Draft version May 13, 2015
Preprint typeset using LaTeX style emulateapj v. 12/16/11

ASTRO 702 FINAL PROJECT

Zachary J. Pace
Department of Astronomy, University of Wisconsin-Madison, Madison, WI 53706
Draft version May 13, 2015

ABSTRACT

I explore distinctions between finite-difference and finite-volume schemes, and their abilities to propagate shocks/discontinuities in fluid flows. The full codebase used can be found at my Github page, but the results are summarized below.

1. FINITE-DIFFERENCE SCHEMES

I slightly modified the code I wrote for the midterm project, to work with Burgers' equation:

$$\delta_t u + u \delta_x u = 0 \tag{1}$$

Burgers' equation resembles the linear advection equation, but the speed at any given point is proportional to the quantity present at that point. I constructed FTBS, FTCS, and FTFS schemes in order to propagate the initial conditions for the so-called "rising" and "falling" cases. The schemas themselves are implemented in the functions `propagate_FTBS`, `propagate_FTCS`, and `propagate_FTFS`, respectively, in `burgers.py`. The results are shown in Figure 1:

$$f(x) = \begin{cases} 0 & : x < -1 \\ \frac{x+1}{2} & : -1 < x < 1 \\ 1 & : x > 1 \end{cases} \quad \text{(rising)} \tag{2}$$

$$f(x) = \begin{cases} 0 & : x < -1 \\ \frac{1-x}{2} & : -1 < x < 1 \\ 1 & : x > 1 \end{cases} \quad \text{(falling)} \tag{3}$$

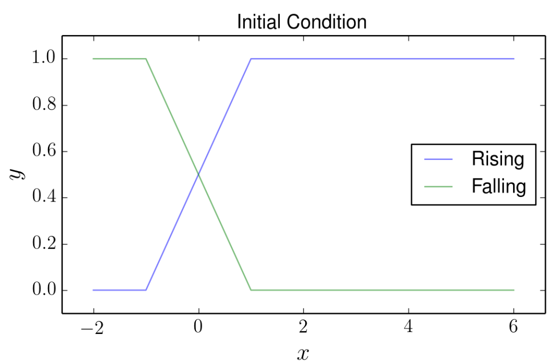

Figure 1. The two Initial Conditions used to test finite-difference and finite-volume schemes. In *blue* is the "rising" case, and in *green* is the "falling" case.

This setup necessitates careful attention to boundary conditions, which are used for determining the 0^{th} and -1^{th} elements in the $(n+1)^{\text{th}}$ timestep. Instead of periodic boundary conditions, where the -1^{th} element is used for the 0^{th} element's propagation (and vice-versa), each is propagated by its own value. This is equivalent to doubling the first and last cells for the purposes of propagation.

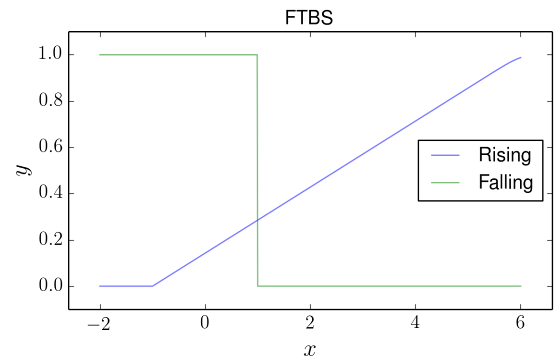

Figure 2. FTBS propagation of the ICs (*blue*: rising; *green*: falling). The IC is stably propagated through $t = 5$.

1.1. *Q1*

> For each of the three schemes, determine whether the scheme is stable or unstable. In light of the results of the stability analyses you undertook in the midterm project, are your findings here expected or surprising?

The results of FTBS, FTCS, and FTFS analysis are given in Figure 2, Figure 3, and Figure 4, through $t = 5$. Note that FTBS is the only stable scheme (i.e., both FTCS and FTFS quickly blow up either in part or in full).

In the midterm project, we discovered that so-called "upstream" methods were stable, and "downstream" methods unstable. For a flow in the positive direction, the terms "upstream" and "downstream" refer to whether the values in timestep $n+1$ for the k^{th} cell are propagated from the $(k-1)^{\text{th}}$ cell or the $(k+1)^{\text{th}}$ cell. The cases are opposite for flows in the negative direction. In this case, we also found that the upstream case (FTBS, for positive flow speed) was stable.

1.2. *Q2*

> For the stable scheme (there should be only one!), plot y(x) snapshots for the rising and falling initial states at $t = 0$, $t = 1$, $t = 2$ and $t = 4$. Compare these snapshots against the analytic solutions presented in 9.2 of the Lecture Notes. Specifically,

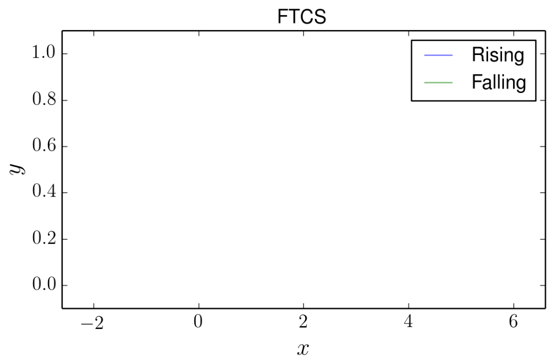

Figure 3. FTCS propagation of the ICs (*blue*: rising; *green*: falling). The FTCS scheme makes the solution blow up everywhere (the script used to propagate the ICs shows overflow errors at a very early time).

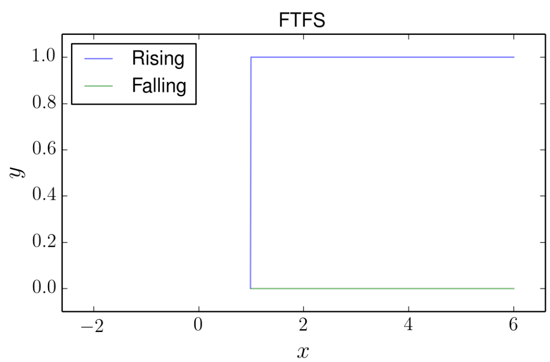

Figure 4. FTFS propagation of the ICs (*blue*: rising; *green*: falling). The FTFS scheme is only stable in part of the domain (and only for the "rising" case), and blows up everywhere else.

1. For the rising initial state, measure the propagation speeds of the head of the ramp (initially at $x = 1$) and the tail of the ramp (initially at $x = 1$). Compare these against the characteristic speeds of the head and tail, shown in Fig. 9.5 of the Notes how good is the match?
2. For the falling initial state, confirm that a flow discontinuity develops at $x = 1$, $t = 2$, as predicted in Fig. 9.10 of the Notes.
3. Again for the falling initial state, determine the propagation speed of the discontinuity for $t > 2$. Youll notice that the speed is different from the $S = 1/2$ shown in Fig. 9.10 (and derived in the accompanying text). With reference to the details of the numerical scheme, can you explain why this is?

In Figure 5, we indeed see that the falling case presents a flow discontinuity. Also, we note that the rising case's base does not move at all, but the peak moves with a speed of 1, though it does spread out (so does the other concave-down piecewise boundary, at the head of the falling case). However, the discontinuity in the falling case does not propagate forward (i.e., it has a speed of zero). This would seem to be an artifact of the backwards-space scheme: in particular, this is because in the FTBS scheme, $y_k^{n+1} = y_k^n + C(y_{k+1}^n - y_k^n)$, where the Courant number C is defined as the inverse of the "code speed" $(s_{code})^{-1} = \frac{\Delta t}{\Delta x}$ times the flow speed, which in this case is equal to the value in the cell, y_k^n. Since to the right of the discontinuity, the value is zero, the shock never propagates forward (because the next cell "doesn't know" about the shock!)

2. FINITE-VOLUME SCHEMES

I implemented Godunov's method in the `godunov` function in burgers.py.

2.1. *Q3*

> Plot $\bar{y}(x)$ snapshots for the rising and falling initial states at $t = 0$, $t = 1$, $t = 2$ and $t = 4$. Following the same three steps as in Q2, compare these snapshots against the analytic solutions presented in Section 9.2 of the Lecture Notes. Is the propagation speed of the discontinuity now correctly reproduced?

Figure 6 shows the results of Godunov's method as applied to both the rising and falling cases. By comparing the bottom-left and bottom-right panes ($t = 2$ and $t = 4$, respectively), we find that $S = \frac{\Delta x}{\Delta t} = 1/2$. This correctly reproduces the analytic result.

2.2. *Q4*

> For both initial states, calculate the integrated quantity
>
> $$Y(t) \equiv \int_{y_0}^{y_f} \bar{y}(t)dx = \sum_{k=0}^{K-1} \bar{y}(t)\Delta x \tag{4}$$
>
> at each time step. This represents the total amount of the quantity y in the calculation domain. Plot $Y(t)$ as a function of time can you explain why it decreases (increases) steadily with time for the rising (falling) initial state, and why in particular $|\frac{dY}{dt}| = 1/2$?

Figure 7 shows the amount of material $Y(t)$, as defined in Equation 4, through time, for both the rising and falling cases. The slope $\frac{dY}{dt}$ is $-1/2$ for the falling case and $1/2$ for the rising case. The latter can be explained by the system being "fed" more material from the LHS, which piles up behind the discontinuity, while in the former case, the material to the right of the slope is escaping through the RHS boundary at a rate of 1 t^{-1}, while the expansion of the slope is increasing the material at a rate of 1/2 t^{-1}, for a total rate of change of $(\frac{dY}{dt})_{rising} = -1/2$.

2.3. *Q5*

> A restriction on the application of Godunovs method is that, across a time step of extent Δt, the discontinuity from one interface shouldnt be able to reach one of the neighboring interfaces (otherwise, eqn. 4.18—from assignment text—becomes invalid). Use this restriction to determine an upper limit on the time steps used in your code. Confirm (with suitable plots) that when run with Δt above this limit, on either initial state, your code becomes unstable.

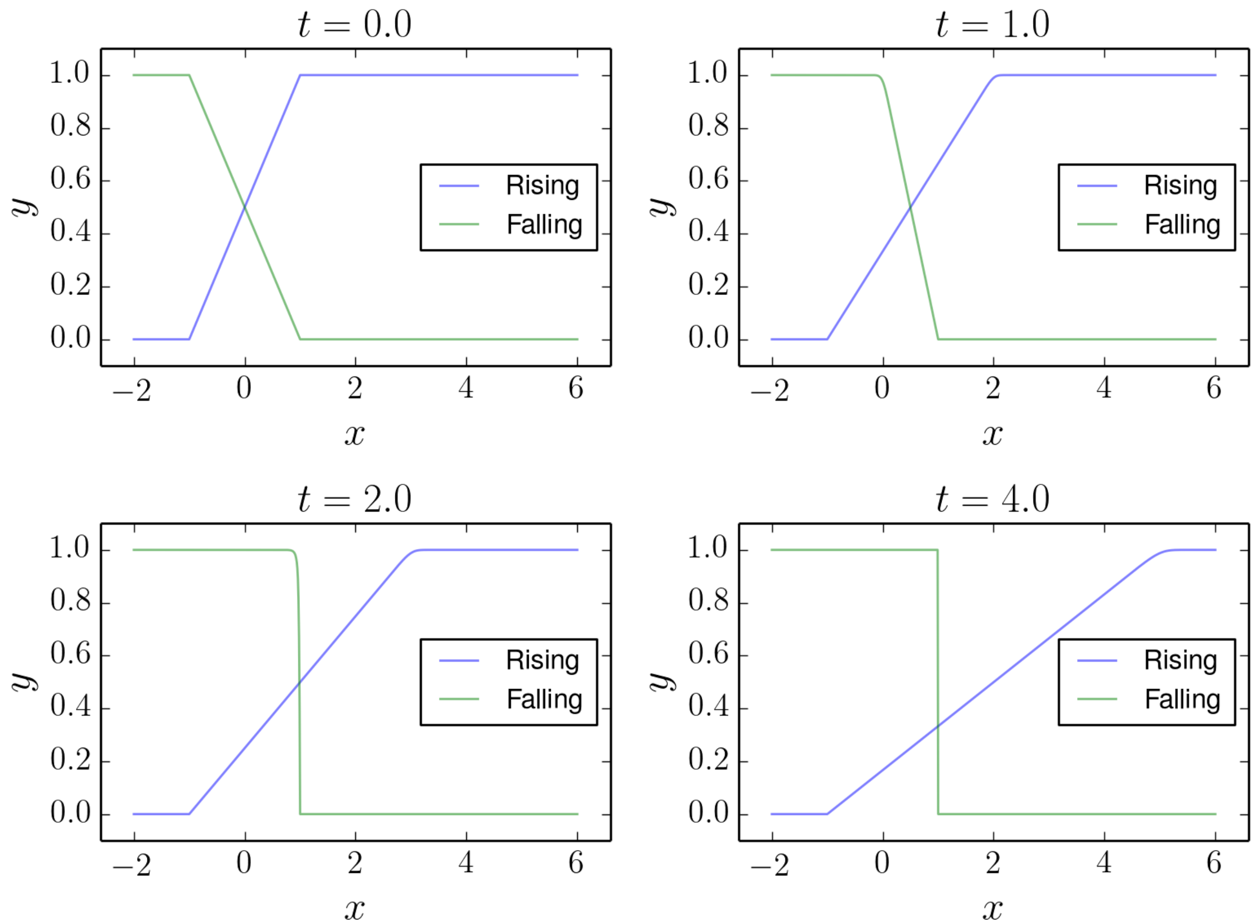

Figure 5. FTBS propagation results for $t = 0$, $t = 1$, $t = 2$, and $t = 4$, with the rising case in *blue* and the falling case in *green*.

When $\Delta x > \Delta t$, information from one particular cell can transport beyond the next cell, which causes instabilities. This is equivalent to demanding dense spatial sampling, compared to a typical flow speed $\bar{a}$, times the time step Δt. Figure 8 shows both IC cases at various timesteps with $\Delta t = 1.25\ \Delta x$. The instabilities take hold very quickly in all cases, and for the rising case (solid lines), we can see the exact x-cell that begins to be affected in each timestep: in particular, the instability moves left as t increases from 0 to 5 (in order: blue, red, purple, black, green, cyan). For the falling case, the simulation immediately becomes unstable and blows up. As a general rule, we can conclude that cells with higher values become unstable faster. This agrees with the notion that flow speed goes with the value in the cell, and so higher-valued cells will need less time to “catch” up to the next boundary.

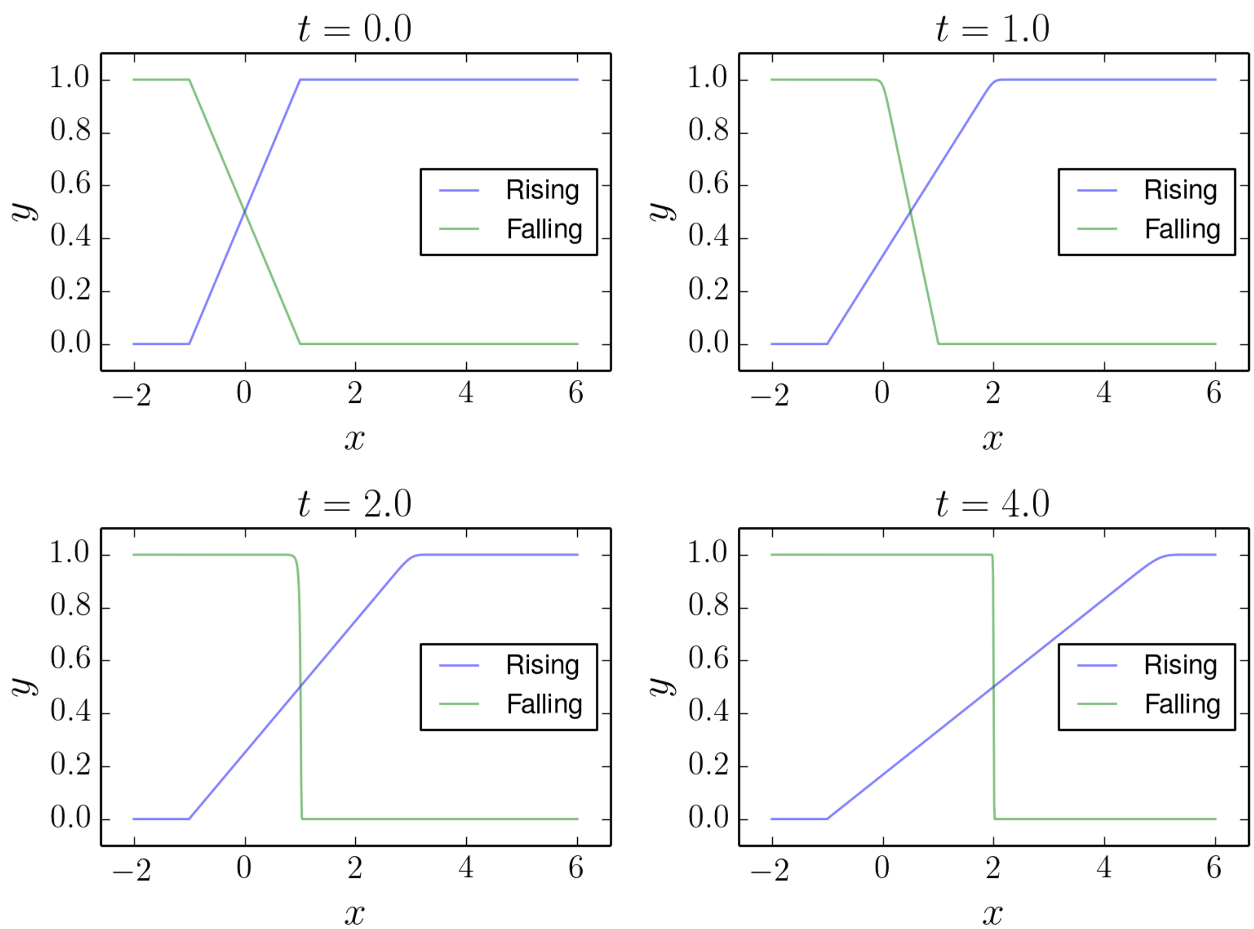

Figure 6. Godunov's method propagation for $t = 0$, $t = 1$, $t = 2$, and $t = 4$, with the rising case in *blue* and the falling case in *green*.

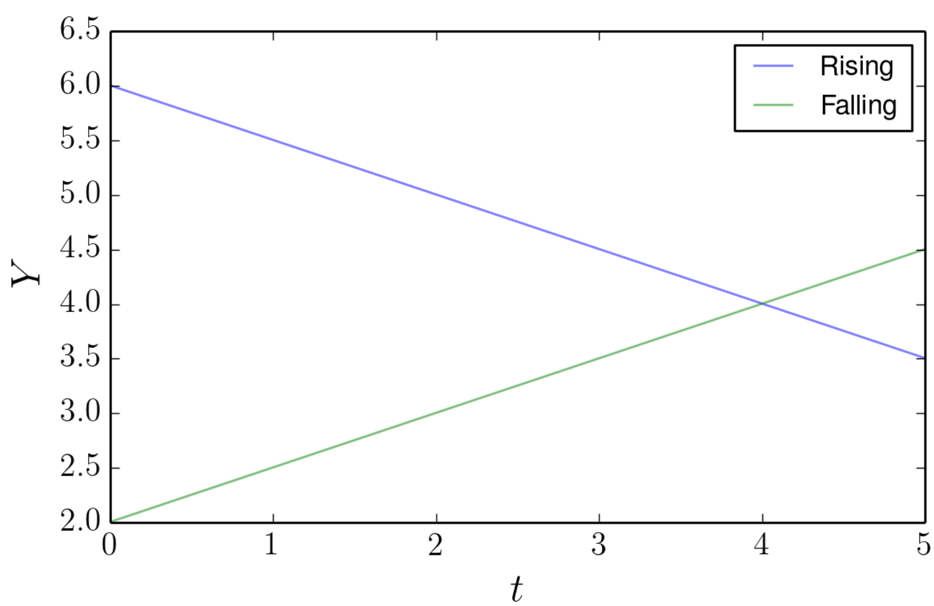

Figure 7. The total quantity $Y(t)$ plotted against t, with the rising case in *blue* and the falling case in *green*.

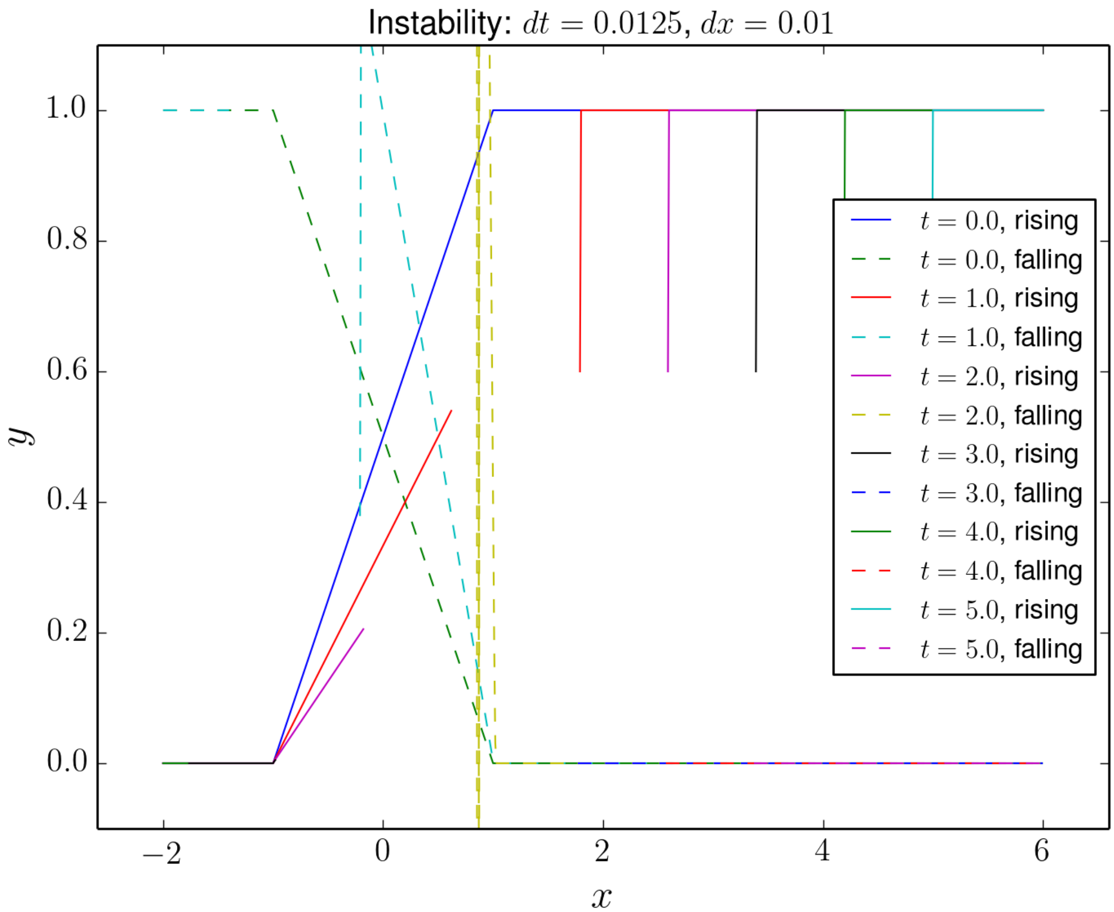

Figure 8. The propagation of instabilities as time increases, with $\Delta t = 1.25\ \Delta x$. The solid lines are the rising case, and the dashed lines are the falling case.