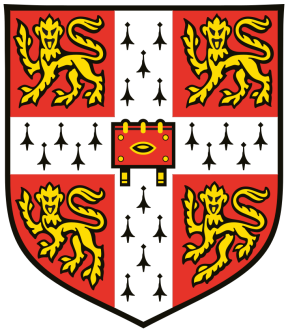
4.5

University of Cambridge

CATAM

Part II

# Euler's Equations



UNIVERSITY OF CAMBRIDGE

# Section 1: Investigating Euler's Equations

The program we will use to investigate the behaviour of Euler's Equations is written in MATLAB and the code is included at the end of this document. The aim is to solve Euler's equations numerically by implementing an RK-4 algorithm, which I have also programmed and included. Sample data for $A = 1.4$, $B = 1$, $C = 0.7$ is included in figure 1 below.

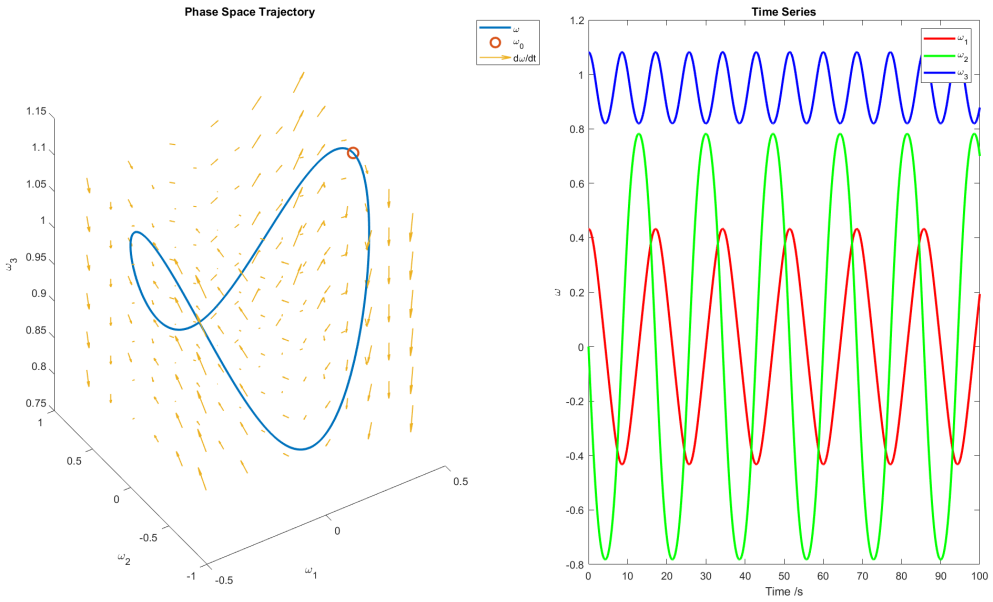



Figure 1: Sample program output for initial data $\omega_0 = (0.4, 0, 1)$.

# Question 1

For this question it will be convenient to refer to $A$ as $A_1$, $B$ as $A_2$ and $C$ as $A_3$. Since these parameters are distinct, there are 6 possible decreasing orderings. If this ordering is a cyclic permutation $\sigma$ of $A_1 > A_2 > A_3$ then by the change of variables $\nu_i = \omega_{\sigma(i)}$, Euler's equations for $\nu_i$, $(i = 1, 2, 3)$ have parameters in decreasing order. The other 3 decreasing orderings are cyclic permutations $\tau$ of the ordering $A_3 > A_2 > A_1$. By the change of variables $\nu_i = -\omega_{\tau(i)}$, we get Euler's equations in $\nu$ with parameters decreasing. So we can without loss of generality assume $A > B > C$.

If $A = B = C$, then the time-derivatives of all $w_i$'s vanish, so we have constant angular velocity in each component. For the case of 2 parameters equal without loss of generality take $A_1 = A_2 = \alpha$, $A_3 = \beta \neq \alpha$, then Euler's equations give $\omega_3 = k = \text{const}$. The equations for $\omega_1$ and $\omega_2$ are

$$\begin{pmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \end{pmatrix} = k\left(\frac{\beta - \alpha}{\alpha}\right) \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix} = M \begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix}. \tag{1}$$

These have general solution

$$\begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix} = \boldsymbol{C}_1 \exp(\lambda_1 t) + \boldsymbol{C}_2 \exp(\lambda_2 t), \tag{2}$$

where $\lambda_{1,2}$ are the eigenvalues of the matrix in (1). Since $\det M = k^2(\beta - \alpha)^2/\alpha^2 > 0$, and $\operatorname{Tr} M = 0$, we have that $\lambda_{1,2}$ are purely imaginary complex conjugates, so the solution (2) is bounded periodic motion for $\omega_1$ and $\omega_2$, while the motion for $\omega_3$ is constant.

We may multiply Euler's equations through by $1/B$ to yield the Euler's Equations with parameters $(A/B, 1, C/B)$. This will always work, since we have the values of $A/B$ and $C/B$ given, so $B \neq 0$. Thus the parameter $(A, B, c)$ generate the same general solution as $(A/B, 1, C/B)$.

Next, let $\hat{t} = \lambda t$ and $\hat{\boldsymbol{\omega}} = \lambda \boldsymbol{\omega}$. Then the re-scaled equations become

$$\begin{aligned}
\lambda^2(A\dot{\omega}_1 + (C-B)\omega_2\omega_3) &= 0,\\
\lambda^2(B\dot{\omega}_1 + (A-C)\omega_2\omega_3) &= 0,\\
\lambda^2(C\dot{\omega}_1 + (B-A)\omega_2\omega_3) &= 0.
\end{aligned}$$

So setting $\lambda = \sqrt{E}$ yields $\frac{1}{2}(A\omega_1^2 + B\omega_2^2 + C\omega_3^2) = 1$.

## Question 2

We begin by making a small perturbation in the $\omega_2$ direction to the point $\boldsymbol{\omega}_0 = (\sqrt{2/A}, 0, 0)$. Plotting a vector field of derivatives around $\boldsymbol{\omega}_0$ shows rotational symmetry in the $y-z$ plane and we do indeed observe periodic rotational motion in the $y-z$ plane, while we observe no motion in $\omega_1$.

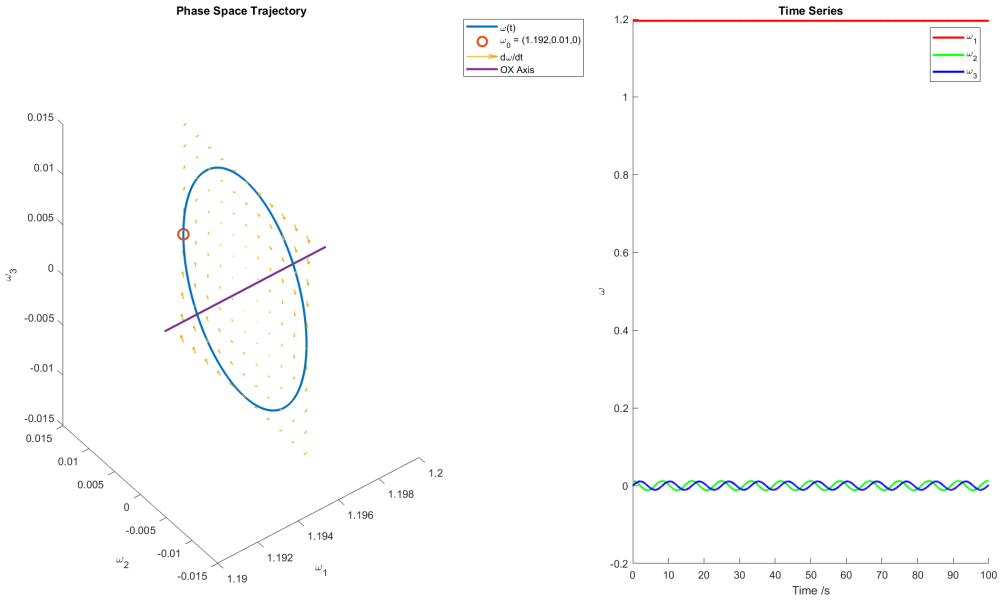



Figure 2: Small oscillations about the OX axis

## Question 3

i) Suppose $\omega_1 \approx \sqrt{2/A}$ and $\omega_1$, $\omega_2$ are small. Then, approximating Euler's Equations to first order in $||\mathbf{w}||$ yields

$$\begin{aligned}
\dot{\omega}_1 &= 0\\
B\dot{\omega}_2 + (A-C)\sqrt{2/A}\,\omega_3 &= 0\\
C\dot{\omega}_3 + (B-A)\sqrt{2/A}\,\omega_2 &= 0.
\end{aligned}$$

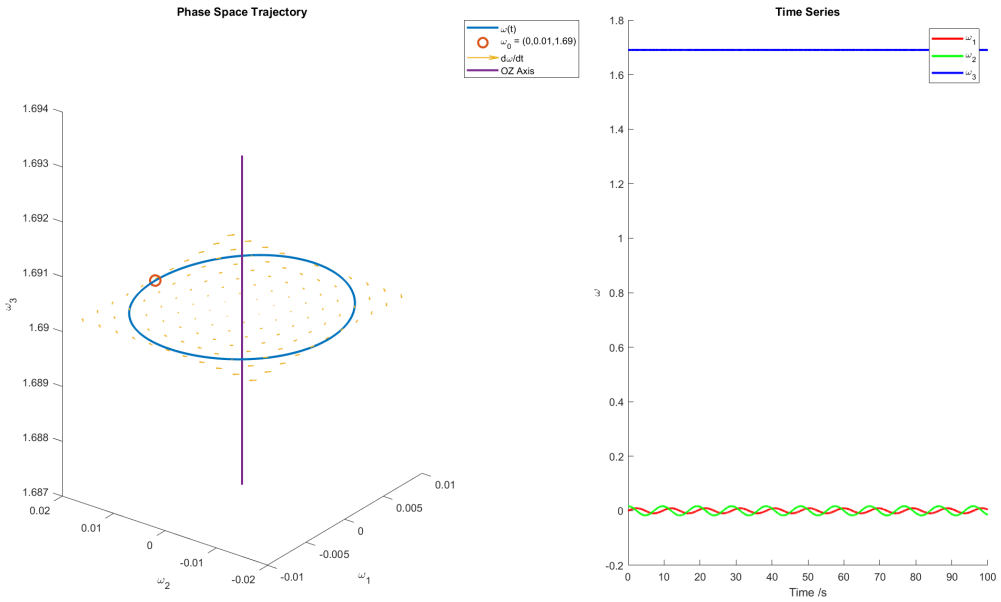



Figure 3: Small oscillations about the OZ axis

This has analytic solution

$$\begin{aligned}\omega_1 &= \sqrt{2/A}\\ \omega_2 &= D\sin\eta t + E\cos\eta t\\ \omega_3 &= F\sin\eta t + G\cos\eta t,\end{aligned}$$

where

$$\begin{aligned}\eta = \sqrt{\frac{2(A-C)(A-B)}{ABC}} = \frac{2}{\sqrt{7}} \approx 0.76\\ \implies T = \frac{2\pi}{\eta} = \pi\sqrt{7} \approx 8.31.\end{aligned}$$

ii) Continuing in the same way as above yields

$$\begin{aligned}A\dot{\omega}_1 + (C-B)\sqrt{2/C}\,\omega_2 = 0\\ B\dot{\omega}_2 + (A-C)\sqrt{2/C}\,\omega_1 = 0\\ \dot{\omega}_3 = 0.\end{aligned}$$

This has analytic solution

$$\begin{aligned}\omega_1 &= D\sin\eta t + E\cos\eta t\\ \omega_2 &= F\sin\eta t + G\cos\eta t\\ \omega_3 &= \sqrt{2/C},\end{aligned}$$

where

$$\eta = \sqrt{\frac{2(A-C)(B-C)}{ABC}} = \sqrt{\frac{3}{7}} \approx 0.65,$$

and

$$T = \frac{2\pi}{\eta} = 2\pi\sqrt{\frac{7}{3}} \approx 9.60.$$

These analytic solutions to the linearised problem approximate well the behaviour observed in Q2 above. Both sets of solutions give one constant solution and two oscillatory solutions, both with the same frequency. The value of the constant solutions, $\omega_1 = \sqrt{2/A}$ in (i) and $\omega_3 = \sqrt{2/C}$ in (ii), matches the value of our constant solutions and the period of oscillatory motion in both cases looks very accurate, with the linearised solutions predicting a slightly higher period in case (ii), just as observed.

In case (i), we have initial conditions $\omega_0 = (\sqrt{2/A}, \epsilon, 0)$, $\dot{\omega}_0 = (0, 0, \frac{4\sqrt{10}}{7\sqrt{7}}\epsilon)$ so $D = G = 0$, $E = \epsilon$, $F = \frac{2\sqrt{10}}{7}\epsilon \approx 0.904\epsilon$, so amplitudes are very similar, while for case (ii) we have $\omega_0 = (0, \epsilon, \sqrt{2/C})$ and $\dot{\omega}_0 = (\frac{3}{14}\sqrt{\frac{10}{7}}\epsilon, 0, 0)$, so $E = F = 0$, $G = \epsilon$ and $D = \frac{\sqrt{30}}{14}\epsilon \approx 0.391\epsilon$, a much larger difference in amplitudes.

With regards to equations (6) and (7), something can also be said here. Without loss of generality we work in case (i), as case (ii) will follow analogously. We compute

$$\begin{aligned} E(t) &= \frac{1}{2}(2 + \epsilon^2 \sin^2 \eta t + \frac{4}{7}\epsilon^2 \cos^2 \eta t) \\ &= 1 + \frac{2}{7}\epsilon^2 + \frac{3}{14}\epsilon^2 \sin^2 t \\ &= 1 + \mathcal{O}(\epsilon^2), \end{aligned}$$

with a similar result holding for $H$, so to first order in $\epsilon$, energy and angular momentum are indeed conserved.

# Question 4

RK-4 is a 4th-order approximation method, and we have used a stepsize of $h = 10^{-2}$, so we have

$$\begin{aligned} ||\tilde{\boldsymbol{\omega}}(100) - \boldsymbol{\omega}(100)|| &\leq Qh^4 \\ &= \mathcal{O}(10^{-8}), \end{aligned}$$

where $\tilde{\boldsymbol{\omega}}$ is the RK-4 approximation of $\boldsymbol{\omega}$. Then we have

$$\begin{aligned} \Delta E(100) &= |\tilde{E}(100) - E(100)| \\ &= \frac{1}{2}|A(\omega_1^2 - \tilde{\omega_1}^2) + B(\omega_2^2 - \tilde{\omega_2}^2) + C(\omega_3^2 - \tilde{\omega_3}^2)| \\ &\leq \frac{1}{2}A|\omega_1 - \tilde{\omega_1}||\omega_1 + \tilde{\omega_1}| \\ &+ \frac{1}{2}B|\omega_2 - \tilde{\omega_2}||\omega_2 + \tilde{\omega_2}| \\ &+ \frac{1}{2}C|\omega_3 - \tilde{\omega_3}||\omega_3 + \tilde{\omega_3}| \\ &= \mathcal{O}(10^{-8}|\epsilon|) \\ &= \mathcal{O}(10^{-10}), \end{aligned}$$

since we only considered perturbations of size $\epsilon = 10^{-2}$.

Relabelling $A$ by $A^2$, $B$ by $B^2$ and $C$ by $C^2$, and omitting the factor of $\frac{1}{2}$ shows that $\Delta H^2 = \mathcal{O}(10^{-10})$. Our results yield $\Delta E = 10^{-12}$ and $\Delta H^2 = 10{-}13$, which are well within the range, so

these are consistent with (6) and (7).

Further checks on numerical accuracy could be carried out by looking at $\Delta E$ and $\Delta H^2$ for larger values of time, for example $T = 1000$, though this is not necessary since the behaviour looks highly periodic, so it would probably suffice to simulate for only a few cycles and use these as the basis for long term predictions.

# Question 5

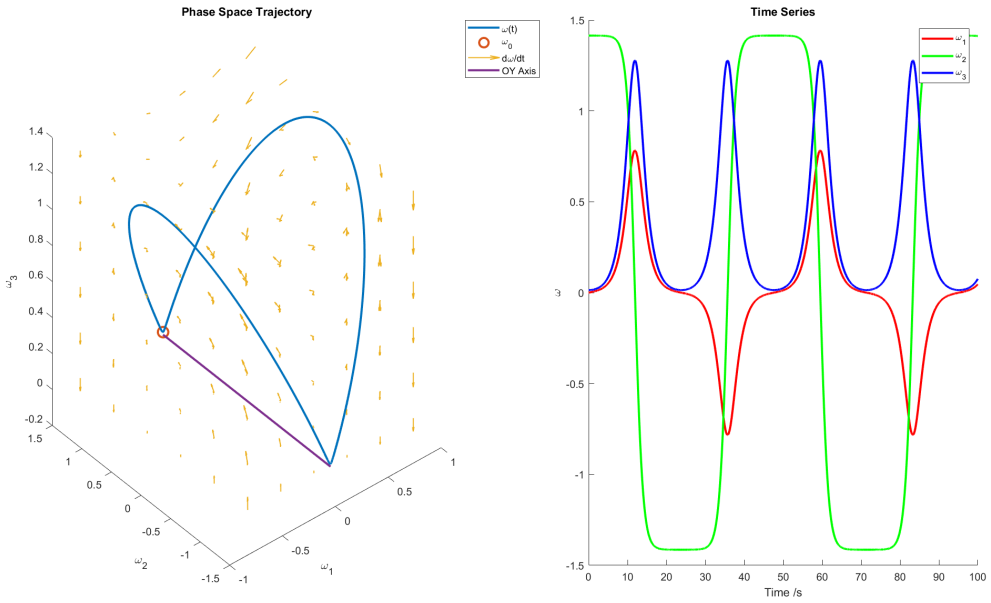



Figure 4: A small perturbation about the OY axis.

Figure 4 shows the solutions obtained by perturbing $\boldsymbol{\omega}_0$ from $(0, 1, 0)$. This behaviour is really intriguing and complex. We see that the fixed point $\boldsymbol{\omega} = (0, 1, 0)$ is highly unstable. The solution at this nudged initial condition is periodic, going on a wild excursion before returning. As the solution goes through a complete cycle, one of the angular velocities oscillates through positive and negative values ($\omega$, in figure 4), while one only oscillates through positive values ($\omega_3$ in figure 4). Physically, what this represents is that while rotating very close to the $OY$-axis, the rigid body flips along either the $OX$ or $OZ$ axis. In figure 4, the rigid body flips around the $OZ$-axis, while all rotation in the $OX$-axis cancels out.

# Question 6

The solution to Euler's equations in phase space is specified by the intersection of two ellipsoids, one given by energy conservation, and the other by angular momentum conservation:

$$\frac{1}{2}(A\omega_1^2 + B\omega_2^2 + C\omega_3^2) = 1 \tag{3}$$

$$A^2\omega_1^2 + B^2\omega_2^2 + C^2\omega_3^2 = H^2 \tag{4}$$

$$\tag{5}$$

Setting $\boldsymbol{\omega} = (0, \sqrt{2}, 0)$ yields $H = \sqrt{2}$. Subtracting (3) from (4) then gives

$$8\omega_1^2 - 3\omega_3^2 = 0 \tag{6}$$

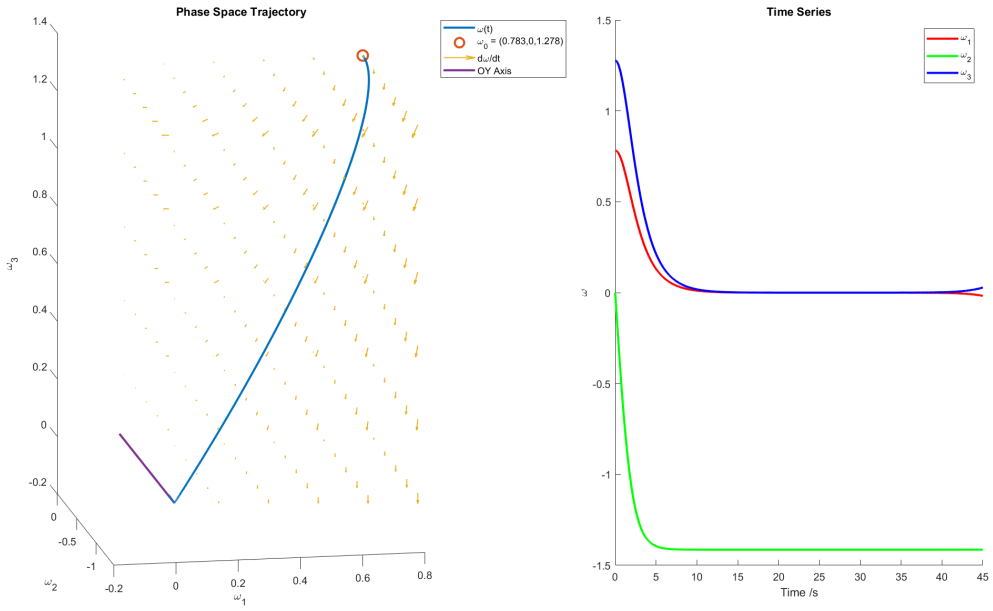



Figure 5: An approximation to a solution beginning away from the OY Axis but tending to some point along this axis.

Our trajectories in figures 4 and 5 are approximately smooth to (**??**), owing to the fact that finite precision and rounding errors always push our simulation off this critical unstable set. Since all solutions are restricted to their polhodes, and polhodes starting outside of (6) are closed and bounded (i.e. compact), its distance to $(0, \sqrt{2}, 0)$ will be bounded below. This means that our initial condition lying in (6) is a necessary condition for $\omega(t) \to (0, \pm\sqrt{2}, 0)$ as $t \to \infty$.

Furthermore, we can plot our vector field along (6) to show that all starting points in this set tend either to $(0, \sqrt{2}, 0)$ or to $(0, -\sqrt{2}, 0)$. This is given in figure 6.

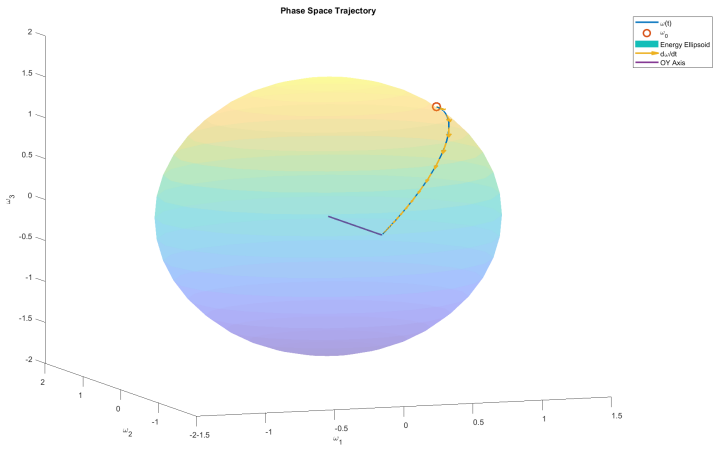



Figure 6: The vector field along the intersection of the ellipsoids tends to 0 as we approach the OY axis.

# Question 7

Since $A, B, C$ are distinct, we may assume without loss of generality that $A > 1, B = 1, C < 1$ and $E = 1$.

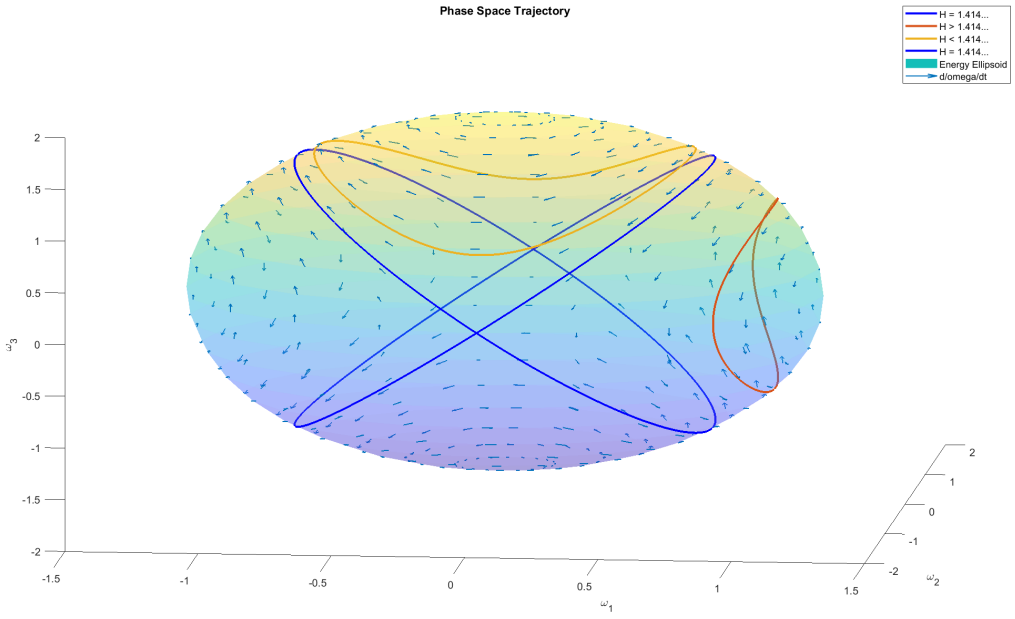



Figure 7: The 3 distinct types of solutions.

Figure 7 shows how the solution set in question 6 separates solutions into green and red solutions. Quantitatively, the behaviour for each solution class is as follows

$\boxed{\boldsymbol{H < \sqrt{2}}}$

- $\boldsymbol{\omega_0} = (\pm\sqrt{2/A}, 0, 0) : \boldsymbol{\omega} = \boldsymbol{\omega_0} =$ constant.
- Periodic precession around the stable rotational axis $OZ$. Stable against small perturbations in that for sufficiently small perturbations of $\boldsymbol{\omega_0}$, the resulting solution is in the same class.

$\boxed{\boldsymbol{H = \sqrt{2}}}$

- $\boldsymbol{\omega_0} = (0, \pm\sqrt{2}, 0) : \boldsymbol{\omega} = \boldsymbol{\omega_0} =$ constant.
- Otherwise, $\omega(t) \to (0, \pm\sqrt{2}, 0)$ as $t \to \infty$. This is very unstable against small perturbations.

$\boxed{\boldsymbol{H > \sqrt{2}}}$

- $\boldsymbol{\omega_0} = (0, 0, \pm\sqrt{2/C}) : \boldsymbol{\omega} = \boldsymbol{\omega_0} =$ constant.
- Otherwise, periodic precession around the stable $OX-$axis.

## Physical Behaviour

For the precession solution classes, the rigid body will either precess around the $OX$ or $OZ$ axis, depending on how close it is to both at the start. If it lies in the critical region, then it does not precess at all, but rather the axis of rotation slowly tends toward the intermediate $(OY)$ axis.

Near the boundary, we get progressively slower precession, with the majority of time spent by the rigid body rotating about an axis progressively closer to the $OY$ axis, followed by a short and sudden change in the rotation direction around the $OY$ axis, as demonstrated in figure 8 and this video.

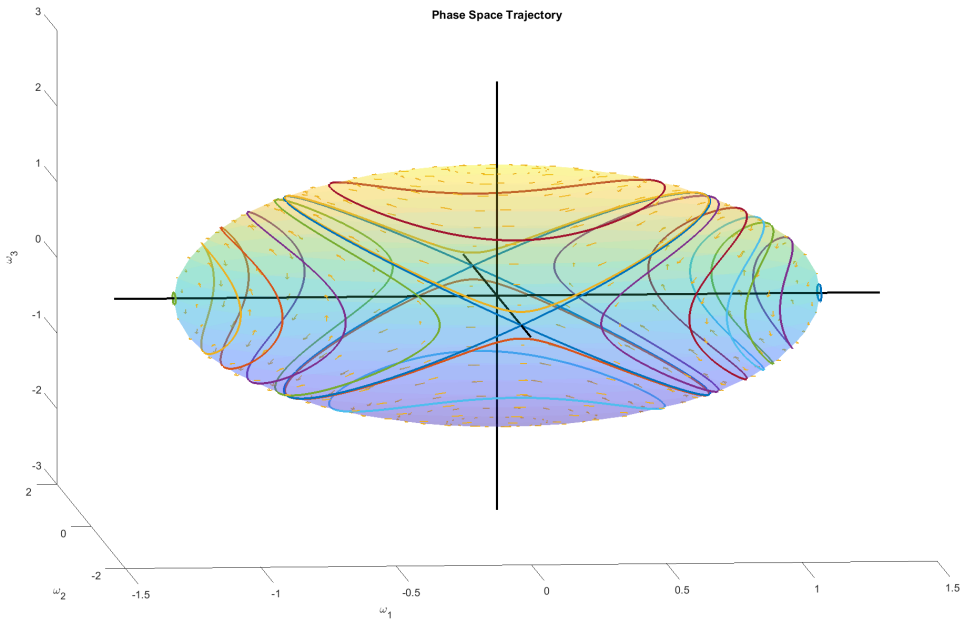
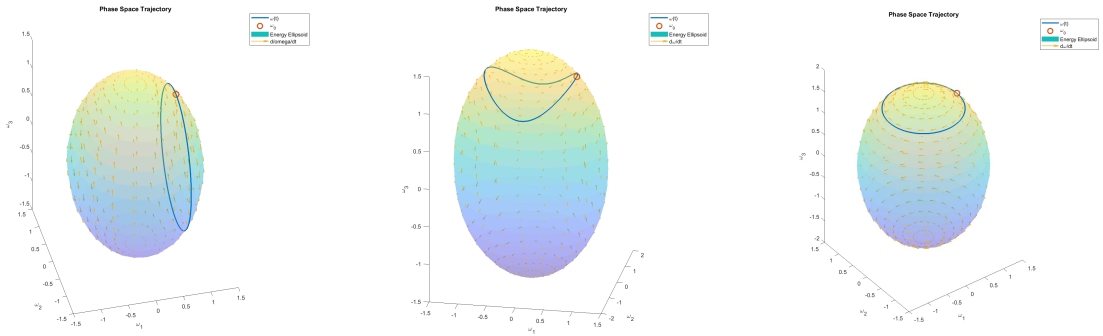



Figure 8: The behaviour of solutions as they tend toward the boundary.

Without loss of generality, assume $B = 1$, so that we care only about varying $A$ and $C$. At $A = 1$, we have $\dot{\omega}_3 = 0$, so all polhodes are horizontal, which at $C = 1, \dot{\omega}_1 = 0$, so all polhodes are vertical. These correspond to $\sqrt{\frac{A(A-1)}{C(1-C)}} = 0$ and $\infty$ respectively. As $\sqrt{\frac{A(A-1)}{C(C-1)}}$ varies between these extremes, the vector field shifts to allow combinations of vertical and horizontal polhodes, as in figure 9.2 below.

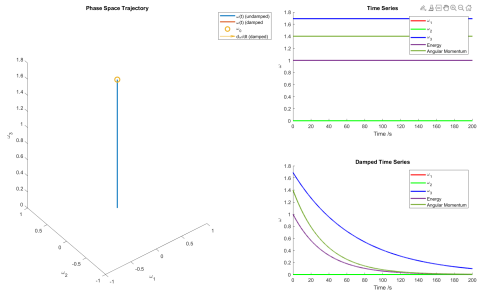
# Question 8

Equations (5) become

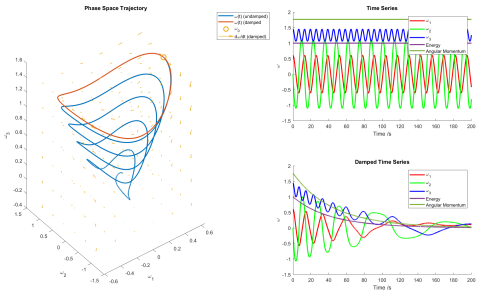



(a) 9.1 (b) 9.2 (c) 9.3

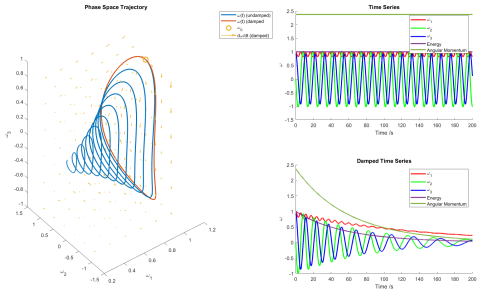
Figure 9: Varying parameters $A/B$ and $C/B$.



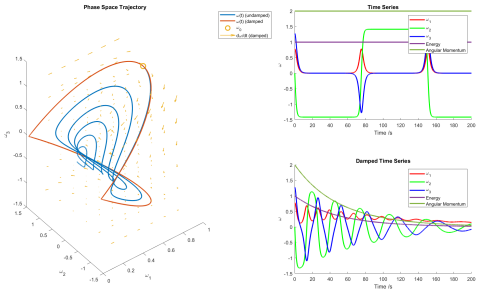
(a) 10.1



(b) 10.2



(c) 10.3



(d) 9.4

Figure 10: Different Solution Types for the Damped Equation.

$$\frac{d\boldsymbol{h}}{dt} + \boldsymbol{w} \times \boldsymbol{h} = -k\boldsymbol{w},$$

or in components:

$$\begin{aligned} A\dot{\omega}_1 + (C-B)\omega_2\omega_3 + k\omega_1 &= 0 \\ B\dot{\omega}_2 + (A-C)\omega_1\omega_3 + k\omega_2 &= 0 \\ C\dot{\omega}_3 + (B-A)\omega_1\omega_2 + k\omega_3 &= 0 \end{aligned}$$

Equations (6) and (7) no longer hold, since the presence of the external couple (torque) means that angular momentum is not conserved, and since work is done by the torque, kinetic energy is lost.

We see very dissipative behaviour with $\boldsymbol{\omega} \to 0$ in every case, with solutions parallel to coordinate axes approaching zero exponentially in a straight line, and other solutions spiralling in toward the origin exponentially fast. We also observe an exponential decay in $H^2$ and $E$, which is consistent with our assertion that these quantities are not conserved.

# Question 9

If we observe figure 10.4, we see that as we approach the boundary between the two 'precession regions', the damped solutions enter precession around the axis with a larger moment of inertia ($OX$). Therefore, the intermediate class solutions don't exist in the damped case. We also see that solutions starting in the first region (precession about $OZ$) may enter the third region, but not vice-versa. This is since at the boundary, small perturbations are caused by damping push solutions into region 3 rather than region 1. An example of this for the case $h = 0.01$ and $A, B, C$ standard is below.

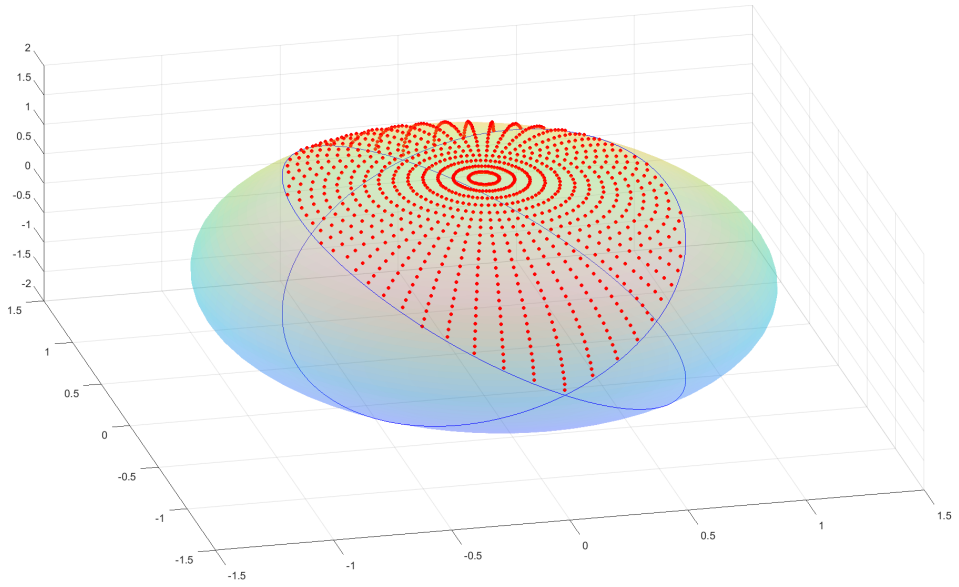



Figure 11: Starting points in the region above our intermediate solutions tend to the region below intermediate solutions in the damped case.

I believe that this is actually generic. If we take $h = 0.1$, simulate a collection of points in region 1, and colour them red if they end up in region 3, then we get the plot in figure 11. I believe that the same result will hold for smaller $h$, but the convergence to region 3 may be longer. The glaring exception is the point $(0, 0, \sqrt{2/C})$. This is since these solutions stay parallel to the OZ axis.

# Appendix

```
function f = func(T,W,A,B,C)
%This function is the RHS of Euler's Equations
%Although this func has no explicit T dependence, RK4 asks for a function
%of bot h x and T
    f = [(B-C)*W(2)*W(3)/A; (C-A)*W(1)*W(3)/B; (A-B)*W(1)*W(2)/C];
end
```

```
function [T,Y] = RK(a,b,f,h,n)

% T =[a,a+h,a+2h,..,a+nh]
% Y = RK approTimant on this interval
% a = T_0,
% b = Y_0;
% f = RHS of diff eqn,
% h = step size,
% n = No. of steps

k = zeros(length(b),4);
T = zeros(1,n+1);
Y = zeros(length(b),n+1);
```

```
T(1) = a;
Y(:,1) = b;

for i = 1:n
   T(i+1) = a+i*h;
   k(:,1)=h*f(T(i),Y(:,i));
   k(:,2)=h*f(T(i)+h/2,Y(:,i)+k(:,1)/2);
   k(:,3)=h*f(T(i)+h/2,Y(:,i)+k(:,2)/2);
   k(:,4)=h*f(T(i)+h,Y(:,i)+k(:,3));
   Y(:,i+1)=Y(:,i)+(k(:,1)+2*k(:,2)+2*k(:,3)+k(:,4))/6;
end

end
```

```
function [W,delta_E,delta_H] = Euler(A,B,C,h,n,W0)
% This code runs a numerical solution and creates various visualisations of
% important data to qualitatively understand the dynamics of Euler's
% Equations in phase space and in time series. It allows us to calculate
% the difference in energy and angular momentum from beginning to end to
% confirm whether or not our numerical simulations preserve these. Code
% also calls our Rk-4 solver and our vector field plotter. Later we
% introduce modified versions of this code to introduce a damping factor.

% A,B,C: princ. moments of inertia
% h = step size in RK4
% n = no. of steps in RK4
% W0 = initial conditions
% W = Solution

    [X,Y,Z] = ellipsoid(0,0,0,(2/A)^(0.5),(2/B)^(0.5),(2/C)^(0.5));
    % This is the Energy ellipsoid

    E0 = (A*W0(1)^2+B*W0(2)^2+C*W0(3)^2)/2; %Initial Energy IE
    W0 = W0/sqrt(E0);   %Rescaling IC so that IE = 1

    [T,W] = RK(0,W0,@(t,w) func(t,w,A,B,C),h,n);    %Numerical solution of W over time vector T

    E = (A*W(1,:).*W(1,:)+B*W(2,:).*W(2,:)+C*W(3,:).*W(3,:))/2; %Energy as func of time
    H = A^2*W(1,:).*W(1,:)+B^2*W(2,:).*W(2,:)+C^2*W(3,:).*W(3,:); % H as a func of time

    tiledlayout(1,2)    %This ensures that our output is nicely formatted

    ax1 = nexttile;
    view(3)
    hold on
    plot3(W(1,:),W(2,:),W(3,:));    %Plot of the solution found by numerical integration in phase
    plot3(W0(1),W0(2),W0(3),'o','MarkerSize',10);   %Plot point of the IC
    s = surf(X,Y,Z,'FaceAlpha',0.25);   %Plot of Energy Ellipsoid in phase space
    s.EdgeColor = 'None';
    vec_field(A,B,C,X,Y,Z); %Plot of Vec field
```

```
    xlabel("\omega_1");
    ylabel("\omega_2");
    zlabel("\omega_3");
    legend;
    title(ax1,'Phase Space Trajectory');

    ax2 = nexttile;
    hold on
    plot(T,W(1,:),'r',T,W(2,:),'g',T,W(3,:),'b');   %Plot of Time Series
    xlabel("Time /s")
    ylabel("\omega")
    legend;
    title(ax2,'Time Series')

    delta_E = E(n+1)-E(1);
    delta_H = H(n+1)-H(1);
end




function [] = vec_field(A,B,C,X,Y,Z)
%Plot a vector field of the system of ODEs for parameters A,B,C over set of points X,Y,Z

    U = ((B-C)/A)*(Y.*Z);   %X-Component of vec field
    V = ((C-A)/B)*(X.*Z);   %Y-Component of vec field
    W = ((A-B)/C)*(X.*Y);   %Z-Component of vec field

    quiver3(X,Y,Z,U,V,W,0.5,'Color',[0.9290 0.6940 0.1250],'LineWidth',1)
end




function f = modified_func(T,W,A,B,C,k)
%This is the modified equation of motion including damping
%Although this func has no explicit T dependence, RK4 asks for a function
%of bot h x and T
    f = [((B-C)*W(2)*W(3)-k*W(1))/A; ((C-A)*W(1)*W(3)-k*W(2))/B; ((A-B)*W(1)*W(2)-k*W(3))/C];
end



function [] = Modified_Euler(A,B,C,k,h,n,W0)
% Modified Euler Equation Code with damping factor
% A,B,C: princ. moments of inertia
% h = step size in RK4
% n = no. of steps in RK4
% W0 = initial conditions

    E0 = (A*W0(1)^2+B*W0(2)^2+C*W0(3)^2)/2; %Initial Energy IE
    W0 = W0/sqrt(E0);   %Rescaling IC so that IE = 1
```

```
    [T,W] = RK(0,W0,@(t,w) modified_func(t,w,A,B,C,k),h,n);    %Numerical solution of W over time
    [T,V] = RK(0,W0,@(t,w) func(t,w,A,B,C),h,n);    %Numerical solution of V over time vector T to

    dx = max(W(1,:))-min(W(1,:));
    dy = max(W(2,:))-min(W(2,:));
    dz = max(W(3,:))-min(W(3,:));
    x = min(W(1,:)):(dx/5):max(W(1,:));
    y = min(W(2,:)):(dy/5):max(W(2,:));
    z = min(W(3,:)):(dz/5):max(W(3,:));
    [X,Y,Z] = meshgrid(x,y,z);

    E = (A*W(1,:).*W(1,:)+B*W(2,:).*W(2,:)+C*W(3,:).*W(3,:))/2; %Energy as func of time
    H = A^2*W(1,:).*W(1,:)+B^2*W(2,:).*W(2,:)+C^2*W(3,:).*W(3,:); % H as a func of time

    tiledlayout(2,2)

    ax1 = nexttile(1,[2 1]);
    view(3)
    hold on
    plot3(W(1,:),W(2,:),W(3,:));    %Plot of the solution found by numerical integration
    plot3(V(1,:),V(2,:),V(3,:));    %Plot of the solution found by numerical integration
    plot3(W0(1),W0(2),W0(3),'o','MarkerSize',10);   %Plot of the IC
    modified_vec_field(A,B,C,k,X,Y,Z); %Plot of Vec field
    xlabel("\omega_1");
    ylabel("\omega_2");
    zlabel("\omega_3");
    legend('\omega(t) (undamped)','\omega(t) (damped','\omega_0','d\omega/dt (damped)');
    title(ax1,'Phase Space Trajectory');

    ax2 = nexttile(2,[1 1]);
    hold on
    plot(T,V(1,:),'r',T,V(2,:),'g',T,V(3,:),'b',T,ones(length(T),1),T,H(1)*ones(length(T),1));   %
    xlabel("Time /s")
    ylabel("\omega")
    legend('\omega_1','\omega_2','\omega_3','Energy','Angular Momentum');
    title(ax2,'Time Series')

    ax3 = nexttile(4,[1 1]);
    hold on
    plot(T,W(1,:),'r',T,W(2,:),'g',T,W(3,:),'b',T,E,T,H);   %Plot of Time Series
    xlabel("Time /s")
    ylabel("\omega")
    legend('\omega_1','\omega_2','\omega_3','Energy','Angular Momentum');
    title(ax3,'Damped Time Series')

    E(n+1)-E(1)
    H(n+1)-H(1)
    W0
end
```

```
function [] = modified_vec_field(A,B,C,k,X,Y,Z)
%Plot a vector field of the system of ODEs for parameters A,B,C over set of points X,Y,Z for the m

    U = ((B-C)/A)*(Y.*Z)-k*X/A;   %X-Component of vec field
    V = ((C-A)/B)*(X.*Z)-k*Y/B;   %Y-Component of vec field
    W = ((A-B)/C)*(X.*Y)-k*Z/C;   %Z-Component of vec field

    quiver3(X,Y,Z,U,V,W,0.25,'Color',[0.9290 0.6940 0.1250],'LineWidth',1)
end
```

```
function [] = IC_Classifier(n,N)
% This function classifies whether an initial condition of region 1 enters region 3. If so it puts
% n = number of iterations of RK4
% N = resolution of Inertia Ellipsoid


[X,Y,Z] = ellipsoid(0,0,0,(2/1.4)^(0.5),(2)^(0.5),(2/0.7)^(0.5),N); %Energy ellipsoid
view(3)
s = surf(X,Y,Z,'FaceAlpha',0.25);   %plot energy ellipsoid
s.EdgeColor = 'None';
hold on

for i = 1:N     %loop over i coords of points in energy ellipsoid
    for j = 1:N     %loop over j coords of points in energy ellipsoid
        if (8*X(i,j)^2-3*Z(i,j)^2 < 0 && Z(i,j)>0)
            W = Simple_Modified_Euler(1.4,1,0.7,0.1,0.01,n,[X(i,j),Y(i,j),Z(i,j)]);
            if abs(W(3,n)) < 8*abs(W(1,n))/3
                p = plot3(X(i,j),Y(i,j),Z(i,j));
                p.Color = 'red';
                p.Marker = '.';
                p.MarkerSize = 10;
            end
        end
    end
end

W0 = [sqrt(1/56),0,sqrt(1/21)];
E0 = (1.4*W0(1)^2+W0(2)^2+0.7*W0(3)^2)/2; %Initial Energy IE
W0 = W0/sqrt(E0);   %Rescaling IC so that IE = 1
[~,W] = RK(0,W0,@(t,w) func(t,w,1.4,1,0.7),0.01,n);
plot3(W(1,:),W(2,:),W(3,:),'b');

W0 = [-sqrt(1/56),0,-sqrt(1/21)];
E0 = (1.4*W0(1)^2+W0(2)^2+0.7*W0(3)^2)/2; %Initial Energy IE
W0 = W0/sqrt(E0);   %Rescaling IC so that IE = 1
[~,W] = RK(0,W0,@(t,w) func(t,w,1.4,1,0.7),0.01,n);
plot3(W(1,:),W(2,:),W(3,:),'b');
```