0.1 DG Tests

0.1.1 The Aim

We will conduct tests on a 2D code that uses discontinuous Galerkin finite elements to solve the neutron transport equation. The equation that will be solved is given by

$$\Omega \cdot \nabla\psi(\mathbf{r}, \Omega) + \sigma_T(\mathbf{r})\psi(\mathbf{r}, \Omega) = \sigma_S(\mathbf{r})\phi(\mathbf{r}) + Q(\mathbf{r}), \tag{1}$$

with $\mathbf{r} \in V \subset \mathbb{R}^2$ and $\Omega \in \mathbb{S}^1$, the unit sphere with radius 1 centred at the origin. The so-called *neutron flux*, ψ, is subject to the incoming boundary condition

$$\psi(\mathbf{r}) = g(\mathbf{r}), \quad \text{if} \quad \Omega \cdot n(\mathbf{r}) < 0, \forall \mathbf{r} \in \delta V. \tag{2}$$

The value, ϕ, is called the *scalar flux* and is defined to be

$$\phi(\mathbf{r}) \equiv \frac{1}{2\pi}\int_{\mathbb{S}^1} \psi(\mathbf{r}, \Omega)\, \mathrm{d}\Omega. \tag{3}$$

To discretise in space, the code uses a mesh with the structure given in figure 0-1 (where M_x and M_y are also defined). We will always take $V \equiv [0,1] \times [0,1]$, but will vary M_x, M_y, σ_S and σ_T.

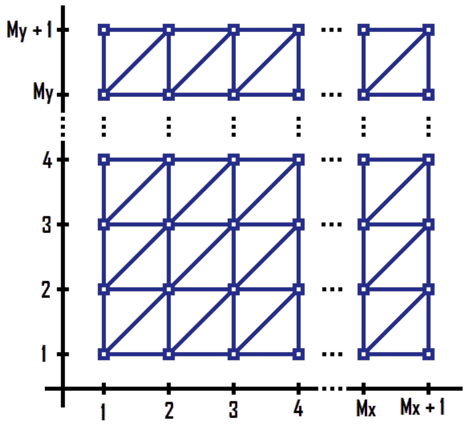

Figure 0-1: *Mesh structure, and definitions of M_x and M_y.*

The code uses a discrete ordinates discretisation for N different angles, chosen via the quadrature rule

$$\Omega \in \left\{ \left(\cos\frac{2\pi j}{N}, \sin\frac{2\pi j}{N} \right) \right\}_{j=1}^{N}. \tag{4}$$

with weights $\omega = 2\pi/N$ for each angle.

Whenever we talk about the *error* in an approximate solution we mean the L^2-norm of the difference between the approximate and true solutions. This is calculated using a quadrature rule as follows. We denote by κ_i the ith element of the mesh, and define $|\kappa_i|$ to be the area of the ith element. Also, let m_1^i, m_2^i and m_3^i be the three midpoints of the edges of κ_i (see figure 0-2), and use ϕ_A and ϕ_T to denote the approximate and true solutions respectively. Then we calculate the error as follows

$$\|\phi_T - \phi_A\|^2_{L^2(V)} \equiv \sum_{i=1}^{2M_xM_y} \frac{|\kappa_i|}{3} \sum_{j=1}^{3} \left(\phi_T(m_j^i) - \phi_A(m_j^i)\right)^2 \tag{5}$$

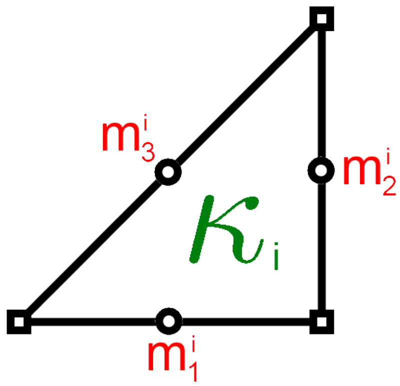

Figure 0-2: *Mipoints, m_1^i, m_2^i and m_3^i for a standard mesh element, κ_i.*

The code solves for the solution by forming a full matrix vector system, representing the full transport equation, and then solving using Matlab's inbuilt 'backslash' solver. As a partial explanation, we are solving

$$\int_\kappa (\Omega\cdot\nabla\psi_h)v_h \,\mathrm{d}\mathbf{r} + \sigma_{T,\kappa}\int_\kappa \psi_h v_h \,\mathrm{d}\mathbf{r} - \int_{\delta\kappa}(\Omega\cdot n(\mathbf{r}))(\psi_h^+ - \psi_h^-)v_h \,\mathrm{d}\mathbf{r} - \sigma_{S,\kappa}\int_\kappa \phi_h v_h \,\mathrm{d}\mathbf{r} = \int_\kappa Q_h v_h \,\mathrm{d}\mathbf{r}, \tag{6}$$

and build a matrix-vector system of the form

$$\begin{bmatrix} T_1 & & & | & -\Sigma_1 \\ & \ddots & & | & \vdots \\ & & T_N & | & -\Sigma_N \\ -- & -- & -- & | & -- \\ -W_1 & \cdots & -W_N & | & I \end{bmatrix} \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_N \\ -- \\ \phi \end{bmatrix} = \begin{bmatrix} Q - F_1 \\ \vdots \\ Q - F_N \\ --- \\ 0 \end{bmatrix}. \tag{7}$$

In this: the T-blocks contain the '∇' and 'σ_T' integrals from (6), as well as the internal element boundary integrals; the 'Σ'-blocks contain the 'σ_S' integral; the 'Q'-blocks contain the 'Q' integral; and the 'F'-blocks contain the part of the element boundary integrals that lie on the domain boundary. For the bottom row, the 'W'-blocks contain quadrature weights scaled by $1/2\pi$, and that line imposes the relationship between ψ and ϕ.

0.1.2 Test 1

In this test we see if the code can produce the correct solution in the case where

$$\psi(\mathbf{r}, \Omega) = \mathbf{r} \cdot \Omega. \tag{8}$$

In this case,

$$\begin{aligned} \phi(\mathbf{r}) &= \tfrac{1}{2\pi} \int_{\mathbb{S}^1} \mathbf{r} \cdot \Omega \, \mathrm{d}\Omega \\ &= \tfrac{1}{2\pi} \int_{[0,2\pi]} x \cos(\theta) + y \sin(\theta) \, \mathrm{d}\theta \\ &= 0, \end{aligned} \tag{9}$$

and so

$$Q(\mathbf{r}) = \Omega \cdot \Omega + \sigma_T(\mathbf{r}) \mathbf{r} \cdot \Omega. \tag{10}$$

Using this source, we solved the transport equation (with constant cross-sections) for varying mesh resolutions, and tabulated the error in each case. These results are in the table below, and show that the code finds the exact solution to machine precision.

h	N	$\lVert\phi_T - \phi_A\rVert_{L^2(V)}$
1/2	12	6e-017
1/4	16	5e-017
1/8	23	5e-017
1/16	32	6e-017

Table 1

0.1.3 Test 2

Johnson and Pitkäranta, 1983, provide an error estimate for the solution to the neutron transport equation in terms of the number of discrete ordinates used, N, and the mesh width, h. This estimate is in remark would like to verify the error estimate given in Remark 5.1, and is as follows

$$\left\lVert \phi - \phi_N^h \right\rVert_{L^2(V)} \leq C(N^{-1} + \sqrt{h})(\lVert\phi\rVert_1 + \lVert Q\rVert_1). \tag{11}$$

Here ϕ_N^h is the solution obtained via DG with discrete ordinates, using N angles and spatial mesh width h. The value C is a constant and $\lVert\cdot\rVert_1$ denotes the H^1-norm.

In this test we want to verify that the estimate is satisfied by our code. We chose a model solution of $\psi(\mathbf{r}) = \mathbf{r} \cdot \mathbf{r}$ which implies

$$Q(\mathbf{r}, \Omega) = 2\Omega \cdot \mathbf{r} + \sigma_A(\mathbf{r}) \mathbf{r} \cdot \mathbf{r}. \tag{12}$$

To test Johnson and Pitkäranta's estimate we solved the transport equation with this source and with boundary conditions implied by the true solution. We did this for a range of mesh widths, h, and chose N such that N^{-1} and $\sqrt{h}$ varied proportionally. For this we used the relation

$$N = 8\left\lceil \sqrt{1/h} \right\rceil, \tag{13}$$

where the scaling factor of 8 ensured a reasonable number of angles were used for all mesh widths. For each mesh width we measured the error in the solution, and have tabulated these below along with the corresponding values of $N^{-1} + \sqrt{h}$.

h	N	$\|\phi_T - \phi_A\|_{L^2(V)}$	$N^{-1} + \sqrt{h}$
1/2	12	3e-002	8e-001
1/4	16	8e-003	6e-001
1/8	23	2e-003	4e-001
1/16	32	5e-004	3e-001
1/32	46	1e-004	2e-001

Table 2

This does follow the estimate above for some constant C, as can be seen easily by plotting the last two columns, as in the following figure.

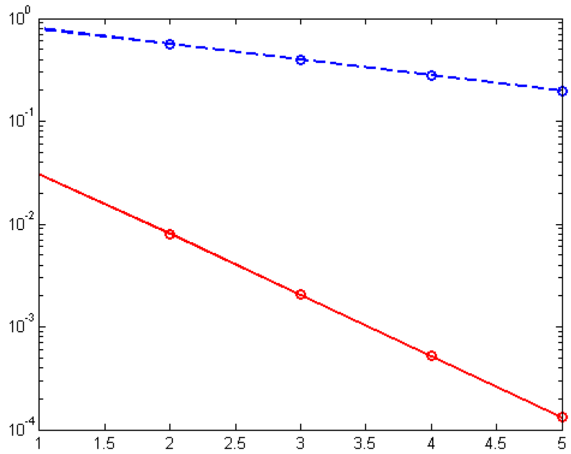

Figure 0-3: *Graph of the Table 2 data. The red continuous line is column 3, the blue dashed line is column 4*