Ay190 – Worksheet 11
Daniel DeFelippis
Date: February 18, 2014

Advection Equation

We are solving

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = 0$$

Using various schemes. The function we are advecting is a Gaussian, starting the peak at $x = 30$.

1

To implement the moving Gaussian, we must, in part, define a function to apply the outflow boundary conditions, which is done with the following two lines of code:

```
y[0] = x[1]
y[-1] = x[-2]
```

which sets the boundary values of the array "y" (0th and last terms) equal to the last interior points of the array "x" (1st and second to last terms).

2

The main lines of code used for implementing the upwind scheme are

```
for j in range(1, n-1):
    yupwind[j] = yold_up[j] - \
    (v*dt/dx)*(yold_up[j] - yold_up[j-1])
```

which define each interior gridpoint for timestep $n + 1$, and

```
apply_bcs(yold_up, yupwind)
```

which uses the boundary condition function to define the outmost gridpoints at timestep $n + 1$.

Since $v = \Delta x = 0.1$, $v\Delta t/\Delta x = c_{CFL}$, where $c_{CFL} \leq 1$. So, in implementing the upwind scheme, we can change c_{CFL} to see when it is stable. We find that if $c_{CFL} = 0$, the scheme doesn't move the Gaussian from its initial position, while if $c_{CFL} = 1$, the scheme immediately makes the Gaussian equal to 0 everywhere. If it is in between those values, the Gaussian is moved to the right, although both the peak height is reduced and the width is increased over time. The closer c_{CFL} is to 0, the slower both of those processes happen. If we increase c_{CFL} so it is greater than 1, we get oscillations in the peak height that increase very rapidly. If we make $c_{CFL} < 0$, we quickly get an oscillatory pattern where the Gaussian should be, as well as a very large and quickly increasing peak at around $x = 5$, where the scheme should remain essentially 0. So, the scheme is stable only between 0 and 1 as expected. For all subsequent plots, we choose $c_{CFL} = 0.5$.

Defining the error to be the difference in height between the peaks of the analytic solution and the upwind scheme solution at a given timestep, we can make a plot of the error of the scheme over time. This is shown in figure 1. We only look at time t up to 350 (even though the trial ran to $t = 500$) because after about $t = 350$, the peak of the scheme's Gaussian went past $x = 100$ which is the maximum value shown in the graph, which means the error very rapidly went to 1 in a way inconsistent with the trend shown in the graph.

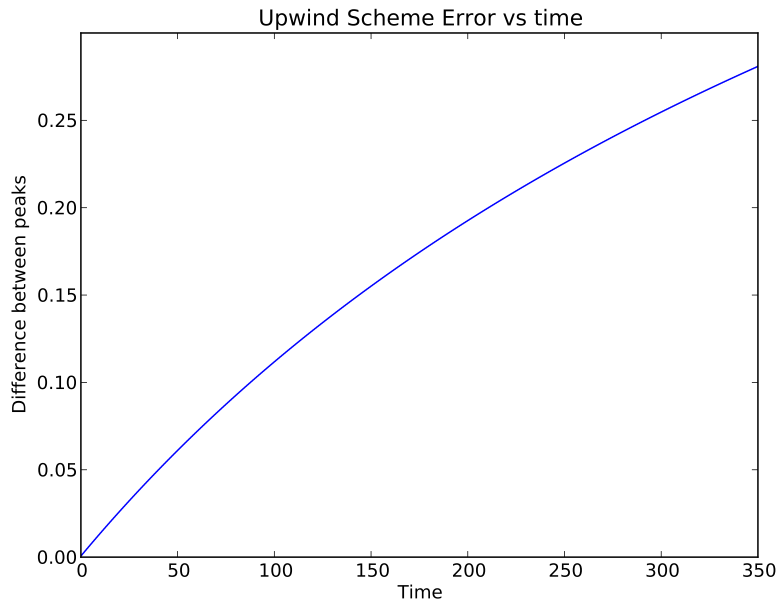

Figure 1: Upwind scheme error for $\sigma = \sqrt{15}$ vs time.

To see what happens to the error if we change the analytic form of the Gaussian, we replace $\sigma = \sqrt{15}$ with one 5 times smaller, $\sigma = \sqrt{15}/5$. Running the same code, we see that the with a smaller σ, the scheme's Gaussian drops off more quickly, achieving larger errors at smaller times. However, as shown in figure 2, it stll moves at the same speed, reaching the boundary of $x = 100$ at the same time as before.

Both of the errors here are clearly not linear. To see what form the might have, we make a log-log plot and notice that, for smaller t, the error is linear on that plot with a slope of about 4/5. So, for smaller t, the error goes as $t^{4/5}$.

For larger t, a plot of error vs log(t) (figure 4 looks more linear for larger t, indicating that the error goes as the log of the time for larger t.

3

I'm confused about the implementation of the unstable FTCS scheme. I used the lines

```
for j in range(1, n-1):

    yftcs[j] = yold_ftcs[j] - \
    (v*dt/(2.*dx))*(yold_ftcs[j+1] - yold_ftcs[j-1])
```

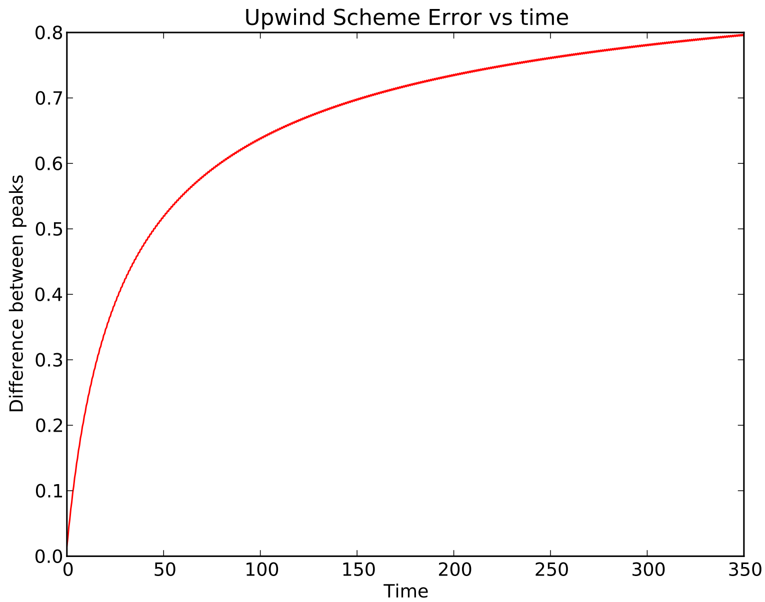

Figure 2: Upwind scheme error for $\sigma = \sqrt{15}/5$ vs time

and

```
apply_bcs(yold_ftcs, yftcs)
```

like before, but the scheme only seemed to maintain the same peak height and overall width as the analytic solution, but just move forward at a faster speed. I didn't notice any instability except even at large times. However, I did notice, for the smaller sigma, some instability in the FTCS scheme: like the upwind scheme when $c_{CFL} < 0$, the gaussian became oscillatory after many iterations.

4

Implementing the Lax-Friedrich Method with the lines

```
for j in range(1, n-1):

    ylax[j] = 0.5*(yold_lax[j+1] + yold_lax[j-1]) - \
    (v*dt/(2.*dx))*(yold_lax[j+1] - yold_lax[j-1])
```

and

```
apply_bcs(yold_lax, ylax)
```

we get a very similar result that the upwind scheme gives. The behavior of a decreasing peak and inreasing width is the same, which we can see by comparing figure 5 to figure 1. The graphs look almost identical. However, the Lax-Friedrich method reaches almost twice the error in half the time. That is, its peak decreases in height twice as fast, and it moves to the right at twice the speed.

Figure 3: Log-log plot of Upwind scheme error for $\sigma = \sqrt{15}/5$ vs time

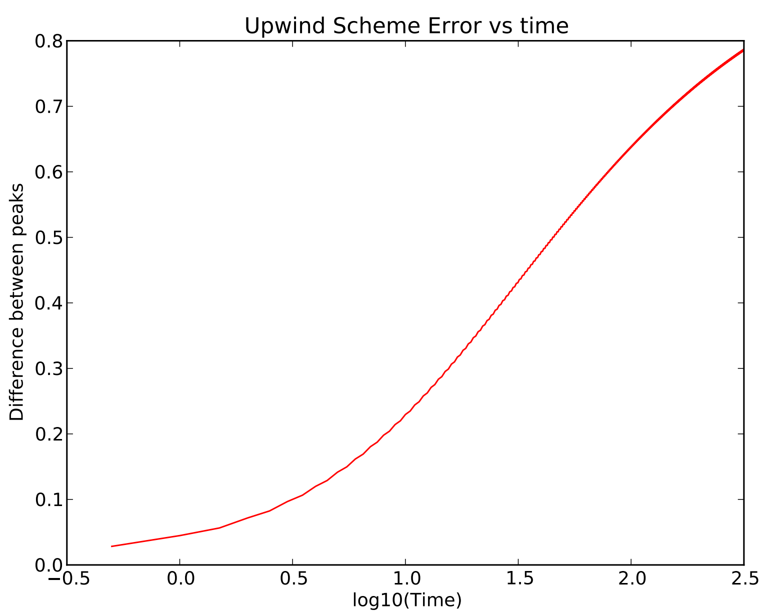

Figure 4: Upwind scheme error for $\sigma = \sqrt{15}/5$ vs log(time)

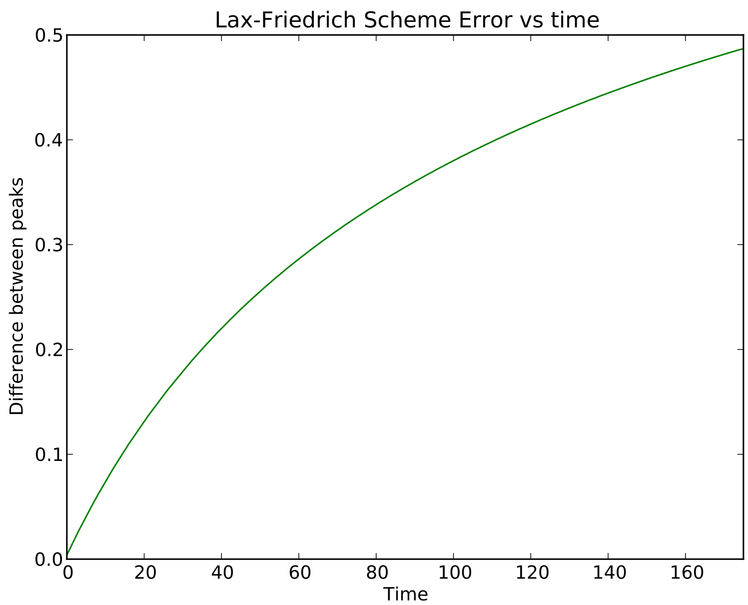

Figure 5: Lax-Friedrich scheme error for $\sigma = \sqrt{15}$ vs time