Homework 5-NE 255

Information

Author: Mario I. Ortega
Organization: Department of Nuclear Engineering, University of California, Berkeley

1 Problem 1

We consider the operator form of the Transport Equation

$$\mathbf{L}\psi = \mathbf{MS}\phi + \mathbf{M}q_e \tag{1.1}$$

$$\phi = \mathbf{D}\psi \tag{1.2}$$

with the following discretizations:

- 3 energy groups
- P_2 (Number of moments equal to $(N+1)^2 \rightarrow (2+1)^2 = 9$ moments)
- S_2 (Number of angles equal to $N(N+2) \rightarrow 2(2+2) = 8$ angles)
- $4 \times 4 \times 4$ spatial mesh (27 cells)
- Diamond Difference (one unknown per cell).

1.1 (a)

The dimensions of each matrix in Equation (1.18) are given in terms of two parameters, α and β, given by the following equations

$$\alpha = G \times n \times c \times u \tag{1.3}$$

$$\beta = G \times N \times c \times u \tag{1.4}$$

where

- G = number of energy groups,
- N = number of moments,
- n = number of angular unknowns
- c = number of cells
- u = number of unknowns per cell.

For our specific set of parameters, we obtain the following matrix sizes:

$$dim(\mathbf{L}) = (\alpha \times \alpha) = (648 \times 648) \tag{1.5}$$

$$dim(\psi) = (\alpha \times 1) = (648 \times 1) \tag{1.6}$$

$$dim(\mathbf{M}) = (\alpha \times \beta) = (648 \times 729) \tag{1.7}$$

$$dim(\mathbf{S}) = (\beta \times \beta) = (729 \times 729) \tag{1.8}$$

$$dim(\phi) = (\beta \times 1) = (729 \times 1) \tag{1.9}$$

$$dim(q_e) = (\beta \times 1) = (729 \times 1). \tag{1.10}$$

1.2 (b)

$$[\mathbf{M}]_{gg} = \begin{pmatrix} Y_{00}^e(\hat{\Omega}_1) & Y_{10}^e(\hat{\Omega}_1) & Y_{11}^o(\hat{\Omega}_1) & Y_{11}^e(\hat{\Omega}_1) & Y_{20}^e(\hat{\Omega}_1) & \cdots & Y_{99}^o(\hat{\Omega}_1) & Y_{99}^e(\hat{\Omega}_1) \\ Y_{00}^e(\hat{\Omega}_2) & Y_{10}^e(\hat{\Omega}_2) & Y_{11}^o(\hat{\Omega}_2) & Y_{11}^e(\hat{\Omega}_2) & Y_{20}^e(\hat{\Omega}_2) & \cdots & Y_{99}^o(\hat{\Omega}_2) & Y_{99}^e(\hat{\Omega}_2) \\ Y_{00}^e(\hat{\Omega}_3) & Y_{10}^e(\hat{\Omega}_3) & Y_{11}^o(\hat{\Omega}_3) & Y_{11}^e(\hat{\Omega}_3) & Y_{20}^e(\hat{\Omega}_3) & \cdots & Y_{99}^o(\hat{\Omega}_3) & Y_{99}^e(\hat{\Omega}_3) \\ Y_{00}^e(\hat{\Omega}_4) & Y_{10}^e(\hat{\Omega}_4) & Y_{11}^o(\hat{\Omega}_4) & Y_{11}^e(\hat{\Omega}_4) & Y_{20}^e(\hat{\Omega}_4) & \cdots & Y_{99}^o(\hat{\Omega}_4) & Y_{99}^e(\hat{\Omega}_4) \\ Y_{00}^e(\hat{\Omega}_5) & Y_{10}^e(\hat{\Omega}_5) & Y_{11}^o(\hat{\Omega}_5) & Y_{11}^e(\hat{\Omega}_5) & Y_{20}^e(\hat{\Omega}_5) & \cdots & Y_{99}^o(\hat{\Omega}_5) & Y_{99}^e(\hat{\Omega}_5) \\ Y_{00}^e(\hat{\Omega}_6) & Y_{10}^e(\hat{\Omega}_6) & Y_{11}^o(\hat{\Omega}_6) & Y_{11}^e(\hat{\Omega}_6) & Y_{20}^e(\hat{\Omega}_6) & \cdots & Y_{99}^o(\hat{\Omega}_6) & Y_{99}^e(\hat{\Omega}_6) \\ Y_{00}^e(\hat{\Omega}_7) & Y_{10}^e(\hat{\Omega}_7) & Y_{11}^o(\hat{\Omega}_7) & Y_{11}^e(\hat{\Omega}_7) & Y_{20}^e(\hat{\Omega}_7) & \cdots & Y_{99}^o(\hat{\Omega}_7) & Y_{99}^e(\hat{\Omega}_7) \\ Y_{00}^e(\hat{\Omega}_8) & Y_{10}^e(\hat{\Omega}_8) & Y_{11}^o(\hat{\Omega}_8) & Y_{11}^e(\hat{\Omega}_8) & Y_{20}^e(\hat{\Omega}_8) & \cdots & Y_{99}^o(\hat{\Omega}_8) & Y_{99}^e(\hat{\Omega}_8) \end{pmatrix} \tag{1.11}$$

$$\mathbf{S} = \begin{pmatrix} [\mathbf{S}_{11}] & [\mathbf{S}_{12}] & [\mathbf{S}_{13}] \\ [\mathbf{S}_{21}] & [\mathbf{S}_{22}] & [\mathbf{S}_{23}] \\ [\mathbf{S}_{31}] & [\mathbf{S}_{32}] & [\mathbf{S}_{33}] \end{pmatrix} \tag{1.12}$$

$$[\mathbf{S}_{21}] = \begin{pmatrix} \Sigma_{s0}^{21} & 0 & \cdots & 0 \\ 0 & \Sigma_{s1}^{21} & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma_{s8}^{21} \end{pmatrix} \tag{1.13}$$

$$\psi = \begin{pmatrix} [\psi]_1 & [\psi]_2 & [\psi]_3 \end{pmatrix}^T \tag{1.14}$$

$$[\psi]_1 = \begin{pmatrix} \psi_1^1 & \psi_2^1 & \psi_3^1 & \psi_4^1 & \psi_5^1 & \psi_6^1 & \psi_7^1 & \psi_8^1 \end{pmatrix}^T \tag{1.15}$$

1.3 (c)

$$\mathbf{D} = \mathbf{M}^T\mathbf{W} \tag{1.16}$$

$$\mathbf{D} = \begin{pmatrix} Y_{00}^e(\hat{\Omega}_1) & Y_{10}^e(\hat{\Omega}_1) & Y_{11}^o(\hat{\Omega}_1) & Y_{11}^e(\hat{\Omega}_1) & Y_{20}^e(\hat{\Omega}_1) & \cdots & Y_{99}^o(\hat{\Omega}_1) & Y_{99}^e(\hat{\Omega}_1) \\ Y_{00}^e(\hat{\Omega}_2) & Y_{10}^e(\hat{\Omega}_2) & Y_{11}^o(\hat{\Omega}_2) & Y_{11}^e(\hat{\Omega}_2) & Y_{20}^e(\hat{\Omega}_2) & \cdots & Y_{99}^o(\hat{\Omega}_2) & Y_{99}^e(\hat{\Omega}_2) \\ Y_{00}^e(\hat{\Omega}_3) & Y_{10}^e(\hat{\Omega}_3) & Y_{11}^o(\hat{\Omega}_3) & Y_{11}^e(\hat{\Omega}_3) & Y_{20}^e(\hat{\Omega}_3) & \cdots & Y_{99}^o(\hat{\Omega}_3) & Y_{99}^e(\hat{\Omega}_3) \\ Y_{00}^e(\hat{\Omega}_4) & Y_{10}^e(\hat{\Omega}_4) & Y_{11}^o(\hat{\Omega}_4) & Y_{11}^e(\hat{\Omega}_4) & Y_{20}^e(\hat{\Omega}_4) & \cdots & Y_{99}^o(\hat{\Omega}_4) & Y_{99}^e(\hat{\Omega}_4) \\ Y_{00}^e(\hat{\Omega}_5) & Y_{10}^e(\hat{\Omega}_5) & Y_{11}^o(\hat{\Omega}_5) & Y_{11}^e(\hat{\Omega}_5) & Y_{20}^e(\hat{\Omega}_5) & \cdots & Y_{99}^o(\hat{\Omega}_5) & Y_{99}^e(\hat{\Omega}_5) \\ Y_{00}^e(\hat{\Omega}_6) & Y_{10}^e(\hat{\Omega}_6) & Y_{11}^o(\hat{\Omega}_6) & Y_{11}^e(\hat{\Omega}_6) & Y_{20}^e(\hat{\Omega}_6) & \cdots & Y_{99}^o(\hat{\Omega}_6) & Y_{99}^e(\hat{\Omega}_6) \\ Y_{00}^e(\hat{\Omega}_7) & Y_{10}^e(\hat{\Omega}_7) & Y_{11}^o(\hat{\Omega}_7) & Y_{11}^e(\hat{\Omega}_7) & Y_{20}^e(\hat{\Omega}_7) & \cdots & Y_{99}^o(\hat{\Omega}_7) & Y_{99}^e(\hat{\Omega}_7) \\ Y_{00}^e(\hat{\Omega}_8) & Y_{10}^e(\hat{\Omega}_8) & Y_{11}^o(\hat{\Omega}_8) & Y_{11}^e(\hat{\Omega}_8) & Y_{20}^e(\hat{\Omega}_8) & \cdots & Y_{99}^o(\hat{\Omega}_8) & Y_{99}^e(\hat{\Omega}_8) \end{pmatrix}^T \begin{pmatrix} w_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & w_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & w_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & w_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & w_6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & w_7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & w_8 \end{pmatrix} \tag{1.17}$$

1.4 (d)

We usually do not from the matrix $\mathbf{L}$ due to its large size and storage requirements. We can invert the matrix using the transport sweep algorithm that gives us the action of the matrix on the vector without having to explicit invert the matrix.

Table 1: My caption

g / g'	1	2	3
1	0.1	0.0	0.0
2	0.3	0.1	0.1
3	0.1	0.3	0.3

1.5 (e)

Starting from

$$\mathbf{L}\psi = \mathbf{MS}\phi + \mathbf{M}q_e \tag{1.18}$$

$$\phi = \mathbf{D}\psi \tag{1.19}$$

we invert and substitute Equation (1.19) into Equation (1.18).

$$\mathbf{LD}^{-1}\phi = \mathbf{MS}\phi + \mathbf{M}q_e \tag{1.20}$$

$$\mathbf{LD}^{-1}\phi - \mathbf{MS}\phi = \mathbf{M}q_e \tag{1.21}$$

$$(\mathbf{LD}^{-1} - \mathbf{MS})\phi = \mathbf{M}q_e \tag{1.22}$$

$$\phi = (\mathbf{LD}^{-1} - \mathbf{MS})^{-1}\mathbf{M}q_e \tag{1.23}$$

2 Problem 2

The three group scalar fluxes can be seen in Figure 2.1. For this problem, the boundary conditions consist of 0.5 for the incoming group one angular flux on the left and reflective boundary conditions on the right boundary. Cross section values used in the problems are given below. We also used the parameter $\alpha = 0.5$. We see that group one scalar flux has the largest magnitude due to the smaller absorption cross section and source in the group.

$$\Sigma_{tg} = [0.5, 0.8, 1.0] \tag{2.1}$$

$$q_{eg} = [1.5, 0.0, 0.2] \tag{2.2}$$

2.1 Problem 2 MATLAB Functions and Scripts

```
function [xi,scalar_flux] = OneDDiscreteOrdinates(mu,wi,h,NEgrps,alpha,L,sigt,sigs,qex,tol)

xii = 0:h:L;
xi = h/2:h:L;

N = length(xi);

half_angular_flux = zeros(length(mu),length(xii),NEgrps);
angular_flux = zeros(length(mu),length(xi),NEgrps);

half_angular_flux(1:length(mu)/2,1,1) = 0.5;

for iter = 1:1000

```

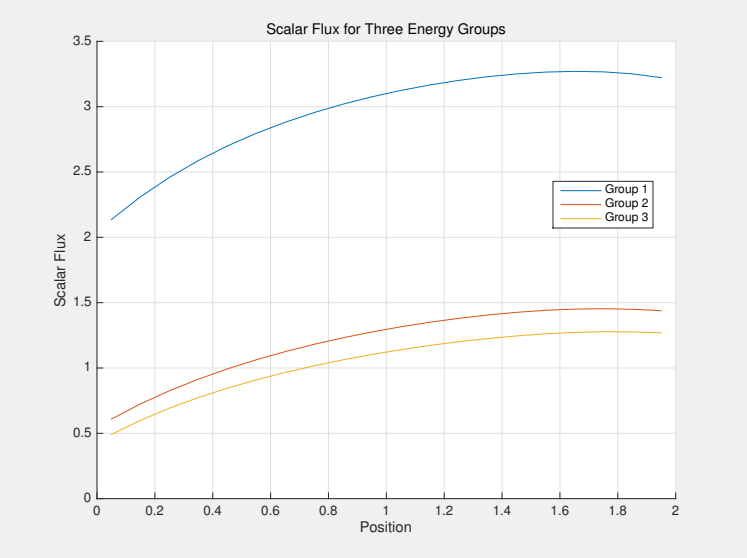

Figure 2.1: Energy Scalar Flux for $\alpha = 0.5$

```
    %scalar_flux_old = Calculate_ScalarFlux(xi,wi,NEgrps,angular_flux);
    scalar_flux_old_total = Calculate_ScalarFlux(xi,wi,NEgrps,angular_flux);

    for e = 1:NEgrps

    for InIter = 1:1000;

        scalar_flux_old = Calculate_ScalarFlux(xi,wi,NEgrps,angular_flux);

    for j = 1:length(mu)

        if ( mu(j) > 0 )

            %Set boundary condition on angular flux
            %half_angular_flux(j,1,1) = 0.5;

            for k = 1:length(xi)

                qscat = 0;

                for srcE = 1:NEgrps

                for l = 1:length(wi)

                    qscat = qscat + (1/2)*sigs(e,srcE)*wi(l)*angular_flux(l,k,srcE);

                end

                end

                q = (1/1)*qscat + qex(e);

                angular_flux(j,k,e) = (2*abs(mu(j))/(1+alpha) + sigt(e)*h)^(-1)*(h*q + ...
                    abs(mu(j))*half_angular_flux(j,k,e)*(1 + (1-alpha)/(1+alpha)));

                half_angular_flux(j,k+1,e) = (2/(1+alpha))*angular_flux(j,k,e) - ...
                    ((1-alpha)/(1+alpha))*half_angular_flux(j,k,e);

            end

```

```
        elseif ( mu(j) < 0 )

            %Set boundary condition on angular flux
            half_angular_flux(j,N+1,e) = half_angular_flux(length(wi)+1-j,N+1,e);

            for k = length(xi):-1:1

                qscat = 0;

                for srcE = 1:NEgrps

                for l = 1:length(wi)

                    qscat = qscat + (1/2)*sigs(e,srcE)*wi(l)*angular_flux(l,k,srcE);

                end

                end

                q = (1/1)*qscat + qex(e);

                angular_flux(j,k,e) = (2*abs(mu(j))/(1+alpha) + sigt(e)*h)^(-1)*(h*q + ...
                    abs(mu(j))*half_angular_flux(j,k+1,e)*(1 + (1-alpha)/(1+alpha)));

                half_angular_flux(j,k,e) = (2/(1-alpha))*angular_flux(j,k,e) - ...
                    ((1+alpha)/(1-alpha))*half_angular_flux(j,k+1,e);

            end

        end

    end

    scalar_flux = Calculate_ScalarFlux(xi,wi,NEgrps,angular_flux);

    norm_flux = sqrt(sum((scalar_flux(e,:) - scalar_flux_old(e,:)).^2));

    fprintf('Energy Group %i, Iteration: %i, Norm: %f \n',e,InIter,norm_flux);

    if ( norm_flux < tol)

        break

    end

    end

    end

    norm_flux_total = sqrt(sum(scalar_flux - scalar_flux_old_total).^2);

     if ( norm_flux_total < tol)

        break

    end

end

return
```

```
%NE255 Homework 5
%Problem 2

clc, clear, clf, close all

```

```
%Geometry Information
x0 = 0.0;
x1 = 2.0;

h = 0.1;
tol = 1e-4;
alpha = 0.5;

%Number of Energy Groups
Nenergy = 3;

%Angular Discretization
mu = [0.7 0.5 0.2 -0.2 -0.5 -0.7];
wi = [1/3 1/3 1/3 1/3 1/3 1/3];

%Cross Section Data
sigt = [0.5 0.8 1.0];
%sigt = [1 0 0]
sigs = [0.1 0.0 0.0;
        0.3 0.1 0.1;
        0.1 0.3 0.3];
%sigs = zeros(3,3); sigs(1) = 0.5;

%Source
qex = [1.5 0.0 0.2];
%qex = [1 0 0];

[xi,scalar_flux] = OneDDiscreteOrdinates(mu,wi,h,Nenergy,alpha,x1,sigt,sigs,qex,tol);

for i = 1:Nenergy
    %figure(i)
    hold on
    plot(xi,scalar_flux(i,:))
    grid on
    xlabel('Position')
    ylabel('Scalar Flux')
    titl = sprintf('Scalar Flux for Three Energy Groups',i);
    title(titl);
end

legend('Group 1','Group 2','Group 3','Location','Best');
```