SP_N Equations

Brody Bassett

February 17, 2016

1 Derivation of the Slab-Geometry P_N Equations

1.1 Derivation of the Slab-Geometry P_N Equations

First, define the following:

$$\begin{aligned} \text{natural numbers :} \quad & \mathbb{N} = (0, 1, 2, 3, \dots), \\ \text{odd numbers :} \quad & \mathbb{O} = (1, 3, 5, 7, \dots), \\ \text{even numbers :} \quad & \mathbb{E} = (0, 2, 4, 5, \dots). \end{aligned}$$

The standard first-order multigroup, P_N, slab geometry differential equations are

$$\frac{d}{dx}\left\{\frac{n}{2n+1}\phi_{n-1}(x) + \frac{n+1}{2n+1}\phi_{n+1}(x)\right\} + \Sigma_t(x)\phi_n(x) = \Sigma_{sn}(x)\phi_n(x) + Q(x)\delta_{n,0}, \quad n \in \mathbb{N}. \tag{1.1}$$

Solve Eq. 1.1 for ϕ_n in terms of ϕ_{n-1} and ϕ_{n+1}:

$$\begin{aligned} \phi_n(x) &= \left[\Sigma_t(x) - \Sigma_{sn}(x)\right]^{-1}\left[Q(x)\delta_{n,0} - \frac{d}{dx}\left\{\frac{n}{2n+1}\phi_{n-1}(x) + \frac{n+1}{2n+1}\phi_{n+1}(x)\right\}\right] \\ &= \frac{n}{2n+1}\left[\Sigma_t(x) - \Sigma_{sn}(x)\right]^{-1}\left[Q(x)\delta_{n,0} - \frac{d}{dx}\left\{\phi_{n-1}(x) + \frac{n+1}{n}\phi_{n+1}(x)\right\}\right] \\ &= \left(\frac{2n-1}{n}\right)D_{n-1}(x)\left[Q(x)\delta_{n,0} - \frac{d}{dx}\left\{\phi_{n-1}(x) + \frac{n+1}{n}\phi_{n+1}(x)\right\}\right], \quad n \in \mathbb{N}. \end{aligned} \tag{1.2}$$

Substitute ϕ_{n-1} and ϕ_{n+1} into the previous equation for ϕ_n to develop a relation for ϕ_n in terms of ϕ_{n-2} and ϕ_{n+2} and to eliminate the terms ϕ_n for which $n \in \mathbb{O}$:

$$\begin{aligned} \left[\Sigma_t(x) - \Sigma_{sn}(x)\right]\phi_n(x) =& Q(x)\delta_{n,0} \\ & - \left(\frac{n}{2n+1}\right)\left(\frac{2n-3}{n-1}\right)\frac{d}{dx}\left\{D_{n-2}(x)\left[\cancelto{0}{Q(x)\delta_{n-1,0}} - \frac{d}{dx}\left\{\phi_{n-2}(x) + \frac{n}{n-1}\phi_n(x)\right\}\right]\right\} \\ & - \left(\frac{n}{2n+1}\right)\left(\frac{n+1}{n}\right)\left(\frac{2n+1}{n+1}\right)\frac{d}{dx}\left\{D_n(x)\left[\cancelto{0}{Q(x)\delta_{n+1,0}} - \frac{d}{dx}\left\{\phi_n(x) + \frac{n+2}{n+1}\phi_{n+2}(x)\right\}\right]\right\} \\ =& \left(\frac{n}{2n+1}\right)\left(\frac{2n-3}{n-1}\right)\frac{d}{dx}\left\{D_{n-2}(x)\frac{d}{dx}\left\{\phi_{n-2}(x) + \frac{n}{n-1}\phi_n(x)\right\} - Q(x)\delta_{n-1,0}\right\} \\ & + \frac{d}{dx}\left\{D_n(x)\frac{d}{dx}\left\{\phi_n(x) + \frac{n+2}{n+1}\phi_{n+2}(x)\right\}\right\} + Q(x)\delta_{n,0}, \quad n \in \mathbb{E}. \end{aligned} \tag{1.3}$$

Written in standard form, the equation will read

$$
\begin{aligned}
-\left(\frac{n}{2n+1}\right)\left(\frac{2n-3}{n-1}\right)\frac{d}{dx}\left\{D_{n-2}\left(x\right)\frac{d}{dx}\left\{\phi_{n-2}\left(x\right)+\frac{n}{n-1}\phi_{n}\left(x\right)\right\}\right\} \\
-\frac{d}{dx}\left\{D_{n}\left(x\right)\frac{d}{dx}\left\{\phi_{n}\left(x\right)+\frac{n+2}{n+1}\phi_{n+2}\left(x\right)\right\}\right\}+\left[\Sigma_{t}\left(x\right)-\Sigma_{s,n}\left(x\right)\right]\phi_{n}\left(x\right) \\
=\delta_{n,0}Q\left(x\right), \quad n\in\mathbb{E},
\end{aligned}
\tag{1.4}
$$

or

$$
\begin{aligned}
-\frac{d}{dx}\left\{\left(\frac{n}{2n+1}\right)\left(\frac{2n-3}{n-1}\right)D_{n-2}\left(x\right)\frac{d}{dx}\phi_{n-2}\left(x\right)\right\} \\
-\frac{d}{dx}\left\{\left[\left(\frac{n}{n-1}\right)^{2}\left(\frac{2n-3}{2n+1}\right)D_{n-2}\left(x\right)+D_{n}\left(x\right)\right]\frac{d}{dx}\phi_{n}\left(x\right)\right\} \\
-\frac{d}{dx}\left\{\left(\frac{n+2}{n+1}\right)D_{n}\left(x\right)\frac{d}{dx}\phi_{n+2}\left(x\right)\right\}+\left[\Sigma_{t}\left(x\right)-\Sigma_{s,n}\left(x\right)\right]\phi_{n}\left(x\right) \\
=\delta_{n,0}Q\left(x\right), \quad n\in\mathbb{E}.
\end{aligned}
\tag{1.5}
$$

This form of the equation is solvable by itself, but with derivative terms for the n equation in ϕ_n, ϕ_{n-2} and ϕ_{n+2}. This can be traded for an equation with a derivative in ϕ_n and non-derivative terms in every even ϕ_n. To do this, define

$$
\Phi_{n}\left(x\right)=\phi_{n}\left(x\right)+\frac{n+2}{n+1}\phi_{n+2}\left(x\right)
\tag{1.6}
$$

and use it to simplify Eq. 1.4:

$$
\begin{aligned}
-\left(\frac{n}{2n+1}\right)\left(\frac{2n-3}{n-1}\right)\frac{d}{dx}\left\{D_{n-2}\left(x\right)\frac{d}{dx}\Phi_{n-2}\left(x\right)\right\} \\
-\frac{d}{dx}\left\{D_{n}\left(x\right)\frac{d}{dx}\Phi_{n}\left(x\right)\right\}+\left[\Sigma_{t}\left(x\right)-\Sigma_{s,n}\left(x\right)\right]\phi_{n}\left(x\right) \\
=\delta_{n,0}Q\left(x\right), \quad n\in\mathbb{E}.
\end{aligned}
\tag{1.7}
$$

Solving Eq. 1.6 for $\phi_n\left(x\right)$ allows the development of a recursion relationship,

$$
\begin{aligned}
\phi_{n}\left(x\right)&=\Phi_{n}\left(x\right)-\frac{n+2}{n+1}\phi_{n+2}\left(x\right) \\
&=\Phi_{n}\left(x\right)-\frac{n+2}{n+1}\left(\Phi_{n+2}\left(x\right)-\frac{n+4}{n+3}\psi_{n+4}\right) \\
&=\Phi_{n}\left(x\right)-\frac{n+2}{n+1}\left(\Phi_{n+2}\left(x\right)-\frac{n+4}{n+3}\left(\Phi_{n+4}-\frac{n+6}{n+5}\psi_{n+6}\right)\right) \\
&=\Phi_{n}\left(x\right)+\sum_{m\in\mathbb{E}}^{(n+2,N-1)}\left[\prod_{\ell\in\mathbb{E}}^{(n+2,m)}\left(-\frac{\ell}{\ell-1}\right)\right]\Phi_{m}\left(x\right).
\end{aligned}
\tag{1.8}
$$

With

$$
k_{nm}=\prod_{\ell\in\mathbb{E}}^{(n+2,m)}\left(-\frac{\ell}{\ell-1}\right),
\tag{1.9}
$$

the equation simplifies to

$$\phi_n(x) = \Phi_n(x) + \sum_{m\in\mathbb{E}}^{(n+2,N-1)} k_{nm}\Phi_m(x)\,. \tag{1.10}$$

Define the temporary variables

$$\beta_n(x) = \frac{d}{dx}\left\{D_n(x)\frac{d}{dx}\Phi_n(x)\right\}, \tag{1.11a}$$
$$\alpha_n(x) = \left[\Sigma_t(x) - \Sigma_{s,n}(x)\right]\phi_n(x) - \delta_{n,0}Q(x)\,, \tag{1.11b}$$

solve for $\beta_n(x)$ in the context of Eq. 1.7, and develop another recursion relationship:

$$\begin{aligned}
\beta_n(x) &= \alpha_n(x) - \left(\frac{n}{2n+1}\right)\left(\frac{2n-3}{n-1}\right)\beta_{n-2} \\
&= \alpha_n(x) - \left(\frac{n}{2n+1}\right)\left(\frac{2n-3}{n-1}\right)\left[\alpha_{n-2}(x) - \left(\frac{n-2}{2n-3}\right)\left(\frac{2n-7}{n-3}\right)\beta_{n-4}\right] \\
&= \sum_{n\in\mathbb{E}}^{(0,n)}\left[\prod_{\ell\in\mathbb{E}}^{(0,n)}(-1)\left(\frac{\ell}{2\ell+1}\right)\left(\frac{2\ell-3}{\ell-1}\right)\right]\alpha_n(x) \\
&= \left(\frac{2n+5}{n+2}\right)\left(\frac{n+1}{2n+1}\right)\sum_{\ell\in\mathbb{E}}^{(0,n)} m_\ell\alpha_\ell(x)
\end{aligned} \tag{1.12}$$

with

$$m_n = \left(\frac{n+2}{2n+5}\right)\left(\frac{2n+1}{n+1}\right)\prod_{\ell\in\mathbb{E}}^{(0,n)}(-1)\left(\frac{\ell}{2\ell+1}\right)\left(\frac{2\ell-3}{\ell-1}\right). \tag{1.13}$$

Substitute $\beta_{n-2}(x)$ in Eq. 1.7 to get

$$-\sum_{\ell\in\mathbb{E}}^{(0,n-2)} m_\ell\alpha_\ell(x) - \frac{d}{dx}\left\{D_n(x)\frac{d}{dx}\Phi_n(x)\right\} + \left[\Sigma_t(x) - \Sigma_{s,n}(x)\right]\phi_n(x) = \delta_{n,0}Q(x)\,, \quad n\in\mathbb{E}, \tag{1.14}$$

or

$$-\frac{d}{dx}\left\{D_n(x)\frac{d}{dx}\Phi_n(x)\right\} + \left[\Sigma_t(x) - \Sigma_{s,n}(x)\right]\phi_n(x) = \sum_{\ell\in\mathbb{E}}^{(0,n-2)} m_\ell\alpha_\ell(x) + \delta_{n,0}Q(x)\,, \quad n\in\mathbb{E}. \tag{1.15}$$

Writing out $\alpha_\ell(x)$ explicitly gives

$$\begin{aligned}
-\frac{d}{dx}&\left\{D_n(x)\frac{d}{dx}\Phi_n(x)\right\} + \left[\Sigma_t(x) - \Sigma_{s,n}(x)\right]\phi_n(x) \\
&= \sum_{\ell\in\mathbb{E}}^{(0,n-2)} m_\ell\left(\left[\Sigma_t(x) - \Sigma_{s,\ell}(x)\right]\phi_\ell(x) - \delta_{\ell,0}Q(x)\right) + \delta_{n,0}Q(x)\,, \quad n\in\mathbb{E}.
\end{aligned} \tag{1.16}$$

Use the recursion relationship developed earlier for ϕ_n in Eq. 1.10 to eliminate it from the equation:

$$
\begin{aligned}
-\frac{d}{dx}\left\{D_n(x)\frac{d}{dx}\Phi_n(x)\right\} + \left[\Sigma_t(x) - \Sigma_{s,n}(x)\right]\Phi_n(x) = -\left[\Sigma_t(x) - \Sigma_{s,n}(x)\right] \sum_{\mathbb{E}(\ell)}^{(n+2,N-1)} k_{n\ell}\Phi_\ell(x) \\
+ \sum_{\ell\in\mathbb{E}}^{(0,n-2)} m_\ell \left(\left[\Sigma_t(x) - \Sigma_{s,\ell}(x)\right]\left(\Phi_\ell(x) + \sum_{\mathbb{E}(\ell')}^{(\ell+2,N-1)} k_{\ell\ell'}\Phi_{\ell'}(x)\right) - \delta_{\ell,0}Q(x)\right) \\
+ \delta_{n,0}Q(x), \quad n\in\mathbb{E}.
\end{aligned}
\tag{1.17}
$$

The final sum in Eq. 1.17 could be simplified to get a linear equation for all the Φ_ℓ with spatially-dependent coefficients. Note that the derivation of the boundary conditions is not included here, but for ϕ_n are reproduced from Larsen's notes:

$$
\sum_{m=0}^{N} \frac{2m+1}{2}\left(\int_0^1 \mu P_n(\mu) P_m(\mu)\, d\mu\right)\phi_m(0) = \int_0^1 \mu P_n(\mu)\phi^b(\mu)\, d\mu, \quad n\in\mathbb{E}, \quad x=0, \tag{1.18a}
$$

$$
\sum_{m=0}^{N} \frac{2m+1}{2}\left(\int_{-1}^0 \mu P_n(\mu) P_m(\mu)\, d\mu\right)\psi_m(X) = \int_{-1}^0 \mu P_n(\mu)\phi^b(\mu)\, d\mu, \quad n\in\mathbb{E}, \quad x=X. \tag{1.18b}
$$

1.2 Simplified Forms of the Slab-Geometry P_N Equations

For $N=1$, the equation simplies to

$$
-\frac{d}{dx}\left\{D_0(x)\frac{d}{dx}\Phi_0(x)\right\} + \left[\Sigma_t(x) - \Sigma_{s,0}(x)\right]\Phi_0(x) = Q(x), \tag{1.19}
$$

which is equivalent to

$$
-\frac{d}{dx}\left\{D_0(x)\frac{d}{dx}\phi_0(x)\right\} + \left[\Sigma_t(x) - \Sigma_{s,0}(x)\right]\phi_0(x) = Q(x). \tag{1.20}
$$

For $N=3$, the equations are

$$
\begin{aligned}
-\frac{d}{dx}\left\{D_0(x)\frac{d}{dx}\Phi_0(x)\right\} + \left[\Sigma_t(x) - \Sigma_{s,0}(x)\right]\Phi_0(x) &= -\left[\Sigma_t(x) - \Sigma_{s,0}(x)\right]k_{0,2}\Phi_2(x) + Q(x), \\
&= 2\left[\Sigma_t(x) - \Sigma_{s,0}(x)\right]\Phi_2(x) + Q(x),
\end{aligned}
\tag{1.21}
$$

and

$$
\begin{aligned}
-\frac{d}{dx}\left\{D_2(x)\frac{d}{dx}\Phi_2(x)\right\} &+ \left[\Sigma_t(x) - \Sigma_{s,2}(x)\right]\Phi_2(x) \\
&= m_0\left(\left[\Sigma_t(x) - \Sigma_{s,0}(x)\right]\left(\Phi_0(x) + k_{0,2}\Phi_2(x)\right) - \delta_{0,0}Q(x)\right) \\
&= \frac{2}{5}\left(\left[\Sigma_t(x) - \Sigma_{s,0}(x)\right]\left(\Phi_0(x) - 2\Phi_2(x)\right) - Q(x)\right).
\end{aligned}
\tag{1.22}
$$

1.3 Adjoint form of the SP_1 Equation

The adjoint form of the SP_1 equation is identical to its standard form, but with the scattering matrix transposed. The boundary conditions will be

$$\psi(0,\mu) = \psi_0^b(\mu), \quad 0 \le \mu \le 1$$
$$\psi(X,\mu) = \psi_X^b(\mu), \quad -1 \le \mu \le 0$$

Use Fick's law to get

$$\psi(x,\mu) = \frac{1}{2}\left(\phi(x) - 3D\frac{d}{dx}\phi(x)\right).$$

Multiply by μ and integrate:

$$\int_0^1 \mu\psi(0,\mu)\,d\mu = \frac{1}{2}\int_0^1 \left(\mu\phi(0) - 3D\mu^2\frac{d}{dx}\phi(x)\right)d\mu$$
$$J^+(0) = \frac{1}{2}\left[\frac{1}{2}\phi(x) - D\frac{d}{dx}\phi(x)\right].$$

The negative signs in the below equations may need reversal, then, for the adjoint problem:

$$\frac{1}{2}\phi(0) - D\frac{d}{dx}\phi(0) = 2J^+(0),$$
$$\frac{1}{2}\phi(X) + D\frac{d}{dx}\phi(X) = 2J^-(X),$$

2 Discretization of the SP_1 Equation

2.1 SP_1 Equation

The SP1 equation is

$$-\frac{d}{dx}D(x)\frac{d}{dx}\phi(x)+\left[\Sigma_t(x)-\Sigma_s(x)\right]\phi(x)-Q(x)=0, \tag{2.1}$$

with

$$D(x)=\left[\Sigma_t(x)-\Sigma_{s,1}(x)\right]^{-1}. \tag{2.2}$$

Their discretized form will be derived for the energy-independent problem, which is very easily generalizable to the multigroup approximation.

Convert the equation to local coordinates, with

$$\xi=\frac{\Delta\xi_i}{\Delta x_i}\left(x-\bar{x}_i\right) \tag{2.3}$$

to get

$$-\frac{d}{d\xi}D(\xi)\frac{d}{d\xi}\phi(\xi)+\left(\frac{dx}{d\xi}\right)^2\left[\Sigma_t(\xi)-\Sigma_s(\xi)\right]\phi(\xi)-\left(\frac{dx}{d\xi}\right)^2 Q(\xi)=0. \tag{2.4}$$

Use the continuous finite element method with linear basis functions,

$$b_1=\frac{1}{2}(1-\xi), \tag{2.5a}$$

$$b_2=\frac{1}{2}(\xi+1), \tag{2.5b}$$

and identical (Galerkin) weight functions,

$$w_1=\frac{1}{2}(1-\xi), \tag{2.6a}$$

$$w_2=\frac{1}{2}(\xi+1). \tag{2.6b}$$

Operate on the equation by $\int_{-1}^{1} w_\beta(\xi)(\cdot)\,d\xi$ and integrate by parts to get

$$\begin{aligned}
0&=\int_{-1}^{1} w_\beta(\xi)\left[-\frac{d}{d\xi}\left\{D(\xi)\frac{d}{d\xi}\phi(\xi)\right\}+\left(\frac{dx}{d\xi}\right)^2\left[\Sigma_t(\xi)-\Sigma_s(\xi)\right]\phi(\xi)-\left(\frac{dx}{d\xi}\right)^2 Q(\xi)\right]d\xi\\
&=\int_{-1}^{1}\left[\frac{d}{d\xi}w_\beta(\xi)D(\xi)\frac{d}{d\xi}\phi(\xi)+w_\beta(\xi)\left(\frac{dx}{d\xi}\right)^2\left[\Sigma_t(\xi)-\Sigma_s(\xi)\right]\phi(\xi)-w_\beta(\xi)\left(\frac{dx}{d\xi}\right)^2 Q(\xi)\right]d\xi\\
&\quad-\oint_\Gamma w_\beta(\xi)D(\xi)\frac{d}{d\xi}\phi(\xi)\,d\Gamma.
\end{aligned} \tag{2.7}$$

Note that as this is the continuous finite element method and not its discontinuous counterpart, that the bounds of the integral are only $-1\le\xi\le1$ because the weight functions are defined on a cellwise basis and are zero outside that cell. The boundary integral is then only pertinent at the edge of the system (Γ)

and is zero otherwise. Continuing on for the interior cells, expand $\phi(\xi)$ and $Q(\xi)$ out in terms of the basis functions,

$$\phi(\xi) = \sum_{\alpha} b_{\alpha}(\xi)\,\tilde{\phi}_{\alpha}, \tag{2.8a}$$

$$Q(\xi) = \sum_{\alpha} b_{\alpha}(\xi)\,\tilde{Q}_{\alpha}, \tag{2.8b}$$

and assume that the material properties are constant in each cell to get (for each β, with the sum over α implied)

$$\begin{aligned}
0 &= D_i\tilde{\phi}_{\alpha}\int_{-1}^{1}\frac{d}{d\xi}b_{\alpha}(\xi)\,\frac{d}{d\xi}w_{\beta}(\xi)\,d\xi + \left(\frac{\Delta x_i}{\Delta\xi_i}\right)^2\left[\Sigma_{t,i}-\Sigma_{s,i}\right]\tilde{\phi}_{\alpha}\int_{-1}^{1}b_{\alpha}(\xi)\,w_{\beta}(\xi)\,d\xi \\
&\quad - \left(\frac{\Delta x_i}{\Delta\xi_i}\right)^2\tilde{Q}_{\alpha}\int_{-1}^{1}b_{\alpha}(\xi)\,w_{\beta}(\xi)\,d\xi \\
&= D_i\tilde{\phi}_{\alpha}k_{\alpha,\beta} + \left(\frac{\Delta x_i}{\Delta\xi_i}\right)^2\left[\Sigma_{t,i}-\Sigma_{s,i}\right]\tilde{\phi}_{\alpha}\ell_{\alpha,\beta} - \left(\frac{\Delta x_i}{\Delta\xi_i}\right)^2\tilde{Q}_{\alpha}\ell_{\alpha,\beta},
\end{aligned} \tag{2.9}$$

with

$$k_{\alpha,\beta} = \int_{-1}^{1}\frac{d}{d\xi}b_{\alpha}(\xi)\,\frac{d}{d\xi}w_{\beta}(\xi)\,d\xi, \tag{2.10}$$

$$\ell_{\alpha,\beta} = \int_{-1}^{1}b_{\alpha}(\xi)\,w_{\beta}(\xi), \tag{2.11}$$

or evaluated and matrix form for the defined linear basis and weight functions,

$$\boldsymbol{k} = \begin{bmatrix}\frac{1}{2} & -\frac{1}{2}\\ -\frac{1}{2} & \frac{1}{2}\end{bmatrix}, \tag{2.12}$$

$$\boldsymbol{\ell} = \begin{bmatrix}\frac{2}{3} & \frac{1}{3}\\ \frac{1}{3} & \frac{2}{3}\end{bmatrix}. \tag{2.13}$$

The matrix system of equations for a single cell with

$$\tilde{\boldsymbol{Q}} = \begin{bmatrix}\tilde{Q}_1\\ \tilde{Q}_2\end{bmatrix}, \tag{2.14}$$

$$\tilde{\boldsymbol{\phi}} = \begin{bmatrix}\tilde{\phi}_1\\ \tilde{\phi}_2\end{bmatrix}, \tag{2.15}$$

and $\Delta\xi_i = 2$ will be

$$\left(D_i\boldsymbol{k} + \frac{1}{4}\left(\Delta x_i\right)^2\left[\Sigma_{t,i}-\Sigma_{s,i}\right]\boldsymbol{\ell}\right)\tilde{\boldsymbol{\phi}} = \frac{1}{4}\left(\Delta x_i\right)^2\boldsymbol{\ell}\tilde{\boldsymbol{Q}}. \tag{2.16}$$

2.2 SP_1 Boundary Conditions

While the boundary integral in Eq. 2.7 will correct the transport equation appropriately for the edge cells, a simpler approach is to simply remove the equation in the boundary cell for which $w_\beta(\xi) = 1$ (which will apply to only one equation for the one-dimensional Lagrange basis functions) and enforce the boundary condition directly by discretizing the boundary equation itself.

The boundary conditions for the SP1 equations are

$$\begin{aligned}\frac{1}{2}\phi(0) - D\frac{d}{dx}\phi(0) &= 2J^{+}(0),\\ \frac{1}{2}\phi(X) + D\frac{d}{dx}\phi(X) &= 2J^{-}(X),\end{aligned}$$

which in cellwise coordinates are

$$\begin{aligned}\frac{1}{2}\left(\frac{\Delta x_i}{\Delta \xi_i}\right)\phi(-1) - D\frac{d}{d\xi}\phi(-1) &= 2\left(\frac{\Delta x_i}{\Delta \xi_i}\right)J^{+}(-1), \qquad i = 1,\\ \frac{1}{2}\left(\frac{\Delta x_i}{\Delta \xi_i}\right)\phi(1) + D\frac{d}{d\xi}\phi(1) &= 2\left(\frac{\Delta x_i}{\Delta \xi_i}\right)J^{-}(1), \qquad i = I.\end{aligned}$$

This equation can easily be evaluated in terms of the basis functions and their derivatives,

$$\begin{aligned}\frac{d}{d\xi}\phi(\xi) &= \sum_{\alpha}\frac{d}{d\xi}b_\alpha(\xi)\tilde{\phi}_\alpha\\ &= -\frac{1}{2}\tilde{\phi}_1 + \frac{1}{2}\tilde{\phi}_2,\end{aligned}$$

to get

$$\begin{aligned}\frac{1}{2}\left(\frac{\Delta x_i}{\Delta \xi_i}\right)\tilde{\phi}_1 - D\left(-\frac{1}{2}\tilde{\phi}_1 + \frac{1}{2}\tilde{\phi}_2\right) &= 2\left(\frac{\Delta x_i}{\Delta \xi_i}\right)J^{+}(-1), \qquad i = 1,\\ \frac{1}{2}\left(\frac{\Delta x_i}{\Delta \xi_i}\right)\tilde{\phi}_2 + D\left(-\frac{1}{2}\tilde{\phi}_1 + \frac{1}{2}\tilde{\phi}_2\right) &= 2\left(\frac{\Delta x_i}{\Delta \xi_i}\right)J^{-}(1), \qquad i = I,\end{aligned}$$

or

$$\begin{aligned}\frac{1}{2}\left[\left(\frac{\Delta x_i}{\Delta \xi_i}\right) + D\right]\tilde{\phi}_1 - \frac{1}{2}D\tilde{\phi}_2 &= 2\left(\frac{\Delta x_i}{\Delta \xi_i}\right)J^{+}, \qquad i = 1,\\ -\frac{1}{2}D\tilde{\phi}_1 + \frac{1}{2}\left[\left(\frac{\Delta x_i}{\Delta \xi_i}\right) + D\right]\tilde{\phi}_2 &= 2\left(\frac{\Delta x_i}{\Delta \xi_i}\right)J^{-}, \qquad i = I.\end{aligned}$$

3 Source Code

Class	Description
Manatee	Main file for running from an input file (working, but not very developed).
tst_Manatee	Includes simple tests for the program (doesn't compare to known results yet).
Data	Includes all nuclear data for the problem.
Finite_Element	Represents a single finite element.
Mesh	Includes all the finite elements and general data about the problem geometry.
Parser	Reads data from input and outputs data after calculations.
Ordinates	Contains the ordinates, weights, and other data for the discrete ordinates (S_N) method.
Sn_Transport	Contains a discontinuous finite element code for spherical and slab geometries.
SP1_Transport	Contains a continuous finite element code for slab geometry with forward and adjoint capabilities. The Trilinos classes Epetra and Amesos handle the sparse matrix storage and linear solution, respectively.
Transport_Model	General class for transport problems. Will eventually be a virtual class for Sn_Transport, SP1_Transport, and any other transport types.

A few things need work:

- Using virtual functions for the scattering (to make the adjoint calculation efficient) and the other virtual capabilities mentioned in the table above.
- The data is checked decently, but needs additional safeguards for data whose size changes between methods (the boundary source, for instance, which depends on ordinates for the S_N calculations but is simply the incoming current for the SP_N equations).
- The SP_N equations in the class SPn_Transport (not included below) need some correcting and possibly need to be recast in a different form. If they're written in the summation form, then it could be possible to solve them iteratively for Ψ_n, where $n \in \mathbb{E}$.
- The direct Amesos solver from Trilinos should be swapped out for an iterative Belos solver from the same package.
- The Epetra storage isn't parallelized yet, although that should be a relatively easy task once a parallel solver from Belos is implemented.