Exercise

A particle of mass m moves in a one-dimensional double-well potential and is subject to an additional frictional force proportional to its velocity. Thus the equation of motion for the position $x(t)$ as a function of time t is

$$m\ddot{x} = \mu\dot{x} + ax - bx^3$$

with positive constants μ, a, and b. Depending on the initial state, the particle will end up either in the left ($x < 0$) or the right potential well.

Using the constants $\mu/m = 0.1$, $a/m = b/m = 1$ (in suitable units), calculate and plot those regions in the plane of initial states $(x(0), \dot{x}(0))$ for which the particle ends up in the left well, i.e., at $x(\infty) = -1$.

Literature

Crandall R.E. (1991) Mathematica for the Sciences. Addison-Wesley, Redwood City, CA

Koonin S.E., Meredith D.C. (1990) Computational Physics, Fortran Version. Addison-Wesley, Reading, MA

Jodl H.-J., Korsch H.J. (1994) Chaos: A Program Collection for the PC. Springer, Berlin, Heidelberg, New York

Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P. (1992) Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press, Cambridge, New York

Schmid E.W., Spitz G., Lösch W. (1990) Theoretical Physics on the Personal Computer. Springer, Berlin, Heidelberg, New York

Stoer J., Bulirsch R. (1996) Introduction to Numerical Analysis. Springer, Berlin, Heidelberg, New York

DeVries P.L. (1994) A First Course in Computational Physics. Wiley, New York

4.2 The Chaotic Pendulum

The French mathematician Henri Poincaré showed, as early as the end of the nineteenth century, that a simple mechanical system can exhibit complex behavior. The idea that one merely has to specify the initial conditions as accurately as possible to be able, in principle, to precisely predict the motion of a mechanical system from its Newtonian equations of motion is, in practice, taken to the point of absurdity, even by simple models. In general, a system reacts so sensitively to the initial conditions that a minute uncertainty of the initial values will lead to a large uncertainty in the prediction after just a short time. This is true not only for the weather forecast, but also for a simple, externally driven pendulum.

Though this fact has been known for a long time, it took almost 100 years for its significance to be recognized and investigated by the scientific community. Only the computer made it feasible to investigate chaotic motion in detail and – what is possibly just as important – display it graphically.

In this section, we want to numerically calculate a simple example: the nonlinear pendulum with frictional force, which is driven by a periodic external force.

Physics

We first consider the pendulum from Sect. 1.2 and add a frictional force. In dimensionless form, the equation of motion for the displacement angle $\varphi(t)$ is

$$\ddot{\varphi} + r\dot{\varphi} + \sin\varphi = 0 \tag{4.17}$$

with a friction coefficient $r \geq 0$. This second-order differential equation can be rewritten in the form of two first-order equations by introducing $\omega(t) = \dot{\varphi}(t)$:

$$\begin{aligned} \dot{\varphi} &= \omega \ , \\ \dot{\omega} &= -r\omega - \sin\varphi \ . \end{aligned} \tag{4.18}$$

The motion of the pendulum can therefore be represented in two-dimensional phase space. Each value (φ, ω) uniquely determines its change $(\mathrm{d}\varphi, \mathrm{d}\omega)$ over a time interval $\mathrm{d}t$. Therefore, an orbit $(\varphi(t), \omega(t))$ cannot intersect itself.

It can be shown that this means that there cannot be any chaotic motion. In two dimensions, there is simply no room for trajectories that do anything other than form closed orbits or approach them. In our case, the pendulum relaxes to the rest position $(\varphi = 0, \omega = 0)$ for $r > 0$, possibly after several oscillations or full turns. The origin of phase space is an attractor for nearly all starting points $(\varphi(0), \omega(0))$. Eventually, all orbits spiral towards the rest position (see Fig. 4.2).

The picture changes if we allow a third direction in phase space. To do so, we drive the pendulum by a periodic torque of strength a and frequency ω_{D},

$$\ddot{\varphi} + r\dot{\varphi} + \sin\varphi = a\cos(\omega_{\mathrm{D}}t) \ . \tag{4.19}$$

Using $\theta = \omega_{\mathrm{D}}t$ we obtain

$$\begin{aligned} \dot{\varphi} &= \omega \ , \\ \dot{\omega} &= -r\omega - \sin\varphi + a\cos\theta \ , \\ \dot{\theta} &= \omega_{\mathrm{D}} \ . \end{aligned} \tag{4.20}$$

The motion of the pendulum is now described in three-dimensional space $(\varphi, \omega, \theta)$. There are three parameters $(r, a, \omega_{\mathrm{D}})$. Without driving torque and friction $(a = r = 0)$ and for small angles $\varphi \ll 1$ we get a harmonic oscillation with frequency $\omega_0 = 1$. Therefore there are, generally, three competing time scales: the period $2\pi/\omega_{\mathrm{D}}$ of the driving torque, the period $2\pi/\omega_0$ of the free oscillation, and the relaxation time $1/r$. Here, just as before in the cases of the Hofstadter butterfly and the Frenkel–Kontorova model, the competition between different length or time scales leads to interesting physical phenomena.

The three-dimensional motion $(\varphi(t), \omega(t), \theta(t))$ is difficult to analyze, even in cases like ours, where θ is just a linear function of time. In order to reduce the multitude of possible forms of motion to the essential structures, the motion is observed only at fixed time intervals, as if the system were illuminated stroboscopically. The result is called a *Poincaré section.*

For the time interval, we take the period of the driving torque and consider

$$(\varphi(t_j), \omega(t_j)) \quad \text{with} \quad t_j = \frac{2\pi j}{\omega_{\mathrm{D}}} \quad \text{and} \quad j = 0, 1, 2, \ldots \,. \tag{4.21}$$

This gives us a sequence of points in the plane, where each point is uniquely determined by its predecessors. As in the case of the chain on a corrugated surface (Sect. 3.2), we reduce the problem to a two-dimensional discrete function, which this time, however, is not area-preserving, if $r > 0$. Rather, an area segment gets smaller in the iteration and attractors are generated, as for the one-dimensional logistic map (Sect. 3.1).

What do the orbits $(\varphi(t_j), \omega(t_j))$ look like in the Poincaré section? For a periodic motion with period $2\pi/\omega_0$, the stroboscopic exposure reveals either individual points or a closed curve. For $\omega_0 = \omega_{\mathrm{D}}$, we get a single point in the (φ, ω) plane. For $\omega_0 = (p/q)\omega_{\mathrm{D}}$ with p and q integer and relatively prime, one obtains q different points, and p determines the order in which the q points are accessed. If, on the other hand, ω_0 is an irrational multiple of ω_{D}, we get an infinite number of points $(\varphi(t_j), \omega(t_j))$ filling a closed curve. Thus, periodic motion results in either individual points or closed curves in the Poincaré section. Such orbits can be attractors, i.e., if one starts with points $(\varphi(0), \omega(0))$ in a certain basin of attraction, then all these values move towards such a periodic attractor. Multiple attractors with their associated basins of attraction are possible as well.

There is a second kind of attractors, though, so-called *strange attractors.* They can be visualized as a kind of puff pastry dough, which is obtained by repeated stretching and folding. Such attractors correspond to chaotic orbits which traverse phase space in a seemingly unpredictable manner. The strange attractors are fractals (see Sect. 3.3), i.e., they are more than a line, but less than an area.

As in Sect. 3.3, the fractal dimension D can be determined by covering the attractor with N squares whose edges have the length ε:

$$D_{\mathrm{B}} = -\lim_{\varepsilon \to 0} \frac{\log N(\varepsilon)}{\log \varepsilon} \,. \tag{4.22}$$

A method that is numerically more efficient was suggested by Grassberger and Procaccia: We generate N points $\boldsymbol{x}_i$ on the attractor, which should have as little correlation as possible. Then we count the number of points $\boldsymbol{x}_j$ whose distance from $\boldsymbol{x}_i$ is less than a given value R, and average this number over the $\boldsymbol{x}_i$. We can formally express this correlation using the step function $\Theta(x)$:

$$C(R) = \lim_{N\to\infty} \frac{1}{N(N-1)} \sum_{i\neq j}^{N} \Theta\left(R - |\boldsymbol{x}_i - \boldsymbol{x}_j|\right) . \tag{4.23}$$

$C(R)$ can be interpreted as the average mass of a section of the attractor, and the relation between mass and length already used in Sect. 3.3 defines a fractal dimension D_{C} via

$$C(R) \propto R^{D_{\mathrm{C}}} . \tag{4.24}$$

A log–log plot of this equation should therefore yield a straight line with slope D_{C}. This, however, is only true for values of R which are larger than the average distance between the data points, and smaller than the size of the attractor.

Another method for determining the fractal dimension uses the concept of information entropy. Again, the attractor is covered with squares whose edges have the length ε. Then one counts how many points of the orbit generated lie in each square. Let p_i be the probability that a point (φ, ω) of the attractor is found in square i. Then the entropy is defined as

$$I(\varepsilon) = -\sum_i p_i \ln p_i , \tag{4.25}$$

and the information dimension D_{I} is obtained from

$$D_{\mathrm{I}} = -\lim_{\varepsilon\to 0} \frac{I(\varepsilon)}{\ln \varepsilon} . \tag{4.26}$$

Now it can be shown that the following relation between the three dimensions holds:

$$D_{\mathrm{B}} \leq D_{\mathrm{I}} \leq D_{\mathrm{C}} . \tag{4.27}$$

In practice the three dimensions often agree within the experimental error.

Algorithm

To solve the differential equation (4.20) numerically, we choose the fourth-order Runge–Kutta method from the previous section. As we want to be able to display the movement on the screen while the calculation is still in progress, we use the language C with the appropriate graphics environment. In our program the strength a of the driving torque can be changed by a keystroke during the run; also, we can switch back and forth between displaying the continuous motion and the Poincaré section.

Programming the Runge–Kutta step from (4.11) oneself is not difficult, but we want to demonstrate at this point how to incorporate the routine `odeint` from *Numerical Recipes* into one's own program.

The function odeint integrates the system of equations (4.20) using an adaptive control of the step size. Therefore the points of the orbit that are calculated do not represent its actual development in time. In order to observe that correlation, one should directly use the routine rk4 from *Numerical Recipes.*

First, all variables of the routines used must be declared, and the routines must be added to one's own program. In doing so, it has to be noted that odeint uses the programs rk4 and rkqc and that all routines call certain error-handling routines which are contained in nrutil. One can directly copy all programs into one's own code or add them via #include by, e.g.,

```
#define float double
#include "\tc\recipes\nr.h"
#include "\tc\recipes\nrutil.h"
#include "\tc\recipes\nrutil.c"
#include "\tc\recipes\odeint.c"
#include "\tc\recipes\rkqc.c"
#include "\tc\recipes\rk4.c"
```

If you only have access to *Numerical Recipes 2.0*, you should replace rkqc.c with rkqs.c and rk4.c with rkck.c. The path name has to be modified, of course, to indicate where you have stored the programs. We have added the first command so that all real variables in the program and in the *Numerical Recipes* are of the same type double. The routine odeint is called in the following form:

```
odeint(y,n,t1,t2,eps,dt,0.,&nok,&nbad,derivs,rkqc)
```

Here, y is a vector with the components $(\varphi(t_1), \omega(t_1))$. It is integrated over the interval $[t_1, t_2]$ and then y is replaced by the result $(\varphi(t_2), \omega(t_2))$. The variable n denotes the number of variables. Since we replace θ by $\omega_\mathrm{D}\, t$ again in (4.20), there are only $\mathtt{n} = 2$ variables. With eps we can specify the desired accuracy, and dt is an estimate of the step size required. On output, the variables nok and nbad contain information about the number of steps needed. The parameter derivs is the name of a function that is used to evaluate the right-hand side of (4.20). In other words, derivs(t,y,f) calculates the components of f from t and y. In our example, we have

$$\begin{aligned} \mathsf{y}[1] &= \varphi\,, \\ \mathsf{y}[2] &= \omega\,, \end{aligned}$$

and correspondingly

$$\begin{aligned} \mathtt{f}[1] &= \mathsf{y}[2]\,, \\ \mathtt{f}[2] &= \;-\mathtt{r} * \mathsf{y}[2] \quad - \quad \mathtt{sin}(\mathsf{y}[1]) \;\; + \;\; \mathsf{a} * \mathsf{cos}(\omega_\mathrm{D} * \mathtt{t})\,. \end{aligned}$$

Finally, the routine rkqc, which in turn calls the Runge–Kutta step rk4, is passed to odeint. One can, however, pass other integration procedures, possibly one's own, just as well.

In summary, then, the essential part of the program pendulum.c consists of the following commands:

```
main()
{
y[1]=pi/2.;
y[2]=0.;

while(done==0)
 {
  odeint(y,2,t,t+3.*pi,eps,dt,0.,&nok,&nbad,derivs,rkqc);
  xold=fmod(y[1]/2./pi +100.5,1.)*ximage;
  yold=y[2]/ysc*yimage/2+yimage/2;
  rectangle(xold,yold,xold+1,yold+1);
  t=t+3.*pi;
 }
}

void derivs(double t,double *y,double *f)
 {
   f[1]=y[2];
   f[2]=-r*y[2]-sin(y[1])+a*cos(2./3.*t);
 }
```

Here, too, a good programmer should have all constants calculated in advance. The vectors y and f were declared longer than necessary, by one element, because the indices in the *Numerical Recipes* start at 1 rather than 0 (as is customary in C).

The program above uses $\omega_{\mathrm{D}} = 2/3$ and plots the points $(\varphi(t_i), \omega(t_i))$ (as small rectangles), each of them one period $[t, t + 3\pi]$ of the driving torque after the previous one. In order to observe the continuous orbit, one has to replace $t + 3\pi$ by $t + \mathrm{d}t$, where the time step $\mathrm{d}t$ should be chosen such that the orbit progresses by only a few pixels each time step. In this case, of course, a line from the initial to the final point is plotted instead of a single point.

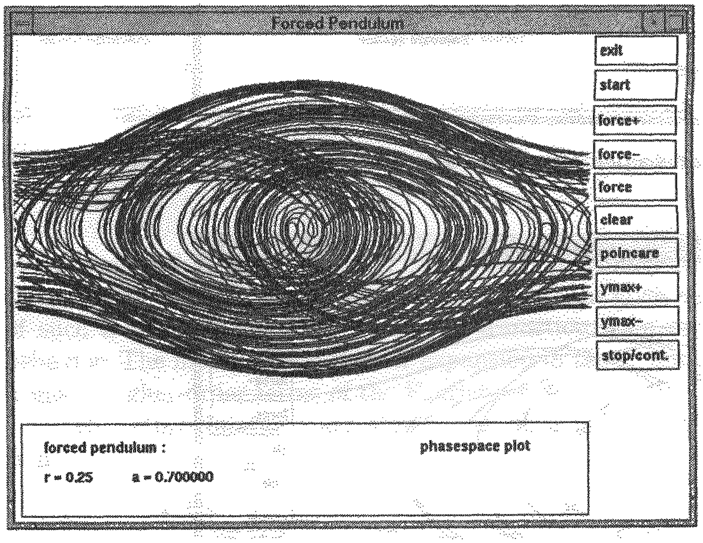
Results

As an example we investigate on screen the forced pendulum with a friction coefficient $r = 0.25$ and a driving frequency $\omega_{\mathrm{D}} = 2/3$. For the initial angle we pick $\varphi(0) = \pi/2$ and $\omega(0) = 0$ for the initial angular velocity. The keys i and d respectively allow us to increment and decrement the driving torque a in steps of 0.01.

Without external torque ($a = 0$), the pendulum executes a damped oscillation and eventually stops at the rest position $(\varphi, \omega) = (0, 0)$. For small values of a we obtain, after a transient phase, a periodic motion with a period $2\pi/\omega_{\mathrm{D}}$, i.e., a single point in the Poincaré section. As a reaches a value of 0.68, the drive is strong enough that the pendulum can turn over at the top. The orbit leaves the screen and reenters on the opposite side, owing to the periodicity of the angle φ (=horizontal axis).

For $a = 0.7$, we observe a chaotic attractor (Fig. 4.3). In the Poincaré section, a fractal "puff pastry dough" begins to show up (Fig. 4.4). For $a = 0.85$, the orbit still looks irregular (Fig. 4.5), but the Poincaré section clearly shows a cycle of length $7 \times 2\pi/\omega_{\mathrm{D}}$. Even period doubling can be observed. The single cycle, with a turnover at $a = 0.97$, doubles for $a = 0.98$ and $a = 0.99$, and quadruples at $a = 1.00$. By the time a reaches 1.01, the orbit looks chaotic yet again. For $a = 1.2$, three points can be seen, and for $a = 1.3$, we get a chaotic orbit with three bands, whose Poincaré section is shown in Fig. 4.6.

As the strength a of the drive increases, we observe a continuous change back and forth between periodic and chaotic motion.



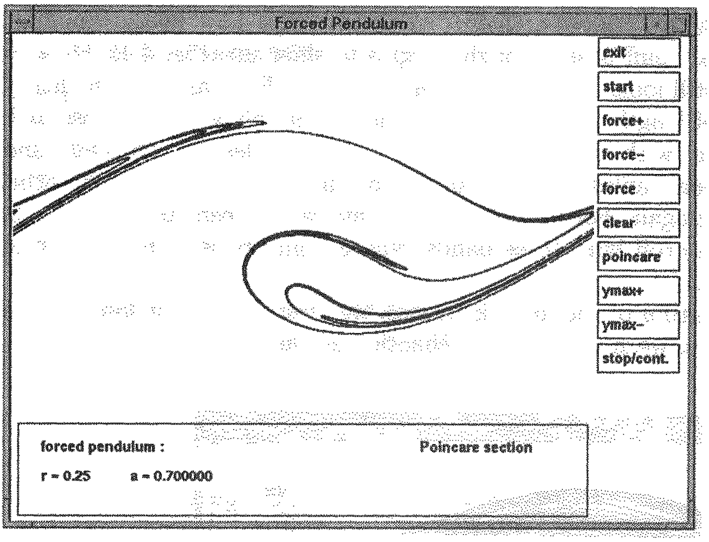
forced pendulum : phasespace plot

r = 0.25 a = 0.700000

Fig. 4.3. Motion of the forced pendulum in (φ, ω) phase space; the driving torque a is 0.7

Exercise

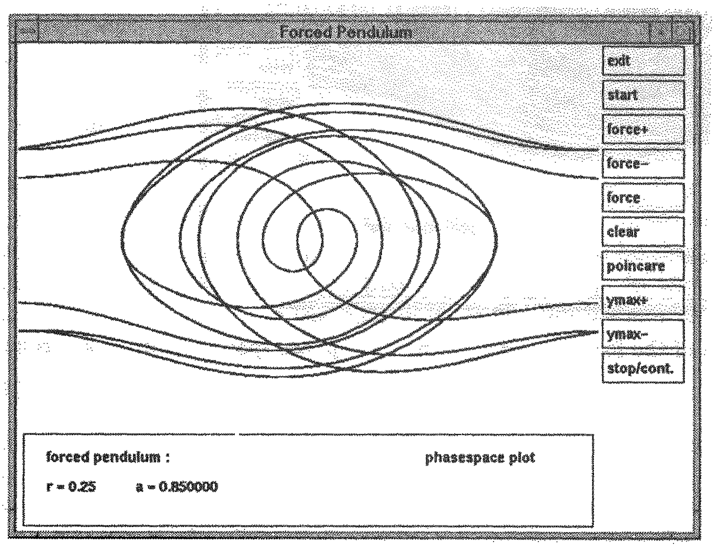
A liquid of given viscosity μ, heat conductivity κ, and density ρ is enclosed between two plates. The upper plate has a temperature T, the lower a temperature $T + \Delta T$. If the temperature difference is small, only heat conduction takes place; any convective motion is prevented by viscous resistance. As the temperature difference increases, the hot liquid starts to rise in some areas and sink again in others; so-called convection cells result. As the temperature increases even further, these static cells are broken up and chaotic motion results.



forced pendulum : Poincare section

r = 0.25 a = 0.700000

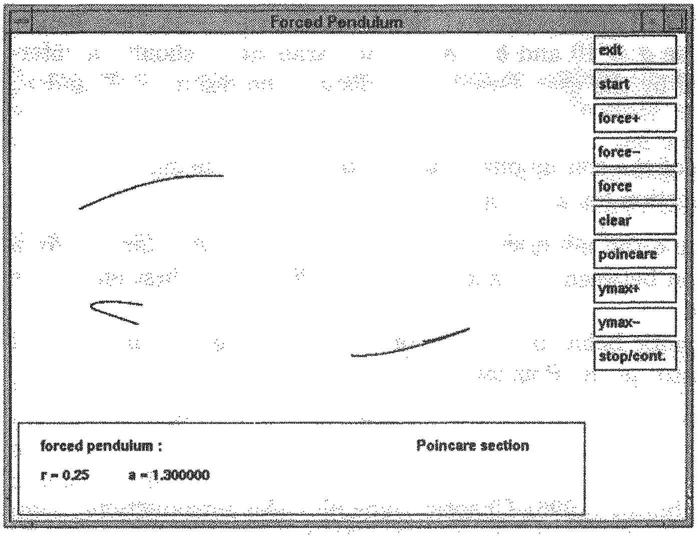
Fig. 4.4. Poincaré section for the motion from Fig. 4.3, taken at time intervals $2\pi j/\omega_D$



forced pendulum : phasespace plot

r = 0.25 a = 0.850000

Fig. 4.5. Phase-space plot for $a = 0.85$. The motion is a cycle of period $7 \times 2\pi/\omega_D$. The corresponding Poincaré section shows exactly seven points



forced pendulum :

r = 0.25 a = 1.300000

Poincare section

Fig. 4.6. Poincaré section for $a = 1.3$. The motion alternates between three chaotic orbits

Assuming that the convection cells extend to infinity in the y-direction, the meteorologist Lorenz has developed a model that gives a good description of such a Bénard experiment near the transition from ordered to chaotic behavior. The model uses a Fourier expansion in the x- and z-directions and neglects higher-order terms of the Fourier series.

In this approximation the system is described by the following three equations:

$$\dot{X} = -\sigma X + \sigma Y \, ,$$
$$\dot{Y} = -XZ + rX - Y \, ,$$
$$\dot{Z} = XY - bZ \, .$$

Here, X represents the velocity of the circular motion, Y the temperature difference between the rising and the sinking liquid, and Z represents the deviation of the resulting temperature profile from the equilibrium profile. The parameters b and σ are determined solely by the material constants (μ, ρ, κ) and the geometric dimensions; consequently, they have to be regarded as constants. The quantity r is proportional to the temperature difference applied and thus serves as an external control parameter that determines the system's behavior.

This system is interesting for various applications:

- in meteorology: air movement;
- in astronomy: stellar structure of convective stars;
- in energy technology: heat conduction of insulating materials.

Integrate the differential equations of the Lorenz model with the Runge–Kutta method. Use $\sigma = 10$ and $b = 8/3$. The parameter r should be interactively controlled by the user. Graphically display the result for X and Y, spe~~ci~~fically

- as a projection of the continuous motion onto the X–Y plane,
- as a Poincaré section for Z = constant = 20.0.

To better determine the intersection with the plane Z = constant = 20, a linear interpolation between the last value below 20 and the first one above 20 should be used.

When does chaotic behavior set in? For $r = 28$, what is the dimension of the strange attractor in the Poincaré section mentioned above?

Literature

Baker G.L., Gollub J.P. (1996) Chaotic Dynamics: An Introduction. Cambridge University Press, Cambridge, New York

Jodl H.-J., Korsch H.J. (1994) Chaos: A Program Collection for the PC. Springer, Berlin, Heidelberg, New York

Ott E. (1993) Chaos in Dynamical Systems. Cambridge University Press, Cambridge, New York

Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P. (1992) Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press, Cambridge, New York

4.3 Stationary States

Newton's equations of motion, however successfully they describe our macroscopic world, turn out to be unsuitable if electrons in atoms, molecules, or solids are concerned. In the realm of the microscopically small, only probability statements for the position and momentum of a particle are possible; their time dependence is determined by the quantum-mechanical Schrödinger equation. Under certain conditions, this dynamic equation reduces to the so-called stationary Schrödinger equation, an eigenvalue equation for the energy operator, which in the coordinate representation takes the form of a linear differential equation. In principle, this equation can be solved numerically using the methods from the previous sections. There is, however, a more efficient method for numerically solving the Schrödinger equation, which we want to illustrate using a simple example, the anharmonic oscillator. For the Schrödinger equation we have the additional problem that the energy cannot take all real values, but only certain discrete ones, which are determined by the normalizability requirement.