**NE 255: HW5** April Novak

① The $S_N$ form of the neutron transport equation is:

$$\begin{split}\hat{\Omega}_a\cdot\nabla\psi_a^g(\vec{r})+\Sigma_t^g(\vec{r})\psi_a^g(\vec{r}) =\\ \sum_{l=0}^{N}\left[Y_{l0}^e(\hat{\Omega}_a)s_{l0}^g(\vec{r})+\sum_{m=1}^{l}\left(Y_{lm}^e(\hat{\Omega}_a)s_{lm}^g(\vec{r})+Y_{lm}^o(\hat{\Omega}_a)v_{lm}^g(\vec{r})\right)\right]+\\ \sum_{g'=1}^{G}\sum_{l=0}^{N}\Sigma_{s,l}(\vec{r},g'\to g)\left[Y_{l0}^e(\hat{\Omega}_a)e_{l0}^{g'}(\vec{r})+\sum_{m=1}^{l}\left(Y_{lm}^e(\hat{\Omega}_a)e_{lm}^{g'}(\vec{r})+Y_{lm}^o(\hat{\Omega}_a)o_{lm}^{g'}(\vec{r})\right)\right]\end{split} \tag{1}$$

The $g$ superscript refers to the energy group number, and $G$ to the total number of energy groups. Within-group scattering is counted in the scattering term, but is canceled by the total interaction term, which avoids double counting. The above equation can be cast in operator form. The external group source is defined as:

$$q_e^g(\vec{r})=\sum_{l=0}^{N}\left[Y_{l0}^e(\hat{\Omega}_a)s_{l0}^g(\vec{r})+\sum_{m=1}^{l}\left(Y_{lm}^e(\hat{\Omega}_a)s_{lm}^g(\vec{r})+Y_{lm}^o(\hat{\Omega}_a)v_{lm}^g(\vec{r})\right)\right] \tag{2}$$

in order to permit easier equation manipulation. In addition, several operators are defined. The transport operator is:

$$\mathbf{L}\equiv\hat{\Omega}_a\cdot\nabla+\Sigma_t^g \tag{3}$$

and the solution vector $\Psi$ is a column vector containing each energy group flux vector $\vec{\psi}$:

$$\Psi=\begin{bmatrix}\vec{\psi}_1 & \vec{\psi}_2 & \vec{\psi}_3 & \cdots & \vec{\psi}_G\end{bmatrix}^T \tag{4}$$

where $\vec{\psi}$ is then a column vector that contains the flux for each discrete angle for that particular energy group $g$:

$$\vec{\psi}_g=\begin{bmatrix}\psi_1^g & \psi_2^g & \psi_3^g & \cdots & \psi_n^g\end{bmatrix}^T \tag{5}$$

So, the vector of unknowns is organized by energy group, and in the section pertaining to each group, by angle. In order to discuss the size of these matrices and vectors, several variables are defined:

$$\begin{aligned}G&=\text{number of energy groups}\\ t&=\text{number of moments}\\ n&=\text{number of angles}\\ N&=P_N\text{ order}\\ N_c&=\text{number of cells}\\ N_e&=\text{number of unknowns per cell}\end{aligned} \tag{6}$$

So, if the problem consisted of a single cell with a single node (i.e. only one spatial unknown), then $\Psi$ would be an $(Gn)\times 1$ vector. However, the problems solves will consist of a spatial domain that is also present in $\Psi$. In reality, each $\psi_{\hat{\Omega}_n}^g$ term is solved as a function of space, and is therefore not a scalar value, but an $N_cN_e\times 1$ vector. The $S_N$ equation for a particular angle and group is solved as a function of space using a variety of methods, such as the finite element method or the finite difference method. So, the total size of $\Psi$ is $N_cN_eGn\times 1$. The external source $q_e^g$ is defined in a similar manner, and is represented as $Q$, an $N_cN_eGn\times 1$ vector.

Expressing the mapping between flux moments and the scattering term is more difficult simply due to the difficulty in visualizing the matrix multiplication. The moment-to-discrete matrix is used to project harmonic moments onto discrete angle space. It is defined as:

$$\mathbf{M} = \left\{ \begin{array}{ccccccccccc} Y^o_{00}(\hat{\Omega}_1) & Y^e_{00}(\hat{\Omega}_1) & Y^o_{01}(\hat{\Omega}_1) & Y^e_{01}(\hat{\Omega}_1) & Y^o_{10}(\hat{\Omega}_1) & Y^e_{10}(\hat{\Omega}_1) & Y^o_{11}(\hat{\Omega}_1) & Y^e_{11}(\hat{\Omega}_1) & \cdots & Y^o_{NN}(\hat{\Omega}_1) & Y^e_{NN}(\hat{\Omega}_1) \\ Y^o_{00}(\hat{\Omega}_2) & Y^e_{00}(\hat{\Omega}_2) & Y^o_{01}(\hat{\Omega}_2) & Y^e_{01}(\hat{\Omega}_2) & Y^o_{10}(\hat{\Omega}_2) & Y^e_{10}(\hat{\Omega}_2) & Y^o_{11}(\hat{\Omega}_2) & Y^e_{11}(\hat{\Omega}_2) & \cdots & Y^o_{NN}(\hat{\Omega}_2) & Y^e_{NN}(\hat{\Omega}_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ Y^o_{00}(\hat{\Omega}_n) & Y^e_{00}(\hat{\Omega}_n) & Y^o_{01}(\hat{\Omega}_n) & Y^e_{01}(\hat{\Omega}_n) & Y^o_{10}(\hat{\Omega}_n) & Y^e_{10}(\hat{\Omega}_n) & Y^o_{11}(\hat{\Omega}_n) & Y^e_{11}(\hat{\Omega}_n) & \cdots & Y^o_{NN}(\hat{\Omega}_n) & Y^e_{NN}(\hat{\Omega}_n) \end{array} \right\} \tag{7}$$

But, some of these components are actually always zero - for instance, from the expansion in Eq. (1), if $l = 0$, then there are no spherical harmonics for which $m \neq 0$. This eliminates $Y^e_{01}$ and $Y^o_{01}$. In addition, $Y^o_{00} = 0$ and $Y^o_{10} = 0$ since there is no odd component for $m = 0$. This gives a simplified version of the moment-to-discrete matrix, where the only difference between this matrix and the more explicit one above is that several of the low $l$ and $m$ spherical harmonics that are zero are removed.

$$\mathbf{M} = \left\{ \begin{array}{cccccccc} Y^e_{00}(\hat{\Omega}_1) & Y^e_{10}(\hat{\Omega}_1) & Y^o_{11}(\hat{\Omega}_1) & Y^e_{11}(\hat{\Omega}_1) & Y^e_{20}(\hat{\Omega}_1) & \cdots & Y^o_{NN}(\hat{\Omega}_1) & Y^e_{NN}(\hat{\Omega}_1) \\ Y^e_{00}(\hat{\Omega}_2) & Y^e_{10}(\hat{\Omega}_2) & Y^o_{11}(\hat{\Omega}_2) & Y^e_{11}(\hat{\Omega}_2) & Y^e_{20}(\hat{\Omega}_2) & \cdots & Y^o_{NN}(\hat{\Omega}_2) & Y^e_{NN}(\hat{\Omega}_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ Y^e_{00}(\hat{\Omega}_n) & Y^e_{10}(\hat{\Omega}_n) & Y^o_{11}(\hat{\Omega}_n) & Y^e_{11}(\hat{\Omega}_n) & Y^e_{20}(\hat{\Omega}_n) & \cdots & Y^o_{NN}(\hat{\Omega}_n) & Y^e_{NN}(\hat{\Omega}_n) \end{array} \right\} \tag{8}$$

The scattering matrix contains along its diagonal the scattering moments:

$$\mathbf{S}_{g' \to g} = \left[\Sigma_{s0}\right] \tag{9}$$

With this definition, the discretization of the transport equation leads to the following matrix system:

$$\mathbf{L} \begin{bmatrix} \Psi_1 \\ \Psi_2 \\ \Psi_3 \\ \vdots \\ \Psi_G \end{bmatrix} = \begin{bmatrix} \mathbf{M} & 0 & 0 & 0 & 0 \\ 0 & \mathbf{M} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{M} & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} & \mathbf{S}_{13} & \cdots & \mathbf{S}_{1G} \\ \mathbf{S}_{21} & \mathbf{S}_{22} & \mathbf{S}_{23} & \cdots & \mathbf{S}_{2G} \\ \mathbf{S}_{31} & \mathbf{S}_{32} & \mathbf{S}_{33} & \cdots & \mathbf{S}_{3G} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{S}_{G1} & \mathbf{S}_{G2} & \mathbf{G}_{23} & \cdots & \mathbf{S}_{GG} \end{bmatrix} \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \vdots \\ \Phi_G \end{bmatrix} + \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ \vdots \\ Q_G \end{bmatrix} \tag{10}$$

Hence, the upper triangular portion of the scattering matrix represents upscattering, while the diagonal represents within-group scattering, and the lower triangular portion represents downscattering. $\Phi$ represents a vector of the angular flux moments used in the expansion of the scattering term, organized by group.

$$\Phi_g = \begin{bmatrix} o^g_{00} & e^g_{00} & o^g_{01} & e^g_{01} & o^g_{10} & e^g_{10} & \cdots & o^g_{NN} & e^g_{NN} \end{bmatrix}^T \tag{11}$$

where $e_{lm}$ and $o_{lm}$ are defined by Eq. (13). Note, however, that some of these terms are zero due to how the expansion is performed. Moments with $l = 0$ are only nonzero for $m = 0$. And, there is no odd component for $l = 0$.

$$\Phi_g = \begin{bmatrix} e^g_{00} & e^g_{10} & o^g_{11} & e^g_{11} & \cdots & o^g_{NN} & e^g_{NN} \end{bmatrix}^T \tag{12}$$

$$\begin{aligned} e_{lm}(\vec{r}, E') &= \int_{4\pi} d\hat{\Omega}' Y^e_{lm}(\hat{\Omega}') \psi(\vec{r}, E', \hat{\Omega}') \\ o_{lm}(\vec{r}, E') &= \int_{4\pi} d\hat{\Omega}' Y^o_{lm}(\hat{\Omega}') \psi(\vec{r}, E', \hat{\Omega}') \end{aligned} \tag{13}$$

For this specific problem, the discretized $S_N$ equations become:

$$\underbrace{\mathbf{L}}_{\textcircled{1}} \underbrace{\begin{bmatrix} \Psi_1 \\ \Psi_2 \\ \Psi_3 \end{bmatrix}}_{\textcircled{2}} = \underbrace{\begin{bmatrix} \mathbf{M} & 0 & 0 \\ 0 & \mathbf{M} & 0 \\ 0 & 0 & \mathbf{M} \end{bmatrix}}_{\textcircled{3}} \underbrace{\begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} & \mathbf{S}_{13} \\ \mathbf{S}_{21} & \mathbf{S}_{22} & \mathbf{S}_{23} \\ \mathbf{S}_{31} & \mathbf{S}_{32} & \mathbf{S}_{33} \end{bmatrix}}_{\textcircled{4}} \underbrace{\begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \end{bmatrix}}_{\textcircled{5}} + \underbrace{\begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}}_{\textcircled{6}} \tag{14}$$

where the groups are numbered 1, 2, and 3, with 1 being the highest-energy group.

**(a)**: The dimensions of each of the matrices shown in Eq. (14) are summarized in the table below, according to the numbering scheme shown above. For this problem, $G = 3, n = 8, N_c = 4^3, N_e = 4, N = 2, t = (N+1)^2 = 9$. $N_e = 4$ because for each cell, the cell-centered flux and the outgoing fluxes for three of the six faces are unknown.

**Table 1.** Sizes of matrices in Eq. (14).

| Matrix or Vector | Size (Symbolic) | Size (Actual) |
|---|---|---|
| $\textcircled{1}$ | $GnN_cN_e \times GnN_cN_e$ | 6144 × 6144 |
| $\textcircled{2}$ | $GnN_cN_e \times 1$ | 6144 × 1 |
| $\textcircled{3}$ | $GnN_cN_e \times GtN_cN_e$ | 6144 × 6192 |
| $\textcircled{4}$ | $GtN_cN_e \times GtN_cN_e$ | 6192 × 6192 |
| $\textcircled{5}$ | $GtN_cN_e \times 1$ | 6192 × 1 |
| $\textcircled{6}$ | $GnN_cN_e \times 1$ | 6144 × 1 |

**(b)**: For this particular problem:

$$\mathbf{M}_{gg} = \left\{ \begin{matrix} Y^e_{00}(\hat{\Omega}_1) & Y^e_{10}(\hat{\Omega}_1) & Y^o_{11}(\hat{\Omega}_1) & Y^e_{11}(\hat{\Omega}_1) & Y^e_{20}(\hat{\Omega}_1) & Y^o_{21}(\hat{\Omega}_1) & Y^e_{21}(\hat{\Omega}_1) & Y^o_{22}(\hat{\Omega}_1) & Y^e_{22}(\hat{\Omega}_1) \\ Y^e_{00}(\hat{\Omega}_2) & Y^e_{10}(\hat{\Omega}_2) & Y^o_{11}(\hat{\Omega}_2) & Y^e_{11}(\hat{\Omega}_2) & Y^e_{20}(\hat{\Omega}_2) & Y^o_{21}(\hat{\Omega}_2) & Y^e_{21}(\hat{\Omega}_2) & Y^o_{22}(\hat{\Omega}_2) & Y^e_{22}(\hat{\Omega}_2) \\ Y^e_{00}(\hat{\Omega}_3) & Y^e_{10}(\hat{\Omega}_3) & Y^o_{11}(\hat{\Omega}_3) & Y^e_{11}(\hat{\Omega}_3) & Y^e_{20}(\hat{\Omega}_3) & Y^o_{21}(\hat{\Omega}_3) & Y^e_{21}(\hat{\Omega}_3) & Y^o_{22}(\hat{\Omega}_3) & Y^e_{22}(\hat{\Omega}_3) \\ Y^e_{00}(\hat{\Omega}_4) & Y^e_{10}(\hat{\Omega}_4) & Y^o_{11}(\hat{\Omega}_4) & Y^e_{11}(\hat{\Omega}_4) & Y^e_{20}(\hat{\Omega}_4) & Y^o_{21}(\hat{\Omega}_4) & Y^e_{21}(\hat{\Omega}_4) & Y^o_{22}(\hat{\Omega}_4) & Y^e_{22}(\hat{\Omega}_4) \\ Y^e_{00}(\hat{\Omega}_5) & Y^e_{10}(\hat{\Omega}_5) & Y^o_{11}(\hat{\Omega}_5) & Y^e_{11}(\hat{\Omega}_5) & Y^e_{20}(\hat{\Omega}_5) & Y^o_{21}(\hat{\Omega}_5) & Y^e_{21}(\hat{\Omega}_5) & Y^o_{22}(\hat{\Omega}_5) & Y^e_{22}(\hat{\Omega}_5) \\ Y^e_{00}(\hat{\Omega}_6) & Y^e_{10}(\hat{\Omega}_6) & Y^o_{11}(\hat{\Omega}_6) & Y^e_{11}(\hat{\Omega}_6) & Y^e_{20}(\hat{\Omega}_6) & Y^o_{21}(\hat{\Omega}_6) & Y^e_{21}(\hat{\Omega}_6) & Y^o_{22}(\hat{\Omega}_6) & Y^e_{22}(\hat{\Omega}_6) \\ Y^e_{00}(\hat{\Omega}_7) & Y^e_{10}(\hat{\Omega}_7) & Y^o_{11}(\hat{\Omega}_7) & Y^e_{11}(\hat{\Omega}_7) & Y^e_{20}(\hat{\Omega}_7) & Y^o_{21}(\hat{\Omega}_7) & Y^e_{21}(\hat{\Omega}_7) & Y^o_{22}(\hat{\Omega}_7) & Y^e_{22}(\hat{\Omega}_7) \\ Y^e_{00}(\hat{\Omega}_8) & Y^e_{10}(\hat{\Omega}_8) & Y^o_{11}(\hat{\Omega}_8) & Y^e_{11}(\hat{\Omega}_8) & Y^e_{20}(\hat{\Omega}_8) & Y^o_{21}(\hat{\Omega}_8) & Y^e_{21}(\hat{\Omega}_8) & Y^o_{22}(\hat{\Omega}_8) & Y^e_{22}(\hat{\Omega}_8) \end{matrix} \right\} \tag{15}$$

Then, matrix $\textcircled{3}$ is assembled by aligning these **M** matrices along the diagonal, as shown in Eq. (14). The scattering matrix **S** is comprised of scattering matrices $\mathbf{S}_{gg'}$, and is shown in Eq. (14). One of these scattering matrices is, for example:

$$\mathbf{S}_{21} = \begin{bmatrix} \Sigma^{21}_{s0} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \Sigma^{21}_{s1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \Sigma^{21}_{s1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Sigma^{21}_{s1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \Sigma^{21}_{s2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \Sigma^{21}_{s2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \Sigma^{21}_{s2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Sigma^{21}_{s2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Sigma^{21}_{s2} \end{bmatrix} \tag{16}$$

The vector of unknowns is:

$$\Psi = \begin{bmatrix} \Psi_1 & \Psi_2 & \Psi_3 \end{bmatrix} \tag{17}$$

where, for example,

$$\Psi_1 = \begin{bmatrix} \psi^1_1 & \psi^1_2 & \psi^1_3 & \psi^1_4 & \psi^1_5 & \psi^1_6 & \psi^1_7 & \psi^1_8 \end{bmatrix} \tag{18}$$

where the subscripts indicates the angle, and the superscript indicates the energy group. The vector of flux moments is:

$$\Phi = \begin{bmatrix}\Phi_1 & \Phi_2 & \Phi_3\end{bmatrix} \tag{19}$$

where, for example,

$$\Phi_1 = \begin{bmatrix}e^1_{00} & e^1_{10} & o^1_{11} & e^1_{11} & e^1_{20} & o^1_{21} & e^1_{21} & o^1_{22} & e^1_{22}\end{bmatrix}^T \tag{20}$$

where the superscript indicates the energy group. The moments $e$ and $o$ are defined in Eq. (13).

**(c)**: The $\mathbf{D}$ matrix is used to convert between the discrete angular fluxes to the moments of the angular flux by $\phi = \mathbf{D}\psi$. The discrete form of $\mathbf{D}$ is defined by:

$$\mathbf{D} = \mathbf{M}^T\mathbf{W} \tag{21}$$

where $\mathbf{M}$ is defined in Eq. (14) as containing $\mathbf{M}_{gg}$ matrices along its diagonal and $\mathbf{W}$ is a diagonal matrix containing $n$ quadrature points per group (since we're using $S_N$) - these quadrature points are aligned along the diagonal in a repeated fashion for each energy group. The form of $\mathbf{D}$ is therefore:

$$\mathbf{D} = \begin{bmatrix}
w_1Y^e_{00} & w_2Y^e_{00} & w_3Y^e_{00} & w_4Y^e_{00} & w_5Y^e_{00} & w_6Y^e_{00} & w_7Y^e_{00} & w_8Y^e_{00}\\
w_1Y^e_{10} & w_2Y^e_{10} & w_3Y^e_{10} & w_4Y^e_{10} & w_5Y^e_{10} & w_6Y^e_{10} & w_7Y^e_{10} & w_8Y^e_{10}\\
w_1Y^o_{11} & w_2Y^o_{11} & w_3Y^o_{11} & w_4Y^o_{11} & w_5Y^o_{11} & w_6Y^o_{11} & w_7Y^o_{11} & w_8Y^o_{11}\\
w_1Y^e_{11} & w_2Y^e_{11} & w_3Y^e_{11} & w_4Y^e_{11} & w_5Y^e_{11} & w_6Y^e_{11} & w_7Y^e_{11} & w_8Y^e_{11}\\
w_1Y^e_{20} & w_2Y^e_{20} & w_3Y^e_{20} & w_4Y^e_{20} & w_5Y^e_{20} & w_6Y^e_{20} & w_7Y^e_{20} & w_8Y^e_{20}\\
w_1Y^o_{21} & w_2Y^o_{21} & w_3Y^o_{21} & w_4Y^o_{21} & w_5Y^o_{21} & w_6Y^o_{21} & w_7Y^o_{21} & w_8Y^o_{21}\\
w_1Y^e_{21} & w_2Y^e_{21} & w_3Y^e_{21} & w_4Y^e_{21} & w_5Y^e_{21} & w_6Y^e_{21} & w_7Y^e_{21} & w_8Y^e_{21}\\
w_1Y^o_{22} & w_2Y^o_{22} & w_3Y^o_{22} & w_4Y^o_{22} & w_5Y^o_{22} & w_6Y^o_{22} & w_7Y^o_{22} & w_8Y^o_{22}\\
w_1Y^e_{22} & w_2Y^e_{22} & w_3Y^e_{22} & w_4Y^e_{22} & w_5Y^e_{22} & w_6Y^e_{22} & w_7Y^e_{22} & w_8Y^e_{22}
\end{bmatrix} \tag{22}$$

**(d)**: Because the matrices involved in only this $4 \times 4$ mesh are relatively large, none of them are directly formed so that zero entries do not need to be stored. Instead, all matrix-matrix or matrix-vector products are performed according to their indices. For instance, to multiply a $n \times 1$ vector $V$ by a $n \times n$ matrix $M$ to give a resulting vector $r$, each component of $r$ would be dictated by:

$$r_i = \sum_{j=1}^{n} M_{ij}V_j \tag{23}$$

and so on for the matrix multiplication. Avoiding storage of the matrices is a huge advantage, since a majority of their contents are zero anyways, and for iterative solution methods, it is common practice to take advantage of the fact that the matrix does not need to be performed to further reduce the runtime.

**(e)**: To get a matrix system entirely in terms of $\Psi$, begin with the operator form:

$$\mathbf{L}\Psi = \mathbf{MS}\Phi + \mathbf{M}q_e \tag{24}$$

and insert $\Psi = \mathbf{D}\Phi$. This can transform the above equation either into an equation for $\Phi$ or $\Psi$. I'm not sure which approach is done in practice, so I'll write both forms:

$$\mathbf{LD}\Phi = \mathbf{MS}\Phi + \mathbf{M}q_e \quad \rightarrow \quad (\mathbf{LD} - \mathbf{MS})\Phi = \mathbf{M}q_e \tag{25}$$

And for the equation in terms of $\Psi$:

$$\mathbf{L}\Psi = \mathbf{MSD}^{-1}\Psi + \mathbf{M}q_e \quad \rightarrow \quad (\mathbf{L} - \mathbf{MSD}^{-1})\Psi = \mathbf{M}q_e \tag{26}$$

$\textcircled{2}$: This question implements a Jacobi solver to create a multigroup transport solver using the Weighted Diamond Difference (WDD) code developed from the previous assignment. This code from the previous assignment is used to perform the within-group solves for the angular flux in the $\pm 0.2, \pm 0.5, \pm 0.7$ directions. Within each of these iterations, the scattering source within each group (angle to angle) is iteratively updated. Then, outside of each within-group calculation, the overall group source is updated. I assume that the right boundary condition is kept as a reflective boundary condition, since this was performed in the previous assignment (and we're not given a condition there). I also assumed that the incoming flux for group 1 was evenly split amongst the three different angles. The group source is updated according to:

$$\text{group source} = \sum_{g'=1}^{G} \Sigma_s(g^{'} \rightarrow g)\psi_{g'} \tag{27}$$

using the same set of quadrature rules as from the previous assignment (note: not a real quadrature set, but am using this from the previous assignment). Fig. 1 shows the scalar fluxes obtained using this multigroup solver. The convergence of the energy iteration was evaluated based on a comparison of the group energy sources (from scattering amongst energy sources) and the previous group energy sources. All three fluxes satisfy the reflective boundary condition, and the accuracy is difficult to judge, except that the relative magnitudes of the fluxes does scale with their total cross sections, while I would expect that downscattering would lead the group 3 flux to have a higher magnitude.

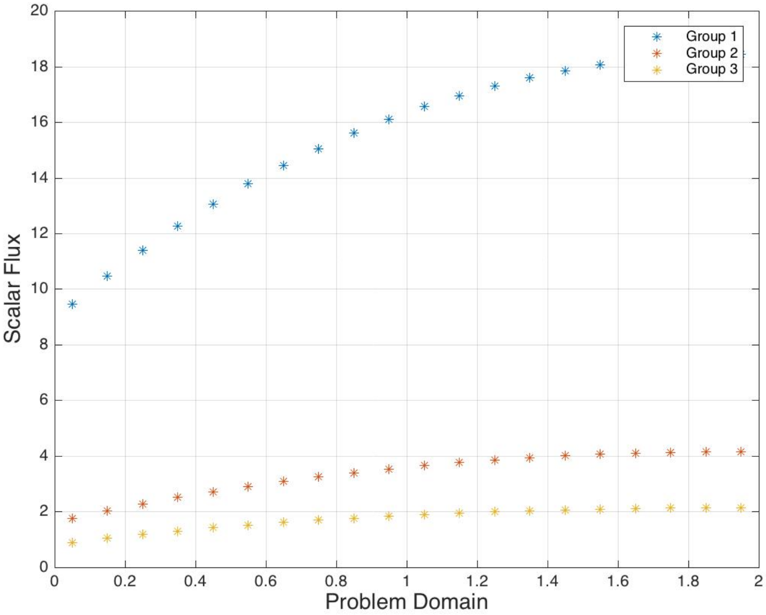

**Figure 1.** Scalar flux for the three energy groups.

# 1 Appendix

This section contains all the code used for each question.

## 1.1 Question 2

```
% HW4, Question 3
clear all

% logistical things
fontsize = 16;

% discrete angles
mu02 = 0.2;
```

```
mu05 = 0.5;
mu07 = 0.7;

% group cross sections
Et = [0.5, 0.8, 1.0];
Es = [0.1+0.3+0.1, 0.1+0.3, 0.1+0.3];
Es_mat = [0.1, 0.0, 0.0; 0.3, 0.1, 0.1; 0.1, 0.3, 0.3];

% convergence specifications
tolerance = 1e-6;
max_iterations = 1000;
energy_tolerance = 1e-6;
num_energy_iterations = 0;
max_energy_iterations = 2000;

alpha = 0.5;
h = 0.10;
G = 3;

mesh = 0:h:2;
elem_length = mesh(2);
coordinates = 0:(elem_length / 2):2;

% external sources
qe(1,:) = 1.5 .* ones(1, length(coordinates));
qe(2,:) = 0.0 .* ones(1, length(coordinates));
qe(3,:) = 0.2 .* ones(1, length(coordinates));

% initial guesses for scattering sources
q_energy = zeros(3, length(coordinates));

% initial guess for the scattering source for __each__ group. After each 1
% within-group calculation, this source will be specialized depending
% on the energy group
energy_norm = 1; % arbitrary initial value
while ((energy_norm > energy_tolerance) && (num_energy_iterations <
   ↪ max_energy_iterations))
    for g = 1:G

            if num_energy_iterations == 0
                q = 1.0 .* coordinates;
            else
                q = q_energy(g,:);
            end

            norm = 1;
            % each within-group calculation finds the angular flux for that
                ↪ group

            % initial guess for the scattering source is simply set to the
            % external source (equivalent to guessing psi = 0.0)

            % iterate
            num_iterations = 0;
```

```
while ((norm > tolerance) && (num_iterations < max_iterations))

    % initialize
    psi02 = zeros(1, length(coordinates));
    psi_n02 = zeros(1, length(coordinates));
    psi05 = zeros(1, length(coordinates));
    psi_n05 = zeros(1, length(coordinates));
    psi07 = zeros(1, length(coordinates));
    psi_n07 = zeros(1, length(coordinates));

    if g == 1
        psi02(1) = 0.5/3; % incoming flux boundary condition
        psi05(1) = 0.5/3; % incoming flux boundary condition
        psi07(1) = 0.5/3; % incoming flux boundary condition
    end

    % positive sweep
    i = 1;
    while i <= (length(coordinates) - 2)
        % compute cell-centered value
        psi02(i + 1) = (q(i+1) + qe(g, i+1) + (2 / (1 + alpha)) *
           ↪ (abs(mu02) * psi02(i) ./ elem_length)) / (Et(g) + (2
           ↪  / (1 + alpha)) * abs(mu02) / elem_length);
        psi05(i + 1) = (q(i+1) + qe(g, i+1) + (2 / (1 + alpha)) *
           ↪ (abs(mu05) * psi05(i) ./ elem_length)) / (Et(g) + (2
           ↪  / (1 + alpha)) * abs(mu05) / elem_length);
        psi07(i + 1) = (q(i+1) + qe(g, i+1) + (2 / (1 + alpha)) *
           ↪ (abs(mu07) * psi07(i) ./ elem_length)) / (Et(g) + (2
           ↪  / (1 + alpha)) * abs(mu07) / elem_length);

        % compute out-going value
        psi02(i + 2) = (2 / (1 + alpha)) * psi02(i + 1) - ((1 -
           ↪ alpha)/(1 + alpha)) * psi02(i);
        psi05(i + 2) = (2 / (1 + alpha)) * psi05(i + 1) - ((1 -
           ↪ alpha)/(1 + alpha)) * psi05(i);
        psi07(i + 2) = (2 / (1 + alpha)) * psi07(i + 1) - ((1 -
           ↪ alpha)/(1 + alpha)) * psi07(i);

        % move to next cell
        i = i + 2;
    end


    % negative sweep - store the outgoing flux to apply periodic
       ↪ BC
    i = length(coordinates);
    psi_n02(i) = psi02(end);
    psi_n05(i) = psi05(end);
    psi_n07(i) = psi07(end);

    while i > 2

        % compute the cell-centered flux
        psi_n02(i - 1) = (q(i-1) + qe(g, i-1) + (2/(1 - alpha)) *
            ↪ (mu02 * psi_n02(i) / elem_length)) / (Et(g) + 2 *
            ↪ mu02 / (elem_length * (1 - alpha)));
        psi_n05(i - 1) = (q(i-1) + qe(g, i-1) + (2/(1 - alpha)) *
            ↪ (mu05 * psi_n05(i) / elem_length)) / (Et(g) + 2 *
            ↪ mu05 / (elem_length * (1 - alpha)));
        psi_n07(i - 1) = (q(i-1) + qe(g, i-1) + (2/(1 - alpha)) *
            ↪ (mu07 * psi_n07(i) / elem_length)) / (Et(g) + 2 *
            ↪ mu07 / (elem_length * (1 - alpha)));

        % compute the outgoing face flux
        psi_n02(i - 2) = (2 / (1 - alpha)) * psi_n02(i-1) - ((1+
            ↪ alpha)/(1-alpha)) * psi_n02(i);
        psi_n05(i - 2) = (2 / (1 - alpha)) * psi_n05(i-1) - ((1+
            ↪ alpha)/(1-alpha)) * psi_n05(i);
        psi_n07(i - 2) = (2 / (1 - alpha)) * psi_n07(i-1) - ((1+
            ↪ alpha)/(1-alpha)) * psi_n07(i);

        i = i - 2;
    end

    % update the within-group scattering source
    q_new = zeros(1, length(psi02));
    q_new = Es(g) .* (1/6) .* (psi02 + psi_n02 + psi05 + psi_n05 +
       ↪  psi07 + psi_n07) + q_energy(g,:)/3;

    % test for convergence
    norm = 0;
    for u = 1:length(q_new)
        norm = norm + (q_new(u) - q(u)).^2;
    end
    norm = sqrt(norm);

    q = q_new;

    num_iterations = num_iterations + 1;
    if num_iterations >= max_iterations
        disp('Maximum␣number␣of␣iterations␣reached!')
    end

end

% compute scalar flux (cell-centered)
phi = zeros(1, length(psi02(2:2:end)));
phi = (1/6) .* (psi02(2:2:end) + psi_n02(2:2:end) + psi05(2:2:end)
   ↪  + psi_n05(2:2:end) + psi07(2:2:end) + psi_n07(2:2:end));

phi_total = zeros(1, length(psi02));
phi_total = (1/6) .* (psi02 + psi_n02 + psi05 + psi_n05 + psi07 +
   ↪ psi_n07);

% save the scalar flux to the appropriate energy group
group_flux(g,:) = phi;
group_flux_total(g,:) = phi_total;
```

```
    end

    % after updating all of the groups, update the group sources using the
    % scattering matrix.
    for gg = 1:G
        q_energy(gg,:) = Es_mat(gg,1).*group_flux_total(1,:) + Es_mat(gg,2).*
           ↪ group_flux_total(2,:) + Es_mat(gg,3).*group_flux_total(3,:);
    end


    % compute whether or not the energy iteration has converged
    energy_norm = 0;
    for u = 1:length(q_energy(1,:))
        energy_norm = energy_norm + (q_energy(1,u) - q(1,u)).^2;
    end
    energy_norm = sqrt(energy_norm);

    num_energy_iterations = num_energy_iterations + 1;
    if num_energy_iterations >= max_energy_iterations
        disp('Maximum␣number␣of␣energy␣iterations␣reached!')
    end

end






% plot the scalar flux
plot(coordinates(2:2:end), group_flux(1,:), '*', coordinates(2:2:end),
   ↪ group_flux(2,:), '*', coordinates(2:2:end), group_flux(3,:), '*')
hold on
grid on
legend('Group␣1', 'Group␣2', 'Group␣3')
xlabel('Problem␣Domain', 'FontSize', fontsize-2)
ylabel('Scalar␣Flux', 'FontSize', fontsize-2)
saveas(gcf, 'ScalarFlux_3Group', 'jpeg')
```