NE 255: HW4 April Novak

(1) The S_N equations provide the framework to discretize the angular flux in angle, but the equations still must be discretized in space. In general, there are two different means to perform spatial discretization - cell balance methods, which preserve conservation of the solution, and finite element methods, which do not necessarily preserve conservation of the solution. In a single mesh cell, cell balance methods will provide a statement that is equivalent to conservation of neutrons, while finite element methods satisfy the governing equation in a weighted-integral sense. For simplicity, all of the spatial discretization schemes discussed here are applied to the simplified transport equation that neglects time dependence and assumes some angular discretization has already been applied to ψ such that it is independent of angle (in each of these equations):

$$\hat{\Omega}\cdot\nabla\psi(\vec{r},E)+\Sigma_t(\vec{r},E)\psi(\vec{r},E)=S(\vec{r},E) \tag{1}$$

where S represents the scattering, fission, and external sources. There will be one of these equations for each discrete direction in the S_N method. For all the following discretization schemes, a Cartesian mesh given in the figure in the `spatial-disr.pdf` is assumed, with i representing indices in the x-direction, j in the y-direction, and k in the z-direction.

Cell balance schemes point values for the solution ψ over a discretized spatial mesh. With the definitions for the components of $\hat{\Omega}$ as $\hat{\Omega}=\mu\hat{x}+\eta\hat{y}+\xi\hat{z}$, integrating Eq. (1) over the mesh cell gives:

$$\int_{x_{i-1/2}}^{x_{i+1/2}}dx\mu\frac{\partial\psi}{\partial x}+\int_{y_{j-1/2}}^{y_{j+1/2}}dy\eta\frac{\partial\psi}{\partial y}+\int_{z_{k-1/2}}^{z_{k+1/2}}dz\xi\frac{\partial\psi}{\partial z}+\Sigma_{t,ijk}E\psi_{ijk}(\vec{r},E)=S_{ijk}(E) \tag{2}$$

where it has been assumed that Σ_t and S are constant over the cell, and are represented by their cell-centered values. In addition, the energy-dependence has been dropped. Without approximation, the integrals in the above reduce to:

$$\frac{\mu}{\Delta_i}(\psi_{i+1/2}-\psi_{i-1/2})+\frac{\eta}{\Delta_j}(\psi_{j+1/2}-\psi_{j-1/2})+\frac{\xi}{\Delta_k}(\psi_{k+1/2}-\psi_{k-1/2})+\Sigma_{t,ijk}\psi_{ijk}=S_{ijk} \tag{3}$$

With this discretization method, there must be some way to relate the cell-centered value ijk to the face values. The different options we have for discretization are related to the choices we can make for how to relate these quantities. The Diamond Difference and Step Difference schemes choose the following relationship:

$$\psi_{i\pm1/2}=\frac{2}{1\pm\alpha}\psi_{ijk}-\frac{1\mp\alpha}{1\pm\alpha}\bar{\psi}_{i\mp1/2} \tag{4}$$

and likewise for the y and z directions. $\bar{\psi}$ represents a known flux, and depending on whether a sweep is being performed in the positive-μ direction ($\bar{\psi}_{i-1/2}$ is known) or in the negative-μ direction ($\bar{\psi}_{i+1/2}$ is known), either the + or - option is selected in Eq. (4). Once the sweep direction is known, the relationship in Eq. (4) is used in the governing equation Eq. (3) to solve for the cell-centered flux in terms of only known quantities:

$$\psi_{ijk}=\frac{S_{ijk}+\frac{2}{1\pm\alpha}\left(\frac{|\mu|}{\Delta_i}\bar{\psi}_{i\mp1/2}+\frac{|\eta|}{\Delta_j}\bar{\psi}_{j\mp1/2}+\frac{|\xi|}{\Delta_k}\bar{\psi}_{k\mp1/2}\right)}{\Sigma_{t,ijk}+\frac{2}{1\pm\alpha}\left(\frac{|\mu|}{\Delta_i}+\frac{|\eta|}{\Delta_j}+\frac{|\xi|}{\Delta_k}\right)} \tag{5}$$

where it has been assumed that α is selected to be the same in all directions. α is a parameter used to tune the spatial discretization. Common choices for α have their own names. The Diamond Difference method sets $\alpha=0$. With this choice, the cell-centered flux is an average of the face fluxes. The Step Difference method, on the other hand, sets $\alpha=\pm1$ such that the cell-centered flux is set equal to the incoming face flux (choose 1 or -1 accordingly so that Eq. (4) provides an expression for ψ_{ijk} in terms of the known incoming face flux).The Diamond Difference method is second-order in space, while the Step Difference method is only first-order.

The objective is to choose the correct relationship in Eq. (4) such that the known quantities, or the incoming fluxes to each cell, can be used to compute the unknown quantities, or the outgoing fluxes for each cell, in a "sweep" over the spatial mesh. Each of these sweeps is performed for each individual flux $\psi(\hat{\Omega}_a$.

Now that the numerical method has been set up, the sweeping process can be described in more detail for the different situations prompted in the homework:

(a): For a sweep in the positive-μ direction, we supposedly know $\bar{\psi}_{i-1/2}$, so we would relate the cell-centered flux to the outgoing flux in the following way:

$$\psi_{i+1/2} = \frac{2}{1+\alpha}\psi_{ijk} - \frac{1-\alpha}{1+\alpha}\bar{\psi}_{i-1/2} \quad \rightarrow \quad \psi_{ijk} = \frac{1}{2}\left[(1+\alpha)\psi_{i+1/2} + (1-\alpha)\bar{\psi}_{i-1/2}\right] \tag{6}$$

So, to perform the sweep, we would begin at the boundary at which a boundary condition is specified. At this boundary, the incoming flux $\bar{\psi}_{i-1/2}$ is known. Then, Eq. (5) is used to determine the cell-centered flux ψ_{ijk}. Then, finally, Eq. (6) is used to compute the outgoing fluxes from the cell-centered fluxes. To be more precise, the steps for a Diamond Difference method ($\alpha = 0$) are:

1. Begin at the boundary at which the incoming fluxes are known. Compute the cell-centered fluxes using:

$$\psi_{ijk} = \frac{S_{ijk} + 2\left(\frac{|\mu|}{\Delta_i}\bar{\psi}_{i-1/2} + \frac{|\eta|}{\Delta_j}\bar{\psi}_{j-1/2} + \frac{|\xi|}{\Delta_k}\bar{\psi}_{k-1/2}\right)}{\Sigma_{t,ijk} + 2\left(\frac{|\mu|}{\Delta_i} + \frac{|\eta|}{\Delta_j} + \frac{|\xi|}{\Delta_k}\right)} \tag{7}$$

2. Then, compute the outgoing fluxes using:

$$\psi_{i+1/2} = 2\psi_{ijk} - \bar{\psi}_{i-1/2} \tag{8}$$

3. Repeat steps 1 and 2 for the remaining cells, using the outgoing fluxes from step 2 as the new incoming fluxes in step 1. If there is no fission, scattering, or external source, then only one sweep is needed. On the other hand, if any of these sources are present, then after each sweep, an updated source S_{ijk} must be computed, and the process repeated until reaching convergence.

(b): For a sweep in the negative-μ direction, we supposedly know $\bar{\psi}_{i+1/2}$, so we would relate the cell-centered flux to the outgoing flux in the following way:

$$\psi_{i-1/2} = \frac{2}{1-\alpha}\psi_{ijk} - \frac{1+\alpha}{1-\alpha}\bar{\psi}_{i+1/2} \quad \rightarrow \quad \psi_{ijk} = \frac{1}{2}\left[(1+\alpha)\psi_{i+1/2} + (1-\alpha)\bar{\psi}_{i-1/2}\right] \tag{9}$$

Very similar to a sweep in the positive direction, we would begin at the boundary at which a boundary condition is specified. At this boundary, the incoming flux $\bar{\psi}_{i+1/2}$ is known. Then, Eq. (5) is used to determine the cell-centered flux ψ_{ijk}. Then, finally, Eq. (9) is used to compute the outgoing fluxes from the cell-centered fluxes. To be more precise, the steps for a Diamond Difference method ($\alpha = 0$) are:

1. Begin at the boundary at which the incoming fluxes are known. Compute the cell-centered fluxes using:

$$\psi_{ijk} = \frac{S_{ijk} + 2\left(\frac{|\mu|}{\Delta_i}\bar{\psi}_{i+1/2} + \frac{|\eta|}{\Delta_j}\bar{\psi}_{j+1/2} + \frac{|\xi|}{\Delta_k}\bar{\psi}_{k+1/2}\right)}{\Sigma_{t,ijk} + 2\left(\frac{|\mu|}{\Delta_i} + \frac{|\eta|}{\Delta_j} + \frac{|\xi|}{\Delta_k}\right)} \tag{10}$$

2. Then, compute the outgoing fluxes using:

$$\psi_{i-1/2} = 2\psi_{ijk} - \bar{\psi}_{i+1/2} \tag{11}$$

3. Repeat steps 1 and 2 for the remaining cells, using the outgoing fluxes from step 2 as the new incoming fluxes in step 1. If there is no fission, scattering, or external source, then only one sweep is needed. On the other hand, if any of these sources are present, then after each sweep, an updated source S_{ijk} must be computed, and the process repeated until reaching convergence.

The only difference between a sweep in the positive and negative μ-directions is the notation on which flux is known - either $\psi_{i+1/2}$ or $\psi_{i-1/2}$ is known.

(c) If reflective boundary conditions are present, then begin from the side of the domain for which the conditions are known. Apply steps 1-3 listed in part (a) (if moving in the $\mu > 0$ direction) or steps 1-3 listed in part (b) (if moving in the $\mu < 0$ direction) in order to reach the reflecting boundary from a boundary at which the incoming flux is known. Using the equations in either (a) or (b) is determined by which boundary has values specified - begin from that boundary, and sweep until reaching the reflecting boundary. Then, once at the reflecting boundary, apply the condition that $\psi_{N+1,i+1/2} = \psi_{N+1,i-1/2}$, where N is the number of mesh cells such that $N+1$ is the number of the node on the right face, assuming a linear, Cartesian grid. This condition is applied such that a sweep then begins moving in the opposite direction of the first sweep (and so apply steps 1-3 in part (a) if initially the steps from part (b) were applied, and vice versa. These steps are not repeated here to avoid redundancy). After each double sweep, updates are performed to S_{ijk}. This process is repeated until reaching convergence. If reflective boundary conditions exist on both ends of the domain, then an initial guess for the incoming fluxes at the starting point must be made, and then each double sweep is again halted to perform updates to S_{ijk} until reaching convergence.

(d) In a single sweep, each cell requires that the incoming fluxes and source S_{ijk} to proceed with the sweep. To compute S_{ijk}, the scattering, fission, and external source, we need to have all flux moments for that cell. Once an updated S_{ijk} is computed however, those flux moments can be discarded, except those that contribute to the incoming flux of the next cell. So, as we sweep through the mesh, once a cell computation is complete, the incoming and cell-centered flux values for that cell can be discarded so long as S_{ijk} has been updated for the next sweep.

② Truncation error can be quantified for a 1-D, zero-source flux. This flux signifies the flux of neutrons moving with angle $\hat{\Omega}_a$, but for simplicity the a subscripts on ψ and μ are not retained throughout this problem, but are left to be implied.

(a): The equation to be solved analytically is:

$$\mu\frac{d\psi}{dx} + \Sigma_t\psi = 0 \tag{12}$$

The analytical solution to this equation is:

$$\frac{d\psi}{\psi} = \frac{-\Sigma_t}{|\mu|}dx \quad \rightarrow \quad \int_x^{x^{'}} \frac{d\psi}{\psi} = \int_x^{x^{'}} \frac{-\Sigma_t}{|\mu|}dx \tag{13}$$

Performing the integration:

$$\ln\left(\frac{\psi_{x'}}{\psi_x}\right) = \frac{-\Sigma_t}{|\mu|}\delta_i \quad \rightarrow \quad \psi_{x'} = \exp\left(\frac{-\Sigma_t}{|\mu|}\delta_i\right)\psi_x \tag{14}$$

where $\delta_i \equiv x^{'} - x$.

(b): For a Cartesian grid with $x^{'} = x_{i+1/2}$ and $x = x_{i-1/2}$, Eq. (14) becomes:

$$\psi_{i+1/2} = \exp\left(\frac{-\Sigma_t}{|\mu|}\Delta_i\right)\psi_{i-1/2} \tag{15}$$

$$\ln\left(\frac{\psi_{i+1/2}}{\psi_{i-1/2}}\right) = \frac{-\Sigma_t}{|\mu|}\Delta_i \quad \rightarrow \quad \psi_{i+1/2} = \exp\left(\frac{-\Sigma_t}{|\mu|}\delta_i\right)\psi_{i-1/2} \tag{16}$$

where $\Delta_i \equiv x_{i+1/2} - x_{i-1/2}$. Introducing the definition $h = \Sigma_t\Delta_i/2|\mu|$, the above becomes:

$$\ln\left(\frac{\psi_{i+1/2}}{\psi_{i-1/2}}\right) = -2h \quad \rightarrow \quad \psi_{i+1/2} = \exp\left(-2h\right)\psi_{i-1/2} \tag{17}$$

(c): To determine the numerical equivalent of Eq. (17), the diamond difference equations give ψ_{ijk} in terms of the face flux values as:

$$\psi_{i+1/2} = 2\psi_{ijk} - \bar{\psi}_{i-1/2} \tag{18}$$

The discretized form of the transport equation is already given in Eq. (3). Inserting Eq. (18) into the 1-D form of Eq. (3) (with the source term neglected for this specific problem) for ψ_{ijk} gives a second relationship between $\psi_{i+1/2}$ and $\psi_{i-1/2}$:

$$\frac{\mu}{\Delta_i}(\psi_{i+1/2} - \psi_{i-1/2}) + \Sigma_t \frac{1}{2}\left(\psi_{i+1/2} + \psi_{i-1/2}\right) = 0 \tag{19}$$

Rearranging, and inserting the definition for h:

$$\psi_{i+1/2} = \frac{1-h}{1+h}\psi_{i-1/2} \tag{20}$$

(d): The power series of e^{-2h} is:

$$e^{-2h} \approx 1 + (-2h) + \frac{(-2h)^2}{2!} + \cdots = 1 - 2h - 2h^2 + \cdots \tag{21}$$

The coefficient in Eq. (20) can be written as:

$$\frac{1-h}{1+h} = \frac{(1-h)^2}{1-h^2} = (1-h)^2 \sum_{i=0}^{\infty}(h^2)^n \approx (1-h)^2\left(1 + (h^2) + \frac{h^4}{2} + \cdots\right) \tag{22}$$

Expanding the first term in the above equation fully:

$$\frac{1-h}{1+h} \approx (1-h)^2 + \cdots = 1 - 2h - h^2 + \cdots \tag{23}$$

By comparing Eq. (23) with Eq. (21), it can be seen that the truncation error is very nearly second order, and hence the discretization is of second order.

(e): From Eq. (20), it is clear that $\psi_{i+1/2}$ could be negative if $h > 1$. In other words, from the definition of h, the flux can be negative if:

$$\Delta_i > \frac{2|\mu|}{\Sigma_t} \tag{24}$$

From a cost perspective, the coarser the mesh, the better, since then the simulation will not be overly-limited by the mesh requirements. The conditions for which the mesh refinement will cause the most complications (i.e. produce negative fluxes) is for small $|\mu|$ or large Σ_t. In order to avoid negative fluxes, we require $\Delta_i < \frac{2|\mu|}{\Sigma_t}$.

③ For this problem, the boundary conditions are:

$$\begin{gathered}\psi(0) = 0 \text{ for } \mu > 0 \\ \psi(2) \text{ for } \mu > 0 = \psi(2) \text{ for } \mu < 0\end{gathered} \tag{25}$$

Due to the reflecting boundary conditions present at $x = 2$, sweeps in both the positive and negative μ directions are needed. The numerical algorithm is:

1. Begin at $x = 0$ with the boundary condition $\psi_{i-1/2} = 0$.

2. Compute the cell-centered flux using:

$$\psi_i = \frac{S_i + \frac{2}{1+\alpha}\left(\frac{|\mu|}{\Delta_i}\bar{\psi}_{i-1/2}\right)}{\Sigma_{t,i} + \frac{2}{1+\alpha}\left(\frac{|\mu|}{\Delta_i}\right)} \tag{26}$$

3. Compute the outgoing face flux using:

$$\psi_{i+1/2} = \frac{2}{1+\alpha}\psi_{ijk} - \frac{1-\alpha}{1+\alpha}\bar{\psi}_{i-1/2} \tag{27}$$

4. Repeat steps 2 and 3 until reaching the very last cell. At the last cell, after performing step 3, the outgoing flux has been computed. This must be reflected to serve as the incoming flux for a sweep in the negative-μ direction. At the last cell, set $\psi_{i+1/2} = \psi_{i+1/2}$ to serve as the incoming face flux for a sweep in the negative-μ direction.

5. Begin the sweep in the negative-μ direction. Compute the cell-centered flux using:

$$\psi_i = \frac{S_i + \frac{2}{1-\alpha}\left(\frac{|\mu|}{\Delta_i}\bar{\psi}_{i+1/2}\right)}{\Sigma_{t,i} + \frac{2}{1-\alpha}\left(\frac{|\mu|}{\Delta_i}\right)} \tag{28}$$

6. Compute the outgoing face flux using:

$$\psi_{i-1/2} = \frac{2}{1-\alpha}\psi_{ijk} - \frac{1+\alpha}{1-\alpha}\bar{\psi}_{i+1/2} \tag{29}$$

7. Repeat steps 5 and 6 for all cells until reaching the very first cell again. At this point, if there is no source (fission, scattering, or external), then no further sweeps are needed, and the solution is complete.

(a): Using the numerical algorithm described above, Fig. 1 shows the angular flux for positive μ and negative μ (no quadrature rule has been applied to obtain the scalar flux yet), for a mesh spacing of $h = 0.08$. This is included purely to show some results before the quadrature is applied.

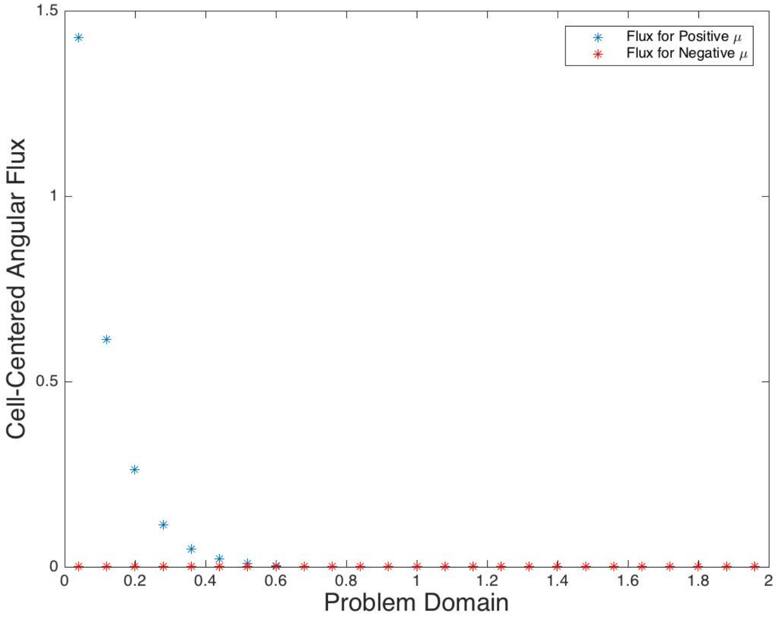

Figure 1. Angular flux for positive and negative μ for $h = 0.08$.

To obtain the scalar flux, the angular flux must be integrated over angle:

$$\phi = \int_{-1}^{1} \psi(\mu) d\mu \approx \sum_{i=1}^{N} w_i \psi(\mu_i) \tag{30}$$

This integration is approximated with a quadrature rule. Because there is only one value of $|\mu|$, $\mu_i = 0.1$, and because the problem statement says to use equi-probable weights, the weights are set to $w_i = 1.0$ so that integration over $-1 \leq \mu \leq 1$ gives 2 if the angular flux is not dependent on angle. Note that this is not a real quadrature set. With this quadrature rule, Fig. 2 shows the scalar flux for $h = 0.08, 0.1, 0.125, 0.2, 0.4$.

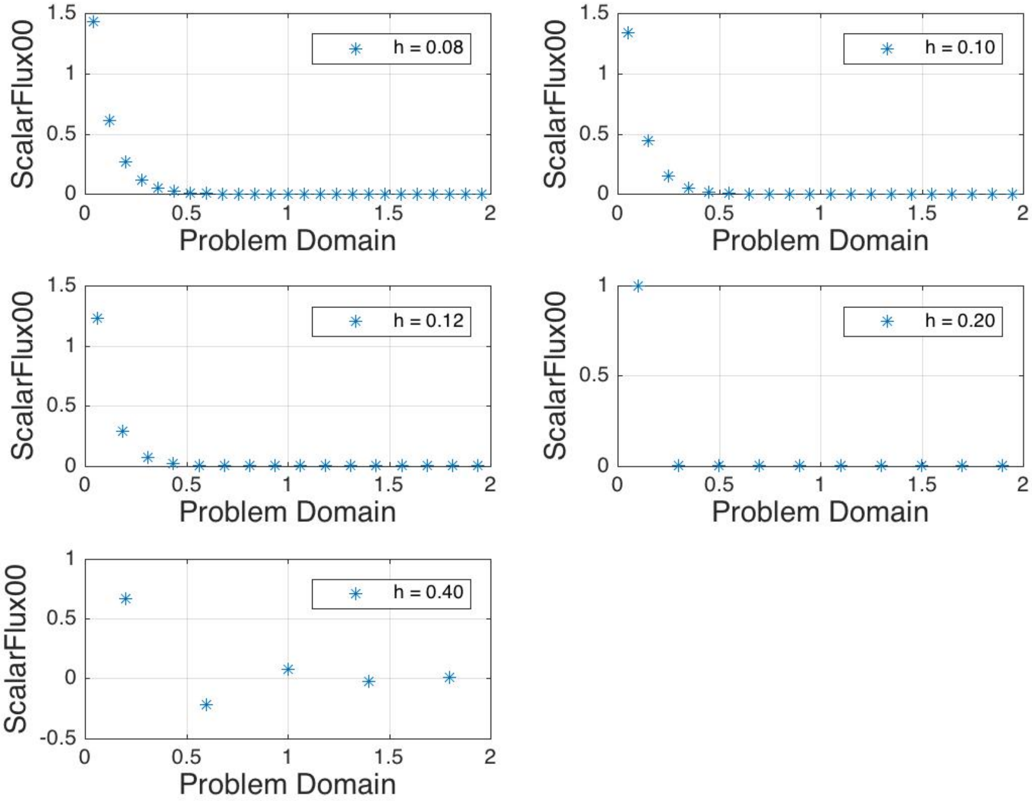

Figure 2. Cell-centered scalar flux for $h = 0.08, 0.1, 0.125, 0.2, 0.4$ and $\alpha = 0$.

As can be seen, negative fluxes are obtained for $h = 0.4$, the largest of the mesh spacings. From Eq. (24), the flux can be negative for:

$$\Delta_i > 0.2 \tag{31}$$

The only mesh spacing with $\Delta_i > 0.2$ is the last spacing, of $h = 0.4$. Because negative flux is observed, this shows that the predictions from question 2 are accurate in terms of the needed mesh refinement to avoid the unphysical negative flux that can result.

(b): Using the same property data, part (a) is repeated for $\alpha = -0.9, -0.5, 0.25, 0.5, 0.9$.

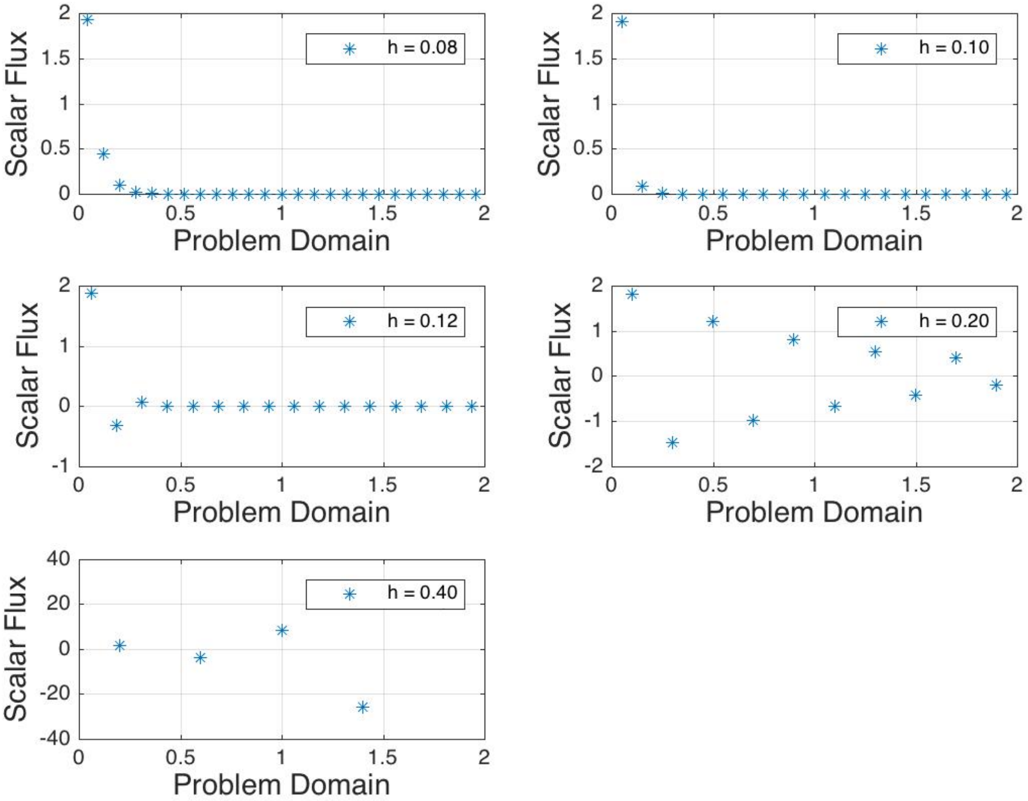

Figure 3. Cell-centered scalar flux for $h = 0.08, 0.1, 0.125, 0.2, 0.4$ and $\alpha = -0.9$.

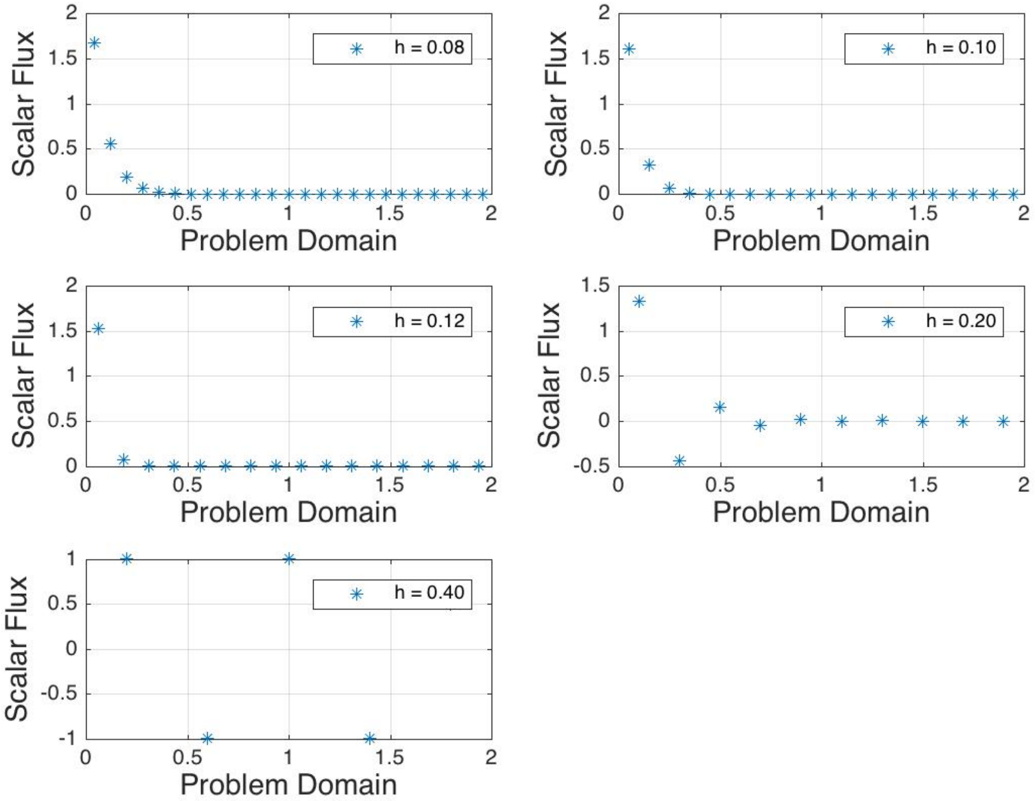

Figure 4. Cell-centered scalar flux for $h = 0.08, 0.1, 0.125, 0.2, 0.4$ and $\alpha = -0.5$.

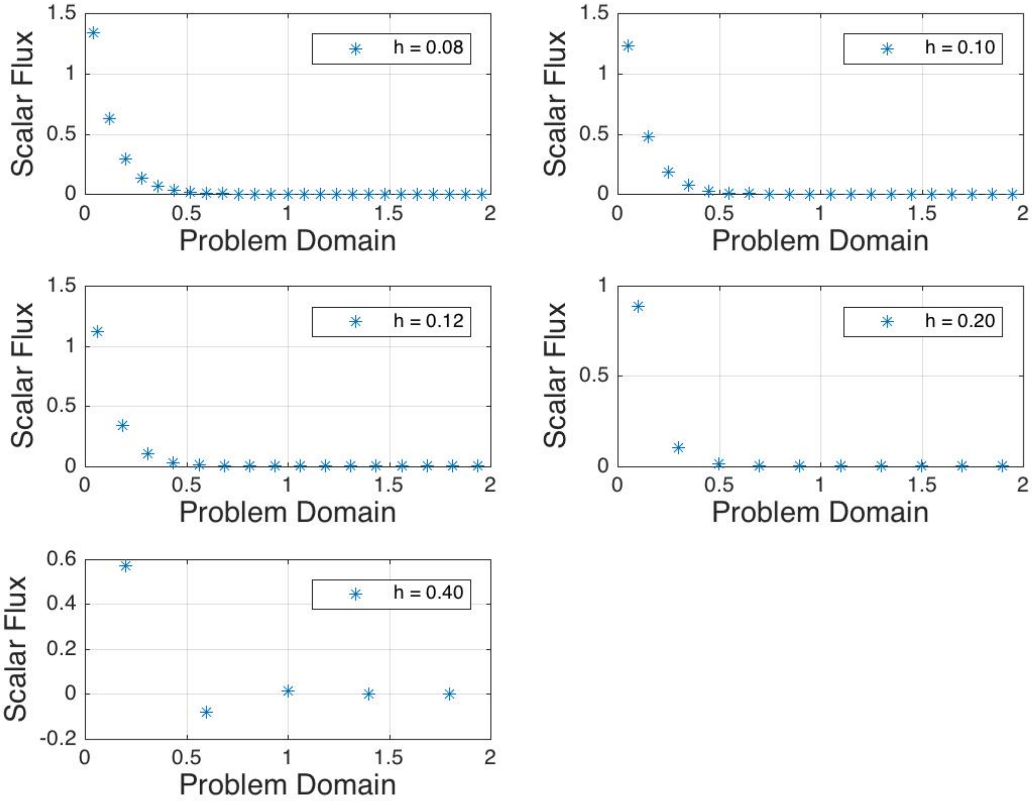

Figure 5. Cell-centered scalar flux for $h = 0.08, 0.1, 0.125, 0.2, 0.4$ and $\alpha = 0.25$.

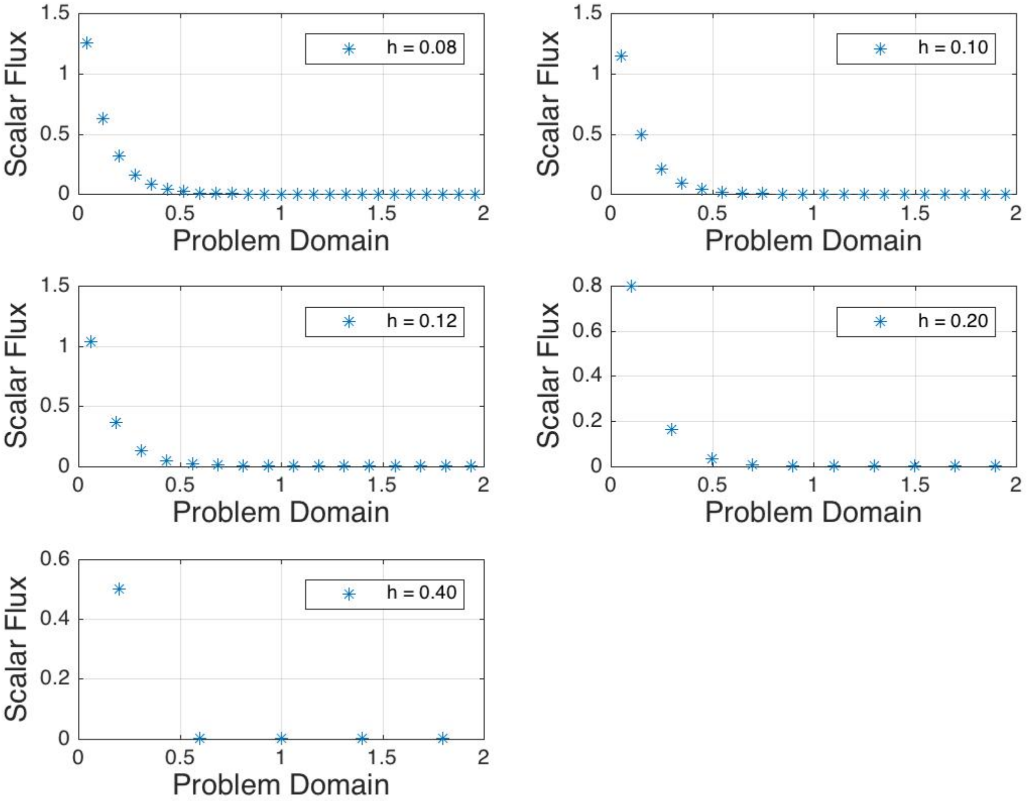

Figure 6. Cell-centered scalar flux for $h = 0.08, 0.1, 0.125, 0.2, 0.4$ and $\alpha = 0.5$.

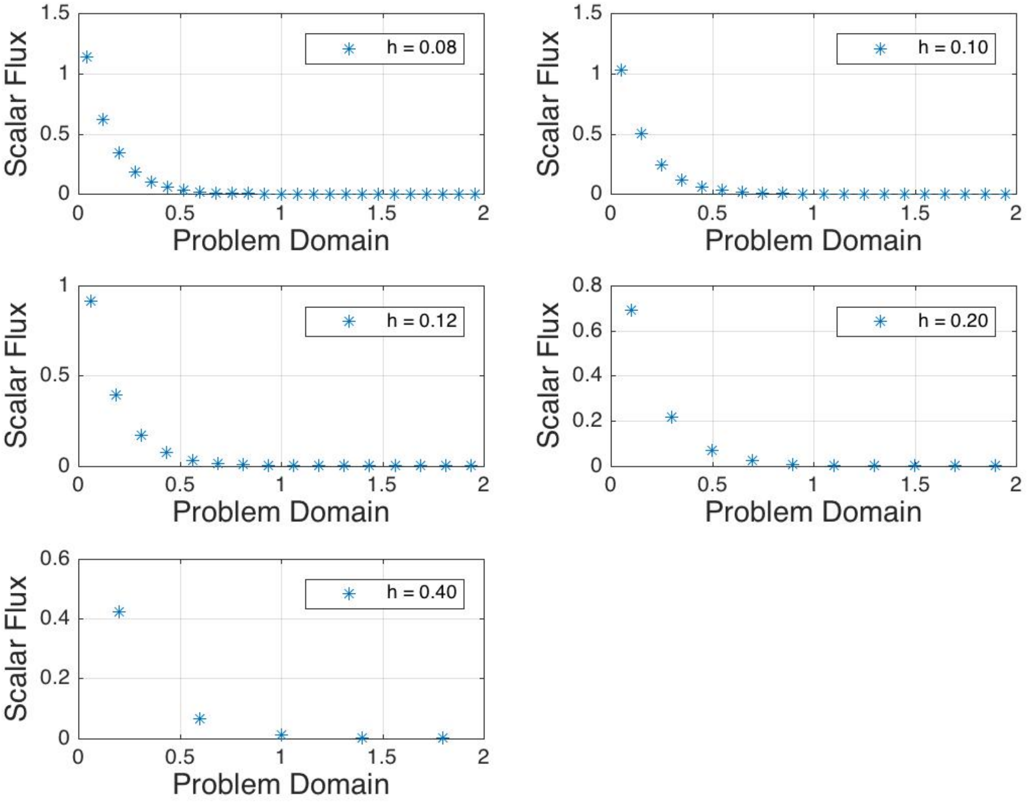

Figure 7. Cell-centered scalar flux for $h = 0.08, 0.1, 0.125, 0.2, 0.4$ and $\alpha = 0.9$.

The impact of changing α on the accuracy of the solution can be judged by comparing the results with those obtained for $h = 0.08$ and $\alpha = 0$ in Fig. 2 (assuming that such a fine mesh gives the correct answer). The table below summarizes the impact of changing α on whether or not negative fluxes are obtained or if the solution is unphysical (for example, has oscillations, since that would not be expected to occur for a problem with no complicated source profile). From the figures, it is clear that the less positive the α, the more probable the occurrence of negative flux, oscillations between positive and negative flux, and an inability to capture the smooth exponential decay of the flux beginning at $x = 0$. For instance, while both $\alpha = 0.5$ and $\alpha = 0.9$ give reasonably good results, for $h = 0.4$, $\alpha = 0.5$ fails to capture the smooth decay in flux to zero, and after only one cell sets the flux to zero. On the other hand, for $\alpha = 0.9$, the flux decays with an exponential profile as expected. Hence, $\alpha = 0.9$ should be selected when using this code, at least for the specific case here with no source - it is possible that changing conditions would make the other choices of α better-suited.

Table 1. Impact on changing α for the solution in terms of negative flux and physical reality. Solutions that appear to display the correct behavior are indicated as "good."

h	$\alpha = -0.9$	$\alpha = -0.5$	$\alpha = 0.25$	$\alpha = 0.5$	$\alpha = 0.9$
0.080	**good**	**good**	**good**	**good**	**good**
0.100	too-sharp drop-off	**good**	**good**	**good**	**good**
0.125	negative ϕ	too-sharp drop-off	**good**	**good**	**good**
0.200	negative ϕ	negative ϕ	**good**	**good**	**good**
0.400	negative ϕ	oscillations	negative ϕ	too-sharp drop-off	**good**

(c): Once a scattering source is present ($\Sigma_s \neq 0$), this scattering source must be updated after each forward and backward sweep. From the form of the transport equation, the scattering source is:

$$S = \int_{-1}^{1} \Sigma_s(\mu^{'} \rightarrow \mu)\psi(\mu^{'})d\mu^{'} \approx \sum_{i=1}^{N} \Sigma_s w_i \psi(\mu_i) = 0.5(\psi(0.2) + \psi(-0.2) + \psi(0.7) + \psi(-0.7)) \tag{32}$$

Because two angles are investigated for this problem, and we are told to use equal weights, $w_i = 0.5$ in order for the integral to normalize to 2 if ψ is not a function of angle.

So, the numerical approach for this case is slightly different from that taken in parts (a) and (b). For this case, in order to have an initial scattering source, an initial guess for the flux must be made (you were right, you do need an initial guess for the flux - my bad!). This initial guess is set to zero for simplicity. Then, a forward and backward sweep are performed. After each double sweep, new values for $\psi(\pm 0.2)$ and $\psi(\pm 0.7)$ are obtained in order to update the scattering source. Then, the process is repeated until reaching convergence.

Convergence is estimated for this method by looking at the norm of the difference in the scattering source between two successive iterations, since once the scattering source is constant, the method only requires one iteration to obtain the correct results. This is equivalent in practice to convergence based on when the flux is constant. Convergence is defined to occur once:

$$\|q^i - q^{i-1}\|_{L^2} \leq 0.01 \tag{33}$$

where i indicates iteration count. Based on the results from part (b), $\alpha = 0.5$ is selected, somewhat arbitrarily, since all three values of α gave non-negative and relatively similar results. I wasn't sure of the problem wording on "then choose one alpha to use with 3 angles" (we're only given two angles?), so I just used the two angles given in the problem. Fig. 8 shows the scalar flux for $\alpha = 0.5$ and $\mu = \pm 0.2, \pm 0.7$. Note that the same boundary conditions from part (a) were kept since no other information was given. As can be seen, the reflective boundary at the right end is obeyed, since the slope at that point is zero. Due to the external and scattering source in the domain, the flux increases towards the reflecting boundary, and decreases at the left edge due to leakage.

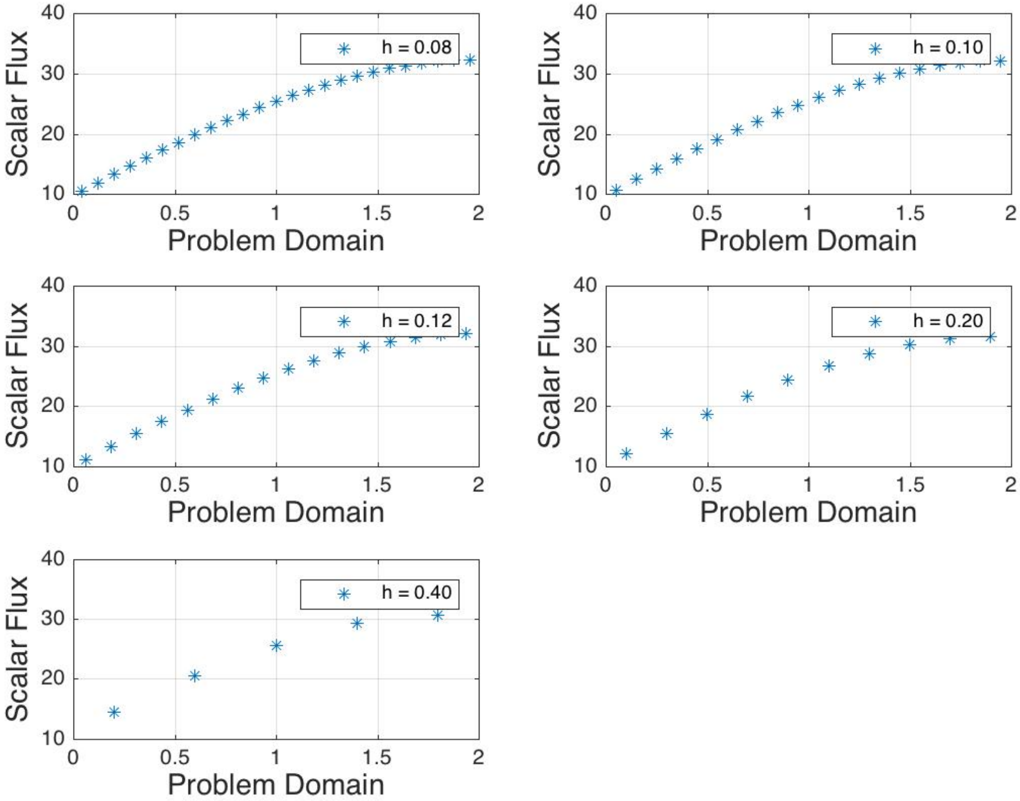

Figure 8. Cell-centered scalar flux for $h = 0.08, 0.1, 0.125, 0.2, 0.4$ and $\alpha = 0.5$ and for $\mu = \pm 0.2, \pm 0.7$.

(d): With these particular specifications, the solve never converges. When stopping the solve at 1000 iterations (much larger than the number, of order 50, required for part (c)), the following flux is obtained. This might be occurring as the scattering to total interaction approaches 1, which means that it becomes very difficult to converge with a finite flux.

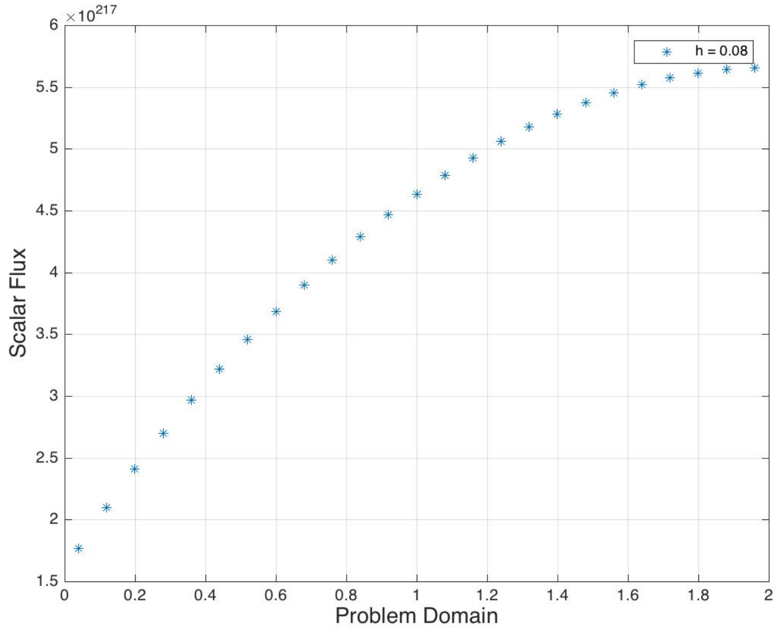

Figure 9. Cell-centered scalar flux for $h = 0.08$ and $\alpha = 0.0$ and for $\mu = \pm 0.2, \pm 0.7$.

④ The 1-D form of the multigroup transport equation is:

$$\frac{\mu_a}{h_i}\left(\psi^g_{a,i+1/2} - \psi^g_{a,i-1/2}\right) + \Sigma^g_{t,i}\psi^g_{a,i} = 2\pi \sum_{a=1}^{N} w_a \underbrace{\sum_{g^{'}=1}^{G} \Sigma_{s,i}(g^{'} \to g, a^{'} \to a)\psi^{g^{'}}_{a^{'},i}}_{\text{in-scattering}} + \underbrace{\frac{\chi_g}{2}\sum_{g^{'}=1}^{G} \nu_{g^{'}}\Sigma^{g^{'}}_{f,i}\phi^{g^{'}}_i}_{\text{fission}} + \frac{1}{2}Q^g_i \tag{34}$$

Writing the general system of equations for five groups ($G = 5$) (each equation applies to a single angular direction μ_a) gives:

$$\begin{aligned}
\frac{\mu_a}{h_i}\left(\psi^1_{a,i+1/2} - \psi^1_{a,i-1/2}\right) + \Sigma^1_{t,i}\psi^1_{a,i} &= 2\pi \sum_{a=1}^{N} w_a \sum_{g^{'}=1}^{2} \Sigma_{s,i}(g^{'} \to 1, a^{'} \to a)\psi^{g^{'}}_{a^{'},i} + \frac{\chi_1}{2}\sum_{g^{'}=1}^{G} \nu_{g^{'}}\Sigma^{g^{'}}_{f,i}\phi^{g^{'}}_i + \frac{1}{2}Q^1_i \\
\frac{\mu_a}{h_i}\left(\psi^2_{a,i+1/2} - \psi^2_{a,i-1/2}\right) + \Sigma^2_{t,i}\psi^2_{a,i} &= 2\pi \sum_{a=1}^{N} w_a \sum_{g^{'}=1}^{2} \Sigma_{s,i}(g^{'} \to 2, a^{'} \to a)\psi^{g^{'}}_{a^{'},i} + \frac{\chi_2}{2}\sum_{g^{'}=1}^{G} \nu_{g^{'}}\Sigma^{g^{'}}_{f,i}\phi^{g^{'}}_i + \frac{1}{2}Q^2_i \\
\frac{\mu_a}{h_i}\left(\psi^3_{a,i+1/2} - \psi^3_{a,i-1/2}\right) + \Sigma^3_{t,i}\psi^3_{a,i} &= 2\pi \sum_{a=1}^{N} w_a \sum_{g^{'}=1}^{5} \Sigma_{s,i}(g^{'} \to 3, a^{'} \to a)\psi^{g^{'}}_{a^{'},i} + \frac{\chi_3}{2}\sum_{g^{'}=1}^{G} \nu_{g^{'}}\Sigma^{g^{'}}_{f,i}\phi^{g^{'}}_i + \frac{1}{2}Q^3_i \\
\frac{\mu_a}{h_i}\left(\psi^4_{a,i+1/2} - \psi^4_{a,i-1/2}\right) + \Sigma^4_{t,i}\psi^4_{a,i} &= 2\pi \sum_{a=1}^{N} w_a \sum_{g^{'}=1}^{5} \Sigma_{s,i}(g^{'} \to 4, a^{'} \to a)\psi^{g^{'}}_{a^{'},i} + \frac{\chi_4}{2}\sum_{g^{'}=1}^{G} \nu_{g^{'}}\Sigma^{g^{'}}_{f,i}\phi^{g^{'}}_i + \frac{1}{2}Q^4_i \\
\frac{\mu_a}{h_i}\left(\psi^5_{a,i+1/2} - \psi^5_{a,i-1/2}\right) + \Sigma^5_{t,i}\psi^5_{a,i} &= 2\pi \sum_{a=1}^{N} w_a \sum_{g^{'}=1}^{5} \Sigma_{s,i}(g^{'} \to 5, a^{'} \to a)\psi^{g^{'}}_{a^{'},i} + \frac{\chi_5}{2}\sum_{g^{'}=1}^{G} \nu_{g^{'}}\Sigma^{g^{'}}_{f,i}\phi^{g^{'}}_i + \frac{1}{2}Q^5_i
\end{aligned} \tag{35}$$

Because neutrons are assumed to not be able to scatter from thermal groups to fast groups, $\Sigma_{s,i}(3,4,5 \to 1,2$ is zero, where the numbers listed as comma-separated values are intended to indicate that, for example,

$\Sigma_{s,i}(3 \rightarrow 1) = 0$, $\Sigma_{s,i}(3 \rightarrow 2) = 0$, $\cdots$, and $\Sigma_{s,i}(5 \rightarrow 2) = 0$. It is assumed that neutrons of any energy levels can induce fission, and the external source contributes neutrons of all energies. Also, while it is typical to assume that no thermal neutrons are born from fission ($\chi_3 = \chi_4 = \chi_5 = 0$), this is not assumed here based on the problem statement.

Because the thermal groups are assumed to upscatter, for groups 3, 4, and 5, neutrons of any group can scatter into groups 3, 4, and 5.

1 Appendix

This section contains all the code used for each question.

1.1 Question 3, parts (a) and (b)

```
% HW4, Question 3
clear all

% logistical things
fontsize = 16;

% material data
mu = 0.1;
Et = 1.0;
Es = 0.0;

for alpha = [-0.9, -0.5, 0.25, 0.5, 0.9]
    plot_num = 1;

    if alpha < 0
        neg = 1;
    else
        neg = 2;
    end

    for h = [0.08, 0.10, 0.125, 0.20, 0.40]

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

        q = 0.0 .* coordinates;

        if (elem_length > 2 * mu / Et)
            sprintf('Warning:␣Possibly␣negative␣flux!')
        end

        % initialize
        psi = zeros(1, length(coordinates));
        psi_n = zeros(1, length(coordinates));
        phi = zeros(1, length(coordinates(2:2:end)));

        psi(1) = 2.0; % incoming flux boundary condition

        % positive sweep
```

```
i = 1;

while i <= (length(coordinates) - 2)
    % compute cell-centered value
    psi(i + 1) = (q(i+1) + (2 / (1 + alpha)) * (abs(mu) * psi(i) ./ elem_length)) / (Et + (2 / (1 + alpha)) * abs(mu) / elem_length);

    % compute out-going value
    psi(i + 2) = (2 / (1 + alpha)) * psi(i + 1) - ((1 - alpha)/(1 + alpha)) * psi(i);

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

% negative sweep - store the outgoing flux to apply periodic BC
i = length(coordinates);
psi_n(i) = psi(end);

while i > 2

    % compute the cell-centered flux
    psi_n(i - 1) = (q(i-1) + (2/(1 - alpha)) * (mu * psi_n(i) / elem_length)) / (Et + 2 * mu / (elem_length * (1 - alpha)));

    % compute the outgoing face flux
    psi_n(i - 2) = (2 / (1 - alpha)) * psi_n(i-1) - ((1+alpha)/(1-alpha)) * psi_n(i);

    i = i - 2;
end


% plot the cell-centered values of angular flux
% plot(coordinates(2:2:end), psi(2:2:end), '*', coordinates(2:2:end), psi_n(2:2:end), 'r*')
% legend('Flux for Positive \mu', 'Flux for Negative \mu')
% xlabel('Problem Domain', 'FontSize', fontsize)
% ylabel('Cell-Centered Angular Flux', 'FontSize', fontsize)
% saveas(gcf, 'AngularFluxh08', 'jpeg')
% close all


% apply "quadrature rule" to get scalar flux
phi = psi(2:2:end) + psi_n(2:2:end); % equal weights of 1

% plot the scalar flux for part a
subplot(3, 2, plot_num)
plot(coordinates(2:2:end), phi, '*')
hold on
grid on
legend(sprintf('h = %.2f', h))
xlabel('Problem Domain', 'FontSize', fontsize -2)
```

```
        ylabel('Scalar␣Flux', 'FontSize', fontsize-2)

        plot_num = plot_num + 1;
    end

    saveas(gcf, sprintf('ScalarFlux%i%i', neg, 100*alpha), 'jpeg')
    close all
end
```

1.2 Question 3, parts (c) and (d)

```
% HW4, Question 3
clear all

% logistical things
fontsize = 16;

% material data
mu = 0.2;
mu07 = 0.7;
Et = 1.0;
Es = 0.9;
tolerance = 0.01;
norm = 1; % intial arbitrary value
num_iterations = 0;
max_iterations = 1000;
plot_num = 1;

for alpha = [0.0]

    for h = [0.08]

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

        % initial guess for the scattering source is simply set to the
        % external source (equivalent to guessing psi = 0.0)
        q = 1.0 .* coordinates;

        % iterate
        while ((norm > tolerance) && (num_iterations < max_iterations))

            if (elem_length > 2 * mu / Et)
                sprintf('Warning:␣Possibly␣negative␣flux!')
            end

            % initialize
            psi = zeros(1, length(coordinates));
            psi_n = zeros(1, length(coordinates));
            psi07 = zeros(1, length(coordinates));
            psi_n07 = zeros(1, length(coordinates));
```

```
psi(1) = 2.0; % incoming flux boundary condition

% positive sweep
i = 1;

while i <= (length(coordinates) - 2)
    % compute cell-centered value
    psi(i + 1) = (q(i+1) + (2 / (1 + alpha)) * (abs(mu) * psi(i)
       ↪ ./ elem_length)) / (Et + (2 / (1 + alpha)) * abs(mu) /
       ↪ elem_length);
    psi07(i + 1) = (q(i+1) + (2 / (1 + alpha)) * (abs(mu07) *
       ↪ psi07(i) ./ elem_length)) / (Et + (2 / (1 + alpha)) *
       ↪ abs(mu07) / elem_length);

    % compute out-going value
    psi(i + 2) = (2 / (1 + alpha)) * psi(i + 1) - ((1 - alpha)/(1
       ↪ + alpha)) * psi(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_n(i) = psi(end);
psi_n07(i) = psi07(end);

while i > 2

    % compute the cell-centered flux
    psi_n(i - 1) = (q(i-1) + (2/(1 - alpha)) * (mu * psi_n(i) /
       ↪ elem_length)) / (Et + 2 * mu / (elem_length * (1 - alpha
       ↪ )));
    psi_n07(i - 1) = (q(i-1) + (2/(1 - alpha)) * (mu07 * psi_n07(i
       ↪ ) / elem_length)) / (Et + 2 * mu07 / (elem_length * (1 -
       ↪  alpha)));

    % compute the outgoing face flux
    psi_n(i - 2) = (2 / (1 - alpha)) * psi_n(i-1) - ((1+alpha)/(1-
       ↪ alpha)) * psi_n(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 scattering source, and add the external source
q_new = zeros(1, length(psi));
q_new = Es .* 0.5 .* (psi + psi_n + psi07 + psi_n07) + 1.0.*
   ↪ coordinates;
```

```
            % 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
        phi = 0.5 .* (psi(2:2:end) + psi_n(2:2:end) + psi07(2:2:end) + psi_n07(2:2:end));

        % plot the scalar flux
        %subplot(3, 2, plot_num)
        plot(coordinates(2:2:end), phi, '*')
        hold on
        grid on
        legend(sprintf('h = %.2f', h))
        xlabel('Problem Domain', 'FontSize', fontsize-2)
        ylabel('Scalar Flux', 'FontSize', fontsize-2)
    end
end

saveas(gcf, 'ScalarFlux_partD', 'jpeg')
%close all
```