Numerical Methods in ESP

Numerical Methods II

Graziano Giuliani

International Centre for Theoretical Physics

Second Semester 2023-24

`/afs/ictp.it/public/g/ggiulian/WORLD/num2_lesson1.pdf`

Course Outline

- Numerical approximations and the finite difference method for ODE
- 1D Linear advection
- 1D Heat equation
- Thomas algorithm and linear systems (TDMA)
- Basic data analysis and statistic (M. V. Guarino)

Typical morning schedule

- Lesson 45 minutes
- Exercise 45 minutes
  - Any programming language accepted.
  - Resource for python: `https://jupyter.ictp.it`

Send codes by email to: `ggiulian@ictp.it` `mguarino@ictp.it`

References

- Durran D. R. (1999) Numerical Methods for Wave Equations in Geophysical Fluid Dynamics. New York, Springer-Verlag.
- William Menke, Environmental Data Analysis with MATLAB or Python: Principles, Applications, and Prospects, Elsevier

Numerical Solutions

Differences between analytical and numerical solution methods

- Analytical solutions are:
  - exact
  - difficult to find (available only for very simple special cases)
- Numerical solutions are:
  - approximate W.R.T. the original problem
  - always particular to a given set of parameters (discretization parameters e.g. $\Delta x, \Delta t$)
- the difference between the numerical and the true solution is the error of the numerical solution

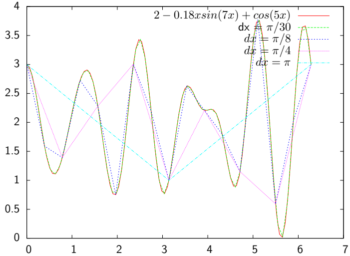
Computer floating points

Floating points numbers are NOT real numbers:

- Finite number of bits $16 - 32 - 64 - 128$:
  - Base β and precision p
    - With $\beta = 10$ and $p = 3$: $0.1 \rightarrow 1.00 \times 10^{-1}$
    - With $\beta = 2$ and $p = 24$: $0.1 \rightarrow 1.10011001100110011001101 \times 2^{-4}$
- Round-off error
- No associative or distributive properties
  - $(a + b) * c \neq a * c + b * c$
  - $(a + b) + c \neq a + (b + c)$
- IEEE standard for Basic Algorithms
  - Machine ϵ
    - 32 bit: $\epsilon = 1.19209290E - 07$
    - 64 bit: $\epsilon = 2.22044604925031308E - 16$
- Out of range numbers
  - Infinity
  - De-normalized number

Discretization

- Discretization: *continuous* functions are approximated by a finite set of degrees of freedom e.g. point values.
- Generally speaking, the more values are used to approximate a function, the more accurate the approximation will be.



- The resolution required to reduce discretization error to an acceptable level depends on smoothness of the function to be approximated.

Finite differences

- Finite difference approximation is based on truncated Taylor series. Given $x_j = j\Delta x, j = 1, 2, \ldots$ a function can be approximated in a given point x as:

$$f(x) = f(x_j) + (x - x_j)f'(x_j) + \frac{(x - x_j)^2}{2}f''(x_j) + \frac{(x - x_j)^3}{6}f'''(x_j) + \mathcal{O}(x - x_j)^4 \tag{1}$$

- Truncating [1] at second order and expanding in $\pm\Delta x$:

$$f(x_{j+1}) = f(x_j) + \Delta x f'(x_j) + \mathcal{O}(\Delta x)^2 \tag{2}$$

$$f(x_{j-1}) = f(x_j) - \Delta x f'(x_j) + \mathcal{O}(\Delta x)^2 \tag{3}$$

Approximation of first derivative

- Rearranging [2] and [3]:

$$f'(x_j) = \frac{f(x_{j+1}) - f(x_j)}{\Delta x} + \mathcal{O}(\Delta x) \tag{4}$$

$$f'(x_j) = \frac{f(x_j) - f(x_{j-1})}{\Delta x} + \mathcal{O}(\Delta x) \tag{5}$$

These are first order accurate approximations to the first derivative of a function in a given point and are known from the positions of the points used for the approximation as *forward difference* [4] and *backward difference* [5]. Truncating [1] at third order and expanding in $\pm\Delta x$ and then subtracting the two equation we obtain instead:

$$f'(x_j) = \frac{f(x_{j+1}) - f(x_{j-1})}{2\Delta x} + \mathcal{O}(\Delta x)^2 \tag{6}$$

or the *centered difference* second order accurate approximation to the first derivative of a function in a given point.

Approximation of second derivative

- Truncating [1] at fourth order and expanding in $\pm\Delta x$:

$$f(x_{j+1}) = f(x_j) + \Delta x f'(x_j) + \frac{\Delta x^2}{2} f''(x_j) + \frac{\Delta x^3}{6} f'''(x_j) + \mathcal{O}(\Delta x)^4 \tag{7}$$

$$f(x_{j-1}) = f(x_j) - \Delta x f'(x_j) + \frac{\Delta x^2}{2} f''(x_j) - \frac{\Delta x^3}{6} f'''(x_j) + \mathcal{O}(\Delta x)^4 \tag{8}$$

- Adding together [7] and [8]:

$$f''(x_j) = \frac{f(x_{j+1}) - 2f(x_j) + f(x_{j-1})}{\Delta x^2} + \mathcal{O}(\Delta x^2) \tag{9}$$

or the second order accurate approximation to the second derivative of a function in a given point.

Accuracy is related to the convergence velocity of the approximation to the true value when the distance $\Delta x \rightarrow 0$. Second order accuracy grants the approximation to converge faster than the first order ones, or equivalently to be nearer the true value with a fixed value of Δx.

Linear ordinary differential equation 1

- Consider a linear ordinary differential equation:

$$\frac{dy}{dt} = -\lambda y \tag{10}$$

  where $\lambda > 0$, and initial condition $y(0) = y_0$.
- The exact solution is : $y(t) = y(0)e^{-\lambda t}$.

We will solve this equation numerically using the approximations to the derivatives. Let us introduce the following notation:

$$\begin{aligned} n &= 1, 2, \ldots \\ t_n &= n\Delta t \\ y^n &\approx y(t_n) \end{aligned} \tag{11}$$

$$\text{Truncation error} \Rightarrow e^n = y(t_n) - y^n \tag{12}$$

Linear ordinary differential equation 2

The equation [10] can be approximated with FORWARD IN TIME differencing as:

$$\frac{y^{n+1} - y^n}{\Delta t} = -\lambda y^n \tag{13}$$

Thus:

$$y^{n+1} = (1 - \lambda \Delta t) y^n \tag{14}$$

The equation [14] provides a recursive formula to solve the linear ordinary differential equation for any time given the value of the initial condition at time t_0.

Convergence and Stability

- A finite difference numerical scheme for the solution of a linear differential equation is *consistent* if the truncation error of the scheme approaches zero as the interval $\Delta t \to 0$.
- Consistency is not granting nevertheless the *convergence*, i.e. the fact that the distance between the numerical scheme solution of the linear differential equation and the true solution of the same problem approaches zero as $\Delta t \to 0$.
- *Lax equivalence theorem* states that for a consistent, linear method, *stability* is the necessary and sufficient condition for the convergence.

In other words, we must keep an eye on the accumulation of the error. Even if for a small timestep Δt the error ϵ is small, it can accumulate with time rapidly and give a solution divergent from the true one. In this case the numerical method is said to be *unstable*. For a non-linear differential equation, the stability condition of the linearized form is still necessary, but not sufficient for the convergence.

Stability analysis for a linear ODE 1

- Let us assume that the approximate numerical solution of [10] is of the form:

$$y^n = A^p \tag{15}$$

where the factor A is called the *amplification factor*.

- The solution of the numerical scheme in [14] is thus:

$$y^{n+1} = A^p(1 - \lambda\Delta t) \tag{16}$$

or, given the [15]:

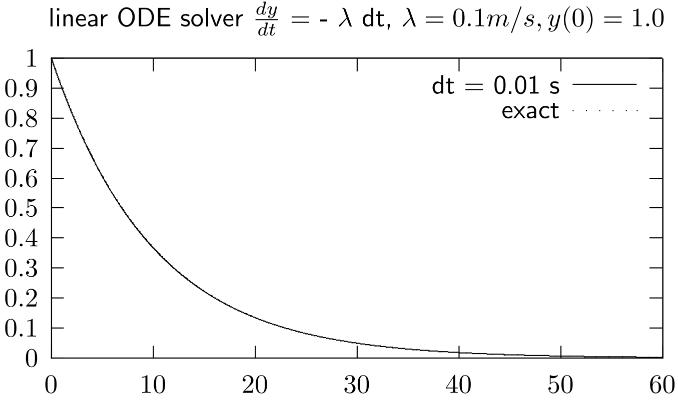
$$A^{p+1} = A^p(1 - \lambda\Delta t) \Rightarrow A = (1 - \lambda\Delta t) \tag{17}$$

- The numerical scheme is stable if: $|A| \leq 1$

Stability analysis for a linear ODE 2

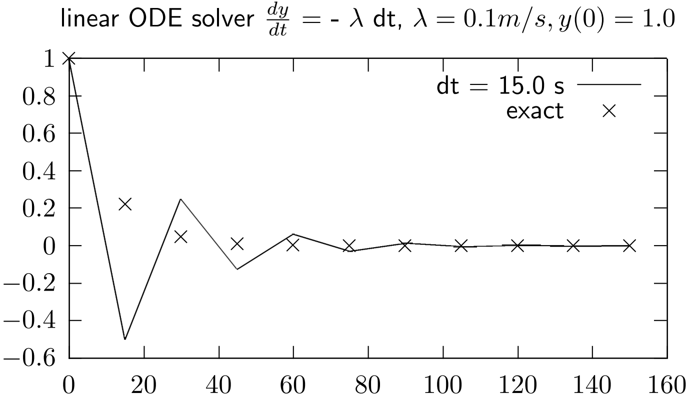
Let us analyze now the choices we have for selecting the timestep Δt fixed the problem constant λ:

- $\Delta t < \frac{1}{\lambda}$
  - In this case $0 < (1 - \lambda\Delta t) < 1$ and the numerical solution is a decreasing function of time.



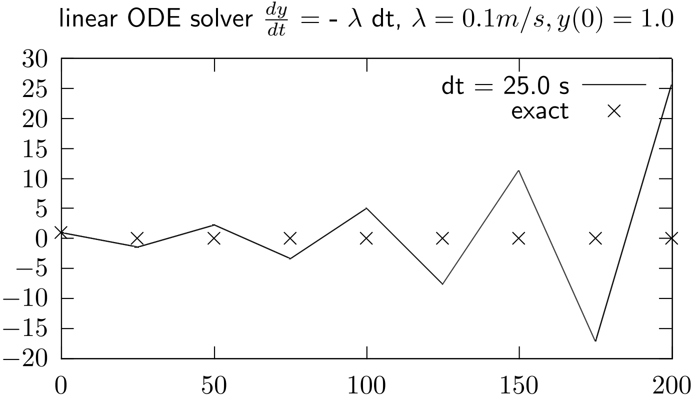
Stability analysis for a linear ODE 3

- $\frac{1}{\lambda} < \Delta t < \frac{2}{\lambda}$
  - In this case the numerical solution is decreasing function of time in magnitude, but oscillates in sign. The scheme is stable but not accurate.



Stability analysis for a linear ODE 4

- $\Delta t > \frac{2}{\lambda}$
  - In this case $(1 - \lambda\Delta t) < -1$ and the numerical solution is an increasing function of time in magnitude, and oscillates in sign. The scheme is unstable.



Exercise on linear ODE

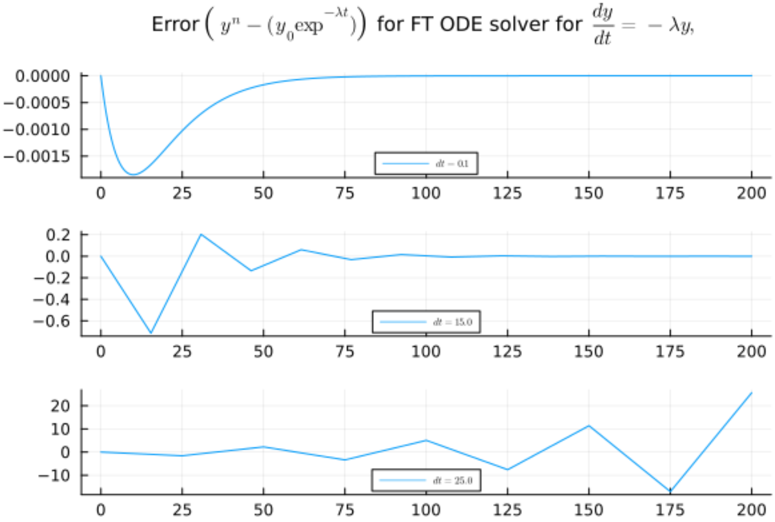
- Write a computer program to integrate the linear ODE in [10] using the FORWARD in TIME scheme in the formula in [14] with $\lambda = 0.1 s^{-1}$ and $y(0) = 1.0$ and compare the solution with the exact solution for the three cases:
  - $\Delta t < \frac{1}{\lambda}$
  - $\frac{1}{\lambda} < \Delta t < \frac{2}{\lambda}$
  - $\Delta t > \frac{2}{\lambda}$

  Plot $e^n = y^n - y(t_n)$ vs time t_n.
- Show that the BACKWARD in time difference scheme in [18] is always stable for the differential equation in [10] repeating the stability analysis we have done for the FORWARD in time difference scheme.

$$\frac{y^n - y^{n-1}}{\Delta t} = -\lambda y^n \tag{18}$$

Hint : Start substituting the generic form of the solution [15].

Expected result



Julia Code

```
t0 = 0.0; t1 = 200.0; y0 = 1.0; e0 = 0.0; lm = 0.1;
function integrate_ft(y,dt)
  (1.0 - lm*dt) * y;
end;
function exact(y0,t)
  y0 * exp(-lm*t);
end;
for dt in [ 0.1, 15.0, 25.0 ]
  nt = round(Int64,(t1-t0)/(dt)) + 1;
  sol_t = LinRange(t0,t1,nt);
  sol_e = fill(e0,nt);
  y = y0;
  for (n,t) in enumerate(sol_t[2:nt])
    y = integrate_ft(y,dt);
    sol_e[n+1] = y - exact(y0,t);
  end;
```