

# Dynamics of discrete clusters of weakly interacting waves in the Charney-Hasegawa-Mima equation.

Ellen O'Carroll

August 2021

## 1 Abstract

The Charney-Hasegawa-Mima (CHM) equation is a partial differential equation which is used to model Rossby waves in the atmosphere and drift waves in plasmas. We study the dynamics of clusters of waves that satisfy this equation. In this project we explore invariants of clusters, different initial amplitude conditions, average amplitude of the modes and phase analysis of the clusters.

## 2 Introduction

The CHM equation is used to model waves in the atmosphere and in plasmas. Waves that are solutions to this equation must satisfy a resonant condition  $\mathbf{k}_1 + \mathbf{k}_2 = \mathbf{k}_3$ , which is called a triad. This causes the solutions to the CHM equation to occur in clusters of connected triads with the wavevectors as the modes in the triads. The goal of this project is to understand the dynamics of these clusters such as the interactions between the modes in the clusters. These clusters have quantities that are invariant with time - the Hamiltonian which is a cubic invariant, quadratic invariants such as energy, enstrophy and the Manley-Rowe invariants. These invariants can have finite or infinite capacity depending on the order of the initial amplitudes of the modes in the cluster. Also the order of the average amplitude of each mode over time can be predicted by the order of the initial amplitude. Phase analysis was also conducted through the order parameter  $R(t)$  and the triad phases  $\Phi(t)$ .

## 3 Charney-Hasegawa-Mima Equation

The CHM equation is below where  $\psi(\mathbf{x}, t)$  is the streamfunction, and  $F_0$  and  $\beta$  are constants.

$$(\nabla^2 - F) \frac{\partial \psi}{\partial t} + \beta \frac{\partial \psi}{\partial x} + \frac{\partial \psi}{\partial x} \frac{\partial \nabla^2 \psi}{\partial y} - \frac{\partial \psi}{\partial y} \frac{\partial \nabla^2 \psi}{\partial x} = 0 \quad (1)$$

### 3.1 Fourier Series Expansion

It is helpful to transform the CHM equation into Fourier Series form for further analysis, as done before [1], so using Fourier mode expansion below and subbing into (1) gives (3).

$$\psi(\mathbf{x}, t) = \sum_{\mathbf{k} \in \mathbb{Z}^2} A_{\mathbf{k}} e^{i\mathbf{k} \cdot \mathbf{x}} + c.c. \quad (2)$$

$$\begin{aligned} \sum_{\mathbf{k} \in \mathbb{Z}^2} A_{\mathbf{k}} (-|\mathbf{k}|^2 - F) e^{i\mathbf{k} \cdot \mathbf{x}} + \beta \sum_{\mathbf{k} \in \mathbb{Z}^2} A_{\mathbf{k}} i k_x e^{i\mathbf{k} \cdot \mathbf{x}} + \sum_{\mathbf{k}_1 \in \mathbb{Z}^2} A_{\mathbf{k}_1} i k_{1x} e^{i\mathbf{k}_1 \cdot \mathbf{x}} \sum_{\mathbf{k}_2 \in \mathbb{Z}^2} A_{\mathbf{k}_2} (-|\mathbf{k}_2|^2) i k_{2y} e^{i\mathbf{k}_2 \cdot \mathbf{x}} \\ - \sum_{\mathbf{k}_3 \in \mathbb{Z}^2} A_{\mathbf{k}_3} i k_{3y} e^{i\mathbf{k}_3 \cdot \mathbf{x}} \sum_{\mathbf{k}_4 \in \mathbb{Z}^2} A_{\mathbf{k}_4} (-|\mathbf{k}_4|^2) i k_{4x} e^{i\mathbf{k}_4 \cdot \mathbf{x}} = 0 \end{aligned} \quad (3)$$

Changing the indexing from  $k_3$  and  $k_4$  to  $k_1$  and  $k_2$  in term 4 and splitting the third and fourth terms in half and switching the indexing in the new terms produces (4).

$$\begin{aligned}
& \sum_{\mathbf{k} \in \mathbb{Z}^2} \dot{A}_{\mathbf{k}}(-|\mathbf{k}|^2 - F)e^{i\mathbf{k} \cdot \mathbf{x}} + \beta \sum_{\mathbf{k} \in \mathbb{Z}^2} A_{\mathbf{k}} i k_x e^{i\mathbf{k} \cdot \mathbf{x}} + \frac{1}{2} \sum_{\mathbf{k}_1 \in \mathbb{Z}^2} A_{\mathbf{k}_1} k_{1x} e^{i\mathbf{k}_1 \cdot \mathbf{x}} \sum_{\mathbf{k}_2 \in \mathbb{Z}^2} A_{\mathbf{k}_2} (|\mathbf{k}_2|^2) k_{2y} e^{i\mathbf{k}_2 \cdot \mathbf{x}} \\
& + \frac{1}{2} \sum_{\mathbf{k}_2 \in \mathbb{Z}^2} A_{\mathbf{k}_2} k_{2x} e^{i\mathbf{k}_2 \cdot \mathbf{x}} \sum_{\mathbf{k}_1 \in \mathbb{Z}^2} A_{\mathbf{k}_1} (|\mathbf{k}_1|^2) k_{1y} e^{i\mathbf{k}_1 \cdot \mathbf{x}} - \frac{1}{2} \sum_{\mathbf{k}_1 \in \mathbb{Z}^2} A_{\mathbf{k}_1} k_{1y} e^{i\mathbf{k}_1 \cdot \mathbf{x}} \sum_{\mathbf{k}_2 \in \mathbb{Z}^2} A_{\mathbf{k}_2} (|\mathbf{k}_2|^2) k_{2x} e^{i\mathbf{k}_2 \cdot \mathbf{x}} \\
& - \frac{1}{2} \sum_{\mathbf{k}_2 \in \mathbb{Z}^2} A_{\mathbf{k}_2} k_{2y} e^{i\mathbf{k}_2 \cdot \mathbf{x}} \sum_{\mathbf{k}_1 \in \mathbb{Z}^2} A_{\mathbf{k}_1} (|\mathbf{k}_1|^2) k_{1x} e^{i\mathbf{k}_1 \cdot \mathbf{x}} = 0
\end{aligned} \tag{4}$$

Since the equation is equal to 0, multiply both sides by  $e^{-i\mathbf{k}' \cdot \mathbf{x}}$  and integrate over the volume of the domain.

$$\begin{aligned}
& \sum_{\mathbf{k} \in \mathbb{Z}^2} \dot{A}_{\mathbf{k}}(-|\mathbf{k}|^2 - F) \int_0^{2\pi} \int_0^{2\pi} e^{i(\mathbf{k}-\mathbf{k}') \cdot \mathbf{x}} dx dy + \beta \sum_{\mathbf{k} \in \mathbb{Z}^2} A_{\mathbf{k}} i k_x \int_0^{2\pi} \int_0^{2\pi} e^{i(\mathbf{k}-\mathbf{k}') \cdot \mathbf{x}} dx dy \\
& + \frac{1}{2} \sum_{\mathbf{k}_1 \in \mathbb{Z}^2} A_{\mathbf{k}_1} k_{1x} \sum_{\mathbf{k}_2 \in \mathbb{Z}^2} A_{\mathbf{k}_2} (|\mathbf{k}_2|^2) k_{2y} \int_0^{2\pi} \int_0^{2\pi} e^{i(\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}') \cdot \mathbf{x}} dx dy \\
& + \frac{1}{2} \sum_{\mathbf{k}_2 \in \mathbb{Z}^2} A_{\mathbf{k}_2} k_{2x} \sum_{\mathbf{k}_1 \in \mathbb{Z}^2} A_{\mathbf{k}_1} (|\mathbf{k}_1|^2) k_{1y} \int_0^{2\pi} \int_0^{2\pi} e^{i(\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}') \cdot \mathbf{x}} dx dy \\
& - \frac{1}{2} \sum_{\mathbf{k}_1 \in \mathbb{Z}^2} A_{\mathbf{k}_1} k_{1y} \sum_{\mathbf{k}_2 \in \mathbb{Z}^2} A_{\mathbf{k}_2} (|\mathbf{k}_2|^2) k_{2x} \int_0^{2\pi} \int_0^{2\pi} e^{i(\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}') \cdot \mathbf{x}} dx dy \\
& - \frac{1}{2} \sum_{\mathbf{k}_2 \in \mathbb{Z}^2} A_{\mathbf{k}_2} k_{2y} \sum_{\mathbf{k}_1 \in \mathbb{Z}^2} A_{\mathbf{k}_1} (|\mathbf{k}_1|^2) k_{1x} \int_0^{2\pi} \int_0^{2\pi} e^{i(\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}') \cdot \mathbf{x}} dx dy = 0
\end{aligned} \tag{5}$$

Since,

$$\int_0^{2\pi} \int_0^{2\pi} e^{i(\mathbf{k}-\mathbf{k}') \cdot \mathbf{x}} dx dy = \begin{cases} 0 & k \neq k' \\ 4\pi^2 & k = k' \end{cases} \tag{6}$$

and,

$$\int_0^{2\pi} \int_0^{2\pi} e^{i(\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}') \cdot \mathbf{x}} dx dy = \begin{cases} 0 & k_1 + k_2 \neq k' \\ 4\pi^2 & k_1 + k_2 = k' \end{cases} \tag{7}$$

(5) simplifies to:

$$\begin{aligned}
& \dot{A}_{\mathbf{k}}(-|\mathbf{k}|^2 - F)4\pi^2 + \beta A_{\mathbf{k}} i k_x 4\pi^2 + \frac{1}{2} \sum_{\mathbf{k}_1 \in \mathbb{Z}^2} A_{\mathbf{k}_1} k_{1x} \sum_{\mathbf{k}_2 \in \mathbb{Z}^2} A_{\mathbf{k}_2} (|\mathbf{k}_2|^2) k_{2y} 4\pi^2 \delta_{\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}} \\
& + \frac{1}{2} \sum_{\mathbf{k}_2 \in \mathbb{Z}^2} A_{\mathbf{k}_2} k_{2x} \sum_{\mathbf{k}_1 \in \mathbb{Z}^2} A_{\mathbf{k}_1} (|\mathbf{k}_1|^2) k_{1y} 4\pi^2 \delta_{\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}} - \frac{1}{2} \sum_{\mathbf{k}_1 \in \mathbb{Z}^2} A_{\mathbf{k}_1} k_{1y} \sum_{\mathbf{k}_2 \in \mathbb{Z}^2} A_{\mathbf{k}_2} (|\mathbf{k}_2|^2) k_{2x} 4\pi^2 \delta_{\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}} \\
& - \frac{1}{2} \sum_{\mathbf{k}_2 \in \mathbb{Z}^2} A_{\mathbf{k}_2} k_{2y} \sum_{\mathbf{k}_1 \in \mathbb{Z}^2} A_{\mathbf{k}_1} (|\mathbf{k}_1|^2) k_{1x} 4\pi^2 \delta_{\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}} = 0
\end{aligned} \tag{8}$$

Grouping terms gives

$$\begin{aligned}
& \dot{A}_{\mathbf{k}}(-|\mathbf{k}|^2 - F)4\pi^2 + \beta A_{\mathbf{k}} i k_x 4\pi^2 \\
& + \frac{1}{2} \sum_{\mathbf{k}_1, \mathbf{k}_2 \in \mathbb{Z}^2} A_{\mathbf{k}_1} A_{\mathbf{k}_2} 4\pi^2 \delta_{\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}} (k_{1x} k_{2y} |\mathbf{k}_2|^2 + k_{2x} k_{1y} |\mathbf{k}_1|^2 - k_{1y} k_{2x} |\mathbf{k}_2|^2 - k_{2y} k_{1x} |\mathbf{k}_1|^2) = 0
\end{aligned} \tag{9}$$

This simplifies to

$$\dot{A}_{\mathbf{k}}(-|\mathbf{k}|^2 - F) + \beta A_{\mathbf{k}} i k_x + \frac{1}{2} \sum_{\mathbf{k}_1, \mathbf{k}_2 \in \mathbb{Z}^2} A_{\mathbf{k}_1} A_{\mathbf{k}_2} \delta_{\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}} (k_{1x} k_{2y} - k_{2x} k_{1y}) (|\mathbf{k}_2|^2 - |\mathbf{k}_1|^2) = 0 \tag{10}$$

Letting

$$Z_{\mathbf{k}_1\mathbf{k}_2}^{\mathbf{k}} = \frac{(k_{1x}k_{2y} - k_{2x}k_{1y})(|\mathbf{k}_1|^2 - |\mathbf{k}_2|^2)}{|\mathbf{k}|^2 + F} \quad (11)$$

$$w_{\mathbf{k}} = \frac{-\beta k_x}{|\mathbf{k}|^2 + F} \quad (12)$$

Equation (10) becomes

$$\dot{A}_{\mathbf{k}} + iw_{\mathbf{k}}A_{\mathbf{k}} + \frac{1}{2} \sum_{\mathbf{k}_1\mathbf{k}_2 \in \mathbb{Z}^2} A_{\mathbf{k}_1}A_{\mathbf{k}_2}\delta_{\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}}Z_{\mathbf{k}_1\mathbf{k}_2}^{\mathbf{k}} = 0 \quad (13)$$

So the final equation is

$$\dot{A}_{\mathbf{k}} + iw_{\mathbf{k}}A_{\mathbf{k}} = -\frac{1}{2} \sum_{\mathbf{k}_1\mathbf{k}_2 \in \mathbb{Z}^2} A_{\mathbf{k}_1}A_{\mathbf{k}_2}\delta_{\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}}Z_{\mathbf{k}_1\mathbf{k}_2}^{\mathbf{k}} \quad (14)$$

This form in (14) is easier to work with as it shows that solutions must be in the form of triads that satisfy  $\mathbf{k}_1 + \mathbf{k}_2 = \mathbf{k}_3$ .

### 3.2 Restricting $k_x$ to positive values only

By restricting  $k_x$  to positive values only, the wavevector space is reduced by half which simplifies the process of finding wavevectors which satisfy the triad relation. Taking the simplified CHM equation

$$\dot{A}_{\mathbf{k}} + iw_{\mathbf{k}}A_{\mathbf{k}} = -\frac{1}{2} \sum_{\mathbf{k}_1\mathbf{k}_2 \in \mathbb{Z}^2} A_{\mathbf{k}_1}A_{\mathbf{k}_2}\delta_{\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}}Z_{\mathbf{k}_1\mathbf{k}_2}^{\mathbf{k}} \quad (15)$$

Split the right hand side into 4 separate summations

$$\begin{aligned} \dot{A}_{\mathbf{k}} + iw_{\mathbf{k}}A_{\mathbf{k}} = & -\frac{1}{2} \sum_{\mathbf{k}_1\mathbf{k}_2 \in \mathbb{Z}^+ \times \mathbb{Z}} A_{\mathbf{k}_1}A_{\mathbf{k}_2}\delta_{\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}}Z_{\mathbf{k}_1\mathbf{k}_2}^{\mathbf{k}} - \frac{1}{2} \sum_{\mathbf{k}_1 \in \mathbb{Z}^+ \times \mathbb{Z}} \sum_{\mathbf{k}_2 \in \mathbb{Z}^- \times \mathbb{Z}} A_{\mathbf{k}_1}A_{\mathbf{k}_2}\delta_{\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}}Z_{\mathbf{k}_1\mathbf{k}_2}^{\mathbf{k}} \\ & - \frac{1}{2} \sum_{\mathbf{k}_1 \in \mathbb{Z}^- \times \mathbb{Z}} \sum_{\mathbf{k}_2 \in \mathbb{Z}^+ \times \mathbb{Z}} A_{\mathbf{k}_1}A_{\mathbf{k}_2}\delta_{\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}}Z_{\mathbf{k}_1\mathbf{k}_2}^{\mathbf{k}} - \frac{1}{2} \sum_{\mathbf{k}_1\mathbf{k}_2 \in \mathbb{Z}^- \times \mathbb{Z}} A_{\mathbf{k}_1}A_{\mathbf{k}_2}\delta_{\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}}Z_{\mathbf{k}_1\mathbf{k}_2}^{\mathbf{k}} \end{aligned} \quad (16)$$

Since

$$A_{-\mathbf{k}} = A_{\mathbf{k}}^* \quad (17)$$

Equation (16) can be rewritten as

$$\begin{aligned} \dot{A}_{\mathbf{k}} + iw_{\mathbf{k}}A_{\mathbf{k}} = & -\frac{1}{2} \sum_{\mathbf{k}_1\mathbf{k}_2 \in \mathbb{Z}^+ \times \mathbb{Z}} A_{\mathbf{k}_1}A_{\mathbf{k}_2}\delta_{\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}}Z_{\mathbf{k}_1\mathbf{k}_2}^{\mathbf{k}} - \frac{1}{2} \sum_{\mathbf{k}_1 \in \mathbb{Z}^+ \times \mathbb{Z}} \sum_{\mathbf{k}_2 \in \mathbb{Z}^+ \times \mathbb{Z}} A_{\mathbf{k}_1}A_{\mathbf{k}_2}^*\delta_{\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}}(-Z_{\mathbf{k}_1\mathbf{k}_2}^{\mathbf{k}}) \\ & - \frac{1}{2} \sum_{\mathbf{k}_1 \in \mathbb{Z}^+ \times \mathbb{Z}} \sum_{\mathbf{k}_2 \in \mathbb{Z}^+ \times \mathbb{Z}} A_{\mathbf{k}_1}^*A_{\mathbf{k}_2}\delta_{\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}}(-Z_{\mathbf{k}_1\mathbf{k}_2}^{\mathbf{k}}) - \frac{1}{2} \sum_{\mathbf{k}_1\mathbf{k}_2 \in \mathbb{Z}^+ \times \mathbb{Z}} A_{\mathbf{k}_1}^*A_{\mathbf{k}_2}^*\delta_{\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}}Z_{\mathbf{k}_1\mathbf{k}_2}^{\mathbf{k}} \end{aligned} \quad (18)$$

So restricting  $\mathbf{k} \in \mathbb{Z}^+ \times \mathbb{Z}$ , (18) becomes

$$\begin{aligned} \dot{A}_{\mathbf{k}} + iw_{\mathbf{k}}A_{\mathbf{k}} = & -\frac{1}{2} \sum_{\mathbf{k}_1\mathbf{k}_2 \in \mathbb{Z}^+ \times \mathbb{Z}} \delta_{\mathbf{k}_1+\mathbf{k}_2-\mathbf{k}}(A_{\mathbf{k}_1}A_{\mathbf{k}_2}Z_{\mathbf{k}_1\mathbf{k}_2}^{\mathbf{k}} - A_{\mathbf{k}_1}A_{\mathbf{k}_2}^*Z_{\mathbf{k}_1\mathbf{k}_2}^{\mathbf{k}} \\ & - A_{\mathbf{k}_1}^*A_{\mathbf{k}_2}Z_{\mathbf{k}_1\mathbf{k}_2}^{\mathbf{k}} + A_{\mathbf{k}_1}^*A_{\mathbf{k}_2}^*Z_{\mathbf{k}_1\mathbf{k}_2}^{\mathbf{k}}) \end{aligned} \quad (19)$$

### 3.3 Equations of motion for triad

Equations of motion for one triad.

Using the substitution

$$a_{\mathbf{k}} = A_{\mathbf{k}} e^{-i w_{\mathbf{k}} t} \quad (20)$$

equation (15) becomes

$$\dot{a}_{\mathbf{k}} = -\frac{1}{2} \sum_{\mathbf{k}_1, \mathbf{k}_2 \in \mathbb{Z}^2} a_{\mathbf{k}_1} a_{\mathbf{k}_2} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} Z_{\mathbf{k}_1, \mathbf{k}_2}^{\mathbf{k}} e^{-i w_{\mathbf{k}_1, \mathbf{k}_2}^{\mathbf{k}} t} \quad (21)$$

where

$$w_{\mathbf{k}_1, \mathbf{k}_2}^{\mathbf{k}} = w_{\mathbf{k}_1} + w_{\mathbf{k}_2} - w_{\mathbf{k}} \quad (22)$$

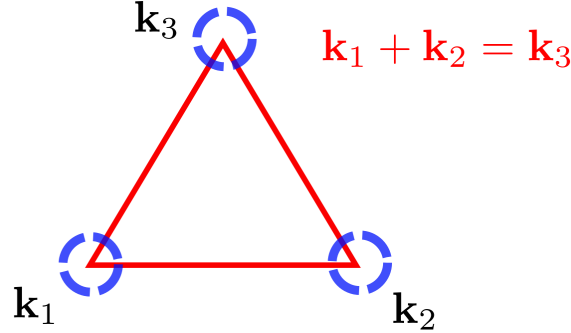


Figure 1: Triad with  $\mathbf{k}_1 = (1,-8)$ ,  $\mathbf{k}_2 = (15,10)$  and  $\mathbf{k}_3 = (16,2)$

Taking the triad above where (1,-8) and (15,10) are passive modes and (16,2) is an active mode. An active mode is the mode that corresponds to  $\mathbf{k}_3$  in the equation  $\mathbf{k}_1 + \mathbf{k}_2 = \mathbf{k}_3$ , and the passive modes are the other 2 modes. Each triad has one active and two passive modes. Allow the variable  $a_1$  correspond to the value for (1,-8),  $a_2$  correspond to the value for (15,10) and  $a_3$  correspond to the value for (16,2), the equations of motion are as follows:

**For  $a_1$**

$$\dot{a}_1 = -\frac{1}{2} \sum_{\mathbf{k}_1, \mathbf{k}_2 \in \mathbb{Z}^2} a_{\mathbf{k}_1} a_{\mathbf{k}_2} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} Z_{\mathbf{k}_1, \mathbf{k}_2}^{\mathbf{k}} \quad (23)$$

since  $w_{\mathbf{k}_1, \mathbf{k}_2}^{\mathbf{k}} = 0$ .

The only  $\mathbf{k}$  values that fit for this triad, when consider it by itself are  $\mathbf{k} = -\mathbf{K}_2(15,10) + \mathbf{K}_3(16,2)$ .

$$\dot{a}_1 = -\frac{1}{2} (a_2^* a_3 Z_{-\mathbf{K}_2, \mathbf{K}_3}^{\mathbf{k}} + a_2^* a_3 Z_{-\mathbf{K}_2, \mathbf{K}_3}^{\mathbf{k}}) \quad (24)$$

The same term appears twice due to the  $\mathbf{k}_2$  equalling  $\mathbf{K}_2$  and  $\mathbf{K}_3$  and same for  $\mathbf{k}_3$ . Subbing in the triad values, with  $F = 0$ , the equation becomes

$$\dot{a}_1 = -130 a_2^* a_3 \quad (25)$$

**For  $a_2$**

$\mathbf{k} = -\mathbf{K}_2(15,10) + \mathbf{K}_3(16,2)$  Similarly,

$$\dot{a}_2 = -78 a_1^* a_3 \quad (26)$$

**For  $a_3$**

$\mathbf{k} = \mathbf{K}_1(1,-8) + \mathbf{K}_2(15,10)$  Similarly,

$$\dot{a}_3 = 130 a_1 a_2 \quad (27)$$

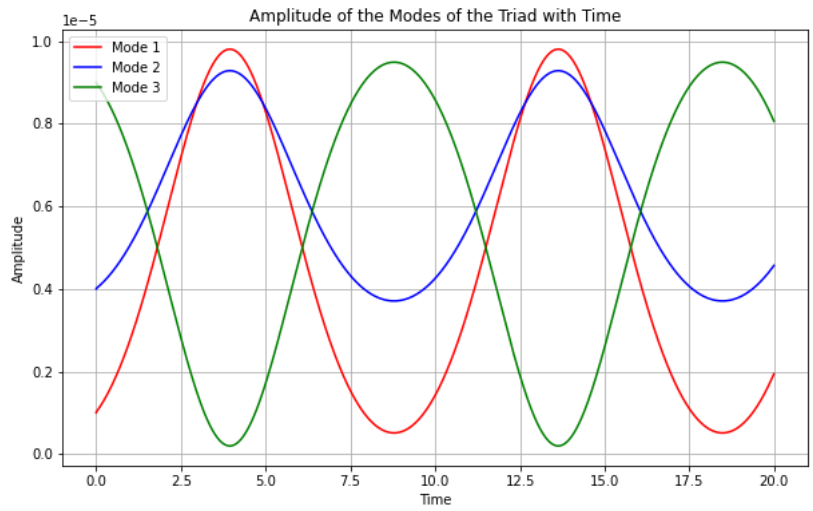


Figure 2: This shows the amplitudes of the three modes in the triad over time with a set of initial conditions.

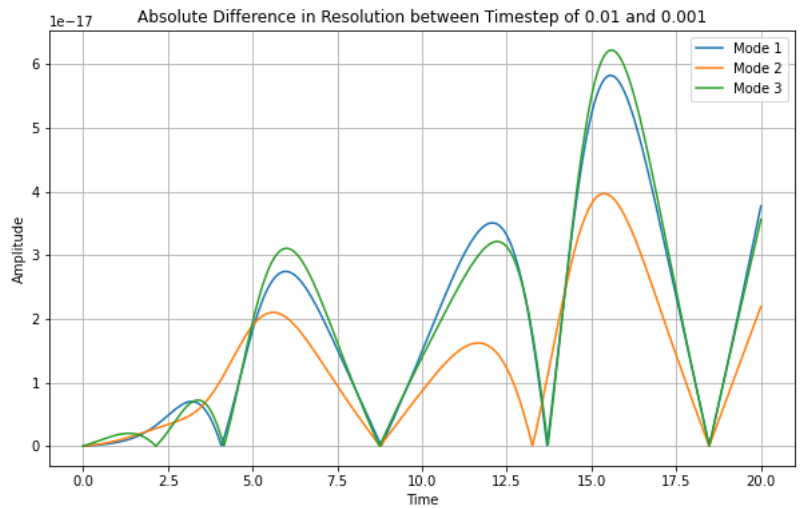


Figure 3: This shows the absolute difference in amplitude between the timestep of 0.01 and 0.001 of the three modes in the triad over time with a set of initial conditions.

The relative error of the L2 norm between the amplitudes of the two different timesteps is  $1.2025472552945099e-12$ . This indicates that the error is low between timesteps so it is possible to increase timestep in order to ensure a faster output of results.

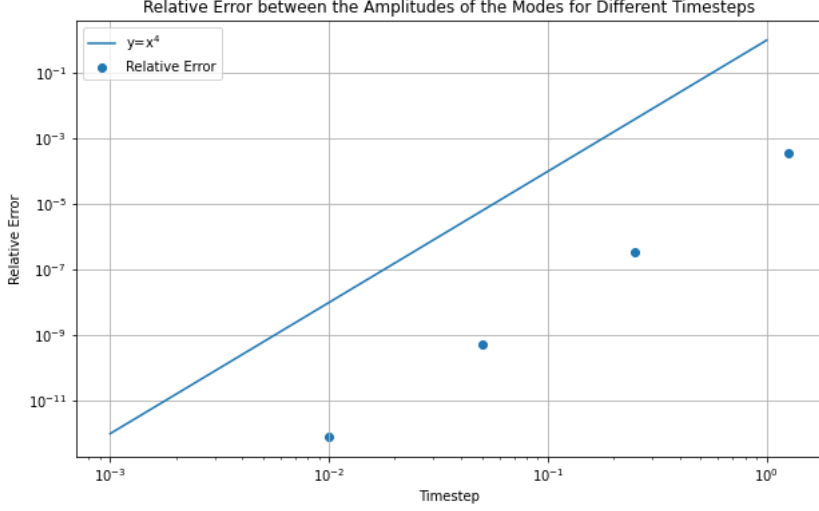


Figure 4: This shows the relative error of the amplitudes of the modes in the triad for different timesteps with the same set of initial conditions. The error increases by  $10^4$  for every increase in timestep by a factor of 10.

### 3.4 Hamiltonian form of CHM Equation

The Hamiltonian is an invariant quantity of a triad or any cluster of triads. In order to calculate this invariant, it is easiest to transform the Fourier series expansion of the CHM to Hamiltonian form, as explained in Nazarenko's Wave Turbulence [2].

Taking the Taylor series expansion for of the CHM equation in (14).

$$\dot{A}_{\mathbf{k}} + iw_{\mathbf{k}}A_{\mathbf{k}} = -\frac{1}{2} \sum_{\mathbf{k}_1, \mathbf{k}_2 \in \mathbb{Z}^2} A_{\mathbf{k}_1} A_{\mathbf{k}_2} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} Z_{\mathbf{k}_1, \mathbf{k}_2}^{\mathbf{k}} \quad (28)$$

Using the transformation

$$b_{\mathbf{k}} = \frac{|\mathbf{k}|^2 + F}{\sqrt{\beta k_x}} A_{\mathbf{k}} \quad (29)$$

Equation (28) becomes

$$\frac{\sqrt{\beta k_x}}{|\mathbf{k}|^2 + F} \dot{b}_{\mathbf{k}} + iw_{\mathbf{k}} \frac{\sqrt{\beta k_x}}{|\mathbf{k}|^2 + F} b_{\mathbf{k}} = -\frac{\beta}{2} \sum_{\mathbf{k}_1, \mathbf{k}_2 \in \mathbb{Z}^2} b_{\mathbf{k}_1} b_{\mathbf{k}_2} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} (|k_{1x} k_{2x}|)^{1/2} \frac{(k_{1x} k_{2y} - k_{2x} k_{1y})(|\mathbf{k}_1|^2 - |\mathbf{k}_2|^2)}{(|\mathbf{k}|^2 + F)(|\mathbf{k}_1|^2 + F)(|\mathbf{k}_2|^2 + F)} \quad (30)$$

Which simplifies to

$$\dot{b}_{\mathbf{k}} + iw_{\mathbf{k}} b_{\mathbf{k}} = -\frac{\beta^{1/2}}{2} \sum_{\mathbf{k}_1, \mathbf{k}_2 \in \mathbb{Z}^2} b_{\mathbf{k}_1} b_{\mathbf{k}_2} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} (|k_x k_{1x} k_{2x}|)^{1/2} \frac{(k_{1x} k_{2y} - k_{2x} k_{1y})(|\mathbf{k}_1|^2 - |\mathbf{k}_2|^2)}{k_x (|\mathbf{k}_1|^2 + F)(|\mathbf{k}_2|^2 + F)} \quad (31)$$

Since

$$\begin{aligned} k_{1x} &= k_x - k_{2x} \\ k_{1y} &= k_y - k_{2y} \end{aligned} \quad (32)$$

$$\frac{k_{1x} k_{2y} - k_{2x} k_{1y}}{k_x} = k_{2y} - \frac{k_{2x} k_y}{k_x} \quad (33)$$

Also, since  $F = 0$

$$\frac{(|\mathbf{k}_1|^2 - |\mathbf{k}_2|^2)}{(|\mathbf{k}_1|^2 + F)(|\mathbf{k}_2|^2 + F)} = \frac{1}{|\mathbf{k}_2|^2 + F} - \frac{1}{|\mathbf{k}_1|^2 + F} \quad (34)$$

Therefore,

$$\begin{aligned} \frac{(k_{1x}k_{2y} - k_{2x}k_{1y})(|\mathbf{k}_1|^2 - |\mathbf{k}_2|^2)}{k_x(|\mathbf{k}_1|^2 + F)(|\mathbf{k}_2|^2 + F)} &= \frac{k_{2y}}{|\mathbf{k}_2|^2 + F} - \frac{k_{2y}}{|\mathbf{k}_1|^2 + F} - \frac{k_y}{k_x} \left( \frac{k_{2x}}{|\mathbf{k}_2|^2 + F} - \frac{k_{2x}}{|\mathbf{k}_1|^2 + F} \right) \\ &= \frac{k_{2y}}{|\mathbf{k}_2|^2 + F} - \frac{k_{2y}}{|\mathbf{k}_1|^2 + F} - \frac{k_y}{k_x} \left( \frac{k_{2x}}{|\mathbf{k}_2|^2 + F} - \frac{k_x - k_{1x}}{|\mathbf{k}_1|^2 + F} \right) \\ &= \frac{k_{2y}}{|\mathbf{k}_2|^2 + F} - \frac{k_{2y}}{|\mathbf{k}_1|^2 + F} - \frac{k_y}{|\mathbf{k}_2|^2 + F} + \frac{k_y}{|\mathbf{k}_1|^2 + F} \\ &= \frac{k_{1y}}{|\mathbf{k}_1|^2 + F} + \frac{k_{2y}}{|\mathbf{k}_2|^2 + F} - \frac{k_y}{|\mathbf{k}_1|^2 + F} \end{aligned} \quad (35)$$

(31) becomes

$$\dot{b}_{\mathbf{k}} + iw_{\mathbf{k}}b_{\mathbf{k}} = -\frac{\beta^{1/2}}{2} \sum_{\mathbf{k}_1, \mathbf{k}_2 \in \mathbb{Z}^2} b_{\mathbf{k}_1} b_{\mathbf{k}_2} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} (|k_x k_{1x} k_{2x}|)^{1/2} \left( \frac{k_{1y}}{|\mathbf{k}_1|^2 + F} + \frac{k_{2y}}{|\mathbf{k}_2|^2 + F} - \frac{k_y}{|\mathbf{k}|^2 + F} \right) \quad (36)$$

Multiplying by  $i$  gives the Hamiltonian form of the CHM equation

$$i\dot{b}_{\mathbf{k}} = w_{\mathbf{k}}b_{\mathbf{k}} + \sum_{\mathbf{k}_1, \mathbf{k}_2 \in \mathbb{Z}^2} b_{\mathbf{k}_1} b_{\mathbf{k}_2} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} V_{12k} \quad (37)$$

with

$$V_{12k} = -\frac{i\beta^{1/2}}{2} (|k_x k_{1x} k_{2x}|)^{1/2} \left( \frac{k_{1y}}{|\mathbf{k}_1|^2 + F} + \frac{k_{2y}}{|\mathbf{k}_2|^2 + F} - \frac{k_y}{|\mathbf{k}|^2 + F} \right) \quad (38)$$

### 3.5 Proof in Hamiltonian Form

In order for an equation to be in Hamiltonian form it must satisfy that  $\frac{\partial \mathcal{H}}{\partial b_{\mathbf{k}}^*} = i\dot{b}_{\mathbf{k}}$  where  $\mathcal{H} = \mathcal{H}_2 + \mathcal{H}_3 + \dots$  and

$$\mathcal{H}_2 = \sum_{\mathbf{k}} \omega_{\mathbf{k}} |b_{\mathbf{k}}|^2 \quad (39)$$

and

$$\mathcal{H}_3 = \sum_{\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3 \in \mathbb{Z}^2} (V_{123} b_{\mathbf{k}_1} b_{\mathbf{k}_2} b_{\mathbf{k}_3}^* \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} + c.c.) + (U_{123} \delta_{\mathbf{k}_1 + \mathbf{k}_2 + \mathbf{k}} b_{\mathbf{k}_1} b_{\mathbf{k}_2} b_{\mathbf{k}_3} + c.c.) \quad (40)$$

So to prove (41) split the equation in two parts.

$$\frac{\partial \mathcal{H}}{\partial b_{\mathbf{k}}^*} = i\dot{b}_{\mathbf{k}} = w_{\mathbf{k}}b_{\mathbf{k}} + \sum_{\mathbf{k}_1, \mathbf{k}_2 \in \mathbb{Z}^2} V_{12k} b_{\mathbf{k}_1} b_{\mathbf{k}_2} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} \quad (41)$$

Starting with  $\mathcal{H}_2$

$$\begin{aligned} \frac{\partial \mathcal{H}_2}{\partial b_{\mathbf{k}}^*} &= \frac{\partial}{\partial b_{\mathbf{k}}^*} \left( \sum_{\mathbf{k}} \omega_{\mathbf{k}} |b_{\mathbf{k}}|^2 \right) \\ &= \frac{\partial}{\partial b_{\mathbf{k}}^*} \left( \sum_{\mathbf{k}} \omega_{\mathbf{k}} b_{\mathbf{k}} b_{\mathbf{k}}^* \right) \\ &= \sum_{\mathbf{k}} \omega_{\mathbf{k}} b_{\mathbf{k}} \end{aligned} \quad (42)$$

Moving onto  $\mathcal{H}_3$ , restrict  $k_x$  positive so that

$$\sum_{\mathbf{k}_1 \mathbf{k}_2 \in \mathbb{Z}^2} V_{12k} b_{\mathbf{k}_1} b_{\mathbf{k}_2} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} = \sum_{\mathbf{k}_1 \mathbf{k}_2 \in \mathbb{Z}^2} (V_{12k} b_{\mathbf{k}_1} b_{\mathbf{k}_2} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} + V_{-12k} b_{\mathbf{k}_1}^* b_{\mathbf{k}_2} \delta_{-\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} + V_{1-2k} b_{\mathbf{k}_1} b_{\mathbf{k}_2}^* \delta_{\mathbf{k}_1 - \mathbf{k}_2 - \mathbf{k}} + V_{-1-2k} b_{\mathbf{k}_1}^* b_{\mathbf{k}_2}^* \delta_{-\mathbf{k}_1 - \mathbf{k}_2 - \mathbf{k}}) \quad (43)$$

The last term of (43) is zero due to the Kronecker delta as with the restriction that all  $k_x$  are positive this term is always zero. Also, the U term of  $\mathcal{H}_3$  is zero.

$$\begin{aligned} \frac{\partial \mathcal{H}_3}{\partial b_{\mathbf{k}}^*} &= \frac{\partial}{\partial b_{\mathbf{k}}^*} \left( \sum_{\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3 \in \mathbb{Z}^2} (V_{123} b_{\mathbf{k}_1} b_{\mathbf{k}_2} b_{\mathbf{k}_3}^* \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} + V_{123}^* b_{\mathbf{k}_1}^* b_{\mathbf{k}_2}^* b_{\mathbf{k}_3} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}}) \right) \\ &= \sum_{\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3 \in \mathbb{Z}^2} \left( V_{123} \frac{\partial}{\partial b_{\mathbf{k}}^*} (b_{\mathbf{k}_1} b_{\mathbf{k}_2} b_{\mathbf{k}_3}^*) \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} + V_{123}^* \frac{\partial}{\partial b_{\mathbf{k}}^*} (b_{\mathbf{k}_1}^* b_{\mathbf{k}_2}^* b_{\mathbf{k}_3}) \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} \right) \\ &= \sum_{\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3 \in \mathbb{Z}^2} (V_{123} b_{\mathbf{k}_1} b_{\mathbf{k}_2} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} + V_{123}^* (b_{\mathbf{k}_2}^* b_{\mathbf{k}_3} + b_{\mathbf{k}_1}^* b_{\mathbf{k}_3}) \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}}) \end{aligned} \quad (44)$$

By changing the indexing in the second term where the 3 becomes a 1, and the 1 becomes a k, and in the third term 3 becomes a 2 and the 2 a k, (44) becomes

$$\frac{\partial \mathcal{H}_3}{\partial b_{\mathbf{k}}^*} = \sum_{\mathbf{k}_1 \mathbf{k}_2 \mathbf{k} \in \mathbb{Z}^2} (V_{12k} b_{\mathbf{k}_1} b_{\mathbf{k}_2} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} + V_{k21}^* b_{\mathbf{k}_2}^* b_{\mathbf{k}_1} \delta_{\mathbf{k} + \mathbf{k}_2 - \mathbf{k}_1} + V_{1k2}^* b_{\mathbf{k}_1}^* b_{\mathbf{k}_2} \delta_{\mathbf{k}_1 + \mathbf{k} - \mathbf{k}_2}) \quad (45)$$

Since  $V_{k21}^* = V_{1-2k}$  and  $V_{1k2}^* = V_{-12k}$ , the RHS of (45) is equivalent to the RHS of (43), so (41) is true and the CHM is in Hamiltonian form.

### 3.6 Hamiltonian is Real

The Hamiltonian (below) is a real invariant, this can be seen

$$\mathcal{H} = \sum_{\mathbf{k}} \omega_{\mathbf{k}} |b_{\mathbf{k}}|^2 + \sum_{\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3 \in \mathbb{Z}^2} (V_{123} b_{\mathbf{k}_1} b_{\mathbf{k}_2} b_{\mathbf{k}_3}^* \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} + V_{123}^* b_{\mathbf{k}_1}^* b_{\mathbf{k}_2}^* b_{\mathbf{k}_3} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}}) \quad (46)$$

The Hamiltonian (above) is a real invariant, this can be seen as the first term is real as it contains only the magnitude squared of the b-variable, also the complex conjugate of the second term is equal to the third term and vice versa. Since the complex conjugate of the Hamiltonian is equal to the Hamiltonian, the Hamiltonian is real.

### 3.7 Hamiltonian is conserved

Since the Hamiltonian is an invariant it must be conserved. By using the chain rule this can be seen.

$$\frac{\partial \mathcal{H}}{\partial t} = \frac{\partial \mathcal{H}}{\partial b_{\mathbf{k}}} \dot{b}_{\mathbf{k}} + \frac{\partial \mathcal{H}}{\partial b_{\mathbf{k}}^*} \dot{b}_{\mathbf{k}}^* \quad (47)$$

$$\frac{\partial \mathcal{H}}{\partial b_{\mathbf{k}}^*} = i \dot{b}_{\mathbf{k}} \quad (48)$$

$$\frac{\partial \mathcal{H}}{\partial b_{\mathbf{k}}} = -i \dot{b}_{\mathbf{k}}^*$$

So

$$\frac{\partial \mathcal{H}}{\partial t} = -i \dot{b}_{\mathbf{k}}^* \dot{b}_{\mathbf{k}} + i \dot{b}_{\mathbf{k}} \dot{b}_{\mathbf{k}}^* = 0 \quad (49)$$



## 4 Invariants

### 4.1 Invariants for Triad

Taking the triad (1,-8) (15,10) (16,2) as before and allowing the variable  $b_1$  correspond to the value for (1,-8),  $b_2$  correspond to the value for (15,10) and  $b_3$  correspond to the value for (16,2), using the same process as in 1.3, the equations of motion in Hamiltonian form are as follows:

$$\begin{aligned} \dot{b}_1 &= \frac{i\beta}{65}b_1 - \frac{2\beta^{1/2}\sqrt{15}}{5}b_2^*b_3 \\ \dot{b}_2 &= \frac{3i\beta}{65}b_2 - \frac{2\beta^{1/2}\sqrt{15}}{5}b_1^*b_3 \\ \dot{b}_3 &= \frac{4i\beta}{65}b_3 + \frac{2\beta^{1/2}\sqrt{15}}{5}b_1b_2 \end{aligned} \quad (50)$$

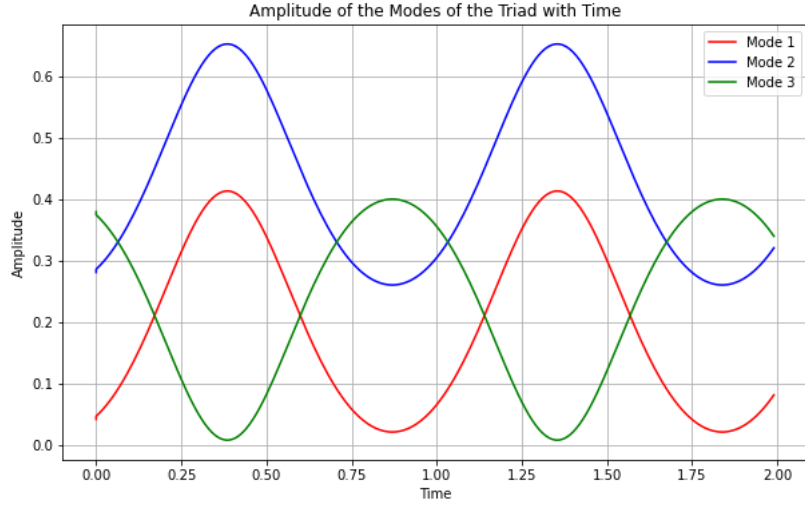


Figure 5: This shows the amplitudes of the three modes in the triad in  $b$ -variables over time with a set of initial conditions.

#### 4.1.1 Hamiltonian for Triad

$$\mathcal{H} = \sum_{\mathbf{k}} \omega_{\mathbf{k}} |b_{\mathbf{k}}|^2 + \sum_{\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3 \in \mathbb{Z}^2} (V_{123} b_{\mathbf{k}_1} b_{\mathbf{k}_2} b_{\mathbf{k}_3}^* \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} + V_{123}^* b_{\mathbf{k}_1}^* b_{\mathbf{k}_2}^* b_{\mathbf{k}_3} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}}) \quad (51)$$

Substituting the values for the triad

$$\mathcal{H} = -\frac{\beta}{65}|b_1|^2 - \frac{3\beta}{65}|b_2|^2 - \frac{4\beta}{65}|b_3|^2 - \frac{6i\beta^{1/2}\sqrt{15}}{5}(b_1^*b_2^*b_3 - b_1b_2b_3^*) \quad (52)$$

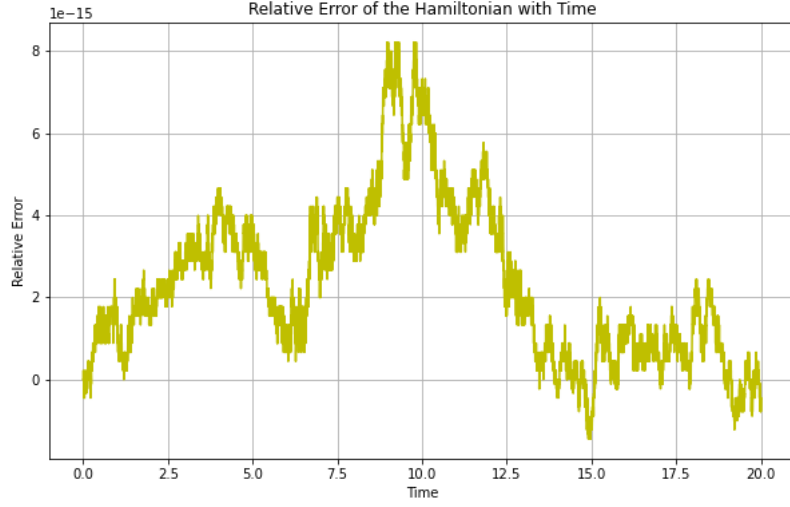


Figure 6: This shows the relative error between the initial Hamiltonian and the Hamiltonian at each timestep. As the error is  $\sim e - 15$ , the Hamiltonian for the triad is invariant.

#### 4.1.2 Energy and Enstrophy for Triad

Energy and enstrophy are quadratic invariants for a cluster. The general form of the energy and enstrophy are sums over the number of modes ( $M$ ) in the cluster.

$$Energy = \sum_i^M |\mathbf{k}_i|^2 |A_{\mathbf{k}_i}|^2 \quad (53)$$

$$Enstrophy = \sum_i^M |\mathbf{k}_i|^4 |A_{\mathbf{k}_i}|^2 \quad (54)$$

For ease transforming the energy and enstrophy to  $b$ -variables ensures that the invariants use the same variables as the Hamiltonian.

$$Energy = \sum_i^M \frac{\beta k_{i,x}}{|\mathbf{k}_i|^2} |b_{\mathbf{k}_i}|^2 \quad (55)$$

$$Enstrophy = \sum_i^M \beta k_{i,x} |b_{\mathbf{k}_i}|^2 \quad (56)$$

As the triad has 3 modes the Energy and Enstrophy are as follows

$$Energy = \frac{\beta}{65} |b_{\mathbf{k}_1}|^2 + \frac{3\beta}{65} |b_{\mathbf{k}_2}|^2 + \frac{4\beta}{65} |b_{\mathbf{k}_3}|^2 \quad (57)$$

$$Enstrophy = \beta |b_{\mathbf{k}_1}|^2 + 15\beta |b_{\mathbf{k}_2}|^2 + 16\beta |b_{\mathbf{k}_3}|^2 \quad (58)$$

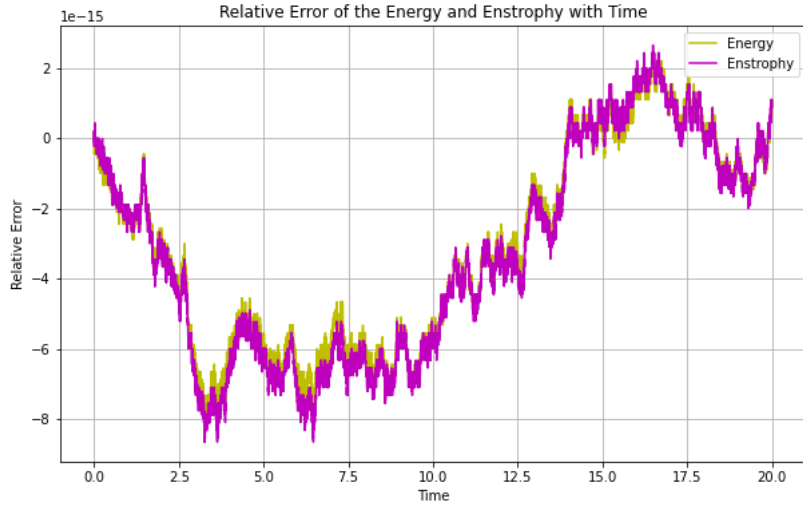


Figure 7: This shows the relative error between the initial Energy and Enstrophy and the Energy and Enstrophy at each timestep. As the error is  $\sim e - 15$ , the Energy and Enstrophy for the triad are invariant.

#### 4.1.3 Manley-Rowe Variants for Triad

The Manley-Rowe Invariants are quadratic invariants that are found using the nullspace of the N by M cluster matrix of the cluster of triads which has N triads and M modes [3]. Each row is a triad and each column is a mode. Passive modes are represented by a 1 and active modes a -1, if an entry is 0 in a row then it means that that mode is not in that triad.

The cluster matrix for the triad is  $\begin{bmatrix} 1 & 1 & -1 \end{bmatrix}$ .

The nullspace of this cluster matrix is  $\begin{bmatrix} 1 & -1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$ .

Meaning the Manley-Rowe Invariants are

$$\begin{aligned} MR_1 &= |b_1|^2 + |b_3|^2 \\ MR_2 &= |b_2|^2 - |b_1|^2 \end{aligned} \tag{59}$$

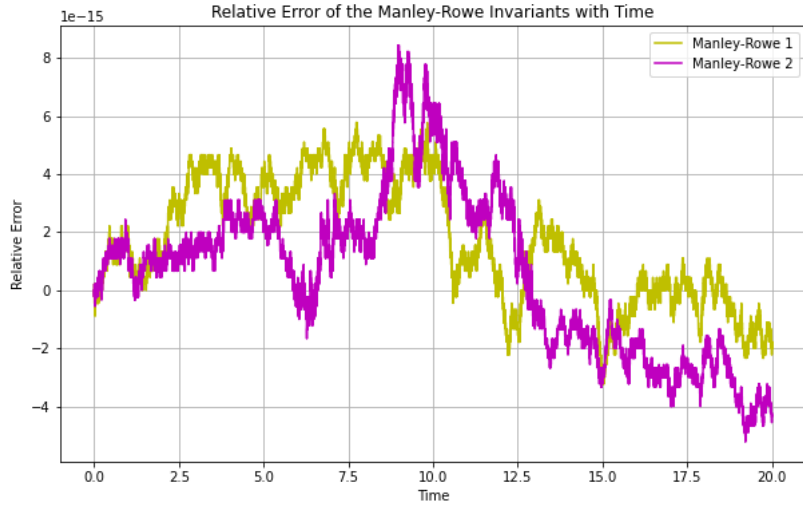


Figure 8: This shows the relative error between the initial Manley-Rowe Invariants and the Manley-Rowe Invariants at each timestep. As the error is  $\sim e - 15$ , the Manley-Rowe Invariants for the triad are invariant.

## 4.2 Invariants for Butterfly

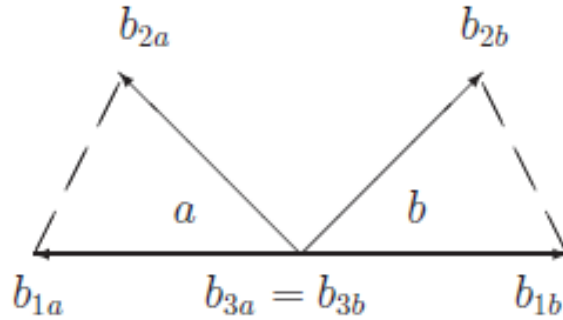


Figure 9: This shows 5 modes interconnected in a butterfly structure.

This butterfly cluster structure consists of 2 triads connected by an active mode which is the active mode in both triads, the passive modes in the first triad are (1,-8) and (15,10) and the active mode is (16,2) and the passive modes in the second triad are (3,-11) and (13,13).

The equations of motion for this triad are

$$\begin{aligned}
 \dot{a}_1 &= -130a_2^*a_3 \\
 \dot{a}_2 &= -78a_1^*a_3 \\
 \dot{a}_3 &= 130a_1a_2 + \frac{728}{5}a_4a_5 \\
 \dot{a}_4 &= -\frac{546}{5}a_5^*a_3 \\
 \dot{a}_5 &= -70a_4^*a_3
 \end{aligned} \tag{60}$$

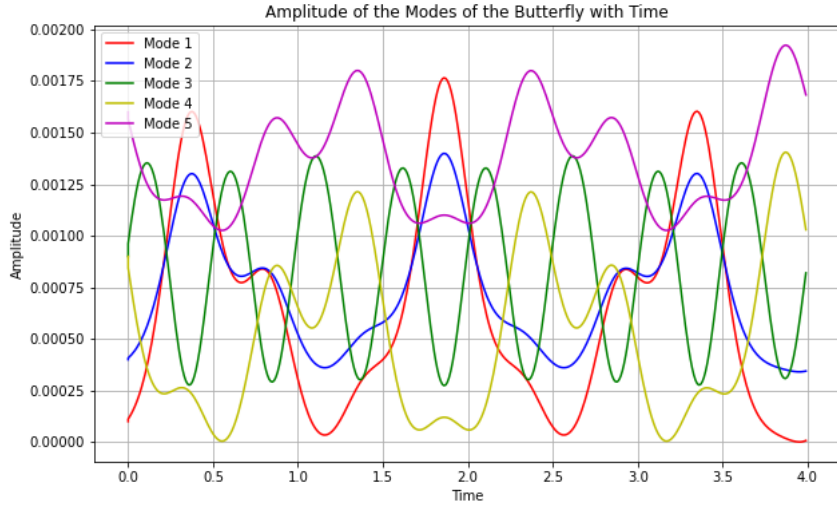


Figure 10: This shows the amplitudes of the five modes in the butterfly over time with a set of initial conditions.

Transforming the equations of motion to b-variables gives

$$\begin{aligned}
 \dot{b}_1 &= \frac{i\beta}{65}b_1 - \frac{2\beta^{1/2}\sqrt{15}}{5}b_2^*b_3 \\
 \dot{b}_2 &= \frac{3i\beta}{65}b_2 - \frac{2\beta^{1/2}\sqrt{15}}{5}b_1^*b_3 \\
 \dot{b}_3 &= \frac{4i\beta}{65}b_3 + \frac{2\beta^{1/2}\sqrt{15}}{5}b_1b_2 + \frac{14\beta^{1/2}\sqrt{39}}{65}b_4b_5 \\
 \dot{b}_4 &= \frac{3i\beta}{130}b_4 - \frac{14\beta^{1/2}\sqrt{39}}{65}b_5^*b_3 \\
 \dot{b}_5 &= \frac{i\beta}{26}b_5 - \frac{14\beta^{1/2}\sqrt{39}}{65}b_4^*b_3
 \end{aligned} \tag{61}$$

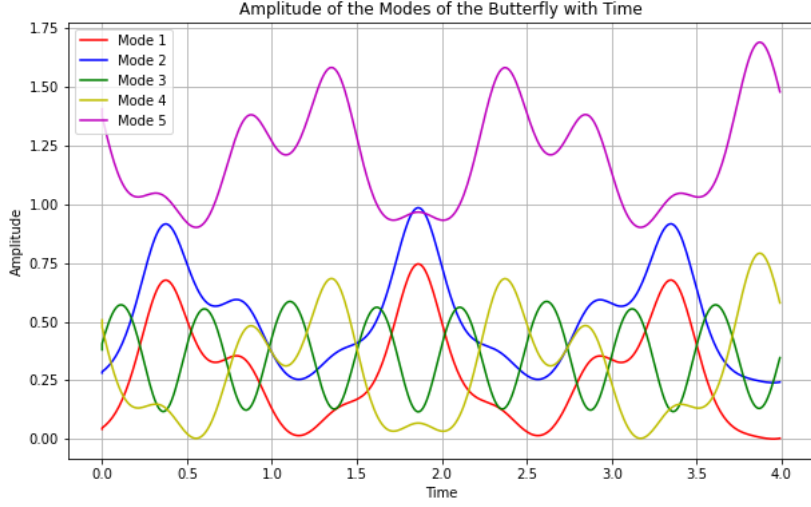


Figure 11: This shows the amplitudes of the five modes in the butterfly in b-variables over time with a set of initial conditions.

#### 4.2.1 Hamiltonian for Butterfly

$$\mathcal{H} = -\frac{\beta}{65}|b_1|^2 - \frac{3\beta}{65}|b_2|^2 - \frac{4\beta}{65}|b_3|^2 - \frac{3\beta}{130}|b_4|^2 - \frac{\beta}{26}|b_5|^2 - \frac{6i\beta^{1/2}\sqrt{15}}{5}(b_1^*b_2^*b_3 - b_1b_2b_3^*) - \frac{42i\beta^{1/2}\sqrt{39}}{65}(b_4^*b_5^*b_3 - b_4b_5b_3^*) \quad (62)$$

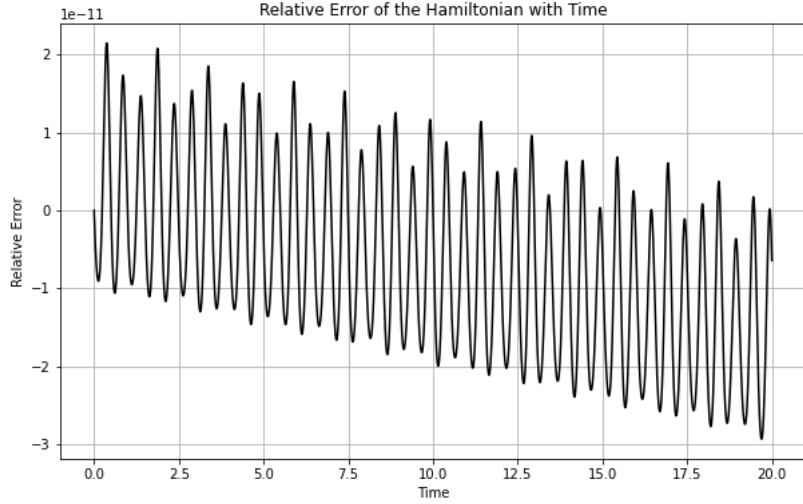


Figure 12: This shows the relative error between the initial Hamiltonian and the Hamiltonian at each timestep. As the error is  $\sim e - 11$ , the Hamiltonian for the butterfly is invariant.

#### 4.2.2 Energy and Enstrophy for butterfly

$$Energy = \frac{\beta}{65}|b_{\mathbf{k}_1}|^2 + \frac{3\beta}{65}|b_{\mathbf{k}_2}|^2 + \frac{4\beta}{65}|b_{\mathbf{k}_3}|^2 + \frac{3\beta}{130}|b_{\mathbf{k}_4}|^2 + \frac{\beta}{26}|b_{\mathbf{k}_5}|^2 \quad (63)$$

$$Enstrophy = \beta|b_{\mathbf{k}_1}|^2 + 15\beta|b_{\mathbf{k}_2}|^2 + 16\beta|b_{\mathbf{k}_3}|^2 + 3\beta|b_{\mathbf{k}_4}|^2 + 13\beta|b_{\mathbf{k}_5}|^2 \quad (64)$$

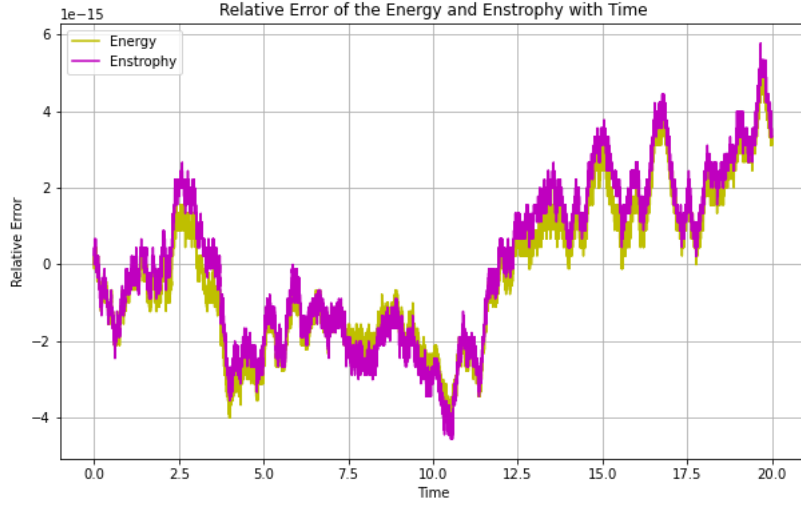


Figure 13: This shows the relative error between the initial Energy and Enstrophy and the Energy and Enstrophy at each timestep. As the error is  $\sim e - 15$ , the Energy and Enstrophy for the butterfly are invariant.

#### 4.2.3 Manley-Rowe Invariants for Butterfly

Cluster Matrix for the butterfly is  $\begin{bmatrix} 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 \end{bmatrix}$ .

So the nullspace is  $\begin{bmatrix} -1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ .

Meaning the Manley-Rowe Invariants are

$$\begin{aligned} MR_1 &= |b_2|^2 - |b_1|^2 \\ MR_2 &= |b_1|^2 + |b_3|^2 + |b_4|^2 \\ MR_3 &= |b_1|^2 + |b_3|^2 + |b_4|^2 \end{aligned} \quad (65)$$

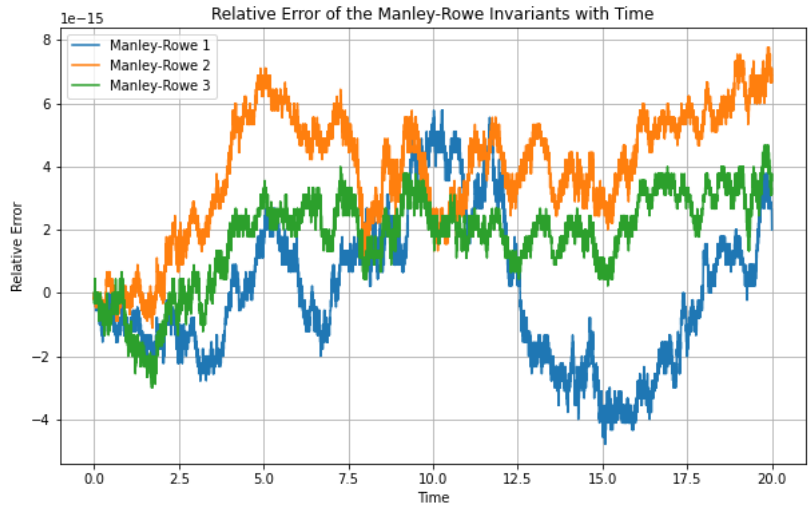


Figure 14: This shows the relative error between the initial Manley-Rowe Invariants and the Manley-Rowe Invariants at each timestep. As the error is  $\sim e - 15$ , the Manley-Rowe Invariants for the butterfly are invariant.

## 5 Dynamics of Cluster

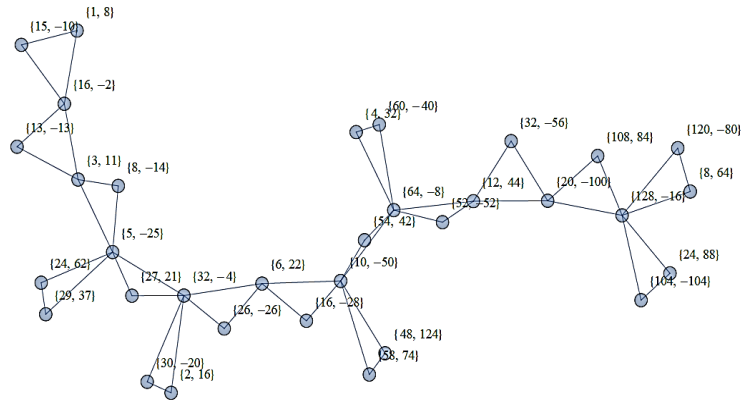


Figure 15: This figure shows a special cluster system that consists of modes which double every 10 modes.

In this project, a special type of cluster (above) is analysed which consists of ten modes which are doubled and then mapped and connected to the original ten modes to create a cluster of connecting triads. This means that the 11th mode is double the 1st, the 12th mode is double the 2nd and the 21st mode is double the 11th and so on. Thus the wavevectors are generated by  $\mathbf{k}_j = 2^j \mathbf{k}$  where  $j$  is the generation number (the number of times it has been mapped) [4]. This continues infinitely, but this project considers only to a maximum size of 81 modes.



For the purposes of the project the initial conditions of the modes are initialised by a certain method with quasi random initial phase and initial amplitudes of  $\sim |\mathbf{k}|^n$ .

## 5.1 Capacity

Invariants can have either finite or infinite capacity. This means that depending on the initial condition of the modes in the cluster, as the number of modes in the cluster tends to infinity, the invariants may converge to a certain value (finite capacity) or not (infinite capacity) [2].

In this project, the capacity of the Hamiltonian, energy and enstrophy are studied. In order to simplify the study of the Hamiltonian it is shown (below) that the first term in the Hamiltonian is equal to the negative of the energy of the cluster, so the  $\mathcal{H}_3$  must be an invariant. So  $\mathcal{H}_3$ , energy and enstrophy are the only invariants for which the capacity is studied.

$$\begin{aligned} \text{Energy} &= \sum_{\mathbf{k}} \frac{\beta k_x}{|\mathbf{k}|^2} |b_{\mathbf{k}}|^2 \\ \mathcal{H}_2 &= \sum_{\mathbf{k}} \omega_{\mathbf{k}} |b_{\mathbf{k}}|^2 \end{aligned} \quad (66)$$

and  $\omega_{\mathbf{k}} = \frac{-\beta k_x}{|\mathbf{k}|^2}$  since  $F = 0$ . So  $\mathcal{H}_2 = -\text{Energy} = \text{Constant}$

### 5.1.1 C-Variables

Before studying the capacity of the invariants, performing a transformation on the b-variables to c-variables removes the fast moving linear term from our model which optimises the time needed to run code.

The transformation is  $c_{\mathbf{k}} = b_{\mathbf{k}} e^{i\omega_{\mathbf{k}} t}$ . So,

$$\dot{c}_{\mathbf{k}} = \dot{b}_{\mathbf{k}} e^{i\omega_{\mathbf{k}} t} + i\omega_{\mathbf{k}} b_{\mathbf{k}} e^{i\omega_{\mathbf{k}} t} \quad (67)$$

and rearranging this gives

$$i\dot{b}_{\mathbf{k}} = i\dot{c}_{\mathbf{k}} e^{-i\omega_{\mathbf{k}} t} + \omega_{\mathbf{k}} b_{\mathbf{k}} \quad (68)$$

The Hamiltonian form of the CHM equation is

$$i\dot{b}_{\mathbf{k}} = \omega_{\mathbf{k}} b_{\mathbf{k}} + \sum_{\mathbf{k}_1 \mathbf{k}_2 \in \mathbb{Z}^2} b_{\mathbf{k}_1} b_{\mathbf{k}_2} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} V_{12k} \quad (69)$$

So using this substitution into our Hamiltonian form equations of motion gives

$$i\dot{c}_{\mathbf{k}} e^{-i\omega_{\mathbf{k}} t} = \sum_{\mathbf{k}_1 \mathbf{k}_2 \in \mathbb{Z}^2} c_{\mathbf{k}_1} e^{-i\omega_{\mathbf{k}_1} t} c_{\mathbf{k}_2} e^{-i\omega_{\mathbf{k}_2} t} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} V_{12k} \quad (70)$$

$$i\dot{c}_{\mathbf{k}} = \sum_{\mathbf{k}_1 \mathbf{k}_2 \in \mathbb{Z}^2} c_{\mathbf{k}_1} c_{\mathbf{k}_2} e^{-i\omega_{\mathbf{k}_1 \mathbf{k}_2}^{\mathbf{k}} t} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} V_{12k} \quad (71)$$

where  $\omega_{\mathbf{k}_1 \mathbf{k}_2}^{\mathbf{k}} = \omega_{\mathbf{k}_1} + \omega_{\mathbf{k}_2} - \omega_{\mathbf{k}}$ . So in resonant conditions  $\omega_{\mathbf{k}_1 \mathbf{k}_2}^{\mathbf{k}} = 0$ , giving

$$i\dot{c}_{\mathbf{k}} = \sum_{\mathbf{k}_1 \mathbf{k}_2 \in \mathbb{Z}^2} c_{\mathbf{k}_1} c_{\mathbf{k}_2} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} V_{12k} \quad (72)$$

The transformed invariants with C-Variables are as follows

$$\mathcal{H} = \sum_{\mathbf{k}} \omega_{\mathbf{k}} |c_{\mathbf{k}}|^2 + \sum_{\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3 \in \mathbb{Z}^2} (V_{123} c_{\mathbf{k}_1} c_{\mathbf{k}_2} c_{\mathbf{k}_3}^* \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} + V_{123}^* c_{\mathbf{k}_1}^* c_{\mathbf{k}_2}^* c_{\mathbf{k}_3} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}}) \quad (73)$$

$$\text{Energy} = \sum_i^M \frac{\beta k_{ix}}{|\mathbf{k}_i|^2} |c_{\mathbf{k}_i}|^2 \quad (74)$$

$$Enstrophy = \sum_i^M \beta k_{ix} |c_{\mathbf{k}_i}|^2 \quad (75)$$

and the Manley-Rowe invariants are unchanged with c instead of b as  $|e^{i\omega_{\mathbf{k}}t}|^2 = 1$ . The figure below shows the difference in amplitude of the modes between the b-variables and c-variables for the butterfly cluster, as the difference in amplitude is  $\sim e - 14$  there is little added error by performing this transformation.

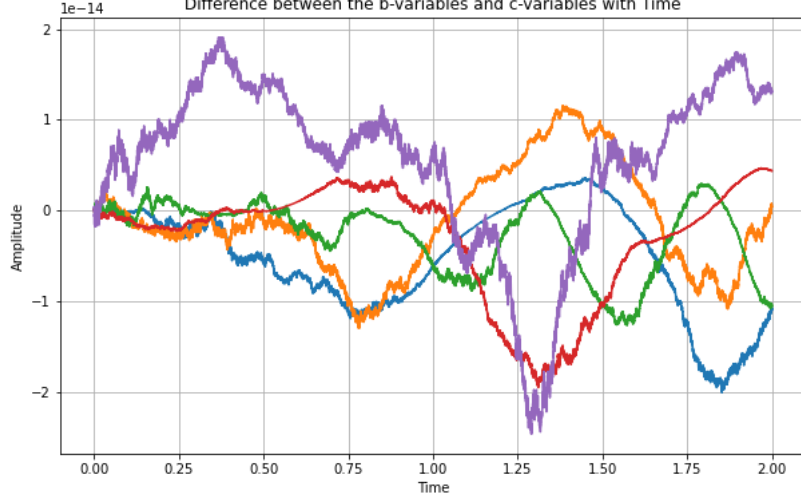


Figure 16: This shows the difference in amplitudes of the five modes in the butterfly in b-variables and c-variables over time with a set of initial conditions.

### 5.1.2 Capacity

The initial amplitude of the modes are  $|c_{\mathbf{k}}| \sim |\mathbf{k}|^n$ . As the number of modes in the cluster matrix tends to infinity, different values of n result in the capacity of the invariants being finite and infinite. In this section, the boundary value of n between finite and infinite capacity for each invariant is found.

#### Hamiltonian

Since the Hamiltonian is invariant and the first term is invariant, since it equals the negative energy, so the second term is invariant also.

$$\mathcal{H}_3 = \sum_{\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3 \in \mathbb{Z}^2} (V_{123} c_{\mathbf{k}_1} c_{\mathbf{k}_2} c_{\mathbf{k}_3}^* \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} + V_{123}^* c_{\mathbf{k}_1}^* c_{\mathbf{k}_2}^* c_{\mathbf{k}_3} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}}) \quad (76)$$

$$\mathcal{H}_3 = \sum_{\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3 \in Gen} \sum_j -\frac{i\beta^{1/2}}{2} (|k_x 2^j k_{1x} 2^j k_{2x} 2^j|)^{1/2} \left( \frac{k_{1y} 2^j}{|\mathbf{k}_1 2^j|^2} + \frac{k_{2y} 2^j}{|\mathbf{k}_2 2^j|^2} - \frac{k_y 2^j}{|\mathbf{k} 2^j|^2} \right) \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} (c_{\mathbf{k}_1} c_{\mathbf{k}_2} c_{\mathbf{k}_3}^* - c_{\mathbf{k}_1}^* c_{\mathbf{k}_2}^* c_{\mathbf{k}_3}) \quad (77)$$

$$\mathcal{H}_3 = \sum_{\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3 \in Gen} \sum_j -\frac{i\beta^{1/2}}{2} (|k_x k_{1x} k_{2x}|)^{1/2} \left( \frac{k_{1y}}{|\mathbf{k}_1|^2} + \frac{k_{2y}}{|\mathbf{k}_2|^2} - \frac{k_y}{|\mathbf{k}|^2} \right) 2^{j/2} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} (c_{\mathbf{k}_1} c_{\mathbf{k}_2} c_{\mathbf{k}_3}^* - c_{\mathbf{k}_1}^* c_{\mathbf{k}_2}^* c_{\mathbf{k}_3}) \quad (78)$$

Letting

$$\begin{aligned} c_{\mathbf{k}_1} &= c_{1a} + c_{1b}i \\ c_{\mathbf{k}_2} &= c_{2a} + c_{2b}i \\ c_{\mathbf{k}_3} &= c_{3a} + c_{3b}i \end{aligned} \quad (79)$$

$$\mathcal{H}_3 = \sum_{\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3 \in \text{Gen}} \sum_j V_{123} 2^{\frac{j}{2}} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} ((c_{1a} + c_{1b}i)(c_{2a} + c_{2b}i)(c_{3a} - c_{3b}i) - (c_{1a} - c_{1b}i)(c_{2a} - c_{2b}i)(c_{3a} + c_{3b}i)) \quad (80)$$

$$\begin{aligned} \mathcal{H}_3 = \sum_{\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3 \in \text{Gen}} \sum_j V_{123} 2^{\frac{j}{2}} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} & (c_{1a}c_{2a}c_{3a} - c_{1a}c_{2a}c_{3b}i + c_{1a}c_{2b}c_{3a}i + c_{1a}c_{2b}c_{3b} + c_{1b}c_{2a}c_{3a}i + c_{1b}c_{2a}c_{3b} \\ & - c_{1b}c_{2b}c_{3a} + c_{1b}c_{2b}c_{3b}i - c_{1a}c_{2a}c_{3a} - c_{1a}c_{2a}c_{3b}i + c_{1a}c_{2b}c_{3a}i - c_{1a}c_{2b}c_{3b} + c_{1b}c_{2a}c_{3a}i - c_{1b}c_{2a}c_{3b} + c_{1b}c_{2b}c_{3a} \\ & + c_{1b}c_{2b}c_{3b}i) \end{aligned} \quad (81)$$

$$\mathcal{H}_3 = \sum_{\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3 \in \text{Gen}} \sum_j 2V_{123} 2^{\frac{j}{2}} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} (-c_{1a}c_{2a}c_{3b}i + c_{1a}c_{2b}c_{3a}i + c_{1b}c_{2a}c_{3a}i + c_{1b}c_{2b}c_{3b}i) \quad (82)$$

Since  $|c_{\mathbf{k}}| \sim |\mathbf{k}|^n$

$$\begin{aligned} |c_{1a}| &= |2^j \mathbf{k}|^n |\cos(\theta_1)| \\ |c_{1b}| &= |2^j \mathbf{k}|^n |\sin(\theta_1)| \end{aligned} \quad (83)$$

$$\begin{aligned} |\mathcal{H}_3| &= \sum_{\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3 \in \text{Gen}} \sum_j i2V_{123} 2^{\frac{j}{2}} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} (|2^j \mathbf{k}_1|^n \cos(\theta_1) |2^j \mathbf{k}_2|^n \cos(\theta_2) |2^j \mathbf{k}_3|^n \sin(\theta_3) \\ &+ |2^j \mathbf{k}_1|^n \cos(\theta_1) |2^j \mathbf{k}_2|^n \sin(\theta_2) |2^j \mathbf{k}_3|^n \cos(\theta_3) \\ &+ |2^j \mathbf{k}_1|^n \sin(\theta_1) |2^j \mathbf{k}_2|^n \cos(\theta_2) |2^j \mathbf{k}_3|^n \cos(\theta_3) \\ &+ |2^j \mathbf{k}_1|^n \sin(\theta_1) |2^j \mathbf{k}_2|^n \sin(\theta_2) |2^j \mathbf{k}_3|^n \sin(\theta_3)) \end{aligned} \quad (84)$$

$$\begin{aligned} |\mathcal{H}_3| &= \sum_{\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3 \in \text{Gen}} \sum_j i2V_{123} |\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3|^n 2^{\frac{j}{2} + 3jn} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} (|\cos(\theta_1) \cos(\theta_2) \sin(\theta_3) \\ &+ \cos(\theta_1) \sin(\theta_2) \cos(\theta_3) + \sin(\theta_1) \cos(\theta_2) \cos(\theta_3) + \sin(\theta_1) \sin(\theta_2) \sin(\theta_3)|) \end{aligned} \quad (85)$$

$$|\mathcal{H}_3| = \sum_{\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3 \in \text{Gen}} \sum_j i2V_{123} |\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3|^n 2^{\frac{j}{2} + 3jn} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} (|\cos(\theta_1) \sin(\theta_2 - \theta_3) + \sin(\theta_1) \cos(\theta_2 - \theta_3)|) \quad (86)$$

$$|\mathcal{H}_3| = \sum_{\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3 \in \text{Gen}} \sum_j i2V_{123} |\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3|^n 2^{\frac{j}{2} + 3jn} \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} |\sin(\theta_1 + \theta_2 - \theta_3)| \quad (87)$$

$$|\mathcal{H}_3| = \sum_{\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3 \in \text{Gen}} i2V_{123} |\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3|^n \delta_{\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}} |\sin(\theta_1 + \theta_2 - \theta_3)| \sum_j 2^{\frac{j}{2} + 3jn} \quad (88)$$

$|\mathcal{H}_3|$  will converge if  $2^{\frac{1}{2} + 3n} < 1$ , as  $\sum_j 2^{\frac{j}{2} + 3jn}$  converges to  $\frac{1}{1 - 2^{j(\frac{1}{2} + 3n)}}$  if this condition is satisfied.

So for the Hamiltonian to have finite capacity  $2^{\frac{1}{2} + 3n} < 1$  so  $n < -\frac{1}{6}$ .

## Energy

The energy is also invariant, with  $|c_{\mathbf{k}}| \sim |\mathbf{k}|^n$  and the energy as below.

$$\text{Energy} = \sum_i^M \frac{\beta k_{ix}}{|\mathbf{k}_i|^2} |c_{\mathbf{k}_i}|^2 \quad (89)$$

$$\text{Energy} = \sum_{\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3 \in \text{Gen}} \sum_j \frac{\beta k_x 2^j}{|\mathbf{k} 2^j|^2} |(\mathbf{k} 2^j)^n|^2 \quad (90)$$

$$\text{Energy} = \sum_{\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3 \in \text{Gen}} \frac{\beta k_x}{|\mathbf{k}|^2} |\mathbf{k}|^{2n} \sum_j 2^{j-2j+2jn} \quad (91)$$

Energy will converge if  $2^{2n-1} < 1$ , as  $\sum_j 2^{2jn-j}$  converges to  $\frac{1}{1 - 2^{j(2n-1)}}$  if this condition is satisfied.

So for the Energy to have finite capacity  $2^{2n-1} < 1$  so  $n < \frac{1}{2}$ .

## Enstrophy

Enstrophy is the third invariant that the capacity will be studied, with  $|c_{\mathbf{k}}| \sim |\mathbf{k}|^n$  and the enstrophy as below.

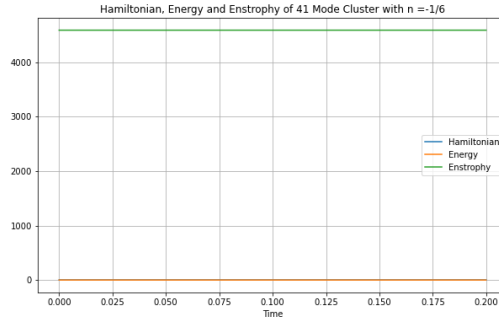
$$Enstrophy = \sum_i^M \beta k_{ix} |c_{\mathbf{k}_i}|^2 \quad (92)$$

$$Enstrophy = \sum_{\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3 \in Gen} \sum_j \beta k_x 2^j |(\mathbf{k} 2^j)^n|^2 \quad (93)$$

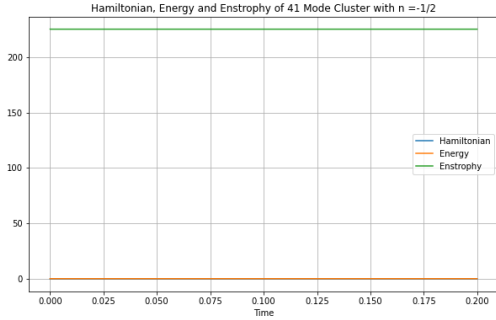
$$Enstrophy = \sum_{\mathbf{k}_1 \mathbf{k}_2 \mathbf{k}_3 \in Gen} \beta k_x |\mathbf{k}|^{2n} \sum_j 2^{j+2n j} \quad (94)$$

Enstrophy will converge if  $2^{2n+1} < 1$ , as  $\sum_j 2^{2jn+j}$  converges to  $\frac{1}{1-2^{j(2n+1)}}$  if this condition is satisfied. So for the Enstrophy to have finite capacity  $2^{2n+1} < 1$  so  $n < -\frac{1}{2}$ .

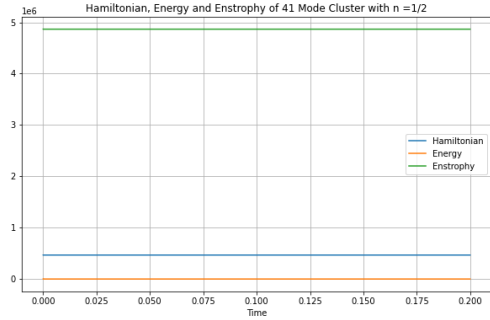
The significant values of  $n$  are therefore  $-1/6$ ,  $-1/2$  and  $1/2$ . The three invariants are plotted below with these values.



(a)  $n=-1/6$



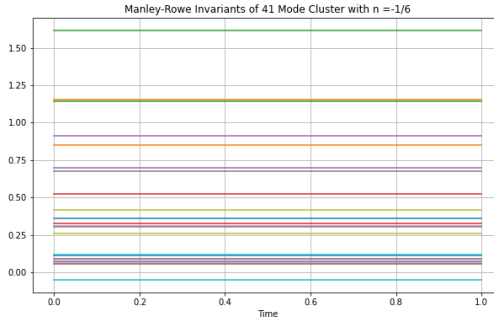
(b)  $n=-1/2$



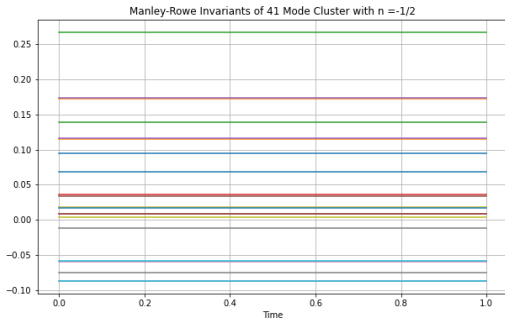
(c)  $n=1/2$

Figure 17: This figures shows the Hamiltonian, Energy and Enstrophy of the 41 mode cluster with (a)  $n=-1/6$ , (b)  $n=-1/2$  and (c)  $n=1/2$ .

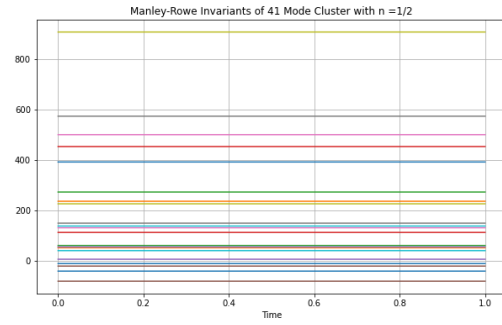
Even though, the capacity of the Manley-Rowe invariants were not explored, these invariants are plotted for the significant values of  $n$ .



(a)  $n = -1/6$



(b)  $n = -1/2$



(c)  $n = 1/2$

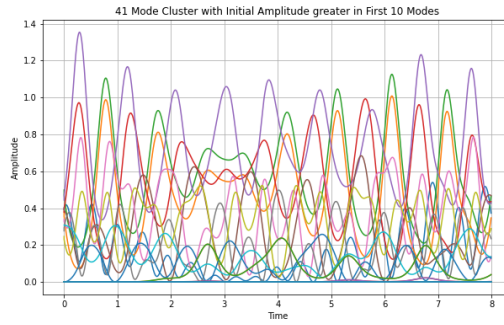
Figure 18: This figures shows the Manley-Rowe Invariants of the 41 mode cluster with (a)  $n = -1/6$ , (b)  $n = -1/2$  and (c)  $n = 1/2$ .

## 5.2 Amplitudes of Modes

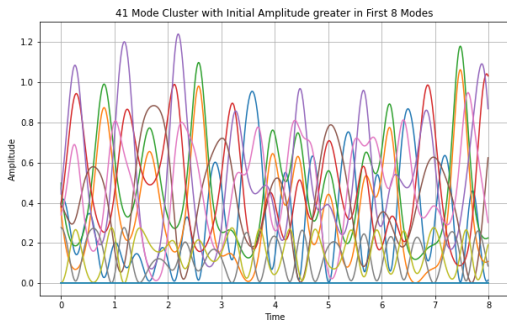
In this section, the effect of putting greater initial amplitude into a certain set of modes is studied. The initial amplitudes of the modes in the cluster were still by the rule  $|c_{\mathbf{k}}| \sim |\mathbf{k}|^n$  where the greater amplitude is  $|c_{\mathbf{k}}| = |\mathbf{k}|^n$  and the rest of the amplitudes of the modes are initialised by  $|c_{\mathbf{k}}| = \frac{|\mathbf{k}|^n}{100}$ . The initial phase of the modes in the cluster is pseudo-random.

### 5.2.1 First Modes

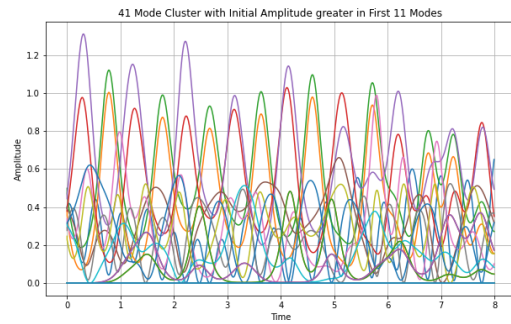
Firstly, greater initial amplitude was allocated to the first 8, 10, and 11 modes in a 41 mode cluster. The results showed that due on the structure of the cluster initially low amplitude modes may gain amplitude if connected to higher amplitude modes.



(a) First 10 Modes



(b) First 8 Modes

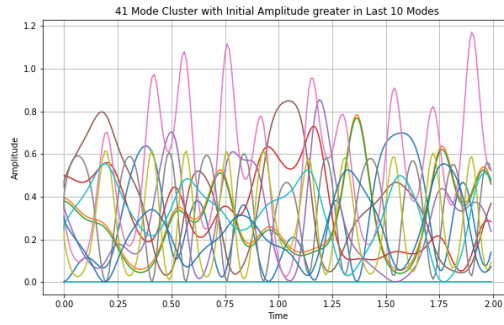


(c) First 11 Modes

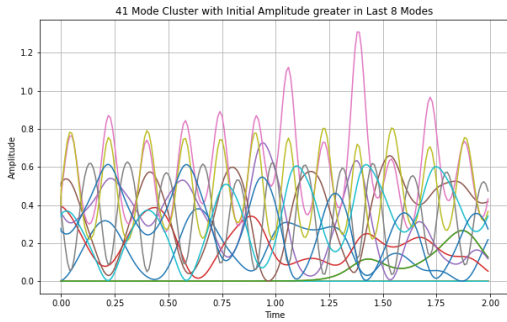
Figure 19: This figure shows a 41 mode cluster with  $n = -1/6$ , with greater initial amplitudes in first (a) 10 modes, (b) 8 modes and (c) 11 modes. In (a) 3 modes that do not have initial amplitude gain amplitude. In (b) 1 mode that does not have initial amplitude gains amplitude. In (c) 2 modes that do not have initial amplitude gain amplitude. This is due to the structure of the cluster. The 10th mode is passive and connected to the 11th mode which is active, and this is the active mode in to two other triads, which produces the extra modes that gain amplitude in (a) and (c). The 8th mode is active and connected to 7th and 9th modes which are passive and hence the 9th mode gains amplitude in (b).

### 5.2.2 Last Modes

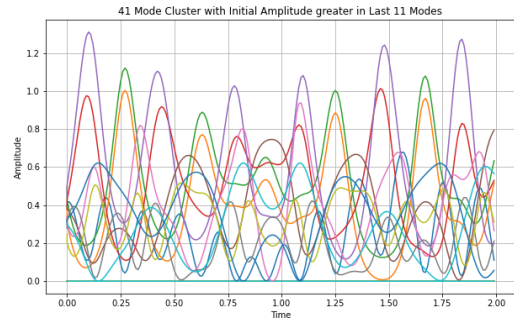
Next, greater initial amplitude was allocated to the last 8, 10, and 11 modes in a 41 mode cluster. The results showed that due on the structure of the cluster initially low amplitude modes may gain amplitude if connected to higher amplitude modes.



(a) Last 10 Modes



(b) Last 8 Modes



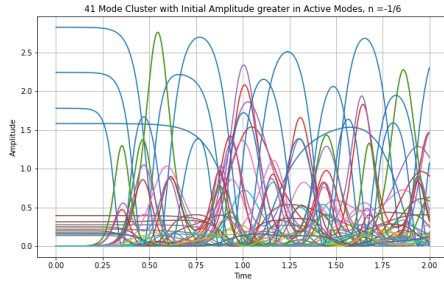
(c) Last 11 Modes

Figure 20: This figure shows a 41 mode cluster with  $n = -1/6$ , with greater initial amplitudes in last (a) 10 modes, (b) 8 modes and (c) 11 modes. In (a) 1 mode that does not have initial amplitude gains amplitude. In (b) 2 modes that do not have initial amplitude gain amplitude. In (c) no mode that does not have initial amplitude gains amplitude. This is due to the structure of the cluster. The 32nd mode is passive and connected to the 31st mode which is active which is the extra mode that gains amplitude in (a). The 31st mode is also active in the triad connected to 34th and 35th modes and hence the 31st mode gains amplitude in (b), also since this mode is active in other triads it is responsible for further modes gaining energy. In (c) as the 31st mode is active and connected to triads with initially greater amplitudes, no further modes gain amplitude.

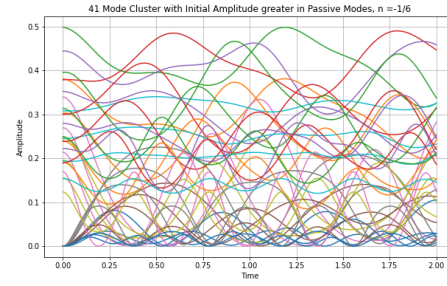
### 5.2.3 Active/Passive Modes

Then, greater initial amplitude was allocated to the active modes in the triad. The results showed that due on the structure of the cluster with active modes connected to generally more other modes than the passive modes, the passive modes with initially low amplitude gain amplitude but there are significantly more modes with lower amplitude than higher amplitude as time progresses.

Further, when greater initial amplitude was allocated to the passive modes in the triad, the results showed that the active modes with initially low amplitude gain amplitude but this time the amplitude is more spread out with the initially higher amplitude passive modes tending to remain at high amplitude. This suggests the passive modes do not transfer their amplitude to the active modes to the same extent as the active modes transfer to passive modes.



(a) Greater Amplitude in Active Modes

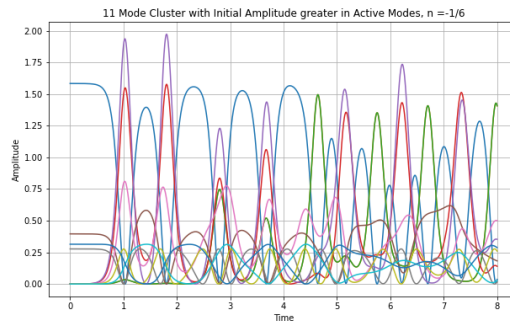


(b) Greater Amplitude in Passive Modes

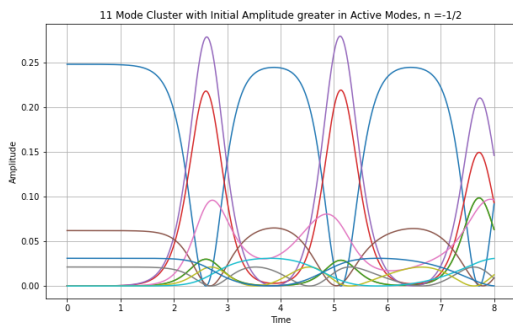
Figure 21: This figure shows a 41 mode cluster with  $n = -1/6$ , with greater initial amplitudes in (a) active modes and (b) passive modes. In both cases all modes gain amplitude with time.

### 5.2.4 11 Modes - Active

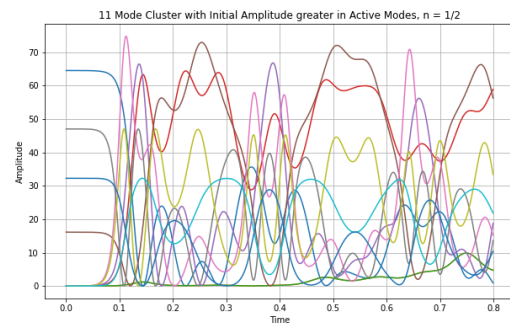
To explore the situation where greater initial amplitude was allocated to the active modes in the triad further, this figure shows the equations of motion of an 11 mode cluster for the significant values of  $n$ . There does not appear to be any pattern in behaviour.



(a)  $n = -1/6$



(b)  $n = -1/2$



(c)  $n = 1/2$

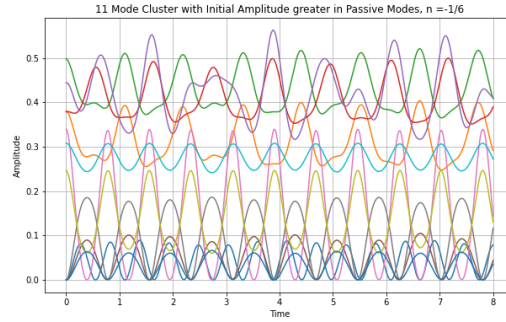
Figure 22: This figure shows an 11 mode cluster with greater initial amplitudes in the active modes with (a)  $n = -1/6$ , (b)  $n = -1/2$  and (c)  $n = 1/2$ . There does not appear to be any form of pattern in the modes.

### 5.2.5 11 Modes - Passive

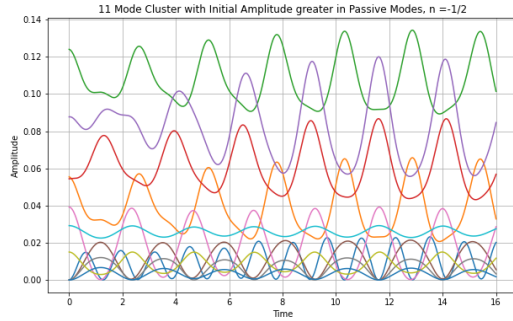
Again to explore the situation where greater initial amplitude was allocated to the passive modes in the triad further, this figure shows the equations of motion of an 11 mode cluster for the significant values of



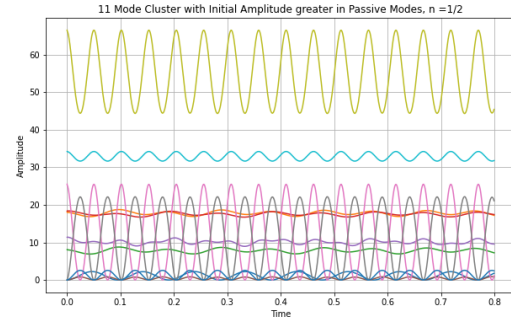
n. There appears to be a periodic behaviour pattern with the modes which were initially higher amplitude staying that way and initially lower amplitude staying lower.



(a)  $n=-1/6$



(b)  $n=-1/2$



(c)  $n=1/2$

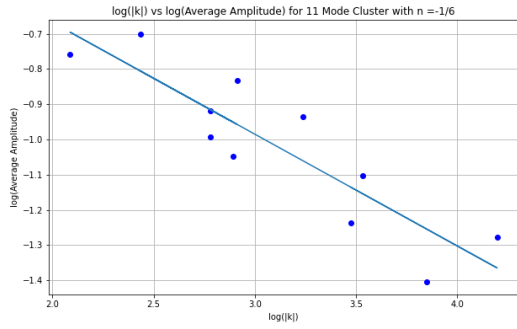
Figure 23: This figure shows an 11 mode cluster with greater initial amplitudes in the passive modes with (a)  $n=-1/6$ , (b)  $n=-1/2$  and (c)  $n=1/2$ . The modes appear to exhibit a periodic pattern.

### 5.3 Average Amplitudes

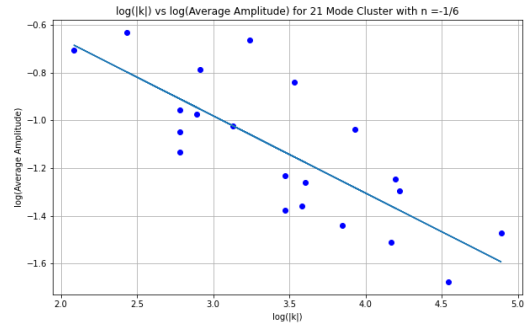
As the initial amplitude of the modes in the cluster followed the rule  $|c_{\mathbf{k}}| \sim |\mathbf{k}|^n$ , it is of interest to check if this value of  $n$  controlled the amplitude of these modes with time, i.e. the average amplitude of the modes. To discover whether this relation existed, the average amplitude of each mode was plotted against square of the magnitude of wavevector of that mode, and the best fit line was plotted between each data point. The slope of this best fit line is the value of interest which we call  $\gamma$  and this is a function of  $n$ . This can be seen as the average of the amplitudes  $\langle |c_{\mathbf{k}}| \rangle = \alpha |\mathbf{k}|^\gamma$  where  $\alpha$  is a constant, so  $\log(\langle |c_{\mathbf{k}}| \rangle) = \log(\alpha) + \gamma \log(|\mathbf{k}|)$ . We will study this relation in this section.

#### 5.3.1 Varying Cluster Size

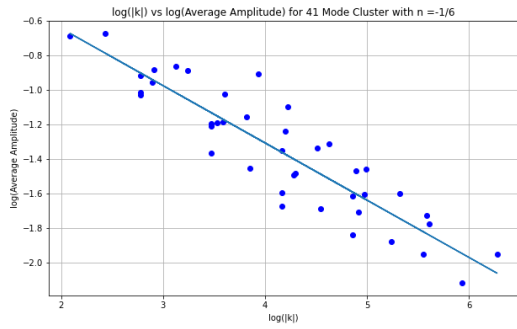
Firstly, the value of  $\gamma$  was studied for different number of modes in the cluster with the same  $n$  value of  $-1/6$ . The value of  $\gamma$  was roughly  $-1/3$  which suggests that  $\gamma = 2n$ . As the data points appeared more strongly correlated as the number of modes increased, a larger cluster would produce more accurate results.



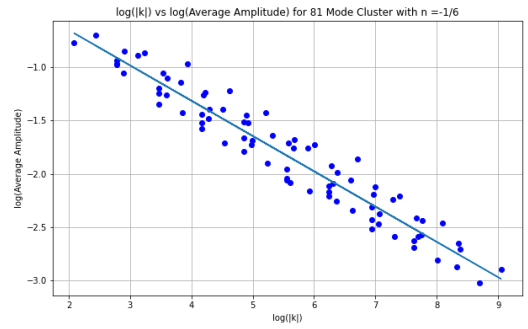
(a) 11 Modes



(b) 21 Modes



(c) 41 Modes

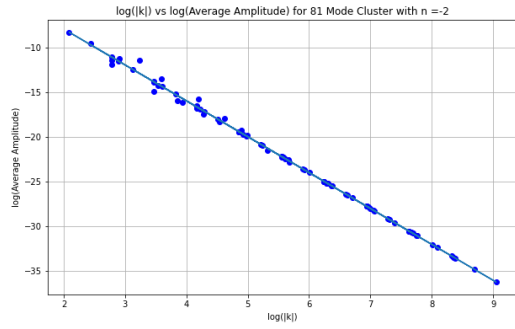


(d) 81 Modes

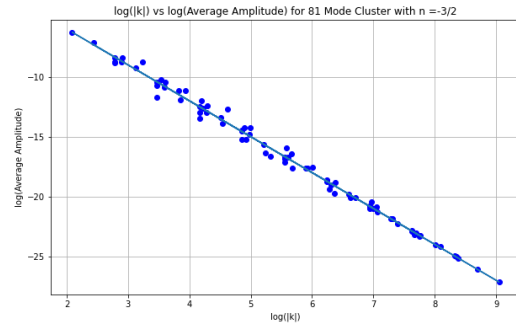
Figure 24: This figure shows the plot of the log of  $|k|$  of the mode vs the log of the average amplitude of the mode, for clusters with (a) 11 modes, (b) 21 modes, (c) 41 modes and (d) 81 modes, when  $n = -1/6$ . The slope of the best fit line for each plot is (a)  $-0.31722918897046$ , (b)  $-0.32436227157450176$ , (c)  $-0.3319914413791856$  and (d)  $-0.3310140369558723$ . This suggests that there is a pattern between initial power of the amplitude ( $n$ ) and the power of the average amplitude of the modes ( $\gamma$ ), of  $\gamma \sim 2n$ . As the cluster size increases the modes become closer toward the best fit line. Timestep = 0.001.

### 5.3.2 Varying $n$ value

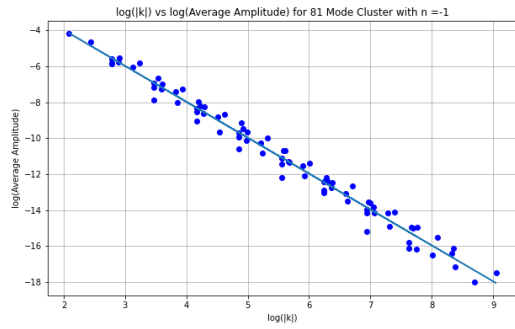
Next, the value of  $n$  was varied for an 81 mode cluster to ensure the hypothesis that  $\gamma = 2n$ . An 81 mode cluster was chosen as a larger cluster appeared to have strongly correlated data points. It is seen that  $\gamma \sim 2n$  for all values of  $n$  chosen, and the lower values of  $n$  had more highly correlated data points.



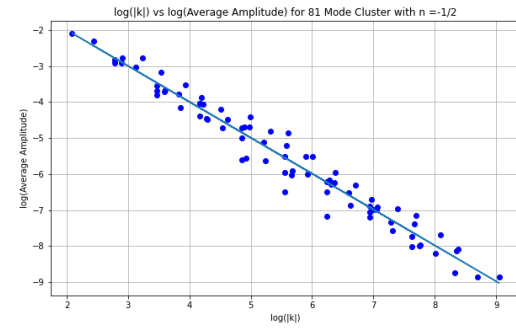
(a)  $n = -2$



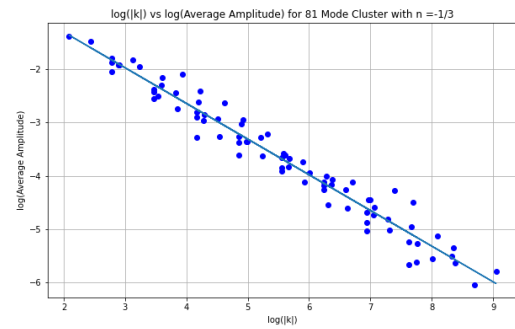
(b)  $n = -3/2$



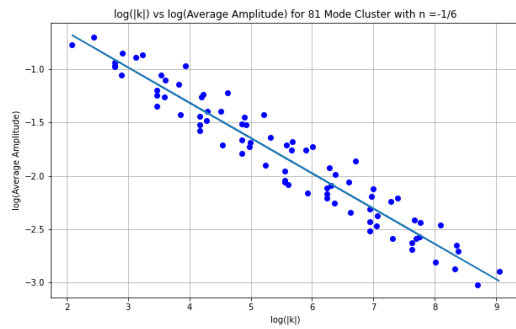
(c)  $n = -1$



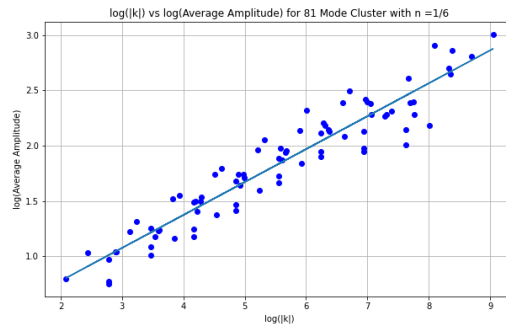
(d)  $n = -1/2$



(e)  $n = -1/3$



(f)  $n = -1/6$

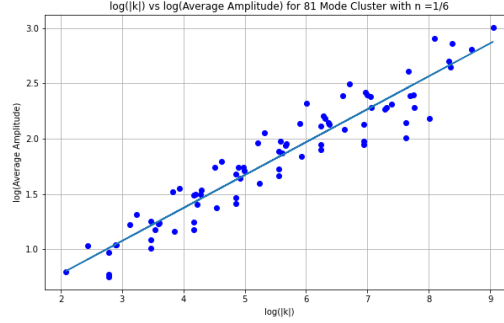


(g)  $n = 1/6$

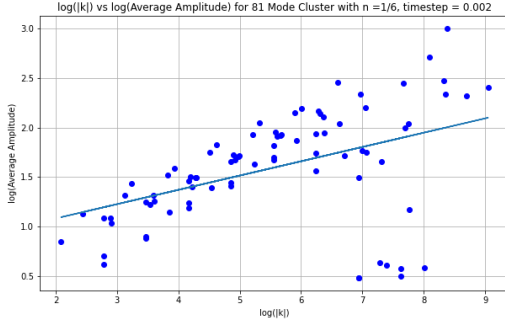
Figure 25: This figure shows the plot of the log of  $|k|^2$  of the mode vs the log of the average amplitude of the mode, for an 81 mode cluster with (a)  $n = -2$ , (b)  $n = -3/2$ , (c)  $n = -1$ , (d)  $n = -1/2$ , (e)  $n = -1/3$ , (f)  $n = -1/6$  and (g)  $n = 1/6$ . The slope of the best fit line for each plot is (a)  $-4.004955558554817$ , (b)  $-3.007288389233271$ , (c)  $-1.9960800115259507$ , (d)  $-0.996640957515859$ , (e)  $-0.6683312636406519$ , (f)  $-0.3310140369558723$  and (g)  $0.2973421728171482$ . This suggests that there is a pattern between initial power of the amplitude ( $n$ ) and the power of the average amplitude of the modes ( $\gamma$ ), of  $\gamma \sim 2n$ . As  $n$  decreases the modes become closer toward the best fit line. Timestep = 0.001.

### 5.3.3 Improvements in timestep

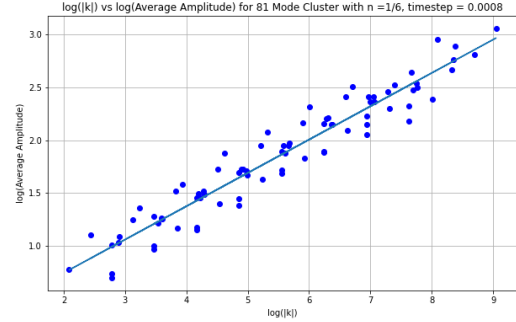
The timestep for all previous plots in this section is 0.001, but it is important to notice if reducing the timestep would improve the estimation of  $\gamma$ . In the plots below the original timestep of 0.001 is compared to the reduced timestep of 0.0008 and increased timestep of 0.002, for a fixed  $n = -1/6$ . It can be seen that reducing the timestep produces a  $\gamma$  value closer to  $2n$ .



(a) Timestep = 0.001



(b) Timestep = 0.002



(c) Timestep = 0.0008

Figure 26: This figure shows the difference in timestep on the slope of the log of  $|k|^2$  of the mode vs the log of the average amplitude of the mode with a timestep of (a) 0.001, (b) 0.002 and (c) 0.0008, with  $n = 1/6$ . The slope of the best fit line for each plot is (a) 0.2973421728171482, (b) 0.14426698366954577, and (c) 0.3144631427642188. This shows that as timestep decreases the slope increases in accuracy to  $1/3$ , which is  $2n$ .

## 5.4 Phase Analysis

The final aspect of the cluster dynamics that was explored in this project is phase analysis of the cluster. This involved examining the order parameter  $R(t)$  and the triad phase, which have been described before by Murray and Bustamante [5].

$$\begin{aligned} \phi_{1,2}^3(t) &= \left| \sum^N \text{arg}(c_1 c_2 c_3^*) \right| \\ R(t) &= \frac{\left| \sum^N \exp(i\phi_{1,2}^3) \right|}{N} \end{aligned} \quad (95)$$

where  $N$  is the number of triads in the cluster.

Both  $R(t)$  and  $\phi_{1,2}^3(t)$  are plotted on the one graph against time. The maximum  $R(t)$  and average  $R(t)$  for each plot are calculated. The graph below shows that when the modes are initialised with zero phase that the modes do not change phase and values of the  $\phi_{1,2}^3(t)$  are either 0 or  $\pi$  as expected.

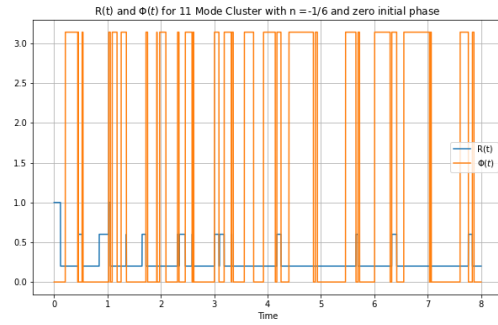
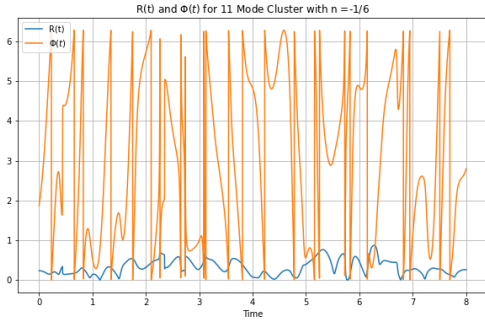


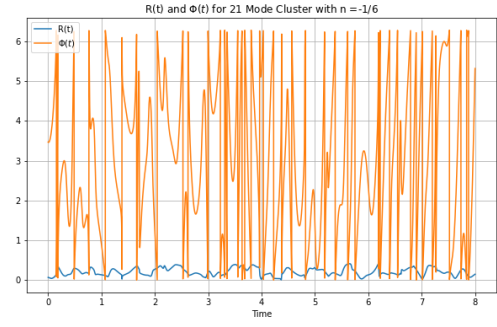
Figure 27: This figure shows the plot of the order parameter  $R(t)$  of the cluster and the triad phases  $\Phi(t)$  with time for an 11 mode cluster with initial zero phase of the modes. As the  $R(t)$  value is either 0 or 1 and the triad phase is 0 or  $\pi$ , this shows the code works.

#### 5.4.1 Varying Cluster Size

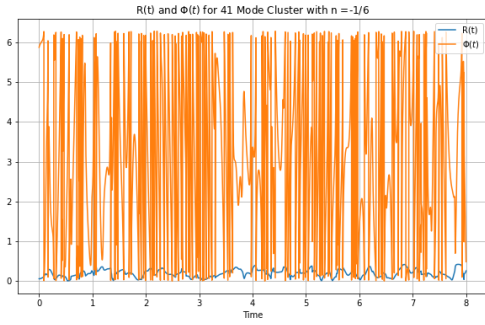
Firstly, the order parameter and triad phases were studied for different number of modes in the cluster with the same  $n$  value of  $-1/6$ . The results show that as the cluster size increases, there are more fluctuations in the plot of both  $R(t)$  and  $\Phi(t)$  with time. Also as the cluster size increases the  $R(t)$  reaches lower peak values. As the cluster size increases both the maximum  $R(t)$  and average  $R(t)$  appear to decrease.



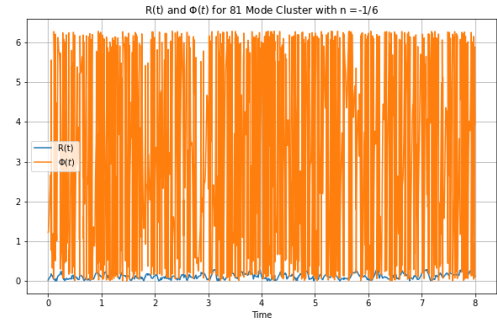
(a) 11 Modes



(b) 21 Modes



(c) 41 Modes

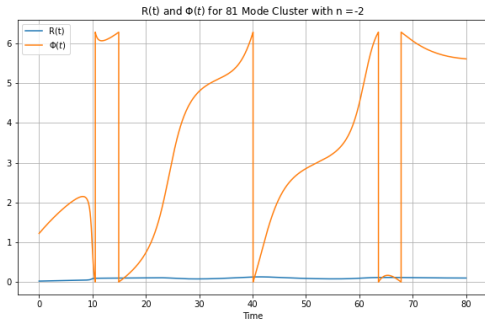


(d) 81 Modes

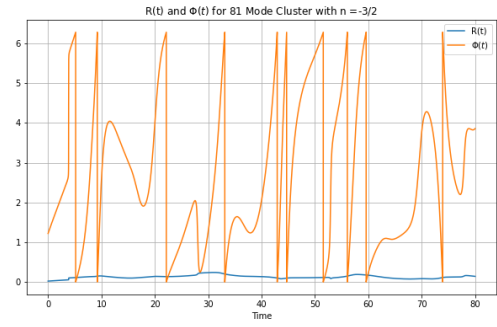
Figure 28: This figure shows the plot of the order parameter  $R(t)$  of the cluster and the triad phases  $\Phi(t)$  with time, for clusters with (a) 11 modes, (b) 21 modes, (c) 41 modes and (d) 81 modes, when  $n = -1/6$ . The maximum value for the  $R(t)$  is (a) 0.8725280260985734, (b) 0.4095003787981561, (c) 0.4195933663126821 and (d) 0.3648872765716297. The average value for the  $R(t)$  is (a) 0.32034835597280037, (b) 0.20336334993482372, (c) 0.19247125621936306 and (d) 0.12317304837143686. As the cluster size increases both the maximum  $R(t)$  and average  $R(t)$  appear to decrease. Timestep = 0.001.

#### 5.4.2 Varying $n$ value

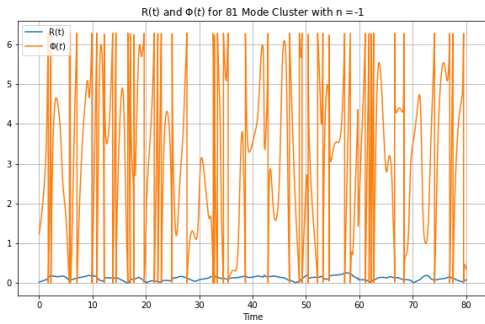
Next, the order parameter and triad phases were studied for different  $n$  value for an 81 mode cluster. The time period differs as to clearly present the graphs. The results show that as the  $n$  value increases, there are more fluctuations in the plot of both  $R(t)$  and  $\Phi(t)$  with time. Also as the  $n$  value increases the  $R(t)$  reaches higher peak values. As  $n$  increases the maximum  $R(t)$  tends to increase and the average  $R(t)$  does not seem to vary in any pattern.



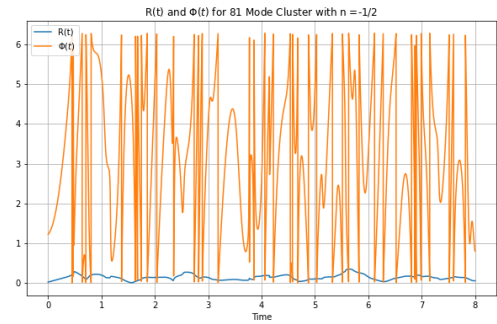
(a)  $n = -2$



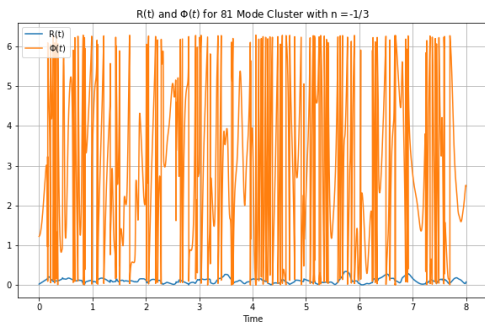
(b)  $n = -3/2$



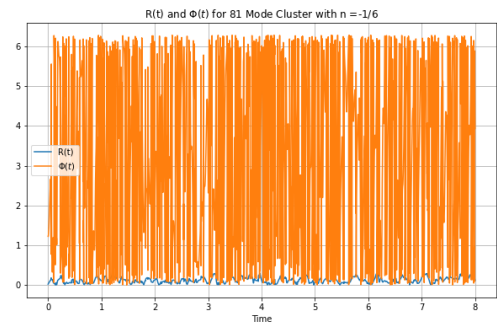
(c)  $n = -1$



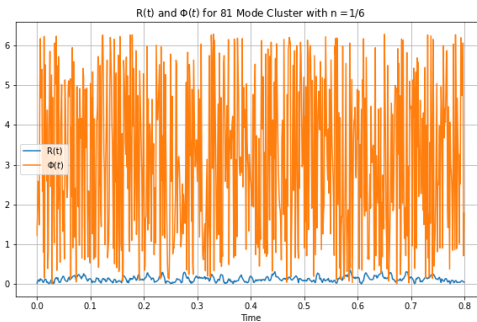
(d)  $n = -1/2$



(e)  $n = -1/3$



(f)  $n = -1/6$



(g)  $n = 1/6$

Figure 29: This figure shows the plot of the order parameter  $R(t)$  of the cluster and the triad phases  $\Phi(t)$  with time, for an 81 mode cluster with (a)  $n = -2$ , (b)  $n = -3/2$ , (c)  $n = -1$ , (d)  $n = -1/2$ , (e)  $n = -1/3$ , (f)  $n = -1/6$  and (g)  $n = 1/6$ . The maximum value for the  $R(t)$  is  $s_3(a)$  0.1291981241390307, (b) 0.2410493694476629, (c) 0.25705191267401906, (d) 0.35038498587296363, (e) 0.3412563595999774, (f) 0.3648872765716297 and (g) 0.35079013904851447. The average value for the  $R(t)$  is (a) 0.092616558062236, (b) 0.1279160598483235, (c) 0.11765192747332175, (d) 0.1411575269765623, (e) 0.10878277194764227, (f) 0.12317304837143686 and (g) 0.12988717254285947. As  $n$  increases the maximum  $R(t)$  tends to increase and the average  $R(t)$  does not seem to vary in any pattern. Timestep = 0.001. Note: (a),(b),(c) have a time period of 80, (d),(e),(f) have a time period of 8 and (g) has a time period of 0.8.

## 6 Conclusion

This project explored the dynamics of clusters of waves in the CHM equation by examining the invariants, capacity, amplitude and phase of modes. It was seen that varying initial conditions, by  $n$  value and cluster size affect the dynamics of the cluster. Further study could include larger clusters and a greater variety of  $n$  values, also the code included (Appendix) could be used to study different types of clusters of waves.

## References

- [1] Shane G Walsh and Miguel D Bustamante. “On the convergence of the normal form transformation in discrete Rossby and drift wave turbulence”. In: *Journal of Fluid Mechanics* 884 (2020).
- [2] Sergey Nazarenko. *Wave turbulence*. Vol. 825. Springer Science & Business Media, 2011.
- [3] Katie L Harper, Miguel D Bustamante, and Sergey V Nazarenko. “Quadratic invariants for discrete clusters of weakly interacting waves”. In: *Journal of Physics A: Mathematical and Theoretical* 46.24 (2013), p. 245501.
- [4] Umar Hayat et al. “Discrete resonant Rossby/drift wave triads: Explicit parameterisations and a fast direct numerical search algorithm”. In: *Communications in Nonlinear Science and Numerical Simulation* 79 (2019), p. 104896.
- [5] Brendan P Murray and Miguel D Bustamante. “Energy flux enhancement, intermittency and turbulence via Fourier triad phase dynamics in the 1-D Burgers equation”. In: *Journal of Fluid Mechanics* 850 (2018), pp. 624–645.

## Appendix

All python code in this section can be modified to study various clusters with different initial conditions.

### Equations of Motion

Python code used to plot equations of motion for cluster with an example of initialising conditions for 41 mode cluster

```
import numpy as np
import matplotlib.pyplot as plt
import cmath as cm
from numpy import linalg as la
from scipy.stats import linregress
import sympy as sy

def V(kx, k1x, k2x, ky, k1y, k2y):
    return -np.sqrt(beta)*np.sqrt(abs(kx*k1x*k2x))*(k1y/(k1x**2+k1y**2)
+k2y/(k2x**2+k2y**2)-ky/(kx**2+ky**2))

def getplot5(matrix1, matrix2, dt, t0, initlist, imax, plot = False, printarray=False):
    fck = sy.zeros(len(matrix2[0]), 1, dtype=np.complex)
    cmatrix = []
    i = 0
    while i<len(matrix2[0]):
        cmatrix.append(sy.symbols('c'+str(i+1)))
        i+=1
    j=0
    while j<len(matrix1[0]):
        i = 0
        while i<len(matrix1[:,0]):
```



```

    if matrix1[i][j] == 1:
        k = 0
        while k < len(matrix1[0]):
            if k == j:
                k += 1
            else:
                if matrix1[i][k] == 1:
                    k1 = -matrix2[:,k]
                    ck1 = sy.conjugate(cmatrix[k])
                if matrix1[i][k] == -1:
                    k2 = matrix2[:,k]
                    ck2 = cmatrix[k]
                k += 1
            fck[j] += ck1*ck2*V(matrix2[0][j],k1[0],k2[0],matrix2[1][j],k1[1],k2[1])
    if matrix1[i][j] == -1:
        k = 0
        k1 = []
        ck1 = []
        while k < len(matrix1[0]):
            if matrix1[i][k] == 1:
                k1.append(matrix2[:,k])
                ck1.append(cmatrix[k])
            k += 1

        fck[j] += ck1[0]*ck1[1]*V(matrix2[0][j],k1[0][0],k1[1][0],
            matrix2[1][j],k1[0][1],k1[1][1])
    i += 1
    j += 1
if plot == True:
    return RK4C4(fck,cmatrix,dt,t0,initlist,imax,True,False,False)
if plot == False:
    return RK4C4(fck,cmatrix,dt,t0,initlist,imax,False,False,False)

def RK4C4(falist,cmatrix,dt,t0,initlist,imax,plot=False,printsteps=False,
returnarray=False):
    ti = t0
    ai = []
    for el in initlist:
        ai.append(el)
    i = 0
    t = np.array([ti])
    aki = []
    aki.append(ai[:])
    tk = sy.symbols('tk')
    falist1 = []
    i=0
    while i < len(falist):
        falist1.append(sy.lambdify([cmatrix,tk],falist[i]))
        i += 1
    i=0
    while i < imax:
        j=0
        k1 = []
        while j < len(falist):

```

```

        k1.append(falist1[j](ai, ti))
        j+=1
    j=0
    k2 = []
    dtk1 = [element *dt/2 for element in k1]
    aik2 = [a + b for a, b in zip(ai, dtk1)]
    while j<len(falist):
        k2.append(falist1[j](aik2, ti+dt/2))
        j+=1
    j=0
    k3=[]
    dtk2 = [element *dt/2 for element in k2]
    aik3 = [a + b for a, b in zip(ai, dtk2)]
    while j<len(falist):
        k3.append(falist1[j](aik3, ti+dt/2))
        j+=1
    j=0
    k4=[]
    dtk3 = [element *dt for element in k3]
    aik4 = [a + b for a, b in zip(ai, dtk3)]
    while j<len(falist):
        k4.append(falist1[j](aik4, ti+dt))
        j+=1
    j = 0
    while j <len(falist):
        ai[j] += (dt/6)*(k1[j]+2*k2[j]+2*k3[j]+k4[j])
        j+=1
    t = np.append(t, ti)
    aki.append(ai[:])
    ti += dt
    i += 1
arrayk = np.array(aki)
if plot == True:
    j=0
    while j < len(falist):
        plt.plot(t, np.abs(arrayk[:, j])**2)
        j+=1
if printsteps == True:
    print(t, ak)
return arrayk, t

```

```

beta = 10
matrix1 = np.array([[ -1, 1, 1, 0, 0, 0, 0, 0, 0, 0], [ -1, 0, 0, 1, 1, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 1, -1, 1, 0, 0, 0], [0, 0, 0, 0, 0, 0, 1, -1, 1, 0, 0], [0, 0, 0, 0, 0, 0, 1, 0, 0, 1, -1]])
rows = 5
columns = 10
Z =np.zeros((rows, columns), dtype=int)
newmatrix1=np.asarray(np.bmat([[matrix1, Z], [Z, matrix1]]))
n= 2
i=0
while i<n:
    newmatrix = np.asarray(np.bmat([[matrix1, Z], [Z, matrix1]]))
    matrix1 = newmatrix
    Z = np.zeros(((2**(i+1))*rows, (2**(i+1))*columns), dtype = int)

```

```

        i+=1
#print(len(matrix1[0]))
list2 = [[16,15,1,13,3,8,5,29,24,27],[-2,-10,8,-13,11,-14,-25,37,62,21]]
m = 4
i =0
while i < m:
    newlist0 = [element *2**(i+1) for element in list2 [0][0:10]]
    newlist1 = [element *2**(i+1) for element in list2 [1][0:10]]
    list2 [0].extend(newlist0)
    list2 [1].extend(newlist1)
    i+=1
list3 = np.asarray(list2)
list4 = list3[:,0:41]
initcond = []
n= -1/6
i = 0
while i <10:
    initcond.append((np.sqrt(list4 [0, i]**2+list4 [1, i]**2))*n*np.exp(1j*i))
    i+=1
j=0
while j <31:
    initcond.append((np.sqrt(list4 [0, j]**2+list4 [1, j]**2))*n*np.exp(1j*j)/100)
    j+=1
getplot5(matrix1, list4, 0.01, 0, initcond, 800, True)
plt.grid(True)
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.title('41_Mode_Cluster_with_Initial_Amplitude_greater_in_First_10_Modes')

```

## Plotting Invariants

```

def getinvariants(matrix1, matrix2, dt, t0, initlist, imax, plot = False, printarray=False):
    fck = sy.zeros(len(matrix2[0]), 1, dtype=np.complex)
    cmatrix = []
    i = 0
    while i<len(matrix2[0]):
        cmatrix.append(sy.symbols('c'+str(i+1)))
        i+=1
    j=0
    while j<len(matrix1[0]):
        i = 0
        while i<len(matrix1[:,0]):
            if matrix1[i][j] == 1:
                k = 0
                while k<len(matrix1[0]):
                    if k ==j:
                        k+=1
                    else:
                        if matrix1[i][k] == 1:
                            k1 = -matrix2[:,k]
                            ck1 = sy.conjugate(cmatrix[k])
                            #print(cmatrix[k])
                            #print(ck1)

```

```

        if matrix1[i][k] == -1:
            k2 = matrix2[:,k]
            ck2 = cmatrix[k]
            k+=1
        fck[j] += ck1*ck2*V(matrix2[0][j],k1[0],k2[0],matrix2[1][j],k1[1],k2[1])
    if matrix1[i][j] == -1:
        k = 0
        k1 = []
        ck1 = []
        while k<len(matrix1[0]):
            if matrix1[i][k] == 1:
                k1.append(matrix2[:,k])
                ck1.append(cmatrix[k])
            k+=1

        fck[j] += ck1[0]*ck1[1]*V(matrix2[0][j],k1[0][0],k1[1][0],
        matrix2[1][j],k1[0][1],k1[1][1])
    i+=1
j+=1

```

```

return RK4CInv(fck, cmatrix, matrix1, matrix2, dt, t0, initlist, imax, True, False, False)

```

```

def RK4CInv(falist, cmatrix, matrix1, matrix2, dt, t0, initlist, imax, plot = False,
printsteps = False, returnarray = False):

```

```

    ti = t0
    ai = []
    for el in initlist:
        ai.append(el)
    i = 0
    t = np.array([ti])
    aki = []
    aki.append(ai[:])
    tk = sy.symbols('tk')
    falist1 = []
    i=0
    while i < len(falist):
        falist1.append(sy.lambdify([cmatrix, tk], falist[i]))
        i+=1
    #print(falist1)
    i=0
    while i < imax:
        j=0
        k1 = []
        while j<len(falist):
            k1.append(falist1[j](ai, ti))
            j+=1
        j=0
        k2 = []
        dtk1 = [element *dt/2 for element in k1]
        aik2 = [a + b for a, b in zip(ai, dtk1)]
        while j<len(falist):
            k2.append(falist1[j](aik2, ti+dt/2))
            j+=1
        j=0

```

```

k3=[]
dtk2 = [element *dt/2 for element in k2]
aik3 = [a + b for a, b in zip(ai, dtk2)]
while j<len(falist):
    k3.append(falist1[j](aik3, ti+dt/2))
    j+=1
j=0
k4=[]
dtk3 = [element *dt for element in k3]
aik4 = [a + b for a, b in zip(ai, dtk3)]
while j<len(falist):
    k4.append(falist1[j](aik4, ti+dt))
    j+=1
j = 0
while j <len(falist):
    ai[j] += (dt/6)*(k1[j]+2*k2[j]+2*k3[j]+k4[j])
    j+=1
t = np.append(t, ti)
aki.append(ai[:])
ti += dt
i += 1
arrayk = np.array(aki)
if plot == True:
    plt.plot(t, np.real(Hamiltonian(matrix1, matrix2, arrayk)))
    plt.plot(t, Energy(matrix2, arrayk))
    plt.plot(t, Enstrophy(matrix2, arrayk))
    plt.legend(['Hamiltonian', 'Energy', 'Enstrophy'])
if printsteps == True:
    print(t, ak)
if returnarray == True:
    return arrayk, t
return arrayk, t

def Hamiltonian(matrix1, matrix2, arrayk):
H = np.zeros(len(arrayk[:,0]), dtype=np.complex)
k=0
#print(len(arrayk[:,0]))
while k<len(arrayk[:,0]):
    i = 0
    while i < len(matrix2[0]):
        H[k] += -beta*matrix2[0][i]*abs(arrayk[k,i])**2/(matrix2[0][i]**2
        +matrix2[1][i]**2)
        i+=1
    i=0
    while i <len(matrix1[0]):
        j=0
        while j<len(matrix1[:,0]):
            if matrix1[j][i]==-1:
                l=0
                klist =[]
                cklist = []
                while l<len(matrix1[0]):
                    if matrix1[j][l]==1:
                        klist.append(matrix2[:,l])

```

```

        cklist.append(arrayk[k,l])
        l+=1
    H[k] += 1j*3*V(matrix2[0][i], klist[0][0], klist[1][0], matrix2[1][i],
        klist[0][1], klist[1][1])* (cklist[0]*cklist[1]*np.conj(arrayk[k,i]) -

        j+=1
        i+=1
    k+=1
    return H

def Energy(matrix2, arrayk):
    E = np.zeros(len(arrayk[:,0]))
    k=0
    while k<len(arrayk[:,0]):
        i = 0
        while i<len(matrix2[0]):
            E[k]+=beta*matrix2[0,i]*abs(arrayk[k,i])**2/(matrix2[0,i]**2+matrix2[1,i]**2)
            i+=1
        k+=1
    return E

def Enstrophy(matrix2, arrayk):
    E = np.zeros(len(arrayk[:,0]))
    k=0
    while k<len(arrayk[:,0]):
        i = 0
        while i<len(matrix2[0]):
            E[k]+=beta*matrix2[0,i]*abs(arrayk[k,i])**2
            i+=1
        k+=1
    return E

from sympy import Matrix

def getinvariantsMR(matrix1, matrix2, dt, t0, initlist, imax, plot = False, printarray=False):
    fck = sy.zeros(len(matrix2[0]), 1, dtype=np.complex)
    cmatrix = []
    i = 0
    while i<len(matrix2[0]):
        cmatrix.append(sy.symbols('c'+str(i+1)))
        i+=1
    j=0
    while j<len(matrix1[0]):
        i = 0
        while i<len(matrix1[:,0]):
            if matrix1[i][j] == 1:
                k = 0
                while k<len(matrix1[0]):
                    if k == j:
                        k+=1
                    else:
                        if matrix1[i][k] == 1:
                            k1 = -matrix2[:,k]

```

```

        ck1 = sy.conjugate(cmatrix[k])
        #print(cmatrix[k])
        #print(ck1)
        if matrix1[i][k] == -1:
            k2 = matrix2[:,k]
            ck2 = cmatrix[k]
            k+=1
        fck[j] += ck1*ck2*V(matrix2[0][j],k1[0],k2[0],matrix2[1][j],k1[1],k2[1])
    if matrix1[i][j] == -1:
        k = 0
        k1 = []
        ck1 = []
        while k<len(matrix1[0]):
            if matrix1[i][k] == 1:
                k1.append(matrix2[:,k])
                ck1.append(cmatrix[k])
            k+=1

        fck[j] += ck1[0]*ck1[1]*V(matrix2[0][j],k1[0][0],k1[1][0],
            matrix2[1][j],k1[0][1],k1[1][1])
    i+=1
j+=1

```

```

return RK4CInvMR(fck, cmatrix, matrix1, matrix2, dt, t0, initlist, imax, True, False, False)

```

```

def RK4CInvMR(falist, cmatrix, matrix1, matrix2, dt, t0, initlist, imax, plot = False,
printsteps = False, returnarray = False):

```

```

    ti = t0
    ai = []
    for el in initlist:
        ai.append(el)
    i = 0
    t = np.array([ti])
    aki = []
    aki.append(ai[:])
    tk = sy.symbols('tk')
    falist1 = []
    i=0
    while i < len(falist):
        falist1.append(sy.lambdify([cmatrix, tk], falist[i]))
        i+=1
    i=0
    while i < imax:
        j=0
        k1 = []
        while j<len(falist):
            k1.append(falist1[j](ai, ti))
            j+=1
        j=0
        k2 = []
        dtk1 = [element *dt/2 for element in k1]
        aik2 = [a + b for a, b in zip(ai, dtk1)]
        while j<len(falist):
            k2.append(falist1[j](aik2, ti+dt/2))

```

```

        j+=1
    j=0
    k3=[]
    dtk2 = [element *dt/2 for element in k2]
    aik3 = [a + b for a, b in zip(ai, dtk2)]
    while j<len(falist):
        k3.append(falist1[j](aik3, ti+dt/2))
        j+=1
    j=0
    k4=[]
    dtk3 = [element *dt for element in k3]
    aik4 = [a + b for a, b in zip(ai, dtk3)]
    while j<len(falist):
        k4.append(falist1[j](aik4, ti+dt))
        j+=1
    j = 0
    while j <len(falist):
        ai[j] += (dt/6)*(k1[j]+2*k2[j]+2*k3[j]+k4[j])
        j+=1
    t = np.append(t, ti)
    aki.append(ai[:])
    ti += dt
    i += 1
arrayk = np.array(aki)
if plot == True:
    nullspace = np.asarray(Matrix(matrix1).nullspace())
    i=0
    while i< len(nullspace):
        k = 0
        mrvn = []
        while k < len(arrayk[:,0]):
            mrvn.append(np.dot(nullspace[i], np.abs(arrayk[k,:])**2))
            k+=1
        mrvnarr = np.asarray(mrvn)
        plt.plot(t, mrvnarr)
        i+=1
if printsteps == True:
    print(t, ak)
if returnarray == True:
    return arrayk, t
return arrayk, t

```

### Average Amplitudes

```

def getaverages(matrix1, matrix2, dt, t0, initlist, imax, plot = False, printarray=False):
    fck = sy.zeros(len(matrix2[0]), 1, dtype=np.complex)
    cmatrix = []
    i = 0
    while i<len(matrix2[0]):
        cmatrix.append(sy.symbols('c'+str(i+1)))
        i+=1
    j=0
    while j<len(matrix1[0]):

```



```

i = 0
while i < len(matrix1[:, 0]):
    if matrix1[i][j] == 1:
        k = 0
        while k < len(matrix1[0]):
            if k == j:
                k += 1
            else:
                if matrix1[i][k] == 1:
                    k1 = -matrix2[:, k]
                    ck1 = sy.conjugate(cmatrix[k])
                if matrix1[i][k] == -1:
                    k2 = matrix2[:, k]
                    ck2 = cmatrix[k]
                k += 1
            fck[j] += ck1*ck2*V(matrix2[0][j], k1[0], k2[0], matrix2[1][j], k1[1], k2[1])
        if matrix1[i][j] == -1:
            k = 0
            k1 = []
            ck1 = []
            while k < len(matrix1[0]):
                if matrix1[i][k] == 1:
                    k1.append(matrix2[:, k])
                    ck1.append(cmatrix[k])
                k += 1
            fck[j] += ck1[0]*ck1[1]*V(matrix2[0][j], k1[0][0], k1[1][0],
            matrix2[1][j], k1[0][1], k1[1][1])
        i += 1
    #print(fck)
    j += 1

```

```

return RK4CAvg(fck, cmatrix, matrix1, matrix2, dt, t0, initlist, imax, True, False, False)

```

```

def RK4CAvg(falist, cmatrix, matrix1, matrix2, dt, t0, initlist, imax, plot = False,
printsteps = False, returnarray = False):

```

```

    ti = t0
    ai = []
    for el in initlist:
        ai.append(el)
    i = 0
    t = np.array([ti])
    aki = []
    aki.append(ai[:])
    tk = sy.symbols('tk')
    falist1 = []
    i=0
    while i < len(falist):
        falist1.append(sy.lambdify([cmatrix, tk], falist[i]))
        i += 1
    i=0
    while i < imax:
        j=0
        k1 = []

```

```

while j<len(falist):
    k1.append(falist1[j](ai,ti))
    j+=1
j=0
k2 = []
dtk1 = [element *dt/2 for element in k1]
aik2 = [a + b for a, b in zip(ai, dtk1)]
while j<len(falist):
    k2.append(falist1[j](aik2,ti+dt/2))
    j+=1
j=0
k3=[]
dtk2 = [element *dt/2 for element in k2]
aik3 = [a + b for a, b in zip(ai, dtk2)]
while j<len(falist):
    k3.append(falist1[j](aik3,ti+dt/2))
    j+=1
j=0
k4=[]
dtk3 = [element *dt for element in k3]
aik4 = [a + b for a, b in zip(ai, dtk3)]
while j<len(falist):
    k4.append(falist1[j](aik4,ti+dt))
    j+=1
j = 0
while j <len(falist):
    ai[j] += (dt/6)*(k1[j]+2*k2[j]+2*k3[j]+k4[j])
    j+=1
t = np.append(t,ti)
aki.append(ai[:])
ti += dt
i += 1
arrayk = np.array(aki)
if plot == True:
    average = []
    j=0
    while j < len(matrix2[0]):
        avg = 0
        for el in arrayk[:,j]:
            avg +=abs(el)**2
        average.append(avg/len(arrayk[:,0]))
        j+=1
    avgarray =np.asarray(average)
    kplot = []
    i=0
    while i <len(matrix2[0]):
        kplot.append(np.sqrt(matrix2[0,i]**2+matrix2[1,i]**2))
        i +=1
    kplotarray = np.asarray(kplot)
    plt.plot(np.log(kplotarray),np.log(avgarray), 'bo')
    m, b = np.polyfit(np.log(kplotarray),np.log(avgarray), 1)
    plt.plot(np.log(kplotarray), m*np.log(kplotarray) + b)
    plt.grid(True)
    print(m,b)

```

```

if printsteps == True:
    print(t, ak)
if returnarray == True:
    #return arrayk, t
return arrayk, t

```

## Phase Analysis

```

def getphases2(matrix1, matrix2, dt, t0, initlist, imax, plot = False, printarray=False):
    fck = sy.zeros(len(matrix2[0]), 1, dtype=np.complex)
    cmatrix = []
    i = 0
    while i < len(matrix2[0]):
        cmatrix.append(sy.symbols('c'+str(i+1)))
        i+=1
    matrix3 = np.zeros((3, len(matrix1[:, 0])), dtype = np.int32)
    j=0
    while j < len(matrix1[0]):
        i = 0
        while i < len(matrix1[:, 0]):
            if matrix1[i][j] == 1:
                k = 0
                while k < len(matrix1[0]):
                    if k == j:
                        k+=1
                    else:
                        if matrix1[i][k] == 1:
                            k1 = -matrix2[:, k]
                            ck1 = sy.conjugate(cmatrix[k])
                        if matrix1[i][k] == -1:
                            k2 = matrix2[:, k]
                            ck2 = cmatrix[k]
                        k+=1
                fck[j] += ck1*ck2*V(matrix2[0][j], k1[0], k2[0], matrix2[1][j], k1[1], k2[1])
            if matrix1[i][j] == -1:
                matrix3[0, i]+=j
                k = 0
                k1 = []
                ck1 = []
                while k < len(matrix1[0]):
                    if matrix1[i][k] == 1:
                        k1.append(matrix2[:, k])
                        ck1.append(cmatrix[k])
                    if matrix3[1, i]==0:
                        matrix3[1, i]+=k
                    else:
                        matrix3[2, i]+=k
                k+=1

                fck[j] += ck1[0]*ck1[1]*V(matrix2[0][j], k1[0][0], k1[1][0],
                matrix2[1][j], k1[0][1], k1[1][1])
            i+=1
        j+=1

```

```

return RK4CPhases2(fck , cmatrix , matrix1 , matrix2 , matrix3 , dt , t0 , initlist , imax , True , False

```

```

def RK4CPhases2(falist , cmatrix , matrix1 , matrix2 , matrix3 , dt , t0 , initlist , imax ,
plot = False , printsteps = False , returnarray = False):

```

```

    ti = t0
    ai = []
    for el in initlist:
        ai.append(el)
    i = 0
    t = np.array([ti])
    aki = []
    aki.append(ai[:])
    tk = sy.symbols('tk')
    falist1 = []
    i=0
    while i < len(falist):
        falist1.append(sy.lambdify([cmatrix, tk], falist[i]))
        i+=1
    i=0
    while i < imax:
        j=0
        k1 = []
        while j<len(falist):
            k1.append(falist1[j](ai, ti))
            j+=1
        j=0
        k2 = []
        dtk1 = [element *dt/2 for element in k1]
        aik2 = [a + b for a, b in zip(ai, dtk1)]
        while j<len(falist):
            k2.append(falist1[j](aik2, ti+dt/2))
            j+=1
        j=0
        k3=[]
        dtk2 = [element *dt/2 for element in k2]
        aik3 = [a + b for a, b in zip(ai, dtk2)]
        while j<len(falist):
            k3.append(falist1[j](aik3, ti+dt/2))
            j+=1
        j=0
        k4=[]
        dtk3 = [element *dt for element in k3]
        aik4 = [a + b for a, b in zip(ai, dtk3)]
        while j<len(falist):
            k4.append(falist1[j](aik4, ti+dt))
            j+=1
        j = 0
        while j < len(falist):
            ai[j] += (dt/6)*(k1[j]+2*k2[j]+2*k3[j]+k4[j])
            j+=1
        t = np.append(t, ti)
        aki.append(ai[:])
        ti += dt

```

```

    i += 1
arrayk = np.array(aki)
if plot == True:
    rlist = []
    philist = []
    k=0
    while k<len(arrayk[:,0]):
        rvalue = 0
        phivalue = 0
        i = 0
        while i <len(matrix3[0]):
            rvalue +=np.exp(1j*np.angle(arrayk[k,matrix3[1,i]]*
            arrayk[k,matrix3[2,i]]*np.conj(arrayk[k,matrix3[0,i]])))
            phivalue +=np.angle(np.exp(1j*np.angle(arrayk[k,matrix3[1,i]]*
            arrayk[k,matrix3[2,i]]*np.conj(arrayk[k,matrix3[0,i]]))))
            i+=1
        rlist.append(abs(rvalue)/len(matrix1[:,0]))
        philist.append(phivalue%(2*np.pi))
        k+=1
    rarray = np.asarray(rlist)
    phiarray = np.asarray(philist)
    print(np.amax(rarray))
    ravg = 0
    for el in rarray:
        ravg+=el
    print(ravg/len(arrayk[:,0]))
    plt.plot(t,rarray)
    plt.plot(t,phiarray)
    plt.legend(['R(t)', '$\Phi(t)$'])
if printsteps == True:
    print(t,ak)
if returnarray == True:
    #return arrayk, t
return arrayk, t

```