EIGENVALUES OF GENERAL (SQUARE) MATRICES AND LINEAR SYSTEMS OF DEs

1 Revision, eigenvalues

In your 2nd year units (M235 or A213) you will have learnt the definition of eigenvalues, eigenvectors, and methods to calculate them. Also in these 2nd year units, you will have seen the definitions of similarity and of jordan forms. In M235, though, to a rather lesser extent in A213, you will have seen applications of these ideas to constant coefficient systems of d.e.s, and this chapter reviews this material.

1.1 Eigenvectors and systems \( y' = Ay \)

Consider \( \frac{dy}{dt} = Ay \), where \( A \) is a square matrix. If we look for solutions \( y = u \exp(\lambda t) \), on substituting into the d.e., we see that, to get a solution, we must have

\[ \lambda u = Au. \]

A nonzero vector \( u \) which satisfies this last equation is called an eigenvector. We will return to how best to use this sort of information to solve d.e.s later.

1.2 Diagonalization

[StrBC] §5.6, [StrWC] §6.2

PROBLEM A1. Given a linear operator \( T : V \to V \) on a finite dimensional vector space \( V \), does there exist a basis for \( V \) w.r.t. which the matrix of \( T \) is diagonal?

PROBLEM A2. Given a linear operator \( T : V \to V \) on a finite dimensional inner product space \( V \) (= a Euclidean space), does there exist an orthonormal basis for \( V \) w.r.t. which the matrix of \( T \) is diagonal? (Well, the definition of orthonormal is a bit later in these notes, but will be advertised in lectures here.)

Matrix forms of these questions.

PROBLEM M1. Given a square matrix \( A \), does there exist an invertible matrix \( P \) such that \( P^{-1}AP \) is diagonal?

PROBLEM M2. Given a square matrix \( A \), does there exist an orthogonal matrix \( Q \) such that \( QT AQ \) is diagonal. (Orthogonal means \( QT = Q^{-1} \).)

In these notes we concentrate of the matrix problems M1 and M2 rather than the abstract problems A1 and A2. We will consider Problem 2 later. Here, however, is a preview - the answer to Problem M2.

Let \( A \) be an \( n \) by \( n \) real matrix. Then \( A \) is orthogonally similar to a diagonal matrix (i.e. \( QT AQ \) is diagonal) if and only if \( A \) is symmetric.

Definition. Matrix \( A \) is said to be diagonable, or diagonalizable, if there is an invertible matrix \( P \) such that \( P^{-1}AP \) is diagonal).
Answer to Problem M1.

**THEOREM.** A is diagonalizable if and only if there exists a basis of \( \mathbb{C}^n \) in which every vector is an eigenvector of \( A \), if and only if for any eigenvalue \( \lambda \) of \( A \), the algebraic and geometric multiplicities of \( \lambda \) are equal.

If the answer to Problem M1 is that for a particular \( A \) there is an invertible matrix \( P \) such that \( P^{-1}AP \) is diagonal this leads to further questions.

How is \( P \) found?

What can be said about the diagonal matrix \( D = P^{-1}AP \)?

**THEOREM.** If \( A \) is similar to the diagonal matrix \( D = \text{diag}(\alpha_1, \ldots, \alpha_n) \), then \( \alpha_1, \ldots, \alpha_n \) are the eigenvalues of \( A \) (with their algebraic multiplicities).

**THEOREM.** Suppose that \( A \) is diagonalizable, so that by earlier results, there is a basis in which every vector is an eigenvector of \( A \). Let \( (X_1, \ldots, X_n) \) be any such basis, with \( AX_k = \alpha_k X_k \), and let

\[
P = [X_1; \ldots; X_n].
\]

Then \( P \) is nonsingular and \( P^{-1}AP = \text{diag}(\alpha_1, \ldots, \alpha_n) \).

For a proof, see [StrBC] Thm 5C.

**THEOREM.** ([StrBC] Thm 5D.) Let \( A \) be an \( n \) by \( n \) matrix. Then eigenvectors corresponding to distinct eigenvalues are linearly independent.

**COROLLARY.** ([StrBC] bottom p256.) An \( n \) by \( n \) matrix with \( n \) distinct eigenvalues is diagonalizable.

If a matrix is diagonalizable, this fact can be useful in calculations. For example, if \( A \) is diagonalizable, \( D = P^{-1}AP \), then any power of \( A \) is easily found, even very large powers, as \( A^k = PD^kP^{-1} \) and \( D^k \) is easily found. ([StrBC] p258.)

Another interesting fact is ([StrBC] Thm 5F.) If \( A \) and \( B \) are diagonalizable, they share the same eigenvector matrix \( P \) if and only if \( AB = BA \).

Although Matlab’s \texttt{eig} function is good for finding float numeric approximations to eigenvalues, it isn’t easy, in general, to quickly look at a matrix and say anything useful about its eigenvalues. However, there are some special cases when something can be said. Here is an example. ([StrBC] Thm 5K p271.) Let \( A \) be a square matrix, all of whose entries are positive numbers. Then, the largest eigenvalue \( \lambda_1 \) of \( A \) is positive, and so are all the components of an eigenvector corresponding to \( \lambda_1 \).

1.3 More about similarity: jordan form

(It is Camille Jordan, [ON] p1088, a different Jordan than the one with the Gauss-Jordan decomposition, ref.)

Not all matrices are diagonalizable, e.g. \[
\begin{bmatrix}
\lambda & 1 \\
0 & \lambda
\end{bmatrix}.
\]

This leads to the following question. What simple form can any \( n \) by \( n \) matrix be reduced to by similarity? I.e., given \( A \), find a simple form for \( B = P^{-1}AP \)?

Answer (but the proof is too advanced to include here):
JORDAN FORM THEOREM. ([StrBC] Thm 5U, [StrWC] Thm 6R.) An $n \times n$ matrix with eigenvalues, repeated with their algebraic multiplicities ($\lambda_1, \ldots, \lambda_n$) is similar to a matrix with these eigenvalues in the diagonal positions, zeros and ones on the superdiagonal, and zeros elsewhere. (The proof is long and definitely not examinable in this unit. Details are in [StrBC] Appendix B.)

EXAMPLE #different eigenvalues #(lin indep) eigenvectors

\[
\begin{bmatrix}
\lambda & 0 & 0 \\
0 & \mu & 0 \\
0 & 0 & \nu
\end{bmatrix}
\]

3 3

\[
\begin{bmatrix}
\lambda & 0 & 0 \\
0 & \lambda & 0 \\
0 & 0 & \mu
\end{bmatrix}
\]

2 3

\[
\begin{bmatrix}
\lambda & 1 & 0 \\
0 & \lambda & 0 \\
0 & 0 & \mu
\end{bmatrix}
\]

2 2

\[
\begin{bmatrix}
\lambda & 0 & 0 \\
0 & \lambda & 0 \\
0 & 0 & \lambda
\end{bmatrix}
\]

1 3

\[
\begin{bmatrix}
\lambda & 1 & 0 \\
0 & \lambda & 0 \\
0 & 0 & \lambda
\end{bmatrix}
\]

1 2

\[
\begin{bmatrix}
\lambda & 1 & 0 \\
0 & \lambda & 1 \\
0 & 0 & \lambda
\end{bmatrix}
\]

1 1

We will see, in a later section of this set of notes, a USE of jordan forms in connection with solving constant coefficient systems of d.e.s - matrix exponentials.

Here are a couple of examples of the Student Matlab one-liner used to find a jordan form. First here is the example we had earlier of a defective matrix: Next, a larger example:

There is an important caution concerning jordan forms, namely that in Float computation, the process of forming them is very badly conditioned. For this reason, Matlab’s jordan function requires a symbolic matrix (i.e. with exact entries).

EXAMPLE. Let $A_\epsilon = \begin{bmatrix} \epsilon & 1 \\ 0 & 0 \end{bmatrix}$. When $\epsilon = 0$, i.e. $A_0$, is already in jordan form. However, for any $\epsilon \neq 0$ it, $A_\epsilon$, is not in jordan form, and $A_\epsilon$ is diagonable. Verify this last statement.

Solution. We have

\[
\begin{bmatrix}
\epsilon & 1 \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
1 \\
0
\end{bmatrix} = \epsilon
\begin{bmatrix}
1 \\
0
\end{bmatrix},
\begin{bmatrix}
\epsilon & 1 \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
1 \\
-\epsilon
\end{bmatrix} =
\begin{bmatrix}
0 \\
0
\end{bmatrix}.
\]

In particular, when $\epsilon$ is small but nonzero we have ‘nearly parallel’ eigenvectors. For the diagonalizing matrix $P_\epsilon$ we have

\[
P_\epsilon^{-1} = \frac{1}{\epsilon}
\begin{bmatrix}
\epsilon & 1 \\
0 & -1
\end{bmatrix},
P_\epsilon =
\begin{bmatrix}
1 & 1 \\
0 & -\epsilon
\end{bmatrix}.
\]
2 SYSTEMS OF LINEAR D.E.S: Intro to matrix exponentials, etc.

References:  [StrBC] §5.4, [StrWC] §6.3

2.1 Example: compartment models involving solute and mixing

There are huge numbers of examples of linear systems of d.e.s, and you will encounter many in your engineering units. Since we are about to treat constant coefficient systems it may be appropriate to begin with an easy-to-derive example of one that arises in connection with mixing of solutes in a system of interconnected tanks. (It is possible to imagine ‘environmental engineering/environmental science’ applications of a related kind, e.g. concerning salinity levels in different parts of the Swan/Canning river/estuary system, treated as a function of time with given inputs via rainfall, streamflow, and mixing rates between different parts. However, we will treat merely an idealized invented system.)

2.1.1 Two tank example: formulating the d.e.s

Consider two interconnected tanks containing water and a solute. Both tanks are kept well mixed.

- Initially, tank 1 contains $v_1$ litres of water and $q_1(0)$ kg of solute, and tank 2 contains $v_2$ litres of water and $q_2(0)$ kg of solute.

- Water containing $g_1$ kg of solute per litre flows into tank 1 at a rate of $s_1$ litres per minute. The mixture in tank 1 flows out to tank 2 at a rate of $r_{12}$ litres per minute. The total volume in tank 1 remains constant, with an overflow of $d_1$ litres per minute going into a drain.

  - Water containing $g_2$ kg of solute per litre flows into tank 2 at a rate of $s_2$ litres per minute. The mixture in tank 2 flows out to tank 1 at a rate of $r_{21}$ litres per minute. The total volume in tank 2 remains constant, with an overflow of $d_2$ litres per minute going into a drain.

Here is a diagram of the situation.

Let $q_i(t)$ denote the quantity of solute in tank $i$ at time $t$. A simple mass balance gives that $q_1(t)$ and $q_2(t)$ satisfy the following system of d.e.s:

$$
d_1 = s_1 - r_{12} + r_{21}
$$

$$
d_2 = s_2 + r_{12} - r_{21}
$$
\[
\frac{dq_1}{dt} = s_1 g_1 - \frac{(r_{12} + d_1)q_1}{v_1} + \frac{r_{21}q_2}{v_2} \\
\frac{dq_2}{dt} = s_2 g_2 + \frac{r_{12}q_1}{v_1} - \frac{(r_{21} + d_2)q_2}{v_2}
\]

It is a bit messy keeping all these symbols general, so let’s consider a couple of numerical examples

- the first example being from the (1st year level) book by Fulford et al. *Modelling with differential and difference equations*
- the second example as in the CODEE *ODE Architect* book published by Wiley (Exploration 6.3, p109), leaving \(g_1, g_2\) general.

**Fulford example, p356**

Assume that initially the two tanks each contain 100 litres of pure water. Pure dye is then pumped into the first tank at a fixed rate of 1 litre/minute, while pure water is pumped into the second tank at the same rate. Pumps exchange the mixtures between the two tanks, – at a rate of 4 litres/minute from tank 1 to tank 2, and at a rate of 3 litres/minute from tank 2 to tank 1. The diluted mixture is drawn off from tank 2 at a rate of 2 litres/minute. Derive the differential equations for the amount of dye in each tank.

\[
v_1 = 100, \quad q_1(0) = 0 \\
v_2 = 100, \quad q_1(0) = 0 \\
s_1 = 1, \quad g_1 = 1, \quad r_{12} = 4, \quad d_1 = 0 \\
s_2 = 1, \quad g_2 = 1, \quad r_{21} = 3, \quad d_2 = 2
\]

This leads to the system of d.e.s

\[
\frac{dq_1}{dt} = 1 - \frac{4}{100}q_1 + \frac{3}{100}q_2 \\
\frac{dq_2}{dt} = 0 + \frac{4}{100}q_1 - \frac{5}{100}q_2
\]

This can be written as a system

\[
\frac{dq}{dt} = b + Aq, \quad b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad A = \frac{1}{100} \begin{bmatrix} -4 & 3 \\ 4 & -5 \end{bmatrix}
\]

The mixing problem we have been given is an ‘initial-value problem’: the system of d.e.s is to be solved for \(q(t)\), with \(q(0)\) given.

**CODEE example, p109**

In this example,

\[
v_1 = 5, \quad q_1(0) = 3 \\
v_2 = 4, \quad q_1(0) = 5 \\
s_1 = 2, \quad r_{12} = 3, \quad d_1 = 3 \\
s_2 = 3, \quad r_{21} = 4, \quad d_2 = 2
\]
This leads to the system of d.e.s

\[
\frac{dq_1}{dt} = 2g_1 - \frac{6}{5}q_1 + \frac{4}{5}q_2 \\
\frac{dq_2}{dt} = 3g_2 + \frac{3}{5}q_1 - \frac{6}{5}q_2
\]

This can be written as a system

\[
\frac{dq}{dt} = b + Aq, \quad b = \begin{bmatrix} 2g_1 \\
3g_2 \end{bmatrix}, \quad A = \begin{bmatrix} -\frac{6}{5} & 1 \\
\frac{3}{5} & -\frac{3}{2} \end{bmatrix}
\]

Again, the mixing problem in this example is an 'initial-value problem': the system of d.e.s is to be solved for \(q(t)\), with \(q(0)\) given.

**The general case**

It is no more difficult to set up the d.e.s for the general case. This can be written as a system

\[
\frac{dq}{dt} = b + Aq, \quad b = \begin{bmatrix} s_1g_1 \\
s_2g_2 \end{bmatrix}, \quad A = \begin{bmatrix} -\frac{s_1+r_1}{v_1} + \frac{r_2}{v_1} & \frac{r_2}{v_2} \\
\frac{s_1+r_1}{v_1} & -\frac{s_2+r_2}{v_2} \end{bmatrix}
\]

### 2.1.2 Two tank example: discussing solving the d.e.s

The approach to the problem is much as with your first year linear d.e. work. There is an easily spotted particular solution, an equilibrium solution where \(q(t)\) stays constant, at \(q_e\) where

\[Aq_e = -b\]

(Note that, in our numerical examples, \(\det(A) \neq 0\) so there is a unique equilibrium solution.) We will see that the easiest way to write the solution is

\[q(t) = q_e + \expm(tA)(q(0) - q_e)\]

Here \(\expm(At)\) is called the ‘exponential-matrix’ of \(At\).

The numerical versions of the examples here do have one feature which simplifies their solution: for them the matrices \(A\) are diagonalizable. Let \(u_j\) denote an eigenvector corresponding to eigenvalue \(\lambda_j\). From the discussion at the beginning of this chapter on eigenvalues we know that the general solution of the d.e. problem is

\[q(t) = q_e + c_1u_1\exp(\lambda_1t) + c_2u_2\exp(\lambda_2t)\]

for constants \(c_1, c_2\). The initial-value problem is solved by finding the \(c_1, c_2\) so that the initial conditions are satisfied. There is a lot of tedious, but routine, calculation in this (and it gets even worse with matrices which are not diagonalizable). Because initial-value problems occur a lot, there is value in tidy formula like that involving the exponential matrix above.

**Fulford example**

In the Fulford example, \(A\) has eigenvalues of \(-1/100\) and \(-8/100\). The formula for the exponential matrix is just given here (use Matlab’s \texttt{expm} if you want to check, but, at this stage I haven’t told you how to calculate \(\expm(tA)\)), and, to save writing, let \(e_1 = \exp(-t/100), e_2 = \exp(-8t/100)\):

\[
\expm(tA) = \frac{1}{7} \begin{bmatrix} 4e_1 + 3e_2 & 3e_1 - 3e_2 \\
4e_1 - 4e_2 & 3e_1 + 4e_2 \end{bmatrix}
\]
Note that this tends to zero as $t$ tends to plus infinity, so that any solution – no matter what the starting values are – tends to the equilibrium solution. With this formula for $\expm(tA)$ and

$$q(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad q_e = \begin{bmatrix} 125/2 \\ 50 \end{bmatrix},$$

we can easily plot out the solutions and the plot is shown in Figure 1.

Figure 1: The quantity of dye in each tank in the Fulford example. The plot is the same as in the Fulford et al. book, p365.

**CODEE example**

In the CODEE example, its $A$ has pretty messy eigenvalues, and as a consequence the formula for the exponential matrix is a mess. This is pretty typical of real problems, I’m afraid. Actually, at least it is possible to find the eigenvalues. Doing this numerically is often enough, and, in particular, with this problem it is easily seen that both eigenvalues are negative, and this has the implication that as $t$ tends to plus infinity, the $\expm(tA)$ term tends to zero, and so any solution tends to $q_e$.

At this stage you can ignore the details of this (and, it is recommended that you don’t, on return to this messy problem, work it out by hand, as matlab is a better tool if you must look at the details). I find that

$$\expm(tA) = \exp(-27t/20) \begin{bmatrix} c - \sqrt{249}/83 s & 20 \sqrt{249}/83 s \\ 4\sqrt{249}/83 s & c + \sqrt{249}/83 s \end{bmatrix},$$

where $c = \cosh(t\sqrt{249}/20)$ and $s = \sinh(t\sqrt{249}/20)$. 
In exercise 2, page 109 of the CODEE book, they ask people to plot the solutions when also \( g_1 = 1 \) and \( g_2 = 1 \). The CODEE code is just numerical solution, effectively equivalent to matlab’s \texttt{ode45} \texttt{ode23} \texttt{ode113} \texttt{ode23s} \texttt{ode15s} , and then it becomes an exploration to see things, easy for us with the formula, that the equilibrium that is reached is independent of the initial values \( q(0) \). In any event, the results with the initial values as given above are shown in Figure 2.

Figure 2: The quantity of solute in each tank in the CODEE example.

**The general case**

When the parameters are all nonnegative, it is easy to see that the trace of \( A \) is nonpositive and the det is nonnegative. With a bit more work it can also be shown that the discriminant of the quadratic characteristic polynomial is nonnegative so that both eigenvalues are real (and, from the det and trace information, are both nonpositive). Again, when \( \det(A) > 0 \), the equilibrium is approached at \( t \) tends to infinity.

2.2 General linear systems

In general the problem is to solve for \( y(t) \) satisfying

\[
\frac{dy}{dt} - A(t)y = f(t) \tag{N}
\]

with \( y(0) = y_0 \) given. Here \( A(t) \) is a given matrix-valued function of \( t \), and \( f(t) \), the ‘forcing’, is a given vector-valued function of \( t \). Think of \( t \) as ‘time’. This is called an initial-value problem.
**Definition.** When \( f(t) \equiv 0 \ \forall \ t, \) 
\[
\frac{dy}{dt} - A(t)y = 0,
\]
we say \((H)\) is a *homogeneous* equation, while \((N)\) is called *nonhomogeneous*.

**MAJOR TOPICS**

1. A formula, the ‘variation of parameters’ formula, for solving \((N)\) in terms of certain matrix solutions \( \Phi(t) \) of 
\[
\frac{d\Phi}{dt} - A(t)\Phi = 0.
\]
   Nonsingular \( \Phi \) satisfying this are called ‘fundamental matrices’. We treated this topic in the earlier section on matrix revision, matrix inverses.

2. The *constant-coefficient* case, the case 
\[
A(t) = A, \quad A \text{ a constant matrix}.
\]
   We will see similarity and eigenvalues used to determine its solution. In particular, the fundamental matrix which also satisfies \( \Phi(0) = I \), the identity matrix, is denoted 
\[
\Phi(t) = \expm(tA).
\]

3. Special methods, ‘method of undetermined coefficients’ for constant coefficients and special forms of the right-hand side \( f \). Definitely not examinable in this unit, but if you ever have to do hand calculation, they are easier than variation of parameters.

### 2.3 Variation of parameters

This is repeated from before, where we gave it as an example of using matrix inverses. If one can find a certain matrix function – the fundamental matrix, defined below – closely related to the homogeneous system, there is a formula (called the variation of parameters formula, and reducing to the integrating factor method when \( n = 1 \)) which solves (after some integrations) the nonhomogeneous problem \((N)\).

**Definition.** A *fundamental matrix* \( \Psi \) for 
\[
\frac{dx}{dt} - A(t)x = 0
\]
is an \( n \) by \( n \) matrix whose columns are linearly independent solutions of the system.

Let \( \Phi(t) \) be the matrix satisfying 
\[
\frac{d\Phi}{dt} - A(t)\Phi = 0,
\]
\[
\Phi(0) = I \text{ (identity matrix)}.
\]

This \( \Phi \) is called *the (principal) fundamental matrix* (with initial time \( t = 0 \)).

**CONSTANT COEFFICIENT EXAMPLES** Consider the two matrices \( A \) and \( M \), listed with their eigenvalues and eigenvectors below:

\[
A = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}, \quad \text{e'val } \sqrt{2}, \quad \text{e'vec } \begin{bmatrix} 1 \\ 1 + \sqrt{2} \end{bmatrix}, \quad \text{e'val } -\sqrt{2}, \quad \text{e'vec } \begin{bmatrix} 1 \\ 1 - \sqrt{2} \end{bmatrix};
\]

\[
M = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad \text{e'val } 1, \quad \text{e'vec } \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \text{e'val } 0, \quad \text{e'vec } \begin{bmatrix} 1 \\ -1 \end{bmatrix}.
\]
You can readily check that the following give fundamental matrices for each of the systems:

\[
\Psi_A = \begin{bmatrix}
\exp(t\sqrt{2}) & \exp(-t\sqrt{2}) \\
(1 + \sqrt{2}) \exp(t\sqrt{2}) & (1 - \sqrt{2}) \exp(-t\sqrt{2})
\end{bmatrix}
\]

\[
\Psi_M = \begin{bmatrix}
\exp(t) & 1 \\
\exp(t) & -1
\end{bmatrix}
\]

Neither of these is equal to the identity matrix at \( t = 0 \). So although they are fundamental matrices for their respective systems, they are not principal fundamental matrices (and, for constant coefficient systems we will call the principal fundamental matrix the matrix exponential, but more on this later).

One of the reasons the principal fundamental matrix is important is the variation of parameters formula: that

\[
y(t) = \Phi(t)(y(0) + \int_0^t (\Phi(s))^{-1}f(s)ds),
\]

solves the initial-value problem (N) above.

### 2.4 Constant coefficient homogeneous systems

An important class of differential equations for which the (principal) fundamental matrix, or equivalently the general solution, can be found is d.e.s with constant coefficients, i.e. \( A(t) \) is constant in \( t \).

For the rest of the treatment of d.e.s here, \( A(t) \) is a constant matrix and we consider \( \frac{dy}{dt} = Ay \).

As in other areas, there are several ways through to solution using CA, relevant CA commands being:

<table>
<thead>
<tr>
<th>Student Matlab</th>
<th>Maple</th>
<th>Mathematica</th>
</tr>
</thead>
<tbody>
<tr>
<td>dsolve</td>
<td>dsolve</td>
<td>Dsolve</td>
</tr>
<tr>
<td>expm(t*A)</td>
<td>exponential(A,t)</td>
<td>MatrixExp[ A t]</td>
</tr>
<tr>
<td>[V,D]=eig(A)</td>
<td>eigenvects(A)</td>
<td>Eigenvectors[ A ]</td>
</tr>
<tr>
<td>[J,P]=jordan(A)</td>
<td>jordan(A,'P')</td>
<td>JordanDecomposition[ A ]</td>
</tr>
</tbody>
</table>

**EXAMPLE.** \( \frac{dy}{dt} = Ay \) with \( A = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix} \)

The eigenvalues of \( A \) are \( +\sqrt{2} \) and \( -\sqrt{2} \). The principal fundamental matrix is

\[
\Phi(t) = \begin{bmatrix}
c - \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\
\frac{\sqrt{2}}{2} & c + \frac{\sqrt{2}}{2}
\end{bmatrix}
\]

where \( s = \sinh(t\sqrt{2}) \) and \( c = \cosh(t\sqrt{2}) \).

Well, you can CHECK this (calculate \( \Phi'(t) - A\Phi(t) \)), but, at this stage, I haven’t told you how to derive it. And, it is a bit messy, so return to the Matlab given here later.

We now return to general constant matrices \( A \). In the case that \( A \) is \( 1 \times 1 \), this was solved in first year maths. \( \frac{dy}{dt} = Ay \), \( y(0) = 1 \) has solution \( \phi(t) = \exp(At) \) where exp can be defined by its taylor series

\[
\exp(At) = 1 + \sum_{j=1}^{\infty} \frac{t^j}{j!} A^j.
\]
This formula generalises to the case $A$ an $n \times n$ matrix, the 1 being replaced by an identity matrix.

The (principal) fundamental matrix solving
\[
\frac{d\Phi}{dt} = A\Phi , \quad \Phi(0) = I,
\]
is, [StrBC] §5.4, p276, equation (3),
\[
\Phi(t) = \expm(tA) = I + \sum_{j=1}^{\infty} \frac{t^j}{j!} A^j.
\]

Though we won’t prove it, the series happens to be sufficiently rapidly convergent to allow the interchange of limit processes – differentiation and infinite summation. Believing this, and using the expm series for $\Phi$, we have
\[
\Phi'(t) = 0 + \sum_{j=1}^{\infty} \frac{t^j}{(j-1)!} A^j
\]
\[
= A \left( I + \sum_{j=2}^{\infty} \frac{t^{j-1}}{(j-1)!} A^{j-1} \right)
\]
\[
= A\Phi
\]
as required.

EXAMPLES. A $2 \times 2$ example
\[
A = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}, \quad \frac{t^j A^j}{j!} = \begin{bmatrix} \frac{t^j \lambda_1^j}{j!} & 0 \\ 0 & \frac{t^j \lambda_2^j}{j!} \end{bmatrix}, \quad \expm(tA) = \begin{bmatrix} \exp(\lambda_1 t) & 0 \\ 0 & \exp(\lambda_2 t) \end{bmatrix}.
\]

A larger example
\[
A = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \quad \expm(tA) = \begin{bmatrix} \exp(\lambda_1 t) & 0 & 0 \\ 0 & \exp(\lambda_2 t) & 0 \\ 0 & 0 & \exp(\lambda_3 t) \end{bmatrix}.
\]

We will learn easier ways – via matrix similarity and jordan forms – to evaluate $\expm(A, t)$. This is easier than the series when the jordan form is available. ([StrBC] 5L, p277 treats the case of the simplest jordan forms, i.e for $A$ diagonalizable.)

2.5 The relationship with eigenvalues

Consider
\[
y' = \frac{dy}{dt} = Ay. \quad (H)
\]
You are reminded that, if $u$ is an eigenvector of $A$, i.e $Au = \lambda u$, then a solution of (H) is given by
\[
y(t) = \exp(\lambda t)u.
\]

Easy Proof.
\[
\frac{d}{dt} \exp(\lambda t)u = \lambda \exp(\lambda t)u. \quad (1)
\]
and
\[ A \exp(\lambda t)u = \exp(\lambda t)Au = \lambda \exp(\lambda t)u. \] (2)

(1) and (2) together give the result.

If there is a full set \((n)\) of linearly independent eigenvectors, we can start from these to construct \(\expm(At)\). See the Matlab for the very first example of a principal fundamental matrix when we had \(A = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix} \). I prefer, though, to organize the calculations along the following lines – using similarity.

**Similarity, matrix exponentials, and jordan forms**

Strategy for solving \( \Phi' = A\Phi \)

1. Find a simpler \(B\) similar to \(A\), \(B = P^{-1}AP\)
2. Solve \(\Psi' = B\Psi\)
3. \(\Phi = P\Psi P^{-1}\)

Check 3 works.

\[
\Phi' = P\Psi'P^{-1} = PB\Psi P^{-1} \quad \text{using 2,}
\]
\[
= PP^{-1}AP\Psi P^{-1} \quad \text{using 1,}
\]
\[
= A\Phi \quad \text{using definition 3 of } \Phi,
\]
as required.

**SUMMARY:** If \(P^{-1}AP = B\), we have

\[
A = PBP^{-1} \quad \text{and } \expm(tA) = P\expm(tB)P^{-1}.
\]

**EXAMPLES** \(\Psi = \expm(tB)\) for simple \(B\).

\[
B = \begin{bmatrix} \lambda & 0 \\ 0 & \mu \\ \lambda & 1 \\ 0 & \lambda \end{bmatrix}, \quad \expm(tB) = \begin{bmatrix} \exp(\lambda t) & 0 \\ 0 & \exp(\mu t) \\ \exp(\lambda t) & 1 \\ t 0 \end{bmatrix}
\]

\[
\expm(\alpha t) = \begin{bmatrix} \exp(\alpha t) & \cos(\omega t) - \sin(\omega t) \\ \sin(\omega t) & \cos(\omega t) \end{bmatrix}
\]

It turns out that this is essentially all the different sorts of behaviour that are possible. Indeed the last matrix is similar to

\[
D = \begin{bmatrix} \alpha + i\omega & 0 \\ 0 & \alpha - i\omega \end{bmatrix}, \quad \expm(Dt) = \exp(\alpha t) \begin{bmatrix} \exp(i\omega t) & 0 \\ 0 & \exp(-i\omega t) \end{bmatrix},
\]

so that there are really only two different sorts of solution behaviour in \(\Phi(t)\) exponentials in \(t\) and (polynomials in \(t\)) times (exponentials in \(t\)).
Numerical EXAMPLE. Find exp\(m(tA)\) when \(A = \begin{bmatrix} 2 & 1 \\ -3 & 6 \end{bmatrix}\).

Solution. First, as Matlab jordan will readily tell you, we have

\[
A = \begin{bmatrix} 2 & 1 \\ -3 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & 5 \end{bmatrix} \begin{bmatrix} \frac{3}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix} = PBP^{-1}.
\]

Following the strategy above we have

\[
\expm(tA) = P \begin{bmatrix} \exp(3t) & 0 \\ 0 & \exp(5t) \end{bmatrix} P^{-1}
\]

\[
= \begin{bmatrix} \frac{3}{2} \exp(3t) - \frac{1}{2} \exp(5t) & -\frac{1}{2} \exp(3t) + \frac{1}{2} \exp(5t) \\ \frac{3}{2} \exp(3t) - \frac{1}{2} \exp(5t) & -\frac{1}{2} \exp(3t) + \frac{1}{2} \exp(5t) \end{bmatrix}.
\]

EXERCISE. Check the details in the preceding example.

Another Numerical EXAMPLE. For the matrix

\[
M = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}
\]

whose eigenvalues and eigenvectors, and for which a fundamental matrix was given before, find \(\expm(Mt)\) beginning with

\[
P = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = P^{-1} \quad \text{and} \quad P^{-1}MP = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = B.
\]

Solution.

\[
\expm(tM) = P \exp(tB)P^{-1} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \exp(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \exp(t) + 1 & \exp(t) - 1 \\ \exp(t) - 1 & \exp(t) + 1 \end{bmatrix}
\]

Carrying out the above strategy in general takes us back to the question we asked after looking at diagonalization.

**What simple form can any \(n \times n\) matrix be reduced to by similarity?**

I.e. given \(A\), find a simple form for \(B = P^{-1}AP\)? The answer, from before, is as follows.

JORDAN FORM THEOREM. [StrBC] 5U, p312. An \(n \times n\) matrix \(A\) with eigenvalues, repeated with their algebraic multiplicities, \(\lambda_1, \lambda_2, \ldots, \lambda_n\) is similar to a matrix with these eigenvalues in the diagonal positions, zeros and ones along the superdiagonal and zeros elsewhere.

Remark. The number and location of the 1s depends on the difference between algebraic and geometric multiplicities.

The easiest nondiagonalizable EXAMPLE. Let

\[
Z = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}.
\]

Find \(\expm(tZ)\).
Solution. The easiest way is, in the series for \( \expm(Zt) \), to use the fact that \( Z^2 = 0 \) and hence \( Z^j = 0 \) for all \( j \geq 2 \). Thus the series has only 2 terms

\[
\expm(tZ) = I + tZ = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}.
\]

This easy example with \( A \) not diagonalizable is given in [StrBC] Q5.4.7, p287.

The general form of \( \expm(Jt) \) for a Jordan block \( J \) is given in [StrBC] §5.6, p314.

A Harder Numerical EXAMPLE. For the defective matrix we have looked at before

\[
A = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & -2 \end{bmatrix}.
\]

find \( \expm(tA) \).

Solution, via Matlab. Cutting and pasting from Matlab:

\[
\expm(tA) = \begin{bmatrix} e^t & -te^t & -1/3 te^t + 1/9 e^t - 1/9 e^{-2t} \\ 0 & e^t & 1/3 e^t - 1/3 e^{-2t} \\ 0 & 0 & e^{-2t} \end{bmatrix}
\]

We remark that, as \( A \) is upper triangular, the series shows that \( \expm(At) \) will also be upper triangular.

The behaviour with the polynomials in \( t \) multiplied by the exponentials is typical of what happens when \( A \) is defective (i.e. not diagonalable).

Observe that the variation of parameters formula for \textit{constant} \( A \) does not actually involve any hard calculations of matrix inverses as, if

\[
\Phi(t) = \expm(tA), \quad \text{then } \Phi(s)^{-1} = \Phi(-s).
\]

Various other comments on matrix exponentials are possible

\[
AB = BA \implies \expm(A + B) = \expm(A)\expm(B).
\]

(See also [StrBC] §5.4, Q5.4.5 p287 and 5F on commuting matrices and simultaneous diagonalizability, or see [StrWC] p261 Thm 6F concerning commuting matrices sharing eigenvectors.)

2.6 Method of undetermined coefficients

This is a generalisation to constant coefficient systems of the method with the same name you learnt for a single d.e. in first year. It is, when applicable, MUCH easier than the more general method of variation of parameters. (You may also use Laplace transforms for these problems. See your M132 work.)

Let’s begin with an example. Find a particular solution of

\[
\frac{dy}{dt} - Ay = \exp(\alpha t)r,
\]

when \( \alpha \) is NOT an eigenvalue of \( A \).
Look for solution \( y(t) = \exp(\alpha t)s \). Substituting, we have
\[
\alpha s - As = r, \quad \text{or} \quad (\alpha I - A)s = r,
\]
and this has a unique solution since \( \alpha I - A \) is invertible.

Remark. The case when the rhs is a constant vector \( r \) is the special case \( \alpha = 0 \).

This generalises.

Conclusions. The algorithm for solving constant coefficient linear systems of d.e.s
\[
\frac{dy}{dt} - Ay = f,
\]
is precisely defined and, being just routine, calculation can be implemented in computer algebra programs. (Student Matlab, or the full Matlab with the Symbolic Toolbox, have computer algebra capabilities.)

Although in the past engineers had to be able to do hand calculations for problems like that on previous page, you can now leave tedious details to mathematical software.

An understanding will always be necessary. In particular both hand-calculation and computer algebra software depend on being able to find eigenvalues. If these can only be found approximately numerically you may have to work harder. The shape of the final solution will also depend on whether exact expressions for eigenvalues are short or not.
2.7 \( y' = Ay \): Eigenanalysis of the stability of the zero solution

As is perhaps correct in a first account, our treatment of \( \expm(tA) \), so far, has concentrated on actually calculating it. However, to do this exactly, one has to find the eigenvalues exactly, and this is often not possible. Fortunately there is easily extracted useful information without having the eigenvalues exactly calculated.

**Definition.** A square matrix \( A \) is said to be **stable** if the real parts of all its eigenvalues are negative.

The zero solution of \( y' = Ay \) is an ‘equilibrium point’. A question is: ‘is it stable?’

It is worth beginning with the \( n = 1 \) case: \( A \) is just a real number. If \( A < 0 \), the solution \( \exp(tA)y(0) \to 0 \) as \( t \to \infty \) so the zero solution is stable. I.e. start near it, and the system takes you closer to it. If \( A > 0 \), the zero solution is unstable. Let’s now return to systems. The general result is that the zero equilibrium (solution) is stable iff the matrix \( A \) is stable.

There is quite a lot more that can be said. Here we confine our further remarks to the case where \( A \) is 2 by 2. (Of course, now we could actually find the eigenvalues exactly if we wished to do so.) Perhaps you might regard this ‘more that can be said’ as telling you a bit more about how the solutions behave close to the equilibrium. To give a 1st year example, the damped linear oscillator has 0 as a stable solution, but the solutions look different in the underdamped and the overdamped cases.

### 2.7.1 A couple of examples of stability for \( \dot{y} = Ay \), \( A \) 2 by 2

**Example.**

\[
A = \begin{bmatrix} \alpha & -\omega \\ \omega & \alpha \end{bmatrix}, \quad \expm(tA) = \exp(\alpha t) \begin{bmatrix} \cos(\omega t) & -\sin(\omega t) \\ \sin(\omega t) & \cos(\omega t) \end{bmatrix}
\]

\( A \) is a stable matrix iff \( \alpha < 0 \).

What do the solutions look like as \( t \) increases?

Describe the trajectory \((y_1(t), y_2(t))\) in the \((y_1, y_2)\)-plane if we start from \((1, 0)\).

**Solution.** Since \( \alpha < 0 \), any solution tends to zero as \( t \to +\infty \).

For the solution starting at \((1, 0)\) we have

\[
y(t) = \exp(\alpha t) \begin{bmatrix} \cos(\omega t) \\ \sin(\omega t) \end{bmatrix}
\]

So, setting \( r^2 = y_1^2 + y_2^2 \), we have

\[
r^2 = \exp(2\alpha t), \quad \theta = \arctan \left( \frac{y_2}{y_1} \right) = \omega t
\]

In polar coordinates this is

\[
r = \exp(\alpha \theta / \omega)
\]

which, with \( \alpha < 0 \) is a spiral tending to zero as \( \theta \) increases.

**Example.** Consider the differential equation, for an unforced damped linear oscillator:

\[
y'' + \frac{1}{2}y' + \frac{257}{16}y = 0
\]
or, equivalently, with \( [y_1; y_2] = y = [y; y'] \),

\[
y' = Ay \quad \text{with} \quad A = \begin{bmatrix} 0 & 1 \\ -\frac{257}{16} & -\frac{1}{2} \end{bmatrix}.
\]

Using Matlab one finds that the eigenvalues of \( A \) are \(-\frac{1}{4} \pm 4i\), and that

\[
\expm(tA) = \exp(-t/4) \begin{bmatrix} \cos(4t) + \frac{1}{16} \sin(4t) & \frac{1}{4} \sin(4t) \\ -\frac{257}{64} \sin(4t) & \cos(4t) - \frac{1}{16} \sin(4t) \end{bmatrix}. 
\]

Now suppose that the system is started with \( y(0) = [0; 1] \), i.e. \( y(0) = 0, y'(0) = 1 \).

- Write down the formula for \( y(t) \) solving this initial value problem.
- How many times does \( y(t) = y_1(t) \) pass through 0 when \( t \) varies from 0 up to 5?
- Sketch, with attention to the main qualitative features, but without too much concern about detailed numerics, the general shape of the trajectory of the solution of part (a) in the \((y_1, y_2)\)-plane.

Matlab’s `ezplot` is the easiest way to get to the drawing of the spiral, but, if you don’t have convenient access to Matlab at present, You may use the following table to get a few points on the plot:

<table>
<thead>
<tr>
<th>( t )</th>
<th>0</th>
<th>0.5</th>
<th>1.0</th>
<th>1.5</th>
<th>2.0</th>
<th>2.5</th>
<th>3.0</th>
<th>3.5</th>
<th>4.0</th>
<th>5.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>( y_1(t) )</td>
<td>0</td>
<td>.20</td>
<td>-.15</td>
<td>-.048</td>
<td>.15</td>
<td>-.073</td>
<td>.10</td>
<td>-.026</td>
<td>.065</td>
<td></td>
</tr>
<tr>
<td>( y_2(t) )</td>
<td>1</td>
<td>-.42</td>
<td>-.47</td>
<td>.67</td>
<td>-.13</td>
<td>-.43</td>
<td>.41</td>
<td>.031</td>
<td>-.35</td>
<td>.10</td>
</tr>
</tbody>
</table>

Solution. With \( \expm(tA) \) as given above

\[
y(t) = \expm(tA) \begin{bmatrix} 0 \\ 1 \end{bmatrix}
\]

so

\[
y_1(t) = \exp(-\frac{t}{4}) \frac{\sin(4t)}{4}.
\]

From this we can find the number of zeros for the interval of \( t, 0 < t \leq 5 \). It is the same as the numbers of zeros of \( \sin(x) \) (where \( x = 4t \)) with \( 0 < x < 20 \). Since \( 6\pi \approx 18.85 \) and \( 7\pi \approx 22 \), there are, after starting at zero, 6 more zeros of \( y_1 \) before \( t = 5 \).

To do the plot the easy way, using Matlab:

```matlab
syms t y1 s4
s4=sym(4)
y1=exp(-t/s4)*sin(s4*t)/s4;
y2=diff(y1,t);
ezplot(y1,y2,[0 5])
```

The result of running the Matlab is the spiral shown in Figure 3.

The examples in this subsection have both sketches/plots as spirals. Other sorts of behaviour are possible. It isn’t hard – merely a bit long – to check out all the various possibilities for 2 by 2 matrices. The next subsection will have an account of this.
Figure 3: Example of a phase portrait for a underdamped linear pendulum. Matlab has labelled the axes $x$ corresponding to our $y_1$ and $y$ corresponding to our $y_2$. 
2.7.2 General comments about $\dot{y} = Ay$, $A$ 2 by 2

**THEOREM.** Let $A$ be a 2 by 2 real matrix, $p$ is the trace of $A$ and $q = \det(A)$. Let $D = p^2 - 4q$. There are three possibilities for the eigenvalues of $A$. 

- The eigenvalues of $A$ are real and distinct ($D > 0$).
- The eigenvalues of $A$ are real and equal ($D = 0$).
- The eigenvalues of $A$ are a complex conjugate pair ($D < 0$).

In the damped nonlinear pendulum example, the first case corresponded to the unstable upwards equilibrium, the last to the downwards equilibrium.

The proof of the theorem follows from calculating the eigenvalues of $A$. They are

$$
\lambda_{\pm} = \frac{1}{2} \left( \text{trace}(A) \pm \sqrt{\text{trace}(A)^2 - 4 \det(A)} \right) \\
= \frac{1}{2} \left( p \pm \sqrt{p^2 - 4q} \right).
$$

There are only a few different sorts of behaviours in the phase-plane for 2nd-order constant coefficient systems. See Figure 4. The different cases can be listed.

- $\det(A) < 0$. The equilibrium point is a **saddle**.
- $\det(A) > 0$.
  - $\text{trace}(A) > 0$. The equilibrium point is a **source** (unstable).
  - $\text{trace}(A) < 0$. The equilibrium point is a **sink** (stable).
  - $D < 0$. The equilibrium point is a **spiral**.
  - $D > 0$. The equilibrium point is a **node**.
  - $D = 0$. $A = cI$, the equilibrium point is a **focus**.
    - $D = 0$. $A \neq cI$, the equilibrium point is an **improper node**.
- $\det(A) > 0$ and $\text{trace}(A) = 0$. **centre**: nonzero purely imaginary eigenvalues.

- $\det(A) = 0$. Three subcases:
  - single zero eigenvalue: **saddle-node**;
  - double zero eigenvalue, one-dimensional nullspace: **shear**;
  - $A = $ zero matrix.
Figure 4: The different sorts of equilibrium points. StrBC Fig 5.2 p281, O’Neil AEM Fig 11.47.
3 NONLINEAR SYSTEMS OF DEs
NUMERICAL SOLUTION
EIGENANALYSIS OF STABILITY OF EQUILIBRIA

3.1 Introduction

The unit as a whole has an emphasis on ‘linear systems’, and will culminate with a treatment of Fourier series and uses in linear d.e.s. However, the handbook entry requires us to treat numerical solution of nonlinear d.e.s too. Let’s treat an initial-value problem, written as a system:

\[
\frac{dy}{dt} = f(t, y), \quad y(0) = y_0 \text{ given.} \tag{1}
\]

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 appropriate software, e.g. the ode23 or ode45 commands in Matlab. The first part of these notes will describe an example (the nonlinear rigid-body pendulum with damping) treated with Matlab. The Matlab code is up at the unit’s web pages.

- Next there is the question of how does it work. There is a treatment of Euler’s method in Stewart ‘Calculus’ Chpt 10. 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. (The lab at http://www.maths.uwa.edu.au/~keady/Matlab/ONlabs/l05sym.html gives an account of another of the methods.) I will let you read these items, and won’t dwell on them now.

3.2 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 autonomous if it can be written

\[
\frac{dy}{dt} = F(y), \tag{2}
\]

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.

3.3 The nonlinear damped pendulum

The differential equation is

\[
\frac{d^2\theta}{dt^2} + c \frac{d\theta}{dt} + \frac{g}{l} \sin(\theta) = 0 .
\]
The first step in getting this into Matlab is to rewrite it as a system. Define \( x_1 = \theta \) and \( x_2 = \frac{d\theta}{dt} \). Then the system is

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

Next put into Matlab code the function on the rhs of this:

```matlab
function dx = dampPendf(t,x)
global c g l
% Denote theta by x(1), its time derivative by x(2)
dx= [x(2);-(c*x(2)+(g/l)*sin(x(1)))];
```

We are now ready to use Matlab to solve initial value problems. Let’s begin by solving the problem with a modest displacement, and zero initial velocity, when we wouldn’t expect (and unless we have the code wrong won’t get) much difference from the linear pendulum we know all about.

```matlab
global c g l
c= 0.1; % damping constant
g= 9.8; % acceleration due to gravity
%m= 0.1; % mass in kg
l= 0.5; % length in m
% Denote theta by x(1), its time derivative by x(2)
%dampPendf= inline(['[x(2),-c*x(2)-(g/l)*sin(x(1))]',','t','','x']);
% Start the pendulum from rest, at theta=x(1)=0.5radians
x10=-0.5;
x20=0;
[ta,xa]= ode45('dampPendf',[0 20],[x10; x20]);
hold on
plot(ta,xa(:,1))
title(['theta against time, theta(0)=',num2str(x10),' dtheta(0)=',num2str(x20)])
hold off
```

The results are shown in Figure 5.

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 6.
Figure 5: $\theta(0) = -0.5, \frac{d\theta}{dt} = 0$: not much different from a linear pendulum with small initial values.

Figure 6: $\theta(0) = -0.5, \frac{d\theta}{dt} = 10$: spins a few times over the top.
3.4 Various graphical ideas for 2nd order systems

Many of you will have looked in Stewart 4th and 5th eds Chapter 10 e.g. §10.7. There you will have seen 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 M131 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 M131.)

The Matlab code which produces Figures 7,8 is at the M235 web pages:

Figure 7: Phase plot: the continuous curves are those from the numerical solution of the d.e.. The arrows show the direction field.

Figure 8: Phase plot: the continuous curve is just the one corresponding to the zero initial velocity.
3.5 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 Stewart, §10.5. 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 9.

Figure 9: A couple of solutions to the logistic equation

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

$$DF = \begin{bmatrix} \frac{\partial F_i}{\partial y_j} \end{bmatrix}.$$  

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.

$$F(y) = \begin{bmatrix} y_2 \\ -cy_2 - \frac{g}{l} \sin(y_1) \end{bmatrix}.$$  

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

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

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 Stewart. 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.**

### 3.6 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.

### 3.7 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:

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

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’.

28
3.7.1 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. For examples of phase plane analysis, see Stewart ‘Calculus’ (e.g. for a different problem, predator-prey, §10.7) or Boyce and Di Prima ‘Elementary differential equations ...’.

  The same 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, using Matlab, next.

3.7.2 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 matlab code given, we ask matlab to find the equilibrium points, to find the jacobian, and then to determine the stability.

```matlab
%This example is treated in many texts, e.g.
%Boyce and Di Prima
%‘Elementary differential equations ...’.
%Barnes and Fulford,
%‘Mathematical Modelling with Case Studies’ §6.3

% rhs of D.E.s describing competing species
sym y_1 beta_1 d_1 c_1 y_2 beta_2 d_2 c_2 f1 f2
%f1= y_1*(beta_1 - d_1*y_1 -c_1*y_2);
f2= y_2*(beta_2 - d_2*y_2 -c_2*y_1);
ejilibSols= solve( f1,f2 , y_1,y_2 )
ejilib1= [eilibSols.y_1(1),eilibSols.y_2(1)]
ejilib2= [eilibSols.y_1(2),eilibSols.y_2(2)]
ejilib3= [eilibSols.y_1(3),eilibSols.y_2(3)]
ejilib4= [eilibSols.y_1(4),eilibSols.y_2(4)]
```

%There are at most four equilibrium solutions:

\%(0,0), \%(0, beta_2/d_2), \%(beta_1/d_1,0),
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 stabilization of the other.

\[
\frac{(c_1 \cdot \beta_2 - d_2 \cdot \beta_1)}{(c_1 \cdot c_2 - d_1 \cdot d_2)}, \quad \frac{(c_2 \cdot \beta_1 - d_1 \cdot \beta_2)}{(c_1 \cdot c_2 - d_1 \cdot d_2)}
\]

\frown{ColVec= [f1; f2]}
\frown{yRowVec= [y_1, y_2]}
\frown{J= jacobian(fColVec, yRowVec)}

\frown{J1=subs(J, {y_1,y_2}, {equilib1(1), equilib1(2)})}
\frown{eig(J1)}
% ans =
% [ beta_1]
% [ beta_2]
% Both eigenvalues are positive; [0,0] is unstable

\frown{J2=subs(J, {y_1,y_2}, {equilib2(1), equilib2(2)})}
equilib2
\frown{eig(J2)}
% ans =
% [ beta_1/d_1, 0]
% [-beta_1]
% [ (-beta_1*c_2+d_1*beta_2)/d_1]
equilib2 is stable iff
% \text{IntY1} = (beta_1/d_1 - beta_2/c_2) > 0

\frown{J3=subs(J, {y_1,y_2}, {equilib3(1), equilib3(2)})}
equilib3
\frown{eig(J3)}
% ans =
% [ 0, beta_2/d_2]
% [-beta_2]
% [- (beta_2*c_1-d_2*beta_1)/d_2]
equilib3 is stable iff
% \text{IntY2} = (beta_2/d_2 - beta_1/c_1) > 0

\frown{J4=subs(J, {y_1,y_2}, {equilib4(1), equilib4(2)})}
equilib4
% both entries nonzero
\frown{eig(J4)}
% after some analysis we find that
% equilib4 is stable iff
% \text{IntY2<0 and 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.

### 3.7.3 Phase plots

Phase plots can be found in the usual way.

```matlab
% 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;
Y1step=0.125; Y2step=0.1;
[Y2vals= cat(2,[-7:1:-3],[-3:0.5:3],[3:1:7]);
[Y1,Y2] = meshgrid(-Y1step:Y1step:1.4, -Y2step:Y2step:1);
f1 = Y1.*(beta1-d1*Y1-c1*Y2);
f2 = Y2.*(beta2-d2*Y2-c2*Y1);
L = sqrt(f1.^2 + f2.^2);
figure
hold on
quiver(Y1,Y2, f1./L, f2./L,0.3)
axis equal tight
xlabel('y1'); ylabel('y2');
str1= 'phase plot for competing species: ';
title(strcat(str1, ...
' beta2= ',num2str(beta2), ' , c2 = ',num2str(c2)))
plot(0,0,'b*')
plot(0,beta2/d2,'b*')
plot(beta1/d1,0,'b*')
plot((c1*beta2-d2*beta1)/(c1*c2-d1*d2), ...
(c2*beta1-d1*beta2)/(c1*c2-d1*d2),'b*');
```

The results of running this are given in Figure 10.

### References


Figure 10: The parameters are 1 except those given on the plot.
