Introduction to Computational Physics

Lecture: Time Integration - 1

Andreas Adelmann

Paul Scherrer Institut, Villigen
E-mail: andreas.adelmann@psi.ch

https://moodle-app2.let.ethz.ch/course/view.php?id=15323

Content I

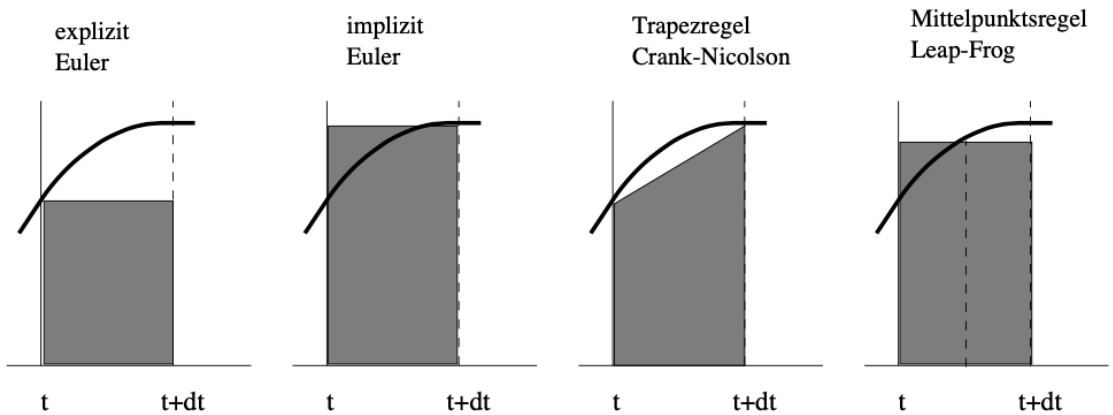
9.1 Introduction I

For unsteady or initial value problems the time will be discretised as a new coordinate direction

$$\frac{d\phi}{dt} = f(t, \phi(t)), \qquad \phi(t_0) = \phi^0.$$

The time dependency is solved by integration in the time direction. That means integration of the partial differential equation from time t^n to $t^{n+1} = t^n + \Delta t$. The exact integral of $\frac{d\phi}{dt}$ is

$$\int_{t^n}^{t^{n+1}} \frac{d\phi}{dt} = \phi^{n+1} - \phi^n.$$



9.2 Numerical Errors I

We consider wo major sources of error

- truncation error
- round-off error.

Remark

1. *Truncation error – as a result from the Taylor expansion – arises in Euler's method because the curve of the solution is not generally a straight-line between the neighbouring grid-points. argument.*
2. *Every time we take a step using Euler's method we collect a truncation error of $\mathcal{O}(\Delta t^2)$, where Δt is the step-length.*

Example:

- Euler's method to integrate our ODE over an interval of order unity.
- This requires $\mathcal{O}(\Delta t^{-1})$ steps.

9.2 Numerical Errors II

- each step incurs an error of $\mathcal{O}(\Delta t^2)$, and the errors are simply cumulative (a fairly conservative assumption), then the net truncation error is $\mathcal{O}(\Delta t)$.

$\rightarrow$ Thus, if we want to keep the relative error in the integration below about 10^{-6} then we would need to take about 10^6 steps per unit interval.

$\rightarrow$ Euler's method is termed a first-order integration method

Remark

Note that truncation error would be present even if computers performed floating-point arithmetic operations to infinite accuracy.

$\rightarrow$ infinite accuracy does not exist on computers

$\rightarrow$ we have floating-point number to a fixed number of decimal places

9.2 Numerical Errors III

→ there is a characteristic number, η, which is defined as the smallest number which when added to a number of order unity gives rise to a new number

→ Every floating-point operation incurs a round-off error of $\mathcal{O}(\eta)$ which arises from the finite accuracy of floating-point numbers

Suppose that we use Euler's method to integrate our ODE over an interval of order unity. This entails $\mathcal{O}(\Delta t^{-1})$ integration steps, and, therefore, $\mathcal{O}(\Delta t^{-1})$ floating-point operations. If each floating-point operation incurs an error of $\mathcal{O}(\eta)$, and the errors are simply cumulative, then the net round-off error is

$$\mathcal{O}\left(\frac{\eta}{\Delta t}\right).$$

9.2 Numerical Errors IV

The total error, ε, associated with integrating our ODE over an interval of order unity is (approximately) the sum of the truncation and round-off errors. Thus, for Euler's method we get

$$\varepsilon \sim \frac{\eta}{\Delta t} + \Delta t.$$

Two cases need to be considererd:

- at large step-lengths Δt the error is dominated by truncation error
- round-off error dominates at small step-lengths

9.2 Numerical Errors V

Remark

1. *There is clearly no point in making the step-length, Δt, any smaller than Δt_0, since this increases the number of floating point operations but does not lead to an increase in the overall accuracy.*
2. *It is also clear that the ultimate accuracy of Euler's method (or any other integration method) is determined by the accuracy, η, to which floating-point numbers are stored on the computer performing the calculation.*

The value of η depends on how many bytes the computer hardware uses to store floating-point numbers. In Julia you can check this with:

9.2 Numerical Errors VI

```
julia> eps(Float32)
1.1920929f-7

julia> eps(Float64)
2.220446049250313e-16
```

It follows that the minimum practical step length for Euler's method using `Float64` is $\Delta t_0 \sim 10^{-8}$, yielding a minimum relative integration error of $\varepsilon_0 \sim 10^{-8}$. This level of accuracy is perfectly adequate for most scientific calculations.
The corresponding η value for single precision, yielding a minimum practical step-length and a minimum relative error for Euler's method of $\Delta t_0 \sim 3 \times 10^{-4}$ and $\varepsilon_0 \sim 3 \times 10^{-4}$, respectively.

9.2 Numerical Errors VII

Remark

This level of accuracy is generally not adequate for scientific calculations. However there is a great interest in mixed precision arithmetic mainly to save memory and make use of the faster low precision arithmetic avaidable in modern CPU's und GPU's.

9.3 A Family of One Step Methods I

The one step methods can be generalized as follows:

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = \alpha f(t + \Delta t, \phi^{n+1}) + (1 - \alpha) f(t, \phi^n) \text{ with } \quad 0 \leq \alpha \leq 1.$$

We get

α	method	stability	order
0	explicit Euler	conditionally	1
1	implicite Euler	stable	1
1/2	class. Crank-Nicholson	stable	2

9.4 A Family of Multi Step Methods I

By using multi step methods a higher approximation order can be obtained. This is achieved by using the information from more time steps. For example the two step method

$$\begin{aligned}\frac{(1+\beta)\phi^{n+1} - (1+2\beta)\phi^n + \beta\phi^{n-1}}{\Delta t} &= \gamma f(t+\Delta t, \phi^{n+1}) \\ &+ (1-\gamma+\delta) f(t, \phi^n) \\ &- \delta(f(t-\Delta t, \phi^{n-1}).\end{aligned}$$

α	β	γ	method	order
0	0	0	explicit Euler	1
1	0	0	implicite Euler	1
1/2	0	0	class. Crank-Nicholson	2
1	1/2	0	Backward difference	2
3/4	0	-1/4	Adams-type	2
0	-1/2	0	Leapfrog	2
5/12	0	1/12	Adams-Moulton	3

9.5 Runge-Kutta methods I

There are two main reasons why Euler's method is not generally used in scientific computing.

1. given value of Δt: the truncation error per step associated with this method is far larger than those associated with other, more advanced, methods
2. Euler's method is too prone to numerical instabilities

The methods most commonly employed to integrate ODE's were first developed by the German mathematicians *C.D.T. Runge* and *M.W. Kutta* in the latter half of the nineteenth century. The basic reasoning behind so-called Runge-Kutta methods is outlined in the following.

- The main reason that Euler's method has such a large truncation error per step is that in evolving the solution from ϕ^n to ϕ^{n+1} the method only evaluates derivatives at the beginning of the interval: i.e., at ϕ^n.
- The method is, therefore, very asymmetric with respect to the beginning and the end of the interval.

9.5 Runge-Kutta methods II

- We can construct a more symmetric integration method by making an Euler-like trial step to the midpoint of the interval, and then using the values at the midpoint to make the real step across the interval.

To be more exact,

$$\begin{aligned} k_1 &= \Delta t \cdot f(t_n, \phi^n) \\ k_2 &= \Delta t \cdot f(t_{n+1/2}, \phi^n + \frac{k_1}{2}) \\ \phi^{n+1} &= \phi^n + k_2 + \mathcal{O}(\Delta t^3). \end{aligned}$$

As indicated in the error term, this symmetrization cancels out the first-order error, making the method second-order.

9.5 Runge-Kutta methods III

Remark

- *the above method is known as a second-order Runge-Kutta method*
- *Euler's method is a first-order Runge-Kutta method*

Of course, there is no need to stop at a second-order method. By using two trial steps per interval, it is possible to cancel out both the first and second-order error terms, and, thereby, construct a third-order Runge-Kutta method.
The general form is

$$\phi^{n+1} = \phi^n + \Delta t \sum_{r=1}^{R} c_r f^r.$$

with

$$f^r = f(t + \Delta t \cdot a_r, \phi^n + \Delta t \sum_{s=1}^{R} b_{r,s} f^s), \quad r = 1, 2, \ldots R$$

9.5 Runge-Kutta methods IV

and

$$a_r = \sum_{s=1} b_{r,s}, \quad r = 1, 2, \ldots R.$$

The widely used RK-method of fourth order now can be written as

$$\begin{aligned}
k_1 &= \frac{\Delta t}{2} f(t_n, \phi^n) \\
k_2 &= \frac{\Delta t}{2} f(t_{n+1/2}, k_1) \\
k_3 &= \frac{\Delta t}{2} f(t_{n+1/2}, k_2) \\
k_4 &= \frac{\Delta t}{2} f(t_n, k_3) \\
\phi^{n+1} &= \phi^n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} + \mathcal{O}(\Delta t^5).
\end{aligned}$$

9.6 Examples I

Consider the following system of ODEs:

$$\begin{aligned} \frac{dx}{dt} &= v, \\ \frac{v}{dt} &= -kx, \\ x(0) &= 0, \quad v(0) = \sqrt{k}, \qquad \text{at } t = 0. \end{aligned}$$

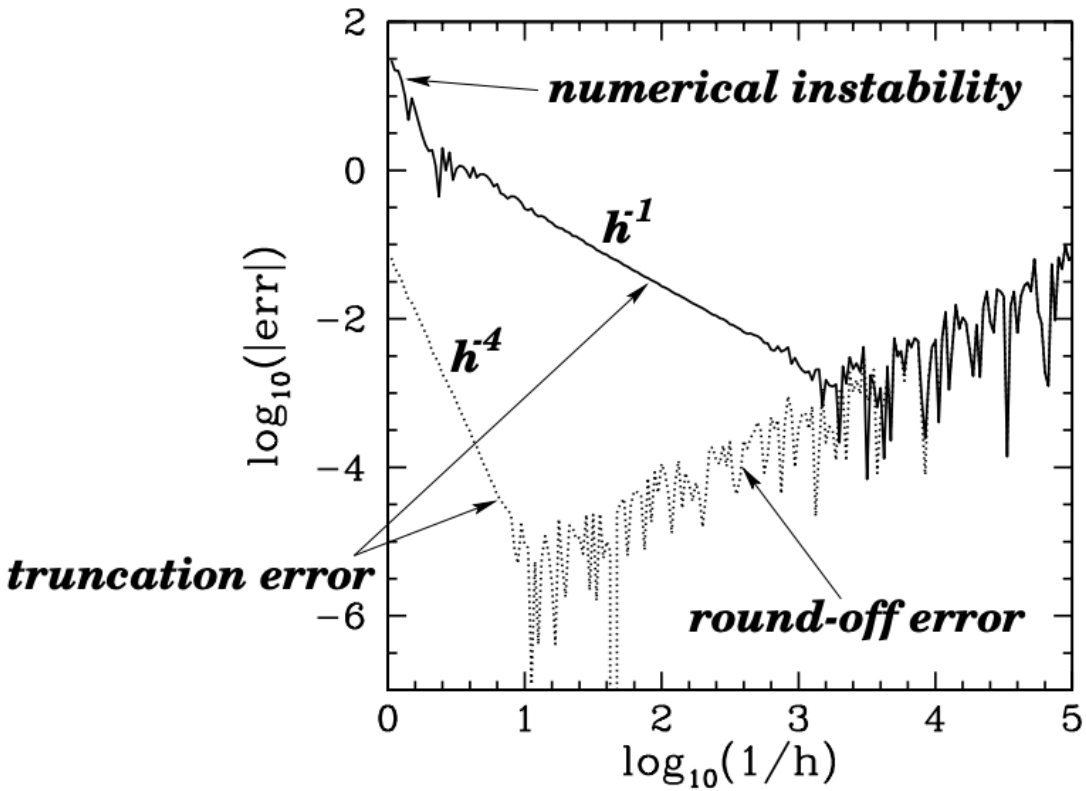
We know the analytic solution to be

$$x = \sin \sqrt{k}\, t.$$

Let us compare the above solution with that obtained numerically using either Euler's method or a fourth-order Runge-Kutta method for double and single precision.

global integration errors associated with Euler's method (solid curve) and a fourth-order Runge-Kutta method (dotted curve) plotted against the step-length Δt I

Single precision calculation



Discussion I

- at large values of Δt, the error associated with Euler's method becomes much greater than unity (i.e., the magnitude of the numerical solution greatly exceeds that of the analytic solution), indicating the presence of a numerical instability
- no similar signs of instability associated with the RK method
- at intermediate Δt, the *truncation error* associated with Euler's method decreases smoothly like Δt^{-1}
- the *truncation error* associated with the RK method scales like Δt^{-4}
- as Δt is decreased, the error associated with both methods eventually starts to rise in a jagged curve that scales roughly like Δt. This is a manifestation of *round-off error*.
- The minimum error associated with both methods corresponds to the boundary between the *truncation error* and *round-off error dominated* regimes.

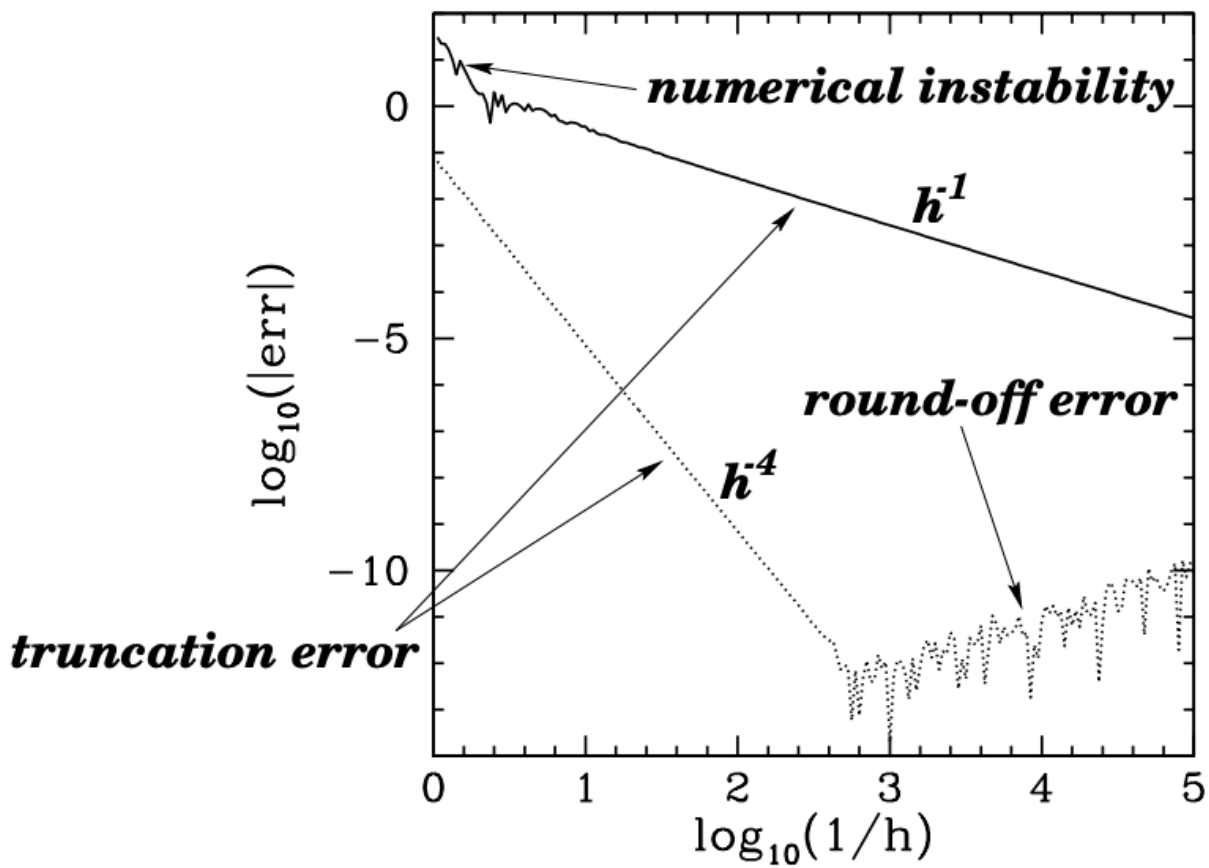
Discussion II

- Thus, for Euler's method the minimum error is about 10^{-3} at $\Delta t \sim 10^{-3}$, whereas for the Runge-Kutta method the minimum error is about 10^{-5} at $\Delta t \sim 10^{-1}$.

Clearly, the performance of the Runge-Kutta method is vastly superior to that of Euler's method, since the former method is capable of attaining much greater accuracy than the latter using a far smaller number of steps (i.e., a far larger Δt).

global integration errors associated with Euler's method (solid curve) and a fourth-order Runge-Kutta method (dotted curve) plotted against the step-length Δt I

Double precision calculation



10.2 Remarks on Conserved Quantities I

- given a conservative hamiltonian system, we recognize energy as a conserved quantity.
- however when we call our integrator, it is not obvious whether energy is conserved to high order
- in fact, it is unlikely
- the integrators we have discussed are designed to match the equations of motion to a certain order and so to low order in step-size they will conserve energy
- so far we have not required energy to be conserved at each step, so on long timescales there – most likely – will be a drift in energy.

10.2 Remarks on Conserved Quantities II

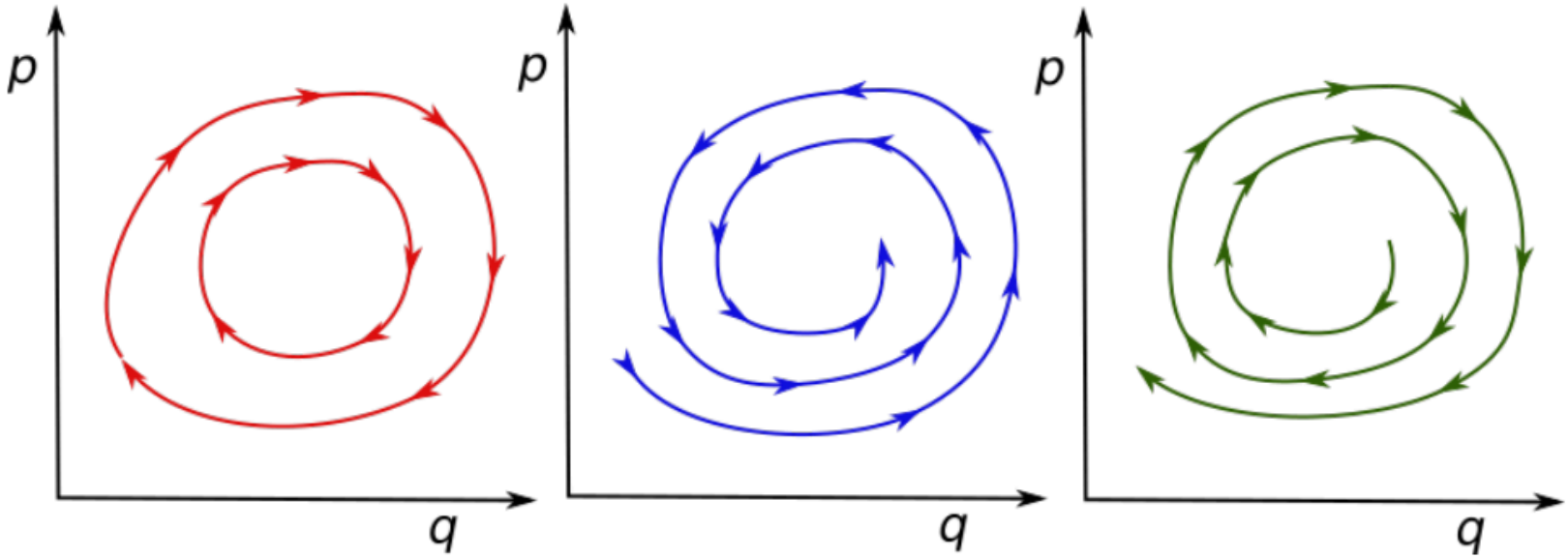


Figure: On left we show trajectories in a system where area is preserved. In the middle we show a system where volume contracts, as would occur when energy dissipation takes place. On the right the system gains energy.

10.2 Remarks on Conserved Quantities III

Consider a two-dimensional system on the plane with density $\rho(x, y, t)$ conservation of mass is

$$\frac{d\rho}{dt} = -\nabla \cdot (\rho \vec{v})$$

with $\vec{v} = (\dot{x}, \dot{y})$. If the system is incompressible and density stays the same at all times then

$$\nabla \cdot \vec{v} = 0$$

i.e. $\partial_x v_x + \partial_y v_y = \partial_x \dot{x} + \partial_y \dot{y} = 0$. Now going back to our 2d integrated system with $\vec{x} = (y, v)$ and the incompressibility condition.
Note: $\dot{v} = -\partial_y U(y)$ is independent of v and so $\partial_v \dot{v} = 0$.

10.2 Remarks on Conserved Quantities IV

Remark

- *our two dimensional system is conserving density in phase space*
- *alternatively, being incompressible in phase space.*
- *this is related to energy conservation and that we have used a conservative force.*
- *if the force depends on velocity, then* $\partial_v \dot{v} \neq 0$ *and volume in phase space is not conserved.*
- *in a dissipating system, volume contracts and in a system that gains energy volume increases as depicted in figure above*

10.3 Symplectic and non-symplectic first order integration for the Harmonic oscillator I

A simple example is that of a harmonic oscillator. Setting the spring constant $k = 1$ and momentum per unit mass $p = \dot{q}$, the energy per unit mass is

$$H = \frac{1}{2}\left(p^2 + q^2\right)$$

The force per unit mass is $f(q) = -q$ and potential energy $U(q) = q^2/2$
The equations of motion are $\dot{q} = p$ and $\dot{p} = -q$, this together with $\ddot{q} = -q$ constitutes oscillation with angular frequency $\omega = 1$.
The exact evolution of the system is

$$\begin{bmatrix} q(\tau) \\ p(\tau) \end{bmatrix} = \begin{bmatrix} \cos\tau & \sin\tau \\ -\sin\tau & \cos\tau \end{bmatrix} \begin{bmatrix} q(0) \\ p(0) \end{bmatrix} \tag{1}$$

10.3 Symplectic and non-symplectic first order integration for the Harmonic oscillator II

and we see that

$$\left|\det \begin{bmatrix} \cos\tau & \sin\tau \\ -\sin\tau & \cos\tau \end{bmatrix}\right| = \cos^2\tau + \sin^2\tau = 1. \tag{2}$$

The matrix is the Jacobian of the transformation, this becomes clear if we think of the transformation as

$$\vec{x}' = Ax.$$

The Jacobian is a matrix composed of elements $\frac{\partial x_i'}{\partial x_j}$ and as the relation between $\vec{x}'$ and $\vec{x}$ is linear $\frac{\partial x_i'}{\partial x_j} = A_{ij}$. If

$$|\det A| = 1$$

the transformation is volume preserving.

10.3 Symplectic and non-symplectic first order integration for the Harmonic oscillator III

Expand the cosine and sine to first order in τ

$$\begin{bmatrix} q' \\ p' \end{bmatrix} = \begin{bmatrix} 1 & \tau \\ -\tau & 1 \end{bmatrix} \begin{bmatrix} q \\ p \end{bmatrix} \tag{3}$$

and see that

$$\left| \det \begin{bmatrix} 1 & \tau \\ -\tau & 1 \end{bmatrix} \right| = 1 + \tau^2 \neq 1. \tag{4}$$

After one timestep the energy is

$$H' = \frac{1}{2}\left(p'^2 + q'^2\right) = \frac{1}{2}\left(1 + \tau^2\right)\left(p^2 + q^2\right).$$

$\rightarrow$ since $1 + \tau^2 > 1$ the energy will increase at every time step

$\rightarrow$ volume in phase space is not conserved

10.3 Symplectic and non-symplectic first order integration for the Harmonic oscillator IV

We can fix this by making A area preserving with

$$\begin{bmatrix} q' \\ p' \end{bmatrix} = \begin{bmatrix} 1 & \tau \\ -\tau & 1-\tau^2 \end{bmatrix} \begin{bmatrix} q \\ p \end{bmatrix} \tag{5}$$

The determinant of the transformation matrix is 1. The transformation is area preserving and so symplectic. Since the transformation is symplectic it does preserve an energy. However the quantity that is conserved is not the original energy H. The transformation gives a different energy

$$H' = \frac{1}{2}(q^2 + p^2 + \tau^2(q^2 - p^2) + \tau^4 p^2 + 2\tau^3 qp)$$

However by inserting q' and p' into

$$H_{integ} = \frac{1}{2}(q^2 + p^2) + \frac{\tau}{2}pq$$

10.3 Symplectic and non-symplectic first order integration for the Harmonic oscillator V

we can show that this quantity is conserved and does not change. The integrated Hamiltonian preserved differs from the true Hamiltonian. The above Hamiltonian is called the **modified Hamiltonian**. Just as finite differencing techniques can better approximate a modified differential equation, a symplectic method preserves a Hamiltonian that differs from that intended. Since the H_{integ} is conserved, it is likely that the difference $H_{integ} - H$ never gets very big. In this sense we expect the error to be bounded and not grow forever

10.3 Symplectic and non-symplectic first order integration for the Harmonic oscillator VI

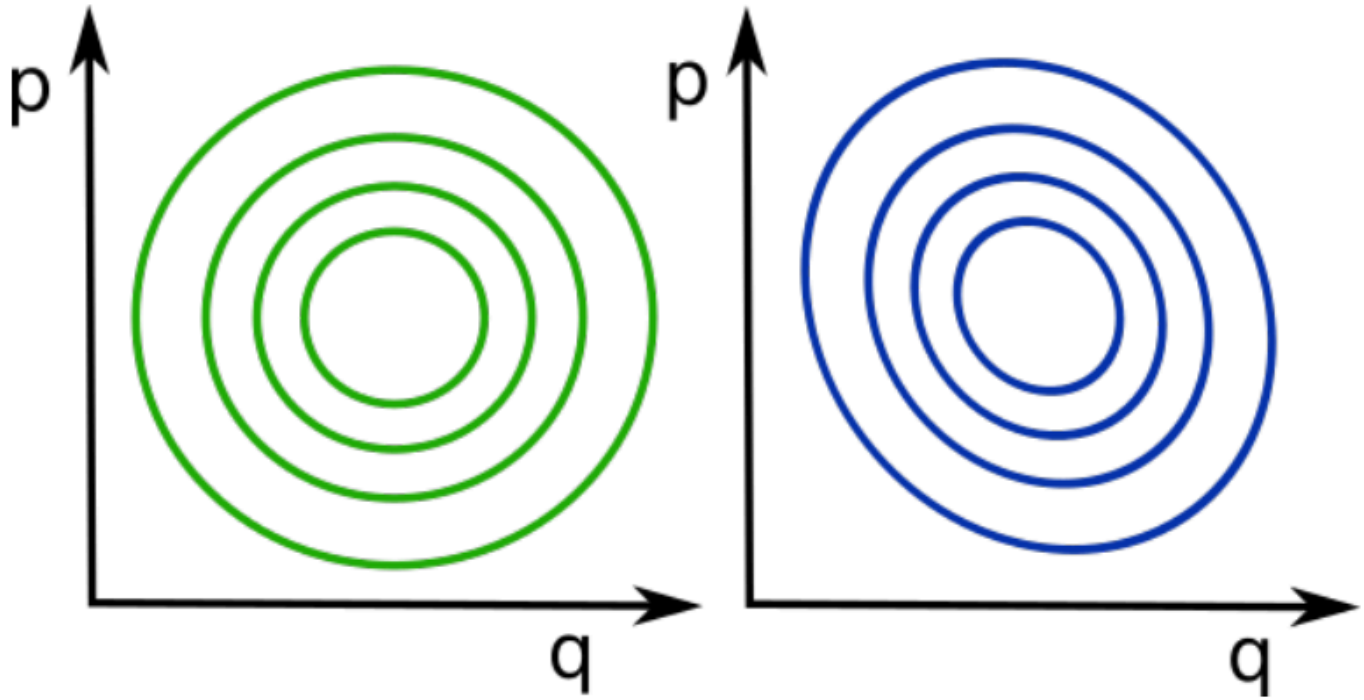


Figure: On the left, we show level curves of the Hamiltonian $H = 1/2(p^2 + q^2)$ for the harmonic oscillator. On the right we show level curves for $H_{integ} = \frac{1}{2}(q^2 + p^2) + \frac{\tau}{2}pq$. Trajectories on the integrated Hamiltonian do not

10.3 Symplectic and non-symplectic first order integration for the Harmonic oscillator VII

continuously diverge from those of the real system because they are confined to level curves.