Available online at www.sciencedirect.com

ScienceDirect

Comput. Methods Appl. Mech. Engrg. 318 (2017) 215–241

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

An adaptive, hanging-node, discontinuous isogeometric analysis method 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

Received 23 May 2016; received in revised form 27 January 2017; accepted 30 January 2017

Available online 4 February 2017

Highlights

- A DG hanging-node IGA discretisation is developed for hyperbolic equations.
- A conservative upwinding scheme across the curved element boundaries is established.
- Optimal convergence rates are achieved when the underlying solution is smooth.
- Adaptive formulation outperforms uniform refinement for a variety of test cases.
- Comparisons made to DG finite elements.

Abstract

In this paper a discontinuous, hanging-node, isogeometric analysis (IGA) method is developed and applied to the first-order form of the neutron transport equation with a discrete ordinate (S_N) angular discretisation in two-dimensional space. The complexities involved in upwinding across curved element boundaries that contain hanging-nodes have been addressed to ensure that the scheme remains conservative. A robust algorithm for cycle-breaking has also been introduced in order to develop a unique sweep ordering of the elements for each discrete ordinates direction. The convergence rate of the scheme has been verified using the method of manufactured solutions (MMS) with a smooth solution. Heuristic error indicators have been used to drive an adaptive mesh refinement (AMR) algorithm to take advantage of the hanging-node discretisation. The effectiveness of this method is demonstrated for three test cases. The first is a homogeneous square in a vacuum with varying mean free path and a prescribed extraneous unit source. The second test case is a radiation shielding problem and the third is a 3 × 3 "supercell" featuring a burnable absorber. In

* Corresponding author.
E-mail address: a.owens12@imperial.ac.uk (A.R. Owens).

http://dx.doi.org/10.1016/j.cma.2017.01.036

the final test case, comparisons are made to the discontinuous Galerkin finite element method (DGFEM) using both straight-sided and curved quadratic finite elements.

Keywords: Adaptive; Hanging-node; Isogeometric analysis; Hyperbolic equations; Neutron transport; Discrete ordinates

1. Introduction

The numerical solution of the neutron transport equation for heterogeneous whole core reactor physics and radiation shielding problems remains a significant computational challenge even for today's numerical algorithms and high performance computing hardware architectures. This is primarily due 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. Current numerical methods deployed in reactor physics involve first solving the neutron transport equation for fuel assembly sized domains with prescribed periodic boundary conditions [1,2]. The heterogeneous material cross section data for the fuel assembly is then homogenised for use in whole core reactor physics problems. The whole core reactor physics problem is then often solved using the neutron diffusion equation with the assembly averaged or homogenised material cross section data by applying nodal discretisation methods [1,3].

The discontinuous Galerkin finite element method (DGFEM) is a commonly used approach if the geometric features are to be modelled directly. DGFEM was originally developed and applied to the discretisation of the first-order form of the neutron transport equation with a discrete ordinates angular discretisation [4,5]. It has since been applied to a wide variety of hyperbolic systems, such as the compressible Navier–Stokes equations [6], the Euler equations of gas dynamics [7] and radiative heat transfer [8].

A wide range of complex geometries can be modelled in the finite element method using basic geometric primitives such as triangles and quadrilaterals in two dimensions and pyramids, prisms, hexahedra and tetrahedra in three dimensions [9]. This is achieved by the use of automatic mesh generation software that generates surface and volumetric meshes of finite elements from the underlying Non-Uniform Rational B-Spline (NURBS) surface descriptions produced by Computer Aided Design (CAD) programs [9]. However, the underlying geometry is often not exactly reproduced when it is discretised into basic geometric primitives as these usually are linear elements with straight or planar sides [9]. This is of particular importance in reactor physics applications, where the fissile mass of the system must be preserved by the polygonal approximations to the geometry in order to obtain accurate criticality solutions [10,11]. This can be corrected by locally modifying the mesh to preserve the fissile mass, but this is a non-trivial task for complex geometries.

Recently, the finite element method (FEM) has been extended to allow exact preservation of the CAD geometry using isogeometric analysis [12]. As in FEM, the weak form of the governing partial differential equation (PDE) is discretised using prescribed shape and test functions to form a system of equations. To preserve the CAD geometry, the same NURBS used in the CAD program are used as the shape and test functions in the spatial discretisation of the PDE. This allows the geometry to be modelled exactly from the coarsest level of refinement, as well as maintaining this exact representation as the mesh is refined.

Since its introduction IGA has been applied to a wide variety of different physical phenomena, including computational fluid dynamics [12], computational solid mechanics [12] and coupled solid–fluid interaction problems [13]. A variety of different IGA discretisations have been developed. These were initially restricted to Bubnov–Galerkin type IGA discretisations for solid mechanics problems and streamline upwind Petrov–Galerkin discretisations for advection–diffusion problems [12]. In the field of reactor analysis, Hall et al. applied a Bubnov–Galerkin projection using NURBS to the one group neutron diffusion equation for a simple pincell problem [14]. This was extended by Welch et al. to a multigroup scheme for large heterogeneous quarter-core reactor physics problems [15].

Discontinuous IGA methods have recently been developed for elliptic systems of equations [16,17]. For hyperbolic systems, two discontinuous IGA discretisations have recently been developed. The Blended Isogeometric Discontinuous Galerkin method [18] uses rational Bernstein–Bézier triangles to mesh the geometry. The authors then used standard polynomial spaces to approximate solution fields to the acoustic wave equations and Maxwell's equations of electromagnetics. The work presented here is an extension of [19], in which a discontinuous Galerkin IGA method was applied to a hyperbolic system of equations, specifically the first-order form of the neutron transport equation with a discrete ordinates (S_N) angular discretisation.

A major drawback of the spatial discretisation presented in [19], as well as in NURBS-based IGA in general is that the tensor product structure of the basis functions leads to the propagation of mesh refinement throughout the entire physical domain. This can lead to computational effort being wasted by over-refining parts of the physical domain that are of little importance to the accuracy of the solution. For continuous discretisations, existing techniques for dealing with this refinement propagation include T-splines [20] and LR B-splines [21].

In a discontinuous setting, the simpler hanging-node approach [22] can be taken to stop this refinement propagation. This has the additional advantage of enabling adaptive mesh refinement (AMR) techniques to refine the mesh, as each element in the mesh can be refined independently of its neighbours. This allows mesh refinement to be prioritised in regions of the physical domain that have a large error associated with them, as calculated by a selection of heuristic error indicators.

The remainder of this paper is organised as follows. Section 2 explains the underlying concepts of isogeometric analysis methods and the NURBS basis functions the method uses. Section 3 derives the spatially discretised system of equations from the mono-energetic neutron transport equation and details the upwinding methodology in the presence of hanging-nodes. Section 4 describes the method used to determine a unique sweep ordering through the mesh for each discrete ordinates direction. Section 5 describes the AMR strategy employed as well as the error indicators used. Section 6 presents numerical results of applying the adaptive hanging-node method to a variety of verification test cases.

2. Isogeometric analysis

Isogeometric analysis was introduced in an attempt to unify the geometric description of physical problems, utilised in CAD programs, with the computational analysis and numerical discretisation of the governing equations [12]. This unification was achieved by using the NURBS representation for both the underlying geometry and the numerical discretisation method. A large proportion of the design and analysis cycle time is typically taken up with generating a gap free NURBS representation of the geometry that can then be meshed using a mesh generator [12]. This is particularly the case if adaptivity of the spatial mesh is required, or if the geometry changes in a time dependent problem. In these cases the CAD geometry may need to be revisited every time the mesh changes if there are curved boundaries. This is due to the fact that conventional finite elements cannot exactly represent conic sections such as circles and ellipses or quadric surfaces such as cylinders, spheres and ellipsoids. These are all commonly found in reactor physics, nuclear criticality safety assessment and radiation shielding applications. In particular in reactor physics problems, the fissile mass of the system must be conserved in order to obtain an accurate solution [10]. Even if the fissile mass is conserved, the geometry is often still poorly approximated, and so the leakage of neutrons through surfaces will contain an error. The impact of this imperfect geometrical representation will be covered in Section 6.4.

As IGA uses the same NURBS representation as the CAD geometry to discretise the weak form of the S_N equations, it can exactly represent the aforementioned geometric primitives, as well many others [23]. As a direct result of this, the geometry used by the analysis program is exact from the coarsest level of refinement. This allows an ancillary mesh generator to be completely bypassed as any refinement can be performed directly in the analysis program. In contrast, the finite element method requires a mesh generator to be employed every time a more accurate representation of the geometry is required. Finally, the parameterisation of the mapping from the parametric to the physical space does not change under h or p refinement when using IGA [12]. This simplifies the implementation of a hanging-node formulation with an arbitrary distribution of nodes on a given element face.

2.1. B-splines

Three properties completely define a B-spline curve, also known as a patch [23]:

1. A polynomial order p.
2. A set of n control points in physical space $\{\mathbf{B}_i : i = 1, \dots, n\}$, where n is the number of basis functions defining the B-Spline curve.

3. A knot vector Ξ, which is 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, in which the first and last knot value are repeated $p+1$ times. In this case, the knot vector will have length $n+p+1$, $\Xi = (\xi_1, \xi_2, \dots, \xi_{n+p+1})$.

B-Spline basis functions in parametric space are completely defined by their polynomial order and knot vector. A uniform knot vector has equally spaced knot points, otherwise it is non-uniform [23]. Knot points separate the parametric space into knot spans. As these knot points are points of reduced continuity, the knot spans can be considered to be analogous to elements in traditional FEM. If a knot point is repeated m times, it has multiplicity m and the basis functions are C^{p-m} continuous at this point. In particular, if a knot point has multiplicity $m = p+1$ then the basis functions will be discontinuous at this point, and the patch is split into two new patches. Denoting basis function i of polynomial order p by $N_{i,p}(\xi)$, then the basis functions are given by the following Cox–de Boor recursion formula [24,25] for $p = 0$:

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

and for $p \ge 1$

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

Due to the recursive nature of their construction, the derivatives of B-splines can be represented in terms of lower order B-splines as follows [23]:

$$\frac{d}{d\xi} N_{i,p}(\xi) = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi). \tag{3}$$

Terms of the form $0/0$ can arise in Eqs. (2) and (3), in which case the term is defined to be zero. These basis functions form a partition of unity at all values of ξ within the knot vector [23], that is:

$$\sum_{i=1}^{n} N_{i,p}(\xi) = 1 \quad \forall \xi \in [\xi_1, \xi_{n+p+1}]. \tag{4}$$

The basis functions are pointwise non-negative, which means that every term in a mass matrix is also non-negative, which can be useful for mass lumping schemes [26]. The basis functions are also interpolatory at the end points of the knot vector, such that $N_{i,p}(\xi_1) = \delta_{i,1}$ and $N_{i,p}(\xi_{n+p+1}) = \delta_{i,n}$.

B-Spline curves are then defined by associating each basis function with a control point in physical space and summing the product of the two:

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

Higher dimensional B-Spline objects are constructed from one dimensional B-splines in a tensor product fashion. In two dimensions, polynomial orders p and q, knot vectors $\Xi = (\xi_1, \xi_2, \dots, \xi_{n+p+1})$ and $H = (\eta_1, \eta_2, \dots, \eta_{m+q+1})$ and a control net $\{\mathbf{B}_{i,j} : i = 1, \dots, n \text{ and } j = 1, \dots, m\}$ 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 $Z = (\zeta_1, \zeta_2, \dots, \zeta_{l+r+1})$ and changing to a control lattice $\{\mathbf{B}_{i,j,k} : i = 1, \dots, n,\ j = 1, \dots, m \text{ and } k = 1, \dots, l\}$ 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, pointwise non-negativity and interpolatory properties extend into 2 and 3 dimensions due to the tensor product structure. In particular, the interpolatory nature of the basis functions limits the number of non-zero basis functions on each face of an element. This can improve the memory efficiency of a sweep based solver by reducing the size of the mass matrix that must be stored on each face of each element for upwinding.

2.2. NURBS

While B-splines can be used to represent a wide variety of geometries, NURBS are needed to exactly represent conic sections. To construct NURBS from B-splines, a weight is associated with each control point, denoted by ω_i, $\omega_{i,j}$ and $\omega_{i,j,k}$ in one, two or three dimensions respectively. One dimensional NURBS basis functions of order p are then defined by [23]:

$$R_i^p(\xi) = \frac{N_{i,p}(\xi)\omega_i}{\sum\limits_{\hat{i}=1}^{n} N_{\hat{i},p}(\xi)\omega_{\hat{i}}} \tag{8}$$

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:

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

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

NURBS surfaces and solids are then constructed by associating each basis function with a control point, analogously to Eq. (9).

All of the problems considered in this work are two-dimensional, and in practice can be constructed from the union of biquadratic ($p = 2$) NURBS surface patches using the open knot vectors $\Xi = H = (0, 0, 0, 1, 1, 1)$. As the length of an open knot vector is $n + p + 1$, there will be 3 basis functions in each parametric direction, for a total of 9 basis functions per patch. An example of a pincell geometry and the associated control points for two of the patches are given in Fig. 1 and Table 1 respectively.

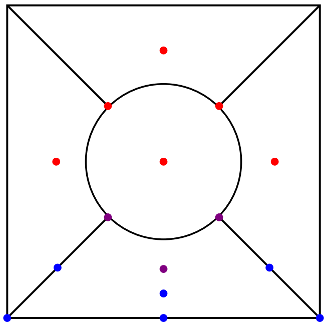


Fig. 1. The 5 biquadratic patches used to create a pincell geometry with a pin of radius 1 cm and a cell pitch of 4 cm. The control points for the circle (red) and one of the moderator patches (blue) have been marked on, with control points used by both patches marked in purple. The origin is taken to be at the centre of the circle, and the physical coordinates and weights of these control points are given in Table 1. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
The physical coordinates and weights for the control points of two of the patches shown in Fig. 1. Control point (i, j) is numbered as in Eq. (10a).

Control point	Circle		Moderator	
	(x, y) coordinate	Weight	(x, y) coordinate	Weight
(1, 1)	$\left(\frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}\right)$	1	$(-2, -2)$	1
(2, 1)	$\left(0, -\sqrt{2}\right)$	$\frac{1}{\sqrt{2}}$	$(0, -2)$	1
(3, 1)	$\left(\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}\right)$	1	$(2, -2)$	1
(1, 2)	$\left(-\sqrt{2}, 0\right)$	$\frac{1}{\sqrt{2}}$	$\left(\frac{1}{2}\left(-2-\frac{1}{\sqrt{2}}\right), \frac{1}{2}\left(-2-\frac{1}{\sqrt{2}}\right)\right)$	1
(2, 2)	$(0, 0)$	1	$\left(0, \frac{1}{2}(-2-\sqrt{2})\right)$	$\frac{1}{2}\left(1+\frac{1}{\sqrt{2}}\right)$
(3, 2)	$\left(\sqrt{2}, 0\right)$	$\frac{1}{\sqrt{2}}$	$\left(\frac{1}{2}\left(2+\frac{1}{\sqrt{2}}\right), \frac{1}{2}\left(-2-\frac{1}{\sqrt{2}}\right)\right)$	1
(1, 3)	$\left(\frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$	1	$\left(\frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}\right)$	1
(2, 3)	$\left(0, \sqrt{2}\right)$	$\frac{1}{\sqrt{2}}$	$\left(0, -\sqrt{2}\right)$	$\frac{1}{\sqrt{2}}$
(3, 3)	$\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$	1	$\left(\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}\right)$	1

The interpolatory nature of the basis functions at the end points of the knot vectors forces the corner control points ((1, 1), (3, 1), (1, 3) and (3, 3)) to coincide with the corners of the physical patch. The mid-side control points ((2, 1), (1, 2), (3, 2) and (2, 3)) of the circle are then fixed by the equation for a circle arc using quadratic NURBS [26], as is control point (2, 3) of the moderator patch. The remaining control points all have some freedom available in their (x, y) coordinates and weight. The values given in Table 1 have been chosen based on symmetry considerations and, in the case of control point (2, 2) of the moderator patch, an attempt to give a smooth variation in the Jacobian over the patch.

When these patches are refined, knot points are inserted with multiplicity $p + 1$ [23], so that the basis functions are discontinuous between each element as in the fully discontinuous Galerkin scheme described in [19].

3. Discrete ordinates equations & weak form

In the following derivation vector quantities are in bold while matrices are double underlined. The steady state, mono-energetic neutron transport equation with isotropic scattering and fixed (extraneous) source is given by [27]:

$$\boldsymbol{\Omega} \cdot \nabla\psi(\mathbf{r}, \boldsymbol{\Omega}) + \Sigma_t(\mathbf{r})\psi(\mathbf{r}, \boldsymbol{\Omega}) = \frac{\Sigma_s(\mathbf{r})}{4\pi}\phi(\mathbf{r}) + \frac{Q(\mathbf{r})}{4\pi} \tag{11a}$$

$$\phi(\mathbf{r}) = \int_{4\pi} d\boldsymbol{\Omega}\psi(\mathbf{r}, \boldsymbol{\Omega}) \tag{11b}$$

where $\mathbf{r}$ is a point in the physical domain V, $\boldsymbol{\Omega}$ is the direction of neutron flight given by a vector on the unit sphere, ψ is the angular neutron flux (cm^{-2} s^{-1} Sr^{-1} MeV^{-1}), ϕ is the scalar neutron flux (cm^{-2} s^{-1} MeV^{-1}), Σ_t is the macroscopic total cross-section (cm^{-1}), Σ_s is the macroscopic scattering cross-section (cm^{-1}) and Q is the isotropic extraneous source strength (cm^{-3} s^{-1} MeV^{-1}). This work will only consider problems in two-dimensional space $V \subset \mathbb{R}^2$, although the derivation is identical in one or three-dimensional problems.

The discrete ordinates angular discretisation then splits the angular dependence into N discrete directions $\boldsymbol{\Omega}_k$, along which Eq. (11a) is solved for the $\psi_k(\mathbf{r}) = \psi(\mathbf{r}, \boldsymbol{\Omega}_k)$. Each ordinate also has a corresponding quadrature weight w_k for performing angular integrals, which gives an approximation to the scalar flux:

$$\phi(\mathbf{r}) \approx \sum_{k=1}^{N} w_k\psi_k(\mathbf{r}). \tag{12}$$

The steady state, mono-energetic, fixed source discrete ordinates equations for neutron transport with isotropic scattering are then given by:

$$\boldsymbol{\Omega}_k \cdot \nabla\psi_k(\mathbf{r}) + \Sigma_t(\mathbf{r})\psi_k(\mathbf{r}) = \frac{\Sigma_s(\mathbf{r})}{4\pi}\phi(\mathbf{r}) + \frac{Q(\mathbf{r})}{4\pi} \quad \text{for } k = 1, \ldots, N. \tag{13}$$

As with traditional DGFEM, Eq. (13) is multiplied by a test function $v(\mathbf{r}) \in H^1(V_e)$, as both the gradient and the trace of v on the element boundary will be needed [28], and then integrated over an element V_e:

$$\int_{V_e} d\mathbf{r} v(\mathbf{r})\boldsymbol{\Omega}_k \cdot \nabla\psi_k(\mathbf{r}) + \int_{V_e} d\mathbf{r}\Sigma_t(\mathbf{r})\psi_k(\mathbf{r})v(\mathbf{r}) = \frac{1}{4\pi}\left(\int_{V_e} d\mathbf{r}\Sigma_s(\mathbf{r})\phi(\mathbf{r})v(\mathbf{r}) + \int_{V_e} d\mathbf{r}Q(\mathbf{r})v(\mathbf{r})\right). \tag{14}$$

The divergence theorem is then applied to the first term of Eq. (14), yielding:

$$\int_{V_e} d\mathbf{r} v(\mathbf{r})\boldsymbol{\Omega}_k \cdot \nabla\psi_k(\mathbf{r}) = \int_{\Gamma_e} dS v(\mathbf{r})\psi_k(\mathbf{r})\boldsymbol{\Omega}_k \cdot \mathbf{n}(\mathbf{r}) - \int_{V_e} d\mathbf{r}\psi_k(\mathbf{r})\boldsymbol{\Omega}_k \cdot \nabla v(\mathbf{r}) \tag{15}$$

where Γ_e is the boundary of element V_e, and $\mathbf{n}(\mathbf{r})$ 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 $\langle u, v\rangle := \int_{\Gamma_e} dSu(s)v(s)$, combining this notation with Eq. (14), substituting in (15) and utilising the fact that the cross-sections and external source term are constant within an element yields:

$$-\boldsymbol{\Omega}_k \cdot (\psi_k, \nabla v) + \Sigma_t(\psi_k, v) + \boldsymbol{\Omega}_k \cdot \langle\psi_k, v\mathbf{n}\rangle = \frac{1}{4\pi}[\Sigma_s(\phi, v) + Q_e(1, v)] \tag{16}$$

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

$$\boldsymbol{\Omega}_k \cdot \langle\psi_k, v\mathbf{n}\rangle = \boldsymbol{\Omega}_k \cdot \langle\psi_k, v\mathbf{n}\rangle^+ + \boldsymbol{\Omega}_k \cdot \langle\psi_k, v\mathbf{n}\rangle^-. \tag{17}$$

As in traditional DGFEM S_N [29], the outflow integral over Γ_e^+ is performed using the flux from within the element. The inflow integral over Γ_e^- is fully upwinded using the flux ψ_{adj} from the adjacent element(s). This process is more complicated with the hanging-node formulation than in the conforming mesh case outlined in [19], and will be covered in detail below. If Γ_e^- is on the physical domain boundary, ψ_{adj} is instead determined by the appropriate boundary condition. Eq. (16) can then be rewritten as:

$$-\boldsymbol{\Omega}_k \cdot (\psi_k, \nabla v) + \Sigma_t(\psi_k, v) + \boldsymbol{\Omega}_k \cdot \langle\psi_k, v\mathbf{n}\rangle^+ = \frac{1}{4\pi}[\Sigma_s(\phi, v) + Q_e(1, v)] - \boldsymbol{\Omega}_k \cdot \langle\psi_{adj}, v\mathbf{n}\rangle^-. \tag{18}$$

A solution $\psi_k \in H^1(V_e)$ is then sought such that Eq. (18) 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:

$$v(\mathbf{r}) = R_A(\mathbf{r}) \quad \text{for } A = 1, \ldots, \text{nnp} \tag{19a}$$

$$\psi_k(\mathbf{r}) = \sum_{B=1}^{\text{nnp}} \psi_k^B R_B(\mathbf{r}) \tag{19b}$$

$$\phi(\mathbf{r}) = \sum_{B=1}^{\text{nnp}} \phi^B R_B(\mathbf{r}) \quad \text{where} \tag{19c}$$

$$\phi^B = \sum_{k=1}^{N} w_i \psi_k^B \tag{19d}$$

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. The source vector, along with mass, streaming and boundary matrices are then defined to be:

$$\mathbf{Q}_{e\{A\}} = Q_e(1, R_A) \tag{20a}$$

$$\underline{\underline{M}}_{\{A,B\}} = (R_B, R_A) \tag{20b}$$

$$\underline{\underline{S_x}}_{\{A,B\}} = \left(R_B, \frac{\partial R_A}{\partial x}\right) \tag{20c}$$

$$\underline{\underline{S_y}}_{\{A,B\}} = \left(R_B, \frac{\partial R_A}{\partial y}\right) \tag{20d}$$

$$\underline{\underline{P_k}}_{\{A,B\}} = \boldsymbol{\Omega}_k \cdot \langle R_B, R_A \mathbf{n}\rangle^+. \tag{20e}$$

The volumetric integrals are performed using $p+1$ Gaussian quadrature points in each parametric direction over a parent element defined to be $[-1,1]^d$ where d is the number of spatial dimensions. For example, if element V_e is the image of the parametric element $[\xi_l, \xi_u] \times [\eta_l, \eta_u]$, with Jacobian matrix $\underline{\underline{J}}$ corresponding to this mapping, entry $\{A, B\}$ of the mass matrix is given by:

$$\underline{\underline{M}}_{\{A,B\}} = (R_B, R_A) \tag{21a}$$

$$= \int_{V_e} d\mathbf{r}\; R_A(\mathbf{r})\; R_B(\mathbf{r}) \tag{21b}$$

$$= \int_{\xi_l}^{\xi_u} d\xi \int_{\eta_l}^{\eta_u} d\eta\; \left|\underline{\underline{J}}(\xi,\eta)\right|\; R_A(\xi,\eta)\; R_B(\xi,\eta) \tag{21c}$$

$$= \frac{(\xi_u - \xi_l)(\eta_u - \eta_l)}{4} \int_{-1}^{1} d\tilde{\xi} \int_{-1}^{1} d\tilde{\eta}\; \left|\underline{\underline{J}}(\tilde{\xi},\tilde{\eta})\right|\; R_A(\tilde{\xi},\tilde{\eta})\; R_B(\tilde{\xi},\tilde{\eta}) \tag{21d}$$

$$\approx \frac{(\xi_u - \xi_l)(\eta_u - \eta_l)}{4} \sum_{i=1}^{p+1}\sum_{j=1}^{p+1} W_i W_j \left|\underline{\underline{J}}(\tilde{\xi}_i,\tilde{\eta}_j)\right|\; R_A(\tilde{\xi}_i,\tilde{\eta}_j)\; R_B(\tilde{\xi}_i,\tilde{\eta}_j) \tag{21e}$$

where the W_i are the Gaussian quadrature weights and the fractional term represents the determinant of the Jacobian of the mapping from $[-1,1]^2$ to $[\xi_l, \xi_u] \times [\eta_l, \eta_u]$. Defining solution vectors $\boldsymbol{\psi}_k$ and $\boldsymbol{\phi}$ in the obvious way from Eqs. (19b) and (19c) respectively, Eq. (18) can be written in the compact form:

$$\left[-\mu_k \underline{\underline{S_x}} - \eta_k \underline{\underline{S_y}} + \Sigma_t \underline{\underline{M}} + \underline{\underline{P_k}}\right] \boldsymbol{\psi}_k = \frac{1}{4\pi}\left(\Sigma_s \underline{\underline{M}}\boldsymbol{\phi} + \mathbf{Q}_e\right) - \boldsymbol{\psi}_{adj} \tag{22}$$

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

$$\underline{\underline{L}} := -\mu_k \underline{\underline{S_x}} - \eta_k \underline{\underline{S_y}} + \Sigma_t \underline{\underline{M}} + \underline{\underline{P_k}} \tag{23a}$$

$$\implies \underline{\underline{L}}\boldsymbol{\psi}_k = \frac{1}{4\pi}\left(\Sigma_s \underline{\underline{M}}\boldsymbol{\phi} + \mathbf{Q}_e\right) - \boldsymbol{\psi}_{adj}. \tag{23b}$$

This system of equations can then be solved for each element in an order that depends on the discrete ordinates direction $\boldsymbol{\Omega}_k$. The difference between the conforming mesh case outlined in [19] and the hanging-node formulation presented here lies in the calculation of the incoming upwinded flux $\boldsymbol{\psi}_{adj}$. In the conforming mesh case, on the element boundary the adjacent flux $\boldsymbol{\psi}_{adj}$ is represented in terms of the same basis functions as the flux $\boldsymbol{\psi}_k$ within the element to be solved for. This is no longer the case if an element face contains a hanging-node, such as in Fig. 2. In this case, projected mass matrices are needed of the form $\int_{\Gamma_e^-} dS R_B(S) R_A(S)$ where the R_A are the basis functions defined over element e that are non-zero along Γ_e^-, and the R_B are the basis functions defined over neighbouring elements that are non-zero along Γ_e^-. In DGFEM these element faces are typically straight, or approximated as such [29], and so a single projected mass matrix is needed on each of the faces Γ_2, Γ_3 and Γ_4 in Fig. 2. The NURBS case is complicated by the fact that, in general, the use of an average normal vector on an element face limits the convergence of the spatial scheme to first order [19]. The product $R_B R_A$ along with a normal vector must be stored at each quadrature point being used for integration on each element face. This represents the only fundamental difference between the

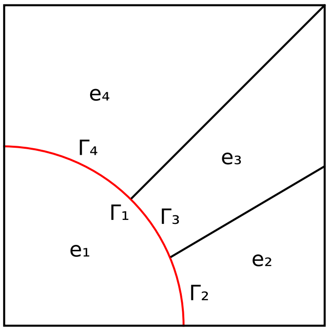


Fig. 2. A typical hanging-node element face highlighted in red. $\varGamma_i$ is the portion of the face on the boundary of element i, and $\varGamma_1 = \varGamma_2 \cup \varGamma_3 \cup \varGamma_4$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

method presented here and existing hanging-node DGFEM methods for the discrete ordinates equations [22,30], and incurs a performance penalty of ≈5%–10% in the transport solve, depending on the polynomial order of the basis functions employed.

In addition, the sign of $\boldsymbol{\Omega}_k \cdot \mathbf{n}$ can change over a curved boundary, and so integration must be performed over partial element boundaries. This is performed as in [19], where the integral over a partial element boundary, e.g. $\varGamma_e^+$ is treated as an integral over the whole boundary in which the integrand vanishes where $\boldsymbol{\Omega}_k \cdot \mathbf{n} < 0$. I.e.

$$\int_{\varGamma_e^+} f(S)\boldsymbol{\Omega}_k \cdot \mathbf{n} dS = \int_{\varGamma_e} g(S)\boldsymbol{\Omega}_k \cdot \mathbf{n} dS \quad \text{where} \tag{24a}$$

$$g(s) = \begin{cases} f(S) & \text{for} \quad \boldsymbol{\Omega}_k \cdot \mathbf{n} \geq 0 \\ 0 & \text{for} \quad \boldsymbol{\Omega}_k \cdot \mathbf{n} < 0. \end{cases} \tag{24b}$$

For example in the quadratic case depicted in Fig. 2, $p = 2$ and therefore 3 quadrature points are required on an element face as demonstrated in [19]. This means integrals over $\varGamma_2^\pm$, $\varGamma_3^\pm$ and $\varGamma_4^\pm$ must be performed using 3 quadrature points each. Similarly, integrals over $\varGamma_1^-$ must be split up into integrals over $\varGamma_2$, $\varGamma_3$ and $\varGamma_4$ and summed, leading to a total of 9 quadrature points being used for this integration. It needs to be investigated when performing integrals over $\varGamma_1^+$ whether we need to use these same 9 quadrature points, or whether 3 are enough.

The method of integration over partial element faces given by Eq. (24) dictates that the method with 9 quadrature points must be used. If not, in cases where the sign of $\boldsymbol{\Omega}_k \cdot \mathbf{n}$ changes over an element face, the outgoing integral over $\varGamma_1^+$ will make one approximation to this partial boundary integration, while the incoming integrals over $\varGamma_2^-$, $\varGamma_3^-$ and $\varGamma_4^-$ will make a different approximation. The number of neutrons leaving e_1 across $\varGamma_1$ will then not in general be equal to the sum of the neutrons entering e_2, e_3 and e_4 over $\varGamma_2$, $\varGamma_3$ and $\varGamma_4$ respectively. The scheme would then not be conservative, and so integration must always be performed using the same quadrature points on any given element face. This simplifies the logic of the scheme, in that integration over any inter-element boundary must always be performed over the finest subdivision of that boundary by the elements on either side of it.

4. Sweep ordering

As with DGFEM, an order must be chosen in which to solve the elements for each discrete ordinates direction to update the solution at each iteration. This is done using the directed graph (digraph) associated with each direction. For each direction, each element in the mesh has a vertex in the digraph. There is an edge from vertex i to vertex j if neutrons can move from element i to element j in the mesh for this direction. If we define $\varGamma_{i,j}$ to be the boundary between element i and element j and $\mathbf{n}$ to be the outward pointing unit normal on $\varGamma_{i,j}$ with respect to element i, then there is an edge from vertex i to vertex j if $\boldsymbol{\Omega}_k \cdot \mathbf{n} > 0$ at any point on $\varGamma_{i,j}$.

In traditional DGFEM, element faces are usually either straight/planar, or approximated as such [29]. In this case, meshes of two-dimensional geometries produce acyclic digraphs [29], and a perfect order can be found in which to sweep the elements for each discrete ordinate direction. This is not the case when curved element faces are explicitly taken into account [19].

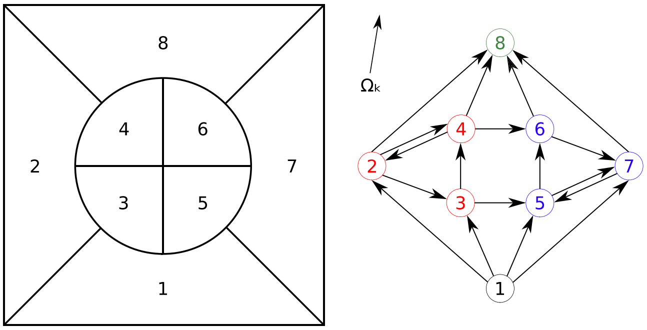


Fig. 3. A typical hanging-node mesh along with its associated digraph for the direction $\boldsymbol{\Omega}_k$. Vertices in the digraph are colour coded by which SCC they are in. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

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 can cause problems in sweep ordering, as it becomes unclear which element should be solved for first for a given ordinate if neutrons can move from element i to element j and vice versa. This can also reduce the efficiency of the scheme, as the solution for a direction cannot be obtained by solving for each element once at each iteration.

Strongly connected components (SCCs) are defined to be a set of vertices $\{v_1, v_2, \ldots, v_n\}$ such that there is a path from v_i to v_j and vice versa for all $i, j \in \{1, 2, \ldots, n\}$, and are directly correlated to the existence of cycles in the mesh. 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 [31].

In the conforming mesh case, it was found in [19] that solving each element within an SCC once is more efficient than iteratively converging the solution within an SCC before moving on to the next one. This approach will be followed here. Therefore a sweep ordering of the elements within the SCC must be found. The method employed for this in [19] involves assigning a weight to each edge in the SCC such that the weight of the edge from vertex i to vertex j $\alpha_{i,j}$ is given by:

$$\alpha_{i,j} = \int_{\Gamma_{i,j}^{+}} \boldsymbol{\Omega}_k \cdot \mathbf{n} dS \tag{25}$$

where $\Gamma_{i,j}^{+}$ is defined to be the portion of $\Gamma_{i,j}$ where $\boldsymbol{\Omega}_k \cdot \mathbf{n}$ is positive. This is then a measure of how "outgoing" the direction $\boldsymbol{\Omega}$ is on this boundary. Where there was an edge both from vertex i to vertex j and vice versa the edge with the lower weight of the two was deleted. In the absence of hanging-nodes this method worked for all examples considered.

However, when there are hanging-nodes present, this method is no longer sufficient, as demonstrated in Fig. 3. In the SCC consisting of vertices 2, 3 and 4, following this method would delete the edge from vertex 2 to vertex 4. The resulting graph would still contain a cycle $2 \to 3 \to 4 \to 2$.

An alternative approach used here is to delete the weakest edge in the *entire* SCC. This is in contrast with the previous method where edges would only be deleted if there was an edge from vertex i to vertex j and vice versa. If the resulting graph is acyclic, an ordering can be found. If not, the next weakest edge is deleted, and so on until the graph is acyclic and an ordering can be found. Note that unlike the previous method, this is guaranteed to provide an ordering, as even in a pathological example eventually there will be no edges left in the SCC and so finding an ordering is trivial.

5. Hanging-node adaptive mesh refinement algorithm

While it is possible to specify hanging-node meshes manually by specifying the knots to insert in each parametric direction in each patch, the number of possible permutations grows rapidly with the number of patches. In addition, this requires *a priori* knowledge of which parts of the geometry will need a finer mesh and which parts can be left with a coarse mesh, which is not always available. In order to take advantage of the discretisation, an adaptive scheme is used here to choose how to refine the patches specified in the geometry.

Here, the coarsest mesh was chosen to be the geometry in an AMR scheme, so that each patch in the geometry becomes an element in the mesh. This geometry is created to be conforming, in the sense that each face of each element that is not on a domain boundary exactly matches the face of another element, i.e. there are no hanging-nodes in the coarsest mesh. This condition could be relaxed, at the expense of complicating the procedure of identifying which elements are adjacent to each other in the coarsest mesh, as the control points on the faces of adjacent elements would no longer coincide.

The solution to the discrete ordinates equations is then calculated on this mesh via the procedure outlined in Section 3. Each element e in the mesh is then assigned an error indicator β^e, designed to estimate the error in the numerical solution in this element. An element is then flagged for refinement if it meets the following criteria [22]:

$$\beta^e \geq \alpha \max_e(\beta^e) \tag{26}$$

with the parameter $0 \leq \alpha \leq 1$. In this study α was chosen to be 0.3 as in [22]. Elements flagged for refinement are bisected in each parametric direction, creating four new elements. This procedure is then repeated until a given exit criteria is reached, such as the number of AMR iterations or the total number of elements in the mesh.

As the initial mesh is conforming and elements are only ever refined by halving them in each parametric direction, this ensures that element faces are always either fully contained by an adjacent face or contain an integer number of element faces. This is deliberate to simplify the integration over element faces for upwinding.

Three different error indicators will be used here to calculate β^e.

Scalar Jump Error Indicator

For this error indicator a variation on the jump-based indicator presented in [30] is used, which measures the jump in the scalar flux around an element boundary. In [30] the square of the scalar flux was used, whereas here the absolute value is used, such that:

$$\beta^e = \int_{\Gamma_e} dS|\phi^e - \phi^{adj}| \tag{27}$$

where Γ_e is the boundary of the element, ϕ^e is the scalar flux within the element and ϕ^{adj} is the scalar flux in adjacent elements.

Angular Jump Error Indicator

This is similar to the jump-based error indicator, only the jump in the angular flux is used, such that:

$$\beta^e = \sum_{k=1}^{N} w_k \int_{\Gamma_e} dS|\psi_k^e - \psi_k^{adj}||\boldsymbol{\Omega}_k \cdot \mathbf{n}|. \tag{28}$$

Residual Error Indicator

This more complex error indicator is based on that described in Duo et al. [32]. Defining the residual in an element e and direction k to be:

$$r_k^e = \boldsymbol{\Omega}_k \cdot \nabla\psi_k + \Sigma_t\psi_k - \frac{\Sigma_s}{4\pi}\phi - \frac{Q}{4\pi} \quad \text{for } k = 1, \ldots, N. \tag{29}$$

The volumetric residual term for element e is then given by:

$$R_v^e = \left(\int_{V_e} d\mathbf{r} \sum_{k=1}^{N} w_k (r_k^e)^2\right)^{\frac{1}{2}}. \tag{30}$$

The jump in the incoming current is also taken into account, through the term:

$$R_c^e = \left(\sum_{k=1}^{N} w_k \int_{\Gamma_e^-} dS \left(\psi_k^e - \psi_k^{adj}\right)^2 |\boldsymbol{\Omega}_k \cdot \mathbf{n}|\right)^{\frac{1}{2}}. \tag{31}$$

The error indicator β_e is then given by:

$$\beta^e = h^e R_v^e + \sqrt{2h^e} R_c^e \tag{32}$$

where h^e is a measure of the size of the element e, taken here to be the square root of the area of the element.

6. Numerical results

6.1. Method of manufactured solutions

In order to check that the hanging-node formulation attains the correct rate of convergence, the Method of Manufactured Solutions was used. For a full description of the method, see [33]. Here, a modified version of the MMS solution presented in [34] is used to ascertain the order of convergence of the hanging-node formulation. The manufactured solution chosen is for the discrete ordinates equations rather than the full transport equation, and can therefore be used to test the convergence properties of only the spatial discretisation.

The spatial domain is given by the unit centimetre square [0, 1] × [0, 1]. The manufactured solution is then given by:

$$\psi_k(x, y) = x^2 y^2 (1 - x^2)(1 - y^2)(1 + \mu_k^2 + \eta_k^2) \quad \text{for } k = 1, \ldots, N \tag{33a}$$

$$\phi_0(x, y) = \sum_{k=1}^{N} w_k \psi_k \tag{33b}$$

where N is the number of directions in the angular quadrature set used. All MMS calculations were performed with an S_4 level symmetric angular quadrature set, and the scalar flux solution is presented in Fig. 4.

The solution is infinitely differentiable in space, allowing a spatial discretisation to attain its full rate of convergence. The manufactured solution is also equal to zero on the domain boundaries, allowing a vacuum boundary condition to be prescribed. The anisotropic source term in each direction can then be calculated:

$$Q_k(x, y) = 4\pi(\boldsymbol{\Omega}_\mathbf{k} \cdot \nabla \psi_k(x, y) + \Sigma_t \psi_k(x, y)) - \Sigma_s \phi_0(x, y) \quad \text{for } k = 1, \ldots, N. \tag{34}$$

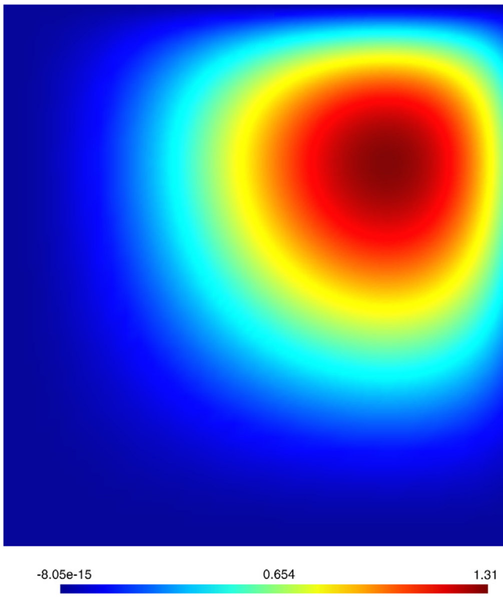


Fig. 4. The scalar flux ϕ_0 for the MMS problem.

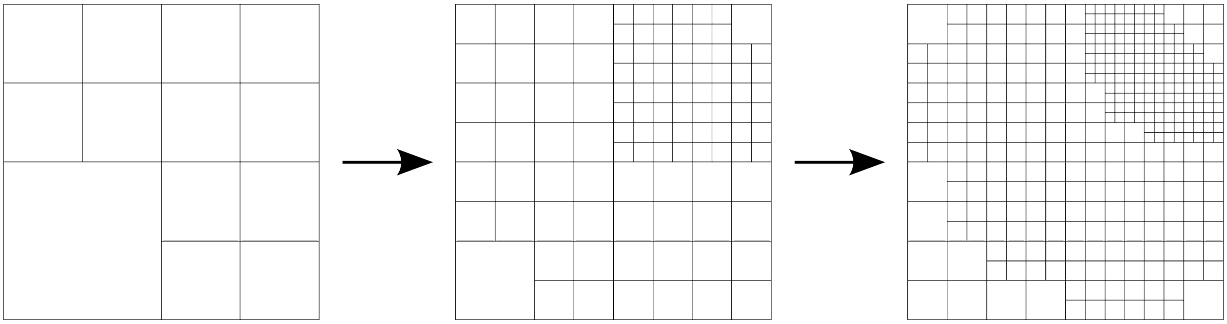


Fig. 5. The meshes from AMR steps 3, 6 and 8 for the MMS problem using orthogonal grids.

$\Sigma_t = 1\ \text{cm}^{-1}$ and $\Sigma_s = 0.5\ \text{cm}^{-1}$ as in [34]. Once the numerical scalar flux solution ϕ_{num} has been calculated, the L_2 error ϵ_2 is then given by:

$$\epsilon_2 = \left(\int_0^1 \int_0^1 (\phi_{num} - \phi_0)^2 \, dxdy \right)^{\frac{1}{2}}. \tag{35}$$

Expression (35) is then evaluated using 16^2 Gaussian quadrature points in each element. ϵ_2 will then be exact (up to machine precision) for numerical solutions with polynomial order $p \leq 15$ on B-Spline meshes. While this is not the case for NURBS meshes, $p \leq 4$ in the following examples and so the error introduced by this integration is expected to be negligible [35].

In order to generate meshes with hanging-nodes to test the convergence rate of the scheme, an adaptive procedure was used. In this rare case when the exact discrete ordinates solution is known, the L_2 error in the scalar flux in each element can be calculated. The error indicator β^e in element e is then given by:

$$\beta^e = \left(\int \int_e (\phi_{num} - \phi_0)^2 \, dxdy \right)^{\frac{1}{2}}. \tag{36}$$

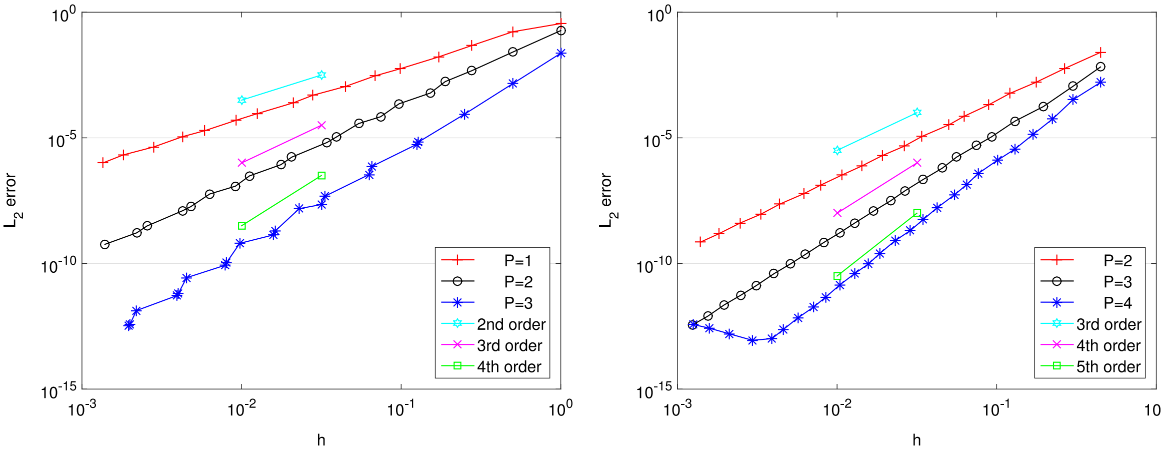
The first mesh types generated were orthogonal grids, in which case the scheme is equivalent to DGFEM. This is achieved by beginning with the square geometry as a single patch. When the AMR iteration procedure is applied to this geometry, the resulting element boundaries are always aligned with the coordinate axes. This allows both the volumetric and boundary integrals to be performed exactly using Gaussian quadrature. Example meshes generated using this procedure are given in Fig. 5.

The results of applying this procedure with first, second and third order basis functions are presented in Fig. 6(a). Fourth order basis functions cannot be used to test convergence rates on orthogonal grids as the manufactured solution itself is fourth order. Fourth order basis functions are therefore capable of exactly representing the solution, and so give an error only due to machine precision, which has been verified to be the case. Fig. 6(a) shows that the hanging-node formulation attains approximately $p + 1$ order convergence with order p basis functions. The convergence rates are not perfectly straight lines as they are for equivalent formulations without hanging-nodes [19]. However this is to be expected, as the element size is no longer a constant within a given mesh, so h is not a perfect metric of element size.

A more stringent test of the convergence rate attained by the hanging-node formulation was performed by applying the above methodology but using an initial geometry represented by a pincell style mesh shown in Fig. 7. The radius of the circle was given by $\frac{1}{\sqrt{5\pi}}$ so that the area of each of the 5 patches in the original geometry is identical.

This is a much tougher test of the spatial discretisation. The basis functions are now rational, so that both the volumetric and boundary integrals performed using Gaussian quadrature are not exact. The element boundaries can now also be curved, so that integration must be performed along partial element boundaries when discrete ordinate directions are tangent to the boundary.

The convergence rates for second, third and fourth order NURBS are shown in Fig. 6(b). Note that fourth order NURBS on these meshes cannot exactly represent the manufactured solution and so convergence tests can be



(a) Orthogonal meshes.

(b) Pincell meshes.

Fig. 6. Convergence plots for the MMS problem on orthogonal (a) and pincell style (b) meshes, along with ideal 2nd–5th order convergence examples. h is a measure of the average element size, given by $\frac{1}{\sqrt{E}}$ where E is the number of elements in the mesh. The $p = 4$ scheme convergence on the pincell style meshes is limited to $\approx 10^{-13}$ due to a combination of machine precision and the condition number of the system.

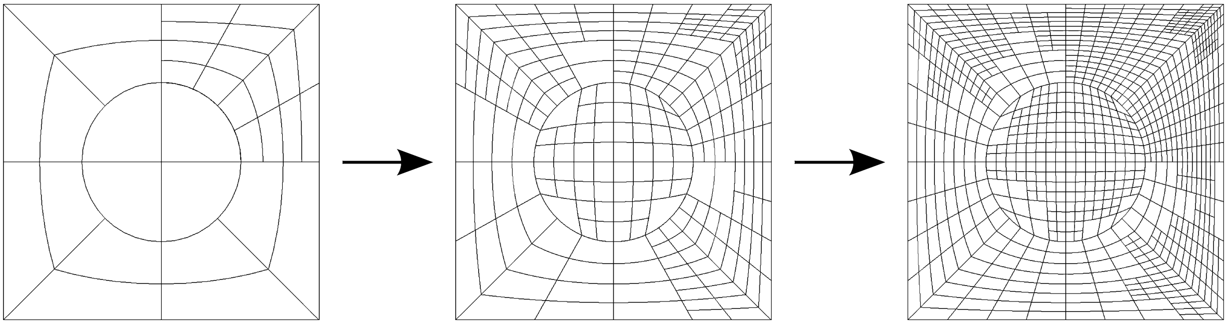


Fig. 7. The meshes from AMR steps 3, 6 and 8 for the MMS problem using pincell-style meshes.

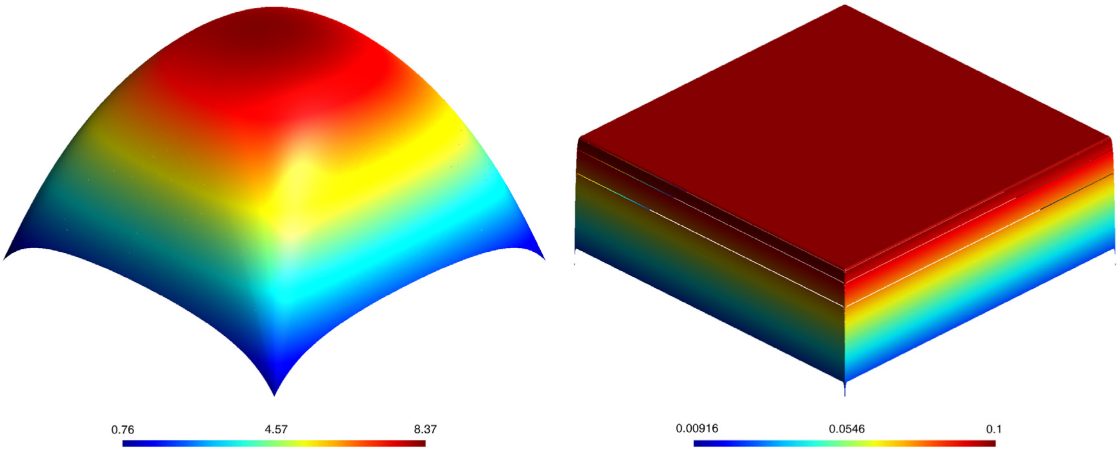
performed using them. Fig. 6(b) shows that with rational basis functions and curved element boundaries, the hanging-node formulation still attains order $p+1$ convergence with order p basis functions. The L_2 error is limited to $\approx 10^{-13}$ due to a combination of machine precision and the condition number of the system.

6.2. *Homogeneous square in a vacuum*

While the solution in Section 6.1 is useful for testing theoretical convergence rates, typical transport solutions are not infinitely differentiable. With piecewise constant cross sections, the solution belongs in $H^{\frac{3}{2}}$ at best [36]. This restricts the attainable rate of convergence of even high order spatial discretisations to $\frac{3}{2}$. A problem that demonstrates this reduced convergence order is that of a homogeneous square with vacuum boundary conditions prescribed [22,30].

The problem consists of a 10 cm × 10 cm square with a uniform isotropic unit source, a scattering ratio $c = \frac{\Sigma_s}{\Sigma_t} = 0.9$ and vacuum boundary conditions. This was solved with an S_4 level symmetric angular quadrature set for $\Sigma_t = 1\ \text{cm}^{-1}$ and $\Sigma_t = 100\ \text{cm}^{-1}$. The scalar flux solutions to this problem are shown in Fig. 8(a) for $\Sigma_t = 1\ \text{cm}^{-1}$ and in Fig. 8(b) for $\Sigma_t = 100\ \text{cm}^{-1}$. In both cases, uniform refinement of the spatial domain was compared to AMR driven by each of the 3 error indicators outlined in Section 5.

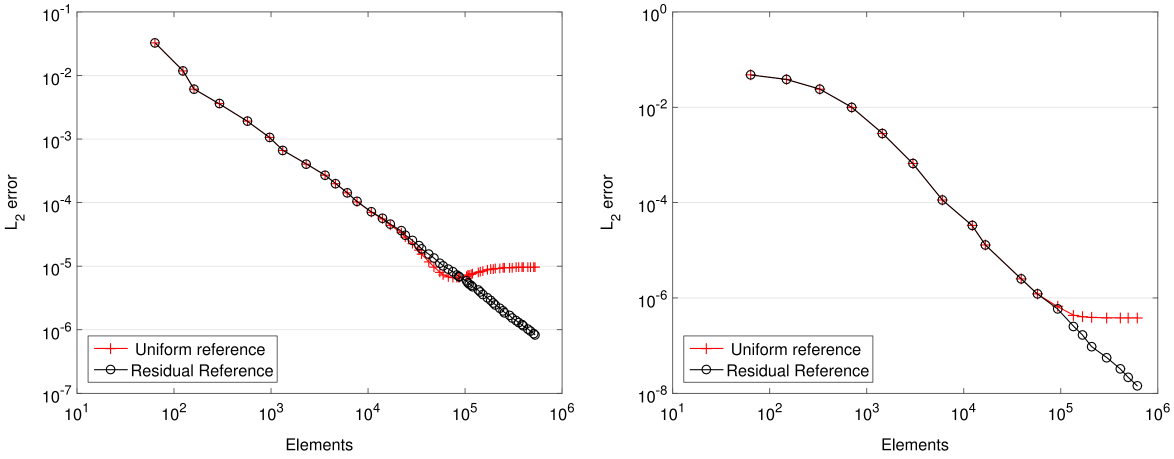
For both values of Σ_t, L_2 errors were calculated using a reference solution. Two different reference solutions were generated for each Σ_t value. In both cases one of these used a uniform refinement of the domain with $4^{11} = 4{,}194{,}304$ quartic ($p = 4$) elements. The other was generated using the Residual error indicator driven AMR technique with $p = 4$, with 1,040,173 elements for $\Sigma_t = 1\ \text{cm}^{-1}$ and with 524,656 elements in the $\Sigma_t = 100\ \text{cm}^{-1}$ case.



(a) $\Sigma_t = 1\ \text{cm}^{-1}$.

(b) $\Sigma_t = 100\ \text{cm}^{-1}$.

Fig. 8. The scalar flux solutions to the homogeneous square problem with $\Sigma_t = 1\ \text{cm}^{-1}$ (a) and $\Sigma_t = 100\ \text{cm}^{-1}$ (b).



(a) $\Sigma_t = 1\ \text{cm}^{-1}$.

(b) $\Sigma_t = 100\ \text{cm}^{-1}$.

Fig. 9. The convergence of the L_2 error calculated using both uniform and adaptive quartic ($p = 4$) reference solutions for the Residual error indicator with $p = 3$ for the $\Sigma_t = 1\ \text{cm}^{-1}$ case (a) and the $\Sigma_t = 100\ \text{cm}^{-1}$ case (b) of the homogeneous square problem. The errors calculated using the uniform references plateau around 10^{-5} and $10^{-6.7}$ for the $\Sigma_t = 1\ \text{cm}^{-1}$ and $\Sigma_t = 100\ \text{cm}^{-1}$ cases respectively. In both cases this corresponds to the L_2 difference between the uniform and adaptive references.

Fig. 9(a) shows that for errors larger than $\approx 10^{-4.5}$ the L_2 error computed using either of the references give indistinguishable results for the $\Sigma_t = 1\ \text{cm}^{-1}$ case. The errors computed using the uniform reference then diverge from those computed using the adaptive reference and plateau at 10^{-5}. This is due to the error in the uniform reference. While it has more elements than the adaptive reference, they are evenly distributed, and so many are used to over-refine areas of the geometry of little significance. Fig. 10 shows that the adaptive scheme has concentrated elements along the ordinate directions emanating from the corners of the geometry. This is to be expected as these are the lines across which the true S_N solution can be discontinuous [22,30].

Similarly, Fig. 9(b) shows indistinguishable errors computed using both references above $\approx 10^{-6}$ for the $\Sigma_t = 100\ \text{cm}^{-1}$ case, with the error compared to the uniform reference plateauing around $10^{-6.7}$. Fig. 11 demonstrates the adaptive scheme concentrating elements along the boundary of the domain as there is an extremely sharp gradient in the solution here as demonstrated in Fig. 8(b). The magnified portion of Fig. 11 shows that near the boundary the elements are also being concentrated along the ordinate directions emanating from the domain corners. For these reasons, the remaining results use the L_2 error computed using the adaptive reference solutions.

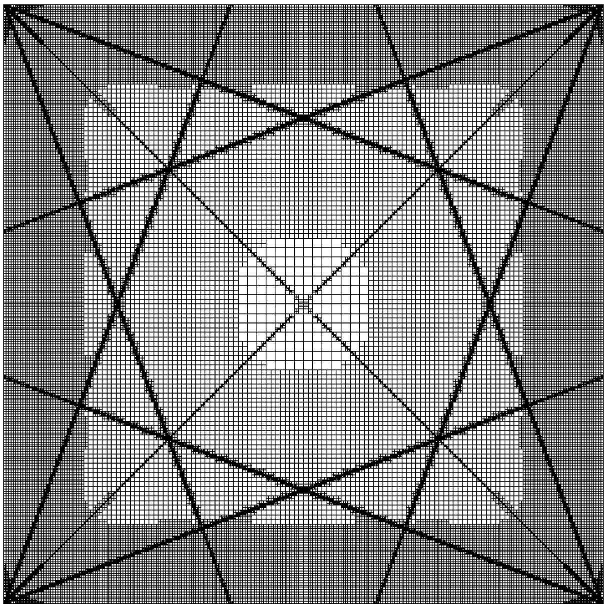


Fig. 10. The reference mesh calculated using the Residual error indicator with $p = 4$ for the homogeneous square problem with $\Sigma_t = 1\ \mathrm{cm}^{-1}$. The elements have been concentrated along the ordinate directions emanating from the corners of the geometry.

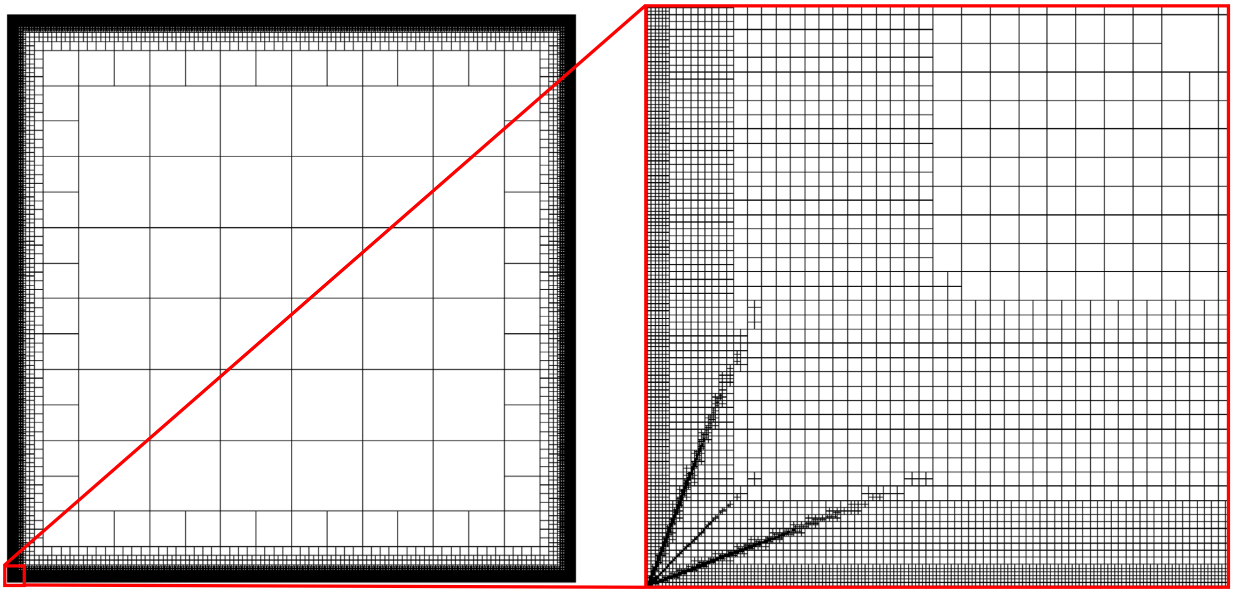
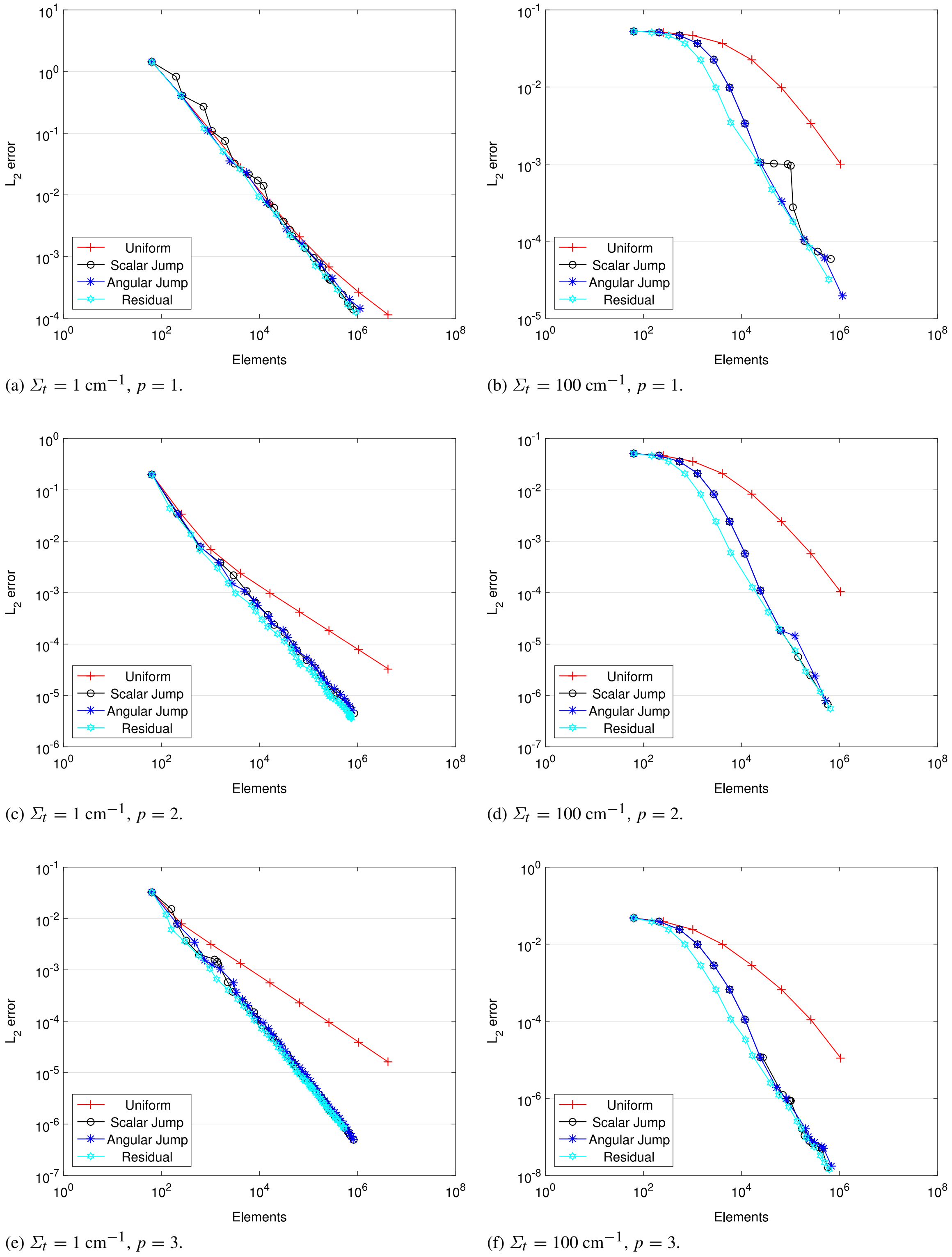


Fig. 11. The reference mesh calculated using the Residual error indicator with $p = 4$ for the homogeneous square problem with $\Sigma_t = 100\ \mathrm{cm}^{-1}$.

Figs. 12(a), 12(c) and 12(e) show the convergence in the $\Sigma_t = 1\ \mathrm{cm}^{-1}$ case of uniform refinement and all 3 AMR schemes for $p = 1$, 2 and 3 respectively. For all polynomial orders all 3 AMR schemes behave very similarly, and in all cases have a higher rate of convergence on a per element basis than the uniform refinement method. In the case of the most refined cubic cases, this results in efficiency savings of two orders of magnitude, either obtaining the same level of accuracy using 100x fewer elements, or using the same number of elements and reducing the L_2 error by a factor of 100.

Figs. 12(b), 12(d) and 12(f) show the convergence in the $\Sigma_t = 100\ \mathrm{cm}^{-1}$ case of uniform refinement and all 3 AMR schemes for $p = 1$, 2 and 3 respectively. Again a higher rate of convergence on a per element basis is observed using



(a) $\Sigma_t = 1\ \text{cm}^{-1}$, $p = 1$.

(b) $\Sigma_t = 100\ \text{cm}^{-1}$, $p = 1$.

(c) $\Sigma_t = 1\ \text{cm}^{-1}$, $p = 2$.

(d) $\Sigma_t = 100\ \text{cm}^{-1}$, $p = 2$.

(e) $\Sigma_t = 1\ \text{cm}^{-1}$, $p = 3$.

(f) $\Sigma_t = 100\ \text{cm}^{-1}$, $p = 3$.

Fig. 12. The L_2 error convergence of uniform refinement along with all 3 AMR schemes for the $\Sigma_t = 1\ \text{cm}^{-1}$ case (left) and $\Sigma_t = 100\ \text{cm}^{-1}$ case (right) of the homogeneous square problem with $p = 1$ (top), 2 (middle) and 3 (bottom). In all cases the AMR schemes have a higher rate of convergence on a per element basis than uniform refinement. This difference in rate markedly increases in the $\Sigma_t = 1\ \text{cm}^{-1}$ case as the polynomial order of the basis is increased, which is not observed in the $\Sigma_t = 100\ \text{cm}^{-1}$ case.

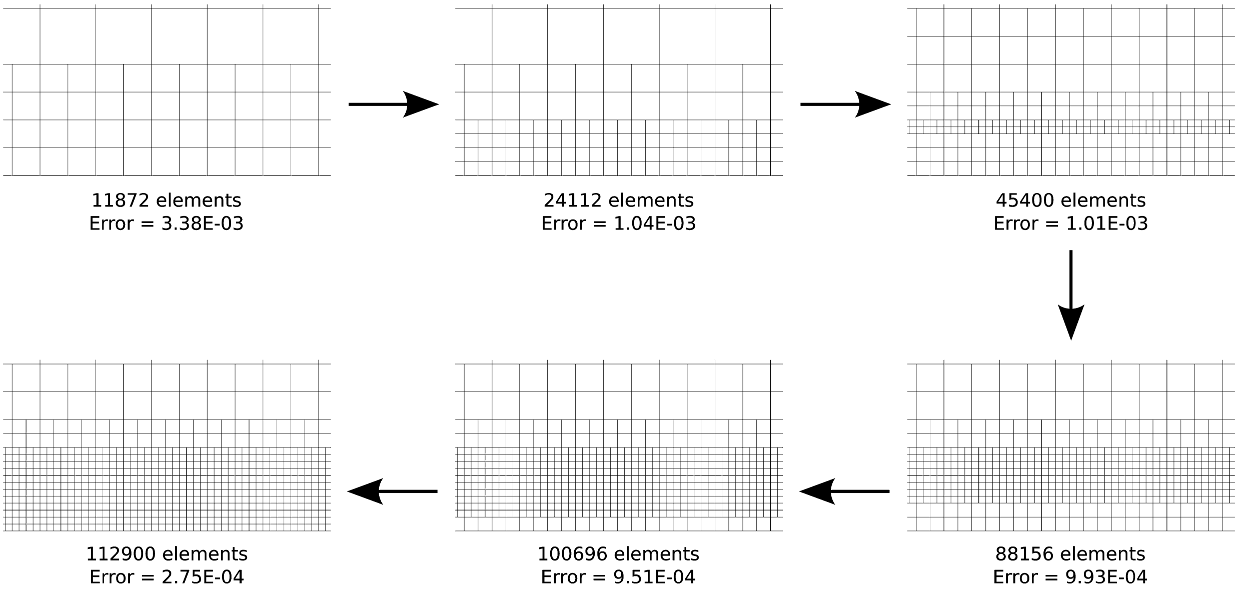


Fig. 13. A section of the mesh in the centre of the bottom ($y = 0$) domain boundary at AMR steps 7–12 for the Scalar Jump error indicator with $p = 1$ for the $\Sigma_t = 100\ \text{cm}^{-1}$ case of the homogeneous square problem. Significant drops in the L_2 error are observed from step 7 to step 8 and from step 11 to step 12, which coincide with elements on the boundary being refined. No significant reduction in the L_2 error is observed between steps 8 and 11 as the elements on the boundary are not being refined.

all 3 of the AMR schemes. All 3 AMR schemes behave qualitatively similarly, although are not in as close agreement as in the $\Sigma_t = 1\ \text{cm}^{-1}$ case. In particular, the Scalar Jump error indicator in the $p = 1$ case shows a plateau in the error in Fig. 12(b) around 10^{-3} before resuming its convergence. Fig. 13 shows the meshes generated by this scheme at the AMR steps involved. The elements on the boundary of the domain are not being refined in the AMR steps when the L_2 error is plateaued. The error then reduces dramatically once these elements are refined at AMR step 12. One possible cause for this behaviour could be that there is no jump in the scalar flux along the face of the elements that form part of the domain boundary, which would artificially suppress the error indicators in boundary elements using this method.

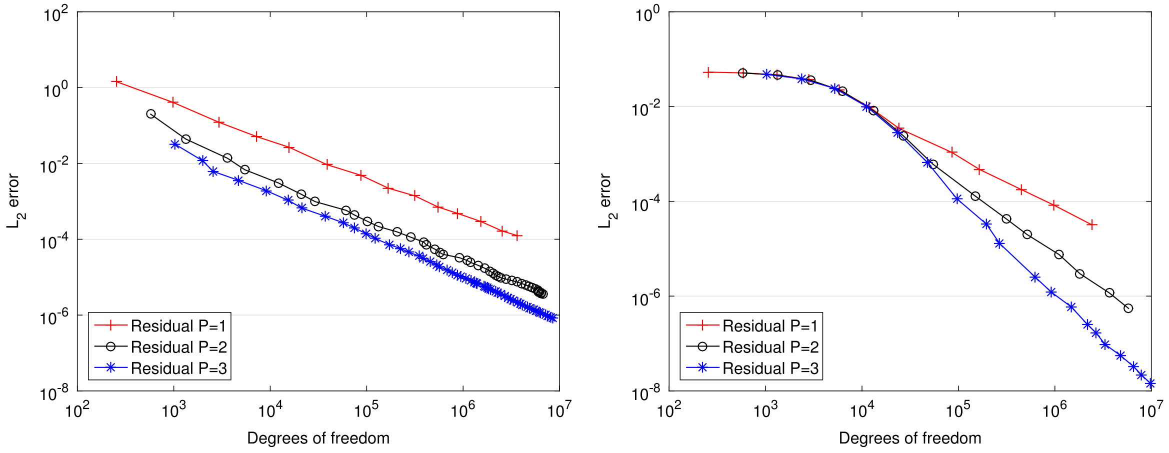
Figs. 14(a) and 14(b) show comparisons between spatial discretisations using the same error indicator but different polynomial orders for $\Sigma_t = 1\ \text{cm}^{-1}$ and $\Sigma_t = 100\ \text{cm}^{-1}$ respectively. As elements of different polynomial orders have different numbers of basis functions, total degrees of freedom are used as the independent variable. For $\Sigma_t = 1\ \text{cm}^{-1}$ there is a significant increase in accuracy on a per degree of freedom basis when moving from linear to quadratic basis functions, and a slight improvement again in using cubics. In the $\Sigma_t = 100\ \text{cm}^{-1}$ case all 3 orders behave very similarly until the error reaches $\approx 10^{-2.5}$. At that point, the rates of convergence differ significantly, with higher convergence rates observed the greater the polynomial order of the spatial discretisation.

6.3. Shielding problem

This problem is an adaptation of Example 2 used in [22], and is representative of a shielding problem [30]. A central source region is surrounded by a highly scattering material, which in turn is surrounded by an absorber. In order to test the NURBS and curved boundaries in a more taxing regime than that in Section 6.1, the Cartesian geometry specified in [22] has been modified so that the central source region is a quarter circle, the surrounding scattering media is a quarter annulus, and the absorber fills the rest of the geometry as shown in Fig. 15. The material properties in each region are given in Table 2.

A level symmetric S_4 angular quadrature set was used to solve the problem. A reference solution was generated using the Residual error indicator with 202,532 quartic elements, the scalar flux of which is presented in Fig. 15.

All three AMR schemes and uniform refinement were compared using quadratic elements. Fig. 16 shows the convergence in the L_2 error of all three AMR schemes and uniform refinement with quadratic elements compared to the reference solution. All of the AMR schemes behave very similarly, and converge significantly faster than the



(a) $\Sigma_t = 1\ \mathrm{cm}^{-1}$.

(b) $\Sigma_t = 100\ \mathrm{cm}^{-1}$.

Fig. 14. The L_2 error convergence of the Residual error indicator for the $\Sigma_t = 1\ \mathrm{cm}^{-1}$ (a) and $\Sigma_t = 100\ \mathrm{cm}^{-1}$ (b) cases of the homogeneous square problem for linear, quadratic and cubic basis functions. Degrees of freedom is now on the x-axis as different polynomial orders have a different number of basis functions per element. For the $\Sigma_t = 1\ \mathrm{cm}^{-1}$ case the rate of convergence is very similar for all polynomial orders, with an increase in accuracy per degree of freedom observed as the polynomial order is increased. In the $\Sigma_t = 100\ \mathrm{cm}^{-1}$ case the rates of convergence on a degree of freedom basis differ significantly once the error is below $\approx 10^{-2.5}$, improving as the polynomial order is increased.

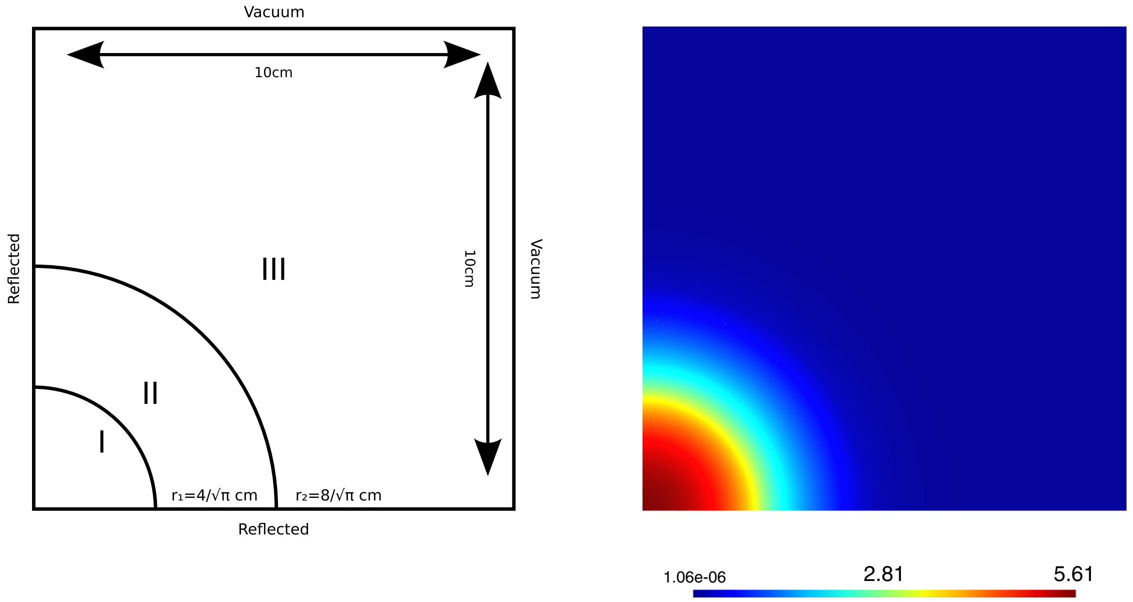


Fig. 15. The geometry (left) and scalar flux (right) for the shielding problem. The domain is a square of side length 10 cm with a reflective boundary condition on the left and bottom sides and a vacuum boundary condition on the right and top sides. The region boundary radii were chosen to preserve the volume of each material found in the original problem specified in [22].

Table 2
The cross section and source data for the shielding problem.

Material	Σ_t	Σ_s	Q
I	1.00	0.90	1.00
II	1.50	1.44	0.00
III	1.00	0.30	0.00

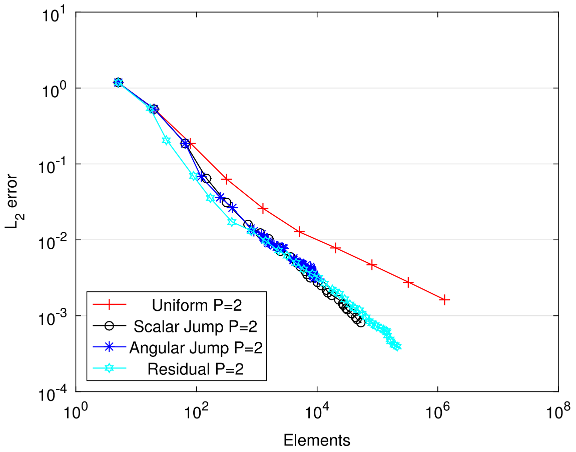


Fig. 16. The convergence of the L_2 error for all three AMR schemes and uniform refinement with quadratic elements for the shielding problem. All 3 AMR schemes behave very similarly, and converge at a higher rate than uniform refinement.

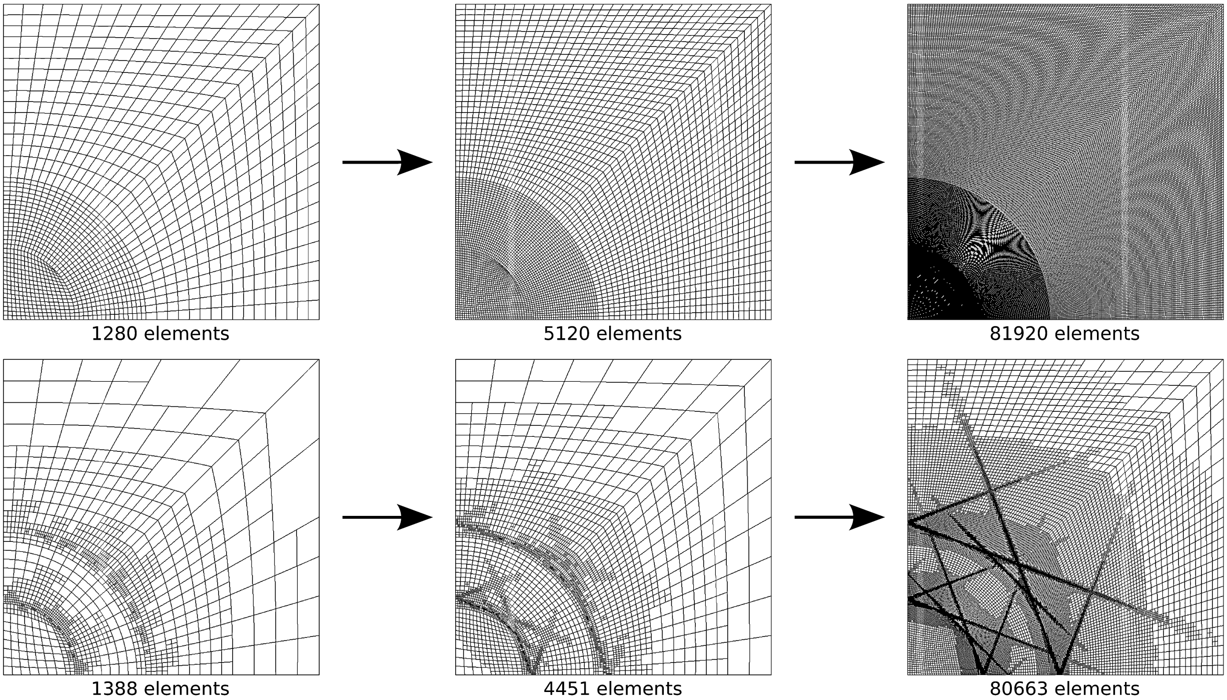


Fig. 17. Uniform (top) and AMR (bottom) meshes for the shielding problem. The AMR schemes concentrate the elements along the characteristics across which the solution can exhibit discontinuities, as well as around the sharp gradients in the flux. The uniform refinement with similar numbers of elements under-resolves these features while over-refining regions of the geometry of little importance to the overall solution, such as the top right corner.

uniform refinement method on a per element basis. Fig. 17 shows both uniform and Residual error indicator meshes with similar numbers of elements (all 3 AMR schemes give qualitatively similar meshes). The AMR schemes refine the mesh along the characteristics across which the S_4 solution can be discontinuous, as well as around the sharp gradient in the flux near the source region. The uniform refinement under-resolves these areas, while over-refining areas of the domain where the solution is flat, such as near the vacuum boundaries.

6.4. Supercell with burnable absorber

This test problem consists of a 3 × 3 array of cells, adapted from [37] by only considering isotropic scattering. The outer 8 cells are identical fuel pins, while the central cell contains a burnable absorber, all surrounded by a moderator. The geometry is given in Fig. 18 and the material properties are given in Table 3. The problem was solved with an S_{32} triangular-Chebyshev angular quadrature set to attempt to mitigate ray effects as much as possible. The strongly absorbing central pin causes a severe depression in the scalar flux, as demonstrated in Fig. 18. A reference solution was generated using the Residual error indicator with 684,064 quartic elements. Uniform refinement here refers to selecting the number of knot insertions so as to keep the average element size in each patch as similar as possible. The absorption rate in the fuel pins in the middle of one side of the geometry was used as the detector response to compare this uniform refinement to each of the three AMR schemes with quadratic elements in all cases.

Fig. 19 shows the convergence in the detector response for quadratic elements using uniform refinement and all three AMR schemes. The AMR schemes are more accurate than uniform refinement for this problem by approximately one order of magnitude when using the same number of elements. The Residual and Angular Jump error indicators give reasonably smooth convergence plots, while the Scalar Jump error indicator has more erratic behaviour. Fig. 20 shows the Residual scheme concentrating the mesh refinement around the sharp gradient induced by the strongly absorbing central pin (all 3 AMR schemes give qualitatively similar meshes). Uniform meshes with a similar number of elements under-resolve this region while over-refining areas of the geometry which are less important, such close to the domain boundary.

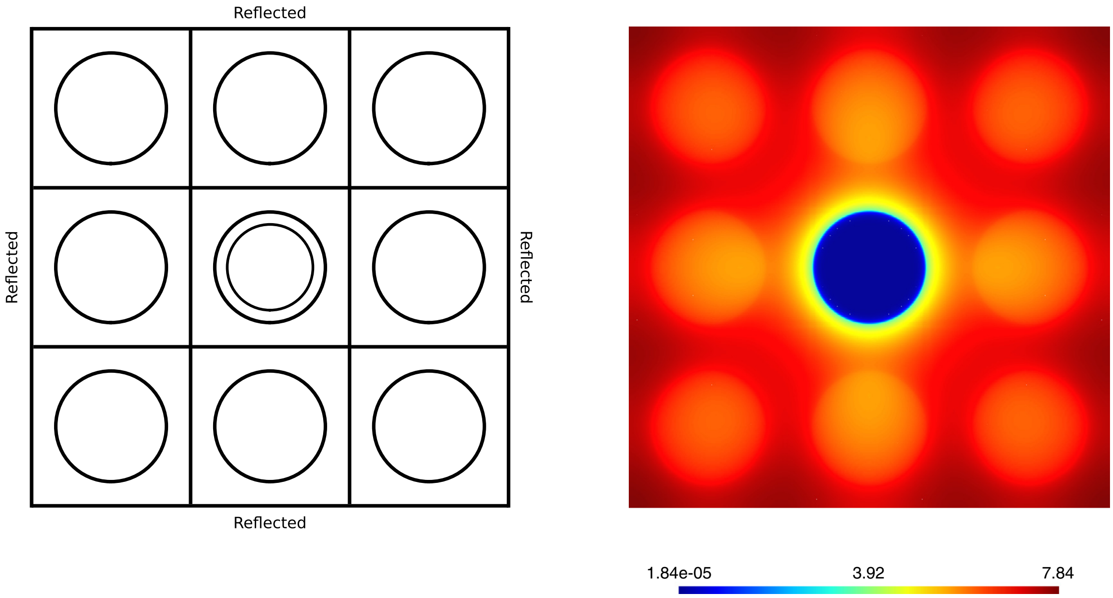


Fig. 18. The geometry (left) and scalar flux (right) for the supercell problem. Each cell has pitch 2 cm and each of the 8 fuel pins has radius 0.71 cm. The central burnable absorber consists of a circle of radius 0.55 cm and an annulus of outer radius 0.71 cm. Reflective boundary conditions are applied on every boundary of the domain. The strongly absorbing burnable absorber in the centre causes a sharp gradient in the flux in its vicinity.

Table 3
The cross section and source data for the supercell problem.

Material	Σ_t	Σ_s	Q
Fuel	0.5145	0.3432	0.0000
Moderator	0.7557	0.7504	1.0000
Burnable absorber circle	14.270	0.3400	0.0000
Burnable absorber annulus	21.630	0.3414	0.0000

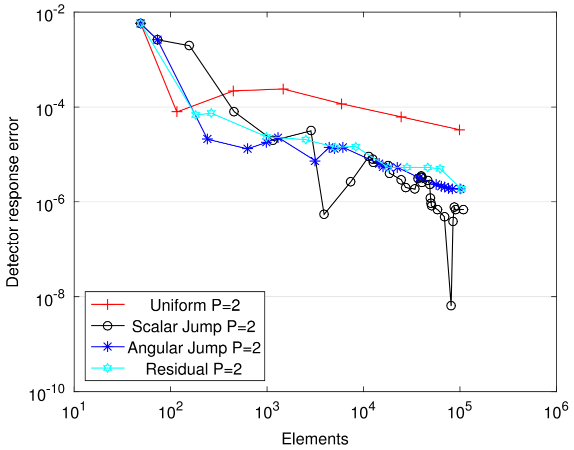


Fig. 19. The convergence of the detector response for all 3 AMR schemes and a uniformly refined method used quadratic elements for the supercell problem. The AMR schemes are approximately an order of magnitude more accurate than uniform refinement for a given number of elements.

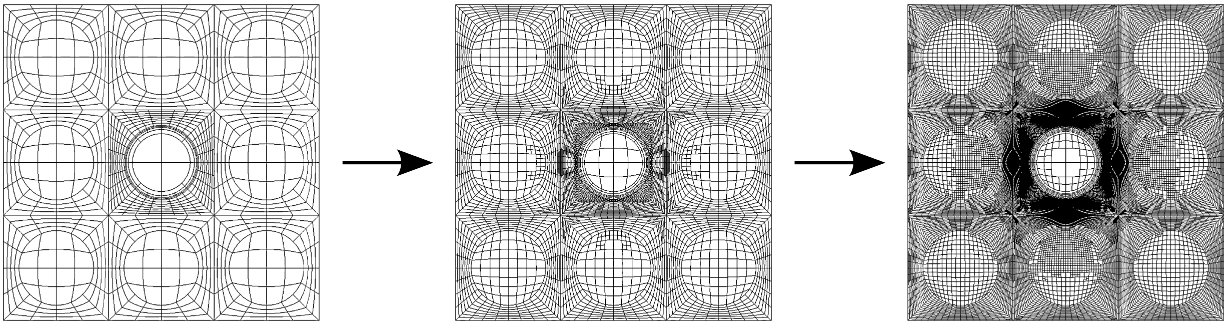


Fig. 20. Meshes for steps 4, 6 and 9 of the AMR process using the Residual error indicator and quadratic elements for the supercell problem. The elements are being concentrated around the sharp gradient in the flux in the vicinity of the burnable absorber. The other error indicators show similar results.

In order to demonstrate the advantage of IGA over FEM in the context of AMR, the same problem was solved using finite elements. This can be performed using the same program, as there is a one-to-one mapping between tensor-product (bilinear, biquadratic etc.) finite elements and two-dimensional B-splines. As the finite element and B-spline basis functions span the same function space in the parent element, the transformation between the two is simply a change of basis. The B-spline control points $\mathbf{B}_i$ can therefore be written as a linear combination of the finite element nodes $\mathbf{N}_j$:

$$\mathbf{B}_i = \sum_j A_{i,j}\,\mathbf{N}_j \tag{37}$$

with the $A_{i,j}$ calculated by Galerkin projection [38].

The issue with DGFEM AMR is that unless the CAD geometry is revisited at every iteration, the representation of the geometry does not change as the AMR proceeds. Therefore if this representation is not exact, as with circular pincells, this does not improve throughout the AMR process. In order to demonstrate this, the supercell geometry was set up using biquadratic finite elements using two different approaches.

In the first, all elements were straight-sided, and so the circles and annuli in the geometry are approximated by regular polygons. This approach is frequently taken in practice, as fuel regions can then easily be mass preserved using *a priori* knowledge of the circle radius and the number of sides in the regular polygon, and fissile mass preservation is crucial to obtaining accurate criticality solutions [10].

In the second, curved elements were used. These are formed from the (non mass preserved) straight-sided meshes described above by moving the mid-side nodes of element faces on a circle boundary onto the centre point of the circle

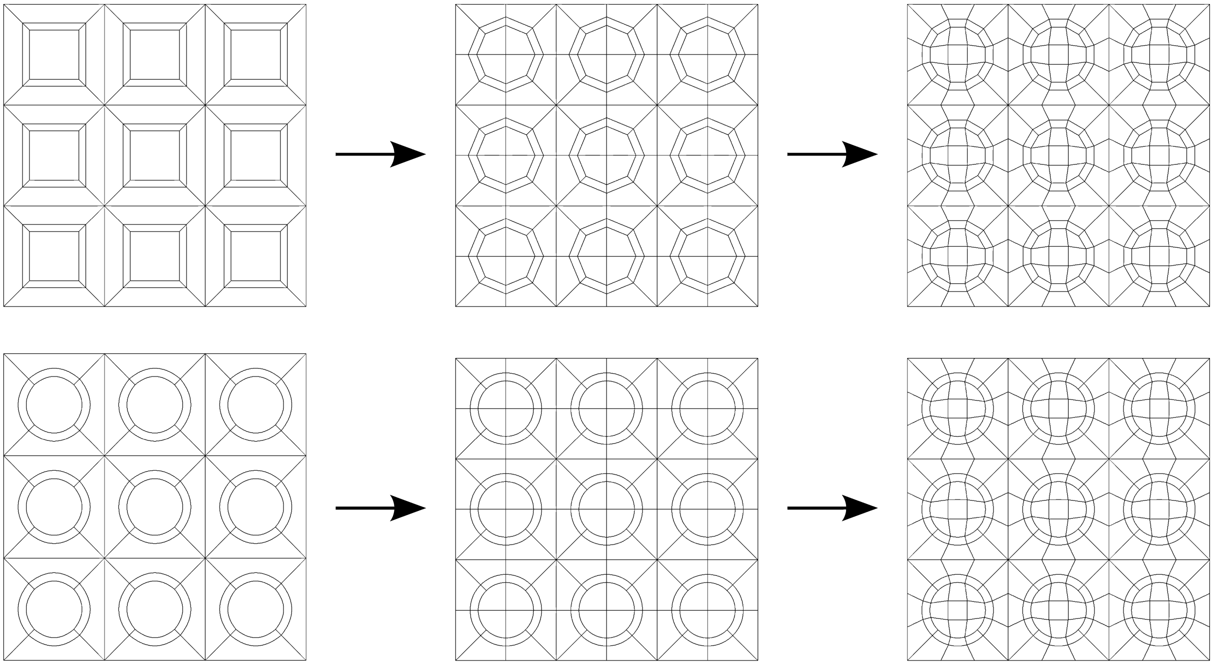


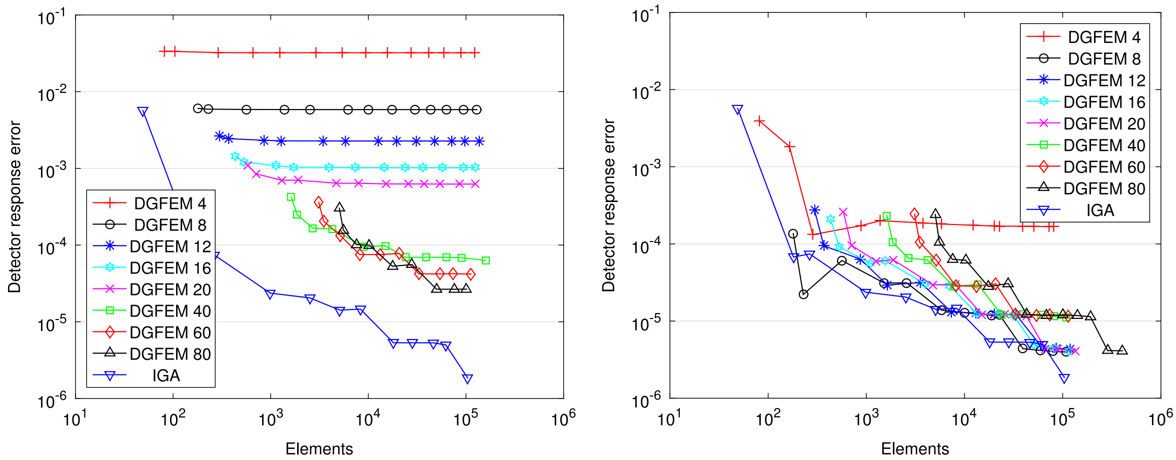
Fig. 21. A selection of the initial meshes used to solve the supercell problem with DGFEM. Biquadratic, straight-sided (top) and curved (bottom) elements were used, with the circular geometry approximated by 4 (left), 8 (middle) and 12 (right) line elements. As rational basis functions are required to model the circle exactly, the biquadratic curved elements cannot capture the geometry exactly, although this is not clear to the human eye at this scale.

arc between the two corner nodes on the same boundary. This produces a quadratic, rather than linear, approximation to the geometry, but it is still not exact. In particular, the fissile mass will always be underestimated using this approach. Conserving region volumes is no longer simple in this case, as regions are bounded by piecewise quadratic curves rather than regular polygons. To do so requires the numerical integration of the area of the elements approximating the circular geometry, followed by moving the nodes on the circumference of the circle away from the centre of the circle by an appropriate scaling factor, which is based on the ratio of this area to the exact area of the circle. Due to this complexity, this approach would not usually be used in practice.

However, in order to facilitate comparison between these two finite element approaches and IGA, all regions will be mass preserved in the following calculations. The supercell geometry was set up with the circle boundaries represented by regular polygons and piecewise quadratic curves with between 4 and 80 sides, some examples of which are shown in Fig. 21. The Residual error indicator driven AMR scheme was then used to refine this initial mesh, without changing the original geometry.

Fig. 22 shows the results of these methods, along with the quadratic IGA scheme using the Residual error indicator for comparison. With the straight-sided elements presented in Fig. 22(a), increasing the accuracy in the geometry representation decreases the error in the detector response. However, for a fixed representation of the geometry, the error in the detector response eventually plateaus as the mesh is adaptively refined. This suggests that the converged detector response for a given polygonal representation of the geometry is not the same as the converged detector response for the true circular geometry, and that this is not a viable method of reducing the error when using straight-sided finite elements. In contrast, the error in the detector response in the IGA scheme continues to decrease on the finest meshes considered, and is approximately an order of magnitude more accurate than any of the straight-sided finite element results.

The results when using the curved elements, presented in Fig. 22(b), are more complex. The coarsest representation of the geometry, denoted by DGFEM 4 in the legend, plateaus after a few AMR steps, similarly to the straight-sided element cases presented in Fig. 22(a). The AMR procedure was initially set to exit once a solution had been calculated on a mesh with at least 10^5 elements. These results implied that the detector response for the less accurate geometries (DGFEM 8–20) plateaued at a lower level than those of the more accurate geometries (DGFEM 40–80).



(a) Straight-sided elements.

(b) Curved elements.

Fig. 22. The detector response error when the supercell problem is solved using the Residual error indicator with straight-sided (a) and curved (b) biquadratic DGFEM. The DGFEM number refers to the number of sides of the piecewise linear (a) or quadratic (b) curve approximating the circles in each mesh.

However, the convergence behaviour of the curved elements, and to a lesser extent the straight-sided and IGA elements, are "stepped", in that many AMR steps produce little or no improvement in the detector response error, while others produce a step change. This could be caused by a phenomenon similar to that described in Fig. 13, in which the detector response accuracy only improves when certain regions of the mesh are refined, although this is more challenging to verify for this problem due to the relative complexity of the geometry. This hypothesis is, however, supported by the fact that the values of the error in the detector response at which the various schemes plateau coincide at a few discrete values, e.g. 1.2×10^{-5}.

In order to verify that the DGFEM 80 curved elements could achieve a detector response error as low as the DGFEM 8–20 schemes, the calculation was repeated with more AMR steps, so that the final mesh had $\approx 4 \times 10^5$ elements. It can be seen in Fig. 22(b) that the error does indeed decrease again.

A likely cause of this seemingly counterintuitive behaviour could be the initial finite element meshes used for the curved DGFEM schemes. As shown in Fig. 21, increasing the accuracy of the geometrical representation increases the number of elements in the initial mesh, e.g. the coarsest DGFEM 80 mesh has ×28 as many elements as the coarsest DGFEM 8 mesh. These extra elements are primarily in the burnable absorber and fuel regions, which are exactly the regions of the geometry that are refined the least by the AMR procedure as demonstrated in Fig. 20. These additional elements, while required to improve the geometrical representation with this mesh construction method, therefore essentially constitute wasted computational effort in terms of the ability of the mesh to accurately represent the solution until many AMR steps have been performed. Forming the initial curved element meshes in a less structured manner so that fewer elements are initially in the fuel pins and burnable absorber would likely alleviate this behaviour.

With the straight-sided elements, improving the accuracy of the geometrical representation seems to decrease the error at which the convergence plots in Fig. 22(a) plateau. In order to quantify the error in the geometry, the error in the perimeter of each fuel pin was used, as mass preservation results in the correct area for each pin. Fig. 23 shows that as more and more sides are used to represent a circle using a regular polygon, the corresponding error in the detector response decreases. With the more accurate representations of the geometry, the rate of convergence appears to decrease, although it is possible that this is because the lowest errors taken from Fig. 22(a) have not truly converged yet for these geometries. The same analysis has not been performed for the curved elements, as it is not clear from Fig. 22(b) if any but the coarsest geometrical representation has reached a true plateau with the number of AMR steps performed.

7. Conclusion

In this paper an adaptive, hanging-node, discontinuous isogeometric spatial discretisation has been developed and applied within the field of neutron transport; although the method is generally applicable to any hyperbolic system of

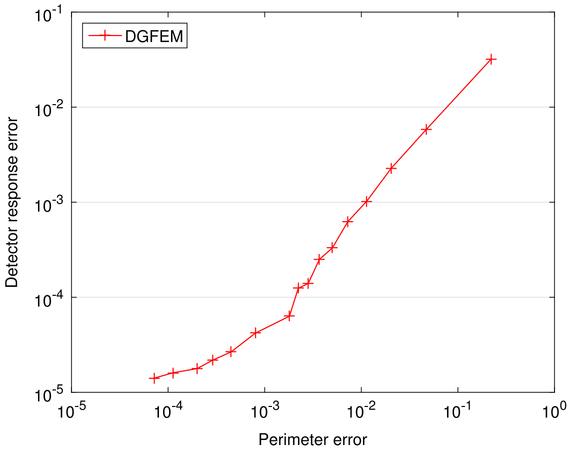


Fig. 23. The error in the detector response of the most refined straight-sided DGFEM meshes versus the error in the perimeter of the circles as modelled by these meshes for the supercell problem. As the geometry becomes more accurate, the error in the detector response decreases.

equations. The complexities that hanging-nodes and strongly curved boundaries introduce to the upwinding process have been overcome by performing all element boundary integration over the finest mesh being used on either side of the element face. This ensures that the scheme is conservative. A robust cycle-breaking algorithm has also been introduced to perform the sweep ordering of the elements for each discrete ordinates direction. In order to take advantage of the hanging-node formulation, simple and heuristic error indicators have been used to select elements for refinement at each step in an AMR scheme.

To verify that the hanging-node formulation obtains the correct order of convergence, a two dimensional, infinitely differentiable MMS problem was solved over the unit square. A discretisation of order p attained order $p + 1$ convergence with hanging-nodes present in the mesh for both Cartesian and pincell-style meshes for this problem.

Two test cases of a homogeneous square in a vacuum were presented to compare the performance of the scheme to uniform refinement when the underlying transport solution is not infinitely differentiable. When the mean free path is large (10% of the domain length), the AMR schemes concentrate the mesh refinement along the discrete ordinate directions emanating from the corners of the domain, as these are the characteristics across which the S_N solution can exhibit discontinuities. When the mean free path is small (0.1% of the domain length) the AMR schemes concentrate the mesh refinement along the domain boundary where there is an extremely sharp gradient in the solution. In both cases the L_2 error in the AMR solutions were consistently lower than the error in the solution when the mesh was uniformly refined on a per element basis for a given polynomial order.

A representative shielding benchmark containing a circular source region, an annulus of highly scattering material and an outer strongly absorbing region was studied to test the L_2 convergence of the scheme when the underlying transport solution is not infinitely differentiable and the geometry requires NURBS rather than polynomials to represent it. The AMR schemes again concentrate the elements in the mesh around sharp gradients in the solution and the discrete ordinate characteristic lines, while uniform refinement under-resolves these regions and over-refines regions of the geometry where the solution is slowly varying. Again the AMR schemes are consistently more accurate than uniform refinement for this problem.

The final test case considered was a 3 × 3 array of fuel pins, with the central pin replaced by a burnable absorber, all surrounded by a moderating material. This produces a sharp depression in the flux inside the burnable absorber, with a sharp gradient in the moderator region in the immediate vicinity of the burnable absorber. This problem was solved with an S_{32} angular quadrature set to try to mitigate the ray effect as much as possible. Again the AMR schemes concentrated the mesh refinement around the sharp gradient in the flux near the burnable absorber material, while uniform refinement under-refined this region and over-resolved other areas of the geometry. In this case the quantity of interest used was the absorption rate in one of the fuel pins. This was found to be approximately an order of magnitude more accurate when calculated using the same number of elements with the AMR schemes compared to uniform refinement.

In order to demonstrate the advantage of IGA for AMR over DGFEM, this same supercell problem was also solved using both straight-sided and curved finite elements. The geometry was fixed by an initial finite element mesh, and then the same AMR procedure as used with the IGA scheme was used to spatially converge the discrete ordinates solution with this geometry. With the straight-sided elements, the errors were initially found to decrease, but after a certain number of AMR steps (dependent on the initial geometry) this error plateaued and no further improvement in accuracy was obtained by refining the mesh further. The errors when the same problem was solved with an IGA scheme of the same polynomial order were approximately an order of magnitude more accurate than the straight-sided DGFEM results when the same number of elements were present in the mesh. The most refined straight-sided DGFEM solutions on each geometry were found to converge to the reference absorption rate with respect to the error in the perimeter of the circle that the polygonal geometry is approximating. This could imply that even when the fuel region in reactor physics problems is mass preserved, the error in the leakage rate across the material boundaries inhibits the achievable accuracy of a DGFEM scheme.

The curved finite elements were more accurate than their straight-sided counterparts, and sometimes even as accurate as the IGA scheme for the same number of elements in the mesh. However, these elements are not used in practice in reactor physics problems because of the additional preprocessing step required to ensure that fissile mass is conserved. The clear advantage of the IGA scheme in this respect is that the geometry, and therefore the fissile mass, is automatically exact from the coarsest level of refinement without any preprocessing.

This work has focused on one group problems in order to avoid introducing any bias by combining error indicators from each group in a multigroup problem into a single error indicator for an element in order to select those to refine. The exact geometry preservation of IGA naturally lends itself to overcoming this obstacle with the use of group dependent meshes. This has already been investigated with Cartesian geometries using continuous finite elements for the multigroup diffusion equation [39], which relies on a common description of the geometry at the coarsest mesh level for every energy group. As demonstrated in the supercell problem, in the presence of curved geometries this is not a feasible strategy with DGFEM as the geometrical errors introduced by this fixed geometry will truncate the achievable accuracy of the scheme. This can be overcome by using IGA, as the coarsest mesh description is exactly the geometry of the problem, and will be investigated in future work.

Acknowledgements

A.R. Owens would like to acknowledge the support of EPSRC under their industrial doctorate programme (EPSRC Grant Number: EP/G037426/1), 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) Ph.D. 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). In accordance with EPSRC funding requirements, all supporting data used to create figures and tables in this paper may be accessed at the following URL: http://doi.org/10.5281/zenodo.259450.

References

[1] K.S. Smith, Assembly homogenization techniques for light water reactor analysis, Prog. Nucl. Energy 17 (1986) 303–335.

[2] Richard Sanchez, Assembly homogenization techniques for core calculations, Prog. Nucl. Energy 51 (2009) 14–31.

[3] R.D. Lawrence, Progress in nodal methods for the solution of the neutron diffusion and transport equations, Prog. Nucl. Energy 17 (1986) 271–301.

[4] W.H. Reed, Triangular Mesh Difference Schemes for the Transport Equation, Technical Report LA–4769, Los Alamos Scientific Laboratory, 1971.

[5] W.H. Reed, T.R. Hill, Triangular Mesh Methods for the Neutron Transport Equation, Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

[6] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations, J. Comput. Phys. 131 (1997) 267–279.

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

[8] B.Q. Li, Discontinuous Finite Elements in Fluid Dynamics and Heat Transfer, Springer-Verlag, 2006.

[9] P.J. Frey, P.-L. George, Mesh Generation: Application to Finite Elements, Wiley, 2008.

[10] J.-Y. Moller, J.-J. Lautard, 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.

[11] K. Manalo, M. Chin, G. Sjoden, Cartesian meshing impacts for PWR assemblies in multigroup Monte Carlo and S_N transport, in: Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2013, 2013.

[12] T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Comput. Methods Appl. Mech. Engrg. 194 (2005) 4135–4195.

[13] Y. Bazilevs, K. Takizawa, T.E. Tezduyar, Computational Fluid-Structure Interaction: Methods and Applications, John Wiley and Sons, 2013.

[14] S.K. Hall, M.D. Eaton, M.M.R. Williams, The application of isogeometric analysis to the neutron diffusion equation for a pincell problem with an analytic benchmark, Ann. Nucl. Energy 49 (2012) 160–169.

[15] J.A. Welch, J. Kópházi, A.R. Owens, M.D. Eaton, Isogeometric analysis for the multigroup neutron diffusion equation with applications in reactor physics, Ann. Nucl. Energy 101 (2017) 465–480.

[16] F. Brunero, Discontinuous Galerkin Methods for Isogeometric Analysis (Master's thesis), Università degli Studi di Milano, 2012.

[17] Ulrich Langer, Angelos Mantzaflaris, Stephen E. Moore, Ioannis Toulopoulos, Multipatch discontinuous Galerkin isogeometric analysis, in: Isogeometric Analysis and Applications 2014, Springer International Publishing, 2015, pp. 1–32 (Chapter 1).

[18] C. Michoski, J. Chan, L. Engvall, J.A. Evans, Foundations of the blended isogeometric discontinuous Galerkin (BIDG) method, Comput. Methods Appl. Mech. Engrg. 305 (2016) 658–681.

[19] A.R. Owens, J.A. Welch, J. Kópházi, M.D. Eaton, Discontinuous isogeometric analysis methods for the first-order form of the neutron transport equation with discrete ordinate (S_N) angular discretisation, J. Comput. Phys. 315 (2016) 501–535.

[20] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, T.W. Sederberg, Isogeometric analysis using T-splines, Comput. Methods Appl. Mech. Engrg. 199 (2010) 229–263.

[21] Kjetil André Johannessen, Trond Kvamsdal, Tor Dokken, Isogeometric analysis using LR B-splines, Comput. Methods Appl. Mech. Engrg. 269 (2014) 471–514.

[22] Jean C. Ragusa, Yaqi Wang, A two-mesh adaptive mesh refinement technique for S_N neutral-particle transport using a higher-order DGFEM, J. Comput. Appl. Math. 233 (2010) 3178–3188.

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

[24] M.G. Cox, The numerical evaluation of B-splines, J. Appl. Math. 10 (1972) 134–149.

[25] Carl de Boor, On calculating with B-splines, J. Approx. Theory 6 (1972) 50–62.

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

[27] James J. Duderstadt, William R. Martin, Transport Theory, Wiley, 1979.

[28] D.A. Di Pietro, A. Ern, Steady advection-reaction, in: D.A. Di Pietro, A. Ern (Eds.), Mathematical Aspects of Discontinuous Galerkin Methods, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 37–65 (Chapter 2).

[29] Todd A. Wareing, John M. McGhee, Jim E. Morel, Shawn D. Pautz, Discontinuous finite element S_N methods on three-dimensional unstructured grids, Nucl. Sci. Eng. 138 (2001) 256–268.

[30] Yaqi Wang, Jean C. Ragusa, Standard and goal-oriented adaptive mesh refinement applied to radiation transport on 2D unstructured triangular meshes, J. Comput. Phys. 230 (2011) 763–788.

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

[32] Jose I. Duo, Yousry Y. Azmy, Ludmil T. Zikatanov, A posteriori error estimator and AMR for discrete ordinates nodal transport methods, Ann. Nucl. Energy 36 (2009) 268–273.

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

[34] James S. Warsa, A continuous finite element-based, discontinuous finite element method for S_N transport, Nucl. Sci. Eng. 160 (2008) 385–400.

[35] T.J.R. Hughes, A. Reali, G. Sangalli, Efficient quadrature for NURBS-based isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 199 (2010) 301–313.

[36] Yaqi Wang, Jean C. Ragusa, On the convergence of DGFEM applied to the discrete ordinates transport equation for structured and unstructured triangular meshes, Nucl. Sci. Eng. 163 (2009) 56–72.

[37] J. Wood, M.M.R. Williams, Recent progress in the application of the finite element method to the neutron transport equation, Prog. Nucl. Energy 14 (1984) 21–40.

[38] P.E. Farrell, J.R. Maddison, Conservative interpolation between volume meshes by local Galerkin projection, Comput. Methods Appl. Mech. Engrg. 200 (2011) 89–100.

[39] Yaqi Wang, Wolfgang Bangerth, Jean Ragusa, Three-dimensional h-adaptivity for the multigroup neutron diffusion equations, Prog. Nucl. Energy 51 (2009) 543–555.