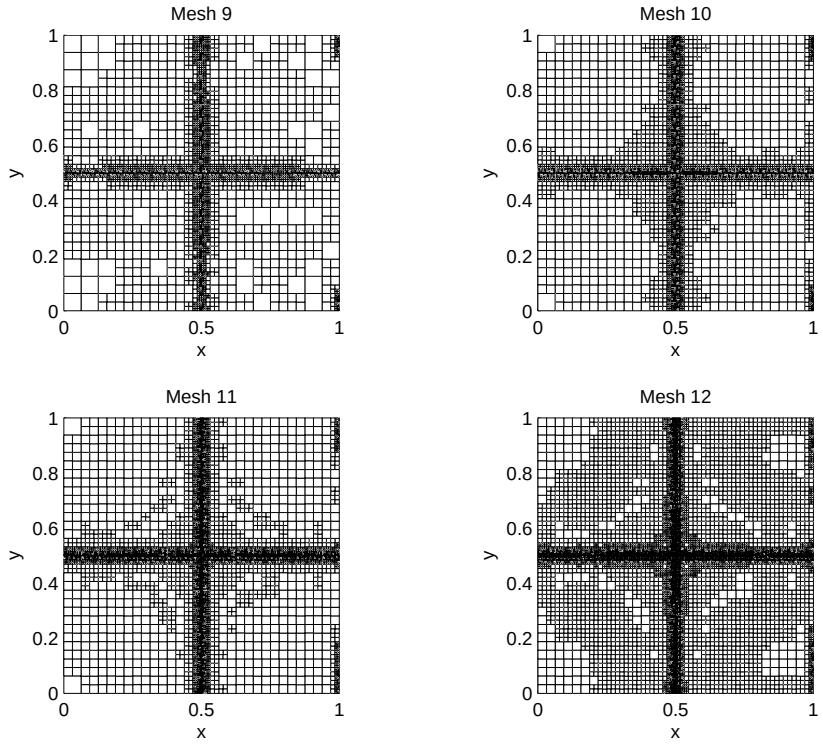


Figure 6.14: The final 4 meshes from adaptive refinement for Adaptive Test Problem 2

Table 6.7: h–adaptive results for Pseudo 3D Adaptive Test Problem 1 with $p = q = 0$

No Angles	No spatial dofs	No total dofs	$J(\psi) - J(\psi_h)$	Error Estimate	Effectivity
16	64	1024	1.237186E-02	7.817863E-03	0.63
28	112	3136	1.020056E-02	6.197600E-03	0.61
43	196	8428	7.284467E-03	4.668876E-03	0.64
73	331	24163	5.422937E-03	3.684849E-03	0.68
127	550	69850	4.197638E-03	2.899793E-03	0.69
229	898	205642	3.317066E-03	2.288969E-03	0.69
400	1456	582400	2.583040E-03	1.790475E-03	0.69

6.4 DWR for the Pseudo 3D Neutron Transport Source Problem

The approach considered previously for the two dimensional neutron transport equation extends directly to the pseudo 3D discretisation also.

Consider prescribing the following problem: find ψ such that

$$\begin{aligned}\boldsymbol{\mu} \cdot \nabla_x \psi(\boldsymbol{x}, \boldsymbol{\mu}) + \sigma_t \psi(\boldsymbol{x}, \boldsymbol{\mu}) = \frac{1}{4\pi} \int_D (\sigma_s + \nu\sigma_f) \psi(\boldsymbol{x}, \boldsymbol{\mu}') \left(1 - |\boldsymbol{\mu}'|^2\right)^{-1/2} \, \mathrm{d}\boldsymbol{\mu}' \\ + Q(\boldsymbol{x}, \boldsymbol{\mu}) \quad \text{in } \Omega \times D, \\ \psi(\boldsymbol{x}, \boldsymbol{\mu}) = g(\boldsymbol{x}, \boldsymbol{\mu}) \quad \text{on } \Gamma_{\boldsymbol{\mu}}^-.\end{aligned}$$

Prescribing the solution

$$\psi = xy \sin(\pi x) \sin(\pi y) \sin(\varphi) \cos(\theta) \tag{6.11}$$

and defining $Q(\boldsymbol{x}, \boldsymbol{\mu})$ appropriately we can apply the DWR procedure to this problem. The numerical results we obtain are given in Table 6.7 The sequence of spatial meshes obtained are shown in Figures 6.15 and 6.16. The corresponding angular meshes are shown in Figures 6.17 and 6.18 Note that the code understands the faces of the outer angular elements coincident with the boundary of the unit disc to be curved, due to the implementation of curved boundaries.

As the spatial component of ψ is the same as that considered in Test Case 1 of Section 5.2.1 we would expect refinement to occur in the top right corner as this is where the solution rises to a peak, indeed we do observe this Figure 6.16. The effectivites are not as close to one as they were for the two dimensional problems; in fact they seem

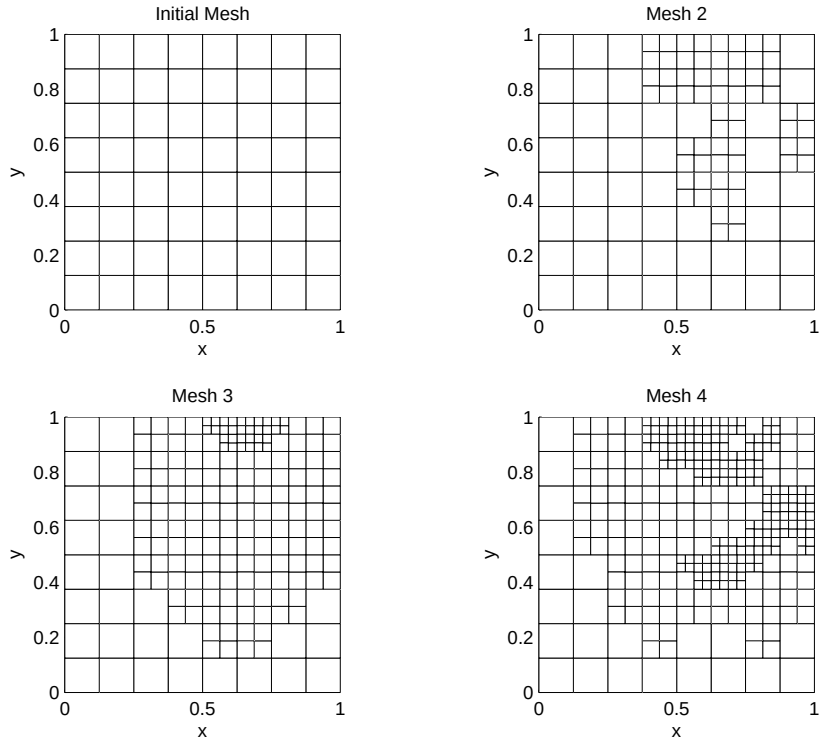


Figure 6.15: The first 4 spatial meshes from adaptive refinement for Pseudo 3D Adaptive Test Problem 1

to have settled at 0.69. This could be due to the difficulty we have experienced performing the quadrature in the angular domain and also the fact that when refining the angular mesh we lose the graded structure which we have observed to be important for convergence. Also, the sub optimal convergence of the angular flux may play a role.

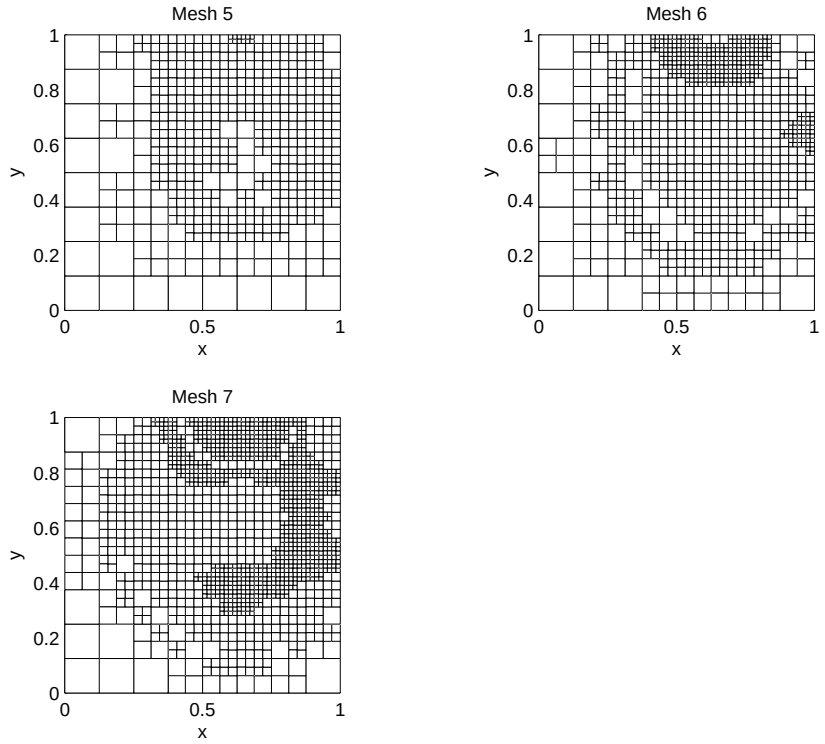


Figure 6.16: The final 3 spatial meshes from adaptive refinement for Pseudo 3D Adaptive Test Problem 1

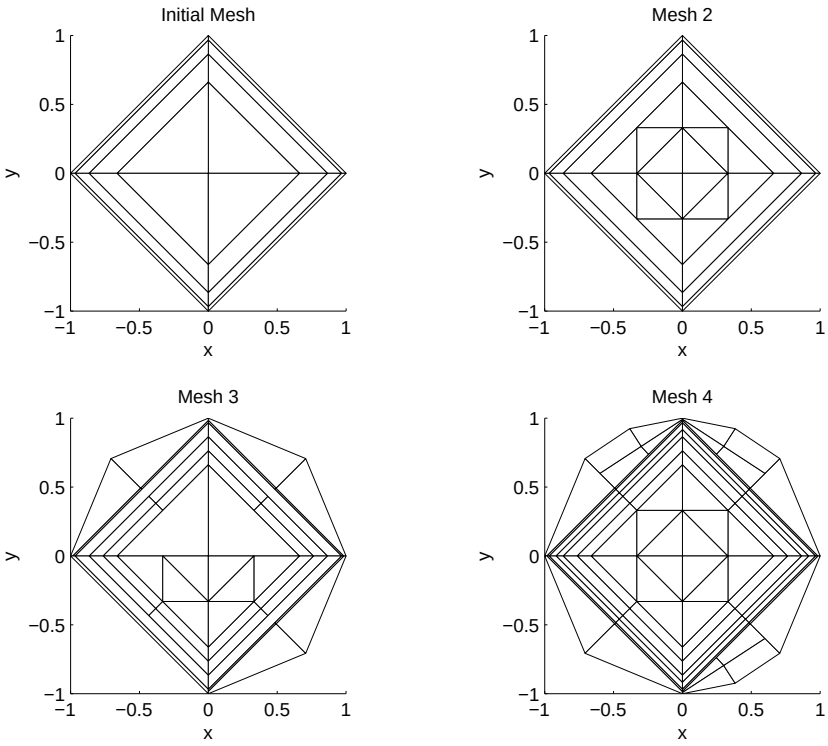


Figure 6.17: The first 4 angular meshes from adaptive refinement for Pseudo 3D Adaptive Test Problem 1

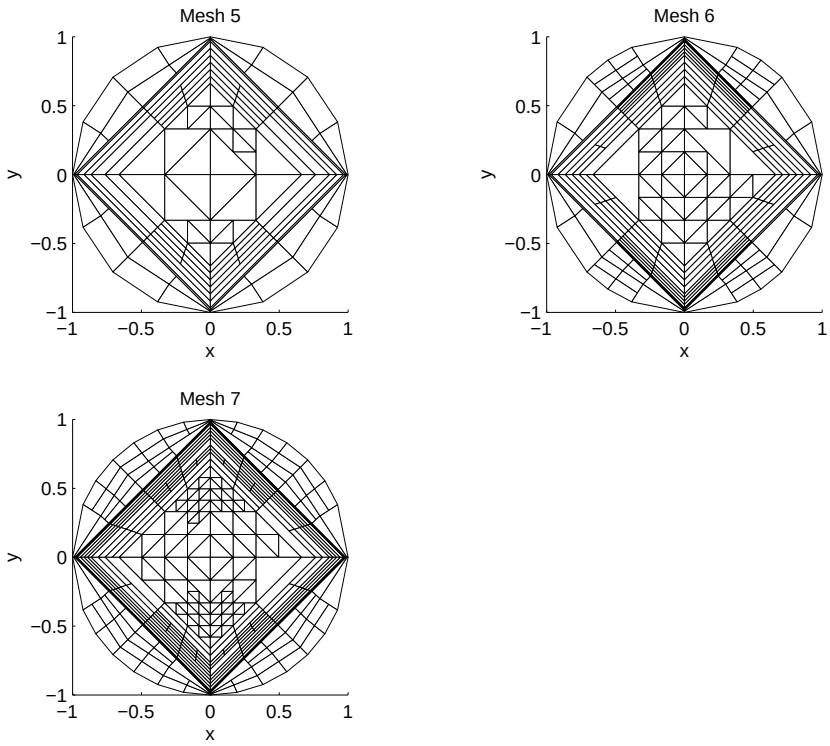


Figure 6.18: The final 3 angular meshes from adaptive refinement for Pseudo 3D Adaptive Test Problem 1

6.5 DWR for Eigenvalue Problems

As discussed in [29, 50, 80] we can also apply the dual weighted residual approach to functionals arising from a variational discretisation of an eigenvalue problem.

We consider the following generalised eigenvalue problem: find $(u_P, \lambda_P) \in V \times \mathbb{R}$ such that

$$Au_P = \lambda_P B u_P$$

with the corresponding variational form: find (u_P, λ_P)

$$A(u_P, v) = \lambda_p B(u_P, v) \tag{6.12}$$

for some operators A and B. If we require that the eigenfunction u_P is normalised so that for some nonlinear operator C, $C(u) = 1$ then we can rewrite (6.12) in the following way: find $\boldsymbol{u}_P = (u_P, \lambda_P) \in V \times \mathbb{R}$ such that

$$N(\boldsymbol{u}, \boldsymbol{v}) = 0 \quad \forall \boldsymbol{v} = (v, \chi) \in V \times \mathbb{R} \tag{6.13}$$

with

$$N(\boldsymbol{u}, \boldsymbol{v}) = -A(u, v) + \lambda_p B(u, v) + \chi(C(u) - 1).$$

Performing a discontinuous Galerkin discretisation we obtain the discretised form of the above: find $\boldsymbol{u}_{h,P} = (bu_{h,P}, \lambda_{h,P})$ such that

$$N(\boldsymbol{u}_{h,P}, \boldsymbol{v}_h) = 0 \quad \forall \boldsymbol{v}_h = (v_h, \chi_h) \in V \times \mathbb{R} \tag{6.14}$$

As before, suppose that we wish to compute the error $J(\boldsymbol{u}_P) - J(\boldsymbol{u}_{h,P})$ in approximating some functional $J(\cdot)$ of the eigenfunction and eigenvalue. Let the functional be defined to be $J(\boldsymbol{u}) = \lambda_P C(u_P)$, then the approximation error in the functional is:

$$J(\boldsymbol{u}_P) - J(\boldsymbol{u}_{h,P}) = \lambda_P C(u_P) - \lambda_{h,P} C(u_{h,P}) = \lambda_P - \lambda_{h,P},$$

where the eigenfunction and approximate eigenfunction have been normalised so that $C(u_P) = C(u_{h,P}) = 1$. Hence, the functional gives us a measure of how closely the approximate eigenvalue estimates the true eigenvalue. Now, computing the Fréchet derivative of $N(\cdot, \boldsymbol{z})$ for $\boldsymbol{z} = (z, \lambda_z)$ in the direction $\boldsymbol{v}$ evaluated at $\boldsymbol{u}$ we obtain

$$N'[\boldsymbol{u}](\boldsymbol{v}, \boldsymbol{z}) = -A(v, z) + \chi B(u, z) + \lambda_p B(v, z) + \lambda_z C'[u](v).$$

Similarly we can compute the Fréchet derivative of the functional $J(\cdot)$,

$$J'[\boldsymbol{u}](\boldsymbol{v}) = \chi C(u) + \lambda_p C'[\boldsymbol{u}](\boldsymbol{v}).$$

With the above we can now state the associated dual problem: Find $\boldsymbol{z}_D = (z_D, \lambda_D)$ such that

$$N'[\boldsymbol{u}](\boldsymbol{v}, \boldsymbol{z}_D) = J'[\boldsymbol{u}](\boldsymbol{v}) \quad \forall \boldsymbol{v} \in \mathbb{R} \times V_h$$

That is,

$$-A(v, z_D) + \chi B(u, z_D) + \lambda_P B(v, z_D) + \lambda_D C'[\boldsymbol{u}](v) = \chi C(u) + \lambda_P C'[\boldsymbol{u}](\boldsymbol{v}) \quad \forall \boldsymbol{v} \in \mathbb{R} \times V, \tag{6.15}$$

with the obvious discretised equivalent. Notice, however, that if instead we find $\boldsymbol{z}_D$ such that

$$A(v, z_D) = \lambda_D B(v, z_D) \quad \forall v \in V$$

where we have normalised z_D such that $B(u, z_D) = C(u)$, then, since $\lambda_P = \lambda_D$ (where we have ensured that the dual and primal eigenvalues correspond with each other), $\boldsymbol{z}_D = (z_D, \lambda_D)$ also solves (6.15).

With this dual solution we can obtain an error estimator of the same form as before, namely,

$$|J(u) - J(u_h)| \leq \sum_{\kappa \in \mathcal{T}_h} |\eta_k| = \sum_{\kappa \in \mathcal{T}_h} \mathcal{R}(u_{h,p}, z - z_h)|_\kappa \tag{6.16}$$

Remark 24. *It is possible to solve* (6.15) *directly by considering an augmented system, this is the approach taken in [80] for example. Due to the setting of the neutron transport problem, we wish to compute the dual solution as an eigenvalue problem in the same way as the primal eigenvalue problem.*

6.5.1 Adaptivity For The Neutron Transport Eigenvalue Problem

Following the above success of dual weighted residual error estimation for the Laplace eigenvalue problem, we return to the underlying critical eigenvalue problem of the neutron transport equation. The numerical results from Chapter 5 show that it can be difficult, in practice, to obtain good approximations to the critical eigenvalue, despite requiring only the first couple of decimal digits to be accurate. The convergence of the pseudo 3D eigenvalue code was seen to be fairly slow, with the initial approximations far from the true eigenvalues, this was primarily due to the difficulty of using quadrature to resolve the singularity on the boundary of the angular mesh. A code with adaptive mesh refinement in both the spatial and angular mesh should go some way to mitigating this problem. With this in mind, we return to the two dimensional steady state eigenvalue problem considered previously.

Recall that we have the following eigenvalue problem to solve: find $\boldsymbol{\psi} = (\psi, k_{eff}) \in V \times \mathbb{R}$ such that

$$(\tilde{T} - \tilde{S})(\psi, v) = \frac{1}{k_{eff}} \tilde{F}(\psi, v) \tag{6.17}$$

where the transport, fission and scattering operators are defined to be

$$\tilde{T}(\psi, v) = \int_\Omega \int_I (\boldsymbol{\mu} \cdot \nabla_x \psi + \Sigma_t \psi)\, v \, \mathrm{d}\varphi \mathrm{d}\boldsymbol{x}, \tag{6.18}$$

$$\tilde{F}(\psi, v) = \int_\Omega \int_I \left(\int_I \frac{\nu \Sigma_f}{2\pi} \psi(\boldsymbol{x}, \varphi')\, \mathrm{d}\varphi' \right) v \, \mathrm{d}\varphi \, \mathrm{d}\boldsymbol{x}, \tag{6.19}$$

$$\tilde{S}(\psi, v) = \int_\Omega \int_I \left(\int_I \frac{\Sigma_s}{2\pi} \psi(\boldsymbol{x}, \varphi')\, \mathrm{d}\varphi' \right) v \, \mathrm{d}\varphi \, \mathrm{d}\boldsymbol{x}. \tag{6.20}$$

Problem (6.17) is of the form for a generalised eigenvalue problem (6.12) with,

$$A = (\tilde{T} - \tilde{S})(\cdot, v),$$

and

$$B = \tilde{F}(\cdot, v).$$

We can use the same approach as for the Laplacian operator to derive an error representation formula for the neutron transport equation and use this to guide mesh refinement. As before, we normalise the eigenfunction ψ so that for some operator $C(\psi) = 1$; in practice we require that $(\psi, \psi) = 1$, i.e., the L_2–norm squared of ψ is equal to one. Defining, in this case,

$$N(\boldsymbol{\psi}, \boldsymbol{v}) = -(\tilde{T} - \tilde{S})(\psi, v) + \frac{1}{k_{eff}} \tilde{F}(\psi, v) + \chi(C(u) - 1),$$

the neutron transport eigenvalue problem is recast to be: find $\boldsymbol{\psi} = (\psi, k_{eff}) \in V \times \mathbb{R}$ such that

$$N(\boldsymbol{\psi}, \boldsymbol{v}) = 0 \quad \forall \boldsymbol{v} = (v, \chi) \in V \times \mathbb{R}$$

With our linear functional $J(\cdot)$ defined in the same way as before, namely $J(\boldsymbol{\psi}) = \frac{1}{k_{eff}} C(\psi)$ we obtain, on taking Fréchet derivatives as previously, the associated dual eigenvalue problem: find $\boldsymbol{z} = (z, \lambda_D)$ such that

$$-(\tilde{T} - \tilde{S})(v, z) + \chi \tilde{F}(\psi, z) + \frac{1}{k_{eff}} \tilde{F}(v, z) + \lambda_D C'[\boldsymbol{\psi}](v) = \chi C(\psi) + \frac{1}{k_{eff}} C'[\boldsymbol{\psi}](v)$$

for all $\boldsymbol{v} \in \mathbb{R} \times V$. Normalising the dual eigenfunction such that $\tilde{F}(\psi, z) = 1$, we again obtain a dual eigenvalue problem of the same form as the primal eigenvalue problem: find $\boldsymbol{z} = (z, \lambda_z)$ such that

$$(\tilde{T} - \tilde{F})(v, z) = \lambda_D \tilde{F}(v, z) \quad \forall v \in V$$

For the error representation formula to accurately represent the true error it is crucial that we normalise the dual eigenfunction as above, which in the full form can be written as,

$$\begin{aligned} 1 &= \tilde{F}(\psi, z) \\ &= \int_\Omega \int_I \left(\int_I \frac{\nu \Sigma_f}{2\pi} \psi(\boldsymbol{x}, \varphi') \, \mathrm{d}\varphi' \right) z \, \mathrm{d}\varphi \, \mathrm{d}\boldsymbol{x} \end{aligned}$$

As discussed previously, in practice we perform a sequence of manipulations to reformulate (6.17) as the standard eigenvalue problem: find $\boldsymbol{\psi} = (\psi, k_{eff})$ such that

$$(I - \tilde{T}^{-1}\tilde{S})^{-1}\tilde{T}^{-1}\tilde{F}(\psi, v) = k_{\text{eff}}(\psi, v). \tag{6.21}$$

To find the largest eigenvalue, k_{eff} we employ the power method where we apply a Neumann sum to approximate the operator $(I - \tilde{T}^{-1}\tilde{S})^{-1}\tilde{T}^{-1}\tilde{F}$. This Neumann sum requires the solution of many linear systems where the right hand side is obtained by applying either the fission or scattering operator to the solution of the previous linear system. We typically find we need to use ~ 30 terms in the Neumann sum, and so each application of the operator requires the solution of 30 linear systems.

When solving the dual eigenvalue problem the eigenfunction that we seek is in the second argument of the continuous bilinear forms $(\tilde{T}(v, z) - \tilde{F})$ and $\tilde{F}$. To investigate the difference this makes to our practical solution procedure it is useful to consider how the scattering operator, for instance, acts over the whole tensor product space–angle domain. It helps to consider a concrete example in this setting; we consider a discretisation with three angular elements, a polynomial approximation order in angle of $q = 1$ and a spatial mesh containing four elements with approximation degree in space of $p = 0$. This means, that on each angular element we have 8 degrees of freedom and 24 degrees of freedom for the entire space–angle domain.

Remark 25. *We note here that we are assuming the use of one spatial mesh for all angular elements; the situation is more complex if we allow a different spatial mesh for each angular element.*

We express the full numerical solution as

$$\psi_h = \sum_{\kappa_I=1}^{3} \sum_{j=1}^{4} \sum_{l=1}^{2} \Psi_{\kappa_I, j, l} \zeta_j(\boldsymbol{x}) \xi_l(\varphi),$$

with coefficients $\Psi_{\kappa_I, j, l}$ where it is understood that that on κ_I the support of the angular basis functions ξ is only κ_I. Forming the vector $\boldsymbol{U}$ from the coefficients appearing in

the expansion we can write,

$$\begin{aligned}\boldsymbol{U} &= \Big(\boldsymbol{U}_1, \boldsymbol{U}_2, \boldsymbol{U}_3\Big) \\ &= \Big(\Psi_{1,1,1}, \Psi_{1,1,2}, \Psi_{1,1,3}, \Psi_{1,1,4}, \Psi_{1,2,1}, \Psi_{1,2,2}, \Psi_{1,2,3}, \Psi_{1,2,4}, \Psi_{2,1,1}, \ldots, \Psi_{3,2,3,}, \Psi_{3,2,4}\Big)\end{aligned}$$

where $\boldsymbol{U}_1$ is the vector corresponding to the first angular element for example. Then, considering the scattering operator we have the following

$$\begin{aligned}\tilde{S}(\psi, v) &= \int_\Omega \int_I \left(\int_I \frac{\Sigma_s}{2\pi} \psi(\boldsymbol{x}, \varphi') \, \mathrm{d}\varphi' \right) v \, \mathrm{d}\varphi \, \mathrm{d}\boldsymbol{x}, \\ &= \frac{\Sigma_s}{2\pi} \int_\Omega \int_I \left(\int_I \psi(\boldsymbol{x}, \varphi') \, \mathrm{d}\varphi' \right) v \, \mathrm{d}\varphi \, \mathrm{d}\boldsymbol{x}, \\ &= \frac{\Sigma_s}{2\pi} \int_\Omega \int_I \left(\int_I \sum_{\kappa_I=1}^{3} \sum_{j=1}^{4} \sum_{l=1}^{2} \Psi_{\kappa_I,j,l} \zeta_j(\boldsymbol{x}) \xi_l(\varphi') \, \mathrm{d}\varphi' \right) \zeta_i(\boldsymbol{x}) \xi_k(\varphi) \, \mathrm{d}\varphi \, \mathrm{d}\boldsymbol{x}, \\ &= \sum_{\kappa_I=1}^{3} \sum_{j=1}^{4} \sum_{l=1}^{2} \frac{\Sigma_s}{2\pi} \int_\Omega \int_I \Psi_{\kappa_I,j,l} \zeta_j(\boldsymbol{x}) \left(\int_I \xi_l(\varphi') \, \mathrm{d}\varphi' \right) \zeta_i(\boldsymbol{x}) \xi_k(\varphi) \, \mathrm{d}\varphi \, \mathrm{d}\boldsymbol{x}, \\ &= \sum_{\kappa_I=1}^{3} \sum_{j=1}^{4} \sum_{l=1}^{2} \frac{\Sigma_s}{2\pi} \int_\Omega \int_I \Psi_{\kappa_I,j,l} \zeta_j(\boldsymbol{x}) \left(\sum_{\kappa_I=1}^{3} \int_{\kappa_I} \xi_l(\varphi') \, \mathrm{d}\varphi' \right) \zeta_i(\boldsymbol{x}) \xi_k(\varphi) \, \mathrm{d}\varphi \, \mathrm{d}\boldsymbol{x},\end{aligned}$$

for i=1,...4, k = 1,2. With the definition $b_l = \int_{\kappa_I} \xi_l(\varphi')$ we can define $\hat{\Psi}_{\kappa_I,j,l} = \Psi_{\kappa_I,j,l} b_l$, hence

$$\begin{aligned}\tilde{S}(\psi, v) &= \sum_{\kappa_I=1}^{3} \sum_{j=1}^{4} \sum_{l=1}^{2} \frac{\Sigma_s}{2\pi} \int_\Omega \int_I \hat{\Psi}_{\kappa_I,j,l} \zeta_j(\boldsymbol{x}) \zeta_i(\boldsymbol{x}) \xi_k(\varphi) \, \mathrm{d}\varphi \, \mathrm{d}\boldsymbol{x}, \\ &= \sum_{\kappa_I=1}^{3} \sum_{j=1}^{4} \sum_{l=1}^{2} \frac{\Sigma_s}{2\pi} \int_\Omega \hat{\Psi} 2_{\kappa_I,j,l} b_l \zeta_j(\boldsymbol{x}) \zeta_i(\boldsymbol{x}) \xi_k(\varphi) \, \mathrm{d}\boldsymbol{x}.\end{aligned}$$

We conclude that the application of the scattering operator may be expressed, in the discretised case as the action of three matrices on the vector $\boldsymbol{U}$, namely

$$S(\psi_h, v_h) = \frac{\Sigma_s}{2\pi} BMB\boldsymbol{U}$$

In the expression above B is a block diagonal matrix such that each sub block is diagonal;

$$B = \begin{pmatrix} B_1 & 0 & 0 \\ 0 & B_2 & 0 \\ 0 & 0 & B_3 \end{pmatrix},$$

where each of the B_r are of size number of spatial degrees of freedom $\times$ number of

angular degrees of freedom per angular element and have the form,

$$B_r = \begin{pmatrix} b_{r,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & b_{r,1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & b_{r,1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b_{r,1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & b_{r,2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & b_{r,2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & b_{r,2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & b_{r,2} \end{pmatrix},$$

where $b_{r,l}$ is the integral of the lth angular basis function over the angular element $\kappa_{I,r}$, $b_{r,l} = \int_{\kappa_{I,r}} \xi_l(\varphi)\, \mathrm{d}\varphi$. The matrix M is composed of a collection of mass matrices m of the spatial basis functions,

$$M = \begin{pmatrix} m & m & m \\ m & m & m \\ m & m & m \end{pmatrix},$$

with each m also of size number of spatial degrees of freedom $\times$ number of angular degrees of freedom per angular element such that

$$m = \begin{pmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} & m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} \\ m_{4,1} & m_{4,2} & m_{4,3} & m_{4,4} & m_{4,1} & m_{4,2} & m_{4,3} & m_{4,4} \\ m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} & m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} \\ m_{4,1} & m_{4,2} & m_{4,3} & m_{4,4} & m_{4,1} & m_{4,2} & m_{4,3} & m_{4,4} \end{pmatrix}$$

with

$$m_{i,j} = \int_\Omega \zeta_i(\boldsymbol{x})\zeta_j(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}, \quad i,j = 1,\ldots 4.$$

Recall that the support of the basis functions $\zeta_i(\boldsymbol{x})$ is not global, and so the sub block matrices m are sparse. As one would expect, after application of the scattering operator the vectors $\boldsymbol{U}_1, \boldsymbol{U}_2, \boldsymbol{U}_3$ are the same.

In the primal problem we apply the operators as stated above, $S(\psi_h, v_h) = \frac{\Sigma_s}{2\pi} BMB\boldsymbol{U}$,

Table 6.8: Uniform Refinement Results for the first 2d Eigenvalue Test Case

Mesh	k_{eff}	$1/k_{eff}$	Error Estimate	Error from reference value	Effectivity
1	0.5761772	1.7355773	-2.8446492E-02	-5.6822759E-02	0.50
2	0.5885543	1.6990785	-1.5992837E-02	-2.0323974E-02	0.79
3	0.5869781	1.7036410	-2.4652957E-02	-2.4886500E-02	0.99
4	0.5890949	1.6975193	-1.8721338E-02	-1.8764826E-02	1.00
5	0.5916588	1.6901634	-1.1403086E-02	-1.1408939E-02	1.00
6	0.5934615	1.6850292	-6.2746814E-03	-6.2746814E-03	1.00

however for the dual eigenvalue problem we must evaluate

$$\begin{aligned} S(v_h, z_h) &= (BMB)^\top \boldsymbol{U} \\ &= B^\top (BM)^\top \boldsymbol{U} \\ &= B^\top M^\top B^\top \boldsymbol{U}, \end{aligned}$$

where we note that the transposition is simple since both B and M are symmetric. The same approach can be used for the fission operator. It now remains to consider the transport operator; this is straightforward, as all coupling between angular elements is performed by the fission and scattering operators. Hence, the full linear system due to the transport operator is block diagonal and thus easily transposed:

$$T = \begin{pmatrix} T_1 & 0 & 0 \\ 0 & T_2 & 0 \\ 0 & 0 & T_3 \end{pmatrix},$$

where the T_r is the application of the transport operator on the angular element r.

With these constructions in place, the dual eigenvalue solver can be implemented giving an error indicator which can be used to guide adaptive mesh refinement for the eigenvalue problem.

In Table 6.8 we present preliminary results for the error indicator. These results are for the first two dimensional eigenvalue problem as presented in Chapter 5 under uniform refinement with the polynomial degree in space and polynomial degree in space both zero. As can be seen when we take the reference value to be the final eigenvalue approximation plus the associated error estimate (since we have no analytical value to take) we see good effectivities of 1.00. Work is ongoing to establish that this error estimate works as an indicator to guide refinement for the eigenvalue problem.

CHAPTER 7

Conclusions And Further Work

7.1 Conclusions

In this thesis we have investigated discretisation methods for the neutron transport equation which are high order in both the spatial domain and the angular domain. The work has been heavily computational, with much time devoted to code development and the derivation of novel solution techniques. The techniques developed here have wide ranging practical applications, indeed we have used the solver to compute critical eigenvalues for three industrial benchmark problems from the literature.

Following a detailed review of common angular and spatial discretisation techniques contained in Chapter 2, we began work on developing a high order discretisation of the neutron transport equation employing discontinuous Galerkin methods in both the spatial and angular domains. We began this development in Chapter 3; here we first employed the popular and intuitive approach to discretise the angular domain and the coupling due to the integral on the right hand side of the neutron transport equation, namely the discrete ordinates method. In conjunction with the discrete ordinates angular discretisation we discretised the spatial domain with a DG method, utilising suitable numerical flux functions to handle the inter-element discontinuities. We have shown, in practice, that there is little point in coupling an arbitrary order discretisation with the discrete ordinates approach as the rate of convergence of the full scheme (referred to as DO-DG scheme) will be limited by the rate of convergence of the discrete ordinates scheme. Note, that this rate has been observed to be $O(h^2)$ with respect to the L_2–norm of the error in the approximation of the scalar flux but $O(h^{3/2})$ with respect to the L_2–norm of the angular flux error. The Source Iteration (SI) technique has been applied in the solution of our discretised scheme and the convergence with respect to varying physical material properties has been investigated. An advantage of the SI is

its ease of use, and for the reactor core calculations that we are primarily interested in, has been demonstrated to be practical, converging in a reasonable number of iterations. However, we note that this aspect of the solvers developed in this work could be considerably improved, and is a topic for further research as there has been little work on acceleration schemes for unstructured grids in the in the literature.

Once the limited rate of convergence of the DO-DG scheme had been established, we derived an arbitrary order scheme employing a DG(p) discretisation the spatial domain and a DG(q) discretisation in the angular domain. This discretisation performs well in the two dimensional case, attaining optimal orders of convergence for linear functionals of the numerical solution in addition to the scalar flux and angular flux. However, when extended to the more difficult pseudo 3D case the presence of a singularity on the boundary of the angular domain limits the convergence of the angular flux, like that which occurred for a discrete ordinates discretisation. Following testing of the source problem, we implemented a critical eigenvalue solver for both the two dimensional and pseudo 3D neutron transport equations. This eigenvalue solver performs well for arbitrary orders of polynomial approximation in both angle and space for the three two dimensional eigenvalue problems considered in this thesis. For the pseudo 3D case, however, the situation is less satisfactory; we note that the eigenvalue code converges to the correct k–effective eigenvalue, however the time taken to compute these at present is a difficulty. Further work is required for the solver to be competitive when performing industrial computations.

Chapter 4 looks closely at the proposed discretisations; examining the matrix structure of the resulting linear systems. We then exploit this structure to develop a linear solver that orders the elements according to the advective direction and then solves for each element one at a time. This new solver, whilst being slower than employing MUMPS on the resulting linear systems, can handle larger problems as the memory requirements are minimal. We have also implemented a mesh generator for the generation of suitable computational meshes for the angular domain.
In Chapter 6, the dual weighted residual error estimation approach was applied to the neutron transport problem. To our knowledge this is the first time that an arbitrary order in the angular dimension solver with adaptivity in both space and angle has been developed. The adaptive code performs well in two dimensional tests, regularly attaining effectivity indices of one; in the Pseudo 3D case we obtain effectivities close to one, however the results aren't as strong as in the two dimensional case. We believe this again is due to the quadrature issues present for integration over the angular do-

main and the suboptimal convergence of the angular flux that this leads to. We note that the results presented in this thesis indicate that h adaptivity in the angular dimension does not present a large benefit over uniform refinement. This is probably due to the relative smoothness of the angular component in the solution of neutron transport source problems and may be different for the industrial eigenvalue computations which are the ultimate goal of this work. The use of the same spatial mesh on each angular element also appears to introduce more refinement than is necessary, however a separate mesh for each angular element dramatically increases the memory storage requirements of the solver. An error estimate for eigenvalue computation has also been derived, and is currently being tested in order to perform adaptive computations. This will be the focus of further work in collaboration with colleagues at the University of Nottingham and the University of Leicester.

7.2 Further Work

Following the original work considered in this thesis, we believe that there is considerable scope for future work to extend and develop the ideas discussed here. In this concluding section of the thesis we make some suggestions for future work, together with a brief discussion for each suggestion.

7.2.1 Increasing the Neutronics Capability

At present, the code is heavily restricted to certain individual benchmarks. We propose two, minor, modifications that will ease these restrictions. The first is implementation of the multigroup discretisation for the energy dimension of the neutron transport equation [74, 145]. Here the energy spectrum $[E_0, E_{max}]$ is partitioned into G non-overlapping groups. We then solve a neutron transport equation in each energy group; where the coupling between energy groups is taken care of by the integral operator on the right hand side of the neutron transport equation and suitable cross sections describing the scattering from one energy group to another. This will increase the run time of the solver, but will enable more realistic benchmarks such as the C5G7, Larsen and Alcouffe [98] and the KNK fast reactor benchmark [155]. The second modification would be to incorporate reflective boundary conditions to reduce the work load required for these more complicated benchmarks.

Table 7.1: Preliminary results for the LA7 benchmark with exact angular integration.

Mesh number	Computed Eigenvalue
1	1.063158
2	1.008232
3	0.984436
4	0.975196

7.2.2 Extension of the h–Adaptivity Procedures

For the neutron transport problems considered we have been restricted to performing h–refinement. This is due to the absence of an appropriate "smoothness indicator" for a numerical solution living in the tensor product space derived from the spatial finite element space and angular finite element space. The adaptivity results for both the advection problem and the Laplace eigenvalue problem suggest that performing hp–adaptivity would be advantageous.
Furthermore, the extension to anisotropic mesh refinement [79] could be beneficial in some practical situations. For example, some industrial benchmarks have moderator regions that are significantly thinner in one direction than the other. Due to the presence of materials with varying properties it is important that such features are resolved; being able to perform refinement targeted in one direction should lead to a reduction in the overall number of degrees of freedom required for solution to a given accuracy.

7.2.3 Improved Quadrature for Pseudo 3D Discretisation

We have seen that for the pseudo 3D eigenvalue benchmarks, even when using a graded angular mesh, the difficulty of performing quadrature in the presence of the boundary singularity of the angular domain severely restricts the accuracy of the computed eigenvalues. We propose two approaches to mitigate this. The first is to perform any angular integration exactly by changing from Cartesian coordinates to polar coordinates so that integrals over angular elements can be computed exactly. This has already been implemented for piecewise constant approximations ($q = 0$) in angle with the preliminary results shown in Table 7.1 for the computation of the critical eigenvalue for the LA7 benchmark considered previously. Noting that the error has not necessarily increased on the fourth mesh as we only know that the k effective eigenvalue should be close to 1 (we don't have an exact value), these results look promising. Ideally, however, for maintaining generality of the code, the integration in the angular domain would be accomplished by the use of some suitable quadrature. We propose the investigation of

adaptive quadratures that are designed to deal effectively with singularities.

7.2.4 Extension to a True 3D Discretisation

In this work, we have only considered the discretisation of a pseudo 3–dimensional problem. As we have observed, this can lead to computational difficulties arising from the projection of the surface of the unit sphere down on to the unit disc. In addition to this, the physical model breaks down where there is a large degree of heterogeneity on the z axis. An obvious remedy would be to discretise the full 3D problem directly. To do this, one would need to triangulate the surface of a sphere, this is harder than discretising a disc, however there is a brief discussion of a suitable approach obtained by successive subdivision of an icosahedron in [95]. The ordered solver that has been discussed in this work would also have to be generalised for this problem.

7.2.5 The Use of Composite and Polygonal Finite Element Methods

Two recent developments in the field of discontinuous Galerkin methods, namely, composite DG methods and DG methods on general polygonal domains, lend themselves well to application for the neutron transport problem.

Many industrial benchmarks such as the C5G7 benchmark discussed in the introduction to this work contain geometrical features that are small compared to the domain. To obtain accurate solutions these features need to be resolved by the underlying finite element mesh; this leads to a large number of degrees of freedom, resulting in a long computational time. The composite DG finite element methods, (as an extension to composite FEMS) are described in the article of Antonietti, Giani and Houston [9] and rely on the construction of a hierarchy of meshes such that the finest mesh represents the geometrical structure. The finite element spaces are then naturally defined in terms of prolongation operators.

Another approach that could be useful for benchmarks, such as the KNK-II fast reactor benchmark whose domain is made up of rings of hexagonal elements is the application of a DG method on general polygonal domains. This has been developed by Cangiani, Georgoulis and Houston [44]. In their paper they employ polynomial basis functions which are defined in the physical space (by the placing of a bounding box around an element, on which basis functions are defined); this technique eliminates the need for a mapping from a reference element. An interesting consequence of this is that spaces

of total degree p can be used on general elements whilst still obtaining optimal convergence, further reducing the number of degrees of freedom employed. We believe this is a promising avenue of research with applications for the neutron transport equation.

7.2.6 Parallelisation

Commercial neutron transport codes make use of parallel programming paradigms to ensure complicated problems can be solved in a reasonable amount of time. This is an obvious extension of the work presented in the thesis since the construction of the method lends itself well to parallelisation. The solution on each angular element could be solved on a different processor node, before being collected together for computation of the coupling integral. In addition, the DG method itself is highly parallelisable, adding a further level of possible parallelisation.

APPENDIX A

Derivation of The Neutron Transport Equation

The derivation presented here largely follows the outline given by Barry Ganapol at the Course on Analytical Benchmarks: Case Studies in Neutron Transport Theory in Paris, 16-19th March 2009 (details of which can be found in [74]).

Before we proceed with the derivation we need to introduce the concept of a solid angle: consider an object and a point in space p_n. If you imagine standing at the point p_n, then the object takes up a certain amount of your "total view". This amount of "total view" can be quantified by projecting every point on the surface of the object to the point p_n. This creates a shape on the surface of an imaginary sphere of radius 1 cm such that p_n is the centre of the sphere. The area of this shape will be in the range 0 to 4π (the total surface area of the sphere) and this is referred to as the solid angle of the object subtended at the point p_n. This quantity is measured in steradians.

The derivation considers a neutron to be a point particle and takes advantage of the statistics of large numbers. For convenience, we define $\boldsymbol{P} := (\boldsymbol{x}, \boldsymbol{\mu}, E)$. Point collisions occur in a statistically averaged phase space continuum, realized by defining the phase volume element $\Delta\boldsymbol{P} = \Delta\boldsymbol{x}\Delta\boldsymbol{\mu}\Delta E$ about $\boldsymbol{P}$. Here, $\Delta\boldsymbol{\mu}$ is the solid angle, where 4π steradians account for all possible directions on a unit sphere, $\Delta\boldsymbol{x}$ is a small region in space and ΔE is a small interval in the energy range. By defining the neutron density distribution to be $N(\boldsymbol{x}, \boldsymbol{\mu}, E, t)$, the number of neutrons in $\Delta\boldsymbol{P}$ at time t is given by $N(\boldsymbol{x}, \boldsymbol{\mu}, E, t)\Delta\boldsymbol{P}$. If the neutron track length is defined to be $v\Delta t$, (i.e., the distance travelled per neutron of speed v), then the total track length of all neutrons in $\Delta\boldsymbol{P}$ is

equal to $vN(\boldsymbol{x}, \boldsymbol{\mu}, E, t)\Delta\boldsymbol{P}\Delta t$. Finally, we define the angular flux at time t as

$$\psi(\boldsymbol{x}, \boldsymbol{\mu}, E, t) = vN(\boldsymbol{x}, \boldsymbol{\mu}, E, t).$$

Neutrons experience three types of interaction: scattering (s), fission (f), and absorption (a). We assume neutrons interact with stationary nuclei (interactions with other moving neutrons are neglected).
The cross section for an interaction of type i with the nucleus of nuclide j is:

$$\begin{aligned}\sigma_{ij}(\boldsymbol{x}, E, t) = {} & \text{Fractional probability of neutron interaction } i \\ & \text{with nuclide } j \text{ per path length travelled,}\end{aligned}$$

where $i = s, f, a$. Therefore, the reaction rate for interaction i in $\Delta\boldsymbol{P}$ with nuclide j is equal to $\sigma_{ij}(\boldsymbol{x}, E, t)\psi(\boldsymbol{x}, \boldsymbol{\mu}, E, t)\Delta\boldsymbol{P}$. Assuming interactions are independent events, the total interaction rate of type i is the sum over all participating nuclear species giving the total macroscopic cross section

$$\sigma_i(\boldsymbol{x}, E, t) = \sum_j \sigma_{ij}(\boldsymbol{x}, E, t).$$

This interaction rate depends only on the motion of neutrons between collisions. However, for scattering we must also consider the direction in which neutrons are scattered. We do this by defining the scattering kernel

$$\begin{aligned}f_s(\boldsymbol{\mu}', \boldsymbol{\mu}, E', E)\Delta\boldsymbol{\mu}\Delta E = {} & \text{Fractional probability of scattering from} \\ & \text{direction } \boldsymbol{\mu}' \text{ and energy } E' \text{ to direction} \\ & \boldsymbol{\mu} \text{ and energy } E.\end{aligned}$$

This scattering kernel is normalised so that

$$\int_0^\infty \int_{4\pi} f_s(\boldsymbol{\mu}', \boldsymbol{\mu}, E', E)\, \mathrm{d}\boldsymbol{\mu}\, \mathrm{d}E = 1 \qquad \forall \boldsymbol{\mu}', E'.$$

Similarly, we define the fractional probability of a fission neutron appearing in the direction range $\Delta\boldsymbol{\mu}$ and energy range ΔE to be

$$\frac{\chi(E)}{4\pi}\Delta\boldsymbol{\mu}\Delta E,$$

where $\chi(E)$ is a distribution, together with normalisation $\int_0^\infty \chi(E)\, \mathrm{d}E = 1$. The average number of neutrons produced per fission is denoted by $\nu(E)$.

Now we can consider the total neutron balance in the entire volume V and partial phase space element $\Delta\boldsymbol{\mu}\Delta E$ during a time Δt as follows

$$\begin{aligned}\text{Number at time } t + \Delta t = {} & \text{Number at time } t + \text{Number gained during } \Delta t \\ & - \text{Number lost during } \Delta t.\end{aligned} \tag{A.1}$$

Number of neutrons in a specific volume

By integrating the neutron density distribution over the volume, the total number of neutrons in $V\Delta\boldsymbol{\mu}\Delta E$ at a given time s is given by

$$\int_V \Delta\boldsymbol{\mu}\Delta E N(\boldsymbol{x},\boldsymbol{\mu},E,s)\,\mathrm{d}\boldsymbol{x}. \tag{A.2}$$

Gain due to scattering

Neutrons from any points within V can be scattered into the element $\Delta\boldsymbol{\mu}\Delta E$. The total number scattered during Δt from any element $\Delta E'\Delta\boldsymbol{\mu}'$ is given by the total scattering rate in V multiplied by Δt, i.e.,

$$\int_V \sigma_s(\boldsymbol{x},E',t)\psi(\boldsymbol{x},\boldsymbol{\mu}',E',t)\Delta t\,\mathrm{d}E'\,\mathrm{d}\boldsymbol{\mu}'\,\mathrm{d}\boldsymbol{x}.$$

However, only $f_s(\boldsymbol{\mu}',\boldsymbol{\mu},E',E)\Delta\boldsymbol{\mu}\Delta E$ reach $\Delta\boldsymbol{\mu}\Delta E$. Therefore the total number of neutrons scattering into $\Delta\boldsymbol{\mu}\Delta E$ during Δt within V is given by integrating over all possible differential phase space elements $\Delta E'\Delta\boldsymbol{\mu}'$:

$$\int_V\int_0^\infty\int_0^{4\pi} f_s(\boldsymbol{\mu}',E',\boldsymbol{\mu},E)\sigma_s(\boldsymbol{x},E',t)\psi(\boldsymbol{x},\boldsymbol{\mu}',E',t)\Delta\boldsymbol{\mu}\Delta E\Delta t\,\mathrm{d}\boldsymbol{\mu}'\,\mathrm{d}E'\,\mathrm{d}\boldsymbol{x}. \tag{A.3}$$

Gain due to fission production

For every fission occuring within V during Δt in a differential element $\Delta E'\Delta\boldsymbol{\mu}'$, $\frac{\chi(E)}{4\pi}\nu(E')\Delta\boldsymbol{\mu}\Delta E$ neutrons appear in $\Delta\boldsymbol{\mu}\Delta E$. This gives a total gain due to fission in $V\Delta\boldsymbol{\mu}\Delta E$ of

$$\int_V \frac{\chi(E)}{4\pi}\int_0^\infty\int_0^{4\pi} \nu(E')\sigma_f(\boldsymbol{x},E',t)\psi(\boldsymbol{x},\boldsymbol{\mu}',E',t)\Delta\boldsymbol{\mu}\Delta E\Delta t\,\mathrm{d}\boldsymbol{\mu}'\,\mathrm{d}E'\,\mathrm{d}\boldsymbol{x}. \tag{A.4}$$

Loss due to leaving V

Neutrons can leave the volume V; consider an elemental area $\mathrm{d}A$ on the surface of V with unit outward normal $\boldsymbol{n}$. Then the number of neutrons leaving through $\mathrm{d}A$ from the element $\Delta\boldsymbol{\mu}\Delta E$ during Δt is:

$$\Delta\boldsymbol{\mu}\Delta E\Delta t\;\boldsymbol{n}\cdot v\boldsymbol{\mu}N(\boldsymbol{x},\boldsymbol{\mu},E,t)\,\mathrm{d}A.$$

Therefore, over the entire surface of V, the total number of neutrons leaving is (after applying the divergence theorem):

$$\int_A \boldsymbol{n}\cdot v\boldsymbol{\mu}N(\boldsymbol{x},\boldsymbol{\mu},E,t)\Delta\boldsymbol{\mu}\Delta E\Delta t\,\mathrm{d}A = \int_V \nabla_x\cdot v\boldsymbol{\mu}N(\boldsymbol{x},\boldsymbol{\mu},E,t)\Delta\boldsymbol{\mu}\Delta E\Delta t\,\mathrm{d}\boldsymbol{x}. \tag{A.5}$$

Loss due to absorption and scattering

The number of neutrons lost due to absorption in $V\Delta\boldsymbol{\mu}\Delta E$ is given by:

$$\int_V \sigma_a(\boldsymbol{x}, E, t)\psi(\boldsymbol{x}, \boldsymbol{\mu}, E, t)\Delta\boldsymbol{\mu}\Delta E\Delta t \, \mathrm{d}\boldsymbol{x}.$$

Neutrons will also be lost when they scatter out of the energy range ΔE or the direction range $\Delta\boldsymbol{\mu}$. Since the scattering kernel has been normalised, the number of neutrons lost due to scattering out of $V\Delta\boldsymbol{\mu}\Delta E$ is:

$$\int_V \sigma_s(\boldsymbol{x}, E, t)\phi(\boldsymbol{x}, \boldsymbol{\mu}, e, t)\Delta\boldsymbol{\mu}\Delta E\Delta t \, \mathrm{d}\boldsymbol{x}.$$

Putting it all together

Combining all the above contributions (by following equation (A.1)), dividing through by $\Delta\boldsymbol{\mu}\Delta E\Delta t$ and taking the limit as $\Delta\boldsymbol{\mu}, \Delta E, \Delta t$ tend to 0, we get

$$\begin{aligned}
&\int_V \left[\frac{1}{v}\frac{\partial}{\partial t} + \boldsymbol{\mu}\cdot\nabla_x + \sigma_t(\boldsymbol{x}, E, t)\right]\psi(\boldsymbol{x}, \boldsymbol{\mu}, E, t)\mathrm{d}\boldsymbol{x} \\
&- \int_V \int_0^\infty \int_{4\pi} \sigma_s(\boldsymbol{x}, \boldsymbol{\mu}', E', \boldsymbol{\mu}, E, t)\psi(\boldsymbol{x}, \boldsymbol{\mu}', E', t) \, \mathrm{d}\boldsymbol{\mu}' \, \mathrm{d}E'\mathrm{d}\boldsymbol{x} \\
&- \int_V \frac{\chi(E)}{4\pi} \int_0^\infty \int_{4\pi} \nu(E')\sigma_f(\boldsymbol{x}, E', t)\psi(\boldsymbol{x}, \boldsymbol{\mu}', E', t) \, \mathrm{d}\boldsymbol{\mu}' \, \mathrm{d}E'\mathrm{d}\boldsymbol{x} = 0,
\end{aligned}$$

where

$$\begin{aligned}
\sigma_t(\boldsymbol{x}, E, t) &= \sigma_a(\boldsymbol{x}, E, t) + \sigma_s(\boldsymbol{x}, E, t), \\
\sigma_s(\boldsymbol{x}, \boldsymbol{\mu}', E', \boldsymbol{\mu}, E, t) &= \sigma_s(\boldsymbol{x}, E', t)f_s(\boldsymbol{\mu}', \boldsymbol{\mu}, E', E).
\end{aligned}$$

Since the volume is arbitrary, the integral is only zero if the integrand is zero, and so we arrive at

$$\begin{aligned}
\left[\frac{1}{v}\frac{\partial}{\partial t} + \boldsymbol{\mu}\cdot\nabla_x + \sigma_t(\boldsymbol{x}, E, t)\right]\psi(\boldsymbol{x}, \boldsymbol{\mu}, E, t) = &\int_0^\infty \int_0^{4\pi} \sigma_s(\boldsymbol{x}, \boldsymbol{\mu}', E', \boldsymbol{\mu}, E, t)\psi(\boldsymbol{x}, \boldsymbol{\mu}', E', t) \, \mathrm{d}\boldsymbol{\mu}' \, \mathrm{d}E' \\
&+ \frac{\chi(E)}{4\pi} \int_0^\infty \int_0^{4\pi} \nu(E')\sigma_f(\boldsymbol{x}, E', t)\psi(\boldsymbol{x}, \boldsymbol{\mu}', E', t) \, \mathrm{d}\boldsymbol{\mu}' \, \mathrm{d}E'.
\end{aligned}$$

Which, on combining the terms in the RHS, becomes

$$\begin{aligned}
&\frac{1}{v}\frac{\partial}{\partial t}\psi(\boldsymbol{x}, \boldsymbol{\mu}, E, t) + \boldsymbol{\mu}\cdot\nabla_x\psi(\boldsymbol{x}, \boldsymbol{\mu}, E, t) + \sigma_t(\boldsymbol{x}, \boldsymbol{\mu})\phi(\boldsymbol{x}, \boldsymbol{\mu}, E, t) \\
&\qquad = \int_0^\infty \int_0^{4\pi} f(\boldsymbol{x}, \boldsymbol{\mu}', E', \boldsymbol{\mu}, E)\psi(\boldsymbol{x}, \boldsymbol{\mu}', E', t) \, \mathrm{d}\boldsymbol{\mu}' \, \mathrm{d}E',
\end{aligned} \tag{A.6}$$

where the function $f(\boldsymbol{x}, \boldsymbol{\mu}', E', \boldsymbol{\mu}, E)$ is defined by

$$f(\boldsymbol{x}, \boldsymbol{\mu}', E', \boldsymbol{\mu}, E) = \sigma_s(\boldsymbol{x}, \boldsymbol{\mu}', E', \boldsymbol{\mu}, E) + \frac{\chi(E)}{4\pi}\nu(E')\sigma_f(\boldsymbol{x}, E', t).$$

This is the same as that stated in equation (1.1).

APPENDIX B

Notes on the Computer Programming for this Work

The extensive computer programming for this work was performed in Fortran, specifically Fortran conforming to the 2003 standard. The aim of this appendix is to briefly describe some of the main design decisions made when implementing this neutron transport solver. Further details are available on request.
Some external packages have been used in conjunction with original code and we now provide a brief listing of these. The general purpose Delaunay mesh generator Triangle [136] was used to triangulate domains which were not easily broken into quadrilateral elements, for example the spatial meshes for the LA7 and LA9 eigenvalue benchmarks considered in Chapter 5. As mentioned in Chapter 4 Triangle was initially used for the generation of the angular mesh for the pseudo 3D code before a custom mesh generator was written to produce the graded meshes required to ensure convergence.
To solve the sparse linear systems arising from the finite element discretisation, when an ordered solver has not been used we have used the MUMPS (MUltifrontal Massively Parallel sparse direct Solver) [3] direct solver. This is a direct (as opposed to iterative) sparse linear system solver developed by a team working primarily at CERFACS and INRIA in France. In addition gotoBLAS has been used as the BLAS implementation of choice.

The computer code is structured in such a way to make use of routines available in the finite element package AptoFEM (www.aptofem.com) developed at the University of Nottingham under Paul Houston. This package contains routines for the computation of finite element basis functions, interfaces to the aforementioned linear solver MUMPS and data types for the storage of a finite element mesh and discontinuous Galerkin finite element solution. The use of these routines has meant that the focus

of programming time could be directed towards the neutron transport problem as opposed to the programming of generic finite element routines.

We shall now briefly provide an introduction to some of the considerations made when developing the neutron transport software.

The discontinuous Galerkin discretisation of the neutron transport equation requires the storage of a finite element solution on each angular element. This information needs to be accessed by many different routines (for example when computing the scalar flux, computing an interpolant of the angular flux and computing norms of the solution) and so the decision was made to define a data type `nt_data_type` that was stored in a module `neutron_transport_type.f90` and so was globally accessible. The attributes of this data type are shown in the code fragment shown below.

```
type nt_data_type

    integer :: no_angles
    type(mesh) :: spatial_mesh
    type(solution), dimension(:), allocatable :: solution_array
    integer, dimension(:), allocatable :: angle_poly_degree
    integer, dimension(:), allocatable :: angle_ele_dofs
    integer, dimension(:), allocatable :: spatial_ele_dofs

    type(solution) :: scalar_flux
    real(db), dimension(:), allocatable :: angular_mesh_coordinates

  end type nt_data_type
```

This data type stores the spatial mesh (which is the same for all angular meshes) and an array of type `solution` (an AptoFEM data type) which stores the finite element solution for each angular element. The code fragment shown above is for the data type associated with the two dimensional problem and so the angular mesh is stored as an array of real numbers which store the coordinates of the mesh nodes. For the pseudo 3D case there is an attribute `angular_mesh` which stores an angular mesh as the AptoFEM data type `mesh` in the same way that the spatial mesh is stored. In addition to these this data type also stores information regarding the polynomial degree, the number of spatial degrees of freedom and the number of angular degrees of freedom on each angular element. This information is used when allocating storage for and constructing

the discretisation matrices arising from the discontinuous Galerkin discretisation. In addition to defining the data type `nt_data_type` this module also contains numerous routines for setting up, modifying and deleting attributes of the data type.

As this neutron transport solver is primarily a research code the decision was made to split up the computation of mesh refinement indicators from the refinement and modification of the spatial and angular meshes and associated solutions. For this reason the routine in the `neutron_transport_type` module which implements the refinement of both the angular and spatial meshes has, as an input, integer valued arrays of refinement indicators. These arrays are of length the number of angular elements and the number of spatial elements respectively. If an entry is labelled one it indicates that the corresponding angular or spatial element is to be refined. The generation of these refinement indicators is accomplished elsewhere in the code and is based on one of the various marking strategies described in Section 6.3. The choice of refinement strategies can be made at run time by the user with the selection of appropriate keys in the control file.

The specific nature of the neutron transport discretisation meant that many key routines in AptoFEM could not be used. The primary example of this would be the matrix set–up and population routines. The tensor product nature of the finite element basis used leads to the blocked matrix structure discussed in Chapter 4 which in turn led to the development of specific matrix population routines which could access particular blocks of the full matrix. Similarly routines were developed which implemented the coupling terms appearing on the right hand side of the neutron transport equation.

As discussed previously, it became apparent when working on the pseudo 3D problem that an arbitrary triangular mesh of the angular domain, which in this case is the unit disc, adversely affected the convergence rates of the solver (even if an approximation to the circle with hundreds of line segments was used). A mesh generator that produced a mesh of simplices and quadrilaterals which were graded towards the edge of the unit disc - the area of singularity - was developed. The method to produce this grading was chosen so that the angular meshes were analogous with the discrete ordinate splitting used by Baker [27] so that the methods could be easily compared. This mesh generator was also written to interface with the mesh representation required by AptoFEM for ease of use.

References

[1] I.K. Abu-Shumays. Compatible product angular quadrature for neutron transport in xy geometry. *Nucl. Sci. Eng.;(United States)*, 64(2), 1977.

[2] M.L. Adams and E. W. Larsen. Fast iterative methods for discrete-ordinates particle transport calculations. *Progress in nuclear energy*, 40(1):3–159, 2002.

[3] Patrick R Amestoy, Iain S Duff, Jean-Yves L'Excellent, and Jacko Koster. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23(1):15–41, 2001.

[4] P.R. Amestoy, I.S. Duff, J. Koster, and J.-Y.L'Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23(1):15–41, 2001.

[5] Paola F Antonietti and Blanca Ayuso. Multiplicative schwarz methods for discontinuous galerkin approximations ofelliptic problems. In *Communications to SIMAI Congress*, volume 1, 2007.

[6] P.F. Antonietti and B. Ayuso. Schwarz domain decomposition preconditioners for discontinuous galerkin approximations of elliptic problems: non-overlapping case. *ESAIM: Mathematical Modelling and Numerical Analysis*, 41(01):21–54, 2007.

[7] P.F. Antonietti and B. Ayuso. Two-level schwarz preconditioners for super penalty discontinuous galerkin methods. *Communications in Computational Physics*, 2007.

[8] P.F. Antonietti and P. Houston. A class of domain decomposition preconditioners for hp-discontinuous galerkin finite element methods. *Journal of Scientific Computing*, 46(1):124–149, 2011.

[9] P.F. Antonietti, S. Giani, and P. Houston. hp-version composite discontinuous galerkin methods for elliptic problems on complicated domains. *SIAM Journal on Scientific Computing*, 35(3):A1417–A1439, 2013.

[10] D.N. Arnold. *An interior penalty finite element method with discontinuous elements*. PhD thesis, The University of Chicago, 1979.

[11] D.N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM journal on numerical analysis*, 19(4):742–760, 1982.

[12] D.N. Arnold, F. Brezzi, B. Cockburn, and D. Marini. Discontinuous Galerkin methods for elliptic problems. In *Discontinuous Galerkin methods (Newport, RI, 1999)*, volume 11 of *Lect. Notes Comput. Sci. Eng.*, pages 89–101. Springer, Berlin, 2000.

[13] D.N. Arnold, D. Boffi, and R.S. Falk. Approximation by quadrilateral finite elements. *Math. Comp.*, 71(239):909–922 (electronic), 2002. ISSN 0025-5718.

[14] D.N. Arnold, F. Brezzi, B. Cockburn, and L.D. Marini. Unified analysis of discontinuous galerkin methods for elliptic problems. *SIAM Journal of Numerical Analysis*, 39(5):1759–1779, 2002.

[15] M. Asadzadeh. Analysis of a fully discrete scheme for neutron transport in two-dimensional geometry. *SIAM J. Numer. Anal.*, 23(3):543–561, 1986. ISSN 0036-1429. doi: 10.1137/0723035. URL `http://dx.doi.org/10.1137/0723035`.

[16] M. Asadzadeh. A finite element method for the neutron transport equation in an infinite cylindrical domain. *SIAM journal on numerical analysis*, 35(4):1299–1314, 1998.

[17] J. Askew. A characteristics formulation of the neutron transport equation in complicated geometries. Technical report, United Kingdom Atomic Energy Authority, Winfrith, 1972.

[18] I. Babuška. The finite element method with penalty. *Math. Comp.*, 27:221–228, 1973. ISSN 0025-5718.

[19] I. Babuška and A.D. Miller. The post-processing approach in the finite element method i: calculations of displacements, stresses and other higher derivatives of the displacement. *Int. J. Numer. Meth. Engrg*, 20:1085–1109, 1984.

[20] I. Babuška and A.D. Miller. The post-processing approach in the finite element method ii: the calculation of stress intensity functions. *Int. J. Numer. Meth. Engrg*, 20:1111–1129, 1984.

[21] I. Babuška and A.D. Miller. The post-processing approach in the finite element method iii: a posteriori error estimation and adaptive mesh selection. *Int. J. Numer. Meth. Engrg*, 20:1085–1109, 1984.

[22] I. Babuška and W.C. Rheinboldt. Error estimates for adaptive finite element computations. *SIAM J. Numer. Anal.*, 15(4):736–754, 1978. ISSN 0036-1429.

[23] I. Babuška and W.C. Rheinboldt. A posteriori error estimates for the finite element method. *Int. J. Numer. Meth. Engrg*, 12:1597–1615, 1978.

[24] I. Babuska and M. Suri. The hp version of the finite element method with quasiuniform meshes. *Modélisation mathématique et analyse numérique*, 21(2):199–238, 1987.

[25] I. Babuška and M. Zlámal. Nonconforming elements in the finite element method with penalty. *SIAM J. Numer. Anal.*, 10:863–875, 1973. ISSN 0036-1429.

[26] D Baker. Characteristic methods for the neutron transport equation - progress report. Technical report, University of Nottingham, 2010.

[27] D. Baker. *Characteristic based methods for modelling neutron transport*. PhD thesis, University of Nottingham, 2012.

[28] Garth A Baker. Finite element methods for elliptic equations using nonconforming elements. *Mathematics of Computation*, 31(137):45–59, 1977.

[29] Wolfgang Bangerth and Rolf Rannacher. *Adaptive finite element methods for differential equations*. Springer, 2003.

[30] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *J. Comput. Phys.*, 131(2):267–279, 1997. ISSN 0021-9991.

[31] R. Becker and R. Rannacher. Weighted a posteriori error control in fe methods. In *Proceedings ENUMATH'97*. World Scientific, Singapore, 1995.

[32] R. Becker and R. Rannacher. A feed-back approach to error control in finite element methods: basic analysis and examples. *East-West J. Numer. Math.*, 4(4): 237–264, 1996. ISSN 0928-0200.

[33] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta numerica*, 10(1):1–102, 2001.

[34] G. Bell and S. Glasstone. *Nuclear Reactor Theory*. Van Nostrand Reinhold Company, 1st edition, 1970.

[35] G.I. Bell and W.B. Goad. Polarization effects on neutron transport. *Nuclear Science And Engineering*, 23:380–391, 1965.

[36] K.S. Bey and J.T. Oden. *hp*-version discontinuous Galerkin methods for hyperbolic conservation laws. *Comput. Methods Appl. Mech. Engrg.*, 133(3-4):259–286, 1996. ISSN 0045-7825.

[37] K.S. Bey, J.T. Oden, and A. Patra. A parallel *hp*-adaptive discontinuous Galerkin method for hyperbolic conservation laws. *Appl. Numer. Math.*, 20(4):321–336, 1996. ISSN 0168-9274. Adaptive mesh refinement methods for CFD applications (Atlanta, GA, 1994).

[38] B. Bollobás. *Modern graph theory*, volume 184. Springer, 1998.

[39] S.C. Brenner and K. Wang. Two-level additive schwarz preconditioners for c 0 interior penalty methods. *Numerische Mathematik*, 102(2):231–255, 2005.

[40] F. Brezzi, G. Manzini, D. Marini, P. Pietra, and A. Russo. Discontinuous Galerkin approximations for elliptic problems. *Numer. Methods Partial Differential Equations*, 16(4):365–378, 2000. ISSN 0749-159X.

[41] R.J. Brissenden. Ten lectures on neutron transport theory. Technical report, Advanced Methods Development Group, Reactor Pysics Division, AEE Winfrith, 1986.

[42] A.N. Brooks and T.J.R. Hughes. Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations. *Computer methods in applied mechanics and engineering*, 32(1):199–259, 1982.

[43] A. Buchan. *Adaptive Spherical Wavelets for the Angular Discretisation of the Boltzmann Transport Equation*. PhD thesis, Imperial College, University of London, 2006.

[44] A. Cangiani, E.H. Georgoulis, and P. Houston. hpÐversion discontinuous galerkin methods on polygonal and polyhedral meshes. *submitted for publication*, 2013.

[45] C. Cercignani. *The Boltzmann equation and its applications*, volume 67 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1988. ISBN 0-387-96637-4.

[46] J.C. Chai, H.S Lee, and S.V. Patankar. Ray effect and false scattering in the discrete ordinates method. *Numerical Heat Transfer, Part B Fundamentals*, 24(4):373–389, 1993.

[47] S. Chandrasekhar. *Radiative Transfer*. Oxford University Press, 1950.

[48] G. Chavent and B. Cockburn. The local projection P^0P^1-discontinuous-Galerkin finite element method for scalar conservation laws. *RAIRO Modél. Math. Anal. Numér.*, 23(4):565–592, 1989. ISSN 0764-583X.

[49] G. Chavent and G. Salzano. A finite-element method for the 1-d water flooding problem with gravity. *Journal of Computational Physics*, 45(3):307–344, 1982.

[50] K.A. Cliffe, E.J.C Hall, and P. Houston. Adaptive discontinuous galerkin methods for eigenvalue problems arising in incompressible fluid flows. *SIAM Journal on Scientific Computing*, 31(6):4607–4632, 2010.

[51] B. Cockburn and C. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework. *Math. Comp.*, 52(186):411–435, 1989. ISSN 0025-5718.

[52] B. Cockburn and C.W. Shu. The Runge-Kutta local projection P^1-discontinuous-Galerkin finite element method for scalar conservation laws. *RAIRO Modél. Math. Anal. Numér.*, 25(3):337–361, 1991. ISSN 0764-583X.

[53] B. Cockburn, S. Hou, and C. Shu. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. The multidimensional case. *Math. Comp.*, 54(190):545–581, 1990. ISSN 0025-5718. doi: 10.2307/2008501. URL `http://dx.doi.org/10.2307/2008501`.

[54] B. Cockburn, G.E. Karniadakis, and C. Shu. The development of discontinuous galerkin methods. In *Recent advances in scientific computing and partial differential equations (Hong Kong, 2002)*, volume 330 of *Contemp. Math.*, pages 89–119. Amer. Math. Soc., Providence, RI, 2003.

[55] B. Cockburn, M. Luskin, C. Shu, and E. Süli. Enhanced accuracy by post-processing for finite element methods for hyperbolic equations. *Mathematics of Computation*, 72(242):577–606, 2003.

[56] Bernardo Cockburn and Chi-Wang Shu. The Runge-Kutta discontinuous Galerkin method for conservation laws. V. Multidimensional systems. *J. Comput. Phys.*, 141(2):199–224, 1998. ISSN 0021-9991.

[57] Bernardo Cockburn and Chi-Wang Shu. The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM J. Numer. Anal.*, 35(6):2440–2463 (electronic), 1998. ISSN 0036-1429.

[58] Richard Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bull. Amer. Math. Soc*, 49(1):23, 1943.

[59] R. Dautray and J.L. Lions. *Mathematical analysis and numerical methods for science and technology. Vol. 6*. Springer-Verlag, Berlin, 1993. ISBN 3-540-50206-8; 3-540-66102-6.

[60] C.R.E. De Oliveira. An arbitrary geometry finite element method for multigroup neutron transport with anisotropic scattering. *Progress in Nuclear Energy*, 18(1): 227–236, 1986.

[61] C.R.E. de Oliveira and A.J.H Goddard. Event-a multidimensional finite element-spherical harmonics radiation transport code. *Three-dimensional deterministic radiation transport computer programs*, page 81, 1997.

[62] A. Dedner and P. Vollmöller. An adaptive higher order method for solving the radiation transport equation on unstructured grids. *J. Comput. Phys.*, 178:263–289, 2002.

[63] *Meeting the Energy Challenge: A White Paper on Nuclear Power*. Department for Business, Enterprise and Regulatory Reform, January 2008.

[64] J. Douglas and T. Dupont. Interior penalty procedures for elliptic and parabolic Galerkin methods. In *Computing methods in applied sciences (Second Internat. Sympos., Versailles, 1975)*, pages 207–216. Lecture Notes in Phys., Vol. 58. Springer, Berlin, 1976.

[65] D. Draess. *Finite Elements: theory, fast solvers and applications in solid mechanics*. CUP, Cambridge, UK, 1997.

[66] M. Dubiner. Spectral methods on triangles and other domains. *Journal of Scientific Computing*, 6(4):345–390, 1991.

[67] M. Dumbser. Arbitrary high order $p_n p_m$ schemes on unstructured meshes for the compressible navier–stokes equations. *Computers & Fluids*, 39(1):60–76, 2010.

[68] M. Dumbser and O. Zanotti. Very high order $p_n p_m$ schemes on unstructured meshes for the resistive relativistic mhd equations. *Journal of Computational Physics*, 228(18):6991–7006, 2009.

[69] M. Dumbser, M. Käser, and E.F. Toro. An arbitrary high-order discontinuous galerkin method for elastic waves on unstructured meshes–v. local time stepping and p-adaptivity. *Geophysical Journal International*, 171(2):695–717, 2007.

[70] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. Introduction to adaptive methods for differential equations. In *Acta numerica, 1995*, Acta Numer., pages 105–158. Cambridge Univ. Press, Cambridge, 1995.

[71] V. Faber and T.A. Manteuffel. A look at transport theory from the point of view of linear algebra. Technical report, Los Alamos National Lab., NM (USA), 1988.

[72] X. Feng and O.A. Karakashian. Two-level additive schwarz methods for a discontinuous galerkin approximation of second order elliptic problems. *SIAM Journal on Numerical Analysis*, 39(4):1343–1365, 2001.

[73] C. Führer and G. Kanschat. A posteriori error control in radiative transfer. *Computing*, 58:317–334, 1997.

[74] B.D. Ganapol. *Anaytical Benchmarks for Nuclear Engineering Applications : Case Studies in Neutron Transport Theory*. NEA Data Bank, 2008.

[75] E.M Gelbard. Simplified spherical harmonics equations and their use in shielding problems. technical report wapd-t-1182. Technical report, Bettis Atomic Power Laboratory, 1961.

[76] E.H. Georgoulis, E. Hall, and P. Houston. Discontinuous galerkin methods for advection-diffusion-reaction problems on anisotropically refined meshes. *SIAM Journal on Scientific Computing*, 30(1):246–271, 2007.

[77] E.H. Georgoulis, E. Hall, and P. Houston. Discontinuous galerkin methods on hp-anisotropic meshes ii: a posteriori error analysis and adaptivity. *Applied numerical mathematics*, 59(9):2179–2194, 2009.

[78] E.H.U.D. Greenspan. Developments in perturbation theory. *Advances in Nuclear Science and Technology*, 9:181–268, 1976.

[79] E.J.C. Hall. *Anisotropic Adaptive Refinement For Discontinuous Galerkin Methods*. PhD thesis, Department of mathematics, University of Leicester, 2007.

[80] EJC Hall and S Giani. Discontinuous galerkin methods for eigenvalue problems on anisotropic meshes. In *Numerical Mathematics and Advanced Applications 2011*, pages 351–359. Springer, 2013.

[81] G.E. Hansen and C. Maier. Perturbation theory of reactivity coefficients for fast-neutron critical systems. *Nuclear Sci. and Eng.*, 8, 1960.

[82] R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic conservation laws. *SIAM J. Sci. Comput.*, 24(3): 979–1004 (electronic), 2002. ISSN 1064-8275.

[83] R. Hartmann and P. Houston. Error estimation and adaptive mesh refinement for aerodynamic flows. In H. Deconinck, editor, *VKI LS 2010-01: 36th CFD/ADIGMA course on hp-adaptive and hp-multigrid methods, Oct. 26-30, 2009*. Von Karman Institute for Fluid Dynamics, Rhode Saint Genèse, Belgium, 2009.

[84] J.S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*, volume 54. Springer, 2008.

[85] P. Houston and E. Süli. hp-adaptive discontinuous galerkin finite element methods for first-order hyperbolic problems. *SIAM Journal on Scientific Computing*, 23 (4):1226–1252, 2001.

[86] P. Houston and E. Süli. A note on the design of *hp* adaptive finite element methods for elliptic partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 194(2):229–243, 2005.

[87] P. Houston, R. Rannacher, and E. Süli. A posteriori error analysis for stabilised finite element approximations of transport problems. *Computer methods in applied mechanics and engineering*, 190(11):1483–1508, 2000.

[88] P. Houston, C. Schwab, and E. Süli. Stabilized *hp*-finite element methods for first-order hyperbolic problems. *SIAM J. Numer. Anal.*, 37(5):1618–1643 (electronic), 2000. ISSN 0036-1429.

[89] P. Houston, Ch. Schwab, and E. Süli. Discontinuous hp-finite element methods for advection-diffusion-reaction problems. *SIAM Journal on Numerical Analysis*, 39(6):2133–2163, 2002.

[90] J.L. Hutton and N.R. Smith. Use of a hybrid monte carlo technique for power shape calculations. *MC2000, Lisbon, Portugal*, 2000.

[91] I.C.F. Ipsen. Computing an eigenvector with inverse iteration. *SIAM review*, 39 (2):254–291, 1997.

[92] C. Johnson and J. Pitkäranta. Convergence of a fully discrete scheme for two-dimensional neutron transport. *SIAM Journal on Numerical Analysis*, 20(5):951–966, 1983.

[93] C. Johnson and J. Pitkäranta. An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation. *Math. Comp.*, 46(173):1–26, 1986. ISSN 0025-5718.

[94] C. Johnson, U. Nävert, and J. Pitkäranta. Finite element methods for linear hyperbolic problems. *Comput. Methods Appl. Mech. Engrg.*, 45(1-3):285–312, 1984. ISSN 0045-7825.

[95] G. Kanschat. *Parallel and Adaptive Galerkin Methods for Radiative Transfer Problems*. PhD thesis, Ruprecht-Karls-Universität Heidelberg, 1996.

[96] M. Käser and M. Dumbser. An arbitrary high-order discontinuous galerkin method for elastic waves on unstructured meshes–i. the two-dimensional isotropic case with external source terms. *Geophysical Journal International*, 166 (2):855–877, 2006.

[97] R.C. Kirby. Algorithm 839: Fiat, a new paradigm for computing finite element basis functions. *ACM Transactions on Mathematical Software (TOMS)*, 30(4):502–516, 2004.

[98] EW Larsen and RE Alcouffe. Linear characteristic method for spatially discretizing the discrete ordinates equations in (x, y)-geometry. Technical report, Los Alamos Scientific Lab., NM (USA), 1981.

[99] E.W. Larsen, J.M. McGhee, and J.E. Morel. The simplified pn equations as an asymptotic limit of the transport equation. *Trans. Amer. Nucl. Soc.*, 66:427–430, 1992.

[100] P. LaSaint and P.A. Raviart. On a finite element method for solving the neutron transport equation. In *Mathematical aspects of finite elements in partial differential equations*, pages 89–145. Academic Press, 1974.

[101] K.D. Lathrop. Spatial differencing of the transport equation: positivity vs. accuracy. *Journal of computational physics*, 4(4):475–498, 1969.

[102] M. Lemanska. On the simplified pn method in the 2-d diffusion code exterminator. *Atomkernenergie*, 37:173–175, 1981.

[103] R.K. Lestner and R. Rosner. The growth of nuclear power - drivers and constraints. Technical report, Massachusetts Institute of Technology, 2009.

[104] R.J. LeVeque. *Finite volume methods for hyperbolic problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge, 2002. ISBN 0-521-81087-6; 0-521-00924-3.

[105] E.E. Lewis, C.B. Carrico, and G. Palmiotti. Variational nodal formulation for the spherical harmonics equations. *Nuclear science and engineering*, 122(2):194–203, 1996.

[106] E.E. Lewis, M.A. Smith, N. Tsoulfanidis, G. Palmiotti, T.A. Taiwo, and RN Blomquist. Benchmark specification for deterministic 2-d/3-d mox fuel

assembly transport calculations without spatial homogenisation (c5g7 mox). *NEA/NSC*, 2001.

[107] Q. Lin and A.H. Zhou. Convergence of the discontinuous Galerkin method for a scalar hyperbolic equation. *Acta Math. Sci. (English Ed.)*, 13(2):207–210, 1993. ISSN 0252-9602.

[108] J.L. Lions. Problèmes aux limites non homogénes à données irrégulières. In *Numerical Analysis of Partial Differential Equations*, pages 283–292. Springer, 2011.

[109] S. Maire and D. Talay. On a monte carlo method for neutron transport criticality computations. *IMA journal of numerical analysis*, 26(4):657–685, 2006.

[110] T. Manteuffel, S. McCormick, J. Morel, S. Oliveira, and G. Yang. A fast multigrid algorithm for isotropic transport problems i: Pure scattering. *SIAM Journal on Scientific Computing*, 16(3):601–635, 1995.

[111] J. Mark. The spherical harmonic method i. Technical report, National Research Council of Canada, Atomic Energy Project, 1944.

[112] J. Mark. The spherical harmonic method ii. Technical report, National Research Council of Canada, Atomic Energy Project, 1945.

[113] W Mitchell and Marjorie A McClain. A comparison of hp-adaptive strategies for elliptic partial differential equations, 2011. URL `http://math.nist.gov/~WMitchell/#pubs`.

[114] S. Murphy. *Parallel solvers for a higher order DG method for applications including neutron transport*. PhD thesis, University of Nottingham, to appear.

[115] C.R. Nastase and D.J. Mavriplis. High-order discontinuous galerkin methods using a spectral multigrid approach. *AIAA Paper*, 1268, 2005.

[116] Cristian R Nastase and Dimitri J Mavriplis. High-order discontinuous galerkin methods using an *hp* multigrid approach. *Journal of Computational Physics*, 213 (1):330–357, 2006.

[117] J. Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abh. Math. Sem. Univ. Hamburg*, 36:9–15, 1971. ISSN 0025-5858. Collection of articles dedicated to Lothar Collatz on his sixtieth birthday.

[118] P.F. Nowak, E.W. Larsen, and W.R. Martin. Multigrid methods for sn problems. *Transactions of the American Nuclear Society*, 57:355–356, 1987.

[119] P.F. Nowak, E.W. Larsen, and W.R. Martin. A multigrid method for sn calculations in xy geometry. *Transactions of the American Nuclear Society*, 56:291–292, 1988.

[120] O.A. Oleĭnik and E.V. Radkevič. *Second order equations with nonnegative characteristic form*. Plenum Press, New York, 1973. ISBN 0-306-30751-0. Translated from the Russian by Paul C. Fife.

[121] C. Pain, M. Eaton, R. Smedley-Stevenson, A. Goddard, M. Piggott, and C. De Oliveira. Space-time streamline upwind *p*etrov-*g*alerkin methods for the boltzmann transport equation. *Computer Methods in Applied Mechanics and Engineering*, 195:4334–4357, 2006.

[122] C. Pain, M. Eaton, R. Smedley-Stevenson, A. Goddard, M. Piggott, and C. De Oliveira. Streamline upwind *p*etrov-*g*alerkin methods for the steady-state boltzmann transport equation. *Computer Methods in Applied Mechanics and Engineering*, 195:4448–4472, 2006.

[123] G Palmiotti, EE Lewis, and CB Carrico. Variant: Variational anisotropic nodal transport. Technical report, ANL-95/40, October, 1995.

[124] T.E. Peterson. A note on the convergence of the discontinuous Galerkin method for a scalar hyperbolic equation. *SIAM J. Numer. Anal.*, 28(1):133–140, 1991. ISSN 0036-1429.

[125] J. Ragusa and Y. Wang. A two-mesh adaptive mesh refinement technique for sn neutral-particle transport using a higher-order dgfem. *Journal of computational and applied mathematics*, 233(12):3178–3188, 2010.

[126] W.H. Reed. The effectiveness of acceleration techniques for iterative methods in transport theory. *Nucl. Sci. Eng.*, 45, 1971.

[127] W.H. Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

[128] S. Richling, E. Meinköhn, N. Kryzhevoi, and G. Kanschat. Radiative transfer with finite elements - basic method and tests. *Astronomy and Astrophysics*, 380:776–778, 2001.

[129] G.R. Richter. An optimal-order error estimate for the discontinuous Galerkin method. *Math. Comp.*, 50(181):75–88, 1988. ISSN 0025-5718.

[130] G.R. Richter. The discontinuous Galerkin method with diffusion. *Math. Comp.*, 58(198):631–643, 1992. ISSN 0025-5718.

[131] B. Riemann. *Bernhard Riemann's Gesammelte Mathematische Werke*. Dover Publications, 1953.

[132] A. Santamarina, D. Bernard, Blaise P., Coste M., and Courcelle A. et al. *The JEFF 3.1.2 Nuclear Data Library*. OECD NEA, 2012.

[133] N. Sasamoto and K. Takeuchi. An improvement of the pallas discrete ordinates transport code. *Nucl. Sci. Eng.*, 71:330–342, 1979.

[134] F. Scheben. *Iterative Methods for Criticality Computations in Neutron Transport Theory*. PhD thesis, University of Bath, 2011.

[135] Ch. Schwab. *p- and hp-finite element methods*. Numerical Mathematics and Scientific Computation. The Clarendon Press Oxford University Press, New York, 1998. ISBN 0-19-850390-3. Theory and applications in solid and fluid mechanics.

[136] J.R. Shewchuk. Triangle: Engineering a 2d quality mesh generator and delaunay triangulator. In *Applied computational geometry towards geometric engineering*, pages 203–222. Springer, 1996.

[137] K. Shibata, O. Iwamoto, T. Nakagawa, N. Iwamoto, A. Ichihara, S Kuneida, S. Chiba, N. Furutaka, N. Otuka, and T. et al. Ohsawa. Jendl 4.0: A new library for nuclear science and engineering. *J. Nucl. Sci. Tech.*, 48:1–30, 2011.

[138] C. Shu. Total-variation-diminishing time discretizations. *SIAM J. Sci. Statist. Comput.*, 9(6):1073–1084, 1988. ISSN 0196-5204.

[139] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially nonoscillatory shock-capturing schemes. *J. Comput. Phys.*, 77(2):439–471, 1988. ISSN 0021-9991.

[140] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially nonoscillatory shock-capturing schemes. II. *J. Comput. Phys.*, 83(1):32–78, 1989. ISSN 0021-9991.

[141] K.S. Smith. Multidimensional nodal transport using the simplified pl method. *American Nuclear Society*, 52:427–430, 1986.

[142] P. Solin, K. Segeth, and I. Dolezel. *Higher-order finite element methods*. CRC Press, 2003.

[143] A. Sood, R.A. Forster, and Kent P.D. Analytical benchmark test set for criticality code verification. *Progress in Nuclear Energy*, 42(1):55–106, 2003.

[144] J. Spanier and E. Gelbard. *Monte Carlo Principles and Neutron Transport Problems*. Dover Publications, 2008.

[145] W.M. Stacey. *Nuclear Reactor Physics*. Wiley-VCH, 2nd edition, 2007.

[146] T. Strouboulis and J.T. Oden. A posteriori error estimation of finite element approximations in fluid mechanics. *Computer methods in applied mechanics and engineering*, 78(2):201–242, 1990.

[147] Endre Süli, Christoph Schwab, and Paul Houston. *hp*-DGFEM for partial differential equations with nonnegative characteristic form. In *Discontinuous Galerkin methods (Newport, RI, 1999)*, volume 11 of *Lect. Notes Comput. Sci. Eng.*, pages 221–230. Springer, Berlin, 2000.

[148] G Szegö. Orthogonal polynomials, ams colloq. *Publ*, 23:154, 1939.

[149] K. Takeuchi. A numerical method for solving the neutron transport equation in finite cylindrical geometry. *J. Nucl. Sci. Tech.*, 6:466–473, 1969.

[150] K. Takeuchi. Numerical solution to space-angle energy dependent neutron integral transport equation. *J. Nucl. Sci. Tech.*, 8:141–152, 1971.

[151] J.B. Taylor. *The development of a three-dimensional nuclear reactor kinetics methodology based on the method of characteristics*. PhD thesis, The Pennsylvania State University, 2007.

[152] E.F. Toro and V.A. Titarev. Solution of the generalized riemann problem for advection–reaction equations. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 458(2018):271–281, 2002.

[153] EF Toro, RC Millington, and LAM Nejad. Towards very high order godunov schemes. In *Godunov Methods*, pages 907–940. Springer, 2001.

[154] R. Verfürth. *A review of a posteriori error estimation and adaptive mesh refinement techniques*. Wiley - Teubner, 1996.

[155] Y. Wang. *Adaptive mesh refinement solution techniques for the multigroup SN transport equation using a higher-order discontinuous finite element method*. PhD thesis, Texas A & M University, 2009.

[156] Y. Wang and J. Ragusa. Standard and goal-oriented adaptive mesh refinement applied to radiation transport on 2d unstructured triangular meshes. *Journal of Computational Physics*, 230(3):763–788, 2011.

[157] Y. Wang, W. Bangerth, and J. Ragusa. Three-dimensional< i> h</i>-adaptivity for the multigroup neutron diffusion equations. *Progress in Nuclear Energy*, 51(3): 543–555, 2009.

[158] T. Warburton. *Spectral/hp methods on polymorphic multi-domains: algorithms and applications*. PhD thesis, Brown University, 1999.

[159] R.M. Westfall. Benchmark solutions for the infinite critical cylinder. Technical report, Oak Ridge National Lab., TN (USA), 1983.

[160] M. Wheeler. An elliptic collocation-finite element method with interior penalties. *SIAM J. Numer. Anal.*, 15(1):152–161, 1978. ISSN 0036-1429.

[161] G.C. Wick. Über eine diffusionsprobleme. *Zeitscrift für Physik A: Hadrons and Nuclei*, 121, November 1943.

[162] J.H. Wilkinson. *The algebraic eigenvalue problem*, volume 155. Oxford Univ Press, 1965.

[163] E. Woodcock, T. Murphy, P. Hemmings, and T. Longworth. Techniques used in the gem code for monte carlo neutronics calculations in reactors and other systems of complex geometry. In *Proc. Conf. Applications of Computing Methods to Reactor Problems*. 1965.

[164] *Supply of Uranium, Briefing Paper*. World Nuclear Association, 75 edition, 2008.

[165] M. Zhang and C. Shu. An analysis of three different formulations of the discontinuous galerkin method for diffusion equations. *Mathematical Models and Methods in Applied Sciences*, 13(03):395–413, 2003.

[166] OC Zienkiewicz. *Finite Element Method in Structural and Continuum Mechanics*. McGraw Hill, 1967.