THE COLLEGE OF WILLIAM AND MARY

Homework 8 Reflection

Alexander Powell

April 16, 2015

1 REFLECTION

The video on lesson four continues to descibe step 1 of the Navier-Stokes equation. Basically, they describe an ititial profile and apply a space-time discretization to it. Numerical schemes like forward and backward difference methods are applied to the discretization (time using forward difference and space using backwards). By calculating the transpose, we are able to obtain values t_{n+1} from t_n values, thus making it an iterative method, easily implemented in a language like MATLAB.
The next part of the video presents step 2 of Navier-Stokes equation. The only difference in the discretization is changing the constant c into u_i^n.
Step 3 introduces the one dimension diffusion formula. Also, according to the video, the physics of diffusion is isotrophic in nature, and for that reason the finite difference method that represents it best is the central difference method (combination of forward and backward differences). In step 3, she uses a forward difference in time, and a central difference in space. Again, the equation is discretized and written as an iterative scheme by taking the transpose.
Step 4 outlines Burgers' equation, and again the strategy is the same as the previous steps. In this step, the plotted numerical results follow a periodic pattern.

Typo Warning: They typo warnings in the video were referencing the boundary conditions for steps 1 and 2. The video stated that $u = 0$ for the interval $(0,2)$ but it was corrected to be $u = 1$ for $(0,2)$.

2 STEP 2 NONLINEAR CONVECTION CODE

The following MATLAB code is an implementation of step 2 non-linear convection.

```
function Step2( )
%initial condition
nx = 20;
dx = 2 / (nx - 1);
nt = 25; %250 for nonlin
dt = 0.025; %0.0025 for nonlin
c = 1;
lamda = abs(c * dt/dx)
u = ones(nx, 1);
u(0.5/dx : 1/dx + 1) = 2;
plot(u);

frame_counter = 1;
for n = 1:nt
    un = u;
    for i = 2:nx-1
            % The only difference occurs right below this comment.
        % The Constant c is replaced with un(i).
        u(i) = un(i) - un(i) * (dt/dx) * (un(i) - un(i-1));
    end
    plot(linspace(0,2,nx),u);
    F(frame_counter) = getframe;
    frame_counter = frame_counter + 1;
end
end
```