Imperial College London

Spatial Goal-Based Error Estimation and Adaptive Mesh Refinement (AMR) for Diamond Difference Discrete Ordinate (DD-S_N) Methods

Department of Mechanical Engineering

Academic Supervisors:
Dr. Matthew Eaton
Dr. Michael Bluck

Industrial Supervisors:
Dr. Frank Hülsemann
Dr. François Févotte

Author:
Rebecca Siân Jeffers

Funded by EPSRC and EDF R&D
Submitted in part fulfilment of the requirements for the degree of Doctor of Engineering in Nuclear Engineering of Imperial College London and the Diploma of Imperial College London

January 2017

Abstract

A derivation of the dual weighted residual (DWR) goal-based error estimator is given for the 1-D DD-S_N discretisation of the fixed source and eigenvalue neutron transport equations. The quantity of interest (QoI) is linear and non-linear respectively. Isotropic scattering is assumed. Error correction and adaptive mesh refinement (AMR) were implemented for the 1-D code to reduce the error in the QoI as a function of the number of degrees of freedom (DoF) required for the forward solution. Higher order DD methods in the 1-D code allowed for h, p and hp AMR. Cell-wise DWRs were used to select cells for refinement. In the hp case, a merit function was used to choose between h or p refinement for a given cell.

The extension of the weighted residual (WR) view of the traditional 1-D DD-S_N equations to multidimensions results in a bilinear/trilinear approximation within the cell and cell vertex values as unknowns. However, traditional DD codes express the unknowns as cell-average and cell-edge average fluxes. The bilinear and trilinear components of the flux within a cell cannot be recovered from these values for use in the DWR calculation. Two 2-D codes were written, one keeping the cell-vertex values as unknowns and the other implementing the traditional DD scheme. In the former the DWR is calculated as in the 1-D case. The flux result of the latter is mapped into a discontinuous finite element space with a zero bilinear term before calculating the DWR. The vertex code provided the best error estimates. The DWR error estimate had, in general, the same convergence rate as the reference error, though the effectivity index tended towards unity only for fixed source problems with high scatter ratios. The equivalence between the vertex method and the traditional DD method means that the SP_N acceleration scheme used by EDF can still be applied.

Declaration of originality

I hereby declare that the work contained in this thesis is my own unless stated otherwise. In such cases the work will be appropriately referenced.

Copyright Declaration

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Acknowledgements

First and foremost I would like to thank my supervisors: Matthew Eaton, Mike Bluck, Frank Hülsemann and François Févotte. Thank you to the EPSRC and EDF R&D for jointly funding the project. I'm sure I will look back at these years as some of the best in my life.

I would like to thank Matthew Eaton for everything he has taught me throughout the doctorate, for generously funding my attendance on many courses and providing extensive reading material that was essential for the work to progress. The French courses in particular allowed my transition to life in Paris to go much more smoothly than it would have done otherwise. József Kópázi, despite being incredibly busy, always found the time to advise me on programming and neutron transport theory with immense rigour. Thank you especially for providing reference solutions for all the 2-D test cases used in the final part of the thesis work. I also thank Rolls Royce for allowing the use of their radiation transport code: RADIANT-II for this purpose. Thank you to Frank Hülsemann and François Févotte for their advice and guidance throughout my time at EDF R&D. The heavy fog that surrounded my functional analysis text books would not have cleared (or at least reduced!) without them. They also went out of their way to make me feel welcome. Thank you for helping to restore my confidence, caffeine and sugar supply. It was also my privilege to have interesting discussions with other academics including Mike Bluck (who was also my first IMechE mentor), M.M.R. Williams, Alan Copestake, Jean Ragusa, Richard Smedley-Stevenson, Ed Larsen and Anil Prinja.

Thank you for the continued friendship of my colleagues at Imperial College London, particularly Jo, Lavinia, Alex, James, Ben and Antonello. It doesn't get much better than working in an office full of friends. Thank you to Jo and Duncan for my French themed farewell party when I left the UK. You know your friends are engineers when your departure provides an excuse to construct a moving *Moulin Rouge* from lollipop sticks and a motor!

The friendly and welcoming atmosphere of I23 at EDF is the perfect environment for research and progress. I hope that many more students have the pleasure of working with this group. They should also be thanked for tolerating my school-girl French during coffee breaks. Thank you in particular to those with whom I share an office: Romain and Helin, for being friendly faces throughout the write-up and Alexandra who took a lot of time out of her day to help me with my CV.

Thank you to my long-distance friends: Dom, Jess, Holly, Kathryn, Chris, Raihan, Yusuf, Taf, Sita, Tillmann and Grace for your constant joy and support. Your friendship means a lot to me and the *hiraeth* is constant. To newer, more local friends, Jean, Maria and Elie, thank you for your endless support and friendship in a new city and for making sure I never go hungry at Fadi's! A special thanks to Chris and Jean for helping me with the practicalities of moving city. Felipe and Nicolò, my London housemates, thank you for 2 happy years in your company, you are the big brothers I never had. Thank you to the Masson family for looking after me for 4 months, your patience increased my French vocabulary and your cooking increased my waistline! Thank you to Barbara and Laura for welcoming me into the house-share in Paris and for your understanding in the final months of the write-up. Thank you to Mike Probyn for being my friend, IMechE mentor and for informing me of this EngD opportunity.

Finally, I would like to thank my family. My sister Lisa for keeping me in my place with blunt comments whilst still being fun to be around. My Nain (grandmother), Marina Owen, for never expecting thanks for hard work and for teaching me that one needs to have patience when fixing cars (it's good advice for bug hunting too). Ceinwen and Thomas Jeffers, my parents, have sacrificed a lot over the years for the happiness of my sister and myself. You are more like friends than parents and I owe you more than you could ever know. Diolch o galon i chi. I dedicate this thesis to the four of you and to the memory of family members no longer with us.

Dyfal donc a dyrr y garreg (Welsh proverb)

Contents

Abstract 1

Declaration of originality 2

Copyright Declaration 3

Acknowledgements 4

List of Figures 8

List of Tables 10

1 Introduction 11
- 1.1 EDF's activities in the UK nuclear industry . . . 12
- 1.2 The need for neutron flux models for nuclear reactors . . . 12
- 1.3 Reactor codes used by EDF R&D . . . 13
- 1.4 Sources of error in numerical simulations . . . 14
- 1.5 Project scope and objectives . . . 16
- 1.6 Thesis outline . . . 17

2 Theory and background 18
- 2.1 Reactor Physics . . . 18
  - 2.1.1 Material properties . . . 19
  - 2.1.2 Calculation of reaction rates . . . 21
  - 2.1.3 Neutron transport equation . . . 22
  - 2.1.4 Simplifications to the neutron transport equation . . . 24
- 2.2 Solving the neutron transport equation in practice . . . 25
  - 2.2.1 Stochastic methods . . . 25
  - 2.2.2 Deterministic methods . . . 26
- 2.3 Error estimation and AMR . . . 36
- 2.4 A summary of the test cases used throughout the thesis . . . 41

3 DWR error estimation and functional correction in 1-D slab geometry 43
- 3.1 Introduction . . . 44
- 3.2 Higher order 1D-DD discretisation of the already S_N angularly discretised transport equation as a WR method . . . 45
  - 3.2.1 Traditional DD discretisation of the single-ordinate problem . . . 46
  - 3.2.2 The higher-order DD single ordinate equations. . . . 49
  - 3.2.3 Multi-angle problems . . . 54

3.2.4 Eigenvalue problems . . . 56
3.3 DWR error estimation theory applied to the 1-D DD-S_N method . . . 58
3.3.1 General non-Bubnov-Galerkin case . . . 59
3.3.2 Fixed source problem with detector function goal . . . 62
3.3.3 Eigenvalue problem with eigenvalue goal . . . 63
3.3.4 Calculating the adjoint solution in practice . . . 66
3.4 QoI approximation improvement by DWR error correction . . . 68
3.5 Application to test cases . . . 69
3.5.1 Implementation . . . 69
3.5.2 Results . . . 70
3.6 Closing comments for this chapter . . . 80

4 DWR driven h, p and hp AMR in 1-D slab geometry 84
4.1 Introduction . . . 84
4.2 AMR strategy for h or p type mesh refinement . . . 85
4.3 Combined hp AMR strategy algorithm . . . 87
4.4 Application to test cases . . . 90
4.4.1 h, p and hp AMR . . . 90
4.4.2 Starting meshes and refinement strategies . . . 98
4.5 Closing comments for this chapter . . . 99

5 AMR and functional correction for linear functionals in 1-D slab geometry 106
5.1 Introduction . . . 106
5.2 A more appropriate error indicator to obtain an optimal mesh for the error corrected functional . . . 107
5.3 Application to test cases . . . 109
5.3.1 Implementation . . . 110
5.3.2 Results and discussion . . . 110
5.4 Closing comments for this chapter . . . 116

6 Multidimensional DWR error estimation 118
6.1 Introduction . . . 119
6.2 The box method . . . 119
6.3 2-D DD method . . . 123
6.4 The equivalence between the 2-D DD and box method . . . 126
6.4.1 Proof of DD and box method equivalence . . . 126
6.4.2 Practical conclusions that can be drawn from this equivalence . . . 129
6.5 DWR error estimator . . . 130
6.5.1 DWR for the box method . . . 130
6.5.2 DWR for the 2-D DD method . . . 131
6.6 Application to test cases . . . 132
6.6.1 Test case 7 . . . 132
6.6.2 Test case 8a . . . 136
6.6.3 Test case 8b . . . 136
6.6.4 Test case 9 . . . 139
6.6.5 Test case 10 . . . 141
6.6.6 Test case 11 . . . 141

6.7 Comment on 3-D DWR . . . 141
6.8 Comment on multi-dimension AMR . . . 144
6.9 Comment on multi-dimensional higher order DD . . . 145
6.10 Closing comments for this chapter . . . 147

7 Conclusions **149**
7.1 Chapter summaries . . . 149
7.1.1 DWR error estimation and functional correction in 1-D slab geometry . . . 149
7.1.2 DWR driven h, p or hp AMR in 1-D slab geometry . . . 150
7.1.3 AMR and functional correction for linear functionals in 1-D slab geometry 150
7.1.4 Multidimensional DWR error estimation . . . 151
7.2 Further work . . . 152

Bibliography **154**

Appendix **168**

A Test cases used throughout the thesis **168**
A.1 1D test cases . . . 168
A.1.1 Test case 1 . . . 168
A.1.2 Test case 2 . . . 169
A.1.3 Test case 3 . . . 169
A.1.4 Test case 4 . . . 170
A.1.5 Test case 5 . . . 171
A.1.6 Test case 6 . . . 171
A.1.7 Benchmarked solutions . . . 172
A.2 2D test cases . . . 172
A.2.1 Test case 7 . . . 173
A.2.2 Test case 8 . . . 174
A.2.3 Test case 9 . . . 175
A.2.4 Test case 10 . . . 175
A.2.5 Test case 11 . . . 175

B Adjoint equation definition and explanation **177**

C The inability to guarantee positive fluxes using the DD scheme in 2-D **180**
C.1 DD method . . . 180
C.2 Box method . . . 181

D Equivalence of DD1-Hébert scheme with a bi-quadratic box scheme **183**

List of Figures

1.1 Sources of error in results from computer programs modelling physical phenomena 15

2.1 Scattering event geometry 20
2.2 Scattering 21
2.3 Example of the ray effect 33

3.1 Example 2 cell domain and associated function spaces 49
3.2 Results for test case 1 72
3.3 Results for test case 2 73
3.4 Results for test case 3a 75
3.5 Results for test case 3b 76
3.6 The spatial variation of the error estimators, $(R_{\mathrm{h}}, \psi^{\dagger}_{approx})_{cell}$, as uniform refinement is performed for test case 3b with 64 cell starting mesh 77
3.7 The spatial variation of the adjoint solution for the smallest positive $|\mu|$ for the original mesh, one twice as refined and one with an increased polynomial order . 78
3.8 Results for test case 3c 79
3.9 Results for test case 4a 80
3.10 Results for test case 4b 81
3.11 Results for test case 5 82
3.12 Results for test case 6 82

4.1 One-type refinement algorithm flow chart 87
4.2 hp refinement algorithm flow chart 90
4.3 Results for test case 1 93
4.4 Results for test case 2 93
4.5 Results for test case 3 95
4.6 Results for test case 4 96
4.7 Test case 5 analytical goal error convergence results 97
4.8 Test case 6 analytical goal error convergence results 97
4.9 The absolute value of the goal error as a function of cell number. AMR refinement criterion = refine 50% of the cells at each mesh iteration (those with the highest error indicator values). 98
4.10 Convergence results for test case 3a 101
4.11 Results for test case 3b 102
4.12 Convergence results for test case 3c 103
4.13 Convergence results for test case 4a 104
4.14 Convergence results for test case 4b 105

5.1 Test case 1, Adaptivity with corrections (n: no error correction, l: low order error correction). . . . 112
5.2 Test case 2, Adaptivity with corrections (n: no error correction, l: low order error correction). . . . 114
5.3 Test case 3, adaptivity with corrections (n: no error correction, l: low order error correction). . . . 115
5.4 Test case 4, adaptivity with corrections (n: no error correction, l: low order error correction). . . . 117

6.1 Node numbering for box method and DD scheme 121
6.2 Test case 7a. Effectivity index as a function of the scatter ratio for a range of different error indicators and mesh iterations. . . . 134
6.3 Test case 7a. Convergence graphs. . . . 135
6.4 Test case 7b. Effectivity index as a function of the scatter ratio for a range of different error indicators and mesh iterations. . . . 137
6.5 Test case 7b. Convergence graphs. . . . 138
6.6 Test case 8a results. . . . 139
6.7 Test case 8b results. . . . 140
6.8 Test case 9 results. . . . 140
6.9 Test case 10 results. . . . 141
6.10 Test case 11 results. . . . 142
6.11 Node numberings for a rectangular prism 143
6.12 2-D hanging node example 144
6.13 Bi-quadratic cell 145

A.1 Reed cell problem scalar flux solution 170
A.2 EIRA-1-D problem 171
A.3 1-D-Mini-BWR geometry and material property definition. . . . 172
A.4 EIRA 2-A 2-D problem [1]. . . . 174
A.5 2-D-Mini-BWR geometry and material property definition. . . . 176

List of Tables

2.1 1-D test cases 41
2.2 2-D test cases. All reference solutions provided by RADIANT-II [2] 42

4.1 α values used for each test case. 91

5.1 Refinement criteria for the three different error indicators. 110

1. Introduction

This engineering doctorate is sponsored by the EPSRC and EDF R&D. The nuclear EngD programme is comprised of four elements that are to be completed over four years:

1. A doctoral level research project,
2. Post-graduate diploma in Enterprise Management,
3. Taught technical modules,
4. Professional development programme.

All technical modules, professional development courses and the post-graduate diploma have been completed. The final component is the doctoral thesis which is contained in this document.

The research work was carried out at Imperial College London, within Matthew Eaton's reactor physics group, and at EDF R&D (Clamart and Saclay) in SINETICS group "I23 - *analysis and numerical modelling*". As well as achieving an overall contribution to knowledge (a requirement of a PhD), an EngD needs to also recognise the "commercial drivers behind the research and the business impact (potential or real) of the results" [3]. Therefore, the first section (section 1.1) of this introductory chapter will give some background behind the current state of EDF activities in the UK nuclear industry. A brief explanation of the need for reliable neutron transport codes is given in section 1.2 and an overview of those used at EDF is given in section 1.3. The various sources of error in computer codes that are used to model real world physical phenomena are discussed in section 1.4. The scope of the project, how it has contributed to knowledge in the field and how it could contribute to EDF's business is described in section 1.5. Finally, section 1.6 will describe the thesis outline.

1.1 EDF's activities in the UK nuclear industry

The UK government is now committed to new nuclear as a means of replacing the generating capacity that will be lost as ageing nuclear power stations are taken off-line in the next decade [4]. New nuclear is also a way of providing low carbon electricity generation in the UK, which was one of the aims set out by the Energy Act 2013 [5].

About a fifth (21%) of the UK's electricity is currently produced by nuclear power [4]. All 15 remaining nuclear reactors in operation are owned and operated by EDF Energy [4], a subsidiary of The EDF Group. The UK's current reactor fleet comprises 14 advanced gas-cooled reactors (AGRs) and one pressurised water reactor (PWR). EDF plans to build new reactors of the European Pressurised Reactor (EPR) design in the UK. They are now able to proceed with the Hinkley Point C project since government approval was granted on September the 15th 2016 [6].

EDF has also been able to export electricity generated in France to the UK since 1985, via the cross-channel inter-connector. France's ability to export is mainly due to their large fleet of nuclear reactors generating about 80% of the country's electricity. The decision to go nuclear was taken after the 1973 oil crisis as a route to energy independence [7].

1.2 The need for neutron flux models for nuclear reactors

Nuclear reactors are large and complex systems involving multi-scale and multi-physics phenomena. It is necessary to predict the behaviour of reactors by using computational models in order to perform safety analysis for the commissioning of reactors, fuel loading preparation and real-world operating scenarios. Many quantities of engineering interest associated with safety analysis depend on the population and behaviour of the neutrons in the reactor. Codes that model neutron behaviour are therefore an essential element of reliably modelling the overall behaviour of nuclear reactors and power stations.

The source of power in a fission reactor is the release of energy upon the fission of large nuclei as they disintegrate to a more stable configuration of nucleons. The activation energy for the fission of a nucleus can be provided by neutron bombardment. A neutron incident on a nucleus can induce the fission of the nucleus. This results in daughter nuclei, energy release and neutron release. The neutrons released during fission can then go on to bombard further fissile nuclei in a self-sustaining chain reaction.

If the average number of neutrons being produced is equal to the average number being absorbed or leaked from the reactor system, the reactor is said to be critical and the power produced by the reactor will be constant. In this case, K_{eff}, the ratio of neutrons produced in one generation compared to the previous generation [8], will be equal to one. If the reactor power is to be increased or decreased, the reactor must go super-critical ($K_{eff} > 1$) or sub-critical ($K_{eff} < 1$) respectively. To model the criticality and fission rate distribution in the reactor for a given configuration, neutron transport codes are required. The more accurate and reliable the models, the easier it is to define appropriate safety factor margins for reactor construction and operation. This has potential safety and economic benefits.

1.3 Reactor codes used by EDF R&D

The reactor codes currently in use on nuclear sites for whole reactor core calculations need to provide answers on short time-scales. In order for the codes to run within reasonable timescales, approximations and simplifications are made to the problem in order to make it less computationally intensive to solve. COCCINELLE is a 2 group diffusion code that is used to calculate quantities at the scale of the whole nuclear reactor core [9] and for coupling with codes that model other physical phenomena (such as reactor core thermal hydraulics codes). This is a simplified model that assumes linear variation in angle by using the diffusion approximation to the neutron transport equation. Input material data that is homogenised over whole (or quarter) assemblies and broad energy ranges are obtained from calculations made on the scale of a single (or quarter) assembly where the spatial, energy and angular discretisation is more refined. These codes are often called *reactor lattice codes* or *lattice physics codes* [10]. The code used by EDF for this purpose is currently APOLLO2. The Monte Carlo code TRIPOLI is also used for reference calculations and radiation shielding type calculations [11].

The next generation of reactor physics solvers are currently being developed at EDF R&D and are part of the COCAGNE code platform. One branch of the COCAGNE code platform is a finite element transport-simplified P_N solver (SP_N) [12]. To ensure that the effect of the low order SP_N approximation on whole core calculation results is known, EDF require reference calculations for code validation of the full core SP_N solver. For this purpose, a diamond difference (DD) S_N solver: DOMINO [13] was developed. A high order S_N solver takes into account the angular variation of the flux in more detail than low order SP_N solvers and, unlike the SP_N equations, the S_N equations are guaranteed to converge to the transport solution with increased

N (assuming a converged spatial solution is obtained) [12, 14]. Part-validation of the S_N solver was made by comparing the results obtained on a 2D PWR assembly with the results of the MCNP Monte Carlo code [13]. Fine resolution results throughout a 3D nuclear reactor core is difficult to obtain with Monte Carlo codes, hence the motivation for a deterministic code for validation purposes [13].

1.4 Sources of error in numerical simulations

The result given by a computer code for a quantity of engineering interest will be associated with an error. The error comes from a variety of sources. In order to give an idea of the safety factors required in reactor design and predictions of reactor behaviour, a measure of the error is required. This section briefly discusses the different sources of error in the quantity of interest (QoI) prediction given by a piece of computing software and ways in which users can approximate the order of magnitude of the error.

Figure 1.1 is based on the discussion in section 2.1 of a book that discusses accuracy and reliability in scientific computing [15]. The figure summarises the sources of error that enter a given computer code, provided that there are no programming errors.

At the lowest level there are rounding errors that present themselves due to the use of floating point arithmetic on computer hardware. The error due to this can change as a function of computer hardware, compiler optimisations, use of parallelism and what precision is prescribed for the floating point numbers [15, 16]. Verification of the reliability of certain algorithms can be done by using stochastic arithmetic. This is a topic that is currently being researched by the analysis and numerical modelling group at EDF R&D [16].

The next level of error is where the continuous mathematical equations are modified so that they can be solved numerically by a computer. In deterministic schemes, continuous integro-differential equations (such as the neutron transport equation) are discretised. Code verification ensures that the discretisation error and the convergence rate upon uniform refinement are as expected for a given scheme. The method of manufactured solutions (MMS) is one way of conducting code verification [17]. Such analysis is more complicated when non-uniform adaptive mesh refinement (AMR) is conducted in order to reduce the number of unknowns for a given discretisation error, especially when the QoI is not the solution to the PDE itself but a functional of the solution. Dual weighted residual (DWR) error estimates are one way of providing an *a*

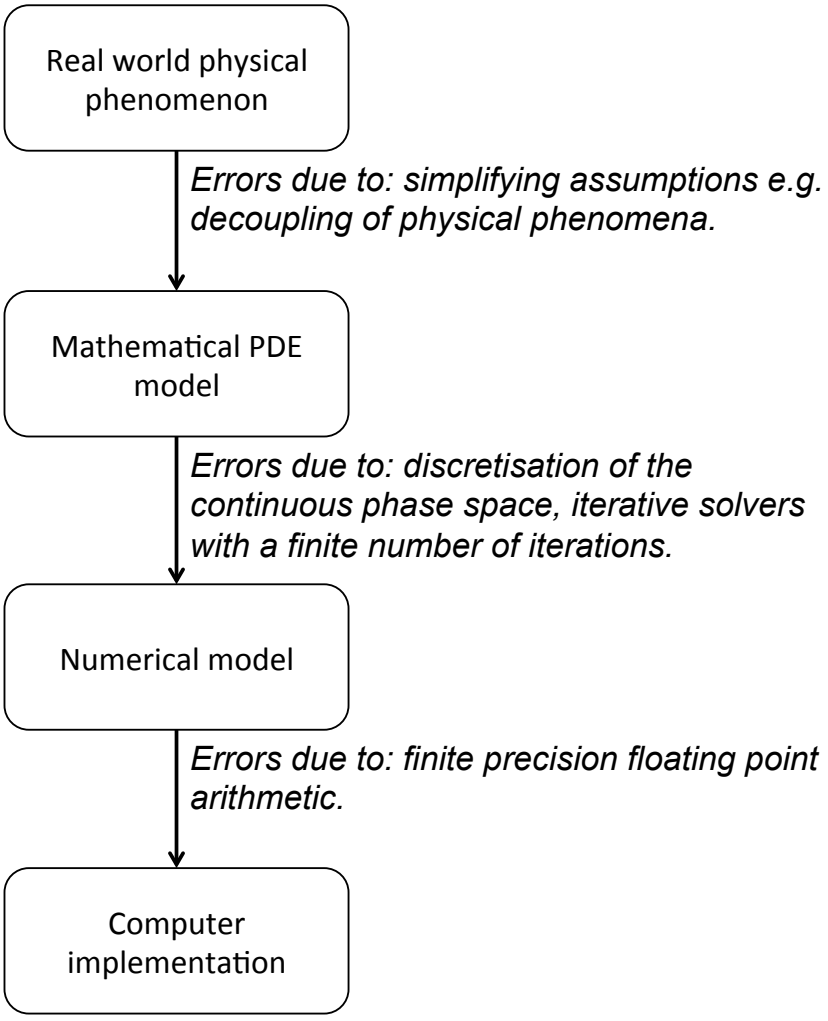


FIGURE 1.1: Sources of error in results from computer programs modelling physical phenomena

posteriori estimation of the error due to discretisation in a given QoI [18]. Another source of numerical error is in linear solvers where the numerically discretised equations are solved iteratively. An estimate of the error in a QoI due to the implementation of iterative solvers using a finite number of iterations can also be obtained by a DWR process [19, 20]. The error due to discretisation and iteration schemes can be reduced by using a larger number of iterations and by refining the discretisation scheme. However, at some point the amount of refinement will be limited by memory and execution time constraints.

The next level of error is in the assumptions made in the mathematical model, which is a simplification of the physical phenomena that occur in the real world scenario. For example, the population of neutrons in the reactor depends on many coupled physical phenomena such as the reactor temperature and the material properties of the reactor that change with time. In

neutron transport calculations, the assumption is made that the neutron transport phenomena being analysed occur at a time-scale that is much shorter than the time-scale of changing temperature or material properties in the reactor. If the reactor is assumed to be in a steady state, then discretisation in time is no longer needed. Another approximation may be to use the diffusion approximation to the neutron transport equation which removes the need for angular discretisation. Code validation must be conducted to ensure that the assumptions made in the construction of the mathematical model represent reality to a satisfactory degree of accuracy. This can be done by comparing the result given by a code to experimental measurements (if available). It can also be done by comparing simplified codes with more complex ones that make fewer assumptions. This is the aim of DOMINO in the COCAGNE code, as explained in section 1.3. The DOMINO solver validates that the low order SP_N approximation is appropriate by representing the angular variable via a large number of discrete ordinate directions. Another source of error compared to reality is uncertainties in the input data to the model (coefficients of the differential equation and boundary value data) [21, 22]. *Uncertainty quantification* due to uncertainty in parametric input data is an active area of research in neutron transport [23].

1.5 Project scope and objectives

The DOMINO solver can solve full core pressurised water reactor (PWR) eigenvalue (K_{eff}) problems with $\mathcal{O}(10^{12})$ degrees of freedom (DoF, number of unknowns) [24, 25]. These unknowns arise from discretising the six-dimensional phase space of the steady state angular flux. This six dimensional phase space consists of three spatial (x, y, z), two angular (θ, ω) and one energy dimension (E) [24, 25]. The aim of these whole core simulations is often to calculate the power distribution and reaction rates within the reactor core, as well as the K_{eff} of the reactor. Large numbers of DoF are required to ensure a low discretisation error (see section 1.4). The results are obtained at reasonable timescales due to making full use of nested levels of parallelism (multiple cores and vectorised operations) [25].

Estimating the error in a given QoI due to the discretisation of the phase space is essential for determining whether the discretisation is sufficiently refined. Deriving dual weighted residual (DWR) error estimators of the error in both linear and non-linear QoI due to the spatial discretisation of the fixed source and eigenvalue angularly discretised neutron transport equations is the first aim of this thesis.

The second aim of the thesis is to reduce the number of DoF required by only refining the spatial mesh where needed, for a specific QoI. It has been shown that adaptive mesh refinement (AMR) using error estimators that aim to reduce the global error in the forward solution can, in certain circumstances, fail to yield an accurate answer to the QoI compared to uniform refinement (see [26, p783] and [27, p306]). Heuristic error estimators are often inaccurate [28, p3], but there are examples of where heuristic error estimators for refinement can be improved by weighting with an adjoint solution [29, fig.6.7]. Therefore, reliable local goal-based error indicators should yield more robust approaches to mesh adaptivity that efficiently reduce the error in the QoI upon mesh refinement.

The above will be achieved by experimenting with my own code. EDF can then make an informed decision on whether to investigate this topic further for application to industrial reactor physics problems.

1.6 Thesis outline

The remaining 7 chapters are set out as follows. Chapter 2 covers the theory required in the rest of the thesis as well as giving a brief overview of the previous work that has been conducted on error estimation and AMR in the field of neutron transport. A summary of the test cases used in the thesis is also given. Chapter 3 derives the theory required for DWR error estimation of linear and non-linear quantities of interest for the DD-S_N discretised 1-D fixed source and eigenvalue neutron transport equations respectively. Use of this error estimator as an error correction term to obtain a better approximation of a QoI is also discussed and implemented. Chapter 4 uses the locally calculated cell values of the DWR estimator as error indicators for h, p or combined hp refinement of the DD scheme. Chapter 5 combines the advantages of chapter 3 and 4 by using a different error indicator for refinement in order to obtain the optimal mesh for the error corrected approximation to the QoI. Chapter 6 implements the necessary extensions required to implement the DWR theory to the multi-dimensional equations. Finally chapter 7 concludes the findings of the thesis and highlights how this research could be extended in the future.

2. Theory and background

This chapter will cover the basic theory required before undertaking the main subject material in the thesis. Section 2.1 gives some of the reactor physics theory required to obtain the governing mathematical equations for neutron transport. Section 2.2 gives an overview of the various methods used to obtain an approximate solution to the governing mathematical equations for neutron transport. Section 2.3 gives an overview of previous work that has been completed by the neutron transport community on AMR and error estimation. Section 2.4 gives a summary of the test cases that are used throughout this thesis.

2.1 Reactor Physics

The neutron *angular flux* is a distribution with respect to six different variables as well as a function of time [12, 30]:

$$\begin{aligned} &\Phi(\mathbf{r}, \mathbf{\Omega}, E, t) = \Phi(x, y, z, \theta, \psi, E, t) = vN(\mathbf{r}, \mathbf{\Omega}, E, t) \\ &\text{Units} = \text{cm}^{-2}\text{s}^{-1}\text{eV}^{-1}\text{sr}^{-1}. \end{aligned} \tag{2.1}$$

v (Units: cms^{-1}) represents the speed of the neutrons. The number density of neutrons ($N(\mathbf{r}, \mathbf{\Omega}, E, t)$, Units: $\text{cm}^{-3}\text{eV}^{-1}\text{sr}^{-1}$) is a distribution with respect to the six-dimensions of the phase-space defined by the spatial, angular and energy variables. $N(\mathbf{r}, \mathbf{\Omega}, E, t)$ is defined such that $N(\mathbf{r}, \mathbf{\Omega}, E, t)dEd\mathbf{\Omega}d\mathbf{r}$ is defined as the number of neutrons in the phase space volume $dEd\mathbf{\Omega}d\mathbf{r}$ at time t [12].

The value of the angular flux at any point in the six-dimensional phase space at a particular time t will depend on the size and composition of the reactor. Neutrons will be absorbed, be scattered, initiate fission and be produced by fission as a function of the material composition of

the reactor. Neutrons can also leak out of the reactor. For all these phenomena to be represented in a mathematical equation, a way of describing the nuclear properties of the reactor material and the resulting neutron reaction rates is required. This is explained in sections 2.1.1 and 2.1.2. How these values are used to give a representation of neutron behaviour in a reactor via the neutron transport equation is given in section 2.1.3. Some of the simplifications that can be made to the neutron transport equation in practice are discussed in section 2.1.4.

2.1.1 Material properties

The nuclear properties of isotopes are described by their *microscopic cross sections*, $\sigma_x(E)$, where x represents the specific type of neutron interaction under consideration [31]. This has units of area, and can be thought of as the effective cross-sectional area for neutron interaction with the nucleus. These can often be much larger than the actual surface area of the nucleus [32]. Cross sections vary with the type of nucleus, type of reaction, and the energy of the incident neutron.

There is a partial microscopic cross section associated with each possible neutron reaction, including the scattering microscopic cross section, $\sigma_{\mathrm{s}}(E)$, capture microscopic cross section, $\sigma_{\mathrm{c}}(E)$, and fission microscopic cross section, $\sigma_{\mathrm{f}}(E)$. The sum of the capture and fission microscopic cross sections can be represented by the absorption microscopic cross section $\sigma_{\mathrm{a}}(E)$ and the sum of all partial microscopic cross sections gives the total microscopic cross section $\sigma_{\mathrm{t}}(E)$ [8, 32].

By multiplying the microscopic cross sections with the number density of the particular isotope under consideration, one obtains the *macroscopic cross section*, $\Sigma_x(E, \mathbf{r})$ [31], which has dimensions L^{-1} and can be thought of as the effective area for interaction per unit volume or the probability of neutron interaction per unit distance travelled by the neutron [31]. It is a function of spatial coordinates since the number density of the isotope under consideration can vary in space. Similar to the microscopic cross section, the partial macroscopic cross sections for any nuclide present can be summed to provide the total macroscopic cross section [31]. Values of macroscopic cross section can also be given for a medium that is a mix of nuclides.

A *differential cross section* is also defined for reactions which produce neutrons [31]:

$$\Sigma_x(\mathbf{r}, E' \to E, \boldsymbol{\Omega}' \to \boldsymbol{\Omega}) = \Sigma_x(\mathbf{r}, E') f_x\left(\mathbf{r}, E' \to E, \boldsymbol{\Omega}' \to \boldsymbol{\Omega}\right), \tag{2.2}$$

$\Sigma_x(\mathbf{r}, E' \to E, \boldsymbol{\Omega}' \to \boldsymbol{\Omega})$ is the area for interaction per unit volume for an incident neutron at $\mathbf{r}$, of energy E' and direction $\boldsymbol{\Omega}'$ partaking in reaction x and in the process releasing a neutron

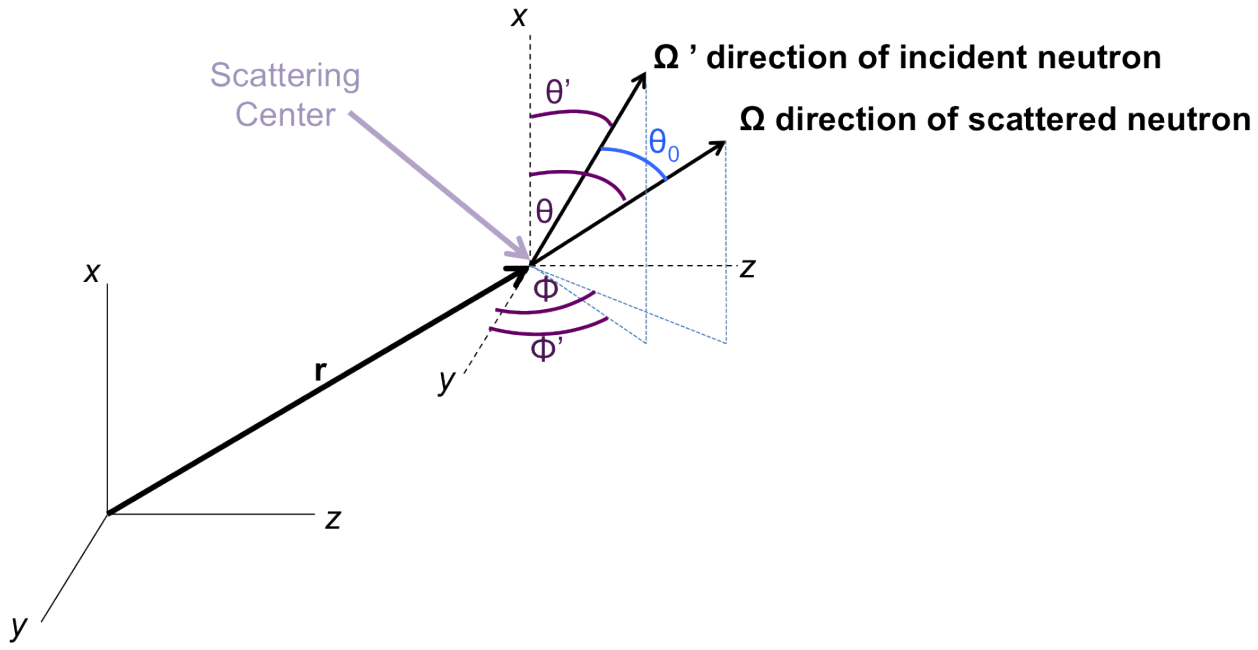


FIGURE 2.1: Scattering event geometry

to direction $\mathbf{\Omega}$ at energy E. $\Sigma_x(\mathbf{r}, E')$ is the area for interaction per unit volume for neutrons at position $\mathbf{r}$ and energy E'. $f_x\left(\mathbf{r}, E' \to E, \mathbf{\Omega}' \to \mathbf{\Omega}\right)$ is the probability density that this will result in a neutron being released with energy E and direction of travel $\mathbf{\Omega}$. If the reaction, x, under consideration is neutron scattering (in which case x is replaced by s), the directional dependence of f_s is a function of $\mathbf{\Omega}' \cdot \mathbf{\Omega}$ only [33], which is the cosine of the scattering angle, $\mu_0 = \cos\theta_0$, shown in figure 2.1. This indicates that the *material is isotropic*, i.e. there is no preferred direction in the material itself [34]. It also means that all scattering directions along the rim of the circle in figure 2.2 are of equal probability for the given incoming direction. Despite Kulikowska [31] showing f_s as a function of position and time, Ganapol [33] notes that it is generally independent of time and position, and that the spatial and temporal variation of the differential cross section is entirely due to the atomic density of the various nuclides in the medium which is included in the macroscopic scattering cross section. For scattering events, where one neutron interaction results in the release of only one neutron, f_s must be normalised in the following way [33]:

$$\int_0^\infty dE \int_{4\pi} d\mathbf{\Omega} f_s(\mathbf{\Omega}' \cdot \mathbf{\Omega}, E' \to E) \equiv 1. \tag{2.3}$$

The *mean free path* (mfp) of a neutron in a particular medium is a measure of the distance a

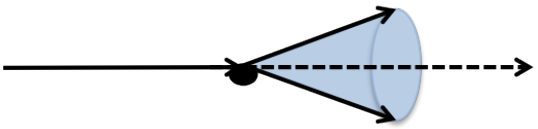


FIGURE 2.2: Scattering

neutron can travel in a straight line (assuming straight line travel between collisions) without interacting with the nuclei of the medium in any way. It is given by [31]:

$$\text{mfp} = \frac{1}{\Sigma_t(E, \mathbf{r})}. \tag{2.4}$$

The equation is derived in Lewis [8]. *Optically thick* media have short mean free paths, whereas those of *optically thin* media are large.

2.1.2 Calculation of reaction rates

Reaction rates can be calculated from the macroscopic cross section for the reaction and the neutron flux. The reaction rate as a result of neutrons travelling in a particular direction can be obtained by the angular flux as follows:

$$\text{Rate of reaction } x \text{ per unit volume due to neutrons travelling in direction } \mathbf{\Omega} = \Sigma_x \Phi. \tag{2.5}$$

Φ gives the number of neutrons incident on a unit area perpendicular to $\mathbf{\Omega}$ at a certain point in time, and Σ_x gives the area for interaction per unit volume. Given that the material can mostly be assumed isotropic, the cross section will be the same no matter which directions the neutrons are travelling. Hence, multiplying these two terms gives the number of reactions per unit volume due to neutrons travelling in this direction. Integrating over all directions simply results in the following equation, because Σ_x is independent of the incoming neutron angle:

$$\text{Rate of reaction } x \text{ per unit volume} = \int_{4\pi} \Sigma_x \Phi d\mathbf{\Omega} = \Sigma_x \phi, \tag{2.6}$$

where:

$$\phi(\mathbf{r}, E, t) = \int_{4\pi} \Phi(\mathbf{r}, \mathbf{\Omega}, E, t) d\mathbf{\Omega} \tag{2.7}$$

is the scalar flux and has units $\text{cm}^{-2}\text{s}^{-1}\text{eV}^{-1}$. It is the total number of neutrons per unit volume, with energy E (regardless of their direction), multiplied by their speed.

Terms of the kind expressed in equation (2.6) are important in the neutron transport equation given in section 2.1.3. As well as reactions, neutrons enter and leave a point in a phase space via transport. The rate at which neutrons leave a point in the phase space per unit phase space volume at time t is given by:

$$\boldsymbol{\Omega} \cdot \nabla \Phi \left(\mathbf{r}, E, \boldsymbol{\Omega}, t\right). \tag{2.8}$$

This is called the *streaming term*.

2.1.3 Neutron transport equation

The angular flux is the solution to the linear Boltzmann transport equation, also called the neutron transport equation, which can be written in its unsimplified form in equation (2.9) [30]:

$$\begin{aligned}
\frac{1}{v}\frac{\partial \Phi(\mathbf{r}, E, \boldsymbol{\Omega}, t)}{\partial t} = & -\left[\Sigma_a(\mathbf{r}, E) + \Sigma_s(\mathbf{r}, E)\right] \Phi(\mathbf{r}, E, \boldsymbol{\Omega}, t) \\
& - \boldsymbol{\Omega} \cdot \nabla \Phi(\mathbf{r}, E, \boldsymbol{\Omega}, t) \\
& + \int_{4\pi} d\boldsymbol{\Omega}' \int_0^\infty dE' \Sigma_s(E' \to E, \boldsymbol{\Omega}' \to \boldsymbol{\Omega}, \mathbf{r}) \Phi(\mathbf{r}, E', \boldsymbol{\Omega}', t) \\
& + \chi_p(E)(1-\beta)\frac{1}{4\pi}\int_0^\infty dE' \nu\Sigma_f(\mathbf{r}, E') \int_{4\pi} d\boldsymbol{\Omega}' \Phi(\mathbf{r}, E', \boldsymbol{\Omega}', t) \\
& + \frac{1}{4\pi}\sum_l \chi_l(E)\lambda_l C_l(\mathbf{r}, t) \\
& + Q(\mathbf{r}, E, \boldsymbol{\Omega}, t),
\end{aligned} \tag{2.9}$$

where, given the assumption that each neutron precursor (C_l: a fission product that releases a neutron after the initial fission has happened) emits one delayed neutron [32], the precursor concentration is given by:

$$\frac{\partial}{\partial t} C_l(\mathbf{r}, t) = \beta_l \frac{1}{4\pi}\int_0^\infty dE' \nu\Sigma_f(\mathbf{r}, E', t) \int_{4\pi} d\boldsymbol{\Omega}' \Phi(\mathbf{r}, E', \boldsymbol{\Omega}', t) - \lambda_l C_l(\mathbf{r}, t), \tag{2.10}$$

where:

$$\beta = \sum_l \beta_l. \tag{2.11}$$

Equation (2.9) is derived by applying conservation of neutrons to every possible control volume in the seven dimensional phase space. The term on the left hand side of equation (2.9) is the rate of increase of neutron density (neutrons per unit volume) with time of neutrons with energy E, travelling in a direction defined by the unit vector $\boldsymbol{\Omega}$ at point $\mathbf{r}$. The first term on the right

hand side is the removal of neutrons due to neutron interaction with the reactor material. The second is the removal of neutrons due to transport. The third term on the right hand side is the addition of neutrons due to those scattered from different directions and energies to energy E and direction $\mathbf{\Omega}$. The fourth term on the right hand side is the addition of neutrons due to prompt fission neutrons, released isotropically at the instant of fission. The fifth term is due to delayed neutrons which are released isotropically when the fission products decay. The factor of $\frac{1}{4\pi}$ in the fourth and fifth term is due to the fact that neutrons released as a result of fission are assumed to be emitted isotropically. The final term on the right hand side is an extraneous (fixed) source of neutrons.

The fourth and fifth terms on the right hand side of equation (2.9) introduce new symbols. β is the fraction of fission neutrons that are delayed. ν is the average number of neutrons released per fission. $\chi_p(E)$ and $\chi_l(E)$ are the prompt and delayed neutron fission spectra respectively. $\chi_p(E)$ gives the proportion of prompt neutrons produced with energy E. Similarly $\chi_l(E)$ gives the proportion of delayed neutrons produced as a result of neutron precursor l, with energy E. They must be normalised as follows:

$$\int_0^\infty \chi_p(\mathbf{r}, E, t)dE \equiv 1 \tag{2.12}$$

and

$$\int_0^\infty \chi_l(\mathbf{r}, E, t)dE \equiv 1 \tag{2.13}$$

and they are both assumed independent of the isotope being fissioned.

Neutron precursors are grouped according to their half-life [32]. $C_l(\mathbf{r}, t)$ is the number of neutron pre-cursor nuclei in neutron pre-cursor group l per unit volume, λ_l is the delayed neutron pre-cursor decay coefficient for precursors of group l.

Even in the above rather complex set of equations, simplifying assumptions have been made. The effect of fission and neutron bombardment on the material macroscopic neutron cross sections is assumed to be independent of time (occur at time-scales much slower than those considered in equation (2.9)). Also, the fission spectra ($\chi_p(E)$ and $\chi_l(E)$) and proportion of prompt and delayed neutrons are assumed the same for all fissile nuclei undergoing fission.

2.1.4 Simplifications to the neutron transport equation

The neutron transport equation cannot be solved analytically because of its complexity, therefore there is the requirement to either solve it numerically, or simplify it. For time dependent problems it must be discretised in time, and stability issues become important depending on which discretisation scheme is used. An extensive explanation will not be given here, since the application in this project will be to steady state problems.

Assuming steady state conditions simplifies the problem immediately:

$$\begin{aligned}[\Sigma_a(\mathbf{r},E)+\Sigma_s(\mathbf{r},E)]\,\Phi(\mathbf{r},E,\mathbf{\Omega})+\mathbf{\Omega}\cdot\nabla\Phi(\mathbf{r},E,\mathbf{\Omega}) = \\ + \int_{4\pi} d\mathbf{\Omega}' \int_0^\infty dE' \Sigma_s(E'\to E, \mathbf{\Omega}'\to\mathbf{\Omega},\mathbf{r})\Phi(\mathbf{r},E',\mathbf{\Omega}') \\ + \frac{1}{4\pi}\chi(\mathbf{r},E)\int_0^\infty dE'\nu\Sigma_f(\mathbf{r},E')\int_{4\pi} d\mathbf{\Omega}'\Phi(\mathbf{r},E',\mathbf{\Omega}') \\ + Q(\mathbf{r},E,\mathbf{\Omega}). \end{aligned} \tag{2.14}$$

For cases where the external source, Q, is zero, the resulting equation is only valid if the system under consideration is critical. In the case where the reactor is not critical, the equation needs to be modified to obtain a well-posed problem. One way of doing this is by dividing the fission source by a factor $\mathrm{K_{eff}}$ to enforce the steady-state condition. $\mathrm{K_{eff}}$ is equal to the neutrons produced divided by the number that are absorbed or leaked from the core in each generation. The system being analysed is supercritical, critical or subcritical if $\mathrm{K_{eff}}$ is greater than, equal to or less than one respectively. This transforms the equation into a generalised eigenvalue problem. Alternatively, a fixed source problem is one where there is no fission but there is an extraneous source of neutrons, Q.

An alternative to the $\mathrm{K_{eff}}$ eigenvalue method is the α-eigenvalue method, where steady state is *not* assumed. The flux variation with time, however, is assumed independent of the other parameters and assumed to be of the form $e^{\alpha t}$. Substituting this into equation (2.9), and assuming no delayed neutrons, also results in an eigenvalue equation where α is the eigenvalue and the time-independent part of the flux is the eigenfunction. The α term appears in an absorption term, unlike $\mathrm{K_{eff}}$ which appears in the source term. The system being analysed is critical (steady state) if α is equal to zero. In this case the $\mathrm{K_{eff}}$ and α-eigenvalue equations are equivalent. The equations are not equivalent when the system is supercritical ($\alpha > 0$) or subcritical ($\alpha < 0$).

In either the steady state fixed source or eigenvalue case, analytical solutions only exist for simple problems. Hence, the equation must be solved approximately using stochastic or deterministic methods. The different methods are discussed in the following section.

2.2 Solving the neutron transport equation in practice

Methods for obtaining an approximate solution to the neutron transport equation can be divided into two families of methods: stochastic methods and deterministic methods. The method used in this thesis is a particular type of deterministic method. A brief description of the methods currently available is given in section 2.2.1 for stochastic methods and section 2.2.2 for deterministic methods.

2.2.1 Stochastic methods

This usually means applying a Monte Carlo method to solve the neutron transport equation. MCNP and TRIPOLI are examples of Monte Carlo codes [12]. The Monte Carlo method was first developed for the nuclear industry around the same time as the deterministic S_N method at Los Alamos National Laboratory [35]. The interaction of neutrons with materials is a statistical process [36], where the *probabilistic events* governing the the overall *process* of neutron transport consist of collisions with varying different outcomes, such as scatter, fission, capture, or if incident on a boundary, leakage or reflection. Each of the possible events have an associated probability distribution which governs them. These distributions are statistically sampled in the Monte Carlo method and many particles are followed from their source through to the "death" of the particle by absorption or leakage [36]. The source of a neutron could be an extraneous source, or produced in the process of following another neutron [36]. Many "neutron histories" [36] are found and after following many of them, the neutron distribution becomes better known.

There are advantages to using Monte Carlo techniques over deterministic methods. Ray effects are not an issue as they are in the S_N method (see section 2.2.2.2). The spatial domain can be modelled exactly because there is no need for a mesh. The energy dependence of nuclear properties are modelled continuously [36]. The continuous energy spectrum is sampled without the need for a multi group approximation [12]. There is, however, the need for "bins" (in energy, space and angle) in which neutron tallies are taken for the specific quantity of interest being calculated [36]. The method therefore gives integral quantities. The number of DoF required to

accurately model complex 3-D problems is large when the total phase space is discretised such as in the S_N or P_N method. However, for an accurate statistical representation, many independent samples must be taken [37], which is also computationally expensive. In fact, the error in the solution of interest is proportional to $\frac{1}{\sqrt{N}}$ where N is the number of particle histories taken [38]. The uncertainty is increased for quantities for which only a few of the particle histories (N) contribute to the overall tally (such as localised flux values, where small "bins" are used) compared to K_{eff} or integral values over larger volumes where a larger proportion of particle histories contributes to the tally of the quantity of interest [32].

Monte Carlo methods are appropriate for the calculation of integral quantities in complex geometry reactors. However, to obtain information about the flux (or other quantities) as a function of space and energy, the Monte Carlo method is inefficient and prone to inaccuracies [39].

For full 3-D geometries, the number of neutron histories required to produce an accurate and spatially detailed solution to the flux is significantly higher than a 2-D case [13]. It is for this reason that EDF have decided to use a 3-D S_N solver to provide reference solutions for validation purposes rather than a Monte Carlo code [13].

2.2.2 Deterministic methods

All methods involve the discretisation of a six-dimensional phase space (for steady state problems). A given combination of different discretisation techniques for each parameter in the phase space gives a specific type of deterministic method. The type of discretisation for energy, angle and space used in various deterministic methods are briefly discussed in sections 2.2.2.1, 2.2.2.2 and 2.2.2.3 respectively.

2.2.2.1 Discretisation in energy

The microscopic cross sections of atoms are functions of the incident neutron energy, hence the energy spectrum must be discretised in a deterministic code. In deterministic codes the energy domain is discretised using the multi-group method. In this method the energy domain, from 0 to a prescribed upper bound (e.g. 10MeV [40]), is discretised into G discrete energy regions, often called energy groups. Normally the energy groups are numbered with integers from G to 1, with energy group G being of the lowest energy and group 1 being of the highest energy

[12, 32, 40]. The integrated behaviour of the neutrons in each energy group is calculated and the energy dependent input parameters are averaged over each energy group.

To transform equation (2.14) into a multi-group equation, the equation is integrated over the energy width (ΔE_g) of an arbitrary group g. This is done for all groups, to produce G algebraically coupled equations. The integrals over the whole energy domain are replaced by the sums of the integrals over the individual groups [32]:

$$\begin{aligned}\int_g dE\Sigma_t(\mathbf{r},E)\Phi(\mathbf{r},E,\mathbf{\Omega})+\mathbf{\Omega}\cdot\nabla\int_g dE\Phi(\mathbf{r},E,\mathbf{\Omega}) = \\ + \int_g dE\int_{4\pi} d\mathbf{\Omega}'\sum_{g'=1}^{G}\int_g dE'\Sigma_s(E'\to E,\mathbf{\Omega}'\to\mathbf{\Omega},\mathbf{r})\Phi(\mathbf{r},E',\mathbf{\Omega}') \\ + \int_g dE\frac{1}{4\pi}\chi(\mathbf{r},E)\sum_{g'=1}^{G}\int_{g'} dE'\nu\Sigma_f(\mathbf{r},E')\phi(\mathbf{r},E') \\ + \int_g dEQ(\mathbf{r},E,\mathbf{\Omega}) \\ \text{for } g=1,2,\dots G. \end{aligned} \tag{2.15}$$

Note that in the above equation $\int_g dE = \int_{g_{min}}^{g_{max}} dE$.

Equation 2.15 can be written in terms of *group angular fluxes*, *group scalar fluxes*, *group external sources* and *group constant cross sections* using the following definitions [41]:

$$\phi_g(\mathbf{r}) = \int_g dE\phi(\mathbf{r},E), \tag{2.16}$$

$$\Phi_g(\mathbf{r},\mathbf{\Omega}) = \int_g dE\Phi(\mathbf{r},E,\mathbf{\Omega}), \tag{2.17}$$

$$Q_g(\mathbf{r},\mathbf{\Omega}) = \int_g dEQ(\mathbf{r},E,\mathbf{\Omega}), \tag{2.18}$$

$$\chi_g(\mathbf{r}) = \int_g dE\chi(\mathbf{r},E). \tag{2.19}$$

Group-constant cross sections and differential cross sections are denoted by the following symbols: $\Sigma_{t,g}(\mathbf{r})$, $\Sigma_{s,g'\to g}(\mathbf{r},\mathbf{\Omega}'\to\mathbf{\Omega})$ and $\nu\Sigma_{f,g}(\mathbf{r})$. They represent the averaged behaviour within a given energy group. The evaluation of the multi-group cross sections can be quite involved due to the complicated variation of the microscopic cross sections with energy. The variation can include a resonance region where the cross section varies significantly within a very short range of neutron energies [12]. The process of calculating multi-group cross sections that take

these aspects into account is therefore a complex process that is outside the scope of this thesis. Nuclear data files such as ENDF or JEFF are processed by a cross section processing code such as NJOY [12] to give multi-group cross sections. Lattice calculations are then conducted to homogenise the material cross sections over regions in the reactor and for reducing the number of energy groups required for whole core calculations. In this work the energy group cross sections are assumed to be known inputs to the neutron transport code.

Substituting in for all the group-constant cross sections and equations (2.16)-(2.19) results in the multi-group equations:

$$\begin{aligned}\Sigma_{t,g}(\mathbf{r})\Phi_g(\mathbf{r},\mathbf{\Omega}) + \mathbf{\Omega}\cdot\nabla\Phi_g(\mathbf{r},\mathbf{\Omega}) = &\int_{4\pi} d\mathbf{\Omega}' \sum_{g'=1}^{G} \Sigma_{s,g'\to g}(\mathbf{r},\mathbf{\Omega}'\to\mathbf{\Omega})\Phi_{g'}(\mathbf{r},\mathbf{\Omega}) \\ &+ \frac{1}{4\pi}\chi_g(\mathbf{r}) \sum_{g'=1}^{G} \nu\Sigma_{f,g'}(\mathbf{r})\phi_{g'}(\mathbf{r}) \\ &+ Q_g(\mathbf{r},\mathbf{\Omega}) \\ \text{for } g = 1,2,\ldots G.&\end{aligned} \tag{2.20}$$

Isotropic scattering is assumed in this project. Because of this, the first term on the right-hand side can be simplified to give the following equation:

$$\Sigma_{t,g}(\mathbf{r})\Phi_g(\mathbf{r},\mathbf{\Omega}) + \mathbf{\Omega}\cdot\nabla\Phi_g(\mathbf{r},\mathbf{\Omega}) = \frac{1}{4\pi}\sum_{g'=1}^{G}\Sigma_{s0,g'\to g}(\mathbf{r})\phi_{g'}(\mathbf{r}) + \frac{1}{4\pi}\chi_g(\mathbf{r})\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\mathbf{r})\phi_{g'}(\mathbf{r}) + Q_g(\mathbf{r},\mathbf{\Omega})$$
$$\text{for } g = 1,2,\ldots G, \tag{2.21}$$

where $\Sigma_{s0,g'\to g}(\mathbf{r})$ represents a group isotropic scattering coefficient.

A fixed source multi-group problem, without the presence of a neutron multiplying material is denoted as follows:

$$\Sigma_{t,g}(\mathbf{r})\Phi_g(\mathbf{r},\mathbf{\Omega}) + \mathbf{\Omega}\cdot\nabla\Phi_g(\mathbf{r},\mathbf{\Omega}) = \frac{1}{4\pi}\sum_{g'=1}^{G}\Sigma_{s0,g'\to g}(\mathbf{r})\phi_{g'}(\mathbf{r}) + Q_g(\mathbf{r},\mathbf{\Omega}) \quad \text{for } g = 1,2,\ldots G. \tag{2.22}$$

In practical eigenvalue solvers such as DOMINO, a multi-group problem is solved at each eigenvalue iteration. The fission source is fixed for each iteration and represented by a fixed source term based on the flux of the previous iteration.

Given that the group scalar flux is a function of the group angular fluxes, it is clear that equation (2.22) needs to be solved iteratively as well. If an initial guess for each group scalar flux is given, and each equation from $g = 1$ to $g = G$ is solved, one can then integrate these group angular fluxes to produce new group scalar fluxes. These can then be used as the source for G equations of the next multi group iteration. Iterations continue until the result is deemed to have converged. Solving the problem in this way is called a Jacobi iteration [42]. Defining l as the iteration number, this scheme can be written as follows:

$$\Sigma_{t,g}(\mathbf{r})\Phi_g^l(\mathbf{r},\mathbf{\Omega}) + \mathbf{\Omega}\cdot\nabla\Phi_g^l(\mathbf{r},\mathbf{\Omega}) = \frac{1}{4\pi}\sum_{g'=1}^{G}\Sigma_{s0,g'\to g}(\mathbf{r})\phi_g^{l-1}(\mathbf{r}) + Q_g(\mathbf{r},\mathbf{\Omega}) \quad \text{for } g = 1,2,\dots G. \tag{2.23}$$

Faster convergence can be achieved by splitting the sum in equation (2.23) so that for each equation, g, the scalar flux from the current multi group iteration is used for $g' < g$. This is called a Gauss-Seidel iteration [42]. It can be represented by equation (2.24):

$$\Sigma_{t,g}(\mathbf{r})\Phi_g^l(\mathbf{r},\mathbf{\Omega}) + \mathbf{\Omega}\cdot\nabla\Phi_g^l(\mathbf{r},\mathbf{\Omega}) = \frac{1}{4\pi}\sum_{g'=g}^{G}\Sigma_{s0,g'\to g}(\mathbf{r})\phi_g^{l-1}(\mathbf{r}) + \frac{1}{4\pi}\sum_{g'=1}^{g-1}\Sigma_{s0,g'\to g}(\mathbf{r})\phi_g^{l}(\mathbf{r}) + Q_g(\mathbf{r},\mathbf{\Omega}) \quad \text{for } g = 1,2,\dots G. \tag{2.24}$$

In sequential implementations of codes (e.g. the sequential implementation of [43]) it is the Gauss-Seidel scheme that is used because of its obvious advantages. However, the Jacobi iteration is used in parallel implementations e.g. [43, 44]. In these cases it is assumed that the increased efficiency offered by parallelising the code makes up for the increased number of iterations required for a converged answer.

Each g equation in equation (2.24) is a mono-energetic problem with the in-scatter from other energy groups being represented as a fixed source for that problem. Hence, in a practical code a multi-group solver will call a fixed source solve for each group. For each group, the fixed source is defined as the sum of incoming neutrons from all groups other than g, and any external sources.

2.2.2.2 Discretisation in angle

The angular variation of the flux can be represented by many different procedures, including the P_N, SP_N and S_N methods.

The spherical harmonic method (P_N) is based on the expansion of the angular flux and source terms in terms of spherical harmonics. The SP_N equations are a simplified version of the P_N equations that have less complicated couplings between unknowns in multi-dimensions [12]. The spherical harmonic expansion of the P_N method reduces to a Legendre polynomial expansion in 1-D problems. The multidimensional SP_N method is then developed from the 1-D P_N equations by replacing the 1-D differential term by either a divergence or gradient operator. The original 1-D equations obtained by weighting by the even Legendre Polynomial basis functions use the divergence operator, those obtained by weighting with the odd Legendre Polynomial basis functions use the gradient operator [12].

The P_1, SP_1 and diffusion approximations are equivalent if the source term is isotropic [12]. As mentioned in section 1.3, the SP_N scheme is used in the COCAGNE code. Another branch of the COCAGNE code (DOMINO) uses the S_N method, and it is this branch that is of interest to the work of this thesis.

S_N methods are also called discrete ordinate methods. They discretise the angular domain into a finite number of angular directions and evaluate the neutron transport equation along each ordinate. The directions are chosen using a quadrature scheme (for example, the Level Symmetric quadrature scheme [32]) so that the scalar flux (and higher flux moments) can be obtained by a weighted sum over all flux ordinates. The S_N method was first developed at the same time as the Monte Carlo method at Los Alamos National Laboratory [35]. The name comes from the "angular segmentation" method developed by Carlson [45], from which the discrete ordinate method evolved [35].

To illustrate the idea of the S_N method, S_N equations will be shown without discretising in space and assuming a fixed source 1 group problem. This is in an attempt to not over-complicate the presentation. When level symmetric quadrature is used, the angular flux is computed at a number of discrete angles, which have N discrete and distinct values of μ (the azimuthal angular cosine). The total number of ordinates required for a 3-D S_N approximation is then [32]:

$$N(N+2). \tag{2.25}$$

The following notation will be adopted:

$$\Phi(\mathbf{r}, \mathbf{\Omega}_m) = \Phi_m(\mathbf{r}). \tag{2.26}$$

The mono-energetic equation can then be re-written in its discretised form:

$$\left[\Sigma_t(\mathbf{r}) + \mathbf{\Omega}_m \cdot \nabla\right] \Phi_m(\mathbf{r}) = \sum_{m'=1}^{N(N+2)} \frac{\pi}{2} w_{m'} \Sigma_s(\mathbf{\Omega}_{m'} \cdot \mathbf{\Omega}_m, \mathbf{r}) \Phi_{m'}(\mathbf{r}) + Q_m(\mathbf{r}) \quad \text{for} \quad m = 1, 2, \ldots, N(N+2), \tag{2.27}$$

where:

$$\int_{4\pi} \Sigma_s(\mathbf{\Omega}_{m'} \cdot \mathbf{\Omega}_m, \mathbf{r}) \Phi_{m'}(\mathbf{r}) d\mathbf{\Omega}' \approx \sum_{m'=1}^{N(N+2)} \frac{\pi}{2} w_{m'} \Sigma_s(\mathbf{\Omega}_{m'} \cdot \mathbf{\Omega}_m, \mathbf{r}) \Phi_{m'}(\mathbf{r}). \tag{2.28}$$

The factor of $\frac{\pi}{2}$ in the scattering source is due to the weights, w_m, being normalised such that they sum to 1 over one octant. In the 3-D case there are 8 octants in total.

The streaming operator can be separated into the different coordinates, and the equation re-written as:

$$\begin{aligned}\left[\Sigma_t(x,y,z) + \mu_m \frac{\partial}{\partial x} + \eta_m \frac{\partial}{\partial y} + \xi_m \frac{\partial}{\partial z}\right] \Phi_m(x,y,z) \\ = \sum_{m'=1}^{N(N+2)} \frac{\pi}{2} w_{m'} \Sigma_s(\mathbf{\Omega}_{m'} \cdot \mathbf{\Omega}_m, x, y, z) \Phi_{m'}(x,y,z) + Q_m(x,y,z) \\ \text{for} \quad m = 1, 2, \ldots, N(N+2).\end{aligned} \tag{2.29}$$

For isotropic scattering, the equation is simplified to:

$$\begin{aligned}\left[\Sigma_t(x,y,z) + \mu_m \frac{\partial}{\partial x} + \eta_m \frac{\partial}{\partial y} + \xi_m \frac{\partial}{\partial z}\right] \Phi_m(x,y,z) = \frac{\Sigma_{s0}(x,y,z)}{4\pi} \phi(x,y,z) + Q_m(x,y,z) \\ \text{for} \quad m = 1, 2, \ldots, N(N+2),\end{aligned} \tag{2.30}$$

where

$$\phi(x,y,z) = \sum_{m=1}^{N(N+2)} \frac{\pi}{2} w_m \Phi_m(x,y,z), \tag{2.31}$$

and where Σ_{s0} is the zeroth moment of the within-group scattering kernel. All other moments equal zero when scattering is isotropic.

In 3-D problems, $\Phi(\mathbf{r}, \mathbf{\Omega})$ must be determined over 8 octants of the unit sphere swept out by $\mathbf{\Omega}$. However, in 2-D, symmetry in the $x - y$ plane implies that the calculations need only be

performed for 4 octants. Provided the set of weights normalise to 1 over an octant, the equation simplifies to:

$$\left[\Sigma_t(x,y)+\mu_m\frac{\partial}{\partial x}+\eta_m\frac{\partial}{\partial y}\right]\Phi_m(x,y)=\frac{\Sigma_{s0}(x,y)}{4\pi}\sum_{m'=1}^{\frac{N(N+2)}{2}}\pi w_{m'}\Phi_{m'}(x,y)+Q_m(x,y)$$
$$\text{for}\quad m=1,2,\ldots,\frac{N(N+2)}{2}. \tag{2.32}$$

Similarly for a 1-D problem, azimuthal symmetry requires only one set of directional cosines to be evaluated to find the full solution to equation (2.30) [32]. Therefore, taking discrete values of μ in the range $[-1,+1]$, the equations simplify to:

$$\left[\Sigma_t(x)+\mu_m\frac{\partial}{\partial x}\right]\Phi_m(x)=\frac{\Sigma_{s0}(x)}{4\pi}\sum_{m'=1}^{N}2\pi w_{m'}\Phi_{m'}(x)+Q_m(x)\quad\text{for}\quad m=1,2,\ldots,N. \tag{2.33}$$

In the 1-D code written for this thesis, the whole equation is integrated over the azimuthal angle so that the flux being calculated is the angular flux integrated over the azimuthal angle, denoted by ψ_m. q_m is the source integrated over the azimuthal angle:

$$\left[\Sigma_t(x)+\mu_m\frac{\partial}{\partial x}\right]\psi_m(x)=\frac{\Sigma_{s0}(x)}{2}\sum_{m'=1}^{N}w_{m'}\psi_{m'}(x)+q_m(x)\quad\text{for}\quad m=1,2,\ldots,N. \tag{2.34}$$

In 1-D, when the S_N method uses Gaussian quadrature, it is equivalent to the $P_{N\text{-}1}$ method [32, 46]. Cullen [46] explains this using both a physical and mathematical argument.

The solution of the one-group angularly discretised problem can be solved using a source iteration scheme (SI) given by:

$$\left[\Sigma_t(x,y,z)+\mathbf{\Omega}_m\cdot\nabla\right]\Phi_m^{l+1}(x,y,z)=\frac{\Sigma_{s0}(x,y,z)}{4\pi}\phi^l(x,y,z)+Q_m(x,y,z), \tag{2.35}$$

where l is the iteration index.

The S_N method has the advantage of not being constrained to media in which the angle is assumed near-isotropic and of not being constrained to isotropic scattering [47]. The scheme has the advantage of not requiring the matrix to be formed, since the solvers are sweep-based, meaning memory requirements are less constraining. However, in highly scattering media the standard source iteration (SI) technique takes a long time to converge [39]. In practice, solver acceleration is essential. Various acceleration schemes are available e.g. the use of a Krylov

solver, diffusion synthetic acceleration (DSA) or a DSA pre-conditioned Krylov Solver [48]. DSA is implemented in EDF's DOMINO solver. DSA, a Krylov solver and a DSA pre-conditioned Krylov solver are acceleration options in the 2-D DD code written for this thesis.

S_N methods have the disadvantage of "ray effects" in multiple dimensions. These were discovered in the 1960s by both Bettis Atomic Laboratory and Los Alamos National Laboratory [35]. Figure 2.3 illustrates how the ray effect manifests itself in a point source, pure absorber, 2-D problem. The arrows in figure 2.3 are the directions of the discrete ordinates. The neutrons are modelled to travel along these directions only. As a result, each ordinate results in a Dirac delta function along the direction perpendicular to the ordinate direction. In reality there would be neutrons throughout the square domain. This issue is not as clear, but can still be identified [49], in cases where sources are distributed throughout the domain and scattering is present. To reduce the impact of the ray effect, one can refine the angular discretisation [49] or use alternative angular discretisation methods such as spherical harmonics (that do, however, suffer from Gibbs phenomenon [50]) or finite elements (that mitigate but do not eliminate the ray effect [51]).

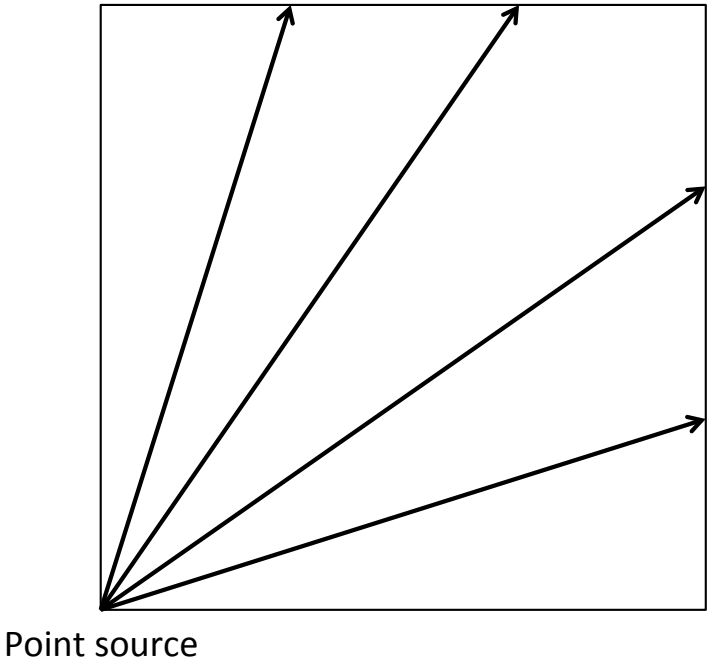


FIGURE 2.3: Example of the ray effect

2.2.2.3 Discretisation in space

Many different options are available: finite difference methods, finite volume methods, nodal methods, Bubnov-Galerkin and Petrov-Galerkin discretisation schemes. Which scheme is most appropriate depends on the form of the transport equation used. Researchers began to consider the use of finite elements for the neutron transport equation for both the even parity [52] and first order [53] approximations in the 1970s. Both continuous and discontinuous finite elements have been applied to the transport equations. Reed and Hill suggest that the discontinuous finite element approximation should be used with the first order transport equation since it was found to be more stable [53].

In this work the diamond difference (DD) scheme is implemented. Despite the increased use of finite element codes in the industry, many full scale production codes use the diamond difference (DD) method for solving the discrete ordinate (S_N) angular discretisation of the neutron transport equation. Examples include the PARTISN code developed at LANL [54], the Denovo S_N module in the SCALE code (developed at ORNL) [55] and the DOMINO solver that is a part of EDF's COCAGNE code [25].

For a given angular direction of the S_N scheme, with components μ_m and η_m in the x and y direction respectively, the 2-D diamond difference equations are as follows:

$$\mu_m \frac{\bar{\Phi}_{i+\frac{1}{2},j,m} - \bar{\Phi}_{i-\frac{1}{2},j,m}}{\Delta x_{i,j}} + \eta_m \frac{\bar{\Phi}_{i,j+\frac{1}{2},m} - \bar{\Phi}_{i,j-\frac{1}{2},m}}{\Delta y_{i,j}} + \Sigma_t \bar{\Phi}_{i,j,m} = \frac{\Sigma_{s0}}{4\pi} \sum_{m=1}^{\frac{N(N+2)}{2}} \pi w_m \bar{\Phi}_{i,j,m} + \bar{Q}_{i,j,m}, \tag{2.36}$$

$$\bar{\Phi}_{i,j,m} = \frac{\bar{\Phi}_{i+\frac{1}{2},j,m} + \bar{\Phi}_{i-\frac{1}{2},j,m}}{2}, \tag{2.37}$$

$$\bar{\Phi}_{i,j,m} = \frac{\bar{\Phi}_{i,j+\frac{1}{2},m} + \bar{\Phi}_{i,j-\frac{1}{2},m}}{2}, \tag{2.38}$$

where the flux terms are defined as follows:

$$\bar{\Phi}_{i,j,m} = \frac{1}{\Delta x_{i,j} \Delta y_{i,j}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \int_{y_{i-\frac{1}{2}}}^{y_{i+\frac{1}{2}}} \Phi_m(x,y) dx dy, \tag{2.39}$$

$$\bar{\Phi}_{i\pm\frac{1}{2},j,m} = \frac{1}{\Delta y_{i,j}} \int_{y_{i-\frac{1}{2}}}^{y_{i+\frac{1}{2}}} \Phi_m(x_{i\pm\frac{1}{2}},y) dy, \tag{2.40}$$

$$\bar{\Phi}_{i,j\pm\frac{1}{2},m} = \frac{1}{\Delta x_{i,j}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \Phi_m(x,y_{j\pm\frac{1}{2}}) dx. \tag{2.41}$$

Equations (2.36)-(2.38) are an extension of the DD equations in 1-D:

$$\mu_m \frac{\psi_{i+\frac{1}{2},m} - \psi_{i-\frac{1}{2},m}}{\Delta x_i} + \Sigma_t \bar{\psi}_{i,m} = \frac{\Sigma_{s0}}{2} \sum_{m=1}^{N} w_m \bar{\psi}_{i,m} + \bar{q}_{i,m}, \tag{2.42}$$

$$\bar{\psi}_{i,m} = \frac{\psi_{i+\frac{1}{2},m} + \psi_{i-\frac{1}{2},m}}{2}. \tag{2.43}$$

However, in the 1-D example, ψ represents the angular flux integrated over the azimuthal angle (hence the division by 2 instead of 4π in the isotropic scatter term in equation (2.42) compared to (2.36)). In the 1-D example, $\psi_{i+\frac{1}{2},m}$ and $\psi_{i-\frac{1}{2},m}$ represent cell-edge point values and and $\bar{\psi}_i$ the average over cell i. The connection between the 1-D DD scheme and a weighted residual scheme is discussed in chapter 3.

Both Hennart and Hébert have extended the DD-S_N method to give higher-order DD methods [56–58]. Hennart studied the 1-D case whereas Hébert extended the methods to multi-dimensions. Both methods are exactly equivalent in 1-D (see section 3.2.2.2). In Hébert's approach, however, the higher order variation within the cell is ignored when reconstructing the flux variation in the cell from the unknowns solved for. A 1-D code using Hennart's approach was written for this thesis. This allows the use of a higher-order approximation for the adjoint solution compared to the forward solution to give a more accurate error estimate. This point is discussed further in section 3.3.4. Two 2-D codes were written for this thesis, one using the traditional DD equations (2.36)-(2.38) and another using the *box method* [59] which has the same cell-average and edge-average fluxes as the DD scheme but is a direct extension of the 1-D WR formulation of the DD equations using point values as the unknown parameters. This is discussed further in chapter 6.

Although the cartesian mesh discrete ordinate methods (e.g. DD, box method, cartesian mesh FEs) must *staircase* or *tessellate* curved surfaces, Schunert and Azmy showed that the cartesian mesh discrete ordinates method applied in the DORT code provided accuracy comparable to other methods used for fuel assembly calculations, but at a reduced computational cost due to the advantage of structured grids [47, 60]. Azmy et al. showed that a square approximation to curved rod surfaces can still give an accurate estimate for the eigenvalue [61]. This can be achieved in two ways: by a square approximation of the same area as the curved geometry, or a square with the same wetted perimeter as the curved geometry (with a density factor to keep fissile mass constant). Azmy et al. [61] showed that the accuracy was higher in the case of constant area.

The tessellation required for curved surfaces makes adaptive meshing and ensuring the conservation of fissile mass at each mesh iteration rather difficult. The application for this project is whole reactor cores, so homogenised pin cells are used. Because of this, there is no need to model curved surfaces.

2.3 Error estimation and AMR

As previously discussed (section 1.5), the aim of the thesis is twofold. Firstly, to derive reliable DWR error estimates that provide a goal-based error estimate for a range of different QoI. Secondly, to reduce the number of DoF required for a given accuracy in QoI through goal-based AMR. Error estimation and AMR have been implemented previously in the field of neutron transport. A brief overview of the work already implemented in the field is given here.

AMR can take the form of h, p, hp or r type AMR. h AMR increases the number of cells in localised regions while keeping the polynomial order of the approximation within the cell the same. p AMR keeps the number of cells in the mesh the same but increases the polynomial order within individual cells. hp AMR can refine non-uniformly using both h and p type refinements in the same mesh. r AMR keeps the same number of DoF in the overall mesh, but moves nodes to where they are most needed to obtain a better solution. In this work only h, p and hp AMR are considered.

Heuristic AMR has previously been applied to the discrete ordinate finite volume (FV) equations [43, 62, 63]. Jessee et al. discretise the angular domain using the discrete ordinate technique and use the first order step (upwind) scheme to discretise the spatial dimension. The local truncation error was used as the local error indicator for refinement [62]. Aussourd et al. apply AMR to the DD-S_N method but choose refinement based on *a priori* knowledge of the mean free path [43, 63]. The lack of *a posteriori* information in the refinement indicator means that it does not take into account the spatial variation of the numerical solution to a particular problem, hence the refinement indicator can lead to sub-optimal meshes. Baker et al. applied block-AMR to the DD-S_N equations [44]. Work extending the possible acceleration techniques for this code to include diffusion synthetic acceleration (DSA) was later published [64, 65]. The non-uniform refinement method implemented was the "mesh potential" function method described in [66]. This method starts from a fine mesh and then indicates where the mesh can be coarsened without

sacrificing the accuracy of the solution. However, this method does not include the option of refining the mesh to give improved accuracy.

Jatuff et al. [67, 68] applied adaptive p refinement to the nodal diffusion equations. The error estimator was based on the L^2 norm of the residual. Zamonsky et al. applied p refinement to the discrete ordinate arbitrarily high order transport nodal method [69]. Ruggieri et al. applied h refinement to the variational nodal transport method [70]. Zhang and Lewis applied p adaptivity to the variational nodal diffusion equations [71] and later on to the variational nodal spherical harmonics method [72], the error estimator was given by the L^2 norm of the relative difference between the approximate solution and a reference solution.

Many reactor physics researchers have implemented error estimation and mesh refinement techniques to FE approximations of the neutron transport equations. Heuristic and goal-based refinement techniques have been applied.

Warsa and Prinja implemented p refinement for the 1-D discontinuous finite element discrete ordinate equations (DG-FEM-S_N) [73]. The solution over a cell is computed for p and $p+1$ for the same incoming flux in the sweep. If these do not agree to within a specified tolerance, the flux is calculated for $p+2$ and compared to $p+1$. This continues until the specified tolerance is achieved.

Wang and Ragusa in particular have done a lot of work on AMR and error estimation for the diffusion approximation, SP_N equations (along with Turcksin and Bangerth) and the discrete ordinate approximations of the neutron transport equation. The spatial dimension was discretised using either continuous (for diffusion and SP_N) or discontinuous (for S_N transport) FEs. Various non-goal-based and goal-based error indicators were investigated.

In [74] an estimator based on the difference between the approximate solution and a reference solution provided by a refined solution is used to drive hp refinement. This estimator was previously used in other applications by Demkowitz et al. [75, 76]. Later, Wang and Ragusa used an interpolation of a fine solution on a coarser mesh so that two solutions are not required for only one error estimator [77, 78]. [78] extends the work in [77] to give goal-oriented hp adaptivity for the multi group 1-D diffusion equations by applying the work of Šolín and Demkowicz [79] to the existing multi group hp adaptive scheme. Šolín and Demkowicz's work involves weighting the original error estimator by the same estimator applied to the adjoint solution (the source of which is determined by the QoI) [79]. Wang and Ragusa have also applied AMR in multiple dimensions.

In [80] an h adaptive scheme driven by a gradient jump error estimator is implemented. [81] extends the 1-D hp goal adaptive work for the diffusion equation to multi-dimensions. A Hessian based error estimator was implemented for the multigroup diffusion equations by Ragusa in [82, 83] which was used to drive h refinement. In [82] a goal-based adaptive scheme is driven by the Hessian for a QoI, and does not involve the solution of an adjoint equation.

Ragusa applied hp refinement to the SP_3 equations, where the error estimator uses the difference between the solution on a current mesh and one twice as refined with increased polynomial orders in each cell [84]. Goal-based h adaptive refinement was applied to the SP_N equations by Turcksin et al. [27, 85]. The error estimator was given by the norm of the difference between the approximate solution and a reference solution to the forward equation, multiplied by the norm of the difference between the approximate solution and a reference solution to the adjoint equation.

Wang and Ragusa extended their work to the multi-dimensional discrete ordinate neutron transport equations using the discontinuous Galerkin FE method (DG-FEM-S_N) [26, 86, 87]. First, a non-goal-based error estimator was applied to a h refinement scheme in 2-D [86]. The estimator used the difference between the solution of either the neutron current or the scalar flux computed on two meshes with different levels of refinement [86]. This work was extended to also include projection based error estimators or jump based error estimators based on moment values of the flux [26]. They made these estimators goal-based estimators by weighting the original estimator by an indicator of the error in the adjoint equation. This allowed for goal-based AMR. It has been shown that standard AMR "failed to provide an accurate answer" [26, p783]. This was because all the computational effort was devoted to refinement of the mesh in regions away from the area of interest; where fluxes were larger and therefore contributed more to the global error in the computational domain. This provides motivation for applying goal-based rather than standard AMR when solving for a particular QoI.

Lathouwers applied a DWR goal-based scheme described in a text book by Bangerth and Rannacher [88] to drive h adaptation in the 2-D DG-FEM-S_N equations. In this case the analytical residual of the forward approximation to the flux is weighted by the error in the adjoint solution [89]. The error in the adjoint solution is approximated by calculating the adjoint solution on the same mesh as the forward solution, but with a polynomial order that is everywhere increased. This work has been extended to eigenvalue problems, where the eigenvalue is the QoI [90]. Note

that unlike the work carried out by Wang and Ragusa, this error estimator uses the angular quantities and not angular integrated quantities.

Goffin et al. [91] also applied a DWR h adaptive mesh refinement scheme to the transport equations with the eigenvalue being the QoI. In this case the angular discretisation used was the spherical harmonics (P_N) discretisation with a sub-grid scale spatial discretisation. This work was later extended to provide both regular and goal-based angular adaptivity, by allowing the order of the spherical harmonic (P_N) expansion to be different at each node of the mesh for each energy group [92].

An error indicator for spatial refinement of the Arbitrarily High Order Transport Method of the Nodal type (AHOT-N) was derived by Duo et al. [93], with the aim of reducing the global L^2 error norm. Although duality arguments are used in the derivation of the error estimator, a dual solution is not required for its evaluation. The error in the adjoint solution is replaced by bounds to that term, given in terms of the forward solution. Similar arguments were used by Park et al. [94] for spatial and angular adaptivity of the FE-P_N discretisation of the even parity equations. It is noted that, in both of these cases, the goal quantity is limited to be the L^2 error, hence the mesh which results may not be optimal for other quantities of interest.

Gastaldo et al. applied hp adaptive refinement to the DG-FEM-S_N discretisation of the time dependent first order Boltzmann equation on non-conforming 2-D cartesian meshes. The refinement indicator used the difference between the approximate solution and reference solution [95].

Despite the wealth of error estimators, adaptive schemes, and more recently goal-based error estimators and adaptive schemes in the field of FEs applied to neutron transport, many industrial codes are prevented from implementing these techniques. This is because they use methods other than Galerkin FEs to solve the neutron transport equations. As previously mentioned, EDF uses the DD-S_N method. It may therefore be beneficial to derive equivalent error estimators for the DD-S_N scheme so that goal-based (specifically, DWR) error estimation and AMR can be applied to this widely used scheme.

Researchers outside of the neutronics field have investigated how to apply goal-based error estimators to methods other than FEs [96].

Giles and Pierce looked at ways of improving the approximate solution of a functional by using adjoint methods and to obtain a “computable correction” term. They lay out the theory for

general discretisation schemes noting that the function given by a FE solution could be used, or a function could be fit through nodal values of other discretisation schemes [97–101]. Chen et al. take a similar approach to presenting a general functional analytic framework for the DWR (goal-based) *a posteriori* error estimation for general discretisation methods, but handle the hyperbolic case more rigorously [102]. Chen et al. use adjoint equations of varying accuracy and use different methods for the forward and adjoint equations. They apply their methods to the 1-D linear advection problem.

Giles and Müller utilise the local computable correction terms as error indicators for AMR in a node-centered finite volume code [103]. In this case it was noticed that the residual distribution was noisy and therefore smoothing of the refinement indicator, or of the resulting mesh after adaptation was required. At the same time Venditti and Darmofal applied a discrete DWR error estimate to provide a correction to the result obtained for a functional (e.g. lift or drag) on a particular mesh. They then derive a separate error indicator to drive the AMR. They follow similar ideas to those presented by Giles, Pierce and Müller but with some differences. One difference is that they obtain a discrete adjoint equation from the discretised forward equations rather than discretising the continuous adjoint equations [104]. Another difference is that the indicator for refinement is not based on what they call the "computable correction" term (the case in [103]); instead, it is based on an estimate of the error in the computable correction term. This error indicator aims to reduce the residual in both the forward and adjoint solutions throughout the domain [104–107]. This method was applied in a quasi-1-D setting [104, 105] before being extended to 2-D flows [106, 107]. Balasubramanian and Newman apply the same ideas to 2-D inviscid, incompressible flows [108]. They compare the error estimator used by Müller and Giles [103] with that used by Venditti and Darmoval [105–107]. They concluded that the latter was more robust in their study. Kuzmin et al. also employed the technique of obtaining a DWR method from an arbitrary numerical scheme by deriving a finite element interpolant of the resulting values [109, 110].

Dwight devised a heuristic goal-based error estimator that does not involve evaluating a residual but does involve computing an adjoint solution [111]. More recent developments with regards to adjoint methods for mesh adaptation can be found in work by Angermann et al. [112], Shi et al. [113], Ponsin et al. [114] and Todarello et al. [115].

Some attempts have been made to reformulate difference schemes as variational problems so that a variation on the DWR scheme used in FE can be employed. Collins et al. applied this idea to

the Lax-Wendroff finite difference scheme [116]. This work was then extended by Chaudhry et al. to the IMEX multi-step time integration method [117].

2.4 A summary of the test cases used throughout the thesis

To validate the theoretical work of this thesis, the error estimators are tested on various test cases. A detailed description of each test case is given in appendix A along with an explanation of why each test case was chosen. A brief description of the 1-D and 2-D test cases are given in tables 2.1 and 2.2 respectively.

Test case	Problem type	QoI	Reference solution	Description
1	Fixed source	a: Integral of ϕ throughout the domain b: Integral of ϕ throughout the first 1cm of the domain	Analytical (by hand)	Homogeneous zero-scatter, vacuum boundary condition
2	Fixed source	a: Integral of ϕ throughout the domain b: Integral of ϕ throughout the first 1cm of the domain	Analytical (by hand)	Homogeneous scatter, vacuum boundary condition
3	Fixed source	a: Integral of ϕ throughout the domain b: Integral of ϕ throughout the first 1cm of the domain c: Integral of ϕ in the domain between 1.6cm and 2.4cm	Analytical [118]	Reed cell [119]
4	Fixed source	a: Integral of ϕ throughout the domain b: Integral of ϕ throughout the first 1cm of the domain	Analytical [120]	EIRA-1-D (cross section values taken from the EIRA-2A test case [1])
5	Eigenvalue	K_{eff}	Analytical (by hand [121, 122])	Homogeneous vacuum boundary condition
6	Eigenvalue	K_{eff}	RADIANT-II [2]	1-D-Mini-BWR (cross section values from the mini-BWR test case given in [123])

Table 2.1: 1-D test cases

Test case	Problem type	QoI	Reference solution	Description
7	Fixed source	a: Integral of ϕ throughout the domain b: Integral of ϕ throughout the first quarter of the domain	Analytical (for $c = 0.0$), RADIANT-II [2] for $c > 0.0$.	Homogeneous, vacuum boundary condition, various different values of scatter ratio (c)
8	Fixed source	a: Integral of ϕ throughout the domain b: Total absorption in the $\Sigma_t = 0.7$ box	RADIANT-II [2]	EIRA-2A [1]
9	Eigenvalue	K_{eff}	RADIANT-II [2]	Homogeneous high leakage
10	Eigenvalue	K_{eff}	RADIANT-II [2]	Homogeneous less leakage
11	Eigenvalue	K_{eff}	RADIANT-II [2]	2-D BWR test case (modification of the mini-BWR test case given in [123])

TABLE 2.2: 2-D test cases.
All reference solutions provided by RADIANT-II [2]

3. DWR error estimation and functional correction in 1-D slab geometry

This chapter derives the dual weighted residual (DWR) error estimate for a functional of the flux calculated using a DD discretisation of the discrete ordinate (S_N) approximation to the neutron transport equation. The method is derived for both linear fixed source type problems with detector function goal (which is also linear) as well as the eigenvalue criticality type problem with eigenvalue (K_{eff}) goal (which is a non-linear equation with non-linear goal). The performance of the error estimator is evaluated by applying it to test cases with reference solutions.

The novel parts of this chapter include the following:

- *Extending the WR view of the 1-D DD method to multiple angles.*
- *Showing that the DWR error estimation theory can be applied to the 1-D DD method.*
- *Showing why the adjoint solution using the same mesh as the forward solution does not give a zero error estimate in the DD method.*
- *Show that the DD method is not adjoint consistent and so the approximate adjoint solution must be derived by discretising the adjoint equation and not by taking the transpose of the already discretised forward equations.*
- *The application of DWR functional error correction to neutron transport problems.*
- *The application of DWR functional error correction to higher order DD schemes.*

3.1 Introduction

The quantity of interest (QoI) associated with a solution of a partial differential equation (PDE) is not, in general, the solution itself, but a functional of the solution. Goal-based error estimation aims to provide an estimate to the error resulting in the QoI, when the QoI has been calculated using an approximate solution to the PDE. Dual weighted residual (DWR) error estimators are one way of providing a goal-based error estimate, and will be investigated in this chapter.

This chapter aims to provide an estimate of the error in the QoI due to the spatial discretisation of the 1-D fixed source and eigenvalue (K_{eff} criticality) neutron transport problems. The discretisation scheme being used is the diamond difference (DD) method in space and discrete ordinate (S_N) method in angle. The QoIs investigated are the reaction rate in a detector and the value of the eigenvalue (K_{eff}) for the fixed source and criticality problems respectively.

This chapter not only investigates the traditional DD scheme [32] but also the use of the higher order DD scheme outlined by Hennart et al. [56]. The higher order DD scheme allows for higher order polynomial approximations within the cells.

The error estimates are formed by taking the cell-wise inner product over the continuous in space-discontinuous in angle phase space of the analytically calculated residual that results from the approximate solution and the solution of an appropriately defined adjoint equation. This quantity is calculated for all cells in the discretisation and summed over all cells to provide the error estimator for the goal quantity. In practice, the analytical solution to the adjoint equation is not available and it must be estimated numerically. In this work, the use of the following approximations to the adjoint solution for use in the DWR error estimator is investigated: the adjoint solution computed on the same mesh as the forward solution; the adjoint solution computed on a mesh with twice the number of cells as the mesh used for the forward equation; and a mesh with the same number of cells as the mesh used for the forward equation but using an increased polynomial order in each cell.

Once an estimation for the error has been calculated it can be used to determine whether the mesh is sufficiently refined to obtain the required accuracy in the QoI. The error estimation can also be used to improve the approximate solution for the QoI. Error correction of the QoI is investigated in this chapter.

This chapter is organised as follows. The traditional and higher order DD schemes applied to the S_N equations are described in section 3.2. Section 3.2 also shows how the DD schemes can

be viewed as weighted residual (WR) and Petrov-Galerkin methods. The theory required for applying the DWR error estimator to the 1-D DD-S_N equations is described in section 3.3. A description of how the DWR estimate can be used to correct the approximation to the QoI is given in section 3.4. The first part of section 3.5 gives a description of code implementation. The second part shows the results of applying DWR error estimation and functional correction to the 1-D test cases described in appendix A. Concluding comments to this chapter are given in section 3.6.

3.2 Higher order 1D-DD discretisation of the already S_N angularly discretised transport equation as a WR method

The view of the traditional, low order DD scheme as a WR scheme is described in Pitkäranta using a single ordinate, purely absorbing, fixed source problem [124]. This work shall be repeated in section 3.2.1 but specifying also that it is a strong solution for the transport equation that's being sought. Section 3.2.2 then shows how this can be extended to higher orders using the work done by Hennart et al. who extended the DD scheme in 1-D to higher orders using the polynomial case [56]. Hennart showed the method for a single ordinate. Section 3.2.3 extends this to multiple angles where the analytical solution is considered to be the solution to the continuous in space, but already discretised in angle, S_N equations. The WR representation will be written in such a way that all angles are considered in one expression with an inner product defined for the combined spatial and discrete angular space. This gives a straightforward way of expressing the scatter source in the WR representation of the equations. This representation also makes it easier to rigorously derive the DWR error estimator for the 1-D DD-S_N scheme in section 3.3. Finally, section 3.2.4 extends the multiple angle representation to eigenvalue problems where the ordinates are also coupled by the fission operator.

3.2.1 Traditional DD discretisation of the single-ordinate problem

In this section, the notation $\psi(x)$ represents an angular flux along a particular angular cosine. The single ordinate, fixed source, pure absorption problem is then written as:

$$\mu \frac{d\psi(x)}{dx} + \Sigma_{\mathrm{t}}(x)\psi(x) = q(x), \tag{3.1}$$

$$\psi(0) = \psi_{\mathrm{D}}, \tag{3.2}$$

assuming $\mu > 0$ and where ψ_{D} is a fixed incoming flux. Note that the j subscript is omitted in this section because only a single ordinate is being considered.

The WR method is a general mathematical approach to solving partial differential equations (PDEs) that encompasses many different types of numerical discretisation schemes. The point collocation, sub-domain, Bubnov-Galerkin (traditional method for FE codes) and Petrov-Galerkin methods all satisfy the definition of a WR method. This is because they involve weighting the residual of the PDE by a series of test functions, integrating over the solution domain and setting the resulting integral to zero. An approximate solution is formed by assuming the solution can be written as a finite sum of prescribed trial functions. Depending on the trial and test functions used, an approximation to the *classical*, *strong*, or *weak* formulation is found.

Süli and Jovanović distinguish between *classical*, *strong* and *weak* solutions of elliptic PDEs [125]. Oden and Reddy also define this distinction but for a more general linear operator that is not necessarily elliptic [126, eq. 2.30]. It must be noted that there is some inconsistency of nomenclature in the literature. For example, Allaire uses the terms *classical* and *strong* solution interchangeably [127, p66]. Quartering et al. refer to the *classical* form of the equation as the *strong* form [128, p544]. In this discussion the view of Oden and Reddy is taken, where there is a distinguishable difference between the terms *classical*, *strong* and *weak*, as defined in their book [126]. By this definition, the DD approximation seeks an approximation to the *strong* form of the governing equation (3.1) and its boundary condition (3.2), where the source term and cross sections are functions in L^2. Note, as an aside, that mono-directional point sources will not be well represented by the DD scheme due to their lack of regularity. For this, an approximation to the *weak* form of the governing equations (3.1) would be needed.

Given that the data and source terms exist in L^2, the differential term must also exist in L^2 for equation (3.1) to be satisfied in an integral sense:

$$\int_0^\ell R(x)v(x)dx = 0, \quad \forall v \in L^2, \tag{3.3}$$

where ℓ is the length of the spatial domain, and the integrand is the residual:

$$R(x) = q(x) - \mu\frac{d\psi(x)}{dx} - \Sigma_{\mathrm{t}}(x)\psi(x), \tag{3.4}$$

multiplied by any test function in L^2. A *strong* solution in H^1 (as opposed to a classical solution in C^1) to the integral equation (3.3), that also satisfies the boundary condition (3.2) is sought. Due to the residual being in L^2 the resulting integral equation must hold if weighted by any function in L^2.

In the DD scheme an approximate solution, $\psi_{\mathrm{h}}(x)$, is sought in an n dimensional subspace of H^1: $\psi_{\mathrm{h}}(x) \in U^h \subset H^1$. $\psi_{\mathrm{h}}(x)$ can be expanded in terms of a set of independent basis functions of U^h: the set of trial functions, $N_i(x)$:

$$\psi_{\mathrm{h}}(x) = \sum_{i=1}^{n} c_i N_i(x). \tag{3.5}$$

It is assumed, for the moment, that the lowest order DD scheme is being used. In this case, U^h represents the space of piece-wise linear functions over the $n-1$ cell discretisation, the subspace is n dimensional. The n basis functions of this subspace can be defined in such a way that each basis function is associated with a node in the discretisation of the physical domain. $N_i(x)$ shall be defined to be a piece-wise linear function that is unity at node i and zero at all other nodes (see figure 3.1b for a two-cell domain):

$$N_i(x) = \begin{cases} \frac{x-x_{i-1}}{x_i-x_{i-1}} & \text{if } x \in [x_{i-1}, x_i] \\ \frac{x_{i+1}-x}{x_{i+1}-x_i} & \text{if } x \in [x_i, x_{i+1}] \\ 0 & \text{otherwise} \end{cases} \quad \text{for} \quad i = 1, 2, \ldots n. \tag{3.6}$$

In this case, c_i in equation (3.5) gives the approximated flux value at nodal point i, which can be denoted by ψ_i. Thus, the lowest order DD approximation of the solution over a domain divided into $n-1$ cells can be written as:

$$\psi_{\mathrm{h}} = \sum_{i=1}^{n} \psi_i N_i(x). \tag{3.7}$$

The boundary conditions are essential, so the space of allowable functions is further restricted to U_{b}^h: the space that has a Dirichlet boundary condition at ψ_{b} ($\psi_{\mathrm{b}} = \psi_1$ for right-going ordinates, $\psi_{\mathrm{b}} = \psi_n$ for left-going ordinates). This space is $n-1$ dimensional and has $n-1$ unknowns.

Note that U_{b}^h can only be considered a vector space when the boundary conditions are homogeneous, otherwise the set is not closed under addition [129]. When the boundary conditions are not homogeneous, U_{b}^h is the set of allowable functions.

To solve for the remaining coefficients in equation (3.7), a set of $n-1$ independent test functions is required. In the DD scheme, these are obtained from a subset of L^2 (a different subset from the set of trial functions). The DD scheme is therefore a Petrov-Galerkin scheme, where the trial and test functions are not the same. As the dimension of the subspaces tend to infinity, a strong solution to the transport equation (3.3) is obtained. The advantage of a strong solution is that the resulting approximation, $\psi_{\mathrm{h}}(x)$, is regular enough to apply the operators of the equation to it. The residual that results from $\psi_{\mathrm{h}}(x)$ can therefore be calculated easily. In this work the residual is weighted by the adjoint solution to obtain a DWR error estimator (see section 3.3).

The lowest order DD scheme uses piece-wise constant test functions (see figure 3.1c for a two-cell domain). In this case, the test function space, V^h, is:

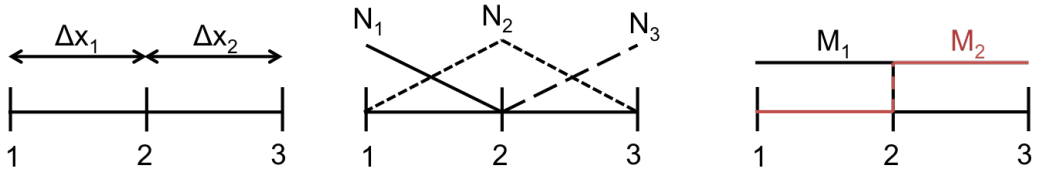
$$M_i(x) = \begin{cases} 1 & \text{if } x \in [x_i, x_{i+1}] \\ 0 & \text{otherwise} \end{cases} \quad \text{for} \quad i = 1, 2, \ldots n-1. \tag{3.8}$$

Hence, the lowest order DD scheme in 1-D is equivalent to a subdomain method. Equivalently, it is a Petrov-Galerkin scheme where the space of weighting functions is the space of cell-wise constant weighting functions and the trial functions are piece-wise linear globally continuous functions. The resulting set of algebraic equations to be solved, once divided through by Δx_i, is:

$$\mu \frac{[\psi_{i+1} - \psi_i]}{\Delta x_i} + \Sigma_{\mathrm{t},i} \frac{\psi_{i+1} + \psi_i}{2} = \frac{\int_i^{i+1} q dx}{\Delta x_i} \text{ for } i = 1, 2, \ldots, n-1. \tag{3.9}$$

This is the set of equations that results if equation (2.43) is substituted into (2.42), for the case where $\Sigma_{\mathrm{s}0} = 0$.

The lowest order DD scheme can also be viewed as a FV scheme; where the definition of a FV technique is taken to be a conservation equation that equates the net amount produced in the cell with the net amount that leaves through streaming.



(A) The two cell domain (B) Trial functions (C) Test functions

FIGURE 3.1: Example 2 cell domain and associated function spaces

3.2.2 The higher-order DD single ordinate equations.

The 1-D DD-S_N method implemented for this chapter allows for higher-order approximations in space. This is useful so that a more accurate solution to the adjoint equation (and hence a more accurate error estimator) can be calculated (see section 3.3.4).

There are two approaches for extending the traditional DD discretisation to higher orders. The first is provided by Hennart [56] and is used in this thesis since it is an extension of the WR view of the traditional DD equations (discussed in section 3.2.1) to higher orders. This gives a rigorous mathematical basis to the final equations. This method is explained in section 3.2.2.1 below.

The second method is provided by Hébert [57]. The resulting equations are equivalent to the Hennart scheme. However, all the spatial moments are not retained in the final solution. For example, in the traditional DD scheme, although the numerical scheme is equivalent to assuming a linear variation within the cell, only the cell-averaged fluxes are kept (see equation (16) of [57]) and the linear variation is ignored at output. This is explained further in section 3.2.2.2.

3.2.2.1 Hennart's higher order DD formulation

The theory underlying the higher-order DD-S_N method is described in detail by Hennart [56] for a single ordinate case. He discusses both *polynomial* and *analytical* basis functions in his paper, but it is only the polynomial case that is considered here. The lowest order polynomial scheme corresponds to the traditional DD WR scheme described above.

In Hennart's formulation, the flux within cell i is expanded in terms of the set of basis functions:

$$\psi_{\mathrm{h},p,i}(x) = c_i N_i(x) + c_{i+1} N_{i+1}(x) + \sum_{k=0}^{p-2} c_{c,i}^k N_{c,i}^k(x), \tag{3.10}$$

where h indicates that the solution is approximate and p denotes the order of approximation in the cell.

Defining:

$$\zeta = \frac{2x - x_i - x_{i+1}}{\Delta x_i}, \tag{3.11}$$

the basis functions are then defined in terms of normalised Legendre polynomials as follows [56]:

$$N_i(x) = \frac{1}{2}(-1)^{p-1}\left[P_{p-1}(\zeta) - P_p(\zeta)\right] \tag{3.12}$$

$$N_{i+1}(x) = \frac{1}{2}\left[P_{p-1}(\zeta) + P_p(\zeta)\right] \tag{3.13}$$

$$N^k_{c,i}(x) = P_k(\zeta) + P_{p-2+m(k)}(\zeta), \quad \text{for } k = 0, \ldots, p-2; \tag{3.14}$$

$m(k) = 1$ or 2 such that $p-2+m(k)$ has the same parity as k. $P_k(\zeta)$ is the normalised Legendre polynomial of degree k over the reference cell $[-1, +1]$. Equations (3.12)-(3.14) are equivalent to equation (3.6) within a cell when $p = 1$. Basis functions (3.12)-(3.14) always map out a subspace of H^1. The Legendre polynomials have all the properties specified in equation 1.5 of Hennart's paper [56], these are repeated below:

$$P_k(+1) = 1, \tag{3.15}$$

$$P_k(-1) = (-1)^k, \tag{3.16}$$

$$\int_{-1}^{+1} P_k(\zeta)P_l(\zeta)d\zeta = \delta_{kl}N_k, \tag{3.17}$$

$$\int_{x_i}^{x_{i+1}} P_k\left(\frac{2x - x_i - x_{i+1}}{\Delta x_i}\right) P_l\left(\frac{2x - x_i - x_{i+1}}{\Delta x_i}\right) dx = \frac{\Delta x_i}{2}\int_{-1}^{+1} P_k(\zeta)P_l(\zeta)d\zeta \quad = \delta_{kl}N_k\frac{\Delta x_i}{2}, \tag{3.18}$$

where $N_k = \frac{2}{2k+1}$ and δ_{kl} is the Kronecker delta function.

Globally, this is similar to standard vertex FEs, but instead of the basis functions being chosen such that the unknowns to be solved for are the values at the vertices and specified internal point values (for orders higher than linear), the coefficients represent the boundary vertices and internal cell moments. With this in mind, equation (3.10) can be re-written as [56, p455]:

$$\psi_{\mathrm{h},p,i}(x) = \psi_i N_i(x) + \psi_{i+1}N_{i+1}(x) + \sum_{k=0}^{p-2} \psi^k_{c,i}N^k_{c,i}(x), \tag{3.19}$$

where the subscripts i or $i+1$ indicate cell node values, and c,i indicates a cell moment value for cell number i. The cell central moments of a function $u(x)$ defined over a cell i are as follows:

$$m_{c,i}^{k}(u(x)) = \frac{\int_{x_i}^{x_{i+1}} u(x) P_k\left(\frac{2x-x_i-x_{i+1}}{\Delta x_i}\right) dx}{\int_{x_i}^{x_{i+1}} P_k\left(\frac{2x-x_i-x_{i+1}}{\Delta x_i}\right) P_k\left(\frac{2x-x_i-x_{i+1}}{\Delta x_i}\right) dx} \tag{3.20}$$

$$= \frac{\int_{-1}^{+1} u(\frac{\zeta\Delta x_i+x_i+x_{i+1}}{2}) P_k(\zeta)\, d\zeta}{N_k}. \tag{3.21}$$

The weighting (test) functions are different from the set of trial functions. Firstly, instead of being taken from a subspace of H^1, they are taken from a subspace of L^2, since they are allowed to be discontinuous over cell boundaries. In addition to this, given that one of the unknowns is provided by either a boundary condition, or the calculation over the previous cell, one less dimension is required compared to the the space of cell trial functions. The set of locally defined test functions for a p-ordered cell, defined over the interval $[x_i, x_{i+1}]$ is therefore:

$$M_{i,k}(x) = \begin{cases} P_k\left(\frac{2x-x_i-x_{i+1}}{\Delta x_i}\right) & \text{if } x \in [x_i, x_{i+1}] \\ 0 & \text{otherwise} \end{cases} \quad \text{for} \quad k = 0, 1, \ldots p-1 \text{ and } i = 1, 2, \ldots, nc, \tag{3.22}$$

where nc is the number of cells in the discretisation. Clearly equation (3.22) reduces to (3.8) for the case where $p = 1$. In that case, $nc = n - 1$, where n is the total number of degrees of freedom.

The cell equations for cell number i, with cell-order p can be written as [56]:

$$\frac{2\mu}{N_k \Delta x_i}\left\{\left[\psi_{i+1} - (-1)^k \psi_i\right] - \int_{-1}^{+1} \psi_{\mathrm{h}}(\zeta) \frac{dP_k(\zeta)}{d\zeta} d\zeta\right\} + \Sigma_{t,i}\psi_c^k = m_c^k(q), \quad k = 0, \ldots p-1. \tag{3.23}$$

For the integration by parts to be possible in equation (3.23) the test functions must be at least H^1 within an element so that a trace function can be defined [130]. This is the case here, the test functions are globally L^2 and C^∞ within a cell.

In the 1-D code written for this thesis, the cell edge values and cell central moments are rearranged to give the Legendre Polynomial moments for each cell. This simplifies the integration when calculating the DWR for each cell. The formula for the elements of the inverse of the $p \times p$

matrix, given by equation (3.23), is hard-coded into the 1-D code for a given p. At the moment, the code can accommodate up to $p = 4$.

To conclude this section, one remarks that the lowest order DD scheme can still be considered a WR method that approximates a strong solution, even after extension to higher-order. However, the method can no longer be seen as a subdomain method or a FV method. It is not a subdomain method because the weighting functions are of higher-order than piece-wise constant within a cell (though they remain as globally L^2). The higher-order DD can no longer be considered a FV method either, since there is more detail being kept than just the concept of the outgoing flux being equal to the incoming flux plus the net production in the cell. On the contrary, the higher-order DD gives a more detailed representation of how the solution varies within the cell. It can, however, still be considered a Petrov-Galerkin scheme, since the weighting and trial functions are different. It has also been demonstrated that Hennart's generalised case reduces to the traditional DD scheme when $p = 1$.

3.2.2.2 Hébert's higher order DD formulation

First, Hébert's interpretation of the DD formulation is explained. Then it is explained how the Mth order Hébert scheme is equivalent to the $M + 1$th order Hennart scheme but with the highest order spatial moment being ignored at output.

Hébert explained in his 2006 article [57] that the flux is expanded in terms of a Legendre polynomial basis in each cell:

$$\psi(u)_i = \sum_{k=0}^{M} \tilde{P}_k(u)\psi_{c,i}^k. \tag{3.24}$$

$\tilde{P}_k(u)$ are the normalised Legendre Polynomials where u varies from -0.5 to $+0.5$ as follows:

$$u = \frac{2x - x_i - x_{i+1}}{2\Delta x_i} \tag{3.25}$$

In Hennart's derivation, Legendre Polynomials were a function of a parameter that varied from $[-1, +1]$. The Legendre Polynomials used in Hébert's derivation are normalised differently such that the equivalent of equation (3.18) for the normalised Legendre Polynomials is:

$$\int_{x_i}^{x_{i+1}} \tilde{P}_k\left(\frac{2x - x_i - x_{i+1}}{2\Delta x_i}\right) \tilde{P}_l\left(\frac{2x - x_i - x_{i+1}}{2\Delta x_i}\right) dx = \Delta x_i \int_{-0.5}^{+0.5} \tilde{P}_k(u)\tilde{P}_l(u)du = \delta_{kl}\Delta x_i, \tag{3.26}$$

Thus the following equality should be kept in mind when finding the equivalence between the two schemes:

$$\tilde{P}_k(u) = \sqrt{\frac{2}{N_k}} P_k(2u). \tag{3.27}$$

A set of WR equations are then derived by weighting the residual by the normalised Legendre polynomial moments. Equations (19)-(21) in Hébert's paper [57] are repeated in equations (3.28)-(3.30) below (but using this thesis's node numbering and notation):

$$\frac{\mu}{\Delta x_i}(\psi_{i+1} - \psi_i) + \Sigma_{t,i}\psi^0_{c,i} = Q^0_{c,i} \quad M = 0 \text{ WR equation} \tag{3.28}$$

$$\sqrt{3}\frac{\mu}{\Delta x_i}(\psi_{i+1} + \psi_i - 2\psi^0_{c,i}) + \Sigma_{t,i}\psi^1_{c,i} = Q^1_{c,i} \quad M = 1 \text{ WR equation} \tag{3.29}$$

$$\sqrt{5}\frac{\mu}{\Delta x_i}(\psi_{i+1} - \psi_i - 2\sqrt{3}\psi^1_{c,i}) + \Sigma_{t,i}\psi^2_{c,i} = Q^2_{c,i} \quad M = 2 \text{ WR equation} \tag{3.30}$$

It is noted that equations (3.28)-(3.30) have not yet substituted in for the *approximate flux solution* (3.24). Hébert then proceeds to find auxiliary equations to close the system by writing the the outgoing edge's flux in terms of the incoming edge's flux and the cell central moments "in such a way to cancel the $M+1$-order term" [57]:

$$\psi_{i+1} + \psi_i - 2\psi^0_{c,i} = 0 \quad M = 0 \text{ auxiliary equation} \tag{3.31}$$

$$\psi_{i+1} - \psi_i - 2\sqrt{3}\psi^1_{c,i} = 0 \quad M = 1 \text{ auxiliary equation} \tag{3.32}$$

$$\psi_{i+1} + \psi_i - 2\psi^0_{c,i} - 2\sqrt{5}\psi^2_{c,i} = 0 \quad M = 2 \text{ auxiliary equation} \tag{3.33}$$

It is the $M+1$th order term of the *gradient* of the flux that is made equal to zero. The Mth order term of the gradient of the flux is non-zero and thus the $M+1$th order term in the flux must also be non-zero. Hébert explains that $M = 0$ corresponds to the traditional DD equations. The 0th order WR equation provides the traditional DD integral equation and ensuring the $M+1$th order term of the gradient of the flux is equal to zero gives the traditional DD auxiliary equation. Therefore Hébert's DD0 method does have a non-zero first order Legendre polynomial term (with all higher order terms equal to zero) and thus the representation of the flux given by equation (3.24) and equation (16) of Hébert's paper does not represent the assumptions made of the solution in the derivation of the numerical scheme.

The representation of the flux given by equation (3.24) could lead one to think that the $M+1$th order flux and source terms are zero. If this were the case, substituting in for these terms in the $M+1$th WR equation then provides the auxiliary equations. However, this would lead to

a mathematical inconsistency, and is therefore not the case. This can be easily demonstrated in the lowest order (traditional DD) case. If all flux spatial moments of order > 0 were equal to zero, the right and left flux values would be the same by definition. Equation (3.28) would then be reduced to a simple relation between that which is produced and consumed within the cell (with no streaming term). It also means that the solution of equation (3.29) is trivial. The only way ψ_i, ψ_{i+1} and $\psi^0_{c,i}$ can be different from each other is if the flux variation within the cell is of a higher order than zero. The WR equation used (3.28) is the same as Hennart's scheme and the auxiliary equation (3.31) is obtained by enforcing a linear variation in the cell (setting the first order component of the flux gradient to zero). This means that Hébert's method is equivalent to Hennart's method. Hébert's scheme is therefore equivalent to a WR scheme that has trial and test functions from different spaces.

This equivalence extends to higher orders once the differences in the normalisation of the Legendre polynomials are taken into account. Equation (3.27) means that the cell central moments provided by the Hébert's scheme relate to the cell central moments provided by Hennart's scheme as follows:

$$\psi^k_{c,i,\text{Hennart}} = \sqrt{\frac{2}{N_k}}\psi^k_{c,i,\text{Hébert}} \tag{3.34}$$

The cell node values are the same in both cases. Therefore, substituting in for (3.34) in (3.28)-(3.30) and (3.31)-(3.33) for a given M order scheme results in the same equations as in the $M+1$ order Hennart scheme. Thus Hébert's Mth order method is equivalent to the Hennart's $M+1$th method with the $M+1$th moment ignored when reconstructing the flux in the spatial domain.

3.2.3 Multi-angle problems

To extend Hennart's method to multiple angle problems, the solution, $\Psi \in \left(H^1\right)^N := \bar{U}$, must be considered. Elements 1 to $\frac{N}{2}$ are the flux solutions for angular ordinates with $\mu_n > 0$. Elements $\frac{N}{2}+1$ to N are the flux solutions for angular ordinates with $\mu_n < 0$.

The continuous-in-space and discretised-in-angle equation to be approximated, (2.34), can therefore be represented as follows (in the fixed source case):

$$W\mathcal{A}\Psi = WQ. \tag{3.35}$$

$\Psi \in \bar{U}$, with the m'th element corresponding to the continuous function that represents the angular flux in direction m. $Q \in \left(L^2\right)^N := \bar{V}$ is the discretised in angle fixed source vector. Each

element m corresponds to the fixed source in direction m, which is a function in the L^2 space. $\mathcal{A}$ is an $N \times N$ matrix, the diagonal contains the streaming and absorption operators for each angle and the off-diagonals contain isotropic scattering terms multiplied by the corresponding angular weight, as shown for the general case below:

$$\mathcal{A} = \begin{bmatrix} \mu_1 \frac{d}{dx} + \Sigma_{\mathrm{t}} - \frac{\Sigma_{s0} w_1}{2} & \cdots & -\frac{\Sigma_{s0} w_m}{2} & \cdots & -\frac{\Sigma_{s0} w_N}{2} \\ -\frac{\Sigma_{s0} w_1}{2} & \cdots & \mu_m \frac{d}{dx} + \Sigma_{\mathrm{t}} - \frac{\Sigma_{s0} w_m}{2} & \cdots & -\frac{\Sigma_{s0} w_N}{2} \\ -\frac{\Sigma_{s0} w_1}{2} & \cdots & -\frac{\Sigma_{s0} w_m}{2} & \cdots & \mu_N \frac{d}{dx} + \Sigma_{\mathrm{t}} - \frac{\Sigma_{s0} w_N}{2} \end{bmatrix}. \tag{3.36}$$

$\mathcal{A}$ represents a mapping $\bar{U} \to \bar{V}$. W is a diagonal matrix consisting of the angular weights. Because its inverse exists, solving $\mathcal{A}\Psi = Q$ is equivalent to solving equation (3.35). It is used here to represent the importance of the individual angular equations relative to one another. The following notation is adopted:

$$\mathcal{K} = W\mathcal{A}. \tag{3.37}$$

The $\mathrm{S_N}$ problem, before spatial discretisation, is a matrix equation to solve for Ψ, which exists in a subset of the space $\bar{U}$, denoted by $\bar{U}_{\mathrm{b}}$. $\bar{U}_{\mathrm{b}}$ is the set of all $\bar{U}$ with the appropriate boundary conditions (left-most node given for right-going ordinates and right-most node given for left-going ordinates). For simplicity of explanation vacuum boundary conditions are assumed, hence $\bar{U}_{\mathrm{b}}$ shall be replaced by $\bar{U}_0$. It is clear that $\bar{U}_0 \subset \bar{V}$, it is also dense in $\bar{V}$. Due to all the terms in equation (3.35) existing in $\bar{V}$, the integral statement of the equation can be denoted by taking the scalar product defined in this space with a test function in this space. An appropriate scalar product defined for the space $\bar{V}$ would be:

$$\langle a, b \rangle_{\bar{V}} = \int a^T b dx \quad \forall a, b \in \bar{V}. \tag{3.38}$$

Hence the strong form integral statement of the equation is defined as:

Find $\Psi \in \bar{U}_0$ such that:

$$\langle \mathcal{K}\Psi, v \rangle_{\bar{V}} = \langle WQ, v \rangle_{\bar{V}} \quad \forall v \in \bar{V}. \tag{3.39}$$

This equation can be solved approximately using the DD equations for multiple angles. This is achieved by taking the solution to be in a finite dimensional subset: $\bar{U}_0^h \subset \bar{U}_0$. For the DD equations $\bar{U}_0^h$ is defined by the space of all N sized vectors, where elements 1 to $\frac{N}{2}$ are for functions in $U_{0_{\mathrm{L}}}^h \subset H_{0_{\mathrm{L}}}^1$ (where the subscript 0_{L} denotes that the left boundary of the spatial domain is

a vacuum boundary condition), and elements $\frac{N}{2}+1$ to N are functions in $U^h_{0_\mathrm{R}} \subset H^1_{0_\mathrm{R}}$ (where the subscript 0_R denotes that the right boundary of the spatial domain is a vacuum boundary condition). This space is mapped out in each ordinate by n independent basis functions (of the form (3.12)-(3.14)). This gives $n \times N$ basis functions in total. The unknown coefficients are solved by N boundary conditions (one for each ordinate) and $(n-1) \times N$ weighting functions. These weighting functions are taken from a subspace of $\bar{V}$ denoted by $\bar{V}_\mathrm{h} \subset \bar{V}$, mapped out in each ordinate by $n-1$ basis functions (of the form given in (3.22)). This gives $(n-1) \times N$ basis functions in total.

3.2.4 Eigenvalue problems

The same method can be applied to the eigenvalue equation. In this case the equation to be solved is:

$$\mu_m \frac{d\psi_m}{dx} + \Sigma_\mathrm{t}\psi_m - \frac{\Sigma_{\mathrm{s}0}}{2}\phi - \frac{\nu\Sigma_\mathrm{f}}{2\mathrm{K}_\mathrm{eff}}\phi = 0 \quad \text{for } m = 1, 2, \ldots, N \quad , \tag{3.40}$$

with the following normalisation condition [88, 90]:

$$\sum_{m=1}^{N} w_m \int_0^\ell \frac{\nu\Sigma_\mathrm{f}}{2}\phi\psi_m dx = \int_0^\ell \frac{\nu\Sigma_\mathrm{f}}{2}\phi\phi dx = 1. \tag{3.41}$$

Equations (3.40) and (3.41) can be expressed as a matrix equation in angle, and a differential equation in space as follows:

Find $\Psi \in \bar{U}_\mathrm{b}$ and $\lambda \in \mathbb{R}$ such that:

$$\mathcal{K}\Psi = \lambda W \mathcal{F}\Psi, \tag{3.42}$$

$$\langle W\mathcal{F}\Psi, \Psi\rangle_{\bar{V}} = 1. \tag{3.43}$$

Everything is defined as before. $\lambda = \frac{1}{\mathrm{K}_\mathrm{eff}}$. $\mathcal{F}$ is an $N \times N$ matrix, each element contains the function that represents the spatial variation of the fission cross section (in this case single group): $\frac{\nu\Sigma_\mathrm{f}}{2}$ multiplied by an angular weight w_m, where m represents the column number as follows:

$$\mathcal{F} = \begin{bmatrix} \frac{\nu\Sigma_\mathrm{f}w_1}{2} & \ldots & \frac{\nu\Sigma_\mathrm{f}w_m}{2} & \ldots & \frac{\nu\Sigma_\mathrm{f}w_N}{2} \\ \frac{\nu\Sigma_\mathrm{f}w_1}{2} & \ldots & \frac{\nu\Sigma_\mathrm{f}w_m}{2} & \ldots & \frac{\nu\Sigma_\mathrm{f}w_N}{2} \\ \frac{\nu\Sigma_\mathrm{f}w_1}{2} & \ldots & \frac{\nu\Sigma_\mathrm{f}w_m}{2} & \ldots & \frac{\nu\Sigma_\mathrm{f}w_N}{2} \end{bmatrix}. \tag{3.44}$$

The operator $\mathcal{F}$ represents a mapping: $\bar{V} \to \bar{V}$. Note that equation (3.42) is equivalent to equation (3.40) but with each ordinate equation multiplied by its corresponding weighting function.

Both the eigenvalue equation and the normalisation condition can be expressed as WR statements. This is done by expressing the residual of equation (3.42) as:

$$R = \mathcal{K}\Psi - \lambda W \mathcal{F}\Psi, \tag{3.45}$$

then multiplying this by a test function $v \in \bar{V}$ and integrating over the phase space and setting the result to zero. This is equivalent to writing:

$$\langle R, v \rangle_{\bar{V}} = 0 \quad \forall v \in \bar{V}. \tag{3.46}$$

The normalisation condition can be multiplied by a test scalar [88, 90]:

$$\tau r = \tau \left[1 - \langle W \mathcal{F}\Psi, \Psi \rangle_{\bar{V}}\right] = 0 \quad \forall \tau \in \mathbb{R}. \tag{3.47}$$

Both of these combined result in the WR statement for the eigenvalue problem:

$$\langle R, v \rangle_{\bar{V}} + \tau r = 0 \quad \forall \hat{v} \in \hat{\bar{V}}. \tag{3.48}$$

The test function space is defined as [88, 90]:

$$\hat{v} = \{v, \tau\} \in \hat{\bar{V}} \text{ where } \hat{\bar{V}} := \bar{V} \times \mathbb{R}. \tag{3.49}$$

The aim is to find both a function, Ψ, and a scalar, λ, that satisfies the WR statement (3.48). This problem can be viewed as finding a member:

$$\hat{\Psi} = \{\Psi, \lambda\} \in \hat{\bar{U}}_0, \tag{3.50}$$

where

$$\hat{\bar{U}}_0 := \bar{U}_0 \times \mathbb{R}, \tag{3.51}$$

such as in [90].

The WR statement of equation (3.48) can be expressed using similar notation to [90]:

Find $\hat{\Psi} \in \hat{\bar{U}}_0$ such that:

$$A(\hat{\Psi}; \hat{v}) = B(\Psi, v) - \lambda F(\Psi, v) + \tau(1 - F(\Psi, \Psi)) = 0 \quad \forall \hat{v} \in \hat{\bar{V}}, \tag{3.52}$$

where $A(\cdot;\cdot)$ is a semi-linear form: non-linear in the argument that precedes the semi-colon and linear in the argument that follows the semi-colon. The operators in equation (3.52) are defined in terms of the inner product over $\bar{V}$ as follows:

$$F(\cdot,\cdot) = \langle W\mathcal{F}\cdot, \cdot\rangle_{\bar{V}} \quad : \quad \bar{V} \times \bar{V} \to \mathbb{R}, \tag{3.53}$$

$$B(\cdot,\cdot) = \langle \mathcal{K}\cdot, \cdot\rangle_{\bar{V}} \quad : \quad \bar{U} \times \bar{V} \to \mathbb{R}. \tag{3.54}$$

In the DD case the weighting functions are taken from a different space to the trial functions, this is not the case for the equivalent expression given in [90].

From this point on, the arguments for the DD approximation are very similar to the fixed source multi-angle approach. The solution $\hat{\Psi}_\mathrm{h}$ is taken from a space $\hat{\bar{U}}_0^h \subset \hat{\bar{U}}_0$. The $N \times n$ coefficients for the basis functions of the approximate solution space are solved for by N boundary conditions and $N \times (n-1)$ WR equations. The WR equations are given by defining the weighting functions as the basis functions of the space $\hat{\bar{V}}_\mathrm{h} \subset \hat{\bar{V}}$.

3.3 DWR error estimation theory applied to the 1-D DD-S_N method

In this section the DWR scheme for the DD-S_N fixed source and eigenvalue equations is explained. An estimation of the numerical error in a goal quantity due to the spatial discretisation is sought. In the fixed source case the goal quantity is a detector function (a linear QoI) and in the eigenvalue case, it is the eigenvalue itself (a non-linear QoI). The work in this section combines the original ideas expressed by Rannacher et al. [18, 88], that were applied specifically to the DG-FEM-S_N neutron transport method by Lathouwers [89, 90], with the generalisation to non-Bubnov-Galerkin discretisation schemes given by Giles, Pierce and Chen [97–102]. Section 3.3.1 shows the general scheme for any linear or semi-linear form. In both cases the form is obtained by weighting an equation for which the solution is in a space U, by a space V, of which U is a dense subspace. The main distinction from Rannacher's work is that $U \neq V$ in the

non-Bubnov-Galerkin case. The specific application to fixed source and eigenvalue problems in neutron transport is explained in sections 3.3.2 and 3.3.3 respectively.

3.3.1 General non-Bubnov-Galerkin case

Consider the general semi-linear form on a vector space V:

$$A(\cdot;\cdot) : U \times V \to \mathbb{R} \text{ where } U \text{ is dense in } V, \tag{3.55}$$

which is non-linear with respect to the parameter before the semi-colon and linear with respect to anything after it.

There is a problem to be solved for $u \in U$ defined by the WR equation (3.56):

$$A(u;v) = 0 \quad \forall v \in V, \tag{3.56}$$

where U is dense in V.

There exists a QoI defined by a functional (linear or non-linear) of the solution to equation (3.56) which is denoted by:

$$J(u) : U \to \mathbb{R}. \tag{3.57}$$

Becker and Rannacher derive an approach to estimating the error in the functional: $J(u) - J(u_{\mathrm{G}})$ where u_{G} is the solution to a Bubnov-Galerkin approximation to equation (3.56) by using "the framework of optimal control" [18, p5 and p9]. A similar method is followed here to obtain an estimate for the numerical error due to spatial discretisation when the approximate solution is solved using a DD-$\mathrm{S_N}$ discretisation method.

One can look at the problem of finding $J(u)$ from the solution $u \in U$ of equation (3.56) as a trivial equality constrained optimisation problem [18]:

$$\text{Minimise } J(u), \text{ given the constraint that: } A(u;v) = 0 \quad \forall v \in V \text{ and } u \in U. \tag{3.58}$$

It is trivial because the value of $J(u)$ is unique provided that u is the solution of the equality constraint equation. The method of Lagrangian multipliers is used, where a Lagrange multiplier,

z, is defined and the following Lagrangian obtained [130]:

$$L(u,z) = J(u) + A(u;z) \text{ given the constraint that } u \in U \text{ and } z \in V. \tag{3.59}$$

If $J(\psi)$ is a minimum of $J(u)$ for the original constrained problem, there exists $\psi^\dagger$ such that $L(\psi,\psi^\dagger)$ is a stationary point of the Lagrangian (3.59) [130].

A stationary point is where the Gateaux partial derivatives of $L(u,z)$ with respect to both parameters u and z are zero in all directions. The following notation is used for the Gateaux partial derivatives of the Lagrangian:

$$L'_u(u,z)[y] \quad \text{and} \quad L'_z(u,z)[g]. \tag{3.60}$$

The quantity in square brackets denotes the direction in which the Gateaux derivative is taken. The $'$ denotes a first order derivative and the corresponding subscript denotes with respect to which parameter the derivative is being taken.

To find the stationary point of $L(u,z)$, denoted by $L(\psi,\psi^\dagger)$, the following equations must be satisfied:

$$L'_u(\psi,\psi^\dagger)[y] = J'_u(\psi)[y] + A'_u(\psi;\psi^\dagger)[y] = 0 \quad \forall y \in U, \tag{3.61}$$

$$L'_z(\psi,\psi^\dagger)[g] = J'_z(\psi)[g] + A'_z(\psi;\psi^\dagger)[g] = 0 \quad \forall g \in V. \tag{3.62}$$

Equation (3.62) can be simplified to:

$$L'_z(\psi,\psi^\dagger)[g] = A'_z(\psi;\psi^\dagger)[g] = 0 \quad \forall g \in V. \tag{3.63}$$

This equation is satisfied if ψ is the solution to the forward equation, no matter what the definition of $\psi^\dagger \in V$. $\psi^\dagger$ in this equation can be replaced by any $z \in V$ and the equality will still hold. It is known from the forward problem that $A(\psi;z)$ is a constant value of zero for all functions $z \in V$. Hence the gradient with respect to the second parameter is zero for all functions $z \in V$. Therefore this component of the derivative cannot be used to determine the value of $\psi^\dagger$.

The adjoint problem therefore reduces to finding a value $\psi^\dagger$ that satisfies the equation:

$$L'_u(\psi, \psi^\dagger)[y] = J'_u(\psi)[y] + A'_u(\psi; \psi^\dagger)[y] = 0 \quad \forall y \in U. \tag{3.64}$$

The means of finding the value $\psi^\dagger$ will depend on the particular problem (two examples are treated in the following sections).

Equation (3.64) can also be used to give a value for $J(\psi_\mathrm{h}) - J(\psi)$, where ψ_h is an estimate of ψ in the U space (for example, the lowest order DD approximation). This is done by considering the case where y in equation (3.64) is equal to $e = \psi_\mathrm{h} - \psi \in U$:

$$J'_u(\psi)[\psi_\mathrm{h} - \psi] + A'_u(\psi; \psi^\dagger)[\psi_\mathrm{h} - \psi] = 0, \tag{3.65}$$

which can be expressed as:

$$J(\psi_\mathrm{h}) - J(\psi) = A(\psi; \psi^\dagger) - A(\psi_\mathrm{h}; \psi^\dagger) + \epsilon, \tag{3.66}$$

where:

$$\epsilon = \sum_{l=2}^{t-1} \frac{\left\{J^l_u(\psi)[\psi_\mathrm{h} - \psi] + A^l_u(\psi; \psi^\dagger)[\psi_\mathrm{h} - \psi]\right\}}{l!}(\psi_\mathrm{h} - \psi)^l + \mathcal{O}(\psi_\mathrm{h} - \psi)^t. \tag{3.67}$$

$A(\psi; \psi^\dagger) = 0$ by definition of the WR problem. Therefore an estimate for the error in the QoI (goal functional) is given by an adjoint WR and a linearisation error ϵ:

$$J(\psi_\mathrm{h}) - J(\psi) = -A(\psi_\mathrm{h}; \psi^\dagger) + \epsilon, \tag{3.68}$$

where:

$$\epsilon = \frac{\left\{J''_u(\psi)[\psi_\mathrm{h} - \psi] + A''_u(\psi; \psi^\dagger)[\psi_\mathrm{h} - \psi]\right\}}{2!}(\psi_\mathrm{h} - \psi)^2 + \mathcal{O}(\psi_\mathrm{h} - \psi)^3. \tag{3.69}$$

Two examples are discussed in the sections that follow. Section 3.3.2 discusses the example where $A(u; v) = 0$ represents the fixed source neutron transport problem with $J(u)$ representing a detector function goal value. Section 3.3.3 discusses the example where $A(u; v) = 0$ represents a neutron transport eigenvalue ($\mathrm{K_{eff}}$ criticality) problem with $J(u)$ being the eigenvalue. It is assumed for now that any iteration error in the scatter/eigenvalue problems are negligible (see section 3.3.4 for further discussion). To simplify the discussion, vacuum boundary conditions are assumed. In this case the space of allowable functions can be considered as a vector space. The subscript 0_i will be used to note a space where the vacuum boundary conditions are defined

on the left boundary for $\mu > 0$ and on the right boundary for $\mu < 0$. The opposite will be noted by the subscript 0_o.

3.3.2 Fixed source problem with detector function goal

In this case the semi-linear form is, in fact, linear and defined as follows:

$$A(u; v) = \langle WQ, v\rangle_{\bar{V}} - \langle \mathcal{K}u, v\rangle_{\bar{V}} \text{ with } u \in \bar{U}_{0_i} \text{ and } v \in \bar{V}. \tag{3.70}$$

The problem to be solved is the following:

$$\begin{aligned} &\text{Find } \Psi \in \bar{U}_{0_i} := D(\mathcal{K})[102] \text{ such that:} \\ &\qquad A(\Psi; v) = \langle WQ, v\rangle_{\bar{V}} - \langle \mathcal{K}\Psi, v\rangle_{\bar{V}} = 0 \quad \forall v \in \bar{V}. \end{aligned} \tag{3.71}$$

The goal quantity is a linear functional on $\bar{U}_{0_i}$:

$$J(\Psi) = \langle \Psi, W\boldsymbol{\Sigma}_d\rangle_{\bar{V}}. \tag{3.72}$$

$\boldsymbol{\Sigma}_d$ is an N sized vector, each element m is the detector cross section distribution for angle m, denoted by: $\Sigma_{dm} \in L^2$. Ψ is the solution to equation (3.71).

Hence, equation (3.64) becomes:

$$\begin{aligned} L'_u(\Psi, \Psi^\dagger)[y] = \frac{\partial}{\partial u}\left(\langle \Psi, W\boldsymbol{\Sigma}_d\rangle_{\bar{V}} + \langle WQ, \Psi^\dagger\rangle_{\bar{V}} - \langle \mathcal{K}\Psi, \Psi^\dagger\rangle_{\bar{V}}\right)[y] = 0 \\ \forall y \in \bar{U}_{0_i}\ ,\ \Psi^\dagger \in \bar{V} \text{ and } \Psi \text{ a solution to equation (3.71)}, \end{aligned} \tag{3.73}$$

which reduces to:

$$\begin{aligned} \frac{\partial}{\partial u}\left(\langle \Psi, W\boldsymbol{\Sigma}_d\rangle_{\bar{V}} - \langle \mathcal{K}\Psi, \Psi^\dagger\rangle_{\bar{V}}\right)[y] = 0 \\ \forall y \in \bar{U}_{0_i}\ ,\ \Psi^\dagger \in \bar{V} \text{ and } \Psi \text{ a solution to equation (3.71)}. \end{aligned} \tag{3.74}$$

$\frac{\partial}{\partial u}(\cdot)[y]$ denotes the partial Gateaux derivative with respect to parameter u in the direction y. In equations (3.73) and (3.74) u is denoted by Ψ, since Ψ is the value taken by u at the stationary point.

It can be shown that equation (3.74) holds for the solution of the following adjoint equation:

$$\langle v, W\Sigma_{\mathrm{d}}\rangle_{\bar{V}} - \langle v, \mathcal{K}^{\dagger}\Psi^{\dagger}\rangle_{\bar{V}} = 0 \quad \forall v \in \bar{V} \text{ and } \Psi^{\dagger} \in D(\mathcal{K}^{\dagger}) \subset \bar{V}, \tag{3.75}$$

where $D(\mathcal{K}^{\dagger})$ is the space of functions in the domain of the adjoint operator. Note that $D(\mathcal{K}^{\dagger}) \neq D(\mathcal{K})$ [102]. This is explained in appendix B.

Equation (3.74) can now be used to obtain an expression for the error in the linear functional $J(\Psi_{\mathrm{h}}) - J(\Psi)$ where Ψ_{h} is the DD approximation. This is done by taking $y = \Psi_{\mathrm{h}} - \Psi$ in equation (3.73) and following the same logic as in equations (3.65)-(3.69) to arrive at:

$$J(\Psi_{\mathrm{h}}) - J(\Psi) = -A(\Psi_{\mathrm{h}}; \Psi^{\dagger}) + \epsilon. \tag{3.76}$$

$\epsilon = 0$ due to linearity, consequently equation (3.76) reduces to:

$$J(\Psi_{\mathrm{h}}) - J(\Psi) = -\langle WQ - \mathcal{K}\Psi_{\mathrm{h}}, \Psi^{\dagger}\rangle_{\bar{V}} \tag{3.77}$$

$$= -\langle R_{\mathrm{h}}, \Psi^{\dagger}\rangle_{\bar{V}}. \tag{3.78}$$

Hence the error in the goal quantity can be obtained by weighting the residual of the DD solution with the solution to the adjoint equation defined by equation (3.75).

3.3.3 Eigenvalue problem with eigenvalue goal

In this case definitions (3.50)-(3.54) are recalled and the semi-linear form is defined as [88, 90]:

$$A(\hat{u}; \hat{v}) = B(u, v) - \lambda F(u, v) + \tau(1 - F(u, u)) \quad : \quad \hat{\bar{U}} \times \hat{\bar{V}} \to \mathbb{R}, \tag{3.79}$$

where $\hat{u} \in \hat{\bar{U}}$ and $\hat{v} \in \hat{\bar{V}}$.

The problem to be solved is:

Find $\hat{\Psi} \in \hat{\bar{U}}_{0_i}$ such that:

$$A(\hat{\Psi}; \hat{v}) = B(\Psi, v) - \lambda F(\Psi, v) + \tau(1 - F(\Psi, \Psi)) = 0 \quad \forall \hat{v} \in \hat{\bar{V}}. \tag{3.80}$$

The boundary conditions are contained in the space of allowable functions $\hat{\bar{U}}_{0_i}$.

The goal quantity is the non-linear functional on $\hat{\bar{U}}_{0_i}$ defined by [88, 90]:

$$J(\hat{\Psi}) = \lambda F(\Psi, \Psi) = \lambda. \tag{3.81}$$

Therefore equation (3.64) becomes:

$$L'_{\hat{u}}(\hat{\Psi}, \hat{\Psi}^\dagger) = J'_{\hat{u}}(\hat{\Psi})[\hat{y}] + A'_{\hat{u}}(\hat{\Psi}; \hat{\Psi}^\dagger)[\hat{y}] = 0$$
$$\forall \hat{y} \in \hat{\bar{U}}_{0_i}, \hat{\Psi}^\dagger \in \hat{\bar{V}} \text{ and } \hat{\Psi} \text{ being the solution to equation (3.80).} \tag{3.82}$$

Setting $\hat{y} = \{y, \xi\}$ and $\hat{\Psi}^\dagger = \{\Psi^\dagger, \lambda^\dagger\}$ this becomes:

$$\lambda\{F(\Psi, y) + F(y, \Psi)\} + \xi F(\Psi, \Psi) = \lambda F(y, \Psi^\dagger) - B(y, \Psi^\dagger) + \xi F(\Psi, \Psi^\dagger) + \lambda^\dagger\{F(\Psi, y) + F(y, \Psi)\}$$
$$\forall \hat{y} \in \hat{\bar{U}}_{0_i}, \hat{\Psi}^\dagger \in \hat{\bar{V}} \text{ and } \hat{\Psi} \text{ the solution to equation (3.80),} \tag{3.83}$$

which rearranges to:

$$B(y, \Psi^\dagger) - \lambda F(y, \Psi^\dagger) + \xi\{F(\Psi, \Psi) - F(\Psi, \Psi^\dagger)\} + (\lambda - \lambda^\dagger)\{F(\Psi, y) + F(y, \Psi)\} = 0$$
$$\forall \hat{y} \in \hat{\bar{U}}_{0_i}, \hat{\Psi}^\dagger \in \hat{\bar{V}} \text{ and } \hat{\Psi} \text{ the solution to equation (3.80).} \tag{3.84}$$

This is the same as in Lathouwers's paper [90], but equation (3.84) is the non-discretised formulation of the WR.

It can be seen from equation (3.84) that this equation is satisfied by using the value of $\hat{\Psi}^\dagger = \{\Psi^\dagger, \lambda^\dagger\}$ given by the solution to the adjoint eigenvalue equation, with homogeneous essential boundary conditions and the following specific normalisation condition:

Find $\Psi^\dagger \in \bar{U}_{0_o}$ and $\lambda^\dagger \in \mathbb{R}$ such that:

$$\mathcal{K}^\dagger \Psi^\dagger = \lambda^\dagger \mathcal{G}^\dagger \Psi^\dagger \tag{3.85}$$
$$\langle \mathcal{G}^\dagger \Psi^\dagger, \Psi \rangle_{\bar{V}} = 1. \tag{3.86}$$

$\mathcal{G}^\dagger$ is the adjoint of the operator $\mathcal{G} = W\mathcal{F}$. $\mathcal{G}$ is a linear bounded operator on $\bar{V}$ and so its adjoint operator is easily defined by [130]:

$$\langle \mathcal{G}u, v \rangle_{\bar{V}} = \langle u, \mathcal{G}^\dagger v \rangle_{\bar{V}} \quad \forall v, u \in \bar{V}. \tag{3.87}$$

Therefore $\langle \mathcal{G}^\dagger \Psi^\dagger, \Psi \rangle_{\bar{V}} \equiv F(\Psi, \Psi^\dagger)$. In fact, $\mathcal{G}$ is self-adjoint so that $\mathcal{G} = \mathcal{G}^\dagger$. However, $\Psi^\dagger$ is still

restricted to be part of the subspace $\bar{U}_{0_o}$, due to the $\mathcal{K}^\dagger$ operator. That is, for the solution of equations (3.85)-(3.86) to satisfy (3.84) the term $B(y, \Psi^\dagger) = \langle \mathcal{K}y, \Psi^\dagger \rangle_{\bar{V}}$ in equation (3.84) would have to equal $\langle y, \mathcal{K}^\dagger \Psi^\dagger \rangle_{\bar{V}}$. It is shown in appendix B that for this to be true, $\Psi^\dagger \in \bar{U}_{0_o}$.

The integral statement of (3.85) and (3.86) can be expressed as:

$$\langle \mathcal{K}^\dagger \Psi^\dagger, v \rangle_{\bar{V}} - \lambda^\dagger \langle \mathcal{G}^\dagger \Psi^\dagger, v \rangle_{\bar{V}} + \tau \left(1 - \langle \mathcal{G}^\dagger \Psi^\dagger, \Psi \rangle_{\bar{V}} \right) = 0 \quad \forall \hat{v} := \{v, \tau\} \in \hat{\bar{V}}. \tag{3.88}$$

Bell and Glasstone show that $\lambda = \lambda^\dagger$ [41]. Note that the solution $\hat{\Psi}^\dagger$ is the same for all $\hat{v} \in \hat{\bar{V}}$. Hence the solution will remain the same if $\hat{v}$ was replaced by any $\hat{y} \in \hat{\bar{U}}_{0_i}$, since $\hat{\bar{U}}_{0_i}$ forms a dense subset of $\hat{\bar{V}}$. When taken from this subset, the $\langle \mathcal{K}^\dagger \Psi^\dagger, y \rangle_{\bar{V}}$ term can be replaced by $\langle \mathcal{K}y, \Psi^\dagger \rangle_{\bar{V}}$. With these considerations taken into account, the first 3 terms of equation (3.84) are equivalent to equation (3.88) with the weighting function being taken from a particular subspace of $\hat{\bar{V}} : \hat{\bar{U}}_{0_i}$. Then the fact that the adjoint and forward eigenvalues are the same means that the final term in (3.84) also goes to zero. Hence, the solution to the integral statement of the adjoint eigenvalue problem (3.88) gives the value $\hat{\Psi}^\dagger$ required in equation (3.82).

Now that the values of both $\hat{\Psi}$ and $\hat{\Psi}^\dagger$ are known, it is desirable to have equation (3.82) in a form such that it gives an approximation to $J(\Psi_h) - J(\Psi)$. This involves considering equation (3.82) for the case where $\hat{y} = e = \hat{\Psi}_h - \hat{\Psi} \in \hat{\bar{U}}_{0_i}$ and following the logic of equations (3.65)-(3.69) to arrive at the following error estimator:

$$J(\hat{\Psi}_h) - J(\hat{\Psi}) = -A(\hat{\Psi}_h, \hat{\Psi}^\dagger) + \epsilon, \tag{3.89}$$

where

$$\epsilon = \frac{\left\{ J''_{\hat{\Psi}}(\hat{\Psi})[\hat{\Psi}_h - \hat{\Psi}] + A''_{\hat{\Psi}}(\hat{\Psi}; \hat{\Psi}^\dagger)[\hat{\Psi}_h - \hat{\Psi}] \right\}}{2!} (\hat{\Psi}_h - \hat{\Psi})^2 + \mathcal{O}(\hat{\Psi}_h - \hat{\Psi})^3 \tag{3.90}$$

and

$$\begin{aligned} A(\hat{\Psi}_h; \hat{\Psi}^\dagger) &= B(\Psi_h, \Psi^\dagger) - \lambda F(\Psi_h, \Psi^\dagger) + \lambda^\dagger (1 - F(\Psi_h, \Psi_h)) \\ &= B(\Psi_h, \Psi^\dagger) - \lambda F(\Psi_h, \Psi^\dagger), \end{aligned} \tag{3.91}$$

since $F(\Psi_h, \Psi_h) = 1$ is the normalisation applied in the numerical solver.

It is assumed that ϵ is small compared to the WR part since it is second order with respect to the discretisation error. However, there are cases where ϵ is not small compared to the DWR.

For example, where the second order derivatives are very large. This effect should reduce upon refinement as the error in the forward solution is reduced.

3.3.4 Calculating the adjoint solution in practice

In general, the analytical solution to the adjoint equation will not be available for use in the DWR error estimators derived in this chapter. Therefore, an approximation to $\Psi^\dagger$ must be calculated.

It can be seen from equations (3.75) and (3.88) that the discrete operators obtained by the DD-S_N discretisation of the forward equations cannot simply be transposed to give a discrete approximation of the adjoint equation (as in the work of Venditti and Darmofal [104–107]). This is because the discretisation would not be consistent with the strong formulation of the continuous adjoint equations ((3.75) and (3.88) for fixed source and eigenvalue problems respectively). The DD method is therefore not adjoint consistent, in that the transpose of the forward discretised operator is not equivalent to the operator that results from discretising the continuous adjoint operator [131]. Although the weighting functions used in the DD approximation of the forward equations are members of $\bar{V}$ and are therefore consistent with the strong formulation of the forward equations (equations (3.39) and (3.52) for the fixed source and eigenvalue problems respectively), they are not members of $D(\mathcal{K}^\dagger)$. Solving the DD-S_N discretisation of the strong formulation of the adjoint equation with appropriate boundary conditions gives an approximate adjoint solution in the correct phase space, $D(\mathcal{K}^\dagger)$, and the discretised equations are consistent with the continuous equations.

If the lowest order DD approximation is used to obtained Ψ_h, then due to the fact that Ψ_h is generated from a WR method that uses piece-wise constant test functions, a piece-wise constant approximation to $\Psi^\dagger$ would result in an error estimate of zero. It is similar reasoning that means that the adjoint solution must be computed in a different space from the forward equation in Bubnov-Galerkin FE approaches [88–90]. Either a higher-order FE needs to be used or the number of cells in the discretisation increased (or both). However, if an approximate adjoint solution is obtained from a DD discretisation of the adjoint equations, using the same order and the same mesh as the forward equation, a piece-wise linear approximation to the adjoint solution is obtained, not piece-wise constant. Hence the approximate adjoint solution is not in the space of test functions and the DWR using this approximation will not give zero.

Chen and Gunzburger [102] note that because FV methods do not naturally fit into variational forms, the Galerkin orthogonality problem does not apply. However, in the DD case, this chapter has shown that the residual error given by the approximate solution is not orthogonal to the trial function space because the test function space is different from it (piece-wise constant rather than piece-wise linear when $p = 1$). Galerkin orthogonality does still apply to approximations of the adjoint solution that lie in the test function space (piece-wise constant approximations in the case of $p = 1$).

As was noted by Chen and Gunzburger [102], the more accurate the approximation to the adjoint solution, the more accurate the error estimator, thus there may be benefits to obtaining a higher-order solution to the adjoint equation, despite the fact that the estimator is non-zero when the solution is sought in the same space as the forward equation. In this work the estimated errors calculated by using an adjoint computed on the same mesh, one twice as refined, and one using a higher-order DD approximation using Hennart's method [56] are compared.

In addition to this, it must be noted that instead of weighting the residual by $\Psi^\dagger$, one could have weighted the residual by $\Psi^\dagger - \bar{\Psi}^\dagger$. Where $\bar{\Psi}^\dagger$ is the cell-wise projection of the solution $\Psi^\dagger$ into the space of $(p-1)$ order cell-wise Legendre polynomials. For a cell with $p = 1$, $\bar{\Psi}^\dagger$, represents a cell-wise average value. Assuming zero iteration error, these two choices result in identical answers. This can be explained by taking the $p = 1$ case as an example. Any constant component of the adjoint solution weighted with the residual will go to zero, due to the fact that the test functions are piece-wise constant, hence the residual of the DD solution is orthogonal to this space. In reality, the component weighted by the average will give a measure of the iteration error [19, 20]. This thesis does not investigate the use of goal-based iteration error approximation for use in solver iteration exit criteria. In this work, iterations continue until the residual weighted by unity is as close to machine precision as possible (sometimes this is of the order of about 10^{-12} for a machine precision of 10^{-15} due to the loss of accuracy in calculating the residual). The use of goal-based DWR error estimators for stopping criteria in FE methods was investigated by Rannacher et al. [19, 20] where it was deemed suitable for the error due to iteration to be of the same order as the error due to mesh discretisation. This would be an interesting avenue for future work in the case of the DD scheme, but is outside the scope of this thesis.

3.4 QoI approximation improvement by DWR error correction

It is possible to use the DWR error estimator as a mesh iteration stopping criteria when performing uniform h, p or hp refinement. Another option would be to use the DWR error estimator to correct the approximate value for the functional of interest obtained using the solution to the discretised PDE. DWR correction of functionals has, to the author's knowledge, not been applied to the field of neutron transport before, but has been applied in other fields [97–101, 104–108]. The theory behind using the DWR error estimate to correct the approximation to the goal functional is described in this section.

In reality, the analytical adjoint equation given by equation (3.64) can only be derived from a numerical solution to the forward equation. The analytical solution to this equation shall be denoted by $\tilde{\Psi}^{\dagger}$ and the equation resulting from an approximate forward solution, $\Psi_{\mathrm{h},p}$, as:

$$J'_{\Psi}(\Psi_{\mathrm{h},p})[v] + A'_{\Psi}(\Psi_{\mathrm{h},p}; \tilde{\Psi}^{\dagger})[v] = 0 \quad \forall v \in \bar{V}. \tag{3.92}$$

Further, this will be approximated numerically using the DD scheme:

$$J'_{\Psi}(\Psi_{\mathrm{h},p})[v] + A'_{\Psi}(\Psi_{\mathrm{h},p}; \tilde{\Psi}^{\dagger}_{h,p})[v] = 0 \quad \forall v \in \bar{V}_h \subset \bar{V}. \tag{3.93}$$

Hence the error in the approximation to the QoI is given by:

$$J(\Psi_{\mathrm{h},p}) - J(\Psi) = -A(\Psi_{\mathrm{h},p}; \tilde{\Psi}^{\dagger}_{\mathrm{h},p}) - A(\Psi_{\mathrm{h},p}; \tilde{\Psi}^{\dagger} - \tilde{\Psi}^{\dagger}_{\mathrm{h},p}) - A(\Psi_{\mathrm{h},p}; \Psi^{\dagger} - \tilde{\Psi}^{\dagger}) + \epsilon. \tag{3.94}$$

All terms in ϵ are $\mathcal{O}(\Psi_{\mathrm{h},p} - \Psi)^2$ or smaller. The second and third term are a residual weighted by an error term and so will be smaller than the first term. The dominant error term when the approximation to the QoI is calculated using $\Psi_{\mathrm{h},p}$ is therefore $A(\Psi_{\mathrm{h},p}; \tilde{\Psi}^{\dagger}_{\mathrm{h},p})$ and this should give a good approximation of the error in $J(\Psi_{\mathrm{h},p})$. This value can be used to obtain a more accurate approximation to the QoI. In this case the approximation to the QoI becomes:

$$J(\Psi) \approx J(\Psi_{\mathrm{h},p}) + A(\Psi_{\mathrm{h},p}; \tilde{\Psi}^{\dagger}_{\mathrm{h},p}). \tag{3.95}$$

In the literature, the dominant error term in fully discrete formulations has been referred to as the *computable correction term* [96]. Though in the CFD literature it often refers to a weighted residual problem where the discrete adjoint operator matrices are derived by taking the transpose

of the forward problem's matrix [96, 105]. This is not the case in the work of this thesis since the 1-D DD discretisation is not adjoint consistent (see section 3.3).

The remaining error is then given by:

$$\text{Remaining error} = A(\Psi_{\mathrm{h},p}; \tilde{\Psi}^{\dagger} - \tilde{\Psi}^{\dagger}_{\mathrm{h},p}) + A(\Psi_{\mathrm{h},p}; \Psi^{\dagger} - \tilde{\Psi}^{\dagger}) - \epsilon. \tag{3.96}$$

ϵ is second order with respect to the error in the forward solution, and the first two terms in (3.96) are combinations of a residual and an error term. Given that the residual is proportional to the error in the solution, the error in the QoI after correction is proportional to the product of two error terms. This means that the error in the DWR-corrected value for the QoI should give super-convergent results with h refinement. Super-convergence refers to the increased convergence of the functional compared to the flux solution and the non-corrected functional approximation [100]. Super-convergence of non-corrected integral functions is inherent in Bubnov-Galerkin finite element schemes [98].

3.5 Application to test cases

The theory derived and described in the previous sections was implemented in a 1-D arbitrary order DD-S_N code. The implementation is described in section 3.5.1. The results of applying the code to the 1-D test cases (described in section A.1) are shown and discussed in section 3.5.2.

3.5.1 Implementation

The code written for this work is a 1-D, one-group, fixed source and eigenvalue DD-S_N code. The spatial discretisation is achieved by using an arbitrary order DD formulation as described in section 3.2.2.

After the forward equations are solved on the simple mesh, one of three options for the adjoint approximation is calculated. Then the DWR inner product is calculated over each cell individually. The cell values are summed over all cells in order to give an approximation to the error. Two summed values are calculated: the sum of the cell-wise DWR quantities and the sum of the absolute values of the cell-wise DWR quantities. The latter should be a more conservative estimate of the error than the former.

If the forward solution is expressed as $\psi_{\mathrm{h},p}$ then the resulting residual from that is expressed as $R_{\mathrm{h},p}$. h represents the mesh on which the solution is calculated and p represents the order of accuracy at which the solution is calculated in the cells throughout the mesh. Three different DWR error estimators are tested in this chapter:

$$E_1 = \sum_{i=1}^{nc} E_{1i} \quad \text{where} \quad E_{1i} = \sum_{m=1}^{N} w_m (R_{\mathrm{h},p,m}, \psi^{\dagger}_{m,\mathrm{h},p})_i, \tag{3.97}$$

$$E_2 = \sum_{i=1}^{nc} E_{2i} \quad \text{where} \quad E_{2i} = \sum_{m=1}^{N} w_m (R_{\mathrm{h},p,m}, \psi^{\dagger}_{m,\frac{\mathrm{h}}{2},p})_i, \tag{3.98}$$

$$E_3 = \sum_{i=1}^{nc} E_{3i} \quad \text{where} \quad E_{3i} = \sum_{m=1}^{N} w_m (R_{\mathrm{h},p,m}, \psi^{\dagger}_{m,\mathrm{h},p+1})_i. \tag{3.99}$$

$(\cdot,\cdot)_i$ is the L^2 inner product over cell i. Using E_1 and E_3 as functional correction terms is also tested.

3.5.2 Results

For each test case, the effectivity index (I) of the three different DWR error estimates are plotted against the number of cells in the approximation. Uniform h refinement is used for these graphs and the within-cell order is kept constant and first order throughout (lowest order DD scheme: DD1). The effectivity index is the ratio of the estimated error in the QoI to the analytically calculated error in the QoI. The convergence graph of the analytical error in the approximate functional value is compared to that of the analytical error in the DWR corrected approximate value.

3.5.2.1 Test case 1

It was verified, for test case 1a, that the DWR using an analytically calculated adjoint solution was the same as the difference between the analytically calculated goal quantity and the numerically calculated goal quantity. Any discrepancy was due to loss of numerical accuracy in calculating these quantities by using floating point operations. This conclusion was drawn because the equality improved with the use of quadruple precision floating point computer arithmetic.

In test case 1a, due to the symmetry of the S_2 adjoint and forward equations, there is a large DWR for the right-going ordinate where there is a small DWR value for the left-going ordinate

and vice-versa. The resulting error estimates are therefore fairly uniform throughout the domain. This is shown in figure 3.2a, which shows the spatial variation of the value of local error estimators for the initial mesh. Each point represents the value of the error estimator for one cell, and is placed on the graph at the position of the midpoint of the corresponding cell (fig. 3.2a). The distribution is shown for the initial mesh of 16 cells. Figure 3.2a also shows that an improved approximation of the adjoint in the DWR gives a more accurate distribution and overall error estimator. The error estimator that uses the same mesh cell number and mesh order as the forward equation to calculate the adjoint gives a uniform distribution for this example. The improved approximations give a slight curve, hence revealing a subtle variation in the importance of different regions for refinement. In this case, the best approximation of the true error is given by the DWR using the adjoint computed on the same number of cells as the forward equation but with an increased mesh order. It also gives a more conservative estimate since it slightly over-estimates the error rather than underestimating it.

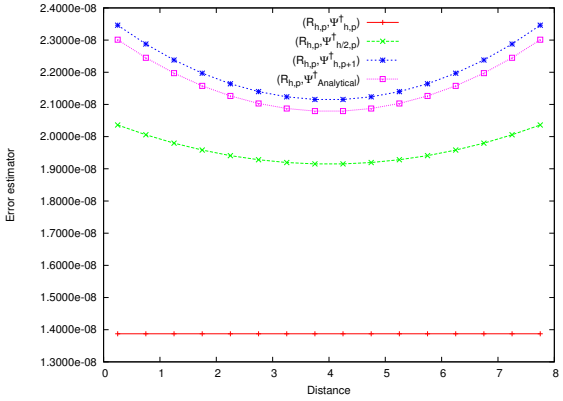
Figure 3.2b shows that there is negligible difference in the values of the error estimators for the initial uniform mesh in test case 1b. Figure 3.2b also shows a significant preference for refinement in the first cm: the region of interest. This is as expected, given that the goal is the value of the scalar flux there.

The error correction reduces the absolute error in the result for a given mesh refinement. The increase in the correction obtained when using the higher order adjoint is less when the goal is localised (figure 3.2c) compared to when the goal is a quantity throughout the whole of the domain (figure 3.2d). The rate of convergence is also improved in the h refinement case (as was predicted in section 3.4).

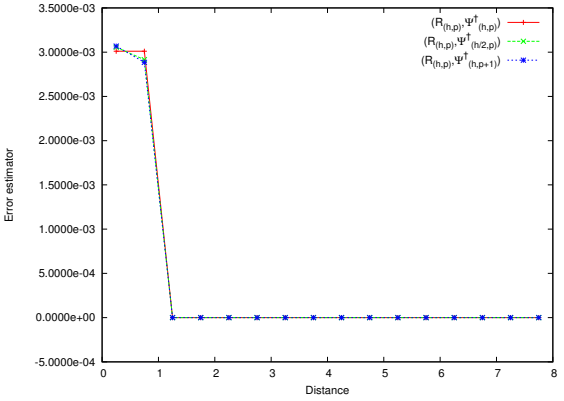
3.5.2.2 Test case 2

The results here are very similar to those of test case 1 (section 3.5.2.1). There is very little absolute difference between the value of the different error estimators for test case 2a. This is shown in figure 3.3a. However, it is clear that a more accurate adjoint solution gives a more accurate spatial distribution of the goal-based error estimator.

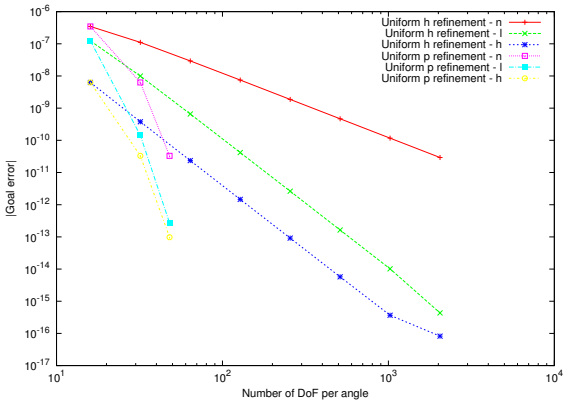
The error correction is limited at a certain point (figures 3.3b and 3.3c for test case 2a and 2b respectively) due to the fact that machine precision limits the extent to which the scatter source problem can converge. Before this point however, an improved convergence rate is observed



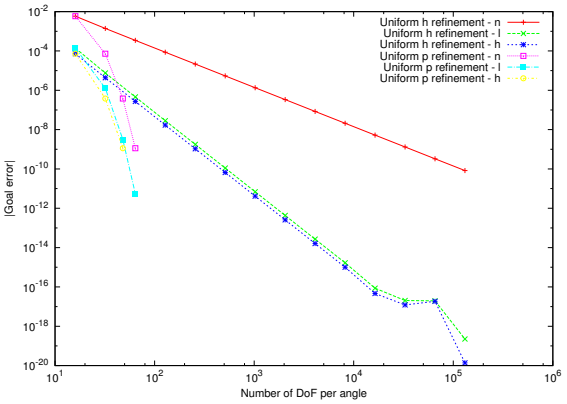
(A) Test case 1a: Spatial variation of error estimates as a function of space for a 16 cell mesh.



(B) Test case 1b: Spatial variation of error estimates as a function of space for 16 cell mesh.



(C) Test case 1a: Uniform refinement with error correction (n: no error correction, l: low order error correction, h: high order error correction)



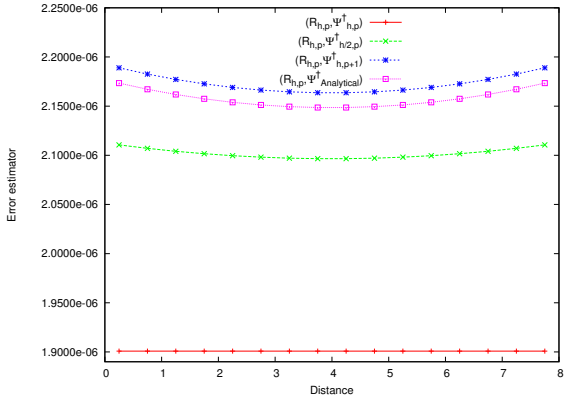
(D) Test case 1b: Uniform refinement with error correction (n: no error correction, l: low order error correction, h: high order error correction)

FIGURE 3.2: Results for test case 1

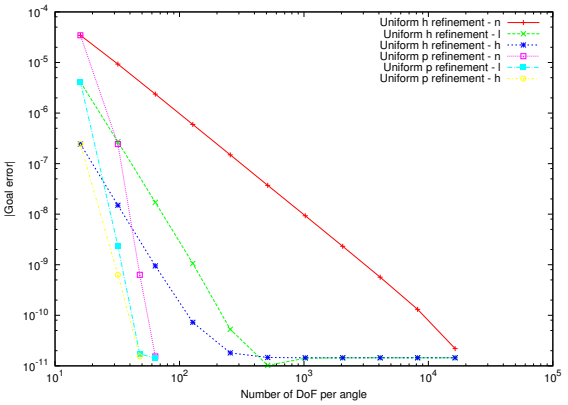
in h refinement for the corrected value compared to the non-corrected result. As for test case 1, absolute error reduction for a given refinement is observed upon adding the correction term to the goal quantities of test cases 2a and 2b, for both h and p refinement. The increase in correction when using a higher order adjoint solution is less for test case 2b (fig. 3.3b) compared to 2a (fig. 3.3c).

3.5.2.3 Test case 3a

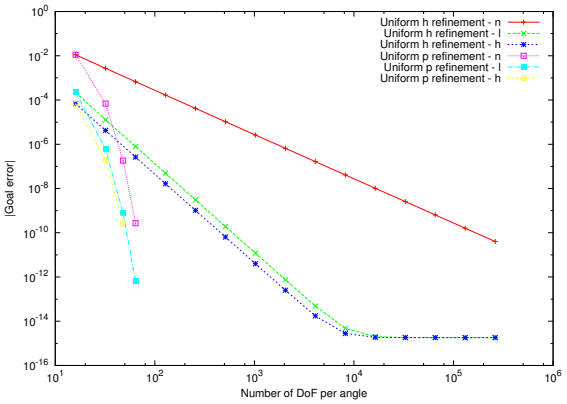
Figure 3.4a shows the accuracy of the error estimators at various uniform h refinement mesh iterations (using DD1 for the forward equation in all cases). Figure 3.4a shows that the error estimator using the adjoint solution of the same accuracy as the forward solution gives a very bad estimate of the error on the 64 cell mesh, by vastly over-estimating it. To better see how the accuracy of the error estimators vary with mesh refinement this first point on the graph is



(A) Test case 2a: Spatial variation of error estimates as a function of space for a 16 cell mesh.



(B) Test case 2a: Uniform refinement with error correction (n: no error correction, l: low order error correction, h: high order error correction)



(C) Test case 2b: Uniform refinement with error correction (n: no error correction, l: low order error correction, h: high order error correction)

FIGURE 3.3: Results for test case 2

removed and the graph re-plotted in figure 3.4b. The error estimator consistently over-estimates the error and converges to a constant value of over-estimation. This is is because this error estimator was calculated by summing the absolute value of all of the local error estimators. We have that:

$$|J(\psi) - J(\psi_{\mathrm{h}})| = \left| \sum_{i=1}^{n} \sum_{m=1}^{N} (R_{\mathrm{h},m}, \psi_m^{\dagger})_i \right| \leq \sum_{\mathrm{i}=1}^{\mathrm{n}} \left| \sum_{m=1}^{N} |(R_{\mathrm{h},m}, \psi_m^{\dagger})_i \right| \tag{3.100}$$

Hence if $(R_{\mathrm{h}}, \psi^{\dagger}_{approx.})$ is a good approximation to $(R_{\mathrm{h}}, \psi^{\dagger})_{cell}$, an error estimator calculated in this way will tend to over-estimate the error unless the error has the same sign throughout the domain. This is desirable if a conservative stopping criteria is sought, but it may lead to the mesh being overly refined. If the the absolute value is taken after summing the individual cell contributions rather than before, the graph of figure 3.4c is obtained. This graph shows that all of the DWR error estimators tested converge to the true error upon mesh refinement. It also shows that the best approximation is given by the more accurate solution calculated on the

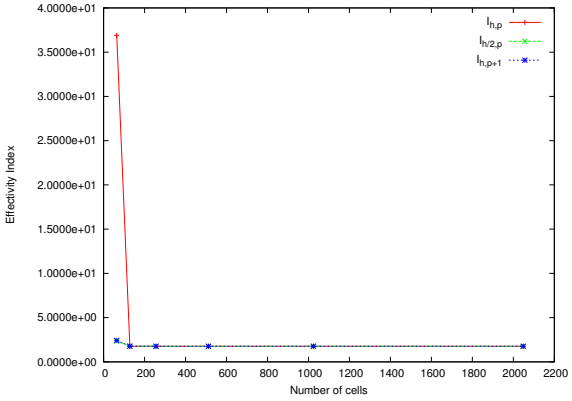
same mesh as the forward equation but using higher-order elements, it also consistently provides a conservative estimate in this example. The second best approximation is that given by the DWR using an adjoint calculated on a mesh that is twice as refined as the mesh used for the forward equation, using the first-order elements. Finally the worst estimate is given by using the adjoint solution calculated with the same cell number and order as the forward equation.

The results of coupling uniform h refinement or p refinement with DWR error correction is shown in figure 3.4d. For the initial mesh, the lower order error estimate is so inaccurate that using it as an error correction makes the approximated solution worse. This is a danger of the error correction procedure at low mesh resolutions where there are areas of the domain that do not satisfy the negative flux criterion. In this case, the error in the error estimator is reduced such that a significant error correction is given for the first uniform h refined mesh. An error correction (instead of error augmentation) is obtained for p refinement at the second mesh iteration after the original mesh. The higher order correction provided a mesh correction for all mesh iterations, including the original mesh.

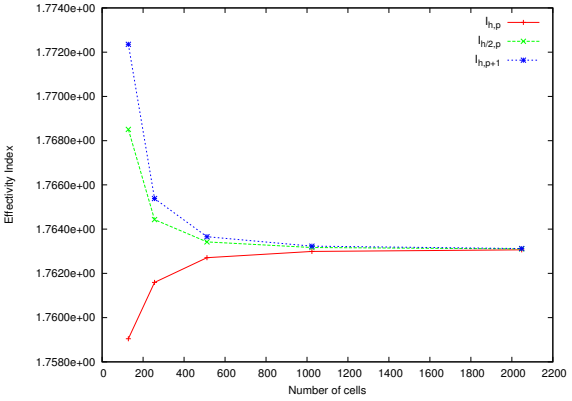
3.5.2.4 Test case 3b

The DWR calculated using the same number of cells and the same order within each cell as that used for the forward equation overestimates the error by at least an order of magnitude (figures 3.5a and 3.5b). Figure 3.5c shows only the effectivity indices calculated using an adjoint computed on a mesh twice as refined, and the case where it is computed on a mesh using everywhere higher-order elements. The error is much better estimated, particularly by the higher-order approximation, until the 4th mesh iteration. At this point, both estimators significantly underestimate the error.

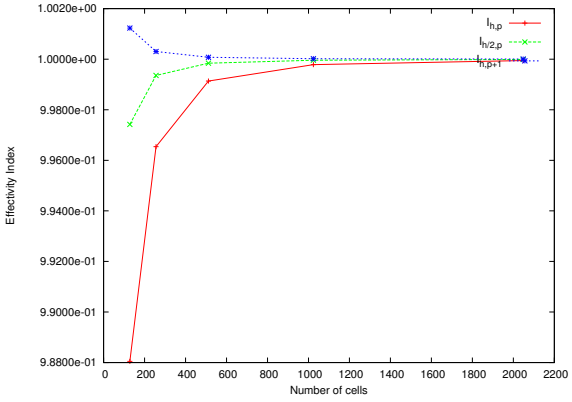
The comments made above on figures 3.5a-3.5c can be explained by figure 3.6 and figure 3.7. In the zeroth mesh (figure 3.6a) it is clear that in the problem regions, i.e. the first cm where the goal region is, and the second cm where there is a high value of error, the less accurate error estimator drastically over-estimates the error. This is most likely due to the fact that the DD solution oscillates in regions of high total cross sections and mesh cell sizes that are not small enough. This is shown by looking at figure 3.7. This figure plots the smallest positive ordinate of the adjoint solution. It is the average value per cell that is plotted as a function of cell number from 0 for the left-most cell to $nc - 1$ on the right. It is clear that there are oscillations in the first cm when the same mesh and order is used to calculate this flux ordinate (figure 3.7a),



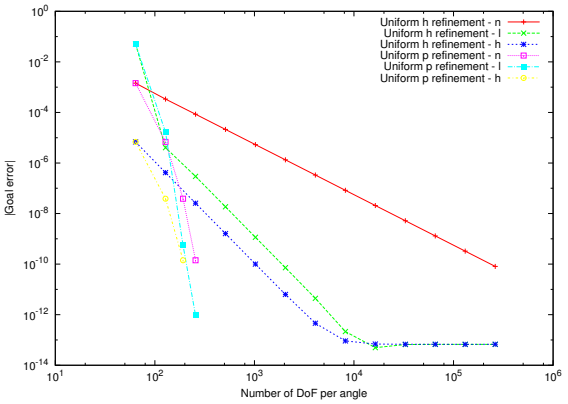
(A) $\frac{\sum_{i=1}^{nc} |\sum_{m=1}^{N} (R_{m,\mathrm{h}}, \psi^{\dagger}_{m,approx})_i|}{|truegoalerror|}$ for the first 6 meshes generated by uniform refinement from a uniform 64 cell starting mesh.



(B) $\frac{\sum_{i=1}^{nc} |\sum_{m=1}^{N} (R_{m,\mathrm{h}}, \psi^{\dagger}_{m,approx})_i|}{|truegoalerror|}$ for the first 5 meshes generated by uniform refinement from a uniform 128 cell starting mesh.



(C) $\left| \frac{\sum_{i=1}^{nc} \sum_{m=1}^{N} (R_{m,\mathrm{h}}, \psi^{\dagger}_{m,approx})_i}{truegoalerror} \right|$ for the first 5 meshes generated by uniform refinement from a uniform 128 cell starting mesh.



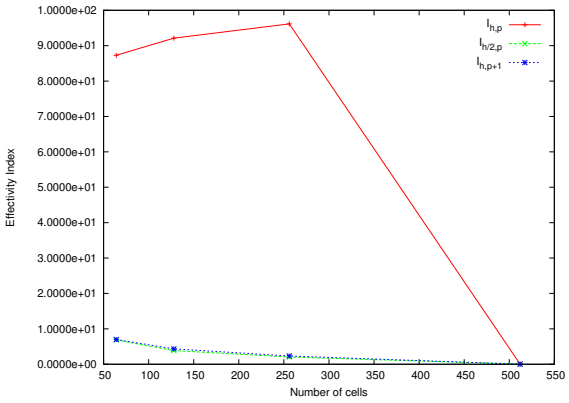
(D) Uniform refinement and error correction (n: no error correction, l: low order error correction, h: high order error correction).

FIGURE 3.4: Results for test case 3a

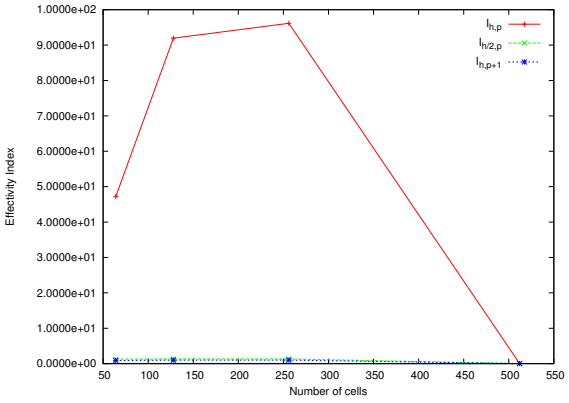
the magnitude of the oscillations are smaller throughout most of the first cm when the mesh is refined everywhere once (figure 3.7b) and even more so when the order is increased in each cell (figure 3.7c). The fact that the error estimator deteriorates significantly by mesh iteration 3 (see figure 3.5c) is explained by looking at figure 3.6d. Here, all local error indicators have a value below that which can be reliably represented by double floating point precision, so the results cannot be trusted.

The result of coupling uniform h and p refinement with DWR error correction for test case 3b are plotted in figure 3.5d. The goal is in the very under-resolved region of the domain, so the issues due to errors in the error correction are amplified compared to in test case 3a (figure 3.4d). The lower order error estimation gives an error augmentation in the calculated approximation to the QoI for each uniform h and p refined mesh iteration. The higher order error estimation, however,

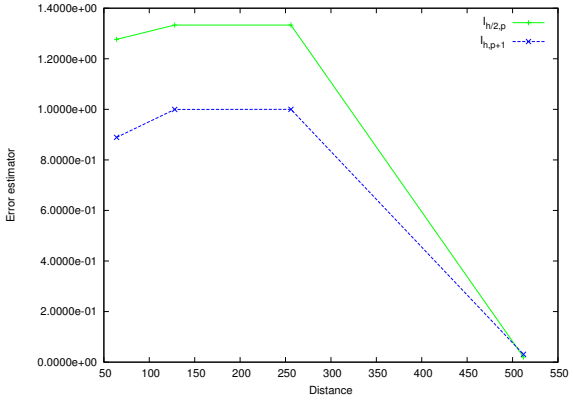
gives an error correction to the uniform h and p refined quantities at each mesh iteration.



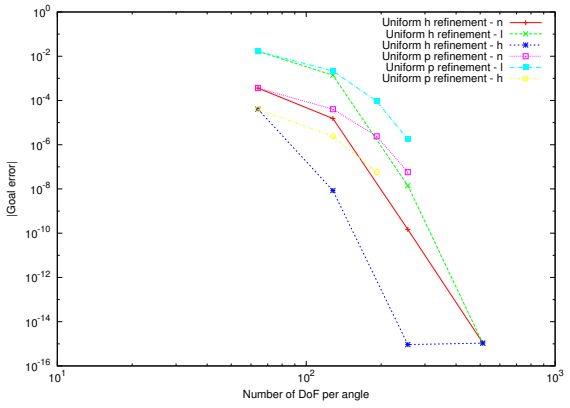
(A) $\frac{\sum_{i=1}^{nc} |\sum_{m=1}^{N} (R_{m,\mathrm{h}}, \psi^{\dagger}_{m,approx})_i|}{|truegoalerror|}$ for the first 4 meshes generated by uniform refinement from a uniform 64 cell starting mesh.



(B) $\left|\frac{\sum_{i=1}^{nc} \sum_{m=1}^{N} (R_{m,\mathrm{h}}, \psi^{\dagger}_{m,approx})_i}{truegoalerror}\right|$ for the first 4 meshes generated by uniform refinement from a uniform 64 cell starting mesh.



(C) $\left|\frac{\sum_{i=1}^{nc} \sum_{m=1}^{N} (R_{m,\mathrm{h}}, \psi^{\dagger}_{m,approx})_i}{truegoalerror}\right|$ for the first 4 meshes generated by uniform refinement from a uniform 64 cell starting mesh.



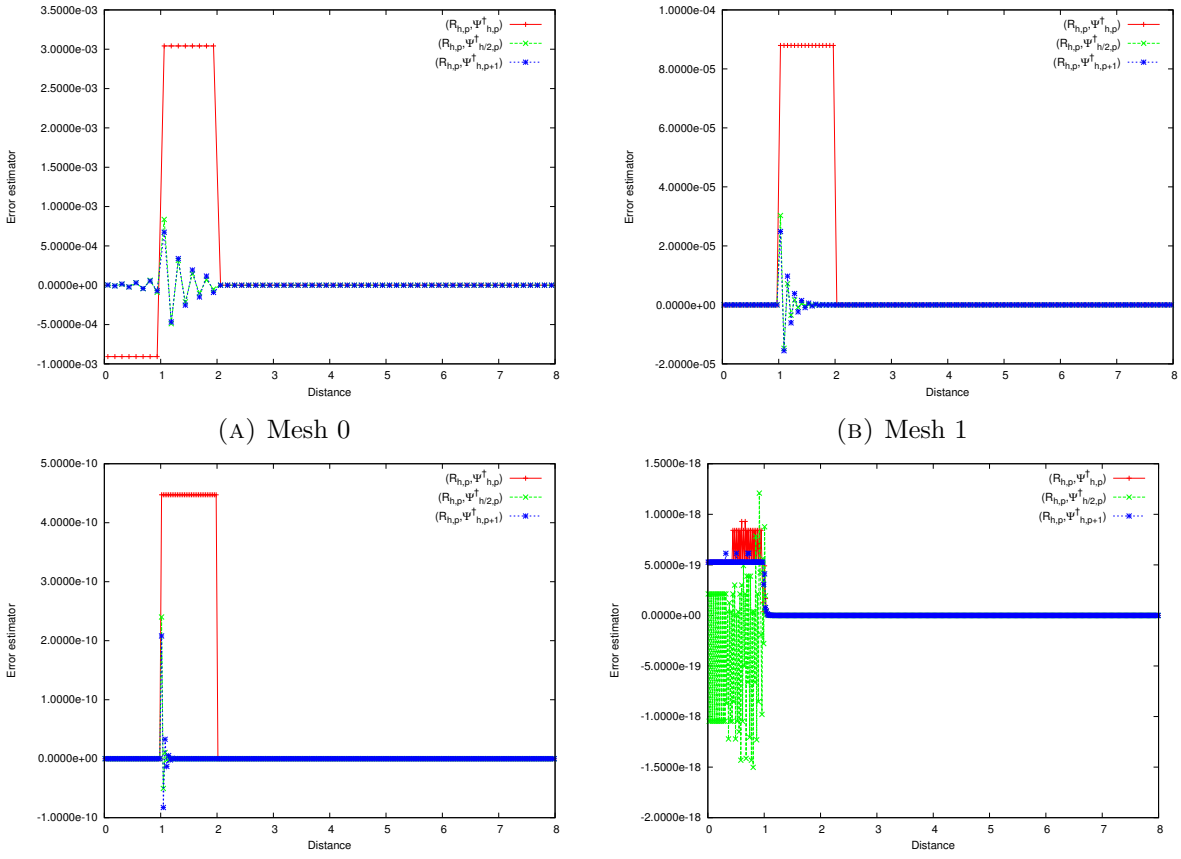
(D) Test case 3b uniform refinement and error correction.
(n: no error correction, l: low order error correction, h: high order error correction).

FIGURE 3.5: Results for test case 3b

3.5.2.5 Test case 3c

Figures 3.8a and 3.8b show that the DWR using the adjoint calculated on the same mesh as the forward equation drastically over-estimates the error for the original and first refined mesh, even when the absolute value is taken after the summation (figure 3.8b). All estimators in figure 3.8b converge to the correct estimation of the error, but the more accurate DWR errors converge faster.

Similarly to test case 3b (figure 3.5d), the DWR calculated with an adjoint computed using the same mesh as the forward solution results in an error augmentation for the initial, first h refined



(A) Mesh 0 (B) Mesh 1

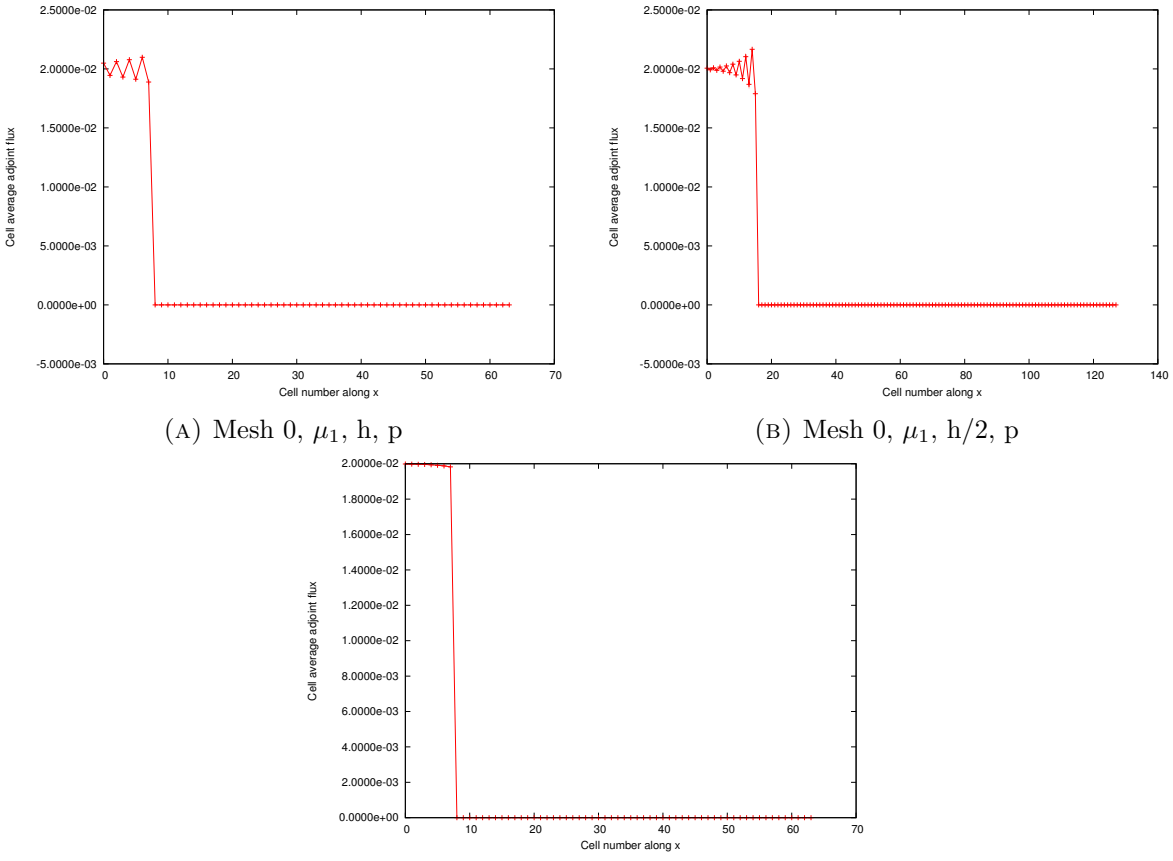
(C) Mesh 2 (D) Mesh 3

FIGURE 3.6: The spatial variation of the error estimators, $(R_{\mathrm{h}}, \psi^{\dagger}_{approx})_{cell}$, as uniform refinement is performed for test case 3b with 64 cell starting mesh

mesh and all p refined meshes. An error correction is given for the 2nd h refined mesh. The higher order DWR error estimate results in an error correction at all mesh refinement iterations, including the initial mesh.

3.5.2.6 Test case 4

When the goal is the integral of the scalar flux throughout the domain and when it is the integral throughout the first 1cm of the domain, summing the absolute values of all the error indicators for each cell over-estimates the error no matter which approximation to the adjoint solution is used (figures 3.9a and 3.10a). Figures 3.9b and 3.10b show that all error estimates converge to the analytical error if the absolute values of the local indicators are not taken before summing over all cells. The effectivity indices as plotted in figures 3.9b and 3.10b show that the best approximation to the error, for both test cases 4a and 4b, is given by the one using an adjoint solution calculated on a mesh everywhere twice as refined as the forward solution. In test



(A) Mesh 0, μ_1, h, p

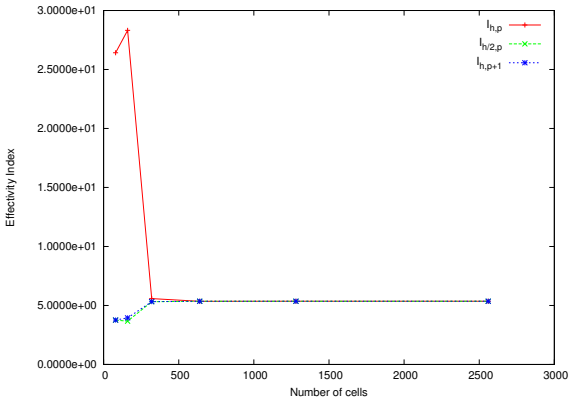
(B) Mesh 0, μ_1, h/2, p

(C) Mesh 0, μ_1, h, p+1

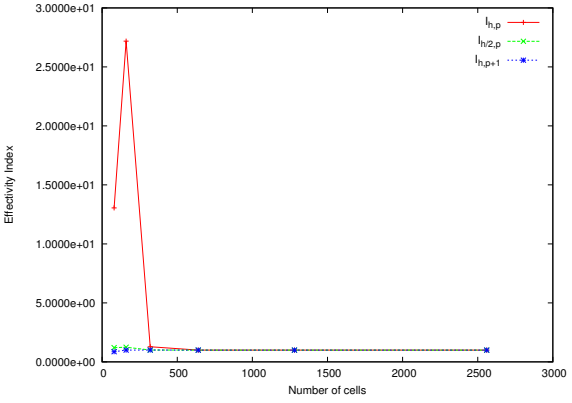
FIGURE 3.7: The spatial variation of the adjoint solution for the smallest positive $|\mu|$ for the original mesh, one twice as refined and one with an increased polynomial order

case 4b, the effectivity index resulting from using everywhere higher order cells for the adjoint solution deviates from one slightly more than the case where an adjoint solution is calculated on the same mesh as the forward equation, at low mesh cell numbers but has the advantage of not under-estimating the error.

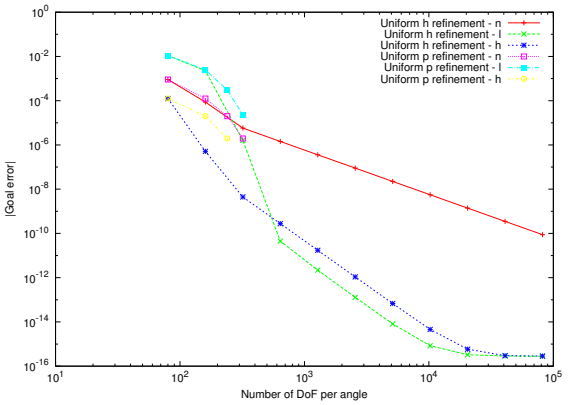
Test cases 4a (figure 3.9c) and 4b (figure 3.10c) show significant error correction of the QoI even when using the low order DWR error estimator. In test case 4a the increase in error correction obtained in using a higher order adjoint solution in the calculation of the DWR error estimator is very small and in test case 4b the error correction obtained is slightly worse due to the fact that for this test case the increased order adjoint solution gives a less accurate DWR then the low order adjoint solution. The difference is minimal, however, and is most likely due to large localised gradients not being represented much better by a higher order method.



(A) $\frac{\sum_{i=1}^{nc} |\sum_{m=1}^{N} (R_{m,\mathrm{h}}, \psi^{\dagger}_{m,approx})_i|}{|truegoalerror|}$ for the first 6 meshes generated by uniform refinement from a uniform 80 cell starting mesh.



(B) $\left| \frac{\sum_{i=1}^{nc} \sum_{m=1}^{N} (R_{m,\mathrm{h}}, \psi^{\dagger}_{m,approx})_i}{truegoalerror} \right|$ for the first 6 meshes generated by uniform refinement from a uniform 80 cell starting mesh.



(C) Test case 3c uniform refinement and error correction.
(n: no error correction, l: low order error correction, h: high order error correction).

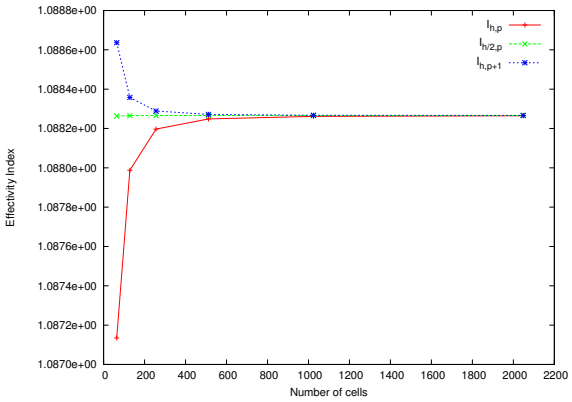
FIGURE 3.8: Results for test case 3c

3.5.2.7 Test case 5

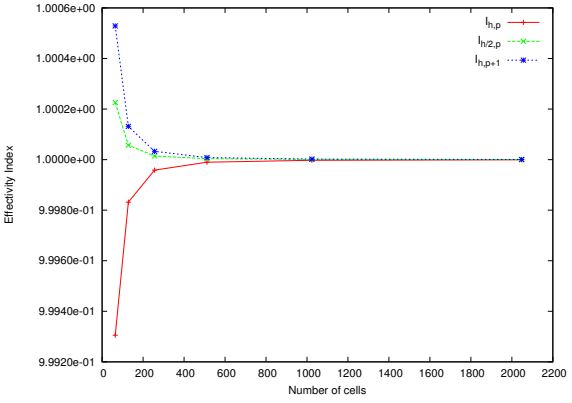
Figure 3.11a shows the accuracy of the error estimators. It is clear that for very low cell numbers the DWR error estimator using the higher-order adjoint solution is the most accurate. However, both converge quickly towards the true error as the mesh is refined. There is also a benefit to both low order and high order error correction (figure 3.11b).

3.5.2.8 Test case 6

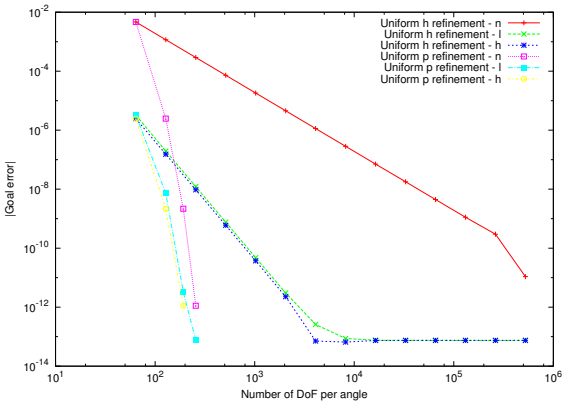
In the heterogeneous eigenvalue test case (test case 6) the error estimators do not converge to give the same error as the true error (see figure 3.12a). There is only a very small benefit to error correction (see figure 3.12b). Using a higher order adjoint gives no benefit compared to the



(A) $\frac{\sum_{i=1}^{nc} |\sum_{m=1}^{N} (R_{m,\mathrm{h}}, \psi^{\dagger}_{m,approx})_i|}{|truegoalerror|}$ for the first 6 meshes generated by uniform refinement from a uniform 64 cell starting mesh.



(B) $\left| \frac{\sum_{i=1}^{nc} \sum_{m=1}^{N} (R_{m,\mathrm{h}}, \psi^{\dagger}_{m,approx})_i}{truegoalerror} \right|$ for the first 6 meshes generated by uniform refinement from a uniform 64 cell starting mesh.



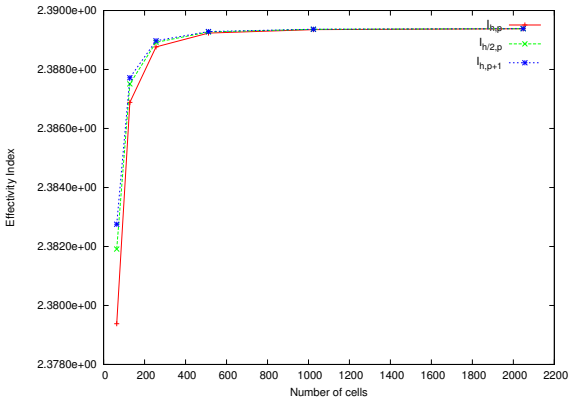
(C) Test case 4a uniform refinement and error correction. (n: no error correction, l: low order error correction, h: high order error correction).

FIGURE 3.9: Results for test case 4a

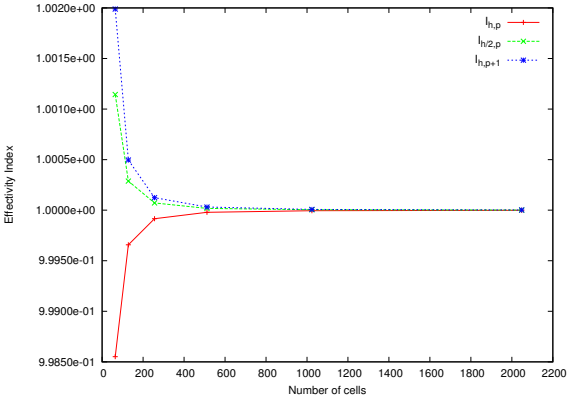
correction calculated using the lower order adjoint (see figure 3.12b). This is most likely due to the extra errors that come into play in calculating the solution to a non-linear adjoint problem as discussed in section 3.4.

3.6 Closing comments for this chapter

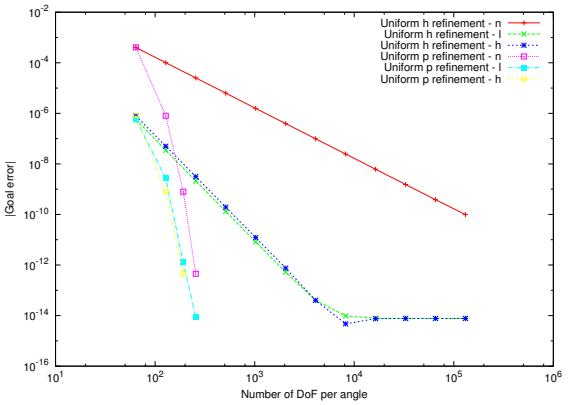
This chapter has described the development and application of DWR type error estimators for the standard and higher-order DD-S_N discretisation of the neutron transport equation in 1-D. Using the same mesh cell number and order for the adjoint solution as was used for the forward solution gave a good approximation to the true error in the QoI for most cases. The only time it was not sufficiently accurate for fixed source cases was for low order meshes in test case 3a and



(A) $\frac{\sum_{i=1}^{nc} |\sum_{m=1}^{N} (R_{m,\mathrm{h}}, \psi^{\dagger}_{m,approx})_i|}{|truegoalerror|}$ for the first 6 meshes generated by uniform refinement from a uniform 64 cell starting mesh.



(B) $\left|\frac{\sum_{i=1}^{nc} \sum_{m=1}^{N} (R_{m,\mathrm{h}}, \psi^{\dagger}_{m,approx})_i}{truegoalerror}\right|$ for the first 6 meshes generated by uniform refinement from a uniform 64 cell starting mesh.



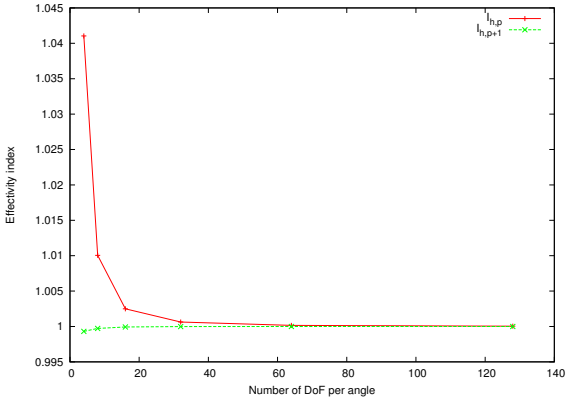
(C) Test case 4b uniform refinement and error correction. (n: no error correction, l: low order error correction, h: high order error correction).

FIGURE 3.10: Results for test case 4b

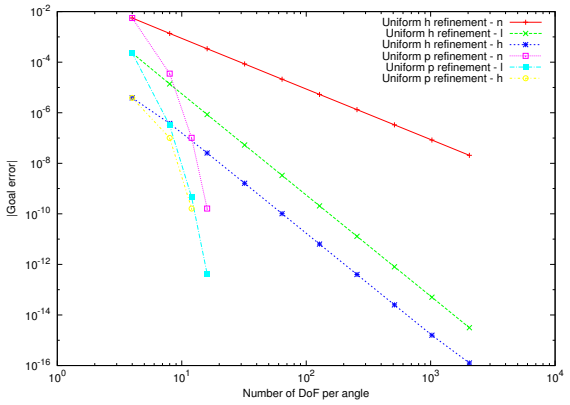
3c, and for all meshes in test case 3b. This test case, however, had regions that were not refined sufficiently to satisfy the negative flux criterion. In more challenging test cases, it may be useful to use a more accurate mesh representation for the adjoint solution if the DWR estimator is to be used as a stopping criteria. A compromise suggested here is to use the least accurate error estimator for mesh refinement until that gives a goal error below a user-defined tolerance, and then use the more accurate error estimator to verify this convergence.

Both error estimators tested in test case 6 converge to give an effectivity index that is greater than 1. This is most likely due to the linearisation error (due to the non-linear equation and goal). This is non-zero for eigenvalue problems and is not taken into account in the DWR estimator.

DWR error correction was applied to test cases in combination with uniform h and uniform p

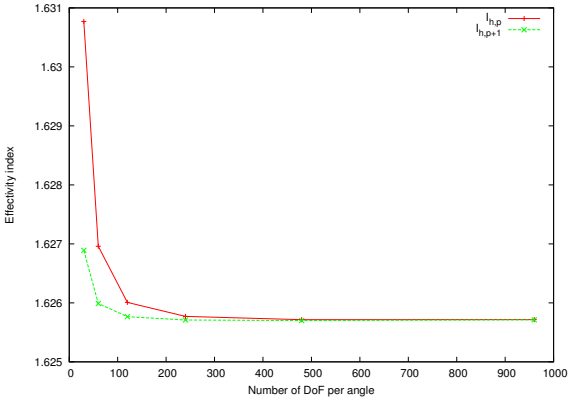


(A) $\left|\frac{\sum_{i=1}^{nc}\sum_{m=1}^{N}(R_{m,\mathrm{h}},\psi^{\dagger}_{m,approx})_i}{truegoalerror}\right|$ for the first 6 meshes generated by uniform refinement from a uniform 4 cell starting mesh.

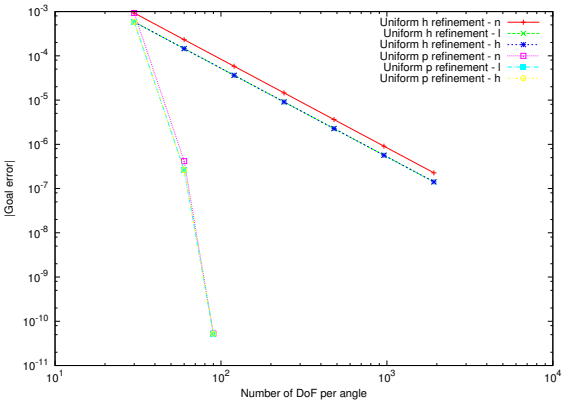


(B) Test case 5 uniform refinement and error correction. (n: no error correction, l: low order error correction, h: high order error correction).

FIGURE 3.11: Results for test case 5



(A) $\left|\frac{\sum_{i=1}^{nc}\sum_{m=1}^{N}(R_{m,\mathrm{h}},\psi^{\dagger}_{m,approx})_i}{truegoalerror}\right|$ for the first 6 meshes generated by uniform refinement from a uniform 4 cell starting mesh.



(B) Test case 6 uniform refinement and error correction. (n: no error correction, l: low order error correction, h: high order error correction).

FIGURE 3.12: Results for test case 6

refinement. A general comment can be made for the uniform h refinement fixed source results: the convergence rate with respect to cell size is increased for the corrected QoI. This is the super-convergence of DWR-corrected integral functionals as discussed in section 3.4. This is also observed for the simple eigenvalue test case, but not in test case 6, where the corrected value has the same slope as the uncorrected QoI. This is most likely due to the ϵ term being large due to the non-linearity of the problem.

Given that using Hennart's formulation for the DD technique allows the specification of higher-order solutions within a cell and that there is no requirement for cells to be of uniform size, the cell-wise calculated DWR error values could be used as error indicators for h, p or hp refinement.

Such refinement would be goal-based and reduce the number of degrees of freedom required to obtain a given accuracy in $J(\Psi_{\mathrm{h},p})$. This would be an alternative way to reduce the error for a given number of DoF. This is investigated in the following chapter.

4. DWR driven h, p and hp AMR in 1-D slab geometry

This chapter investigates the use of cell-wise DWR values as error indicators for non-uniform mesh refinement. Goal-based h, p and hp adaptive mesh refinement (AMR) for the 1-D DD-S_N discretisation of the neutron transport equation is investigated for both fixed source and eigenvalue problems. Goal-based refinement aims to produce a mesh that is optimal for a given QoI. Both linear and non-linear QoIs are investigated.

The novel parts of this chapter include the following:

- *Application of goal-based h AMR to the traditional DD method*
- *Application of goal-based p AMR to the higher order DD method*
- *Application of goal-based hp AMR to the higher order DD method using a goal-based merit function*

4.1 Introduction

The DWR error estimator described in chapter 3 is calculated by summing values calculated over a cell. The cell values can be used as refinement indicators. Cells with larger DWR values should be prioritised for refinement. With the higher order DD capability refinement can occur by either increasing the order within the cell (p AMR) or by increasing the number of cells within a given region (h AMR), or a combination of the two (hp AMR).

In general, h AMR is appropriate for cases where the solution is not smooth, whereas p AMR is more appropriate for smooth solutions [81]. However, with no *a priori* knowledge of the solution,

the most effective refinement to apply to a given problem is unknown. Another possibility is to have a problem that has both smooth and irregular regions. In that case, h adaptivity would be most appropriate in some areas of the domain but p adaptivity would be more appropriate in others. Due to these considerations it is desirable to have a combined hp AMR algorithm.

This chapter investigates the use of h, p and hp AMR. When either h or p AMR are used independently, the local DWR error indicators are sufficient for identifying which cells should be refined. However, when hp AMR is required, extra information is needed to choose whether h or p refinement is best for any given cell. In this chapter a *merit function* is derived which is to be calculated for each refinement option for a given cell that has been flagged for refinement. The refinement option with the highest merit function is implemented in the following mesh iteration.

This chapter is organised as follows. Section 4.2 explains how goal-based h or p refinement can be carried out independently using local DWR error indicators. Section 4.3 explains the hp AMR strategy implemented in this work and how a merit function is required to choose between h or p refinement for a given cell. Section 4.4 applies the theory of the first two sections to test cases. The chapter is concluded with some closing comments in section 4.5.

4.2 AMR strategy for h or p type mesh refinement

Chapter 3 showed that an approximate solution to the S_N-discretised transport equation, obtained by the DD method can be improved by either increasing the number of cells in the mesh (h refinement) or by increasing the order of the polynomial representation within the cells of the mesh (p refinement). In the DD method, the cell-size is not limited to being uniform throughout, and therefore the mesh cell size can be made such that there is a higher cell density in some regions than others. The method for higher order DD described by Hennart [56] retains the left and right nodes of a given element as unknowns and all higher order terms are cell moment values that take a value of zero at the cell nodes. This allows for two neighbouring cells to be of different order p. This allows for non-uniform p refinement as a way to increase the convergence rate for a given QoI. These properties of the DD method allow for non-uniform h, p or hp AMR such as in traditional finite element schemes [132, 133].

In chapter 3 the following error estimates were used for estimating the error in the calculation of the goal quantity in the fixed source and eigenvalue problems, where the goal quantities are

the reaction rate in a detector and the value of the K_{eff} respectively:

$$\text{Error in detector function} \approx \langle R_{\text{f.s.,h},p}, \Psi^{\dagger}_{\text{h},p} \rangle_{\bar{V}}, \tag{4.1}$$

$$\text{Error in } K_{\text{eff}} \text{ value} \approx -\langle R_{\text{eigen.,h},p}, \Psi^{\dagger}_{\text{h},p} \rangle_{\bar{V}}, \tag{4.2}$$

where:

$$R_{\text{f.s.,h},p} = WQ - \mathcal{K}\Psi_{\text{h},p}, \tag{4.3}$$

$$R_{\text{eigen.,h},p} = \lambda_h W \mathcal{F} \Psi_{\text{h},p} - \mathcal{K}\Psi_{\text{h},p}, \tag{4.4}$$

and $\Psi^{\dagger}_{\text{h},p}$ is an approximate solution to the appropriate adjoint equation. Approximations using the same mesh as the forward equation and more accurate meshes compared to the forward equation were tested in chapter 3. Adjoint approximations of increased accuracy gave better estimations of the error, in general.

Local calculations of equations (4.1) and (4.2) over a cell can be used as error indicators to flag cells for refinement. Assuming that the type of refinement is limited to one kind (h or p) decided beforehand, there are a number of options for how to use the indicator information to prioritise cells for refinement. Chapter 2 in Verfürth's book [134] gives examples of some possibilities including the *maximum strategy*. If the DWR indicator for cell i is denoted by ρ_i, and the maximum cell-indicator value in the mesh is given by $\rho_{\max}$, then the *maximum strategy* flags cell i for refinement if the following condition is true:

$$|\rho_i| \geq \alpha |\rho_{\max}|, \tag{4.5}$$

where α is a fraction input by the user. Wang notes in his master's thesis that $\alpha = \frac{1}{3}$ is a good rule of thumb [135]. This is the strategy used in this thesis, though there is a capability in the code to refine a fixed % of cells per mesh iteration. In this strategy, a pre-defined fraction of cells are refined, those being the cells with highest error. This would require ordering the values of the local estimators and refining a pre-defined fraction α. This scheme was used by Lathouwers [89, 90].

The AMR strategy which incorporates the "maximum" marking strategy is represented by the flow-chart shown in figure 4.1. This flow-chart applies to h AMR or p AMR, the difference between the two schemes being that flagged cells are refined by dividing the cell by two or by increasing the order of the polynomial within the cell by one respectively.

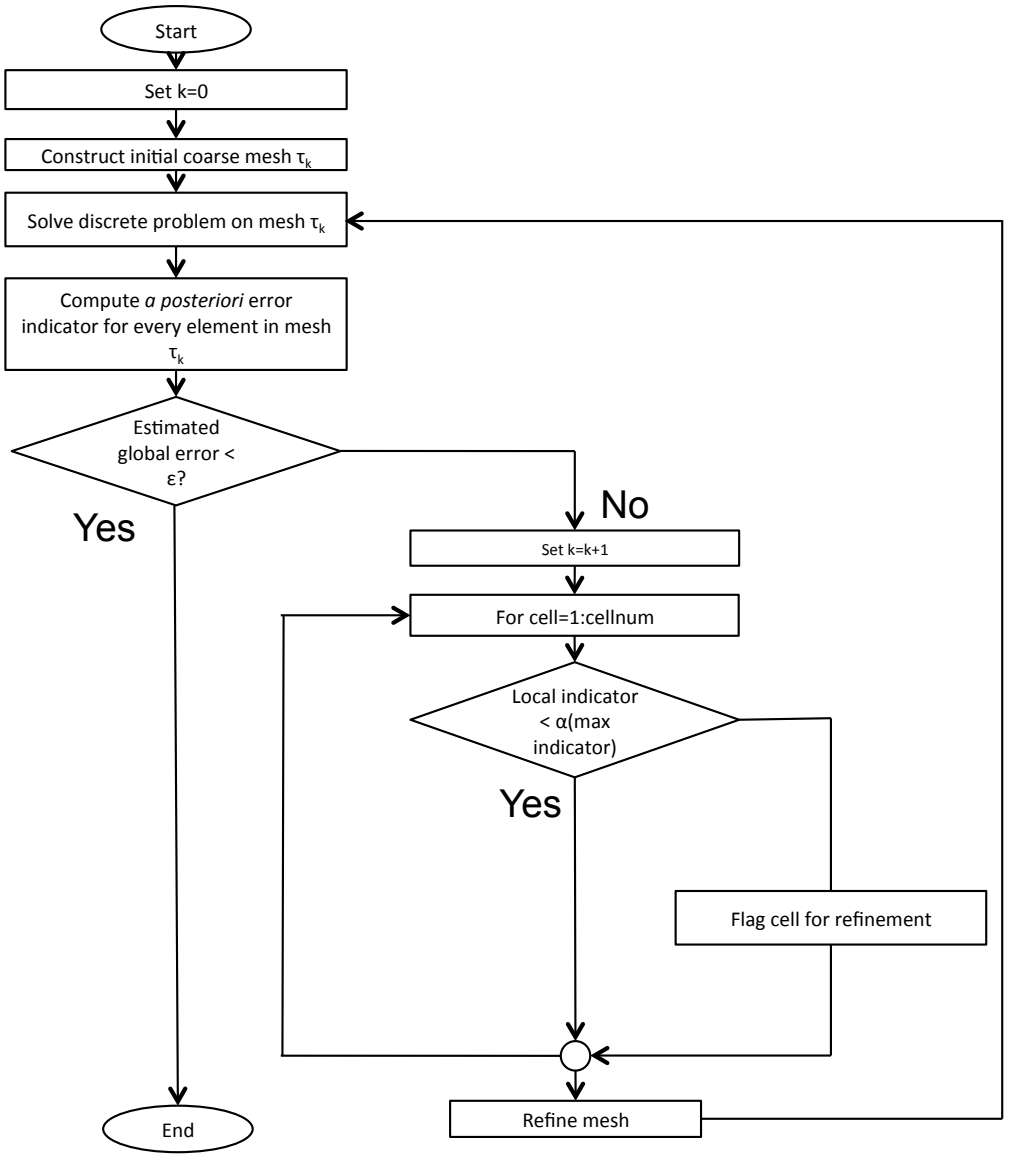


FIGURE 4.1: One-type refinement algorithm flow chart

4.3 Combined hp AMR strategy algorithm

Given that h refinement can be more beneficial in some circumstances and p refinement can be more beneficial in others, combining both types to give an hp adaptive refinement strategy should give the optimal combination of refinement strategies [81].

However, the DWR error indicator derived in chapter 3 and described in section 4.2 only identifies where is best to refine, it does not give any information as to whether it would be better to refine the cell via p or h refinement. For this, the idea of a "merit function" will be used. This terminology was used in the work of Ceze et al. [136] . They calculate a *merit* function for a

particular refinement type, m_{r}, by dividing a *benefit* function, b_{r}, by a *cost* function c_{r}:

$$m_r = \frac{b_{\mathrm{r}}}{c_{\mathrm{r}}}. \tag{4.6}$$

This is a local quantity that is calculated only for cells already flagged for refinement by the previously described DWR error indicators (section 4.2). The refinement type with the largest merit function is the refinement that is chosen for that cell in generating the next mesh.

Once a cell is identified for refinement there is an infinite number of refinement options. The cell could be split in two but have the daughter cells be of different cell sizes, or have different polynomial representations within each daughter cell. The cell could even be split into more than 2 daughter cells. In this work the options are limited to 2 refinement options: increasing the order of the polynomial representation of the solution within the cell by one without cell division (p refinement) or splitting the cell in half and keeping the order of the polynomial representation of the solution within the two daughter cells the same as that of the parent cell (h refinement).

The cost function used in this work is the same as the DoF cost function described in Ceze's work [136]. It is defined as the total number of DoF required for the cell after refinement:

$$c_{\mathrm{h}} = 2p, \tag{4.7}$$

$$c_p = p + 1. \tag{4.8}$$

Note that for a cell with a linear flux approximation, h and p refinement have the same computational cost in terms of DoF.

The benefit function calculated in this work is different from that of Ceze et al.'s work [136]. They calculate their benefit function by taking the current solution and calculating the discrete residual of the locally refined cell given by the current solution to give an idea of how incorrect the current solution is compared to the refined version. This is then weighted by the adjoint solution. The benefit function used in this work is based upon calculating the flux that would result in the cell by taking the previously calculated incoming flux and scatter source and calculating the flux distribution that would result with both different kinds of refinement: p and h refinement. The difference between the magnitude of the DWR associated with the improved and original solution is calculated over the cell for each angle. These values are then integrated numerically over the angular space.

Note that integration over angle is conducted after taking the magnitude of the DWR for both solutions and finding the difference. Even though, when calculating the DWR error indicators for identifying cells to be refined, the integration over angle occurs without taking the absolute value of the angular DWR (this follows from the derivation shown in chapter 3). In the case of the DWR error indicator, the accumulated error due to the sweep is taken into account by weighting by the adjoint solution. In the case of the benefit function, however, given that the "improved" forward solution is calculated locally, it does not take into account the changes that refinements occurring in other areas of the mesh would have on incoming fluxes and the scatter source. These errors will be greater for more downstream cells. For example, in a zero-scatter problem with vacuum boundary conditions, the approximated improved solution in the first cell on the left would be the same as the final improved solution with this refinement for any ordinates where $\mu_m > 0$. However, for $\mu_m < 0$, the incoming flux used for estimating the improved solution (namely, the incoming flux of the current solution) could be vastly different from the flux that results after refinement. Taking the magnitudes of the values for each ordinate before summing means that when the benefit functions for h and p refinement are calculated, there is no cancellations of error due to summing up benefit approximations that are more or less accurate for different angles. The benefit function for h and p refinement are given in equation (4.9) and (4.10) respectively.

$$b_{\mathrm{h}} = \sum_{m=1}^{N} w_m |(R_{h,p,m}, \psi_m^{\dagger})| - \sum_{m=1}^{N} w_m |(R_{h/2,p,m}, \psi_m^{\dagger})|_{\text{left cell}} - \sum_{m=1}^{N} w_m |(R_{h/2,p,m}, \psi_m^{\dagger})|_{\text{right cell}}, \tag{4.9}$$

$$b_p = \sum_{m=1}^{N} w_m |(R_{h,p,m}, \psi_m^{\dagger})| - \sum_{m=1}^{N} w_m |(R_{h,p+1,m}, \psi_m^{\dagger})|. \tag{4.10}$$

A current limitation of the code is that the highest order that can be represented within a cell is a 4th order polynomial. This means that if the solution within a cell is already approximated by a 4th order polynomial and it is flagged for refinement by the algorithm, a merit function is not calculated and local h adaptivity is automatically chosen for the cell.

The hp AMR method used is summarised in the flow chart of figure 4.2.

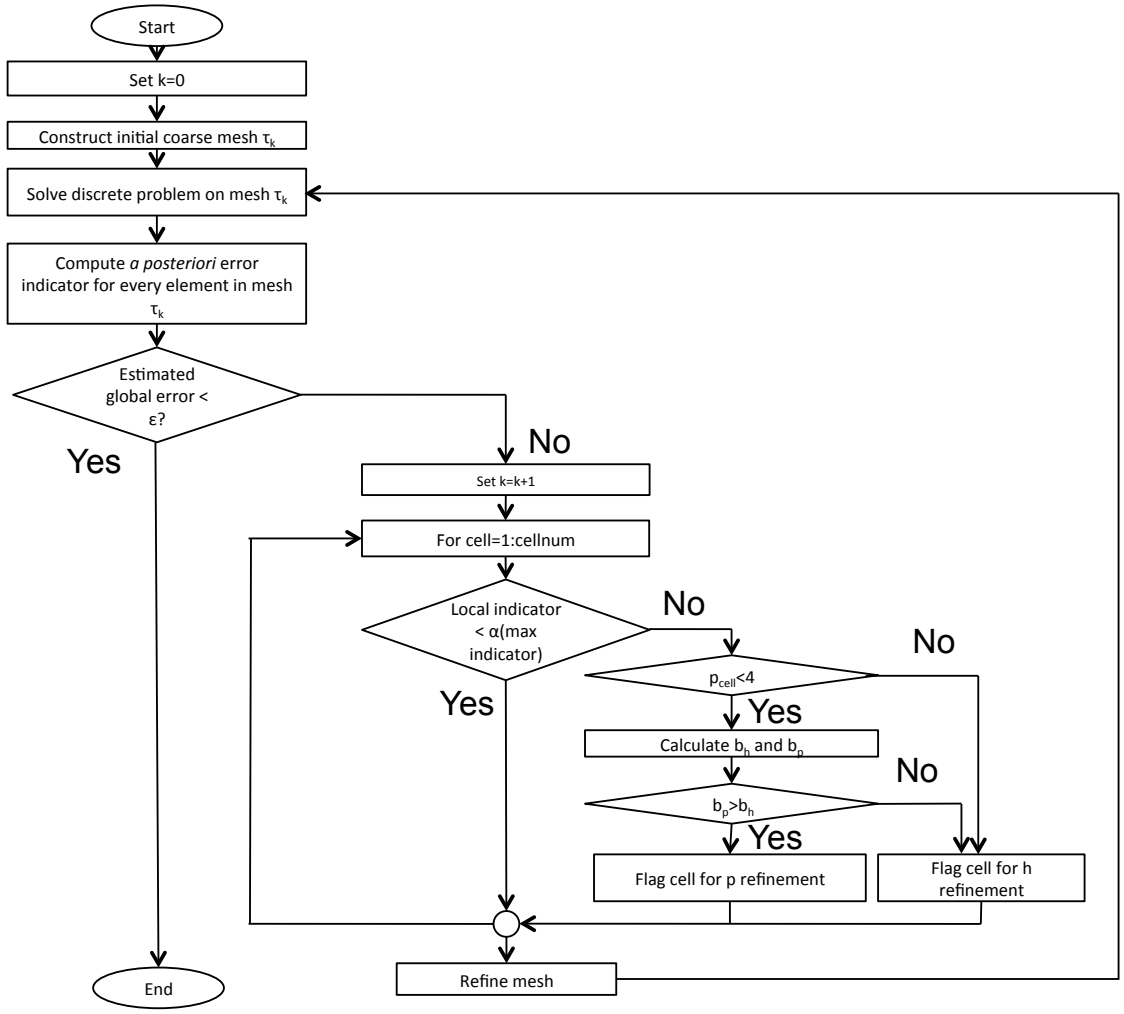


FIGURE 4.2: hp refinement algorithm flow chart

4.4 Application to test cases

This section applies the ideas described in sections 4.2 and 4.3 of this chapter to the 1D test cases described in appendix A. Section 4.4.1 applies h, p and hp adaptive strategies using the maximum marking strategy described in section 4.2. Section 4.4.2 briefly investigates the effect that different refinement strategies can have and the effect of different starting meshes on the resulting convergence graph of h AMR schemes.

4.4.1 h, p and hp AMR

Two different refinement indicators were used to flag cells for refinement in the h, p and hp AMR schemes of this section. Both are DWR error estimators, however, one uses the adjoint flux computed on the same mesh as the forward equation, whereas the other uses an adjoint

Test case	α
1	0.5
2	0.5
3	0.3
4	0.3
5	0.5
6	0.3

TABLE 4.1: α values used for each test case.

solution calculated using a mesh with the same number of cells as the forward equation but with the polynomial order that approximates the adjoint solution in each cell being one order higher than that used for the forward equation.

For the eigenvalue problem, the adjoint solution depends on the forward solution (see equation (3.92)). The same order forward equation is used in both cases (i.e. a higher order forward equation is not used in the normalisation of the higher order adjoint equation).

The values of α used for each test case are specified in table 4.1.

4.4.1.1 Test case 1 results.

The results for test case 1a are shown in figure 4.3a. The error indicators vary in such a way throughout the domain that all cells are flagged for refinement. This is due to the domain being homogeneous and the goal being the integral of the scalar flux throughout the domain. Areas of high DWR for $\mu_m > 0$ have low DWR for $\mu_m < 0$ and vice versa. Hence when h AMR is chosen with constant DD1 order cells, uniform h refinement is chosen by the adaptivity algorithm. Similarly, when p AMR is conducted, uniform p refinement is chosen by the adaptivity algorithm. Given that the solution is very smooth, it is expected and observed that p refinement gives better convergence results than h refinement. The hp refinement algorithm picks out the path of p refinement in all cases as it gives the best convergence rate.

In both cases 1a and 1b (figures 4.3a and 4.3b), uniform hp refinement results in a lower error in fewer mesh iterations with the disadvantage of a shallower convergence graph gradient compared to p refinement.

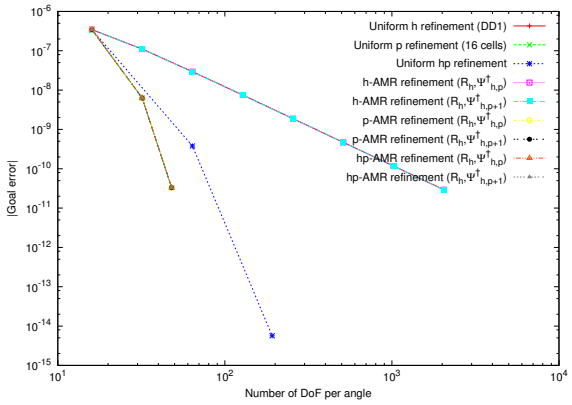
The results for test case 1b are shown in figures 4.3b-4.3d. The magnitude of the goal error as a function of the number of DoF per angular ordinate for various different refinement schemes is plotted in figure 4.3b. Uniform p refinement outperforms uniform h refinement due to the

smoothness of the forward solution. Due to the localised goal in test case 1b, gains can be made through goal-based AMR. Goal-based p AMR outperforms goal-based h AMR, and hp AMR chooses the path of p AMR up until $p = 4$, where h refinement is forced due to the limitations of the code. It can be seen from the p and hp refinement curves that the goal error increases from mesh iteration 2 to 3. This can be explained by looking at the plots of the spatial distribution in the cell-averaged scalar flux error in figures 4.3c and 4.3d. It is observed in figure 4.3c that as p AMR is conducted, the error in the first 2 cells (those in the goal-zone) are reduced significantly compared to other regions in the domain, as expected. To see the variation in this region more clearly for mesh iterations 2 and 3 the first 1cm is plotted in figure 4.3d. From figure 4.3d one observes that the absolute value of the error in each cell has reduced from mesh iteration 2 to mesh iteration 3. However, the two cells have errors of opposite sign in mesh 2 but not in mesh 3. For this reason the integral value of the error of the scalar flux in the first 1cm results in error cancellation in mesh iteration 2 that is not present in mesh iteration 3. Hence the kink observed in figure 4.3b for p and hp AMR is explained.

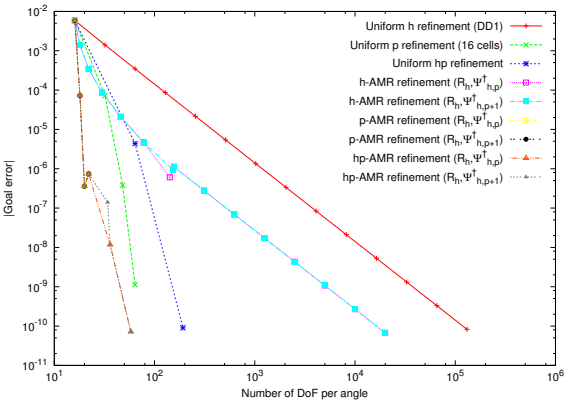
A final comment to be made is that there is no difference in the AMR convergence graphs when the higher order adjoint is used in the calculation of the DWR error indicators compared to the adjoint computed on the same mesh as the forward equation. There are a couple of exceptions to this rule. One example is for hp AMR in test case 1b, where the higher order scheme results in an extra mesh iteration before following the same convergence graph as the lower order scheme. The second example is for h AMR in test case 1b where the 6th mesh iteration gives slightly different errors when using adjoint solutions of different accuracy.

4.4.1.2 Test case 2 results

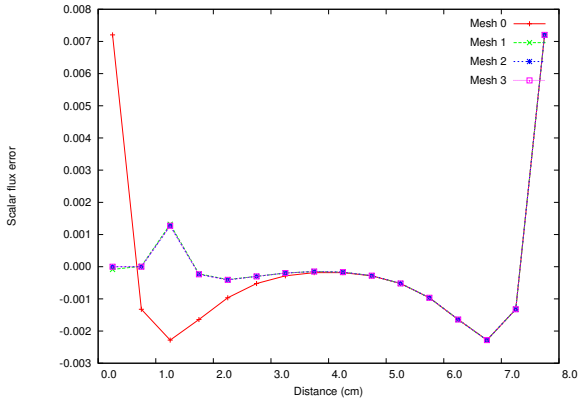
The convergence graphs for test cases 2a and 2b are shown in figures 4.4a and 4.4b respectively. The results for test case 2 are very similar to those for test case 1 (section 4.4.1.1). The smoothness of the forward solution, in both cases, results in p refinement being superior to h refinement. No gains are obtained for adaptive h, p or hp AMR in test case 2a due to a uniform mesh and polynomial order being optimal for this problem. The localised goal in test case 2b results in gains being made with the use of h, p or hp AMR. The largest gains being made by p AMR. The hp AMR graph follows the p AMR graph up to mesh iteration 3 where the p AMR algorithm is stopped due to the limit of p=4 in the current code.



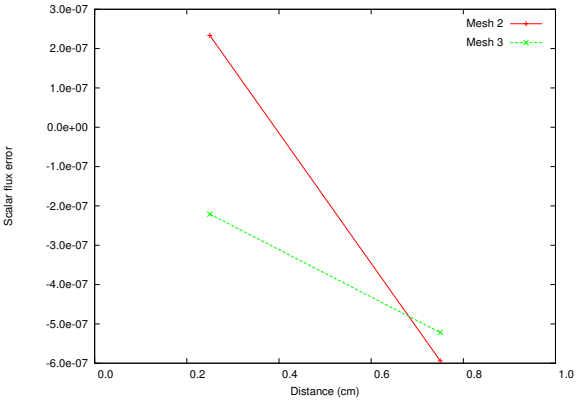
(A) Test case 1a convergence.



(B) Test case 1b convergence.

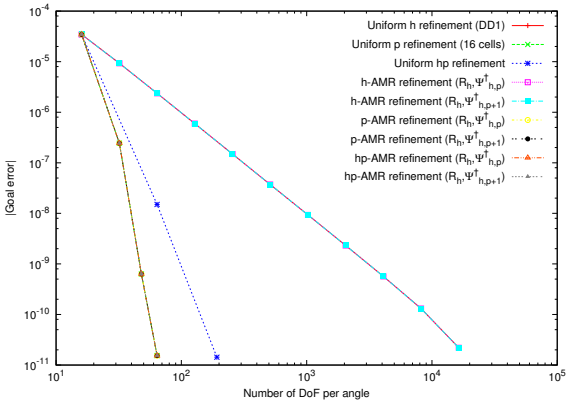


(C) Spatial variation of error in scalar flux (whole domain) for test case 1b.

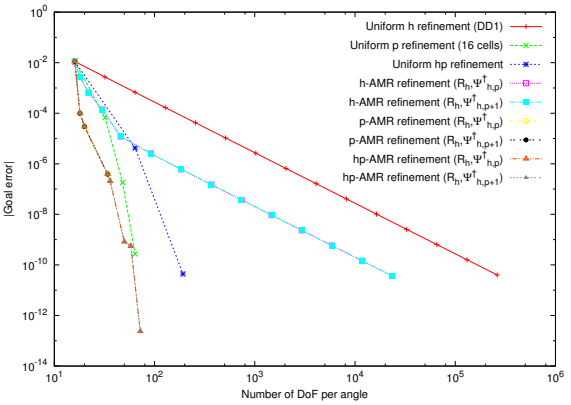


(D) Spatial variation of error in scalar flux (first 1cm) for test case 1b.

FIGURE 4.3: Results for test case 1



(A) Test case 2a convergence.



(B) Test case 2b convergence.

FIGURE 4.4: Results for test case 2

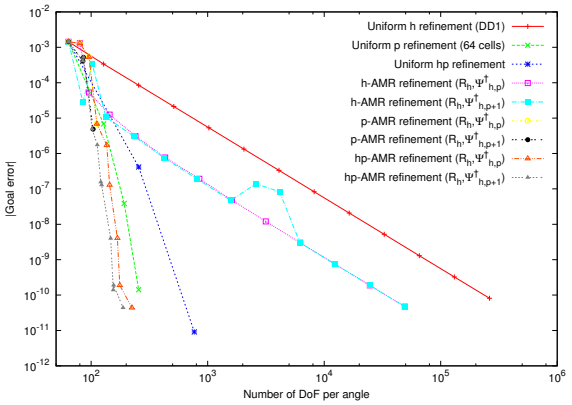
4.4.1.3 Test case 3 results

This test case is more complicated than previous test cases and the initial mesh does not satisfy the negative flux criterion (that guarantees non-negative fluxes) throughout the mesh [32].

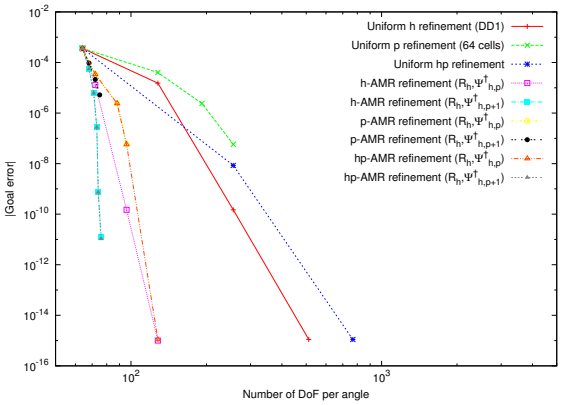
In test case 3a, where the goal quantity is the integral of the scalar flux throughout the domain, the region of high total cross section is not enough to result in p refinement being less effective than h refinement (figure 4.5a). Due to the heterogeneity of the problem, gains can be made via h, p or hp AMR compared to uniform refinement. In the case of p AMR, using an improved adjoint solution significantly affects the number of DoF that results. When hp AMR is conducted, a combination of h and p refinement takes place between mesh iteration 2 and 3 when the hp refinement algorithm is used in combination with the higher order DWR, since the number of DoF required when p refinement is used is very slightly more than in the equivalent hp AMR case. This is the first example where hp AMR outperforms either h or p AMR, rather than just picking the most favourable path out of h AMR or p AMR.

Test case 3b has the goal quantity in the first 1cm of the domain, where the mesh is significantly under-resolved and the non-negative flux criterion is not adhered to. In this case, uniform h refinement gives a better convergence plot than uniform p refinement, as shown in figure 4.5b. Due to the heterogeneity of the problem and the localised goal quantity, significant gains can be made though h, p or hp AMR. Figure 4.5b shows that for this more complex problem, the use of a higher order adjoint solution in the computation of the DWR error indicator results in a better convergence graph when driving h or p AMR (compare the pink and light blue lines for h refinement and yellow and black lines for p refinement in figure 4.5b). In both cases, adaptive h refinement is more accurate than adaptive p refinement using the same error indicator. In hp AMR, when the low order adjoint is used in calculating the error indicators for flagging cells for refinement and in the calculation of the merit function (red line with triangle pointers), the hp algorithm does not initially choose the most efficient refinement. p adaptation is chosen up until the 4th mesh iteration, when the 4th refinement results in h AMR that gives a very similar error to the 3rd mesh refinement of the h AMR scheme using the equivalent error indicator. When a high order DWR indicator is used and the high order adjoint is used in the calculation of the merit functions of the hp algorithm, the hp algorithm chooses the path of h refinement over p refinement, thereby choosing the best option for this test case.

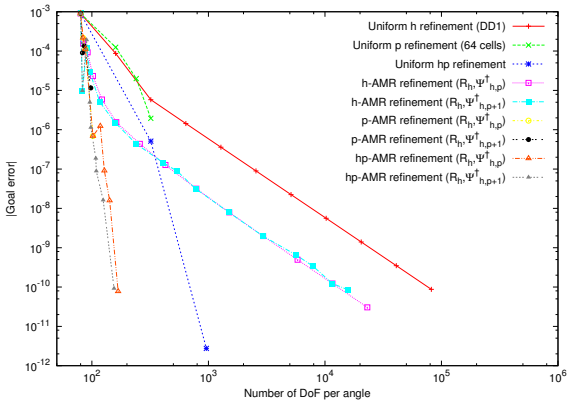
The results for test case 3c are given in figure 4.5c. Uniform h refinement out-performs uniform p refinement due to the goal region being placed where the flux varies sharply. p AMR and h



(A) Test case 3a.



(B) Test case 3b.



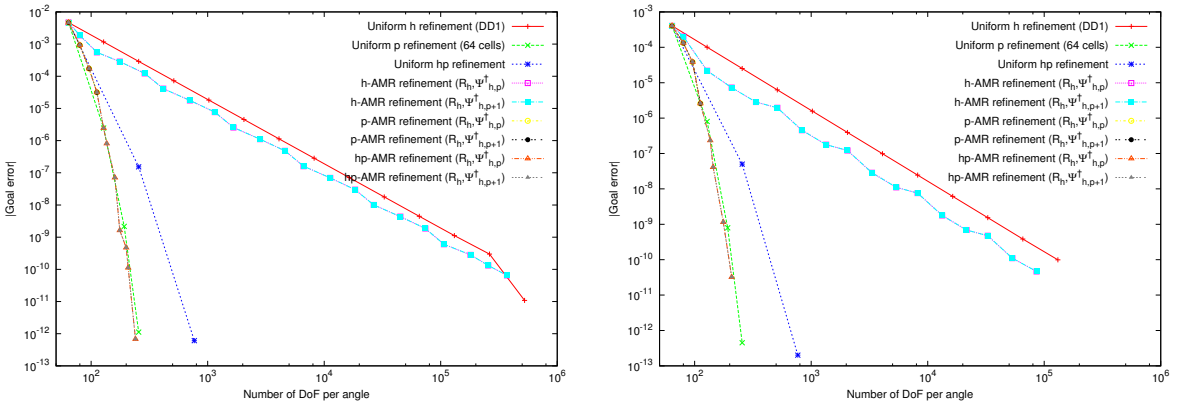
(C) Test case 3c.

FIGURE 4.5: Results for test case 3

AMR give similar slopes in the pre-asymptotic region. hp AMR performs best and the higher order DWR gives a slightly improved convergence graph.

4.4.1.4 Test case 4 results

The convergence results for both test case 4a and 4b are given in figure 4.6. Due to the smoothness of the forward solution, uniform p refinement gives a better convergence graph than uniform h refinement in both test case 4a (figure 4.6a) and 4b (figure 4.6b). Due to the heterogeneity of the problem, gains can be made by using h, p or hp AMR even when the goal is the integral of the scalar flux throughout the whole domain. However, the gains made by h AMR compared to uniform h refinement are more in test case 4b (figure 4.6b) compared to 4a (figure 4.6a). For p AMR, the first 2 mesh iterations are above the uniform hp refinement line in the case of the localised-goal problem (figure 4.6b), whereas they are below the uniform hp refinement line in the case where the goal is the integral of the scalar flux throughout the domain.



(A) Test case 4a. (B) Test case 4b.

FIGURE 4.6: Results for test case 4

There are not enough p AMR points for the p AMR convergence line to cross the uniform p refinement line in test case 4a (figure 4.6a). The hp AMR in this test case follows the p AMR for the first 3 mesh iterations and coincides with the uniform p refinement graph for the fourth iteration, it roughly follows the uniform p refinement line after that, often resulting in a smaller error per number of DoF. It is likely that allowing for higher order polynomials in the cell would benefit the hp algorithm for test case 4a.

For test case 4b, the p AMR line coincides with the uniform p refinement line on the 3rd mesh iteration. The hp AMR line follows that of the p AMR line and continues beyond the 3rd mesh iteration in such a way that it gives consistently lower error than the uniform p refinement line thereafter.

In this test case, using a higher order adjoint approximation for the DWR error indicators and for the merit functions does not give an improved AMR convergence graph.

4.4.1.5 Test case 5 results

Figure 4.7 shows the convergence behaviour of the error in K_{eff} for various refinement schemes applied to test case 5. The fact that the geometry is homogeneous, means that a uniform mesh size and local polynomial order are optimal for this test case, hence all non-uniform refinement options follow uniform refinement (uniform h refinement in the case of h AMR and uniform p refinement in the case of p and hp AMR). The smoothness of the problem means that uniform p refinement out-performs uniform h refinement.

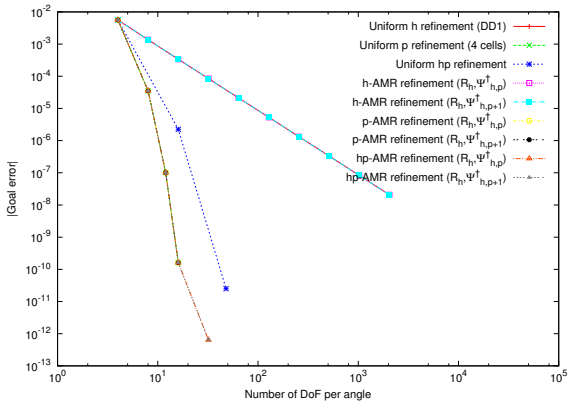


FIGURE 4.7: Test case 5 analytical goal error convergence results

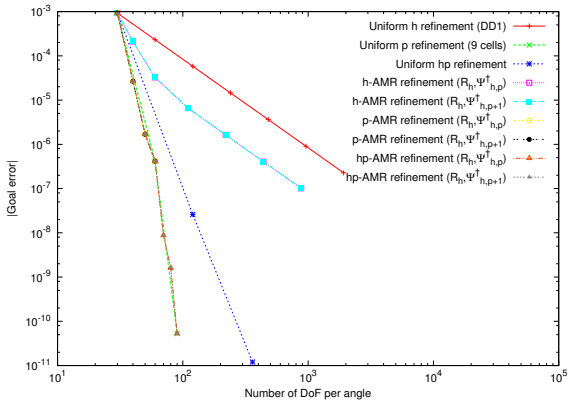


FIGURE 4.8: Test case 6 analytical goal error convergence results

4.4.1.6 Test case 6 results

Figure 4.8 shows the convergence behaviour of the error in K_{eff} for various refinement schemes applied to test case 6. Due to the smoothness of the solution, p refinement outperforms h refinement. Gains can be made using non-uniform refinement due to the heterogeneity of the problem. It is clear from figure 4.8 that using a higher order adjoint for calculating the DWR error indicator and for calculating the merit function does not have an effect on the mesh refinement chosen in this test case. However, it is clear that the gains made by using h AMR compared to uniform h refinement is significantly larger than the reduction in error per DoF for p AMR compared to uniform p refinement. The p AMR strategy gives a convergence line that is very close (sometimes coinciding with) the uniform p refinement line. The hp AMR line follows that of the p refinement until the 3rd mesh iteration (where the p AMR line is stopped due to p=4 being a limit in the code).

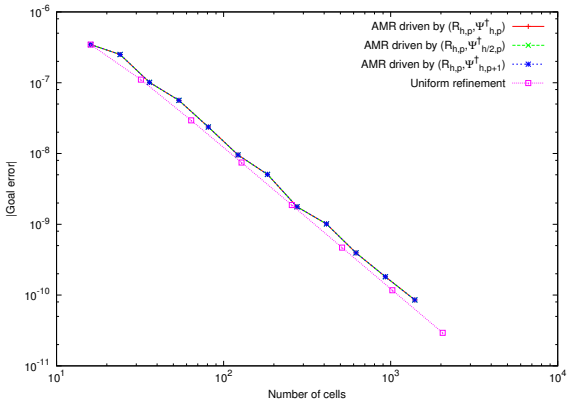


FIGURE 4.9: The absolute value of the goal error as a function of cell number. AMR refinement criterion = refine 50% of the cells at each mesh iteration (those with the highest error indicator values).

4.4.2 Starting meshes and refinement strategies

Section 4.4.1 used the maximum marking strategy defined in equation (4.5) for all test cases with a pre-defined value of α that was found appropriate for the given test case. In this section the influence of the refinement strategy and starting mesh on some of the 1-D test cases is analysed.

It was mentioned in section 4.2 that the 1-D code has the capability of implementing a fixed % strategy, where all cells are ordered in terms of decreasing error indicator magnitude and the top ($\alpha \times 100\%$) cells are refined. This forces non-uniform refinement in cases where a uniform mesh may be the optimal mesh for a given problem. For example, it was shown in figure 4.3a that a uniform mesh is optimal for test case 1a. If it is decided to refine half of all cells (the half with the highest values of error indicators) at each mesh iteration, the convergence behaviour plotted in figure 4.9 is obtained. The accuracy of the adjoint solution does not have any effect on the convergence behaviour of the AMR, however it is noted that the convergence rate is approximately the same as in uniform refinement, but that there are intermediate values that appear due to forcing only a % of cells to be refined. This is not a very efficient way of conducting AMR compared to the maximum marking strategy implemented in section 4.4.1.

Even if the maximum marking strategy applied in section 4.4.1 is chosen, convergence results can be different depending on the choice of α. If $\alpha = 0$ then uniform refinement is conducted no matter what the shape of the error indicator distribution. At the other extreme, if $\alpha = 1$, only one cell is refined at each mesh iteration, and a new calculation would need to be calculated each time. This would be very inefficient. Wang notes in his Master's thesis that $\alpha = 0.3$ is a good rule of thumb [135]. This seems to be true for test case 3a. The results of uniform h refinement

and h AMR are plotted in figure 4.10 for when α takes two different values: $\alpha = 0.3$ (figures 4.10b, 4.10d and 4.10f) and $\alpha = 0.8$ (figures 4.10a, 4.10c and 4.10e). The results are plotted for 3 different starting meshes: a 64 cell starting mesh (figures 4.10a and 4.10b), a 320 cell starting mesh (figures 4.10c and 4.10d) and a 1120 cell starting mesh (figures 4.10e and 4.10f). For all starting meshes in figure 4.10, $\alpha = 0.3$ gives a more stable convergence graph than when $\alpha = 0.8$. This result supports Wang's "rule of thumb" [135]. For this particular test case, where it has been previously discussed that coarse meshes give a bad representation of the error, the initial mesh can have some affect on the convergence graph. For example, the beginning of the graph in figure 4.10b has more oscillations than those with a finer original mesh (figures 4.10d and 4.10f), even when $\alpha = 0.3$.

For test case 3b, it was found that the goal error was already converged at a uniform mesh of 320 cells, and so it is only the results for the 64 cell starting mesh that are shown in figure 4.11. One can however, observe some more oscillation in the convergence graph of the h AMR lines when $\alpha = 0.8$ is used instead of $\alpha = 0.3$.

The same comments can be made for test case 3c on figure 4.12 as were made for test case 3a on figure 4.10.

Test case 4a (figure 4.13) and test case 4b (figure 4.14) do not show any difference in convergence behaviour between starting meshes, for the same value of α. This is most likely because the negative flux criterion is satisfied throughout the domain, even when the initial mesh has only 64 cells. However, as in test case 3, there is more oscillation in the convergence graph when $\alpha = 0.8$ compared to when $\alpha = 0.3$.

4.5 Closing comments for this chapter

This chapter has used DWR error indicators to drive AMR for the 1-D DD-S_N method applied to both fixed source and eigenvalue (K_{eff} criticality) problems.

This work has demonstrated that a reduction in the number of unknowns required to reach an accurate solution to a QoI defined by a functional of the angular flux can be achieved in some cases with the use of DWR driven AMR.

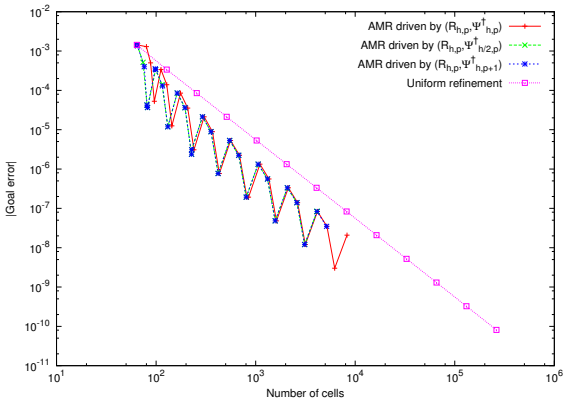
It was found that using a higher-order approximation to the adjoint solution did not result in a better mesh refinement for most of the test cases analysed here. The exception was where the

goal was in a region heavily affected by the instability of the DD-S_N solution in areas of large total cross section.

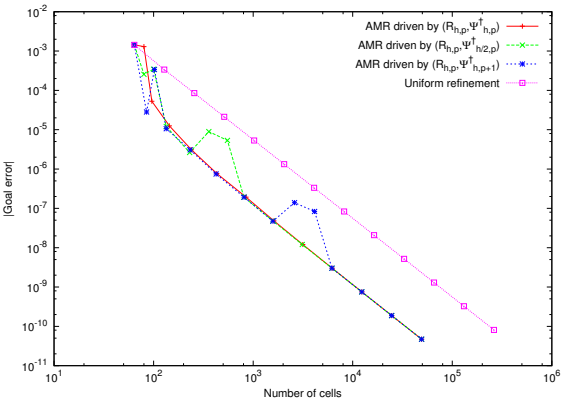
Some sensitivity to the parameter α was observed. The convergence graph also depends on the marking scheme chosen, for example, whether refinement is carried out by refining cells with an error above a fraction α of the maximum local indicator, or by setting a fixed fraction α of cells for refinement. Sometimes there is sensitivity to the initial mesh chosen, especially if the cell sizes do not satisfy the negative flux criterion.

One can conclude that hp AMR is superior to either h AMR or p AMR when there is no *a priori* information about the shape of the solution. In challenging problems such as the Reed cell benchmark (test case 3 in this work) it is best to use a higher order adjoint in the calculations of the error indicator and benefit function.

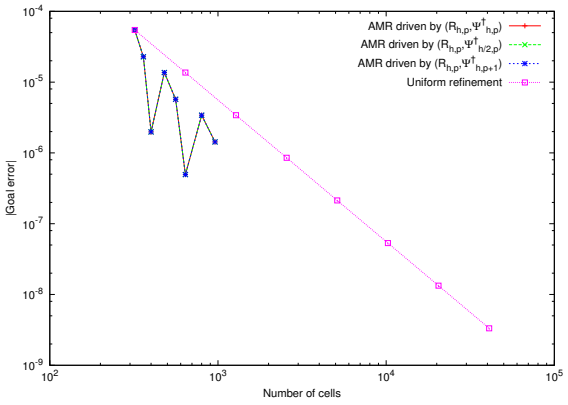
Combining the benefits of DWR functional correction (applied to test cases in chapter 3) and the goal-based AMR in this chapter could give further reductions in the number of unknowns required for a given accuracy of an approximation to a given QoI. This is investigated in the following chapter.



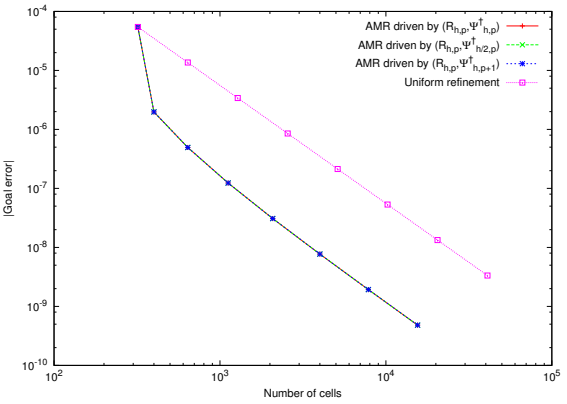
(A) Convergence with uniform starting mesh of 64 cells. Refinement criterion = $0.8(max_cell_error)$



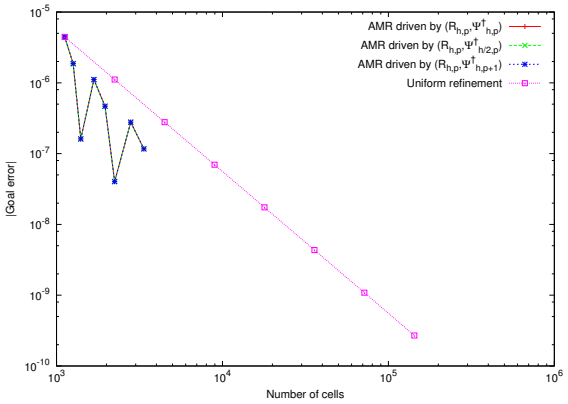
(B) Convergence with uniform starting mesh of 64 cells. Refinement criterion = $0.3(max_cell_error)$



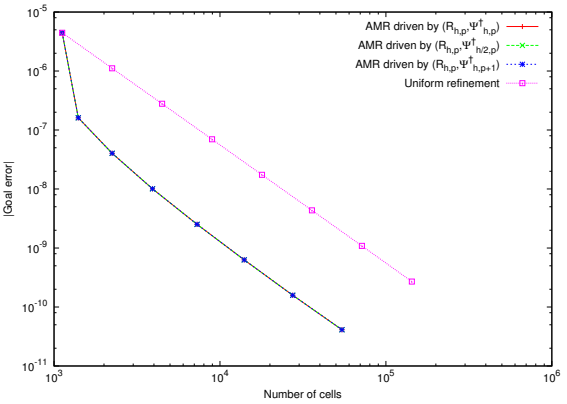
(C) Convergence with uniform starting mesh of 320 cells. Refinement criterion = $0.8(max_cell_error)$



(D) Convergence with uniform starting mesh of 320 cells. Refinement criterion = $0.3(max_cell_error)$

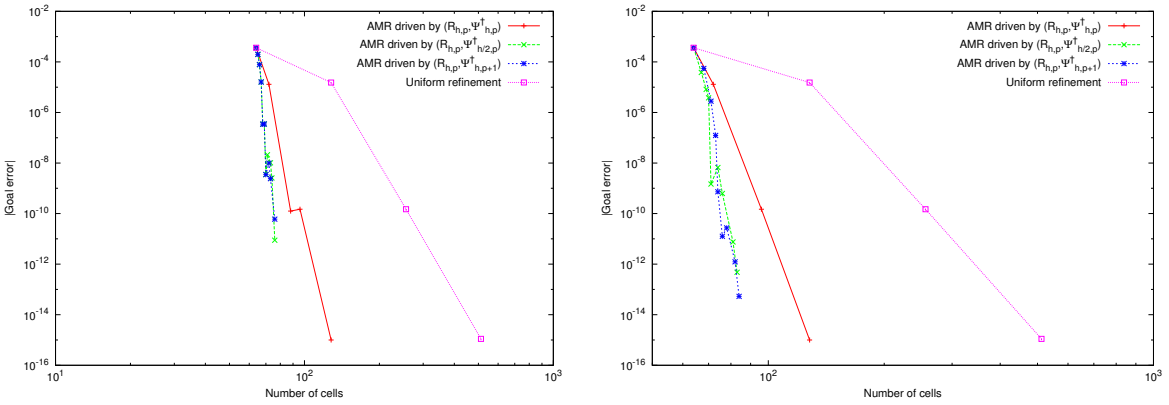


(E) Convergence with uniform starting mesh of 1120 cells. Refinement criterion = $0.8(max_cell_error)$



(F) Convergence with uniform starting mesh of 1120 cells. Refinement criterion = $0.3(max_cell_error)$

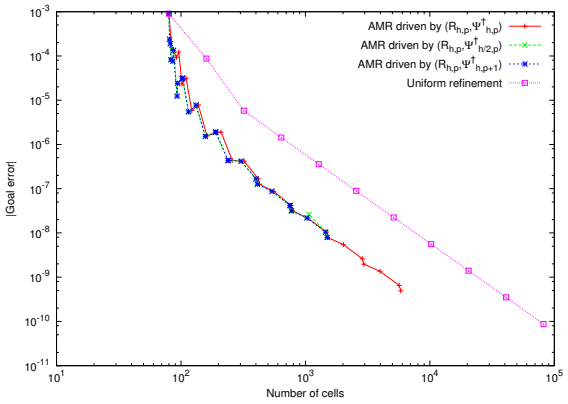
FIGURE 4.10: Convergence results for test case 3a



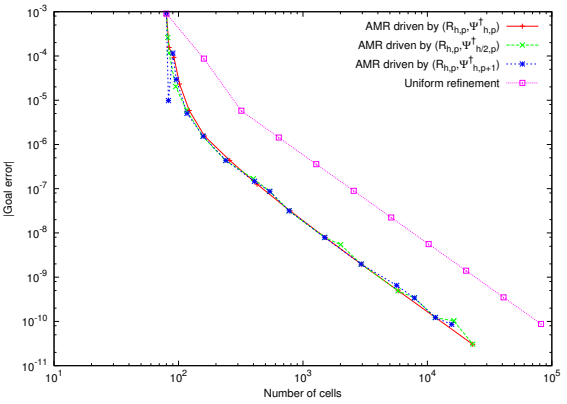
(A) Convergence with uniform starting mesh of 64 cells. Refinement criterion = $0.8(max_cell_error)$

(B) Convergence with uniform starting mesh of 64 cells. Refinement criterion = $0.3(max_cell_error)$

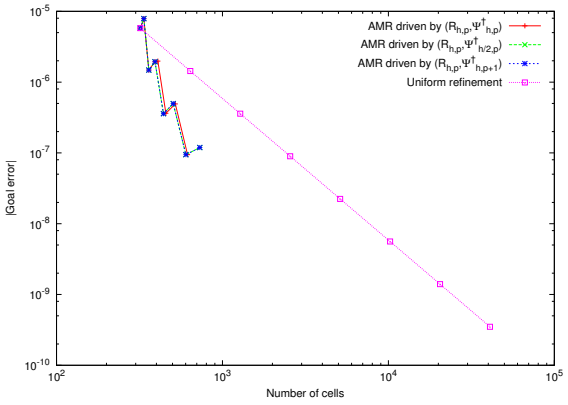
FIGURE 4.11: Results for test case 3b



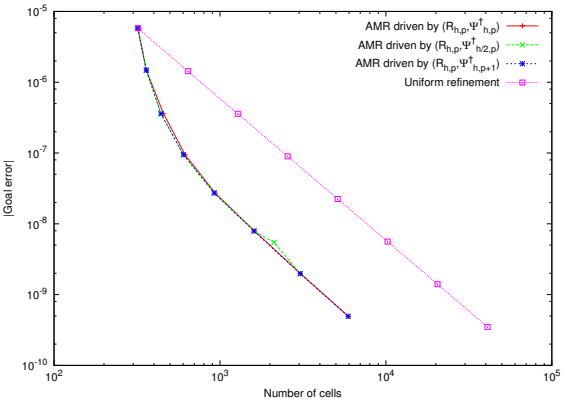
(A) Convergence with uniform starting mesh of 80 cells. Refinement criterion = $0.8(max_cell_error)$



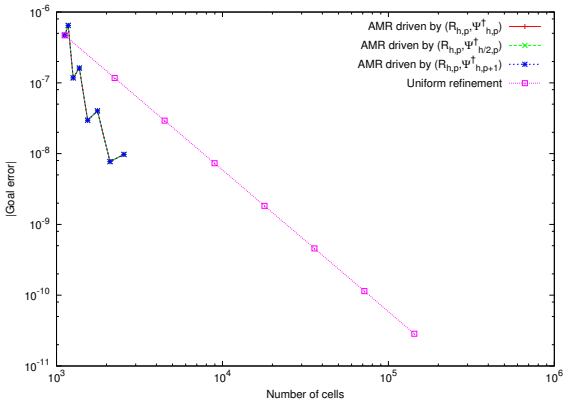
(B) Convergence with uniform starting mesh of 80 cells. Refinement criterion = $0.3(max_cell_error)$



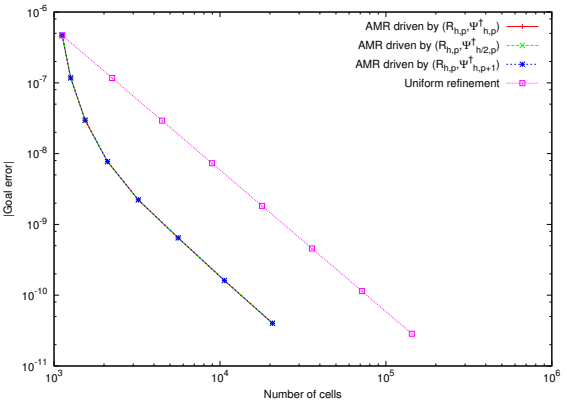
(C) Convergence with uniform starting mesh of 320 cells. Refinement criterion = $0.8(max_cell_error)$



(D) Convergence with uniform starting mesh of 320 cells. Refinement criterion = $0.3(max_cell_error)$

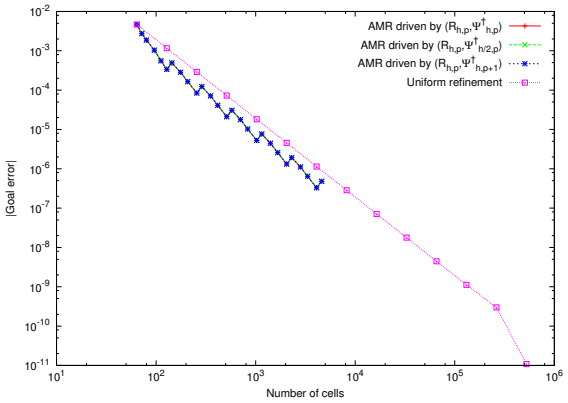


(E) Convergence with uniform starting mesh of 1120 cells. Refinement criterion = $0.8(max_cell_error)$

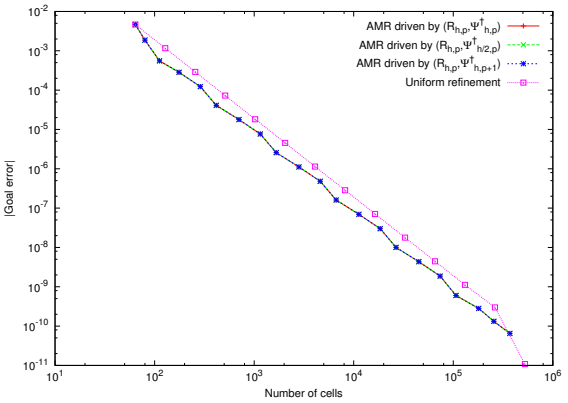


(F) Convergence with uniform starting mesh of 1120 cells. Refinement criterion = $0.3(max_cell_error)$

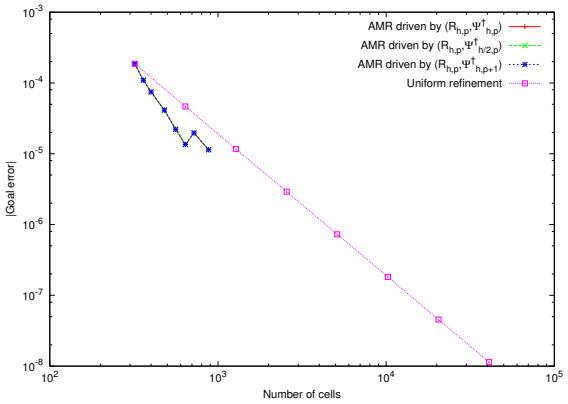
FIGURE 4.12: Convergence results for test case 3c



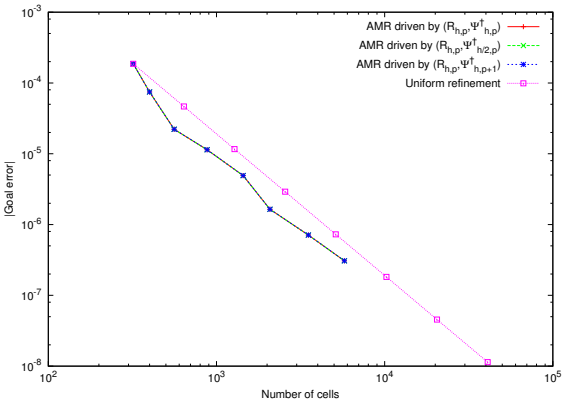
(A) Convergence with uniform starting mesh of 64 cells. Refinement criterion = $0.8(max_cell_error)$



(B) Convergence with uniform starting mesh of 64 cells. Refinement criterion = $0.3(max_cell_error)$

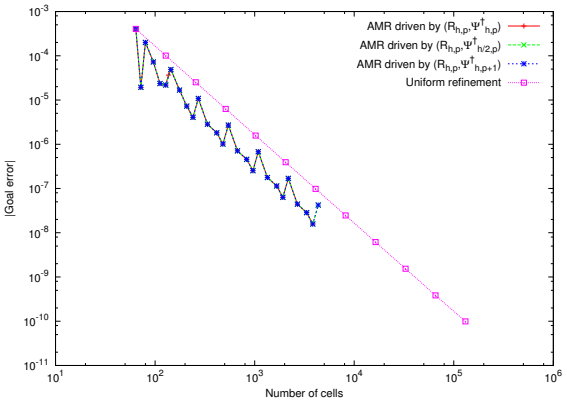


(C) Convergence with uniform starting mesh of 320 cells. Refinement criterion = $0.8(max_cell_error)$

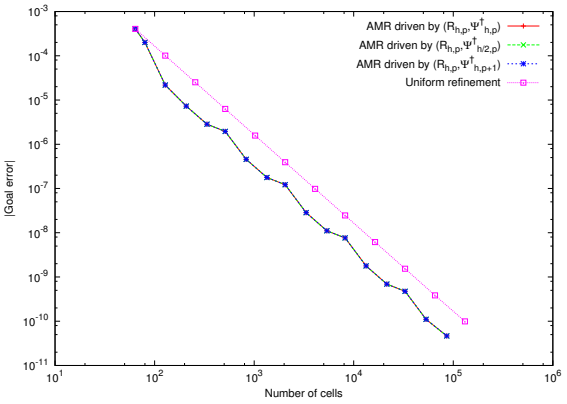


(D) Convergence with uniform starting mesh of 320 cells. Refinement criterion = $0.3(max_cell_error)$

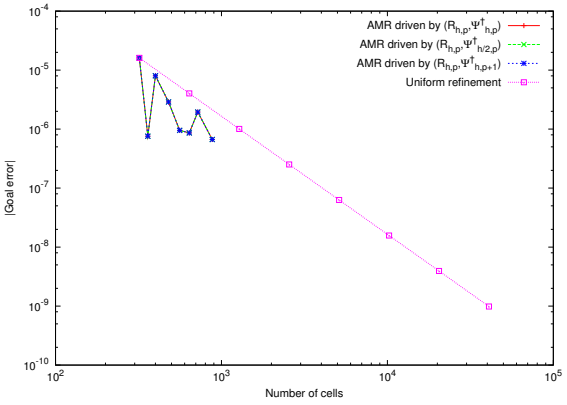
FIGURE 4.13: Convergence results for test case 4a



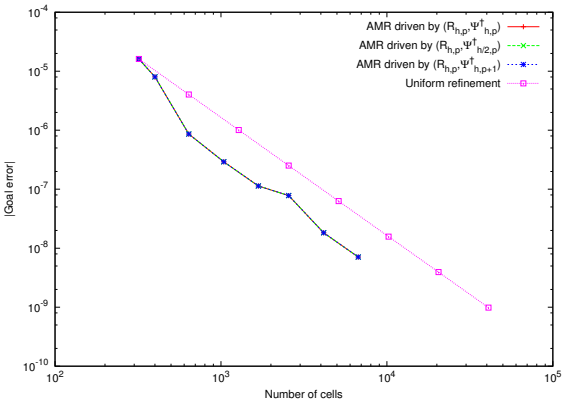
(A) Convergence with uniform starting mesh of 64 cells. Refinement criterion = $0.8(max_cell_error)$



(B) Convergence with uniform starting mesh of 64 cells. Refinement criterion = $0.3(max_cell_error)$



(C) Convergence with uniform starting mesh of 320 cells. Refinement criterion = $0.8(max_cell_error)$



(D) Convergence with uniform starting mesh of 320 cells. Refinement criterion = $0.3(max_cell_error)$

FIGURE 4.14: Convergence results for test case 4b

5. AMR and functional correction for linear functionals in 1-D slab geometry

This chapter investigates combining DWR functional error correction with goal-based h, p and hp AMR for the 1-D DD-S_N discretisation of the neutron transport equation. A new error indicator is used, based on work previously carried out in the CFD industry [105, 108]. The new indicator is an approximation to the error in the error correction rather than the error in the approximation to the functional given by the discretised flux. This indicator is deemed more appropriate since an optimal mesh for the corrected functional value is sought.

The novel parts of this chapter include the following:

- *The combination of goal-based AMR and functional error correction in the neutron transport field*
- *Application of goal-based h AMR and error correction to the traditional DD method*
- *Application of goal-based p AMR and error correction to the higher order DD method*
- *Application of goal-based hp AMR and error correction to the higher order DD method using a goal-based merit function*

5.1 Introduction

In chapter 3, the error in a QoI for a given number of DoF was reduced by adding a DWR error correction term. Chapter 4 used DWR error indicators to drive goal-based AMR so that

computational resources were only being used where needed for a given problem. This also reduced the error in a given QoI as a function of number of DoF for a number of test cases. It is therefore reasonable to assume that combining goal-based AMR with DWR functional correction will reduce the error in the QoI per number of DoF even further.

Müller and Giles's work [103] uses the leading order error term (the *computable correction* term) for local indicators instead of functional correction, in the same way as applied to the 1-D DD-S_N discretisation of the 1-D transport equation in chapter 4 of this manuscript. Venditti and Darmofal [105], however, use the computable correction term for error correction of the functional and use an approximation to the error in the computable correction as error indicators for refinement. Balasubramanian and Newman [108] compare the use of the grid adaptive strategies of Müller and Giles [103] with Venditti and Darmofal [105] to improve the DWR-corrected value of a functional of interest. They concluded that the method used by Venditti and Darmofal was more robust for the 2-D, inviscid, incompressible flow problems in their study [108]. To my knowledge, the idea of error correction of a functional via a DWR error estimate has not yet been applied in the field of neutron transport. It has also not been applied to p or hp AMR options.

The rest of the chapter is organised as follows. The derivation of an error indicator based on the work of Venditti and Darmofal [105], where local values of the error in the error correction are used is described in section 5.2. The results obtained by combining error correction with AMR driven by a number of different error indicators are analysed and compared in section 5.3 for the fixed source test cases. The chapter ends with some closing remarks in section 5.4.

5.2 A more appropriate error indicator to obtain an optimal mesh for the error corrected functional

In chapter 3 it was explained that instead of approximating a QoI denoted by $J(\Psi)$ by $J(\Psi_{\mathrm{h},p})$, a more accurate approximation may be obtained by using the DWR error estimator derived in section 3.3 as a correction term:

$$J(\Psi) \approx J(\Psi_{\mathrm{h},p}) + A(\Psi_{\mathrm{h},p}; \tilde{\Psi}^{\dagger}_{\mathrm{h},p}). \tag{5.1}$$

The error in this approximation is then:

$$e_{\mathrm{c}} = A(\Psi_{\mathrm{h},p}; \tilde{\Psi}^{\dagger} - \tilde{\Psi}^{\dagger}_{\mathrm{h},p}) + A(\Psi_{\mathrm{h},p}; \Psi^{\dagger} - \tilde{\Psi}^{\dagger}) - \epsilon. \tag{5.2}$$

Certain researchers noted that the next largest error term would be better than the computable correction term for use as a refinement indicator when the corrected QoI is to be calculated [105, 108]. This is because this should result in a mesh which is optimal for reducing the error in the *corrected approximation* of the QoI (5.1), rather than the approximation obtained simply from the numerical flux. The aim is therefore to approximate $|e_{\mathrm{c}}|$ over a cell for use as a refinement indicator.

In the linear case $\tilde{\Psi}^{\dagger} = \Psi^{\dagger}$ and $\epsilon = 0$, so equation (5.2) reduces to:

$$e_{\mathrm{c}} = A(\Psi_{\mathrm{h},p}; \Psi^{\dagger} - \Psi^{\dagger}_{\mathrm{h},p}). \tag{5.3}$$

This is the residual of the forward equation $R_{\mathrm{h},p}$ weighted by the error in the adjoint solution and integrated over the phase space. The following equivalence then holds:

$$e_{\mathrm{c}} = \langle R_{\mathrm{h},p}, \Psi^{\dagger} - \Psi^{\dagger}_{\mathrm{h},p} \rangle_{\bar{V}} \equiv \langle R^{\dagger}_{\mathrm{h},p}, \Psi - \Psi_{\mathrm{h},p} \rangle_{\bar{V}}, \tag{5.4}$$

where $R^{\dagger}_{\mathrm{h},p}$ is the residual of the approximate adjoint solution. The equivalence can be made given that both the numerical and analytical solutions of the forward and adjoint equations are in the domain of the forward and adjoint operators respectively and so integration by parts can be performed. A boundary term should appear if the boundary conditions are not vacuum.

The error can be represented by:

$$e_{\mathrm{c}} = \langle R_{\mathrm{h},p}, \Psi^{\dagger}_{\mathrm{h},p+1} - \Psi^{\dagger}_{\mathrm{h},p} \rangle_{\bar{V}} + \langle R_{\mathrm{h},p}, \Psi^{\dagger} - \Psi^{\dagger}_{\mathrm{h},p+1} \rangle_{\bar{V}} \tag{5.5}$$

or equivalently:

$$e_{\mathrm{c}} = \langle \Psi_{\mathrm{h},p+1} - \Psi_{\mathrm{h},p}, R^{\dagger}_{\mathrm{h},p} \rangle_{\bar{V}} + \langle \Psi - \Psi_{\mathrm{h},p+1}, R^{\dagger}_{\mathrm{h},p} \rangle_{\bar{V}} \tag{5.6}$$

The first term of either representation can be taken as an approximation to the error in the corrected value for use as an error indicator. The magnitude of the first term in equation (5.5) shall be denoted by $|e_{\mathrm{c},1}|$ and the magnitude of the first term in equation (5.6) denoted by $|e_{\mathrm{c},2}|$. $|e_{\mathrm{c},1}|$ and $|e_{\mathrm{c},2}|$ will vary in accuracy throughout the domain depending on the problem.

In section 5.3 both $|e_{c,1}|$ and $|e_{c,2}|$ are used separately and a combination of the two to test which works best in different situations. The third option involves first selecting refinement using $|e_{c,1}|$ and separately $|e_{c,2}|$ and then refining the union of all flagged cells. This is expected to be the most robust since a combination error estimator based on both $|e_{c,1}|$ and $|e_{c,2}|$ was suggested for the fully discrete equivalent formulation described by Venditti et al. [106, p47] and because the leading error term in either $|e_{c,1}|$ or $|e_{c,2}|$ will be the most accurate depending on which cell in the domain the quantity is being calculated in.

The definition of $|e_{c,1}|$ and $|e_{c,2}|$ when the lower order approximation to the adjoint equation is used for error correction is as follows (note the use of the subscript, l, to denote the use of the lower order adjoint):

$$|e_{c,1,l}| = |\langle R_{h,p}, \Psi^{\dagger}_{h,p+1} - \Psi^{\dagger}_{h,p} \rangle_{\bar{V}}| \tag{5.7}$$

$$|e_{c,2,l}| = |\langle \Psi_{h,p+1} - \Psi_{h,p}, R^{\dagger}_{h,p} \rangle_{\bar{V}}| \tag{5.8}$$

The equivalent expressions for when a high order approximation to the adjoint is used should be as follows:

$$|e_{c,1,h}| = |\langle R_{h,p}, \Psi^{\dagger}_{h,p+2} - \Psi^{\dagger}_{h,p+1} \rangle_{\bar{V}}| \tag{5.9}$$

$$|e_{c,2,h}| = |\langle \Psi_{h,p+1} - \Psi_{h,p}, R^{\dagger}_{h,p+1} \rangle_{\bar{V}}| \tag{5.10}$$

but $\Psi^{\dagger}_{h,p+2}$ is not calculated in the algorithm used for this work, and therefore the analysis in section 5.3 is conducted only for the low order error correction and refinement indicator.

How to proceed from equation (5.2) in deriving an equivalent error indicator for non-linear problems (such as criticality problems) is less straightforward. This is because, once approximations to $\Psi^{\dagger}$ and $\tilde{\Psi}^{\dagger}$ are made, it is unclear which term is the dominant term in equation (5.2). This investigation is left to further work. For further discussion on the difficulties of computing an *a posteriori* error bound in the non-linear case, see Pierce and Giles's work [98, section 4].

5.3 Application to test cases

Section 5.3.1 describes the implementation of the code and the application to the 1-D test cases described in appendix A. Section 5.3.2 then shows and discusses the results of applying the theory discussed in this chapter to the test cases.

Indicator name	Refine if:
$e_{c,1,l}$	$\lvert e_{c,1,l}\rvert_{cell} < \alpha \max(\lvert e_{c,1,l}\rvert)$
$e_{c,2,l}$	$\lvert e_{c,2,l}\rvert_{cell} < \alpha \max(\lvert e_{c,2,l}\rvert)$
Combination. ind.	$\lvert e_{c,1,l}\rvert_{cell} < \alpha \max(\lvert e_{c,1,l}\rvert)$ or $\lvert e_{c,2,l}\rvert_{cell} < \alpha \max(\lvert e_{c,2,l}\rvert)$

TABLE 5.1: Refinement criteria for the three different error indicators.

5.3.1 Implementation

Four different error indicators for refinement are tested in this section: the locally calculated low order error correction value on the cell and the three variations on a new refinement indicator shown in table 5.1. α is an input parameter defined for the problem. The value of α used for each test case in this section are the same as in chapter 4, denoted in table 4.1.

The low order error correction is the original DWR error estimator using the adjoint solution calculated on the same mesh as the forward equation.

Only fixed source problems are considered in this section due to the complications of deriving an appropriate new error indicator for the corrected value in non-linear problems (as previously explained in section 5.2).

5.3.2 Results and discussion

Figures 5.1-5.4 compare different AMR refinement strategies with no functional correction and low order functional correction using the α values and error indicators described in section 5.3.1. For clarity, h AMR, p AMR and hp AMR are plotted on separate graphs.

5.3.2.1 Test case 1 - AMR with correction

The optimal mesh for test case 1a is one with uniform mesh size and polynomial order. For this reason all refinement indicators choose uniform refinement, be that h AMR (figure 5.1a), p AMR (figure 5.1c) or hp AMR (figure 5.1e).

The results for test case 1b are shown in figures 5.1b, 5.1d and 5.1f.

When h AMR is applied (figure 5.1b), although using the local values of the original DWR error indicators gives the best convergence plots for the non-corrected value of the QoI, they do not give the best convergence plots for the corrected value. A plateau is observed for this

line (green) in figure 5.1b. Better convergence graphs are obtained for the low order corrected quantity using $|e_{c,1,l}|$, $|e_{c,2,l}|$ and the combination error indicator. The mesh iteration algorithm cuts off after a given number of mesh iterations, which is why the graphs for the $|e_{c,1,l}|$ and $|e_{c,2,l}|$ error indicators are cut short compared to the combination error indicator. The combination error indicator results in a lower error in fewer mesh iterations, without the plateau observed when the local error correction value is used as an indicator, thereby reducing the error per DoF significantly.

When p AMR is applied (figure 5.1d), the local error correction values give a good convergence graph, they also exhibit the kink explained in section 4.4.1.1. In this test case the combination error indicator removes the kink in the non-corrected graph, though the slope between mesh iterations two and three is shallow (dark blue). The slope is improved in the low order corrected version (pink). Use of $|e_{c,1,l}|$ or $|e_{c,2,l}|$ independently does not give the best convergence graph for either the non-corrected or low order corrected value.

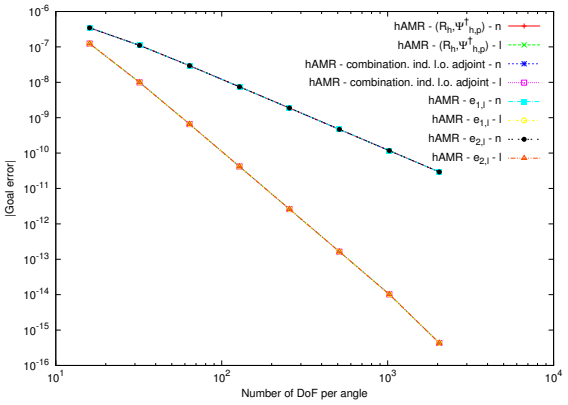
When hp-AMR is applied (figure 5.1f) p refinement is chosen until the polynomial order limit (p=4). Beyond this it is difficult to compare the plots given by the different error indicators, because the algorithm may have followed a p refinement curve if the code was not limited to a fourth order polynomial.

5.3.2.2 Test case 2 - AMR with correction

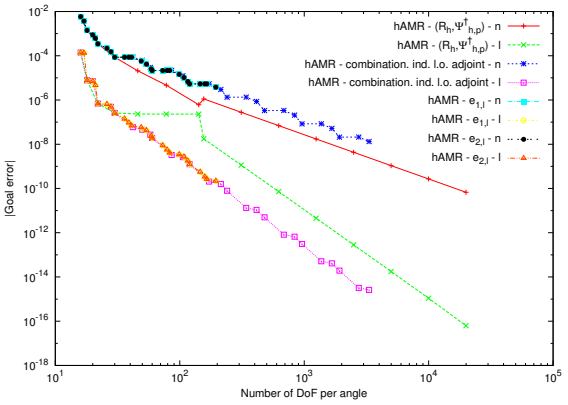
The optimal mesh for test case 2a is one with a uniform mesh size and polynomial order. For this reason all refinement indicators choose uniform refinement, be that h AMR (figure 5.2a), p AMR (figure 5.2c) or p refinement chosen by the hp AMR algorithm (figure 5.2e). It is observed for all AMR types that the error corrected goal error plots plateau at about 10^{-11}. This is due to a limitation in the accuracy obtainable for the solution of the discretised problem: the residual of the discretised problem is non-zero due to it being solved by an iterative procedure which is limited in accuracy by machine precision.

The results for test case 2b are shown in figures 5.2b, 5.2d and 5.2f.

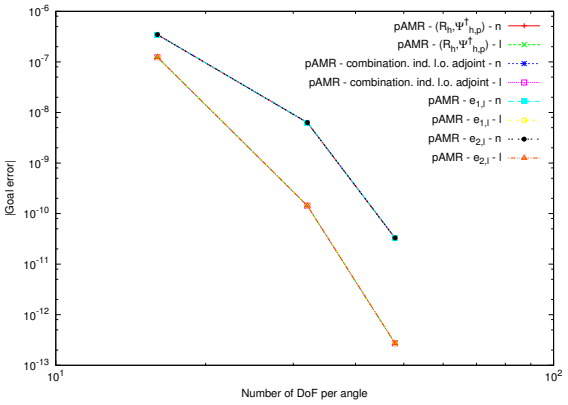
When h AMR is applied (figure 5.2b), although using the local values of the original DWR error indicators gives the best convergence plots for the non-corrected value of the QoI, they do not give the best convergence plots for the corrected value. For the same reasons as for test



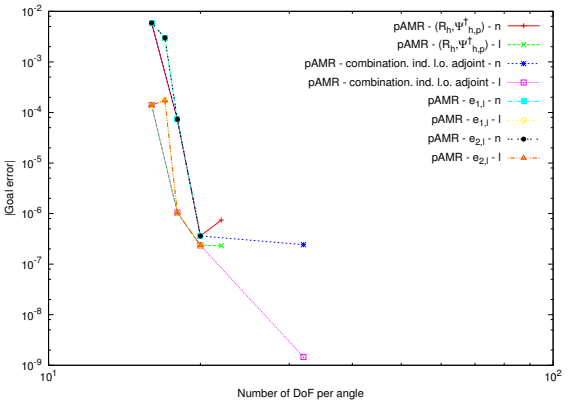
(A) Test case 1a h AMR with corrections.



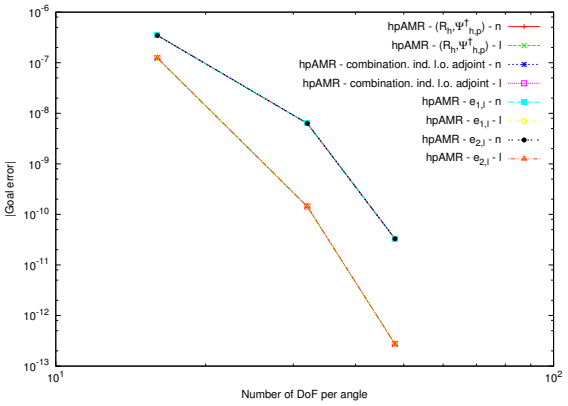
(B) Test case 1b h AMR with corrections.



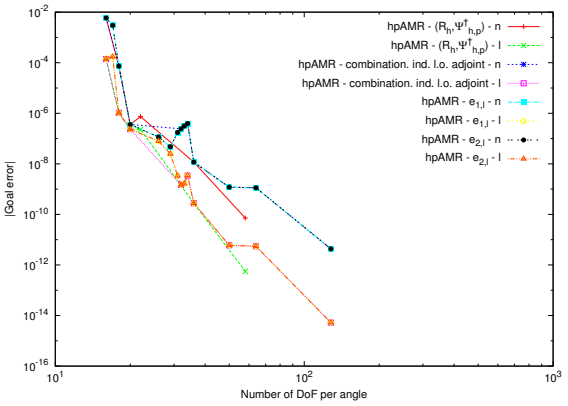
(C) Test case 1a p AMR with corrections.



(D) Test case 1b p AMR with corrections.



(E) Test case 1a hp AMR with corrections.



(F) Test case 1b hp AMR with corrections.

FIGURE 5.1: Test case 1, Adaptivity with corrections (n: no error correction, l: low order error correction).

case 1b (section 5.3.2.1) a better convergence rate is found with the other refinement indicators, provided that the corrected value of the QoI is to be used.

The same comments can not be made when p AMR is applied (figure 5.2d). The combination error indicator results in over-refinement between the first and second mesh iteration due to the rather conservative approach of taking the union of cells flagged for $|e_{c,1,l}|$ or $|e_{c,2,l}|$. In this case, using either $|e_{c,1,l}|$ or $|e_{c,2,l}|$ independently gives a better convergence graph.

For hp AMR (figure 5.2f) p AMR is followed up until the limit (p=4). Beyond this it is difficult to compare the plots given by the different error indicators, because the algorithm may have continued with p refinement if the code was not limited to a fourth order polynomial. However, all error indicators converge to give the same error in the corrected value up until the error is between 10^{-11} and 10^{-12}. These numbers are very small for double precision and so conclusions cannot be drawn from that. It is observed that there seems to be no advantage to using any of the new estimators for hp AMR in this test case.

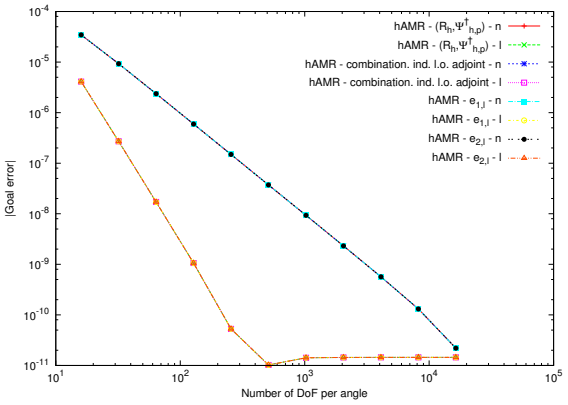
5.3.2.3 Test case 3 - AMR with correction.

This test case has already been shown to be challenging for the DD scheme using a low number of cells in the mesh. Any of the error indicators in table 5.1 applied to test case 3a give a better convergence plot for the corrected QoI than the original DWR indicator (figure 5.3a after the first mesh iteration). The same is observed in figure 5.3c for the p AMR options after the first mesh iteration. For the hp AMR case (figure 5.3e) the combination error indicator performs best in general.

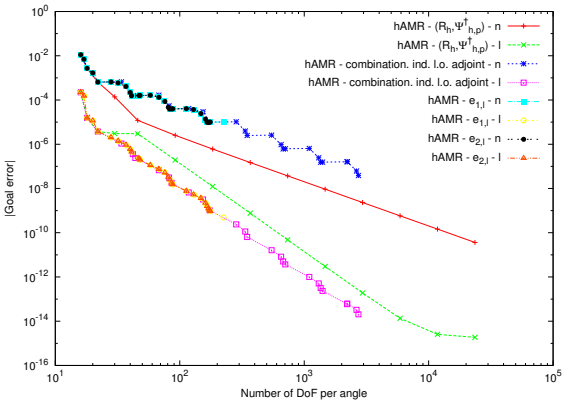
The error correction is false at all mesh iterations for test case 3b (figures 5.3b, 5.3d and 5.3f) due to the adjoint solution, calculated on the same mesh as the forward solution, not being accurate enough for this test case. A higher order adjoint must be used to perform the error correction in this case.

5.3.2.4 Test case 4 - AMR with correction.

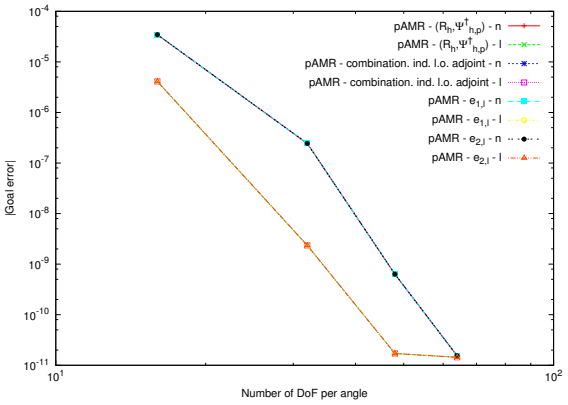
For most test cases the combination error estimate performs best. The convergence graph is less oscillatory for the combination error indicator than for either $|e_{c,1,l}|$ or $|e_{c,2,l}|$, which also perform better than the local value of the error correction as an error indicator. The new error



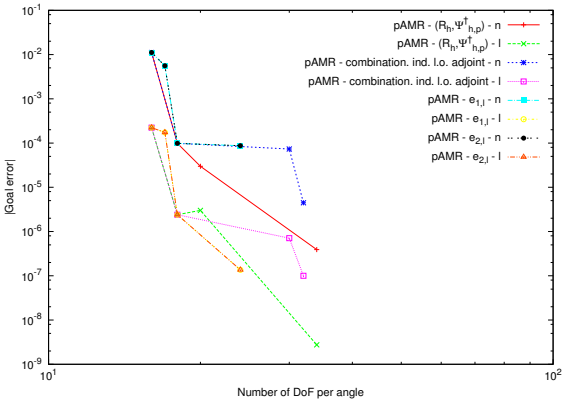
(A) Test case 2a h AMR with corrections.



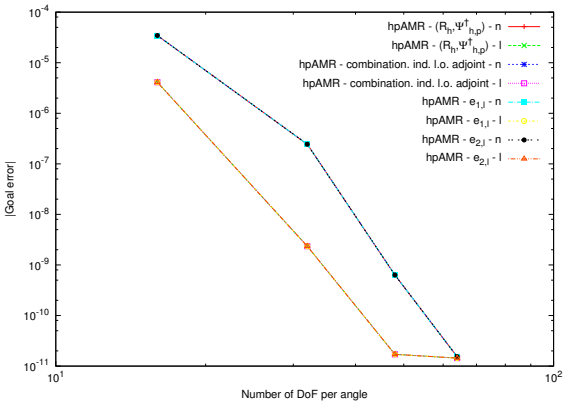
(B) Test case 2b h AMR with corrections.



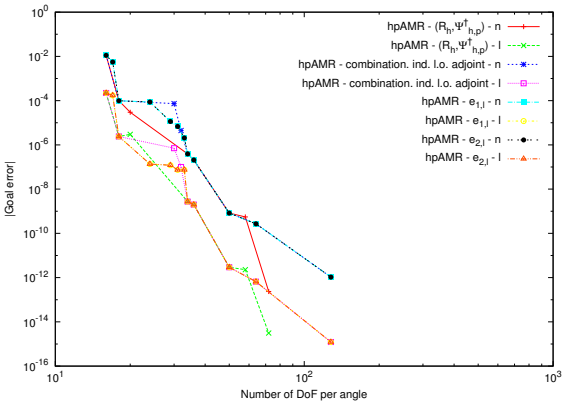
(C) Test case 2a p AMR with corrections.



(D) Test case 2b p AMR with corrections.

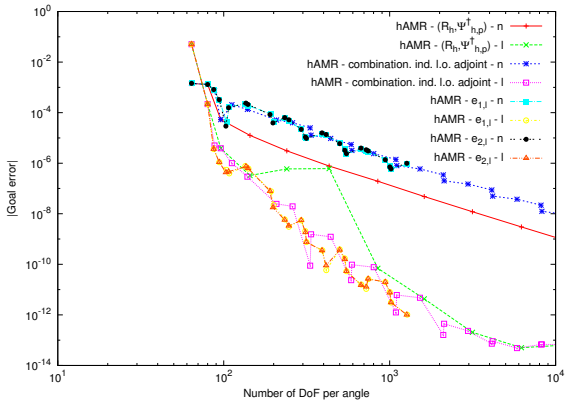


(E) Test case 2a hp AMR with corrections.

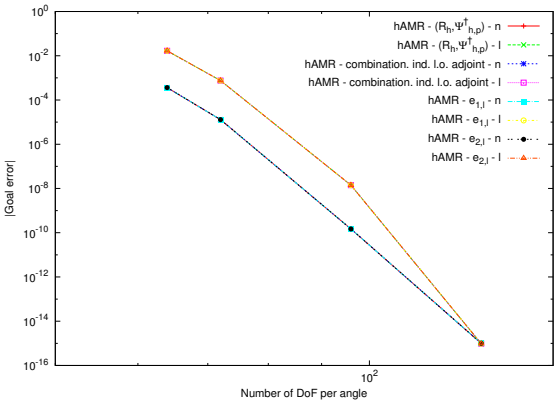


(F) Test case 2b hp AMR with corrections.

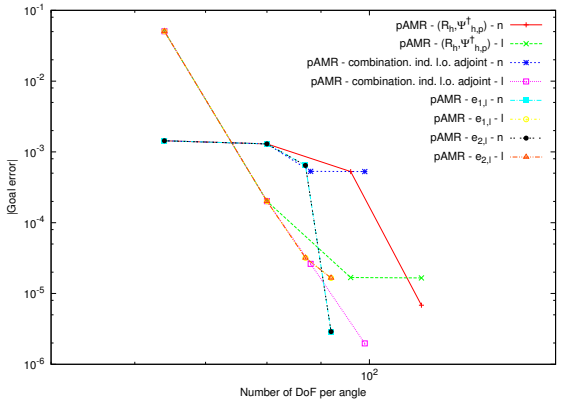
FIGURE 5.2: Test case 2, Adaptivity with corrections (n: no error correction, l: low order error correction).



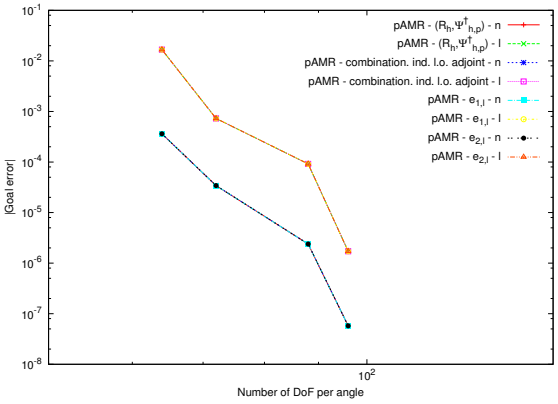
(A) Test case 3a h AMR with corrections.



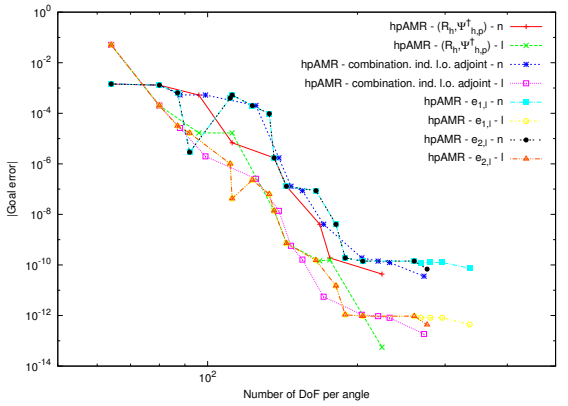
(B) Test case 3b h AMR with corrections.



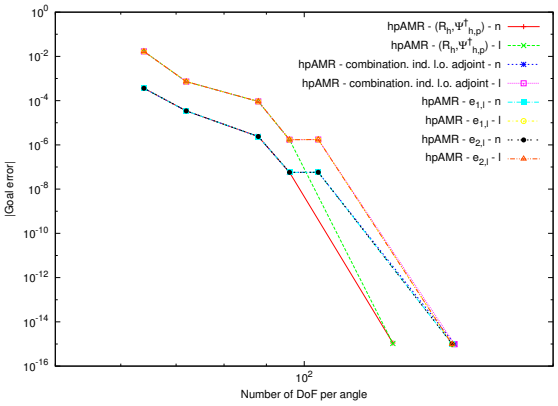
(C) Test case 3a p AMR with corrections.



(D) Test case 3b p AMR with corrections.



(E) Test case 3a hp AMR with corrections.



(F) Test case 3b hp AMR with corrections.

FIGURE 5.3: Test case 3, adaptivity with corrections (n: no error correction, l: low order error correction).

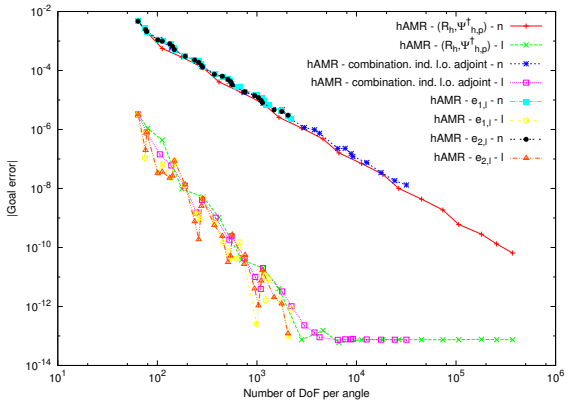
indicators give superior performance at the beginning of the plot in figure 5.4f. Towards the end of the plot in figure 5.4f, the combination error indicator, $|e_{c,1,l}|$ and $|e_{c,2,l}|$ follow the same convergence pattern as the plot given for when the error correction is used as an indicator, but with extra intermediate points.

In test case 4b, in the h AMR case (figure 5.4b), the use of the locally calculated error correction as an indicator for h AMR results in a less accurate corrected functional value per DoF than if the error correction and uniform h refinement were combined (figure 3.10c). Using the combination error indicator removes this issue.

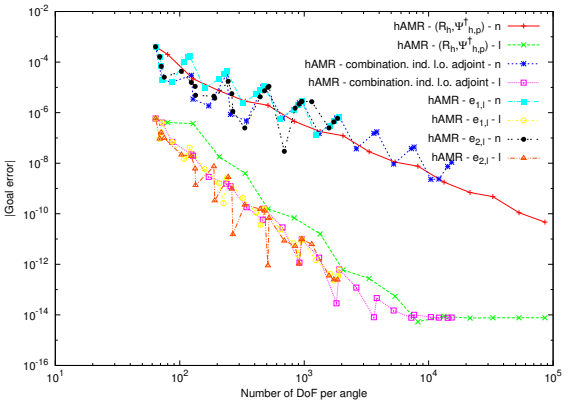
5.4 Closing comments for this chapter

This chapter investigated the combination of functional error correction with AMR for fixed source test cases with a detector type goal. It was shown that although the local value of the error correction quantity over a cell is the optimal error indicator for obtaining an efficient mesh for the non-corrected value of the functional, the same could not always be said for the corrected value of the functional. In test case 4b for example, the result of the corrected functional when h AMR is used gives a worse convergence rate than if uniform h refinement were used in conjunction with error correction. To remedy the situation a new error indicator was used based on the error in the error correction quantity, by weighting the residual of the forward equation with the difference between a reference solution to the adjoint equation (obtained by a higher order approximation within the mesh cells) and the adjoint solution used for the DWR error correction. The convergence graph using this quantity gave better convergence graphs in general. However, they were more oscillatory. It is possible that using a lower value of α would help with this.

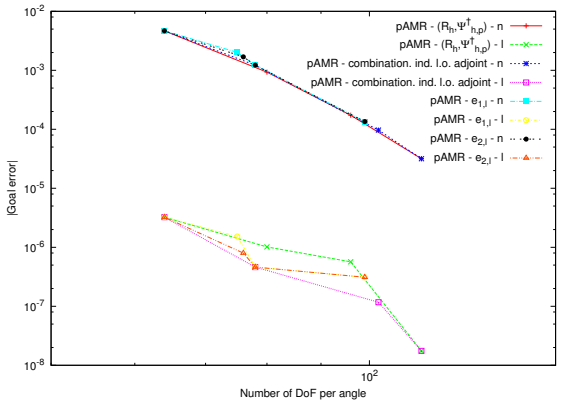
Future work could investigate the use of a new rule of thumb for α for the new error indicator when an error corrected value of the functional is calculated. How to approximate the error in the error correction to the K_{eff} could also be investigated.



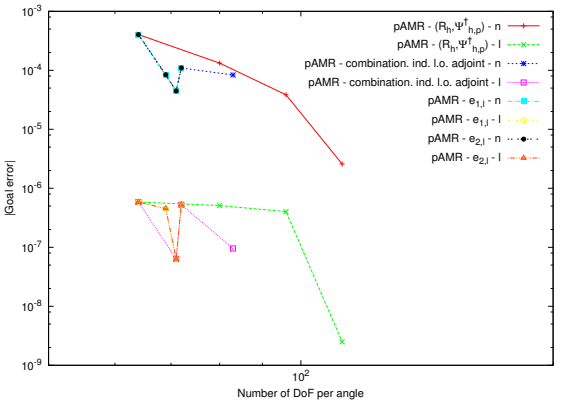
(A) Test case 4a h AMR with corrections.



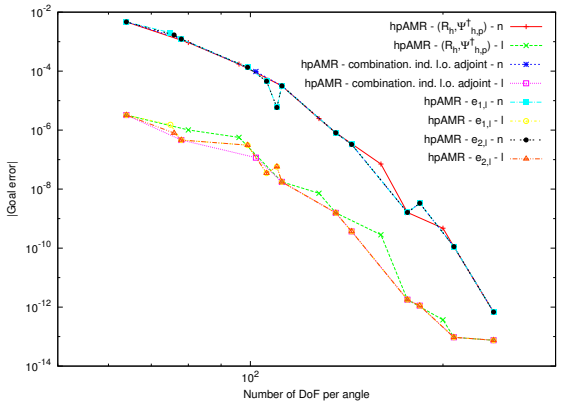
(B) Test case 4b h AMR with corrections.



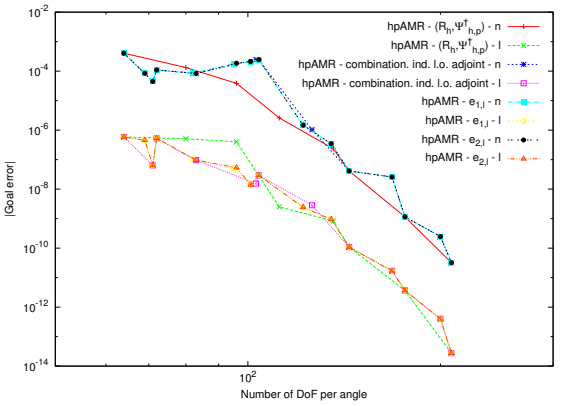
(C) Test case 4a p AMR with corrections.



(D) Test case 4b p AMR with corrections.



(E) Test case 4a hp AMR with corrections.



(F) Test case 4b hp AMR with corrections.

FIGURE 5.4: Test case 4, adaptivity with corrections (n: no error correction, l: low order error correction).

6. Multidimensional DWR error estimation

This chapter extends the DWR formulation to the 2-D DD-S_N scheme. The method is applied to both linear fixed source type problems with detector function goal (which is also linear) as well as the eigenvalue criticality type problem with eigenvalue (K_{eff}) goal which is a non-linear equation with a non-linear goal. The performance of the error estimator is evaluated by applying it to test cases with reference solutions. A comment on the 3-D theory is made but not implemented.

The novel parts of this chapter include the following:

- *Showing that the extension of the WR vertex interpretation of the 1-D DD scheme to 2-D gives the box method*
- *Suggesting a point-wise representation of the DD flux within a cell for use in the DWR error estimator*
- *Showing an equivalence between the box method and the traditional DD method*
- *The application of DWR functional error estimation to the 2-D DD method*
- *The application of DWR functional error estimation to the 2-D box method*
- *Calculation of the DWR functional error estimate using an arbitrary uniform mesh refinement difference between the forward and adjoint solutions for the 2-D box method*
- *A theoretical view of the challenges that a 3-D implementation would bring*

6.1 Introduction

In this chapter, two very similar 2-D spatial discretisation methods are presented: the box method, and the DD method. Their equivalence is then explained. The DD method is the traditional way of extending the 1-D DD equations to 2-D [32] and is the method employed in multi-dimensional DD-S_N neutron transport codes. The box method, however, is what is obtained if the WR view of the 1-D method (see section 3.2) is extended to 2-D. Both methods are explained in sections 6.2 and 6.3. The equivalence between the two is explained in section 6.4.

The equivalence between the two methods means that any stability issues that are evident from the WR method view of the box method will also apply to the DD method. Another consequence of the equivalence is that the analytical error in integral quantities will be the same for both methods.

A DWR technique is suggested for both methods in section 6.5. The results of the box method follows from the rigorous arguments presented in chapter 3. In the DD method the results must be mapped into a FE space, and two options are presented.

The methods are applied to fixed source and criticality 2-D test cases and the results discussed in section 6.6. Although the methods are not applied to 3-D test cases in this thesis, section 6.7 explains how they would be. It also shows that the challenges are very similar to the 2-D case. Closing comments for the chapter are given in section 6.10.

6.2 The box method

In the lowest order 1-D DD case, the vertex values at the edges of each cell in the discretised domain are the unknowns solved for. Linear behaviour within the cell can be inferred from the DD auxiliary equation for each cell. This means that the approximate solution is "uniquely determined by its values at cell boundaries" [124]. It was shown by Pitkäranta and in chapter 3 that the 1-D DD scheme is equivalent to a Petrov-Galerkin finite element scheme, in that it is a WR scheme with piece-wise constant test functions and piece-wise linear trial functions [124, 137]. The representation of the angular flux within a given cell, for a given angle can be expressed as:

$$\Phi(x)_{\text{cell}} = \Phi(\zeta) = c_1 + c_2\zeta = c_1 P_0(\zeta) + c_2 P_1(\zeta), \tag{6.1}$$

where:

$$\zeta = \frac{2x - x_i - x_{i+1}}{x_{i+1} - x_i}, \tag{6.2}$$

$P_l(\zeta)$ is the l'th Legendre polynomial and the subscripts i and $i+1$ are used to indicate the left and right sides of cell i respectively.

If nc is the number of cells, then this gives nc equations for $nc+1$ parameters per angle. The remaining parameter is given by imposing an essential Dirichlet boundary condition for each angle.

Pitkäranta notes that this method can be extended in to 2-D by choosing piece-wise constant test functions and piece-wise bilinear trial functions [124]:

$$\Phi(x,y)_{\text{cell}} = \Phi(\zeta,\gamma) = c_1 + c_2\zeta + c_3\gamma + c_4\zeta\gamma, \tag{6.3}$$

where:

$$\zeta = \frac{2x - x_{\text{w}} - x_{\text{e}}}{x_{\text{e}} - x_{\text{w}}}, \tag{6.4}$$

$$\gamma = \frac{2y - y_{\text{s}} - y_{\text{n}}}{y_{\text{n}} - y_{\text{s}}}, \tag{6.5}$$

and the subscripts e, w, n and s represent the east, west, north and south sides of a given cell, as shown in figure 6.1.

The trial space (for a given angular direction) in this case is a $(n_x+1)(n_y+1)$-dimensional space, where n_x and n_y are the number of cells in the x and y dimension respectively. Pitkäranta notes that the approximate solution is then determined uniquely by the vertex values. Weighting by each basis function of the test space gives $(n_x n_y)$ equations and (n_y+n_x+1) boundary conditions are then imposed on the trial space in a point-wise fashion in order to be able to solve for all vertex unknowns.

The solution within a cell in terms of the vertex values for a given angular direction is:

$$\Phi(x,y)_{\text{cell}} = \Phi(\zeta,\gamma) = \Phi_1 N_1 + \Phi_2 N_2 + \Phi_3 N_3 + \Phi_4 N_4, \tag{6.6}$$

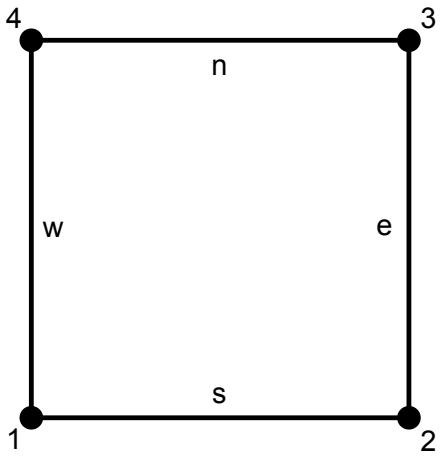


FIGURE 6.1: Node numbering for box method and DD scheme

where

$$N_1 = \frac{1}{4}(1-\zeta)(1-\gamma), \tag{6.7}$$

$$N_2 = \frac{1}{4}(1+\zeta)(1-\gamma), \tag{6.8}$$

$$N_3 = \frac{1}{4}(1+\zeta)(1+\gamma), \tag{6.9}$$

$$N_4 = \frac{1}{4}(1-\zeta)(1+\gamma). \tag{6.10}$$

Equations (6.7)-(6.10) show the local variation of N_{node} over a single cell. Each N_{node} is a globally continuous function in terms of x and y that is unity at the associated node and zero at all other nodes, resulting in a bilinear tent function.

The test function space is discontinuous and defined as follows:

$$M_{i,j}(x) = \begin{cases} 1 & \text{if } x \in (x_i, x_{i+1}) \text{ and } y \in (y_i, y_{i+1}) \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } i = 1, 2, \ldots, n_x \text{ and } j = 1, 2, \ldots, n_y. \tag{6.11}$$

Keeping the vertex values as the unknown parameters gives equation (6.12), see figure 6.1 for node numberings. There are 4 vertex values in each cell, 3 of them (points 1,2 and 4 for $\mu > 0$ and $\eta > 0$) are determined by the imposed essential boundary conditions or values calculated for previous cells in the sweep. These values are assumed continuous between cells by searching for a function in a subspace of H^1 for each ordinate. The remaining unknown (point 3 for $\mu > 0$ and $\eta > 0$) can then be calculated from the equation that results from the WR procedure.

The equations that result are the same as the method used by Valougeorgis [59] called the *box method*:

$$\frac{\mu}{2\Delta x_{ij}}\left(\Phi_3+\Phi_2-\Phi_4-\Phi_1\right)+\frac{\eta}{2\Delta y_{ij}}\left(\Phi_3+\Phi_4-\Phi_1-\Phi_2\right)+\frac{\Sigma_{t,ij}}{4}\left(\Phi_3+\Phi_2+\Phi_4+\Phi_1\right)=q_{ij}. \tag{6.12}$$

However, in Pitkäranta's derivation of the WR DD method, vertex terms are collected together in the resulting equations [124, eq.34] to form the traditional 2-D DD equations (see (2.36)-(2.38)). The implications of this step is discussed in section 6.3.

The WR method is extended to multi-ordinate problems in exactly the same way as in 1-D (section 3.2.3). Hence, the trial functions map out a subspace of $(H^1)^{\frac{N(N+2)}{2}}$ and the test functions map out a subspace in $(L^2)^{\frac{N(N+2)}{2}}$.

The box method seeks a *strong* solution to the neutron transport equation and not a *weak* solution. This is because (in the multi-ordinate case) a solution to the angular flux is sought in a subspace of $(H^1)^{\frac{N(N+2)}{2}}$. Less regular solutions than $(H^1)^{\frac{N(N+2)}{2}}$ will therefore not be well represented by the box method (neither by the DD method due to the equivalence that will be shown in section 6.4).

The most regular the scalar flux can be will be when it is a member of $H^{\frac{3}{2}}$ [138, 139]. However, in pure absorber or void regions, the regularity of the solution can reduce to $H^{\frac{1}{2}}$. It is for this reason that test cases with regions such as this will not be well-represented by the box method. In PWR reactors there are very few void or purely absorptive regions and thus the box method can be deemed appropriate. However, it would not be the most appropriate for other reactor designs such as boiling water reactors (BWRs) or advanced gas cooled reactors (AGRs), where there would be large void regions [40, 140]. Similarly, it would not be the most appropriate discretisation scheme to use when modelling a loss of coolant (LOC) accident in a PWR reactor, as this would also result in the need to model void regions. Finally, shielding applications can also contain discontinuities that would render the use of a DD code unsuitable [141].

One example of a test case that has a solution in $H^{\frac{1}{2}}$ is the case of a homogeneous, purely absorbing, square domain, with zero volumetric source and different incoming fluxes on the two incoming boundaries [142]. In this case the solution is discontinuous, and thus the box method will not converge to the correct point-wise solution as the cell-size reduces.

Oscillations are still observed when the solution is in H^1, such as the vacuum boundary condition, homogeneous, pure-absorber, constant source, square domain problem. In this case the flux is continuous but the first order derivatives are discontinuous [142]. In the box method, the basis functions for the approximation can be discontinuous in gradient at cell edges, but have continuous gradients within the cell (6.7)-(6.10). The oscillations observed are therefore due to the discontinuous gradients not being aligned with mesh cell edges, where the slope is allowed to vary discontinuously in the approximate solution.

6.3 2-D DD method

In Pitkäranta's paper, the point values of equation (6.12) are collected together to give equations (2.36)-(2.38) with no further comment. It is not explained that if the equation using the collected terms is solved (which for a bilinear approximation is the same as averaged cell and cell-edge terms) how can the point-wise approximation be reconstructed?

Once terms in the WR equation for the cell have been collected into averaged face values (6.13), there are still 4 unknowns for the cell. It is now only possible to define 2 boundary terms and there remain 2 unknowns:

$$\frac{\mu}{\Delta x_{ij}}\left(\bar{\Phi}_e - \bar{\Phi}_w\right) + \frac{\eta}{\Delta y_{ij}}\left(\bar{\Phi}_n - \bar{\Phi}_s\right) + \frac{\Sigma_{t,ij}}{4}\left(\bar{\Phi}_e + \bar{\Phi}_w + \bar{\Phi}_n + \bar{\Phi}_s\right) = q_{ij}. \tag{6.13}$$

Both remaining unknowns cannot be solved for using the single WR equation for the cell. The diamond difference auxiliary equations must be used instead:

$$\bar{\Phi}_c = \frac{\bar{\Phi}_e + \bar{\Phi}_w}{2}, \tag{6.14}$$

$$\bar{\Phi}_c = \frac{\bar{\Phi}_n + \bar{\Phi}_s}{2}. \tag{6.15}$$

Equations (6.14) and (6.15) are consistent with a bilinear functional representation within the cell, but it must be noted that in taking this approach the ability to reconstruct the bilinear term from the unknowns solved for is lost.

If the vertex values have been solved for using the box method, it is possible to calculate the cell averaged fluxes using the following matrix multiplication:

$$\begin{bmatrix} \bar{\Phi}_s \\ \bar{\Phi}_n \\ \bar{\Phi}_e \\ \bar{\Phi}_w \end{bmatrix} = \mathcal{T} \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \end{bmatrix}. \tag{6.16}$$

The transformation matrix from cell vertex values to cell edge-averaged values in (6.16) is denoted by $\mathcal{T}$. The determinant of $\mathcal{T}$ is zero and so $\mathcal{T}^{-1}$ does not exist. Hence it is not possible to calculate the DD approximation and recover the bilinear behaviour of the approximate solution within the cell and coefficient c_4 of equation (6.3) is undetermined. It could be set to an infinite number of possibilities. If a point-wise essential boundary condition was imposed instead of a weakly imposed one and the result assumed continuous (as in Pitkäranta's paper [124]), then these values could be inputted into the vector in equation (6.16), thereby determining the nodal values for that cell. To obtain these values for each cell would mean applying the same mapping to each cell by sweeping through the mesh once again. This would lead to the same piece-wise bilinear point-wise representation as the box method.

Alternatively, the DD scheme could be viewed as a scheme that weakly imposes the boundary conditions and interface conditions and allows for a discontinuous representation of the flux (a member of $(L^2)^{\frac{N(N+2)}{2}}$). The integral of the incoming fluxes along the incoming cell edges on the boundary of the domain is set to zero (for vacuum boundary conditions), rather than the boundary condition being set as point-wise zero along the cell edge. Similarly, the information passed on to the following cells in the sweep can be viewed as weaker than if the cell vertex values had been retained. In the box method, the information passed on from the previous cells in the sweep is a linear variation over the cell edge (assuming an angle in the first octant, the linear behaviour of the approximate solution with respect to y at the fixed point x_e of the previous cell in the x direction, and the linear behaviour of the approximate solution with respect to x at the fixed point y_n for the previous cell in the y direction). In the discontinuous view of the DD approximation however, only the integral behaviour of the function within the previous cells along these fixed values of x (for the west face) and y (for the south face) is passed on to the current cell. Due to the "linear information" not being passed on to following cells in the sweep it can be assumed that the approximation has a zero bilinear term within the cell and that the

results within the cell is in fact a linear function as in equation (6.17).

$$\Phi(x,y)_{\text{cell}} = \Phi(\zeta,\gamma) = c_1 + c_2\zeta + c_3\gamma. \tag{6.17}$$

The fact that the approximation within the cell is linear (and not bilinear) means that the gradient with respect to either coordinate direction is constant throughout the whole cell. This means that the gradient w.r.t x within the cell is uniquely determined by the east and west face averaged values (6.18). Similarly, the gradient w.r.t y is uniquely determined by the north and south face averaged values (6.19).

$$c_2 = \Phi_x = \frac{\bar{\Phi}_w - \bar{\Phi}_e}{2}, \tag{6.18}$$

$$c_3 = \Phi_y = \frac{\bar{\Phi}_n - \bar{\Phi}_s}{2}. \tag{6.19}$$

Finally, c_1 is the average value of the flux in the cell denoted by $\bar{\Phi}_c$. It is given by equations (6.14) and (6.15) since for a linear or bilinear approximation, these relations are true by definition. To clarify notation, equation (6.17) can be re-written as:

$$\Phi(\zeta,\gamma) = \bar{\Phi}_c P_{00} + \Phi_x P_{10} + \Phi_y P_{01}, \tag{6.20}$$

given:

$$P_{00} = P_0(\zeta)P_0(\gamma), \tag{6.21}$$

$$P_{10} = P_1(\zeta)P_0(\gamma), \tag{6.22}$$

$$P_{01} = P_0(\zeta)P_1(\gamma). \tag{6.23}$$

Weakly imposing the vacuum boundary conditions results in (6.20) giving a negative flux approximation for the flux at the lower left hand corner of the domain (when $\mu > 0$ and $\eta > 0$). Negative flux is not a physical result, but is as accurate as can be reproduced from the information retained in the DD scheme: assuming linear variation but no bilinear variation in the cell, weakly imposed boundary conditions and discontinuity between cells (though weakly continuous in that the cell edge averages are the same for both neighbouring cells). Flux positivity is never guaranteed for the multi-dimensional DD averaged cell edge values either, an example of where the DD equations can give negative fluxes for any cell size is given in appendix C.

The idea of a weakly continuous scheme is discussed for the 2D nodal schemes presented by Hennart et al. [143]. These are schemes that are point-wise discontinuous over cell edges, but are continuous in some edge moments. The following symbol is given to the various weakly discontinuous schemes: $WD_{p,u}$. p is the number of parameters per cell and u is the number of unknowns. The parameters are cell or edge moments, u of which are unknown and $p-u$ are known thanks to boundary conditions or values calculated for previous cells in the sweep. The unknown parameters are solved by generating u simultaneous equations via a weighted residual method. This is where the traditional, moment based DD scheme differs from Hennart's polynomial nodal schemes. Assuming a linear variation within the cell, although 5 parameters are involved in the equations, only 3 of them are independent, of which 2 are known due to boundary conditions or the results from previous cells in the sweep. Once the incoming boundary edge moments are known and the central moment is known, the outgoing boundary edge moments are fixed by the assumption that the flux is linear with a zero bilinear term. This is why only one WR equation is required, the auxiliary equations are just a means of calculating the values that are already determined and fixed by the scheme.

6.4 The equivalence between the 2-D DD and box method

In this section, the weakly discontinuous view of the DD method described in section 6.3 will be referred to as the DD method, the box method and Pitkäranta's continuous view of the DD method will be referred to as the box method.

Although it is clear that the point-wise representation is different assuming a linear variation within the cell (DD) compared to a bilinear approximation (box method). It is possible to show that the cell and cell-edge averaged fluxes are the same for both methods. This is shown in section 6.4.1. From this equivalence one can draw some practical conclusions. These are discussed in section 6.4.2.

6.4.1 Proof of DD and box method equivalence

In the box method, one of the corner values are unknown and the rest are known from boundary conditions, or the results given from previous cells in the sweep. If the angle is in the first quadrant, then the unknown flux is that at corner 3 of figure 6.1, and is given in terms of the 3

known values as follows:

$$\Phi_3 = \frac{q - \frac{\Sigma_t}{4}(\Phi_1 + \Phi_2 + \Phi_4) - \frac{\mu}{2\Delta x}(\Phi_2 - \Phi_4 - \Phi_1) - \frac{\eta}{2\Delta y}(\Phi_4 - \Phi_1 - \Phi_2)}{\frac{\mu}{2\Delta x} + \frac{\eta}{2\Delta y} + \frac{\Sigma_t}{4}}. \tag{6.24}$$

In the DD method the first unknown to be calculated is the cell-averaged flux, in terms of the averaged incoming fluxes on south and west faces of the cell (figure 6.1). From this, the north and east face average flux values can be calculated [32]. The cell-averaged flux as a function of the known incoming fluxes is as follows:

$$\bar{\Phi}_\mathrm{c} = \frac{q + \frac{2\mu}{\Delta x}\bar{\Phi}_\mathrm{w} + \frac{2\eta}{\Delta y}\bar{\Phi}_\mathrm{s}}{\Sigma_t + \frac{2\mu}{\Delta x} + \frac{2\eta}{\Delta y}}. \tag{6.25}$$

The first step in the proof is to show that the cell average flux given by the box method is the same as that given by the DD method:

$$\bar{\Phi}_\mathrm{c,bm} = \frac{\Phi_1 + \Phi_2 + \Phi_3 + \Phi_4}{4} \tag{6.26}$$

$$= \frac{\Phi_1 + \Phi_2 + \Phi_4}{4} + \frac{q - \frac{\Sigma_t}{4}(\Phi_1 + \Phi_2 + \Phi_4) - \frac{\mu}{2\Delta x}(\Phi_2 - \Phi_4 - \Phi_1) - \frac{\eta}{2\Delta y}(\Phi_4 - \Phi_1 - \Phi_2)}{4\left(\frac{\mu}{2\Delta x} + \frac{\eta}{2\Delta y} + \frac{\Sigma_t}{4}\right)} \tag{6.27}$$

$$= \frac{\begin{array}{r}(\Phi_1 + \Phi_2 + \Phi_4)\left(\frac{\mu}{2\Delta x} + \frac{\eta}{2\Delta y} + \frac{\Sigma_t}{4}\right) + q - \frac{\Sigma_t}{4}(\Phi_1 + \Phi_2 + \Phi_4) - \frac{\mu}{2\Delta x}(\Phi_2 - \Phi_4 - \Phi_1) \\ - \frac{\eta}{2\Delta y}(\Phi_4 - \Phi_1 - \Phi_2)\end{array}}{4\left(\frac{\mu}{2\Delta x} + \frac{\eta}{2\Delta y} + \frac{\Sigma_t}{4}\right)} \tag{6.28}$$

$$= \frac{q + \Phi_1\left(\frac{\mu}{\Delta x} + \frac{\eta}{\Delta y}\right) + \Phi_2\left(\frac{\eta}{\Delta y}\right) + \Phi_4\left(\frac{\mu}{\Delta x}\right)}{4\left(\frac{\mu}{2\Delta x} + \frac{\eta}{2\Delta y} + \frac{\Sigma_t}{4}\right)} \tag{6.29}$$

$$= \frac{q + \left(\frac{\Phi_1+\Phi_4}{2}\right)\left(\frac{2\eta}{\Delta y}\right) + \left(\frac{\Phi_1+\Phi_2}{2}\right)\left(\frac{2\mu}{\Delta x}\right)}{4\left(\frac{\mu}{2\Delta x} + \frac{\eta}{2\Delta y} + \frac{\Sigma_t}{4}\right)} \tag{6.30}$$

The quantities $\left(\frac{\Phi_1+\Phi_4}{2}\right)$ and $\left(\frac{\Phi_1+\Phi_2}{2}\right)$ are the average west face and south face fluxes respectively and so equations (6.30) and (6.25) are equivalent for cells on a Dirichlet boundary. For them to be equivalent in the body of the domain the incoming average fluxes from previous cells in the sweep must also be equivalent. This can be proven by showing that the outgoing face-averaged fluxes are the same for both the box method and DD method.

The auxiliary equations for the DD method (6.14)-(6.15) hold for a cell-wise bilinear approximation, and therefore for the box method. Since it has already been shown that the west face and south face averaged flux values are the same due to the Dirichlet boundary conditions and that the cell averaged fluxes are the same by equation (6.30), then it follows from the auxiliary equations that:

$$\bar{\Phi}_{\mathrm{e,DD}} = \bar{\Phi}_{\mathrm{e,bm}} \tag{6.31}$$

and

$$\bar{\Phi}_{\mathrm{n,DD}} = \bar{\Phi}_{\mathrm{n,bm}} \tag{6.32}$$

for a corner boundary cell. In addition to this, since the above shows that the outgoing averaged fluxes of a corner cell are the same for both methods. The incoming average fluxes will be the same for the following cells. The same logic can be applied to all subsequent cells in the domain, thus the cell average and face average fluxes are equivalent for the box method and DD method throughout the whole domain.

Given the equivalence of the cell centre and averaged fluxes one can also establish that the coefficients of the first three Legendre polynomial products are the same for the DD (equation (6.17)) and box method (equation (6.3)). This is shown as follows:

Proof. The functions associated with each vertex value in a bilinear approximation are given in equations (6.7)-(6.10). Thus, the first three coefficients of equation (6.3) can be given in terms of cell vertex values, or cell edge averaged values as follows:

$$c_1 = \frac{1}{4}\left(\Phi_1 + \Phi_2 + \Phi_3 + \Phi_4\right) = \frac{1}{2}\left(\bar{\Phi}_w + \bar{\Phi}_e\right) = \frac{1}{2}\left(\bar{\Phi}_s + \bar{\Phi}_n\right), \tag{6.33}$$

$$c_2 = \frac{1}{4}\left(-\Phi_1 + \Phi_2 + \Phi_3 - \Phi_4\right) = \frac{1}{2}\left(\bar{\Phi}_e - \bar{\Phi}_w\right), \tag{6.34}$$

$$c_3 = \frac{1}{4}\left(-\Phi_1 - \Phi_2 + \Phi_3 + \Phi_4\right) = \frac{1}{2}\left(\bar{\Phi}_n - \bar{\Phi}_s\right). \tag{6.35}$$

The final coefficient can be given in terms of cell vertex values as follows:

$$c_4 = \frac{1}{4}\left(\Phi_1 - \Phi_2 + \Phi_3 - \Phi_4\right), \tag{6.36}$$

but the vertex values cannot be combined in such a way as to replace them by combinations of the cell-edge averaged fluxes.

6.4.2 Practical conclusions that can be drawn from this equivalence

It is well-known that the DD method suffers from instabilities in 1-D when the cell size is not small enough [32, 119]. It is also never guaranteed that the cell-average fluxes in 2-D shall converge to the analytical solution [141, 142]. It was shown by Duo and Azmy that convergence of the cell-average fluxes was not obtained using the DD method for a homogeneous, non-scattering domain with different incoming boundary conditions [141, 142] since the analytical solution is discontinuous along the singular characteristic. In the case of equal incoming boundary sources, the analytical solution is continuous but the gradient is discontinuous. Although the method does converge in the average cell flux value for this case, the asymptotic convergence rate is only fractional [141, 142, 144].

Given that the box method is a WR method it is easier to view the mathematical arguments for the convergence of the box method. It is also clear that any instabilities that result will also be shown in the DD method thanks to the equivalence shown in section 6.4.1.

It was noted in section 6.2 that the box method seeks a strong solution (as opposed to a weak or classical solution) to the transport equation and thus the solution must be a member of H^1 for the WR integral to be bounded. The solution is sought in a subset of H^1 that is infinitely continuous within a cell and discontinuous in gradient at cell edges. It is for this reason that it is impossible to represent a discontinuous solution to the transport equation (such as the discontinuous benchmark used by Duo and Azmy [141]) using the box method (and therefore the DD method). It is also the reason why oscillations and slow convergence are observed when the gradient discontinuity does not coincide with cell edges (such as Larsen's benchmark [144]).

The above also explains why the diamond difference singular characteristic tracking (DD-SCT) algorithm, devised by Duo et al. [142, 145], has a significantly improved convergence rate compared to the traditional DD scheme for the L^2, L^1 and L^∞ error norms for Larsen's benchmark problem [142, 145]. In the discontinuous test cases [142], the L^2 and L^1 norm error convergence rates are significantly improved when the DD-SCT algorithm is used. The L^∞ error norm, which did not converge for the DD scheme, converges at a rate close to unity for the DD-SCT scheme [145]. The DD-SCT method works by applying a step characteristic in the cells traversed by the discontinuity (in the result itself or its gradient) and the standard DD scheme in all other cells [142, 145].

In addition to the explanation about stability that comes from the mathematical representation of the box method and its cell-average equivalence to the DD method, a comment can be made on the implementation of accelerations schemes. EDF R&D had already developed an SP_N solver that was in a mature state of development when it was decided to develop the DOMINO S_N code. DD was chosen as the spatial discretisation scheme because of the already published results on the compatibility of using a "Raviat-Thomas discretisation of the P_1 spherical harmonic equations with Gauss-Legendre integration of the mass matrices" for use in synthetic acceleration of the source iteration scheme [12]. The fact that the box method gives the same cell and edge-averaged flux results means that the same arguments hold when applying the correction term to the flux in the diffusion synthetic acceleration scheme (DSA). Hence, if the box method is found to give a better approximation to the error than the DD method, the unknowns solved for can be changed without changing the acceleration scheme being used.

6.5 DWR error estimator

The DWR error estimator and local indicators are easy to define for the box method due to it being equivalent to the DD method in 1-D. The form that the DWR estimator and local indicators take are given in section 6.5.1. In the 2-D DD case, as explained in section 6.5.1, one can either calculate the vertex values by taking the DD unknowns and boundary conditions and sweeping through the cells, or it can be assumed that the flux is piece-wise linear, point-wise discontinuous between cells but continuous in cell-edge fluxes. Due to these different options, different forms of the DWR estimator are given for the DD method in section 6.5.2.

6.5.1 DWR for the box method

Since the box method is the direct extension of the WR formulation of the 1-D DD equations, the DWR estimator is easily derived since it follows the exact same arguments as in 1-D:

$$\mathrm{DWR}_{\mathrm{Box}} = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \sum_{m=1}^{\frac{N(N+2)}{2}} w_m \int_{A_{ij}} R_{\mathrm{h,Box}}(x, y, \Omega_m) \Phi_m^{\dagger}(x, y, \Omega_m) dx dy. \tag{6.37}$$

The cell-wise indicators if AMR was to be conducted would then be given by:

$$\text{Indicator}_{i,j} = \left| \sum_{m=1}^{\frac{N(N+2)}{2}} w_m \int_{A_{ij}} R_{\text{h,Box}}(x, y, \Omega_m) \Phi_m^{\dagger}(x, y, \Omega_m) dx dy \right|. \tag{6.38}$$

6.5.2 DWR for the 2-D DD method

If the box vertex values are reconstructed in such a way as to take into account point-wise enforced boundary conditions, then the same DWR error estimator as the box method can be used (6.37) and it would give the same results.

If the DD method results are mapped into a discontinuous bilinear finite element space (with the bilinear term set to zero), the DWR takes the same form as that derived by Lathouwers for the DGFEM scheme [89, 90].

This results in the following DWR for a given cell:

$$\begin{aligned} \text{DWR}_{\text{cell}} = \sum_{m=1}^{\frac{N(N+2)}{2}} \Bigg(& \int_{A_{cell}} \Phi_{\text{int}}^{\dagger}(x, y, \Omega_m) R_{\text{h}}(x, y, \Omega_m) dA \\ & + \int_{y_s}^{y_n} \Phi_{\text{int}}^{\dagger}(x, y, \Omega_m) r_{x_{in}}(x, \Omega_m) dy + \int_{x_w}^{x_e} \Phi_{\text{int}}^{\dagger}(x, y, \Omega_m) r_{y_{in}}(y, \Omega_m) dx \Bigg), \end{aligned} \tag{6.39}$$

where "int" stands for internal cell value, the subscript "in" stands for the value at the incoming boundary and:

$$R_{\text{h}}(x, y, \Omega_m) = q - \mathcal{L}\Phi_h, \tag{6.40}$$

$$r_{x_{in}}(x, y, \Omega_m) = \mu_m \left(\Phi_h(x_{\text{in}} - 0, y, \Omega_m) - \Phi_h(x_{\text{in}} + 0, y, \Omega_m) \right), \tag{6.41}$$

$$r_{y_{in}}(x, y, \Omega_m) = \eta_m \left(\Phi_h(x, y_{\text{in}} - 0, \Omega_m) - \Phi_h(x, y_{\text{in}} + 0, \Omega_m) \right). \tag{6.42}$$

The boundary term arises from the discontinuity and the weak application of boundary conditions. Instead of integrating over a cell, the integration is done over a cell that is shifted by a very small amount ξ in the upwind direction, and the answer as $\xi \to 0$ is computed [143]. If the variation within the cell is approximated via a bilinear variation within the cell and the approximation to the flux is globally continuous, the boundary terms in equation (6.39) go to zero.

The total error in the domain is obtained by summing the values of (6.39) over all cells in the domain. If AMR was to be conducted, an indicator for the cell is found by taking the absolute value of (6.39).

6.6 Application to test cases

This section applies the methods described in the previous sections of this chapter to the various 2-D test cases described in appendix A.2. The reference solutions for calculating the true error and effectivity indices were provided by the RADIANT-II code [2].

In order to improve the error estimate in the box method case without adding too much complication to the code, there is the option of calculating the adjoint solution on a more refined mesh than the forward solution. The more refined solution is then injected into the phase space of the less refined forward solution. This was not conducted for linear DD since how to conduct the injection into the phase space without introducing more errors is not obvious. In the rest of this section *ncr* represents the ratio of the number of cells along one side of the domain for the adjoint solution compared to that used in the mesh for the forward solution.

6.6.1 Test case 7

Test case 7 comprises all of the different variations on test case 7 given in appendix A.2.1. The results for the two different goal values (QoI) considered are discussed in section 6.6.1.1 and section 6.6.1.2 respectively.

6.6.1.1 Test case 7a

Figure 6.2a gives the effectivity index of the 4 different DWR estimators at the 6th mesh iteration. Figure 6.2b is the same but with the DD plot removed for clarity. Figure 6.3 plots the convergence graphs of the true error and all error estimators. The true error resulting from the box method code and DD code are plotted and are exactly the same due to the equivalence between the two methods for integral quantities.

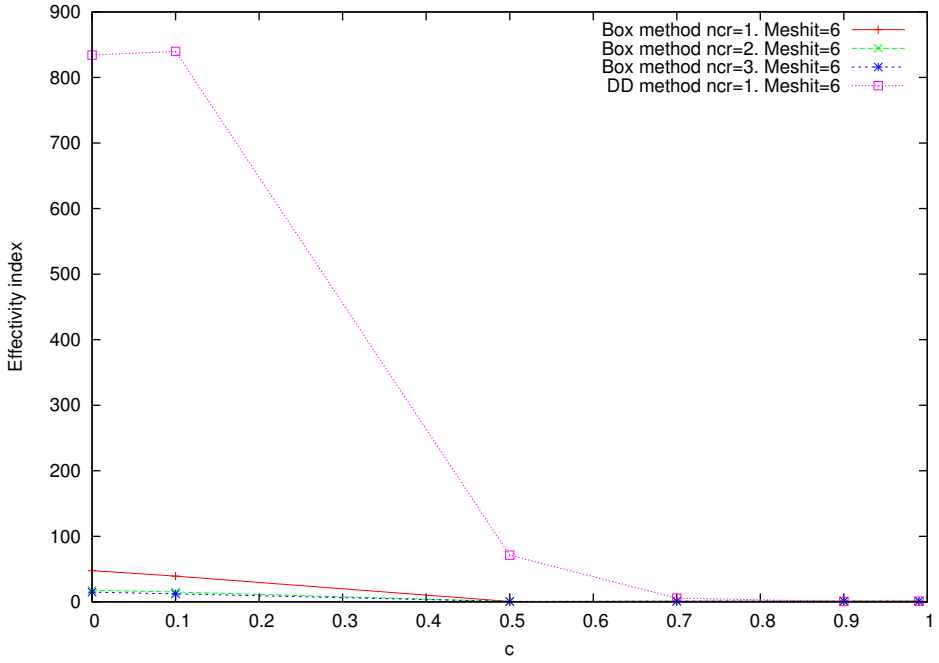
Figure 6.2a shows that all error estimators give an effectivity index near unity for high values of c. The estimator given by the DD method described in section 6.5.2 gives the worst result for

low values of c. Figure 6.3 shows that, for low values of c, the effectivity index given by the DD DWR gets worse as the mesh becomes more refined. This is due to the convergence plot of the DWR error estimate in the DD case having a different slope from that of the reference error. The exception is when $c = 0.99$ (figure 6.3f), where the DD estimator tends to the correct slope and value.

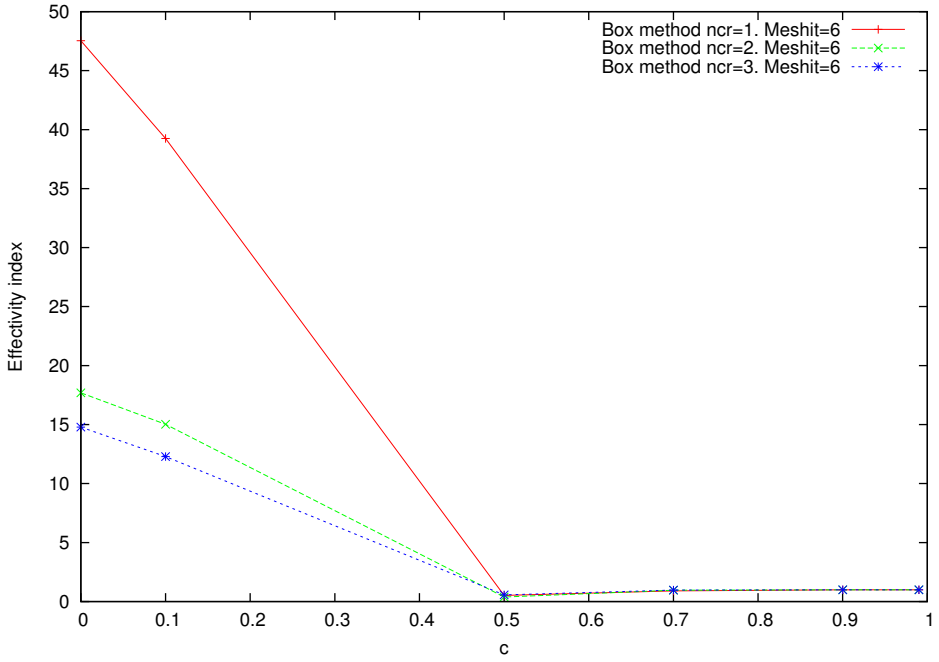
Figure 6.2b shows that the effectivity index tends to unity as c is increased for all box method error estimators. The effectivity indices at low c, although all far from unity, are less incorrect when the error estimator uses a more accurate adjoint solution. The effectivity index reduces from 47 when $ncr = 1$ to 18 when $ncr = 2$, then reducing further to 15 when $ncr = 3$. The increase in the accuracy of the error estimate with an increased value of ncr is limited given that the more accurate result is mapped into the same space as the forward mesh for simplicity. Thus, although an accurate solution to the adjoint solution may be found on a highly refined mesh, some error is re-introduced by mapping it into the low order space.

All box method DWR estimators have the same slope as the true error in figure 6.3, with the exception of $c = 0.5$ (for all mesh iterations) and $c = 0.7$ (for the first few mesh iterations). This suggests that the reason for the DD DWR estimator's incorrect slope is due to the bilinear term being artificially set to zero. The box method representation conserves the bilinear term resulting in a more accurate error estimate.

The box method DWR estimator (which conserves the bilinear term), although having the correct slope, does not coincide with the graph of the true error at low values of c. The approximation to the flux for low values of c is oscillatory due to the lack of regularity in the analytical solution. For example, when $c = 0.0$, the discontinuity in the gradient of the flux is not aligned with cell boundaries and thus the solution will be badly represented by the box/DD method. The discontinuous behaviour reduces as c increases. Thus, the oscillations observed in the approximate solution reduce in magnitude as c increases or as the number of cells in the mesh is increased. The goal quantity is an integral value, thus as long as the solution is oscillating around the correct value of the flux, cancellation of errors in integration will result in only a small contribution from solution oscillation to the final error in the integral functional. However, this is not the case for the error estimator. Oscillations are observed in both the forward and adjoint flux approximations, thus, two oscillating functions are multiplied together before being integrated in the DWR calculation. Because of this, the cancellation of errors observed in the approximation to the goal functional does not occur for the DWR estimate, thus explaining the



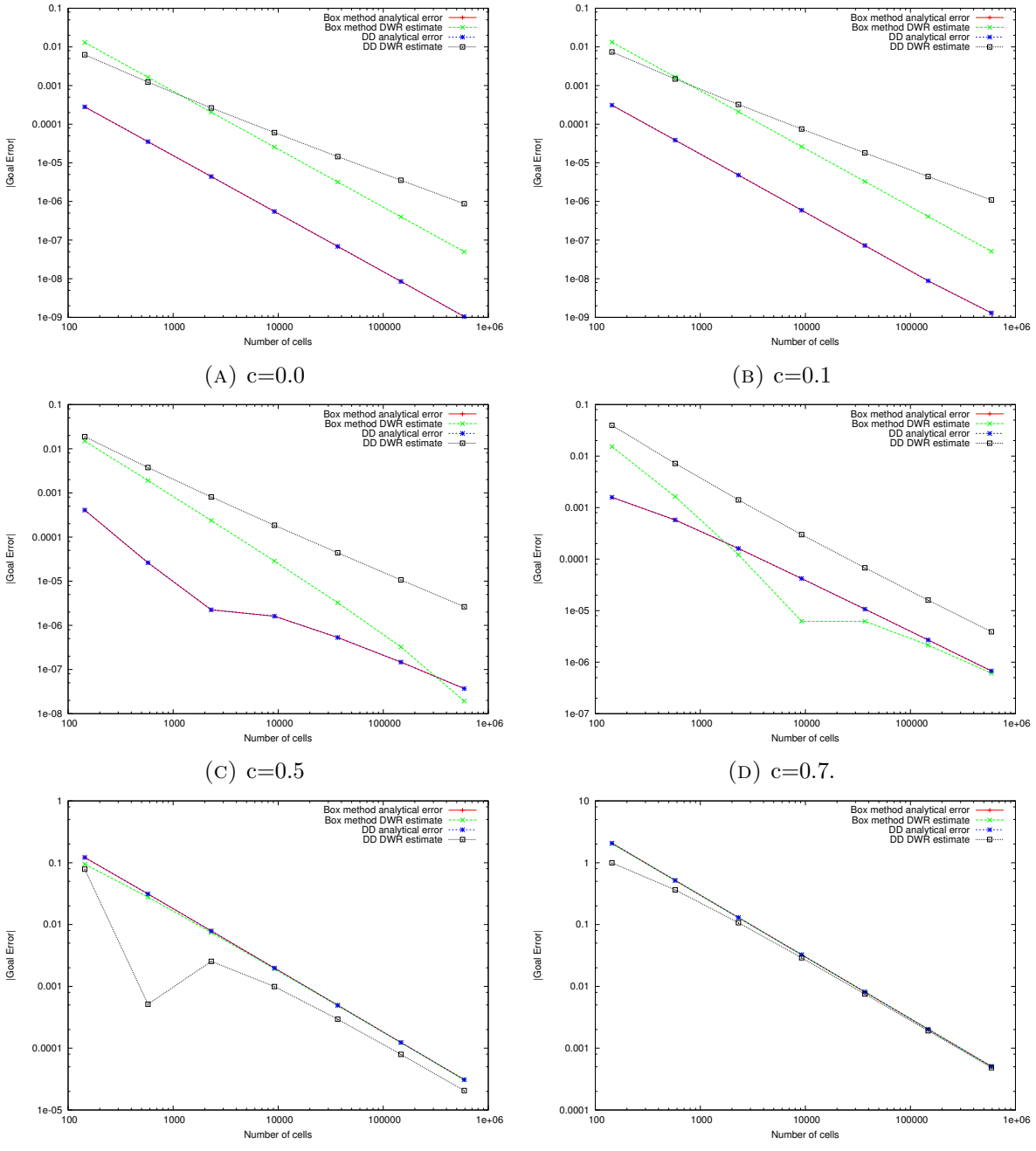
(A) Box methods and DD method



(B) Box method only

FIGURE 6.2: Test case 7a. Effectivity index as a function of the scatter ratio for a range of different error indicators and mesh iterations.

overestimation of the error by the DWR error estimator. The improvement of the effectivity index when using a higher order result for the adjoint solution is mainly due to the reduction observed in the oscillations.



(A) c=0.0 (B) c=0.1

(C) c=0.5 (D) c=0.7.

(E) c=0.9 (F) c=0.99

FIGURE 6.3: Test case 7a. Convergence graphs.

6.6.1.2 Test case 7b

All error estimators give an effectivity index close to unity (figure 6.4a), even the DD DWR error estimator. Though, the estimators that conserve the bilinear term consistently perform better. Figure 6.4b plots only the box method DWR error estimators, and it is observed that the effectivity index improves when a more accurate approximation to the adjoint solution is used. Although the effectivity index is close to unity for all values of c, an improvement is observed as c is increased. The convergence graphs in figure 6.5 also show how all error estimates well-represent the true error for all values of c.

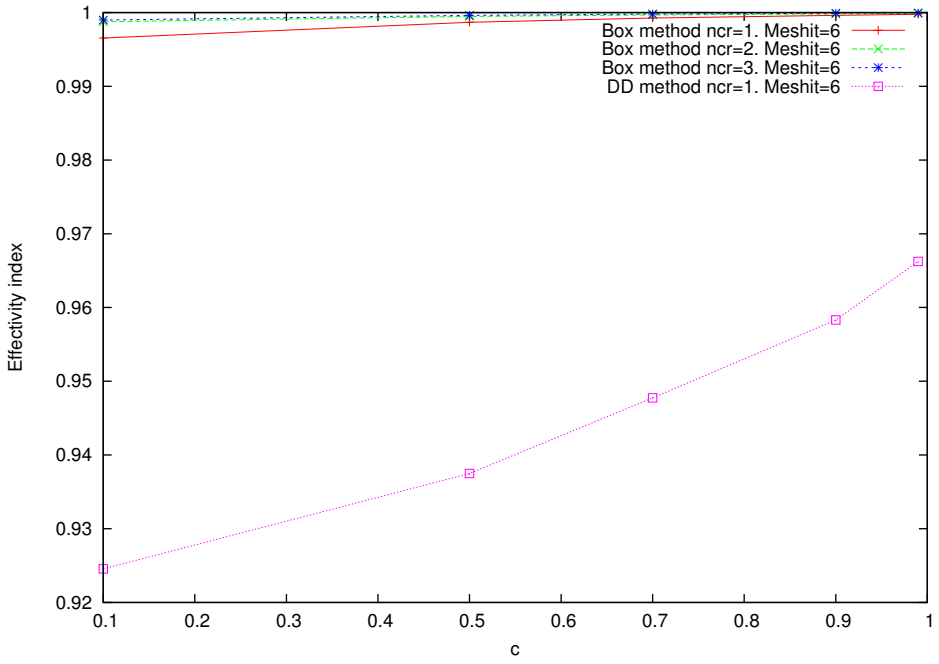
6.6.2 Test case 8a

The convergence plots of the error and error estimators as a function of the number of cells in the forward mesh are plotted in figure 6.6a. Similarly, the effectivity indices as a function of cell number are plotted in figure 6.6b. The effectivity index plots of only the box method DWR error estimators are re-plotted in figure 6.6c for clarity.

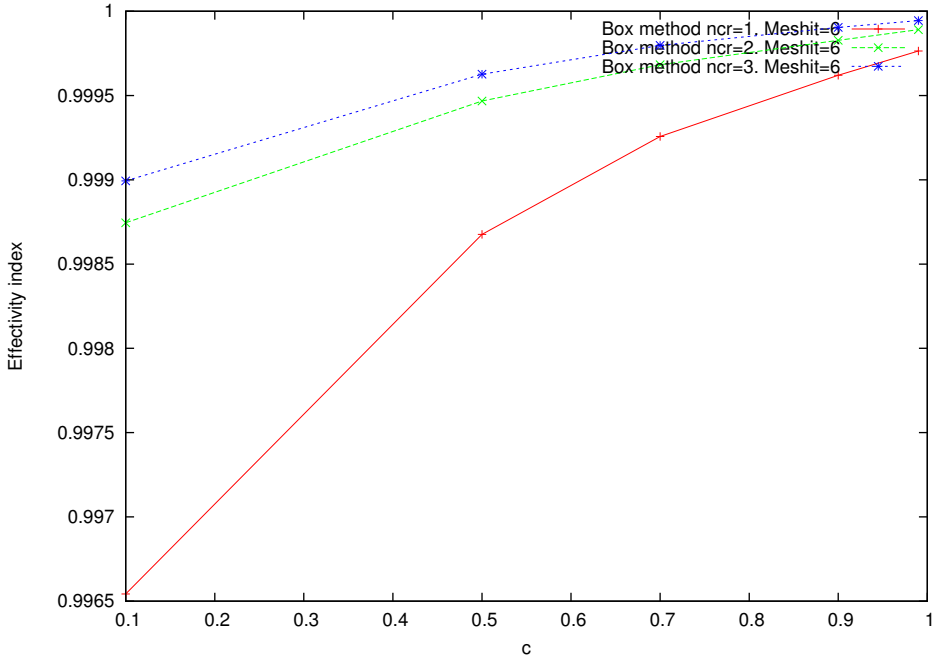
The graphs in figure 6.6 reinforce the conclusions drawn from the homogeneous test cases in section 6.6.1.1: better results are given by the box method error estimators than the DD method and the more accurate the adjoint solution, the closer the effectivity index is to unity. The divergence away from unity at the highest resolution mesh in figure 6.6c is due to reference solution used in the effectivity index being calculated by using a highly refined numerical solution to the forward equation and not an analytical reference. The numerical reference solution was obtained using the RADIANT-II code with quadrilateral finite elements [2]. Reference results were also obtained using other types of elements in the RADIANT-II code. The resulting variation in the effectivity index results were of the order of the deviation shown in figure 6.6c.

6.6.3 Test case 8b

The results for this test case are shown in figure 6.7. There is a big deviation between the DD error estimator and the true error (figure 6.7b), though the slope is correct in figure 6.7a. The first and second points of all box method plots in figure 6.7c seem to be tending towards unity before diverging again. The divergence is because of the limited accuracy of the reference solution, the accuracy is less for test case 8a (section 6.6.2) due to the more localised goal

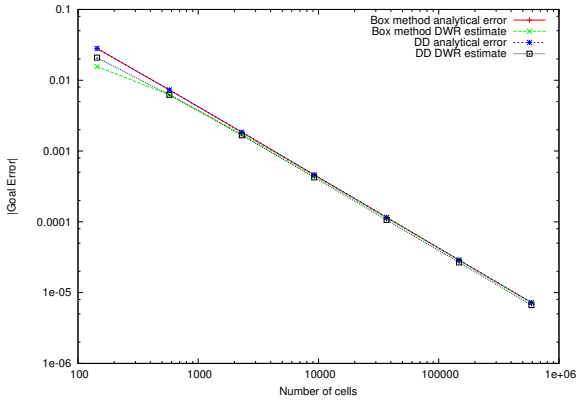


(A) Box methods and DD method.

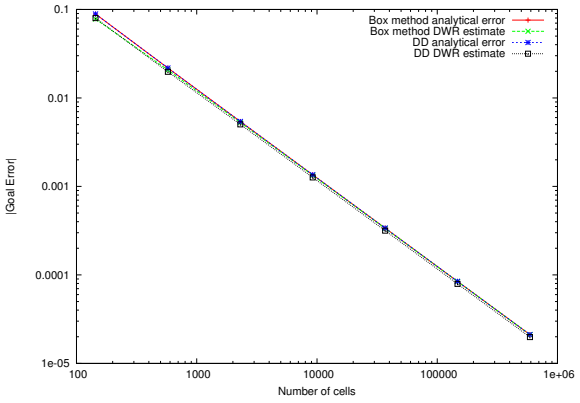


(B) Box method only.

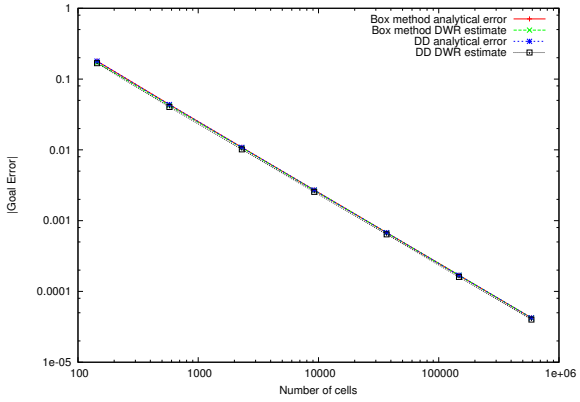
FIGURE 6.4: Test case 7b. Effectivity index as a function of the scatter ratio for a range of different error indicators and mesh iterations.



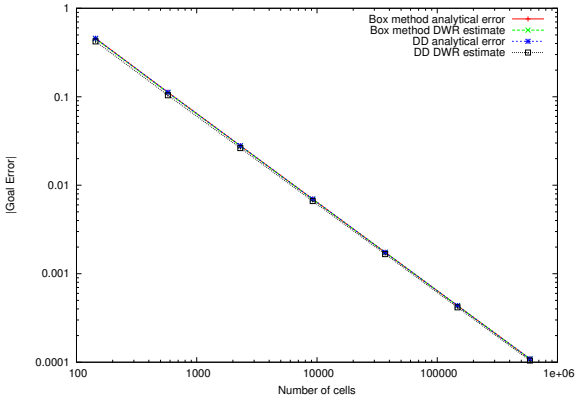
(A) c=0.1



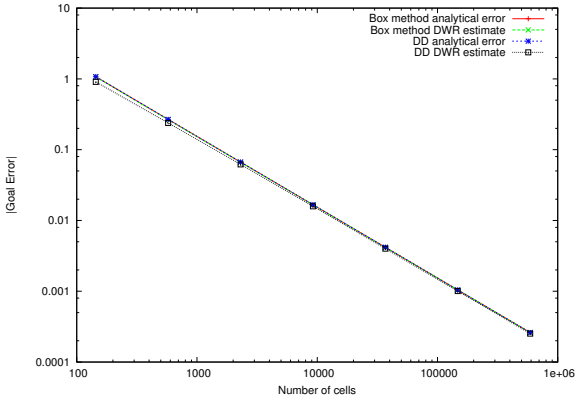
(B) c=0.5



(C) c=0.7.

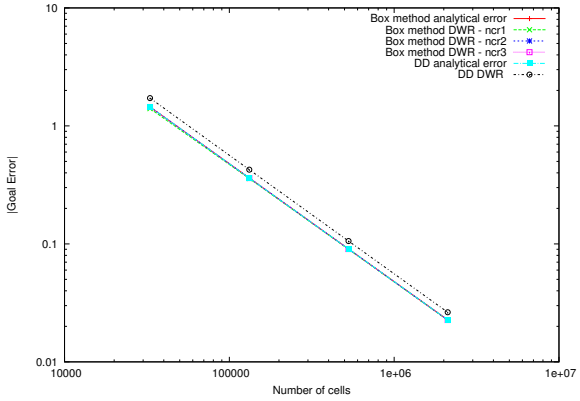


(D) c=0.9

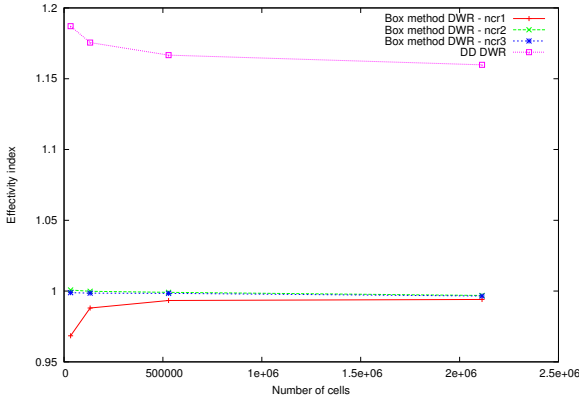


(E) c=0.99

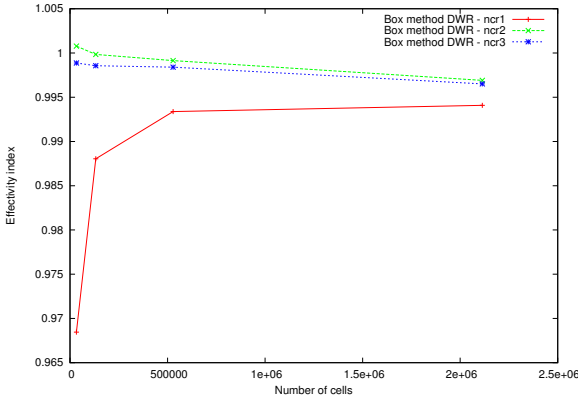
FIGURE 6.5: Test case 7b. Convergence graphs.



(A) Convergence



(B) Effectivity index



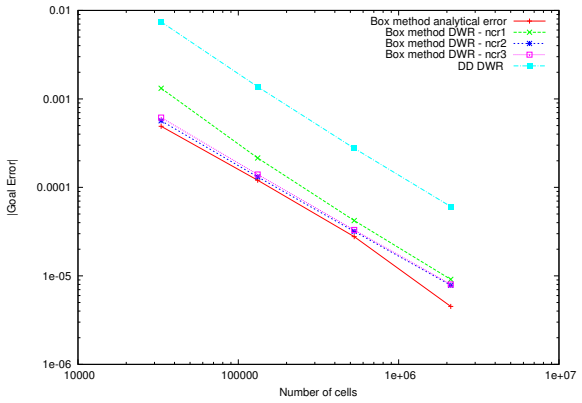
(C) Effectivity Index. Box method only.

FIGURE 6.6: Test case 8a results.

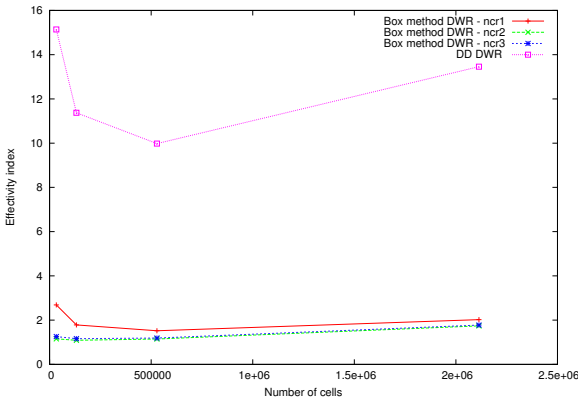
quantity. Increasing the accuracy of the adjoint solution results in a better error estimator. In figure 6.7c the effectivity index results for $ncr = 2$ are closer to one than when $ncr = 3$. This is due to the way the map into the forward mesh's space can distort the results.

6.6.4 Test case 9

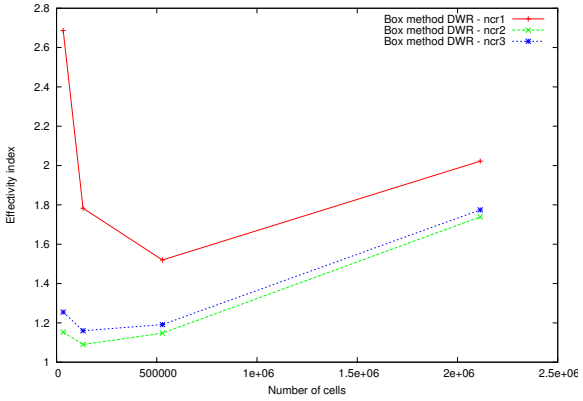
Although the slopes of all error estimators are the same as the reference error (figure 6.8a), they all significantly under-estimate the error (by a factor of the order of 50 - figure 6.8b). This test case has a lot of leakage, and thus suffers from some of the oscillation issues that gave bad results in test case 7a (section 6.6.1.1). This reasoning for the underestimation is supported slightly by the small increase in the effectivity index when a more accurate adjoint solution is used for calculating DWR estimator. The large underestimation could also be due to the linearisation error not being negligible and not decreasing at the same rate as the leading order error term measured by the estimator, thus leading to a convergence to a constant effectivity index that is not unity.



(A) Convergence

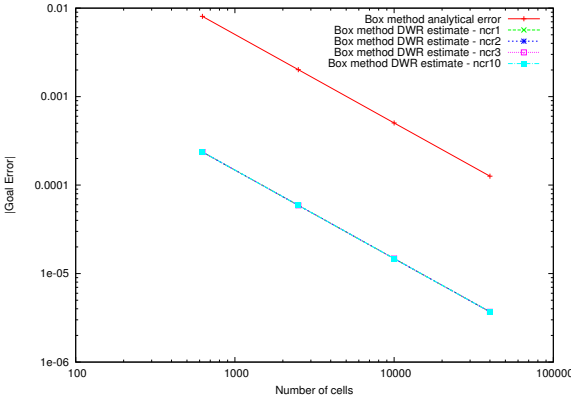


(B) Effectivity index

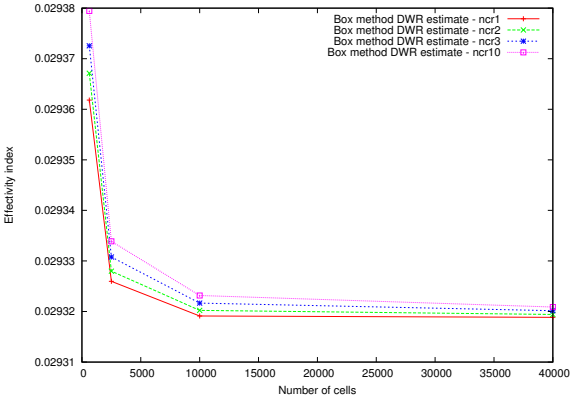


(C) Effectivity Index. Box method only.

FIGURE 6.7: Test case 8b results.

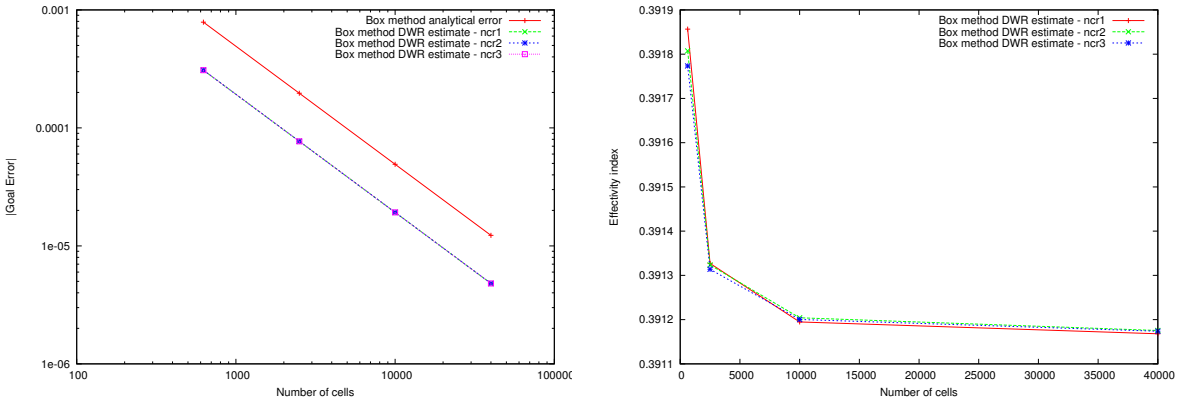


(A) Convergence



(B) Effectivity index

FIGURE 6.8: Test case 9 results.



(A) Convergence (B) Effectivity index

FIGURE 6.9: Test case 10 results.

6.6.5 Test case 10

Figure 6.9 shows the results for test case 10, which has less leakage than test case 9. Similar comments can be made for the results given for this test case as for test case 9 (section 6.6.4). However, in this case the underestimation of the error is less severe (less than a factor of 3). There is also not much difference between the error estimate given by DWRs computed with adjoint solutions calculated on more refined meshes. This suggests that the remaining error is due to the non-linear terms.

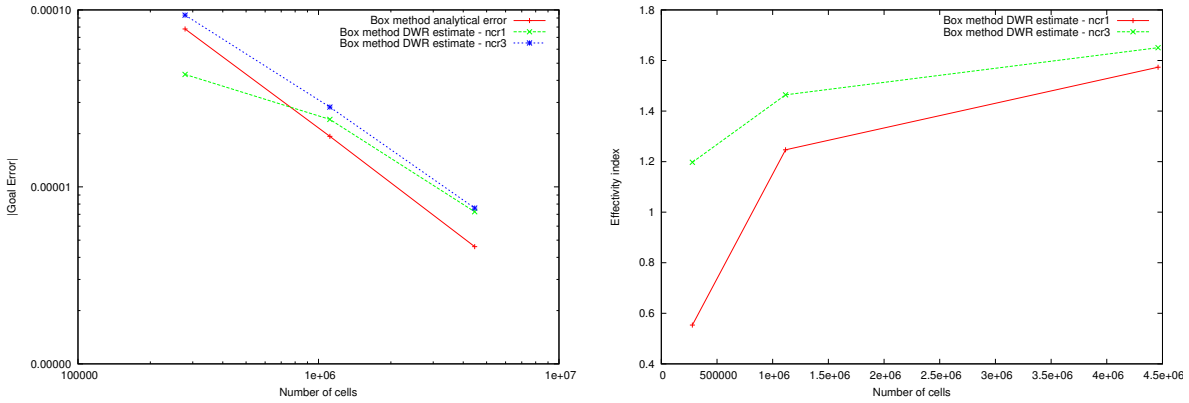
6.6.6 Test case 11

This test case is bigger, with less leakage and has a high scatter ratio throughout, thus the results are improved compared to the other eigenvalue test cases (sections 6.6.4 and 6.6.5). The size of the problem means that the accuracy of results of the RADIANT-II code may be limited to some extent. However, the results given by the DWR error estimators give results that are no more than a factor of 2 away from the reference error (see figure 6.10).

6.7 Comment on 3-D DWR

The arguments in this chapter have been presented for 2-D problems and applied to 2-D test cases. Similar arguments can be made for the 3-D test case.

The equivalent to the box method in 2-D can be found in the 3-D case by taking cells to be simply connected rectangular prisms and keeping the unknowns as the flux value at each vertex



(A) Convergence (B) Effectivity index

FIGURE 6.10: Test case 11 results.

of the prism. A result in H^1 that is uniquely determined by cell vertex values results in a trilinear variation of the flux within the cell. This results in 8 parameters per cell, 7 of which are provided by the continuity condition between cells or by external boundary conditions (set by the fact that a solution in H^1 is sought). The remaining parameter is determined by a WR equation where the test functions are taken from a subset of L^2: piece-wise constant function over the cell. Using the node numbering shown in figure 6.11, a trial function can be associated with each vertex in the cell:

$$N_1 = \frac{1}{8}(1-\zeta)(1-\gamma)(1-\omega) \tag{6.43}$$

$$N_2 = \frac{1}{8}(1+\zeta)(1-\gamma)(1-\omega) \tag{6.44}$$

$$N_3 = \frac{1}{8}(1+\zeta)(1+\gamma)(1-\omega) \tag{6.45}$$

$$N_4 = \frac{1}{8}(1-\zeta)(1+\gamma)(1-\omega) \tag{6.46}$$

$$N_5 = \frac{1}{8}(1-\zeta)(1-\gamma)(1+\omega) \tag{6.47}$$

$$N_6 = \frac{1}{8}(1+\zeta)(1-\gamma)(1+\omega) \tag{6.48}$$

$$N_7 = \frac{1}{8}(1+\zeta)(1+\gamma)(1+\omega) \tag{6.49}$$

$$N_8 = \frac{1}{8}(1-\zeta)(1+\gamma)(1+\omega) \tag{6.50}$$

where:

$$\zeta = \frac{2x - x_w - x_e}{x_e - x_w}, \quad \gamma = \frac{2y - y_s - y_n}{y_n - y_s}, \quad \omega = \frac{2z - z_b - z_f}{z_f - z_b}. \tag{6.51}$$

Where w, e, n, s, represent the west, east, north and south faces (as in 2-D, but now the face

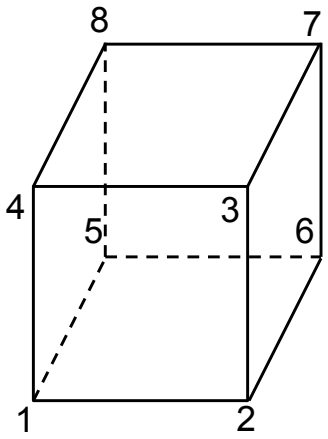


FIGURE 6.11: Node numberings for a rectangular prism

is a rectangular area rather than a straight line). f and b represent the faces that are in the z direction towards the inside of the page and towards the outside of the page respectively.

The function within a cell can be represented in 2 ways, as in the 1-D and 2-D cases:

$$\Phi(x,y)_{\text{cell}} = \Phi(\zeta,\gamma) = \Phi_1 N_1 + \Phi_2 N_2 + \Phi_3 N_3 + \Phi_4 N_4 + \Phi_5 N_5 + \Phi_6 N_6 + \Phi_7 N_7 + \Phi_8 N_8 \quad (6.52)$$

and

$$\Phi(x,y)_{\text{cell}} = \Phi(\zeta,\gamma) = c_1 + c_2\zeta + c_3\gamma + c_4\omega + c_5\zeta\gamma + c_6\zeta\omega + c_7\omega\gamma + c_8\zeta\gamma\omega. \quad (6.53)$$

The WR equation obtained is as follows:

$$\frac{\mu}{4\Delta x_{ijk}}(\Phi_2 + \Phi_3 + \Phi_6 + \Phi_7 - \Phi_1 - \Phi_4 - \Phi_5 - \Phi_8) \quad (6.54)$$

$$+ \frac{\eta}{4\Delta y_{ijk}}(\Phi_3 + \Phi_4 + \Phi_7 + \Phi_8 - \Phi_1 - \Phi_2 - \Phi_5 - \Phi_6) \quad (6.55)$$

$$+ \frac{\xi}{4\Delta z_{ijk}}(\Phi_5 + \Phi_6 + \Phi_7 + \Phi_8 - \Phi_1 - \Phi_2 - \Phi_3 - \Phi_4) \quad (6.56)$$

$$+ \frac{\Sigma_{t,ijk}}{8}\sum_{v=1}^{8}\Phi_v = q_{ijk} \quad (6.57)$$

Grouping appropriate terms together gives the traditional representation of the DD method in 3-D:

$$\frac{\mu}{\Delta x_{ijk}}(\Phi_e - \Phi_w) + \frac{\eta}{\Delta y_{ijk}}(\Phi_n - \Phi_s) + \frac{\xi}{\Delta z_{ijk}}(\Phi_f - \Phi_b) + \Sigma_{t,ijk}\Phi_c = q_{ijk} \quad (6.58)$$

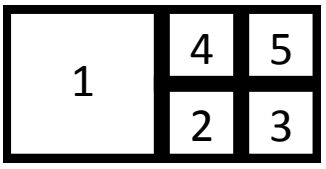


FIGURE 6.12: 2-D hanging node example

However, from these unknowns, the final 4 terms of the within cell flux approximation in equation (6.53) cannot be reconstructed. This means that the gap between the answer of the vertex method DWR and the version where the solution is mapped into a discontinuous FE space with the unknown terms artificially set to zero will be even worse in 3-D compared to 2-D. This is further motivation for calculating the vertex values as unknowns in industrial codes, rather than the cell-averaged and cell face-averaged fluxes.

6.8 Comment on multi-dimension AMR

As previously mentioned (section 2.3), h AMR has been applied to the DD scheme [44, 63]. However, extending the current work to provide multi-dimensional h AMR for the DD scheme driven by DWR error indicators would not be trivial. The WR theory behind the box method (and the equivalent DD method) breaks down with the addition of hanging nodes.

Figure 6.12 shows a hanging node between cell 1 and its two neighbouring cells (cells 2 and 4). For an ordinate that has a positive component going from left to right, the hanging node value could simply take the value given by the linear variation along the eastern cell edge of cell 1. However, the situation is not as clearcut if the component of the flux is going in the opposite direction (from a more refined region to a less refined region). Unless the hanging node happens to take a value that is exactly half the value of the nodes either side in the vertical direction, the condition of continuity along the line will not be satisfied. This means that the approximation will not exist in a subset of $(H_b^1)^{\frac{N(N+2)}{2}}$ and thus the WR theory behind the box method will break down.

One could allow for discontinuity in the solution, thereby making the calculation of the DWR similar to the case where we assume the bilinear term is zero (section 6.5.2). However, the equivalence between the cell-averaged values of the resulting solution and a WR scheme is lost, thereby making the DWR theory applied to the result more questionable than when the bilinear term was simply ignored (in section 6.5.2). Even if the vertex values are the DoF being solved

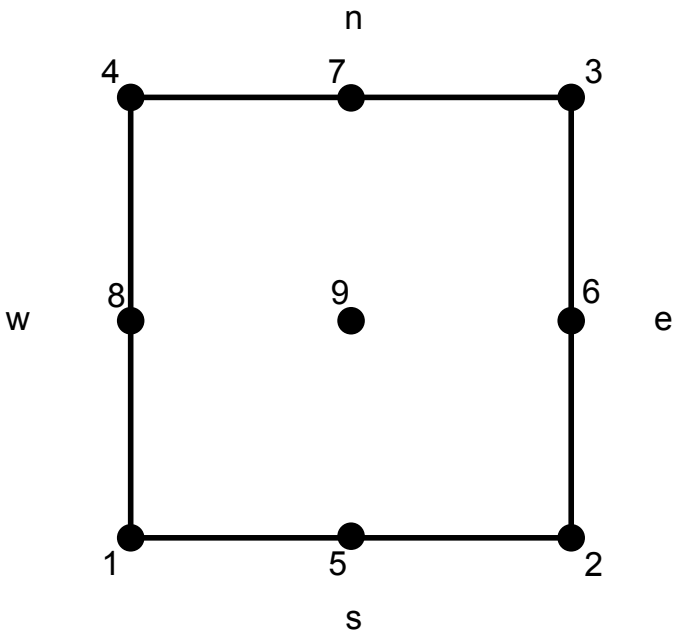


FIGURE 6.13: Bi-quadratic cell

for, flux values for the two right hand nodes of cell 1 (of figure 6.12) cannot be allocated in a unique manner. One could simply take the values at the nodes for the previous cells in the sweep and ignore the result on the hanging node. However, this would result in a lack of conservation of neutrons. One could ensure conservation of neutrons, but the issue of discontinuity and the lack of equivalence with a WR scheme remains.

6.9 Comment on multi-dimensional higher order DD

Hébert's method for higher order 1-D DD was extended to 2-D [57] and 3-D [58]. In 1-D, Hébert's method was found to be equivalent to the WR Hennart method, with the highest order variation being ignored in the reconstruction of the flux in the spatial domain (see section 3.2.2.2).

An equivalence between the box and traditional DD method has been shown in section 6.4.1. The first 3 Legendre polynomial coefficients can be calculated from the traditional DD unknowns, but the fourth (the bilinear term) cannot. Another way of viewing the box method is as a WR method where the trial functions are given by a first order Lagrange [146] interpolation within a cell and the weighting functions are zeroth order Legendre polynomials. The ability to rearrange the coefficients of an Mth order Lagrange polynomial to give an Mth order Legendre polynomial is what made analytical integration possible for the calculation of the DWR error estimates.

An obvious way to extend the box method to higher orders would be to use higher order Lagrange polynomials for the trial functions and higher order Legendre polynomials for the weighting functions. The approximation would be assumed continuous at all orders, as in the higher order 1-D DD methods described by Hennart [56].

In 2-D, an Mth order Lagrange polynomial used as trial functions would be associated with test functions defined as the basis of a $(M-1)$th order Legendre polynomial phase space. This would result in the solution within a cell being represented uniquely by $(M+1)^2$ nodes (see figure 6.13 for the case where $M = 2$). The solution on $(2M + 1)$ nodes would be known from boundary conditions or the results given by previous cells in the sweep. The remaining M^2 unknowns would be solved for by the M^2 WR equations that result from weighting by the basis functions of a $(M - 1)$th order Legendre polynomial function. The resulting flux approximation will be represented as an Mth order 2-D Lagrange polynomial which can be rearranged and expressed as an Mth order 2-D Legendre polynomial. For the higher order box and Hébert methods to be equivalent in the same way as the traditional box and DD methods, the first $M(M+2)$ Legendre polynomial coefficients of the box method (calculated by rearranging the Lagrange polynomial representation) must be equivalent to those that can be calculated from the Hébert method's unknowns. The final Legendre polynomial coefficient provided by the Mth order box method will not be able to be calculated from the Hébert method due to the unknowns being cell-centred and cell-edge spatial moments that average out this term.

The WR equations in Hébert's paper (equations (38), (39), (41) and (42) in Hébert's paper [57]) are analytical before the auxiliary equations are substituted in for the higher order terms. Showing that the Mth order approximate function of the box scheme satisfies the auxiliary equations of a $M-1$th order Hébert DD scheme is sufficient to show that the first M^2 Legendre polynomial coefficients of the two schemes are equivalent. The next $2M$ Legendre polynomial coefficients can then be calculated from known cell edge and cell central spatial moments of the flux (in an analogous way to the linear in x and y terms for the lowest order DD method). The final Legendre polynomial cannot be calculated as a function of the cell centre and cell edge moments as this information is lost by taking integral values.

The above equivalence has been demonstrated for the bi-quadratic scheme in appendix D. Numerical analysis of this equivalence for higher orders is beyond the scope of this thesis.

A comment may be made about the practicality of calculating the DWR for the higher order box method. Rearranging the Lagrange polynomial coefficients to give the equivalent Legendre

polynomial coefficients gets more cumbersome as the order of the polynomials increases. If a higher order box method was to be considered in the future, numerical integration should be considered for calculation of the DWR integrals.

6.10 Closing comments for this chapter

This chapter extended the 1-D WR view of the DD equation to 2-D to give the *box method* where the vertices of the cells are kept as unknowns. These unknowns can be collected together to give the traditional 2-D DD equations. The WR view of the DD equations in 2-D assumes a solution in H^1 with bilinear variation of the flux approximation within the cell. When the unknowns solved for are the cell-averaged and edge-averaged fluxes (such as in traditional implementations of DD codes) however, it is impossible to re-construct the bilinear term without taking the boundary condition and sweeping through the whole domain again. It is for this reason that 2 codes were written, one implementing the traditional box method and the other implementing the traditional DD equations. In the former, the DWR was calculated using the resulting piece-wise bilinear flux distribution that is uniquely determined by the cell vertex flux values. In the latter, the result was mapped into a bilinear DGFEM space where the bilinear term is artificially set to zero.

The DWR error estimates calculated using the vertex unknowns gave very good results for fixed source high c problems. This was due to the increased stability and the lack of oscillations in the result when c is large. PWRs, for which EDF use the DOMINO code, operate with large c values, thus suggesting that the *box method* DWR estimates are appropriate for industrial cases.

The results for eigenvalue problems were less successful than for the fixed source case. Though, in the more realistic example that had reduced leakage and high scatter ratios, the effectivity index of the box method was within a factor of 2 of unity. An opportunity for future work would be to investigate ways of estimating the magnitude of the error terms not taken into account in the DWR for non-linear problems. Such work could investigate whether the results observed for the eigenvalue test cases in this chapter could be improved.

Given the equivalence of the cell-averaged fluxes of the box and DD methods, EDF's $\mathrm{SP_N}$ solver can be applied to the box method and give the same stability results. The advantage of using the vertex values as unknowns instead of cell- and edge-averaged fluxes is that the quality of the error estimate is consistently improved. This would be a recommended change to apply to the

DOMINO code if EDF decide to pursue DWR error estimation for spatial discretisation error. Assuming nc_x denotes the number of cells in the x direction and nc_y denotes the number of cells in the y direction, the number of unknowns per ordinate would be equal to $nc_x nc_y + nc_x + nc_y$ if vertex values were stored, as opposed to $3nc_x nc_y + nc_x + nc_y$ if cell centre and cell edge values were stored.

A comment on how these methods can be extended to 3-D test cases was discussed. The theory shows that the advantages of solving for the vertex values in 3-D could be more significant than in 2-D.

Applying DWR driven AMR in 2-D would be more challenging than in the 1-D case due to the presence of the hanging nodes. Some of the theoretical challenges were discussed in this chapter. A possible higher order extension of the box method was suggested and its equivalence to the higher order Hébert method demonstrated mathematically.

7. Conclusions

This chapter summarises the conclusions of the thesis and provides suggestions for further work.

7.1 Chapter summaries

The four main technical chapters of the thesis are summarised in the following sections.

7.1.1 DWR error estimation and functional correction in 1-D slab geometry

This chapter showed that DWR error estimators could be applied to the 1-D DD-S_N discretisation of both the fixed source and eigenvalue neutron transport equations. Viewing the DD scheme as a WR scheme showed that the test functions were in a different function space to the trial functions. Because of this, unlike for standard Bubnov-Galerkin FEM schemes, weighting the residual by an adjoint solution taken from the same space as the forward solution does not give an estimate of zero. In many cases this gave a good estimate of the goal error, but an improved adjoint solution was required for cases that do not satisfy the negative flux criterion.

The heterogeneous eigenvalue case gave an effectivity index that tended to a constant value above unity. It is assumed that this was due to the DWR estimator not taking into account the lower order error terms that exist for the eigenvalue problem.

Improvements in the accuracy of the QoI were observed when the DWR estimate was used to correct the approximation of the QoI. For uniform h refinement, super-convergence of the integral goal quantities was observed after error correction.

7.1.2 DWR driven h, p or hp AMR in 1-D slab geometry

The cell values that are summed to give the DWR error estimate in chapter 3 were used as error indicators for goal-based AMR in chapter 4. Due to using Hennart's high order extension to the DD equations, both h and p AMR was possible. The *maximum strategy* was used and it was possible to obtain a reduction in error per DoF for heterogeneous test cases and for homogeneous cases with localised goals. For most test cases, there was no benefit to using a more refined/higher order mesh to obtain the adjoint solution used in the DWR estimator. There was a benefit observed for the test case where the QoI is located in an area that is very challenging for the DD method.

The ability to have a non-uniform mesh cell size and cell-wise polynomial order throughout the mesh also allowed for hp AMR. The error indicator does not give sufficient information to choose between h or p refinement for a given cell, thus a merit function was derived. The merit function of both refinement options was calculated for each cell already flagged for refinement by the DWR indicator. The option with the highest merit function was chosen in generating the mesh in the next iteration. hp AMR was shown to give the best results.

Some different starting meshes and values of α were tested for some of the 1-D test cases. It was found that the convergence graphs became very oscillatory when α was too large. For some test cases the starting mesh had an impact on the form of the convergence graphs.

7.1.3 AMR and functional correction for linear functionals in 1-D slab geometry

The advantages of both AMR and functional correction were combined in this chapter. For this, a different error indicator was used that approximated the remaining error after DWR error correction of the functional. It was predicted that this would result in the optimal mesh for the error corrected quantity and this was shown to be the case.

The work of this chapter was only applied to the fixed source problems since the remaining error after the DWR error correction is less obviously approximated for eigenvalue problems.

7.1.4 Multidimensional DWR error estimation

The extension of the WR view of the 1-D DD equations to 2-D requires the vertex values of the flux to be kept as unknowns (resulting in a method that has, in previous work, been referred to as the *box method* [59]). The traditional view of the 2-D DD method however, solves for the cell averaged fluxes and cell-edge averaged fluxes as unknowns. These equations can be obtained from the WR view of the 2-D DD equations by gathering together vertex values. However, it is impossible to reconstruct the bilinear term in the within-cell flux representation from these unknowns without sweeping through the cells again and enforcing point-wise boundary conditions and continuity along internal cell boundaries. Thus two codes were written:

- a box method code that kept the vertex values, allowing for full representation of the flux solution;
- a DD code where the solution was mapped into a discontinuous FEM space and the bilinear term artificially set to zero for each cell. This flux distribution was then used when calculating the DWR.

The box method DWR estimators consistently outperformed the DD DWR estimators. Overestimation of the error in fixed source test cases with low c was observed. This was attributed to the oscillations that appear in the flux approximations given by the DD and box methods when there is a lack of regularity in the analytical solution being approximated.

The results for the eigenvalue test cases also indicated a need for high scatter ratios and low leakage. The non-linear terms in the goal error that are not taken into account in the DWR estimation are also thought to play a part in the effectivity index converging to a value that is not unity.

It was shown that the box method gives the same cell-averaged and edge-averaged fluxes as the traditional DD method. This has two consequences. Firstly, any instabilities in the box method that are obvious from the WR derivation will also apply to the DD method. This is why the DD method does not give good convergence results for problems that have solutions that are less regular than H^1, or where discontinuities in gradients are not aligned along mesh cell edges. Secondly, the equivalence has an industrial consequence for EDF. The equivalence means that the same arguments that motivated the use of the $\mathrm{SP_N}$ solver for DSA of the DD code can also be applied to the box method. Provided EDF wants to pursue DWR error estimation in the

DOMINO code, it would be advisable to change the unknowns solved for in the DOMINO solver to be vertex values rather than the traditional DD unknowns. Doing this would allow for the advantages of a more reliable error estimate without the need for a new acceleration scheme.

A comment on the possible 3-D implementation of the work was given. The theory shows that the advantages of solving for the vertex values instead of cell-averaged and face-averaged fluxes exist also for the 3-D case.

Extending the current work to provide DWR driven AMR in 2-D would be challenging. A brief discussion of why the presence of a hanging node makes the theory break down was given.

A possible extension of the box method to higher order polynomials was suggested and shown to be mathematically equivalent to the higher order Hébert method. The only difference being that the unknowns solved for in a Mth order Hébert method do not allow for the reconstruction of the $\Phi_v^{(M+1)(M+1)}$ moment in space, whereas the higher order box method does.

7.2 Further work

In both the 1-D and 2-D eigenvalue problems, although the slope of the error estimator's convergence graph was the same as the slope of the true error, the effectivity index tended towards a value different from unity. This is potentially due to the extra error terms that result due to the non-linearity of the problem. These are not taken into account by the DWR estimate and could be converging to zero at different rates from the term that's considered in the DWR estimate. Investigating ways of approximating the other error terms could lead to a way of testing this hypothesis. It would also allow a way of finding the dominating error term after error correction of the functional so that a more appropriate error estimate for AMR coupled with error correction could be derived for eigenvalue problems as was done for fixed source problems in chapter 5.

The AMR and error correction techniques applied in 1-D could be applied in 2-D. AMR in multi-dimensions is more complex since it would result in hanging nodes. The 2-D code written for this thesis (using the traditional DD form of the equations) was set up being able to have hanging node meshes that could be accelerated via DSA in the same way as in Baker et al.'s work [44, 65], with the additional ability of a Krylov wrapper as was applied to the DGFEM scheme by Warsa et al. [48]. However, how to incorporate the hanging node formulation into the calculation of the DWR is not a trivial extension.

Further investigation into using the DWR error estimator to approximate the error in a QoI due to the iteration error that results from the source iteration scheme would be an obvious extension to this work. The estimator has the potential to be used for goal-appropriate iteration scheme stopping criteria.

The DOMINO solver does not accommodate hanging node AMR, but it would be possible to add the error estimation routines to this industrial code. If this decision was made by EDF it would also be better to solve for the vertex values instead of the traditional DD unknowns. If the methods were applied to the DOMINO code it would allow for the methods to be tested on industrial test cases.

The DWR estimate studied in this work gives an estimate of the error in a QoI due to spatial discretisation only. The error is with respect to the analytical solution to the sets of equations that are already discretised in angle and energy. These equations are different from the continuous neutron transport equation for which an approximation is sought. Research into how DWR estimation could be extended to the S_N equations is therefore essential future work to ensure that the DWR estimate does not overly underestimate the error with respect to the governing mathematical model.

Bibliography

[1] J. Stepanek, T. Auerbach, and W. Halg. Calculation of four thermal reactor benchmark problems in x-y geometry. Technical Report 464, Federal Institute of Technology Zurich, 1982.

[2] J. Kópházi. Radiant II code. Rolls Royce, 2016. Code written and all test cases using this code completed by J. Kópházi.

[3] Nuclear EngD Centre. *Nuclear Engineering Doctorate Research Engineer Programme Handbook*. Revision 8.1.

[4] World Nuclear Association. Nuclear Power in the United Kingdom. Web page. URL http://www.world-nuclear.org/information-library/country-profiles/countries-t-z/united-kingdom.aspx. Accessed July 2016.

[5] Energy Act 2013, Chapter 32. Act of parliament. URL http://www.legislation.gov.uk/ukpga/2013/32/contents/enacted/data.htm.

[6] UK Gov. Department for Business, Energy & Industrial Strategy. Government confirms Hinkley Point C project following new agreement in principle with EDF. Online, September 2016. URL https://www.gov.uk/government/news/government-confirms-hinkley-point-c-project-following-new-agreement-in-principle-with-edf.

[7] EDF. The EDF adventure. A global electricity company in the spotlight. Web page. URL https://www.edf.fr/en/the-edf-group/world-s-largest-power-company/history. Accessed July 2016.

[8] E.E.Lewis. *Fundamentals of Nuclear Reactor Physics*. Academic Press, 2008.

[9] David Lecarpentier. Calcul neutronique des réacteurs du futur à EDF. In *Séminaire GEDEPEON*, October 2004.

[10] Teresa Kulikowska. Reactor lattice codes. In *Workshop on Nuclear Data and Nuclear Reactors: Physics, design and Safety, Trieste*, 2000.

[11] E. Brun, F. Damian, C.M. Diop, E. Dumonteil, F.X. Hugot, C. Jouanne, Y.K. Lee, F. Malvagi, A. Mazzolo, O. Petit, J.C. Trama, T. Visonneau, and A. Zoia. TRIPOLI-4, CEA, EDF and AREVA reference monte carlo codes. *Annals of Nuclear Energy*, 82:151 – 160, 2015. ISSN 0306-4549. doi: http://dx.doi.org/10.1016/j.anucene.2014.07.053. URL http://www.sciencedirect.com/science/article/pii/S0306454914003843. Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2013, SNA + MC 2013. Pluri- and Trans-disciplinarity, Towards New Modeling and Numerical Simulation Paradigms.

[12] Alain Hébert. *Applied Reactor Physics*. Presses Internationales Polytechnique, 2010.

[13] Angélique Ponçot and Tanguy Courau. Elements of validation of the EDF R&D S_N and SP_N solvers vs. MCNP multigroup assembly calculations. In *PHYSOR 2010 - Advances in reactor physics to power the nuclear renaissance*, Pittsburgh, Pennsylvania, USA, May 2010.

[14] N. K. Madsen. Pointwise convergence of the three-dimensional discrete ordinate method. *SIAM J. Numerical Analysis.*, 8:2, 1971.

[15] Ronald F. Boisvert, Ronald Cools, and Bo Einarsson. *Accuracy and Reliability in Scientific Computing*, chapter Assessment of Accuracy and Reliability, pages 13–32. SIAM, 2005.

[16] François Févotte and Bruno Lathuilière. VERROU: a CESTAC evaluation without recompilation. In *International Symposium on Scientific Computing, Computer Arithmetics and Verified Numerics (SCAN)*, Uppsala, Sweden, September 2016.

[17] Kambiz Salari and Patrick Knupp. Code verification by the method of manufactured solutions. Technical Report SAND2000-1444, Sandia National Laboratories, June 2000.

[18] Roland Becker and Rolf Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica*, 10:1–102, 2001.

[19] D. Meidner, R. Rannacher, and J. Vihharev. Goal-oriented error control of the iterative solution of finite element equations. *Journal of Numerical Mathematics*, 17:143–172, 2009.

[20] R. Rannacher, A. Westenberger, and W. Wollner. Adaptive finite element solution of eigenvalue problems: balancing of discretiszation and iteration error. *Journal of Numerical Mathematics*, 18:303–327, 2010.

[21] Ivo Babuška, Whiteman John R, and Theofanis Strouboulis. *Finite Elements. An introduction to the method and error estimation.* Oxford University Press, 2011.

[22] Ahmed K. Noor and Ivo Babuška. Quality assessment and control of finite element solutions. *Finite Elements in Analysis and Design*, 3(1):1–26, 1987. ISSN 0168-874X. doi: http://dx.doi.org/10.1016/0168-874X(87)90030-8. URL http://www.sciencedirect.com/science/article/pii/0168874X87900308.

[23] Pouya Sabouri. *Application of Perturbation Theory Methods to Nuclear Data Uncertainty Propagation using the Collision Probability Method.* PhD thesis, L'Université de Grenoble, 2013.

[24] Salli Moustafa, Ivan Dutka-Malen, Laurent Plagne, Angélique Ponçot, and Pierre Ramet. Shared memory parallelism for 3-D Cartesian discrete ordinates solvers. In *Joint international conference on supercomputing in nuclear applications and Monte Carlo*, La Cité des Sciences et de l'Industrie, Paris, France, October 2013. Presentation.

[25] Salli Moustafa, Ivan Dutka-Malen, Laurent Plagne, Angélique Ponçot, and Pierre Ramet. Shared memory parallelism for 3D Cartesian discrete ordinates solvers. *Annals of Nuclear Energy*, 82:179 – 187, 2015. ISSN 0306-4549. doi: http://dx.doi.org/10.1016/j.anucene.2014.08.034. URL http://www.sciencedirect.com/science/article/pii/S0306454914004265. Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2013, SNA + MC 2013. Pluri- and Trans-disciplinarity, Towards New Modeling and Numerical Simulation Paradigms.

[26] Yaqi Wang and Jean C. Ragusa. Standard and goal-oriented adaptive mesh refinement applied to radiation transport on 2-D unstructured triangular meshes. *Journal of Computational Physics*, 230(3):763–788, 2011. ISSN 0021-9991. doi: 10.1016/j.jcp.2010.10.018. URL http://www.sciencedirect.com/science/article/pii/S0021999110005747.

[27] Bruno Turcksin, Jean C. Ragusa, and Wolfgang Bangerth. Goal-Oriented h-Adaptivity for the Multigroup SP_N Equations. *Nuclear Science and Engineering*, 165:305319, 2010.

[28] M. Ainsworth and J. Tinsley Oden. A posteriori error estimation in finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 142(12):1 – 88,

1997. ISSN 0045-7825. doi: http://dx.doi.org/10.1016/S0045-7825(96)01107-3. URL http://www.sciencedirect.com/science/article/pii/S0045782596011073.

[29] K.J. Fidkowski and P.L. Roe. An entropy adjoint approach to mesh refinement. *SIAM Journal of Scientific Computing*, 32(3):1261–1287, 2010.

[30] M.M.R. Williams. *Lecture Notes in Reactor Physics and Kinetics*, volume 1. Nuclear Engineering Department, Queen Mary College, University of London, 1981.

[31] Teresa Kulikowska. An introduction to the neutron transport phenomena. In *Workshop on Nuclear Data and Nuclear Reactors: Physics, design and Safety, Trieste*, 2000.

[32] E.E. Lewis and W.F. Miller Jr. *Computational Methods of Neutron Transport.* American Nuclear Society Inc., 1993.

[33] B.D.Ganapol. *Analytical benchmarks for nuclear engineering applications. Case studies in neutron transport theory.* Nuclear Energy Agency, 2008.

[34] R. Douglas O'Dell and Raymond E. Alcouffe. Transport calculations for nuclear analyses: Theory and guidelines for effective use of transport codes. Technical Report LA-10983-MS, Los Alamos, 1987.

[35] K.D.Lathrop. The early days of the S_N methods. In *International Conference American Nuclear Society/European Nuclear Society*, Chicago, Illinois, November 1992.

[36] X-5 Monte Carlo Team. MCNP - A general Monte Carlo N-particle transport code, Version 5. Technical Report LA-UR-03-1987, Los Alamos National Laboratory, 2003.

[37] Iván Lux and László Koblinger. *Monte Carlo Particle Transport Methods: Neutron and Photon Calculations.* CRC Press, 1991.

[38] Esam M. A. Hussein. *Radiation Mechanics.* Elsevier, 2007.

[39] M.D.DeHART. Advancementsin generalized-geometry discrete ordinates transport for lattice physics calculations. In *Proc. Topl. Mtg. Reactor Physics: Advances in Nuclear Analysis and Simulation (PHYSOR 2006)*, Vancouver, British Columbia, Canada, September 2006. American Nuclear Society.

[40] Weston M. Stacey. *Nuclear Reactor Physics.* Wiley-Vch, 2007.

[41] George I. Bell and Samuel Glasstone. *Nuclear Reactor Theory.* Van Nostrand Reinhold Company, 1970.

[42] Shoichiro Nakamura. *Computational Methods in Engineering and Science with applications to fluid dynamics and nuclear systems.* Krieger, 1986.

[43] Christian Aussourd. Styx 3-D implementation on SMP architectures. In *Proc. Int. Topl. Mtg. Advances in Reactor Physics and Mathematics and Computation into the Next Millennium*, Pittsburgh, Pennsylvania, May 2000. American Nuclear Society.

[44] Randal S. Baker. A block adaptive mesh refinement algorithm for the neutral particle transport equation. *Nuclear Science and Engineering*, 141:1–12, 2002.

[45] Bengt G. Carlson. Solution of the transport equation by S_N approximations. October LA-1599, Los Alamos Scientific Laboratory, 1953.

[46] D. E. Cullen. Why are the P_N and S_N methods equivalent? Technical report, U.S. Department of Energy, Lawrence Livermore National Laboratory, September 2001.

[47] Sebastian Schunert and Yousry Y. Azmy. On the adequacy of Cartesian geometry discrete ordinates solutions for assembly calculations. In *International Conference on Mathematics, Computational Methods & Reactor Physics*, Saratoga Springs, New York, May 2009.

[48] James S. Warsa, Todd A. Wareing, and Jim E. Morel. Krylov iterative methods and the degraded effectiveness of diffusion synthetic acceleration for multidimensional S_N calculations in problems with material discontinuities. *Nuclear Science and Engineering*, 147: 218–248, 2004.

[49] Paul Reuss. *Neutron Physics.* EDP Sciences, 2008.

[50] Anne Gelb. The resolution of the Gibbs phenomenon for spherical harmonics. *Mathematics of computation*, 66:699–717, 1997.

[51] William R. Martin, Carl E. Yehnert, Leonard Lorence, and James J. Duderstadt. Phase-space finite element methods applied to the first-order form of the transport equation. *Annals of Nuclear Energy*, 8:633, 1981.

[52] William Russell Martin. *The Application of the finite element method to the neutron transport equation.* PhD thesis, University of Michigan, 1976.

[53] W.H. Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. Technical report, Los Alamos scientific laboratory of the University of California, 1971.

[54] NEA. PARTISN 4.00: 1-D, 2-D, 3-D Time-Dependent, Multigroup Deterministic Parallel Neutral Particle Transport Code. Web, 2009. URL http://www.oecd-nea.org/tools/abstract/detail/ccc-0707. CCC-0707 PARTISN 4.00.

[55] Thomas M. Evans, Alissa S. Stafford, Rachel N. Slaybaugh, and Kevin T. Clarno. Denovo: A new three-dimensional parallel discrete ordinates code in scale. *Nuclear Technology*, 171: 171–200, August 2010.

[56] J. P. Hennart and E. del Valle. A generalized nodal finite element formalism for discrete ordinates equations in slab geometry: Part I: Theory in the continuous moment case. *Transport Theory and Statistical Physics*, 24:449–478, 1995.

[57] Alain Hébert. High order diamond differencing schemes. *Annals of Nuclear Energy*, 33: 1479–1488, 2006.

[58] Nicolas Martin and Alain Hébert. A three-dimensional S_N high-order diamond differencing discretization with a consistent acceleration scheme. *Annals of Nuclear Energy*, 36:1787–1796, 2009.

[59] Dimitris Valougeorgis. Boundary treatment of the diffusion synthetic acceleration method for fixed-source discrete-ordinates problems in x-y geometry. *Nuclear Science and Engineering*, 100:142–148, 1988.

[60] Sebastian Schunert and Yousry Azmy. Using the Cartesian discrete ordinates code DORT for assembly-level calculations. *Nuclear Science and Engineering*, 173:233–258, 2013.

[61] Yousry Y. Azmy, Jess C. Gehin, and Roberto Orsi. DORT solutions to the two-dimensional C5G7 MOX benchmark problem. *Progress in Nuclear Energy*, 45:215–231, 2004.

[62] J.Patrick Jessee, Woodrow A. Fiveland, Louis H. Howell, Phillip Colella, and Richard B. Pember. An adaptive mesh refinement algorithm for the radiative transport equation. *Journal of Computational Physics*, 139(2):380–398, 1998. ISSN 0021-9991. doi: http://dx.doi.org/10.1006/jcph.1997.5870. URL http://www.sciencedirect.com/science/article/pii/S0021999197958708.

[63] Christian Aussourd. Styx: A multi-dimensional AMR S_N schemes. *Nuclear Science and Engineering*, 143:281–290, 2003.

[64] Robert C. Ward, Randal S. Baker, and Jim E. Morel. A diffusion synthetic acceleration method for block adaptive mesh refinement. In *Mathematics and Computation, Supercomputing, Reactor Physics and Nuclear and Biological Applications Conference*, Palais des Papes, Avignon, France, September 2005. Los Alamos National Laboratory.

[65] Robert C. Ward, Randal S. Baker, and Jim E. Morel. A diffusion synthetic acceleration method for block adaptive mesh refinement. *Nuclear Science and Engineering*, 152:164–179, 2006.

[66] S.A. Turner. Automatic mesh coarsening for discrete ordinates codes. In *Int. Conf. Mathematics and Computation, Reactor Physics and Environmental Analysis in Nuclear Applications, Madrid, Spain*, volume 2, pages 1423–1999, September 1999.

[67] F.E. Jatuff. Error estimation and adaptivity for the neutron diffusion problem. *Annals of Nuclear Energy*, 22:775–786, 1995.

[68] F.E. Jatuff and C.J. Gho. A physical interpretation and benchmarking of nodal diffusion error estimators. *Annals of Nuclear Energy*, 25:1235–1246, 1998.

[69] Oscar M. Zamonsky, C.J. Gho, and Y.Y Azmy. Error estimation and adaptive order nodal method for solving multidimensional transport problems. In *American nuclear society radiation protection and shielding division topical conference*, Nashville, TN, April 1998.

[70] J.M. Ruggieri, R. Boyer, J.Y. Doriath, and P.J. Finck. Accounting for strong localized heterogeneities and local transport effect in core calculation. *Nuclear Science and Engineering*, 124:82–88, 1996.

[71] Hui Zhang and E. E. Lewis. An adaptive approach to variational nodal diffusion problems. *Nuclear Science and Engineering*, 137:14–22, 2001.

[72] Hui Zhang and E. E. Lewis. Spatial adaptivity applied to the variational nodal pn equations. *Nuclear Science and Engineering*, 142:57–63, 2002.

[73] James S. Warsa and Anil K. Prinja. p-adaptive numerical methods for particle transport. *Transport Theory and Statistical Physics*, 28:229–270, 1999.

[74] Y. Wang and J.C. Ragusa. Adaptive hp-mesh refinement applied to 1-D, one group diffusion problems. *Trans. Am. Nucl. Soc.*, 93:585, 2005.

[75] Leszek Demkowitz. *Computing with hp-adaptive finite elements. Volume 1. One and two dimensional elliptic and maxwell problems.* Chapman and Hall / CRC, 2007.

[76] Leszek Demkowitz, Jason Kurtz, David Pardo, Maciej Paszyński, Waldemar Rachowicz, and Adam Zdunek. *Computing with hp-adaptive finite elements. Volume 2. Frontiers: three dimensional elliptic and Maxwell problems with applications.* Chapman and Hall / CRC, 2008.

[77] Yaqi Wang and Jean C. Ragusa. Adaptive automated solution of the multi-group diffusion equations. In *PHYSOR-2006, ANS Topical Meeting on Reactor Physics*. Organized and hosted by the Canadian Nuclear Society. Vancouver, BC, Canada., September 2006.

[78] Yaqi Wang and Jean Ragusa. Goal-oriented hp-mesh adaptation for 1-D multigroup diffusion equations. In *Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications*, Monterey, California, April 2007. American Nuclear Society, LaGrange Park, IL.

[79] P. Šolín and L. Demkowicz. Goal-oriented hp-adaptivity for elliptic problems. *Computer methods in applied mechanics and engineering*, 193:449–468, 2004.

[80] Yaqi Wang, Wolfgang Bangerth, and Jean Ragusa. Three-dimensional h-adaptivity for the multigroup neutron diffusion equations. *Progress in Nuclear Energy*, 51(3):543 – 555, 2009. URL http://www.sciencedirect.com/science/article/pii/S0149197008001406.

[81] Yaqi Wang and Jean Ragusa. Application of hp adaptivity to the multigroup diffusion equations. *Nuclear Science and Engineering*, 161:22–48, 2009.

[82] Jean C. Ragusa. Adaptive solution of the multigroup diffusion equation on irregular structured grids using a conforming finite element method formulation. In *PHYSOR - The Physics of Fuel Cycles and Advanced Nuclear Systems: Global Developments Chicago, Illinois*, April 2004.

[83] Jean C. Ragusa. A simple hessian-based 3-D mesh adaptation technique with applications to the multigroup diffusion equations. *Annals of Nuclear Energy*, 35(11): 2006–2018, 2008. ISSN 0306-4549. doi: 10.1016/j.anucene.2008.06.008. URL http://www.sciencedirect.com/science/article/pii/S0306454908001825.

[84] Jean C. Ragusa. Application of h, p, and hp-mesh adaptation techniques to the SP_3 equations. *Transport Theory and Statistical Physics*, 39:234–254, 2011.

[85] Bruno Turcksin and Jean C. Ragusa. Goal-oriented mesh adaptivity for multi-dimension SP_N equations. In *International Conference on Mathematics, Computational Methods & Reactor Physics (M&C 2009)*, Saratoga Springs, New York, May 2009.

[86] Jean C. Ragusa and Yaqi Wang. A two-mesh adaptive mesh refinement technique for S_N neutral-particle transport using a higher-order DG-FEM. *Journal of Computational and Applied Mathematics*, 233:3178–3188, 2010.

[87] Yaqi Wang and Jean C. Ragusa. Diffusion synthetic acceleration for high-order discontinuous finite element S_N transport schemes and application to locally refined unstructured meshes. *Nuclear Science and Engineering*, 166:145–166, 2010.

[88] Wolfgang Bangerth and Rolf Rannacher. *Adaptive Finite Element Methods for Differential Equations*. Lectures in Mathematics ETH Zürich. Birkhäuser Verlag, 2003.

[89] D. Lathouwers. Goal-oriented spatial adaptivity for the S_N equations on unstructured triangular meshes. *Annals of Nuclear Energy*, 38(6):1373–1381, 2011. ISSN 0306-4549. doi: 10.1016/j.anucene.2011.01.038. URL http://www.sciencedirect.com/science/article/pii/S0306454911000594.

[90] D. Lathouwers. Spatially adaptive eigenvalue estimation for the S_N equations on unstructured triangular meshes. *Annals of Nuclear Energy*, 38(9):1867–1876, 2011. ISSN 0306-4549. doi: 10.1016/j.anucene.2011.05.013. URL http://www.sciencedirect.com/science/article/pii/S0306454911002039.

[91] Mark A. Goffin, Christopher M.J. Baker, Andrew G. Buchan, Christopher C. Pain, Matthew D. Eaton, and Paul N. Smith. Minimising the error in eigenvalue calculations involving the boltzmann transport equation using goal-based adaptivity on unstructured meshes. *Journal of Computational Physics*, 242:726–752, 2013.

[92] Mark A. Goffin, Andrew G. Buchan, Anca C. Belme, Christopher C. Pain, Matthew D. Eaton, Paul N. Smith, and Richard P. Smedley-Stevenson. Goal-based angular adaptivity applied to the spherical harmonics discretisation of the neutral particle transport equation. *Annals of nuclear energy*, 71:60–80, 2014.

[93] Jose I. Duo, Yousry Y. Azmy, and Ludmil T. Zikatanov. A posteriori error estimator and AMR for discrete ordinates nodal transport methods. *Annals of Nuclear Energy*, 36(3):268 – 273, 2009. ISSN 0306-4549. doi: http://dx.doi.org/10.1016/j.anucene.2008.12.008. URL

http://www.sciencedirect.com/science/article/pii/S0306454908003368. PHYSOR 2008.

[94] HyeongKae Park and Cassiano R. E. de Oliveira. Coupled space-angle adaptivity for radiation transport calculations. *Nuclear Science and Engineering*, 161:216–234, 2009.

[95] L. Gastaldo, R. Le Tellier, C. Suteau, D. Fournier, and J.M. Ruggieri. High-order discrete ordinate transport in non-conforming 2-D Cartesian meshes. In *International Conference on Mathematics, Computational Methods & Reactor Physics (M&C 2009)*, 2009.

[96] Krzysztof J. Fidkowski and David L. Darmofal. Review of output-based error estimation and mesh adaptation in computational fluid dynamics. *AIAA Journal*, 49(4):673–694, 2011.

[97] M.B. Giles and N.A. Pierce. Improved lift and drag estimates using adjoint euler equations. *AIAA Paper 99-3293*, 1999.

[98] Niles A. Pierce and Michael B. Giles. Adjoint recovery of superconvergent functionals from PDE approximations. *SIAM REVIEW*, 42(2):247–264, 2000.

[99] M.B. Giles and E. Süli. Adjoint methods for PDEs: a posteriori error analysis and post-processing by duality. *Acta Numerica*, 11:145 – 236, 2002.

[100] M.B. Giles, N.A. Pierce, and E. Süli. Progress in adjoint error correction for integral functionals. *Comput Visual Sci*, 6:113–121, 2004.

[101] N. A. Pierce and M.B. Giles. Adjoint and defect bounding and correction for functional estimates. *Journal of Computational Physics*, 200:769–794, 2004.

[102] Qingshan Chen and Max Gunzburger. Goal-oriented a posteriori error estimation for finite volume methods. *Journal of Computational and Applied Mathematics*, 265:69–82, 2014.

[103] Jens-Dominik Müller and Michael B. Giles. Solution adaptive mesh refinement using adjoint error analysis. In *15th Computational Fluid Dynamics Conference*, Anaheim, California, June 2001.

[104] David A. Venditti and David L. Darmofal. A multilevel error estimation and grid adaptive strategy for improving the accuracy of integral outputs. In *14th computational fluid dynamics conference*, number AIAA 99-3292, June 1999.

[105] D.A. Venditti and D.L. Darmofal. Adjoint error estimation and grid adaptation for functional outputs: Application to quasi-one-dimensional flow. *Journal Computational Physics*, 164:204227, 2000.

[106] David A. Venditti and David L. Darmofal. Grid adaptation for functional outputs: Application to two-dimensional inviscid flows. *Journal of Computational Physics*, 176:40–69, 2002.

[107] David A. Venditti and David L. Darmofal. Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows. *Journal of Computational Physics*, 187:22–46, 2003.

[108] R. Balasubramanian and J.C. Newman. Adjoint-based error estimation and grid adaptation for functional outputs: Application to two-dimensional, inviscid, incompressible flows. *Computers and fluids*, 38:320–332, 2009.

[109] Dmitri Kuzmin and Sergey Korotov. Goal-oriented a posteriori error estimates for transport problems. *Mathematics and Computers in Simulation*, 80:1674–1683, 2010.

[110] Dmitri Kuzmin and Matthias Möller. Goal-oriented mesh adaptation for flux-limited approximations to steady hyperbolic problems. *Journal of Computational and Applied Mathematics*, 233:3113–3120, 2010.

[111] Richard P. Dwight. Heuristic a posteriori estimation of error due to dissipation in finite volume schemes and application to mesh adaptation. *Journal of Computational Physics*, 227:2845–2863, 2008.

[112] Lutz Angermann. A posteriori estimates for errors of functionals on finite volume approximations to solutions of elliptic boundary value problems. *arXiv:1205.1980v1 [math.NA]*, May 2012.

[113] Lei Shi and Z.J.Wang. Adjoint-based error estimation and mesh adaptation for the correction procedure via reconstruction method. *Journal of Computational Physics*, 295: 261–284, 2015.

[114] J. Ponsin, F. Fraysse, M. Gómez, and M. Cordero-Gracia. An adjoint-truncation error based approach for goal-oriented mesh adaptation. *Aerospace Science and Technology*, 41: 229–240, 2015.

[115] Giovanni Todarello, Floris Vonck, Sébastien Bourasseau, Jaques Peter, and Jean-Antoine Désidéri. Finite-volume goal-oriented mesh adaptation for aerodynamics using functional derivative with respect to nodal coordinates. *Journal of computational physics*, 2016. doi: http://dx.doi.org/10.1016/j.jcp.2016.02.063.

[116] J.B. Collins, D. Estep, and S. Tavener. A posteriori error estimation for the Lax-Wendroff finite difference schemes. *Journal of Computational and Applied Mathematics*, 263(0):299 – 311, 2014. ISSN 0377-0427. doi: http://dx.doi.org/10.1016/j.cam.2013.12.035. URL http://www.sciencedirect.com/science/article/pii/S0377042713007085.

[117] Jehanzeb H. Chaudhry, Donald Estep, Victor Ginting, John N. Shadid, and SImon Tavener. A posteriori error analysis of IMEX multi-step time integration methods for advection-diffusion-reaction equations. *Computer methods in applied mechanics and engineering*, 285:730–751, 2015.

[118] James. S. Warsa. Analytical S_N solutions in heterogeneous slabs using symbolic algebra computer programs. *Annals of Nuclear Energy*, 29:851–874, 2002.

[119] William H. Reed. New difference schemes for the neutron transport equation. *Nuclear Science and Engineering*, 36:309–314, 1971.

[120] James S. Warsa. Analytical solution of the discrete ordinates transport equations. Online, 2000. URL www.maplesoft.com/applications/view.aspx?SID=3792&view=html.

[121] C.E.Siewert. The critical problem with high-order anisotropic scattering. *Annals of nuclear energy*, 28:825–829, 2001.

[122] C.E.Siewert. A concise and accurate solution to Chandrasekhar's basic problem in radiative transfer. *Journal of Quantitative Spectroscopy & Radiative Transfer*, 64:109–130, 2000.

[123] James Bryce Taylor. *The development of a three-dimensional nuclear reactor kinetics methodology based on the method of characteristics*. PhD thesis, The Pennsylvania State University, 2007.

[124] Juhani Pitkäranta. On the spatial differencing of the discrete ordinate neutron transport equation. *SIAM J. Numerical Analysis*, 15:859–869, 1978.

[125] Boško S. Jovanović and Endre Süli. *Analysis of Finite Difference Schemes*. Springer, 2014.

[126] J.T. Oden and J.N. Reddy. *An introduction to the mathematical theory of finite elements.* John Wiley and Sons, 1976.

[127] Grégoire Allaire. *Numerical Analysis and Optimization.* Oxford Science Publications, 2007.

[128] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical Mathematics.* Springer, 2000.

[129] Howard C. Elman, David J. Silvester, and Andrew J. Wathen. *Finite Elements and Fast Iterative Solvers : with Applications in Incompressible Fluid Dynamics.* Oxford University Press, 2014.

[130] J.N. Reddy. *Applied functional analysis and variational methods in engineering.* McGraw-Hill, 1986.

[131] J.E. Hicken and D.W. Zingg. Dual consistency and functional accuracy: a finite-difference perspective. *Journal of Computational Physics*, 256:161–182, 2014.

[132] B. Guo and I. Babuška. The h-p version of the finite element method part 1: The basic approximation results. *Computational Mechanics*, 1:21–41, 1986.

[133] B. Guo and I. Babuška. The h-p version of the finite element method part 2: General results and applications results. *Computational Mechanics*, 1:203–220, 1986.

[134] Rüdiger Verfürth. *A Posteriori Error Estimation Techniques for Finite Element Methods.* Numerical Mathematics and Scientific Computation. Oxford Science Publications, 2013.

[135] Yaqi Wang. Hp-mesh adaptation for 1-D multigroup neutron diffusion problems. Master's thesis, Texas A&M University, December 2006.

[136] Marco Ceze and Krzysztof J. Fidkowski. Anisotropic hp-adaptation framework for functional prediction. *AIAA Journal*, 51(2):492–509, 2013.

[137] R.S. Jeffers, J. Kópházi, M.D. Eaton, F. Févotte, F. Hülsemann, and J. Ragusa. Goal-based h-adaptivity of the 1-D diamond difference discrete ordinate method. *Submitted for publication*, 2016.

[138] Claes Johnson and Juhani Pitkäranta. Convergence of a fully discrete scheme for two-dimensional neutron transport. *SIAM J. Numer. Anal.*, 20(5):951–966, 1983.

[139] Yaqi Wang and Jean C. Ragusa. On the Convergence of DGFEM Applied to the Discrete Ordinates Transport Equation for Structured and Unstructured Triangular Meshes. *Nuclear Science and Engineering*, 163:56–72, 2009.

[140] R. Le Tellier and A. Hébert. A modeling of BWR-MOX assemblies based on the characteristics method combined with advanced self-shielding Models. *Nuclear Science and Engineering*, 158:231–243, 2008.

[141] J. I. Duo and Y. Y. Azmy. Error comparison of diamond difference, nodal, and characteristic methods for solving multidimensional transport problems with the discrete ordinates approximation. *Nuclear Science and Engineering*, 156:139–153, 2007.

[142] Jose Ignacio Duo. *Error estimates for nodal and short characteristics spatial approximations of two-dimensional discrete ordinates methods.* PhD thesis, The Pennsylvania State University, May 2008.

[143] J.P. Hennart and E del Valle. On nodal transport methods. Technical Report 2947, INRIA, July 1996.

[144] Edward W. Larsen. Spatial convergence properties of the diamond difference method in x,y geometry. *Nuclear Science and Engineering*, pages 710–713, 1981.

[145] J.I. Duo and Y.Y. Azmy. Spatial convergence study of discrete ordinates methods via the singular characteristic tracking algorithm. *Nuclear Science and Engineering*, 162:41–55, 2009.

[146] Thomas J.R. Hughes. *The Finite Element Method. Linear Static and Dynamic Finite Element Analysis.* Dover, 2000.

[147] Walter Rudin. *Functional Analysis.* McGraw-Hill, ISBN 0-07-054236-8., second edition, 1991.

A. Test cases used throughout the thesis

The thesis requires many test cases, some with analytical solutions in order to test the work. The 1D test cases are described in section A.1 and the 2D test cases in section A.2.

A.1 1D test cases

This section first describes 6 1-D test cases and finally describes how analytical or reference solutions were obtained for them.

A.1.1 Test case 1

This is a zero-scatter, homogeneous test case with vacuum boundary conditions on both sides. The slab length is 8cm, 2 angular ordinates are used. There is a homogeneous total cross section (Σ_t) of 1cm^{-1} and a homogeneous and isotropic unit source (0.5cm^{-3}s^{-1}MeV^{-1} per ordinate).

The goal quantity is the integral of the total scalar flux throughout the domain for test case 1a and throughout only the first 1cm of the domain for test case 1b.

This test case is used as an initial test. The fact that the test case is homogeneous and that the angular directions are decoupled means that it is a good test case for the initial debugging stages of the code. The test case also tests the code in the extreme case of zero scatter. The two different goal quantities used in test case 1a and 1b will show the difference in the behaviour of AMR algorithms when a more localised QoI is sought. An analytical solution can be calculated by hand for the angular flux of this test case, and so the DWR calculated with the analytically calculated adjoint solutions was possible to verify the DWR theory.

A.1.2 Test case 2

This is a homogeneous test case with vacuum boundary conditions on both sides. The slab length is 8cm, 2 angular ordinates are used. There is a homogeneous total cross section (Σ_t) of 1cm^{-1}, a homogeneous isotropic scatter cross section (Σ_{s0}) of 0.5cm^{-1} and a homogeneous and isotropic unit source (0.5cm^{-3}s^{-1}MeV^{-1} per ordinate).

The goal quantity is the integral of the total scalar flux throughout the domain for test case 2a and throughout only the first 1cm of the domain for test case 2b.

This test case was used as a simple test case, similar to test case 1, but including the effects of neutron scatter. The different goal quantities used in test case 2a and 2b will also show the different behaviour of the AMR algorithm for the same geometry, but with different QoI. Similarly to test case 1, an analytical solution can be calculated by hand for the angular fluxes which is an advantage for debugging and testing.

A.1.3 Test case 3

This problem uses the Reed cell problem geometry and material properties, described in Reed's paper [119]. It includes very optically thick regions and drastic changes in material properties throughout. An S_8 approximation is used in angle.

The added complexity of this problem (highly heterogeneous cross sections) should give a better idea of the behaviour of the error estimators. It is also known that the DD equations give oscillatory scalar fluxes in areas of large total cross sections and small μ [119] because of the possible presence of negative angular fluxes when the cell width is larger than $\frac{2|\mu|}{\Sigma_t}$ [32]. The worst case of this in the Reed cell problem is where $\Sigma_t = 50.0\text{cm}^{-1}$. When S_8 is used, the smallest value of the cosine of the angle is $\mu = 0.183434642495650$. In this region Δx must be smaller than about 0.007cm to satisfy the negative flux criterion.

The goal quantity is the integral of the total scalar flux throughout the domain for test case 3a, throughout only the first 1cm of the domain for test case 3b, and between 1.6cm and 2.4cm for test case 3c. In test case 3b, the goal region coincides with the area that is most challenging for DD discretisation schemes. The goal region in test case 3c includes regions where the scalar flux has very sharp gradients, as can be seen from the analytical solution plotted in figure A.1.

This test case is known to be a challenge for the DD scheme [119] and is therefore chosen as a test case in this thesis to test the error estimation and AMR algorithms to their limits. It is unlikely that the DD scheme would be applied to such extreme cross-sections in reality, but testing it on this problem will provide some useful insight into the limitations of the method. The different QoI chosen in test cases a, b and c are also chosen to test different aspects of the error estimation calculations and AMR algorithms. Test case 3a is chosen as the least challenging test case and should test whether a reduction in the number of DoF can be obtained with the AMR algorithms even for a global integral due to the strong heterogeneity of the problem. Test case b places the QoI at the most challenging area for the DD discretisation scheme, therefore testing the methods to their limits. Test case c places the QoI where the analytical solution has large gradients, where we would expect local refinement should result in a decrease in the number of DoF required for an accurate solution. The test case also has the advantage of an analytical solution to the scalar flux being available.

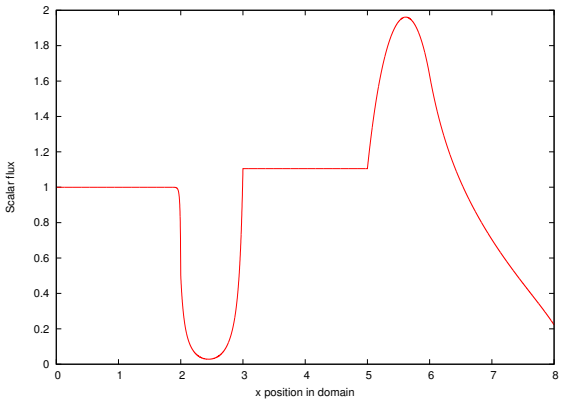
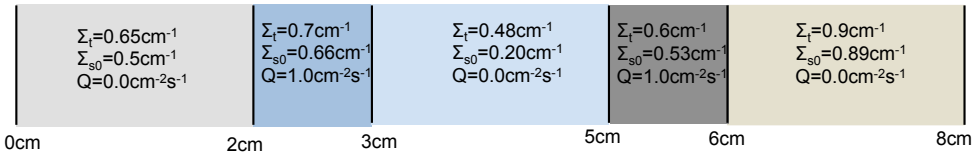


FIGURE A.1: Reed cell problem scalar flux solution

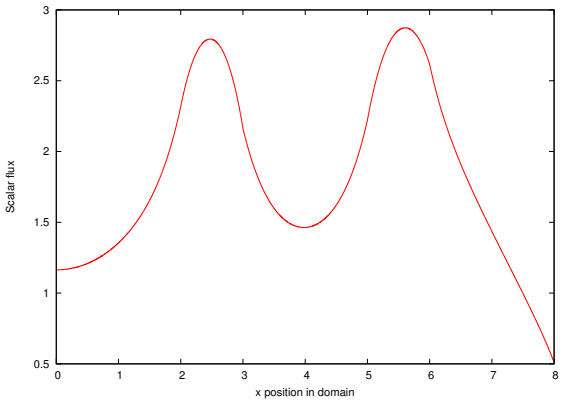
A.1.4 Test case 4

This test case takes the geometry of the Reed cell benchmark, but replaces the cross sections with less extreme values. The cross sections are those defined for the EIRA-2A benchmark problem defined in [1]. The geometry definition is given in figure A.2a. A refined solution to this problem is plotted in figure A.2b. The goal quantity is the integral of the total scalar flux throughout the domain for test case 4a and throughout only the first 1cm of the domain for test case 4b.

This test case was used as a more realistic test case than test case 3. The problem is is still heterogeneous and more complex than the first two test cases, but uses cross-sections that are



(A) Problem definition



(B) Scalar flux solution

FIGURE A.2: EIRA-1-D problem

less extreme and more suitable for the DD discretisation scheme. The test case also has the advantage of an analytical solution to the scalar flux being available.

A.1.5 Test case 5

This is a homogeneous criticality benchmark with vacuum boundary conditions on both sides of the domain. There is a homogeneous total cross section (Σ_t) of 1cm^{-1}, a homogeneous isotropic scatter cross section (Σ_{s0}) of 0.1cm^{-1} and the homogeneous fission (Σ_f) cross section multiplied by the average number of neutrons released per fission (ν), is unity: $\nu\Sigma_f = 1.0\text{cm}^{-1}$. An S_2 approximation is used in angle. The goal quantity is the value of K_{eff}.

This test case was chosen since it is an eigenvalue test case that has an analytical solution for the angular fluxes. This is useful for debugging and testing the error estimators.

A.1.6 Test case 6

This is a benchmark that takes the form shown in figure A.3. The cross sections were taken from a 2-D benchmark described in Taylor's thesis [123]. An S_2 approximation is used in angle. The goal quantity is the value of K_{eff}.

A more complex eigenvalue test case was needed and therefore this test case provides a heterogeneous eigenvalue test case.

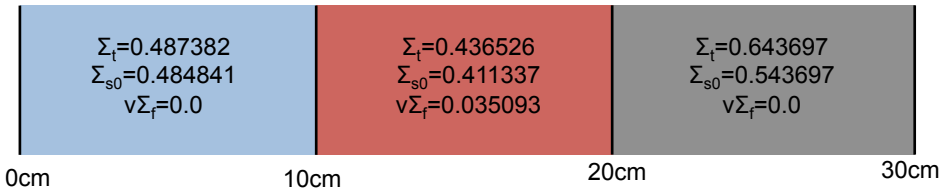


FIGURE A.3: 1-D-Mini-BWR geometry and material property definition.

A.1.7 Benchmarked solutions

Analytical solutions were calculated for test cases 1 and 2 by hand. An analytical solution to test case 3 is provided by Warsa's paper [118]. An analytical solution to test case 4 was calculated using the Maple script written by Warsa [118], which is freely available online [120].

For test case 5, methods described in references [121] and [122] were used to derive the following analytical solution for the critical width of the domain, ℓ_0, for the S_2 case with vacuum boundary conditions on both sides of the slab:

$$\ell_0 = -\frac{2\mu}{\Sigma_t\sqrt{c-1}}\arctan\left(-\frac{1}{c-1}\right), \tag{A.1}$$

where $c = \frac{\nu\Sigma_f+\Sigma_{s0}}{\Sigma_t}$ and $\mu = \frac{1}{\sqrt{3}}$ when Gauss-Legendre quadrature is used. We set ℓ_0 as an input so that the analytical solution should give $K_{eff} = 1$.

Given that an analytical solution for test case 6 is not available, the results are compared against results produced by J. Kópházi using the RADIANT-II DGFEM S_N code developed at Imperial College London [2]. The K_{eff} was spatially converged to within 10 significant figures for a given angularly discretised problem.

An important point to reiterate is that when referring to *analytical solutions* in this paper, we mean the analytical solutions to the already angularly discretised problems.

A.2 2D test cases

This section describes 4 2-D cases. An analytical reference solution is calculated for the zero scatter homogeneous case. The reference solution for all other test cases is provided by the

RADIANT-II DGFEM code [2].

A.2.1 Test case 7

This test case is an S_2, homogeneous square domain (side length 9cm) with vacuum boundary conditions on all sides and unit source throughout. $\Sigma_t = 1.0\text{cm}^{-1}$. Many different scatter ratios $\left(c = \frac{\Sigma_{s0}}{\Sigma_t}\right)$ are tested: $c = 0.0, 0.1, 0.5, 0.7, 0.9, 0.99$.

For $c \geq 0.1$, two goal quantities are tested: the integral of the scalar flux throughout the domain (denoted by a) and the integral throughout a quarter slice of the domain (denoted by b).

When $c = 0$, the analytical solution to the flux can be calculated as described in Duo's thesis [142]. The result is repeated here in equation (A.2).

$$\Phi(x,y,\Omega_m) = \begin{cases} \frac{Q}{4\pi\Sigma_t}\left(1 - e^{-\frac{\Sigma_t}{\mu_m}(x-0.5X[1-sgn(\mu_m)])}\right) & \text{for} \quad \frac{y-0.5Y[1-sgn(\eta_m)]}{\eta_m} > \frac{x-0.5X[1-sgn(\mu_m)]}{\mu_m} \\ \frac{Q}{4\pi\Sigma_t}\left(1 - e^{-\frac{\Sigma_t}{\eta_m}(y-0.5Y[1-sgn(\eta_m)])}\right) & \text{for} \quad \frac{y-0.5Y[1-sgn(\eta_m)]}{\eta_m} \leq \frac{x-0.5X[1-sgn(\mu_m)]}{\mu_m} \end{cases}. \tag{A.2}$$

Where Q is a constant isotropic source throughout the domain, the source in each angular direction being $\frac{Q}{4\pi}$, X and Y are the lengths of the domain in the x and y directions respectively.

When a square domain and S_2 quadrature are assumed this simplifies to:

$$\Phi(x,y,\Omega_1) = \begin{cases} \frac{Q}{4\pi\Sigma_t}\left(1 - e^{-\Sigma_t\sqrt{3}x}\right) & \text{for} \quad y > x \\ \frac{Q}{4\pi\Sigma_t}\left(1 - e^{-\Sigma_t\sqrt{3}y}\right) & \text{for} \quad y \leq x \end{cases} \tag{A.3}$$

$$\Phi(x,y,\Omega_2) = \begin{cases} \frac{Q}{4\pi\Sigma_t}\left(1 - e^{-\Sigma_t\sqrt{3}(L-x)}\right) & \text{for} \quad y < L - x \\ \frac{Q}{4\pi\Sigma_t}\left(1 - e^{-\Sigma_t\sqrt{3}y}\right) & \text{for} \quad y \geq L - x \end{cases} \tag{A.4}$$

$$\Phi(x,y,\Omega_3) = \begin{cases} \frac{Q}{4\pi\Sigma_t}\left(1 - e^{-\Sigma_t\sqrt{3}x}\right) & \text{for} \quad y > L - x \\ \frac{Q}{4\pi\Sigma_t}\left(1 - e^{-\Sigma_t\sqrt{3}(L-y)}\right) & \text{for} \quad y \leq L - x \end{cases} \tag{A.5}$$

$$\Phi(x,y,\Omega_4) = \begin{cases} \frac{Q}{4\pi\Sigma_t}\left(1 - e^{-\Sigma_t\sqrt{3}(L-x)}\right) & \text{for} \quad y < x \\ \frac{Q}{4\pi\Sigma_t}\left(1 - e^{-\Sigma_t\sqrt{3}(L-y)}\right) & \text{for} \quad y \geq x \end{cases} \tag{A.6}$$

These formulae are numerically integrated over angle and integrated analytically over space to give an analytical solution for the integral of the scalar flux throughout the spatial domain.

Reference results for this test case were provided by J. Kópházi via the RADIANT-II code [2], using a highly refined DGFEM mesh and the same angular quadrature.

This collection of test cases was used to provide a set of simple test cases for debugging, but also to show the effects of varying scatter ratios on the accuracy of the DWR error estimators for different goal quantities.

A.2.2 Test case 8

This is the EIRA-2A benchmark problem defined in [1]. The geometry and cross sections are given in figure A.4. Reference results for this test case were provided by J. Kópházi via the RADIANT-II code [2], using a highly refined DGFEM mesh and the same angular quadrature.

The goal quantity in test case 8a is the integral of the scalar flux throughout the whole domain. The goal quantity in test case 8b is the total absorption in the $\Sigma_t = 0.7$ box seen in figure A.4.

This test case was chosen to give a more challenging heterogeneous multi-dimensional fixed source problem.

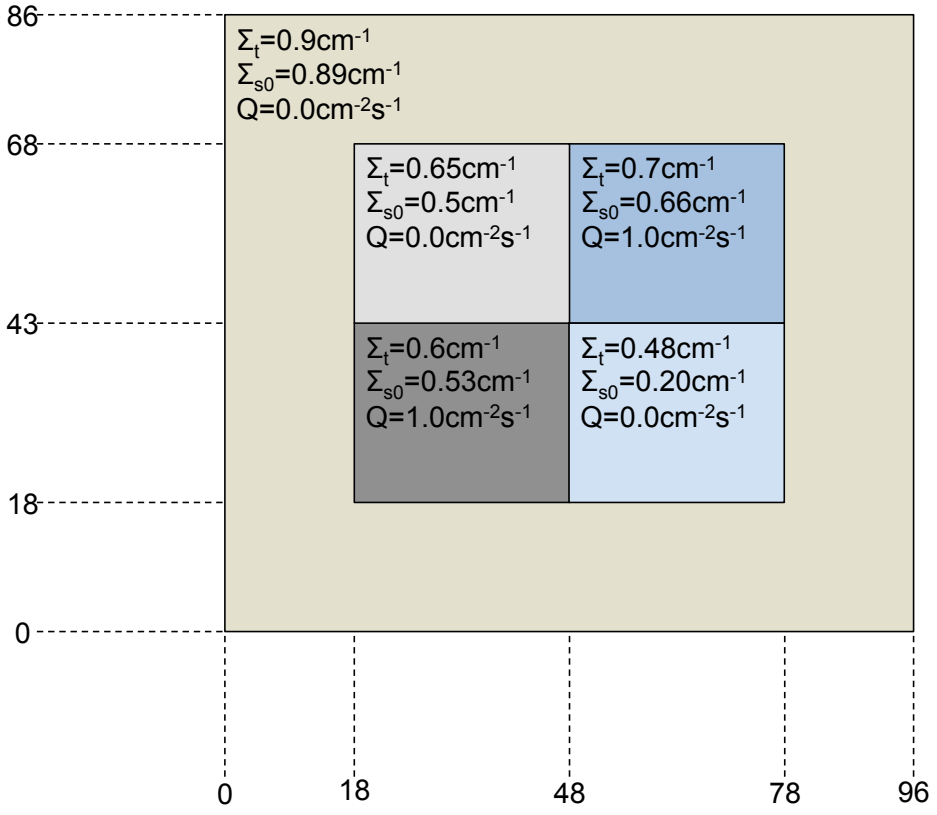


FIGURE A.4: EIRA 2-A 2-D problem [1].

A.2.3 Test case 9

This test case is an S_2 homogeneous square domain (5cm×5cm) with vacuum boundary conditions on all sides. $\Sigma_t = 1.0\text{cm}^{-1}$, $\Sigma_{s0} = 0.99\text{cm}^{-1}$ and $\nu\Sigma_f = 1.0\text{cm}^{-1}$. The QoI is the K_{eff}. Reference results for this test case were provided by J. Kópházi via the RADIANT-II code [2], using a highly refined DGFEM mesh and the same angular quadrature.

This test case was chosen in order to provide a simple eigenvalue test case for debugging.

A.2.4 Test case 10

This test case is an S_2 homogeneous square domain (5cm×5cm) with vacuum boundary conditions on all sides. $\Sigma_t = 10.0\text{cm}^{-1}$, $\Sigma_{s0} = 9.9\text{cm}^{-1}$ and $\nu\Sigma_f = 0.2\text{cm}^{-1}$. The QoI is the K_{eff}. Reference results for this test case were provided by J. Kópházi via the RADIANT-II code [2], using a highly refined DGFEM mesh and the same angular quadrature.

This test case was chosen in order to provide a simple eigenvalue test case that has less leakage than test case 9. This should be more appropriate for the DD discretisation scheme.

A.2.5 Test case 11

This test case is a variation on the mini-BWR test case found in Taylor's thesis [123]. The geometry in [123] is reflected in the x and y axis to produce the geometry of figure A.5. 121 copies of the lattice geometry described in figure A.5 are then placed in a 11×11 configuration. The boundary conditions of this larger geometry are then set to vacuum. The QoI is the K_{eff}. Reference results for this test case were provided by J. Kópházi via the RADIANT-II code [2], using a highly refined DGFEM mesh and the same angular quadrature.

This test case was chosen to provide a more realistic problem to test the use of DWR error estimators on eigenvalue problems.

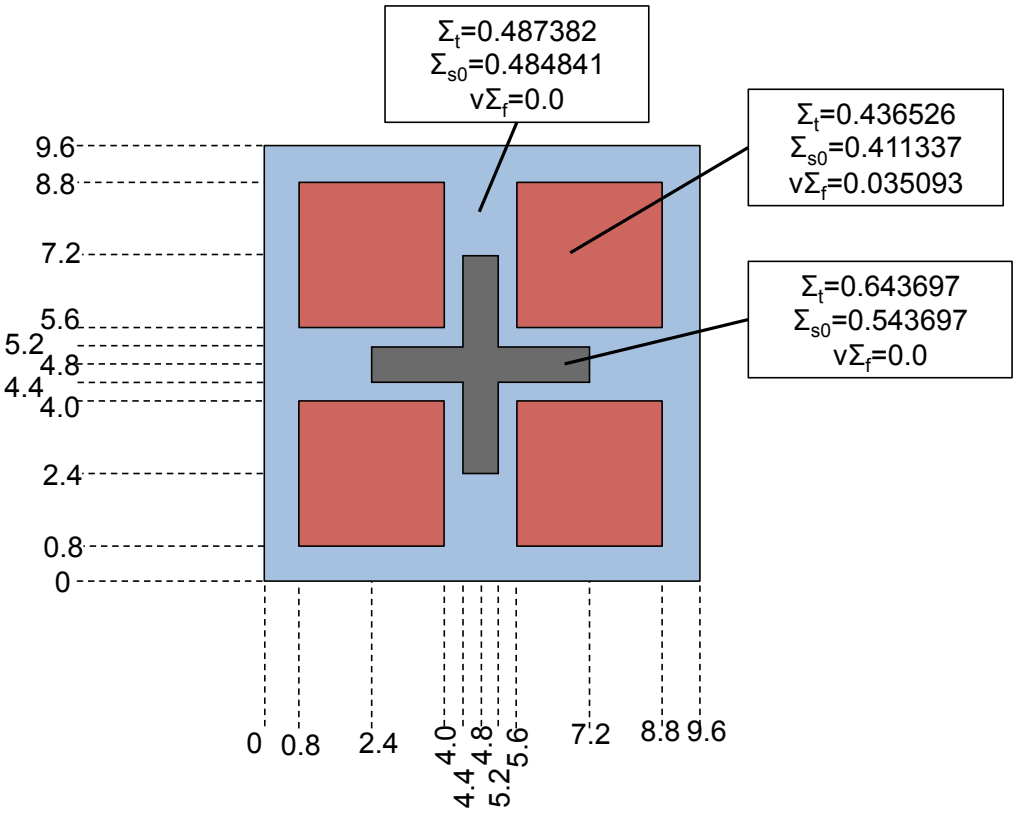


FIGURE A.5: 2-D-Mini-BWR geometry and material property definition.

B. Adjoint equation definition and explanation

This appendix explains how the adjoint equation is derived for the fixed source, detector function case.

Since the argument of the Gateaux derivative in equation (3.74) is linear, and has the value zero when $u = 0$, then the argument of (3.74) must also equal zero for all values of u:

$$\langle \Psi, W\boldsymbol{\Sigma}_{\mathrm{d}} \rangle_{\bar{V}} - \langle \mathcal{K}\Psi, \Psi^{\dagger} \rangle_{\bar{V}} = 0 \quad \Psi^{\dagger} \in \bar{V} \text{ and } \Psi \text{ a solution to equation (3.71).} \tag{B.1}$$

$\mathcal{K}$ is an unbounded linear operator in $\bar{V}$. This is because it contains a first order derivative, and $\bar{V}$ contains elements that contain functions in L^2. L^2 contains functions whose first derivatives are not bounded in the L^2 norm. $\mathcal{K}$ can also be seen as a bounded linear operator on the space $\bar{U} \subset \bar{V}$ to its range $\bar{V}$. In our case Ψ is constrained to a subspace of $\bar{U}$ with the essential boundary conditions of the problem under consideration: $\bar{U}_{0_i}$.

Below, the definition of an adjoint operator for both bounded and unbounded operators given in many textbooks is repeated [130, 147].

Bounded operators:
For a bounded linear operator, e.g. $\mathcal{T}$, on $\bar{V}$ the adjoint operator $\mathcal{T}^{\dagger}$ is defined as follows [130]:

$$\langle \mathcal{T}u, v \rangle_{\bar{V}} = \langle u, \mathcal{T}^{\dagger} v \rangle_{\bar{V}} \quad \forall v, u \in \bar{V}, \tag{B.2}$$

but the derivation of this depends on the fact that $\langle \mathcal{T}\cdot, v \rangle_{\bar{V}}$ for a given $v \in \bar{V}$ is a bounded linear functional on $\bar{V}$. This is because $\bar{V}$ is a Hilbert space, hence by the Riesz representation

theorem there exists a unique vector $v_0 \in \bar{V}$, for any fixed $v \in \bar{V}$ such that [102, 130]:

$$\langle \mathcal{T}u, v\rangle_{\bar{V}} = \langle u, v_0\rangle_{\bar{V}} \quad \forall u \in \bar{V}. \tag{B.3}$$

Equation (B.3) implies that for each $v \in \bar{V}$, there is a unique $v_0 \in \bar{V}$ that is associated with it. This in turn implies that there is a mapping from $\bar{V}$ to itself that shall be denoted by $\mathcal{T}^\dagger$ such that $v_0 = \mathcal{T}^\dagger v$. It is by this definition that we call the operator $\mathcal{T}^\dagger$ the adjoint of the operator of $\mathcal{T}$ [130].

Unbounded operators:

$\mathcal{K}$ is unbounded on $\bar{V}$, hence definition (B.2) cannot be applied in the same way. The definition in (B.2) of the adjoint operator is changed slightly for unbounded operators. The inner product:

$$\langle \mathcal{K}u, v\rangle_{\bar{V}}, \tag{B.4}$$

is only a bounded linear functional such that $u \mapsto \langle \mathcal{K}u, v\rangle_{\bar{V}}$ [102, 130] when u and v are taken from certain subspaces of $\bar{V}$.

Firstly $\mathcal{K}$ is bounded on its domain, $D(\mathcal{K})$. In the case of the continuous in space 1-D S_N equations, $D(\mathcal{K}) = \bar{U}_{0_i}$, a subspace of $\bar{V}$. The inner product:

$$\langle \mathcal{K}\Psi, v\rangle_{\bar{V}} \text{ where } \Psi \in D(\mathcal{K}), \tag{B.5}$$

is then only a bounded linear functional on $D(\mathcal{K})$ for fixed values of v taken from a certain subspace of $\bar{V}$ denoted by $D(\mathcal{K}^\dagger)$. This space is defined by the subspace of $\bar{V}$ for which there is a one-to-one mapping between $v \to v_0$ denoted by $\mathcal{K}^\dagger$ such that:

$$\langle \mathcal{K}\Psi, v\rangle_{\bar{V}} = \langle \Psi, v_0\rangle_{\bar{V}} \quad \forall \Psi \in D(\mathcal{K}). \tag{B.6}$$

Integration by parts must be employed to transfer the differential operator from one function to another in the above definition of $\mathcal{K}^\dagger$. This operation usually results in a boundary integral involving v. This means that a trace function of v needs to exist. It can therefore be deduced that the elements of the vector of v must have H^1 continuity [130]. Thus the domain of the adjoint operator must be a subspace of $\bar{U}$. The domain of the adjoint operator is then further restricted by the requirement that these boundary integrals go to zero to obtain an expression equivalent to equation (B.2). The boundary integrals are already zero for all boundaries for

which functions in $D(\mathcal{K})$ are zero. This means that $D(\mathcal{K}^\dagger) := \bar{U}_{0_o}$ such that the following holds [102, 130, 147]:

$$\langle \mathcal{K}u, v\rangle_{\bar{V}} = \langle u, \mathcal{K}^\dagger v\rangle_{\bar{V}} \text{ where } u \in D(\mathcal{K}) \text{ and } v \in D(\mathcal{K}^\dagger). \tag{B.7}$$

The above has shown that $D(\mathcal{K}) = \bar{U}_{0_i}$ and $D(\mathcal{K}^\dagger) = \bar{U}_{0_o}$, hence $D(\mathcal{K}) \neq D(\mathcal{K}^\dagger)$.

If $\Psi^\dagger$ in equation (B.1) is restricted to $D(\mathcal{K}^\dagger) \subset \bar{V}$, then equation (B.7) can be substituted into equation (B.1) to give:

$$\langle \Psi, W\boldsymbol{\Sigma}_{\mathrm{d}}\rangle_{\bar{V}} - \langle \Psi, \mathcal{K}^\dagger\Psi^\dagger\rangle_{\bar{V}} = 0 \quad \Psi^\dagger \in D(\mathcal{K}^\dagger) \text{ and } \Psi \text{ a solution to equation (3.71).} \tag{B.8}$$

By making the observation that if $\Psi^\dagger$ is the solution to the following strong formulation:

Find $\Psi^\dagger \in D(\mathcal{K}^\dagger)$ such that:

$$\langle v, W\boldsymbol{\Sigma}_{\mathrm{d}}\rangle_{\bar{V}} - \langle v, \mathcal{K}^\dagger\Psi^\dagger\rangle_{\bar{V}} = 0 \quad \forall v \in \bar{V}, \tag{B.9}$$

then equation (B.8) is just a special case of equation (B.9), we can then say that the solution $\Psi^\dagger$ to equation (3.75) satisfies equation (3.74).

C. The inability to guarantee positive fluxes using the DD scheme in 2-D

Here I will show an example of when a negative flux is produced no matter what the cell size, this is in contrast to the 1D case, where flux positivity was guaranteed given a small enough cell width [32].

I will show that the same problems arise for the DD method in section C.1 and the box method in section C.2.

C.1 DD method

$$\mu\frac{\bar{\psi}_e - \bar{\psi}_w}{\Delta x} + \eta\frac{\bar{\psi}_n - \bar{\psi}_s}{\Delta y} + \Sigma_t\bar{\psi}_c = q \tag{C.1}$$

Once the DD auxiliary equations have been substituted into the above equation and we rearrange to obtain the average flux for the cell, we have:

$$\bar{\psi}_c = \frac{q + \frac{2\mu\psi_w}{\Delta x} + \frac{2\eta\psi_s}{\Delta y}}{a}, \tag{C.2}$$

where

$$a = \frac{2\mu}{\Delta x} + \frac{2\eta}{\Delta y} + \Sigma_t. \tag{C.3}$$

Assuming a constant incoming flux along the incoming west face and a vacuum boundary condition on the incoming south face and zero source within the cell, this reduces to:

$$\bar{\psi}_c = \frac{2\mu\psi_w}{a\Delta x}. \tag{C.4}$$

Then using the diamond difference auxiliary equation to obtain the east side flux we have:

$$\bar{\psi}_e = 2\bar{\psi}_c - \bar{\psi}_w = \left(\frac{4\mu}{a\Delta x} - 1\right)\bar{\psi}_w. \tag{C.5}$$

Hence for a non-negative outgoing flux we require that:

$$\frac{4\mu}{a\Delta x} > 1 \tag{C.6}$$

$$\frac{4\mu}{\Delta x} > \frac{2\mu}{\Delta x} + \frac{2\eta}{\Delta y} + \Sigma_t \tag{C.7}$$

$$\frac{2\mu}{\Delta x} > \frac{2\eta}{\Delta y} + \Sigma_t \tag{C.8}$$

$$2 > 2\frac{\eta}{\mu}\frac{\Delta x}{\Delta y} + \frac{\Sigma_t \Delta x}{\mu} \tag{C.9}$$

In the case where square cells are used and the value of μ and η are the same and in the first quadrant, (C.9) reduces to:

$$0 > \frac{\Sigma_t \Delta x}{\mu}. \tag{C.10}$$

Each quantity in the term on the right hand side of equation (C.10) is a positive quantity and therefore it is impossible for the RHS of the equation to be smaller than zero. Hence we have a contradiction and so in this case we will have a negative flux no matter what the value of x.

C.2 Box method

Similarly for the box method, using the numbering shown in figure 6.1 for the nodes, we have, for a cell with zero source:

$$\frac{\mu}{2\Delta x}(\psi_3 + \psi_2 - \psi_4 - \psi_1) + \frac{\eta}{2\Delta y}(\psi_4 + \psi_3 - \psi_1 - \psi_2) + \Sigma_t(\psi_1 + \psi_2 + \psi_3 + \psi_4) = 0 \tag{C.11}$$

Assume that the cell considered has a non-zero source in the cell to the west, but that there is a zero boundary condition along the lower boundary of the cell. This means that ψ_1 and ψ_2 are zero and that ψ_3 and ψ_4 will be non-zero.

ψ_3 can also therefore be expressed in terms of ψ_4 as follows:

$$\psi_3 = \psi_4 \left(\frac{\frac{\mu}{2\Delta x} - \frac{\eta}{2\Delta y} - \Sigma_t}{\frac{\mu}{2\Delta x} + \frac{\eta}{2\Delta y} + \Sigma_t} \right) \tag{C.12}$$

Considering the case where $\mu = \eta$ and $\Delta x = \Delta y$, we have:

$$\psi_3 = \psi_4 \left(\frac{-\Sigma_t}{\frac{\mu}{\Delta x} + \Sigma_t} \right) \tag{C.13}$$

The angular cosine, cell width and total cross chapter are all positive and so given a positive incoming flux at 4 it will produce a negative flux at 3 no matter what the cell size.

D. Equivalence of DD1-Hébert scheme with a bi-quadratic box scheme

As previously mentioned in section 6.9, it needs to be shown that the trial functions in the 2nd order box scheme satisfy the auxiliary equations given by the 1st order Hébert scheme. This is sufficient to show that the first 4 Legendre polynomials of the two methods are equivalent.

To make the comparison slightly easier, the normalisation of the Legendre Polynomials used in Hébert's method will be changed such that the reference cell varies from $[-1,+1]$ along each axis and the polynomials (for a given axis) satisfy equation (3.18) when integrated together. Hébert's WR equations then become:

$$\frac{\mu}{\Delta x}(\Phi_e^0 - \Phi_w^0) + \frac{\eta}{\Delta y}(\Phi_n^0 - \Phi_s^0) + \Sigma_t \Phi_c^{00} = Q_c^{00} \tag{D.1}$$

$$\frac{3\mu}{\Delta x}(\Phi_e^0 + \Phi_w^0 - 2\Phi_c^{00}) + \frac{\eta}{\Delta y}(\Phi_n^1 - \Phi_s^1) + \Sigma_t \Phi_c^{10} = Q_c^{10} \tag{D.2}$$

$$\frac{\mu}{\Delta x}(\Phi_e^1 - \Phi_w^1) + \frac{3\eta}{\Delta y}(\Phi_n^0 + \Phi_s^0 - 2\Phi_c^{00}) + \Sigma_t \Phi_c^{01} = Q_c^{01} \tag{D.3}$$

$$\frac{5\mu}{\Delta x}(\Phi_e^1 + \Phi_w^1 - 2\Psi_c^{01}) + \frac{5\eta}{\Delta y}(\Phi_n^1 + \Phi_s^1 - 2\Psi_c^{10}) + \Sigma_t \Phi_c^{11} = Q_c^{11} \tag{D.4}$$

Similarly, the auxiliary equations become:

$$0 = \Phi_e^0 - \Phi_w^0 - 2\Phi_c^{10} \tag{D.5}$$

$$0 = \Phi_e^1 - \Phi_w^1 - 2\Phi_c^{11} \tag{D.6}$$

$$0 = \Phi_n^0 - \Phi_s^0 - 2\Phi_c^{01} \tag{D.7}$$

$$0 = \Phi_n^1 - \Phi_s^1 - 2\Phi_c^{11} \tag{D.8}$$

Φ_n denotes the angular flux on node n of figure 6.13. The bi-quadratic within-cell variation can be represented as a function of the nodal values using Lagrange polynomials as follows:

$$\begin{aligned}\Phi(\zeta,\gamma) =&\frac{1}{4}\zeta\gamma(\zeta-1)(\gamma-1)\Phi_1+\frac{1}{4}\zeta\gamma(\zeta+1)(\gamma-1)\Phi_2+\frac{1}{4}\zeta\gamma(\zeta+1)(\gamma+1)\Phi_3\\ &+\frac{1}{4}\zeta\gamma(\zeta-1)(\gamma+1)\Phi_4+\frac{1}{2}\gamma(1-\zeta^2)(\gamma-1)\Phi_5+\frac{1}{2}\zeta(\zeta+1)(1-\gamma^2)\Phi_6\\ &+\frac{1}{2}\gamma(1-\zeta^2)(\gamma+1)\Phi_7+\frac{1}{2}\zeta(\zeta-1)(1-\gamma^2)\Phi_8+(1-\zeta^2)(1-\gamma^2)\Phi_9\end{aligned} \tag{D.9}$$

This can be rearranged in terms of Legendre polynomials as follows:

$$\begin{aligned}\Phi(\zeta,\gamma) =&\Phi_c^{00}P_{00}(\zeta,\gamma)+\Phi_c^{10}P_{10}(\zeta,\gamma)+\Phi_c^{01}P_{01}(\zeta,\gamma)\\ &+\Phi_c^{11}P_{11}(\zeta,\gamma)+\Phi_c^{20}P_{20}(\zeta,\gamma)+\Phi_c^{02}P_{02}(\zeta,\gamma)\\ &+\Phi_c^{21}P_{21}(\zeta,\gamma)+\Phi_c^{12}P_{12}(\zeta,\gamma)+\Phi_c^{22}P_{22}(\zeta,\gamma)\end{aligned} \tag{D.10}$$

With the cell-central moments defined as follows:

$$\Phi_c^{00}=\frac{1}{36}(\Phi_1+\Phi_2+\Phi_3+\Phi_4+4\Phi_5+4\Phi_6+4\Phi_7+4\Phi_8+16\Phi_9) \tag{D.11}$$

$$\Phi_c^{10}=\frac{1}{12}(-\Phi_1+\Phi_2+\Phi_3-\Phi_4+4\Phi_6-4\Phi_8) \tag{D.12}$$

$$\Phi_c^{01}=\frac{1}{12}(-\Phi_1-\Phi_2+\Phi_3+\Phi_4-4\Phi_5+4\Phi_7) \tag{D.13}$$

$$\Phi_c^{11}=\frac{1}{4}(\Phi_1-\Phi_2+\Phi_3-\Phi_4) \tag{D.14}$$

$$\Phi_c^{20}=\frac{1}{18}(\Phi_1+\Phi_2+\Phi_3+\Phi_4-2\Phi_5+4\Phi_6-2\Phi_7+4\Phi_8-8\Phi_9) \tag{D.15}$$

$$\Phi_c^{02}=\frac{1}{18}(\Phi_1+\Phi_2+\Phi_3+\Phi_4+4\Phi_5-2\Phi_6+4\Phi_7-2\Phi_8-8\Phi_9) \tag{D.16}$$

$$\Phi_c^{21}=\frac{1}{6}(-\Phi_1-\Phi_2+\Phi_3+\Phi_4+2\Phi_5-2\Phi_7) \tag{D.17}$$

$$\Phi_c^{12}=\frac{1}{6}(-\Phi_1+\Phi_2+\Phi_3-\Phi_4-2\Phi_6+2\Phi_8) \tag{D.18}$$

$$\Phi_c^{22}=\frac{1}{9}(\Phi_1+\Phi_2+\Phi_3+\Phi_4+2\Phi_5+2\Phi_6+2\Phi_7+2\Phi_8+4\Phi_9) \tag{D.19}$$

Cell edge variation can then be represented as:

$$\Phi_w(\gamma) = (\Phi_c^{00} - \Phi_c^{10} + \Phi_c^{20})P_0(\gamma) + (\Phi_c^{01} - \Phi_c^{11} + \Phi_c^{21})P_1(\gamma) + (\Phi_c^{02} - \Phi_c^{12} + \Phi_c^{22})P_2(\gamma) \quad \text{(D.20)}$$

$$\Phi_e(\gamma) = (\Phi_c^{00} + \Phi_c^{10} + \Phi_c^{20})P_0(\gamma) + (\Phi_c^{01} + \Phi_c^{11} + \Phi_c^{21})P_1(\gamma) + (\Phi_c^{02} + \Phi_c^{12} + \Phi_c^{22})P_2(\gamma) \quad \text{(D.21)}$$

$$\Phi_s(\zeta) = (\Phi_c^{00} - \Phi_c^{01} + \Phi_c^{02})P_0(\zeta) + (\Phi_c^{10} - \Phi_c^{11} + \Phi_c^{12})P_1(\zeta) + (\Phi_c^{20} - \Phi_c^{21} + \Phi_c^{22})P_2(\zeta) \quad \text{(D.22)}$$

$$\Phi_n(\zeta) = (\Phi_c^{00} + \Phi_c^{01} + \Phi_c^{02})P_0(\zeta) + (\Phi_c^{10} + \Phi_c^{11} + \Phi_c^{12})P_1(\zeta) + (\Phi_c^{20} + \Phi_c^{21} + \Phi_c^{22})P_2(\zeta) \quad \text{(D.23)}$$

Thus the constant and linear cell edge values would be given by:

$$\Phi_w^0 = \Phi_c^{00} - \Phi_c^{10} + \Phi_c^{20} \quad \text{(D.24)}$$

$$\Phi_e^0 = \Phi_c^{00} + \Phi_c^{10} + \Phi_c^{20} \quad \text{(D.25)}$$

$$\Phi_s^0 = \Phi_c^{00} - \Phi_c^{01} + \Phi_c^{02} \quad \text{(D.26)}$$

$$\Phi_n^0 = \Phi_c^{00} + \Phi_c^{01} + \Phi_c^{02} \quad \text{(D.27)}$$

$$\Phi_w^1 = \Phi_c^{01} - \Phi_c^{11} + \Phi_c^{21} \quad \text{(D.28)}$$

$$\Phi_e^1 = \Phi_c^{01} + \Phi_c^{11} + \Phi_c^{21} \quad \text{(D.29)}$$

$$\Phi_s^1 = \Phi_c^{10} - \Phi_c^{11} + \Phi_c^{12} \quad \text{(D.30)}$$

$$\Phi_n^1 = \Phi_c^{10} + \Phi_c^{11} + \Phi_c^{12} \quad \text{(D.31)}$$

Equations (D.24) to (D.31) can be rearranged to give the auxiliary equations provided by Hébert's scheme, and thus both methods give equivalent Legendre polynomial coefficients for the first 4 Legendre polynomials ($\Phi_c^{00}, \Phi_c^{10}, \Phi_c^{01}, \Phi_c^{11}$). Setting the boundary conditions correctly in Hébert's scheme results in the zeroth and first order edge values being equivalent to those given by the box scheme.

Equations (D.24)-(D.31) provide a means of combining the edge values and the cell-central moments calculated in Hébert's scheme to give the next 4 cell central Legendre polynomial coefficients: $\Phi_c^{20}, \Phi_c^{02}, \Phi_c^{21}, \Phi_c^{12}$. However, Φ_c^{22} cannot be calculated from the unknowns solved for in Hébert's scheme, because this information has been averaged out.