Finite Difference Scheme Stability for the Non-linear Shallow Water Equations

Joey Foster

November 3, 2024

1 Setup and motivation

Modelling the evolution of a fluid system is, in-general, exceedingly challenging, with numerous strategies devised, both mathematical and computational, to overcome such difficulties. In this report, we focus on implementing a model of the so-called 'shallow water equations' (SWE) and solve them numerically using a finite difference method.

The full equations can be see in, for example, [Durran, 2010, equations (1.25)-(1.27)] and, collapsing to one space-dimension, take the form

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + g\frac{\partial h}{\partial x} = 0 \tag{1}$$

$$\frac{\partial h}{\partial t} + u\frac{\partial h}{\partial x} + h\frac{\partial u}{\partial x} = 0, \tag{2}$$

where $(t, x) \in \mathbb{R}_{\geq 0} \times \mathbb{R}$ are the usual time and space coordinates, although we shall not be so bold with our domain when solving the system numerically. Additionally, we have the relevant functions to be solved for, of which $u(t, x)$ is the velocity in the x-direction, and $h(t, x)$ is the height of the water above an assumed flat bedrock at $z = 0$ (assuming the water occupies the $x - z$ plane). Lastly, there is of course g taking the value of local gravitational acceleration, and taken to be 9.81 ms^{-2} for the duration of this report.

For a well-posed PDE problem, we will of course also need some boundary and initial conditions. While the latter will be the subject of much discussion moving forward, we will simply assume periodic boundary conditions over our selected numerical solving domain for ease of numerical implementation.

While these equations are self-admittedly a simplification of the most complete physical description, they nonetheless have been used extensively to test or predict physical phenomena. For example, [Brocchini and Dodd, 2008] used these equations for coastal engineering modeling – a clearly applicable use case due to the titular shallow water of coastlines. [Constantin and Escher, 1998] considered the impact of breaking waves as 'finite-time blow up' solutions to these equations and [Courtier and Geleyn, 1988] even used them to test a global weather prediction model. There is also contemporary research in this direction, such as [Khorbatly, 2024]'s study of the 'highly non-linear' shallow water equations.

2 A somewhat naive first attempt

2.1 The Forward-in-Time-Backward-in-Space scheme

To solve the system (1)-(2) numerically, we try one of the simplest discretisation schemes first. Namely, the Forward-in-Time-Backward-in-Space scheme, or FTBS for short.

Consider an arbitrary function $\phi(t, x)$. Denote its discretisation at grid point j and time level n by $\phi_j^{(n)}$. Then first order temporal and spatial derivatives in this scheme become

$$\frac{\partial \phi_j^{(n)}}{\partial t} = \frac{\phi_j^{(n+1)} - \phi_j^{(n)}}{\Delta t}; \qquad \frac{\partial \phi_j^{(n)}}{\partial x} = \frac{\phi_j^{(n)} - \phi_{j-1}^{(n)}}{\Delta x} \tag{3}$$

for time and space steps Δt and Δx respectively. Note that we would expect any implementation of such a scheme to have errors that grow linearly with Δt and Δx as the construction of such derivative approximations neglects terms of order $\Delta(\cdot)^2$ and higher [Weller, 2024, Exercise 2.3(1)].

If we apply the FTBS scheme (3) to our SWE system (1)-(2), we readily arrive at

$$u_j^{(n+1)} = u_j^{(n)} - \frac{\Delta t}{\Delta x}\left[u_j^{(n)}\left(u_j^{(n)} - u_{j-1}^{(n)}\right) + g\left(h_j^{(n)} - h_{j-1}^{(n)}\right)\right] \tag{4}$$

$$h_j^{(n+1)} = h_j^{(n)} - \frac{\Delta t}{\Delta x}\left[u_j^{(n)}\left(h_j^{(n)} - h_{j-1}^{(n)}\right) + h_j^{(n)}\left(u_j^{(n)} - u_{j-1}^{(n)}\right)\right], \tag{5}$$

that is, update formulae for the next timestep in terms of solely the current timestep. This attribute, a so-called *explicit* scheme[a] [Weller, 2024, page 6], is one of the reasons we selected this method to start with. This is arguably the simplest scheme to implement in code.

Our reason for selecting a Backward-in-Space discretisation was two-fold. Firstly, as BS schemes are designed to work[b] with positive velocity, Forward-in-Space (FS) doing the opposite, we had cautiously hoped we could split our scheme such that when u is positive, we apply FTBS, and when u is negative, we apply FTFS. That turned out to not be so simple, but this was our motivation. The other reason considered simply came down to the matter of imposing boundary conditions. A Centered-in-Space scheme comes with an additional boundary point to be dealt with due to the inability to run a loop to the very end of the spatial domain. Granted, dealing with this point is an almost trivial matter but still something we elected to avoid if possible (at least initially).

Unfortunately, the benefits of cheap computational implementation come at the cost of not just rapid error growth ($\mathcal{O}(\Delta t, \Delta x)$), but in this case *unconditional instability*. [Durran, 2010, page 142, problem 8] gives the task of proving that FTBS 'must be unstable' for the linearised SWE. Of course we are dealing with the non-linear SWE but analysing the behaviour of a simpler model is always a sensible place to start.

2.2 Expected behaviour

To devise a test for this first attempt, we turn to [LeVeque, 2002, section 13.1] and consider writing the SWE system (1)-(2) in the form

$$\frac{\partial}{\partial t}\begin{pmatrix} u \\ h \end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix} \frac{1}{2}u^2 + gh \\ uh \end{pmatrix} = 0. \tag{6}$$

[a] Technically, an explicit scheme only requires the next timestep to be defined in terms of any previous timesteps, not just the current one. An example of which would be CTBS (Centered-in-Time) where $\phi^{(n)}$ is computed from $\phi^{(n+1)}$ and $\phi^{(n-1)}$.

[b] A rigorous notion of what it means for a scheme to 'work' with a PDE will follow shortly.

Note that the equations written in this form implicitly assume u and h are smooth. We will accept this limitation for now but if further insight into a formulation that admits discontinuous derivatives is desired, see [LeVeque, 2002, equations (13.5)-(13.7) and the discussion immediately following].

To determine whether our scheme will 'work' in a rigorous sense, we must consider the eigenvalues of the associated Jacobian matrix, as a comparison of their relative signs (and in general, their real and imaginary parts) provides categorisation for the stability of such a system. See, for example, [Morgan, 2017, table at the top of page 77 (the table is not indexed)] although strictly speaking, this analysis is done in the 'theory of ODEs' sense where the notion of stability takes on a slightly different meaning.

For our problem, the associated Jacobian matrix is

$$\begin{pmatrix} u & g \\ h & u \end{pmatrix} \tag{7}$$

with corresponding eigenvalues

$$\begin{vmatrix} u-\lambda & g \\ h & u-\lambda \end{vmatrix} = 0 \implies (u-\lambda)^2 - gh = 0 \implies \lambda_{1,2} = u \pm \sqrt{gh}. \tag{8}$$

Note that our Jacobian matrix takes a slightly different form to [LeVeque, 2002, equation 13.8] due to our a priori smoothness assumption, and yet, our eigenvalues match with [LeVeque, 2002, equation 13.9] suggesting this criterion for stability is independent of the smoothness of the initial data.

From the form of $\lambda_{1,2}$, we see they are obviously always real-valued (as one would expect given the hyperbolic nature of the system), but more notably, they both have the ability to take either sign, depending on the relative sizes of u and $\sqrt{gh}$. If we recall that information propagates along characteristics with speeds given by the eigenvalues of the system [Durran, 2010, discussion between equations (1.6) and (1.7)], and that FTBS assumes a positive wave speed propagation, then in conjunction with the aforementioned instability of the linearised equations, we expect this first attempt to be always unstable.

2.3 Results

As we see from figure 1, the numerical solution becomes unstable at the left boundary point after only a handful of timesteps. Note that the four time slices plotted in figure 1, and in all future figures, do not represent the first four timesteps but rather a representative sample to clearly depict the observed phenomenon. For the precise details, see [Foster, 2024].

Looking at figure 1, one may conclude the instability is arising purely due to a problem at the left boundary. Note, for example, that the selected initial condition for h, namely $h(0,x) = 1 + e^{-5x^2}$ is not a periodic function and hence, while its value at the left and right boundaries agree in this case, its derivative at the same two points do not. One may be forgiven for thinking this is solely (or at least overwhelming) the cause for the observed instability. However, while this is certainly not ideal for a well-posed periodic boundary value problem, we see in figure 2 that this issue is at least not a leading order effect.[c]

Figure 2 shows us that even with a periodic initial condition, the solution becomes unstable and blows up. This time it takes roughly quadruple the number of timesteps to observe this, but still far below any threshold that could reasonably be called 'temporarily stable'.

[c] We will in-fact maintain our Gaussian-based initial condition, despite its flaw, for the rest of this report. This is in part due to its qualitatively pleasing evolution, but mostly due to the numerous other causes of potential instability that so-completely dominate our attention. Relative to these other challenges, this curiosity is of no consequence.

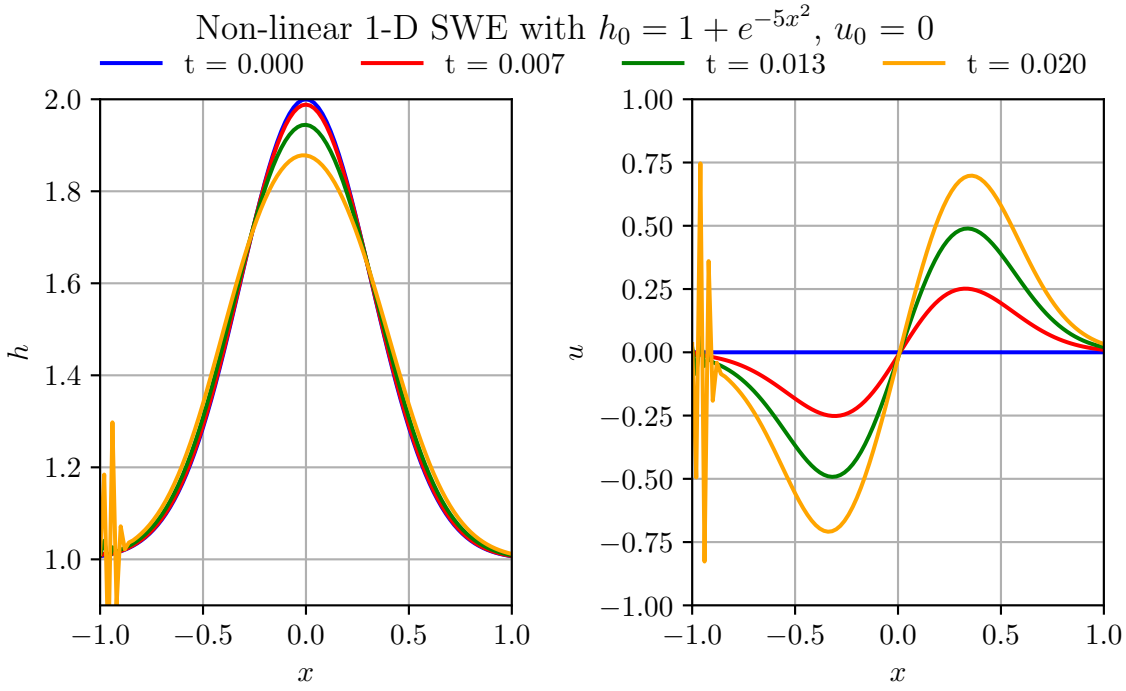

Figure 1: Numerical solution to the system (1)-(2) using the FTBS finite difference scheme over the domain $x \in [-1, 1]$ discretised with $\Delta x = 0.01$ and timestepping obeying (11) with initial conditions $h(0, x) = h_0, u(0, x) = u_0$ and periodic boundary conditions.

Satisfied that this implementation is doing as expected, but no closer to a scheme of reasonable stability, we move on to our second attempt.

3 A slightly cheating second attempt

The cause of our difficulties in the previous section was fundamentally due to the eigenvalues of the system having opposite signs. In our final attempt, we will tackle this problem in a more rigorous and far more generalisable way, but for this attempt we opt to employ a slight cheat. Namely, restricting the set of available problems we can solve to those with a specialised family of initial conditions on u.

Observe the obvious fact that

$$\lambda_2 = u - \sqrt{gh} < u + \sqrt{gh} = \lambda_1 \tag{9}$$

for all u and h. Therefore, if we initialise u such that $\lambda_2 > 0 \ \forall t \geq 0$, every characteristic will propagate information with positive velocity and hence, a necessary condition for FTBS stability will be satisfied.

3.1 The CFL condition

It may surprise readers with expertise in the field to realise we have got this far in our discussion and analysis without mentioning the famous (or perhaps infamous) Courant-Friedrichs-Lewy (CFL) condition. This gives a constraint on the maximum size of the timestep Δt in terms of the grid

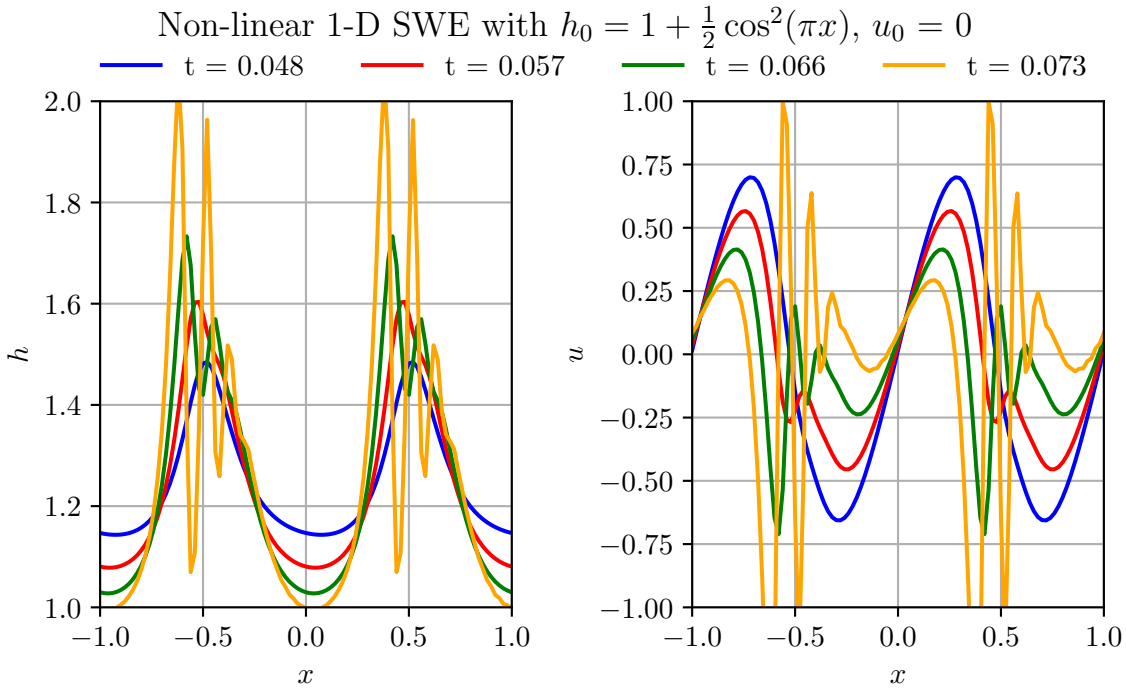

Figure 2: Numerical solution to the system (1)-(2) using the FTBS finite difference scheme over the domain $x \in [-1, 1]$ discretised with $\Delta x = 0.01$ and timestepping obeying (11) with initial conditions $h(0, x) = h_0, u(0, x) = u_0$ and periodic boundary conditions.

spacing Δx and the information propagation speed. For the case of non-linear PDEs, it is a necessary but not sufficient condition for the stability of a numerical scheme [LeVeque, 2002, section 4.4][Durran, 2010, introductory paragraph of section 4.5].

While the nuances of determining the correct CFL condition for a particular problem are plentiful, we will not discuss them here and instead direct readers to the referenced material if they desire more information. For our purposes, the correct CFL condition is given by [LeVeque, 2002, equation 4.17 and the following sentence], namely

$$\Delta t \leq \frac{\Delta x}{\max\{|\lambda_1|, |\lambda_2|\}}. \tag{10}$$

As in this attempt, we assume $\lambda_{1,2} > 0$, clearly $|\lambda_1|$ is larger[d] and so we specify to

$$\Delta t^{(n+1)} \leq \frac{\Delta x}{\max\limits_j\{u_j^{(n)}\} + \sqrt{g \max\limits_j\{h_j^{(n)}\}}} \tag{11}$$

where, since u and h vary over their discretised grids, we find their maximum at each timestep to ensure compliance with (10) and update Δt for the next timestep accordingly.

[d] In the general case where it is unknown which of $|\lambda_1|$ and $|\lambda_2|$ is larger, our specification would simply introduce an additional absolute value to surround $u_j^{(n)}$ before taking its maximum. That way, since the sum of the maxima of the absolute values is greater than the maximum of the absolute value of the sum (by the triangle inequality), we produce a slightly tighter upper bound on Δt without having to actually know which of $|\lambda_1|$, $|\lambda_2|$ is larger.

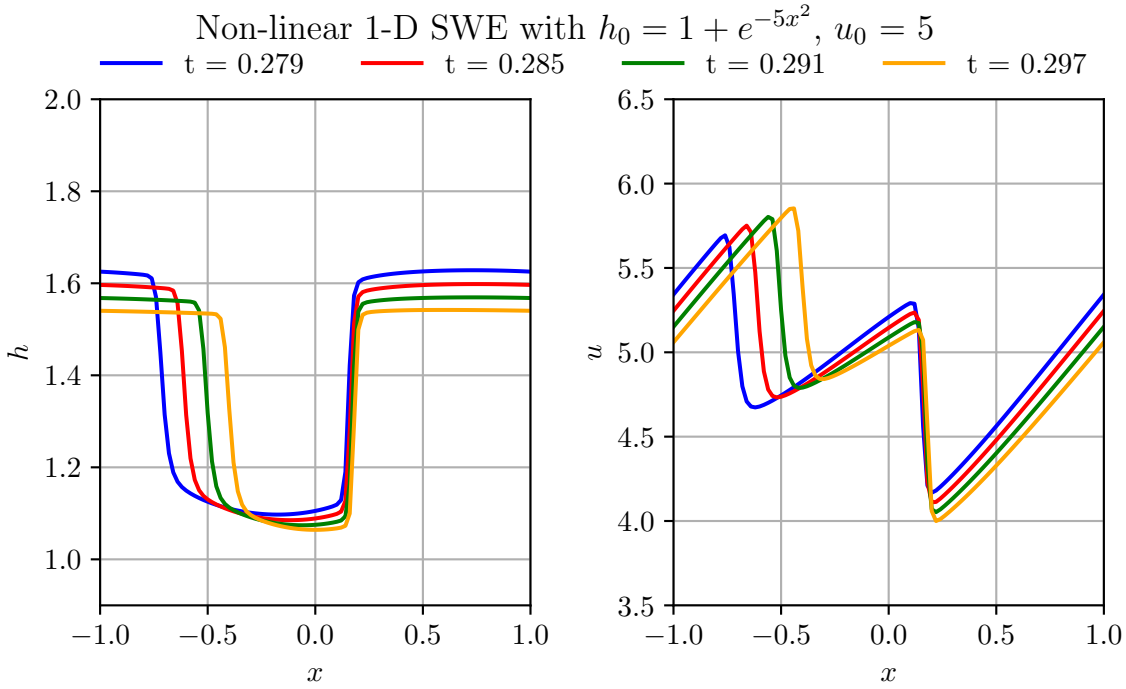

Figure 3: Numerical solution to the system (1)-(2) using the FTBS finite difference scheme over the domain $x \in [-1, 1]$ discretised with $\Delta x = 0.01$ and timestepping obeying (11) with initial conditions $h(0, x) = h_0, u(0, x) = u_0$ and periodic boundary conditions.

Rearranging our expression for λ_2 in accordance with our assumption, we would thus expect our scheme to be stable for an initial condition $u(0, x) > \sqrt{g \max\{h(0, x)\}}$ and for timestepping obeying (11).

3.2 Results

Maintaining our Gaussian-based initial condition for h, the above analysis in isolation would suggest $u(0, x) = \sqrt{2g} \approx 4.2$ would be an appropriate initial condition. To 'play it safe' as it were, figure 3 has $u_0 = 5$ and indeed depicts stable evolution after significant time. Unfortunately, there are more nuances at play here than we accounted for.

Figure 4 shows the same evolution but for $u_0 = 4.3$, still above the theoretic limit of stability under our assumptions, and yet the solution clearly develops an instability. Note however, that this time the solution remains stable for a not-insignificant amount of time, before collapsing. We thus expect this instability to be arising from a different source. Indeed, [Durran, 2010, introductory paragraph of section 4.5] alludes to the fact that non-linear PDEs can develop discontinuities in their finite difference schemes, even starting with smooth initial data. This provides ample motivation to implement the formulation of the SWE [LeVeque, 2002, equation 13.5] which does not assume smoothness of h and u. Regrettably, we did not have the time to implement this, instead choosing to focus on one final attempt to ensure stability for arbitrary smooth initial conditions with our current formulation.

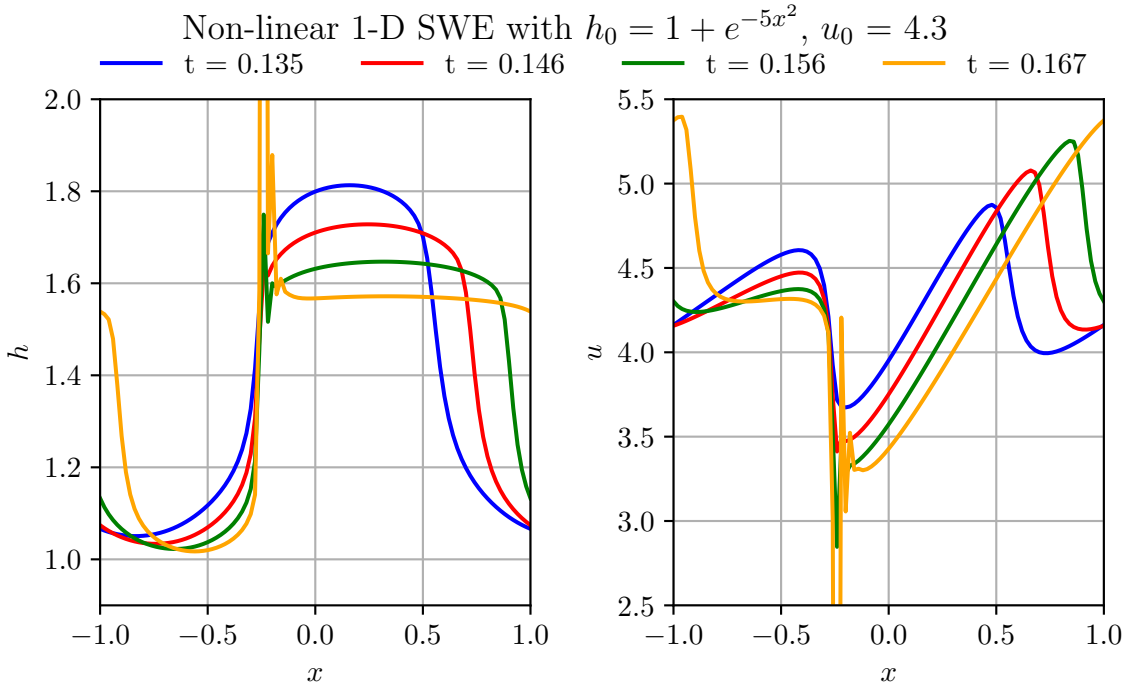

Figure 4: Numerical solution to the system (1)-(2) using the FTBS finite difference scheme over the domain $x \in [-1, 1]$ discretised with $\Delta x = 0.01$ and timestepping obeying (11) with initial conditions $h(0, x) = h_0, u(0, x) = u_0$ and periodic boundary conditions.

4 A more rigorous final attempt

4.1 The characteristic map

To truly tackle the problem of our eigenvalues having different signs, we must invoke a far more rigorous strategy. Namely the so-called 'characteristic variable' approach. There is some brief motivation for this in [LeVeque, 2002, beginning of section 3.11] and a richer description of the procedure in [Durran, 2010, pages 5 and 6] but in both cases, this is only applied to the linear advection equation. Thus, for the sake of completeness, we give an outline of the method applied to the non-linear SWE as follows.

Returning to the vector form (6), or equivalently, using the Jacobian matrix (7), to the form

$$\frac{\partial}{\partial t}\begin{pmatrix} u \\ h \end{pmatrix} + \begin{pmatrix} u & g \\ h & u \end{pmatrix} \frac{\partial}{\partial x}\begin{pmatrix} u \\ h \end{pmatrix} = 0, \tag{12}$$

we diagonalise the matrix (possible by hyperbolicity [Durran, 2010, discussion surrounding equation (1.6)]) using the previously calculated eigenvalues (8) together with the right-eigenvectors

$$\begin{pmatrix} u & g \\ h & u \end{pmatrix} \mathbf{r}_1 = (u + \sqrt{gh})\mathbf{r}_1 \implies \mathbf{r}_1 = \begin{pmatrix} 1 \\ \sqrt{h/g} \end{pmatrix} \tag{13}$$

and

$$\begin{pmatrix} u & g \\ h & u \end{pmatrix} \mathbf{r}_2 = (u - \sqrt{gh})\mathbf{r}_2 \implies \mathbf{r}_2 = \begin{pmatrix} -1 \\ \sqrt{h/g} \end{pmatrix} \tag{14}$$

to construct the equation

$$\begin{pmatrix} u & g \\ h & u \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ \sqrt{h/g} & \sqrt{h/g} \end{pmatrix} \begin{pmatrix} u+\sqrt{gh} & 0 \\ 0 & u-\sqrt{gh} \end{pmatrix} \begin{pmatrix} 1 & -1 \\ \sqrt{h/g} & \sqrt{h/g} \end{pmatrix}^{-1}. \tag{15}$$

Now define

$$\mathbf{Q} = \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ \sqrt{h/g} & \sqrt{h/g} \end{pmatrix}^{-1} \begin{pmatrix} u \\ h \end{pmatrix} \tag{16}$$

$$= \frac{1}{2\sqrt{h/g}} \begin{pmatrix} \sqrt{h/g} & 1 \\ -\sqrt{h/g} & 1 \end{pmatrix} \begin{pmatrix} u \\ h \end{pmatrix} \tag{17}$$

$$= \frac{1}{2} \begin{pmatrix} u+\sqrt{gh} \\ -u+\sqrt{gh} \end{pmatrix} \tag{18}$$

to map into characteristic variables, with the corresponding inverse map,

$$\mathbf{q} = \begin{pmatrix} u \\ h \end{pmatrix} = \begin{pmatrix} Q_1 - Q_2 \\ \frac{1}{g}(Q_1+Q_2)^2 \end{pmatrix}. \tag{19}$$

This allows us to elegantly write the vector equation (12) in the form

$$\frac{\partial \mathbf{Q}}{\partial t} + \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \frac{\partial \mathbf{Q}}{\partial x} = 0 \tag{20}$$

which obviously decouples into the two almost-Burgers' equations,[e]

$$\frac{\partial Q_1}{\partial t} + 2Q_1 \frac{\partial Q_1}{\partial x} = 0 \tag{21}$$

$$\frac{\partial Q_2}{\partial t} - 2Q_2 \frac{\partial Q_2}{\partial x} = 0. \tag{22}$$

Since $\lambda_1 > 0$ and $\lambda_2 < 0$ in the default case of no initial velocity ($u_0 = 0$), we are now legitimately justified in applying FTBS to (21) and FTFS to (22).

4.2 The CFL condition

As we are now formally solving a different equation, we need a new CFL condition for our timestepping to obey. Since finding the true CFL condition for non-linear equations is notoriously difficult, we opt to linearise our equations about some mean value and take the arising CFL condition as a starting point.

Assuming a mean velocity of zero, and taking the mean height of our initial Gaussian to be $H \approx 1.4$,[f] our mean value for the characteristic variables is $\bar{Q}_1 = \bar{Q}_2 = \sqrt{gH}/2$. Thus, linearising (21)-(22) in the form $Q_{1,2} = \bar{Q} + Q'_{1,2}$ for $Q'_{1,2}$ small, our characteristic system further collapses to two linear advection equations with speeds $\pm\sqrt{gH}$ respectively.

The CFL condition for the linear advection equation is well known (see [LeVeque, 2002, equation 4.27] for example) and for our purposes, takes the form

$$0 \leq \sqrt{gH}\frac{\Delta t}{\Delta x} \leq 1 \tag{23}$$

which can easily be rearranged to give a timestepping constraint for Δt as necessary.

[e] 'Almost-Burgers' is not a technical term as far as we are aware, but is used here to note that without the additional factor of ± 2, these equations would reduce to Burgers'. Note also the use of expressions for $\lambda_{1,2}$ in terms of $Q_{1,2}$.

[f] Integrating h_0 over the interval $[-1, 1]$ and then dividing by the interval's length, as standard.

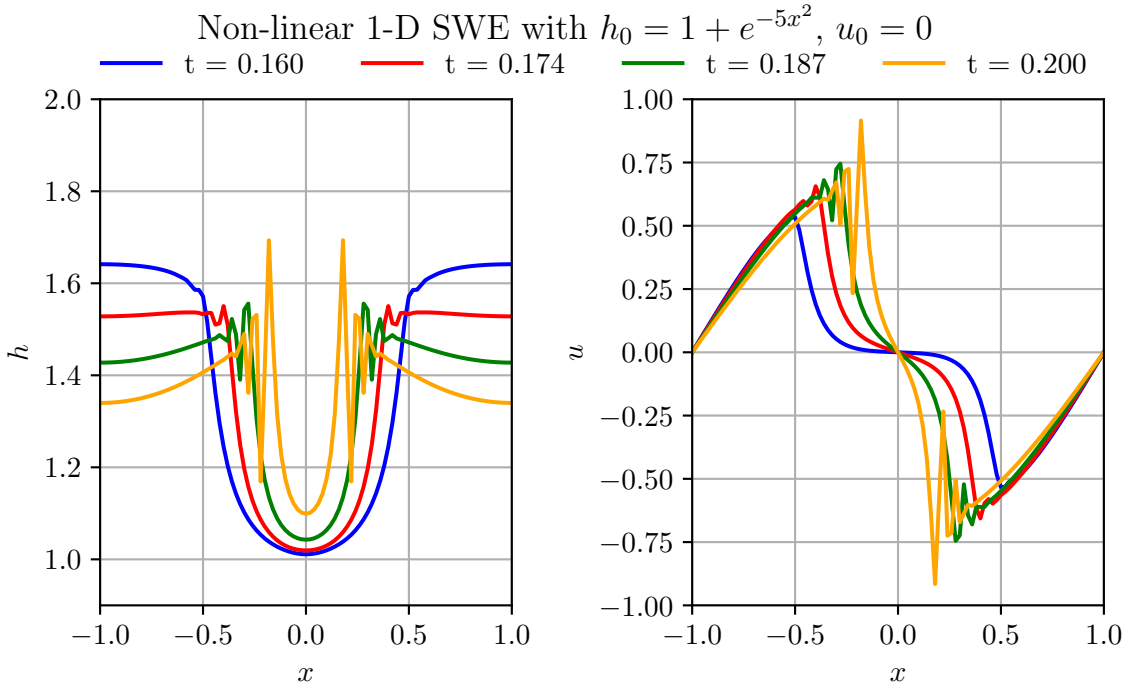

Figure 5: Numerical solution to the system (1)-(2) via solution to the corresponding characteristic system (21)-(22) using an FTBS/FTFS split finite difference scheme over the domain $x \in [-1, 1]$ discretised with $\Delta x = 0.01$ and timestepping obeying (23) with initial conditions $h(0, x) = h_0, u(0, x) = u_0$ and periodic boundary conditions.

4.3 Results

Figure 5 shows the now familiar sight of an arising instability which might, on first inspection, suggest our new-and-improved method has not worked. However, the instability depicted is of the second type discussed, namely one that arises after notable time evolution, and is due to non-linearities. This is a marked improvement for the $u_0 = 0$ case as, recalling figure 1, this time the system is at least temporarily stable.

One may ask as to why the system is not indefinitely stable given the rigour of the scheme and the agreement with the CFL condition (23). However, recall that this condition was derived from the linearised SWE, and even in that case, only admits a necessary condition for stability, not a sufficient one. If we in-fact tighten the upper bound of (23) from 1 to 0.88, we see in figure 6 that for exactly the same initial conditions and essentially[g] the same time slices, the system is now well-behaved.

[Durran, 2010, page 100] highlights the difference between a timestep bound derived from the CFL condition, and one that is *sufficient* for stability. He provides an example where the sufficient condition is, as in our case, simply a more restrictive upper bound, citing von Neumann stability analysis for the derivation. Unfortunately such an analysis is exceedingly challenging in the non-linear case and determining a more precise value than ~ 0.88 empirically would test the limits

[g]The code responsible for plotting these time slices takes an input in 'number of timesteps', not an exact value of t, hence with a different value of Δt it was not possible for the time slices to *exactly* line up.

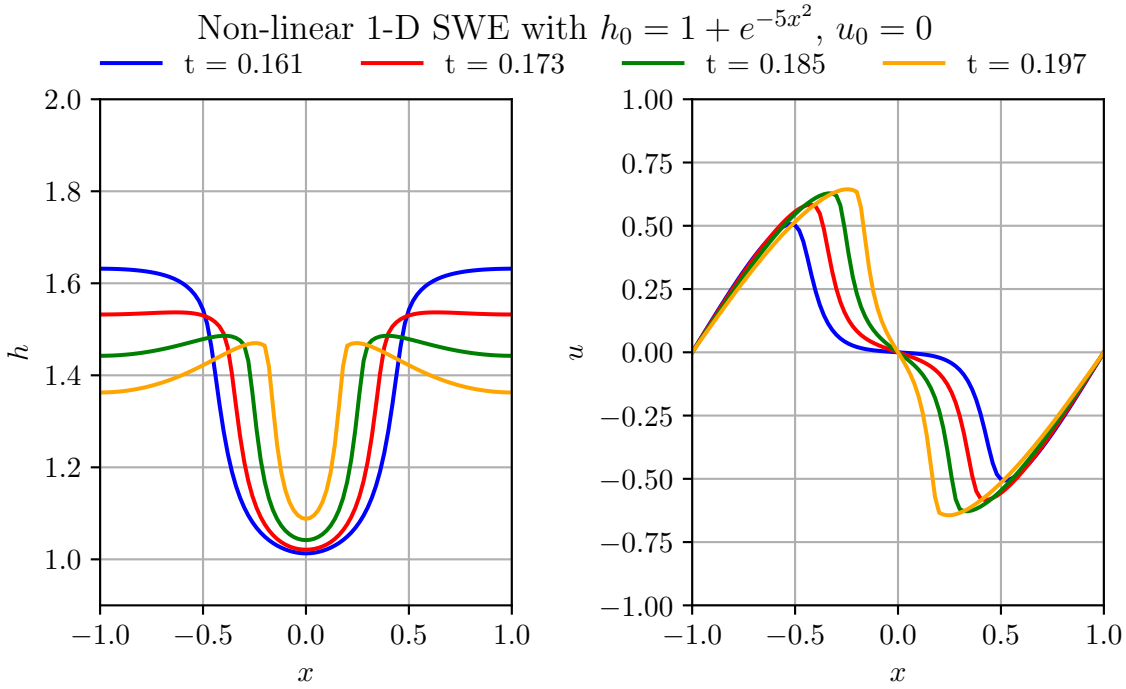

Figure 6: Numerical solution to the system (1)-(2) via solution to the corresponding characteristic system (21)-(22) using an FTBS/FTFS split finite difference scheme over the domain $x \in [-1, 1]$ discretised with $\Delta x = 0.01$ and timestepping obeying $0 \leq \sqrt{gH}\Delta t/\Delta x \leq 0.88$ with initial conditions $h(0, x) = h_0, u(0, x) = u_0$ and periodic boundary conditions.

of our patience. However, to make an entirely qualitative, non-rigorous observation, the evolution seems visually on the cusp on instability as the bound heads towards 0.9 and beyond.

Clearly there are numerous other directions this report could go, given more time. Deriving the sufficient condition for stability in the characteristic framework, and implementing the formulation from [LeVeque, 2002, equation 13.5] that does not assume smooth initial data come to mind as obvious next steps. Of course the other glaring area of investigation is to verify that these schemes do indeed have $\mathcal{O}(\Delta t, \Delta x)$ error growth by computing their error convergence. Although, given the notorious difficultly of getting such computations to work, perhaps we are fortunate that we stuck to stability for this report.

References

Maurizio Brocchini and Nicholas Dodd. Nonlinear shallow water equation modeling for coastal engineering. *Journal of waterway, port, coastal, and ocean engineering*, 134(2):104–120, 2008.

Adrian Constantin and Joachim Escher. Wave breaking for nonlinear nonlocal shallow water equations. 1998.

Philippe Courtier and Jean-Francois Geleyn. A global numerical weather prediction model with variable resolution: Application to the shallow-water equations. *Quarterly Journal of the Royal Meteorological Society*, 114(483):1321–1346, 1988.

Dale R. Durran. *Numerical Methods for Fluid Dynamics*, volume 32 of *Texts in Applied Mathematics*. Springer New York, 2010. ISBN 978-1-4419-6411-3 978-1-4419-6412-0. doi: 10.1007/978-1-4419-6412-0. URL `http://link.springer.com/10.1007/978-1-4419-6412-0`.

Joey Foster. Numerics assignnment GitHub repository. `https://github.com/Joey-Foster/PhD-Code/tree/master/Numerics-assignment`, 2024. Accessed on 1/11/24.

Bashar Khorbatly. The highly nonlinear shallow water equation: local well-posedness, wave breaking data and non-existence of sech 2 solutions. *Monatshefte für Mathematik*, 203(3):635–651, 2024.

Randall J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 1 edition, August 2002. ISBN 978-0-521-81087-6 978-0-521-00924-9 978-0-511-79125-3. doi: 10.1017/CBO9780511791253.

Robert Morgan. Linearization and Stability Analysis of Nonlinear Problems. *Rose-Hulman Undergraduate Mathematics Journal*, 16(2), January 2017.

Hilary Weller. Numerical modelling of the atmosphere and ocean (MFC CDT lecture notes), 2024. Accessed on 25/10/24.