Fundamentals of Wave Simulation: Source Terms

Matilde Tozzi

Abstract

In this paper we present some numerical methods to solve non homogeneous hyperbolic PDEs. In particular we focus on the features of the Godunov and Strang splittings and analyse their accuracy. We also talk briefly about the difficulties that may arise when solving stiff problems.

Index Terms

Wave Simulation, Source Terms, Seminar, hyperbolic PDEs, ODEs, Fractional Methods, Strang Splitting, Godunov Splitting.

I. From Conservation Laws to Balance Laws

IN this paper, we examine some numerical methods for the solution of *non homogeneous balance laws*. These are an extension of the well-known *conservation laws* of the form $q_t + f(q)_x = 0$ by adding a *source term* $\psi(q)$. This is called source term even if physically it represents a sink rather than a source, i.e. a loss of q. Our reference equation is therefore

$$q_t + f(q)_x = \psi(q). \tag{1}$$

We will mostly study problems where the homogeneous equation

$$q_t + f(q)_x = 0 \tag{2}$$

is *hyperbolic* and the source terms don't depend on derivatives of q, so that

$$q_t = \psi(q) \tag{3}$$

is an independent system of ODEs.

II. Godunov-Strang splitting

One standard approach to these problems is using a *fractional-step* or *operator-splitting* method, where we alternate between solving the homogeneous (2) and source (3) term. This allows us to easily use many standard numerical methods for PDEs and ODEs.

A. The Advection-Reaction Equation

The standard example that will be used to illustrate all the following numerical methods is the *advection-reaction* equation, which can be seen as the model for the transport along a flow of a radioactive substance which decays at rate β while it's transported at constant speed $\bar{u}$, with initial conditions $q(x,0) = \mathring{q}(x)$.

$$q_t + \bar{u} q_x = -\beta q \tag{4}$$

We can compute the exact solution of (4), because along the characteristic $\frac{dx}{dt} = \bar{u}$ we have $\frac{dq}{dt} = -\beta q$ and it follows that

$$q(x,t) = e^{-\beta t} \mathring{q}(x - \bar{u}t). \tag{5}$$

In Figure 1, we can see the evolution of the solution for different time steps: we notice the characteristic transport of the initial data $\mathring{q}$ at constant speed $\bar{u}$, typical of hyperbolic equations, combined with a decrease in the concentration of the substance at rate β due to the source term.

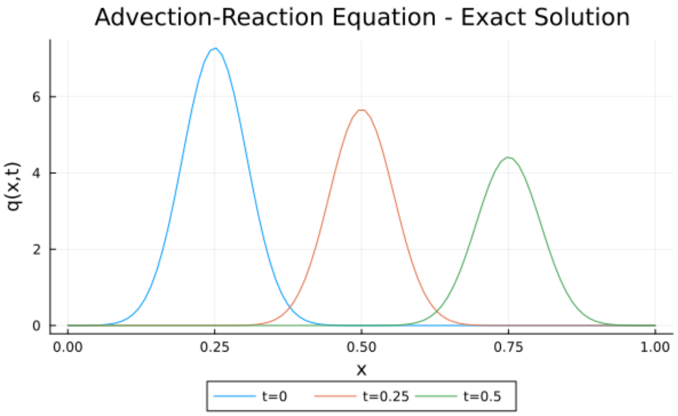


Fig. 1. Evolution of the exact solution of the advection-reaction equation with $\bar{u} = 1$, $\beta = 1$, and $\mathring{q} = \text{Gaussian}(0.25, 0.003)$. We see that the maximum point decreases in time.

B. The Unsplit Method

Of course for this specific example we can easily compute an *unsplit* method

$$\frac{Q_i^{n+1} - Q_i^n}{\Delta t} = -\bar{u}\frac{Q_i^n - Q_{i-1}^n}{\Delta x} - \beta Q_i^n \Rightarrow Q_i^{n+1} = Q_i^n - \bar{u}\frac{\Delta t}{\Delta x}(Q_i^n - Q_{i-1}^n) - \Delta t \beta Q_i^n \tag{6}$$

which is first-order accurate and stable for $0 < \bar{u}\frac{\Delta t}{\Delta x} \leq 1$. A second-order method is also easily obtainable with the Taylor expansion of $q(x, t + \Delta t)$ as explained in [1]. An important thing to note is that the full Taylor expansion of (4) can be written formally as

$$e^{-\Delta t(\bar{u}\partial_x + \beta)} q(x,t) := q(x, t + \Delta t) = \sum_{j=0}^{\infty} \frac{(\Delta t)^j}{j!} \partial_t^j q(x,t) = \sum_{j=0}^{\infty} \frac{(\Delta t)^j}{j!} (-\bar{u}\partial_x - \beta)^j q(x,t). \tag{7}$$

The operator $e^{-\Delta t(\bar{u}\partial_x + \beta)}$ is called *solution operator* for the equation (4) over a time step of length Δt.

C. Godunov Splitting

In the case of the advection equation, we can split it into two subproblems:

$$\text{Problem A: } q_t + \bar{u} q_x = 0, \tag{8}$$

$$\text{Problem B: } q_t = -\beta q. \tag{9}$$

The idea is to solve the two problems in an alternating manner, using standard solving stategies, e.g.:

$$\text{A-step: } Q_i^* = Q_i^n - \frac{\bar{u}\Delta t}{\Delta x}(Q_i^n - Q_{i-1}^n), \tag{10}$$

$$\text{B-step: } Q_i^{n+1} = Q_i^* - \beta \Delta t Q_i^*. \tag{11}$$

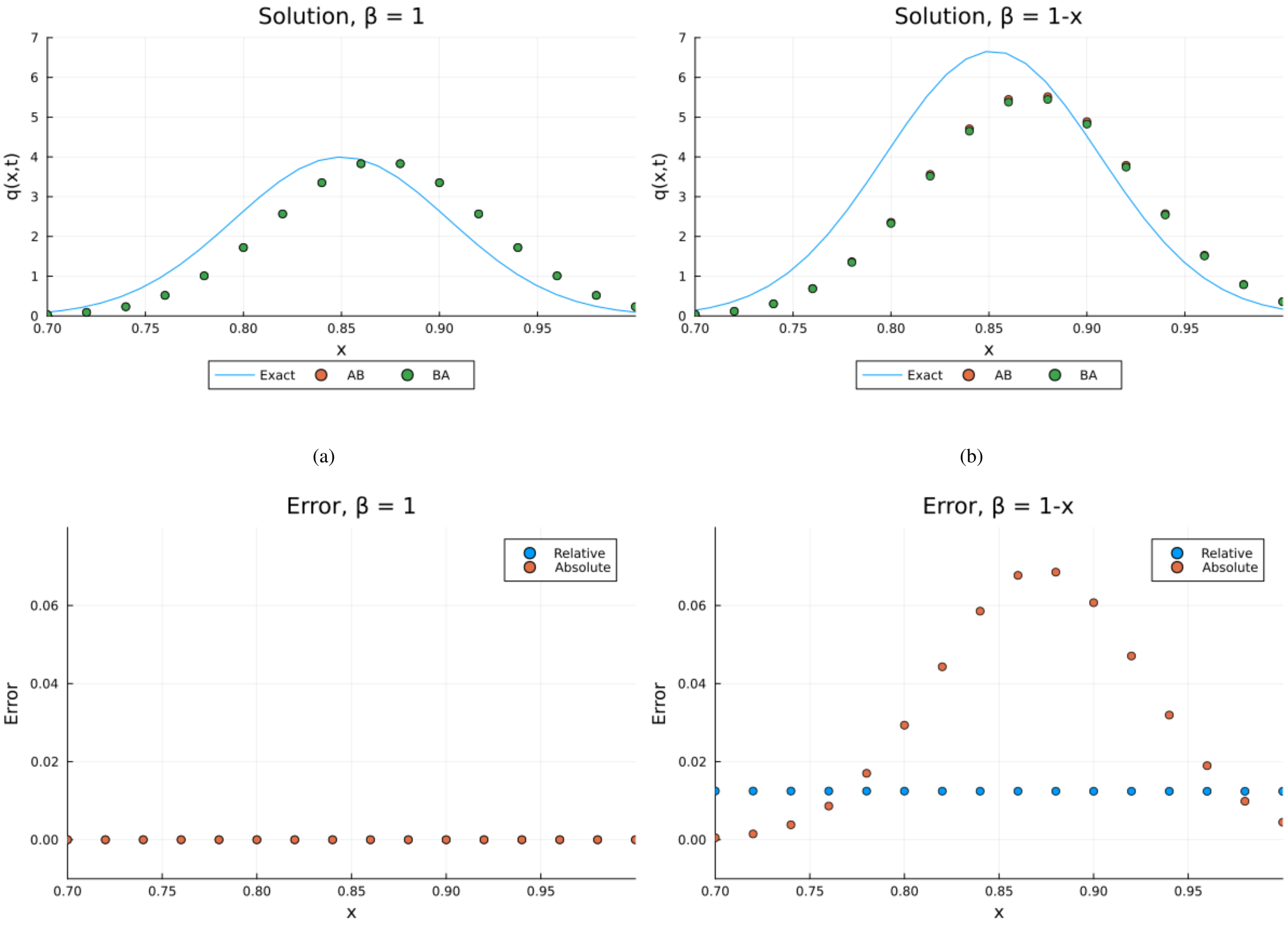
This method is known as *Godunov splitting*, also called Lie-Trotter splitting (mostly outside the Finite Volume setting).

One may think that given that both Q_i^* and Q_i^{n+1} are calculated using Δt, the solution is valid for time $2\Delta t$, but it is not really the case: in fact if we combine the two stages and eliminate Q_i^*, as detailed in [1], we obtain

$$Q_i^{n+1} = Q_i^n - \frac{\bar{u}\Delta t}{\Delta x}(Q_i^n - Q_{i-1}^n) - \beta \Delta t Q_i^n + \frac{\bar{u}\beta \Delta t^2}{\Delta x}(Q_i^n - Q_{i-1}^n). \tag{12}$$

The first three terms on the right hand side agree with (6) and the last one is $\mathcal{O}(\Delta t^2)$, so this method is also consistent and first-order accurate. A question that may arise is: does the accuracy improve if we use a second-order method for both (8) and (9)? In this specific case, yes, but it is not true in general because the splitting introduces a $\mathcal{O}(\Delta t)$ error. The reason why we don't have an error in this case is that we can really decouple the advection and the decay and do these updates in either order. This doesn't happen anymore if for example we take $\beta = 1 - x$, because in this case the value depends on the position and applying the B-step before or after the A-step changes the result. We say that the two subproblems *commute* if there is no splitting error.

This is illustrated in Figure 2, where we compare the case $\beta = 1$ and $\beta = 1 - x$. The error is small and one needs a very coarse approximation to show it clearly. This is the reason why the numerical solution seems so far from the analytical solution.



(a)

(b)

(c)

(d)

Fig. 2. Comparison between the exact solution of the advection-reaction equation and the split method with the two different orders of steps. The difference is hardly noticeable in the plot, but the relative error is non-zero. Note that there is a big gap between the analytical and the numerical solution because we are using a very coarse approximation to highlight the error. The problem has $\bar{u} = 1$, $\mathring{q} = \text{Gaussian}(0.25, 0.003)$, $\Delta x = \Delta t = 0.02$, $t = 0.6$.

D. General Formulation

To better understand the commuting property, we consider the more general formulation

$$q_t = (\mathcal{A} + \mathcal{B})q \tag{13}$$

where $\mathcal{A}$ and $\mathcal{B}$ can be differential operators, such as $-\bar{u}\partial_x$ and $-\beta(x)$ in the case of the advection-reaction equation. To simplify the calculations, we assume that they don't explicitly depend on t, so that we can write

$$q_{tt} = (\mathcal{A} + \mathcal{B})q_t = (\mathcal{A} + \mathcal{B})^2 q \tag{14}$$

without having to take care of the chain rule. If we Taylor expand the solution at time t and use the notation defined in (7), we easily get to

$$q(x, \Delta t) = \sum_{j=0}^{\infty} \frac{\Delta t^j}{j!} (\mathcal{A} + \mathcal{B})^j q(x, 0) = e^{\Delta t(\mathcal{A}+\mathcal{B})} q(x, 0). \tag{15}$$

With Godunov splitting, we obtain

$$q^*(x, \Delta t) = e^{\Delta t \mathcal{A}} q(x, 0) \tag{16}$$

and

$$q^{**}(x, \Delta t) = e^{\Delta t \mathcal{B}} q^*(x, \Delta t) = e^{\Delta t \mathcal{B}} e^{\Delta t \mathcal{A}} q(x, 0). \tag{17}$$

The splitting error is then

$$q(x, \Delta t) - q^{**}(x, \Delta t) = (e^{\Delta t(\mathcal{A}+\mathcal{B})} - e^{\Delta t \mathcal{B}} e^{\Delta t \mathcal{A}}) q(x, 0). \tag{18}$$

If we expand q^{**}, we obtain

$$q^{**}(x, \Delta t) = (I + \Delta t(\mathcal{A} + \mathcal{B}) + \frac{1}{2}\Delta t^2(\mathcal{A}^2 + 2\mathcal{B}\mathcal{A} + \mathcal{B}^2) + \dots) q(x, 0). \tag{19}$$

It is fundamental to note that in the Δt^2 term, in (15) we have $(\mathcal{A} + \mathcal{B})^2 = (\mathcal{A} + \mathcal{B})(\mathcal{A} + \mathcal{B}) = \mathcal{A}^2 + \mathcal{A}\mathcal{B} + \mathcal{B}\mathcal{A} + \mathcal{B}^2$, which in general is *not* the same as the $(\mathcal{A}^2 + 2\mathcal{B}\mathcal{A} + \mathcal{B}^2)$ term that we have in (19). We can clearly see now how there is no splitting error only if $\mathcal{A}\mathcal{B} = \mathcal{B}\mathcal{A}$, that is, if the operators commute. This means that if it is not the case, the Godunov method can only ever be first-order accurate, due to the splitting error.

E. Strang Splitting

Fortunately, there is an easy way to overcome the first-order limitation of the Godunov splitting and offer second-order accuracy in most cases. It is called *Strang splitting* and the idea is to solve $q_t = \mathcal{A}q$ over $\frac{\Delta t}{2}$ at first, then use the result to do a full time step on $q_t = \mathcal{B}q$, and then complete the iteration with the remaining half time step on $q_t = \mathcal{A}q$. Of course it works if we swap $\mathcal{A}$ and $\mathcal{B}$ too. This means that this time we are approximating $e^{\Delta t(\mathcal{A}+\mathcal{B})}$ by $e^{\frac{1}{2}\Delta t \mathcal{A}} e^{\Delta t \mathcal{B}} e^{\frac{1}{2}\Delta t \mathcal{A}}$. The Taylor expansion shows in fact that

$$e^{\frac{1}{2}\Delta t \mathcal{A}} e^{\Delta t \mathcal{B}} e^{\frac{1}{2}\Delta t \mathcal{A}} = I + \Delta t(\mathcal{A} + \mathcal{B}) + \frac{1}{2}\Delta t^2(\mathcal{A}^2 + \mathcal{A}\mathcal{B} + \mathcal{B}\mathcal{A} + \mathcal{B}^2) + \mathcal{O}(\Delta t^3), \tag{20}$$

so this method correctly captures the equation. We can note that after n time steps we obtain

$$Q^n = \underbrace{(e^{\frac{1}{2}\Delta t \mathcal{A}} e^{\Delta t \mathcal{B}} e^{\frac{1}{2}\Delta t \mathcal{A}})(e^{\frac{1}{2}\Delta t \mathcal{A}} e^{\Delta t \mathcal{B}} e^{\frac{1}{2}\Delta t \mathcal{A}}) \dots (e^{\frac{1}{2}\Delta t \mathcal{A}} e^{\Delta t \mathcal{B}} e^{\frac{1}{2}\Delta t \mathcal{A}})}_{n \text{ times}} Q^0. \tag{21}$$

Given that $e^{\frac{1}{2}\Delta t \mathcal{A}} e^{\frac{1}{2}\Delta t \mathcal{A}} = e^{\Delta t \mathcal{A}}$, we see that this method differs from the Godunov one only in the fact that we start and end with a half time step on $\mathcal{A}$. Another way of obtaining the same result is by alternating the oder of application of $\mathcal{A}$ and $\mathcal{B}$, in each time step, i.e.

$$\begin{aligned} Q^1 &= e^{\Delta t \mathcal{A}} e^{\Delta t \mathcal{B}} Q^0 \\ Q^2 &= e^{\Delta t \mathcal{B}} e^{\Delta t \mathcal{A}} Q^1 \\ &\vdots \end{aligned} \tag{22}$$

Analogously to our analysis of (21), we can see that the result is essentially the same, but with Δt instead of $\frac{1}{2}\Delta t$. This is computationally better because it requires fewer function evaluations, but it is more difficult to implement with variable time steps. Furthermore, it needs an even number of iterations in order to obtain the desired cancellation of errors.

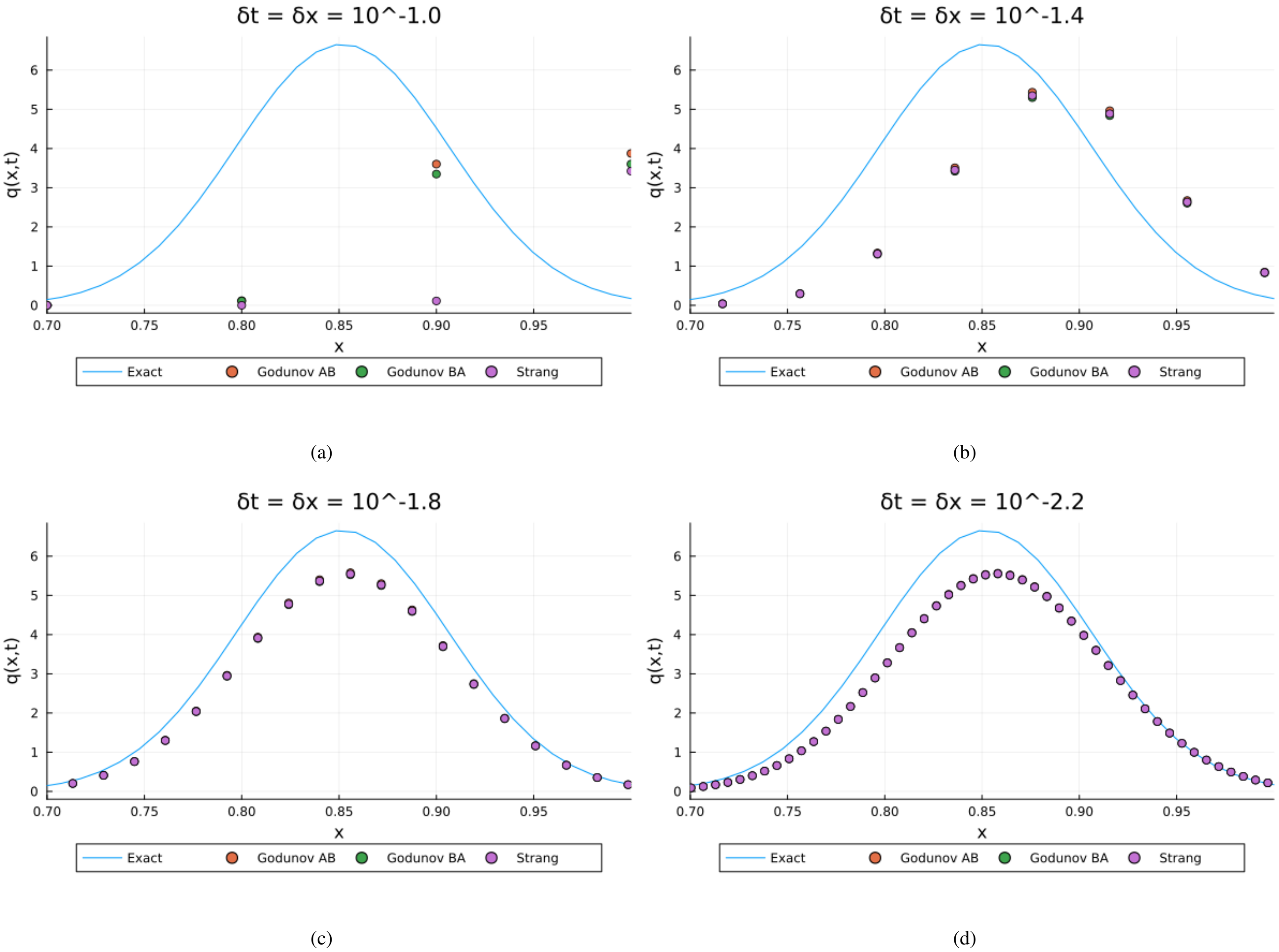
F. Accuracy

We replicated some results from [1] and confirmed that for this specific problem "the Godunov splitting is essentially as accurate as the Strang splitting". We decided to try with a completely different type of problem, namely the *logistic equation* in the form

$$u' = u - u^2 \quad , \qquad u(0) = 0.1 \tag{23}$$

with solution $u(t) = (9e^{-t} + 1)^{-1}$ [8]. Both subproblems, $u' = u$ and $u' = -u^2$, have been discretised with a forward Euler step.

The results show a similar behaviour. The two operators do not commute and in fact we see in Figure 4 and 5 that Godunov usually overestimates or underestimates compared to Strang. In this case, one of the two Godunov versions behaves better, probably because of the difference in magnitude between u and u^2.



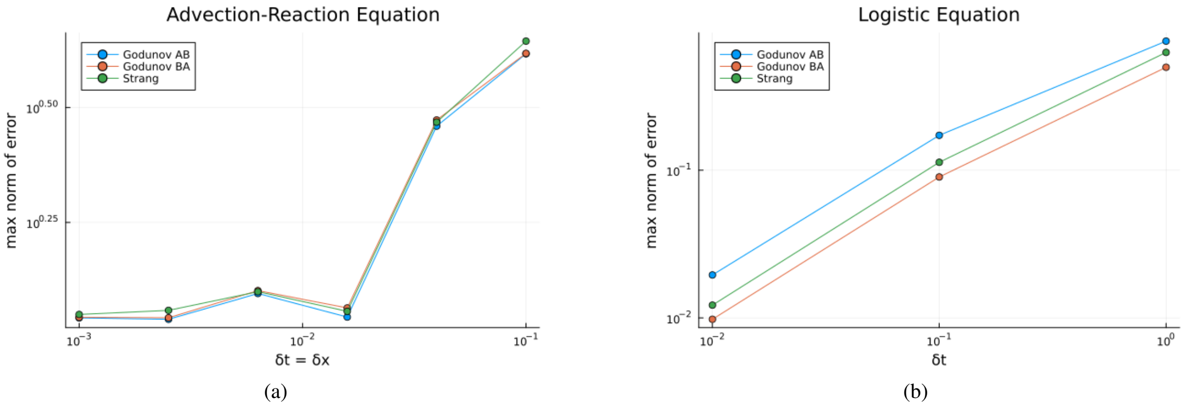
(a)

(b)

(c)

(d)

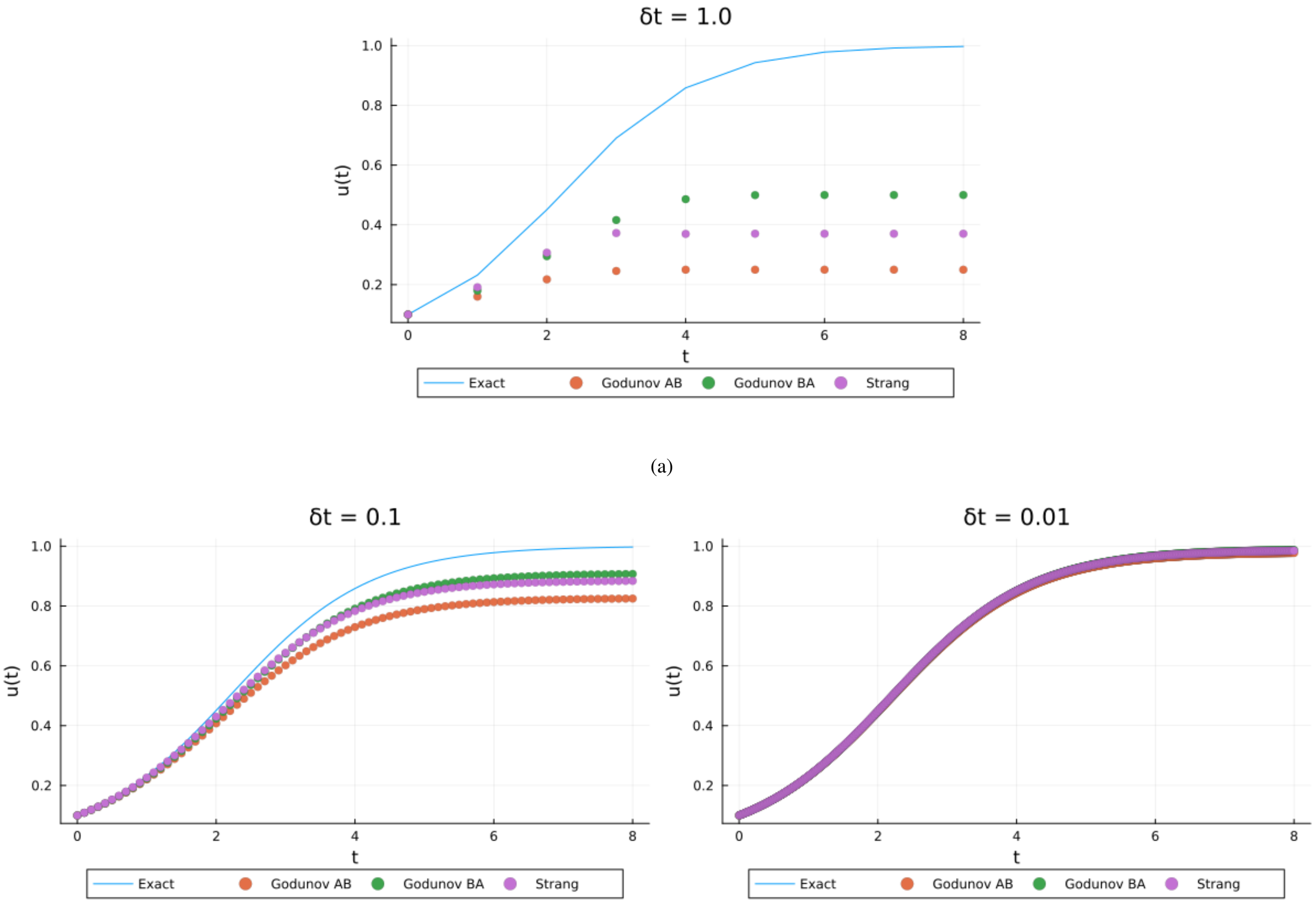
Fig. 3. Comparison of different discretisations. The problem has $\bar{u} = 1$, $\mathring{q} = \text{Gaussian}(0.25, 0.003)$, $\Delta x = \Delta t = 0.02$, $t = 0.6$.



(a)

(b)

Fig. 4. Log-log plot of max-norm errors.



(a)

(b)

(c)

Fig. 5. Comparison of the exact solution of the logistic equation, the Godunov split method with the two different orders of steps, and the Strang method.

III. Implicit methods and choice of ODE solver

In the case of Godunov splitting, we want to use a second-order accurate method in order to maintain the right accuracy. However, in general we cannot use multistep methods that require more than one level of data, because we only have Q_i^* to use to compute Q_i^{n+1}. Values of Q_i^* from previous time steps cannot be used because they are computed by solving a different problem, which is not our ODE. Runge-Kutta methods are very useful because they calculate their own intermediate values to construct higher-order approximations. For explicit methods, we need to make sure that the method is stable with the used time step.

If the ODE $q_t = \psi(q)$ is *stiff* (i.e. such that an extremely small time step is required to solve it with an explicit numerical method), as detailed in Section IV, then an implicit method is needed. The usual choice is the trapezoidal rule:

$$Q_i^{n+1} = Q_i^* + \frac{\Delta t}{2}[\psi(Q_i^*) + \psi(Q_i^{n+1})]. \tag{24}$$

Another nice property of the split methods is that they only require the ODE part to be solved implicitly: the hyperbolic part can still be solved with explicit methods.

If ψ depends on derivatives of q, these need to be discretised. For example, if $\psi(q) = \mu q_{xx}$, then (24) becomes the Crank-Nicolson method and requires solving a tridiagonal system.

IV. Stiff and singular source terms and the associated numerical difficulties

The last consideration of this paper is about stiffness and singularity for source terms. Sometimes it happens that the source term is not distributed in space, but rather behaves like a delta function, i.e. is very large only on a small region compared to our domain. This is the case for example for chemical reactions that can happen on very different time scales compared to the fluid dynamic one. If the reaction zone is concentrated, this source term is called *stiff* in analogy with the the ODEs. Another typical example of stiff reacting flow is the *detonation wave*: an explosion where a gas burns in a thin reaction zone and gets propagated in the rest of the gas like a shock wave. The thin reaction zone can be modeled as a delta function.

We can say that the solution is evolving on a *slow manifold* in state space and perturbing the solution causes it to produce a rapid transient response followed again by a slow evolution. A very simple example is

$$u'(t) = -\frac{u(t)}{\tau} \tag{25}$$

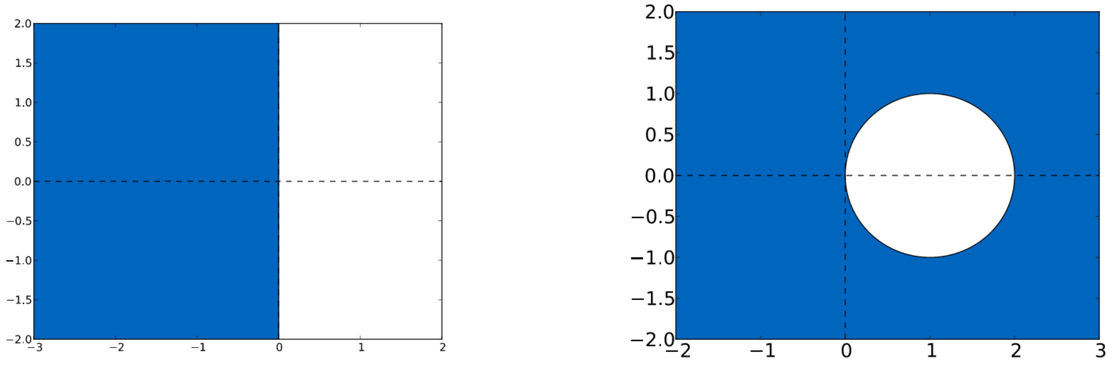
where of course $u \equiv 0$ is the slow manifold. Solving stiff hyperbolic equations is even more difficult than solving stiff ODEs, because the fastest reactions are *not* in equilibrium everywhere: for example in a detonation wave, the wave travels through space, so we cannot concentrate our efforts on a specific area of the domain. For some problems the only solution is using an adaptive mesh refinement whereas in other cases using implicit methods may be enough thanks to their good stability properties. As we saw in Section III, a problem with splitting methods and stiff ODEs is that we cannot use previous time steps, so we are limited in the choice of our solver. The trapezoidal rule seems to work fine for ODEs, but fails for hyperbolic equations with stiff source terms. If we consider (25), the trapezoidal methods yields

$$U^{n+1} = \left(\frac{1 - \frac{1}{2}\frac{\Delta t}{\tau}}{1 + \frac{1}{2}\frac{\Delta t}{\tau}} \right) U^*. \tag{26}$$

If we start on the slow manifold, in this case $U^* = 0$, then $U^{n+1} = 0$ and we remain in the slow manifold. In any other case, if $-\frac{\Delta t}{\tau} \to -\infty$, then $U^{n+1} = -U^*$ and this means that the coefficient in (26) makes the solution oscillate in time rather than decay as we would expect. It is due to the fact that the trapezoidal rule is an *A-stable* method. In fact, the stability region is the left half plane, but the problem is that the stability function doesn't approach zero everywhere as the step size goes to infinity [5]. That's what causes the oscillatory behaviour. We need an *L-stable* method, where the point at infinity is inside the stability region and so the coefficient approaches a value less than 1 in magnitude. The BDF (Backward Differentiation Formulas) methods have this characteristic and the simplest one is the backward Euler method. In fact for (25)

$$U^{n+1} = U^* - \frac{\Delta t}{\tau} U^{n+1} \Rightarrow U^{n+1} = \left(\frac{1}{1 + \frac{\Delta t}{\tau}} \right) U^*. \tag{27}$$

We note that this time the coefficient approaches zero for $\frac{\Delta t}{\tau} \to \infty$. However, the implicit Euler method is only first-order accurate. If we want a second-order, one-step method, a choice can be the TR-BDF2 method. One can find more information in [1].



(a) Trapezoidal Rule [6]

(b) Implicit Euler Method [7]

Fig. 6. Stability region of the trapezoidal rule and implicit Euler.

In Figure 6, we can see that in fact as the distance from the origin increases, for the implicit Euler method all the points become stable, whereas in the trapezoidal rule case the points in the right half plane will never become stable.

Remarks

The code used to produce the plots is original work (unless differently credited) and can be found in [2]. We chose Julia to show the features of the language. Code inspired by [3] and [4].

REFERENCES

[1] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*. Cambridge: Cambridge University Press, 2002.

[2] https://github.com/matilde-t/SeminarCourse-FundamentalsOfWaveSimulation

[3] https://github.com/clawpack/apps/tree/master/fvmbook/chap17

[4] https://github.com/clawpack/riemann_book

[5] B. L. Byron, *On Padé approximations to the exponential function and A-stable methods for the numerical solution of initial value problems.* Waterloo, Ontario: University of Waterloo, 1969.

[6] https://commons.wikimedia.org/wiki/File:Stability_region_for_trapezoidal_method.svg

[7] https://commons.wikimedia.org/wiki/File:Stability_region_for_BDF1.svg

[8] https://hplgit.github.io/fdm-book/doc/pub/book/sphinx/._book018.html