

# COMPARING THE DISCONTINUOUS GALERKIN AND HIGH-ORDER DIAMOND DIFFERENCING METHODS FOR THE TRANSPORT EQUATION ON A LOZENGE-BASED HEXAGONAL GEOMETRY

Atyab A. Calloo<sup>1</sup>, Alain Hébert<sup>1</sup>

<sup>1</sup>Polytechnique Montréal

2500, chemin de Polytechnique, Montréal, QC H3T 1J4, Canada

atyab.calloo@polymtl.ca, alain.hebert@polymtl.ca

## ABSTRACT

This paper presents an implementation and a comparison of two spatial discretisation schemes over a hexagonal geometry for the two-dimensional discrete ordinates transport equation. The methods are a high-order Discontinuous Galerkin (DG) finite element scheme and a high-order Diamond Differencing (DD) scheme. The DG method has been, and is being, studied on the hexagonal geometry, also called a honeycomb mesh – but not the DD method. In this research effort, it was chosen to divide the hexagons into (at least) three lozenges. An affine transformation is then applied onto said lozenges to cast them into the reference quadrilaterals usually studied in finite elements. In practice, this effectively means that the equations used in Cartesian geometry have their terms and operators altered using the Jacobian matrix of the transformation. This was implemented in the discrete ordinates solver of the code DRAGON5. Two 2D benchmark problems were then used for the verification and validation, including one based on the Monju 3D reactor benchmark. It was found that the diamond-differencing scheme seemed better. It converged much faster towards the solution at comparable mesh refinements for first-order expansion of the flux. Even if this difference was not present for second-order, DG was slower, about two to four times slower.

KEYWORDS: reactor physics, DG method, diamond difference, hexagonal geometry, affine transformation

## 1. INTRODUCTION

There are several ways of dealing with this geometry. The hexagon can be kept whole and a Discontinuous Galerkin (DG) formulation using fairly exotic basis functions can be built upon it. Or the hexagon can be sub-divided into six triangular or three lozenge elements - which can then be further refined. The two methods have been respectively implemented in the French DG solvers MINARET and ERANOS [1]. However, neither triangular nor lozenge sub-elements of a hexagonal mesh have been investigated for DD. In addition, there does not exist a comparison of the two schemes on hexagons for eigenvalue problems. This paper addresses these two issues.

It is of note that this work has previously been done in Cartesian geometries [2], where it was shown that at low order, DD performs generally better than DG for comparable mesh refinement. Accuracy is comparable at second-order but DD is generally much faster. Beyond posterity's

sake, it is quite interesting to compare these two methods since they appear to share quite a few similarities in terms of their formulation and implementation. Because of the upwind nature of the scheme, the problem can be solved locally, i.e., on each unit element – where the matrix to be solved is orders of magnitude smaller, albeit full. However, DG is less demanding on the regularity of the solution because of discontinuities at the boundaries.

For this work, it was chosen to divide the hexagons into lozenges. This brings about a certain elegance and ease of implementation. Indeed, an affine transformation is applied on the lozenges to cast them into reference quadrilaterals (usually squares), leading to a Cartesian reference geometry. In fact, this corresponds to transforming the terms and operators of the transport equation using the Jacobian matrix of the transformation. The fundamental DD and DG equations are not changed substantially, which therefore facilitates implementation. This work is being done in the `SNT` : solver of the `DRAGON5` code - an open-source project from Polytechnique Montreal.

## 2. THEORETICAL BACKGROUND

Only the salient points of the theory behind this study will be detailed in this section. The high-order Diamond Difference (DD) and Discontinuous Galerkin (DG) finite element methods have both been covered at length in the literature. The authors invites the reader to peruse [1], [2], [3], and [4], and the references therein.

### 2.1. The Discrete Ordinates Transport Equation

In this angular discretisation method, the angular integral is approximated using a quadrature defined over the unit sphere, with points and weights respectively denoted by  $\Omega_n$  and  $\omega_n$ .

For each point or characteristic of neutron travel,  $\Omega_n$ , the one-speed stationary first-order form of the discrete ordinates transport equation is given by the set of linear hyperbolic equations,  $\forall \mathbf{r} \in \mathcal{D}$ ,

$$\Omega_n \cdot \vec{\nabla} \Phi_n(\mathbf{r}) + \Sigma(\vec{r}) \Phi_n(\mathbf{r}) = Q_n(\mathbf{r}), \quad (1)$$

where  $\mathcal{D}$  is the domain, with boundary  $\Gamma$ , defined in  $\mathbb{R}^2$ .  $\Sigma$  is the macroscopic total cross-section,  $\Phi_n(\mathbf{r}) = \Phi(\mathbf{r}, \Omega)$  is the neutron flux along that characteristic, and  $Q_n(\mathbf{r}) = Q(\mathbf{r}, \Omega)$  is the source consisting of fission and external sources as well neutrons scattering from other directions. The boundary condition is given by,  $\forall \mathbf{r} \in \Gamma_-$ ,

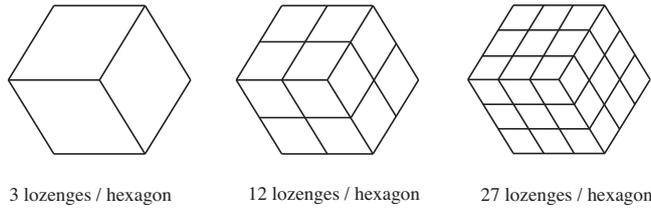
$$\Phi_n(\mathbf{r}) = \phi_n^{BC}(\mathbf{r}), \quad (2)$$

where  $\Gamma_-$  is the inflow flow boundary of  $\mathcal{D}$ , defined by  $\Gamma_- = \{\mathbf{r} \in \Gamma : \Omega_n \cdot \mathbf{n}(\mathbf{r}) < 0\}$ , with  $\mathbf{n}$  being the unit outward normal to  $\Gamma$ .

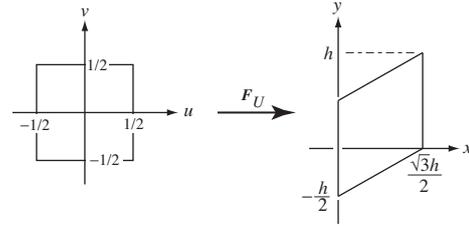
Considering a meshing  $\mathcal{M}_h$  of the domain  $\mathcal{D}$  into elements  $K$  such that  $\bigcup_{K \in \mathcal{M}_h} K = \mathcal{D}$ , the local variational form for the DG method can be expressed by

$$\int_K \left( \Omega_n \cdot \nabla \phi_{n,h} + \Sigma \phi_{n,h} \right) \psi_h d^2r = \int_K Q_n \psi_h d^2r + \int_{\partial K_-} \left( \mathbf{n} \cdot \Omega \right) \left[ \phi_{n,h}^+ - \phi_{n,h}^- \right] \psi_h^+ dr, \quad (3)$$

on an element  $K$ , where the dependency over  $\mathbf{r}$  has been made implicit, and the  $h$  subscript denotes the fact that these functions are obtained from a broken Sobolev space. The  $+$  superscript sign



**Figure 1: Lozenge submesh and its associated refinement in the hexagonal geometry.**



**Figure 2: Affine transformation for one of the lozenges making up the hexagon.**

indicates the restriction of the function taken from within the element  $K$ , while the  $-$  superscript sign the restriction of the function taken from the neighbouring element to  $K$ . Also,  $\partial K_-$  is defined similarly to  $\Gamma_-$ . It should also be noted that the starting point of the DD method is the same as this variational form minus the jump term.

### 2.2. Handling of the Hexagonal Geometry

As mentioned in the introduction, the hexagon is split into (at least) three lozenges. These individual lozenges can then be further sub-meshed, as shown in Fig. 1. These lozenges can then be mapped onto reference quadrilateral elements (denoted by  $\hat{K}$ ) by using a transformation  $F_U$ , as shown in Fig. 2, where  $U$  denotes the specific lozenge. The associated Jacobian matrix is given

by  $\mathbb{J}_{K,U} = \left( \frac{\partial x_i}{\partial u_j} \right)_{i,j}$ , with a determinant  $|\mathbb{J}_{K,U}|$ . Since the basis functions used in the DG and DD

formulations are defined on  $\hat{K}$ , Eq. (3) has to be solved on  $\hat{K}$ . Hence, using the Jacobian matrix previously defined, the following equation is obtained

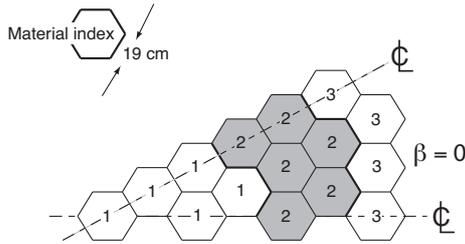
$$\int_{\hat{K}} \left( \boldsymbol{\Omega}_n \cdot (\mathbb{J}_{K,U}^{-T} \nabla \phi_{n,h}) + \Sigma \phi_{n,h} \right) \psi_h |\mathbb{J}_{K,U}| d^2 r = \int_{\hat{K}} Q_n \psi_h |\mathbb{J}_{K,U}| d^2 r + \int_{\partial \hat{K}_-} (\mathbf{n} \cdot \boldsymbol{\Omega}) [\phi_{n,h}^+ - \phi_{n,h}^-] \psi_h^+ \|\mathbb{J}_{K,U} \mathbf{t}\| dr, \quad (4)$$

where  $\|\cdot\|$  is the Euclidean norm in  $\mathbb{R}^2$  and  $\mathbf{t}$  is the unit tangent vector to the inflow boundary,  $\partial \hat{K}_-$ . Of particular note here is that while the Jacobian matrix is different for each of the 3 types of lozenges, the determinant and  $\|\mathbb{J}_{K,U} \mathbf{t}\|$  are in fact the same for all of them. This greatly helps with the implementation.

## 3. NUMERICAL RESULTS

Using the equations obtained in the preceding section, the high-order Diamond Differencing (DD) method and the Discontinuous Galerkin (DG) finite element method were implemented in the DRAGON5 code [5]. The implementation was validated using a simple 2D one-group problem taken from [6]. A more complicated 2D reactor mock-up four-group (without up-scattering) test case was then computed.

The results presented in the following subsections are  $k_{eff}$  values and errors on the absorption rate



**Figure 3: Domain of the 2D one-group benchmark, featuring a 30° symmetry, the material mixtures, and vacuum boundary conditions. Hexagon side is 19 cm.**

Mix.	$\Sigma$ (cm <sup>-1</sup> )	$\Sigma_{s0}$ (cm <sup>-1</sup> )	$\Sigma_{s1}$ (cm <sup>-1</sup> )	$\nu\Sigma_f$ (cm <sup>-1</sup> )
1	0.025	0.013	0.0	0.0155
2	0.025	0.024	0.006	0.0
3	0.075	0.0	0.0	0.0

**Table 1: Cross-section data for the 2D one-group benchmark shown in Fig. 3. Note the anisotropy in mixture 2.**

distribution. The latter is calculated using

$$R_{a,i} = \frac{1}{V_i} \int_{V_i} d^3r [\Sigma(\mathbf{r}) - \Sigma_s(\mathbf{r})] \phi(\mathbf{r}) , \quad (5)$$

where  $i$  denotes the cell element of calculation, and  $V_i$  its volume. The maximum and average errors are then respectively worked out on these reaction rate values using the equations below:

$$\epsilon_{\max} = \max_i \left\{ \frac{|R_{a,i} - R_{a,i}^*|}{R_{a,i}^*} \right\} \text{ and } \bar{\epsilon} = \frac{1}{V_{\text{core}}} \sum_i V_i \frac{|R_{a,i} - R_{a,i}^*|}{R_{a,i}^*} . \quad (6)$$

where the asterisk denotes the reference value, and  $V_{\text{core}}$  is the volume of the whole domain.

Various parameters are investigated for the results, namely, the

- angular discretisation order,  $N$ , which corresponds to  $\frac{N(N+2)}{2}$  allowed directions in 2D.
- spatial discretisation order, with values of 0, 1, 2, and 3, respectively corresponding to flat, linear, parabolic and cubic expansions of the polynomials used to approximate the flux in the discretisation method. DD does not have cubic expansions, and DG does not have flat expansion.
- mesh size. This can be 3, 12, or 27 lozenges per hexagon, as represented in Fig. 1.

Also, no DSA or GMRES(m) acceleration are available for the DG method as of the time of writing this paper. Hence, only a Livolant acceleration was used for the calculations for both methods. Finally, Level-Symmetric (LS) quadratures were used in all computations, and the power iteration is stopped when the maximum relative discrepancy between flux unknowns is less than  $10^{-6}$ .

### 3.1. One-Group Benchmark

The domain is described in Fig. 3, with the cross sections for the different material mixtures given in Table 1. It should be noted that the geometry is first unfolded in the solver before computation.

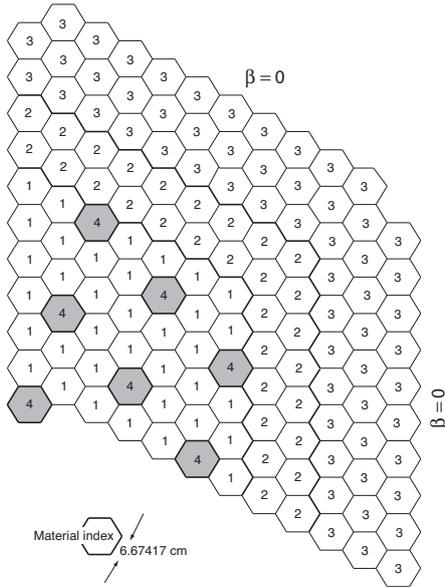
This benchmark is not at all characteristic of a real world reactor. However, transport and anisotropy effects are greatly emphasised over the whole domain and at the void boundary. This allows for a numerical verification and validation of the method, and its implementation. It was obtained from

**Table 2: Results for the 2D one-group benchmark.  $O$ . denotes the polynomial Order, and  $L$ . the number of Lozenges per hexagon. Three significant figures are given, except for  $k_{\text{eff}}$ .**

$S_N$	$O$ .	$L$ .	DD					DG					
			$k_{\text{eff}}$	$\Delta k_{\text{eff}}$ (pcm)	$\epsilon_{\text{max}}$ (%)	$\bar{\epsilon}$ (%)	Time (s)	$k_{\text{eff}}$	$\Delta k_{\text{eff}}$ (pcm)	$\epsilon_{\text{max}}$ (%)	$\bar{\epsilon}$ (%)	Time (s)	
4	0	3	1.001077	-126	7.73	1.77	0.047	-	-	-	-	-	
		12	1.001900	-43.6	4.35	0.740	0.157	-	-	-	-	-	
		27	1.002047	-29.0	3.53	0.571	0.331	-	-	-	-	-	
	1	3	1.002178	-15.9	3.99	0.726	0.140	1.001961	-37.6	3.52	0.681	0.231	
		12	1.002180	-15.7	2.78	0.451	0.556	1.002147	-19.0	3.28	0.478	0.897	
		27	1.002177	-15.9	3.14	0.483	1.180	1.002168	-16.9	3.22	0.483	1.95	
	2	3	1.002171	-16.6	2.75	0.466	0.561	1.002178	-15.9	3.33	0.495	0.743	
		12	1.002178	-15.9	3.14	0.484	2.29	1.002178	-15.9	3.13	0.481	3.06	
		27	1.002179	-15.8	3.10	0.479	4.98	1.002179	-15.8	3.11	0.479	6.84	
3	3	-	-	-	-	-	1.002177	-16.0	3.12	0.482	2.42		
	12	-	-	-	-	-	1.002178	-15.8	3.10	0.479	9.81		
	27	-	-	-	-	-	1.002179	-15.8	3.10	0.479	21.9		
10	0	3	1.001208	-113	5.70	1.33	0.223	-	-	-	-	-	
		12	1.002047	-29.0	0.911	0.282	0.755	-	-	-	-	-	
		27	1.002190	-14.7	0.556	0.132	1.60	-	-	-	-	-	
	1	3	1.002321	-1.60	1.29	0.355	0.703	1.002095	-24.2	1.68	0.484	1.12	
		12	1.002315	-2.20	0.173	0.0393	2.76	1.002281	-5.60	0.342	0.101	4.34	
		27	1.002318	-1.90	0.138	0.0363	5.82	1.002304	-3.30	0.167	0.0473	9.31	
	2	3	1.002310	-2.70	0.201	0.0504	2.96	1.002314	-2.30	0.214	0.0450	3.83	
		12	1.002318	-1.90	0.128	0.0353	11.4	1.002315	-2.20	0.124	0.0361	18.1	
		27	1.002317	-1.90	0.130	0.0357	25.3	1.002317	-2.00	0.131	0.0369	33.6	
	3	3	-	-	-	-	-	1.002313	-2.30	0.119	0.0362	12.5	
		12	-	-	-	-	-	1.002317	-2.00	0.134	0.0370	49.4	
		27	-	-	-	-	-	1.002317	-1.90	0.135	0.0372	113	
	18	0	3	1.001218	-112	5.87	1.28	0.664	-	-	-	-	-
			12	1.002060	-27.7	0.854	0.285	2.26	-	-	-	-	-
			27	1.002211	-12.6	0.455	0.127	5.3	-	-	-	-	-
1		3	1.002337	0.0	1.17	0.360	2.24	1.002114	-22.3	1.67	0.461	3.33	
		12	1.002338	0.1	0.0491	0.011	8.52	1.002301	-3.5	0.292	0.0838	12.8	
		27	1.002336	-0.1	0.0148	3.78E-3	17.9	1.002325	-1.2	0.103	0.0291	28.4	
2		3	1.002334	-0.3	0.0532	0.0111	9.01	1.002335	-0.2	0.128	0.0259	11.4	
		12	1.002336	-0.1	0.0184	3.61E-3	35.0	1.002336	-0.1	0.0216	2.57E-3	47.3	
		27	1.002337	0.0	4.24E-3	6.43E-4	76.8	1.002337	0.0	0.0128	1.32E-3	103	
3		3	-	-	-	-	-	1.002335	-0.2	0.026	5.81E-3	38.1	
		12	-	-	-	-	-	1.002337	0.0	7.92E-3	8.46E-4	156	
		27	-	-	-	-	-	1.002337		Reference		348	

the paper [6], which studied the implementation of an  $SP_n$  solver, and contrasted it with another discrete-ordinates solver, ERANOS/SNATCH – both used a lozenge submeshing of the geometry.

The asymptotic “reference” value for the ERANOS/SNATCH solver was 1.002313. This was obtained using a Gauss-Chebyshev quadrature allowing for 180 directions, a DG method with a flux



**Figure 4: Domain of the Monju-2D benchmark. Hexagon side is 6.67417 cm.**

M.	G.	$\Sigma^g$ (cm <sup>-1</sup> )	$\Sigma_{s0}^{g \rightarrow g}$ (cm <sup>-1</sup> )	$\Sigma_{s0}^{g \rightarrow g+1}$ (cm <sup>-1</sup> )	$\Sigma_{s0}^{g \rightarrow g+2}$ (cm <sup>-1</sup> )	$\nu \Sigma_f^g$ (cm <sup>-1</sup> )
1	1	1.31234E-1	1.00247E-1	2.544E-2	5.625E-4	1.235E-2
	2	1.93349E-1	1.83859E-1	6.551E-3	-	5.225E-3
	3	2.63713E-1	2.56380E-1	-	-	7.684E-3
2	1	1.30822E-1	9.96078E-2	2.497E-2	5.548E-4	1.467E-2
	2	1.93237E-1	1.83362E-1	6.341E-3	-	6.955E-3
	3	2.62674E-1	2.54575E-1	-	-	9.986E-3
3	1	1.53398E-1	1.15467E-1	3.288E-2	7.468E-4	8.631E-3
	2	2.31642E-1	2.19799E-1	1.000E-2	-	5.995E-4
	3	3.24886E-1	3.17275E-1	-	-	1.381E-3
4	1	1.33333E-1	1.10053E-1	2.185E-2	2.163E-4	0.0
	2	1.98295E-1	1.85568E-1	9.379E-3	-	0.0
	3	2.62674E-1	2.47704E-1	-	-	0.0

**Table 3: Cross section data for the Monju-2D benchmark shown in Fig. 4. M. denotes the material Mixture, and G. the energy Group.**

expansion of parabolic order using a Legendre hierarchical basis, and 27 lozenges per hexagons. The results from the current implementation in the SNT : solver are printed in Table 2, and the obtained asymptotic “reference” value is 1.002337, a margin of 2.4 pcm. An  $S_{18}$  also allows for 180 directions; therefore, the differences are the choice of basis and the order of expansion. Taking that into consideration, it can be said that convergence towards the same solution has been achieved – and the solver validated.

### 3.2. The Monju-2D Reactor Benchmark

The Monju-2D reactor benchmark was formulated by selecting the second axial plane from the Monju 3D reactor benchmark. The latter is a three-group Fast Breeder Reactor (FBR) mockup proposed by Komano *et al.* [7]. The description of the domain considered in this paper is given in Fig. 4 and the material cross sections in Table 3. Mixtures 1, 2, 3 and 4 respectively correspond to the active inner core, active outer core, blanket and control rod. Also, again, while the domain features a 120° symmetry, it is first unfolded before computation. The results are printed in Table 4.

### 3.3. Discussion of Results

From both Table 2 and Table 4, looking at each method independently, it can be seen that they perform mostly as expected with each varying parameter. There are a few inconsistencies. For example, for DD, when looking at low angular orders, especially for the linear order, the error on the  $k_{eff}$  can be seen to *increase* as the mesh is refined. This can be attributed to spatial and polynomial convergence being reached at that point, but angular order,  $N$ , still being the limiting factor. Indeed, as  $N$  is increased, this disappears.

**Table 4: Results for the Monju-2D benchmark.  $O$ . denotes the polynomial Order, and  $L$ . the number of Lozenges per hexagon. Three significant figures are given, except for  $k_{\text{eff}}$ .**

$S_N$	$O$ .	$L$ .	DD					DG					
			$k_{\text{eff}}$	$\Delta k_{\text{eff}}$ (pcm)	$\epsilon_{\text{max}}$ (%)	$\bar{\epsilon}$ (%)	Time (s)	$k_{\text{eff}}$	$\Delta k_{\text{eff}}$ (pcm)	$\epsilon_{\text{max}}$ (%)	$\bar{\epsilon}$ (%)	Time (s)	
4	0	3	1.157553	-165	3.82	1.06	1.45	-	-	-	-	-	
		12	1.158748	-45.2	0.669	0.203	4.35	-	-	-	-	-	
		27	1.158951	-25.0	0.500	0.0945	9.11	-	-	-	-	-	
	1	3	1.159169	-3.1	0.633	0.0602	5.00	1.158376	-82.4	0.618	0.120	16.1	
		12	1.159127	-7.3	0.648	0.0503	16.4	1.159000	-20.0	0.737	0.0649	60.5	
		27	1.159122	-7.8	0.679	0.0564	33.4	1.159082	-11.8	0.716	0.0607	127	
	2	3	1.159127	-7.3	0.554	0.0462	17.9	1.159118	-8.2	0.769	0.0664	60.4	
		12	1.159124	-7.6	0.673	0.0557	71.6	1.159124	-7.7	0.687	0.0577	238	
		27	1.159122	-7.8	0.667	0.0555	159	1.159124	-7.6	0.678	0.0566	532	
	3	3	-	-	-	-	-	1.159123	-7.7	0.691	0.0571	202	
		12	-	-	-	-	-	1.159126	-7.4	0.677	0.0564	822	
		27	-	-	-	-	-	1.159126	-7.5	0.671	0.0558	1850	
	10	0	3	1.157634	-157	3.59	1.06	9.03	-	-	-	-	-
			12	1.158824	-37.6	0.601	0.21	29.5	-	-	-	-	-
			27	1.159024	-17.7	0.244	0.0884	64.1	-	-	-	-	-
1		3	1.159237	3.7	0.069	0.0142	25.8	1.158456	-74.4	0.326	0.114	78.6	
		12	1.159196	-0.4	0.053	3.76E-3	106	1.159073	-12.7	0.119	0.0219	295	
		27	1.159195	-0.5	0.073	4.04E-3	233	1.159153	-4.7	0.108	0.0114	637	
2		3	1.159196	-0.4	0.106	0.0146	120	1.159188	-1.2	0.147	0.0176	303	
		12	1.159194	-0.6	0.0702	3.51E-3	504	1.159192	-0.8	0.101	8.19E-3	1220	
		27	1.159194	-0.6	0.0765	4.86E-3	1140	1.159194	-0.6	0.0968	7.45E-3	2680	
3		3	-	-	-	-	-	1.159192	-0.8	0.0938	7.77E-3	1050	
		12	-	-	-	-	-	1.159194	-0.6	0.0947	7.02E-3	4270	
		27	-	-	-	-	-	1.159194	-0.6	0.0917	6.85E-3	9520	
18		0	3	1.157632	-157	3.59	1.06	31.7	-	-	-	-	-
			12	1.158829	-37.1	0.624	0.218	105	-	-	-	-	-
			27	1.159033	-16.7	0.287	0.0990	224	-	-	-	-	-
	1	3	1.159245	4.5	0.0854	0.0186	90.2	1.158465	-73.5	0.322	0.118	237	
		12	1.159204	0.4	0.0563	8.35E-3	368	1.159079	-12.1	0.0595	0.0180	884	
		27	1.159201	0.1	0.0181	2.38E-3	802	1.159160	-4.1	0.0317	7.18E-3	1900	
	2	3	1.159207	0.7	0.142	0.0251	413	1.159194	-0.6	0.0733	0.0113	932	
		12	1.159201	0.1	0.0227	2.89E-3	1770	1.159199	-0.1	0.0119	1.58E-3	3640	
		27	1.159201	0.1	0.0118	1.52E-3	3890	1.159200	0.0	6.59E-3	8.79E-4	8160	
	3	3	-	-	-	-	-	1.159198	-0.2	0.0152	1.12E-3	3190	
		12	-	-	-	-	-	1.159199	-0.1	3.75E-3	5.45E-4	13000	
		27	-	-	-	-	-	1.159200	Reference			29600	

Comparing the two methods is more interesting. Even if DD was better in Cartesian, there was still some expectation among the authors that DG might be more precise and faster than DD on hexagons. However, the results indicate quite the contrary still. At comparable orders, for lower lozenge submeshes, DD does generally better, while DG is persistently *slower* – about  $1.3\times$  to  $1.5\times$  slower for the one-group benchmark and about  $3\times$  to  $4\times$  slower for Monju. This is consistent with what was observed for the Cartesian geometry [2], albeit to a stronger effect. While it could

be ascribed to the source being slightly more complex to build for DG (because of the upwind sources), it is not only that. DG just requires more inner iterations to converge. This has not been illustrated here for lack of space, but it is consistently the case. More investigation is required into their behaviour with preconditioning and a GMRES(m) acceleration.

#### 4. CONCLUSIONS

High-order Discontinuous Galerkin (DG) and high-order Diamond Differencing (DD) schemes for the discrete ordinates transport equation were implemented for hexagonal geometries. This was achieved by dividing the hexagons into lozenges, and transforming them into squares. While it had been done before for DG – but not for DD, the ease of implementation has been more formally shown. This had been lacking in previous literature. The methods were then compared on two eigenvalue problems. It was shown that at low order, DD has smaller margins of error at lower mesh-refinements. It is also consistently significantly faster. In the future, more work still needs to be done with acceleration techniques such as the DSA for better evaluation.

#### ACKNOWLEDGEMENTS

The authors would like to acknowledge the help of Dr. Romain Le Tellier from the CEA Cadarache for the insightful discussions. This work was also supported by a grant from the Natural Sciences and Engineering Research Council of Canada (NSERC).

#### REFERENCES

- [1] R. Le Tellier, C. Suteau, D. Fournier, and J. M. Ruggieri. “High-order discrete ordinate transport in hexagonal geometry: a new capability in ERANOS.” *Il Nuovo cimento della Società italiana di fisica C*, **volume 33**(1), p. 121 (2010).
- [2] A. Calloo and A. Hébert. “Development and Evaluation of a Discontinuous Galerkin Method in the DRAGON5 Code.” In *38th Annual Conference of the Canadian Nuclear Society and 42nd Annual CNS/CNA Student Conference*. Canadian Nuclear Society, Saskatoon, SK, Canada (2018).
- [3] W. H. Reed and T. R. Hill. “Triangular mesh methods for the neutron transport for a scalar hyperbolic equation.” Technical Report LA-UR-73-479, Los Alamos Scientific Lab., N. Mex. (1973).
- [4] A. Hébert. “High order diamond differencing schemes.” *Annals of Nuclear Energy*, **volume 33**(17), pp. 1479–1488 (2006).
- [5] G. Marleau, A. Hébert, and R. Roy. “New computational methods used in the lattice code DRAGON.” In *Topical Meeting on Advances in Reactor Physics*, volume 1, pp. 8–11 (1992).
- [6] A. Hébert. “A Raviart-Thomas-Schneider implementation of the simplified Pn method in 3-D hexagonal geometry.” In *International Conference on the Physics of Reactor, PHYSOR*. American Nuclear Society, Pittsburgh, Pennsylvania, USA (2010).
- [7] Y. Komano, T. Takeda, and T. Sekiya. “Improved few-group coarse-mesh method for calculating three-dimensional power distribution in fast breeder reactor.” Technical Report NEACRP-L-204 (1978).