§3.2.2 The Leapfrog Method

§3.2.2 The Leapfrog Method

- **We have studied various simple solutions of the shallow water equations by making approximations.**

§3.2.2 The Leapfrog Method

- **We have studied various simple solutions of the shallow water equations by making approximations.**
- **In particular, by means of the perturbation method the equations have been linearised, making them amenable to analytical investigation.**

§3.2.2 The Leapfrog Method

- **We have studied various simple solutions of the shallow water equations by making approximations.**
- **In particular, by means of the perturbation method the equations have been linearised, making them amenable to analytical investigation.**
- **However, to obtain solutions in the general case, it is necessary to solve the full nonlinear system.**

§3.2.2 The Leapfrog Method

- **We have studied various simple solutions of the shallow water equations by making approximations.**
- **In particular, by means of the perturbation method the equations have been linearised, making them amenable to analytical investigation.**
- **However, to obtain solutions in the general case, it is necessary to solve the full nonlinear system.**
- **In numerical weather prediction (NWP) the fully nonlinear primitive equations are solved by numerical means.**

§3.2.2 The Leapfrog Method

- **We have studied various simple solutions of the shallow water equations by making approximations.**
- **In particular, by means of the perturbation method the equations have been linearised, making them amenable to analytical investigation.**
- **However, to obtain solutions in the general case, it is necessary to solve the full nonlinear system.**
- **In numerical weather prediction (NWP) the fully nonlinear primitive equations are solved by numerical means.**
- **In the atmosphere, the nonlinear advection process is a dominant factor.**

§3.2.2 The Leapfrog Method

- We have studied various simple solutions of the shallow water equations by making approximations.
- In particular, by means of the perturbation method the equations have been linearised, making them amenable to analytical investigation.
- However, to obtain solutions in the general case, it is necessary to solve the full nonlinear system.
- In numerical weather prediction (NWP) the fully nonlinear primitive equations are solved by numerical means.
- In the atmosphere, the nonlinear advection process is a dominant factor.
- To get some idea of the methods used, we look at the simple problem of formulating time-integration algorithms for the solution of the simple advection equation.

Recap. of Discretization Methods

There are several distinct approaches to the formulation of computer methods for solving differential equations. We will confine ourselves to the *finite difference method*.

Recap. of Discretization Methods

There are several distinct approaches to the formulation of computer methods for solving differential equations. We will confine ourselves to the *finite difference method*.

Other approaches include *finite element* method and the *spectral* method.

Recap. of Discretization Methods

There are several distinct approaches to the formulation of computer methods for solving differential equations. We will confine ourselves to the *finite difference method*.

Other approaches include *finite element* method and the *spectral* method.

The central idea of the finite difference approach is to *approximate the derivatives* in the equation by differences between adjacent points in space or time, and thereby reduce the differential equation to a difference equation.

Recap. of Discretization Methods

There are several distinct approaches to the formulation of computer methods for solving differential equations. We will confine ourselves to the *finite difference method*.

Other approaches include *finite element* method and the *spectral* method.

The central idea of the finite difference approach is to *approximate the derivatives* in the equation by differences between adjacent points in space or time, and thereby reduce the differential equation to a difference equation.

- **An *analytical* problem becomes an *algebraic* one.**

Recap. of Discretization Methods

There are several distinct approaches to the formulation of computer methods for solving differential equations. We will confine ourselves to the *finite difference method*.

Other approaches include *finite element* method and the *spectral* method.

The central idea of the finite difference approach is to *approximate the derivatives* in the equation by differences between adjacent points in space or time, and thereby reduce the differential equation to a difference equation.

- **An *analytical* problem becomes an *algebraic* one.**
- **A problem with an *infinite* degree of freedom is replaced by one with a *finite* degree of freedom.**

Recap. of Discretization Methods

There are several distinct approaches to the formulation of computer methods for solving differential equations. We will confine ourselves to the *finite difference method*.

Other approaches include *finite element* method and the *spectral* method.

The central idea of the finite difference approach is to *approximate the derivatives* in the equation by differences between adjacent points in space or time, and thereby reduce the differential equation to a difference equation.

- **An *analytical* problem becomes an *algebraic* one.**
- **A problem with an *infinite* degree of freedom is replaced by one with a *finite* degree of freedom.**
- **A *continuous* problem goes over to a *discrete* one.**

The Finite Difference Method

We start by looking at the *Taylor expansion* of $f(x)$:

$$f(x + \Delta x) = f(x) + f'(x).\Delta x + \frac{1}{2}f''(x)\Delta x^2 + [O(\Delta x^3)] \qquad (1)$$

$$f(x - \Delta x) = f(x) - f'(x).\Delta x + \frac{1}{2}f''(x)\Delta x^2 + [O(\Delta x^3)] \qquad (2)$$

The Finite Difference Method

We start by looking at the *Taylor expansion* of $f(x)$:

$$f(x+\Delta x) = f(x) + f'(x).\Delta x + \frac{1}{2}f''(x)\Delta x^2 + [O(\Delta x^3)] \qquad (1)$$

$$f(x-\Delta x) = f(x) - f'(x).\Delta x + \frac{1}{2}f''(x)\Delta x^2 + [O(\Delta x^3)] \qquad (2)$$

The higher order terms, represented by $O(\Delta x^3)$, become less important as Δx becomes smaller.

The Finite Difference Method

We start by looking at the *Taylor expansion* of $f(x)$:

$$f(x+\Delta x) = f(x) + f'(x).\Delta x + \frac{1}{2}f''(x)\Delta x^2 + [O(\Delta x^3)] \qquad (1)$$

$$f(x-\Delta x) = f(x) - f'(x).\Delta x + \frac{1}{2}f''(x)\Delta x^2 + [O(\Delta x^3)] \qquad (2)$$

The higher order terms, represented by $O(\Delta x^3)$, become less important as Δx becomes smaller.

We neglect these and obtain approximations for the derivative of $f(x)$ as follows:

$$f'(x) = \frac{f(x+\Delta x) - f(x)}{\Delta x} + O(\Delta x) = f'_F + O(\Delta x)$$

$$f'(x) = \frac{f(x) - f(x-\Delta x)}{\Delta x} + O(\Delta x) = f'_B + O(\Delta x)\,.$$

The Finite Difference Method

We start by looking at the *Taylor expansion* of $f(x)$:

$$f(x+\Delta x) = f(x) + f'(x).\Delta x + \tfrac{1}{2}f''(x)\Delta x^2 + [O(\Delta x^3)] \qquad (1)$$

$$f(x-\Delta x) = f(x) - f'(x).\Delta x + \tfrac{1}{2}f''(x)\Delta x^2 + [O(\Delta x^3)] \qquad (2)$$

The higher order terms, represented by $O(\Delta x^3)$, become less important as Δx becomes smaller.

We neglect these and obtain approximations for the derivative of $f(x)$ as follows:

$$f'(x) = \frac{f(x+\Delta x) - f(x)}{\Delta x} + O(\Delta x) = f'_F + O(\Delta x)$$

$$f'(x) = \frac{f(x) - f(x-\Delta x)}{\Delta x} + O(\Delta x) = f'_B + O(\Delta x)\,.$$

These are called the <u>forward</u> and <u>backward</u> differences.

The Finite Difference Method

We start by looking at the *Taylor expansion* of $f(x)$:

$$f(x+\Delta x) = f(x) + f'(x).\Delta x + \frac{1}{2}f''(x)\Delta x^2 + [O(\Delta x^3)] \qquad (1)$$

$$f(x-\Delta x) = f(x) - f'(x).\Delta x + \frac{1}{2}f''(x)\Delta x^2 + [O(\Delta x^3)] \qquad (2)$$

The higher order terms, represented by $O(\Delta x^3)$, become less important as Δx becomes smaller.

We neglect these and obtain approximations for the derivative of $f(x)$ as follows:

$$f'(x) = \frac{f(x+\Delta x) - f(x)}{\Delta x} + O(\Delta x) = f'_F + O(\Delta x)$$

$$f'(x) = \frac{f(x) - f(x-\Delta x)}{\Delta x} + O(\Delta x) = f'_B + O(\Delta x)\,.$$

These are called the forward and backward differences.

Keeping only leading terms, we incur errors of order $O(\Delta x)$.

***We can do better than this*: subtracting (2) from (1) yields:**

$$f'(x) = \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} + O(\Delta x^2) = f'_C + O(\Delta x^2)$$

***We can do better than this*: subtracting (2) from (1) yields:**

$$f'(x) = \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} + O(\Delta x^2) = f'_C + O(\Delta x^2)$$

This is $O(\Delta x^2)$, therefore *more accurate* for small Δx.

***We can do better than this*: subtracting (2) from (1) yields:**

$$f'(x) = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} + O(\Delta x^2) = f'_C + O(\Delta x^2)$$

This is $O(\Delta x^2)$, therefore *more accurate* for small Δx.

Adding (1) and (2) gives the corresponding expression for the second derivative:

$$f''(x) = \frac{f(x + \Delta x) - 2f(x) + f(x - \Delta x)}{\Delta x^2} + O(\Delta x^2)$$

These <u>centered differences</u> are of accuracy $O(\Delta x^2)$.

We can do better than this: **subtracting (2) from (1) yields:**

$$f'(x) = \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} + O(\Delta x^2) = f'_C + O(\Delta x^2)$$

This is $O(\Delta x^2)$, therefore *more accurate* for small Δx.

Adding (1) and (2) gives the corresponding expression for the second derivative:

$$f''(x) = \frac{f(x+\Delta x) - 2f(x) + f(x-\Delta x)}{\Delta x^2} + O(\Delta x^2)$$

These centered differences are of accuracy $O(\Delta x^2)$.

We can continue taking more and more terms, but obviously there is a trade-off between accuracy and efficiency.

***We can do better than this*: subtracting (2) from (1) yields:**

$$f'(x) = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} + O(\Delta x^2) = f'_C + O(\Delta x^2)$$

This is $O(\Delta x^2)$, therefore *more accurate* for small Δx.

Adding (1) and (2) gives the corresponding expression for the second derivative:

$$f''(x) = \frac{f(x + \Delta x) - 2f(x) + f(x - \Delta x)}{\Delta x^2} + O(\Delta x^2)$$

These <u>centered differences</u> are of accuracy $O(\Delta x^2)$.

We can continue taking more and more terms, but obviously there is a trade-off between accuracy and efficiency.

Fourth-order accurate schemes are sometimes used in NWP, but *second order accuracy is more popular*.

Exercise:

Consider the function $f(x) = A\sin(kx)$**.**

We know that the derivative is $kA\cos(kx)$**.**

Exercise:

Consider the function $f(x) = A\sin(kx)$**.**

We know that the derivative is $kA\cos(kx)$**.**

- **Show that a forward difference approximation gives**

$$f'_F(x) = kA\cos[k(x + \Delta x/2)] \cdot \left[\frac{\sin(k\Delta x/2)}{k\Delta x/2}\right]$$

Exercise:

Consider the function $f(x) = A\sin(kx)$**.**

We know that the derivative is $kA\cos(kx)$**.**

- **Show that a forward difference approximation gives**

$$f'_F(x) = kA\cos[k(x + \Delta x/2)] \cdot \left[\frac{\sin(k\Delta x/2)}{k\Delta x/2}\right]$$

- **Show that the centered difference approximation yields**

$$f'_C(x) = kA\cos[kx] \cdot \left[\frac{\sin(k\Delta x)}{k\Delta x}\right]$$

Exercise:

Consider the function $f(x) = A\sin(kx)$.

We know that the derivative is $kA\cos(kx)$.

- **Show that a forward difference approximation gives**

$$f'_F(x) = kA\cos[k(x + \Delta x/2)] \cdot \left[\frac{\sin(k\Delta x/2)}{k\Delta x/2}\right]$$

- **Show that the centered difference approximation yields**

$$f'_C(x) = kA\cos[kx] \cdot \left[\frac{\sin(k\Delta x)}{k\Delta x}\right]$$

- **Compare these to the true derivative $f'(x)$ and investigate their behaviour for small Δx.**

Exercise:

Consider the function $f(x) = A\sin(kx)$.

We know that the derivative is $kA\cos(kx)$.

- **Show that a forward difference approximation gives**

$$f'_F(x) = kA\cos[k(x+\Delta x/2)] \cdot \left[\frac{\sin(k\Delta x/2)}{k\Delta x/2}\right]$$

- **Show that the centered difference approximation yields**

$$f'_C(x) = kA\cos[kx] \cdot \left[\frac{\sin(k\Delta x)}{k\Delta x}\right]$$

- **Compare these to the true derivative $f'(x)$ and investigate their behaviour for small Δx.**
- **Demonstrate thus that the centered difference is of higher order accuracy.**

Grid Resolution and Accuracy

The size of the gridstep Δx determines the accuracy of the numerical scheme.

For the simple sine function the error depended on $k\Delta x = 2\pi\Delta x/L$, that is, on the ratio of the grid size Δx to the wavelength L.

Grid Resolution and Accuracy

The size of the gridstep Δx determines the accuracy of the numerical scheme.

For the simple sine function the error depended on $k\Delta x = 2\pi\Delta x/L$, that is, on the ratio of the grid size Δx to the wavelength L.

For synoptic scale waves in the atmosphere a typical value of L is 1000 km.

Grid Resolution and Accuracy

The size of the gridstep Δx determines the accuracy of the numerical scheme.

For the simple sine function the error depended on $k\Delta x = 2\pi\Delta x/L$, that is, on the ratio of the grid size Δx to the wavelength L.

For synoptic scale waves in the atmosphere a typical value of L is 1000 km.

To make the ratio equal to 0.1 we need to have a grid size of about 100 km.

This is larger than the typical gridsizes used in operational NWP models.

Grid Resolution and Accuracy

The size of the gridstep Δx determines the accuracy of the numerical scheme.

For the simple sine function the error depended on $k\Delta x = 2\pi\Delta x/L$, that is, on the ratio of the grid size Δx to the wavelength L.

For synoptic scale waves in the atmosphere a typical value of L is 1000 km.

To make the ratio equal to 0.1 we need to have a grid size of about 100 km.

This is larger than the typical gridsizes used in operational NWP models.

The higher the resolution, that is, the smaller the grid-size, the heavier the computational burden.

There is a *trade-off between resolution and accuracy*.

The Leapfrog Method

We consider the equation describing the conservation of a quantity $Y(x,t)$ following the 1D motion of a fluid flow:

$$\frac{dY}{dt} \equiv \left(\frac{\partial Y}{\partial t} + u\,\frac{\partial Y}{\partial x}\right) = 0\,.$$

The Leapfrog Method

We consider the equation describing the conservation of a quantity $Y(x,t)$ following the 1D motion of a fluid flow:

$$\frac{dY}{dt} \equiv \left(\frac{\partial Y}{\partial t} + u \frac{\partial Y}{\partial x} \right) = 0\,.$$

If the velocity is taken to be constant, $u = c$, or if we linearise about a mean flow $\bar{u} = c$, the equation becomes

$$\frac{\partial Y}{\partial t} + c \frac{\partial Y}{\partial x} = 0\,.$$

The Leapfrog Method

We consider the equation describing the conservation of a quantity $Y(x,t)$ following the 1D motion of a fluid flow:

$$\frac{dY}{dt} \equiv \left(\frac{\partial Y}{\partial t} + u\frac{\partial Y}{\partial x}\right) = 0\,.$$

If the velocity is taken to be constant, $u = c$, or if we linearise about a mean flow $\bar{u} = c$, the equation becomes

$$\frac{\partial Y}{\partial t} + c\frac{\partial Y}{\partial x} = 0\,.$$

This is the *linear advection equation*.

The Leapfrog Method

We consider the equation describing the conservation of a quantity $Y(x,t)$ following the 1D motion of a fluid flow:

$$\frac{dY}{dt} \equiv \left(\frac{\partial Y}{\partial t} + u\,\frac{\partial Y}{\partial x}\right) = 0\,.$$

If the velocity is taken to be constant, $u = c$, or if we linearise about a mean flow $\bar{u} = c$, the equation becomes

$$\frac{\partial Y}{\partial t} + c\,\frac{\partial Y}{\partial x} = 0\,.$$

This is the *linear advection equation*.

It is analogous to a factor of the wave equation:

$$\left(\frac{\partial^2}{\partial t^2} - c^2\frac{\partial^2}{\partial x^2}\right)Y = \left(\frac{\partial}{\partial t} + c\frac{\partial}{\partial x}\right)\left(\frac{\partial}{\partial t} - c\frac{\partial}{\partial x}\right)Y = 0\,,$$

and its general solution is $Y = Y(x - ct)$.

Since the advection equation is linear, we can construct a general solution from Fourier components

$$Y = a \exp[ik(x - ct)]\,; \qquad k = 2\pi/L\,.$$

Since the advection equation is linear, we can construct a general solution from Fourier components

$$Y = a \exp[ik(x - ct)]\,; \qquad k = 2\pi/L\,.$$

We take the following *initial condition* for Y:

$$Y(x, 0) = a \exp[ikx]$$

Since the advection equation is linear, we can construct a general solution from Fourier components

$$Y = a \exp[ik(x - ct)] \, ; \qquad k = 2\pi / L \, .$$

We take the following *initial condition* for Y:

$$Y(x, 0) = a \exp[ikx]$$

Next, we approximate the differential equation by a finite difference equation (FDE) using centered differences for both the space and time derivatives.

Since the advection equation is linear, we can construct a general solution from Fourier components

$$Y = a \exp[ik(x - ct)]\,; \qquad k = 2\pi/L\,.$$

We take the following *initial condition* for Y:

$$Y(x, 0) = a \exp[ikx]$$

Next, we approximate the differential equation by a finite difference equation (FDE) using centered differences for both the space and time derivatives.

The continuous variables are replaced by discrete gridpoints at their integral values and the problem is solved on a finite difference grid.

Since the advection equation is linear, we can construct a general solution from Fourier components

$$Y = a\exp[ik(x-ct)]\,; \qquad k = 2\pi/L\,.$$

We take the following *initial condition* for Y:

$$Y(x,0) = a\exp[ikx]$$

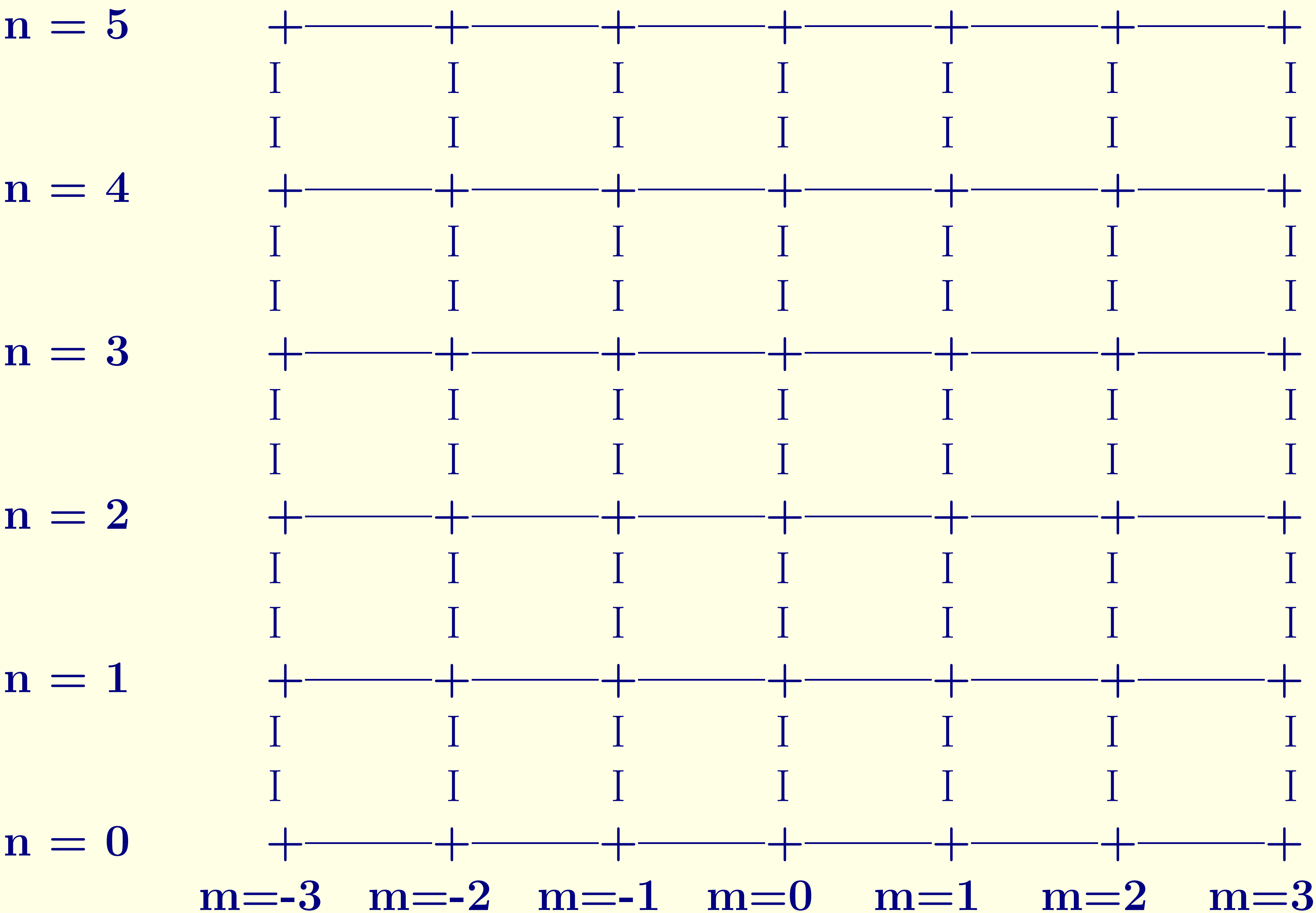
Next, we approximate the differential equation by a finite difference equation (FDE) using centered differences for both the space and time derivatives.

The continuous variables are replaced by discrete gridpoints at their integral values and the problem is solved on a finite difference grid.

Let the variables x and t be represented by the horizontal and vertical axes. Positive time corresponds to the upper half plane. The initial data occur on the x-axis.

Space-Time Grid:

Space axis horizontal
Time axis vertical



We denote the value of Y at a grid point by:

$$Y(m\Delta x, n\Delta t) = Y_m^n \,.$$

We denote the value of Y at a grid point by:

$$Y(m\Delta x, n\Delta t) = Y_m^n \,.$$

Then the (CTCS) finite difference approximation to the differential equation may be written as follows:

$$\left(\frac{Y_m^{n+1} - Y_m^{n-1}}{2\Delta t}\right) + c\left(\frac{Y_{m+1}^n - Y_{m-1}^n}{2\Delta x}\right) = 0 \,.$$

We denote the value of Y at a grid point by:

$$Y(m\Delta x, n\Delta t) = Y_m^n \,.$$

Then the (CTCS) finite difference approximation to the differential equation may be written as follows:

$$\left(\frac{Y_m^{n+1} - Y_m^{n-1}}{2\Delta t}\right) + c\left(\frac{Y_{m+1}^n - Y_{m-1}^n}{2\Delta x}\right) = 0 \,.$$

Solving for the value at time $(n+1)\Delta t$ gives

$$Y_m^{n+1} = Y_m^{n-1} - \left(\frac{c\Delta t}{\Delta x}\right)(Y_{m+1}^n - Y_{m-1}^n)$$

We denote the value of Y at a grid point by:

$$Y(m\Delta x, n\Delta t) = Y_m^n .$$

Then the (CTCS) finite difference approximation to the differential equation may be written as follows:

$$\left(\frac{Y_m^{n+1} - Y_m^{n-1}}{2\Delta t}\right) + c\left(\frac{Y_{m+1}^n - Y_{m-1}^n}{2\Delta x}\right) = 0 .$$

Solving for the value at time $(n+1)\Delta t$ gives

$$Y_m^{n+1} = Y_m^{n-1} - \left(\frac{c\Delta t}{\Delta x}\right)(Y_{m+1}^n - Y_{m-1}^n)$$

The value at the time $(n+1)\Delta t$ is obtained by adding a term to the value at $(n-1)\Delta t$; the method is known as the leapfrog method because of this leap over the time $n\Delta t$.

We denote the value of Y at a grid point by:

$$Y(m\Delta x, n\Delta t) = Y_m^n\,.$$

Then the (CTCS) finite difference approximation to the differential equation may be written as follows:

$$\left(\frac{Y_m^{n+1} - Y_m^{n-1}}{2\Delta t}\right) + c\left(\frac{Y_{m+1}^n - Y_{m-1}^n}{2\Delta x}\right) = 0\,.$$

Solving for the value at time $(n+1)\Delta t$ gives

$$Y_m^{n+1} = Y_m^{n-1} - \left(\frac{c\Delta t}{\Delta x}\right)(Y_{m+1}^n - Y_{m-1}^n)$$

The value at the time $(n+1)\Delta t$ is obtained by adding a term to the value at $(n-1)\Delta t$; the method is known as the leapfrog method because of this leap over the time $n\Delta t$.

The ratio $\mu \equiv \dfrac{c\Delta t}{\Delta x}$ will be found to be crucial.

Inter-dependency of Points

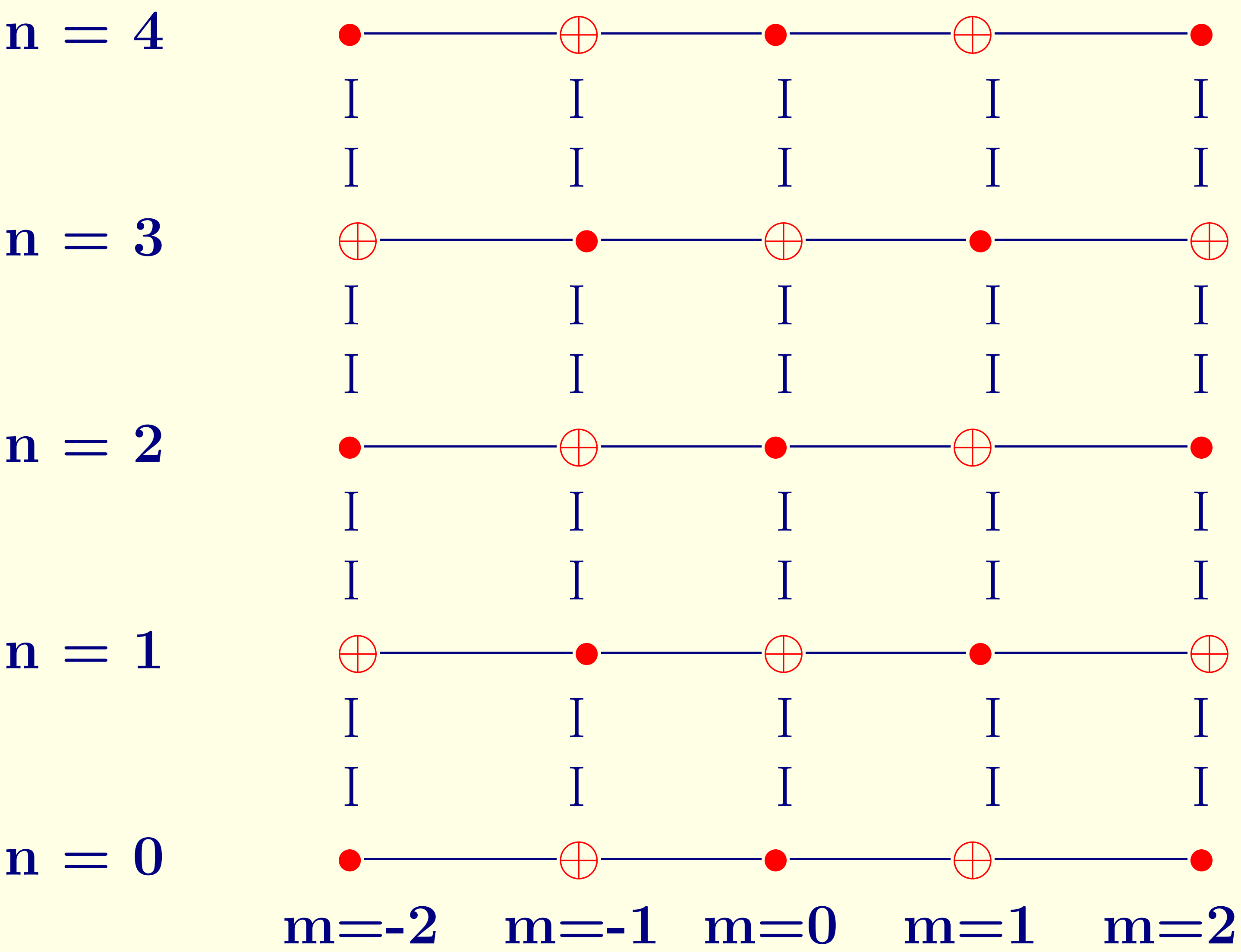
n + 1	$+$	$\oplus$	$+$
n	$\oplus$	$\bullet$	$\oplus$
n - 1	$+$	$\oplus$	$+$
	m-1	**m**	**m+1**

The evaluation of the equation at point $\bullet$ involves values of the variable at points $\oplus$. Solving for Y_m^{n+1} thus requires

$$Y_m^{n-1}, \qquad Y_{m-1}^n \qquad \textbf{and} \qquad Y_{m+1}^n .$$

The leapfrog scheme *splits the grid* into two independent sub-grids.

Grid Splitting



The finite difference grid splits into two sub-grids.

Steps must be taken to avoid divergence of the two solutions.

Recall the (CTCS) finite difference approximation:

$$\left(\frac{Y_m^{n+1} - Y_m^{n-1}}{2\Delta t}\right) + c\left(\frac{Y_{m+1}^{n} - Y_{m-1}^{n}}{2\Delta x}\right) = 0\,.$$

Recall the (CTCS) finite difference approximation:

$$\left(\frac{Y_m^{n+1} - Y_m^{n-1}}{2\Delta t}\right) + c\left(\frac{Y_{m+1}^n - Y_{m-1}^n}{2\Delta x}\right) = 0\,.$$

Solving for the value at time $(n+1)\Delta t$ **gives**

$$Y_m^{n+1} = Y_m^{n-1} - \left(\frac{c\Delta t}{\Delta x}\right)(Y_{m+1}^n - Y_{m-1}^n)$$

or, using the Courant Number,

$$Y_m^{n+1} = Y_m^{n-1} - \mu(Y_{m+1}^n - Y_{m-1}^n)$$

Recall the (CTCS) finite difference approximation:

$$\left(\frac{Y_m^{n+1} - Y_m^{n-1}}{2\Delta t}\right) + c\left(\frac{Y_{m+1}^n - Y_{m-1}^n}{2\Delta x}\right) = 0\,.$$

Solving for the value at time $(n+1)\Delta t$ gives

$$Y_m^{n+1} = Y_m^{n-1} - \left(\frac{c\Delta t}{\Delta x}\right)(Y_{m+1}^n - Y_{m-1}^n)$$

or, using the Courant Number,

$$Y_m^{n+1} = Y_m^{n-1} - \mu(Y_{m+1}^n - Y_{m-1}^n)$$

Assuming that we know the solution up to time $n\Delta t$, the values at time $(n+1)\Delta t$ can be calculated, and the solution advanced by one timestep in this way.

Recall the (CTCS) finite difference approximation:

$$\left(\frac{Y_m^{n+1} - Y_m^{n-1}}{2\Delta t}\right) + c\left(\frac{Y_{m+1}^n - Y_{m-1}^n}{2\Delta x}\right) = 0\,.$$

Solving for the value at time $(n+1)\Delta t$ **gives**

$$Y_m^{n+1} = Y_m^{n-1} - \left(\frac{c\Delta t}{\Delta x}\right)(Y_{m+1}^n - Y_{m-1}^n)$$

or, using the Courant Number,

$$Y_m^{n+1} = Y_m^{n-1} - \mu(Y_{m+1}^n - Y_{m-1}^n)$$

Assuming that we know the solution up to time $n\Delta t$**, the values at time** $(n+1)\Delta t$ **can be calculated, and the solution advanced by one timestep in this way.**

Then the whole procedure can be repeated to advance the solution to $(n+2)\Delta t$**, and so on.**

Question:

Under what conditions does the solution of the finite difference equation approximate that of the original differential equation?

$\star \qquad \star \qquad \star$

Question:

Under what conditions does the solution of the finite difference equation approximate that of the original differential equation?

$$\star \qquad \star \qquad \star$$

Intuitively, we would expect that a good approximation would be obtained provided the grid steps Δx and Δt are small enough.

Question:

Under what conditions does the solution of the finite difference equation approximate that of the original differential equation?

$\star \quad \star \quad \star$

Intuitively, we would expect that a good approximation would be obtained provided the grid steps Δx and Δt are small enough.

However, it turns out that this is not enough, and that the value of the ratio

$$\mu = \frac{c\Delta t}{\Delta x}$$

is found to be of critical importance.

Question:

Under what conditions does the solution of the finite difference equation approximate that of the original differential equation?

$\star \quad \star \quad \star$

Intuitively, we would expect that a good approximation would be obtained provided the grid steps Δx and Δt are small enough.

However, it turns out that this is not enough, and that the value of the ratio

$$\mu = \frac{c\Delta t}{\Delta x}$$

is found to be of critical importance.

This "*surprising result*" has important practical implications for operational NWP.

Question:

Under what conditions does the solution of the finite difference equation approximate that of the original differential equation?

$\star \qquad \star \qquad \star$

Intuitively, we would expect that a good approximation would be obtained provided the grid steps Δx and Δt are small enough.

However, it turns out that this is not enough, and that the value of the ratio

$$\mu = \frac{c\Delta t}{\Delta x}$$

is found to be of critical importance.

This "*surprising result*" has important practical implications for operational NWP.

Break here

The CFL Stability Criterion

Let us assume a solution of the FDE in the form

$$Y_m^n = a\,A^n \exp[ikm\Delta x]$$

The CFL Stability Criterion

Let us assume a solution of the FDE in the form

$$Y_m^n = a\,A^n \exp[ikm\Delta x]$$

Substituting this in the equation, defining $\mu = c\Delta t/\Delta x$ **and dividing by a common factor gives**

$$A^2 + (2i\mu \sin k\Delta x)A - 1 = 0$$

The CFL Stability Criterion

Let us assume a solution of the FDE in the form

$$Y_m^n = a\, A^n \exp[ikm\Delta x]$$

Substituting this in the equation, defining $\mu = c\Delta t/\Delta x$ **and dividing by a common factor gives**

$$A^2 + (2i\mu \sin k\Delta x)A - 1 = 0$$

This is a quadratic for the amplitude A**, with solutions**

$$A_{\pm} = -i\mu \sin k\Delta x \pm \sqrt{1 - \mu^2 \sin^2 k\Delta x}\,.$$

The CFL Stability Criterion

Let us assume a solution of the FDE in the form

$$Y_m^n = a\,A^n \exp[ikm\Delta x]$$

Substituting this in the equation, defining $\mu = c\Delta t/\Delta x$ **and dividing by a common factor gives**

$$A^2 + (2i\mu \sin k\Delta x)A - 1 = 0$$

This is a quadratic for the amplitude A**, with solutions**

$$A_{\pm} = -i\mu \sin k\Delta x \pm \sqrt{1 - \mu^2 \sin^2 k\Delta x}\,.$$

The roots of a quadratic equation may be either real or complex, depending on the value of the coefficients.

The CFL Stability Criterion

Let us assume a solution of the FDE in the form

$$Y_m^n = a\,A^n \exp[ikm\Delta x]$$

Substituting this in the equation, defining $\mu = c\Delta t/\Delta x$ **and dividing by a common factor gives**

$$A^2 + (2i\mu \sin k\Delta x)A - 1 = 0$$

This is a quadratic for the amplitude A**, with solutions**

$$A_\pm = -i\mu \sin k\Delta x \pm \sqrt{1 - \mu^2 \sin^2 k\Delta x}\,.$$

The roots of a quadratic equation may be either real or complex, depending on the value of the coefficients.

We write

$$A_\pm = -i\sigma \pm \sqrt{1-\sigma^2} \qquad \textbf{where} \qquad \sigma \equiv \mu \sin k\Delta x$$

We consider in turn the two cases.

Case I: $|\mu| \leq 1$

The quantity under the square-root sign is positive, so the modulus of A is given by

$$|A|^2 = (1 - \sigma^2) + \sigma^2 = 1\,.$$

Case I: $|\mu| \leq 1$

The quantity under the square-root sign is positive, so the modulus of A is given by

$$|A|^2 = (1 - \sigma^2) + \sigma^2 = 1\,.$$

The modulus of A is seen to be unity. Thus, we may write

$$A = \exp(i\psi) \qquad \textbf{where } \psi \textbf{ is real}\,.$$

Case I: $|\mu| \leq 1$

The quantity under the square-root sign is positive, so the modulus of A is given by

$$|A|^2 = (1 - \sigma^2) + \sigma^2 = 1\,.$$

The modulus of A is seen to be unity. Thus, we may write

$$A = \exp(i\psi) \qquad \textbf{where } \psi \textbf{ is real}\,.$$

Note that

$$\Re\{A_+\} = +\sqrt{1-\sigma^2} \qquad \Im\{A_+\} = -\sigma$$
$$\Re\{A_-\} = -\sqrt{1-\sigma^2} \qquad \Im\{A_+\} = -\sigma$$

Case I: $|\mu| \leq 1$

The quantity under the square-root sign is positive, so the modulus of A is given by

$$|A|^2 = (1 - \sigma^2) + \sigma^2 = 1\,.$$

The modulus of A is seen to be unity. Thus, we may write

$$A = \exp(i\psi) \qquad \textbf{where } \psi \textbf{ is real}\,.$$

Note that

$$\Re\{A_+\} = +\sqrt{1-\sigma^2} \qquad \Im\{A_+\} = -\sigma$$
$$\Re\{A_-\} = -\sqrt{1-\sigma^2} \qquad \Im\{A_+\} = -\sigma$$

The two values of the phase are

$$\psi_1 = -\arcsin \sigma$$

and

$$\psi_2 = \pi - \psi_1\,.$$

The solution of the equation may now be written

$$Y_m^n = \Big[D\exp(i\psi_1 n) + E\exp[i(-\psi_1+\pi)n]\Big]\exp(ikm\Delta x)$$

where D **and** E **are arbitrary constants.**

The solution of the equation may now be written

$$Y_m^n = \Big[D\exp(i\psi_1 n) + E\exp[i(-\psi_1 + \pi)n]\Big]\exp(ikm\Delta x)$$

where D and E are arbitrary constants.

Setting $n = 0$, this implies

$$Y_m^0 = [D + E]\exp(ikm\Delta x)$$

which, on account of the initial condition, means $D + E = a$.

The solution of the equation may now be written

$$Y_m^n = \Big[D\exp(i\psi_1 n) + E\exp[i(-\psi_1+\pi)n]\Big]\exp(ikm\Delta x)$$

where D and E are arbitrary constants.

Setting $n = 0$, this implies

$$Y_m^0 = [D + E]\exp(ikm\Delta x)$$

which, on account of the initial condition, means $D + E = a$.

Thus, we may write the solution as

$$Y_m^n = \underbrace{(a-E)\exp[ik(m\Delta x + \psi_1 n/k)]}_{\text{Physical Mode}} + \underbrace{(-1)^n E\exp[ik(m\Delta x - \psi_1 n/k)]}_{\text{Computational Mode}}$$

⋆ ⋆ ⋆

The solution of the equation may now be written

$$Y_m^n = \Big[D\exp(i\psi_1 n) + E\exp[i(-\psi_1 + \pi)n]\Big]\exp(ikm\Delta x)$$

where D and E are arbitrary constants.

Setting $n = 0$, this implies

$$Y_m^0 = [D + E]\exp(ikm\Delta x)$$

which, on account of the initial condition, means $D + E = a$.

Thus, we may write the solution as

$$Y_m^n = \underbrace{(a - E)\exp[ik(m\Delta x + \psi_1 n/k)]}_{\text{Physical Mode}} + \underbrace{(-1)^n E\exp[ik(m\Delta x - \psi_1 n/k)]}_{\text{Computational Mode}}$$

$\star \qquad \star \qquad \star$

Exercise:

Check in detail the algebra leading to this solution.

Once again, the solution is

$$Y_m^n = \underbrace{(a - E)\exp[ik(m\Delta x + \psi_1 n/k)]}_{\text{Physical Mode}} + \underbrace{(-1)^n E\exp[ik(m\Delta x - \psi_1 n/k)]}_{\text{Computational Mode}}$$

Once again, the solution is

$$Y_m^n = \underbrace{(a-E)\exp[ik(m\Delta x + \psi_1 n/k)]}_{\text{Physical Mode}} + \underbrace{(-1)^n E \exp[ik(m\Delta x - \psi_1 n/k)]}_{\text{Computational Mode}}$$

The first term of this solution is called the physical mode and corresponds to the solution of the differential equation.

Once again, the solution is

$$Y_m^n = \underbrace{(a - E)\exp[ik(m\Delta x + \psi_1 n/k)]}_{\text{Physical Mode}} + \underbrace{(-1)^n E\exp[ik(m\Delta x - \psi_1 n/k)]}_{\text{Computational Mode}}$$

The first term of this solution is called the physical mode and corresponds to the solution of the differential equation.

The second term is called the computational mode.

Once again, the solution is

$$Y_m^n = \underbrace{(a-E)\exp[ik(m\Delta x + \psi_1 n/k)]}_{\text{Physical Mode}} + \underbrace{(-1)^n E \exp[ik(m\Delta x - \psi_1 n/k)]}_{\text{Computational Mode}}$$

The first term of this solution is called the physical mode and corresponds to the solution of the differential equation.

The second term is called the computational mode.

It arises through the use of centered differences resulting in the approximation of a first order differential equation by a second order difference equation (with an extra solution).

Once again, the solution is

$$Y_m^n = \underbrace{(a - E)\exp[ik(m\Delta x + \psi_1 n/k)]}_{\text{Physical Mode}} + \underbrace{(-1)^n E\exp[ik(m\Delta x - \psi_1 n/k)]}_{\text{Computational Mode}}$$

The first term of this solution is called the physical mode and corresponds to the solution of the differential equation.

The second term is called the computational mode.

It arises through the use of centered differences resulting in the approximation of a first order differential equation by a second order difference equation (with an extra solution).

For μ and $k\Delta x$ small we have

$$\psi_1 \approx -\mu k\Delta x = -kc\Delta t \qquad \textbf{and} \qquad \psi_2 \approx \pi + \mu k\Delta x = \pi + kc\Delta t$$

Once again, the solution is

$$Y_m^n = \underbrace{(a-E)\exp[ik(m\Delta x + \psi_1 n/k)]}_{\text{Physical Mode}} + \underbrace{(-1)^n E\exp[ik(m\Delta x - \psi_1 n/k)]}_{\text{Computational Mode}}$$

The first term of this solution is called the physical mode and corresponds to the solution of the differential equation.

The second term is called the computational mode.

It arises through the use of centered differences resulting in the approximation of a first order differential equation by a second order difference equation (with an extra solution).

For μ and $k\Delta x$ small we have

$$\psi_1 \approx -\mu k\Delta x = -kc\Delta t \qquad \textbf{and} \qquad \psi_2 \approx \pi + \mu k\Delta x = \pi + kc\Delta t$$

If the ratio μ is small, the physical mode solution is given approximately by

$$Y \approx a\exp[ik(m\Delta x - cn\Delta t)]$$

which is just the analytical solution.

The centered difference approximation cannot be used for the first timestep: We don't know the values at time $t = -\Delta t$.

The centered difference approximation cannot be used for the first timestep: We don't know the values at time $t = -\Delta t$.

Instead, we must use another approximation for the first step, typically an uncentered forward time step.

The centered difference approximation cannot be used for the first timestep: We don't know the values at time $t = -\Delta t$.

Instead, we must use another approximation for the first step, typically an uncentered forward time step.

In essence, this amounts to specifing another *"initial contition"*, the computational initial condition, at $t = \Delta t$.

The centered difference approximation cannot be used for the first timestep: We don't know the values at time $t = -\Delta t$.

Instead, we must use another approximation for the first step, typically an uncentered forward time step.

In essence, this amounts to specifing another *"initial contition"*, the computational initial condition, at $t = \Delta t$.

The value of the additional "initial condition" determines the amplitude of the computational mode. It should be chosen to minimize this.

The centered difference approximation cannot be used for the first timestep: We don't know the values at time $t = -\Delta t$.

Instead, we must use another approximation for the first step, typically an uncentered forward time step.

In essence, this amounts to specifing another *"initial contition"*, the computational initial condition, at $t = \Delta t$.

The value of the additional "initial condition" determines the amplitude of the computational mode. It should be chosen to minimize this.

The above solution implies

$$
\begin{aligned}
Y_m^0 &= a \exp(ikm\Delta x) \\
Y_m^1 &= [a \exp(i\psi_1) - 2E \cos\psi_1] \exp(ikm\Delta x)
\end{aligned}
$$

The centered difference approximation cannot be used for the first timestep: We don't know the values at time $t = -\Delta t$.

Instead, we must use another approximation for the first step, typically an uncentered forward time step.

In essence, this amounts to specifing another *"initial contition"*, the computational initial condition, at $t = \Delta t$.

The value of the additional "initial condition" determines the amplitude of the computational mode. It should be chosen to minimize this.

The above solution implies

$$
\begin{aligned}
Y_m^0 &= a \exp(ikm\Delta x) \\
Y_m^1 &= [a \exp(i\psi_1) - 2E \cos\psi_1] \exp(ikm\Delta x)
\end{aligned}
$$

Requiring $E = 0$, we find that $Y_m^1 = \exp(i\psi_1) Y_m^0$.

The centered difference approximation cannot be used for the first timestep: We don't know the values at time $t = -\Delta t$.

Instead, we must use another approximation for the first step, typically an uncentered forward time step.

In essence, this amounts to specifing another *"initial contition"*, the computational initial condition, at $t = \Delta t$.

The value of the additional "initial condition" determines the amplitude of the computational mode. It should be chosen to minimize this.

The above solution implies

$$\begin{aligned} Y_m^0 &= a \exp(ikm\Delta x) \\ Y_m^1 &= [a \exp(i\psi_1) - 2E\cos\psi_1] \exp(ikm\Delta x) \end{aligned}$$

Requiring $E = 0$, we find that $Y_m^1 = \exp(i\psi_1) Y_m^0$.

In this simple case, we can eliminate the computational mode. In general, it is much more difficult.

Case II: $|\mu| > 1$

Recall that the roots of the quadratic are

$$A_{\pm} = -i\sigma \pm \sqrt{1-\sigma^2} \qquad \textbf{where} \qquad \sigma \equiv \mu \sin k\Delta x$$

Case II: $|\mu| > 1$

Recall that the roots of the quadratic are

$$A_\pm = -i\sigma \pm \sqrt{1-\sigma^2} \qquad \textbf{where} \qquad \sigma \equiv \mu \sin k\Delta x$$

If $|\mu| > 1$, there will be some wavelengths for which

$$\sigma^2 = \mu^2 \sin^2 k\Delta x > 1 \,.$$

Case II: $|\mu| > 1$

Recall that the roots of the quadratic are

$$A_\pm = -i\sigma \pm \sqrt{1-\sigma^2} \qquad \textbf{where} \qquad \sigma \equiv \mu \sin k\Delta x$$

If $|\mu| > 1$, there will be some wavelengths for which

$$\sigma^2 = \mu^2 \sin^2 k\Delta x > 1\,.$$

Then the two roots of the quadratic are pure imaginary

$$A = i\left(-\sigma \pm \sqrt{\sigma^2 - 1}\right)$$

Case II: $|\mu| > 1$

Recall that the roots of the quadratic are

$$A_\pm = -i\sigma \pm \sqrt{1-\sigma^2} \qquad \textbf{where} \qquad \sigma \equiv \mu \sin k\Delta x$$

If $|\mu| > 1$, there will be some wavelengths for which

$$\sigma^2 = \mu^2 \sin^2 k\Delta x > 1\,.$$

Then the two roots of the quadratic are pure imaginary

$$A = i\left(-\sigma \pm \sqrt{\sigma^2 - 1}\right)$$

Therefore either $|A_+| > 1$ or $|A_-| > 1$, i.e., the modulus of one of the roots will exceed unity.

In that case the amplitude of the solution of the finite difference equation will grow without bound for large time.

Case II: $|\mu| > 1$

Recall that the roots of the quadratic are

$$A_{\pm} = -i\sigma \pm \sqrt{1-\sigma^2} \qquad \textbf{where} \qquad \sigma \equiv \mu \sin k\Delta x$$

If $|\mu| > 1$, there will be some wavelengths for which

$$\sigma^2 = \mu^2 \sin^2 k\Delta x > 1\,.$$

Then the two roots of the quadratic are pure imaginary

$$A = i\left(-\sigma \pm \sqrt{\sigma^2 - 1}\right)$$

Therefore either $|A_+| > 1$ or $|A_-| > 1$, i.e., the modulus of one of the roots will exceed unity.

In that case the amplitude of the solution of the finite difference equation will grow without bound for large time.

This phenomenon is called computational instability.

In case of *computational instability*, the solution of the finite difference equation cannot possibly resemble the physical solution.

In case of *computational instability*, the solution of the finite difference equation cannot possibly resemble the physical solution.

The physical solution remains of constant amplitude for all time. The numerical solution grows without limit with time.

In case of *computational instability*, the solution of the finite difference equation cannot possibly resemble the physical solution.

The physical solution remains of constant amplitude for all time. The numerical solution grows without limit with time.

We thus require that $|\mu| \le 1$. This condition for stability is known as the CFL Criterion:

$$\frac{c\Delta t}{\Delta x} \le 1$$

after Courant, Friedichs and Lewy (1928), who first published the result.

In case of *computational instability*, the solution of the finite difference equation cannot possibly resemble the physical solution.

The physical solution remains of constant amplitude for all time. The numerical solution grows without limit with time.

We thus require that $|\mu| \le 1$. This condition for stability is known as the CFL Criterion:

$$\frac{c\Delta t}{\Delta x} \le 1$$

after Courant, Friedichs and Lewy (1928), who first published the result.

It implies that, <u>if we refine the space grid</u>, that is, decrease Δx, <u>we must also shorten the time step Δt</u>.

In case of *computational instability*, the solution of the finite difference equation cannot possibly resemble the physical solution.

The physical solution remains of constant amplitude for all time. The numerical solution grows without limit with time.

We thus require that $|\mu| \le 1$. This condition for stability is known as the CFL Criterion:

$$\frac{c\Delta t}{\Delta x} \le 1$$

after Courant, Friedichs and Lewy (1928), who first published the result.

It implies that, <u>if we refine the space grid</u>, that is, decrease Δx, <u>we must also shorten the time step Δt</u>.

Thus, halving the grid size in a two dimensional domain results in an eightfold increase in computation time.

Unconditionally Stable Schemes.

A large part of the research effort in Met Éireann recently has been devoted to the development of integration schemes which are free of the CFL constraint.

Unconditionally Stable Schemes.

A large part of the research effort in Met Éireann recently has been devoted to the development of integration schemes which are free of the CFL constraint.

The semi-Lagrangian scheme for advection is based on the idea of approximating the Lagrangian form of the time derivative.

Unconditionally Stable Schemes.

A large part of the research effort in Met Éireann recently has been devoted to the development of integration schemes which are free of the CFL constraint.

The semi-Lagrangian scheme for advection is based on the idea of approximating the Lagrangian form of the time derivative.

It is so formulated that the numerical domain of dependence always includes the physical domain of dependence. This necessary condition for stability is satisfied automatically by the scheme.

Unconditionally Stable Schemes.

A large part of the research effort in Met Éireann recently has been devoted to the development of integration schemes which are free of the CFL constraint.

The semi-Lagrangian scheme for advection is based on the idea of approximating the Lagrangian form of the time derivative.

It is so formulated that the numerical domain of dependence always includes the physical domain of dependence. This necessary condition for stability is satisfied automatically by the scheme.

The semi-Lagrangian algorithm has enabled us to integrate the primitive equations using a time step of 15 minutes.

This can be compared to a typical timestep of 2.5 minutes for Eulerian schemes.

The consequential saving of computation time means that the operational numerical guidance is available to the forecasters much earlier than would otherwise be the case.

The consequential saving of computation time means that the operational numerical guidance is available to the forecasters much earlier than would otherwise be the case.

Lagrangian time-stepping is now used in the majority of global and regional NWP Models.

The consequential saving of computation time means that the operational numerical guidance is available to the forecasters much earlier than would otherwise be the case.

Lagrangian time-stepping is now used in the majority of global and regional NWP Models.

The Irish Meteorological Service (now Met Éireann) was the first NWP centre to implement such a scheme in an operational setting.

The consequential saving of computation time means that the operational numerical guidance is available to the forecasters much earlier than would otherwise be the case.

Lagrangian time-stepping is now used in the majority of global and regional NWP Models.

The Irish Meteorological Service (now Met Éireann) was the first NWP centre to implement such a scheme in an operational setting.

We discuss semi-Lagrangian schemes in a later lecture.

End of §3.2.2