MATH2200 NonLinear Systems of ODEs
A first look
G. Keady

1 NONLINEAR SYSTEMS OF DEs, and EQUILIBRIA
EIGENANALYSIS OF STABILITY OF EQUILIBRIA

1.1 Introduction
Many of our previous lectures - Eigenvalues and MatrixExp, normal modes - have concerned linear
d.e. problems. However, it is easy (with mathematica) to obtain a numerical solution of nonlinear
d.e.s too. Let’s treat an initial-value problem, written as a system:
\[
\frac{dy}{dt} = f(t,y), \quad y(0) = y_0 \text{ given. (1)}
\]

There are couple of aspects to the numerical solution.

• At the most practical level, there is ‘how do we solve d.e.s numerically’. Answer: use ap-
propriate software, e.g. the \texttt{NDSolve} commands in Mathematica. The first part of these
notes will describe an example (the nonlinear rigid-body pendulum with damping) treated
with Mathematica. The code is up at the unit’s web pages.

• Next there is the question of how does it work. The simplest method to describe is Euler’s
method. There is a treatment of Euler’s method in many first year level ‘Calculus’ texts.
Admittedly, this is just for a single nonlinear equation, but it works the same with systems,
i.e. vectors of unknowns.
There are various other methods too. There is an extended account of various methods
in Chapter 5 of Hassani’s \textit{Math Methods using Mathematica} book. I will let you read these
items, and won’t dwell on them now.

1.2 Autonomous equations
Just because a system of d.e.s is nonlinear (and impossible to solve analytically) doesn’t mean that
we give up all hope of analysing its behaviour. Numerical solution is easy enough, as is plotting
out solutions. However it is still worth being able to talk about what the solutions are like. If
your engineering task requires you to solve d.e.s numerically, chances are you will have to explain
something about the solutions to others. One of the standard things to explain is the ‘stability (or
otherwise) of the equilibria’.

A system of d.e.s is said to be \textit{autonomous} if it can be written
\[
\frac{dy}{dt} = F(y), \quad \text{(2)}
\]
i.e. the \( f(t,y) \) of equation (1) doesn’t have any explicit \( t \) dependence.

The *equilibria*, also called *equilibrium points*, are the values \( y_e \) at which \( F(y_e) = 0 \). Hence the constant function of time \( y(t) = y_e \) solves the d.e. (2).

In the last section of the notes we will look at how eigenvalues help us understand the stability of equilibria.

### 1.3 The nonlinear damped pendulum

The differential equation is

\[
\frac{d^2 \theta}{dt^2} + c \frac{d \theta}{dt} + \frac{g}{l} \sin(\theta) = 0 .
\]

I prefer to re-write equations or systems involving higher derivatives than first as systems of first order d.e.s. Define \( x_1 = \theta \) and \( x_2 = \frac{d \theta}{dt} \). Then the system is

\[
\begin{align*}
\frac{dx_1}{dt} &= x_2 , \\
\frac{dx_2}{dt} &= -cx_2 - \frac{g}{l} \sin(x_1) .
\end{align*}
\]

Mathematica:

```mathematica
(* FILE MathematicaNLStab/dampPendmma.txt *)
c = 0.1; (* damping constant *)
g = 9.8; (* acceleration due to gravity *)
m = 1; (* mass in kg *)
l = 0.5; (* length in m *)
(* Start the pendulum from rest, at theta= -0.5radians *)
theta0 = -0.5;
ans = NDSolve[{theta'[t]==w[t],m*w'[t]== -c*w[t]-(g/l)*Sin[theta[t]],
theta[0]== theta0, w[0]==0},{theta[t],w[t]},{t,0,20}];
theta0 = theta0;
ct= Plot[Evaluate[theta[t] /. ans],{t,0,20},
PlotLabel-> StringJoin["theta against time, theta(0)=",theta0,", theta'(0)=0"]]
(* looks like ordinary damped linear pendulum *)
```

The results are shown in Figure 1.

Next we start the pendulum at the same position and give it a large initial velocity. It spins over the top (it is a rigid body pendulum, a metal bar or something like that). Eventually it starts oscillating, and remember that, to us, \( \theta = 2n\pi \) for integer \( n \) all look the same: it is the position hanging downwards (and being still). The results are shown in Figure 2.
Figure 1: $\theta(0) = -0.5, \frac{d\theta}{dt} = 0$: not much different from a linear pendulum with small initial values.

Figure 2: $\theta(0) = -0.5, \frac{d\theta}{dt} = 10$: spins a few times over the top.
1.4 Various graphical ideas for 2nd order systems

Let’s take up the idea of representing the solutions by plotting curves in $(\theta, \dot{\theta})$-space. In our case this is in $(x_1, x_2)$-space.

Of course methods like this are natural only for systems of small order.

A lot can be done without actually solving the d.e., just as you learnt in first year when treating direction fields for single first order d.e.s. (Actually, by dividing the equation for $\dot{x}_2$ by that for $\dot{x}_1$, we actually get back to a first order d.e. for $x_2(x_1)$, so it is the same as you did in first year.)

Figure 3: Phase plot: the continuous curves are those from the numerical solution of the d.e.. The arrows show the direction field.

Figure 4: Phase plot: the continuous curve is just the one corresponding to the zero initial velocity.
1.5 Stability of equilibria

An equilibrium point is a solution of the d.e. which stays constant in time. So, for our pendulum example, we have, for integer \( n \), \( \theta = 2n\pi \), \( \dot{\theta} = 0 \) corresponding to the pendulum hanging downwards, and \( \theta = (2n + 1)\pi \), \( \dot{\theta} = 0 \) corresponding to the pendulum balanced precariously pointing upwards. It is physically obvious in this case which is stable and which is unstable, but we will push on and see how eigenvalues will enable the general situation to be analysed and then check that the eigenvalue analysis agrees with common sense for the pendulum problem.

**Stability in a one-dimensional example**

It may be worth going back to first year examples. The logistic equation

\[
\frac{dy}{dt} = y(1 - y)
\]

was used as an example (or a separable d.e.) then and is treated in your 1st year Calculus text. Its solution is given there at

\[
y(t) = \frac{1}{1 + (1/y(0) - 1)\exp(-t)}
\]

but, actually, we don’t need to solve it exactly to understand how its solutions will look. If \( 0 < y(0) < 1 \) the solution increases, and asymptotes to the line \( y = 1 \) as \( t \) tends to plus infinity. If \( y(0) > 1 \) the solutions decrease and again asymptote to the line \( y = 1 \) as \( t \) tends to plus infinity. See Figure 5.

![Figure 5: A couple of solutions to the logistic equation](image)

With our preceding definition of equilibrium points, we have that the two equilibrium points for this equation are \( y_0 = 0 \) and \( y_1 = 1 \). \( y_0 = 0 \) is unstable: if we start near it we move away. \( y_1 = 1 \) is stable: if we start near it we move even closer to it. The way we really see this is by ‘linearising about the equilibrium solution’. For small \( y \), i.e. near \( y_0 = 0 \), the d.e. is approximately \( \dot{y} = y \).
which has solution $c_0 \exp(\lambda_0 t)$ with $\lambda_0 = 1 > 0$, so solutions grow away from $y_0 = 0$. To linearise about $y_1$ set $y(t) = y_1 + \rho$ and substitute this into the logistic equation, and then neglect terms of order $\rho^2$: the d.e. becomes $\dot{\rho} = -\rho$ so the solutions $\rho(t) = c_1 \exp(\lambda_1 t)$ have $\lambda_1 = -1 < 0$.

**The general case - systems of n d.e.s**

Let's now move onto autonomous systems.

$$\dot{y} = F(y),$$

and suppose $y_e$ is an equilibrium point. Again linearise about the equilibrium point, setting $y = y_e + \rho$. Using a multivariate Taylor series expansion we have

$$F(y_e + \rho) \approx F(y_e) + DF(y_e)\rho + \text{negligibly small terms} .$$

Here $DF$ is a matrix

$$DF = \begin{bmatrix} \frac{\partial F_1}{\partial y_1} \\ \vdots \\ \frac{\partial F_n}{\partial y_n} \end{bmatrix} .$$

Thus the equation for the remainder $\rho$ is a system of constant coefficient linear equations. We know how to solve them using matrix exponentials. Here however, we don’t even need to solve it completely. An equilibrium point is stable if all the solutions tend to zero as $t$ tends to plus infinity and this is guaranteed if the real parts of all the eigenvalues of the matrix $DF(y_e)$ are negative. For the purposes of deciding on the stability of the equilibrium point we just need to look at these properties of its eigenvalues.

**Return to the pendulum example**

Sorry, class. I wrote $x$ in my Matlab code, but I’m using $y$ now in the analysis of the equilibrium points. So the $x$ from the previous bit on the pendulum is now equal to $y$. That is $y_1 = \theta$, and $y_2 = \dot{\theta}$.

Let’s look at the ‘hanging down’ equilibrium point. We linearise $y_1 = \rho_1$, $y_2 = \rho_2$ with both components of $\rho$ vector small.

$$F(y) = \begin{bmatrix} y_2 \\ -c y_2 - \frac{g}{l} \sin(y_1) \end{bmatrix} .$$

The jacobian matrix $DF$ is

$$DF = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} \cos(y_1) & -c \end{bmatrix} .$$

(Don’t worry if you get the transpose of this. We are only after the eigenvalues and the eigenvalues of $A^T$ are the same as the eigenvalues of $A$.) For our hanging-down equilibrium point the $DF$ becomes

$$DF_{\text{down}} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} & -c \end{bmatrix} .$$

The product of the eigenvalues is positive ($g/l$) and the sum is negative $-c$. Either we have a complex conjugate pair with real parts of both being $-c/2$ or we have two negative eigenvalues. In both cases, the real parts of every eigenvalue here are negative.

The situation changes if the rigid pendulum is stationary and upright $y_1 = \theta = \pi$.

$$DF_{\text{up}} = \begin{bmatrix} 0 & 1 \\ \frac{g}{l} & -c \end{bmatrix} .$$

Now the product of the eigenvalues is negative. It is an easy sum to find both eigenvalues. Both are real this time, and one is positive and one negative. The differential equation for the remainder
\(\rho\) now has components which grow. The upward pointing equilibrium is unstable, which accords with our commonsense.

There is less algebraic clutter in the case \(c = 0\), the undamped nonlinear pendulum. The vertically upright position has

\[
DF_{up,c=0} = \begin{bmatrix} 0 & 1 \\ \frac{q}{l} & 0 \end{bmatrix}
\]

Its eigenvalues are real, equal in magnitude and opposite in sign. This vertically upright equilibrium is unstable. For the vertically downward equilibrium with \(c = 0\), the eigenvalues are pure imaginary, one being the negative of the other (as the trace is 0), and this situation is sometimes called ‘neutrally stable’. It isn’t really stable as the departures from equilibrium don’t get smaller, but then again they don’t get larger either.

There are millions of other examples which could be done. Environmental engineering students should probably look at the competing-species d.e. example given below and at the predator-prey d.e. example in many texts. The same ‘eigenvalues of the jacobian matrix’ method can be used to investigate the stability of equilibria in these cases too.

What is the shortest summary of the main point, for now, from all this? **Eigenvalues are used in studies of stability of equilibrium points.**

### 1.6 Further comments about 2nd order autonomous systems

The general story for our 2nd-order autonomous d.e.s is

- Find the equilibrium points \(y_e\).
- For each \(y_e\)
  - Find the 2 by 2 jacobian matrix \(J(y_e)\), and its eigenvalues.
  - Classify the equilibrium point as stable or unstable.
  - Draw in the phase-plane a few trajectories near the equilibrium point.

The key idea for the last part is that we can build up a table of possibilities for \(A = J(y_e)\); and we did this in the treatment of stability for linear systems.

### 1.7 D.E.s describing competing species

Suppose we have two competing species (rabbits and sheep, for example, which are competing to eat the same pasture). Let \(y_1\) and \(y_2\) denote the numbers of each species. A model for the way the populations change is:

\[
\begin{align*}
\frac{dy_1}{dt} &= y_1(\beta_1 - d_1 y_1 - c_1 y_2) \\
\frac{dy_2}{dt} &= y_2(\beta_2 - d_2 y_2 - c_2 y_1)
\end{align*}
\]

If there were to be just one species present, the system reduces to a single equation, a logistic equation, describing the population that is present. We will refer to the quantities \(c_1, c_2\) as ‘interaction coefficients’.
1.7.1 The equilibrium points

There are at most four equilibrium solutions:

\[(0, 0), \quad \left(0, \frac{\beta_2}{d_2}\right), \quad \left(\frac{\beta_1}{d_1}, 0\right), \quad \left(\frac{c_1 \beta_2 - d_2 \beta_1}{c_1 c_2 - d_1 d_2}, \frac{c_2 \beta_1 - d_1 \beta_2}{c_1 c_2 - d_1 d_2}\right)\].

The fourth equilibrium point is relevant only if both components are positive. The middle two equilibrium points suggest the extinction of one species and the survival and stabilisation of the other.

We will investigate two methods of deciding on which equilibrium point is likely to be reached from given initial values.

- For autonomous d.e. problems involving just 2 variables, \(y_1, y_2\), ‘phase plane’ analysis is useful. This example is treated in many texts, e.g. Barnes and Fulford, ‘Mathematical Modelling with Case Studies’ §6.3
- The stability of the equilibrium points can be analyzed using the methods we have discussed above. We will treat this next.

1.7.2 The stability of the equilibrium points

The methods used here would also be applicable for more competing species (sheep, rabbits and kangaroos, for example). In the code given, we find the equilibrium points, then the jacobian, and from the jacobian evaluated at each equilibrium point determine its stability.
The fourth equilibrium point is relevant only if both components are positive. The middle two equilibrium points suggest the extinction of one species and the survival and stabilisation of the other.

\( J = \text{Outer}[D, \{f1, f2\}, \{y1, y2\}] \)

\( J1 = J /. \text{equilibSols}[[1]] \)
\( \text{Eigenvalues}[J1] \)

(* % beta1 and beta2 are the eigenvalues for the \( \{0,0\} \) euilib. Both eigenvalues are positive; \( \{0,0\} \) is unstable *)

\( J2 = J /. \text{equilibSols}[[2]] \)
\( \text{Eigenvalues}[J2] \)

\( J3 = J /. \text{equilibSols}[[3]] \)
\( \text{Eigenvalues}[J2] \)

\( J4 = J /. \text{equilibSols}[[4]] \)
\( \text{Eigenvalues}[J2] \)

(* % The equilib \( \{ beta1/d1, 0\} \) has eigenvalues
\( [ \begin{array}{c} -\beta 1 \\ (-\beta 1 c2+d1\beta 2)/d1 \end{array} ] \)
so this equilib is stable iff
\( \text{IntY1} = (\beta 1/d1 - \beta 2/c2) > 0 \)

% The equilib \( \{ 0, \beta 2/d2\} \) has eigenvalues
\( [ \begin{array}{c} -(\beta 2\times c1-d2\times \beta 1)/d2 \\ -\beta 2 \end{array} ] \)
This equilib is stable iff
\( \text{IntY2} = (\beta 2/d2 - \beta 1/c1) > 0 \)

% The equilib with both entries nonzero
After some analysis we find that
this equilib is stable iff
\( \text{IntY2} < 0 \) and \( \text{IntY1} < 0 \)
See Barnes and Fulford, p174
For autonomous d.e. problems involving just 2 variables, \( y_1 \), \( y_2 \), 'phase plane' analysis is useful.

1.7.3 Phase plots

Phase plots can be found in the usual way.

(* FILE /u/keady/WWW/Teaching/M2en/LecNotesLA/05EvalsGen/ CompetingSpeciesMma/competingSpeciesPhasemma.txt *)

(* phase plot for competing species
Example 1, Fig 9.4.1 of Boyce and DiPrima *)

beta1=1; d1=1; c1=1;
beta2=0.75; d2=1; c2=0.5;
f1 = Y1*(beta1-d1*Y1-c1*Y2);
f2 = Y2*(beta2-d2*Y2-c2*Y1);
<<Graphics'PlotField'

pvf=PlotVectorField[{f1,f2},{Y1,-0.1,1.3},{Y2,-0.1,1},
ScaleFunction->(1&), Frame -> True, AspectRatio ->1]

pequilib= ListPlot[{{0, 0}, {0, beta2/d2}, {beta1/d1, 0},
{(c1*beta2-d2*beta1)/(c1*c2-d1*d2),(c2*beta1-d1*beta2)/(c1*c2-d1*d2)}},
PlotStyle -> PointSize[0.04],

de1 = D[y1[t], t] == f1 /. {Y1 -> y1[t], Y2 -> y2[t]}
de2 = D[y2[t], t] == f2 /. {Y1 -> y1[t], Y2 -> y2[t]}
ans = NDSolve[{de1, de2, y1[0] == 0.2, y2[0] == 0.1},
{y1[t], y2[t]}, {t, 0, 10}]

pp=ParametricPlot[{y1[t],y2[t]}/. ans, {t,0,10},
PlotStyle-> RGBColor[1,0,0]]

Show[{pvf, pequilib, pp}, AspectRatio -> 1]

The results of running this are given in Figure 6.
Figure 6: The parameters are 1 except those given on the plot.
References


