

I. INTRODUCTION

In this exercise you will be introduced to various useful concepts in numerical simulations, such as finite differencing, characteristic analysis, the Method of Lines for evolving hyperbolic PDEs, stability analysis, the Courant-Friedrichs-Lewy stability criterion, convergence testing and different boundary conditions. The theoretical concepts will be explained and then you will see them in action by writing a 1+1 dimensional (1 spatial dimension + time) code to solve the model problem of the wave equation in flat space with a non-zero shift and periodic, reflecting and outgoing (radiative) boundary conditions.

In a flat spacetime with zero shift, the line element is simply

$$ds^2 = -d\tilde{t}^2 + d\tilde{x}^2.$$

Note that we work in units where the speed of light is $c = 1$. The wave equation is

$$\square\phi = -\frac{\partial^2\phi}{\partial\tilde{t}^2} + \frac{\partial^2\phi}{\partial\tilde{x}^2} = 0, \quad (1)$$

where $\square\phi$ is the flat space D'Alembertian applied to ϕ . The introduction of a non-zero shift is simply a coordinate transformation of the form

$$dt = d\tilde{t}, \quad dx = d\tilde{x} - \beta d\tilde{t},$$

where t and x are the new coordinates and β is the shift¹. In the new coordinates, the metric tensor is given by

$$g_{\mu\nu} = \begin{pmatrix} (-1 + \beta^2) & \beta \\ \beta & 1 \end{pmatrix},$$

which could be generalized to have a non trivial lapse function. This is left as an exercise to the reader. The inverse of the metric is also useful

$$g^{\mu\nu} = \begin{pmatrix} -1 & \beta \\ \beta & (1 - \beta^2) \end{pmatrix}.$$

The wave equation generalizes for a general metric to $\square\phi = \frac{1}{\sqrt{-g}}\partial_\mu[\sqrt{-g}g^{\mu\nu}\partial_\nu\phi] = 0$. We would like to rewrite this in first order form (i.e. with no second order time derivatives). In the present case we have $\sqrt{-g} = 1$ so the wave equation $\square\phi = 0$ becomes:

$$\begin{aligned} \square\phi &= \frac{1}{\sqrt{-g}}\partial_\mu[\sqrt{-g}g^{\mu\nu}\partial_\nu\phi] \\ &= \partial_t[g^{t\nu}\partial_\nu\phi] + \partial_x[g^{x\nu}\partial_\nu\phi] \\ &= \partial_t[g^{tt}\partial_t\phi + g^{tx}\partial_x\phi] + \partial_x[g^{xt}\partial_t\phi + g^{xx}\partial_x\phi] \\ &= \partial_t[-\partial_t\phi + \beta\partial_x\phi] + \partial_x[\beta\partial_t\phi + (1 - \beta^2)\partial_x\phi] \\ &= 0 \end{aligned} \quad (2)$$

This form of the equation suggests the definition of two new variables: $\psi := \partial_x\phi$ and $\pi := \partial_t\phi - \beta\partial_x\phi$ that will allow us to obtain a system of equations with first order time derivatives only. Notice that π is the argument of a first order

* diener@aei.mpg.de

† guzman@aei.mpg.de

¹ Since we are working in flat space-time, we do not have to distinguish between covariant and contravariant components of the shift.

derivative with respect to t in (2). The idea is to split the wave equation above into evolution equations for these new variables. The first of the equations is evident from (2), and reads

$$\partial_t \pi = \partial_x (\psi + \beta \pi) \quad (3)$$

If we assume that the wave function is C^2 , then an equation for ψ is found to be $\partial_t \psi = \partial_x (\partial_t \phi)$, and after using the definition of the variable π we end up with

$$\partial_t \psi = \partial_x (\pi + \beta \psi) \quad (4)$$

Now we have the wave equation in terms of the first order variables π and ψ , in fact, notice that if $\beta = 0$ we have the simplest wave equation $\partial_t^2 \phi = \partial_x^2 \phi$ plus one identity. This reminds us that we were interested in the solution for the original field ϕ , but we can obtain that easily just with the help of the definition of π , which states that

$$\partial_t \phi = \pi + \beta \psi. \quad (5)$$

Equations (3) and (4) constitute the complete wave equation, and the relation (5) can be integrated along with the evolution equations to give the physical variable ϕ as a function of time.

II. CHARACTERISTIC ANALYSIS

If we introduce the vector $\mathbf{u} = (\pi, \psi)^T$ we can write down the wave equations ((3) and (4)) as

$$\partial_t \mathbf{u} + \mathbf{A} \partial_x \mathbf{u} = \mathbf{u} \partial_x \beta, \quad (6)$$

where

$$\mathbf{A} = - \begin{pmatrix} \beta & 1 \\ 1 & \beta \end{pmatrix}.$$

The characteristic directions, that is, the local directions of propagation in the (t, x) plane of the variables can be calculated. We only need to find the eigenvalues of A , that is we have to solve $\det(\mathbf{A} - I_2 \tau) = 0$ for $\tau = dx/dt$, where I_2 is the 2x2 unit matrix. The result is

$$\tau_1 = -\beta + 1, \quad \tau_2 = -\beta - 1 \quad (7)$$

As the two eigenvalues are distinct and real, we have a strictly hyperbolic system. In the simple case $\beta = 0$ we end up with the light cone $x = x_0 \pm t$ which determines the local domains of influence and dependence for the fields propagating with the same speed in the two directions from the given point x_0 . This is fundamental to understand the stability of the evolution schemes and the discretization of our equations.

Example 1: $\beta = 0$. At every point, x_0 of the space-time the causal structure is bounded by the characteristic directions $x = x_0 \pm t$.

Example 2: $\beta = 1$. In this case the two values of τ indicate that the cone restricting the domain of dependence and of influence will be given by the region bounded by the straight lines $x = x_0 - 2t$ and $x = x_0$.

Example 3: $\beta = -1$. In this case the two values of τ indicate that the cone restricting the domain of dependence and of influence will be given by the straight lines $x = x_0$ and $x = x_0 + 2t$.

These three examples are illustrated in Figure 1.

Eigenvectors corresponding to the two eigenvalues τ_1 and τ_2 are $v_1 = (1, -1)^T$ and $v_2 = (1, 1)^T$. Forming the matrix of eigenvectors, $\mathbf{V}$, as

$$\mathbf{V} = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$$

and its inverse

$$\mathbf{V}^{-1} = \frac{1}{2} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$$

the matrix $\mathbf{A}$ can be decomposed as $\mathbf{A} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^{-1}$, where $\mathbf{\Lambda} = \text{diag}(\tau_1, \tau_2)$. Multiplying equation (6) by $\mathbf{V}^{-1}$ gives

$$\begin{aligned} \mathbf{V}^{-1} \partial_t \mathbf{u} + \mathbf{V}^{-1} \mathbf{A} \partial_x \mathbf{u} &= \mathbf{V}^{-1} \mathbf{u} \partial_x \beta \\ \partial_t \mathbf{w} + \mathbf{\Lambda} \partial_x \mathbf{w} &= \mathbf{w} \partial_x \beta, \end{aligned} \quad (8)$$

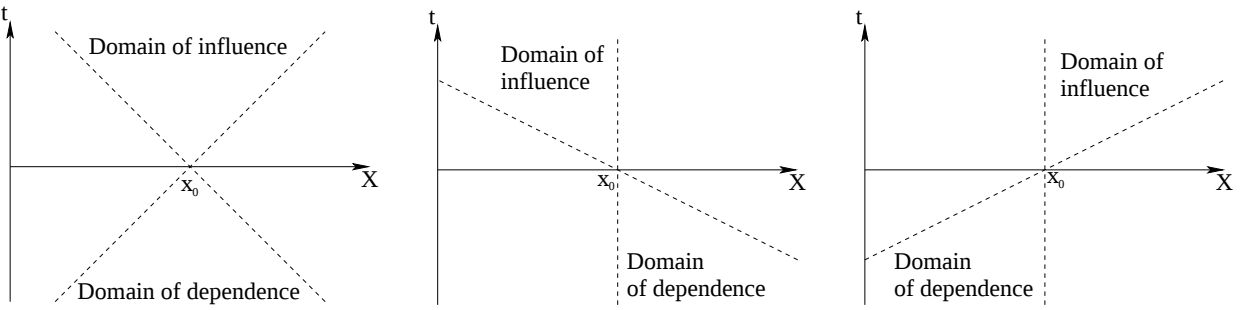


FIG. 1: The domain of dependence and influence for the three different cases from (left to right) $\beta = 0$, $\beta = 1$, $\beta = -1$

where

$$\mathbf{w} = \mathbf{V}^{-1}\mathbf{u} = \frac{1}{2}(\pi - \psi, \pi + \psi)^T = (R, L)^T \quad (9)$$

are the “characteristic variables”. Written in this way, the two equations decouple (since $\mathbf{\Lambda}$ is diagonal) and the scalar field has been decomposed into a right going ($R = 1/2(\pi - \psi)$) and a left going mode ($L = 1/2(\pi + \psi)$). This will be useful when implementing the outgoing boundary conditions.

III. INITIAL DATA

In our new formulation we have to provide initial data for $\pi(0, x)$ and $\psi(0, x)$. It is equivalent to provide initial data for the second order form of the wave equation, where we are required to provide data for $\phi(0, x)$ and $\partial_t(\phi(0, x))$.

For the present course we will restrict ourselves to the case of time-symmetric initial data, which are required to have an initially zero time derivative of the wave function $\phi(0, x)$ and letting the field itself to be an arbitrary distribution with compact support. This is equivalent to require $\pi(0, t) = 0$. In order to set up the initial condition for $\psi(0, t)$ we just need to give an arbitrary distribution for the wave function and just calculate its spatial derivative. In this exercise we choose an initial Gaussian distribution in the following way

$$\begin{aligned} \pi(0, x) &= 0 \\ \phi(0, x) &= Ae^{-(x-x_0)^2/\sigma^2} \Rightarrow \\ \psi(0, x) &= -2\frac{(x-x_0)}{\sigma^2}\phi(0, x). \end{aligned} \quad (10)$$

As shown in section II the solution to the wave equation is the superposition of a right going mode and a left going mode ($\phi(t, x) = f(x+t) + g(x-t)$ in the case of zero shift).

When assuming time-symmetric initial data and a Gaussian distribution we are assuming that

$$\phi(0, x) = \frac{1}{2}Ae^{-([x+t]-x_0)^2/\sigma^2} + \frac{1}{2}Ae^{-([x-t]-x_0)^2/\sigma^2}|_{t=0}. \quad (11)$$

The evolution of such initial data should show the splitting of the initial Gaussian into two Gaussians of the same width but half amplitude.

IV. NUMERICAL METHODS

We wrote down the wave equation in a first order form, so that all the equations we have to deal with (3), (4) and (5) are of the form

$$\partial_t f = S(f), \quad (12)$$

where $S(f)$ is a spatial operator applied to the function we want to integrate, including spatial derivatives of f . We will approximate our equations using finite differencing, which means that we discretize the continuum equations and we'll try to find the values of the functions f at a numerable set of points in space and time. We discretize the

space-time by saying that the time coordinate at label n is given by $t^n = n\Delta t$ and the spatial coordinate $x_i = i\Delta x$. Here n and i are integers and $i = 1, \dots, J$, where J is the number of gridpoints. We simplify the notation by denoting the value of the function $f(t^n, x_i)$ by f_i^n .

In order to update the value f_i^n it is necessary to know information about f in other points of the spacetime. We illustrate this fact with a very simple finite differencing of equation (3) for the case $\beta = 0$, and let the reader discretize the complete set of equations. In this particular case, to discretize the equation at the point (t^n, x_i) , we write down the following

$$\begin{aligned} \frac{\pi_i^n - \pi_i^{n-1}}{\Delta t} &\approx \frac{\psi_{i+1}^{n-1} - \psi_{i-1}^{n-1}}{2\Delta x} \Rightarrow \\ \pi_i^n &= \pi_i^{n-1} + \frac{\Delta t}{2\Delta x}(\psi_{i+1}^{n-1} - \psi_{i-1}^{n-1}), \end{aligned} \quad (13)$$

where the derivatives were approximated according to the standard definition of derivative, assuming that Δt and Δx are small. Now in order to know the value of $\pi(t^n, x_i)$ we need to know the value of ψ also at the neighbouring points (t^{n-1}, x_{i+1}) , (t^{n-1}, x_{i-1}) and the value of π at the point (t^{n-1}, x_i) . This is known as the FTCS (Forward in Time, Centered in Space). It is very easy and simple to implement but is unfortunately unstable as we will see later.

A. MoL and stability analysis

A convenient way to be able to implement different evolution schemes using the same spatial finite differencing and to understand the stability properties of these evolution schemes is to use the Method of Lines (MoL). Here the discretization is done in space only, yielding a set of coupled (through the spatial differences) ODEs for the time dependence of the field variables at the spatial grid points. These ODEs can then be integrated in time with a suitable ODE integrator².

In the Method of Lines, the stability properties can be analyzed in two steps. The first step is to perform a Fourier analysis of the spatial finite differencing equations³.

In our case we simply Fourier transform equation (6). Then the centered second order finite differencing applied to the Fourier modes of form $\xi \exp(ik\Delta j)$, where ξ is the amplitude of the Fourier mode, k is the wave number, $\Delta = \Delta x$ is the grid spacing and j is the index of the grid point is

$$\begin{aligned} \frac{\xi \exp(ik\Delta(j+1)) - \xi \exp(ik\Delta(j-1))}{2\Delta} &= \xi \exp(ik\Delta j) \frac{\exp(ik\Delta) - \exp(-ik\Delta)}{2\Delta} \\ &= \xi \exp(ik\Delta j) i \frac{\sin(k\Delta)}{\Delta}. \end{aligned} \quad (14)$$

So in this case the finite differencing operator acting on a Fourier mode simply gives back the mode multiplied by $i \sin(k\Delta)/\Delta$. Then we can rewrite the evolution equation (6) for the Fourier transformed variables as

$$\partial_t \tilde{\mathbf{u}} + \tilde{\mathbf{A}} \tilde{\mathbf{u}} = \tilde{\mathbf{u}} \partial_x \beta, \quad (15)$$

where $\tilde{\mathbf{A}}$ is

$$\tilde{\mathbf{A}} = -\frac{i \sin(k\Delta)}{\Delta} \begin{pmatrix} \beta & 1 \\ 1 & \beta \end{pmatrix}.$$

Assuming that we use a constant shift (which we will do in all the exercises) the stability properties is determined by the location of the eigenvalues of $\tilde{\mathbf{A}}$ in the complex plane. As this matrix is simply $i \sin(k\Delta)/\Delta$ times the matrix $\mathbf{A}$ from equation (6) we obtain that the eigenvalues are purely imaginary

$$\lambda = (-\beta \pm 1) i \frac{\sin(k\Delta)}{\Delta}.$$

² The FTCS scheme is in fact an example of a MoL scheme with centered differences in space and the time integration done with the simple Euler scheme.

³ If the problem is non-linear there is an additional step in linearizing the system of equations around a solution to the non-linear equations.

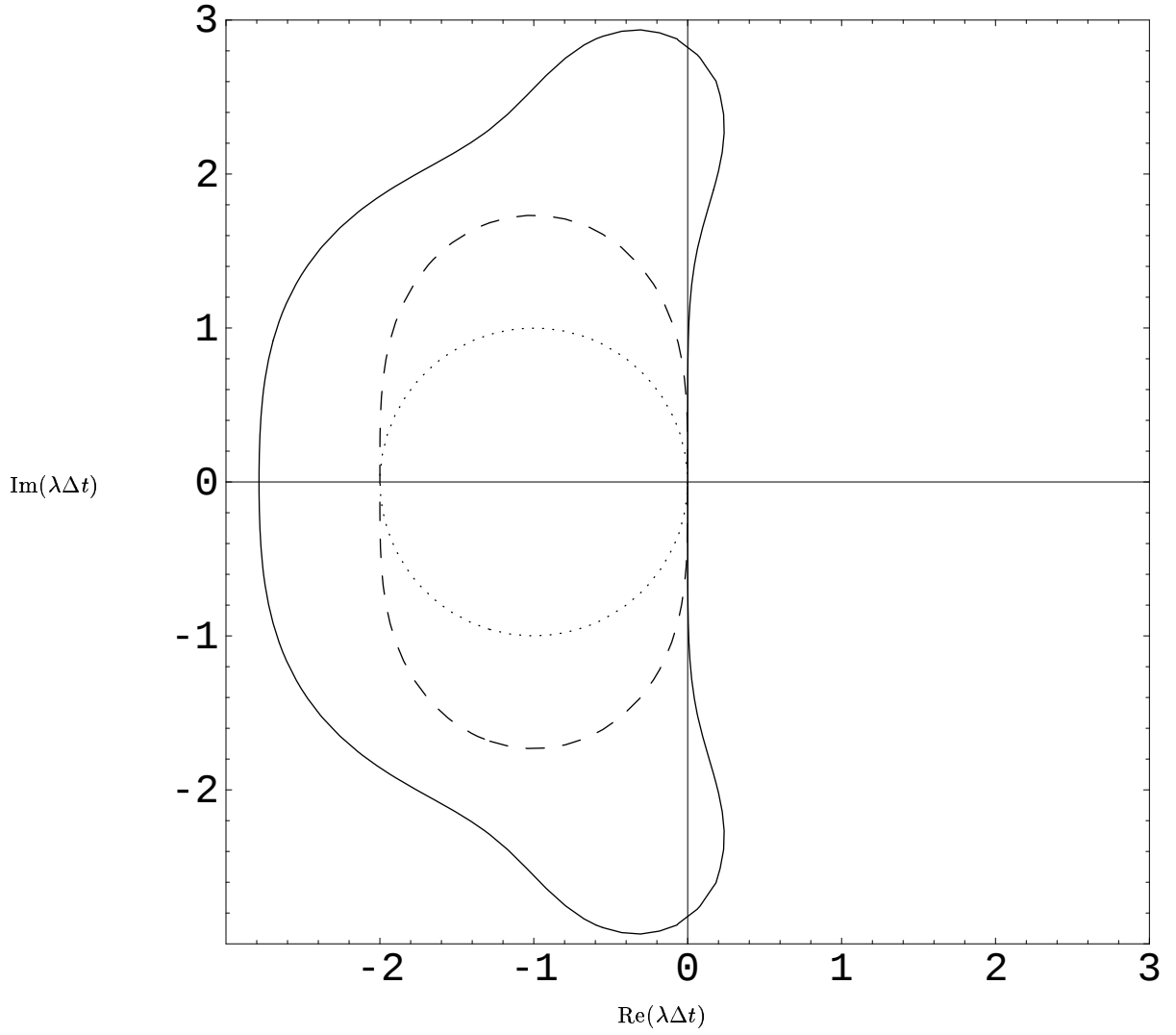


FIG. 2: The stability regions in the complex plane for RK1 or Euler (dotted line), RK2 (dashed line) and RK4 (solid line). In order for a numerical evolution all the eigenvalues of the Fourier transformed evolution system multiplied by the time step, should lie within the stability region.

A given ODE method will then give a stable evolution if $\lambda\Delta t$ lie within the methods region of stability for each eigenvalue λ . Determining the region of stability of a given ODE method is beyond the scope of this exercise so in Figure 2 we plot the stability region for three standard ODE methods: The Euler (or RK1) scheme, the RK2 scheme and the RK4 scheme. Both the Euler (or RK1) and RK2 schemes have stability regions that are tangent to the imaginary axis at the origin, but curves away from the imaginary axis immediately. Thus for our system with purely imaginary eigenvalues, it is impossible to choose a timestep for which $\lambda\Delta t$ lies within the stability region. Thus RK1 and RK2 are unconditionally unstable for our system of equations. On the other hand the RK4 scheme has a much larger stability region, that actually encompasses part of the imaginary axis. From the figure it can be seen that the curve defining the stability region crosses the imaginary axis at around 2.8 (the exact value is $\sqrt{8} \approx 2.82843$). As an example take the case of $\beta = 0$. In this case the maximal absolute value of the eigenvalues are $1/\Delta$ since $|\sin(k\Delta)| \leq 1$. Thus a RK4 evolution is stable if $\Delta t/\Delta \leq \sqrt{8}$ or $\Delta t \leq \sqrt{8}\Delta$. This is known as the Courant-Friedrichs-Lewy stability criterion (Courant or CFL condition for short).

There are many different ways of organizing an RK4 ODE integration scheme. In this exercise we will implement the classical RK4 formula to integrate the state vector f^{n-1} one timestep to obtain f^n :

$$\begin{aligned}
 k_1 &= S(t^{n-1}, f^{n-1}) \\
 k_2 &= S\left(t^{n-1} + \frac{\Delta t}{2}, f^{n-1} + \frac{\Delta t}{2} k_1\right) \\
 k_3 &= S\left(t^{n-1} + \frac{\Delta t}{2}, f^{n-1} + \frac{\Delta t}{2} k_2\right) \\
 k_4 &= S(t^{n-1} + \Delta t, f^{n-1} + \Delta t k_3) \\
 f^n &= f^{n-1} + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4) + O(\Delta t^5)
 \end{aligned} \tag{16}$$

V. PERIODIC BOUNDARY CONDITIONS

The numerical description of the problem is not complete until the boundary conditions are specified. One of the simplest boundary conditions to implement (and therefore useful for testing the evolution code) is the periodic boundary condition. This boundary, as the name implies, ensures that the solution is periodic, meaning that a wave going out of the right boundary reappears at the left boundary.

There are two simple ways to implement this. The first way is to simply modify the finite differencing at the boundary to use the data at the other boundary appropriately. The other way is to introduce a ghost point at both ends, use standard centered finite differencing in the interior and then synchronize the ghost points with the correct values from the other boundaries.

VI. EXERCISES

A. Exercise 1

- Write an evolution code for equations (3), (4) and (5) using centered finite differencing, the RK4 time integration scheme and periodic boundary conditions.
- Choose the spatial domain to be $[-1, 1]$.
- Use $\beta = 0$.
- Run your code with a courant factor of 2 and make sure that the signal reaches your boundaries at $t = 1$. If it does the propagation speed is correct.
- You should find similar results to what we show in Fig. 3, where we used the values $A = 0.01$, $\sigma = 0.1$ and a modest resolution $\Delta x = 0.005$. You could use any desired values, but it is important to keep the physical domain to be $[-1, 1]$.
- Try to run with a Courant factor larger than $\sqrt{8}$. How does the instability manifest itself?

B. Exercise 2

Assuming the function f is reasonably smooth we can use a Taylor expansion to estimate the accuracy of the centered finite difference approximation of the $\partial_x f$. Expanding f around the point x^i to estimate the function value in $x^{i-1} = x^i - \Delta$ and $x^{i+1} = x^i + \Delta$ we get

$$\begin{aligned}
 f_{i-1} &= f_i - f'_i \Delta + \frac{1}{2} f''_i \Delta^2 - \frac{1}{6} f'''_i \Delta^3 + \frac{1}{24} f^{(4)}_i \Delta^4 + O(\Delta^5) \\
 f_{i+1} &= f_i + f'_i \Delta + \frac{1}{2} f''_i \Delta^2 + \frac{1}{6} f'''_i \Delta^3 + \frac{1}{24} f^{(4)}_i \Delta^4 + O(\Delta^5).
 \end{aligned}$$

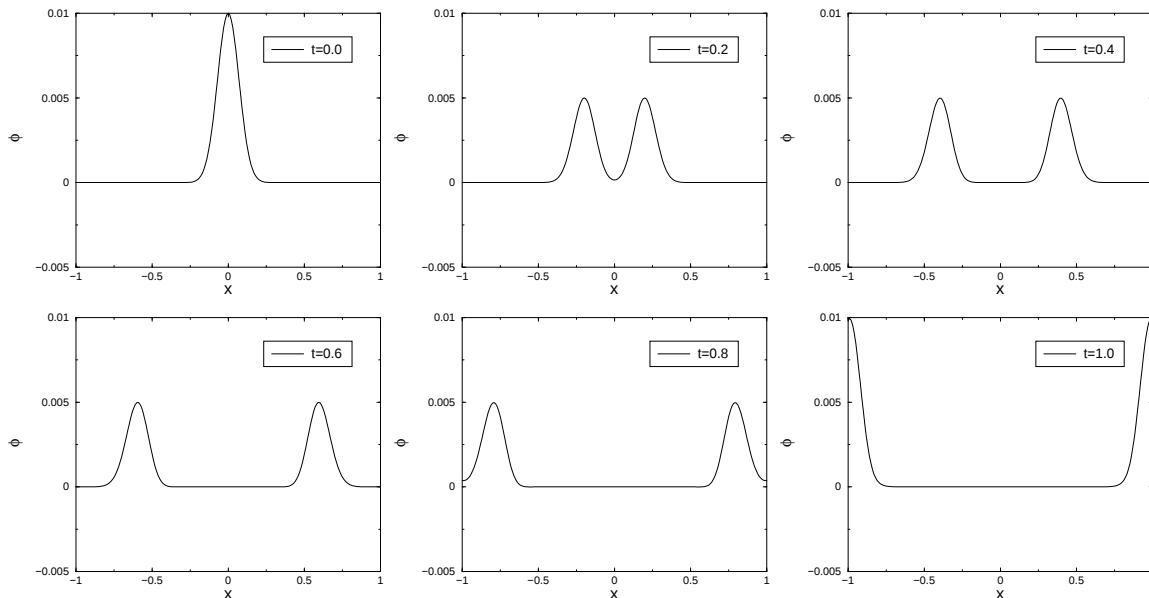


FIG. 3: Exercise 1. Snapshots of the evolution of a Gaussian at different times is shown. It can be seen that the initial distribution splits into two peaks of the same width and half amplitude. The maximum of the signal reaches the boundaries at $t = 1$, which means that the speed of propagation of our field is 1. At that time the two pulses meet each other at the boundaries.

Inserting this into the centered finite differencing formula we get

$$\frac{f_{i+1} - f_{i-1}}{2\Delta} = f'_i + \frac{1}{6}f'''_i \Delta^2 + O(\Delta^4).$$

Thus we see that the spatial finite differencing used here is second order accurate. Even though the RK4 scheme is fourth order accurate for the time evolution, we expect to be limited by the second order accurate spatial finite differences. If the higher order errors are small compared to the second order error (i.e. if we are in the convergent regime) then by increasing the resolution by a factor of 2 the error we make should decrease by a factor of 4. If the solution is known, this can be tested by running at two different resolutions. Denote the exact solution by f_e and the numerical solutions at the two different resolutions by f_Δ and $f_{2\Delta}$, then we expect

$$\begin{aligned} f_{2\Delta} - f_e &= C(2\Delta)^2 = 4C\Delta^2 \\ f_\Delta - f_e &= C\Delta^2, \end{aligned}$$

where C does not depend on the resolution but only on the properties of f_e .

Thus if we over plot $f_{2\Delta} - f_e$ and $4(f_\Delta - f_e)$ we should expect the curves to lie on top of each other.

When the exact solution is not known, we can still test the convergence of the code, but it requires running at least at 3 different resolutions. Again if the higher order error terms are negligible we expect

$$\begin{aligned} f_{4\Delta} &= f_e + C(4\Delta)^2 = f_e + 16C\Delta^2 \\ f_{2\Delta} &= f_e + C(2\Delta)^2 = f_e + 4C\Delta^2 \\ f_\Delta &= f_e + C\Delta^2. \end{aligned}$$

Thus we see that

$$\begin{aligned} f_{4\Delta} - f_{2\Delta} &= 12C\Delta^2 \\ f_{2\Delta} - f_\Delta &= 3C\Delta^2. \end{aligned}$$

So over plotting $f_{4\Delta} - f_{2\Delta}$ and $4(f_{2\Delta} - f_\Delta)$ the curves should lie on top of each other.

- Run your code with $\beta = 0$, $\sigma = 0.1$ at the 3 resolutions $\Delta = (0.005, 0.0025, 0.00125)$ and Courant factor 2 until $t = 2$. To make the convergence test easier, down sample the higher resolutions appropriately so that for all resolutions the output is done at the same spatial locations. Also make sure to output the data at the same times by skipping an appropriate number of time steps at higher resolutions.

- Use `ygraph` to perform the 3 resolution convergence test.

1. Load the difference between the low and medium resolution output for ϕ :
`ygraph -S phi_low.xg phi_med.xg`.
2. Load the difference between the medium and high resolution into the same `ygraph` session by placing the mouse in the `ygraph` window and pressing `Ctrl-S` or choosing the `Subtract A-B` option from the `Plot` menu. Fill out the filenames in the pop-up window and press okay.
3. In order to make the convergence obvious rescale the medium minus the high resolution by pressing `Ctrl-M` or choose the `Scale` option from the `Plot` menu and change the scale factor from 1 to 4.

If the two curves do not follow each other closely when you press the play button, there must be a bug in your code.

C. Exercise 3

- Use $\beta = 1$ everywhere. What happens now? If you don't get the same picture as in Figure 4 you have some debugging to do.
- Experiment with different constant values of the shift (positive, negative, smaller than 1, larger than 1). You will need to adjust the size of your timestep when using larger values of the shift. Does your code behave as expected?

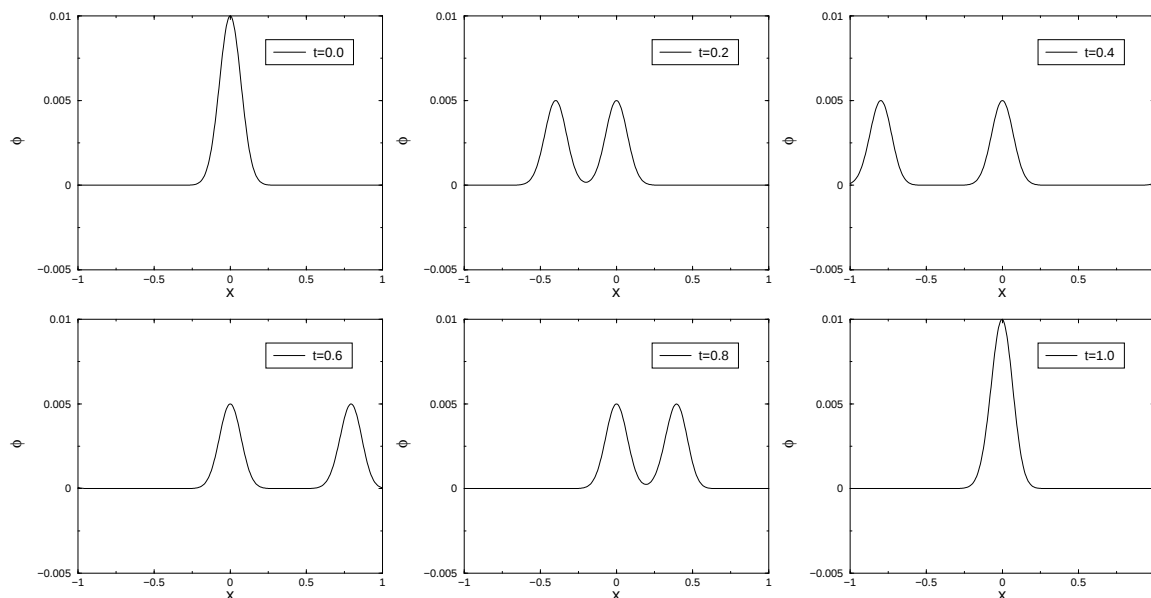


FIG. 4: Exercise 3. In this case we show the solution of the wave equation with $\beta = 1$. The Gaussian is split into two as in the former case, but this time the part travelling to the left reaches the boundary at time $t = 0.5$, which means that for the reference frame we have chosen the speed of propagation is 2. On the other hand, the pulse that originally should be travelling to the right just freezes and travels at speed 0 with respect to our reference frame. All this means that we have chosen a reference system moving at speed 1 to the right. As we are using periodic boundary conditions, we see at $t = 0.5$ the pulse moving to the left entering the grid from the right edge, and joins the original pulse at $t = 1$.

D. Exercise 4

In this exercise we will implement a reflecting boundary condition. Naively it would be expected that simply keeping the evolution static at the boundary, the wave would be reflected from the boundary.

- Implement the following boundary condition:

$$\begin{aligned}\pi_1^n &= \pi_J^n = 0 \\ \psi_1^n &= \psi_J^n = 0\end{aligned}$$

- Run with $\beta = 0$. What happens when the wave hits the boundary?

The previous reflecting boundary condition is actually not what we really want. We want the field ϕ to be static at the boundary. Looking at the definition of π , we see that with $\beta = 0$ we have $\pi = \partial_t \phi$ so in order to have ϕ static at the boundary we need $\pi = 0$. So the first part of the boundary conditions is okay. However, we also want $\partial^2 \phi / \partial t^2 = 0$ and from equation (3) we see that this requires $\partial_t \pi = \partial_x \psi = 0$ instead of $\psi = 0$. At the left boundary we can approximate the spatial derivative (after evolving everything in the interior) of ψ as $\partial_x \psi = (\psi_2^n - \psi_1^n) / dx = 0$ implying that we have to set $\psi_1^n = \psi_2^n$. A similar argument at the right boundary implies that $\psi_J^n = \psi_{J-1}^n$.

- Implement the following correction to the previous boundary condition:

$$\begin{aligned}\pi_1^n &= \pi_J^n = 0 \\ \psi_1^n &= \psi_2^n \\ \psi_J^n &= \psi_{J-1}^n\end{aligned}$$

- Run with $\beta = 0$ and see what happens when the wave hits the boundary.

E. Exercise 5

Now in many real applications periodic or reflecting boundaries are not what we really want. Often we don't have enough resources to discretize the equations we are interested in on a big enough computational domain and we are forced to place some computational boundaries at a finite distance. In that case we typically want any fields reaching the boundaries to simply propagate out of the grid with a minimum of reflections. These are the so called outgoing boundary conditions. This is where the characteristic analysis comes in handy. By decomposing the field into its right and left going modes we can make sure that the outgoing mode at a boundary just propagates out (to a prescribed accuracy) and that the ingoing mode is zero.

One basic recipe for our case is the following. Evolve all fields in the interior points. At the left boundary estimate the left going mode (this is the mode that is outgoing at this boundary) by extrapolating from the points next to the boundary. Let's denote this left going mode extrapolated to the boundary by LL . The outgoing boundary condition requires that the incoming mode is zero. At the left boundary this means that we should enforce $RR = 0$, where RR is the right going mode at the boundary. Remembering the equations for the right and left going modes in equation (9) we see that this implies

$$\begin{aligned}1/2(\pi_1^n + \psi_1^n) &= LL \\ 1/2(\pi_1^n - \psi_1^n) &= RR = 0.\end{aligned}$$

Solving these equations we find that at the left boundary we need to set

$$\pi_1^n = \psi_1^n = LL. \tag{17}$$

It's left as an exercise to the reader to figure out the correct formula at the right boundary.

- Implement the outgoing boundary conditions using zeroth order extrapolation (just copy the outgoing mode from the first point in).
- Notice that there are reflections from the boundary. how does these reflections scale with the resolution?
- Now implement a linear extrapolation scheme (using the two closest points to the boundary). What happened to the reflections? How do they now scale with resolutions? How can you further improve the outgoing boundary conditions?
- Now that we have mentioned the left and right going modes, try to output them and see if they really behave as expected.

Note that this boundary condition only works for $|\beta| \leq 1$, since otherwise both modes are moving in the same direction.

The reflecting boundary condition from exercise 4 does not work with a non-zero shift. However, we can use the same idea to derive a boundary condition for the more general case.

As before we want the first and second time derivative of ϕ to vanish at the boundary in order to keep ϕ static at the boundary. Equation (5) gives the first condition for π and ψ at the boundary

$$\pi_1^n + \beta_1 \psi_1^n = 0. \quad (18)$$

To find the second time derivative of ϕ we take the time derivative of Equation (5)

$$\frac{\partial^2 \phi}{\partial t^2} = \partial_t \pi + \beta \partial_t \psi = 0.$$

Using the evolution equations (3) and (4) we convert the time derivatives into spatial derivatives

$$\partial_x(\psi + \beta\pi) + \beta\partial_x(\pi + \beta\psi) = 0.$$

We finite difference this equation at the left boundary using one sided finite differencing

$$\psi_2^n + \beta_2 \pi_2^n - \psi_1^n - \beta_1 \pi_1^n + \beta_1(\pi_2^n + \beta_2 \psi_2^n - \pi_1^n - \beta_1 \psi_1^n) = 0.$$

Collecting the unknowns on the left hand side we get

$$\psi_1^n(1 + \beta_1^2) + 2\beta_1 \pi_1^n = \psi_2^n(1 + \beta_1 \beta_2) + \pi_2^n(\beta_1 + \beta_2). \quad (19)$$

Equations (18) and (19) can now be solved for the values of π_1^n and ψ_1^n . The result is

$$\begin{aligned} \psi_1^n &= \frac{\psi_2^n(1 + \beta_1 \beta_2) + \pi_2^n(\beta_1 + \beta_2)}{1 - \beta_1^2} \\ \pi_1^n &= -\beta_1 \psi_1^n. \end{aligned} \quad (20)$$

The boundary condition on the right boundary is obtained in a similar way. This is left as an exercise to the reader.

- Implement the reflecting boundary condition for non-zero shift.
- Run with $\beta = 0.2$ everywhere. What happens when the wave hits the left boundary? What happens when the wave hits the right boundary? What is the physical reason for these effects.
- Run with higher values of $\beta < 1$. What happens when β approaches 1?
- In the boundary condition we divide by $1 - \beta^2$ so this obviously doesn't work for $\beta = \pm 1$. What is the physical reason for this? (Hint: What happens if you try to reflect a wave from a boundary moving at the same speed as the reflected wave?)

APPENDIX A: SECOND ORDER RUNGE-KUTTA METHODS

In Figure 2 we showed the stability region for the Euler (or RK1) scheme, the RK2 scheme and the RK4 scheme. For completeness we list here two different ways to implement the RK2 scheme, which you can try and implement (if you have time or later after the school) and check the claim that these methods are unstable for our model problem. In the notation of equation (16) the methods are

$$\begin{aligned} k_1 &= S(t^{n-1}, f^{n-1}) \\ k_2 &= S\left(t^{n-1} + \frac{\Delta t}{2}, f^{n-1} + \frac{\Delta t}{2} k_1\right) \\ f^n &= f^{n-1} + \Delta t k_2 + O(\Delta t^3) \end{aligned} \quad (A1)$$

and

$$\begin{aligned}k_1 &= S(t^{n-1}, f^{n-1}) \\k_2 &= S(t^{n-1} + \Delta t, f^{n-1} + \Delta t \, k_1) \\f^n &= f^{n-1} + \frac{\Delta t}{2}(k_1 + k_2) + O(\Delta t^3).\end{aligned}\tag{A2}$$
