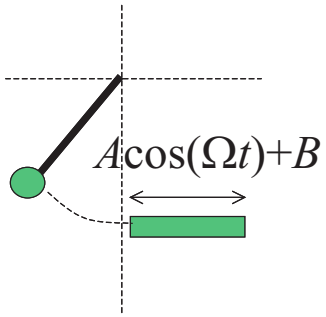
Advanced Programming in Engineering
Exercise ODE 2

Problem

Although the equations of motion of classical mechanics are fully deterministic, the systems described by these equations may show surprisingly complex – even 'chaotic' – behaviour. As an example hereof, we are going to study a pendulum bouncing against a periodically oscillating wall, see figure below. The goal of the exercise is to study the motion of the pendulum, with a focus on the collision series. We will explore the repeating part of the collision series as a function of the system parameters, such as the oscillation amplitude and frequency of the wall. We will also study how the collision series convergences toward the repeating series, and how smooth changes in the continuous system parameters result in transitions between the discrete collision series.



Background

For computational ease, the pendulum used in experiments will be replaced by a damped harmonic oscillator in the simulations. The equation of motion for the displacement x of the pendulum, relative to its equilibrium position, is then given by

$$m\frac{d^2x}{dt^2} = -kx - \gamma\frac{dx}{dt}, \tag{1}$$

where t denotes time, m the mass, k the spring constant and γ the friction coefficient. The position of the periodically moving wall reads as

$$x_{wall}(t) = A\cos(\Omega t) + B, \tag{2}$$

where A denotes the amplitude, Ω the angular frequency and B the average position of the wall relative to the average position of the pendulum. The pendulum is always to the *left* of the impenetrable wall,

$$x(t) \leq x_{wall}(t), \tag{3}$$

where the equal sign holds true at the moments of collision. The motion of the wall is not perturbed by the collisions; the motion of the pendulum is perturbed only at the exact moment of collision. At every collision, the velocity difference between pendulum and wall instantaneously inverts sign. Hence, the initial and final velocities of the pendulum at the n^{th} collision are related by

$$\dot{x}_{final}(t_n) - \dot{x}_{wall}(t_n) = -\left\{\dot{x}_{initial}(t_n) - \dot{x}_{wall}(t_n)\right\}, \tag{4}$$

with the wall velocity given by $\dot{x}_{wall}(t) = -A\Omega\sin(\Omega t)$. The collisions turn the simple deterministic motion of the pendulum into the surprisingly complex collection of orbitals traced out by the bouncing pendulum.

To highlight the relevant parameters of the systems, we convert the equations of motion to *dimensionless scaled* quantities. We chose to scale time by the eigenfrequency ω of the undamped pendulum,

$$\tau = \omega t \quad , \quad \omega^2 = c/m, \tag{5}$$

while the positions are scaled by the amplitude A of the wall oscillation. The equation of motion for the scaled sway of the pendulum, $y(\tau) \equiv x(\tau/\omega)/A$, then reads as

$$\frac{d^2y}{d\tau^2} = -y - \delta\frac{dy}{d\tau}, \tag{6}$$

with $\delta = \gamma/(m\omega)$. The scaled motion of the wall follows from

$$y_{wall}(\tau) = \cos(\rho\tau) + b, \tag{7}$$

where $\rho = \Omega/\omega$ and $b = B/A$. The boundary condition for the pendulum becomes

$$y(\tau) \leq y_{wall}(\tau), \tag{8}$$

and the change of the pendulum's velocity during the n^{th} collision turns into

$$\dot{y}_{final}(\tau_n) = -\dot{y}_{initial}(\tau_n) + 2\dot{y}_{wall}(\tau_n), \tag{9}$$

with the wall velocity given by $\dot{y}_{wall}(\tau) = -\rho\sin(\rho\tau)$.

The above conversions have reduced 'parameter space' from six dimensional parameters[1] to a mere three relevant dimensionless parameters: ρ, δ en b. In combination with the starting point, $y(\tau_0)$ and $\dot{y}(\tau_0)$, they fully determine the path of the pendulum.[2]

We are dealing with a periodically driven non-linear dynamical system with two degrees of freedom (y en $\dot{y}$) and friction ($\delta > 0$). Because of the friction, the path of the system converges to a 'discrete stationary state': either a periodically repeating orbit (known as an *attractor*) or a chaotic orbit (a so-called *strange attractor*), depending on the initial conditions and the dimensionless parameters.

Our first objective, in questions 1 to 3, is to explore and characterize the discrete stationary states of the pendulum orbit. Our second objective, in questions 4 to 6, is to study and understand how the pendulum makes transitions from one discrete state to another discrete state upon changing a continuous system parameter.

The core of the numerical problem in Eqs (6) to (9) is to solve the pendulum's motion, and in particular its collision with the wall, with sufficient accuracy. The result of a simulation of N collisions is the *collision series* stored in the arrays

$$\texttt{tcoll[n]} = \tau_n, \quad \texttt{ycoll[n]} = y_n = y(\tau_n) \quad \text{and} \quad \texttt{vcoll[n]} = v_n = \dot{y}_{initial}(\tau_n), \tag{10}$$

for $1 \leq n \leq N$. This series contains all essential information on the orbit of the pendulum. To solve the motion $\texttt{yslinger}[\ldots] = y(\tau)$ of the pendulum between two consecutive

[1] Namely: m, c, γ, A, B and Ω. These parameters will not return in the course of this exercise, and hence will *not* appear in your codes.

[2] This type of analysis is often useful to identify the relevant independent parameters of any system under investigation. The less parameters to vary, the quicker one can scan parameter space.

collisions, $\tau_n \leq \tau \leq \tau_{n+1}$, we will use matlab's `ode45` routine.[3] The `ode45` solver can be instructed to automatically abort at the point of collision by supplying an appropriate `Events` option, i.e. a user-defined function reaching a value of zero at the moment of collision. This option forces `ode45` to adjust the last couple of time steps before the collision in such a way that the pendulum stops at $y(\tau) \approx y_{wall}(\tau)$. Note that numerically solving the zero-point $f(x) = 0$ of a function f rarely yields the exact zero point x. In stead, a zero-point solver returns a value `x` that satisfies `abs(f(x)) < epsilon`, where `epsilon` is the user-specified or default accuracy. The calculated collision time could be either before or after the true moment of collision, and in the latter case your pendulum may behave oddly. If this happens in your simulations, you have to think of a solution.

From prior experience we know that it is tempting to produce a long report with many pictures. *Please constrain yourselves and keep the report to the point.* Formulate your results and conclusions clearly, and illustrate them, where necessary, with a *representative* plot. Always mention all parameters used and provide the simulation code, so the calculations are repeatable by the reader.

Questions

1 Write an algorithm to calculate the motion $y(\tau)$ of the bouncing pendulum. Calculate the path of the pendulum for $b = 0$, $\rho = 2$ and $\delta = 0.1$, using a starting point left of the wall.

- Show the paths of the pendulum and the wall, $y(\tau)$ and $y_{wall}(\tau)$, in one figure, and verify whether your code works correctly.
- Analyse whether and how the path depends on the starting point.
- Study the influence of the friction coefficient, for $0 \leq \delta \leq 1$.

Compare, for $b = 0$ and $\delta = 0.1$, the $\rho = 2$ path with the $\rho = 3$ path.

2 In stead of looking at the entire trajectory $y(\tau)$ of the pendulum, we now restrict ourselves to the collision series. Plot, for $b = 0$ and $\delta = 0.1$, with $\rho = 2$ and $\rho = 3$:

- the collision positions y_n versus n.
- the collision velocities v_n versus n.
- the intervals between consecutive collisions: $T_n = \tau_n - \tau_{n-1}$ versus n.

  Make sure that your simulations are long enough to have reached the final periodically repeating collision series.

Make *phase portraits* of both trajectories. That is, use markers (no lines) to plot:

- the collision positions y_n against the collision velocities v_n.
- the collision positions y_n against the collision intervals T_n.

  To clarify your results, draw the first part of the trajectory, in which transient effects still dominate, in a different colour than the second part of the run, in which the path has converged to the stable series.

Describe what you see and what you conclude.

[3] In exercise ODE 1 we have seen that `ode45` offers the best accuracy over short time intervals. Furthermore, we need the position and velocity of the pendulum at the moment of collision. Both are readily calculated by `ode45`, while requiring a number of non-trivial changes of the leap-frog and Verlet algorithms.

An important signature of the calculated collision series is the periodicity of the series, after the start-up effects have faded. The series of collision positions, velocities and intervals repeat themselves with smallest recurring unit of p steps:

$$y_{n+p} = y_n, \quad v_{n+p} = v_n \quad \text{and} \quad T_{n+p} = T_n, \quad \forall\, n. \tag{11}$$

Your graphs are readily used to determine the repeat length p. The series of collision times rises monotonically, though here too there is a regularity:

$$\tau_{n+p} = \tau_n + T, \quad \forall\, n. \tag{12}$$

During the time interval T the wall performs q oscillations, with q an integer number. One readily shows that

$$q = \frac{\rho T}{2\pi}. \tag{13}$$

The integers p and q are useful to 'characterize' the collision series.

3 Characterize the collision series for the following parameters:

$$b = -1, 0, +1, \quad \text{en} \quad \rho = 3/2, 2, 5/2, 3, 7/2.$$

Keep the friction fixed at $\delta = 0.1$ and start every simulation from the same initial point. Collect your results in a table, showing[4]

- the repeat length p of the repeating collision series,
- the number of oscillations q of the wall during the repeating series,

for the above combinations of b and ρ.

Since we are interested here in the periodic part of the series, one should make sure that your simulations are long enough for the start-up effects to have vanished, by adjusting the run length accordingly.
Hint: in the analysis it might be convenient to relate T to the collision intervals T_n.

Most of the parameter values in question 3 saw the collision series converge to a stable repeating unit, the so-called *attractor*. We are now going to explore the rate of convergence, i.e. the extinction rate of the transient behaviour, which is also a measure for the stability of the attractor. This we do by looking at the norm d_n, which measures the distance[5] in the (y, v) phase portrait between the n^{th} collision point in the simulation and the matching stable point in the repeating series. For a $p = 1$ orbital:

$$d_n = \|(y_n, v_n) - (y_\infty, v_\infty)\|\,. \tag{14}$$

It has been predicted that the norm decays exponentially to zero,

$$d_n \propto \exp(-\alpha n), \tag{15}$$

where α is a function of the parameters of the system.

[4] Do not waste your time on an algorithm to automatically determine p and q.
[5] There is no problem here with the units, since both y and v are dimensionless.

4
- Study the convergence rate of a $p = 1$ collision series; the series with $\rho = 2$, $b = 0$ and $0.1 \leq \delta \leq 1.0$ is nicely behaved. Make a clear plot of the norm against collision number, and determine the convergence rate α.
- The character (p and q) of an orbit is typically conserved under moderate changes of the friction parameter, but in question 1 we already observed that δ affect the convergence rate. Use your simulation and a clear graph to express α as a function of δ.

In the preceding questions we have seen how the repeating collision series of a stable attractor (the block of p values in y_n, v_n and τ_n) depends on the system parameters ρ, b and δ. Upon a slight change of one of these parameters, all collision points will change a little bit. One readily imagines the consequences of a small change of a continuous variable y_n, but it is much more difficult to imagine what happens when a stable collision series with one unique collision changes in a series with two alternating collisions. What happens when discrete variables like p or q change value?

5 Study the transition between two stable collision series with distinct repeat lengths p. A nice example can be found going from $\rho = 2.8$ to $\rho = 3.0$, for $b = -1$ and $\delta = 0.1$.

- Plot the *bifurcation diagram* of this transition, i.e. a scatterplot of the p stable collision postitions y_n against the varying frequency ratio ρ. To generate this plot, we perform a series of simulations at various values of ρ, in which each simulation is long enough to converge. Plan ahead, as this can take some time... Note that it is not necessary to determine the value of p for every individual simulation; in stead, we simply plot for every ρ the positions y_n of the last 25 collisions.
- Zoom in on the interesting area(s) of your diagram, e.g. the range surrounding a a bifurcation point,[6] by calculating a more detailed diagram. This procedure may be repeated several times to get a clear picture of what happens at the transition(s).
- Can you explain the mechanism by which a series with given p and q develops, through a small change of ρ, into a series with a different p and/or q?

Additional questions (for a grade > 8)

a Calculate the convergence rate(s) for a collision with $p > 1$.
Hint: find the proper modification to Eq. (14).

b If you found a nice bifucation point in question 5: Use a procedure like the one developed in the above question to investigate the convergence rate α as a function of ρ in the vicinity of the bifurcation point.

An often used technique in the analysis of periodic data is the *Fourier transformation*. The function $f(x)$, over the interval $0 \leq x \leq L$, is expressed as the summation

$$f(x) = \frac{1}{L} \sum_{m=1}^{\infty} \beta_m \exp\left(2\pi i \frac{x}{L} m\right), \tag{16}$$

[6] Latin: two-fork.

where $i^2 = -1$. The complex Fourier coefficients are solved from

$$\beta_m = \int_0^L f(x) \exp\left(-2\pi i \frac{x}{L} m\right)\, dx. \tag{17}$$

We will now transform *not* the pendulum's path $y(\tau)$ but its collision series y_n, where $1 \leq n \leq N$ en $N \gg p$ (and excluding start-up effects). Matlab contains a version of the famous 'fast Fourier transform' routine, `beta = fft(y)`, to calculate[7]

$$\texttt{beta[k]} = \sum_{\texttt{j}=1}^{N} \texttt{y[j]} \exp\left[-2\pi i \frac{\texttt{j}-1}{N}(\texttt{k}-1)\right], \qquad 1 \leq \texttt{k} \leq N, \tag{18}$$

$$\texttt{y[j]} = \frac{1}{N} \sum_{\texttt{k}=1}^{N} \texttt{beta[k]} \exp\left[2\pi i \frac{\texttt{j}-1}{N}(\texttt{k}-1)\right]. \tag{19}$$

In the original fft-routine the length N had to be a power of 2, which is not required by matlab's `fft`.[8] The Fourier coefficients contain information on the periodicity of the collision series. To visualize this information, we look at the *power spectrum*, i.e. the plot of $\|\beta_k\|^2$ against k/N. Your result will look at their nicest when N is an integer multiple of p, though this becomes less significant with increasing N.

c Calculate the power spectrum for a couple of long stable collisions series with various values of p. What strikes you about these graphs, and can you explain this?
Hint: do not plot the dominating DC component.

d Finally, calculate the power spectra for some of the 'chaotic' orbits that you came across in the preceding questions. Wat do you see?

[7] A daft '–1' appears in these equations because matlab does not allow a zero$^{\text{th}}$ array-element.

[8] There are a number of computational niceties when N is a power of 2, which drastically speed up the calculation. If your data set has a different length, matlab tacitly extend your series to a power of 2.