The analytic solution and the error between the numeric and analytic solutions for the four schemes with four mesh resolutions are shown in Figure 1. For this condition, the problem is dominated by diffusion over advection and hence, the gradients are nearly constant along the domain. Because of this, the numerical schemes can capture well the behavior of the function, even with a very coarse grid. The first order upwind scheme has high levels of error compared to the higher order schemes, though it is still negligible. Similar trends hold in Figure 2, where the Peclet number is still low (0.1) and hence the problem is dominated by diffusion. For a Peclet number of 1, shown in Figure 3, where the advection is of the same order as the diffusion, the gradient in the solution increases significantly in the streamwise direction, and hence the more complex schemes (QUICK and exponential) are better able to capture its behavior than either of the upwind schemes, though the second order upwind scheme has comparable errors to the QUICK and exponential schemes for the finer meshes. As the Peclet number is increased to 10 in Figure 4, and the advection terms begin to dominate over the diffusion terms, the effects of the different numerical differencing schemes becomes more apparent. Now, both the upwind schemes have non-negligible errors near the right boundary. Additionally, for the coarse mesh, even the QUICK scheme results in a noticeable error at the boundary. For a Peclet number of 100 (Figure 5), significant errors occur at the right boundary for the upwind and QUICK schemes; on the coarsest mesh this error approaches 100% of the analytic solution. Even on the finest mesh, the errors for the upwind and QUICK schemes are non-negligible, due to the very high gradients in this case. It should be noted that the exponential scheme produces errors on the order of machine precision error regardless of the Peclet number or the number of grid points used. This is due to the fact that the exponential scheme represents a linear system of equations which are analytic solutions to the governing equation at each cell. Hence, the exponential scheme is analytic. Based off of these results, it is clear that the choice of numeric scheme and grid resolution is highly dependent on the Peclet number (Reynolds number is fluid flow problems). For low Reynolds number flows, simple first or second order upwind schemes and a coarse mesh can be sufficient. However, for high Reynolds number flow, a more complex differencing scheme and/or higher grid resolution may be needed even for flows where turbulence is negligible.

|  |  |
| --- | --- |
| Z:\My Documents\Matlab\Coursework\ME 811\HW 3\Phia Pe0_01.png | |
| Z:\My Documents\Matlab\Coursework\ME 811\HW 3\N20 Pe0_01.png | Z:\My Documents\Matlab\Coursework\ME 811\HW 3\N40 Pe0_01.png |
| Z:\My Documents\Matlab\Coursework\ME 811\HW 3\N80 Pe0_01.png | Z:\My Documents\Matlab\Coursework\ME 811\HW 3\N160 Pe0_01.png |

Figure 1: Analytic solution and numerical errors for 20, 40, 80 and 160 respectively, for a Peclet number of 0.01.

|  |  |
| --- | --- |
|  | |
|  |  |
|  |  |

Figure 2: Analytic solution and numerical errors for 20, 40, 80 and 160 respectively, for a Peclet number of 0.1.

|  |  |
| --- | --- |
|  | |
|  |  |
|  |  |

Figure 3: Analytic solution and numerical errors for 20, 40, 80 and 160 respectively, for a Peclet number of 1.

|  |  |
| --- | --- |
|  | |
|  |  |
|  |  |

Figure 4: Analytic solution and numerical errors for 20, 40, 80 and 160 respectively, for a Peclet number of 10.

|  |  |
| --- | --- |
|  | |
|  |  |
|  |  |

Figure 5: Analytic solution and numerical errors for 20, 40, 80 and 160 respectively, for a Peclet number of 100.

# Appendix

function [phia] = H3P1(N,Pe)

%Code completed by Michael Crawley for ME 811 HW#3

x = 1/(2\*N):1/N:1-1/(2\*N);

dx = mean(diff(x));

bound0 = 0;

bound1 = 1;

gamma = 1;

rhou = Pe\*gamma;

%%Analytic

C = (exp(Pe)-1)^-1;

phia = C\*(exp(Pe\*x)-1);

%%First order upwind

X0 = repmat(rhou+2\*gamma/dx,N,1);

X0([1 end]) = rhou+4\*gamma/dx; %apply boundary conditions

Xn1 = repmat(-0.5\*(abs(rhou)+rhou)-gamma/dx,N,1);

X1 = repmat(-0.5\*(abs(rhou)-rhou)-gamma/dx,N,1);

% i = 1 boundary conditions

Xn1(1) = 0;

X1(1) = X1(1)-gamma/dx/3;

% i = N boundary conditions

X1(end) = 0;

Xn1(end) = Xn1(end)-gamma/dx/3;

X0(end) = 0.5\*(abs(rhou)-rhou)+4\*gamma/dx;

X1 = circshift(X1,1);

Xn1 = circshift(Xn1,-1);

A = spdiags([Xn1 X0 X1], -1:1,N,N);

b = zeros(N,1);

b(1) = (0.5\*(abs(rhou)+rhou)+8\*gamma/dx/3)\*bound0; %apply boundary condition at x = 0

b(end) = (0.5\*(abs(rhou)-rhou)-0.5\*(abs(rhou)+rhou)+8\*gamma/dx/3)\*bound1; %apply boundary condition at x = 1

phiFUD = TDMsolver(A,b);

%%Second order upwind

b = zeros(N,1);

X0 = repmat(0.75\*(abs(rhou)+rhou)+0.75\*(abs(rhou)-rhou)+2\*gamma/dx,N,1);

Xn1 = repmat(-(0.25\*(abs(rhou)+rhou)+0.75\*(abs(rhou)+rhou)+gamma/dx),N,1);

X1 = repmat(-(0.25\*(abs(rhou)-rhou)+0.75\*(abs(rhou)-rhou)+gamma/dx),N,1);

Xn2 = repmat(0.25\*(abs(rhou)+rhou),N,1);

X2 = repmat(0.25\*(abs(rhou)-rhou),N,1);

%i = 1 boundary conditions

X0(1) = abs(rhou)+rhou+4\*gamma/dx;

Xn1(1) = 0;

X1(1) = -(0.75\*(abs(rhou)-rhou)+4\*gamma/dx/3);

Xn2(1) = 0;

b(1) = -(abs(rhou)+rhou+0.5\*(abs(rhou)-rhou)+8\*gamma/dx/3)\*bound0;

%i = 2 boundary conditions

Xn1(2) = -(1.25\*(abs(rhou)+rhou)+gamma/dx);

Xn2(2) = 0;

b(2) = -0.5\*(abs(rhou)+rhou)\*bound0;

%i = N boundary conditions

X0(end) = (abs(rhou)-rhou)+4\*gamma/dx;

Xn1(end) = -(0.75\*(abs(rhou)+rhou)+4\*gamma/dx/3);

X1(end) = 0;

X2(end) = 0;

b(end) = (abs(rhou)-rhou-0.5\*(abs(rhou)+rhou)+8\*gamma/dx/3)\*bound1;

%i = N-1 boundary conditions

X1(end-1) = -(1.25\*(abs(rhou)-rhou)+gamma/dx);

X2(end-1) = 0;

b(end-1) = -0.5\*(abs(rhou)-rhou)\*bound1;

X1 = circshift(X1,1);

X2 = circshift(X2,2);

Xn1 = circshift(Xn1,-1);

Xn2 = circshift(Xn2,-2);

A = spdiags([Xn2 Xn1 X0 X1 X2],-2:2,N,N);

phiSUD = A\b;

%%QUICK

b = zeros(N,1);

X0 = repmat((3/16)\*(abs(rhou)+rhou)+(3/16)\*(abs(rhou)-rhou)+2\*gamma/dx,N,1);

Xn1 = repmat(-((1/16)\*(abs(rhou)+rhou)+(3/8)\*(abs(rhou)+rhou)-(3/16)\*(abs(rhou)-rhou)+gamma/dx),N,1);

X1 = repmat((3/16)\*(abs(rhou)+rhou)-(3/8)\*(abs(rhou)-rhou)-(1/16)\*(abs(rhou)-rhou)-gamma/dx,N,1);

Xn2 = repmat((1/16)\*(abs(rhou)+rhou),N,1);

X2 = repmat((1/16)\*(abs(rhou)-rhou),N,1);

%i = 1 boundary conditions

X0(1) = 0.5\*(abs(rhou)+rhou)-(3/16)\*(abs(rhou)-rhou)+4\*gamma/dx;

Xn1(1) = 0;

X1(1) = (1/6)\*(abs(rhou)+rhou)-(3/8)\*(abs(rhou)-rhou)-4\*gamma/dx/3;

Xn2(1) = 0;

b(1) = ((1/6)\*(abs(rhou)+rhou)+rhou+8\*gamma/dx/3)\*bound0;

%i = 2 boundary conditions

X0(2) = (3/8)\*(abs(rhou)+rhou)-(3/16)\*(abs(rhou)-rhou)-(1/6)\*(abs(rhou)+rhou)+(3/8)\*(abs(rhou)-rhou)+2\*gamma/dx;

Xn1(2) = -(1/16)\*(abs(rhou)+rhou)-0.5\*(abs(rhou)+rhou)+(3/16)\*(abs(rhou)-rhou)-gamma/dx;

Xn2(2) = 0;

b(2) = -((1/6)\*(abs(rhou)+rhou))\*bound0;

%i = N boundary conditions

X0(end) = 0.5\*(abs(rhou)-rhou)-(3/16)\*(abs(rhou)+rhou)+4\*gamma/dx;

Xn1(end) = (1/6)\*(abs(rhou)-rhou)-(3/8)\*(abs(rhou)+rhou)-4\*gamma/dx/3;

X1(end) = 0;

X2(end) = 0;

b(end) = ((1/6)\*(abs(rhou)-rhou)-rhou+8\*gamma/dx/3)\*bound1;

%i = N-1 boundary conditions

X0(end-1) = (3/8)\*(abs(rhou)+rhou)-(1/6)\*(abs(rhou)-rhou)-(3/16)\*(abs(rhou)+rhou)+(3/8)\*(abs(rhou)-rhou)+2\*gamma/dx;

X1(end-1) = -(1/16)\*(abs(rhou)-rhou)-0.5\*(abs(rhou)-rhou)+(3/16)\*(abs(rhou)+rhou)-gamma/dx;

X2(end-1) = 0;

b(end-1) = -((1/6)\*(abs(rhou)-rhou))\*bound1;

X1 = circshift(X1,1);

X2 = circshift(X2,2);

Xn1 = circshift(Xn1,-1);

Xn2 = circshift(Xn2,-2);

A = spdiags([Xn2 Xn1 X0 X1 X2],-2:2,N,N);

phiQUICK = A\b;

%%Exponential

Pel = Pe\*dx;

fp = Pel/(exp(Pel)-1);

fm = Pel\*exp(Pel)/(exp(Pel)-1);

b = zeros(N,1);

X0 = repmat(gamma\*fm/dx+gamma\*fp/dx,N,1);

Xn1 = repmat(-gamma\*fm/dx,N,1);

X1 = repmat(-gamma\*fp/dx,N,1);

%i = 1 boundary conditions

X0(1) = gamma\*fm/dx+gamma\*Pel/dx/(exp(Pel/2)-1);

Xn1(1) = 0;

b(1) = (rhou+gamma\*Pel/dx/(exp(Pel/2)-1))\*bound0;

%i = N boundary conditions

X0(end) = gamma\*Pel\*exp(Pel/2)/dx/(exp(Pel/2)-1)+gamma\*fp/dx;

X1(end) = 0;

b(end) = (-rhou+gamma\*Pel\*exp(Pel/2)/dx/(exp(Pel/2)-1))\*bound1;

X1 = circshift(X1,1);

Xn1 = circshift(Xn1,-1);

A = spdiags([Xn1 X0 X1], -1:1,N,N);

phiEXP = TDMsolver(A,b);

%%Plot Results

h(1) = figure;

plot(x,phia);xlabel('x');ylabel('\Phi\_a');title(['Analytic Solution for Pe = ',num2str(Pe)]);

h(2) = figure;

plot(x,phiFUD'-phia,'k',x,phiSUD'-phia,'-sk',x,phiQUICK'-phia,'--k',x,phiEXP'-phia,'-.k');

legend('1^s^t Order Upwind','2^n^d Order Upwind', 'QUICK','Exponential','Location','Best');

xlabel('x');

ylabel('\Phi\_n-\Phi\_a');

title(['Error in numerical solution for N = ',num2str(N),', Pe = ', num2str(Pe)]);

saveas(h(2),['N',num2str(N),' Pe',strrep(num2str(Pe),'.','\_')],'fig');

saveas(h(2),['N',num2str(N),' Pe',strrep(num2str(Pe),'.','\_')],'png');

saveas(h(1),['Phia Pe',strrep(num2str(Pe),'.','\_')],'fig');

saveas(h(1),['Phia Pe',strrep(num2str(Pe),'.','\_')],'png');

close(h);

end