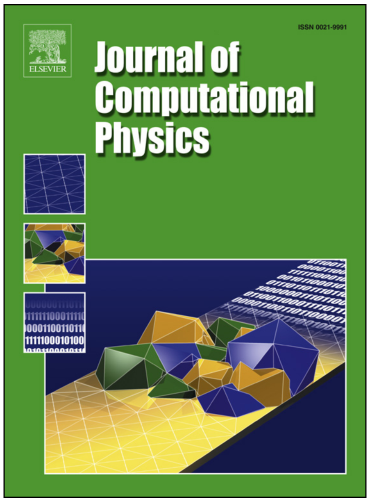

Discontinuous isogeometric analysis methods for the first-order form of the neutron transport equation with discrete ordinate (S_N) angular discretisation

A.R. Owens, J.A. Welch, J. Kópházi, M.D. Eaton

PII:	S0021-9991(16)30025-0
DOI:	http://dx.doi.org/10.1016/j.jcp.2016.03.060
Reference:	YJCPH 6521
To appear in:	*Journal of Computational Physics*
Received date:	21 October 2015
Revised date:	22 March 2016
Accepted date:	27 March 2016

Please cite this article in press as: A.R. Owens et al., Discontinuous isogeometric analysis methods for the first-order form of the neutron transport equation with discrete ordinate (S_N) angular discretisation, *J. Comput. Phys.* (2016), http://dx.doi.org/10.1016/j.jcp.2016.03.060

Discontinuous Isogeometric Analysis Methods for the First-Order form of the Neutron Transport Equation with Discrete Ordinate (S_N) Angular Discretisation

A.R. Owens, J.A. Welch, J. Kópházi, M.D. Eaton

Nuclear Engineering Group, Department of Mechanical Engineering, City and Guilds Building, Imperial College London, Exhibition Road, South Kensington, London, SW7 2AZ, United Kingdom

Abstract

In this paper two discontinuous Galerkin isogeometric analysis methods are developed and applied to the first-order form of the neutron transport equation with a discrete ordinate (S_N) angular discretisation. The discontinuous Galerkin projection approach was taken on both an element level and the patch level for a given Non-Uniform Rational B-Spline (NURBS) patch. This paper describes the detailed dispersion analysis that has been used to analyse the numerical stability of both of these schemes. The convergence of the schemes for both smooth and non-smooth solutions was also investigated using the method of manufactured solutions (MMS) for multidimensional problems and a 1D semi-analytical benchmark whose solution contains a strongly discontinuous first derivative. This paper also investigates the challenges posed by strongly curved boundaries at both the NURBS element and patch level with several algorithms developed to deal with such cases. Finally numerical results are presented both for a simple pincell test problem as well as the C5G7 quarter core MOX/UOX small Light Water Reactor (LWR) benchmark problem. These numerical results produced by the isogeometric analysis (IGA) methods are compared and contrasted against linear and quadratic discontinuous Galerkin finite element (DGFEM) S_N based methods.

Keywords: Isogeometric, Neutron Transport, Hyperbolic Equations, Discontinuous Galerkin, Discrete Ordinates, Reactor Physics

1. Introduction

Solving the neutron transport equation numerically over heterogeneous PWR models remains a challenge to the reactor physics community. This is due in large part to the seven-dimensional phase space of the problem, with solution fidelity required in energy, direction and space, as well as time in transient problems. Traditional techniques for solving the transport equation involve homogenisation [1] [2] of the complex spatial features into cartesian grids, on which either the diffusion equation can be solved or diamond differencing methods [3] can be used.

If the geometric features are to be modelled directly, DGFEM is a commonly used approach. DGFEM is a now standard numerical discretisation technique that has been applied to a diverse variety of physical phenomena from radiation transport [4] [5] and heat transfer to fluid dynamics [6]. DGFEM was originally developed and applied to the discretisation of the first-order form of the neutron transport equation with a discrete ordinate angular discretisation [5]. The solutions of the neutron transport equation can produce both singular transport dominated solutions, as well as smooth and diffusive solutions ($\geq C^0$ continuous). Although DGFEM can exhibit oscillatory solutions in some extreme regimes, such as material interfaces with optically thick cells, these oscillations are bounded [6].

One of the principle advantages of the finite element method is that it can model a wide range of complex geometries using basic geometric primitives such as triangles and quadrilaterals in two dimensions and pyramids, prisms, hexahedra and tetrahedra in three dimensions [7]. Finite element based technologies rely upon the use of automatic

Email address: a.owens12@imperial.ac.uk (A.R. Owens)

mesh generation software that take underlying computer aided design (CAD) descriptions, such as NURBS, that describe the surfaces bounding solid objects, and discretise them into the aforementioned basic geometric primitives [7]. However, there are several issues and problems that arise through the use of automatic mesh generators. One of the major issues is that the underlying geometry is often not exactly reproduced when it is discretised into basic geometric primitives such as triangles or tetrahedra as these usually are linear elements with straight or planar sides [7]. Another issue is that NURBS patches will often have gaps between them which require an automatic CAD repair programme that will generate a NURBS description suitable for use by automatic mesh generators [8]. The preservation of geometry is of critical importance when dealing with reactor physics problems as the polygonal approximations to the geometry produced by finite element meshes must preserve the fissile mass or the criticality solutions will be inaccurate [9] [10]. Usually this is remedied by locally modifying the mesh to preserve the fissile mass but this can be problematic for very complex geometries. The use of polyhedra with straight or planar surfaces means that the NURBS surfaces are not exactly represented in the resulting mesh. Even higher-order spectral elements cannot represent all NURBS surfaces exactly and there are also many issues associated with generating higher-order finite element meshes [11].

Recently, a new numerical discretisation technique has been developed which makes use of the underlying NURBS representation typically found in CAD software packages. This numerical discretisation method is called isogeometric analysis. IGA is similar in form to the finite element method (FEM) [6] and Spectral Element Method [12] in that a discretisation of the weak form of a partial differential equation is used to form a matrix system of equations. However, instead of using standard Lagrange or spectral basis basis functions for the shape and test functions used to discretise the weak form of the PDE, the underlying NURBS basis functions are used instead [13]. The advantages of this approach are manifold and include; improved continuity ($\geq C^0$ continuous), exact representation of the underlying NURBS geometry on the coarsest level of discretisation, as well as exact representation of the geometry as the mesh is refined [13].

So far IGA has been applied to a wide variety of different physical phenomena, including computational solid dynamics problems [13], computational fluid dynamics [13] and coupled solid-fluid interaction problems [14]. A variety of different IGA discretisations have been developed. Initially these were restricted to Bubnov-Galerkin type IGA discretisations for elliptic systems [13]. However, for the advection-diffusion problems, associated with solving the Navier-Stokes equations, variational multiscale IGA discretisations have been developed [13]. More recently, discontinuous IGA methods have been developed for elliptic systems of equations [15].

As yet, no discontinuous IGA methods have been developed for hyperbolic systems of equations. While this paper focuses on neutron transport problems, the spatial discretisations presented are equally applicable to other hyperbolic systems, particularly those that already use DGFEM. Examples of such systems include the Euler equations of gas dynamics [16], the compressible Navier Stokes equations [17] and radiative heat transfer [6].

Following on from Hall et. al's work using IGA with the diffusion equation [18], this work investigates the application of this spatial discretisation to the first order form of the discrete ordinates equations. The remainder of this paper is organized as follows. Section 2 explains the underlying concepts of isogeometric analysis methods and the NURBS basis functions the method uses. Section 3 derives the discrete ordinate equations from the neutron transport equation as well as the weak form of the equation that is to be spatially discretized using the IGA method. The two different shape and test function spaces over which the discontinuous Galerkin isogeometric method discretises the weak form of the angularly discretised neutron transport equation are also described in this section. Section 4 contains the theory behind the dispersion and diffusion analysis of both schemes for a simple one-dimensional homogeneous problem. Section 5 contains the graph theory analysis that is required in order to select the correct sweep ordering for the elements in the sweep based solution algorithm. It also explains two approaches for dealing with cycles as well as the behaviour of the schemes when an average normal vector is used on element boundaries. Section 6 presents numerical results of applying the resulting discontinuous Galerkin isogeometric analysis method to a variety of verification test cases.

2. Isogeometric Analysis

Isogeometric Analysis was originally introduced as a means of developing discretisation methods that were closely related to the underlying geometrical representation within the CAD software [13]. The objective of the development of IGA is to seamlessly link the geometrical description and analysis of engineering problems in order to reduce the time required in the design and analysis cycle. The largest proportion of the design and analysis cycle is usually taken up by developing analysis suitable NURBS representations that are free of gaps which can then be meshed using finite element mesh generators. This is particularly true if the solution method requires adaptivity in the spatial mesh. If there are curved boundaries in the geometry, the CAD model may have to be revisited every time the mesh changes to generate a new one, or if the solution domain varies in time for the same reason. The mesh generation process itself is also very challenging for complex domains. In addition, the resulting mesh often does not exactly represent the underlying NURBS geometry. This can cause issues in reactor physics problems, as the conservation of the fissile mass in the system is crucial to obtaining an accurate solution [9].

IGA closes the gap between the geometrical description and analysis of engineering problems by utilizing the NURBS representation found in CAD software in the discretisation of the weak form of the governing equations for the physical phenomena being modelled. As IGA utilizes NURBS, the method can represent a wide variety of geometrical shapes exactly, such as circles and ellipses in two-dimensions; and spheres, cylinders and other quadric surfaces in three-dimensions [19]. These are all geometric primitives that are typically found in reactor physics and shielding problems. Even higher-order finite element representations and spectral elements have issues representing the wide variety of shapes that can be described using NURBS [11] [7]. Another important advantage of IGA is that it does not depend upon an ancillary mesh generator as long as an analysis suitable NURBS representation of the geometry is given. IGA also exactly represents the underlying geometry at the coarsest level of discretisation, which is unlike FEM that requires further refinement of the mesh to represent the geometry more accurately. However, it is important to note that creating solid NURBS objects from the surface NURBS descriptions generated by CAD programs is still an active area of research. Finally, another important advantage of IGA is that parameterisation of the physical space from the parametric space does not change under h or p refinement, allowing local adaptive refinement to be performed more easily [13].

2.1. B-Splines

B-Spline curves are completely defined by 3 properties [19]:

1. A polynomial order p. Here we will follow the convention in Reference [13] that order and degree are synonymous, as in the finite element literature.
2. A set of n control points $\{\mathbf{B}_i : i = 1, ..., n\}$, where n is the number of basis functions defining the B-Spline curve.
3. A knot vector Ξ, which is simply a sequence of non-decreasing real numbers. The length of a knot vector is defined to be the number of values, including repeats, in Ξ . Open knot vectors will be used in the remainder of this paper. For open knot vectors the first and last knot value are repeated $p + 1$ times. In this case, the knot vector will be of length $n + p + 1$, $\Xi = (\xi_1, \xi_2, ..., \xi_{n+p+1})$.

Together these properties define a one-dimensional B-Spline patch. A knot vector is said to be uniform when the knot points are equally spaced, and non-uniform otherwise [19]. The basis functions in parametric space are completely defined by their polynomial order and knot vector. The knots in the knot vector partition the B-Spline patch into knot spans. The knot spans of the knot vector are similar in concept to traditional “elements” in FEM, in that they are points of reduced continuity. If a knot point appears only once in a knot vector then it is called a simple knot and the continuity of the B-Spline basis functions is C^{p-1} at this point. If a knot point occurs m times where $m > 1$ then the knot point is a called a multiple knot of multiplicity m. Every additional knot point inserted at a pre-existing knot point reduces the degree of continuity at that repeated knot value by one; therefore the continuity of the B-Spline basis functions at a given knot point becomes C^{p-m} [19]. If basis function i of polynomial order p is denoted by $N_{i,p}(\xi)$, then the basis functions are given by the following Cox-de Boor recursion formula [20] [21] for $p = 0$:

$$N_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1}. \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

and for $p \geq 1$

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \tag{2}$$

These basis functions form a partition of unity at all values of ξ within the knot vector [19], that is:

$$\sum_{i=1}^{n} N_{i,p}(\xi) = 1 \quad \forall \xi \in [\xi_1, \xi_{n+p+1}] \tag{3}$$

They also have compact support, with each basis function spanning at most $p + 1$ elements. In addition, the basis functions are interpolatory at the end points of the knot vector, such that $N_{i,p}(\xi_1) = \delta_{i,1}$ and $N_{i,p}(\xi_{n+p+1}) = \delta_{i,n}$. The basis functions are also pointwise non-negative, which means that every term in a mass matrix is also non-negative, which can be useful for mass lumping schemes [22].

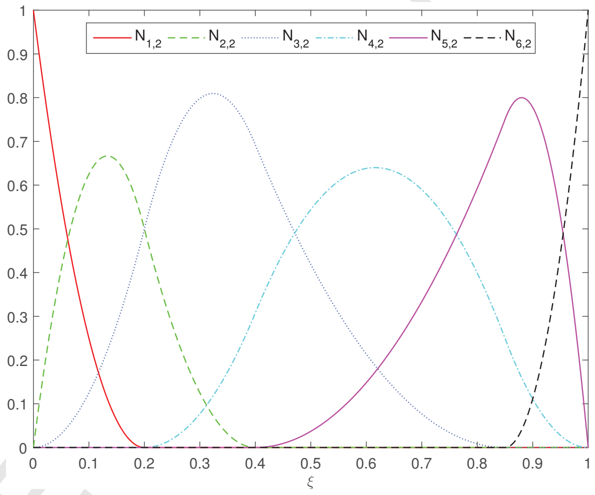

Figure 1: Quadratic B-Splines with open knot vector $\Xi = (0, 0, 0, 0.2, 0.4, 0.85, 1, 1, 1)$. The basis functions clearly span multiple elements (knot spans).

Figure 1 shows a graph of the quadratic B-Spline basis functions formed from a prescribed knot vector. The graph illustrates that B-Spline basis functions typically have support over multiple knot spans i.e. the support of $N_{i,p}(\xi)$ is the interval $[\xi_i, \xi_{i+p+1}]$. The B-Spline basis functions are always C^∞ continuous within knot spans. However, as stated above the internal knot points can be repeated in order to reduce the continuity of the basis function at the repeated knot point in parametric space to C^{p-m}. Figure 2 shows an example of quadratic B-Spline basis functions with a prescribed knot vector which has a repeated internal knot point at $\xi = 0.4$. If the multiplicity of the knot point $m = p$, the B-Spline basis functions are C^0 continuous at that knot point. If the multiplicity of the knot point is $m = p + 1$ the B-Spline basis functions are discontinuous at that knot point and as a result two separate B-Splines patches are formed.

The derivatives of B-Splines can also be represented in terms of lower order B-Splines due to the recursive nature of their construction as follows [19]:

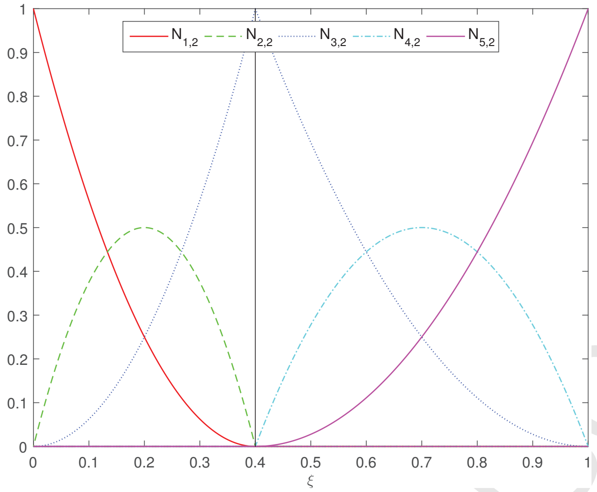

Figure 2: Quadratic B-Splines with reduced continuity at $\xi = 0.4$. The continuity could be C^0, with $\Xi = (0, 0, 0, 0.4, 0.4, 1, 1, 1)$, representing a continuous solution at $\Xi = 0.4$ (this is the basis function numbering used in the legend). Alternatively, the continuity could be C^{-1}, in which case there will be two separate patches, with knot vectors $\Xi = (0, 0, 0, 0.4, 0.4, 0.4)$ and $\Xi = (0.4, 0.4, 0.4, 1, 1, 1)$ respectively. In this case the numerical solution would be upwinded across the boundary $\xi = 0.4$ depending on the direction of neutron travel.

$$\frac{d}{d\xi}N_{i,p}(\xi) = \frac{p}{\xi_{i+p} - \xi_i}N_{i,p-1}(\xi) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}}N_{i+1,p-1}(\xi) \tag{4}$$

Equations 2 and 4 can give rise to terms of the form $0/0$, in which case the term is defined to be zero.

B-Spline curves in real space are then defined by giving each basis function a corresponding control point and summing the products of the two as follows:

$$\mathbf{C}(\xi) = \sum_{i=1}^{n} N_{i,p}(\xi)\mathbf{B}_i \tag{5}$$

Higher dimensional B-Spline objects are constructed in a tensor product fashion from one dimensional B-Splines. In two dimensions, polynomial orders p and q, knot vectors $\Xi = (\xi_1, \xi_2, ..., \xi_{n+p+1})$ and $H = (\eta_1, \eta_2, ..., \eta_{m+q+1})$ and a control net $\{\mathbf{B}_{i,j} : i = 1, ...n \text{ and } j = 1, ..., m\}$ are used to construct B-Spline surfaces as follows:

$$\mathbf{S}(\xi, \eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} N_{i,p}(\xi)M_{j,q}(\eta)\mathbf{B}_{i,j} \tag{6}$$

where the $N_{i,p}$ are as above and the $M_{j,q}$ are defined in the same way using polynomial order q and knot vector H. In three dimensions, we add another parametric direction with polynomial order r, knot vector $Z = (\zeta_1, \zeta_2, ..., \zeta_{l+r+1})$ and changing to a control lattice $\{\mathbf{B}_{i,j,k} : i = 1, ..., n,\ j = 1, ..., m \text{ and } k = 1, ..., l\}$ to get the volume:

$$\mathbf{V}(\xi, \eta, \zeta) = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{l} N_{i,p}(\xi)M_{j,q}(\eta)L_{k,r}(\zeta)\mathbf{B}_{i,j,k} \tag{7}$$

The partition of unity, compact support and pointwise positivity of the basis functions extends into two and three dimensions as a direct result of the one dimensional case combined with the tensor product structure of the two and three dimensional basis functions.

2.2. NURBS

NURBS are constructed from B-Splines by associating a weight with each control point, denoted by w_i, $w_{i,j}$ and $w_{i,j,k}$ in one, two or three dimensions respectively. One dimensional NURBS basis functions of order p are then defined by [19]:

$$R_i^p(\xi) = \frac{N_{i,p}(\xi)w_i}{\sum_{\hat{i}=1}^{n} N_{\hat{i},p}(\xi)w_{\hat{i}}} \tag{8}$$

with NURBS curves given by:

$$\mathbf{C}(\xi) = \sum_{i=1}^{n} R_i^p(\xi)\mathbf{B}_i \tag{9}$$

The rational part of the name NURBS comes from the division by B-Splines. Two and three dimensional basis functions are then defined as follows, with the surfaces and solids constructed from them derived analogously to equation 9:

$$R_{i,j}^{p,q}(\xi,\eta) = \frac{N_{i,p}(\xi)M_{j,q}(\eta)w_{i,j}}{\sum_{\hat{i}=1}^{n}\sum_{\hat{j}=1}^{m} N_{\hat{i},p}(\xi)M_{\hat{j},q}(\eta)w_{\hat{i},\hat{j}}} \tag{10a}$$

$$R_{i,j,k}^{p,q,l}(\xi,\eta,\zeta) = \frac{N_{i,p}(\xi)M_{j,q}(\eta)L_{k,r}(\zeta)w_{i,j,k}}{\sum_{\hat{i}=1}^{n}\sum_{\hat{j}=1}^{m}\sum_{\hat{k}=1}^{l} N_{\hat{i},p}(\xi)M_{\hat{j},q}(\eta)L_{\hat{k},r}(\zeta)w_{\hat{i},\hat{j},\hat{k}}} \tag{10b}$$

2.3. Physical, Parametric and Parent Spaces

For analysis purposes, numerical integration must be performed over knot spans, which are equivalent to elements in FEA. As in FEA, these elements in parametric space are mapped from a canonical reference element , which in IGA is a $[-1,1]^d$ line, square or cube in d dimensions [13]. This reference element is then mapped to the parametric elements, formed by the tensor products of the knot spans in each parametric dimension. These parametric elements are then mapped to the physical space, forming the mesh for analysis purposes as demonstrated in Figure 3. Gaussian quadrature rules using $p+1$, $q+1$ and $r+1$ abscissae are used in each parametric direction to construct local mass matrices and other similar objects. If the weights defining the NURBS are all equal then within each element the basis functions will be polynomials and this quadrature will integrate them exactly. If the functions are rational then this quadrature will not be integrating the basis functions exactly. However, the denominator from equations 8, 10a and 10b does not change under mesh refinement or order elevation. The Jacobian of the mapping from parametric space to real space is also invariant with regard to mesh refinement or order elevation. Therefore as the mesh is refined, the rational basis functions are better approximated by non-rational polynomials, and the errors introduced by this inexact integration rapidly become negligible [23].

3. Discrete Ordinates & Weak Form

The steady state neutron transport equation with multiplication factor k_{eff} and isotropic scattering and fission sources is given by [24]:

$$\mathbf{\Omega}\cdot\nabla\psi(\mathbf{r},\mathbf{\Omega},E) + \Sigma_t(\mathbf{r},E)\psi(\mathbf{r},\mathbf{\Omega},E) = Q(\mathbf{r},E) \tag{11}$$

where $\mathbf{r}$ is a point in the physical domain V, $\mathbf{\Omega}$ is the direction of neutron flight given by a vector on the unit sphere, E is energy, ψ is the angular neutron flux and Σ_t is the macroscopic absorption cross-section.

The source term $Q(\mathbf{r},E)$ is given by:

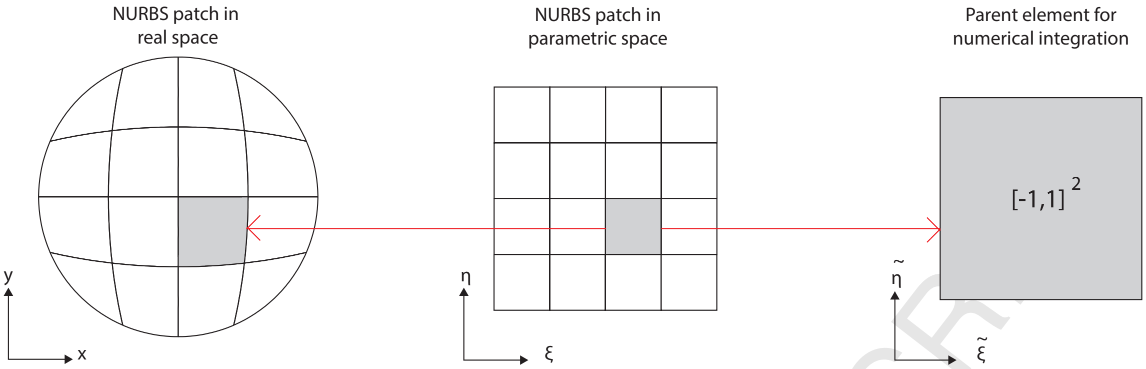
NURBS patch in real space
NURBS patch in parametric space
Parent element for numerical integration
[-1,1] 2
y
x
η
ξ
η̃
ξ̃

Figure 3: An example of a circular NURBS patch in real space, parametric space, and the parent element used to perform numerical integration.

$$Q(\mathbf{r},E) = \int_0^\infty dE'\Sigma_s(\mathbf{r},E'\to E)\phi(\mathbf{r},E') + \frac{\chi(E)}{k_{eff}}\int_0^\infty dE'\nu(E')\Sigma_f(\mathbf{r},E')\phi(\mathbf{r},E') \tag{12}$$

where $\Sigma_s(\mathbf{r},E'\to E)$ is the macroscopic scattering cross-section from energy E' to energy E, $\chi(E)$ is the neutron emission spectrum due to fission, $\nu(E)$ is the average number of neutrons emitted per fission at energy E and $\Sigma_f(\mathbf{r},E)$ is the macroscopic fission cross-section.

The energy dependence of the equation is dealt with by the multigroup method in which the energy dependence in equations 11 and 12 is split into discrete energy groups. The multigroup equations can then be written as:

$$\mathbf{\Omega}\cdot\nabla\psi^g + \Sigma_t^g\psi^g = \sum_{g'=1}^{G}\Sigma_s^{g'\to g}\phi^{g'} + \frac{\chi_g}{K_{eff}}\sum_{g'=1}^{G}\nu^{g'}\Sigma_f^{g'}\phi^{g'} \quad \text{for g=1,...,G} \tag{13a}$$

$$\phi^g(\mathbf{r}) = \int_{4\pi} d\mathbf{\Omega}\psi^g(\mathbf{r},\mathbf{\Omega}) \tag{13b}$$

where ϕ^g, Σ_t^g, $\Sigma_s^{g'\to g}$, χ_g, ν^g and Σ_f^g are now all purely functions of space $\mathbf{r}$, while ψ^g is a function of space and angle $\mathbf{\Omega}$. The discrete ordinates angular discretisation then splits the angular dependence of the multigroup equations into directions $\mathbf{\Omega}_k$, along which equation 13a is solved for the $\psi_k^g(\mathbf{r}) = \psi^g(\mathbf{r},\mathbf{\Omega}_k)$, with k ranging from 1 to N. Each ordinate also has a corresponding quadrature weight w_k for performing angular integrals, which gives an approximation to the group scalar flux:

$$\phi^g(\mathbf{r}) \approx \sum_{k=1}^{N} w_k\psi_k^g(\mathbf{r}) \tag{14}$$

As the projection and solution methods are analogous, the weak form of the one group discrete ordinates equations will be derived here. Dropping the g superscripts, these are given by:

$$\mathbf{\Omega}_k\cdot\nabla\psi_k(\mathbf{r}) + \Sigma_t(\mathbf{r})\psi_k(\mathbf{r}) = \frac{\Sigma_s(\mathbf{r})}{4\pi}\phi(\mathbf{r}) + \frac{1}{k_{eff}}\frac{\nu(\mathbf{r})\Sigma_f(\mathbf{r})}{4\pi}\phi(\mathbf{r}) \tag{15}$$

for $k = 1,...,N$. Two different discontinuous Galerkin spatial discretisations were then applied to these equations. In the first, the basis functions were discontinuous between every element in the mesh, similar to a DGFEM approach. This method will be referred to as the Fully-Discontinuous Galerkin or FDG approach. In the second, the basis functions are continuous within each patch, and discontinuous only on patch boundaries. This method will be referred to as the Patch-Discontinuous Galerkin or PDG approach. See Figure 4 for a graphical representation of the difference. The rationale behind this discretisation is that one of the causes of spurious numerical artefacts in spatial discretisations of the discrete ordinates equations is the presence of adjacent materials that have significant differences in material properties e.g. scattering materials adjacent to highly absorbing materials. Therefore, allowing the solution to be discontinuous at these points could provide improved numerical robustness and reduce the magnitude of these spurious numerical artefacts. In the following discussion, element will be used to refer to an element in the FDG method and a patch in the PDG method.

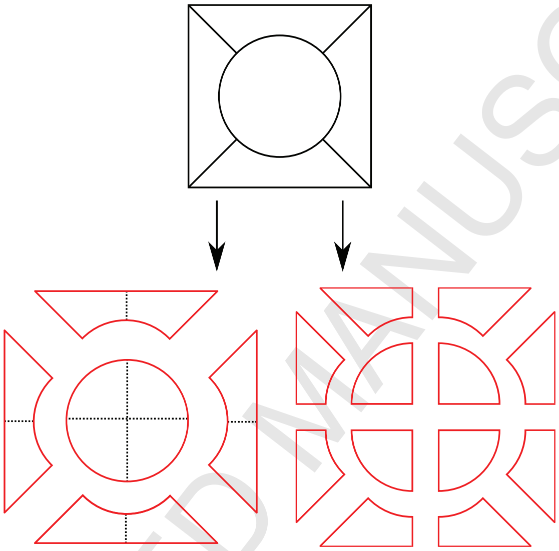

Figure 4: The original CAD geometry (top), along with PDG (left) and FDG (right) meshes after one knot insertion in each direction in the circle. The basis functions are discontinuous across the red lines, while dashed black lines represent knot insertions across which the solution is still continuous.

As with traditional DGFEM, equation 15 is multiplied by a test function $v(\mathbf{r}) \in H^1(V_e)$ and then integrated over a subset of the spatial domain V_e. In the PDG scheme, V_e is the entire patch. This patch may have had knot insertions so that it contains multiple elements, but the solution is at least C^0 everywhere in V_e. In the FDG scheme V_e is an individual element in which the solution is C^∞ everywhere, as with DGFEM. For brevity V_e will be referred to as an element for the remainder of the section.

$$\int_{V_e} d\mathbf{r} v \boldsymbol{\Omega}_k \cdot \nabla\psi_k + \int_{V_e} d\mathbf{r}\Sigma_t\psi_k v = \frac{1}{4\pi}\left(\int_{V_e} d\mathbf{r}\Sigma_s\phi v + \frac{1}{K_{eff}}\int_{V_e} d\mathbf{r} v\Sigma_f\phi v\right) \tag{16}$$

The divergence theorem is then applied to the first term of equation 16, yielding:

$$\int_{V_e} d\mathbf{r} v \boldsymbol{\Omega}_k \cdot \nabla\psi_k = \int_{\Gamma_e} dS\, v\psi_k \boldsymbol{\Omega}_k \cdot \mathbf{n} - \int_{V_e} d\mathbf{r}\psi_k \boldsymbol{\Omega}_k \cdot \nabla v \tag{17}$$

where Γ_e is the boundary of element V_e, and $\mathbf{n}$ is the outward pointing unit normal to this boundary. Defining the two inner products $(u, v) := \int_{V_e} d\mathbf{r} u(\mathbf{r})v(\mathbf{r})$ and $\langle u, v\rangle := \int_{\Gamma_e} dS\, u(s)v(s)$, combining this notation with equation 16, substituting in 17 and utilising the fact that the cross-sections and external source term are constant within an element, we get:

$$-\mathbf{\Omega}_k \cdot (\psi_k, \nabla v) + \Sigma_t(\psi_k, v) + \mathbf{\Omega}_k \cdot \langle \psi_k, v\mathbf{n} \rangle = \frac{1}{4\pi}\left[\Sigma_s(\phi, v) + \frac{1}{K_{eff}}\nu\Sigma_f(\phi, v)\right] \tag{18}$$

To deal with the boundary integral term, it is necessary to split the element boundary Γ_e into inflow and outflow portions; the inflow boundary, where $\mathbf{\Omega}_k \cdot \mathbf{n} < 0$ is denoted by Γ_e^-, while the outflow boundary where $\mathbf{\Omega}_k \cdot \mathbf{n} > 0$ is denoted by Γ_e^+. By defining two additional inner products $\langle u, v\rangle^+ = \int_{\Gamma_e^+} dS\, u(s)v(s)$ and $\langle u, v\rangle^- = \int_{\Gamma_e^-} dS\, u(s)v(s)$, the boundary integral term can be rewritten as:

$$\mathbf{\Omega}_k \cdot \langle \psi_k, v\mathbf{n} \rangle = \mathbf{\Omega}_k \cdot \langle \psi_k, v\mathbf{n} \rangle^+ + \mathbf{\Omega}_k \cdot \langle \psi_k, v\mathbf{n} \rangle^- \tag{19}$$

As in traditional DGFEM S_N [4], the outflow integral over Γ_e^+ is performed using the flux from within the element in the FDG scheme or patch in the PDG scheme. The inflow integral over Γ_e^- is fully upwinded using the flux ψ_{adj} from the adjacent element in the FDG scheme or patch in the PDG scheme. If Γ_e^- is on the physical domain boundary, ψ_{adj} is instead determined by the appropriate boundary condition. Equation 18 can then be rewritten as:

$$-\mathbf{\Omega}_k \cdot (\psi_k, \nabla v) + \Sigma_t(\psi_k, v) + \mathbf{\Omega}_k \cdot \langle \psi_k, v\mathbf{n} \rangle^+ = \frac{1}{4\pi}\left(\Sigma_s(\phi, v) + \frac{1}{K_{eff}}\nu\Sigma_f(\phi, v)\right) - \mathbf{\Omega}_k \cdot \langle \psi_{adj}, v\mathbf{n} \rangle^- \tag{20}$$

In traditional DGFEM, element faces are either planar, or a lumped normal vector is used so that they can be treated as such, in which case $\Gamma_e^\pm$ will be the union of whole faces of the element. The strongly curved element boundaries in IGA mean that $\mathbf{\Omega}_k$ can be tangent to an element face, in which case $\Gamma_e^\pm$ will now contain partial element faces. The strategy employed to deal with integration over these partial element faces is explained in Section 5.

A solution $\psi_k \in H^1(V_e)$ is then sought such that equation 20 holds $\forall v \in H^1(V_e)$. The isoparametric concept is followed, in which the solution representation uses the basis functions that represent the geometry. A Galerkin projection is then performed within the element, such that:

$$v(\mathbf{r}) = R_A(\mathbf{r}) \quad \text{for A = 1,...,nnp} \tag{21a}$$

$$\psi_k(\mathbf{r}) = \sum_{B=1}^{nnp} \psi_k^B R_B(\mathbf{r}) \tag{21b}$$

$$\phi(\mathbf{r}) = \sum_{B=1}^{nnp} \phi^B R_B(\mathbf{r}) \quad \text{where} \tag{21c}$$

$$\phi^B = \sum_{k=1}^{N} w_i \psi_k^B \tag{21d}$$

where there are *nnp* basis functions within the element. The weight functions R_A and the trial functions R_B are taken from the same space, the NURBS used to describe the geometry. Defining mass, streaming and boundary matrices such that:

$$\underline{\underline{M}}_{\{A,B\}} = (R_B, R_A) \tag{22a}$$

$$\underline{\underline{S}}_{x\,\{A,B\}} = \left(R_B, \frac{\partial R_A}{\partial x}\right) \tag{22b}$$

$$\underline{\underline{S}}_{y\,\{A,B\}} = \left(R_B, \frac{\partial R_A}{\partial y}\right) \tag{22c}$$

$$\underline{\underline{P}}_{k\,\{A,B\}} = \mathbf{\Omega}_k \cdot \langle R_B, R_A \mathbf{n}\rangle^+ \tag{22d}$$

and defining solution vectors $\boldsymbol{\psi}_k$ and $\boldsymbol{\phi}$ in the obvious way from equations 21b and 21c respectively, equation 20 can be written in the compact form:

$$\left[-\mu_k \underline{\underline{S}}_x - \eta_k \underline{\underline{S}}_y + \Sigma_t \underline{\underline{M}} + \underline{\underline{P}}_k\right]\boldsymbol{\psi}_k = \frac{1}{4\pi}\left(\Sigma_s + \frac{\nu\Sigma_f}{K_{eff}}\right)\underline{\underline{M}}\boldsymbol{\phi} - \boldsymbol{\psi}_{adj} \tag{23}$$

where $\mathbf{\Omega}_k = (\mu_k, \eta_k)$ in two spatial dimensions and $\boldsymbol{\psi}_{adj\{A\}} = \mathbf{\Omega}_k \cdot \langle \psi_{adj}, R_A \mathbf{n}\rangle^-$. Defining:

$$\underline{\underline{L}} := -\mu_k \underline{\underline{S}}_x - \eta_k \underline{\underline{S}}_y + \Sigma_t \underline{\underline{M}} + \underline{\underline{P}}_k \tag{24a}$$

$$\implies \underline{\underline{L}}\boldsymbol{\psi}_k = \frac{1}{4\pi}\left(\Sigma_s + \frac{\nu\Sigma_f}{K_{eff}}\right)\underline{\underline{M}}\boldsymbol{\phi} - \boldsymbol{\psi}_{adj} \tag{24b}$$

This matrix equation can then be solved for each element in the FDG scheme or patch in the PDG scheme.

4. Dispersion and Diffusion Analysis

A key question when assessing the stability of a spatial discretisation is the ability of that scheme to propagate a travelling wave solution. In the highly heterogeneous geometries found in reactor physics it is difficult to perform such analysis. However the spatial discretisations presented in this paper could be readily applied to problems where this is not the case, such as the Euler equations of gas dynamics [16] and so the analysis is included here. Differences from the true solution appear in two forms:

- Dispersive errors, in which the numerical speed of the wave differs from the true solution.
- Diffusive errors, in which the wave becomes “smeared” over the spatial domain as it is propagated.

A common method for analysing the properties of a spatial discretisation within this context is Eigensolution Analysis (ESA). In this method, the wavenumber k can be shown to behave like a modified wavenumber k^* under the discretisation. Dispersive errors are then related to the difference between the real part of k and k^*, while diffusive errors are attributed to the magnitude of the imaginary part of k^* (which is zero for k) [25]. The analysis, based on that in [26] and [27], proceeds as follows.

Consider a one group, homogeneous, pure absorber problem in one spatial dimension with periodic boundary conditions, that has been angularly discretised using the discrete ordinates method:

$$\frac{1}{v}\frac{\partial \psi_k(x,t)}{\partial t} + \mu_k \frac{\partial \psi_k(x,t)}{\partial x} + \Sigma_t \psi_k(x,t) = 0 \quad \text{for} \quad k = 1, ..., N \tag{25}$$

where N is the number of ordinates being considered. As these equations are uncoupled, it is sufficient to consider just one of them. Without loss of generality, let $\mu_k > 0$. The analytical dispersion relation is obtained by seeking wavelike solutions of the form $\psi_k = e^{i(kx-\omega t)}$ and calculating ω as a function of k. In this case (and dropping the k subscripts), this yields:

$$-\frac{i\omega}{v} + i\mu k + \Sigma_t = 0 \tag{26}$$

$$\implies \omega = v\left(\mu k - i\Sigma_t\right) \tag{27}$$

Now consider a Galerkin projection onto one element of a discontinuous scheme (which is a patch in a PDG scheme, and a single element in a FDG scheme). Denoting the solution in that element by ψ_e and the test function by $w(x)$ we get:

$$\frac{1}{v}\int_{V_e} dx \frac{\partial \psi_e}{\partial t} w + \mu \int_{V_e} dx \frac{\partial \psi_e}{\partial x} w + \Sigma_t \int_{V_e} dx \psi_e w = 0 \tag{28}$$

where the integrals are performed over the physical element V_e. Applying integration by parts to the second term, and fully upwinding with $\mu > 0$ we obtain:

$$\frac{1}{v}\int_{V_e} dx \frac{\partial \psi_e}{\partial t} w - \mu \int_{V_e} dx \psi_e \frac{\partial w}{\partial x} + \Sigma_t \int_{V_e} dx \psi_e w + \mu\left(\psi_e(V_e^+)w(V_e^+) - \psi_L(V_L^+)w(V_e^-)\right) = 0 \tag{29}$$

where the L subscript denotes the element to the left of the one being considered. The Bubnov-Galerkin approach is now taken in space, so that the basis function coefficients are now functions of time rather than constants:

$$\psi_e = \sum_{B=1}^{nnp} c_B^e(t) R_B(x) \tag{30a}$$

$$w = R_A(x) \tag{30b}$$

As NURBS are interpolatory at the end points of their domain of definition and a partition of unity, by the numbering convention used, this implies that:

$$R_B(V_e^+) = \delta_{B,nnp} \tag{31a}$$

$$R_B(V_e^-) = \delta_{B,1} \tag{31b}$$

$$\implies \psi_e(V_e^+) = c_{nnp}^e \tag{31c}$$

$$\implies \psi_L(V_e^+) = c_{nnp}^L \tag{31d}$$

$$\implies w(V_e^+) = \delta_{A,nnp} \tag{31e}$$

$$\implies w(V_e^-) = \delta_{A,1} \tag{31f}$$

Combining these terms gives:

$$\frac{1}{v}\sum_{B=1}^{nnp} \frac{dc_B^e}{dt} \int_{V_e} dx R_B R_A - \mu \sum_{B=1}^{nnp} c_B^e \int_{V_e} dx R_B \frac{dR_A}{dx} + \Sigma_t \sum_{B=1}^{nnp} c_B^e \int_{V_e} dx R_B R_A + \mu(\delta_{A,nnp} c_{nnp}^e - \delta_{A,1} c_{nnp}^L) = 0 \tag{32}$$

For $A = 1, ..., nnp$. Defining the width of V_e to be h, and a parent element $V_P = [-1, 1]$ over which integration will be performed, we can recast the integrals over this new interval:

$$\int_{V_e} dx R_B(x) R_A(x) = \frac{h}{2}\int_{-1}^{1} R_B(\xi) R_A(\xi) d\xi =: \frac{h}{2}\underline{\underline{M}}_{\{A,B\}} \tag{33a}$$

$$\int_{V_e} dx R_B(x) \frac{dR_A(x)}{dx} = \int_{-1}^{1} R_B(\xi) \frac{dR_A(\xi)}{d\xi} \quad =: \underline{\underline{S}}_{\{A,B\}} \tag{33b}$$

These integrals define the mass and streaming matrices respectively. If we additionally define the solution vectors in the elements to be:

$$\mathbf{c}^e = \{c^e_B\}_{B=1,\ldots,nnp} \tag{34a}$$
$$\mathbf{c}^L = \{c^L_B\}_{B=1,\ldots,nnp} \tag{34b}$$

Substituting all of these definitions into 32, the system can be compactly expressed in matrix vector form as:

$$\frac{h}{2v}\underline{\underline{M}}\frac{d\mathbf{c}^e}{dt} - \mu\underline{\underline{S}}\,\mathbf{c}^e + \frac{h\Sigma_t}{2}\underline{\underline{M}}\mathbf{c}^e + \mu(\underline{\underline{R}}\mathbf{c}^e - \underline{\underline{L}}\mathbf{c}^L) = \mathbf{0} \tag{35}$$

The $\underline{\underline{R}}$ and $\underline{\underline{L}}$ matrices represent the outgoing and incoming flux from the cell respectively, which are defined as follows:

$$\underline{\underline{R}}_{i,j} = \delta_{i,nnp} \times \delta_{j,nnp} \tag{36a}$$
$$\underline{\underline{L}}_{i,j} = \delta_{i,1} \times \delta_{j,1} \tag{36b}$$

Following the ESA framework, we seek solutions of the form:

$$\psi = e^{i(kx-\omega t)} \tag{37}$$

To be precise, we seek solutions in which the expansion coefficients $\mathbf{c}^e$ are related to this solution through projection:

$$c^e_B(t) = \int_{-1}^{1} \psi\left(x_e + \frac{\xi h}{2}, t\right) R_B(x)d\xi \tag{38a}$$
$$= e^{i(kx_e-\omega t)} \int_{-1}^{1} e^{\frac{ik\xi h}{2}} R_B(\xi)d\xi \tag{38b}$$
$$=: e^{i(kx_e-\omega t)}\alpha_B \tag{38c}$$

where x_e is the centre point of element e. As $x_L = x_e - h$, this implies that:

$$\mathbf{c}^e = e^{i(kx_e-\omega t)}\boldsymbol{\alpha} \tag{39a}$$
$$\mathbf{c}^L = e^{i(k(x_e-h)-\omega t)}\boldsymbol{\alpha} \tag{39b}$$
$$= e^{-ikh}\mathbf{c}^e \tag{39c}$$

These definitions can now be substituted into equation 35 to get:

$$\frac{h}{2v}\underline{\underline{M}}\frac{d\mathbf{c}^e}{dt} + \frac{h\Sigma_t}{2}\underline{\underline{M}}\mathbf{c}^e = \mu(\underline{\underline{S}}\,\mathbf{c}^e - \underline{\underline{R}}\mathbf{c}^e + e^{-ikh}\underline{\underline{L}}\mathbf{c}^e) \tag{40}$$
$$\implies \frac{h}{2\mu v}(\Sigma_t v - i\omega)\underline{\underline{M}}\boldsymbol{\alpha} = \left(\underline{\underline{S}} + e^{-ikh}\underline{\underline{L}} - \underline{\underline{R}}\right)\boldsymbol{\alpha} \tag{41}$$

Defining $\underline{\underline{A}} = 2\underline{\underline{M}}^{-1}\left(\underline{\underline{S}} + e^{-ikh}\underline{\underline{L}} - \underline{\underline{R}}\right)$ gives the eigenvalue problem:

$$\underline{\underline{A}}\boldsymbol{\alpha} = \frac{h}{\mu v}(\Sigma_t v - i\omega)\boldsymbol{\alpha} \tag{42}$$

where **A** has nnp eigenvalues λ_j such that:

$$\lambda_j = \frac{h}{\mu v}(\Sigma_t v - i\omega_j) \tag{43}$$

The analytic dispersion relation was given by equation 27. We must now find a modified wavenumber k_j^* such that:

$$\omega_j = v\mu k_j^* - iv\Sigma_t \tag{44}$$

Substituting this into equation 43 we get:

$$\lambda_j = \frac{h}{\mu v}(\Sigma_t v - iv\mu k_j^* - \Sigma_t v) = -ihk_j^* \tag{45}$$

Once the eigenvalues of the matrix $\underline{\underline{A}}$ have been found, the real and imaginary parts of k_j^* can be calculated as follows:

$$\text{Real}(k_j^*) = -\frac{\text{Imag}(\lambda_j)}{h} \tag{46a}$$

$$\text{Imag}(k_j^*) = \frac{\text{Real}(\lambda_j)}{h} \tag{46b}$$

This sytem has three important features:

1. The matrix $\underline{\underline{A}}$, and therefore its eigenvalues λ_j have no dependence on Σ_t. Therefore the dispersive and diffusive errors will be the same whatever the total cross section is, and it is sufficient to consider the case $\Sigma_t = 0$.
2. The matrix $\underline{\underline{A}}$ also has no dependence on the ordinate μ under consideration except for the fact that it is assumed to be positive. As the system will be symmetric under reflection in the x-direction, this means that all calculations apply to any ordinate.
3. This system will have nnp eigenvalues, while the analytic system has just one possible wavenumber for each frequency.

5. Sweep Ordering and Normal Averaging

As with any discontinuous Galerkin projection of the discrete ordinates equations, an order must be chosen in which to sweep the elements to calculate the updated solution at each iteration. This can be done using the directed graph, or digraph associated with the mesh for each direction $\mathbf{\Omega}_k$ in the discrete ordinates angular quadrature set. For a particular direction, each vertex in the digraph represents an element in the FDG scheme or a patch in the PDG scheme. For the remainder of the section, both will be referred to as elements for brevity. There is an edge from vertex i to vertex j if information can propagate from element i to element j in the mesh. More precisely, defining the boundary between element i and element j to be $\Gamma_{i,j}$, and the outward unit normal (with respect to element i) on $\Gamma_{i,j}$ to be $\mathbf{n}$, then there is an edge from vertex i to vertex j if $\mathbf{\Omega}_k \cdot \mathbf{n} > 0$ at any point on $\Gamma_{i,j}$.

A digraph G is said to contain a cycle if there exists a vertex v such that by traversing the edges of G, v can be returned to, whilst visiting at least one other vertex inbetween. Cycles are problematic in sweep ordering, as it becomes

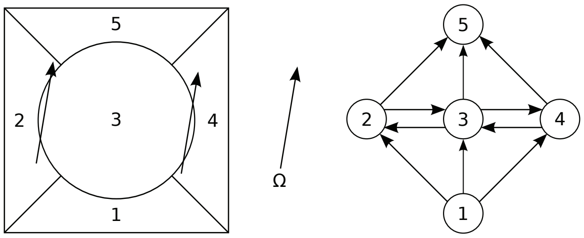

Figure 5: An example of a direction **Ω** and IGA mesh that give a digraph that contains cycles, along with the associated digraph.

unclear which element should be solved for first for a given ordinate if information can propagate from element i to element j and vice versa. This also reduces the efficiency of the scheme, as the solution for a direction cannot be obtained by solving for each element once.

Typically with straight sided finite elements in two dimensions, the resulting digraph is acyclic provided the elements are not too skewed (e.g. have a high aspect ratio), and so a perfect sweep ordering can be found so that the entire mesh can be solved completely in one transport sweep (this is not the case in three dimensions). This is not generally the case even in two dimensions with IGA, due to the strongly curved element boundaries, as demonstrated in Figure 5. In DGFEM when curved boundaries are encountered, lumped normal vectors are typically used so that it can be treated like a straight sided problem [4]. However, due to the highly curved nature of the elements in IGA (particularly in the PDG scheme) this approach does not seem appropriate.

Cycles are the cause of the existence of Strongly Connected Components (SCCs) in the graph. An SCC is defined to be a set of vertices $\{v_1, v_2, ..., v_n\}$ such that there is a path from v_i to v_j and vice versa for all $i, j \in \{1, 2, ..., n\}$. It can be seen in Figure 5 that vertex 1 is one SCC, vertices 2, 3 and 4 are another SCC and vertex 5 is the final SCC. The partitioning of vertices into SCCs is unique, as these sets of vertices cannot overlap. The condensation of a digraph G is formed by collapsing all the vertices in an SCC into a single new vertex, with edges in the condensation between these new vertices if there were any edges between any of the vertices in the original graph G. The condensation digraph is then acyclic, and so a perfect ordering of these is possible [28]. The natural question then arises of how to solve for the elements within an SCC before moving on to the next one.

In [4] these cycles were broken by approximating one or more face flux values with those from the previous iteration. This requires storing the previous iteration's angular flux on reentrant faces. However even on deliberately skewed meshes with high angular quadrature orders, this was less than 4% of faces in their examples, and so represented little storage overhead. With the highly curved element boundaries present in the IGA meshes the reentrant face percentage will be much higher, particularly with the PDG scheme. This would require the storage of a large percentage of the angular flux from the previous iteration, increasing memory requirements substantially.

One alternative option is to solve each SCC completely before moving on to the next one. This can be done by iterating over the elements within the SCC until the angular flux for this direction has converged, keeping the source term constant and updating only the incoming flux values from the neighbouring elements within the SCC. A second option is to simply solve for each element in the SCC once, so that overall during the transport sweep each element is solved for exactly once per direction.

In both cases an order to solve the elements within the SCC must be picked so that either the iteration to convergence is as efficient as possible, or the outer scatter iteration is as efficient as possible if each element is only being solved for once. To this end, the digraph for an individual SCC can be formed, simply by restricting the original digraph G to the vertices contained in the SCC. This will necessarily contain cycles, and so the problem becomes how to delete edges in the graph to remove the cycles so that a sweep can be performed. To this end, each edge in this graph is assigned a weight. The edge connecting vertex i to vertex j is then assigned the weight $w_{i,j}$ as follows:

$$w_{i,j} = \int_{\Gamma^+_{i,j}} \mathbf{\Omega}_k \cdot \mathbf{n} dS \tag{47}$$

where $\Gamma^+_{i,j}$ is defined to be the portion of $\Gamma_{i,j}$ where $\mathbf{\Omega}_k \cdot \mathbf{n}$ is positive. This is then a measure of how "outgoing" the direction $\mathbf{\Omega}$ is on this boundary. The edges to be deleted are then chosen as follows. Where there is both an edge from vertex i to vertex j and an edge from vertex j to vertex i, the edge with the lowest weight is deleted. It can be shown that this is equivalent to using an average normal for $\Gamma_{i,j}$ as follows. Defining $\Gamma^-_{i,j}$ to be the portion of $\Gamma_{i,j}$ where $\mathbf{\Omega}_k \cdot \mathbf{n}$ is negative, we have:

$$\begin{aligned} w_{i,j} - w_{j,i} &= \int_{\Gamma^+_{i,j}} \mathbf{\Omega}_k \cdot \mathbf{n} dS - \int_{\Gamma^+_{j,i}} \mathbf{\Omega}_k \cdot \mathbf{n} dS &&= \int_{\Gamma^+_{i,j}} \mathbf{\Omega}_k \cdot \mathbf{n} dS + \int_{\Gamma^-_{i,j}} \mathbf{\Omega}_k \cdot \mathbf{n} dS \\ &= \int_{\Gamma_{i,j}} \mathbf{\Omega}_k \cdot \mathbf{n} dS &= \mathbf{\Omega}_k \cdot \int_{\Gamma_{i,j}} \mathbf{n} dS &= \mathbf{\Omega}_k \cdot \bar{\mathbf{n}} \end{aligned} \tag{48}$$

Therefore $w_{i,j} > w_{j,i}$ is equivalent to $\mathbf{\Omega}_k \cdot \bar{\mathbf{n}} > 0$ and vice versa. This makes the method analogous to a sweep ordering for straight sided finite elements. Therefore the resulting graph is expected to be acyclic. It is worth noting that this is not guaranteed by the graph theory, and so checks must be performed, although no meshes have been found for which the method does not work.

An associated issue is whether or not an average normal vector to an element boundary can be used, or whether normal vectors at each quadrature point must be used to perform element boundary integration. If an average normal vector can be used without a significant loss in solution accuracy, this would greatly reduce the amount of time spent during the sweeps evaluating boundary integrals, and so would make the whole scheme more efficient. The effect of using an average normal vector will be demonstrated in Section 6.3.

Tied in with this is the question of how to perform the integrals over $\Gamma^+_{i,j}$ and $\Gamma^-_{i,j}$ without incurring a significant loss in accuracy of the solution. As integration in finite element schemes is usually performed over the entire boundary, it is not immediately obvious how to perform integrals over only segments of element boundaries. In [29] this was achieved by splitting elements in two at the point where the direction $\mathbf{\Omega}_k$ was tangent to the element bounday. A Discontinuous Galerkin method was then used separately on each of these two elements and the resulting solutions projected back onto the original element. However this was not found to improve accuracy over using an average normal vector in [9].

The method chosen here was to treat the integral over a partial boundary, say $\Gamma^+_{i,j}$, as an integral over the whole boundary $\Gamma_{i,j}$ in which the integrand vanishes where $\mathbf{\Omega}_k \cdot \mathbf{n} < 0$. I.e.

$$\int_{\Gamma^+_{i,j}} f(S) \mathbf{\Omega}_k \cdot \mathbf{n} dS = \int_{\Gamma_{i,j}} g(S) \mathbf{\Omega}_k \cdot \mathbf{n} dS \quad \text{where} \tag{49a}$$

$$g(s) = \begin{cases} f(S) & \text{for} \quad \mathbf{\Omega}_k \cdot \mathbf{n} \geq 0 \\ 0 \quad \text{for} & \mathbf{\Omega}_k \cdot \mathbf{n} < 0 \end{cases} \tag{49b}$$

As the integrand is therefore not a polynomial even in the B-Spline case, the Gaussian quadrature points used are not guaranteed to provide an accurate evaluation of the integral. However, as the number of quadrature points used is increased, it is expected that the accuracy of these partial boundary integrals will be improved as this is equivalent to approximating the non-polynomial integrand with a higher order polynomial.

6. Numerical Results

6.1. Dispersion and Diffusion Analysis

The real and imaginary parts of the numerical wavespeeds k_j^* are compared to the actual wavenumber k. As only the product kh appears in the equation for $\mathbf{A}$, only this product needs to be considered. Intuitively, this corresponds to the fact that if the wavespeed doubles but the mesh width halves, the numerical dispersion and diffusion are unaffected. The axes are also normalised by $\frac{1}{\text{DOFs}}$, the inverse of the number of degrees of freedom in the system. This is so that the eigenvalues are periodic with period 2π on this scale.

This is demonstrated in Figures 6 and 7, where the real and imaginary parts of k_j^* have been plotted for a fully discontinuous quadratic scheme. The "primary" mode (i.e. the one that recovers $k_j^* = 0$ at $k = 0$) represents the wavenumber that is trying to be approximated by the scheme. The secondary modes in both the dispersive and diffusive aspects are numerical artefacts created by the spatial discretisation as a result of the polynomial basis functions employed being unable to exactly represent a single Fourier component initial condition [27], and are simply shifted versions of the primary mode. As such, all of the information about the secondary modes is encapsulated in the primary mode, and so only the primary mode will be considered from now on.

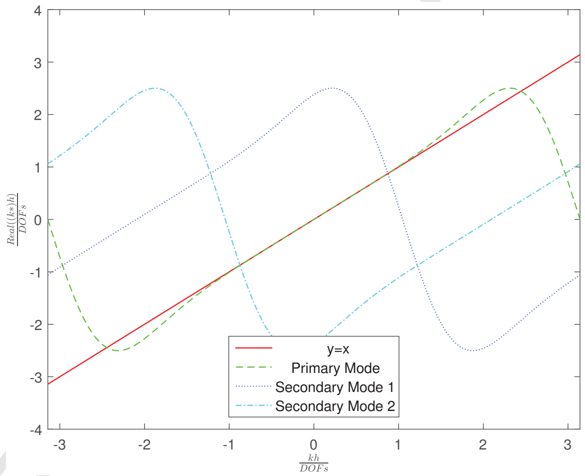

Figure 6: Real parts of the three eigenvalues associated with a fully discontinuous quadratic formulation, representing the dispersive behaviour of the scheme.

For a spatial discretisation to accurately propagate a wave, it needs to exhibit both low dispersion and diffusive behaviour. For dispersion, this corresponds to $|k - \text{Real}(k_j^*)|$ being small, while for diffusion it corresponds to $|\text{Imag}(k_j^*)|$ being small.

In Figure 8 it can be observed that as the polynomial order is increased with a fully discontinuous scheme, the ability of k_j^* to propagate the wavenumber k accurately (i.e. less dispersively) improves. In Figure 9, it can also be seen that as the polynomial order is increased, the spatial discretisation becomes less diffusive, and the difference is a lot more pronounced for the diffusive than for the dispersive behaviour.

In Figure 11, it can be observed that as a patch is h-refined, the numerical diffusion decreases, although nowhere near as drastically as when the order is elevated. However, Figure 10 shows more interesting behaviour. As more

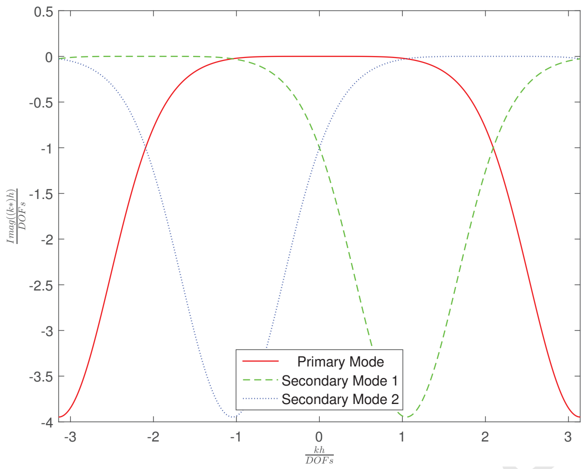

Figure 7: Imaginary parts of the three eigenvalues associated with a fully discontinuous quadratic formulation, representing the diffusive behaviour of the scheme.

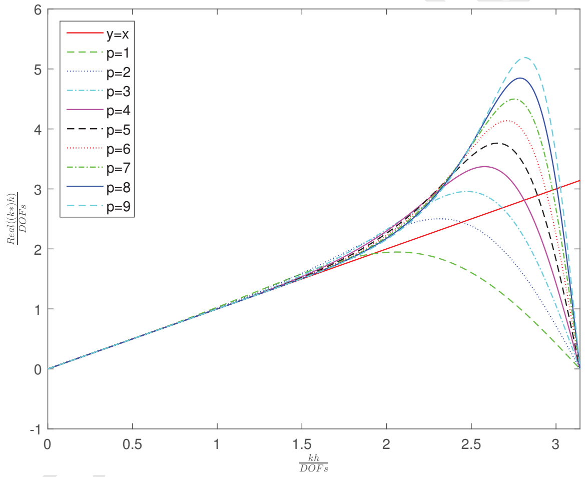

Figure 8: The real parts of the primary modes for fully discontinuous schemes with varying polynomial orders p.

knots are inserted, the dispersive behaviour initially seems to get better, but then dips below the $k_j^* = k$ line, and so it is unclear which discretisation is performing the best here.

A rigorous definition is needed to define which spatial discretisation is the "best" at propagating a wave. If we define $\bar{h} = \frac{h}{DOFs}$, the "length" of one degree of freedom within a patch, then one such possible definition is to find the maximum value of kh such that both the dispersive and diffusive errors introduced by the spatial scheme are less than 1% per $\bar{h}$.

Figure 12 clearly demonstrates that the ability of the spatial discretisations to model a travelling wave is monotonically improving both as the polynomial order is increased, and as the number of knot insertions is increased. However many of these schemes will have the same number of degrees of freedom (and therefore, roughly, the same difficulty to solve). For example, a quadratic patch with no knot insertions has 3 basis functions, as does a linear patch with 1 knot insertion.

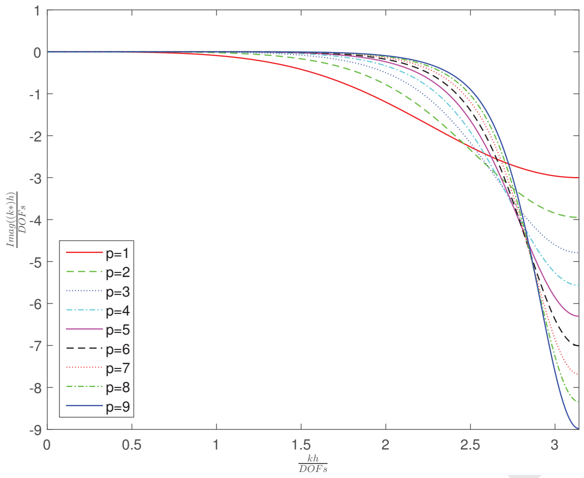

Figure 9: The imaginary parts of the primary modes for fully discontinuous schemes with varying polynomial orders p.

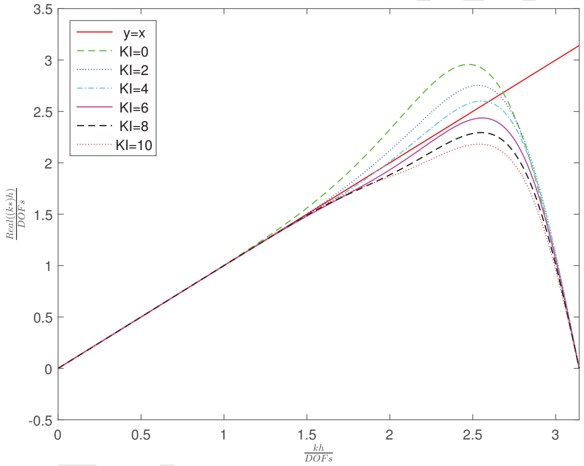

Figure 10: The real parts of the primary modes for a cubic patch discontinuous scheme with varying numbers of knot insertions (K.I.).

Figure 13 shows how various spatial schemes compare when the number of basis functions in a patch is kept constant. There is a general trend that increasing the polynomial order has a more positive impact than knot refining the patches, particularly when moving from linears to quadratics. This trend is not monotonic. However, the highest order polynomial (i.e. with zero knot insertions, representing a FDG) scheme is consistently the best choice no matter how many degrees of freedom are analysed.

6.2. *Strongly Connected Component Convergence*

In order to test the effect of fully converging the SCC's versus solving each element only once for a given element, a variant of the problem from Section 6.5 was used with the randomized meshes. However, the straight sided elements from these meshes do not give rise to any SCC's, and so the boundaries must be forced to be curved in order to compare the two solution methods. This was achieved as follows. The parameter α from Section III of [30] was allowed to vary between 0 and 0.3. The linear elements in this resulting randomized mesh were then order elevated to quadratic

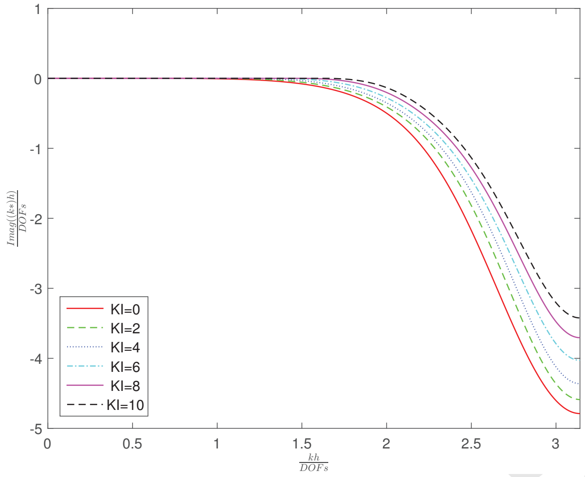

Figure 11: The imaginary parts of the primary modes for a cubic patch discontinuous scheme with varying numbers of knot insertions (K.I.).

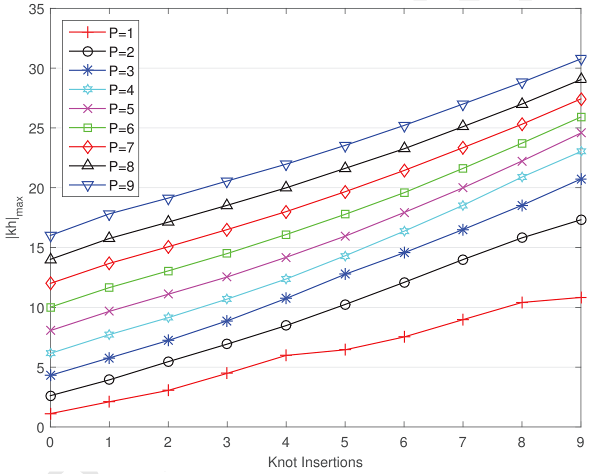

Figure 12: The maximum value of $|kh|$ for which the error in the numerical wavenumber is less than 1% in both the real and imaginary parts per unit $\bar{h}$, for various polynomial orders and numbers of knot insertions.

elements, introducing a new control point in the centre of each boundary of each element. This control point was then perturbed along the normal direction to the boundary it is on. The final position of the control point $\mathbf{C_F}$ was then calculated from its initial position $\mathbf{C_I}$ by the following formula:

$$\mathbf{C_F} = \mathbf{C_I} + \beta r \mathbf{n} \tag{50}$$

where r is a random number between -1 and 1, and β is a new parameter, which was allowed to vary between 0 and 0.5. This curves the boundary, creating SCC's. The following results were obtained using the FDG scheme with quadratic elements.

By developing the mesh in this way, it is possible that some of the element boundaries could cross, as demonstrated in Figure 14. A mesh that is invalid in this respect will have a Jacobian whose determinant changes sign within the element. These meshes were guarded against by checking that the determinant of the Jacobian did not

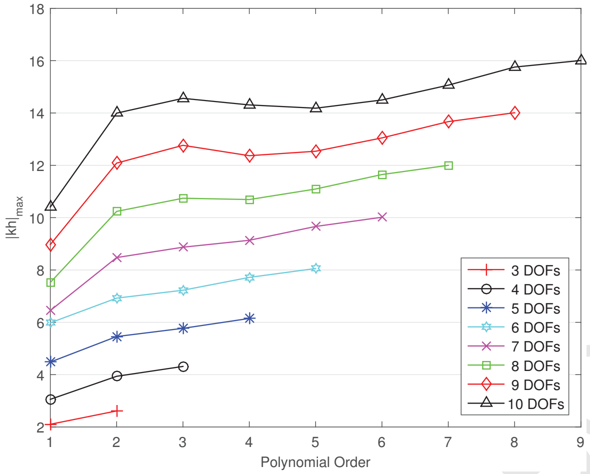

Figure 13: The maximum value of $|kh|$ for which the error in the numerical wavenumber is less than 1% in both the real and imaginary parts per unit $\bar{h}$, for various numbers of degrees of freedom as the polynomial order is increased.

change sign at any of the 16^2 quadrature points used in each element, and discarded if the determinant did change sign. This occurred for relatively few (α, β) pairs.

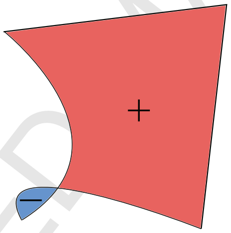

Figure 14: An invalid mesh where two boundaries of an element have crossed over, where the Jacobian determinant is positive in the red (upper right) region and negative in the blue (lower left) region.

Three different metrics of mesh complexity were used, presented in Figures 15, 16 and 17. Figure 15 shows that the average size of a non-trivial SCC did not vary strongly as α and β were varied. Note that for $\beta = 0$ all of the elements have straight sides and therefore there are no non-trivial SCCs. However Figures 16 and 17 demonstrate that the largest SCC in the mesh over all directions and the average value of the largest SCC for each direction both vary strongly with β, as well as weakly with α.

An SCC was considered to be converged if the average change in the solution was an order of magnitude lower than the tolerance for the scalar flux in the source iteration. Figure 18 shows that the number of matrix solves required to converge the system when the SCCs were converged was considerably higher than the number required when each element was solved only once. It can also be seen that as the number and size of the SCCs increases (i.e. as β increases), this disparity increases.

However, as the LU decomposition of the matrix for each element in an SCC was stored while this SCC was being solved, the average time for each matrix solve was lower when fully converging the SCCs than when each element was only being solved once. Figure 19 shows that the disparity between fully converged SCCs and a single solve per element does indeed decrease as the mesh complexity increases. However, this is not enough to make the scheme even comparably efficient.

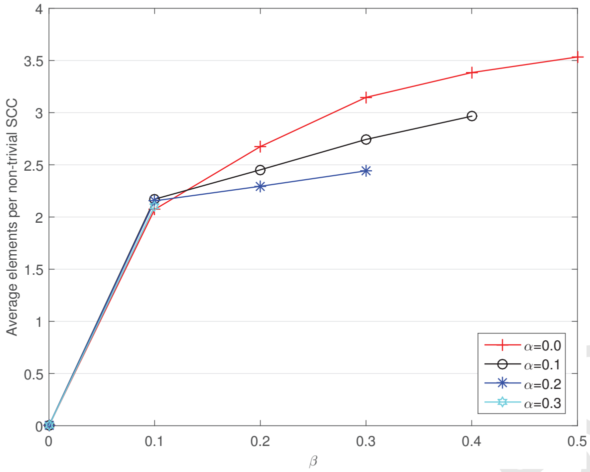

Figure 15: The change in size of the average number of elements in a non-trivial SCC as α and β vary.

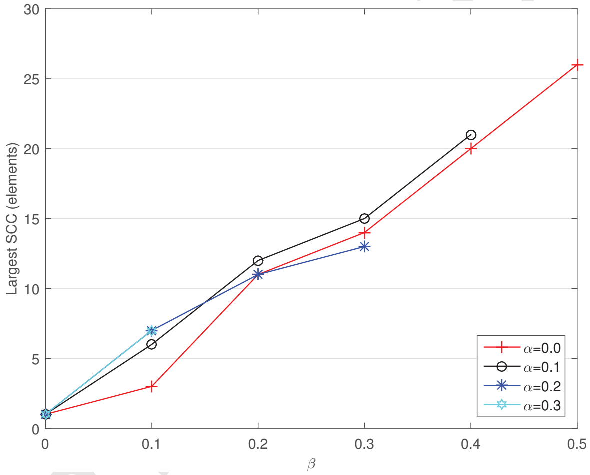

Figure 16: The change in size of the largest number of elements in an SCC over all directions as α and β vary.

6.3. Normal Averaging

In order to test the impact of using an average normal vector on element boundaries, a variant of the problem solved in Section 6.2 was used. In this case, α was fixed at 0, as this has no impact on the curvature of the elements, while β was varied from 0 to 0.3, increasing the curvature of the element boundaries. Figure 20 shows the L_2 error in the solution when 3 and 16 quadrature points were used to integrate along the boundary, along with when an average normal vector was used for the $\beta = 0$ case. As all the elements have straight sides in this case, the average normal vector on an element boundary is exact along that boundary, and so no error is introduced in using an average normal vector.

Figures 21, 22 and 23 show the same results as β is increased from 0.1 to 0.3. It can be seen that there is a negligible difference in the solution accuracy when using 3 or 16 quadrature points on the boundary. As the method of mesh creation for this problem involves only B-Splines and not NURBS, the basis functions evaluated on the

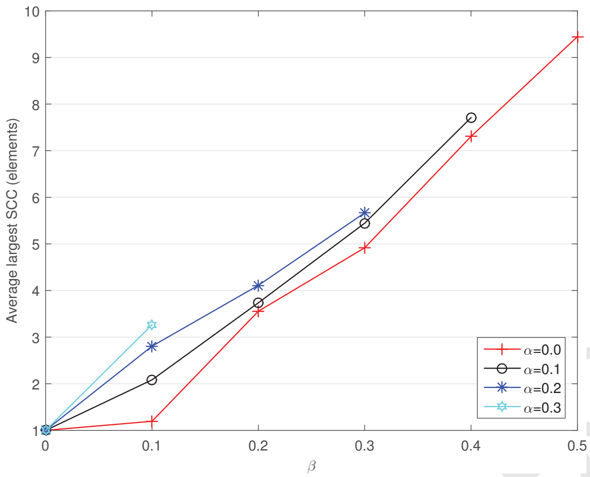

Figure 17: The change in size of the average number of elements in the largest SCC for each direction as α and β vary.

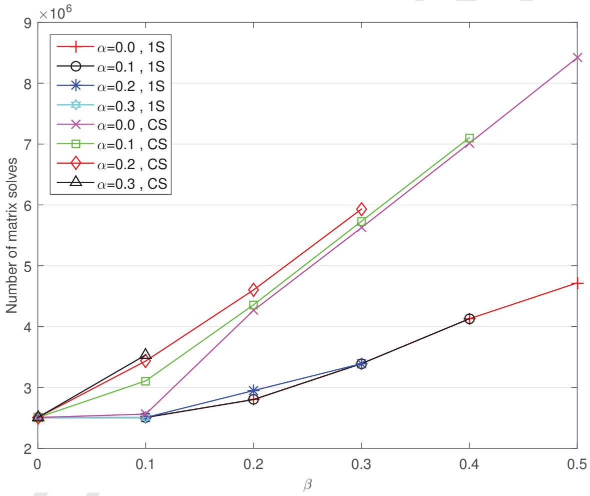

Figure 18: The number of matrix solves required to converge the system as α and β vary. 1S means each element was solved exactly once, whereas CS means each SCC was converged fully.

boundary are polynomials. The integrals being performed along the boundary are of the form $\int_{\Gamma_{i,j}} R_A(\xi)R_B(\xi)\mathbf{n}(\xi)dS$, where R_A and R_B are polynomials of order p in ξ, while the components of the normal vector $\mathbf{n}$ are sums of derivatives of these basis functions, and so are polynomials of order $p-1$. Therefore overall the integrand is a polynomial of order $p + p + (p-1) = 3p - 1$. To integrate this exactly requires n Gaussian quadrature points, where $2n - 1 \geq 3p - 1 \implies n = \frac{3p}{2}$ when p is even or $\frac{3(p+1)}{2}$ when p is odd. In this case $p = 2$, so 3 Gaussian quadrature points are required to perform the integral exactly. These integrals will only be exact when $\Gamma^{\pm}_{i,j} = \Gamma_{i,j}$. Therefore there will only be a difference between the integrals calculated with 3 or 16 quadrature points on element boundaries where $\Gamma^{\pm}_{i,j} \neq \Gamma_{i,j}$ and this appears to make a negligible difference to the solution accuracy.

However when an average normal vector is used (denoted by 1 QP in the Figure legends), a minimum drop in accuracy of an order of magnitude is observed, along with a reduction in the convergence rate of the scheme. Figure 24 shows that the schemes where 3 quadrature points are used for boundary integrals obtain approximately 3rd order

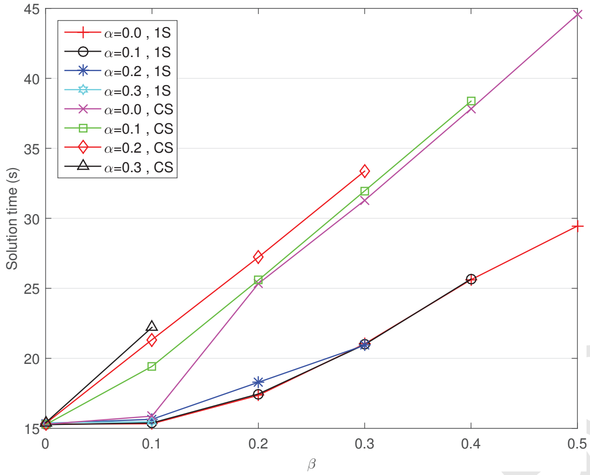

Figure 19: The solution time as α and β vary, with 1S and CS defined as in Figure 18.

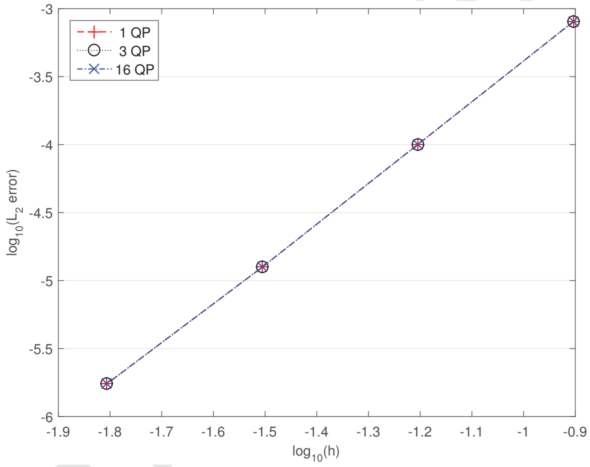

Figure 20: The convergence under mesh refinement when the boundary integrals are performed using 1, 3 and 16 quadrature points (QP) for $\beta = 0.0$.

convergence as expected, whereas the schemes using averaged normal vectors obtains only approximately 1st order convergence.

Figure 25 shows the convergence of the maximum error in the angle between the average normal on an element boundary and the actual normals at the quadrature points that it is trying to approximate. It can be seen that for all values of β, this error converges only at first order. It is hypothesised that this is causing the loss of accuracy in the schemes utilising an average normal vector, and that this is not a viable method.

6.4. *Analytical S_N Comparisons in 1D*

One method to test the convergence of a spatial discretisation is by computing the error between the numerical solution and the analytical one. Unfortunately, these are in general difficult to calculate, especially in multiple dimensions. One problem for which it is possible to compute the analytical solution is the discrete ordinates equations applied to the Reed Cell in one spatial dimension [31]. This is a one group fixed source problem designed to test the

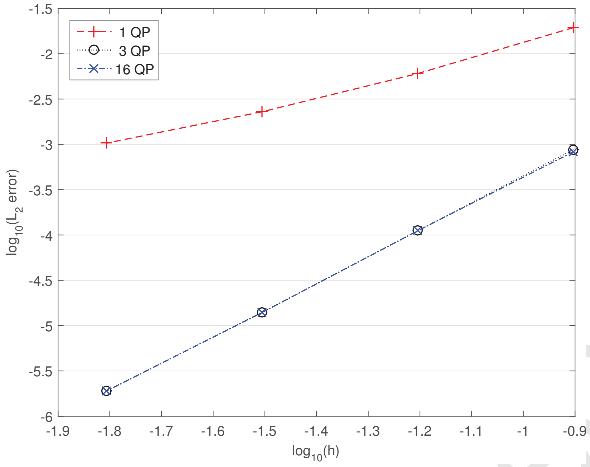

Figure 21: The convergence under mesh refinement when the boundary integrals are performed using 1, 3 and 16 quadrature points (QP) for $\beta = 0.1$.

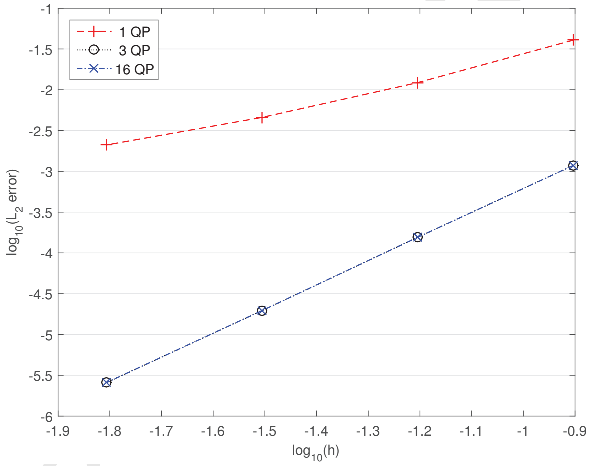

Figure 22: The convergence under mesh refinement when the boundary integrals are performed using 1, 3 and 16 quadrature points for $\beta = 0.2$.

stability of spatial discretisations as it has highly discontinuous Σ_t values in adjacent regions.

The geometry and cross sections for the problem are given in Table 1. There is a reflective boundary at $x = 0$ cm and a vacuum boundary at $x = 8$ cm.

Using these values and a given angular quadrature set, the discrete ordinates equations can then be solved exactly using a symbolic algebra program. For Gaussian quadrature using 8 angles, the scalar flux is given in the Appendix of [31] in terms of exponential and hyperbolic functions. One important point to note is that the analytical solution has a discontinuous first derivative (for example at $x = 3.0$ cm). The scalar flux solution to this problem is plotted in Figure 26.

For a given mesh, every element was the same width. The numerical solution is then evaluated at 16 points in each element and a trapezoidal rule used to integrate the expression given in equation 51 to calculate the L_2 error ϵ_2.

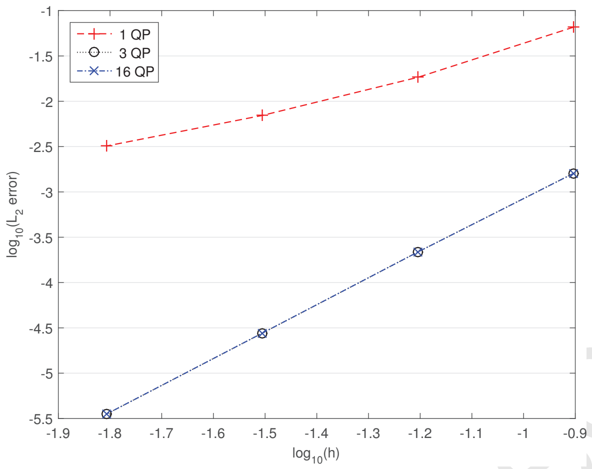

Figure 23: The convergence under mesh refinement when the boundary integrals are performed using 1, 3 and 16 quadrature points for $\beta = 0.3$.

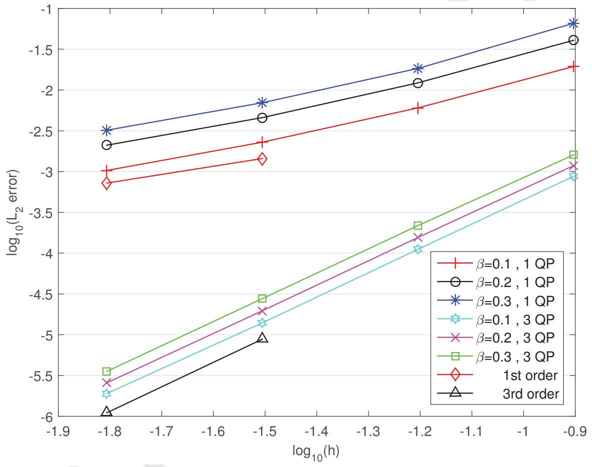

Figure 24: The convergence under mesh refinement for $\beta = 0.1, 0.2$ and 0.3 when 3 quadrature points are used compared to when an average normal is used, along with lines demonstrating 1st and 3rd order convergence.

$$\epsilon_2 = \left(\int_0^8 \left(\phi_{analytical}(x) - \phi_{numerical}(x) \right)^2 dx \right)^{\frac{1}{2}} \tag{51}$$

The calculated L_2 errors for both patch and fully discontinuous schemes are given in Figure 27 at various polynomial orders as the mesh is refined. The same errors are also plotted in Figure 28 but on a degree of freedom basis.

Figure 27 shows that for both schemes as either the polynomial order is increased or the mesh is refined the numerical solution converges to the analytical solution. For polynomial orders higher than linears, the fully discontinuous scheme exhibits more rapid convergence on an element width basis than the patch discontinuous scheme, and this difference becomes more pronounced as the polynomial order is increased. As the analytical solution has a discontinuous first derivative at several points in the problem domain where there are material boundaries, a full order of convergence cannot be expected from either of the schemes. An approximate order of convergence of each scheme is given in Table 2.

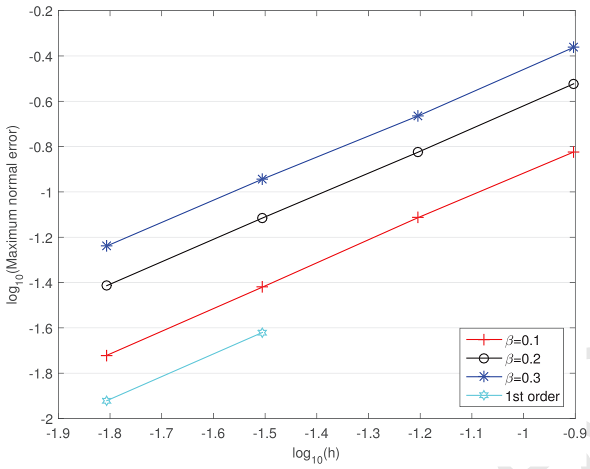

Figure 25: The convergence under mesh refinement of the maximum error in the angle between the average normal and the actual normal on an element boundary measured in radians, along with a line demonstrating 1st order convergence.

Material	Domain	Σ_t	Σ_s	Source
1. Fuel	0cm-2cm	50.0	0.0	50.0
2. Can	2cm-3cm	5.0	0.0	0.0
3. Void	3cm-5cm	0.0	0.0	0.0
4. Moderator	5cm-6cm	1.0	0.9	1.0
5. Moderator	6cm-8cm	1.0	0.9	0.0

Table 1: The cross section data for the Reed Cell problem.

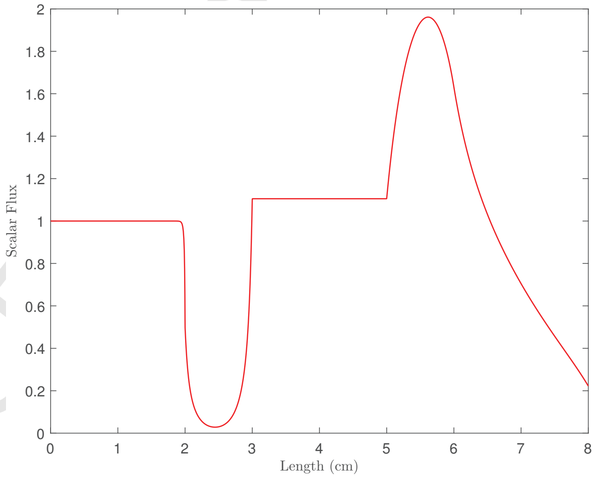

Figure 26: The analytical scalar flux solution to the S_8 discrete ordinates equations for the Reed Cell problem.

While Table 2 shows that no scheme is attaining the maximum possible order of convergence, there is a general trend that as the polynomial order is increased within one of the schemes the order of convergence increases. It also demonstrates that for cubic polynomials and higher, the order of convergence is greater for the fully discontinuous

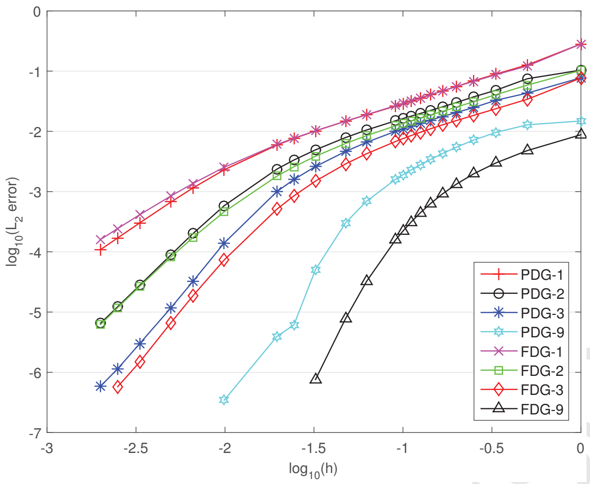

Figure 27: L_2 error convergence for both schemes at various polynomial orders as the mesh is refined.

Polynomial Order	PDG	FDG
1	2.01	1.85
2	2.86	2.86
3	3.37	3.72
4	3.47	4.33
9	5.13	5.23

Table 2: Approximate orders of convergence for each scheme. These were calculated using the penultimate two points for each scheme from Figure 27.

scheme than for the patch discontinuous.

However, this is not a completely fair comparison as the number of basis functions in the spatial discretisation increases much more rapidly as the mesh is refined with the fully discontinuous scheme than for the patch discontinuous version. Figure 28 shows the same error convergence as Figure 27 but on a degree of freedom basis.

Using this metric, the patch discontinuous scheme converges much more rapidly than the fully discontinuous scheme for a given polynomial order. It is difficult to directly compare the efficiency of the two schemes using either of these metrics. While the solution of both schemes involves sweeping through the mesh in the direction of neutron travel, the FDG scheme involves the solution of many small, dense linear systems, while the PDG scheme involves the solution of a few, much larger and sparser linear systems. An efficient method of solving these “patch matrices” has yet to be developed and so a direct comparison of efficiency cannot yet be performed. One possible method would be to take advantage of the banded nature of the matrices, as all of the non-zero entries lie in a finite number of bands of fixed width for a given polynomial order.

6.5. *Method of Manufactured Solutions in 2D*

In multiple dimensions, it is usually more practical to use the Method of Manufactured Solutions (MMS) to assess the convergence properties of a spatial discretisation. This has several advantages over comparisons to analytical solutions. An exact solution to the transport problem does not need to be calculated, which is frequently intractable in multiple dimensions. Also, the desired level of smoothness of the solution can be chosen, allowing full convergence rates to be achieved. Frequently realistic transport problems exhibit discontinuous first derivatives, throttling the attainable convergence rate of any scheme [32].

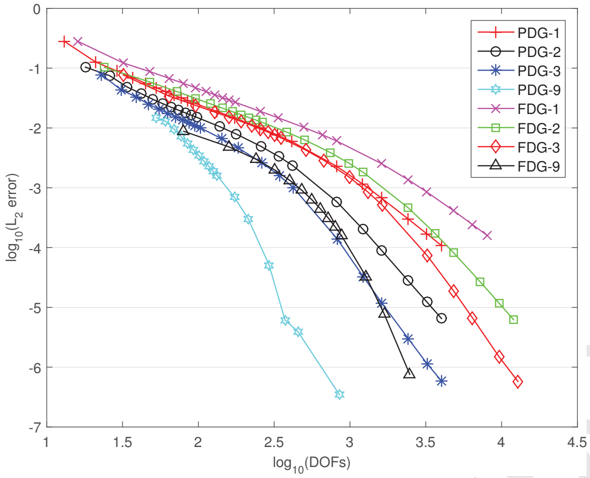

Figure 28: L_2 error convergence for both schemes at various polynomial orders, with the horizontal axis representing the total number of basis functions in the system.

For a full treatment of MMS, see [33]. Here, a modified version of the manufactured solution presented in [30] was used in order to verify the order of convergence of the two different spatial schemes. Note that the manufactured solution chosen is for the discrete ordinates equations, not the full transport equation, and as such can be used to test only the effect of the spatial discretisations. The spatial domain is given by the unit square [0,1] x [0,1]. If the discrete ordinates equations are solved with N ordinates, such that the solution ψ_k corresponds to direction $\mathbf{\Omega}_k$ and weighting w_k, with $\mathbf{\Omega}_k = (\mu_k, \eta_k)$, then the manufactured solution chosen is given by:

$$\psi_k(x, y) = x^2 y^2 (1 - x^2)(1 - y^2)(1 + \mu_k^2 + \eta_k^2) \tag{52a}$$

$$\phi_0(x, y) = \sum_{k=1}^{N} w_k \psi_k \tag{52b}$$

This manufactured solution has several useful properties. Firstly, it is infinitely differentiable, and therefore a spatial scheme can exhibit its full convergence rate. Secondly, it is equal to zero on the domain boundaries, and so a natural vacuum boundary condition can be used. The source term (which is now anisotropic) can then be calculated in each direction as follows:

$$Q_k(x, y) = 4\pi(\mathbf{\Omega_k} \cdot \nabla\psi_k(x, y) + \Sigma_t \psi_k(x, y)) - \Sigma_s \phi_0(x, y) \tag{53}$$

Cross section values were chosen as in [30], so that $\Sigma_t = 1$ and $\Sigma_s = 0.5$. For a given spatial discretisation, once the numerical scalar flux solution ϕ_{num} had been calculated, the L_2 error ϵ_2 was calculated by numerically integrating expression 54 using 16^2 Gaussian quadrature points per element.

$$\epsilon_2 = \left(\int_0^1 \int_0^1 (\phi_{num} - \phi_0)^2 \, dx dy \right)^{\frac{1}{2}} \tag{54}$$

The convergence rates of the scheme were calculated on 3 different types of mesh: orthogonal grids, “randomized” grids and pincell-like grids.

The first mesh types considered were simple orthogonal grids. The original geometry was split into 2^k elements in each direction, for $k = 3, ...8$, and the L_2 error calculated for each mesh, for polynomial orders $p = 1, 2$ and 3. Convergence plots for quartic splines (and higher orders) are not presented on these grids, as they are capable of exactly representing the solution, and thus give an error only due to machine precision, which was verified to be the case.

The results for the pure PDG and FDG schemes are shown in Figure 29. In this case, the PDG scheme reduces to a continuous Galerkin approach. Figure 29 shows that the FDG schemes attain order $p + 1$ convergence. The PDG scheme on the other hand, exhibits only order p convergence rates as the mesh is refined.

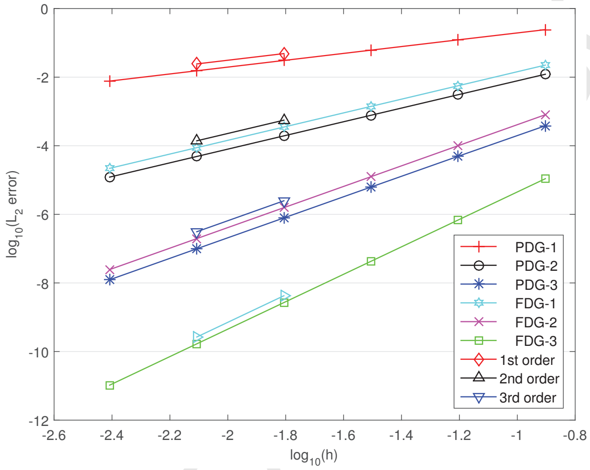

Figure 29: Convergence plots for the PDG and FDG schemes of various polynomial orders for the MMS problem on orthogonal grids.

A third scheme, which is a hybrid of the PDG and FDG schemes has also been implemented. In this hybrid scheme, the spatial domain is first discontinuously subdivided into patches first. These patches are then h-refined with the same number of knot insertions in each parametric direction, maintaining C^{p-1} continuity at the refinement positions. In this case, h was considered to be the distance between discontinuities in the mesh.

Figures 30, 31 and 32 show the convergence plots for the hybrid schemes for linear, quadratic and cubic basis functions respectively. In all cases it can be seen that while h-refining a patch does increase the accuracy of a scheme, it does not change the order of convergence, which is $p + 1$ in all cases. This suggests that for a given number of basis functions per element, order elevating is likely to be more beneficial for accuracy than knot insertion. Figure 33 shows the results of a comparison performed on this basis.

Figure 33 demonstrates that in this case, for the same number of basis functions per patch, and therefore a comparable amount of computation in solving each patch, order elevation does indeed provide more accurate solutions than knot refinement.

As orthogonal grids do not accurately represent the meshes the method is to be used on, a more stringent test of the convergence rates of the schemes is given by a perturbed grid. An identical procedure as that used in [30] was implemented to generate these grids, examples of which are shown in Figure 34.

Figure 35 shows the convergence rates of the FDG schemes on both the orthogonal and randomized grids. It can

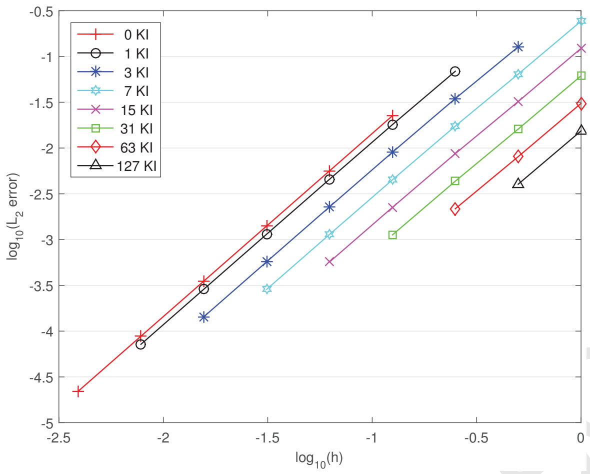

Figure 30: Convergence plots for the linear hybrid schemes with varying numbers of knot insertions for the MMS problem on orthogonal grids.

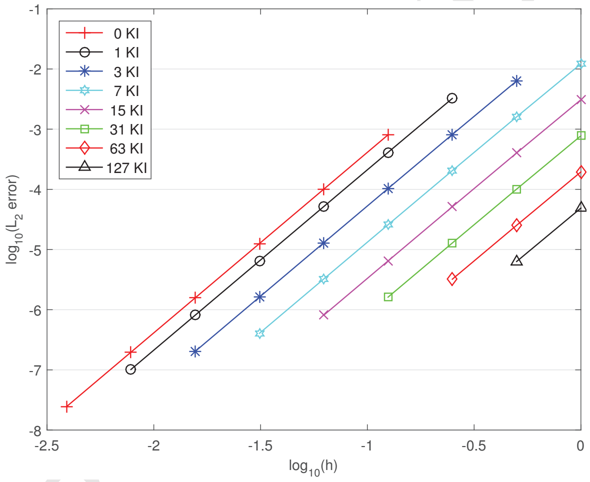

Figure 31: Convergence plots for the quadratic hybrid schemes with varying numbers of knot insertions for the MMS problem on orthogonal grids.

be seen that identical convergence rates are obtained on the randomized grids as were obtained on the orthogonal grids, although for a given mesh sizing the randomized grids are slightly less accurate. The $p = 4$ case has also been included, as on the randomized grids, the basis functions are no longer capable of representing the solution exactly. They are observed to give 5th order convergence. Plots similar to Figures 30, 31 and 32 on randomized grids have been omitted for brevity, as they exhibit the same behaviour as on the orthogonal grids.

A yet more stringent test of the methods is to use a pincell-like mesh as shown in Figure 36, incorporating both rational basis functions and curved boundaries. The radius of the circle was taken to be $\frac{1}{\sqrt{5\pi}}$, so that each patch has the same area and therefore the average element size is identical in all patches. The measure of h was then taken to be $\frac{1}{\sqrt{E}}$, where E is the number of elements in the mesh.

Figure 37 shows the convergence plots for the PDG and FDG schemes as the mesh is refined for various polynomial orders. All of the schemes achieved the same order of accuracy as on both the orthogonal and randomized grids. The L_2 error of the schemes was limited to $\approx 10^{-12}$ due to a combination of the machine precision used and the

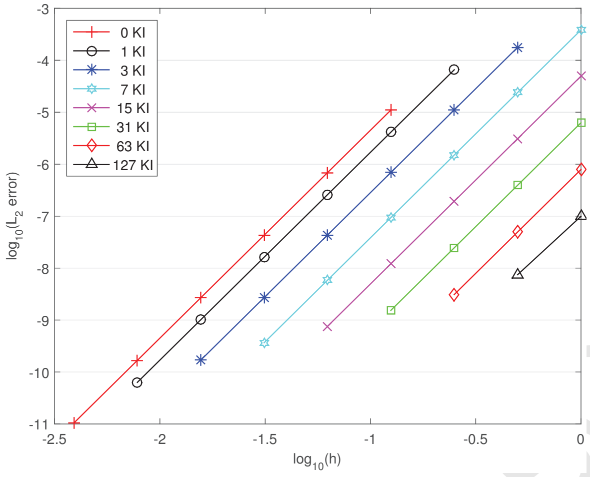

Figure 32: Convergence plots for the cubic hybrid schemes with varying numbers of knot insertions for the MMS problem on orthogonal grids.

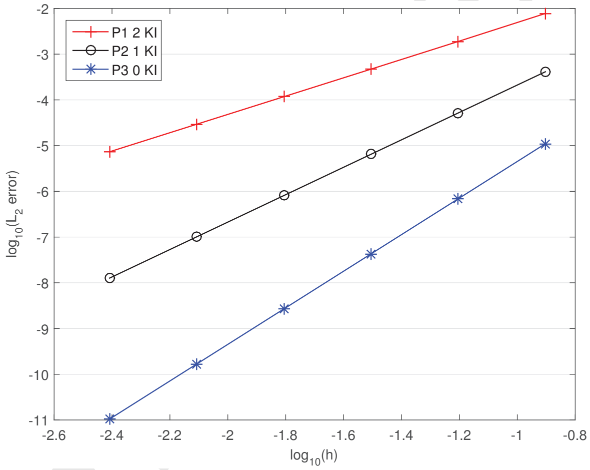

Figure 33: Convergence plots for 3 schemes of 3 different polynomial orders which all have the same number of basis functions per patch (16) for the MMS problem on orthogonal grids.

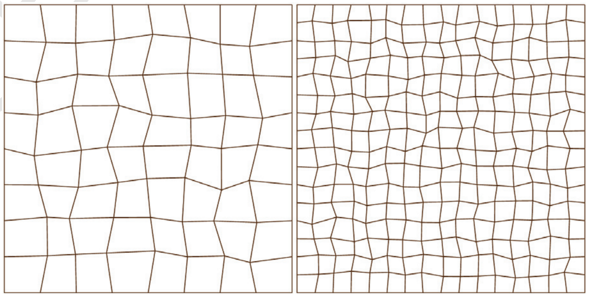

Figure 34: A coarse (k=3) and more refined (k=4) randomized mesh for the MMS problem on randomized grids with $\alpha = 0.2$.

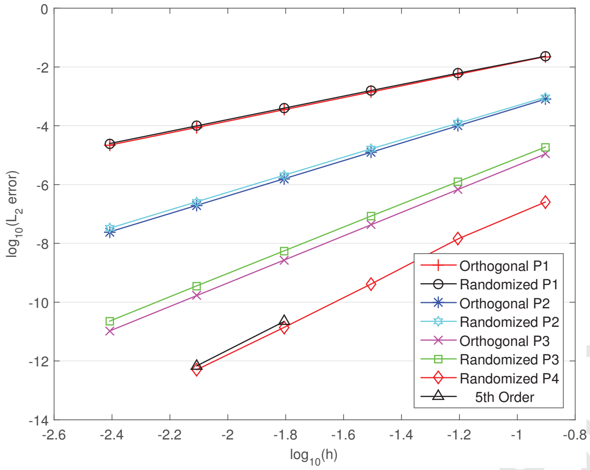

Figure 35: Convergence plots for the FDG scheme of varying polynomial orders for the MMS problem on both orthogonal and randomized grids.

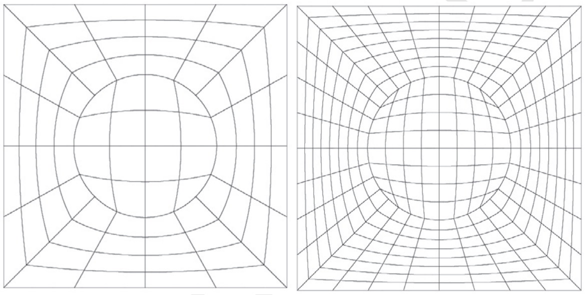

Figure 36: Examples of the pincell-like meshes used for the MMS problem.

condition number of the system.

6.6. KAIST Pincell

The solution used in Section 6.5 is infinitely differentiable, and therefore unrepresentative of typical transport solutions. The theoretically achievable convergence rate on topologically regular triangular meshes is given by [32]:

$$||\phi_h - \phi_{exact}||_{L_2} \leq Ch^{\min(p+1,r)} \tag{55}$$

where C is an arbitrary constant, h is a measure of the mesh size, p is the polynomial order of the basis functions employed and r is the regularity of the solution. With piecewise constant cross-sections r is at most $\frac{3}{2}$, and in the presence of voids or pure absorbers the exact transport solution can be discontinuous, giving $r = \frac{1}{2}$ [32]. This lowers the order of convergence attainable with the higher order schemes for realistic problems. To test this, an individual 3.3% enriched Uranium Oxide pincell from the KAIST 2B Benchmark problem was used [34]. This problem features a fuel pin, an extremely thin gas gap, cladding and moderator, shown in Figure 38, with seven groups. The problem was solved with an S_{16} level symmetric angular quadrature set [35], and a highly refined reference solution was calculated using the FDG scheme with 50688 elements and 5th order basis functions. In all cases the power iteration exit criteria for both the eigenvalue and scalar flux were set to 10^{-11}.

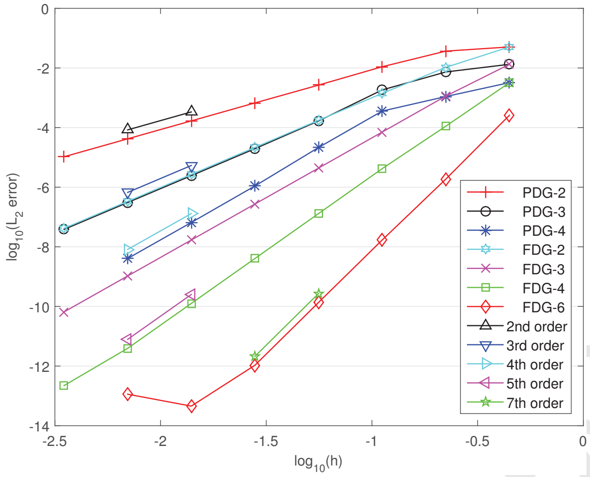

Figure 37: Convergence plots for the PDG and FDG schemes of varying polynomial orders for the MMS problem on the pincell meshes.

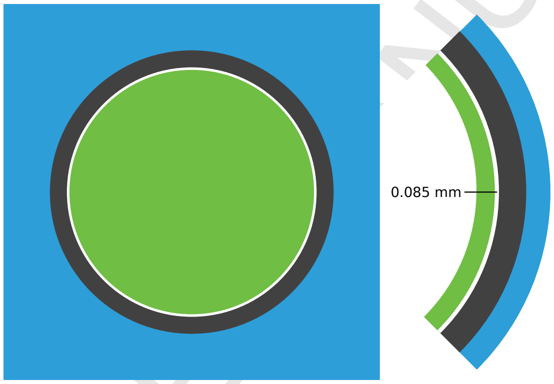

Figure 38: The KAIST pincell geometry, with green fuel, white gas, grey cladding and blue water.

The L_2 error ϵ_g for each group was then calculated using 16^2 Gaussian quadrature points per element on the fine mesh, as the shape functions are now NURBS, and therefore cannot be integrated exactly with Gaussian quadrature. A total L_2 error ϵ_2 was then calculated from these group values by the formula:

$$\epsilon_2 = \sqrt{\sum_{g=1}^{7} \epsilon_g^2} \tag{56}$$

The error in k_{eff} was also calculated from the eigenvalue obtained using the highly refined 5th order mesh.

Figure 39 shows how the 2 schemes converge as the mesh is refined for quadratic, cubic and quartic NURBS. In contrast to the MMS plots, the order of convergence of the schemes does not vary significantly as the polynomial order is increased. If the measure of h is taken to be proportional to $\frac{1}{\sqrt{E}}$ where E is the number of elements as in Section 6.5, the convergence rate is ≈ 0.7 for all of the schemes. The PDG schemes all behave very similarly, with little benefit seen from increasing the polynomial order. The FDG schemes display a much larger gain in accuracy from increasing the polynomial order, but the rate of convergence does not really change. While the gradients of the lines

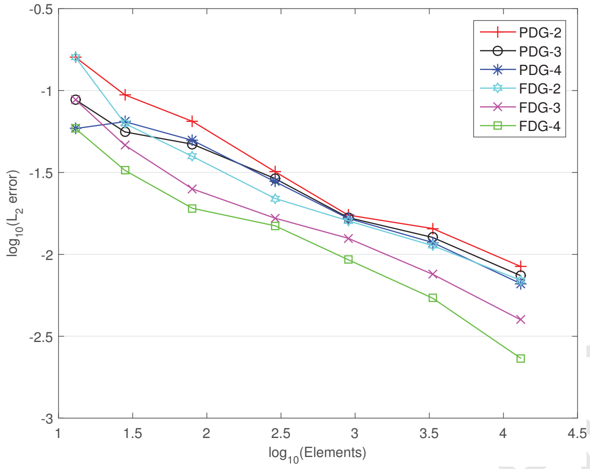

Figure 39: The L_2 error of the 2 schemes for various polynomial orders as the mesh is refined for the KAIST pincell problem.

seem to be different by the last 2 points, it is unclear if this is an actual difference in the rate of convergence, or an artefact of the refined reference solution only having 4 times as many elements as those that are being compared to it. Figure 40 shows that the error in the eigenvalue exhibits very similar behaviour to the L_2 errors.

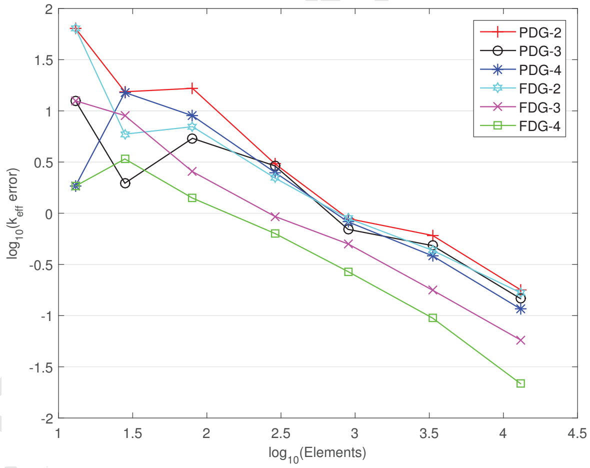

Figure 40: The error in k_{eff} (measured in pcm) of the 2 schemes for various polynomial orders as the mesh is refined for the KAIST pincell problem.

Comparisons like Figures 39 and 40 are not entirely fair, as there are $(p + 1)^2$ basis functions per element in the FDG scheme, and therefore larger dense matrices must be solved for with higher polynomial orders than lower ones. A fairer comparison is the time taken for one iteration, defined here to be one transport sweep for each direction and for each group, as this will be approximately proportional to the solution time. Timing comparisons will not be made here for the PDG scheme as the solution with this method requires solving fewer, much larger sparse matrices, for which an efficient algorithm has not yet been implemented.

Figure 41 shows the results of this comparison. It does not clearly show any particular polynomial order to be the most efficient for this problem, although it would appear that the 4th order scheme is becoming the most efficient at the highest refinement levels. However these refinement levels are far above what would realistically be used in a reactor physics problem, particularly for these higher order elements.

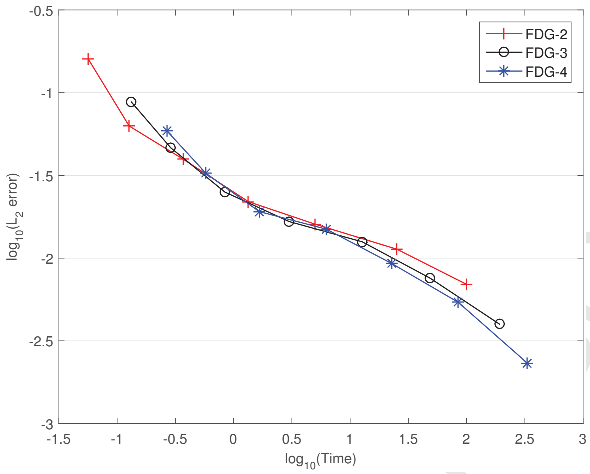

Figure 41: The L_2 error of various polynomial orders with the FDG scheme versus the time taken to perform a single iteration for the KAIST pincell problem.

6.7. Solution of the C5G7 Quarter Core Benchmark

The C5G7 is a quarter-core OECD transport benchmark, designed to test multigroup deterministic transport codes [36]. It consists of four fuel assemblies of MOX and UOX fuel, each containing a lattice of 17x17 fuel pins and guide tubes. Each pincell contains a circle of radius 0.54 cm inside a square of moderator with cell pitch 1.26 cm. These assemblies are surrounded on two sides by a water reflector, and symmetry conditions are imposed on the other two boundaries to simulate the presence of the rest of the core (see Figure 42). The problem has seven groups, and two separate angular quadrature sets were used. In order to perform spatial convergence studies and DGFEM comparisons, an S_8 level symmetric quadrature set was used [35]. The IGA spatial meshes used in the pins are shown in Figure 43, while straight sided DGFEM meshes that were as similar as possible were also used. In the DGFEM case, the fissile mass was preserved in each of the pins, as large errors are introduced if this is not the case. While this is trivial in 2D with circles, it is worth noting that fissile mass preservation is automatic with IGA, which could be a potential advantage in other problems or in three dimensions. In the reflector, elements were kept square to facilitate the comparison between the IGA and DGFEM methods. These meshes were then compared to a reference solution calculated using quadratic IGA, utilising 1,152 elements per pincell, which equates to 2,164,032 elements in the whole geometry. Calculations were performed using only the FDG scheme, as the results from the previous sections suggest that the PDG scheme does not converge as quickly as the FDG scheme.

In all cases the power iteration exit criteria for both the eigenvalue and scalar flux were set to 10^{-11}. The reference calculation quantities of interest, after normalisation as in [36], are shown in table 3.

The errors in various quantities of interest are plotted against the number of elements in the mesh in Figures 44-47. In all cases once the mesh is reasonably refined there is very little difference between the quadratic IGA and the quadratic FEM. This is to be expected, as the finer the spatial mesh the closer the two schemes become as the geometry approximation in the FEM gets better and better. Surprisingly the exact geometry of the IGA scheme seems to have little impact from the third spatial mesh onwards, with significant differences in errors only observed on the coarsest two meshes.

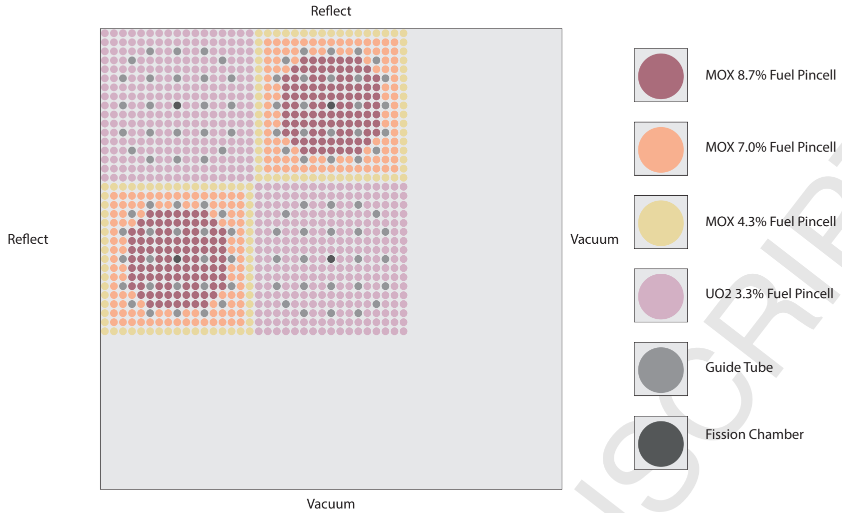

Figure 42: The geometry of the C5G7 benchmark. Reflective boundaries are on the top and left sides, with vacuum boundaries on the bottom and right sides.

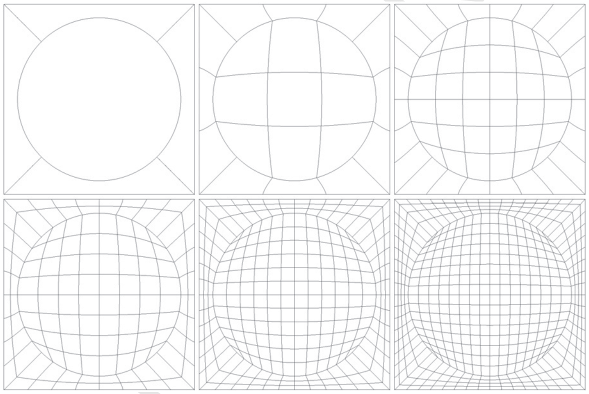

Figure 43: The 6 meshes used to study the spatial convergence of the IGA scheme on this problem. DGFEM meshes were formed that were as similar as possible to these for comparison purposes.

Conversely, for the same spatial mesh the linear FEM converges much more slowly to the reference solution in all quantities of interest. On the coarser meshes it is approximately an order of magnitude less accurate, dropping to half an order of magnitude on the most refined meshes. Both Figures 44 and 45 contain an anomalously accurate result on the second and third most refined meshes respectively. In both cases this is due to a change in sign of the error, such that at some point the Linear FEM numerical solution crosses the reference solution in both of these quantities of interest.

An S_{44} triangular-Chebyshev quadrature set (1012 directions) was then used in order to get as close to the PDT calculated reference solution [37] as possible. Initially comparisons were performed to the MCNP reference solution [36]. However the maximum error in the pin powers plateaued around 0.8% as both angular and spatial fidelity was increased. A much closer agreement was obtained with the PDT reference solution. A likely cause for the disparity between these solutions is too high a variance in the individual pin powers in the MCNP solution for a comparison at this level. These calculations were performed using the FDG scheme on the first 2 meshes from Figure 43, as well as the third with 1 extra knot insertion in the moderator patches to try and reduce the spatial error further.

	Reference Result
K_{eff}	1.183960
Max Pin Power	2.518297
Min Pin Power	0.2315063
UO2-1 Power	495.5315
MOX Power	210.5186
UO2-2 Power	139.4312

Table 3: The quantities of interest calculated using an S_8 level symmetric quadrature set with 1,152 quadratic IGA elements per pincell, which equates to 2,164,032 elements in the whole geometry.

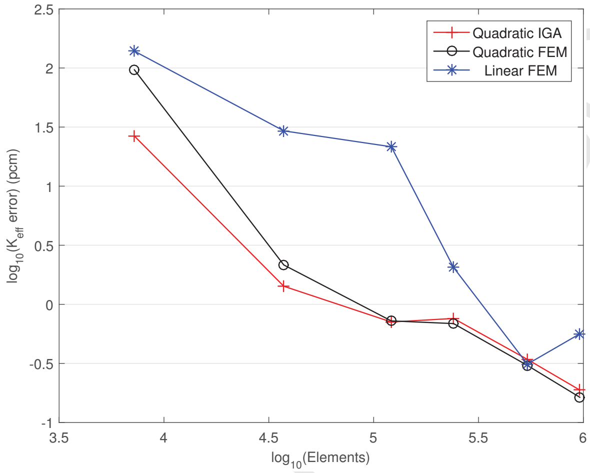

Figure 44: k_{eff} convergence as the spatial mesh is refined for the C5G7 benchmark using an S_8 level symmetric quadrature set. The linear FEM gives an anomalously good result with the penultimate mesh as the quadratic schemes converge monotonically to the reference solution, while the linear FEM error changes in sign at that point.

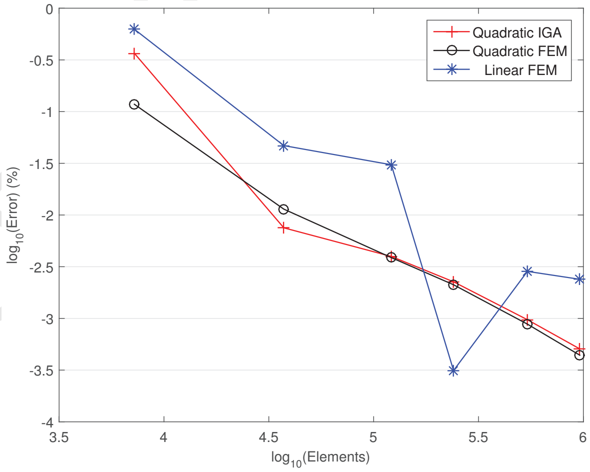

Figure 45: Maximum pin power convergence as the spatial mesh is refined for the C5G7 benchmark using an S_8 level symmetric quadrature set. As in Figure 44, the linear FEM has one anomalously good result where the sign of the error changes.

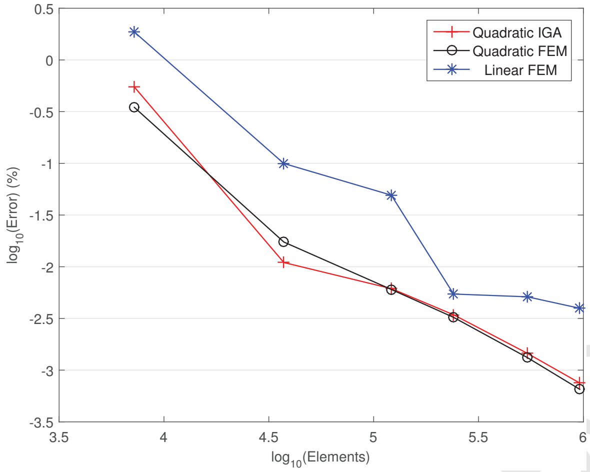

Figure 46: The largest power error in any pin as the spatial mesh is refined for the C5G7 benchmark using an S_8 level symmetric quadrature set.

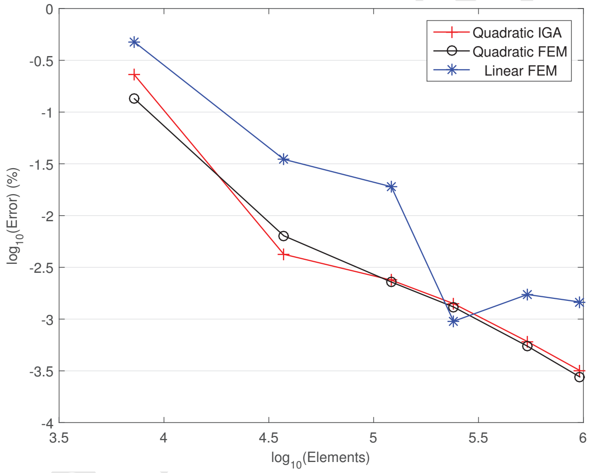

Figure 47: The average pin power error as the mesh is refined for the C5G7 benchmark using an S_8 level symmetric quadrature set.

These results are presented in Table 4, along with Figure 48 for the assembly power errors. The definitions of the average error (AE), root mean square error (RMS) and mean relative error (MRE) are taken from the original benchmark description [36]. The power iteration exit criteria were set at 10^{-8} for both the eigenvalue and scalar flux convergence.

A significant drop in the errors in all of the measured parameters is seen between the coarse and the intermediate meshes in table 4. However there is very little difference in the results between the intermediate and fine meshes solutions, and so this mesh seems to be spatially converged for these quantities of interest for this angular quadrature set. This error convergence can also be seen in the pin power error maps in Figure 49.

These results indicate that for realistic reactor physics calculations, in which the eigenvalue is required to be accurate to within tens of pcm and the pin power errors must be around 1%, the intermediate spatial mesh from this study would be suitable, and possibly even the coarsest mesh.

Mesh Level	**Coarse**	**Intermediate**	**Fine**
Elements	7,225	37,281	149,124
Elements per pincell	5	21	84
K-Effective	1.18670	1.18632	1.18630
PCM Error	21	-11	-13
Maximum Pin Power	2.4906	2.4995	2.4995
% Error	-0.33	0.028	0.029
Minimum Pin Power	0.2323	0.2317	0.2317
% Error	0.235	-0.019	-0.020
Largest Pin Error (%)	0.491	0.0436	0.0450
AE	0.203	0.0172	0.0180
RMS	0.235	0.0200	0.0210
MRE	0.203	0.0168	0.0176

Table 4: Quantities of interest with comparisons to the PDT reference solution [37] for an S_{44} triangular-Chebyshev angular quadrature set and quadratic IGA spatial discretisation.

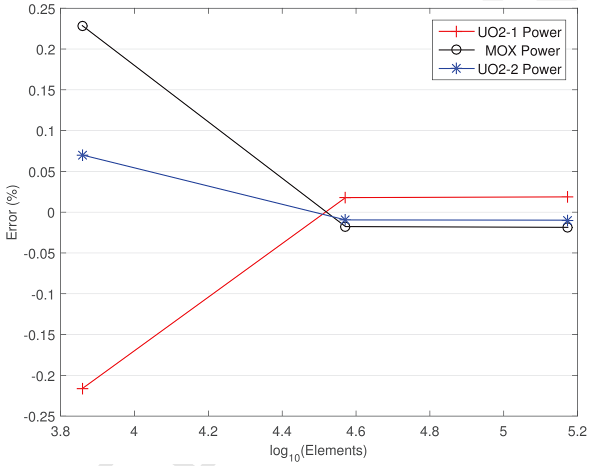

Figure 48: The percentage error of the power in each assembly as the spatial mesh is refined for the C5G7 benchmark using an S_{44} triangular-Chebyshev quadrature set.

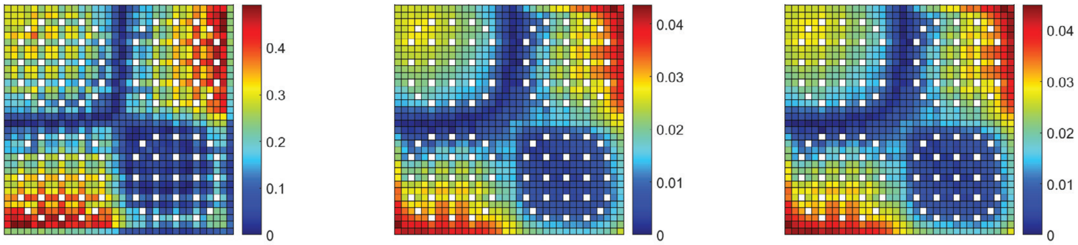

Figure 49: The pin power error distribution over the four assemblies for the coarsest to finest mesh from left to right for the C5G7 benchmark using an S_{44} triangular-Chebyshev quadrature set.

7. Conclusion

In this paper discontinuous Galerkin isogeometric analysis methods have been developed and applied within the field of neutron transport; although the method is generally applicable to all hyperbolic systems of equations. The discontinuous Galerkin numerical discretisation schemes for IGA were developed on both the patch level (PDG) and element level (FDG). The numerical dispersive and diffusive properties of both schemes were compared to the dispersion relationship for a time-dependent one-dimensional analytical hyperbolic wave equation with an absorption term and periodic boundary conditions. It was found that the FDG scheme exhibited lower dispersive and diffusive numerical errors than the PDG scheme on an equivalent mesh.

The convergence of both schemes were compared using an analytical solution to the one-dimensional Reed Cell problem, which is a demanding transport dominated test case. The numerical results of applying the PDG and FDG schemes to the Reed cell problem indicated that for basis functions that are cubic order and higher the FDG scheme exhibited a higher-order convergence than the equivalent PDG scheme on an element level basis. It is also important to note that the PDG scheme has an improved order of convergence if this comparison is performed on a degree of freedom basis. However, this then requires the solution of large sparse matrices.

Further analysis of the both the FDG and PDG schemes in multi-dimensions was performed using the method of manufactured solutions. The method of manufactured solutions, applied to both schemes using an infinitely differentiable solution, demonstrated that the PDG scheme of polynomial order p achieved a convergence rate of p. However, the FDG scheme with the same polynomial order achieved a convergence rate of $p + 1$. For both the FDG and PDG schemes the overall accuracy achieved was only marginally lower when using non-orthogonal compared to orthogonal (or Cartesian) meshes. On meshes where elements form cycles for some or all discrete ordinate directions, solving these strongly connected components has been found to be inefficient compared to sweeping through the mesh once in each direction.

Curved boundaries are very common in IGA meshes and the analysis of both the PDG and FDG schemes indicate that using an average normal vector for the surfaces of bounding NURBS volumes reduces the computational time required to perform the sweeps. However, using an average normal vector also reduces the order of accuracy of both schemes to first-order regardless of the polynomial order of the NURBS basis functions used in the schemes. This is caused by the first order convergence of the average normal to the correct normal direction at each quadrature point, leading to poor accuracy of surface integration of the fluxes.

Finally, two realistic reactor physics verification benchmarks were investigated using both the PDG and FDG schemes. The first verification problem was taken from the KAIST2B reactor physics benchmark suite and consisted of a single 3.3% enriched uranium fuel pin from this benchmark. This problem is challenging as it consists of the fuel pin, a very thin gas gap and surrounding cladding and moderator. The problem was solved using an S_{16} level symmetric angular quadrature set. A highly refined reference solution was calculated for the problem using the FDG scheme. The numerical results from applying the PDG and FDG schemes to this verification test case indicated that both the PDG and FDG yield the same order of accuracy for all NURBS basis functions regardless of polynomial order. This is due to the underlying regularity of the transport solution in this case. However, it is important to note that the errors observed in the FDG solutions of the KAIST2B pincell problem are consistently lower than the corresponding PDG solutions.

The second verification test case was the OECD quarter core C5G7 reactor physics benchmark which consists of two MOX fuel assemblies and two UOX fuel assemblies in a colour-set configuration with a water reflector. Comparisons of the convergence rates of the FDG scheme using quadratic NURBS with linear and quadratic discontinuous Galerkin finite elements were performed for this verification test case. The DGFEM meshes were locally modified to ensure that the fissile mass of the pincells and the volumes of the moderator regions were preserved. The quadratic IGA and DGFEM schemes converge more rapidly for all quantities of interest, such as K_{eff} and reaction rates, than the linear DGFEM scheme, as one would expect. However, no discernible difference was observed in the convergence rates between the quadratic IGA and DGFEM solutions once more than sixty elements were used per pincell. Even on the coarser meshes the differences in the spatial errors between the quadratic IGA and DGFEM schemes is relatively small. However, reactor physics problems usually require levels of accuracy on approximately this scale. Therefore, further numerical investigations are required to fully ascertain the most efficient scheme to reduce the numerical errors to acceptable levels for typical industrial reactor physics applications.

The PDG and FDG schemes for the IGA methods hold great potential for the future in that the methods can be applied to a wide variety of advection dominated phenomena whilst exactly modelling the underlying geometry provided by CAD software. The initial application of the schemes were for the neutron transport equation with application to reactor physics and shielding problems. However, as the neutron transport equation is essentially a hyperbolic equation, the same schemes could be applied to other related fields such as the Euler equations of gas dynamics [16].

The DGFEM comparisons presented here suggest that there is little to be gained in accuracy by using discontinuous IGA of the same polynomial order for these problems. It is suspected a cancellation of errors in the DGFEM scheme is contributing to this similarity, and that a greater difference would be observed in problems that are less forgiving of DGFEM's inexact geometry. However the fact that IGA meshes are exact from the coarsest level of refinement naturally lends itself to Adaptive Mesh Refinement (AMR) schemes. When the elements used in a DGFEM scheme cannot exactly represent the geometry, AMR schemes must revisit the CAD description to refine the mesh. Otherwise there is no improvement in the geometrical representation, which may dominate the errors in the solution. IGA does not face this restriction, and so future work will focus on AMR for discontinuous IGA.

Acknowledgements

A.R. Owens would like to acknowledge the support of EPSRC under their industrial doctorate programme, Rolls-Royce for industrial support and the Imperial College High Performance Computing Service for technical support. J.A. Welch would like to acknowledge EPSRC through the Doctoral Training Award (DTA) PhD scheme and also the support of Rolls-Royce. M.D. Eaton and J. Kópházi would like to thank EPSRC for their support through the following grants: "Adaptive Hierarchical Radiation Transport Methods to Meet Future Challenges in Reactor Physics" (EPSRC grant number: EP/J002011/1) and "RADIANT: A Parallel, Scalable, High Performance Radiation Transport Modelling and Simulation Framework for Reactor Physics, Nuclear Criticality Safety Assessment and Radiation Shielding Analyses" (EPSRC grant number: EP/K503733/1).

References

[1] Richard Sanchez. Assembly homogenization techniques for core calculations. *Progress in Nuclear Energy*, 51:14–31, 2009.

[2] K. S. Smith. *Spatial Homogenization Methods for Light Water Reactor Analysis*. PhD thesis, Massachusetts Institute of Technology, 1980.

[3] B.G. Carlson and K.D. Lathrop. Transport Theory: The Method of Discrete Ordinates. Technical report, Los Alamos Scientific Laboratory, 1965.

[4] Todd A. Wareing, John M. McGhee, Jim E. Morel, and Shawn D. Pautz. Discontinuous Finite Element S_N Methods on Three-Dimensional Unstructured Grids. *Nuclear Science and Engineering*, 138:256–268, 2001.

[5] W.H. Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. Technical report, Los Alamos Scientific Laboratory, 1973.

[6] B.Q. Li. *Discontinuous finite elements in fluid dynamics and heat transfer*. Springer-Verlag, 2006.

[7] P.J.Frey and P-L.George. *Mesh generation: application to finite elements*. Wiley, 2008.

[8] N.A. Peterson and K. Kyle. Detecting translation errors in cad surfaces and preparing geometries for mesh generation. Technical report, Lawrence Livermore National Labs, 2001.

[9] Moller J-Y. and Lautard J-J. MINARET, A Deterministic Neutron Transport Solver For Nuclear Core Calculations. In *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, 2011.

[10] K. Manalo, M. Chin, and G. Sjoden. Cartesian Meshing Impacts for PWR Assemblies in Multigroup Monte Carlo and Sn Transport. In *Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2013*, 2013.

[11] S.J.Sherwin and J.Peiró. Mesh generation in curvilinear domains using high-order elements. *International Journal for Numerical Methods in Engineering*, 53:207–223, 2002.

[12] G.E.M. Karniadakis and S. Sherwin. *Spectral/hp element methods for computational fluid dynamics*. Oxford University Press, 2012.

[13] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geoemtry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.

[14] Y. Bazilevs, K. Takizawa, and T.E. Tezduyar. *Computational Fluid-Structure Interaction: methods and applications*. John Wiley and Sons, 2013.

[15] F. Brunero. Discontinuous galerkin methods for isogeometric analysis. Master's thesis, Università degli Studi di Milano, 2012.

[16] Culbert B. Laney. *Computational Gasdynamics*. Cambridge University Press, 1998.

[17] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible navier-stokes equations. *Journal of Computational Physics*, 131:267–279, 1997.

[18] S.K. Hall, M.D. Eaton, and M.M.R. Williams. The application of isogeometric analysis to the neutron diffusion equation for a pincell problem with an analytic benchmark. *Annals of Nuclear Energy*, 2012.

[19] Les Piegl and Wayne Tiller. *The NURBS Book: Monographs in Visual Communication*. Springer, 1997.

[20] M.G. Cox. The Numerical Evaluation of B-Splines. *Journal of Applied Mathematics*, 10:134–149, 1972.

[21] Carl de Boor. On Calculating with B-Splines. *Journal of Approximation Theory*, 6:50–62, 1972.

[22] J. Austin Cottrell, Thomas J.R. Hughes, and Yuri Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley, 2009.

[23] T.J.R. Hughes, A. Reali, and G. Sangalli. Efficient quadrature for NURBS-based isogeomtric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199:301–313, 2010.

[24] James J. Duderstadt and William R. Martin. *Transport Theory*. Wiley, 1979.

[25] Fang Q. Hu, M.Y. Hussaini, and Patrick Rasetarinera. An Analysis of the Discontinuous Galerkin Method for Wave Propogation Problems. *Journal of Computational Physics*, 151:921–946, 1999.

[26] Spencer Sherwin. *Discontinuous Galerkin Methods: Theory, Computation and Applications*. Springer, 2000.

[27] R.C.Moura, S.J.Sherwin, and J.Peiró. Linear dispersion-diffusion analysis and its application to under-resolved turbulence simulations using discontinuous galerkin spectral/hp methods. *Journal of Computational Physics*, 2015.

[28] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *Data Structures and Algorithms*. Addison-Wesley, 1987.

[29] Jean-Yves Moller. *Eléments finis courbes et accélération pour le transport de neutrons*. PhD thesis, Université Henri Poincaré, 2012.

[30] James S. Warsa. A Continuous Finite Element-Based, Discontinuous Finite Element Method for S_N Transport. *Nuclear Science and Engineering*, 160:385–400, 2008.

[31] James. S. Warsa. Analytical S_N solutions in heterogeneous slabs using symbolic algebra computer programs. *Annals of Nuclear Energy*, 29:851–874, 2002.

[32] 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.

[33] Kambiz Salari and Patrick Knupp. Code Verification by the Method of Manufactured Solutions. Technical report, Sandia National Laboratories, 2000.

[34] Nam Zin Cho. Benchmark Problem 2B: MOX Fuel-Loaded Small PWR Core. Technical report, Korea Advanced Institute of Science and Technology, 2000.

[35] E.E. Lewis and Jr. W.F. Miller. *Computational Methods of Neutron Transport*. American Nuclear Society, Inc., 1993.

[36] E.E. Lewis, M.A. Smith, N. Tsoulfanidis, G. Palmiotti, T.A. Taiwo, and R.N. Blomquist. Benchmark on Deterministic Transport Calculations Without Spatial Homogenisation. Technical report, Organisation for Economic Co-Operation and Development, 2003.

[37] Carolyn N. McGraw, Marvin L. Adams, W. Daryl Hawkins, Michael P. Adams, and Timmie Smith. Accuracy of the Linear Discontinuous Galerkin Method for Reactor Analysis with Resolved Fuel Pins. In *PHYSOR 2014 - The Role of Reactor Physics toward a Sustainable Future*, 2014.