Exercises to numerical Integration and ODEs II
Practical in Numerical Astronomy SS 2010

Siegfried Eggl

1 Exercises in short

- Compare the behavior of Euler's explicit first order method to the symplectic Euler-Cromer method (also first order) in the *frictionless* Harmonic Oscillator's phase space for step-sizes (h) from $h = 1$ to $h = 0.01$, $k = 1$.
- Equip Euler's first order and Euler-Cromer's schemes with adaptive step-size control. Compare their performances to their fixed step-size counterparts' with respect to the *frictionless* Harmonic Oscillator problem.
- Solve the ordinary differential equation of the Nonlinear Pendulum (one dimensional) numerically. Use all of your Euler's methods. How can you check the quality of the solutions produced by your algorithms?

1.1 Adaptive Step-size Control

Behold - once again - the Harmonic Oscillator. This time, we are not interested in friction, so the equations simplify to

$$\frac{d^2x(t)}{dt^2} + kx(t) = 0$$

and consequently

$$\begin{aligned} \dot{x}(t) &= v(t) \\ \dot{v}(t) &= -kx(t) \end{aligned} \tag{1}$$

$x(t)$ being the actual position at time t, $v(t)$ the velocity and k denotes the spring constant. Among the things you should remember reside the following:

- For numerical integrations in general smaller step-sizes (e.g. h) in the parameter (e.g. t) mean more accurate results. This is a direct consequence of the requirement of *convergence* of our method, meaning that if our discretization parameters (e.g. h) tend towards zero, the solution of our difference equation converges against the true solution of our differential equation.
- More steps also mean more evaluations and calculation processes increasing round-off errors and time requirements.

Choosing the proper step-size can be difficult, when you know absolutely nothing about the equations you are dealing with. So, wouldn't it be nice, to make the algorithm choose its own step-size, and constantly adapt it to the required accuracy? This is easier than you might think. Let us assume the requirement of convergence is fulfilled. A direct application of this theorem would be the following:

1. Choose an arbitrary step-size h.
2. Do one step. Memorize the result: $R(h)$.
3. Take a quarter of the same step-size: $h/4$.
4. Do four steps to get to the same point you reached in you first calculation: $R(h/4)$.
5. The convergence condition holds, therefore your new result should be four times better than the old one.[1]
6. Compare your results: if the difference is smaller than your required accuracy (ϵ): $|R(h) - R(h/4)| < \epsilon$, your step-size is fine. Your equation is currently well behaved, and not so much would change, if you took smaller steps. You could actually think of making your step-size larger.
7. If $|R(h) - R(h/4)| > \epsilon$, the difference in results gained via larger and smaller step-sizes is too big, your solution with the larger h will have large errors, therefore reduce the step-size.

Of course, comparing different runs with different step-sizes is cumbersome. Cash & Karp (1990) found a very neat method to boost the performances of Runge Kutta algorithms with adaptive stepping. They compare different orders of the Runge-Kutta method using the same step-size, plus they found a way to use the results from the lower order to calculate the higher order. Therefore comparison of e.g. RK4 and RK5 can be done very efficiently. For our purposes, the points listed above should do the trick.

1.2 Symplecti...what?

Now something completely different. Have another look at the equations of motion for the Harmonic Oscillator (1). In order to jump from 17th to 19th century nomenclature in the description of our problems, we better introduce the Hamiltonian formalism. The Hamiltonian $H(q, p, t)$ is a function of generalized coordinates q (in our case $x = q$) and momenta p (in our case $v = p$).[2] Since the Hamiltonian can be expressed as the sum of kinetic (T) and potential energy (U) we can write it down easily enough for the Harmonic Oscillator

$$H(q, p) = T(p) + U(q) = \frac{p^2}{2} + k\frac{q^2}{2}$$

[1] Since we usually operate way beyond the influence of round-off errors, it is save to claim this.

[2] That is not true for all problems. Usually the momenta are defined as $p = \frac{\partial L(q,\dot{q},t)}{\partial \dot{q}}$, L being the Lagrangian function.

As we can see, the Hamiltonian in our case does not depend on time (t) explicitly. That is quite nice, because that means, that the total energy will stay constant over time, since it can be shown that (Goldstein, 2001)

$$\begin{aligned} \frac{dH}{dt} &= \frac{\partial H}{\partial t} \\ \frac{\partial H}{\partial t} = 0 \rightarrow \frac{dH}{dt} &= 0 \rightarrow H = const \end{aligned} \tag{2}$$

So - in theory, once initial conditions, e.g. $q = 1, p = 0$ are fixed, we can calculate the value of Hamiltonian and it should stay constant throughout our integration. Of course, having $H(q, p)$ is fine, but what to integrate? Don't panic, of course there are beautiful formulations for the equations of motions in terms of first order differential equations:

$$\begin{aligned} \dot{q} &= \frac{\partial H(q,p)}{\partial p} \\ \dot{p} &= -\frac{\partial H(q,p)}{\partial q} \end{aligned} \tag{3}$$

which become in our case

$$\begin{aligned} \dot{q}(t) &= p(t) \\ \dot{p}(t) &= -kq(t) \end{aligned} \tag{4}$$

That should look familiar... Once again we may find Euler's integration formulae from equations (4), and carry on with it. They are, after all, first order ODEs, and we know how to solve them - right? As so often happens in Physics, it pays off to take a step back, refrain from bulldozing everything with the same stuff we always do, and *think*. Having a decent look at equations (3) we find, that there seems to be a structure hidden within. Let us reformulate these equations in order to see more clearly. Introducing the phase-space location vector (Phasenraum-Ortsvektor) $\vec{z} = \binom{\vec{q}}{\vec{p}}$ equations (3) become

$$\dot{\vec{z}} = J \cdot \vec{\nabla} H \tag{5}$$

where $\vec{\nabla}$ is nothing more than the commonly known gradient $\vec{\nabla} = \binom{\frac{\partial}{\partial q}}{\frac{\partial}{\partial p}}$. J is the so-called "SYMPLECTIC STRUCTURE MATRIX" (in our case it's just a 2x2 matrix), that impresses a special structure on the coordinate space. How J looks like, is one thing YOU are going to find out - if you have not payed attention during the lecture. *Small hint:* Have a look at equations (3) and (5), since they are equivalent...

Yeah, yeah, very interesting... *so what*? Well, it can be shown (Hairer, Lubich, Wanner, 2006) that an integration algorithm which preserves J has very good properties in terms of preservation of integrals of motion (e.g. energy or angular momentum). So, if an integration method is called *symplectic* it basically conserves the special geometry of Hamiltonian systems. As very, very many systems in physics can be described via Hamilton's formalism, that's quite good news. Not convinced? Don't worry, you are going to see for yourself! Now is the time for getting acquainted with symplectic algorithms. And, surprisingly,

it turns out, that a simple combination of explicit and implicit Euler's algorithms provides one:

$$\begin{aligned} p_{n+1} &= p_n - h\frac{\partial H}{\partial q}|_{p_{n+1},q_n} \\ q_{n+1} &= q_n + h\frac{\partial H}{\partial p}|_{p_{n+1},q_n} \end{aligned} \tag{6}$$

This one is called Euler-Cromer algorithm. It's still of first order. The vertical line just tells you what to insert into your Hamiltonian, once you derived it. Deriving, of course is done analytically. For the Harmonic Oscillator, this will result in

$$\begin{aligned} p_{n+1} &= p_n - h \cdot k \cdot q_n \\ q_{n+1} &= q_n + h \cdot p_{n+1} \end{aligned} \tag{7}$$

Hereby the important thing is, that the *updated* velocities are used, instead of the old ones. The proof that this method is indeed symplectic may be found in Hairer, Lubich, Wanner (2006), or in your notes to the corresponding lecture. Symplecticity also means a conservation of the phase space volume. Can you actually see this behavior for the Euler-Cromer algorithm?

2 Exercises in detail

2.1 Symplectic???

In order to see what symplectic integration algorithms can do for you, compare the behavior of Euler's explicit first order method to the symplectic Euler-Cromer method (7) with the *frictionless* Harmonic Oscillator's analytic solutions for step-sizes (h) from $h = 1$ to $h = 0.01$ and $k = 1$. Also check the conservation of total energy, and the method's behaviors in phase space.

2.2 Adaptive Step-size

Implement adaptive step-size control in Euler's first order and Euler-Cromer's schemes! Compare their performances to their fixed step-size counterparts' with respect to the *frictionless* Harmonic Oscillator problem. Don't forget to have a look on the total energy... Also calculate the mean step-size your algorithm uses. Hint: just sum all the performed step-sizes, and divide them by the number of total loops done within you program.

2.3 The Nonlinear Pendulum

It's time to deal with something more nasty - the Nonlinear Pendulum!

$$\ddot{q} = -k \cdot Sin(q)$$

The Hamiltonian of this problem can be written as

$$H(q,p) = \frac{p^2}{2} + k \cdot Cos(q)$$

Derive Euler's and Euler-Cromer's equations for this problem. Choose initial conditions for $q \in [-2, 2]$ and $p \in [-4, 4]$ and solve the equations for a total time of 50 time-units. Employ different step-sizes, and see how things turn out. How can you check whether your algorithm performs well? *Hint:* Have a look at equation (2).
Include adaptive step-size control in your algorithms. Compare the results in phase-space (plot q against p)!

Bonus1: find the analytic solution of the pendulum equation in the region $q \in [0, \pi]$ *and* $p \in [-4, 4]$
Bonus2: Use the RK4 algorithm for solving the Nonlinear Pendulum problem

3 Requirements

One protocol containing

- an introduction to the problems posed.
- answers to *each and every* question.
- *no source code!* Latter shall be sent to *siegfried.eggl@univie.ac.at* in form of a *compilable* textfile.

Please send the protocol as a PDF file to *siegfried.eggl@univie.ac.at*

References

Cash, J. R., Karp, A. H.: *A Variable Order Runge-Kutta Method for Initial Value Problems with Rapidly Varying Right-Hand Sides* ACM Transactions on Mathematical Software, Vol. 16, No. 3, *p. 201-222* (1990)

Goldstein, H., Safko, J., Poole, Ch.: *Classical Mechanics*, Addison Wesles, 3rd edition (2001)

Hairer, E., Lubich, C., Wanner, G.: *Geometric Numerical Integration, Structure-Preserving Algorithms for Ordinary Differential Equations*, Springer, 2nd edition (2006)