- ↪ We now turn our attention to systems of differential equations. These are needed for modelling of more complex systems, such as the population dynamics of interacting species (predator-prey, host-parasite, etc) or the concentration of reactants in reaction kinetics or communication within the nervous system, etc.
- ↪ As for single ODEs, we will use three approaches to studying systems of ODEs:
  - Analytical (finding exact solutions).
  - Numerical (approximating solutions to initial value problems).
  - Geometrical (determining solution trends without solving the equations).
- ↪ We will then turn our focus to deriving models which involve systems of ODEs and will use some combination of the three approaches/tools above to look at solutions to such systems.
- ↪ I want you to acquire a good understanding of the three approaches but also don't focus exclusively on them; learn to regard them as tools to help you with exploring solutions to the systems of ODEs which arise in your models.

- As with single ODEs, probably the newest thing to most of you will be the geometrical approach and that is likely what we will spend most of our time on.

- The analytical approach - in which you learn to solve linear, constant coefficient systems of ODEs using techniques and concepts from linear algebra *(notably eigenvalues and eigenvectors)* - will be useful also for the insight it gives to the later geometrical approach. *The truth is, however, with many of the models we study it will be very difficult or impossible to find exact solutions*.

- For specific models, with initial conditions, often the numerical approach will be invaluable.

Advice On Navigating This Massive (almost 150 pages) Lecture

- ↪ Anyone planning on doing **postgraduate studies**, particularly in any field related to *Applied Mathematics*, should try to become familiar with **all** of this material, including the supplementary reading. Having a good understanding of ODEs and the related theory is important for many areas of application, for understanding some aspects of PDEs, as well as in the study of other areas of Mathematics such as Dynamical Systems, Differential Geometry, and Lie Groups.
- ↪ Otherwise, the main things to try to understand are: how to solve constant coefficient homogeneous systems of linear ODEs, how to convert higher order ODEs to systems of first order ODEs, how to solve IVPs numerically using something like MATLAB's `ode45()`, and how to classify steady states of linear and nonlinear systems of ODEs *(including some intuitive idea of what those steady states look like geometrically and how to interpret them in a real-world application)*.

End of Section

Analytical Solutions to Systems of First Order ODEs

- Much of what we will do next will be similar to what one does to solve linear, constant coefficient ODEs: - for example, the linear second order constant coefficient ODE

$$a\frac{d^2x}{dt^2} + b\frac{dx}{dt} + cx = Q(t)$$

(especially the simple case when $Q(t) = 0$ - A HOMOGENEOUS ODE).

- We will also be using many of the ideas from the *Supplementary Lecture on Eigenvalues/Eigenvectors*.

Definitions and Conventions

DEFINITIONS: A general system of n first order linear differential equations is one which can be written in the form

$$\begin{aligned}
x_1'(t) &= p_{11}(t)x_1(t) + p_{12}(t)x_2(t) + \ldots + p_{1n}(t)x_n(t) + g_1(t)\\
x_2'(t) &= p_{21}(t)x_1(t) + p_{22}(t)x_2(t) + \ldots + p_{2n}(t)x_n(t) + g_2(t)\\
\vdots &= \vdots\\
x_n'(t) &= p_{n1}(t)x_1(t) + p_{n2}(t)x_2(t) + \ldots + p_{nn}(t)x_n(t) + g_n(t)
\end{aligned}$$

or, in matrix form, $\boldsymbol{\vec{x}\,'(t) = P(t)\vec{x}(t) + \vec{g}(t)}$, where $\vec{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}$,

$P(t) = \begin{bmatrix} p_{11} & p_{12} & \ldots & p_{1n} \\ p_{21} & p_{22} & \ldots & p_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ p_{n1} & p_{n2} & \ldots & p_{nn} \end{bmatrix}$, and $\vec{g}(t) = \begin{bmatrix} g_1(t) \\ g_2(t) \\ \vdots \\ g_n(t) \end{bmatrix}$.

↪ EXAMPLE 1 For example,

$$\begin{aligned} e^{2t}x'(t) + \sin^2(t)x(t) \quad + 3y(t) &= 10\cos t \\ y'(t) \quad + x(t) - \ln(t^2+1)y(t) &= t^3 - 4t^2 \end{aligned}$$

is a linear ODE system of equations since it can be rewritten as

$$\begin{aligned} x'(t) &= -\frac{\sin^2 t}{e^{2t}}x(t) - \frac{3}{e^{2t}}y(t) + \frac{10\cos t}{e^{2t}} \\ y'(t) &= -x(t) + \ln(t^2+1)y(t) + t^3 - 4t^2 \end{aligned}$$

or, in matrix form,

$$\underbrace{\frac{d}{dt}\begin{bmatrix} x(t) \\ y(t) \end{bmatrix}}_{\uparrow\,\vec{x}\,'(t)} = \underbrace{\begin{bmatrix} -\frac{\sin^2 t}{e^{2t}} & -\frac{3}{e^{2t}} \\ -1 & \ln(t^2+1) \end{bmatrix}}_{\uparrow\,P(t)} \underbrace{\begin{bmatrix} x(t) \\ y(t) \end{bmatrix}}_{\uparrow\,\vec{x}(t)} + \underbrace{\begin{bmatrix} \frac{10\cos t}{e^{2t}} \\ t^3 - 4t^2 \end{bmatrix}}_{\uparrow\,\vec{g}(t)}$$

↪ If $\vec{g}(t) = \vec{0}$ in $\vec{x}\,'(t) = P(t)\vec{x}(t) + \vec{g}(t)$ then the linear system is said to be **HOMOGENEOUS**.
So the linear system in EXAMPLE 1 above is *NOT* homogeneous, but

$$\frac{d}{dt}\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} -\frac{\sin^2 t}{e^{2t}} & -\frac{3}{e^{2t}} \\ -1 & \ln(t^2+1) \end{bmatrix} \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} \quad \text{IS HOMOGENEOUS.}$$

↪ So a **homogeneous** linear system of ordinary differential equations can always be written in matrix form as $\boldsymbol{\vec{x}\,'(t) = P(t)\vec{x}(t)}$.

↪ EXAMPLE 2 Is the following system of ODEs linear, and if so is it homogeneous?

$$\begin{aligned} 3\frac{dx}{dt} + 3\frac{dy}{dt} - 2x &= e^t \\ \frac{dx}{dt} + 2\frac{dy}{dt} - y &= 1 \end{aligned}$$

or, in matrix form, $\begin{bmatrix} 3 & 3 \\ 1 & 2 \end{bmatrix} \frac{d}{dt}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} e^t \\ 1 \end{pmatrix}$.

- ANSWER

↪ Similar to the case with n^{th} order linear ODEs in the **Calculus** course in year 1, we will mostly restrict ourselves to the simpler case in which the *COEFFICIENT MATRIX*, $P(t)$, in the homogeneous system $\vec{x}\,'(t) = P(t)\vec{x}(t)$, is a *CONSTANT* matrix. So we focus on solving

$$\vec{x}\,'(t) = A\vec{x}(t), \quad \text{where } A_{n\times n} \text{ is a constant matrix.}$$

↪ EXAMPLE 1 does not have a constant coefficient matrix, but the following does:

$$\frac{d}{dt}\begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = \begin{pmatrix} -3 & 0 & 17 \\ -4 & 1 & 2 \\ 2 & 2 & -25 \end{pmatrix}\begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix}$$

Converting n^{th} order Linear ODEs to Linear Systems of First Order ODEs

- We focus on first order systems of ODEs because, essentially, *all other (systems of) ODEs can be converted to a first order system of ODEs*.
- There is a simple way of transforming an n^{th} order single ODE into a system of n *first order* ODEs. This is probably best demonstrated by an example:
- **EXAMPLE 3** Transform $u^{iv} - 17u''' + tu'' + (\cos t)u' - 23u = 0$ into a system of *four* first order ODEs.
- **ANSWER** Let

$$y_1(t) = u(t); \quad y_2(t) = u'(t); \quad y_3(t) = u''(t); \quad y_4(t) = u'''(t) \qquad (1)$$

We then automatically have that

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= y_3 \\ y_3' &= y_4 \\ y_4' &= u^{iv} = 23u - (\cos t)u' - tu'' + 17u''' \\ \Rightarrow y_4' &= 23y_1 - (\cos t)y_2 - ty_3 + 17y_4 \end{aligned}$$

- We can now solve for y_1, y_2, y_3, and y_4 and use Equations (1) to convert back to a solution in terms of $u(t)$.

$$\text{REMINDER}: \begin{array}{lcllcl} y_1' &=& y_2; & y_2' &=& y_3; \\ y_3' &=& y_4; & y_4' &=& 23y_1-(\cos t)y_2-ty_3+17y_4 \end{array}$$

$\hookrightarrow$ And the matrix form of this system is $\vec{y}\,'(t) = P(t)\vec{y}$, or equivalently

$$\frac{d}{dt}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 23 & -\cos t & -t & 17 \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

In general, to convert the n^{th} order ODE $\boldsymbol{y^{(n)} = F(t, y, y', \ldots, y^{(n-1)})}$ into a system of *n first order ODEs*

$$\text{Let } x_1 = y, \quad x_2 = y', \quad \ldots, \quad x_n = y^{(n-1)}. \quad \text{THEN}$$

$y^{(n)} = F(t, y, y', \ldots, y^{(n-1)})$ is equivalent to $y^{(n)} = F(t, x_1, x_2, \ldots, x_n)$, and

$$\begin{array}{lcl} x_1' &=& x_2 \\ \vdots & \vdots & \vdots \\ x_{n-1}' &=& x_n \\ x_n' &=& F(t, x_1, x_2, \ldots, x_n). \end{array}$$

$\hookrightarrow$ $\boxed{\text{EXAMPLE 4}}$ Transform $\boxed{\textbf{(a)}}$ $y'' + 0.5y' + 2y = 3\cos t$ and $\boxed{\textbf{(b)}}$ $w''' - 3w = 0$ into systems of first order equations.

$\hookrightarrow$ $\boxed{\text{ANSWER}}$

- ASIDE - if that last problem had initial conditions, *e.g.*, $w(t_0) = a$, $w'(t_0) = b$, and $w''(t_0) = c$, they would become $\begin{pmatrix} x_1(t_0) \\ x_2(t_0) \\ x_3(t_0) \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$.

Some General Theory of Linear Systems of First Order ODEs

$$\boxed{\dot{\vec{x}}(t) = P(t)\vec{x}(t) + \vec{g}(t)}$$

↪ As with n^{th} order linear single ODEs, we can find a *general solution* to the **inhomogeneous** system of n first order linear ODEs $\dot{\vec{x}}(t) = P(t)\vec{x}(t) + \vec{g}(t)$ by

1. finding a *general solution to the **homogeneous** system* $\dot{\vec{x}}(t) = P(t)\vec{x}(t)$,
2. then finding *ANY particular solution to the full system* $\dot{\vec{x}}(t) = P(t)\vec{x}(t) + \vec{g}(t)$ (using a systems version of the Method of Undetermined Coefficients or other techniques to be discussed later), and
3. adding the two solutions from steps (1) and (2).

- Partial Proof: I'll leave you to show that if $\vec{x_c}(t)$ is a solution to $\dot{\vec{x}}(t) = P(t)\vec{x}(t)$ and $\vec{x_p}(t)$ is a solution to $\dot{\vec{x}}(t) = P(t)\vec{x}(t) + \vec{g}(t)$, then $\vec{x_c}(t) + \vec{x_p}(t)$ is a solution to $\dot{\vec{x}}(t) = P(t)\vec{x}(t) + \vec{g}(t)$.

Therefore, as with n^{th} order single equations, we will concentrate first on finding *general solutions* to the **homogeneous system**, $\dot{\vec{x}}(t) = P(t)\vec{x}(t)$, then we will spend (a little) time learning how to find particular solutions to the **non-homogeneous system**, $\dot{\vec{x}}(t) = P(t)\vec{x}(t) + \vec{g}(t)$ - at least, in the special case of a *constant* matrix $P(t) = A$.

- NOTE as for homogeneous linear ODEs, *the principle of superposition* applies to systems of first order linear ODEs: If the vector functions $\vec{x}_1(t)$ and $\vec{x}_2(t)$ are solutions to $\boldsymbol{\vec{x}\,' = P(t)\vec{x}}$, then so is the linear combination $c_1\vec{x}_1 + c_2\vec{x}_2$ for any constants c_1 and c_2.
  (An equivalent pair of statements is that if x_1 and x_2 are solutions to $\vec{x}\,' = P(t)\vec{x}$ and c_1 is any constant, then BOTH $\boxed{\mathbf{1}}$ *$x_1 + x_2$ AND* $\boxed{\mathbf{2}}$ *$c_1 x_1$ are also solutions to $\vec{x}\,' = P(t)\vec{x}$).*
- $\boxed{\text{EXERCISE:}}$ Prove this statement.
- By applying the principle of superposition repeatedly, we see that if $\boxed{\vec{x}_1(t),\ \vec{x}_2(t),\ \ldots,\ \vec{x}_k(t)}$ is any finite set of solutions to the homogeneous system $\vec{x}\,' = P(t)\vec{x}$, then so is any finite linear combination of those solutions, $\boxed{c_1\vec{x}_1(t) + c_2\vec{x}_2(t) + \ldots + c_k\vec{x}_k(t)}$.
- It can be shown that for a general solution of n linear homogeneous ODEs $\vec{x}\,' = P(t)\vec{x}$ on an interval $\alpha < t < \beta$, we need exactly n **LINEARLY INDEPENDENT** solution vector functions $\vec{x}_1(t), \vec{x}_2(t), \ldots, \vec{x}_n(t)$ on that same interval $\alpha < t < \beta$. In that case, the general solution of $\vec{x}\,' = P(t)\vec{x}$ is

$$\boldsymbol{\vec{x}(t) = c_1\vec{x}_1(t) + c_2\vec{x}_2(t) + \ldots + c_n\vec{x}_n(t).}$$

$\hookrightarrow$ Recall *(see Supplementary Lecture on Eigenvalues/Eigenvectors)* that we can check if a set of n dimensional vectors is linearly independent by forming the matrix A whose columns consist of those n vectors, then checking that $\det(A) \neq 0$.

$\hookrightarrow$ We can do something similar for vector functions $\vec{x}_1(t)$, $\vec{x}_2(t)$, ..., $\vec{x}_n(t)$: Form the matrix $X(t)$ whose columns consist of the vector functions $\vec{x}_1(t)$, $\vec{x}_2(t)$, ..., $\vec{x}_n(t)$ and if for a fixed t value, $t = t_0$ we have that $\det(X(t_0))$ - also denoted $W[\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n](t_0)$ and called the **WRONSKIAN** of the vector functions $\vec{x}_1(t)$, $\vec{x}_2(t)$, ..., $\vec{x}_n(t)$ at $t = t_0$ - is NOT $= 0$, then the vector functions $\vec{x}_1(t)$, $\vec{x}_2(t)$, ..., $\vec{x}_n(t)$ are **linearly independent** at $t = t_0$.

$\hookrightarrow$ Furthermore, in fact if t_0 is *any* point in the interval $\alpha < t < \beta$ on which $\vec{x}_1(t)$, $\vec{x}_2(t)$, ..., $\vec{x}_n(t)$ are solutions to $\vec{x}\,' = P(t)\vec{x}$, then EITHER

$$W[\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n](t) \equiv 0 \quad \forall\ \alpha < t < \beta$$

OR

$$W[\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n](t) \neq 0 \quad \forall\ \alpha < t < \beta.$$

$\mapsto$ In other words, if we have n solutions to $\vec{x}\,' = P(t)\vec{x}$ on the interval $\alpha < t < \beta$, we need only evaluate the Wronskian of those n solution vector functions at **ONE** point in the interval $\alpha < t < \beta$ to find out if they are linearly independent or linearly dependent.

- In the approach that we will use for solving the homogeneous system $\vec{x}\,' = A\vec{x}$ (where $A_{n\times n}$ is a constant matrix), we will ensure that the n solutions we obtain are **linearly independent** from the outset - and hence that their linear combination forms a *general solution* to the system $\vec{x}\,' = A\vec{x}$.

- So there will be no real need to check the linear independence of our solution vector functions using the Wronskian.

Solving the homogeneous constant coefficient system $\vec{x}\,' = A\vec{x}$

- Here we are assuming that $A_{n\times n}$ is a constant matrix. Note that when $n = 1$ we have the simple first order system $x' = ax$ whose solution is $x(t) = ce^{at}$, where c is an arbitrary constant.
- Recall more generally for ODEs such as $ax'' + bx' + cx = 0$ (and higher order linear constant coefficient homogeneous ODEs), we assumed solutions of the form $x(t) = ce^{rt}$ and used a *method of undetermined coefficients approach* to determine what appropriate values for r were (treating c as an arbitrary constant which could be different for different values of r).
  - Using a similar logic as in that 2^{nd} (or higher) order single homogeneous linear ODE case, we assume solutions to $\vec{x}\,' = A\vec{x}$ of the form $\boldsymbol{\vec{x}(t) = \vec{c}e^{rt}}$ where the constant vector $\vec{c} = [c_1, c_2, \ldots, c_n]^T$ and the exponent r are to be determined.
  - As always, we proceed by differentiating $\vec{x}(t) = \vec{c}e^{rt}$ and substituting into the ODE system $\vec{x}\,' = A\vec{x}$.

REMINDER: Solving $\vec{x}\,' = A\vec{x}$ by assuming $\vec{x}(t) = \vec{c}e^{rt}$

↪ So $\vec{x}\,' = r\vec{c}e^{rt}$, and substituting into the ODE system we get

$$r\vec{c}e^{rt} = A\vec{c}e^{rt} \quad \Rightarrow \quad r\vec{c} = A\vec{c} \ \ \text{(since } e^{rt} > 0 \ \forall \ t, \text{we can divide by it).}$$

$$\text{Equivalently,} \quad (A - r\mathrm{I})\vec{c} = \vec{0}.$$

⇝ So the r values we seek are exactly the **EIGENVALUES** of A and the $\vec{c}$ values we seek are the corresponding **EIGENVECTORS** of A.

↪ To get a general solution of $\vec{x}\,' = A\vec{x}$ if $A_{n\times n}$, we will need n **linearly independent** eigenvectors $\vec{c}_1$, $\vec{c}_2$, ..., $\vec{c}_n$ along with their corresponding eigenvalues r_1, r_2, ..., r_n. In that case, it's easy to show that the solution vector functions $\vec{c}_1e^{r_1t}$, $\vec{c}_2e^{r_2t}$, ..., $\vec{c}_ne^{r_nt}$ must also be linearly independent. And therefore a general solution to $\vec{x}\,' = A\vec{x}$ would be

$$\boldsymbol{\vec{x}(t) = B_1\vec{c}_1e^{r_1t} + B_2\vec{c}_2e^{r_2t} + \ldots + B_n\vec{c}_ne^{r_nt}},$$

where $B_1, B_2, \ldots, B_n$ are arbitrary constants.

*NOTE it does not matter if there are repeated eigenvalues, i.e. $r_i = r_j$ for some $1 \le i < j \le n$, PROVIDED there are n **linearly independent** eigenvectors (see also EXAMPLES 4, 5, 6 and 10 of the Supplementary Lecture on Eigenvalues/Eigenvectors).*

- So in summary, to find the general solution to $\boldsymbol{\vec{x}\,' = A\vec{x}}$, we find the **eigenvalues** and corresponding **eigenvectors** of A.

- For each eigenvalue-eigenvector pair r_i, $\vec{c}_i$ (so that $A\vec{c}_i = r_i\vec{c}_i$), $1 \le i \le n$, the vector function $B_i\vec{c}_i e^{r_i t}$ is a solution to $\boldsymbol{\vec{x}\,' = A\vec{x}}$ (where B_i is an arbitrary constant).

- If we find n **linearly independent** eigenvectors $\vec{c}_1, \vec{c}_2, \ldots, \vec{c}_n$ with corresponding eigenvalues $r_1, r_2, \ldots, r_n$, then a general solution to $\boldsymbol{\vec{x}\,' = A\vec{x}}$ is

$$\boldsymbol{\vec{x}(t) = B_1\vec{c}_1 e^{r_1 t} + B_2\vec{c}_2 e^{r_2 t} + \ldots + B_n\vec{c}_n e^{r_n t}}, \tag{2}$$

  where $B_1, B_2, \ldots, B_n$ are arbitrary constants.

- So, for example, if $A_{n\times n}$ has n different eigenvalues, then we easily get a general solution of the form of Equation (2).

- So our solution will depend on the eigenvalues (distinct, repeated, complex) and most importantly on the *number of linearly independent eigenvectors* (repeated eigenvalues case only) that we get. We will consider all of the relevant scenarios in the following examples.

Finding Eigenvalues and Eigenvectors Using Matlab

- Although in the following examples I will show some of the details of how to find the eigenvalues and eigenvectors, for practical purposes it will typically be more convenient to calculate these using Matlab's in-built **eig()** function.
- Crucially, **eig()** will also give a complete diagonalisation of a *diagonalisable* square matrix A $\rightsquigarrow$ *i.e.*, it will find an invertible matrix P such that $P^{-1}AP = D$, a diagonal matrix with the eigenvalues of A on the main diagonal.

$\mapsto$ If A is an $n \times n$ matrix, then

\>> eig(A) returns a column vector with the eigenvalues of A.

\>> [P, D] = eig(A) returns a diagonal matrix D *(or whatever else you want to call it)* with the eigenvalues of A on its main diagonal, and an invertible matrix P *(or whatever else you want to call it)* whose columns are normalised (scaled so that have length 1) eigenvectors of A in the same order as the corresponding eigenvalues in D. Thus $P^{-1}AP = D$.

- For example *(see also EXAMPLE 5 next)*, with $A = \begin{bmatrix} 4/3 & -1 \\ -2/3 & 1 \end{bmatrix}$, then

$>>$ eig(A) returns $\begin{bmatrix} 2.0000 \\ 0.3333 \end{bmatrix}$ and

$>>$ [P, D] = eig(A) returns

$$P = \begin{bmatrix} 0.8321 & 0.7071 \\ -0.5547 & 0.7071 \end{bmatrix} \quad \text{and} \quad D = \begin{bmatrix} 2.0000 & 0 \\ 0 & 0.3333 \end{bmatrix}$$

- I'll leave you to figure out how you could scale the eigenvectors (columns of P) to have eigenvectors with whole number entries.
- Type `help eig` in the Matlab command window for more information on eig().

$\boxed{\textbf{CASE 1: Solving } \vec{x}\,' = A\vec{x},\ A_{n\times n} \textbf{ with } n \textbf{ different eigenvalues}}$

$\hookrightarrow$ $\boxed{\text{EXAMPLE 5}}$ Consider the (modified form of the) homogeneous system from EXAMPLE 2, $\frac{d}{dt}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 4/3 & -1 \\ -2/3 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}$. Find a general solution.

First we find the eigenvalues and corresponding eigenvectors of $\begin{bmatrix} 4/3 & -1 \\ -2/3 & 1 \end{bmatrix}$

So $\lambda_1 = \frac{1}{3}$ and $\lambda_2 = 2$ are the two (different) eigenvalues of $A = \begin{bmatrix} 4/3 & -1 \\ -2/3 & 1 \end{bmatrix}$.

REMINDER: Solving $\frac{d}{dt}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 4/3 & -1 \\ -2/3 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}$, eigenvalues $\frac{1}{3}, 2$.

- To find an eigenvector $\vec{c} = [c_1, c_2]^T$ corresponding to eigenvalue $\lambda_1 = \frac{1}{3}$, we solve $(A - \frac{1}{3}\mathrm{I})\vec{c} = \vec{0}$, or equivalently

$$\begin{bmatrix} 1 & -1 \\ -2/3 & \frac{2}{3} \end{bmatrix}\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Rightarrow \quad c_1 = c_2 \text{ and } \vec{c} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

is an eigenvector $(c_2 = 1)$ corresponding to eigenvalue $\lambda_1 = \frac{1}{3}$.

- Similarly,

$$\vec{c} = \begin{bmatrix} 3 \\ -2 \end{bmatrix}$$

is an eigenvector corresponding to $(c_1 = 3)$ eigenvalue $\lambda_2 = 2$.

- And a general solution is $\boldsymbol{\vec{x}(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = B_1 \begin{bmatrix} 1 \\ 1 \end{bmatrix} e^{\frac{1}{3}t} + B_2 \begin{bmatrix} 3 \\ -2 \end{bmatrix} e^{2t}}$, where B_1 and B_2 are arbitrary constants.

- $\boxed{\text{EXAMPLE 6}}$ Recall from EXAMPLE 2 of the *Supplementary Lecture on Eigenvalues/Eigenvectors* that $A = \begin{bmatrix} 2 & 5 \\ 6 & 1 \end{bmatrix}$, has eigenvalue $\lambda_1 = 7$ with corresponding eigenvector $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and eigenvalue $\lambda_2 = -4$ with corresponding eigenvector $\begin{bmatrix} -5 \\ 6 \end{bmatrix}$.
  Use this information to find a general solution to $\dot{\vec{x}}(t) = A\vec{x}$.

- $\boxed{\text{ANSWER:}}$

$\hookrightarrow$ $\boxed{\text{EXAMPLE 7}}$ Similarly, from EXAMPLE 3 of the *Supplementary Lecture on Eigenvalues/Eigenvectors*, since for $A = \begin{bmatrix} 2 & 0 & 0 \\ -4 & -5 & 0 \\ 1 & 0 & 4 \end{bmatrix}$ we have eigenvalues $2, -5$, and 4 with corresponding eigenvectors $\begin{bmatrix} -14 \\ 8 \\ 7 \end{bmatrix}$, $\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$, and $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ respectively, then a general solution to $\dot{\vec{x}} = A\vec{x}$ is

- NOTE since the coefficient matrix A in $\dot{\vec{x}} = A\vec{x}$ is (lower) triangular, we can also solve this system by solving for $x_1(t)$, then substituting that into the second equation and solving for $x_2(t)$, then substituting those two solutions into the third equation and solving for $x_3(t)$.

$\hookrightarrow$ EXAMPLE 8 Solve the system $\begin{bmatrix} x \\ y \\ z \end{bmatrix}' = \begin{bmatrix} 3 & 0 & 0 \\ 0 & -\frac{1}{4} & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$

$\hookrightarrow$ ANSWER

$\hookrightarrow$ NOTE it will always be similarly easy to solve systems of the form $\vec{x}\,' = D\vec{x}$ where D is a diagonal matrix. With this in mind, I'll present an alternative way of viewing the solution of a more general $\vec{x}\,' = A\vec{x}$ which uses the diagonalisation of A (if it exists) and is useful when solving non-homogeneous systems.

$\hookrightarrow$ This leads to another useful way of viewing the solution of $\vec{x}\,' = A\vec{x}$ *when $A_{n\times n}$ has n linearly independent eigenvectors*, which involves the **diagonalisation** of A.

Although this approach is equivalent to the one we have been using so far, it has the advantage of being very useful when solving NON-HOMOGENEOUS problems (so we don't have to use the Method of Undetermined Coefficients or other approaches).

$\hookrightarrow$ SOLVING $\dot{\vec{x}} = A\vec{x}$ USING THE DIAGONALISATION OF A:

- Assuming A can be diagonalised so that $P^{-1}AP = D$ is a diagonal matrix (see the *Supplementary Lecture on Eigenvalues/Eigenvectors* for details), then $\boldsymbol{A = PDP^{-1}}$.
- So $\vec{x}\,' = A\vec{x}$ can be written as

$$\begin{aligned}
\vec{x}\,' &= PDP^{-1}\vec{x} \Rightarrow \\
P^{-1}\vec{x}\,' &= DP^{-1}\vec{x} \Rightarrow \\
(P^{-1}\vec{x})\,' &= D(P^{-1}\vec{x}) \quad \text{since } P \text{ is a constant matrix.}
\end{aligned}$$

$\hookrightarrow$ This last equation involves an unknown vector $P^{-1}\vec{x}$ and its derivative, and as in the last example the coefficient matrix D is *DIAGONAL*. Hence it is easy to solve for $P^{-1}\vec{x}$ - we just solve each equation in the system separately!
(If you have difficulty seeing this as a "diagonal" linear homogeneous system of ODEs, make the substitution $\vec{y} = P^{-1}\vec{x}$ and then the system is $\vec{y}\,' = D\vec{y}$ where D is a diagonal matrix).

$\hookrightarrow$ Having solved for $P^{-1}\vec{x}$ and *written the solution in vector form*, then to find the solution, $\vec{x}$, to the original problem we simply multiply $P^{-1}\vec{x}$ on the left by P.

$\hookrightarrow$ $\boxed{\text{EXAMPLE 9}}$ Redo EXAMPLE 5 using the diagonalisation approach: Solve $\frac{d}{dt}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 4/3 & -1 \\ -2/3 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}$.

$\hookrightarrow$ $\boxed{\text{EXAMPLE 10}}$ Similarly, redo EXAMPLE 7 with the diagonalisation approach:

CASE 2: Solving $\vec{x}\,' = A\vec{x}$, $A_{n\times n}$ with complex eigenvalues

(but still n linearly independent eigenvectors)

$\hookrightarrow$ Approach will be similar to how we extract real valued (linearly independent) solutions to $ax'' + bx' + cx = 0$ when the characteristic polynomial $ar^2 + br + c$ had (non-real) complex roots *(ignore this comment if you have not solved second order constant coefficient linear differential equations before)*.

$\hookrightarrow$ First observe that if A is a matrix with real entries, then when finding eigenvalues the characteristic equation $\det(A - \lambda \mathrm{I}) = 0$ has only real coefficients so all (non-real) complex eigenvalues occur in *conjugate pairs*: $\lambda = a \pm bi$.
It is not difficult to show that the corresponding eigenvectors also occur in conjugate pairs. So if $\vec{x}$ is an eigenvector corresponding to eigenvalue $a + ib$, then $\overline{\vec{x}}$ is an eigenvector corresponding to the eigenvalue $a - ib$ *($\overline{\vec{x}}$ denotes the vector whose entries are the complex conjugates of the corresponding entries in $\vec{x}$)*.
$\boxed{\text{PROOF:}}$ $A\vec{x} = (a + ib)\vec{x}$ so taking the complex conjugate of both sides of this equation (recalling that A has only real entries so $\overline{A} = A$ and that $\overline{\alpha\beta} = \overline{\alpha}\,\overline{\beta}$), then $A\overline{\vec{x}} = (a - ib)\overline{\vec{x}}$.

↪ So let's see how to extract real solutions from solutions which involve eigenvalue $\lambda_1 = a + ib$ with corresponding eigenvector $\vec{u} + i\vec{v}$ *(where $\vec{u}$ and $\vec{v}$ have real entries only)* and the complex conjugate eigenvalue $\lambda_2 = a - ib$ with corresponding eigenvector $\vec{u} - i\vec{v}$.

- Recall that in general if r is an eigenvalue of A and $\vec{c}$ is the corresponding eigenvector, then $\vec{c}e^{rt}$ is a solution to $\vec{x}\,' = A\vec{x}$.
  Applying that in this case, we have (complex-valued) solutions $(\vec{u} + i\vec{v})e^{at+ibt}$ and $(\vec{u} - i\vec{v})e^{at-ibt}$.

⇝ Clearly, since both complex-valued vector functions above are solutions to $\vec{x}\,' = A\vec{x}$ *then so are their real and imaginary parts*. Without loss of generality,

$$(\vec{u} + i\vec{v})e^{at+ibt} = (\vec{u} + i\vec{v})(e^{at}\cos bt + ie^{at}\sin bt) =$$

$$\boldsymbol{e^{at}(\vec{u}\cos bt - \vec{v}\sin bt) + i\; e^{at}(\vec{u}\sin bt + \vec{v}\cos bt).}$$

So the two vector functions $e^{at}(\vec{u}\cos bt - \vec{v}\sin bt)$ and $e^{at}(\vec{u}\sin bt + \vec{v}\cos bt)$ are *real-valued* solutions to $\vec{x}\,' = A\vec{x}$. Furthermore, it can be shown that they are linearly independent (so we have **2** linearly independent eigenvectors to go along with the **2** complex-conjugate eigenvalues $a \pm ib$).

↪ $\boxed{\text{SUMMARY}}$ `Instead of trying to remember the previous result as a formula, just use the following steps to get 2 linearly independent solutions from two complex conjugate eigenvalues` $a \pm bi$ `to the matrix` A `when solving` $\vec{\boldsymbol{x}}\,' = \boldsymbol{A}\vec{\boldsymbol{x}}$:

1. Pick *ONE* of the eigenvalues, $\lambda_1 = a + ib$ or $\lambda_2 = a - ib$, and find the corresponding eigenvector $\vec{v}$ by solving $(\boldsymbol{A} - \lambda \mathrm{I})\vec{\boldsymbol{v}} = \vec{\boldsymbol{0}}$ for $\vec{v}$.
2. A COMPLEX-VALUED solution to $\vec{\boldsymbol{x}}\,' = \boldsymbol{A}\vec{\boldsymbol{x}}$ would then be $\boxed{\vec{\boldsymbol{u}} = \boldsymbol{e}^{\lambda_1 t}\vec{\boldsymbol{v}}}$ or $\boxed{\vec{\boldsymbol{u}} = \boldsymbol{e}^{\lambda_2 t}\vec{\boldsymbol{v}}}$ depending on which eigenvalue, λ_1 or λ_2, was used in Step 1.
3. Write the *complex-valued* solution, $\vec{u}$, from Step 2 in terms of its real and imaginary parts: $\vec{\boldsymbol{u}} = \vec{\boldsymbol{u_1}} + \boldsymbol{i}\vec{\boldsymbol{u_2}}$, where both $\vec{u_1}$ and $\vec{u_2}$ are vectors containing only real-valued entries.
4. $\vec{\boldsymbol{u_1}}$ and $\vec{\boldsymbol{u_2}}$ will be two *linearly independent* REAL-VALUED solutions to $\vec{\boldsymbol{x}}\,' = \boldsymbol{A}\vec{\boldsymbol{x}}$.

So, for example, any general solution to $\vec{\boldsymbol{x}}\,' = \boldsymbol{A}\vec{\boldsymbol{x}}$ would include the terms $\boldsymbol{B_1}\vec{\boldsymbol{u_1}} + \boldsymbol{B_2}\vec{\boldsymbol{u_2}}$, where B_1 and B_2 are arbitrary constants.

$\hookrightarrow$ $\boxed{\text{EXAMPLE 11}}$ Solve the initial value problem $\vec{x}\,' = A\vec{x}$, $\vec{x}(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, where $A = \begin{bmatrix} 1 & -5 \\ 1 & -3 \end{bmatrix}$.

- First to find the eigenvalues we solve $\begin{vmatrix} 1-\lambda & -5 \\ 1 & -3-\lambda \end{vmatrix} = 0$ $\Rightarrow (1-\lambda)(-3-\lambda)+5 = 0$ or $\lambda^2 + 2\lambda + 2 = 0 \Rightarrow$

$$\boldsymbol{\lambda = -1 \pm i.}$$

Next, let's select the eigenvalue $\lambda = -1+i$ and find a corresponding eigenvector: We solve the equation

$\begin{bmatrix} 2-i & -5 \\ 1 & -2-i \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. *BOTH* equations $\Rightarrow$ $v_1 = (2+i)v_2$ so $\begin{bmatrix} 2+i \\ 1 \end{bmatrix}$

is an eigenvector (setting $v_2 = 1$) corresponding to eigenvalue $\lambda = -1+i$.
So a complex valued solution to $\boldsymbol{\vec{x}\,' = A\vec{x}}$ is

$$e^{(-1+i)t} \begin{bmatrix} 2+i \\ 1 \end{bmatrix} = e^{-t}(\cos t + i \sin t) \begin{bmatrix} 2+i \\ 1 \end{bmatrix} = e^{-t} \begin{bmatrix} 2\cos t - \sin t \\ \cos t \end{bmatrix} + i\, e^{-t} \begin{bmatrix} 2\sin t + \cos t \\ \sin t \end{bmatrix}$$

So the general solution to $\boldsymbol{\vec{x}\,' = A\vec{x}}$ is

$$\boldsymbol{\vec{x}(t) = B_1 e^{-t} \begin{bmatrix} 2\cos t - \sin t \\ \cos t \end{bmatrix} + B_2 e^{-t} \begin{bmatrix} 2\sin t + \cos t \\ \sin t \end{bmatrix}}$$ (B_1, B_2 arbitrary constants).

REMINDER: General solution

$$\vec{x}(t) = B_1 e^{-t}\begin{bmatrix} 2\cos t - \sin t \\ \cos t \end{bmatrix} + B_2 e^{-t}\begin{bmatrix} 2\sin t + \cos t \\ \sin t \end{bmatrix}$$

Next, we use the initial condition $\vec{x}(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ to find B_1 and B_2. We get

$$\begin{bmatrix} 2B_1 + B_2 \\ B_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\Rightarrow B_1 = 1 \quad \text{and} \quad B_2 = -1.$$

So the solution to the initial value problem is

$$\vec{x}(t) = e^{-t}\begin{bmatrix} 2\cos t - \sin t \\ \cos t \end{bmatrix} - e^{-t}\begin{bmatrix} 2\sin t + \cos t \\ \sin t \end{bmatrix}$$

$$= e^{-t}\begin{bmatrix} \cos t - 3\sin t \\ \cos t - \sin t \end{bmatrix}.$$

$\hookrightarrow$ $\boxed{\text{EXAMPLE 12}}$ Find the general solution of $\vec{x}\,' = A\vec{x}$, where
$A = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & -2 \\ 3 & 2 & 1 \end{bmatrix}$.

REMINDER:

CASE 3: Solving $\vec{x}\,' = A\vec{x}$, $A_{n\times n}$ with repeated eigenvalues

↪ First, if $A_{n\times n}$ still has n **linearly independent** eigenvectors, then the approach is exactly as in CASE 1 (see EXAMPLES 5–10).

↪ $\boxed{\text{EXAMPLE 13}}$ Recall from EXAMPLES 4, 9, and 11 of the *Supplementary Lecture on Eigenvalues/Eigenvectors* that the matrix $A = \begin{bmatrix} 0 & 0 & -2 \\ 1 & 2 & 1 \\ 1 & 0 & 3 \end{bmatrix}$ has eigenvalues $\lambda = 1, 2, 2$, but that the *repeated* eigenvalue of 2 had *TWO* **linearly independent** eigenvectors. So that in total A had *THREE* linearly independent eigenvectors $\vec{p}_1 = \begin{bmatrix} -2 \\ 1 \\ 1 \end{bmatrix}$, $\vec{p}_2 = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$, and $\vec{p}_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ respectively.

So a general solution to $\dot{\vec{x}} = A\vec{x}$ is

$$\vec{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} = B_1 e^{t} \begin{bmatrix} -2 \\ 1 \\ 1 \end{bmatrix} + B_2 e^{2t} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} + B_3 e^{2t} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix},$$

where B_1, B_2, and B_3 are arbitrary constants.

$\hookrightarrow$ So what really concerns us is the case where the *multiplicity* of a repeated eigenvalue of A *(= the number of times that the eigenvalue is a root of A's characteristic polynomial)* is *greater than* the number of linearly independent eigenvectors we can find for that eigenvalue - see, for example, EXAMPLE 10 of the *Supplementary Lecture on Eigenvalues/Eigenvectors*. In this case, we look for **generalised eigenvectors** corresponding to that repeated eigenvalue.

- The overall approach is similar to what we do when solving $ax'' + bx' + c = 0$ and the characteristic equation $ar^2 + br + c = 0$ has a repeated root.

- E.g. Suppose A has eigenvalue of multiplicity TWO λ with only 1 linearly independent family of eigenvectors, with $\vec{v}$ being one of them. So $A\vec{v} = \lambda\vec{v}$. In the past, we have assumed a solution to $\boldsymbol{\vec{x}\,' = A\vec{x}}$ to be of the form $e^{\lambda t}\vec{v}$. Based on what we did for n^{th} order equations whose characteristic equations had repeated roots, how do you propose we adjust the form of our solution $e^{\lambda t}\vec{v}$ if λ is an eigenvalue of multiplicity TWO?

  ANSWER:

- $\vec{x}(t) = te^{\lambda t}\vec{u} \Rightarrow \vec{x}\,'(t) = e^{\lambda t}\vec{u} + \lambda te^{\lambda t}\vec{u}$. Substituting this into $\boldsymbol{\vec{x}\,' = A\vec{x}}$, we get

$$e^{\lambda t}\vec{u} + \lambda te^{\lambda t}\vec{u} = Ate^{\lambda t}\vec{u} \quad \Rightarrow \quad \vec{u}e^{\lambda t} + \lambda\vec{u}te^{\lambda t} - A\vec{u}te^{\lambda t} = \vec{0}.$$

  This is only possible for all t if the coefficients of both $e^{\lambda t}$ and $te^{\lambda t}$ are zero vectors. In particular, we must have $\vec{u} = \vec{0}$, so there is NO non-zero vector solution (hence no eigenvector) if we assume the solution is of the form $\vec{x}(t) = te^{\lambda t}\vec{u}$.

$\hookrightarrow$ So, observing the appearance of the $\vec{u}e^{\lambda t}$ term when we substituted into the ODE system, we adjust our assumption by including lower order terms:

$$\text{Let} \quad \vec{x}(t) = te^{\lambda t}\vec{u_1} + e^{\lambda t}\vec{u_2} \tag{3}$$

where $\vec{u_1}$ and $\vec{u_2}$ are constant vectors to be determined.
$\vec{x}\,'(t) = \lambda te^{\lambda t}\vec{u_1} + e^{\lambda t}(\vec{u_1} + \lambda\vec{u_2})$ and substituting into $\boldsymbol{\vec{x}\,' = A\vec{x}}$,

$$\lambda te^{\lambda t}\vec{u_1} + e^{\lambda t}(\vec{u_1} + \lambda\vec{u_2}) = Ate^{\lambda t}\vec{u_1} + Ae^{\lambda t}\vec{u_2} \Rightarrow te^{\lambda t}(\lambda\vec{u_1} - A\vec{u_1}) + e^{\lambda t}(\vec{u_1} + \lambda\vec{u_2} - A\vec{u_2}) = \vec{0}.$$

So again, we must have $\lambda\vec{u_1} - A\vec{u_1} = \vec{0}$ AND $\vec{u_1} + \lambda\vec{u_2} - A\vec{u_2} = \vec{0}$.

REMINDER: $\lambda\vec{u_1} - A\vec{u_1} = \vec{0}$ AND $\vec{u_1} + \lambda\vec{u_2} - A\vec{u_2} = \vec{0}$.

- The first equation is, of course, equivalent to $A\vec{u_1} = \lambda\vec{u_1}$ so that $\vec{u_1}$ is simply an eigenvector of A corresponding to eigenvalue λ (so it would already be known!!!).
- The second equation is equivalent to $\boxed{(A - \lambda\mathrm{I})\vec{u_2} = \vec{u_1}}$, and a solution $\vec{u_2}$ is known as a **generalised eigenvector** of A.
- Returning to Equation (3), a solution to $\vec{x}\,' = A\vec{x}$ is $\vec{x(t)} = te^{\lambda t}\vec{u_1} + e^{\lambda t}\vec{u_2}$, where $\vec{u_1}$ is an eigenvector of A corresponding to eigenvalue λ and $\vec{u_2}$ is a *GENERALISED* eigenvector of A corresponding to eigenvalue λ. It can be shown that this solution is **linearly independent** from $\vec{x}(t) = e^{\lambda t}\vec{u_1}$.
  - NOTE 1: that $\vec{u_2}$ will typically contain a sum of vectors, one of which will be a multiple of $\vec{u_1}$. We can ignore that multiple of $\vec{u_1}$ since the term $e^{\lambda t}\vec{u_1}$ would appear elsewhere in a general solution to $\vec{x}\,' = A\vec{x}$.
  - NOTE 2: we have only discussed the case in which λ is an eigenvalue of A of multiplicity TWO. Other cases are "fairly easily" generalisable from this and will be discussed briefly after the next example.

$\hookrightarrow$ EXAMPLE 14 (a) *(Based on EXAMPLE 10 of the Supplementary Lecture on Eigenvalues/ Eigenvectors)*: Find a general solution of $\vec{x}\,' = A\vec{x}$ where $A = \begin{bmatrix} 3 & 1 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 4 \end{bmatrix}$.

By solving $\det(A - \lambda \mathrm{I}) = (3-\lambda)^2(4-\lambda) = 0$, we get eigenvalues $\boxed{\lambda_1 = 3 \text{ (multiplicity TWO)}}$ and $\boxed{\lambda_2 = 4}$. We have already seen in the *Supplementary Lecture on Eigenvalues/Eigenvectors* EXAMPLE 10 that there is only one family of linearly independent eigenvectors corresponding to $\lambda_1 = 3$, of which $\vec{u_1} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ is a representative. So we seek a *generalised eigenvector* by assuming a solution of the form $\vec{x}(t) = te^{3t}\vec{u_1} + e^{3t}\vec{u_2}$ and, upon substitution into $\boldsymbol{\vec{x}\,' = A\vec{x}}$ solving the resulting new equation

$$(A - 3I)\vec{u_2} = \vec{u_1} \quad \Rightarrow \quad \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{21} \\ u_{22} \\ u_{23} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

So $u_{23} = 0$, $u_{22} = 1$, and u_{21} can take on any (non-zero) value, so that a typical generalised eigenvector is of the form $u_{21} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$. Hence, setting $u_{21} = 1$, a generalised eigenvector is $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$.

REMINDER: Generalised eigenvector : $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$

$\hookrightarrow$ Because the first vector in this sum is simply the eigenvector $\vec{u_1}$, we ignore it and take the *generalised eigenvector* to be simply $\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$.

- *ASIDE: Alternatively, we could have simply taken $u_{21} = 0$ and gotten the generalised eigenvector, $\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$, directly.*

$\rightsquigarrow$ So from Equation (3) a solution to $\vec{\boldsymbol{x}}\,' = \boldsymbol{A}\vec{\boldsymbol{x}}$ is $\vec{x}(t) = te^{3t}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + e^{3t}\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$

Next to find an eigenvector corresponding to single eigenvalue $\lambda = 4$ we solve

$$\begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \Rightarrow \ldots \Rightarrow \quad \vec{v} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

is an eigenvector corresponding to eigenvalue $\lambda_2 = 4$. So a general solution to $\vec{\boldsymbol{x}}\,' = \boldsymbol{A}\vec{\boldsymbol{x}}$ is

$$\vec{x}(t) = B_1 e^{4t}\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + B_2 e^{3t}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + B_3\left(te^{3t}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + e^{3t}\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}\right).$$

$\hookrightarrow$ $\boxed{\text{EXAMPLE 14 (b)}}$: Find a general solution of $\vec{x}\,' = A\vec{x}$ where $A = \begin{bmatrix} 3 & -1 \\ 1 & 1 \end{bmatrix}$.

We first find the eigenvalues of A by solving $\det(A - \lambda I) = 0$, so $(3-\lambda)(1-\lambda) + 1 = 0 \Rightarrow \lambda^2 - 2\lambda + 4 = 0 \Rightarrow (\lambda - 2)^2 = 0$, so that $\boxed{\lambda = 2}$ is the (repeated) eigenvalue.

To find one or more corresponding linearly independent eigenvectors, solve $(A - 2I)\vec{x} = \vec{0}$: $\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \Rightarrow x_1 = x_2$ so $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ or any (non-zero) scalar multiple thereof is the only (family of) linearly independent eigenvector(s) associated directly with the eigenvalue $\lambda = 2$.

We therefore need to find a *generalised eigenvector* by solving

$$\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \Rightarrow u_1 - u_2 = 1 \text{ or } u_1 = 1 + u_2.$$

So any vector of the form $\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 1 + u_2 \\ u_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + u_2\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ is a generalised eigenvector of A. Specifically, since the second vector already appears in the linear span of the first eigenvector, we take just $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ as the *generalised eigenvector* (i.e, we set $u_2 = 0$). And the general solution to the system of ODEs is

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = C_1 e^{2t}\begin{pmatrix} 1 \\ 1 \end{pmatrix} + C_2 e^{2t}\begin{pmatrix} 1 \\ 0 \end{pmatrix} + C_2 t e^{2t}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

- The techniques described generalise in a fairly easy way to situations in which the multiplicity of the eigenvalue is greater than 2.
- For example if A has an eigenvalue λ of multiplicity THREE with only TWO linearly independent eigenvectors $\vec{v_1}$ and $\vec{v_2}$ corresponding to that eigenvalue, then an analysis similar to the one used to come up with Equation (3) would again lead to the conclusion that a solution of the form $\vec{x}(t) = te^{\lambda t}\vec{v} + e^{\lambda t}\vec{u} \Rightarrow (A - \lambda \mathrm{I})\vec{v} = \vec{0}$ and $(A - \lambda \mathrm{I})\vec{u} = \vec{v}$. NOTING that the most general solution to the first equation is $\vec{v} = c_1\vec{v_1} + c_2\vec{v_2}$, it will often be necessary to assign specific values to c_1 and c_2 to ensure that $(A - \lambda \mathrm{I})\vec{u} = \vec{v}$ has a solution, $\vec{u}$.
- Likewise, if A has an eigenvalue of multiplicity 3 but only one linearly independent corresponding eigenvector $\vec{v}$, then assume $\vec{x}(t) = te^{\lambda t}\vec{u_1} + e^{\lambda t}\vec{u_2}$ and follow the analysis leading up to Equation (3). THEN to get a third linearly independent solution, assume $\vec{x}(t) = \frac{1}{2}t^2e^{\lambda t}\vec{w_1} + te^{\lambda t}\vec{w_2} + e^{\lambda t}\vec{w_3}$ and repeat the analysis leading up to Equation (3) to conclude

$$(A - \lambda I)\vec{w}_1 = 0, \quad (A - \lambda I)\vec{w}_2 = \vec{w}_1, \quad \text{and} \quad \boxed{\boldsymbol{(A - \lambda I)\vec{w}_3 = \vec{w}_2}}.$$

- SEE a standard introductory ODE book, such as the one by *Boyce and DiPrima*, for more on this topic.

Inhomogeneous Systems $\vec{x}\,' = A\vec{x} + \vec{g}(t)$, $A_{n\times n}$

- There are several techniques for solving inhomogeneous systems, two of which are discussed in **APPENDIX B**.

- This is material is interesting but is not absolutely necessary for what we will be doing here which is classifying the behaviour of solutions to systems of ODEs; for that, it is enough to consider the solution of homogeneous linear (constant matrix coefficient) systems of the form

$$\frac{d\vec{x}}{dt} = A\vec{x}, \qquad A_{n\times n}.$$

- Note examples are numbered in Appendix B and the rest of this document as if Appendix B were inserted here.

End of Section

Numerical Methods for Systems of First Order ODEs

Notation and Conventions

- As always, **NOTATION** is going to be very important in what follows. **PAY CLOSE ATTENTION TO IT!**
- The use of **vector notation** and **vector functions** and **transformations** will make *very easy* the transition from studying and approximating the solution to a *single first order ODE* (and IVP) to studying and approximating the solution to a *system of first order ODEs* (and IVPs).
- In what follows, vectors will be denoted by **boldface**, e.g. $\boldsymbol{v}$, or by a vector symbol $\vec{\ }$, such as $\vec{v}$.
  - *Assume all vectors are column vectors unless otherwise stated.*
  - A function whose output is a vector will follow the above convention of having its name in boldface or with a vector symbol: eg $\vec{f}(t, y) = \begin{bmatrix} t^2 y \\ sin(t+2y) \end{bmatrix}$ or $\boldsymbol{g}(t) = \begin{bmatrix} t^2+2t-1 \\ sin(t)e^{-t} \end{bmatrix}$.

$\hookrightarrow$ We differentiate (or integrate) vector functions by differentiating (or integrating) each term individually:

$$\vec{r}(t) = \begin{bmatrix} t^3 - 4t \\ \sin t \\ e^{2t} \end{bmatrix} \quad \Rightarrow \quad \vec{r}\,'(t) \text{ or } \frac{d\vec{r}}{dt} \text{ or } \dot{\mathbf{r}}(t) = \begin{bmatrix} 3t^2 - 4 \\ \cos t \\ 2e^{2t} \end{bmatrix}.$$

$\hookrightarrow$ It is often convenient to name the component functions of a vector function or transformation with the *same name* as the vector function or transformation, *but with subscripts to indicate their position in the vector*. For example

$$\mathbf{y}(t) = \begin{pmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \end{pmatrix} \quad \text{or} \quad \mathbf{f}(t, y_1, y_2, y_3) = \begin{pmatrix} f_1(t, y_1, y_2, y_3) \\ f_2(t, y_1, y_2, y_3) \\ f_3(t, y_1, y_2, y_3) \end{pmatrix}.$$

This convention will be useful to adopt when we write general programs (Euler's, Heun's, RK(4), etc.) to solve systems of n first order ODEs.

↪ In light of the notation/conventions just established, a system of IVPs such as

$$\begin{aligned}\frac{dy_1}{dt} &= f_1(t, y_1, y_2, y_3)\\ \frac{dy_2}{dt} &= f_2(t, y_1, y_2, y_3)\\ \frac{dy_3}{dt} &= f_3(t, y_1, y_2, y_3)\end{aligned}$$

with $t \in [t_o, T]$ and $y_1(t_0) = y_{1,0}$, $y_2(t_0) = y_{2,0}$, and $y_3(t_0) = y_{3,0}$, can be written in vector form as

$$\frac{d\mathbf{y}}{dt} \text{ or } \dot{\mathbf{y}}(t) \text{ or } \vec{y}\,'(t) \quad = \quad \vec{f}(t, \vec{y}) \text{ or } \vec{f}(t, y_1, y_2, y_3)$$

with $t \in [t_o, T]$ and $\vec{y}(t_0) = \vec{y}_0 = \begin{pmatrix} y_{1,0} \\ y_{2,0} \\ y_{3,0} \end{pmatrix}$, where

$$\vec{y}(t) = \begin{pmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \end{pmatrix} \quad \text{and} \quad \vec{f}(t, \vec{y}) = \begin{pmatrix} f_1(t, y_1, y_2, y_3) \\ f_2(t, y_1, y_2, y_3) \\ f_3(t, y_1, y_2, y_3) \end{pmatrix}.$$

General First Order System of n IVPs

↪ In discussing numerical methods for systems of ODEs, we will focus on the **general first order system of n IVPs**:

↪ Find $\vec{y}(t)$ such that ↩

$$\frac{d\vec{y}}{dt} = \vec{f}(t, \vec{y}), \quad \forall\ t \in [t_0, T]$$

where $\vec{y}(t_0) = \vec{y}_0$ is a given initial value of the unknown *vector* function, $\vec{y}(t)$,

and $\vec{f}(t, \vec{y}) = \vec{f}(t, \vec{y}(t))$ is a given *vector* transformation.

Here $\vec{y}(t) = \begin{pmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_n(t) \end{pmatrix}$, $\vec{y}_0 = \begin{pmatrix} y_1(t_0) \\ y_2(t_0) \\ \vdots \\ y_n(t_0) \end{pmatrix}$, and $\vec{f}(t, \vec{y}) = \vec{f}(t, y_1, y_2, \ldots, y_n) = \begin{pmatrix} f_1(t, y_1, y_2, \ldots, y_n) \\ f_2(t, y_1, y_2, \ldots, y_n) \\ \vdots \\ f_n(t, y_1, y_2, \ldots, y_n) \end{pmatrix}$.

- NOTE the similarity to the single first order IVP.
- *Often, $\vec{t}_0$ will be $\vec{0}$, and we will focus on the $n = 2$ and $n = 3$ cases.*

General First Order System of n IVPs - Autonomous Systems

↪ Since many of the systems we look at will also be **autonomous**, here is the **general first order system of n IVPs** for that special case:

↪ Find $\vec{y}(t)$ such that ↩

$$\frac{d\vec{y}}{dt} = f(\vec{y}), \quad \forall\ t \in [t_0, T]$$

where $\vec{y}(t_0) = \vec{y}_0$ is a given initial value of the unknown *vector* function, $\vec{y}(t)$,

and $\vec{f}(\vec{y}) = f(\vec{y}(t))$ is a given *vector* transformation.

$$\text{Here } \vec{y}(t) = \begin{pmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_n(t) \end{pmatrix}, \quad \vec{y}_0 = \begin{pmatrix} y_1(t_0) \\ y_2(t_0) \\ \vdots \\ y_n(t_0) \end{pmatrix}, \quad \textit{and} \quad \vec{f}(\vec{y}) = \vec{f}(y_1, y_2, \ldots, y_n) = \begin{pmatrix} f_1(y_1, y_2, \ldots, y_n) \\ f_2(y_1, y_2, \ldots, y_n) \\ \vdots \\ f_n(y_1, y_2, \ldots, y_n) \end{pmatrix}.$$

- NOTE the similarity to the single first order **autonomous** IVP *BUT* also NOTE that as for single ODEs *we will solve these systems using programs written for the general case on the preceding slide.*

Euler's Method for Systems of First Order ODEs

The main reason why we made such a fuss about expressing all of our IVPs in vector form is that **the equations for the different approximation methods** *(giving the formula for Y_{i+1})* **remain the same**, with

- operations like $+$, $-$ ***BECOMING*** vector $+$, vector $-$ where appropriate,
- multiplication by $\boldsymbol{h}$ ***BECOMING*** *scalar multiplication* by $\boldsymbol{h}$.

I will show next why this is true for Euler's method by deriving the method for the special case of 2 ODEs, using Taylor series of the two solution functions, similar to what we did in **Lecture 2** when deriving Euler's method for single ODEs.

- Suppose we want to use Euler's method to approximate the solutions to

$$\begin{aligned} \frac{dy_1}{dt} &= f_1(t_1, y_1, y_2) \\ \frac{dy_2}{dt} &= f_2(t_1, y_1, y_2), \quad t \in [t_0, T], \quad y_1(t_0) = y_{1,0}, \quad y_2(t_0) = y_{2,0}. \end{aligned}$$

- Then, using Taylor series,

$$\begin{aligned} y_1(t_{i+1}) = y_1(t_i + h) &= y_1(t_i) + hy_1'(t_i) + O(h^2) \approx y_1(t_i) + hf_1(t_i, y_1(t_i), y_2(t_i)) \\ y_2(t_{i+1}) = y_2(t_i + h) &= y_2(t_i) + hy_2'(t_i) + O(h^2) \approx y_2(t_i) + hf_2(t_i, y_1(t_i), y_2(t_i)) \end{aligned}$$

$\hookrightarrow$ If we replace the functions by their approximations, we get the *systems version of Euler's method (using SUPERSCRIPTS to indicate the timestep (iteration) number and SUBSCRIPTS to indicate the function number)*:

$$\begin{aligned} Y_1^{(i+1)} &= Y_1^{(i)} + hf_1(t_i, Y_1^{(i)}, Y_2^{(i)}) \\ Y_2^{(i+1)} &= Y_2^{(i)} + hf_2(t_i, Y_1^{(i)}, Y_2^{(i)}) \end{aligned}$$

$$\boxed{\begin{array}{c} \text{or } \vec{Y}^{(i+1)} = \vec{Y}^{(i)} + h\vec{f}(t_i, Y_1^{(i)}, Y_2^{(i)}) \\ \\ \text{where } \vec{Y} = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} \text{ and } \vec{f}(t, Y_1, Y_2) = \begin{bmatrix} f_1(t, Y_1, Y_2) \\ f_2(t, Y_1, Y_2) \end{bmatrix} \end{array}}$$

Euler's Method for Systems

Euler's method for approximating the solution to the *general first order system of n IVPs*, $\frac{d\vec{y}}{dt} = \vec{f}(t, \vec{y}), \ \ \forall \ t \in [t_0, T], \ \ \vec{y}(t_0) = \vec{y}_0$:

Euler's Method for Systems

$$\boxed{\begin{gathered} \boldsymbol{\vec{Y}_0 = \vec{y}(t_0)} \text{ THEN} \\ \boldsymbol{\vec{Y}^{(i+1)} = \vec{Y}^{(i)} + h\vec{f}(t_i, \vec{Y}^{(i)})} \text{ for } \boldsymbol{i = 0, 1, 2 \ldots, N-1}. \end{gathered}}$$

where

$$\vec{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} \text{ and } \vec{f}(t, \vec{Y}) = \vec{f}(t, Y_1, Y_2, \ldots, Y_n) = \begin{bmatrix} f_1(t, Y_1, Y_2, \ldots, Y_n) \\ f_2(t, Y_1, Y_2, \ldots, Y_n) \\ \vdots \\ f_n(t, Y_1, Y_2, \ldots, Y_n) \end{bmatrix}$$

and $Y_i^{(j)}$ is the Euler approximation to $y_i(t_j)$ (for $i = 1, 2, \ldots, n$ and $j = 0, 1, 2, \ldots, N$).

Reminder: $\vec{Y}^{(i+1)} = \vec{Y}^{(i)} + h\vec{f}(t_i, \vec{Y}^{(i)})$ for $i = 0, 1, 2 \ldots, N-1$

In summary, Euler's method for a first order system of ODEs simply consists of applying the *scalar* Euler's method to a vector of differential equations one component at a time.

- I include in the following pages a sample Euler's method program for a system of two differential equations. Modifying it for a system of 3 or more equations and for Heun's method and the Runge-Kutta (fourth order) methods is relatively straightforward. NOTE a somewhat more sophisticated version will be also provided on the course Moodle page.

```
clear
clf

f = @(t,y) [-4*y(1)-2*y(2) + cos(t) + 4*sin(t);
            3*y(1)+y(2)-3*sin(t)];

% Here we give the exact solution if known. If not known, set to the
% this to return an appropriately-sized vector of zeros and ignore all
% subsequent references to the exact solution in this program
exact = @(t) [2*exp(-t) - 2*exp(-2*t) + sin(t);
                -3*exp(-t) + 2*exp(-2*t)];

n = input('Enter the number of equations in your system of ODEs ');

h=0.1;
t0 = 0; tN = 2;
y0 = [0; -1];

if length(y0) ~= n
    disp('Error, you entered an incorrect number of equations. Try again ')
    return;
end
```

```
t = [t0:h:tN];
sizet = length(t);


% y(i,j) is the approximation to y_i(t_j)
y = zeros(n,sizet);
yexact = zeros(n, sizet);
for (k = 1:n)
     y(k,1) = y0(k);
end

% Main Euler's method loop
for k = 2:sizet
  y(:,k) = y(:,k-1) + h*f(t(k-1), y(:,k-1));
end

for(k = 1:sizet)
  yexact(:,k) = exact(t(k));
end


for (mm = 1:n)
```

```
        fprintf('\nPRINTING INFORMATION FOR FUNCTION %d\n\n',mm);
        fprintf('  i   TIME        Yi (APPROX)     y(ti) (EXACT)     ABS. ERROR\n')
        for k = 1:sizet
fprintf('%3d   %8f   %10f       %10f         %10f\n',k-1,t(k),y(mm,k),yexact(mm,k), abs(y(mm,k)-yexact(mm,k)))
        end
    end

    disp(''); % blank line

    plotsoln=input('Hit return for graphs of solutions versus time ')
    if isempty(plotsoln)
        set(gca,'fontsize',14)
        for k = 1:n
            plot(t,y(k,:),'linewidth',2)
            xlabel('t')
            fprintf('\nPLOTTING INFORMATION FOR FUNCTION %d\n\n',k);
            if k < n
                disp('Hit any key to see the next graph ');
                pause
            end
        end
    end
```

```
disp(' ') % blank line

if n == 2 % phase plane plot
    plotsoln=input('Hit return for phase plane plot ')
    if isempty(plotsoln)
        plot(y(1,:), y(2,:),'-r');
        xlabel('y1')
        ylabel('y2')
    end
end
```

↪ EXAMPLE 17 Use the systems Euler's method with $\boxed{h = 0.1}$ to solve

$$\frac{dy_1}{dt} = -4y_1 - 2y_2 + \cos(t) + 4\sin(t)$$

$$\frac{dy_2}{dt} = 3y_1 + y_2 - 3\sin(t)$$

$$t \in [0,2], \quad y_1(0) = 0,\ y_2(0) = -1$$

(EXACT SOLUTION)

$$y_1(t) = 2e^{-t} - 2e^{-2t} + \sin(t)$$

$$y_2(t) = -3e^{-t} + 2e^{-2t}$$

↪ In vector form, this is $\frac{d\vec{y}}{dt} = \vec{f}(t, y_1, y_2), \quad t \in [0,2], \quad \vec{y}(0) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$, where

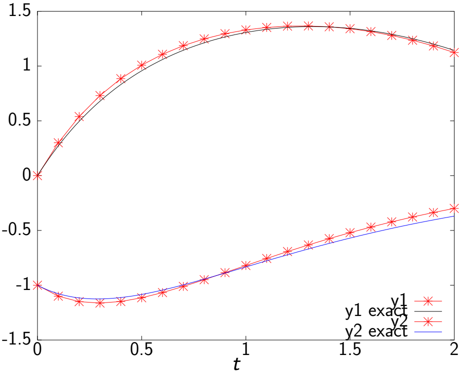
$$\vec{y} = \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} \quad \text{and} \quad \vec{f}(t, y_1, y_2) = \begin{bmatrix} -4y_1 - 2y_2 + \cos(t) + 4\sin(t) \\ 3y_1 + y_2 - 3\sin(t) \end{bmatrix}$$

- You will be expected to know how to change easily between the vector and non-vector form of such systems of ODEs.

Reminder: solving $\frac{d\vec{y}}{dt} = \vec{f}(t, y_1, y_2), \quad t \in [0, 2], \quad \vec{y}(0) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$ with

$$\vec{y} = \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} \text{ and } \vec{f}(t, y_1, y_2) = \begin{bmatrix} -4y_1 - 2y_2 + \cos(t) + 4\sin(t) \\ 3y_1 + y_2 - 3\sin(t) \end{bmatrix}$$

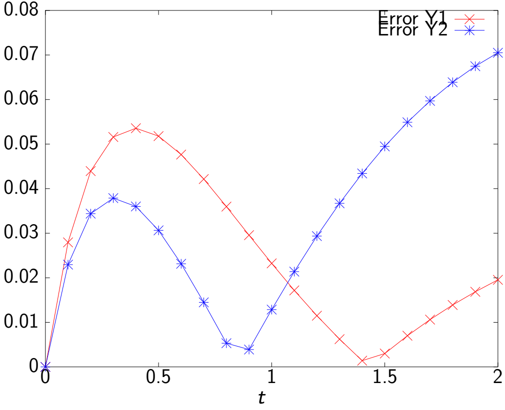
$y_1(t)$ and $y_2(t)$- Exact Solutions and Euler's Method Approximations



Reminder: solving $\frac{d\vec{y}}{dt} = \vec{f}(t, y_1, y_2), \quad t \in [0, 2], \quad \vec{y}(0) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$ with

$$\vec{y} = \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} \text{ and } \vec{f}(t, y_1, y_2) = \begin{bmatrix} -4y_1 - 2y_2 + \cos(t) + 4\sin(t) \\ 3y_1 + y_2 - 3\sin(t) \end{bmatrix}$$

$y_1(t)$ and $y_2(t)$- Error in Euler's Method Approximations



Y_1

i	TIME	Y_i (APPROX)	$y(t_i)$ (EXACT)	ABS. ERROR
0	0.000000	0.000000	0.000000	0.000000
1	0.100000	0.300000	0.272047	0.027953
2	0.200000	0.539434	0.495491	0.043943
3	0.300000	0.731125	0.679533	0.051591
4	0.400000	0.884960	0.831401	0.053559
5	0.500000	1.008510	0.956728	0.051782
		⋮		
16	1.600000	1.314846	1.321842	0.006996
17	1.700000	1.279670	1.290285	0.010615
18	1.800000	1.235906	1.249798	0.013892
19	1.900000	1.183836	1.200696	0.016860
20	2.000000	1.123791	1.143337	0.019546

Y_2

i	TIME	Y_i (APPROX)	$y(t_i)$ (EXACT)	ABS. ERROR
0	0.000000	-1.000000	-1.000000	0.000000
1	0.100000	-1.100000	-1.077051	0.022949
2	0.200000	-1.149950	-1.115552	0.034398
3	0.300000	-1.162716	-1.124831	0.037884
4	0.400000	-1.148306	-1.112302	0.036004
5	0.500000	-1.114474	-1.083833	0.030641
		⋮		
16	1.600000	-0.469265	-0.524165	0.054900
17	1.700000	-0.421610	-0.481304	0.059694
18	1.800000	-0.377370	-0.441249	0.063880
19	1.900000	-0.336489	-0.403964	0.067475
20	2.000000	-0.298877	-0.369375	0.070497

↪ EXAMPLE 18 Use the systems Euler's method to solve $x''' - x' = t$, $t \in [0, 4]$, $x(0) = 6$, $x'(0) = -5$, $x''(0) = 0$. (EXACT SOLUTION, $x(t) = 5 - 2e^t + 3e^{-t} - \frac{1}{2}t^2$).

≻ ANSWER First we convert it to a system:
Let $y_1 = x, \;\; y_2 = x', \;\; y_3 = x'' \;\; \Rightarrow$

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= y_3 \\ y_3' &= y_2 + t. \end{aligned}$$

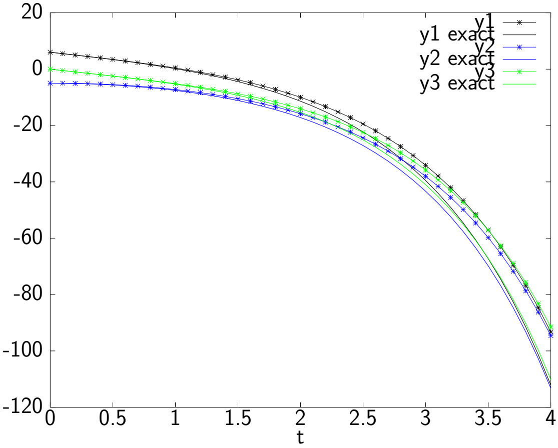
Meanwhile, the initial conditions become

$$\vec{y}(0) = \begin{bmatrix} y_1(0) \\ y_2(0) \\ y_3(0) \end{bmatrix} = \begin{bmatrix} 6 \\ -5 \\ 0 \end{bmatrix}.$$

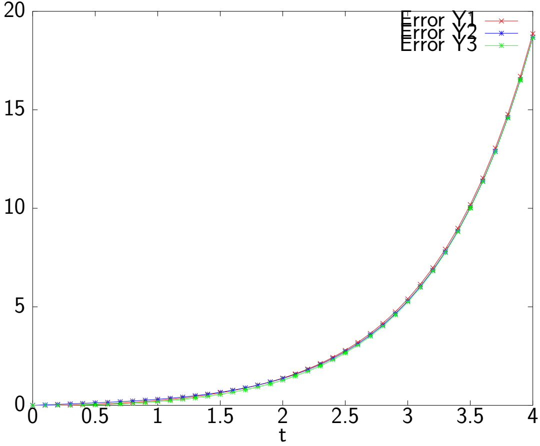
(EXACT SOLUTION
$y_1(t) = 5 - 2e^t + 3e^{-t} - \frac{1}{2}t^2, \quad y_2(t) = \qquad = \qquad$).

- We now need only make minor modifications to the earlier Euler's method program for a system of 2 equations to get it to work for 3 equations. The results are summarised on the following pages, first for $h = 0.1$ then for $h = 0.01$.

$y_1(t)$, $y_2(t)$, and $y_3(t)$ - Exact Solutions and Euler's Method Approximations with $h = 0.1$



$y_1(t)$, $y_2(t)$, and $y_3(t)$ - Error in Euler's Method Approximations with $h = 0.1$



Y_1

i	TIME	Y_i (APPROX)	$y(t_i)$ (EXACT)	ABS. ERROR
0	0.000000	6.000000	6.000000	0.000000
1	0.100000	5.500000	5.499170	0.000830
2	0.200000	5.000000	4.993387	0.006613
			⋮	
38	3.800000	−76.783943	−91.555257	14.771314
39	3.900000	−84.650286	−101.349172	16.698887
40	4.000000	−93.274168	−112.141353	18.867185

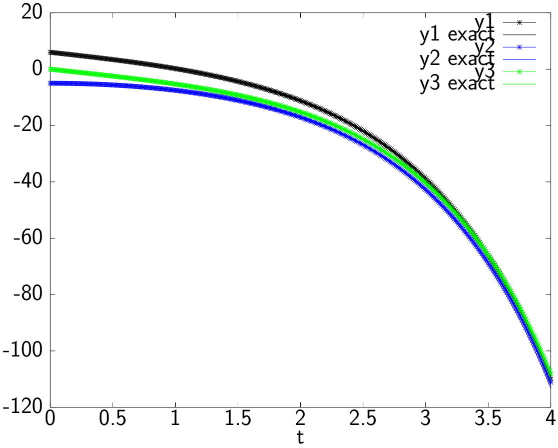
Y_2

i	TIME	Y_i (APPROX)	$y(t_i)$ (EXACT)	ABS. ERROR
0	0.000000	−5.000000	−5.000000	0.000000
1	0.100000	−5.000000	−5.024854	0.024854
2	0.200000	−5.050000	−5.098998	0.048998
			⋮	
38	3.800000	−78.663431	−93.269481	14.606050
39	3.900000	−86.238825	−102.765624	16.526799
40	4.000000	−94.562854	−113.251247	18.688393

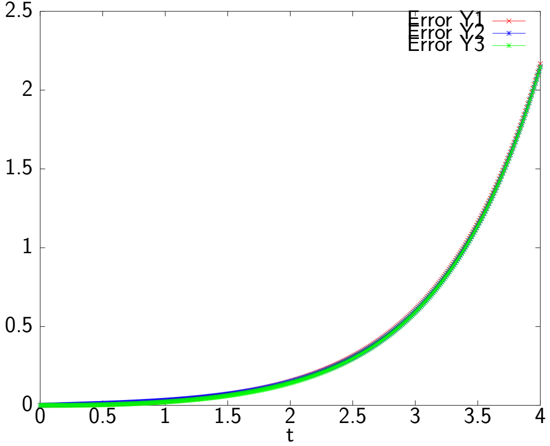
Y_3

i	TIME	Y_i (APPROX)	$y(t_i)$ (EXACT)	ABS. ERROR
0	0.000000	0.000000	0.000000	0.000000
1	0.100000	−0.500000	−0.495830	0.004170
2	0.200000	−0.990000	−0.986613	0.003387
			⋮	
38	3.800000	−75.753943	−90.335257	14.581314
39	3.900000	−83.240286	−99.744172	16.503887
40	4.000000	−91.474168	−110.141353	18.667185

$y_1(t)$, $y_2(t)$, and $y_3(t)$ - Exact Solutions and Euler's Method Approximations with $h = 0.01$



$y_1(t)$, $y_2(t)$, and $y_3(t)$ - Error in Euler's Method Approximations with $h = 0.01$



Y_1

i	TIME	Y_i (APPROX)	$y(t_i)$ (EXACT)	ABS. ERROR
0	0.000000	6.000000	6.000000	0.000000
1	0.010000	5.950000	5.949999	0.000001
2	0.020000	5.900000	5.899993	0.000007
		⋮		
398	3.980000	−107.784316	−109.898212	2.113895
399	3.990000	−108.874055	−111.014330	2.140274
400	4.000000	−109.974383	−112.141353	2.166970

Y_2

i	TIME	Y_i (APPROX)	$y(t_i)$ (EXACT)	ABS. ERROR
0	0.000000	−5.000000	−5.000000	0.000000
1	0.010000	−5.000000	−5.000250	0.000250
2	0.020000	−5.000500	−5.000999	0.000499
		⋮		
398	3.980000	−108.973906	−111.070125	2.096219
399	3.990000	−110.032747	−112.155278	2.122531
400	4.000000	−111.102086	−113.251247	2.149161

Y_3

i	TIME	Y_i (APPROX)	$y(t_i)$ (EXACT)	ABS. ERROR
0	0.000000	0.000000	0.000000	0.000000
1	0.010000	−0.050000	−0.049951	0.000049
2	0.020000	−0.099900	−0.099807	0.000093
		⋮		
398	3.980000	−105.884016	−107.978012	2.093995
399	3.990000	−106.933955	−109.054280	2.120324
400	4.000000	−107.994383	−110.141353	2.146970

Other Numerical Methods for Systems of First Order ODEs

- The relatively large errors in **EXAMPLE 18** with stepsize $h = 0.1$ are a good reason why we move on now to other *(higher order)* numerical methods for systems of first order IVPs.
- As mentioned earlier, the equations for the systems version of the different numerical methods remain the same as their scalar counterparts when written in vector notation *(with the appropriate vectorised interpretation of $+, -$ and multiplication by h)*, and applying a numerical IVP method to a system of ODEs simply consists of applying the *scalar* form of that method to a vector of differential equations one component at a time.
  - In these notes, we will only look at systems versions of Heun's method and the 4th order Runge-Kutta method.

    (For systems versions of other methods, such as TS(2) and AB(2), you can consult the MATH1106 Lecture Notes [contact me if you do not have access to those notes and wish to see them] or books on numerical solutions to ODEs).

Heun's Method

↪ **In summary, Heun's method for a first order system of ODEs simply consists of applying the *scalar* Heun's method to a vector of differential equations one component at a time.**

↪ The *TWO-STEP (see **Lecture 2**)* version of the method is summarised on the following page *for a system of n first order IVPs*.

Heun's Method for Systems

Heun's method for approximating the solution to the *general first order system of n IVPs*, $\frac{d\vec{y}}{dt} = \vec{f}(t, \vec{y}), \quad \forall\ t \in [t_0, T], \quad \vec{y}(t_0) = \vec{y}_0$:

Heuns's Method for Systems

$$\vec{Y}_0 = \vec{y}(t_0) \text{ THEN}$$

$$\overrightarrow{Ytemp}^{(i+1)} = \vec{Y}^{(i)} + h\vec{f}(t_i, \vec{Y}^{(i)}) \text{ AND}$$

$$\vec{Y}^{(i+1)} = \vec{Y}^{(i)} + \frac{h}{2}\left[\vec{f}(t_i, \vec{Y}^{(i)}) + f(t_{i+1}, \overrightarrow{Ytemp}^{(i+1)}\right] \text{ for } i = 0, 1, 2 \ldots, N-1.$$

where

$$\vec{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} \text{ and } \vec{f}(t, \vec{Y}) = \vec{f}(t, Y_1, Y_2, \ldots, Y_n) = \begin{bmatrix} f_1(t, Y_1, Y_2, \ldots, Y_n) \\ f_2(t, Y_1, Y_2, \ldots, Y_n) \\ \vdots \\ f_n(t, Y_1, Y_2, \ldots, Y_n) \end{bmatrix}$$

and $Y_i^{(j)}$ is the Heun approximation to $y_i(t_j)$ (for $i = 1, 2, \ldots, n$ and $j = 0, 1, 2, \ldots, N$).

$\hookrightarrow$ EXAMPLE 19 - Redo **EXAMPLE 17** using Heun's method: *use the systems Heun's method with* $\boxed{h = 0.1}$ *to solve*

$$\frac{dy_1}{dt} = -4y_1 - 2y_2 + \cos(t) + 4\sin(t)$$

$$\frac{dy_2}{dt} = 3y_1 + y_2 - 3\sin(t)$$

$$t \in [0,2], \quad y_1(0) = 0, \; y_2(0) = -1$$

(EXACT SOLUTION)

$$y_1(t) = 2e^{-t} - 2e^{-2t} + \sin(t)$$

$$y_2(t) = -3e^{-t} + 2e^{-2t}$$

$\hookrightarrow$ Recall that in vector form, this is

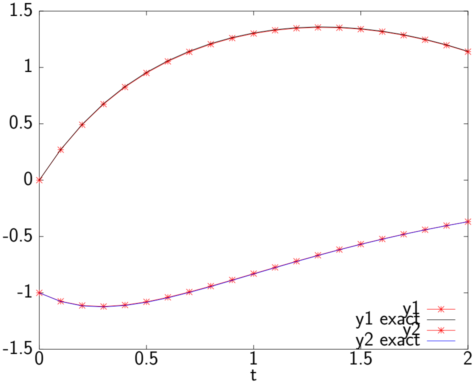
$$\frac{d\vec{y}}{dt} = \vec{f}(t, y_1, y_2), \quad t \in [0,2], \quad \vec{y}(0) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \text{ where}$$

$$\vec{y} = \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} \text{ and } \vec{f}(t, y_1, y_2) = \begin{bmatrix} -4y_1 - 2y_2 + \cos(t) + 4\sin(t) \\ 3y_1 + y_2 - 3\sin(t) \end{bmatrix}.$$

Reminder: solving $\frac{d\vec{y}}{dt} = \vec{f}(t, y_1, y_2)$, $\quad t \in [0, 2]$, $\quad \vec{y}(0) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$ with

$$\vec{y} = \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} \text{ and } \vec{f}(t, y_1, y_2) = \begin{bmatrix} -4y_1 - 2y_2 + \cos(t) + 4\sin(t) \\ 3y_1 + y_2 - 3\sin(t) \end{bmatrix}$$

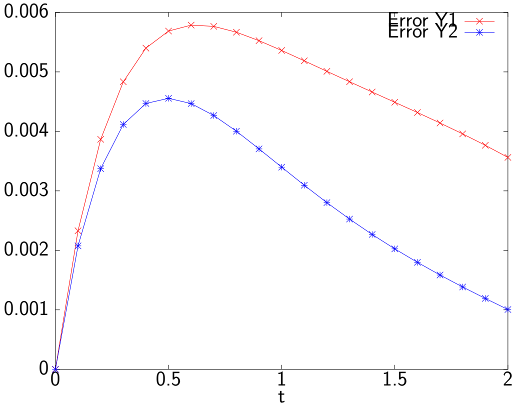
$y_1(t)$ and $y_2(t)$- Exact Solutions and Heun's Method Approximations



Reminder: solving $\frac{d\vec{y}}{dt} = \vec{f}(t, y_1, y_2), \quad t \in [0, 2], \quad \vec{y}(0) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$ with

$$\vec{y} = \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} \text{ and } \vec{f}(t, y_1, y_2) = \begin{bmatrix} -4y_1 - 2y_2 + \cos(t) + 4\sin(t) \\ 3y_1 + y_2 - 3\sin(t) \end{bmatrix}$$

$y_1(t)$ and $y_2(t)$- Error in Heun's Method Approximations



Y_1

i	TIME	Y_i (APPROX)	$y(t_i)$ (EXACT)	ABS. ERROR	Euler's Error
0	0.000000	0.000000	0.000000	0.000000	0.000000
1	0.100000	0.269717	0.272047	0.002330	0.027953
2	0.200000	0.491624	0.495491	0.003867	0.043943
3	0.300000	0.674699	0.679533	0.004834	0.051591
4	0.400000	0.826001	0.831401	0.005400	0.053559
5	0.500000	0.951041	0.956728	0.005687	0.051782
			⋮		
16	1.600000	1.317525	1.321842	0.004318	0.006996
17	1.700000	1.286144	1.290285	0.004141	0.010615
18	1.800000	1.245840	1.249798	0.003958	0.013892
19	1.900000	1.196929	1.200696	0.003766	0.016860
20	2.000000	1.139774	1.143337	0.003563	0.019546

Y_2

i	TIME	Y_i (APPROX)	$y(t_i)$ (EXACT)	ABS. ERROR	Euler's Error
0	0.000000	−1.000000	−1.000000	0.000000	0.000000
1	0.100000	−1.074975	−1.077051	0.002076	0.022949
2	0.200000	−1.112178	−1.115552	0.003374	0.034398
3	0.300000	−1.120715	−1.124831	0.004117	0.037884
4	0.400000	−1.107832	−1.112302	0.004470	0.036004
5	0.500000	−1.079276	−1.083833	0.004557	0.030641
			⋮		
16	1.600000	−0.522366	−0.524165	0.001799	0.054900
17	1.700000	−0.479718	−0.481304	0.001586	0.059694
18	1.800000	−0.439865	−0.441249	0.001385	0.063880
19	1.900000	−0.402772	−0.403964	0.001192	0.067475
20	2.000000	−0.368369	−0.369375	0.001006	0.070497

RK4 Method

$\hookrightarrow$ **In summary, the RK4 method for a first order system of ODEs simply consists of applying the *scalar* RK4 method to a vector of differential equations one component at a time.**

$\hookrightarrow$ You should try to implement this for two equations by modifying the earlier Euler's or Heun's method program; this is easier if you write it out in vector form and then think of how to update the components of those vectors.

RK4 Method for Systems

$\overrightarrow{\mathbf{Y}}_{\mathbf{0}} = \vec{\mathbf{y}}(\mathbf{t_0})$ THEN

$$
\begin{aligned}
\vec{k}_1 &= h\vec{f}(t_i, \overrightarrow{Y}^{(i)}) \\
\vec{k}_2 &= hf(t_i + \tfrac{1}{2}h, \overrightarrow{Y}^{(i)} + \tfrac{1}{2}\vec{k}_1) \\
\vec{k}_3 &= hf(t_i + \tfrac{1}{2}h, \overrightarrow{Y}^{(i)} + \tfrac{1}{2}\vec{k}_2) \\
\vec{k}_4 &= hf(t_i + h, \overrightarrow{Y}^{(i)} + \vec{k}_3)
\end{aligned}
$$

AND

$$
\overrightarrow{\mathbf{Y}}^{(\mathbf{i+1})} = \overrightarrow{\mathbf{Y}}^{(\mathbf{i})} + \tfrac{1}{6}\vec{\mathbf{k}}_1 + \tfrac{1}{3}\vec{\mathbf{k}}_2 + \tfrac{1}{3}\vec{\mathbf{k}}_3 + \tfrac{1}{6}\vec{\mathbf{k}}_4
$$

for $i = 0, 1, 2 \ldots, N-1$

where

$$
\overrightarrow{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} \quad \text{and} \quad \vec{f}(t, \overrightarrow{Y}) = \vec{f}(t, Y_1, Y_2, \ldots, Y_n) = \begin{bmatrix} f_1(t, Y_1, Y_2, \ldots, Y_n) \\ f_2(t, Y_1, Y_2, \ldots, Y_n) \\ \vdots \\ f_n(t, Y_1, Y_2, \ldots, Y_n) \end{bmatrix}
$$

and $Y_i^{(j)}$ is the RK4 approximation to $y_i(t_j)$ (for $i = 1, 2, \ldots, n$ and $j = 0, 1, 2, \ldots, N$).

$\hookrightarrow$ EXAMPLE 20 - Redo **EXAMPLES 17 & 19** using the RK4 method: *use the systems RK4 method with* $\boxed{h = 0.1}$ *to solve*

$$\frac{dy_1}{dt} = -4y_1 - 2y_2 + \cos(t) + 4\sin(t)$$

$$\frac{dy_2}{dt} = 3y_1 + y_2 - 3\sin(t)$$

$$t \in [0,2], \quad y_1(0) = 0, \; y_2(0) = -1$$

(EXACT SOLUTION)

$$y_1(t) = 2e^{-t} - 2e^{-2t} + \sin(t)$$

$$y_2(t) = -3e^{-t} + 2e^{-2t}$$

$\hookrightarrow$ Recall that in vector form, this is

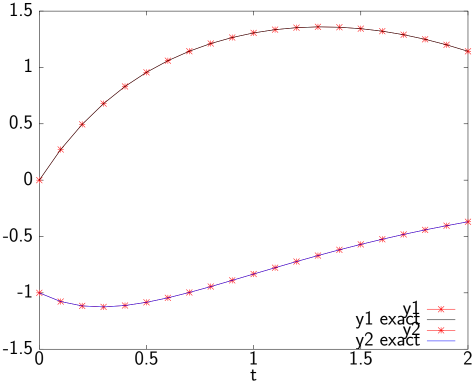
$$\frac{d\vec{y}}{dt} = \vec{f}(t, y_1, y_2), \quad t \in [0,2], \quad \vec{y}(0) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \text{ where}$$

$$\vec{y} = \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} \text{ and } \vec{f}(t, y_1, y_2) = \begin{bmatrix} -4y_1 - 2y_2 + \cos(t) + 4\sin(t) \\ 3y_1 + y_2 - 3\sin(t) \end{bmatrix}.$$

Reminder: solving $\frac{d\vec{y}}{dt} = \vec{f}(t, y_1, y_2)$, $\quad t \in [0, 2]$, $\quad \vec{y}(0) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$ with

$$\vec{y} = \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} \text{ and } \vec{f}(t, y_1, y_2) = \begin{bmatrix} -4y_1 - 2y_2 + \cos(t) + 4\sin(t) \\ 3y_1 + y_2 - 3\sin(t) \end{bmatrix}$$

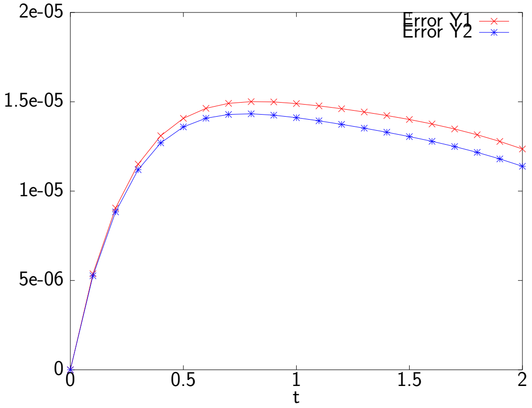
$y_1(t)$ and $y_2(t)$- Exact Solutions and RK4 Method Approximations



Reminder: solving $\frac{d\vec{y}}{dt} = \vec{f}(t, y_1, y_2), \quad t \in [0, 2], \quad \vec{y}(0) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$ with

$$\vec{y} = \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} \text{ and } \vec{f}(t, y_1, y_2) = \begin{bmatrix} -4y_1 - 2y_2 + \cos(t) + 4\sin(t) \\ 3y_1 + y_2 - 3\sin(t) \end{bmatrix}$$

$y_1(t)$ and $y_2(t)$- Error in RK4 Method Approximations



Y_1

i	TIME	Y_i (APPROX)	$y(t_i)$ (EXACT)	ABS. ERROR	Heun's Error	Euler's Error
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1	0.100000	0.272041	0.272047	0.000005	0.002330	0.027953
2	0.200000	0.495482	0.495491	0.000009	0.003867	0.043943
3	0.300000	0.679522	0.679533	0.000012	0.004834	0.051591
4	0.400000	0.831387	0.831401	0.000013	0.005400	0.053559
5	0.500000	0.956714	0.956728	0.000014	0.005687	0.051782
			⋮			
16	1.600000	1.321828	1.321842	0.000014	0.004318	0.006996
17	1.700000	1.290272	1.290285	0.000013	0.004141	0.010615
18	1.800000	1.249785	1.249798	0.000013	0.003958	0.013892
19	1.900000	1.200683	1.200696	0.000013	0.003766	0.016860
20	2.000000	1.143324	1.143337	0.000012	0.003563	0.019546

Y_2

i	TIME	Y_i (APPROX)	$y(t_i)$ (EXACT)	ABS. ERROR	Heun's Error	Euler's Error
0	0.000000	-1.000000	-1.000000	0.000000	0.000000	0.000000
1	0.100000	-1.077045	-1.077051	0.000005	0.002076	0.022949
2	0.200000	-1.115543	-1.115552	0.000009	0.003374	0.034398
3	0.300000	-1.124820	-1.124831	0.000011	0.004117	0.037884
4	0.400000	-1.112290	-1.112302	0.000013	0.004470	0.036004
5	0.500000	-1.083820	-1.083833	0.000014	0.004557	0.030641
			⋮			
16	1.600000	-0.524152	-0.524165	0.000013	0.001799	0.054900
17	1.700000	-0.481292	-0.481304	0.000012	0.001586	0.059694
18	1.800000	-0.441237	-0.441249	0.000012	0.001385	0.063880
19	1.900000	-0.403953	-0.403964	0.000012	0.001192	0.067475
20	2.000000	-0.369363	-0.369375	0.000011	0.001006	0.070497

End of Section

Geometrical Study of Solutions to Systems of First Order ODEs

Introduction

- More so than for single differential equations, systems of differential equations are hard to solve (in the earlier part of this lecture we only considered a very small subset: linear homogeneous systems with constant coefficient matrices). Furthermore, often we are just interested in *patterns* or *general behaviour* of solutions to the scenario being modelled by a system of ODEs, *and these can be determined by a geometric analysis of the system of ODEs without solving it!*.
- Much of this geometrical work will be done for systems of 2 ODEs, but the (often relatively straightforward) generalisations to systems of 3 or more ODEs will be mentioned.
- Again, much of the work will be done initially for *linear constant coefficient systems of ODEs*, and then the very straighforward generalisation to nonlinear systems will be covered.

$\hookrightarrow$ The geometrical techniques we will use on general systems of first order ODEs will fall into three broad categories:

1. Generating and interpreting direction fields in the phase space.
2. Creating phase portraits (by hand).
3. Finding and classifying the steady states of the system of ODEs using calculus and linear algebra.

- Sometimes information from number 3 is used to help inform the creation of phase portraits in number 2 and/or to help choose a suitable domain in which to generate a direction field in number 1.

Phase Space, Phase Portraits, Direction Fields, Steady States - Vocabulary

↪ This sub-section is essentially a *crucial* vocabulary lesson.

- DEFINITION A **phase space** for a system of n first order differential equations $\frac{d\vec{y}}{dt} = f(t, \vec{y})$ is simply an $n-$dimensional coordinate system with axes y_i, $i = 1, \ldots, n$, in which the trajectory of the solution vector $\vec{y} = \begin{pmatrix} y_1(t) \\ \vdots \\ y_n(t) \end{pmatrix}$ can be traced out as t increases.

≻ In practice, a set of such solution trajectories can only easily be visualised for systems of 2 (or 3) ODEs, in which case it is called a **phase plane**.

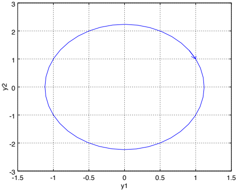
↦ All of the subsequent definitions of key terminology such a *nullclines* and *steady states* will be related to/based on the graphs of solutions in *phase space*.

- ↪ The phase plane (more generally, phase space) is a powerful tool for making sense of the behaviour of solutions to autonomous systems of ODEs - which often arise in modelling biological processes. Indeed, it is often more informative than the graphs of individual solutions as functions of time, $y_i(t)$ versus t.
- ↪ One can also re-create (approximately) those graphs of individual solutions versus time from a phase plane plot of the solutions (and vice versa, although we won't really need this). This is possibly best illustrated by an example.
  - ▶ On the following page is the phase plot for the ODE system

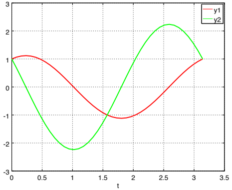
$$\frac{d\vec{y}}{dt} = \begin{pmatrix} 0 & 1 \\ -4 & 0 \end{pmatrix} \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} \quad \textit{with} \quad \vec{y}(0) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

along with plots of y_1 and y_2 versus t and a description of how one could get from the phase plane plot to the solutions versus time plots.

Phase Plane Plot



y_i Versus t Plot



- Starting at $(y_1, y_2) = (1, 1)$ in the phase plane, we see that the value of y_1 increases briefly to its peak slightly above 1 then decreases all the way down to a value slightly below -1 then increases all the way to 1 again.
- At the same time, y_2 decreases to a lowest value of just below -2 (which coincides with when y_1 is 0, then it increases to just above 2 (again coinciding with when y_1 is 0), then descends again to 1.
- These two solution behaviours as functions of time are confirmed by the second plot above.

Steady States

↪ As for single autonomous ODEs, the *equilibrium points* of an autonomous system of ODEs $\dfrac{d\vec{y}}{dt} = f(\vec{y})$ are just the solutions to

$$f(\vec{y}) = \vec{0}.$$

When dealing with systems of ODEs, these equilibrium points are typically called **steady states** (for hopefully obvious reasons).

↪ NOTE to find the steady states of a nonlinear system of ODEs, one has to solve a nonlinear system of equations - so some of the methods mentioned in **Lecture 3** and **Tutorial 3** such as the use of Matlab's in-built function `fsolve` can be used.

- EXAMPLE 21: If $A_{n\times n}$ is a non-singular matrix then the only steady state in the system $\boldsymbol{\dfrac{d\vec{y}}{dt} = A\vec{y}}$ is

.

- EXAMPLE 22: Find the steady state(s) of $\dfrac{d\vec{y}}{dt} = \begin{pmatrix} 6y_1 + 6y_2 + 12 \\ 6y_1 + 6y_2^2 - 24 \end{pmatrix}$.
- ANSWER

$\rightsquigarrow$ Or just use `fsolve` as described in **Tutorial 3**, but note you will have to use different starting points (ordered pairs) to get the two different steady states.

- I'd suggest plotting the two graphs together to see approximately where they intersect and then choosing starting values close to each intersection point.

Types and Stability of Steady States

↪ As for single autonomous ODEs, the steady states of autonomous systems of ODEs can be classifed by their *stability*. They can also be classified by their *type* - which is a way of categorising how solutions close to them behave. For the autonomous ODE system $\dfrac{d\vec{y}}{dt} = f(\vec{y})$ a *steady state* $\vec{y}_0$ is:

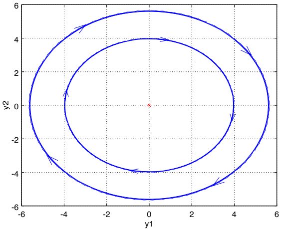
- ▷ **asymptotically stable/a sink** if all solution trajectories in the phase space which start out near to $\vec{y}_0$ move closer to $\vec{y}_0$ as $t \to \infty$;
- ▷ **unstable/repelling/(a source)** if some solution trajectories in the phase space which starts out near to $\vec{y}_0$ move away from $\vec{y}_0$ as $t \to \infty$;
- ▷ **stable** if each solution trajectory in the phase space which starts out near to $\vec{y}_0$ stays the same distance away from $\vec{y}_0$ as $t \to \infty$.

↪ There are four main categories of steady states:

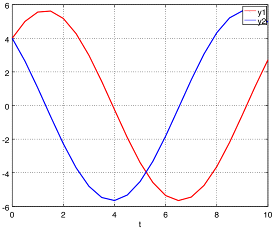
1. **Centres**. *These are always STABLE.*
2. **Spiral Points**, also called **Focuses**. *These can be ASYMPTOTICALLY STABLE or UNSTABLE.*
3. **Saddle Points**. *These are always UNSTABLE*
4. **Nodes**, both **proper** and **improper**. *These can be ASYMPTOTICALLY STABLE or UNSTABLE.*

- Instead of giving formal definitions at this point, I will just show you what they look like in a phase plane and highlight their key features, as well as show representative plots in the t-y plane. Knowing the names is less important than knowing the behaviours and how to determine those behaviours for a given steady state.
- As a useful exercise, with each steady state type in what follows try to think of the types of solutions to linear homogeneous constant coefficient systems $\frac{d\vec{y}}{dt} = A\vec{y}$ (based on the types of eigenvalues of A) which coincide with the steady state type based on the solution plots in the t-y plane.
- In the following diagrams, the origin $(\mathbf{0}, \mathbf{0})$, is the steady state.

Centre



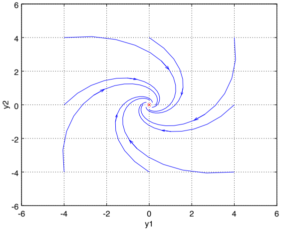
Phase Plane Plot



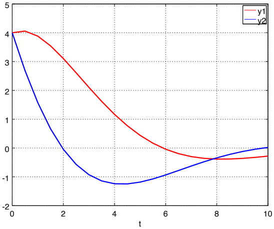
Sample Solution vs Time Plot

- The choice of the name *centre* is obvious from the phase plane plot.
- The solutions are periodic functions which each oscillate around its component of the steady state.

Spiral Point/Focus



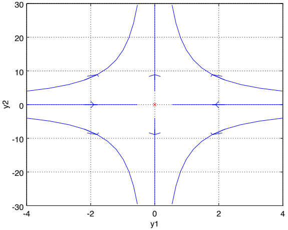
Phase Plane Plot



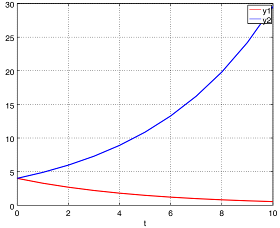
Sample Solution vs Time Plot

- ↪ The choice of the name *spiral point* or *focus* is obvious from the phase plane plot.
- ↪ This shows an asymptotically stable spiral point. In an unstable spiral point, the arrows would point in the opposite direction (away from the origin).
- ↪ The sample solutions for an asymptotically stable spiral point are shown. Each function oscillates around its component of the steady state and the amplitude of those oscillations decrease with increasing time. For an unstable spiral point, the oscillations would increase with increasing time.

Saddle Point



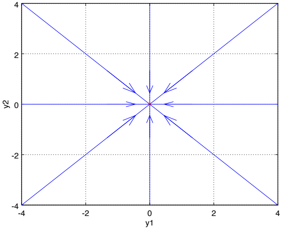
Phase Plane Plot

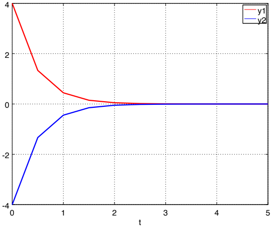


Sample Solution vs Time Plot

- The choice of the name *saddle point* is obvious from the phase plane plot if one remembers that term from optimisation of functions of 2 independent variables.
- The sample solutions are representative: one of the solutions always approaches its component of the steady state and the other always diverges away from its component of the steady state as $t \to \infty$.
- So saddle points are always unstable.

Node

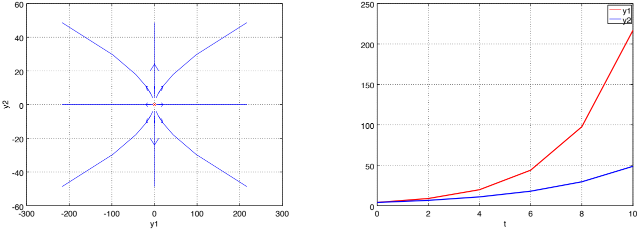




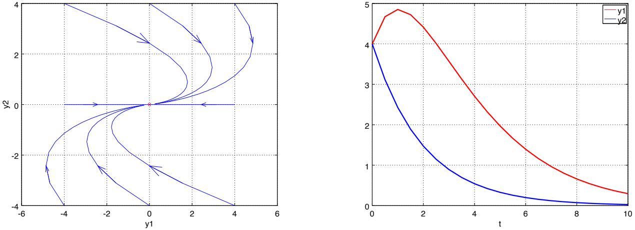
↪ This shows a proper node. The plots on the next page show improper nodes.

↪ This node is asymptotically stable (a sink). On the following page, one node is unstable and the other is asymptotically stable. Obviously, reversing the arrows on the diagrams changes an asymptotically stable node to unstable and vice versa.

Node - continued



An Unstable *Improper* Node



An Asymptotically Stable *Improper* Node

Direction Fields

- As with single ODEs, *direction fields*, drawn in phase space, are helpful in determining the general behaviour of solutions to system of ODEs.
- This is particularly the case if we know the steady states so that we can include them in the region in which we draw direction fields.

- Let us consider the autonomous ODE system $\begin{pmatrix} \frac{dy_1}{dt} \\ \\ \frac{dy_2}{dt} \end{pmatrix} = \begin{pmatrix} F_1(y_1, y_2) \\ F_2(y_1, y_2) \end{pmatrix}$. If we wish to think of $y_2(t)$ as a function (or relation) of $y_1(t)$, then by the **Chain Rule** we have

$$y_2 = y_2(y_1(t)) \Rightarrow \frac{dy_2}{dt} = \frac{dy_2}{dy_1}\frac{dy_1}{dt} \Rightarrow \frac{dy_2}{dy_1} = \frac{dy_2/dt}{dy_1/dt}.$$

KEY RESULT $y_1 = y_1(t)$ and $y_2 = y_2(t)$ then

$$\frac{dy_2}{dy_1} = \frac{dy_2/dt}{dy_1/dt} = \frac{y_2'(t)}{y_1'(t)} = \frac{\dot{y}_2}{\dot{y}_1} = \frac{F_2(y_1, y_2)}{F_1(y_1, y_2)}.$$

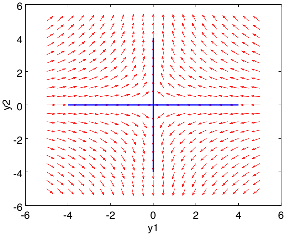
- In particular, at each point t, the vector $\left(\frac{dy_1}{dt}, \frac{dy_2}{dt}\right)$ is *tangent* to the trajectory traced out by $(y_1(t),\ y_2(t))$ in the *phase plane*. Ask if you are not sure why. *(Technically, at each point t the position vector $\left(\frac{dy_1}{dt}, \frac{dy_2}{dt}\right)$, with origin at $(0, 0)$, when shifted to the point (y_1, y_2) in the phase plane is tangent to the curve traced out by $(y_1(t), y_2(t))$).*

$\hookrightarrow$ So if on a grid of $(y_1,\ y_2)$ values we plot a little line segment parallel to $(F_1(y_1,y_2),\ F_2(y_1,y_2))$ at each of the grid points, the overall picture should show how trajectories of solutions behave in the phase plane.

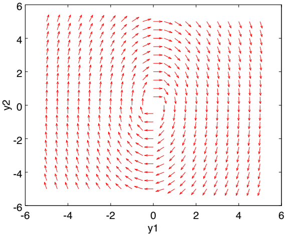
$\hookrightarrow$ Obviously, while straightforward, this would be tedious to do by hand, so see **Tutorial 4** for a link to a simple Matlab program which does this automatically. You may use this program or a slightly modified version of it which I will put up on the course Moodle page.

$\hookrightarrow$ As for single ODEs, this is an easy way to get an idea of how solutions to *a system of ODEs* behave, especially near steady states, **without solving the system of ODEs**.

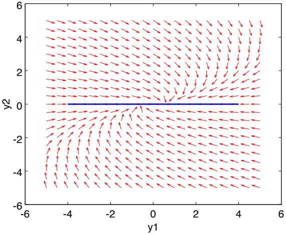
- On the following page are some direction fields around steady states at $(0,0)$. Try to guess the *type* and *stability* of the steady states.



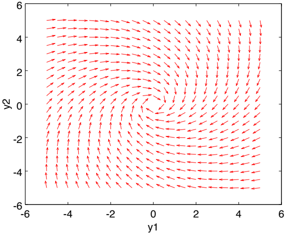
An Unstable ***Saddle Point***



A Stable ***Centre***



An Asymptotically Stable ***Improper Node***



An Asymptotically Stable ***Spiral/Focus***

How to Sketch Phase Portraits

- As we did for single ODEs, we can use calculus to help us sketch what typical solutions look like (in the phase plane) without solving a system of ODEs.
- DEFINITION In a system of n autonomous first order ODEs, $\dfrac{d\vec{y}}{dt} = \vec{f}(\vec{y})$, the j^{th} nullcline is the geometric shape for which $\dfrac{dy_j}{dt}$ *or* $f_i(\vec{y}) = 0$.
- So for systems of 2 ODEs, the nullclines are curves in the phase plane.
- Clearly the *steady states* are where nullclines intersect.
- Because of the topology of $\mathbb{R}^2$, nullclines typically split the plane into regions where the behaviour of solutions is similar so that we can typically generalise the *local behaviour* of solutions near to steady states to the *global behaviour* of solutions. This is not typically true for systems of more than 2 ODEs.
- Observe that in $\mathbb{R}^2$ if $dy_1/dt = 0$ then y_1 does not change with time so all trajectories on that nullcline must be parallel to the y_2 axis (perpendicular to the y_1 axis); similarly, all trajectories on the nullcline $dy_2/dt = 0$ must be parallel to the y_1 axis (perpendicular to the y_2 axis).
- The following summary of how to sketch phase portraits is a slightly modified version of what is in A Primer on Mathematical Models in Biology by Segel and Edelstein-Keshet.

How to Sketch Phase Portraits - continued

Here is a systematic way of sketching trajectories in the phase plane for

$$\begin{pmatrix} \frac{dy_1}{dt} \\ \\ \frac{dy_2}{dt} \end{pmatrix} = \begin{pmatrix} F_1(y_1, y_2) \\ F_2(y_1, y_2) \end{pmatrix}$$ *without solving the system of ODEs*

1. If possible, find the steady states by solving the system of algebraic equations $dy_1/dt = F_1(y_1, y_2) = 0$, $dy_2/dt = F_2(y_1, y_2) = 0$. Otherwise, go to step 2.
2. Plot the **vertical nullcline(s)**, $dy_1/dt = F_1(y_1, y_2) = 0$ and put vertical trajectories along it (them).
3. Plot the **horizontal nullcline(s)**, $dy_2/dt = F_2(y_1, y_2) = 0$ and put horizontal trajectories along it (them).
4. Identify the steady states - where the nullclines intersect. Note some of them might NOT be biologically relevant for a given problem (negative populations, for example).
5. Use the differential equations and select convenient points (y_1, y_2) [for example, y_1 or $y_2 = 0$ or very large] to determine the sign of $dy_1/dt = F_1(y_1, y_2)$ and $dy_2/dt = F_2(y_1, y_2)$ in various regions. Recall that unless these derivatives have discontinuities, one can assume that the signs of dy_1/dt and dy_2/dt change only at the nullclines.
   Put left-pointing arrows where $dy_1/dt < 0$, right-pointing arrows where $dy_1/dt > 0$, downward pointing arrows where $dy_2/dt < 0$, and upward-pointing arrows where $dy_2/dt > 0$.

How to Sketch Phase Portraits - continued

6. If not already done, put arrows along the axes $y_1 = 0$ and $y_2 = 0$ to indicate the direction of trajectories along them.
7. Determine the stability and type of the steady states (if possible) by looking at the direction of the arrows etc. Sometimes this cannot be done fully and the analysis using eigenvalues in the later sub-section can be used.
8. Combine all of the preceding information into a consistent picture, recalling that trajectories can only intersect at steady state points.

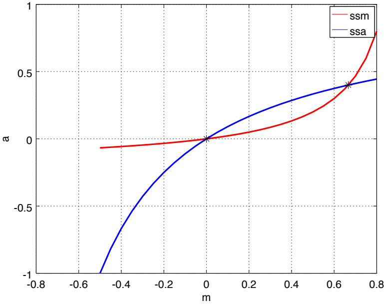
We next give an example of how to construct a phase portrait (taken largely from section 7.6.1 of `A Primer on Mathematical Models in Biology` by Segel and Edelstein-Keshet).

- $\boxed{\text{EXAMPLE 23}}$: A dimensionless model for macrophage cells $m(t)$ removing dead cells $a(t)$ and killing other cells is given by the system:

$$\frac{dm}{dt} = \alpha(1-m)a - \delta m, \qquad \frac{da}{dt} = m - \eta ma - a.$$

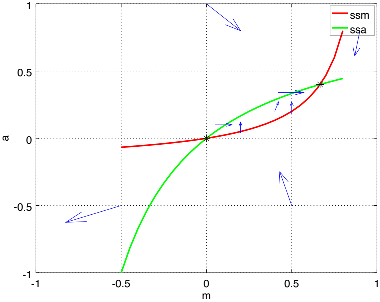
where $\alpha, \delta, \eta > 0$ are constants. Sketch a phase portrait for this system:

- $\boxed{\text{ANSWER}}$



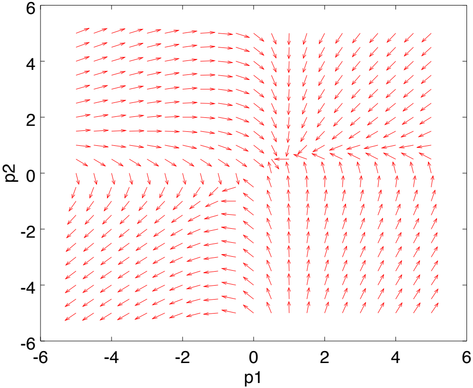
- We will next put vertical arrows across the red curve $dm/dt = 0$ and horizontal arrows across the blue curve $da/dt = 0$.
- We also calculate $dm/dt = \alpha(1-m)a - \delta m$ and $da/dt = m - \eta ma - a$ in various regions (done in Matlab) and insert appropriately-scaled vectors parallel to the $(dm/dt,\ da/dt)$ vectors and emanating from those points.

Sample point	$(dm/dt,\ da/dt)$
$(0, 1)$	$(1, -1)$
$(-0.5, -0.5)$	$(-0.65, -0.25)$
$(0.5, -0.5)$	$(-0.35, 1.25)$
$(0.4, 0.2)$	$(0.04, 0.12)$
$(0.9, 0.8)$	$(-0.1, -0.62)$



- From this we see some trajectories going towards the origin and some moving away, so it is likely an *saddle point* (hence unstable).

- On the other hand, trajectories seem to move towards the steady state in the first quadrant, so it appears asymptotically stable and the pattern of approach suggest it is likely a node (although there are other possibilities). The classification of the steady states can be confirmed by an eigenvalue analysis as described later.



- This direction field plot confirms our conclusions on the preceding page.

Classification of Steady State for Linear Systems of ODEs

- ↪ Recall from earlier that the constant coefficient linear system of ODEs $\vec{y}\,' = A\vec{y}$ has only one steady state: the zero vector.
- ↪ We can easily classify that steady state based on the eigenvalues of the matrix A.
- ↦ These classifications are easy to understand if the form of solutions to the ODE system with different types of eigenvalues are recalled *(See the first section of these lecture notes)*.
  - NOTE there are tests which do not require the calculation of the eigenvalues and just require looking at certain combinations of the entries of the matrix A *(I won't cover those in any detail but you can find them summarised in many standard ODE or Mathematical Biology books, including several on the course's reading list)*.
  - We will discuss this classification based on the eigenvalue types of A, looking at 5 cases and then summarising at the end.

$\mapsto$ First just a few general observations regarding the **EIGENVECTORS** of A, given that the solution of the ODE system is typically of the form $\vec{y} = A\vec{v}_1 e^{\lambda_1 t} + B\vec{v}_2 e^{\lambda_2 t}$.

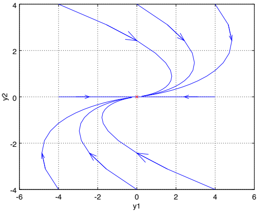
- If initial conditions are such that either A or B is zero, the solution vector will just be a scalar multiple of one of the eigenvectors, hence its trajectory in the phase plane will just be the line through the origin determined by that eigenvector.
- Thus solution trajectories that start out on the line determined by one of the eigenvectors just follows that line as $t \to \infty$, going away from the origin if the corresponding eigenvalue is positive (unstable) or towards the origin if the corresponding eigenvalue is negative (asymptotically stable) .
- Solution trajectories which do not start off on an eigenvector are generally curved and tend towards the eigenvector associated with the largest eigenvalue as t increases.

CASE 1: λ_1, λ_2 real, unequal, and of SAME sign

↪ In that case, the form of the solution is $\vec{y}(t) = A\vec{v}_1 e^{\lambda_1 t} + B\vec{v}_2 e^{\lambda_2 t}$

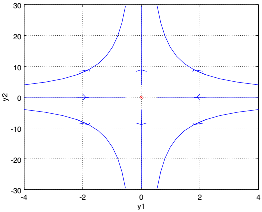
- It is then easy to see that if both eigenvalues are negative, the solution vector $\vec{y}$ must approach $\vec{0}$, the steady state, as $t \to \infty$ hence the steady state is asymptotically stable.
- It is also clear that if both eigenvalues are positive the solution vector diverges away from $\vec{0}$, the steady state, as $t \to \infty$ hence the steady state is unstable.

↪ This type of steady state is called a **Node**.



CASE 2: λ_1, λ_2 real, unequal, and of DIFFERENT signs

- Without loss of generality, assume $\boldsymbol{\lambda_1 < 0 < \lambda_2}$. The form of the solution is $\vec{y}(t) = A\vec{v}_1 e^{\lambda_1 t} + B\vec{v}_2 e^{\lambda_2 t}$
  - It is then easy to see that only if a solution starts out with $B = 0$ (so along the line determined by the eigenvector $\vec{v}_1$ will solutions approach the $(0,0)$ steady state as $t \to \infty$; otherwise, solutions approach ∞ (tangent to the line determined by the eigenvector $\vec{v}_2$). Hence the steady state is unstable.
- This type of steady state is called a **Saddle Point**.



CASE 3: $\lambda_1 = \lambda_2$ real, equal eigenvalues

↪ There are two main cases:

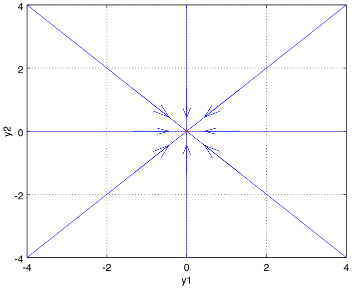
I. **There are two linearly independent eigenvectors $\vec{v}_1$ and $\vec{v}_2$**: In this case, the solution is of the form $\vec{y}(t) = A\vec{v}_1 e^{\lambda_1 t} + B\vec{v}_2 e^{\lambda_1 t}$. NOTE the ratio y_1/y_2 is now independent of t (the $e^{\lambda_1 t}$ terms cancel out) but is dependent on the eigenvectors and the arbitrary constants A and B. Hence trajectories are lines through the origin (steady state) and that steady state is known as a **Proper Node** (sometimes "star point"). Clearly it is asymptotically stable if $\lambda_1 = \lambda_2 < 0$ and is unstable if $\lambda_1 = \lambda_2 > 0$.

II. **There is one linearly independent eigenvector $\vec{v}_1$ and a generalised eigenvector $\vec{\eta}$**: In this case the solution is of the form $\vec{y}(t) = A\vec{v}_1 e^{\lambda_1 t} + B(\vec{v}_1 t e^{\lambda_1 t} + \vec{\eta} e^{\lambda_1 t})$.
In this case, if $\lambda_1 > 0$ the solutions clearly diverges away from 0 as $t \to \infty$ and are therefore unstable. If $\lambda_1 < 0$ - say $\lambda_1 = -r$ where $r > 0$, then by **L'Hôpital's rule**

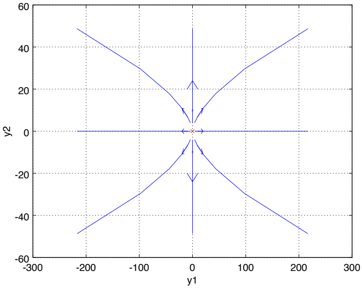
$$\lim_{t\to\infty} te^{-rt} = \lim_{t\to\infty} \frac{t}{e^{rt}} = \lim_{t\to\infty} \frac{\frac{d}{dt}(t)}{\frac{d}{dt}(e^{rt})} = \lim_{t\to\infty} \frac{1}{re^{rt}} = 0.$$

Thus solutions approach 0 as $t \to \infty$, therefore the steady state is asymptotically stable.

Proper Node



Improper Node

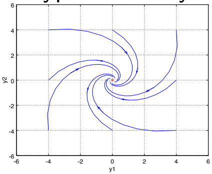


CASE 4: $\lambda_1, \lambda_2 = a \pm ib$ complex conjugate pair with nonzero real and imaginary parts

$\hookrightarrow$ From the earlier examples, the solutions tend to be the product of an exponential term (with exponent at) and a combination of sinusoidal terms.

- It is then easy to see that solutions will have oscillations as time increases, which will either dampen to the zero steady state solution (if $a < 0$) or be continually amplified (if $a > 0$). Hence in the first case the steady state is asymptotically stable and in the second case it is unstable.

$\hookrightarrow$ This type of steady state is called a **Spiral Point** (or focus).

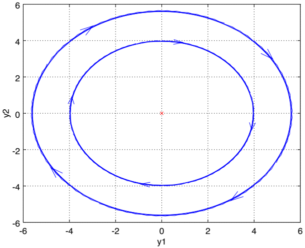


CASE 5: $\lambda_1, \lambda_2 = \pm ib$ complex conjugate pure imaginary

↪ This is as in the previous case but now without the exponential term, so just a sinusoidal solution.

- It is then easy to see that solutions will be periodic and will be represented in the phase plane by closed curves. The steady state is Stable but not Asymptotically Stable since solutions do not approach it over time (but don't diverge from it either).

↪ This type of steady state is called a **Centre**.



SUMMARY: Classification of Steady States for $\frac{d\vec{y}}{dt} = A\vec{y}$, $A_{2\times 2}$ a nonsingular matrix with eigenvalues λ_1 and λ_2:

Eigenvalue	Type of Steady State	Stability
$\lambda_1 > \lambda_2 > 0$	Node	Unstable
$\lambda_1 < \lambda_2 < 0$	Node	Asymptotically stable
$\lambda_1 < 0 < \lambda_2$	Saddle Point	Unstable
$\lambda_1 = \lambda_2 > 0$	Proper or Improper node	Unstable
$\lambda_1 = \lambda_2 < 0$	Proper or Improper node	Asymptotically stable
$\lambda_1, \lambda_2 = a \pm ib$	Spiral Point/Focus	$a < 0 \Rightarrow$ Asymptotically stable, $a > 0 \Rightarrow$ Unstable
$\lambda_1, \lambda_2 = \pm ib$	Centre	Stable

- For linear constant coefficient matrix systems of more than 2 ODEs, a similar (albeit more complicated) analysis can be carried out as was done here.
- The cases in higher dimensions are essentially just combinations of the different cases seen here. For example, for systems of two equations one could have a complex conjugate pair of solutions meaning that solutions along a certain plane may spiral to/from the origin while, for example if the other eigenvalue is a negative number other solutions could tend towards the origin along a line transverse to the plane in which solutions spiral.

- One nice thing that the topology of $\mathbb{R}^2$ allows is that local behaviour (near to steady states) in the phase plane can be generalised to global behaviour and one can get a good idea of how solutions behave everywhere in the plane. **This is not the case in higher dimensions**.

Classification of Steady States for Noninear Systems of ODEs

- The good news is we have done all of the required hard work in the previous sections.
- If the nonlinear autonomous system of ODEs $\boldsymbol{\frac{d\vec{y}}{dt} = \vec{F}(\vec{y})}$ has steady state $\vec{y}_0$ so that $\vec{F}(\vec{y}_0) = \vec{0}$, then a Taylor series expansion about $\vec{y}_0$ (assuming $\vec{F}$ is at least C^2), ignoring second and higher order terms, is $\boldsymbol{\vec{F}(\vec{y}) \approx \vec{0} + \vec{F}'(\vec{y}_0)(\vec{y} - \vec{y}_0)}$ where $\vec{F}'(\vec{y}_0)$ is the **Jacobian** matrix of $\vec{F}$ evaluated at the steady state $\vec{y}_0$. Noting that $\frac{d\vec{y}}{dt} = \frac{d(\vec{y}-\vec{y}_0)}{dt}$, then the differential equation becomes (approximately)

$$\frac{d\vec{y}}{dt} = \frac{d(\vec{y} - \vec{y}_0)}{dt} = \vec{F}'(\vec{y}_0)(\vec{y} - \vec{y}_0).$$

- Crucially, this is a linear constant coefficient system of ODEs with the Jacobian of $\vec{F}$ evaluated at the steady state being the coefficient matrix. It can be shown that the steady states of the nonlinear $\boldsymbol{\frac{d\vec{y}}{dt} = \vec{F}(\vec{y})}$ behave just like the steady states of this linearisation *(with one exception)*, **so we typically only need examine the eigenvalues of the matrix $\vec{F}'(\vec{y}_0)$ at each steady state of $\boldsymbol{\frac{d\vec{y}}{dt} = \vec{F}(\vec{y})}$ to determine the nature of that steady state.**

- The only case in which the earlier classification between linear systems and the nonlinear system may differ is highlighted in yellow in the following table, reproduced from earlier this lecture.

SUMMARY: Classification of Steady States for $\frac{d\vec{y}}{dt} = \vec{F}(\vec{y})$, where $\vec{F}'(\vec{y}_0)$ is a 2×2 Jacobian matrix evaluated at the steady state $\vec{y}_0$ which is nonsingular and has eigenvalues λ_1 and λ_2:

Eigenvalue	Type of Steady State	Stability
$\lambda_1 > \lambda_2 > 0$	Node	Unstable
$\lambda_1 < \lambda_2 < 0$	Node	Asymptotically stable
$\lambda_1 < 0 < \lambda_2$	Saddle Point	Unstable
$\lambda_1 = \lambda_2 > 0$	Proper or Improper node	Unstable
$\lambda_1 = \lambda_2 < 0$	Proper or Improper node	Asymptotically stable
$\lambda_1, \lambda_2 = a \pm ib$	Spiral Point/Focus	$a < 0 \Rightarrow$ Asymptotically stable, $a > 0 \Rightarrow$ Unstable
$\lambda_1, \lambda_2 = \pm ib$	Centre or Spiral Point	Asymptotically stable Stable, or Unstable

In the last, ambiguous case, check the nonlinear terms or use a direction field etc. to confirm the type and stability of the steady state.

- $\boxed{\text{EXAMPLE 24}}$: Returning to the macrophage model of EXAMPLE 23, use eigenvalues to classify the two steady state solutions $(0,0)$ and $\left(\frac{\alpha-\delta}{\delta\eta+\alpha},\ \frac{\alpha-\delta}{\alpha(\eta+1)}\right)$ in the model

$$\frac{dm}{dt} = \alpha(1-m)a - \delta m, \qquad \frac{da}{dt} = m - \eta ma - a.$$

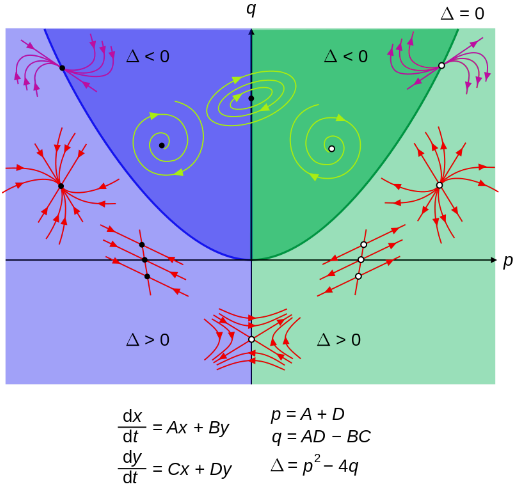
where $\alpha = 1, \delta = 0.2, \eta = 1$ (as in the graphs produced in EXAMPLE 23).

- $\boxed{\text{ANSWER}}$

- Similarly at the steady state

APPENDIX A

Here is an image summarising one way to categorise the steady states of a linear constant coefficient 2×2 system of ODEs without explicitly computing eigenvalues:



APPENDIX B - Inhomogeneous Systems $\vec{x}\,' = A\vec{x} + \vec{g}(t)$, $A_{n\times n}$

$\hookrightarrow$ There are several techniques for solving inhomogeneous systems; we will discuss two:

1. Diagonalisation (inhomgeneous systems for which the homogeneous part, $\vec{x}\,' = A\vec{x}$, has n linearly independent solutions, where $A_{n\times n}$ has n linearly independent eigenvectors).

2. The Method of Undetermined Coefficients for systems.

1. Diagonalisation Approach to Solving $\vec{x}\,' = A\vec{x} + \vec{g}(t)$

- First - review how to solve a first order linear ODE using an **integrating factor** from **Lecture 2** since it will be needed in what follows.

↪ Assuming A can be diagonalised so that $P^{-1}AP = D$ is a diagonal matrix (see the *Supplementary Lecture on Eigenvalues/Eigenvectors* for details), then $\boldsymbol{A = PDP^{-1}}$.

- So $\vec{x}\,' = A\vec{x} + \vec{g}(t)$ can be written as

$$\begin{aligned} \vec{x}\,' &= PDP^{-1}\vec{x} + \vec{g}(t) \Rightarrow \\ P^{-1}\vec{x}\,' &= DP^{-1}\vec{x} + P^{-1}\vec{g}(t) \Rightarrow \\ (P^{-1}\vec{x})\,' &= D(P^{-1}\vec{x}) + P^{-1}\vec{g}(t) \quad \text{since } P \text{ is a constant matrix.} \end{aligned}$$

↪ Because of the diagonal nature of D, each row of the last vector equation is simply an uncoupled first order linear ODE (of the form $\boldsymbol{y' = d_i y + h(t)}$) for the unknown $y_i = (P^{-1}\vec{x})_i$. So we simply solve the equations separately to obtain $P^{-1}\vec{x}$ then multiply on the left by P to obtain $\vec{x}(t)$.

- *NOTE recall that when using this diagonalisation approach with homogeneous linear systems of ODEs, we do not need to know P^{-1}. However, we DO need to know P^{-1} when solving the inhomogeneous system $\vec{x}\,' = A\vec{x} + \vec{g}(t)$ - in order to compute $P^{-1}\vec{g}(t)$.*

$\hookrightarrow$ $\boxed{\text{EXAMPLE 15}}$ Returning to EXAMPLE 2/9, we now solve the full inhomogeneous system :

$$\frac{d}{dt}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 4/3 & -1 \\ -2/3 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \frac{2}{3}e^t - 1 \\ -\frac{1}{3}e^t + 1 \end{bmatrix}.$$

$\hookrightarrow$ We now use the initial conditions $\vec{x}(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ to get

$$\begin{aligned} C_1 + 3C_2 &= 3/2 \\ C_1 - 2C_2 &= 3/2 \end{aligned}$$

$\Rightarrow C_2 = 0, \quad C_1 = \frac{3}{2}$. So the solution to the initial value problem is

$\boldsymbol{\vec{x}(t) = \begin{bmatrix} -\frac{1}{2}e^t + \frac{3}{2}e^{\frac{1}{3}t} \\ -1 + \frac{1}{2}e^t + \frac{3}{2}e^{\frac{1}{3}t} \end{bmatrix}}$, as expected.

2. Method of Undetermined Coefficients Approach to Solving $\vec{x}\,' = A\vec{x} + \vec{g}(t)$

(If pressed for time, you can ignore this is you have not seen the Method of Undetermined Coefficients before for solving constant coefficient linear 2nd order inhomogeneous ODEs)

- ↪ There isn't much new here if you have seen the Method of Undetermined Coefficients for second order constant coefficient linear ODEs. Basically, we can find a **particular solution** to $\boldsymbol{\vec{x}\,' = A\vec{x} + \vec{g}(t)}$ *in the special case where A is constant and $\vec{g}(t)$ contains sines, cosines, polynomials, exponential functions, or sums/products of these.*
- ↪ Again assume a solution $\vec{x}_p(t)$ of the form of the various entries of $\vec{g}(t)$ with undetermined coefficients, substitute this assumption for $\vec{x}_p(t)$ into $\boldsymbol{\vec{x}\,' = A\vec{x} + \vec{g}(t)}$, and find out the values of those coefficients.
- ↪ Once we have a **particular solution** $\vec{x}_p(t)$ to $\boldsymbol{\vec{x}\,' = A\vec{x} + \vec{g}(t)}$ and also know the *general solution of the HOMOGENEOUS system* $\boldsymbol{\vec{x}\,' = A\vec{x}}$ (the **complementary function**) $\vec{x}_c(t)$, then the GENERAL SOLUTION of $\boldsymbol{\vec{x}\,' = A\vec{x} + \vec{g}(t)}$ is simply

$$\vec{x}_p(t) + \vec{x}_c(t).$$

- ▶ *NOTE that this is identical to how we use the Method of Undetermined coefficients to solve linear single ODEs such as $ax'' + bx' + cx = g(t)$.*

$\hookrightarrow$ $\boxed{\text{EXAMPLE 16}}$ We will return to EXAMPLEs 9 and 15 to solve $\frac{d}{dt}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 4/3 & -1 \\ -2/3 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \frac{2}{3}e^t - 1 \\ -\frac{1}{3}e^t + 1 \end{bmatrix}$, *BUT now using the Method of Undetermined Coefficients.*

We already know, from EXAMPLE 9, the general solution of the homogeneous part of this equation: $\vec{x}_c(t) = B_1 \begin{bmatrix} 1 \\ 1 \end{bmatrix} e^{\frac{1}{3}t} + B_2 \begin{bmatrix} 3 \\ -2 \end{bmatrix} e^{2t}$.

Now we find a **particular integral** by observing $\vec{g}(t) = \begin{bmatrix} \frac{2}{3}e^t - 1 \\ -\frac{1}{3}e^t + 1 \end{bmatrix}$ and assuming

$\vec{x}_p(t) = \begin{bmatrix} ae^t + b \\ ce^t + d \end{bmatrix}$.

$\vec{x}_p\,'(t) = \begin{bmatrix} ae^t \\ ce^t \end{bmatrix}$, and substituting this into $\boldsymbol{\vec{x}\,' = A\vec{x} + \vec{g}(t)}$, we get

$$\begin{bmatrix} ae^t \\ ce^t \end{bmatrix} = \begin{bmatrix} 4/3 & -1 \\ -2/3 & 1 \end{bmatrix}\begin{bmatrix} ae^t + b \\ ce^t + d \end{bmatrix} + \begin{bmatrix} \frac{2}{3}e^t - 1 \\ -\frac{1}{3}e^t + 1 \end{bmatrix} =$$

$$\begin{bmatrix} \frac{4}{3}ae^t + \frac{4}{3}b - ce^t - d \\ -\frac{2}{3}ae^t - \frac{2}{3}b + ce^t + d \end{bmatrix} + \begin{bmatrix} \frac{2}{3}e^t - 1 \\ -\frac{1}{3}e^t + 1 \end{bmatrix}$$

REMINDER: $\vec{x}_c(t) = B_1 \begin{bmatrix} 1 \\ 1 \end{bmatrix} e^{\frac{1}{3}t} + B_2 \begin{bmatrix} 3 \\ -2 \end{bmatrix} e^{2t}$ and $\vec{x}_p(t) = \begin{bmatrix} ae^t + b \\ ce^t + d \end{bmatrix}$

This simplifies to
$$\begin{bmatrix} ae^t \\ ce^t \end{bmatrix} = \begin{bmatrix} (\frac{2}{3} + \frac{4}{3}a - c)e^t + \frac{4}{3}b - d - 1 \\ (-\frac{2}{3}a + c - \frac{1}{3})e^t - \frac{2}{3}b + d + 1 \end{bmatrix}.$$

Equating the coefficients of like terms on either side of the equations (and simplifying), we conclude that

$$\begin{aligned} a - 3c &= -2 \\ -2a &= 1 \quad \Rightarrow \quad \boxed{a = -\tfrac{1}{2}} \text{ and } \boxed{c = \tfrac{1}{2}}. \qquad \text{Also,} \end{aligned}$$

$$\begin{aligned} 4b - 3d &= 3 \\ -2b + 3d &= -3 \quad \Rightarrow \quad \boxed{b = 0} \text{ and } \boxed{d = -1}. \end{aligned}$$

So $\vec{x}_p(t) = \begin{bmatrix} -\frac{1}{2}e^t \\ \frac{1}{2}e^t - 1 \end{bmatrix}$ and the general solution to $\vec{x}\,' = A\vec{x} + \vec{g}(t)$ is

$$\begin{bmatrix} -\frac{1}{2}e^t \\ \frac{1}{2}e^t - 1 \end{bmatrix} + B_1 \begin{bmatrix} 1 \\ 1 \end{bmatrix} e^{\frac{1}{3}t} + B_2 \begin{bmatrix} 3 \\ -2 \end{bmatrix} e^{2t} \text{ (as expected).}$$

↪ FINAL NOTE: When solving $\vec{\boldsymbol{x}}\,' = \boldsymbol{A}\vec{\boldsymbol{x}} + \vec{\boldsymbol{g}}(\boldsymbol{t})$ using the Method of Undetermined Coefficients, there is only one case in which the approach differs slightly from that used in the solving of equations like $ax'' + bx' + cx = g(t)$.

⇝ *If the initial assumed form of the particular solution $\vec{x}_p(t) = \vec{a}e^{\lambda t}$, where λ is an eigenvalue of A (so that the term $\vec{a}e^{\lambda t}$ already appears in the complementary function), then instead of adjusting the assumption to $\vec{x}_p(t) = t\vec{a}e^{\lambda t}$, also include lower order terms in the assumption:* $\boxed{\vec{\boldsymbol{x}}_p(\boldsymbol{t}) = \boldsymbol{t}\vec{\boldsymbol{a}}\boldsymbol{e}^{\lambda t} + \vec{\boldsymbol{b}}\boldsymbol{e}^{\lambda t}}$, where $\vec{a}$ and $\vec{b}$ are constant vectors whose entries are to be determined by substitution into the ODE system $\vec{\boldsymbol{x}}\,' = \boldsymbol{A}\vec{\boldsymbol{x}} + \vec{\boldsymbol{g}}(\boldsymbol{t})$.