Contents lists available at ScienceDirect

Annals of Nuclear Energy

journal homepage: www.elsevier.com/locate/anucene

A high-order discontinuous Galerkin method for the S_N transport equations on 2D unstructured triangular meshes

Yaqi Wang, Jean C. Ragusa *

Texas A&M University, Dept. of Nuclear Engineering, College Station, TX 77843-3133, USA

ARTICLE INFO

Article history:
Received 19 August 2008
Received in revised form 4 March 2009
Accepted 5 March 2009
Available online 7 April 2009

ABSTRACT

This paper presents high-order numerical solutions to the S_N transport equation on unstructured triangular meshes using a Discontinuous Galerkin Finite Element Method (DGFEM). Hierarchical basis functions, up to order 4, are used for the spatial representation of the solution. Numerical results are provided for source-driven and eigenvalue problems. Convergence rates (as a function of the mesh size and CPU time) are discussed.

1. Introduction

Due to the relative simplicity of the scheme and its similarity with the finite volume method, the Discontinuous Galerkin Finite Element Method (DGFEM) has been widely used to discretize hyperbolic partial differential equations (e.g., conservation laws) in space. Indeed, the lowest order of DGFEM is a finite volume method. In the transport community, predominantly linear DGFEM (i.e., DGFEM with linear shape functions) has been employed to solve the discrete ordinates S_N transport equation. The DGFEM method was originally derived for neutron/photon transport problems in the early 1970s, (Reed and Hill, 1973; Lesaint and Raviart, 1974). For instance, in the computer code TRIPLET of Reed et al. (1973), the method was presented for 2D regular triangular meshes with various polynomial order approximations. A few years later, the TRIDENT code for 2D multigroup triangular mesh S_N transport was released (Seed et al., 1977; Seed et al., 1978); TRIDENT was based on only a linear DGFEM method, which we denote hereafter by DGFEM(1). Similarly, most of the recent work related to the DGFEM discretization of the transport equation on unstructured meshes has been carried out using DGFEM(1). For instance, a linear function representation has been used for 3D unstructured tetrahedral meshes in Wareing et al. (2001); Morel and Warsa (2005) and a trilinear representation for 3D hexahedral meshes in Wareing et al. (2001). These works have led to the creation of the DGFEM(1)-based commercial code Attila McGhee and Wareing, 2000, developed by Los Alamos National Laboratory and Transpire, Inc. Even though the original work of Reed and Hill was not restricted to order-1 spatial representations, we note that not much research has been carried out for higher-order finite element transport to date. Several reasons can be put forth to explain this:

(1) In the TRIPLET code manual, Reed and Hill recommended the use of "linear finite elements for problems where the computational time was a significant consideration"; furthermore, their high-order basis functions used Lagrange polynomials based on equally-spaced points. Instead, we use a hierarchical basis, which allows for a simpler implementation of coupling between mesh cells; see Section 2 for further details.
(2) Subsequently, the TRIDENT code was developed, employing only linear finite elements; the more recent work by McGhee, Wareing, Morel, and Warsa inherited the legacy of these seminal works with respect to their choice of linear spatial representation, focusing on the accuracy and effectiveness of the transport solution in the thick diffusive limit and the robustness of the spatial discretization.

Nonetheless, in other disciplines, e.g., fluid dynamics, where hyperbolic conservation laws are also present, the development of higher-order DGFEM has significantly progressed. High-order basis functions for triangular meshes based on Jacobi polynomials have been derived and accurate solutions using higher-order functions can be obtained in fewer mesh cells (see Cockburn et al., 2000; Barth and Deconinck, 1999; Nektar code; Karniadakis and Sherwin, 2005). These improvements should be investigated for use in applications within the nuclear science and engineering community.

The case for higher-order methods can be made based on (i) the higher enhanced convergence rates (i.e., better accuracy in the solutions) and (ii) the fact that the computing time for higher-order methods may come virtually for free on modern computer architectures. Today's computational bottleneck for neutron transport is predominantly memory-access related, and low-order spatial discretizations may require more elements, i.e., more solves of a small linear system, than higher-order approximations. We recall here that in S_N transport sweeps, one needs only to solve a small

* Corresponding author. Tel.: +1 979 862 2033; fax: +1 979 845 6443.
E-mail addresses: yaqiwang@tamu.edu (Y. Wang), ragusa@ne.tamu.edu (J.C. Ragusa).

0306-4549/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.anucene.2009.03.002

linear system, obtained by sweeping the mesh element by element. For linear finite elements on triangles, this system is of size 3 × 3, which can potentially leave the processors starved for data to compute, whereas higher-order approximations yield larger elementary systems, of size $N(p) \times N(p)$, with $N(p) = (p+1)(p+2)/2$ and p the polynomial order, thus providing more data to the CPU per elementary solve. Therefore, a logical alternative to circumvent a memory-access constrained environment is to provide the CPU more data to compute at a single time by employing higher-order spatial representations.

Furthermore, another significant reason to consider higher-order polynomials is related to the fact that a DGFEM of order p, hereafter denoted by DGFEM(p), can yield convergence rates of order $q = \min(p+1, r)$ for hyperbolic equations, where r is the solution regularity. Lesaint and Raviart (1974) demonstrated that the convergence rate was p (and $p+1$ for rectangular grids); later, Richter (1988) improved the theoretical result for topologically regular triangular meshes to $p+1$. That is, the error term is of the form Ch^q, with C being a proportionality constant and h the typical mesh size.

In this work, we implemented a DGFEM(p) for polynomial orders up to 4, tested it on several benchmark problems, and report the findings in terms of accuracy (convergence) as a function of both the total number of unknowns and the CPU time. Our choice of high-order basis function consists of hierarchical functions. Such a set of functions is obtained by nesting the basis function sets, such that all lower-degree bases are included. Such sets have attractive features. They have, for example, better conditioning properties (Babuska et al., 1989) and may be implemented more easily through the use of nested elementary matrices and simple procedures to obtain edge values for inter-element communications. For instance, hierarchical bases have been used in Carpenter (1997), Ragusa and Wang (2009) for multigroup diffusion applications.

The outline of the paper is as follows. In Section 2, we present the DGFEM(p) formulation applied to the transport equation and the hierarchical basis functions used. In Section 3, we briefly review the S_N solution procedure for multigroup transport problems. Results are given in Section 4 and concluding remarks in Section 5.

2. Discontinuous finite element formulation

2.1. The discontinuous Galerkin finite element for neutron transport

We briefly present the DGFEM formulation for the transport equation. For brevity's sake, the presentation is done in the one-speed case. In the multigroup case, the only differences will be the addition of mass matrix terms to account for the fission and scattering coupling between groups; thus, the multigroup formulation will be later given as a simple extension of the one-speed case. Given an angular quadrature set $\{\vec{\Omega}_d, w_d\}_{1 \leqslant d \leqslant N_d}$, the one-speed steady state S_N neutron transport equation with anisotropic source and scattering, in a domain D with boundary ∂D, is:

$$\vec{\Omega}_d \cdot \vec{\nabla}\Psi(\vec{r},\vec{\Omega}_d) + \sigma(\vec{r})\Psi(\vec{r},\vec{\Omega}_d) = \sum_{\ell,m} \frac{2\ell+1}{4\pi}\sigma_{s,\ell}(\vec{r})\Phi_{\ell,m}(\vec{r})A_{\ell,m}(\vec{\Omega}_d) + Q(\vec{r},\vec{\Omega}_d) = q(\vec{r},\vec{\Omega}_d) \quad \text{on } D \tag{1}$$

$$\Psi(\vec{r}_b,\vec{\Omega}_d) = \Psi^{inc}(\vec{r}_b,\vec{\Omega}_d) + \sum_{\substack{d' \\ \vec{\Omega}_{d'}\cdot\vec{n} \geqslant 0}} \beta(\vec{r}_b, \vec{\Omega}_{d'} \to \vec{\Omega}_d)\Psi(\vec{r}_b,\vec{\Omega}_{d'}) \quad \text{on } \partial D^- = \{(\vec{r}_b \in \partial D, \vec{\Omega}_d \cdot \vec{n} < 0\}, \tag{2}$$

where the summation on indices ℓ and m is to be understood as a double summation, $A_{\ell,m}(\vec{\Omega}_d)$ is the set of spherical harmonics functions evaluated in direction $\vec{\Omega}_d$, Q the external source term, σ and $\sigma_{s,\ell}$ the total and ℓ^{th} moment scattering cross sections, respectively (they are assumed to be piece-wise constant), q is source term made of the sum of the scattering terms and of the Q source term, β the albedo, D the spatial domain and ∂D its boundary. For Dirichlet conditions, we set the albedo to zero, i.e., $\beta = 0$; for reflective conditions, we set $\beta = \delta(1 - \vec{\Omega}_d \cdot \vec{\Omega}_{d'})$ and $\Psi_{inc} = 0$, where $\vec{\Omega}_{d'}$ is determined by $\vec{\Omega}_{d'} = \vec{\Omega}_d - 2(\vec{\Omega}_d \cdot \vec{n})\vec{n}$, with $\vec{n}$ the outward normal unit vector on the boundary. All notations are standard. The flux moments are defined by:

$$\Phi_{\ell,m}(\vec{r}) = \sum_d w_d \Psi(\vec{r},\vec{\Omega}_d)A_{\ell,m}(\vec{\Omega}_d). \tag{3}$$

For notational simplicity, we let $\Psi_d(\vec{r}) = \Psi(\vec{r},\vec{\Omega}_d)$ and $q_d(\vec{r}) = q(\vec{r},\vec{\Omega}_d)$ and omit the direction subscript d whenever unnecessary.

A mesh T_h is used to discretize the domain D into linear elements K (specifically triangles in our application), such that the union of the all elements fully covers D, i.e., $\bigcup_{K \in T_h} K = D$. We suppose that the boundary of D consists of straight segments.

For the purpose of writing the bilinear form for the DGFEM formulation, we now introduce the volume and surface inner products on any element K and its boundary ∂K, respectively,

$$(f,g)_K = \int_K d^3r f(r)g(r), \tag{4}$$

$$(f,g)_{\partial K} = \int_{\partial K} d^2r\,|\vec{\Omega}\cdot\vec{n}| f(r)g(r). \tag{5}$$

In the case of two-dimensional geometries, these products are to be understood as surface and edge integrals. The DGFEM applied to Eq. (1) is obtained by multiplying Eq. (1) by a (discontinuous) test function b_i and integrating the result over each element, i.e.,

$$\sum_{K\in T_h}\left\{-(\vec{\Omega}\cdot\vec{\nabla}b_i,\,\Psi)_K + (b_i,\,\sigma\Psi)_K + (\Psi^+,\,b_i^+)_{\partial K^+}\right\} = \sum_{K\in T_h}\left\{(q,\,b_i)_K + (\Psi^-,\,b_i^+)_{\partial K^-}\right\}, \tag{6}$$

where ∂K^- is the inflow boundary ($\partial K^- = \{x \in \partial K$ such that $\vec{\Omega}\cdot\vec{n} < 0\}$), ∂K^+ is the outflow boundary ($\partial K^+ = \{x \in \partial K$ such that $\vec{\Omega}\cdot\vec{n} > 0\}$), f^+ denotes the restriction of the function f taken from within the element K and f^- represents the restriction of the function f taken from the neighboring element of K. In the case of straight element boundaries, the quantity $\vec{\Omega}\cdot\vec{n}$ can be factored out of the boundary integrals. The left-hand-side boundary integral in Eq. (6) contains the flux unknowns for each element K, whereas the right-hand-side boundary integral contains the upwind values, i.e., the incoming contributions from the neighboring upwind elements. When an element lies on the domain boundary ∂D, the incoming contribution is given by Eq. (2). It is easy to note that, if some portion of the boundary is supplied with an albedo condition ($\beta \neq 0$), then the (inflow) outer boundary contributions are coupled to the outgoing fluxes, which simply adds another iterative loop in the solution procedure.

2.2. Basis functions

We now introduce the basis and test functions used in this work. A good choice of functions is important for an easy and efficient implementation of high-order DGFEM. Over the course of the last decade, high-order and spectral finite elements have enjoyed many theoretical and applied developments; for instance, in the computational fluid dynamics field, several novel computer code systems are based on DGFEM (Nektar code; Fischer et al., 2002; Fischer and Rønquist, 1994) and several authors describe the various basis function options for high-order finite elements (Solin, 2005; Hesthaven and Warburton, 2007; Karniadakis and Sherwin, 2005).

We have in mind the subsequent development of an mesh adaptive *hp*-code, where the mesh can either be locally subdivided (*h*-refinement) or the polynomial order can be locally increased (*p* -refinement). In such applications, the use of hierarchical basis functions has been shown to allow for an efficient implementation and a flexible polynomial order selection; in neutron diffusion, hierarchical functions have been employed for 2D quadrilateral elements in Carpenter (1997); Ragusa and Wang (2009). Hierarchical bases are sets of functions where bases of lower-order are successively nested in higher-degree bases. Thus, an increase in the polynomial order means only adding new functions to the current set, rather than re-establishing a whole new set as would be the case with the traditional Lagrange polynomial bases. As a result, an elementary matrix for a polynomial approximation of order p contains, embedded in it, the matrices of order $p-1,\ p-2,\ \dots,\ 2$, and 1, as shown on Fig. 1 for $p=4$. The nested feature of the elementary matrices can be exploited for calculations where the polynomial order is not held constant throughout the domain. Even for a constant polynomial order, hierarchical sets are attractive: they possess better condition numbers (Solin, 2005; Demkowicz, 2007; Babuska et al., 1989) and provide a simple representation of the solution on any edge; this latter point is important for inter-element communication in the upwinding procedure. In this work, we implemented and analyzed the convergence properties of hierarchical basis functions for 2D unstructured geometries meshed with triangular elements. The unstructured meshes shown in Section 4 are generated with the Triangle mesh generator (Shewchuk, 1996). Before providing the definition of the basis functions used, we need to note that the basis functions are typically not written for any element of the physical geometry but for a reference element, obtained after applying an affine transformation that maps a physical element onto the reference element. Here, the reference element is the triangle $\widehat{K}$, with the three vertices $V_1=(-1,-1)$, $V_2=(1,-1)$, and $V_3=(-1,1)$; $\widehat{K}$ is defined as follows:

$$\widehat{K} = \{\xi_1, \xi_2 \in \mathbb{R}^2; -1 < \xi_1, \xi_2; \xi_1 + \xi_2 < 0\}, \qquad (7)$$

where we have introduced the reference coordinate system ξ_1, ξ_2. The reference triangle $\widehat{K}$ is shown in Fig. 2. The linear (first-order) basis functions are the standard ones:

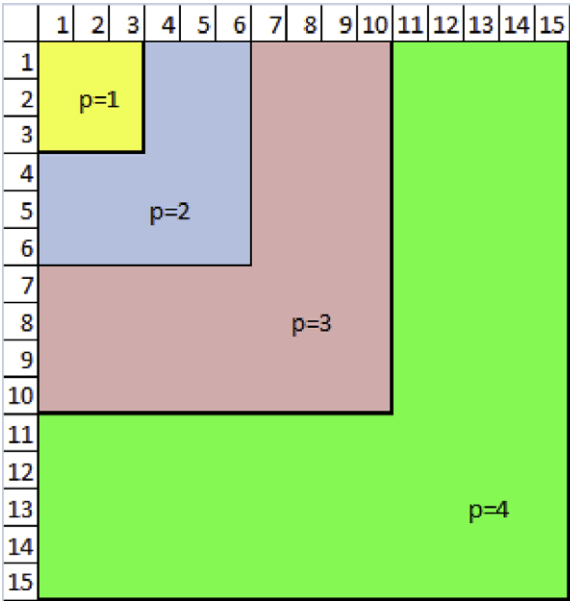

Fig. 1. Nesting of elementary matrices using hierarchical bases, example with $p=4$.

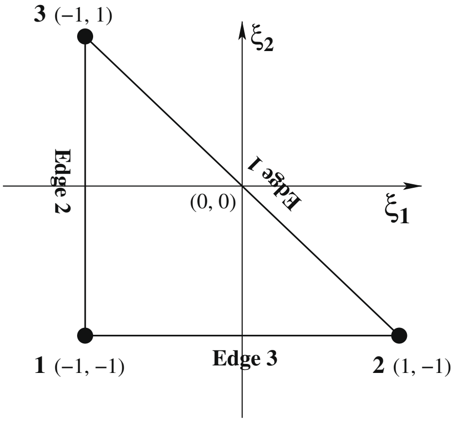

Fig. 2. Reference triangle: vertices and edges numbering.

$$\widehat{b}_1(\xi_1,\xi_2) = -\frac{\xi_1+\xi_2}{2}, \qquad (8)$$

$$\widehat{b}_2(\xi_1,\xi_2) = \frac{\xi_1+1}{2}, \qquad (9)$$

$$\widehat{b}_3(\xi_1,\xi_2) = \frac{\xi_2+1}{2}. \qquad (10)$$

The $\hat{}$ symbol denotes that the functions are defined on the reference element. The higher-order functions consist of edge functions (functions that are non-zero only on one of the edges) and interior functions (functions that are zero on all three edges). For a basis of order p, there are $p-1$ edge functions per edge and $(p-1)(p-2)/2$ interior functions. We have followed the definition by Solin (2005); Karniadakis and Sherwin (2005). We define the following function for indexing purposes, $N(q)=(q+1)(q+2)/2$, which represents the total number of basis functions with polynomial order up to q. Such indexing is consistent with Fig. 1.

For edge 1, linking vertices V_2 and V_3, we have

$$\hat{b}_{N(k-1)+1}(\xi_1,\xi_2) = \hat{b}_2(\xi_1,\xi_2)\hat{b}_3(\xi_1,\xi_2)\phi_{k-2}\left(\frac{\xi_2-\xi_1}{2}\right)$$
$$\text{for } 2 \leqslant k \leqslant p. \qquad (11)$$

For edge 2, linking vertices V_3 and V_1, we have

$$\widehat{b}_{N(k-1)+2}(\xi_1,\xi_2) = \widehat{b}_3(\xi_1,\xi_2)\widehat{b}_1(\xi_1,\xi_2)\phi_{k-2}\left(-\frac{\xi_1+2\xi_2+1}{2}\right)$$
$$\text{for } 2 \leqslant k \leqslant p. \qquad (12)$$

For edge 3, linking vertices V_1 and V_2, we have

$$\hat{b}_{N(k-1)+3}(\xi_1,\xi_2) = \hat{b}_1(\xi_1,\xi_2)\hat{b}_2(\xi_1,\xi_2)\phi_{k-2}\left(\frac{2\xi_1+\xi_2+1}{2}\right)$$
$$\text{for } 2 \leqslant k \leqslant p. \qquad (13)$$

Finally, the interior functions are given by

$$\begin{aligned}\widehat{b}_{N(n_1+n_2)+n_1+3}(\xi_1,\xi_2) &= \widehat{b}_1(\xi_1,\xi_2)\widehat{b}_2(\xi_1,\xi_2)\widehat{b}_3(\xi_1,\xi_2)\\ &\quad\times \phi_{n_1-1}\left(\frac{2\xi_1+\xi_2+1}{2}\right)\phi_{n_2-1}\left(-\frac{\xi_1+2\xi_2+1}{2}\right)\\ \text{for } 1 \leqslant n_1, n_2;\quad & n_1+n_2 \leqslant p-1. \qquad (14)\end{aligned}$$

We note that edge (resp. interior) functions only appear for $p>1$ (resp. $p>2$). The kernel functions $\phi_k(z)$ $(k \geqslant 0)$ are given in terms of the 1D Lobatto polynomials $L_{k+2}(x)$ of order $k+2$ defined on the $[-1,+1]$ interval:

$$\phi_k(z) = \frac{L_{k+2}(z)}{L_1(z)L_2(z)}, \quad k \geqslant 0. \qquad (15)$$

The total number of basis functions, and thus of unknowns per triangle, is given by $N(p) = (p+1)(p+2)/2$. The angular flux is, therefore, expanded on the set of discontinuous spatial basis functions b_j of order p,

$$\Psi_d(\vec{r}) = \sum_{j=1}^{j=N(p)} b_j(\vec{r})\Psi_{d,j} = \mathbf{b}^T\mathbf{\Psi}_d \tag{16}$$

(with $\mathbf{b} = \mathbf{col}(b_j)$ and $\mathbf{\Psi}_d = \mathbf{col}(\Psi_{d,j})$) and the transport equation, tested against any of these functions b_i, yields the DGFEM formulation of order p:

$$\begin{aligned}\sum_{K\in T_h}&\left\{-\vec{\Omega}_d\cdot\left[\int_K \vec{\nabla}\mathbf{b}\,\mathbf{b}^T\right]\mathbf{\Psi} + \left[\int_K \sigma\mathbf{b}\,\mathbf{b}^T\right]\Psi + \vec{\Omega}_d\cdot\left[\int_{\partial K^+}\vec{n}\mathbf{b}\,\mathbf{b}^T\right]\mathbf{\Psi}\right\}\\ &= \sum_{K\in T_h}\left\{\sum_{\ell,m}\left(\frac{2\ell+1}{4\pi}\right)\left[\int_K \sigma_s^\ell\mathbf{b}\,\mathbf{b}^T\right]\mathbf{\Phi}^{\ell,m}A_d^{\ell,m}\right.\\ &\qquad\left.+\left[\int_K \mathbf{b}\,Q\right] - \vec{\Omega}_d\cdot\left[\int_{\partial K^-}\vec{n}\mathbf{b}\,\mathbf{b}^T\right]\mathbf{\Psi}^-\right\}.\end{aligned} \tag{17}$$

Each matrix is computed on the reference triangle via a mapping from K to $\widehat{K}$ that leads to simple elementary matrices which can be pre-computed beforehand. Employing the Jacobian matrix J of the mapping, we have the following gradient, mass, and edge matrices:

$$\begin{aligned}G &= \vec{\Omega}_d\cdot\int_K \nabla_{\mathbf{x}}\mathbf{b}\,\mathbf{b}^T\,dxdy\\ &= \vec{\Omega}_d\cdot|J|\int_{\widehat{K}} J^{-T}\nabla_\xi\widehat{\mathbf{b}}(\xi_1,\xi_2)\widehat{\mathbf{b}}^T(\xi_1,\xi_2)\,d\xi_1 d\xi_2\end{aligned} \tag{18}$$

$$M = \int_K \mathbf{b}\,\mathbf{b}^T\,dxdy = |J|\int_{\widehat{K}}\widehat{\mathbf{b}}(\xi_1,\xi_2)\widehat{\mathbf{b}}^T(\xi_1,\xi_2)\,d\xi_1 d\xi_2 \tag{19}$$

$$E_1 = \int_{edge-1}\mathbf{b}\,\mathbf{b}^T d\mu = |J_{1d,1}|\int_{-1}^{1}\widehat{\mathbf{b}}(s,-1)\mathbf{b}^T(s,-1)ds \tag{20}$$

$$E_2 = \int_{edge-2}\mathbf{b}\,\mathbf{b}^T d\mu = |J_{1d,2}|\int_{-1}^{1}\widehat{\mathbf{b}}(-s,s)\widehat{\mathbf{b}}^T(-s,s)ds \tag{21}$$

$$E_3 = \int_{edge-3}\mathbf{b}\,\mathbf{b}^T d\mu = |J_{1d,3}|\int_{-1}^{1}\widehat{\mathbf{b}}(-1,-s)\widehat{\mathbf{b}}^T(-1,-s)ds, \tag{22}$$

where $\nabla_{\mathbf{x}} = [\partial_x, \partial_y]^T$ is the usual gradient operator in the physical space, ∇_ξ is the gradient in the reference space, and the cosine directors of $\vec{\Omega}_d$ are $\Omega_{d,x} = \vec{\Omega}_d\cdot\vec{e}_x$ and $\Omega_{d,y} = \vec{\Omega}_d\cdot\vec{e}_y$. The determinant of the Jacobian matrix J is simply half of the triangle area. For the edge matrices, the affine transformation is simply a one-dimensional mapping from both extremities of the edge onto the interval $[-1;+1]$. The determinant of the 1D Jacobian matrix, $|J_{1d,i}|$ for the three edges $i = 1,\ldots,3$, is half of the edge length.

We note that the definition of these basis functions allows for an easy access to the edge values, regardless of the chosen polynomial order; indeed, the flux on any edge is obtained by using the two non-zero linear basis functions on that edge and all the edge bubble functions for that edge. This is particularly useful to compute efficiently the upwinding contributions for any polynomial order and is an attractive feature for subsequent mesh adaptivity developments.

3. Solution procedure

In DGFEM applied to the discrete ordinates transport equation, the system of equations formed by Eq. (17) does not need to be assembled for all elements but can be solved element by element one direction at a time, provided that a proper ordering of the elements has be obtained; this mechanism is often referred to as a "transport sweep" and the element ordering depends on the sweeping direction $\vec{\Omega}_d$. With an appropriate numbering of the angular flux and flux moment unknowns into the $\mathbf{\Psi}$ solution vector for the angular fluxes and $\mathbf{\Phi}$ for the flux moments, the discretized transport equation can be represented in the following matrix form:

$$\mathbf{L\Psi} = \mathbf{M\Sigma\Phi} + \mathbf{Q}, \tag{23}$$

$$\mathbf{\Phi} = \mathbf{D\Psi}, \tag{24}$$

where $\mathbf{L}$ is the matrix due to streaming and total collisions, $\mathbf{\Sigma}$ is the scattering mass matrix which operates on the flux moments, $\mathbf{M}$ is the moment-to-direction matrix, $\mathbf{D}$ is the direction-to-moment matrix. The dimensions of $\mathbf{L}$ are the number of spatial degrees of freedom times the number of directions, $(N_{dof}\times N_d)^2$; in a discontinuous method, the number of degrees of freedom is simply the number of elements N_{el} times the number of unknowns per element, i.e., $N_{dof} = N_{el}\times N(p)$, assuming a uniform polynomial order is employed. The dimensions of $\mathbf{\Sigma}$ are $(N_{dof}\times N_{mom})^2$, where N_{mom} is the number of moments employed in the spherical harmonics expansion of the angular fluxes; N_{mom} is equal to $\frac{(L_a+1)(L_a+2)}{2}$ in 2D for standard angular quadratures, with L_a the anisotropic scattering order. $\mathbf{D}$ is of dimension $(N_{dof}\times N_{mom})\times(N_{dof}\times M)$; $\mathbf{M}$ is of dimension $(N_{dof}\times M)\times(N_{dof}\times N_{mom})$. We can recast the problem in terms of flux moments as follows:

$$[\mathbf{I} - \mathbf{DL}^{-1}\mathbf{M\Sigma}]\mathbf{\Phi} = \mathbf{DL}^{-1}\mathbf{Q}. \tag{25}$$

None of these global matrices are explicitly formed and $\mathbf{L}$ is inverted in a "matrix-free" fashion using transport sweeps. A complete transport sweep requires $N_{el}\times N_d$ local solves. In 2D geometries discretized using triangular meshes, there always exists an ordering such that $\mathbf{L}$ is block-lower triangular. For other geometries and/or dimensionalities, the graph of the sweep may present some cycles (or dependencies), which are typically broken and iterated upon in order to retain the "matrix-free" nature of the sweeping process. Note that in the case of albedo conditions on opposing faces, cycles are also present in 2D triangular geometries. The sweep through the domain for all directions makes up one inner iteration of any transport algorithm. Wrapped around this innermost loop, one typically finds the Source Iteration (SI) loop. The SI scheme can be written as

$$\mathbf{\Psi}^{(i+1)} = \mathbf{L}^{-1}\mathbf{M}[\mathbf{\Sigma\Phi}^{(i)} + \mathbf{Q}] \tag{26}$$

$$\mathbf{\Phi}^{(i+1)} = \mathbf{D\Psi}^{(i+1)}. \tag{27}$$

Convergence of the SI scheme is based on the difference between successive iterates of the flux moments and significant angular fluxes (significant angular fluxes are angular flux unknowns on albedo boundaries). The SI scheme is known to be slow converging for highly diffusive problems and a Diffusion Synthetic Acceleration (DSA) (Adams, 2001) has been implemented to address this issue. This DSA technique is not the focus of this paper and will not be presented here but in a subsequent communication; nevertheless, it suffices to say here that it was derived from the discretized transport equation, naturally yielding a DGFEM diffusion equation with interior penalty terms. For more details on interior penalty methods, please refer to Kanschat (2007). Finally, the above description omitted any notion of energy discretization. In the multigroup approximation, the source term Q for group g is simply given by the sum of the external source emitting particles in group g and the inscattering source from all other groups g' to group g:

$$Q^g(r,\vec{\Omega}_d) = Q^g_{ext}(r,\vec{\Omega}_d) + \sum_{\substack{g'\\ g'\neq g}}\sum_{\ell,m}\frac{2\ell+1}{4\pi}\sigma_{s,\ell}^{g'\to g}(\vec{r})\Phi_{\ell,m}^{g'}(\vec{r})A_{\ell,m}(\vec{\Omega}_d) \tag{28}$$

For fast energy groups (i.e., groups without upscattering), one transport solve provides the solution in that group. Thermal groups are iterated upon due to the presence of upscattering. In the case of a multigroup eigenproblem, the external source is omitted and an outer iteration on the fission source $\sum_{g'} \nu\sigma_f^{g'}(\vec{r})\Phi_{0,0}^{g'}(\vec{r})$ is performed. This outer iteration is accelerated by means of a Chebyshev acceleration; see Ferguson and Derstine (1977) for more details on the Chebyshev acceleration technique for k-eigenvalue problems.

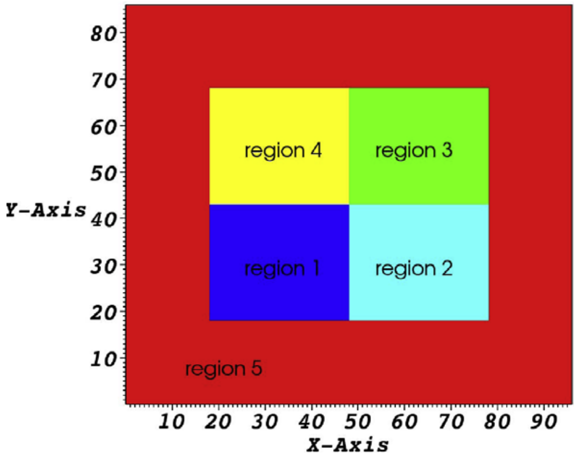

Fig. 3. Geometry of the IAEA-EIR-2 benchmark.

4. Results

We propose three test cases, one source-driven problem and two eigenproblems. The angular quadrature used is the S_8 level symmetric quadrature from the NEWT code of the SCALE package (DeHart, 2006), unless otherwise noted.

4.1. One-group source problem: the EIR-2 benchmark

The definition (geometry and cross-section data) of this problem can be found in several publications (Khalil, 1985; Kavenoky et al., 1979); it is commonly referred to as the “Stepanek problem” or the IAEA-EIR-2 problem. It consists of two source regions (regions #1 and 3) and two absorbing regions (regions #2 and 4). These four regions are arranged in a checkerboard fashion and are surrounded by region #5, see Fig. 3. Vacuum boundary conditions are applied. Various unstructured meshes were obtained using the Triangle mesh generator by employing different maximum element area constraints. A total of seven meshes were created, with the following maximum triangle areas, in cm^2: 60, 30, 15, 5, 2, and 1, resulting in meshes having 20, 206, 431, 869, 2553, 6441, and 12,893 elements, respectively. Fig. 4 shows meshes #1, 2, 6, 7. Fig. 5 presents the convergence rate for the reaction rate in region #3, as a function

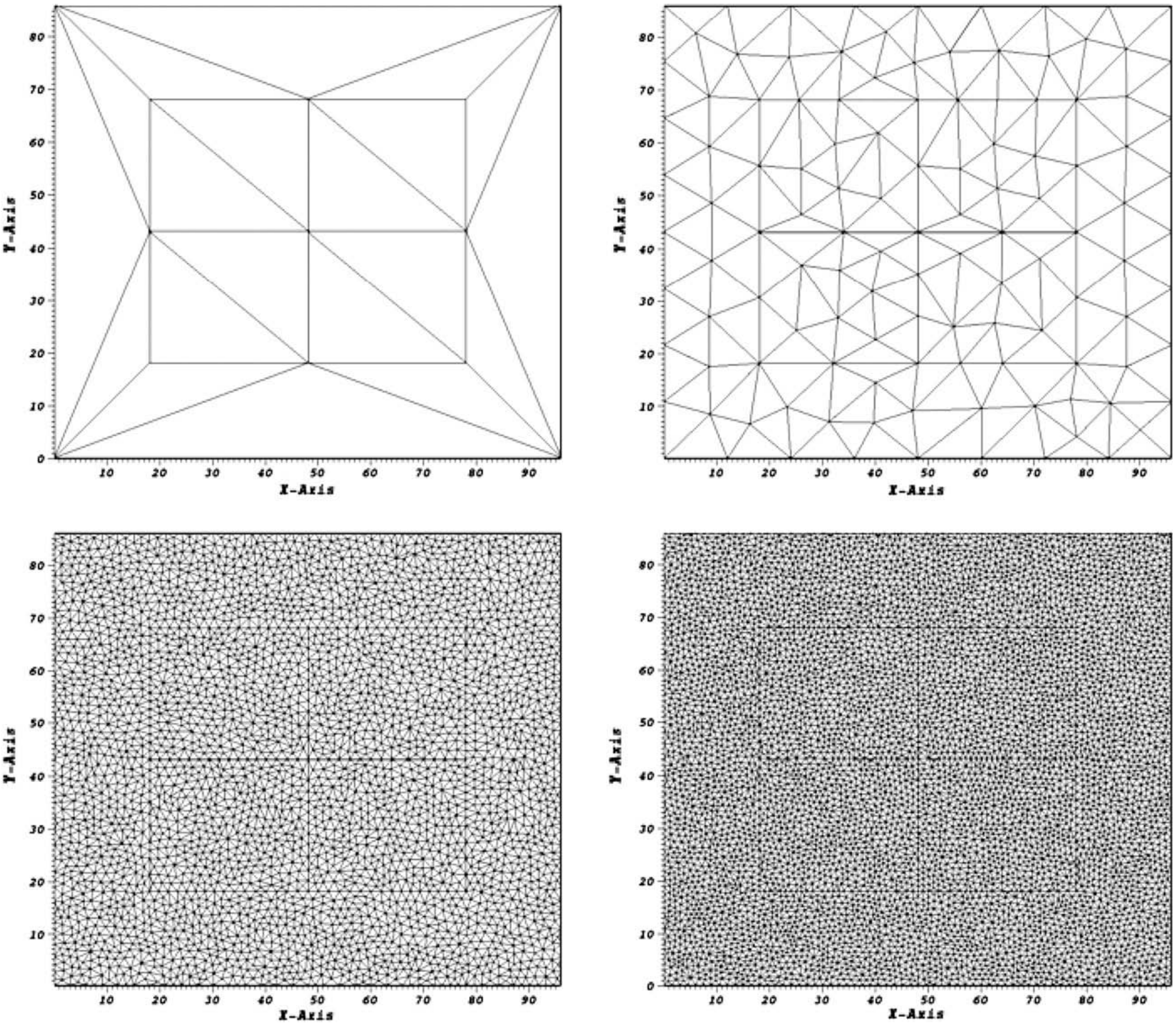

Fig. 4. EIR-2 benchmark: meshes #1 (top left), #2 (top right), #5 (bottom left) and #6 (bottom right).

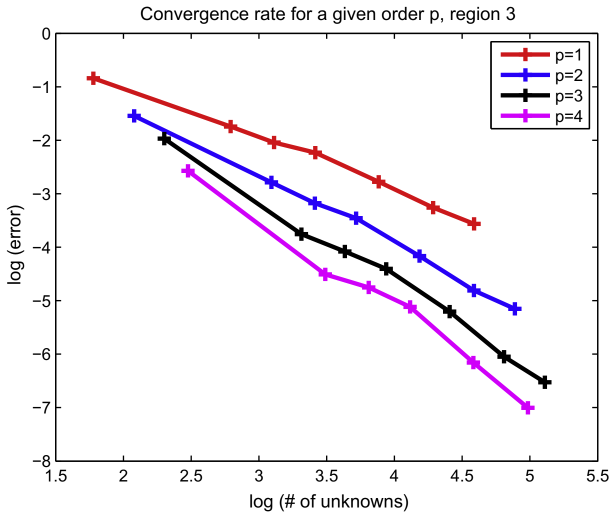

Fig. 5. EIR-2 benchmark: convergence rates for the average flux in region #3 as a function of the number of unknowns, approximations DGFEM(1) through DGFEM(4).

Table 1
EIR-2 benchmark: convergence rates.

p	Convergence of rate in *N*	Convergence rate in *h*
1	1.12	2.24
2	1.47	2.94
3	1.84	3.68
4	2.14	4.28

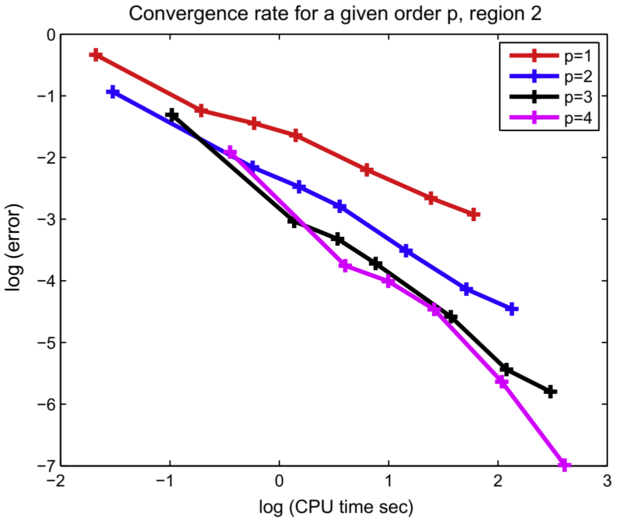

Fig. 6. EIR-2 benchmark: convergence rates for the average flux in region #2 as a function of CPU time, approximations DGFEM(1) through DGFEM(4).

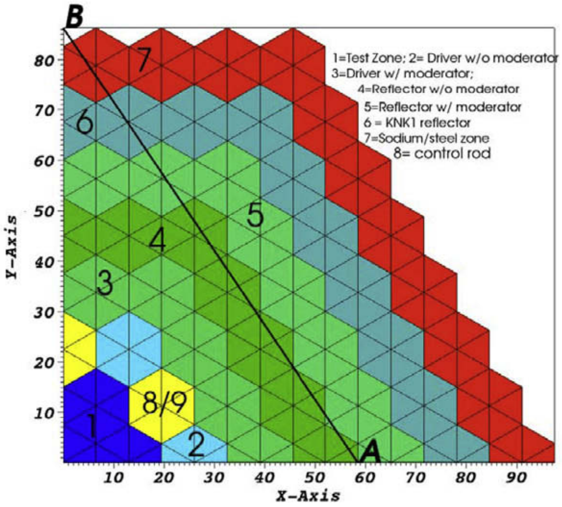

Fig. 7. Geometry of the Takeda benchmark and initial triangular mesh.

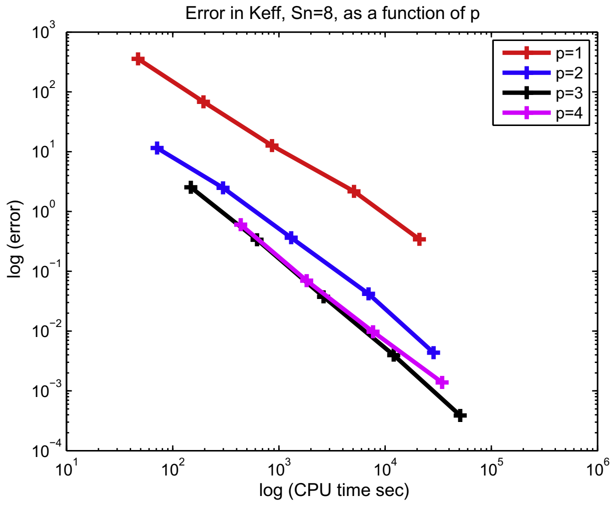

Fig. 8. Takeda benchmark: convergence rates in the k_{eff} as a function of CPU time, approximations DGFEM(1) through DGFEM(4).

of the number of (scalar flux) unknowns. For unstructured meshes, plotting the convergence rate as a function of mesh size is delicate because the element mesh size is not uniform. Nonetheless, in this case, it is common practice to graph convergence rates as a function of the total number of elements in a mesh. Since the total number of elements in a mesh, N_{elem}, is proportional to the square of the typical element size, or $h \propto \sqrt{N_{elem}}$, the slopes inferred from the loglog graphs simply have to be multiplied by two to retrieve the convergence rates as a function of the mesh size. Another, yet similar, option consists in graphing the convergence rates as a function of the number of unknowns, N. In DGFEM, there is a simple relation between N and N_{elem}: namely, for triangular meshes with basis functions of order p, we have $N = \frac{(p+1)(p+2)}{2} N_{elem}$; we have chosen this

Table 2
Takeda benchmark: error in the k_{eff}, in pcm, using approximations DGFEM(1) through DGFEM(4) on the initial mesh and four uniformly refined meshes.

p	Six triangles	24 Triangles	96 Triangles	384 Triangles	1536 Triangles
1	355.161	68.200	12.620	2.166	0.342
2	11.534	2.489	0.364	0.042	0.004
3	2.534	0.336	0.037	0.004	0.0004
4	0.602	0.070	0.010	0.001	Ref. k_{eff} = 1.0108202

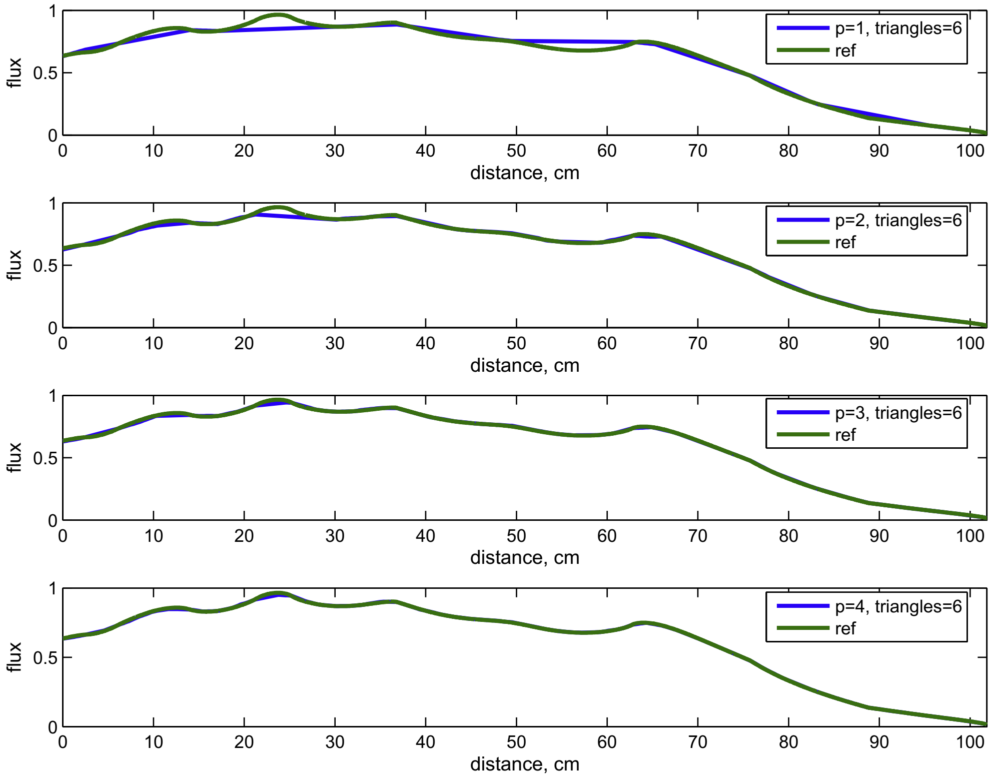

Fig. 9. Takeda benchmark: flux values along line AB for group #4, approximations DGFEM(1) through DGFEM(4), initial mesh.

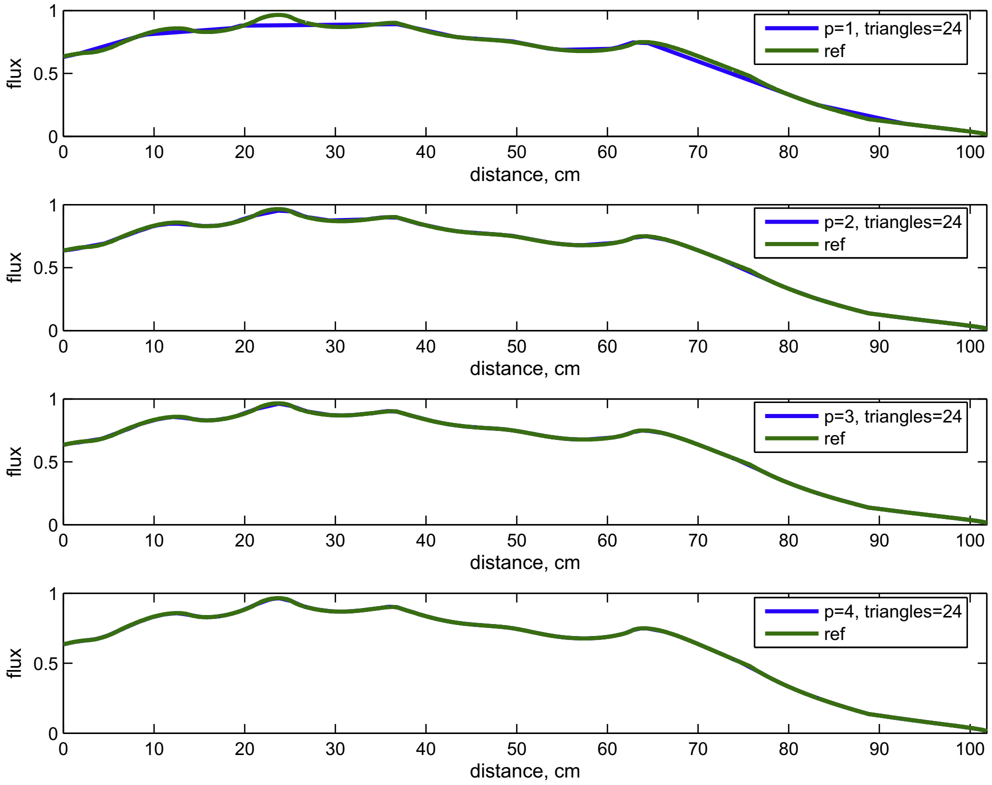

Fig. 10. Takeda benchmark: Flux values along line AB for group #4, approximations DGFEM(1) through DGFEM(4) once-refined mesh.

latter plotting convention because (i) the estimation of the convergence rates is not affected by it and (ii) such a convention accounts for the different memory requirements incurred by the different polynomial orders. In Fig. 5, we graphed the convergence rates for the average flux in region #3 (similar results are obtained for any region). We can note that the various polynomial approximations in DGFEM converge at different rates; Table 1 provides the convergence rates as seen in Fig. 5, as well as convergence rates in terms of mesh size. These values are in good agreement with the theoretical expected rates of $p+1$ (for smooth enough solutions.) For the same mesh, we also observe that the error is significantly smaller as the order p of the numerical approximation is increased, and that these gains diminish as p increases. Clearly, the convergence of the DGFEM(1) is poorer than that of the other methods. Fig. 6 provides the loglog plot of the error versus CPU time, for region #2. A linear approximation requires about 10 times longer than a quadratic approximation in order to reach an error below 0.01. The approximation $p=3$ provides also some benefits over the quadratic approximation for higher-accuracy. Finally, using $p=4$ does not provide much CPU gain versus $p=3$ for a certain accuracy.

4.2. Four-group eigenproblem: the KNK fast reactor benchmark

This benchmark problem, documented by Takeda and Ikeda (1991), is a model of the KNK-II fast reactor core. The geometry, an hexagonal lattice, is given in Fig. 7. The 4-group cross sections can be found in Takeda and Ikeda (1991). The 2D version of this problem has been recently used by Kim and Cho (1996) and Lu et al. (2007) for nodal S_N methods; we present here the results related to the rodded case. The domain is meshed by subdividing every hexagon into six triangles, resulting in the coarsest mesh utilized here. Finer meshes were obtained by regularly subdividing each triangle into four smaller triangles. Four different uniform refinements of the initial mesh were used, thus, each hexagon was treated with either 6, 24, 96, 384, 1536 triangles. Table 2 presents the error, provided in pcm, in the k_{eff} eigenvalue for various polynomial orders and uniform refinement levels. The reference k_{eff} was obtained with $p=4$ and 1536 triangles per hexagon. From Table 2, we note that DGFEM(1) with 96 triangles/hexagon (i.e., 288 unknowns/hexagon) has about the same accuracy as DGFEM(2) with six triangles/hexagon (i.e., 36 unknowns/hexagon). Similarly, DGFEM(2) with 24 triangles/hexagon (i.e., 144 unknowns/hexagon) has about the same accuracy as DGFEM(3) with six triangles/hexagon (i.e., 60 unknowns/hexagon). The convergence of the DGFEM(1) seems relatively poor compared to higher-order approximations; it takes about four levels of uniform refinement for the linear approximation to yield the same accuracy as the fourth order method on the initial mesh or the cubic method on the once-refined mesh. This is also illustrated in Fig. 8, where the convergence in k_{eff} is plotted as a function of CPU time. In Fig. 8, we note again the comparable effectiveness of the cubic and fourth order method, suggesting that approximations with orders greater than 4 may not be cost-effective in terms of CPU time.

We also compared the detailed spatial solutions by graphing a 1D cut throughout the geometry; this cut is represented by the AB line in Fig. 7. Figs. 9 and 10 provide the values along the AB line for the four polynomial orders as well as the first two meshes (six triangles/hexagon and 24 triangles/hexagon). The reference values are taken from a fourth order calculation, using 1536 triangles per hexagon. We note that on the initial mesh (six triangles/hexagon), there is a good agreement for $p \geqslant 3$ and for the next once-refined mesh, a good agreement is achieved starting at $p \geqslant 2$. Here again, the linear solution shows the poorest results.

4.3. Seven-group eigenproblem: the UOX/MOX C5G7 benchmark

This last test is the 2D version of the UOX/MOX C5G7 benchmark, where the fuel pins are represented by homogeneous cylinders (a homogenization of the fuel and its cladding) surrounded by water. The geometry consists of four 17 × 17 fuel assemblies, each composed of 264 fuel pins and 25 water holes. These fuel assemblies are surrounded by a homogeneous reflector (water), resulting in a mini-core geometry of 3 × 3 fuel assemblies, see Lewis (2001) for the complete description of the geometry and the cross-section data. In our triangularization, each fuel pin is approximated by a regular dodecagon (12-sided polygon) whose side is such that

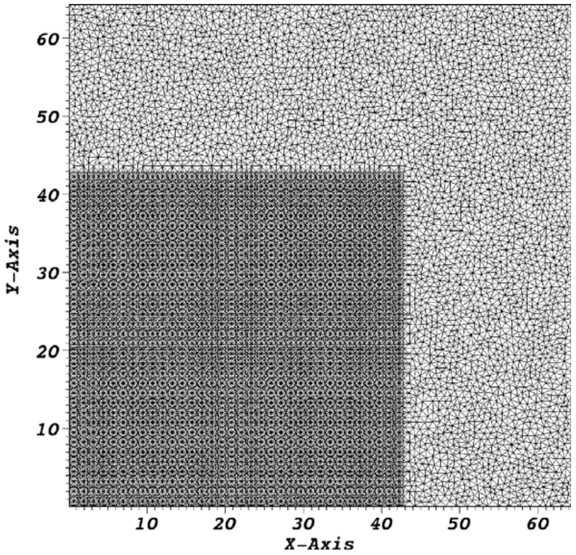

Fig. 11. Mesh used for the C5G7 benchmark.

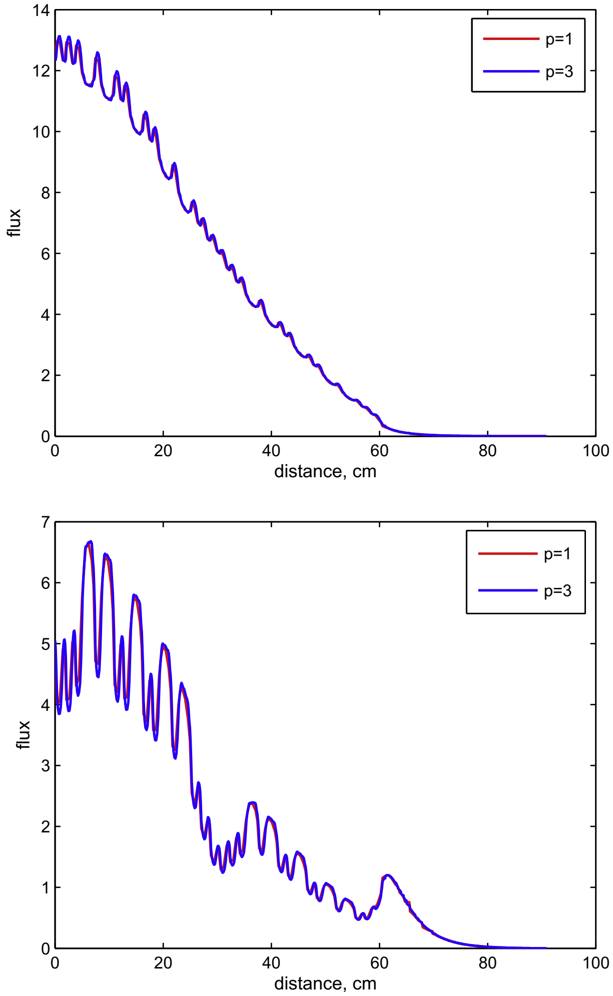

Fig. 12. C5G7 benchmark: flux values along the main diagonal (0,0)–(64.26,64.26) for groups #1 (top) and #7 (bottom), approximations DGFEM(1) and DGFEM(3).

the area of the fuel pin is preserved (a "conservation of fuel" principle). The resulting mesh, containing 39,633 elements, is shown in Fig. 11. A Gauss–Chebyshev product quadrature (with 4 polar angles and 16 azimuthal angles per quadrant) is used. The k_{eff} using a linear approximation is 1.18641(47), whereas the cubic approximation yields 1.18641(90); the published reference value, obtained from a Monte Carlo calculation is 1.18655(0) (Smith et al., 2004). Fig. 12 provides the 1D cross-sectional cut of the flux along from the main diagonal (from point (0,0) to point (64.26,64.26)), for groups 1 and 7 using the $p = 1$ and $p = 3$ methods. Even though the k_{eff} is well approximate using linear finite elements on this fine mesh, some discernible discrepancies in the local flux values can still be seen in Fig. 12 for DGFEM(1) when compared to DGFEM(3).

5. Conclusions

We have presented high-order numerical solutions to the S_N transport equation for unstructured 2D triangular meshes. A Discontinuous Galerkin Finite Element Method (DGFEM) was employed, with polynomial orders up to four. Hierarchical basis functions were chosen for the spatial representation of the solution. This basis set allows for an easy implementation of the upwinding procedure for any polynomial order and is widely used in other engineering disciplines for accurate high-order solutions. We have numerically observed that the solutions converge at the theoretical rate of $p + 1$ for smooth solutions, where p is the order of the approximation. Notably, for a given mesh size, there are always significant gains in accuracy to be obtained from quadratic, cubic, and quartic approximations over using linear DGFEM. In terms of CPU time, cubic and quartic functions yield about the same accuracy per unit time, suggesting that, in 2D, the best accuracy/CPU time compromise may be attained for orders 3 and 4. Our results show that a linear representation of the solution is not optimal due to the lower convergence rate of the method, and we recommend that at least second-order functions be employed. Using the above framework, a mesh adaptive S_N transport solver is under development; the use of a DGFEM method will facilitate the coupling between elements of various refinement levels (in DGFEM, the solution is not required to be continuous across elements) and the hierarchical basis functions will allow for simpler inter-element communications.

References

Adams, Marvin L., 2001. Discontinuous finite element transport solutions in thick diffusive problems. Nucl. Sci. Eng. 137, 298–333.

Babuska, I., Griebel, M., Pitkaranta, J., 1989. The problem of selecting the shape functions for a p-type finite element. Int. J. Numer. Methods Eng., 1891–1908.

Barth, Timothy J., Deconinck, Herman (Eds.), 1999. High-Order Methods for Computational Physics. Lecture Notes in Computational Science and Engineering, vol. 9. Springer-Verlag.

Carpenter, D.C., 1997. Two-dimensional finite element neutron diffusion analysis using hierarchic shape functions, In: Proc. Joint Int. Conf. Mathematical Methods and Supercomputing Applications, Saratoga Springs, NY, ANS.

Cockburn, Bernardo, Karniadakis, G., Shu, C. (Eds.), 2000. Discontinuous Galerkin Methods: Theory Computation and Applications. Lecture Notes in Computational Science and Engineering, vol. 11. Springer-Verlag, New York.

DeHart, M.D., 2006. NEWT: a new transport algorithm for two-dimensional discrete ordinates analysis in non-orthogonal geometries. ORNL/TM-2005/39, Version 5.1, vol. II. Book 4, Section F21 (November).

Demkowicz, L., 2007. In: Computing with hp-Adaptive Finite Elements; vol. 1: One and Two Dimensional Elliptic and Maxwell Problems. Applied Mathematics and Nonlinear Science Series. Chapman & Hall/CRC.

Ferguson, D.R., Derstine, K.L., 1977. Optimized iteration strategies and data management considerations for fast reactor finite difference diffusion theory codes. Nucl. Sci. Eng. 64, 593–604.

Fischer, P.F., Kruse, G.W., Loth, F., 2002. Spectral element methods for transitional flows in complex geometries. J. Sci. Comput. 17 (1), 81–98.

Fischer, P.F., Rønquist, E.M., 1994. Spectral element methods for large scale parallel Navier–Stokes calculations. Comput. Methods Appl. Mech. Eng. 116, 69–76.

Hesthaven, Jan S., Warburton, T., 2007. Nodal Discontinuous Galerkin Methods: Algorithms Analysis and Applications. Springer.

Kanschat, G., 2007. Discontinuous Galerkin Methods for Viscous Flow. Deutscher Universittsverlag, Wiesbaden.

Karniadakis, George Em, Sherwin, Spencer J., 2005. Spectral/hp element methods for computational fluid dynamics, second ed. Oxford University Press.

Kavenoky, A., Stepanek, J., Schmidt, F., 1979. Benchmark problems. Transport Theory and Advanced Reactor Simulations. IAEA-TECDOC-254. IAEA, Vienna, p. 305.

Khalil, H., 1985. A nodal diffusion technique for synthetic acceleration of nodal S_N calculations. Nucl. Sci. Eng. 90, 263.

Kim, T.H., Cho, N.Z., 1996. Source projection analytic nodal S_N method for hexagonal geometry. Ann. Nucl. Energy 23 (2), 133–143.

Lesaint, P., Raviart, P.A., 1974. On a finite element method for solving the neutron transport equation. In: deBoor, C.A. (Ed.), Mathematical Aspects of Finite Elements in Partial Differential Equations. Academic Press, New York, pp. 89–145.

Lewis, E., et al., 2001. Benchmark specification for Deterministic 2D/3D MOX fuel assembly transport calculations without spatial homogenisation (C5G7 MOX). OECD/NEA Expert Group on 3-D Radiation Transport Benchmarks. NEA/NSC/DOC(2001)4.

Lu, Haoliang et al., 2007. Two-dimensional nodal transport method for triangular geometry. Ann. Nucl. Energy 34, 424–432.

McGhee, John M., Wareing, Todd A., 2000. Attila version 2: user's manual. Los Alamos Scientific Laboratory Report LA-UR-00-5778, Los Alamos, NM.

Morel, Jim E., Warsa, James S., 2005. An Sn spatial discretization scheme for tetrahedral meshes. Nucl. Sci. Eng. 151, 157–166.

Nektar Code. A Navier–Stokes spectral/hp solver. available at the Nektar resource website: <http://www2.imperial.ac.uk/ssherw/spectralhp/nektar>/.

Ragusa, J., Wang, Y., 2009. Application of hp-adaptivity to the multigroup diffusion equations. Nucl. Sci. Eng. 161, 1–27.

Reed W.H., Hill T.R., 1973. Triangular mesh methods for neutron transport equation. Los Alamos Scientific Laboratory Report LA-UR-73-479, Los Alamos, NM.

Reed W.H., Hill T.R., Brinkley F.W., Lathrop K.D., 1973. TRIPLET: a two-dimensional, multigroup triangular, planar geometry, explicit transport code, Los Alamos Scientific Laboratory Report LA-5428-MS, Los Alamos, NM.

Richter, Gerard R., 1988. An optimal-order error estimate for the discontinuous Galerkin method. Math. Comput. 50 (181), 75–88.

Seed T.J., Miller W.F., Brinkley F.W., 1977. TRIDENT: a two-dimensional, multigroup triangular discrete ordinates, explicit neutron transport code. Los Alamos Scientific Laboratory Report LA-6735-M, Los Alamos, NM.

Seed T.J., Miller W.F., Bosler G.E., 1978. TRIDENT: a new triangular mesh discrete ordinates code. Topical Meeting on Advances in Reactor Physics, April 9–12, Gatlinburg, TN.

Shewchuk, Jonathan Richard, 1996. Triangle: engineering a 2D quality mesh generator and Delaunay triangulator. In: Lin, Ming C., Manocha, Dinesh (Eds.), Applied Computational Geometry: Towards Geometric Engineering, Lecture Notes in Computer Science, vol. 1148. Springer-Verlag, Berlin, pp. 203–222 (From the First ACM Workshop on Applied Computational Geometry).

Smith, M., Lewis, E., Na, B., 2004. Benchmark on deterministic 2D MOX fuel assembly transport calculations without spatial homogenization. Prog. Nucl. Energy 45 (2–4), 107–118.

Solin, P., 2005. Partial Differential Equations and the Finite Element Method. JohnWiley & Sons Incorporated.

Takeda, T., Ikeda, H., 1991. 3-D neutron transport benchmarks. OECD/NEA Committee on Reactor Physics. NEACRP-L-330 (March).

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