Introduction to Cloud Modeling. Part II: Numerical Methods

Vaughan Phillips

Course on “atmospheric aerosols and clouds with introduction to process oriented modeling”,
Sao Paulo University

Outline

» Introduction

» Finite differences: theory

» Finite differences: examples

» Computational stability

» Summary

INTRODUCTION

Introduction

Cloud modeling:
Aim to predict future state of the clouds and atmospheric circulation from knowledge of present state by using numerical approximations to the dynamical and physical evolution equations

FINITE DIFFERENCES: THEORY

Finite differences

The equations of motion, energy and mass conservation cannot be solved analytically.

They must be approximated and then solved numerically.

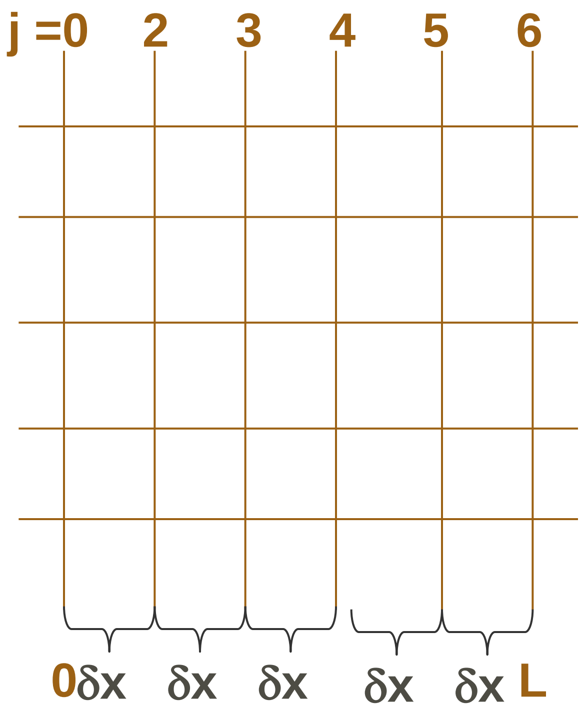
For this approximation, a discretization method is used.

Simplest form: finite difference method

Finite differences

Consider a field variable u(x).

- u(x): solution to a differential equation in the interval $0 \le x \le L$
- the interval can be devided into J (j=0,2,…,J) equally distanced subintervals of length $\delta x = L/J$



Finite differences

Consider a field variable u(x).

- now u(x).can be approximated by a set of J+1 values as

$$U_j = u(j\delta x)$$

which are the values at the J+1 grid points which are given by

$$x = j\delta x, \quad j = 1,2,\ldots,J$$

- if δx is sufficiently small (relative to length-scale of variations in u), then all of the J+1 grid points may provide a good approximation to u(x)

Finite differences

Expressions for derivatives

- Consider the Taylor series expansions about the point x_0:

$$u(x_0+\delta x)=u(x_0)+u'(x_0)\delta x+u''(x_0)\frac{(\delta x)^2}{2}+u'''(x_0)\frac{(\delta x)^3}{6}+O[(\delta x)^4]$$

$$u(x_0-\delta x)=u(x_0)-u'(x_0)\delta x+u''(x_0)\frac{(\delta x)^2}{2}-u'''(x_0)\frac{(\delta x)^3}{6}+O[(\delta x)^4]$$

()‘: differentiation with respect to x

$O[(\delta x)^4]$: terms with order of $(\delta x)^4$ and terms smaller than this are neglected

1st derivative: adding $u(x_0+\delta x)$ and $u(x_0-\delta x)$ and solve for *u‘*

2nd derivative: subtracting $u(x_0-\delta x)$ from $u(x_0+\delta x)$ and solve for *u“*

Finite differences

expressing derivatives

1st derivative: adding $u(x_0+\delta x)$ and $u(x_0-\delta x)$ and solve for u'

$$u'(x_0) = \frac{u(x_0+\delta x) - u(x_0-\delta x)}{2\delta x} + O[(\delta x)^2]$$

2nd derivative: subtracting $u(x_0-\delta x)$ from $u(x_0+\delta x)$ and solve for u''

$$u''(x_0) = \frac{u(x_0+\delta x) - 2u(x_0) + u(x_0-\delta x)}{(\delta x)^2} + O[(\delta x)^2]$$

centered differences

note: Centered differences neglect terms of order $(\delta x)^2$ and higher. The truncation error is of order $(\delta x)^2$.

assessing the limits of accuracy

- any field resolved on a grid can be approximated by a finite Fourier series expansion:

$$u(x) \approx \frac{a_0}{2} + \sum_{m=1}^{J/2} \left[a_m \cos \frac{2\pi\, mx}{L} + b_m \sin \frac{2\pi\, mx}{L} \right]$$

- One can determine a_0, a_m and b_m for wave numbers m=1,2,…,J/2, because J+1 values of u_j together determine the J+1 coefficients in this approximation.

limits of accuracy

- Shortest wavelength component in this approximation has a wavelength of

$$\frac{L}{m} = \frac{2L}{J} = 2\delta x$$

- This is the shortest wave that may be resolved with the FD scheme.

accurate representation is only possible for wavelengths greatly exceeding 2δx

FINITE DIFFERENCES: EXAMPLES

Finite differences

different methods:

centered differences:

$$\left(\frac{du}{dx}\right)_j \rightarrow \frac{u_{j+1} - u_{j-1}}{2\delta x}$$

forward differentiation:

$$\left(\frac{du}{dx}\right)_j \rightarrow \frac{u_{j+1} - u_j}{\delta x}$$

backward differentiation:

$$\left(\frac{du}{dx}\right)_j \rightarrow \frac{u_j - u_{j-1}}{\delta x}$$

Finite differences

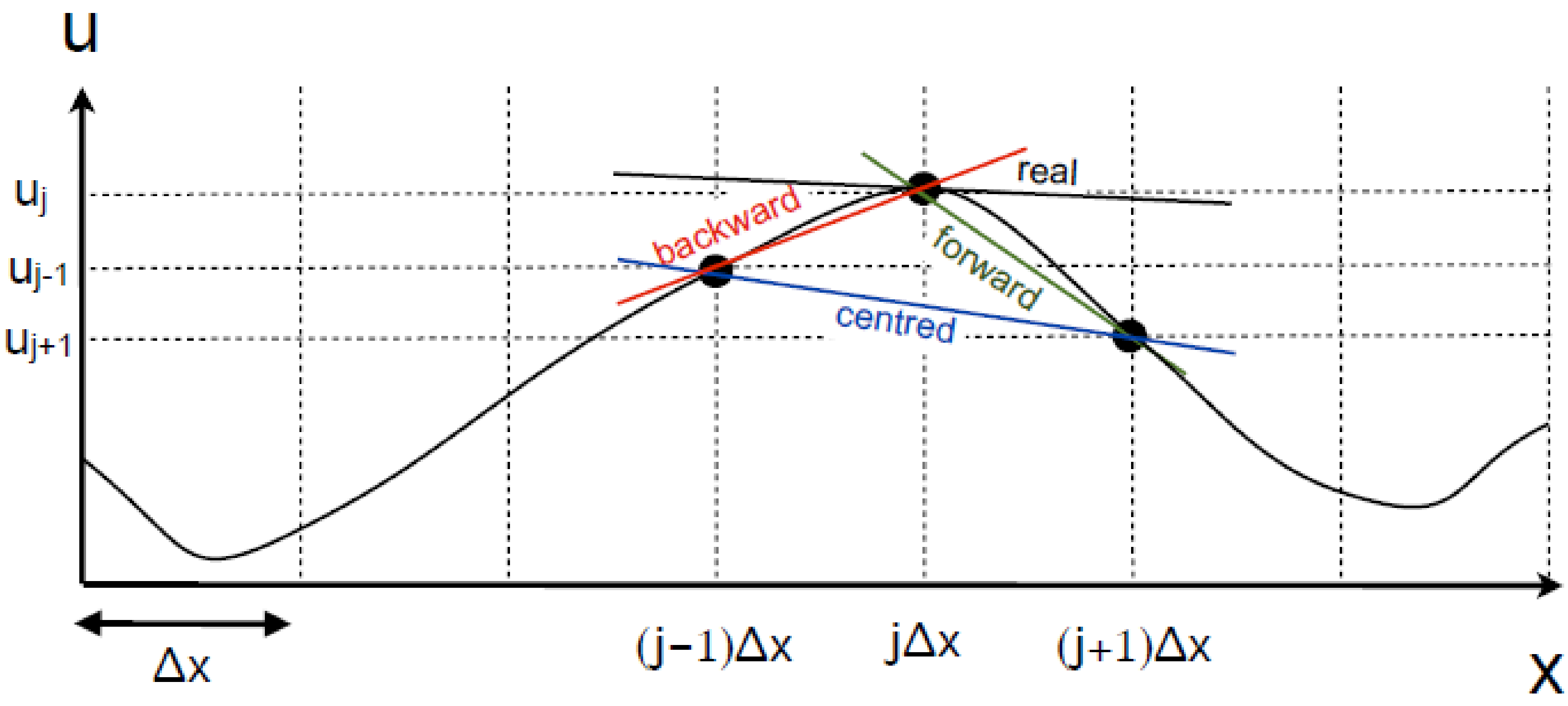


Figure 3.2: *The backward, forward and centred finite differences of a function* $u(x)$*, defined in the grid points* $x = j\Delta x$ *so that* $u_j = u_j(j\Delta x)$*, where* Δx *is the grid length and* $j = 0, 1, 2, \ldots$ *are integers.*

Finite differences

centered differences: explicit time differencing

example: linear 1D advection equation

$$\frac{\partial q}{\partial t} + c\frac{\partial q}{\partial x} = 0$$

c: specified speed

q(x,0): a known initial condition

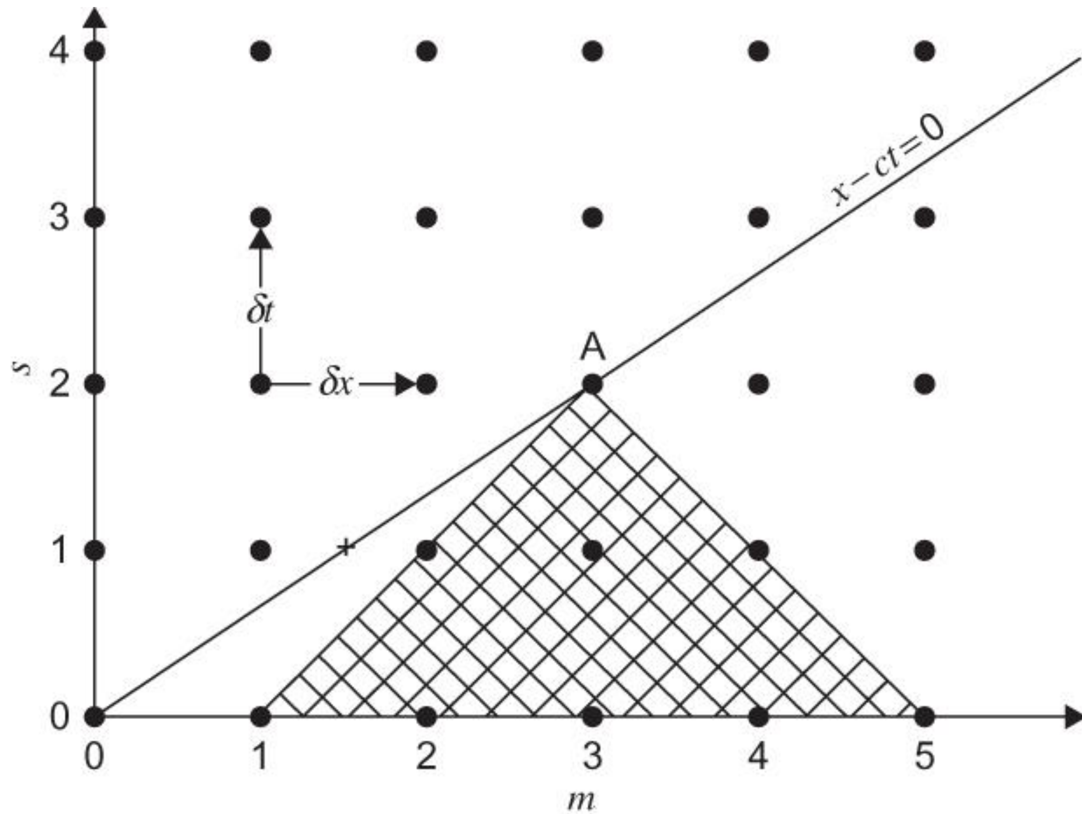
2nd order approximation in *x* and *t* using centered differences:

$$\frac{q(x,t+\delta t) - q(x,t-\delta t)}{2\delta t} = -c\frac{q(x+\delta x,t) - q(x-\delta x,t)}{2\delta x} \qquad \textbf{(1)}$$

Finite differences

centered differences: explicit time differencing

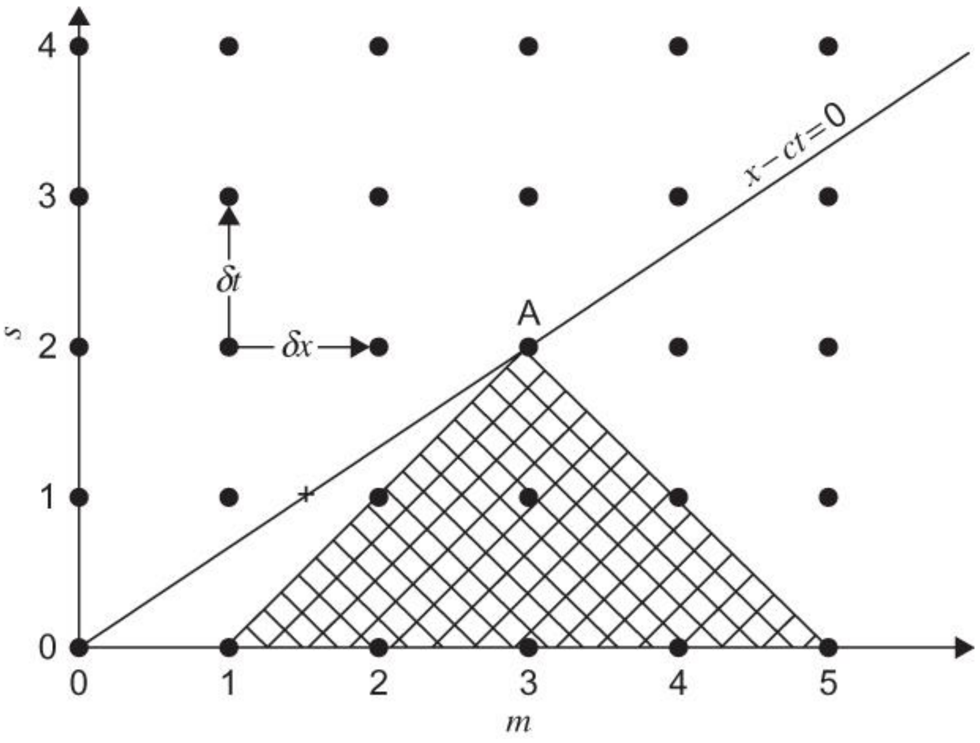
- this set of algebraic equations can be solved to determine solutions for a finite set of points that define a grid mesh in x and t



Finite differences

centered differences: explicit time differencing

- this set of algebraic equations can be solved to determine solutions for a finite set of points that define a grid mesh in x and t



with

$x = m\,\delta x$, $m=0,1,2,\ldots,M$

$t = s\,\delta t$, $s=0,1,2,\ldots,S$

we can define

$$\hat{q}_{m,s} \equiv q(m\,\delta x, s\,\delta t)$$

Finite differences

centered differences: explicit time differencing

then (1) can be written as

$$\hat{q}_{m,s+1} - \hat{q}_{m,s-1} = -\sigma\left(\hat{q}_{m+1,s} - \hat{q}_{m-1,s}\right)$$

with $\sigma \equiv c\dfrac{\delta t}{\delta x}$ Courant number

note: - This form of time differencing is referred to as leapfrog method (the value at time *s* is given by the difference in values computed for time steps s+1 and s-1)

-The leapfrog method cannot be used at the initial time t=0 (s=0); unknown $\hat{q}_{m,s-1}$

- There and alternative method such as forward difference approximation is required.

Finite differences

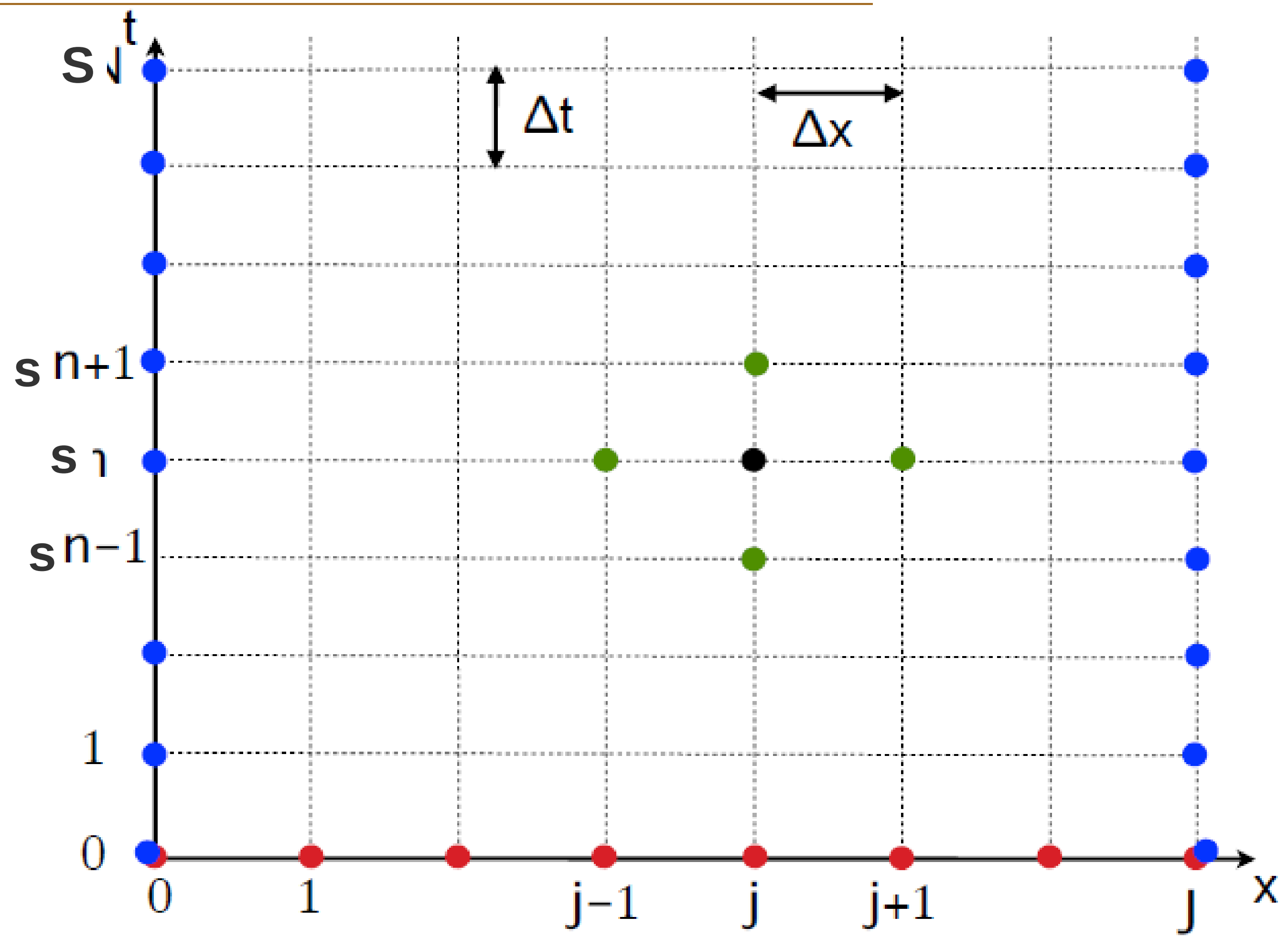


Figure 4.1: *A finite difference grid in time and space. The grid points used by the leap-frog scheme are shown in green . The red points are the necessary initial condition points and the blue points illustrate here two boundaries of the domain, which have to be prescribed.*

COMPUTATIONAL STABILITY

computational stability

- Using finite difference appoximations will not always resemble solutions to the original differential equations, but
- the solutions depend on the computational stability of the difference equations.

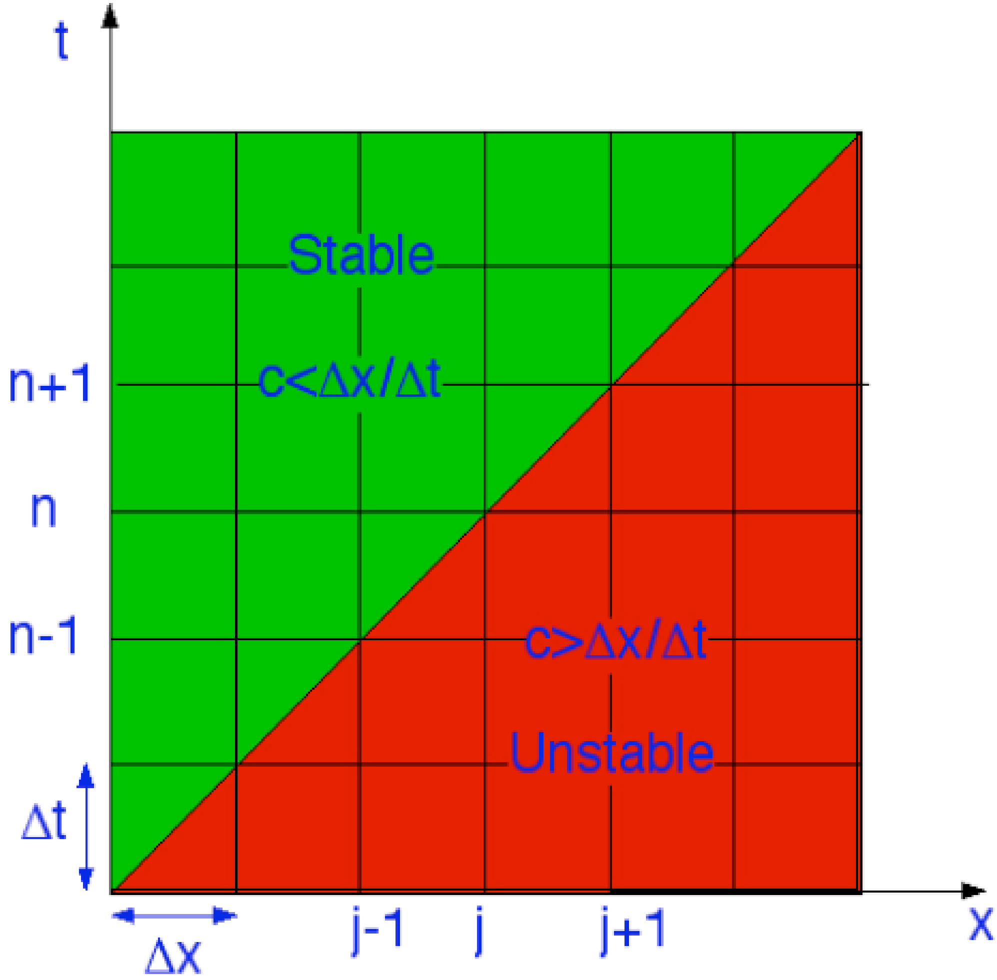
computational unstable: numerical solution will growth exponentially in time even if the original differential equation system has solutions whose amplitudes remain constant in time.

Courant-Friedrichs-Levy (CFL) stability criterion:

A difference equation is computationally stable if for a given space increment δx, the time step δt must be chosen that the dependent field will be advected a distance less than one grid length per time step, i.e. the Courant number is smaller or equal one:

$$\sigma \equiv c \frac{\delta t}{\delta x} \leq 1$$

computational stability



The Courant-Fredrichs-Lewy (CFL) stability criterion for centred schemes in time and space.

SUMMARY

- Finite differences are approximations created by Taylor expansion
- Stability: CFL condition
  - Timestep must be small enough, in view of resolution and flow-speed, if numerical solution is to be stable
- Accuracy
  - Timestep and resolution must be fine enough

Obrigado