

MMA421: Ordinary differential equations and dynamical systems

Computer assignments

The computer assignments described in this document are compulsory. The reports must be delivered in the form of a pdf-document sent by e-mail to `wennberg@chalmers.se` no later than the date of the written exam. The reports will be graded (with grades pass (G) and pass with distinction (VG) for students registered with the University of Gothenburg, and (3), which is equivalent to pass, (4) or (5) for students registered with Chalmers), and the grade will be used when the final grade for the course is determined. To pass the course, the student must obtain pass on the computer assignment.

Purpose

The computer assignments should

- give deeper understanding of the course material;
- introduce the student to numerical methods for studying dynamical systems;
- give a training in written communication of mathematical concepts.

Mathematical software

The assignments are developed to be solved using Matlab, which is available on the computer system. It is allowed to use other types of mathematical software, but a person who chooses a different kind of software must be particularly careful to respect the intention of the assignment. Some familiarity with Matlab is assumed. A booklet with a "Crash course" in Matlab will be available for persons who need it.

The report

You are allowed (and even encouraged) to work in pairs while doing the assignments. However, every student must write his/her own report, and hence must participate in all parts of the work. If you worked with somebody, you should state that in the beginning of the report.

You must write your name and personal identification number in the report

The report does not need to be a complete technical report, and in particular you should not spend a lot of time making it look very nice typographically. It should be sufficiently detailed to be read without the assignment description (this document) in hand. You should present the results of your computer experiments, explain how they were carried out (but I am not very interested in matlab code), and comment on them, and of course, answer all explicit questions given in the text.

Assignment 1: numerical solution of ordinary differential equations

The most elementary way of solving an ordinary differential equation numerically is to approximate derivatives by finite differences. We consider first the initial value problem

$$\begin{aligned}\dot{x}(t) &= f(x(t)) \\ x(0) &= 1\end{aligned}\tag{1}$$

and suppose that we look for approximations in the interval $0 \leq t \leq T$, which we divide into sub intervals $0 = t_0 < t_1 < \dots < t_{N-1} < t_N = T$. We write $h_i = t_i - t_{i-1}$, and from now on we assume that all the h_i are equal, $h = T/N$.

With the *Forward Euler Method* the time derivative is approximated as $\dot{x}(t_i) \approx (x(t_{i+1}) - x(t_i))/h$, and approximations of $x(t_i)$ can be computed as

$$\frac{\tilde{x}(t_{i+1}) - \tilde{x}(t_i)}{h} = f(\tilde{x}(t_i)), \quad (2)$$

or

$$\tilde{x}(t_{i+1}) = \tilde{x}(t_i) + hf(\tilde{x}(t_i)). \quad (3)$$

This is also known as the *explicit Euler method*, because the formula gives an explicit expression for the the following time step. This formula gives the values of $\tilde{x}(t)$ only at discrete time points, and one needs to use some interpolation method to find $\tilde{x}(t)$ for values between these values.

From here, I will let \tilde{x}_k denote $\tilde{x}(t_k)$.

With the *Backward Euler Method*, or *implicit Euler method*, one writes

$$\tilde{x}_{i+1} = \tilde{x}_i + hf(\tilde{x}_{i+1}). \quad (4)$$

This is an implicit method, because one needs to solve an equation to obtain \tilde{x}_{i+1} from \tilde{x}_i – the value of \tilde{x}_{i+1} is only implicitly defined by the formula.

Note that only first order systems can be solved with these methods, and that therefore higher order systems must be rewritten as first order systems.

ASSIGNMENT 1.1: When $f(x) = ax$ for some constant a , one can compute \tilde{x}_j exactly, for both the explicit and implicit Euler methods. Do that, and show that \tilde{x}_N converges to the exact solution e^{aT} when $N \rightarrow \infty$.

By using higher order approximations of the derivative, one can obtain methods that converge faster. *Heun's method* is an explicit method:

$$\tilde{x}_{i+1} = \tilde{x}_i + \frac{h}{2} \left(f(\tilde{x}_i) + f(\tilde{x}_i + hf(\tilde{x}_i)) \right). \quad (5)$$

Here the derivative is computed as the average of the $f(x)$ evaluated at the point \tilde{x}_i and the $f(x)$ evaluated at the point $\tilde{x}_i + hf(\tilde{x}_i)$, which is the next point as computed with the explicit Euler method.

In Matlab there is a whole family of methods for solving systems of ode. In many cases, the best choice is `ode45`, which is a Runge-Kutta method of order 4, that automatically finds the optimal step length for for obtaining the requested accuracy. The manual pages of Matlab give a very good introduction to these methods, and there are several useful examples on how to use them.

ASSIGNMENT 1.2: Consider the pendulum equation

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L} \sin(\theta) \quad (6)$$

where g is the gravitational acceleration, and L is the length of the pendulum. For this exercise, you may assume that $g/L = 1$. Rewrite this as a first order system, and solve it numerically using the the implicit and explicit Euler methods and Heun's method for various values of the step size h . If the error decreases as a power of h , that is, the error is proportional to h^m for some $m > 0$, one says that the method has order m . Try to determine the rate of convergence for the different methods. The time interval should be so large that a several periods of the oscillation can be seen. Compare the solution to the one obtained by using Matlab's `ode45`-method.

For systems of ode:s it is sometimes useful to use explicit methods for some of the variables and implicit methods for other variables. In a system of two equations,

$$\begin{aligned}\dot{x} &= f(x, y) \\ \dot{y} &= g(x, y)\end{aligned}\tag{7}$$

the SYMPLECTIC EULER METHOD is

$$\begin{aligned}\tilde{x}_{i+1} &= \tilde{x}_i + hf(\tilde{x}_i, \tilde{y}_{i+1}) \\ \tilde{y}_{i+1} &= \tilde{y}_i + hg(\tilde{x}_i, \tilde{y}_i).\end{aligned}\tag{8}$$

Symplectic methods are particularly suited for *Hamiltonian* systems. The methods are constructed to conserve certain quantities that are preserved by the system, e.g. the energy.

ASSIGNMENT 1.3: Prove that the total energy¹

$$H = \frac{(\dot{\theta})^2}{2} - \frac{g}{L} \cos \theta\tag{9}$$

is constant for solutions to the system (6). Check how accurately the numerical approximations conserve the energy. For the symplectic method, check if it matters which of θ and $\dot{\theta}$ that is set to x and y in (8).

Assignment 2: Phase portraits

Consider the equation

$$\dot{x} = ax + by\tag{10}$$

$$\dot{y} = cx + dy.\tag{11}$$

The only fixed, or stationary point, to this equation is $x = y = 0$. Whether this point is stable or not depends on the values of a , b , c , and d . Figure 1 shows the phase plane for some different values of the parameters. The vector field is indicated by blue arrows, and some trajectories for different starting points are given in red. In figure (1a), all trajectories are repelled from the origin and approach infinity as t goes to infinity, in (1b), which is then a stable fixed point, and all of \mathbb{R}^2 is the domain of attraction for the origin; in (1c) the eigenvalues of A are purely imaginary, and the trajectories remain on a closed curve around the origin; in (1d), finally, there are two real eigenvalues with opposite signs, $\lambda_1 > 0$ and $\lambda_2 < 0$ with the corresponding eigenvectors \mathbf{u}_1 and \mathbf{u}_2 . In the latter case, trajectory starting on the line $\mathbb{R}\mathbf{u}_1$ will remain on that line, and move away from the origin as t increases. $\mathbb{R}\mathbf{u}_1$ is called the *unstable manifold*. Similarly the line $\mathbb{R}\mathbf{u}_2$ is called the *stable manifold*. These are drawn in green in figure (d).

ASSIGNMENT 2.1: Consider the equation

$$\begin{cases} \dot{x} = x^2 + y^2 - 1 \\ \dot{y} = x - y \end{cases}\tag{12}$$

Make a phase plane analysis of this equation as in the linear example above (this is not linear, so exact calculations are more difficult, if not impossible):

- Determine the fixed points, and their quality (stable, unstable, ...).
- Sketch the vector field, e.g. using the matlab command `quiver`
- Plot a number of trajectories with different starting points.

¹Well, this is not exactly the total energy: assuming that it is an idealized pendulum with a mass m concentrated at the endpoint of the otherwise weightless pendulum, you should multiply by mL^2 . It is also always possible to add a constant.

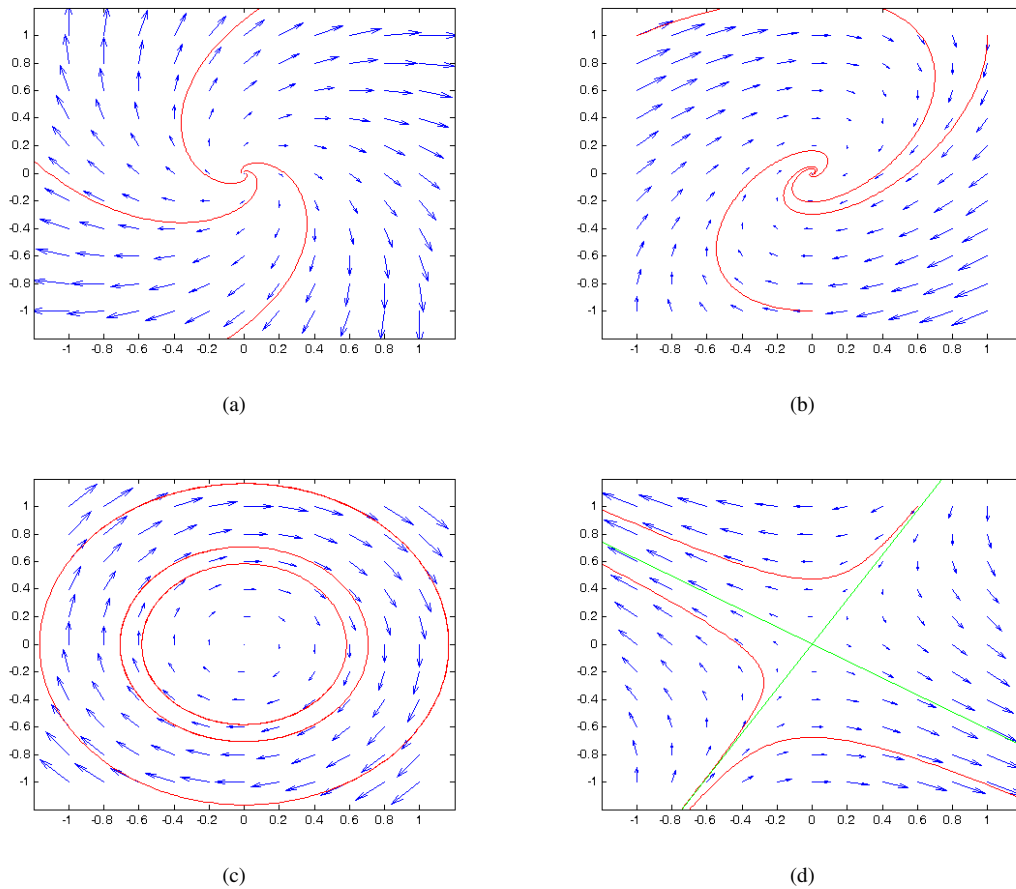


Figure 1: Phase portraits for the linear system $\dot{x} = Ax$. In (a), the origin is an unstable fixed point, in (b) it is stable, in (c) it marginally stable, and in (d) there is one stable and one unstable direction

- In case a fixed point is stable, determine the domain of attraction as accurately as possible (numerically), and give it a distinct color in the phase plane.
- In case a fixed point is similar to (d) above, try to compute the stable and unstable manifolds as accurately as possible (numerically), and plot them using green color.

THE RÖSSLER SYSTEM: This is a famous system of differential equations, which has been thoroughly studied since the first paper by Rössler appeared in 1977:

$$\begin{cases} \dot{x} &= -y - z \\ \dot{y} &= x + ay \\ \dot{z} &= b + z(x - c) \end{cases} \quad (13)$$

Depending on the values of the real parameters a , b and c the solutions may behave very differently: for some values there are periodic solutions, and for some values the solutions are chaotic.

Three dimensional graphs of trajectories may be produced using matlab, but it is equally useful to plot only the x and y components of the trajectories.

ASSIGNMENT 2.2:

- Let $a = b = 0.1$ and plot projections of the trajectories in the $x - y$ -plane for different values of c . In particular, try the values $c = 4, 6, 8.5, 9, 12, 12.6, 18$ (these are standard values to try).
- Check what happens with different starting points. Explain. Are there invariant sets? Are they attracting or repelling?

Parameter values where the solutions change character are called *bifurcation points*. In the case of the Rössler system (and with many other examples) a method that has been used to systematically study bifurcations is the following:

- Fix two of the parameter values, and let the third one vary.
- For a given value of the third parameter (c , for example)
 - Compute a trajectory for such a long time that the attractor has been reached (within reasonable accuracy)
 - The trajectory will continue in a loop, periodically or non periodically. For each loop, make a record of the largest value of $x(t)$, call it \bar{x}_n . Run the system sufficiently long time to get a large number of points \bar{x}_n
 - Plot the point (c, \bar{x}_n) .
- Change the value of the parameter in small steps and repeat the procedure, still plotting in the same graph.

ASSIGNMENT 2.1:

Carry out this procedure using $a = 0.2, c = 5.7$ and varying b in small steps in the interval $0 < b \leq 2$.

Assignment 3: Discrete dynamical systems

If $f : M \rightarrow M$ is a (differentiable) map, we may define a discrete dynamical system by iterations of f . We define

$$f^k(x) = \underbrace{f(f(\dots f(x)\dots))}_{k \text{ times}}$$

Given a starting point x , we may now generate a sequence of points,

$$x_k = f^k(x).$$

This sequence is called the *orbit* of x . It may consist of just one point, x itself, which then a fixed point of the map (i.e. $f(x) = x$), or a larger number of points. If there is $k > 1$ such that $f^k(x) = x$, we say that x is a periodic point, and if k is the smallest integer such that $f^k(x) = x$, the period is k . The orbit may also contain infinitely many different points in M . Chapter 11 in the book by Teschl treats maps of this kind thoroughly.

In this assignment we first consider a family of maps on the real line called “the quadratic family”: For $\mu > 0$ we set

$$f_\mu(x) = \mu x(1 - x)$$

You should convince yourself that if $0 \leq \mu \leq 4$, f_μ maps the interval $[0, 1]$ to itself,

$$f_\mu : [0, 1] \rightarrow [0, 1],$$

but that if $\mu > 4$ there are points $x \in [0, 1]$ such that $f_\mu(x) \notin [0, 1]$. You should also check that if you plot the graphs $y = f_\mu(x)$ and $y = x$ ($0 \leq x \leq 1$), the fixed points of f_μ can be found as the values of x where the two graphs intersect.

ASSIGNMENT 3.1:

- Plot the graphs of $f_\mu^k(x)$ for $k = 1, 2, 3, 4$ and for different values of μ . Use the results to draw conclusions about how the number of fixed points and periodic points depend on μ .
- Try to determine which of the fixed (periodic) points that are stable and which ones are unstable.
- Make a bifurcation diagram similar to the one for the Rössler system: for a given μ compute $x_k = f_\mu^k(x)$ for $k = 1 \dots k_{max}$ (take a large value for k_{max}), and plot the points (μ, x_k) in a $\mu - x$ -graph. You will need to skip a rather large number of points in the beginning of the sequence to get a good result (why). Repeat this for a large number of μ in the interval $0 < \mu \leq 4$.
- Discuss the result in relation to the graphs of $f^k(x)$.

One way to estimate the number of points in the orbit of x is the following:

- Compute $x_k, k = 1, 2, 3, \dots$. If you are only interested in the asymptotic behavior, you must then throw away a (large) number of points in the beginning of the sequence.
- Using the matlab function `hist`, you can make a histogram of the points, *i.e.* you can split the interval $[0, 1]$ (or any other interval) into sub-intervals (bins) $0 < X_1 < X_2, \dots, < X_N = 1$, and the function `hist` can tell you how many points x_k that belongs to each of the sub intervals. By checking how many sub-intervals that are empty, you can get an estimate of the number of different points that belong to the orbit of x .
- You have to experiment a little with the size of the sub intervals to get good results. Read the matlab manual to see more.

ASSIGNMENT 3.2:

Carry out this procedure. Different values of μ will give different results, as we have seen. You should be able to see that for certain values of μ there is only one bin that is not empty (this corresponds to one fixed point), for other values of μ there are two non-empty bins (a periodic point with period 2). How many different periods can you identify in this way? An interesting theorem by Sarkovskii (Theorem 11.2 in Teschl's book) says that if there is an orbit with period three then there are orbits of all orders. Can you find orbits of order three for in the way described above?

If $A \subset [0, 1]$ is a subset of the interval $[0, 1]$, we define

$$f(A) = \bigcup_{x \in A} f(x) = \{y : y = f(x) \text{ for some } x \in A\}$$

In the same way we define $f^k(A)$. By $f^{-1}(A)$ we mean

$$f^{-1}(A) = \{x : f(x) \in A\}$$

In general we know that the set $f(\{x\})$, the image of the set consisting only of the point x , consists of only one point, $f(x)$. But it may very well happen that $f^{-1}(\{x\})$ is a set consisting of many points.

The *measure* of an interval $[a, b] \subset \mathbb{R}$ is defined as the length of the interval:

$$m([a, b]) = b - a$$

(there are many other possibilities, but this is the most familiar one). If a set $A \subset \mathbb{R}$ is the union of finitely many disjoint intervals, $A = I_1 \cup I_2 \cup \dots \cup I_n$, then we define

$$m(A) = m(I_1) + m(I_2) + \dots + m(I_n),$$

the sum of lengths of the intervals. Note that $m([a, b]) = m(]a, b[)$, the measure of an open interval and the measure of its closure are equal. This is because the length of an "interval" consisting of only one point is zero. The measure of more complicated sets can be defined in a certain way by approximating with

finite unions of intervals. A possible numerical method to estimate the measure of a set is by *box counting*: suppose that $A \subset [0, 1]$, and divide $[0, 1]$ in M intervals, $[0, 1/M] \cup [1/M, 2/M] \cup \dots \cup [1 - 1/M, 1] = I_1 \cup \dots \cup I_M$. Check how many of the sub-intervals I_j that are needed to cover the set A , i.e. count the number of intervals such that $I_j \cap A \neq \emptyset$. If that number is N , an estimate of $m(A)$ is N/M . If A is a complicated set, this may converge very slowly (or not at all ...).

However, there are sets for which it does not make sense to speak about their measure. These are the “non measurable sets”, and it is not a trivial matter to construct such sets.

Here we are interested in the following problem: what is the relation between $m(A)$ and $m(f(A))$, and $m(f^{-1}(A))$.

We say that a map f is *measure preserving* if for all sets $A \subset M$

$$m(f^{-1}(A)) = m(A)$$

Our f_μ is clearly not measure preserving: The experiments carried out so far do not seem to depend very much on the starting point, so for example, for $1 < \mu < 3$ it seems that no matter where you start, $f^k(x) \rightarrow x_\infty$, all orbits converge to the same point (you should try to prove this). Does this mean that $m(f^k([] - 1, 1]) \rightarrow 0$ when $k \rightarrow \infty$? The next part of the assignment concerns this question.

ASSIGNMENT 3.2:

- For the quadratic map one can carry out many calculations explicitly. Try to compute $m(f([a, b]))$, where $0 \leq a < b \leq 1$. Then do the same for $m(f^{-1}([a, b] \cap f([0, 1])))$. Then, finally, do the same with the interval $]a, b[$ replaced by a union of two disjoint intervals, $[a, b] \cup [c, d]$, where $0 \leq a < b < c < d \leq 1$.
- Let $a = 0.0, b = 0.3, c = 0.6, d = 1.0$, and let $A = f_\mu([a, b] \cup [c, d])$. You could use a *Monte Carlo* method to estimate the measure of A as follows: take a large set of point x_1, \dots, x_n in $[a, b] \cup [c, d]$, and compute $y_j = f_\mu(x_j)$. Then, as with the box counting method described above, count the number of intervals I_j needed to cover the set $\bigcup_{j=1}^n \{y_j\}$. You can do this most easily with the matlab command `hist`. Try this method for a few different values of μ and for different values of M , but always for a large number of points x_j . Compare with the exact result.
- Discuss whether this is a good method, and, if you can, propose a better one. How does the result depend on M and on the number of points n ? Why doesn't n appear in the estimate?

If $\mu > 4$ there is an interval around 0.5 such that for all points x in this interval, $f(x) > 1$, and another set of points x such that $f_\mu^2(x)$ fall outside the interval. The following exercise amounts to finding a set A such that $f^k(x) \in [0, 1]$ for all k and all $x \in A$. Of course, we cannot try all x , but a large number of them, and of course we cannot compute $f_\mu^k(x)$ for all values of k , but a rather large value can be done.

ASSIGNMENT 3.3:

- Describe why the set $\{x \in [0, 1] : f^k(x) \in [0, 1] \text{ for all } k\}$ is invariant.
- Take a large number of points $x_j \in [0, 1]$, and compute $f_\mu^k(x_j)$ for some large value of k . Then let $A = \bigcup \{x_j \text{ such that } f_\mu^k(x_j) \in [0, 1]\}$. Use the box counting argument to estimate $m(A)$.
- The box counting argument can be expressed as: how many intervals N_ε of length $\varepsilon = 1/M$ are needed to cover a set A , and if A is an interval $I \subset \mathbb{R}$, $N_\varepsilon \sim \varepsilon^{-1}$, or $\frac{\log N_\varepsilon}{-\log \varepsilon} \sim 1$. If you use the same procedure to estimate the measure of a set $A \subset \mathbb{R}^2$ (for example, a disk), in two dimensions, you will get $\lim_{\varepsilon \rightarrow 0} \frac{\log N_\varepsilon}{-\log \varepsilon} = 2$, where N_ε is the number of squares with side length ε needed to cover A . On the other hand, if you consider a curve segment in \mathbb{R}^2 , for example the circular arc with center at $(0, 0)$ starting at $(1, 0)$ and ending at $(0, 1)$, and count the number of squares needed to cover this segment in exactly the same way, you will find $\lim_{\varepsilon \rightarrow 0} \frac{\log N_\varepsilon}{-\log \varepsilon} = 1$. Show that these statements are

correct, and think of why the limit $\lim_{\varepsilon \rightarrow 0} \frac{\log N_\varepsilon}{-\log \varepsilon}$ could be considered as the dimension of the set A (it is known as the *box-counting dimension*).

- Use a Monte Carlo simulation to estimate the box-counting dimension of the set $\{x \in [0, 1] : f^k(x) \in [0, 1] \text{ for all } k\}$. How accurate do you think your result is?