Computational Methods - Lesson 3 (Luxembourg, 30.09.2020)

Differential equations I

Matteo Barborini[*,a], Valentin Vassilev Galindo[*,b], Prof. Alexandre Tkatchenko[*]

Theoretical Chemical Physics group

Faculté des Sciences, des Technologies et de Médecine - Département Physique et sciences des matériaux

[*]Campus Limpertsberg, Université du Luxembourg - 162 A, avenue de la Faïencerie, L-1511 Luxembourg

[a]matteo.barborini@uni.lu, [b]valentin.vassilev@uni.lu

1. Introduction

In this lesson we will start looking into the algorithms used to solve differential equations. These algorithms are used, for example, in molecular dynamics simulations to explore the phase space of large biophysical or biochemical systems, retrieving their thermodynamical properties.

In order to introduce the basic concepts, let us consider the simple case of the harmonic oscillator in the absence of friction, defined as a mass m connected through a spring with positive elastic constant k to a fixed wall (Figure 1).

The kinetic energy associated to the mass will be equal to $T = \frac{1}{2}mv^2$, where v is the velocity of the mass, and the potential energy will be the harmonic potential $V = \frac{1}{2}kx^2$ that depends on the distance x from the resting position of the mass ($x_R = 0$). The Lagrangian associated to the system will thus be

$$\mathcal{L} = T - V = \frac{1}{2}mv^2 - \frac{1}{2}kx^2 \tag{1}$$

and the equation of motion is derived from the Euler-Lagrange equation $\frac{\partial \mathcal{L}}{\partial x} - \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial v} = 0$ from which we have:

$$-kx - m\frac{dv}{dt} = 0 \rightarrow -kx = ma \tag{2}$$

where a is the acceleration[1]. If we define the frequency $\omega = \sqrt{\frac{k}{m}}$ the final equation of motion will be

$$\ddot{x}(t) = -\omega^2 x(t), \tag{3}$$

which has a general solution of the form

$$x(t) = A\cos(\omega t + \delta) \tag{4}$$

where the amplitude of the oscillation A and the phase δ are defined by the initial conditions of the motion. As a matter of fact, if we define at the initial time $t_0 = 0$ the initial conditions for the position $x(0) = x_0$ and velocity $v(0) = v_0$ of the oscillation, we have that

$$\left\{ \begin{array}{l} x_0 = A\cos(\delta) \\ v_0 = -A\omega\sin(\delta) \end{array} \right. \rightarrow \left\{ \begin{array}{l} A = \frac{x_0}{\cos(\delta)} \\ \frac{v_0}{x_0} = -\omega\tan(\delta) \end{array} \right. \rightarrow \left\{ \begin{array}{l} A = \frac{x_0}{\cos(\delta)} \\ \delta = \arctan(-\frac{v_0}{\omega x_0}) \end{array} \right. .$$

Finally, we recall that the energy of the system is given by the Hamiltonian, defined from the Lagrangian as

$$H = \frac{\partial \mathcal{L}}{\partial v}v - \mathcal{L} = \frac{1}{2}mv(t)^2 + \frac{1}{2}kx(t)^2, \tag{5}$$

equal to the sum of the Kinetic and potential energies. Even though $x(t)$ and $v(t)$ depend on the time, the energy is a conserved quantity in a physical system. We will see in the following sections that, due to the algorithmic approach, this principle is violated by the integration algorithms.

[1] Please not that, by definition, $v(t) = \frac{dx(t)}{dt} = \dot{x}(t)$ and $a(t) = \frac{dv(t)}{dt} = \frac{d^2x(t)}{dt^2} = \ddot{x}(t)$

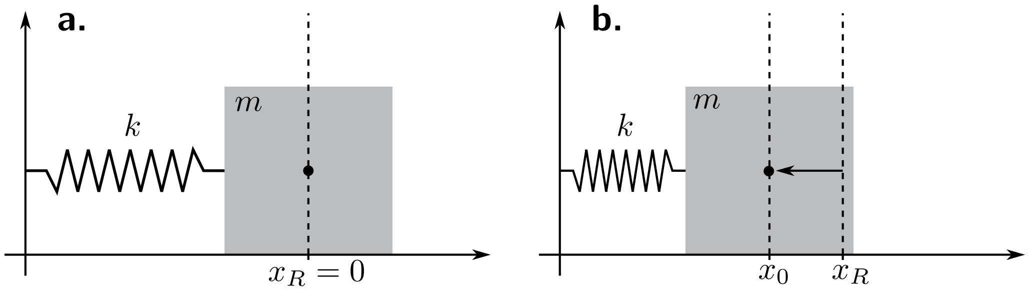

Figure 1: Schematic representation of the harmonic oscillator (without friction) at the resting position $x_R = 0$ (**a**) and at the initial condition (**b**).

2. Numerical Integration

The equation of motion of the harmonic oscillator, $\ddot{x}(t) = -\omega^2 x(t)$, can be rewritten as a system of coupled differential equations of the first order by setting $\phi(t) = \ddot{x}(t)$:

$$\begin{cases} \dot{v}(t) = \phi(t) \\ \dot{x}(t) = v(t) \end{cases} . \tag{6}$$

In order to find the time evolution of these quantities we need to introduce a time discretization. The time thus evolves through small but finite increments Δt and we will be interested in the solution in the limit of continuum $\Delta t \to 0$. Starting at t_0 we describe the progressive evolution at times $t_1 = t_0 + \Delta t$, $t_2 = t_1 + \Delta t = t_0 + 2\Delta t$ up to $t_n = t_{n-1} + \Delta t = t_0 + n\Delta t$.

The quality of the algorithm will depend on two factors. First, the trajectory for different values of Δt should not vary too much. Second, time discretization violates the conservation rules (for example of the energy), thus it is important that the energy fluctuations around the correct value do not exceed a desired threshold during the evolution (we will see this in the following sections).

2.1. The Euler and Euler-Cromer algorithms.

The Euler algorithm is defined by approximating the derivatives on the left sides of the system of eqs. 6 with the forward approach with an error of order $O(dt)$. In order to derive the Euler algorithm, let us compute the values of $x(t + dt)$ and $v(t + dt)$ by expanding them in Taylor series around $x(t)$ and $v(t)$ respectively:

$$\begin{cases} v(t + dt) = v(t) + \phi(t)dt + O(dt^2) \\ x(t + dt) = x(t) + v(t)dt + O(dt^2) \end{cases} . \tag{7}$$

By approximating the expansions in the first order dt, we have that for a finite time step Δt

$$\begin{cases} v(t + \Delta t) \approx v(t) + \phi(t)\Delta t \\ x(t + \Delta t) \approx x(t) + v(t)\Delta t \end{cases} . \tag{8}$$

This result corresponds to substituting the derivatives in eqs. 6 with those approximated by the forward method. By generalizing the system of equations for consequent time steps, we finally obtain the Euler algorithm

$$\begin{cases} v(t_{n+1}) = v(t_n) + \phi(t_n)\Delta t \\ x(t_{n+1}) = x(t_n) + v(t_n)\Delta t \end{cases} . \tag{9}$$

This algorithm is considered quite bad for different reasons. Consider for example the discretization error Δt. The local error of the algorithm is of order Δt^2 since we only consider the terms up to the first order in Taylor expansion. If we want to evolve the system up to the time T with a time step Δt the number of steps to propagate the position will be $n = T/\Delta t$. The global error accumulated during the propagation will be equal to $n\Delta t^2 = T\Delta t$ meaning that, since T is a constant, the error decreases linearly with Δt which is a quite ugly result (See for example the linear error in the energy shown in Figure 3). Moreover, it can be shown that the algorithm is actually unstable (Figure 2), and it does not conserve the total energy of the system (Figure 3).

In order to understand why the algorithm is unstable, let us consider a first order differential equation, of the type

$$\dot{x}(t) + f(x(t), t) = 0 \to \dot{x}(t) = -f(x(t), t). \tag{10}$$

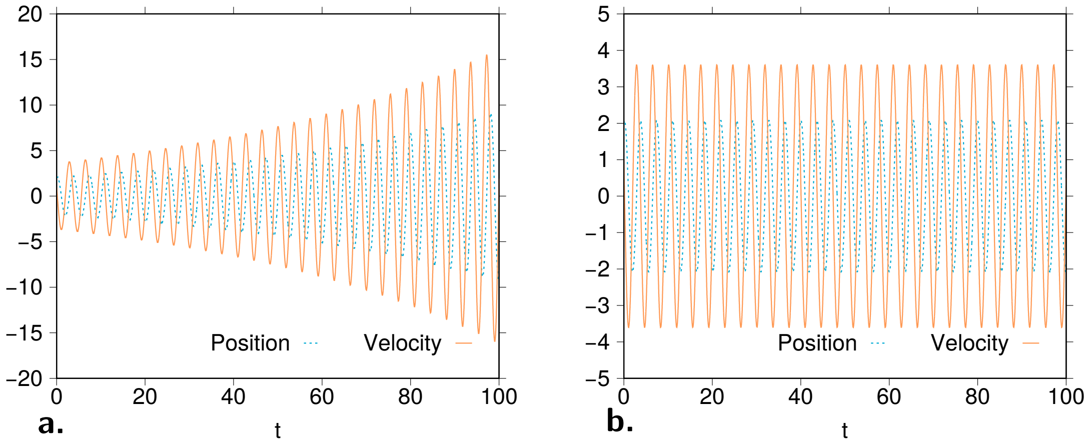

Figure 2: Evolution of the position and velocity of the algorithmic solution of the harmonic oscillator with the Euler (**a**) and Euler-Cromer (**b**) algorithms.

Applying the same discretization of the Euler algorithm we have that

$$x(t_{n+1}) = x(t_n) - f(x(t_n), t_n)\Delta t. \tag{11}$$

If we assume that at the time t_n the algorithmic solution, differs from the correct one for a variation δx_n, we can write the equation

$$x(t_{n+1}) + \delta x_{n+1} = x(t_n) + \delta x_n - \left[f(x(t_n), t_n) + \left.\frac{\partial f(x(t), t)}{\partial x(t)}\right|_{t=t_n} \delta x_n \right] \Delta t \tag{12}$$

where in the square brackets we have written the update of the function $f(x(t), t)$ due to the variation δx_n. By subtracting the two equations we can write the evolution of the error as

$$\delta x_{n+1} = \delta x_n - \left.\frac{\partial f(x(t), t)}{\partial x(t)}\right|_{t=t_n} \delta x_n \Delta t = \left[1 - \left.\frac{\partial f(x(t), t)}{\partial x(t)}\right|_{t=t_n} \Delta t \right] \delta x_n = C(t_n, \Delta t)\delta x_n. \tag{13}$$

It is now clear that the error at time t_{n+1} is the product of the error at time t_n times $C(t_n, \Delta t)$ that is a function t_n and Δt, and as $\Delta t \to 0$ becomes equal to 1 so that the error becomes a constant.

If $|C(t_n, \Delta t)| > 1$ the difference diverges, while for $|C(t_n, \Delta t)| < 1$, that is for $-1 < C(t_n, \Delta t) < 1$, it converges. The right side condition, states that, since $\Delta t > 0$, in order for the Euler method to be stable we must have that

$$1 - \left.\frac{\partial f(x(t), t)}{\partial x(t)}\right|_{t=t_n} \Delta t < 1 \to - \left.\frac{\partial f(x(t), t)}{\partial x(t)}\right|_{t=t_n} \Delta t < 0 \to \left.\frac{\partial f(x(t), t)}{\partial x(t)}\right|_{t=t_n} > 0. \tag{14}$$

Instead the inequality on the right implies that

$$-1 < 1 - \left.\frac{\partial f(x(t), t)}{\partial x(t)}\right|_{t=t_n} \Delta t \to 2 > \left.\frac{\partial f(x(t), t)}{\partial x(t)}\right|_{t=t_n} \Delta t \to 2 \left[\left.\frac{\partial f(x(t), t)}{\partial x(t)}\right|_{t=t_n} \right]^{-1} > \Delta t \tag{15}$$

that is, the time discretization must be small enough in order for the solution to be stable.

Now, for oscillating systems like the harmonic oscillator, the solution will be of the type $x(t) \propto e^{-i\omega t}$ in the complex representation. Since $\dot{x}(t) = -i\omega x(t)$ the derivative $\frac{\partial f(x(t),t)}{\partial x(t)}$ will be a complex number and it is convenient to study the stability condition as $|C|^2 \leq 1$ for which we have $|1 - i\omega\Delta t|^2 \leq 1$. This condition is never verified, in fact $|1 - i\omega\Delta t|^2 = 1 + \omega^2\Delta t^2$ which is strictly greater then one, if $\Delta t \neq 0$, so oscillating systems are always unstable.

It can be shown that a simple modification of this algorithm done by Cromer in 1981, defining the Euler-Cromer algorithm

$$\left\{ \begin{array}{l} v(t_{n+1}) = v(t_n) + \phi(t_n)\Delta t \\ x(t_{n+1}) = x(t_n) + v(t_n + 1)\Delta t \end{array} \right. , \tag{16}$$

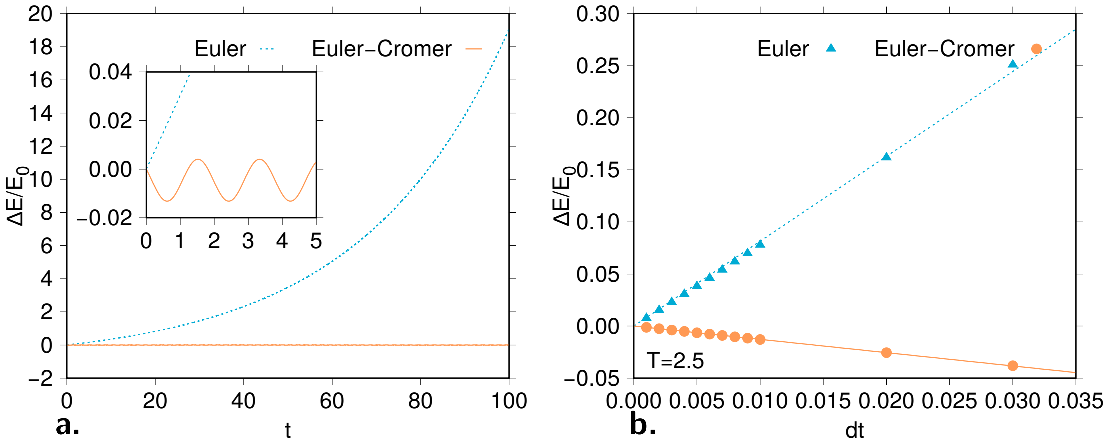

Figure 3: a. Relative error of the energy $\Delta E = \frac{E(t)-E_0}{E_0}$ of the system as a function of time (E_0 is the exact energy defined by the initial conditions $E_0 = \frac{1}{2}mv_0^2 + \frac{1}{2}kx_0^2$). We can see that Euler's error diverges as a function of time. The rate of this divergence depends clearly on the time step Δt; the smaller Δt the slower the divergence. On the other hand, the error of the Euler-Cromer algorithm oscillates between two values, and the amplitude of the oscillation depends on Δt. **b.** Relative error of the energy at time $T = 2.5$, obtained for different values of Δt. We can see that the convergence to zero is linear in Δt.

gives a stable solution that conserves the energy within a certain error (see comparisons in Figure 3 with Euler) also for oscillating systems.[1]

2.2. Generalized stability analysis.

A generalized stability analysis for an algorithm is obtained by studying the eigenvalues of the transition matrix. For example, the Euler integration method for the harmonic oscillator

$$\begin{cases} x(t_{n+1}) = x(t_n) + v(t_n)\Delta t \\ v(t_{n+1}) = v(t_n) - \omega^2 x(t_n)\Delta t \end{cases} \tag{17}$$

is written in matrix form as

$$\begin{pmatrix} x(t_{n+1}) \\ v(t_{n+1}) \end{pmatrix} = \begin{pmatrix} 1 & \Delta t \\ -\omega^2\Delta t & 1 \end{pmatrix} \begin{pmatrix} x(t_n) \\ v(t_n) \end{pmatrix} = \mathbf{T}_E \begin{pmatrix} x(t_n) \\ v(t_n) \end{pmatrix} \tag{18}$$

where $\mathbf{T}_E$ is the transition matrix. The stability of the method is related to the eigenvalues of the transition matrix and also to the determinant. In this case the determinant of the transition matrix is equal to $\det(\mathbf{T}_E) = 1+\omega^2\Delta t$ which for finite Δt is always larger then 1, *ie* the error in the solution always diverges in time. More precisely, if we compute the eigenvalues of the transition matrix through the equation

$$\det\left(\mathbf{T}_E - \lambda\mathbf{I}\right) = 0 \rightarrow \det\begin{pmatrix} 1-\lambda & \Delta t \\ -\omega^2\Delta t & 1-\lambda \end{pmatrix} = 0 \rightarrow (1-\lambda)^2 + \omega^2\Delta t^2 = 0 \tag{19}$$

we obtain

$$\lambda_+ = 1 + i\omega\Delta t \qquad \lambda_- = 1 - i\omega\Delta t \tag{20}$$

which represent two directions, both with modulus that is larger then one, *ie* they lay outside the circle of radius 1 in the complex plane, thus they are unstable.

For Euler-Cromer on the other hand we have that

$$\begin{cases} x(t_{n+1}) = x(t_n) + v(t_n)\Delta t - \omega^2 x(t_n)\Delta t^2 \\ v(t_{n+1}) = v(t_n) - \omega^2 x(t_n)\Delta t \end{cases} \tag{21}$$

that in matrix form can be rewritten as

$$\begin{pmatrix} x(t_{n+1}) \\ v(t_{n+1}) \end{pmatrix} = \begin{pmatrix} 1-\omega^2\Delta t^2 & \Delta t \\ -\omega^2\Delta t & 1 \end{pmatrix} \begin{pmatrix} x(t_n) \\ v(t_n) \end{pmatrix} = \mathbf{T}_{EC} \begin{pmatrix} x(t_n) \\ v(t_n) \end{pmatrix} \tag{22}$$

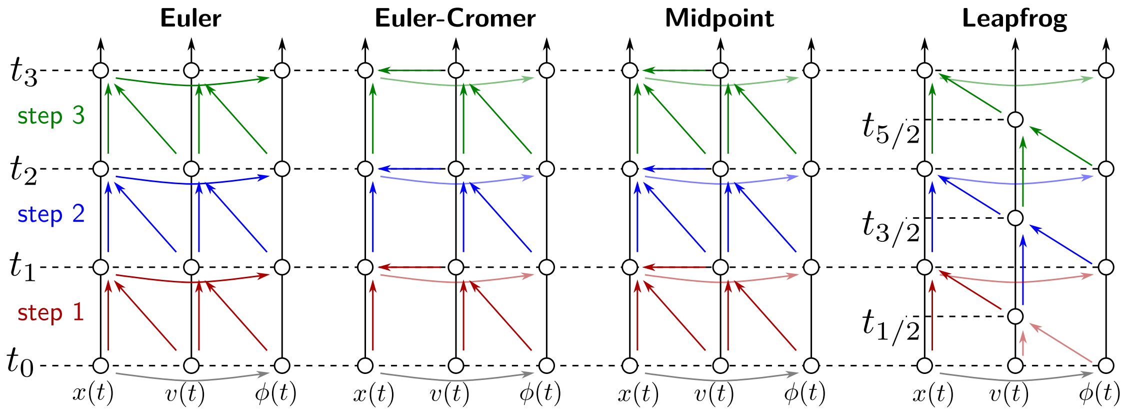

Figure 4: Schemes representing the evolving steps of the four algorithms discussed in this first lesson. We can see how, in order to update the positions, Euler, Euler-Cromer and Midpoint each used different values of the velocity at times t_n and t_{n+1}. These three methods are all self-starting, since through the initial conditions $x(t_0) = x_0$ and $v(t_0) = v_0$ it is possible to compute the function $\phi(t_0) = \phi_0$ and start the update of velocity and position. On the contrary, the leapfrog algorithm, that uses an intermediate step evolution for the velocities, requires the initial computation of the velocity $v(t_{1/2})$. This can be done through $v(t_0)$ and $\phi(t_0)$ using the Euler method.

and the determinant of the transition matrix will be $\det(\mathbf{T}_{EC}) = 1 - \cancel{\omega^2\Delta^2} + \cancel{\omega^2\Delta^2} = 1$, so that the solutions lay exactly on the border. If we compute the eigenvalues we have that

$$\det\left(\mathbf{T}_{EC} - \lambda\mathbf{I}\right) = 0 \rightarrow \det\begin{pmatrix} 1 - \omega^2\Delta t^2 - \lambda & \Delta t \\ -\omega^2\Delta t & 1 - \lambda \end{pmatrix} = 0 \rightarrow 1 + \lambda(\omega^2\Delta t^2 - 2) + \lambda^2 = 0$$

so that the discriminant will be $\Delta = \omega^4\Delta t^4 - 4\omega^2\Delta t^2 < 0$ and the two solutions will be

$$\lambda_+ = \frac{2 - \omega^2\Delta t^2 + i\omega\Delta t\sqrt{4 - \omega^2\Delta t^2}}{2} \qquad \lambda_- = \frac{2 - \omega^2\Delta t^2 - i\omega\Delta t\sqrt{4 - \omega^2\Delta t^2}}{2}$$

which lay on the border of the circle (consider for example the case in which $\omega^2\Delta t^2 = 1$).

2.3. Midpoint and leapfrog.

Euler and Euler-Cromer methods use respectively the velocities at times $v(t_n)$ and $v(t_{n+1})$ to obtain the update of the position $x(t_{n+1})$. A scheme that compares to two procedures can be found in Figure 4.

Two more accurate approaches are represented by the **midpoint** and **leapfrog** methods. The **midpoint** algorithm, in order to update the position uses the mixed value $\frac{v(t_n)+v(t_{n+1})}{2}$ of the velocity, that gives the algorithm

$$\begin{cases} v(t_{n+1}) = v(t_n) + \phi(t_n)\Delta t \\ x(t_{n+1}) = x(t_n) + \frac{v(t_n)+v(t_{n+1})}{2}\Delta t \end{cases} . \tag{23}$$

The reason why we use this intermediate scheme can be understood by substituting the first equation in the second one, for which we obtain

$$x(t_{n+1}) = x(t_n) + \frac{v(t_n) + v(t_n) + \phi(t_n)\Delta t}{2}\Delta t = x(t_n) + v(t_n)\Delta t + \frac{\phi(t_n)}{2}\Delta t^2 \tag{24}$$

which corresponds to the second order Taylor series expansion of the displacement function $x(t)$ around $x(t_n)$. This means that the error of the step is of order $O(\Delta t^2)$ in $v(t)$ and of order $O(\Delta t^3)$ in $x(t)$, which is an improvement with respect to Euler and Euler-Cromer. Moreover, like these first two algorithms, also the midpoint one is self-starting, that is, the initial conditions (x_0, v_0) are sufficient to start the time evolution.

An alternative to the midpoint method is the so called **leapfrog** method for which the velocities are defined at times that are in between the times with which the displacement advances:

$$\begin{cases} v(t_{n+1/2}) = v(t_{n-1/2}) + \phi(t_n)\Delta t \\ x(t_{n+1}) = x(t_n) + v(t_{n+1/2})\Delta t \end{cases} . \tag{25}$$

The scheme is represented in the fourth panel of Figure 4. As shown in the scheme (first two transparent arrows), this method is not self-starting, since in order to compute the position $x(t_1)$ it requires also the value of the velocity $v(t_{1/2})$ at the intermediate time $t_{1/2}$. A way to obtain this value is to do an initial step with the Euler algorithm, evolving the velocity from $v(t_0)$ of half the time step $\Delta t/2$:

$$v(t_{1/2}) = v(t_0) + \frac{1}{2}\phi(t_0)\Delta t. \tag{26}$$

References

(1) Cromer, A. Stable solutions using the Euler approximation. *Am. J. Phys.* **1981**, *49*, 455–459.