



Pattern Formation in Clouds

A study of Turing patterns in a general cloud model using finite element methods.

by Oliver White

A thesis presented for the degree of
Master of Science

College of Engineering, Mathematics and Physical Sciences
University of Exeter

August 21, 2023

Abstract

The clouds in the sky can be seen to produce all manner of patterns across the globe however the mathematics behind the formation of these structures is largely unknown. Unlike general atmospheric flows, which are relatively well represented by the Navier-Stokes equations, clouds are parameterized in a number of different ways. In [Rosemeier et al., 2018] a general cloud model is presented, based off the two operational cloud models and a research model. Following this paper, it is shown in [Rosemeier and Spichtinger, 2021] that the addition of spatially diffusive terms (Laplacian operators) gives the model the required form for Turing patterns to form. Turing patterns have been proven to be the driver behind pattern formation in a multitude of different fields and [Rosemeier and Spichtinger, 2021] proves analytically that the same methodology can be applied to certain types of cloud models. The model is also resolved numerically using a spectral method and [Rosemeier and Spichtinger, 2021] presents results showing the formation of structures away from the equilibrium under different conditions.

Finite element methods have become a popular way to solve partial differential equations, particularly in more complex geometries, due to the flexibility of the grid that the equations are resolved on. Atmospheric models are moving towards this numerical methodology due to this property and hence the physical models that make up these large-scale forecasting models must be converted to this new numerical method. In this dissertation, I recreate the results from [Rosemeier and Spichtinger, 2021] using a finite element method and explore the possibility of pattern formation using assumptions and constants from operational models.



Figure 1: Clouds can regularly be seen to form wavelike patterns yet the mechanism for their formation is not fully understood mathematically [Yates, 2022].

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 4 |
| 1.1 | Importance of Clouds in GCM's | 5 |
| 1.2 | Atmospheric Models | 6 |
| 1.3 | Aims | 7 |
| 2 | The Turing Model | 7 |
| 3 | Cloud Microphysical Processes | 10 |
| 3.1 | Condensation and Evaporation of Cloud Droplets | 12 |
| 3.2 | Autoconversion | 13 |
| 3.3 | Accretion | 13 |
| 3.4 | Evaporation of rain droplets | 14 |
| 3.5 | Precipitation | 14 |
| 3.6 | Comparison of Warm-Rain Cloud Models | 15 |
| 4 | Stability Analysis | 17 |
| 4.1 | No rain from above ($B=0$) | 17 |
| 4.2 | $B = 0$ | 19 |
| 5 | Numerical Methodology | 20 |
| 5.1 | Finite Element Methods | 20 |
| 5.2 | FEM for the general cloud scheme | 21 |
| 6 | Results and Discussion | 22 |
| 6.1 | $\beta = 2, B=0$ | 22 |
| 6.1.1 | 1D case | 22 |
| 6.1.2 | 2D Case | 24 |
| 6.2 | $\beta = 2, B \neq 0$ | 26 |
| 6.3 | $\beta = 3$ | 27 |
| 6.4 | Using the IFS exponents | 27 |
| 7 | Conclusion | 28 |
| A | Evaluating the Jacobian at the non-trivial equilibrium | 31 |
| B | Results from the spectral numerical method | 32 |
| C | 1D Code | 35 |
| D | 2D Code | 36 |
| E | 1D results at $t=120$ | 37 |

List of Figures

| | | |
|---|--|---|
| 1 | Clouds can regularly be seen to form wavelike patterns yet the mechanism for their formation is not fully understood mathematically [Yates, 2022]. | 1 |
| 2 | Simulated Turing Patterns (on the right of each coupled picture) show incredible similarities with Sea Shells and Fish [Kondo and Miura, 2010]. | 4 |
| 3 | The incoming shortwave solar radiation (yellow arrows) is reflected better by low lying clouds, whilst the outgoing longwave radiation (red arrows) is trapped in the atmosphere more effectively by higher clouds [Ceppi and Williams, 2022]. | 5 |
| 4 | Two different types of atmospheric model grids. (a) is an example of a latitude-longitude grid whilst (b) is an example of a reduced grid [Williamson and Laprise, 2000]. One can see that the singularities at the pole in the latitude-longitude grid are removed by using a reduced grid. | 6 |

| | | |
|----|---|----|
| 5 | Figure 7.1 in [ECMWF, 2017] shows the multitude of different variables (in yellow boxes) and processes (arrows) represented in their cloud scheme. For the avoidance of doubt, the comparison and analysis of cloud processes in this dissertation will ignore all solid variables (snow and cloud ice) and any processes involving these. In the following subsections I shall describe in detail the condensation, evaporation, precipitation, auto-conversion and accretion processes labelled here. | 10 |
| 6 | The analytical solution u (blue line) is approximated (red dashed line) by a linear combination of basis functions ψ_i multiplied by coefficients u_i on a uniformly distributed mesh in subfigure (a) and on a non-uniform grid in subfigure (b) [COMSOL, 2017]. | 20 |
| 7 | The variation of q_c in one dimensional space at times $t=20$, $t=200$, $t=2000$. (Note the different scales on the y axis). After 20 timesteps the value of q_c is barely changed from the equilibrium value. However, by 200 timesteps a wavelike pattern is becoming more apparent and a clear sinusoidal wave can be seen at 2000 timesteps. | 23 |
| 8 | The variation of q_r in one dimensional space at times $t=20$, $t=200$, $t=2000$. (Note the different scales on the y axis). In a similar fashion to the cloud droplets q_c , the rain droplets show barely any variation away from the non-trivial equilibrium after 20 timesteps. Again, by $t=200$ there is a clearer wavelike pattern, which is fully realised by $t=2000$ | 24 |
| 9 | The distribution of cloud water q_c in two dimensional space at different times ($t=1/10/60/120$). Noting the different scales in the colour bars, one can see that by the end of the simulation patterns are beginning to emerge. | 25 |
| 10 | The distribution of rain water q_r in two dimensional space at different times ($t=1/10/60/120$). Here the cloud model goes from being very close to the equilibrium at $t=1$ to wavelike patterns becoming more visible at larger times. | 26 |
| 11 | q_c for the case of $B=0.1$ at $t=2000$. Wavelike pattern formation can be seen in both species on a very similar scale to that seen in Figures 7c and 8c. | 27 |
| 12 | When the value of β is increased to 3 wavelike pattern formation is still apparent in both cloud species. | 27 |
| 13 | Figure 4 in [Rosemeier and Spichtinger, 2021]. This shows how structures appear in both the cloud and rain droplets over time. This figure is directly comparable with Figures 7 and 8 | 32 |
| 14 | Figure 5 in [Rosemeier and Spichtinger, 2021] showing the spatial distribution of the cloud droplets. Patterns can be seen to emerge after a shorter period than in the one dimensional case. This figure is directly comparable with Figure 9. | 33 |
| 15 | Figure 6 in [Rosemeier and Spichtinger, 2021] showing the spatial distribution of rain droplets. Similarly to the cloud droplets, patterns are seen much sooner than in the one dimensional case. This figure is directly comparable with Figure 10 | 34 |
| 16 | The first part of code used to solve the cloud scheme in one-dimension. | 35 |
| 17 | The second part of code used to solve the cloud scheme in one-dimension. | 36 |
| 18 | The first part of code used to solve the cloud scheme in two-dimensions. | 36 |
| 19 | The second part of code used to solve the cloud scheme in two-dimensions. | 37 |
| 20 | The spatial distribution of q_c and q_r at $t=120$ | 37 |

1 Introduction

Pattern formations appear everywhere in nature, from animal furs to rock formations, sand dunes to population densities and importantly the weather. Over the years many different mathematical models have been developed to explain many of these beautiful phenomena. One of the most popular models is the Turing model presented in his 1952 paper ‘The chemical basis of morphogenesis’ ([Turing, 1952]). Due to its simplicity and applicability, it has been used to describe pattern formation across a wide spectra of research fields [Ball, 2015].

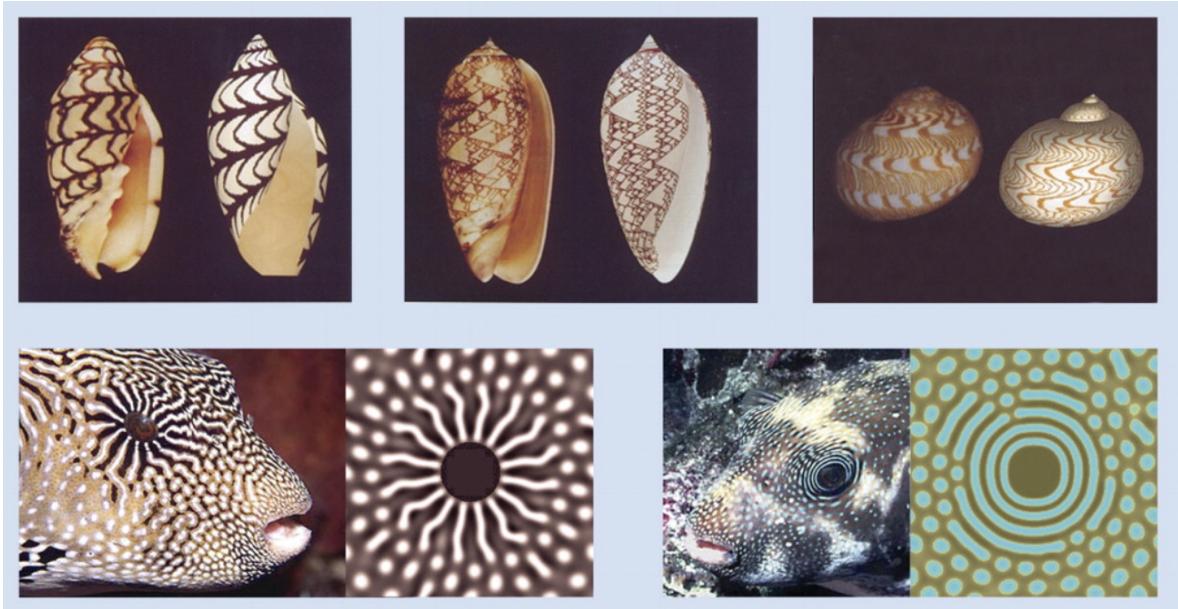


Figure 2: Simulated Turing Patterns (on the right of each coupled picture) show incredible similarities with Sea Shells and Fish [Kondo and Miura, 2010].

Whilst the development of patterns in organisms and between chemical reactants is relatively well understood, there is a lack of research into Turing patterns in atmospheric processes such as clouds ([Rosemeier and Spichtinger, 2021]).

Clouds prove to be a difficult process to model, occurring at scales smaller than the grid boxes in many large-scale models. One way that climate models overcome this is to consider the ‘bulk’ properties of the cloud, such as the water content, averaged mass and drop concentration [Rosemeier et al., 2018], [Khairoutdinov and Kogan, 2000]. This type of parameterisation is called bulk cloud microphysics.

[Kessler, 1969] suggested two types of cloud droplets, (rain and cloud based on their size) from which a set of diagnostic equations can be derived to describe the cloud’s evolution. A number of different cloud models containing this categorisation of droplets exist, both in research papers (e.g. [Wacker, 1992], [Rosemeier et al., 2018]) and in operational models (e.g. [Doms et al., 2011], [ECMWF, 2017]).

The model presented in [Wacker, 1992] is adapted in [Rosemeier and Spichtinger, 2021] by adding a spatial Laplacian so that the cloud model matches the required form to produce Turing patterns. Under certain assumptions on the parameters of the adapted model, a special case is presented in [Rosemeier and Spichtinger, 2021] where Turing patterns can be seen to form over time in both cloud droplet species.

The choice of numerical method to solve equations describing atmospheric processes (such as cloud models) has implications in the accuracy, speed and stability of atmospheric forecasting models. Current models are changing their models’ numerical methodology in favour of finite element methods to increase the speed of the calculations and to make the most of increasing supercomputer processing capability [MetOffice, 2021].

This project brings these two topics together succinctly. The structure of this dissertation is as follows.

Firstly, I discuss the importance of clouds in general circulation models (GCM’s) and why accurate

parameterisation of this sub-grid process is so important. Secondly, I explain why finite element numerical methods are becoming the numerical methodology of choice in modern climate forecasting models. Thirdly, I discuss Turing's model for pattern formation and explain how the model, initially intended for chemical reactions, can be applied across multiple fields of science. Fourthly, a literature review of two operational models and a research model (that has already been used to show pattern formation) is provided. Here, I discuss some of the physical processes that occur within clouds and compare how they are parameterised by these models. Next, I apply linear stability analysis to a simplified version of the general cloud model presented in [Rosemeier and Spichtinger, 2021] to discover the conditions required for Turing pattern formation. The penultimate step contains a brief explanation of the finite element methodology and how this was applied to the cloud model. Finally, I present and discuss the results of solving the general cloud model using finite element methods, under conditions both favourable and unfavourable to pattern formation.

1.1 Importance of Clouds in GCM's

Clouds play a key role in atmospheric processes and influence both incoming and outgoing radiation along with affecting the Earth's hydrological cycle [Borde and Isaka, 1996] [Rosemeier et al., 2018]. As the global temperature rises it is expected to affect the cover, altitude and thickness of clouds [Ceppi and Williams, 2022]. In turn, this will have a knock-on effect on the rainfall across the planet which affects thousands of processes; from permafrost degradation to migration of animals, which will cause further effects to the Earth's energy and carbon budgets.

The Earth is generally a dark object (being over 70% water) and thus absorbs incoming solar radiation well. On the other hand, thick white clouds can act to reflect this incoming radiation back to space thereby cooling the planet. Low lying clouds do this most effectively as they are thicker than clouds higher up in the atmosphere. High, sparse clouds have the opposite effect on the Earth's energy budget. These clouds are easily penetrated by shortwave solar radiation but stop reflected longwave infrared radiation from escaping to space and therefore act as a 'blanket' for the planet [Ceppi and Williams, 2022].

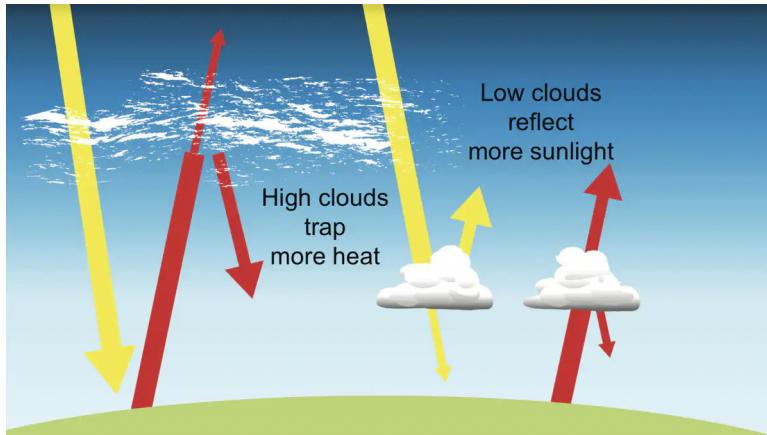


Figure 3: The incoming shortwave solar radiation (yellow arrows) is reflected better by low lying clouds, whilst the outgoing longwave radiation (red arrows) is trapped in the atmosphere more effectively by higher clouds [Ceppi and Williams, 2022].

The lack of a set of general equations for cloud processes is demonstrated by the large range of different cloud models used. The difference in how cloud processes are parameterised in terms of the bulk properties of the cloud causes uncertainty between such models. In turn, the interaction of clouds with other atmospheric processes causes massive uncertainty when modelling the future weather and climate. In fact, such is the uncertainty around the positive and negative feedbacks that clouds give to different atmospheric and land processes that overall, clouds constitute one of the largest discrepancies between different models [Khouider et al., 2019]. In some models, clouds increase the global warming effects dramatically, whilst in others they provide a slight cooling effect or have a negligible impact [Ceppi and Williams, 2022].

The radiative transfer of either the longwave radiation from the earth or the shortwave radiation from the sun through the clouds in the atmosphere is a complex and key process [Khouider et al., 2019], [Grabowski and Wang, 2013]. Observations show that generally cumulus and stratocumulus clouds (low lying thick clouds [Office, ndb]) are inhomogeneous (structured) both vertically and horizontally. However, most models (due to the grid size) cannot allow for this structured form and thus consider the properties of the cloud in each grid optically homogeneous [Zhang et al., 2018]. Unfortunately, there are massive differences between radiative transfer in homogeneous and inhomogeneous media [Rosemeier and Spichtinger, 2021]. A greater understanding of how and when structured clouds occur would be very relevant to understanding when the radiative transfer assumptions used in models may be inaccurate.

1.2 Atmospheric Models

The world's largest atmospheric models used by companies such as the Met Office are being updated as computers' processing speed progressively increase. It is important that we use this increase in computational power to produce more accurate future weather forecasts and predictions. This will allow further understanding of the carbon budget and how the climate will change over the next century as well as providing more accurate weather forecasting in the short term.

A simple way to increase the accuracy of these models is to increase the resolution of the grid that the equations are defined on and solved by the supercomputer [MetOffice, 2021]. For example, many models in the 1950s had a grid resolution of several hundred kilometres whereas newer models have a grid resolution of tens of kilometres [Collins et al., 2013]. Many of these atmospheric models (including the Met Office's own atmospheric model) have used a latitude longitude grid as shown in Figure 4a. However, as can be easily seen in this figure, the grid points on a latitude-longitude grid cluster at the poles. This congregation of grid points slows the calculations massively, causing a 'bottleneck' for the entire model [MetOffice, 2021].

Therefore, to overcome this issue, models are moving away from latitude-longitude meshes towards other more evenly spaced meshes such as that shown in Figure 4b. For example, the Met Office has moved towards a 'cubed-square mesh', where the surface of the sphere is divided into approximately equal sized quadrilateral grid boxes [MetOffice, nd].

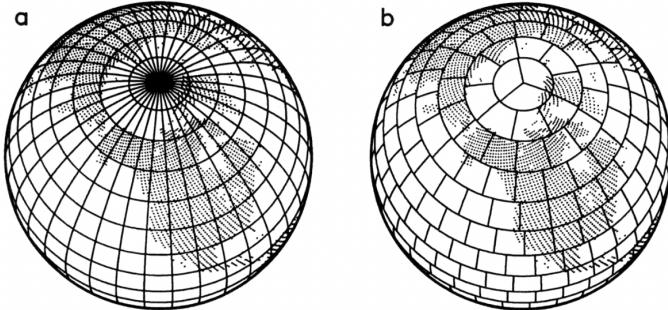


Figure 4: Two different types of atmospheric model grids. (a) is an example of a latitude-longitude grid whilst (b) is an example of a reduced grid [Williamson and Laprise, 2000]. One can see that the singularities at the pole in the latitude-longitude grid are removed by using a reduced grid.

Furthermore, one of the advantages of a latitude-longitude grid is that a finite difference numerical method can be used to solve the atmospheric equations. This is possible as the lines of latitude and longitude lie perfectly orthogonal to one another. This numerical method is one of the oldest and most well researched methods of solving partial differential equations.

This numerical method approximates the equations using a localised Taylor expansion and then solves these on a uniform grid ([Peiro and Sherwin, 2005]). One of the main sources of error in the finite difference approximation arises as a result of neglecting the higher order Taylor series terms [Caendkoelsch, 2018]. Finite difference numerical methods have many useful attributes dependant on the spatial discretisation used (see [LeVeque, 2007] for more detail on how finite difference methods are applied to partial differential equations (PDE's))

When models move away from the longitude-latitude grid the orthogonality is lost and thus a different numerical method is required to solve the equations. One such method that can deal with more complex geometries effectively is the finite *element* numerical method. Many operational models are thus moving towards this methodology.

The concept of predicting future weather patterns using numerical methods based on partial differential equations was originally suggested by Lewis Fry Richardson [[Collins et al., 2013](#)], whom the Met Office's project to convert to this new finite element system is named after [[MetOffice, nd](#)]. The key aim of this project was to maintain the accuracy and stability of the Unified Model (the Met Office's extensive numerical atmospheric model) whilst removing the issues arising from singularities at the Poles [[MetOffice, nd](#)].

A more detailed explanation of how the finite element method is used to solve partial differential equations (and reaction diffusion equations which are the basis of this dissertation) will be provided in Section 5.

1.3 Aims

[[Rosemeier and Spichtinger, 2021](#)] uses a generalised cloud model that shares a number of similarities with models used in current weather forecasting models to prove that Turing patterns are theoretically possible under certain assumptions. However, the numerical method chosen to solve this model is a 'pseudo-spectral' method.

As discussed above, there are a range of benefits to be obtained from finite element methods in climate modelling. Hence, the aim of this dissertation will be to reproduce the results showing pattern formation (see Section 7 of [[Rosemeier and Spichtinger, 2021](#)]) using a finite element numerical method. Upon solving the general cloud model, I will then aim to compare the two sets of results to identify the differences between the two numerical methods.

Furthermore, I aim to compare several different cloud models to understand their differences in parameterisation of cloud processes. This will allow me to compare the possibility for pattern formation and the types of patterns formed in these different models.

2 The Turing Model

The patterns seen throughout both the natural and human world are fascinating, beautiful and complex. Many different explanations and models have been suggested to account for natural phenomena.

For example, the quasi-hexagonal basalt pillars of the Giants Causeway can be explained by the lava cooling and releasing energy. The optimal angle at which energy is released is 120° , and hence a hexagonal pattern is observed [[Perkins, 2015](#)].

Another model of pattern formation, this time in chemical reactions, is the Belousov-Zhabotinsky reaction. Here a mixture of chemicals oscillates in colour over several hours with a regular period [[Taylor, 2002](#)].

However, there are few models that can be seen across quite so many facets of the natural and human world as the Turing Model. The simplicity of the model means that it can be easily applied to different species, geometries and physical systems. Another advantage of this model is that the equations can easily be linked to physical quantities [[Rosemeier and Spichtinger, 2021](#)].

In 1952 when contemplating how patterns in species arose, Alan Turing published a paper on morphogenesis which proposed how a system of equations relating to different chemicals could produce the patterns shown on so many species [[Turing, 1952](#)]. He showed that if at least two chemical species were in a steady state and if certain conditions were satisfied then instabilities could occur, and patterns evolve. One of the fascinating results from his paper is that adding diffusion to the equations (through a spatial Laplacian) acts to destabilise the system and create the patterns. This is counter intuitive as in most systems diffusion acts to smooth out patterns and shapes.

In this section, I will explain the model presented by Turing in his paper and the conditions for instability that result from this. Instability here refers to the system moving away from its equilibrium which causes the growth of patterns or structures.

Turing suggested a model of the form:

$$\frac{\partial u_1}{\partial t} = f_1(u_1, u_2) + D_1 \nabla^2 u_1, \quad (1a)$$

$$\frac{\partial u_2}{\partial t} = f_2(u_1, u_2) + D_2 \nabla^2 u_2. \quad (1b)$$

Here u_1 and u_2 correspond to the population/ concentration of two species whilst f_1 and f_2 are reaction terms that describe how the two species interact. D_1 and D_2 are the diffusion coefficients for each species respectively and we assume that these are constant (this is a reasonable assumption for many experimental situations [Cross and Greenside, 2009]). This model can be extended to more than two species by considering it in vector form:

$$\frac{\partial \vec{u}}{\partial t} = \vec{f}(\vec{u}_1, \vec{u}_2) + D \nabla^2 \vec{u}$$

where $\vec{u} \in \mathbb{R}^n$ and D is now a matrix of diffusion coefficients.

$$D = \begin{pmatrix} D_1 & 0 & \dots & 0 \\ 0 & D_2 & \dots & 0 \\ \dots & 0 & \dots & 0 \\ 0 & \dots & \dots & D_n \end{pmatrix}.$$

However, as our cloud model only contains two species, we will retain the two-component form for the remainder of this section.

To understand what conditions are required for pattern formation, we apply linear stability analysis to this model. Assuming that system 1 evolves within a bounded domain Ω , we take Neumann (zero flux) boundary conditions on the boundary $\partial\Omega$. Mathematically:

$$\frac{\partial u_1}{\partial n} = \vec{n} \cdot \nabla u_1, \quad (2a)$$

$$\frac{\partial u_2}{\partial n} = \vec{n} \cdot \nabla u_2, \quad (2b)$$

where \vec{n} is the outward pointing unit normal vector to Ω at $\partial\Omega$. Next, we assume that we have a steady state solution (u_{1e}, u_{2e}) satisfying system 1 which, in the absence of diffusion, gives $f_1(u_{1e}, u_{2e}) = 0$ and $f_2(u_{1e}, u_{2e}) = 0$.

By linearising about the steady state, it can be shown that an infinitely small perturbation (u'_1, u'_2) results in the following linearised dynamics (see cite a good book here for a more detailed analysis):

$$\frac{\partial u'_1}{\partial t} = a_{11}u'_1 + a_{12}u'_2, \quad (3a)$$

$$\frac{\partial u'_2}{\partial t} = a_{21}u'_1 + a_{22}u'_2, \quad (3b)$$

where $a_{i,j}$ represent the partial derivatives of \vec{f} with respect to $u_{i,j}$ respectively, evaluated at the steady state (u_{1e}, u_{2e}) . These entries make up the 2x2 Jacobian matrix, A :

$$A = \begin{pmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}. \quad (4)$$

The stability of the steady state is determined by looking at the eigenvalues, λ_i , of this matrix. The solution growth or decays over time proportionally to $e^{\lambda t}$ and hence the necessary condition for uniform stability is that $\text{Re}(\lambda_i) < 0$. Alternatively, for instability $\text{Re}(\lambda_i) > 0$ [Cross and Greenside, 2009].

Stability in the absence of diffusion is key as Turing hypothesised that it was the addition of diffusive terms that cause the instability.

The eigenvalues of the Jacobian satisfy the following:

$$\begin{vmatrix} a_{11} - \lambda & a_{12} \\ a_{21} & a_{22} - \lambda \end{vmatrix} = (a_{11} - \lambda)(a_{22} - \lambda) - a_{12}a_{21} = \lambda^2 - (a_{11} + a_{22})\lambda + a_{11}a_{22} - a_{12}a_{21} = 0. \quad (5)$$

One can see that the solutions to this equation are:

$$\lambda_{1,2} = \frac{1}{2} \text{Tr}(A) \pm \frac{1}{2} \sqrt{\text{Tr}(A)^2 - 4 \text{Det}(A)}, \quad (6)$$

where the Trace of A , $\text{Tr}(A) = a_{11} + a_{22}$, and the Determinant of A , $\text{Det}(A) = a_{11}a_{22} - a_{12}a_{21}$. Hence, for $\lambda_{1,2}$ to be negative and real we require the following two conditions:

- $\text{Tr}(A) < 0$;
- $\text{Det}(A) > 0$.

If both conditions are satisfied the system is uniformly stable. To understand what happens when diffusion is added, we consider a small perturbation to the basic state that is proportional to e^{iqx} where q is the wavenumber in the x direction. This results in the modified eigenvalue problem:

$$\mathbf{A}\mathbf{u} = \lambda\mathbf{u}, \quad (7)$$

where \mathbf{A} now is equal to:

$$\mathbf{A} = \begin{pmatrix} a_{11} - q^2 D_1 & a_{12} \\ a_{21} & a_{22} - q^2 D_2 \end{pmatrix}. \quad (8)$$

Assuming the two conditions above hold, and as q^2 , D_1 and D_2 are all positive, the $\text{Tr}(A)$ remains negative. Thus, the change in stability must come from the Determinant of A becoming negative under the addition of the diffusion terms. The Determinant has now become a quadratic in q^2 :

$$\text{Det}(A) = a_{11}a_{22} - D_2q^2a_{11} - D_1q^2a_{22} + D_1D_2q^4 - a_{12}a_{21}. \quad (9)$$

We find the minimum of $\text{Det}(A)$ by taking the derivative with respect to q^2 and setting this to zero.

$$\frac{d(\text{Det}(A))}{dq^2} = 2D_1D_2q^2 - a_{11}D_2 - a_{22}D_1 = 0. \quad (10)$$

Thus the minimum value of $\text{Det}(A)$ occurs at:

$$q^2 = \frac{D_2a_{11} + D_1a_{22}}{2D_1D_2}. \quad (11)$$

By substituting this back into equation 9 and simplifying, this gives us the minimum of $\text{Det}(A)$:

$$\min(\text{Det}(A)) = a_{11}a_{22} - a_{12}a_{21} - \frac{(D_1a_{22} + D_2a_{11})^2}{4D_1D_2}. \quad (12)$$

This is negative when the following inequality is satisfied:

$$D_1a_{22} + D_2a_{11} > 2\sqrt{D_1D_2(a_{11}a_{22} - a_{12}a_{21})}. \quad (13)$$

Finally, the 3 conditions for a system such as system 2, derived by Turing, that are required for a system that is initially stable and becomes unstable under the addition of diffusion terms are:

- $a_{11} + a_{22} < 0$;
- $a_{11}a_{22} - a_{12}a_{21} > 0$;
- $D_1a_{22} + D_2a_{11} > 2\sqrt{D_1D_2(a_{11}a_{22} - a_{12}a_{21})}$.

Interpreting these conditions leads to a number of further insights into the values of a_{ij} .

As the right hand side of the third condition must be positive a corollary is: $D_1a_{22} + D_2a_{11} > 0$. As both D_1 and D_2 are positive, this corollary along with the first condition implies that a_{11} and a_{22} must have the opposite sign.

Furthermore, this implies that the first term in the second condition ($a_{11}a_{22}$) is negative and thus for this condition to be satisfied overall, a_{12} and a_{21} must also have opposite signs.

These conditions can then be applied to any number of systems with differing functions f_i and species u_i dependant on the field in question. For example this process has been used to explain the patterns seen on the fur of a zebra to the patterns in sand ripples [Ball, 2015].

3 Cloud Microphysical Processes

In order to understand the physical processes described in the equations used to model pattern formation it is useful to have a thorough understanding of cloud microphysical processes and how they are parameterised within different large-scale models. This allows one to look at the equations used to model the pattern formation and understand whether the choices for the parameters are realistic.

A simple explanation of how clouds form and dissipate is as follows. When enough surface water has evaporated the air in the atmosphere can become saturated with water vapour. When this occurs alongside cooling or a drop in the atmospheric pressure then the water vapour condenses and becomes water droplets or ice crystals. These water droplets and ice crystals attach themselves to particles such as volcanic ash, dust, or salt crystals to form what we know as clouds. When these droplets acquire enough mass through various coalescence processes, they fall as rain, hail, or snow [UCAR, 2019].

A typical m^3 of cloudy air contains between 10^7 and 10^9 separate droplets [Grabowski et al., 2019] and these vary in many ways depending on the atmospheric conditions (such as temperature and pressure). For example, some clouds are made entirely of ice crystals (so called cirrus clouds), whilst warm-rain clouds, where the droplets are all in liquid phase, show two distinct sizes of droplets [Kessler, 1969].

There are several different ways to model the evolution of these cloud particles over time and on different length scales, each with different advantages and drawbacks.

Some of these include spectral continuity models and Lagrangian particle-based probability models ([Grabowski et al., 2019]), however the cloud models that I will be comparing will be bulk continuity models. These predict the mass fraction of different types of cloud particles that may be based on particle size or what phase they are in.

Modern forecasting models using this method contain cloud parameterisations that are based largely on the work done by [Kessler, 1969] who treats each of the different type of particle as a separate variable and then represents the processes between the particles as nonlinear terms. An example of the different types of cloud variables and the process between them are shown in Figure 5.

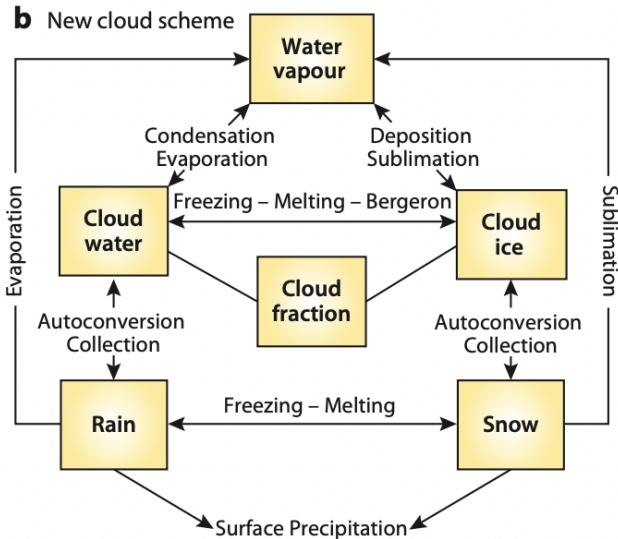


Figure 5: Figure 7.1 in [ECMWF, 2017] shows the multitude of different variables (in yellow boxes) and processes (arrows) represented in their cloud scheme. For the avoidance of doubt, the comparison and analysis of cloud processes in this dissertation will ignore all solid variables (snow and cloud ice) and any processes involving these. In the following subsections I shall describe in detail the condensation, evaporation, precipitation, autoconversion and accretion processes labelled here.

The work in [Rosemeier and Spichtinger, 2021] shows the potential for pattern formation in warm-rain clouds (i.e where the water is of entirely liquid or gaseous phase) and hence I omit explanations for any processes relating to the formation or precipitation of ice-based particles from this point onwards. For further information on modelling the formation of ice and snow within clouds please refer to either

of the following two large-scale models discussed later or see [Spreitzer et al., 2017] for a model of how cirrus clouds (clouds made entirely of ice crystals) form. Precipitation from so called warm-rain clouds is estimated to account for 30% of the rain across the planet ([Grabowski and Wang, 2013]) which demonstrates the relevance of this simpler model despite neglecting the ice and snow droplets.

Although there are many different ways to define cloud particles, and indeed different models use size, phase and fall speed to define different types of cloud variables, the two main particles considered in this dissertation are so called cloud droplets and rain droplets. This differentiation was originally proposed by [Kessler, 1969] and these particles are separated by their size. The smaller cloud droplets are typically less than $40\mu\text{m}$ in diameter [Rosemeier and Spichtinger, 2021] whilst the rain droplets are larger than this. These two key variables can be seen in many different cloud models ([ECMWF, 2017], [Rosemeier and Spichtinger, 2021], [Wacker, 1992], [Doms et al., 2011]) and are denoted as q_c and q_r respectively. These variables represent mass mixing ratios, i.e the mass of droplet species M_{q_i} to the mass of dry air M_d [Rosemeier et al., 2018]:

$$q_i = \frac{M_{q_i}}{M_d} \quad (14)$$

There are a multitude of processes occurring within clouds and it would be impossible to describe them all. However, there is a consensus between models of the key processes within clouds that affect the mass fractions of any particle. Specifically, the processes that I shall be discussing are condensation/evaporation, autoconversion, accretion, and precipitation/sedimentation.

In the following sections I shall explain the physics behind their processes and then compare how each of these processes is parameterised by different models. I include in this section how each of the processes were theoretically derived from various papers and how they are presented in two large-scale forecasting models. I have adapted the notation throughout so that like for like terms have the same notation to make it easier to compare between models.

The first of the large-scale operational models was developed by the European Center for Medium Range Weather Forecasts (ECMWF) and is called the Integrated Forecasting System (IFS). The warm-rain model contained within this model is adapted from [Khairoutdinov and Kogan, 2000] and can be found in [ECMWF, 2017].

The other operational model discussed here was developed by the Consortium for Small Scale Modelling (COSMO). This is a large scale model based on the well-established thermo-hydronamical equations which describe a moist planet and atmosphere and is used by the German Meteorological Service (Deutscher Wetterdienst, DWD). The model covers several countries including Germany, Austria and Switzerland and has over 9.7million grid points, each around 2.8km apart. Although this model has been combined with the Icosahedral Nonhydrostatic (ICON) model and is no longer operational, the warm-rain scheme remains relevant and is therefore still included here.

Both of these two models, the model presented in [Wacker, 1992], and the general cloud model in [Rosemeier and Spichtinger, 2021] are of the following form:

$$\frac{\partial q_c}{\partial t} = A_{q_c} + S_c - S_{au} - S_{ac}, \quad (15a)$$

$$\frac{\partial q_r}{\partial t} = A_{q_r} + P_r - S_{ev} + S_{au} + S_{ac}. \quad (15b)$$

The terms A_q correspond to processes such as advection which aren't directly related to the cloud microphysics. (These aren't addressed in [Rosemeier and Spichtinger, 2021] but are apparent in the other three models).

Each of the terms S_i represent source or sink terms for the cloud or rain droplets. They represent the following processes which will be explained thoroughly in the following subsections:

- S_c = the rate of evaporation and condensation of cloud droplets, q_c . This can be positive to represent condensation or negative to represent evaporation of q_c .
- S_{au} = the autoconversion rate. This represents the rate of increase in q_r through smaller cloud droplets combining to form a larger rain droplet or through the diffusion of water vapour.
- S_{ac} = the accretion rate. The accretion process is another collision process like autoconversion. Here, a large rain drop absorbs a smaller cloud droplet thus increasing the mass fraction of q_r ;

- P_r = the precipitation rate of the rain droplets.
- S_{ev} = the rate of evaporation of rain droplets q_r .

3.1 Condensation and Evaporation of Cloud Droplets

The warm-rain clouds being considered in this dissertation contain the two variables discussed above (q_c and q_r) and the remainder of the cloud fraction is made up of water vapour (sometimes denoted as q_v in models).

The first term that is considered in most cloud models is how small cloud droplets q_c are created and destroyed through condensation and evaporation. The cloud droplet change in size is driven by diffusion of the water vapour which is driven by the changes in saturation. The cloud droplets can also evaporate back into water vapour and hence this term can be either positive or negative depending on the environmental conditions.

[Rosemeier and Spichtinger, 2021]’s general cloud model can be seen throughout these processes to allow for a large degree of flexibility with parameters in order to find the assumptions that permit pattern formation. Hence, the simplest representation of the condensation and evaporation of cloud droplets is present in this model:

$$S_c = cq_c \quad (16)$$

This is a linear relationship with the cloud droplets and the constant c is dependant on the atmospheric supersaturation, temperature and pressure only. The work in [Rosemeier and Spichtinger, 2021] is largely based off the model presented in [Wacker, 1992] and hence the same expression for S_c is found there. [Wacker, 1992] also noted other potential sources of cloud droplets through other processes such as advection and entrainment.

The large scale models have a much more complex way to deal with the condensation and evaporation of cloud droplets although it is based on the same physics. Firstly, the COSMO model uses a ‘saturation adjustment technique’. This simply means that should a grid box in the model become supersaturated (i.e. there is more water vapour in a parcel of air than can be held at equilibrium) then the cloud droplets increase in size through condensation and the water vapour concentration goes back to its saturated state. Equally, a decrease in the grid box saturation is adjusted back up to the saturated state by evaporation of the cloud droplets. This means that the source term S_c is linearly represented as:

$$S_c = c(q_c - \tilde{q}_c). \quad (17)$$

In this representation c is based on the Clausius-Clapeyron equation and \tilde{q}_c is an estimate for q_c based on the budget equations for the hydrological cycle in a moist atmosphere (see [Doms et al., 2011]).

The IFS model considers the problem in a similar way. The term is separated into its condensation and evaporation parts.

Similarly, the condensation part of the term is treated by considering the change in saturated cloud mass fraction over time and thus can again be considered linear in q_c ($S_c = cq_c$). The evaporation part of the term is considered separately as the diffusion of the saturated cloud back into water vapour and hence is represented as:

$$S_{c(evaporation)} = -\phi q_c, \quad (18)$$

where ϕ is proportional to the diffusion coefficient (3×10^{-6}) and the saturation deficit of the air (similar to the COSMO case). Note that this term is negative as it results in a decrease in cloud droplets overall.

All of the models agree with the linearity of the condensation and evaporation term and the only differences are in the constant of proportionality. The coefficient c varies between the models and is considered a function of temperature and pressure in the COSMO and IFS models and a constant in [Wacker, 1992]. [Rosemeier and Spichtinger, 2021] considers c to be a dependant on atmospheric variables in their description of the model but when studying pattern formation considers it constant for simplicity.

3.2 Autoconversion

Autoconversion is the first of the collision processes that occur between cloud and rain droplets. In essence it is where two cloud droplets collide to form a larger rain droplet.

There is significant disparity between different models for expressions to parameterise this process though. One of the first expressions was presented in [Kessler, 1969] and defined a linear rate of increase of rain droplets through autoconversion as:

$$S_{au} = a_1(q_c - q_{c0})H(q_c - q_{c0}), \quad (19)$$

where a_1 , H and q_{c0} are a tuning constant, Heaviside step function and threshold water fraction respectively.

In the COSMO model, the autoconversion is similarly considered linear above the threshold value. The constant a_1 is considered to be the inverse of a time constant that determines the speed of this process (named τ^{-1} in the model's literature). [Wacker, 1992] also suggests a linear relationship of the form $a_1 q_c$.

The IFS model takes a very different approach to the other models and is based off the work done in [Khairoutdinov and Kogan, 2000]. Here, large scale data analysis was used to determine that the 'best-fit' autoconversion rate is:

$$S_{au} = a_1 q_c^{2.47} \quad (20)$$

where a_1 here is related to the cloud drop concentration. This shows a clear difference in the nonlinearity between the previous two models. There is however theoretical evidence to suggest that this exponent may be close to this. [Manton and Cotton, 1977] derived a theoretical form for S_{au} :

$$S_{au} = a_1 q_c^{\frac{7}{3}} H(r_{vc} - r_0). \quad (21)$$

Here a_1 is considered to be dependant on the mean droplet efficiency, Stokes constant and cloud drop concentration whilst $H(r_{vc} - r_0)$ is again a Heaviside step function but this time using the droplets mean volume radius and a threshold radius (r_{vc} and r_0 respectively).

There is clearly two different approaches between clouds models in the representation of the autoconversion rate.

The general model presented at the beginning of [Rosemeier and Spichtinger, 2021] cleverly allows for both of these scenarios by representing the autoconversion rate as

$$S_{au} = a_1 q_c^\gamma. \quad (22)$$

However, in the later analysis it is considered to be a linear relationship ($\gamma = 1$) to allow for simplicity in the stability analysis. I will aim to test various values of γ to see if pattern formation is possible under linear and nonlinear accretion.

3.3 Accretion

The second collision process that occurs within clouds is more complex than autoconversion and has a significant amount of variation in how it is represented within literature. Accretion is the process whereby cloud droplets are depleted by a falling rain drop which absorbs the smaller cloud droplets.

Similarly to autoconversion, the COSMO model shares a great deal of similarity with the work in [Kessler, 1969]. The derivation for the accretion rate is as follows.

The model assumes that rain droplets are exponentially distributed with respect to the drop diameter. Moreover, the increase in mass of the rain droplets due to accretion is assumed to obey the continuous collection equation where the mass (m) changes over time (t) according to:

$$\frac{dm}{dt} = \frac{4\pi}{3} \int E_c(R, r) \pi(R + r)^2 (v_T(R) - v_T(r)) f_c(r) r^3 dr, \quad (23)$$

where R and r are the radii of the large rain and small cloud droplets respectively, v_T the fall speed of the particles, E_c the collection efficiency, and f_c the number density distribution of the smaller cloud droplets. This is then integrated over all raindrops to give the following expression for S_{ac} .

$$S_{ac} \approx 1.72 q_c q_r^{7/8} \quad (24)$$

$$= a_2 q_c q_r^{7/8}, \quad (25)$$

where a_2 is a function of the variables described in Equation 23.

Alternatively, the IFS model follows a numerical pathway for determining S_{ac} , again based on [Khairoutdinov and Kogan, 2000]. The accretion rate is defined by [Khairoutdinov and Kogan, 2000] as ‘total mass increase per unit time... corresponding to collisions between drops smaller than [a threshold value]’. The experimental data is analysed using regression analysis to determine that the best fit for the accretion rate is:

$$S_{ac} = a_2 (q_c q_r)^{1.15}, \quad (26)$$

where a_2 was a constant determined to be equal to 67. This representation is what is used in the IFS model, however, [Khairoutdinov and Kogan, 2000] notes that a linear version could also be used ($S_{ac} = 3.7 q_c q_r$) although he shows that this is less accurate than the previous representation for his experimental data.

[Wacker, 1992] presented a formulation based on the Kessler-like parameterisation where the dependence on q_c was considered linear but here q_r has a flexible parameter β as the exponent. The model in [Rosemeier and Spichtinger, 2021] extends this flexibility by applying the non-linearity to both droplet species so that the accretion rate is defined as:

$$S_{ac} = a_2 q_c^{\beta_c} q_r^{\beta_r}. \quad (27)$$

In both research models a_2 is considered constant.

3.4 Evaporation of rain droplets

Both of the operational models consider a term that represents the loss of rain droplets through evaporation S_{ev} . Condensation is not considered in the rain droplet case as this process causes the creation of small cloud droplets before these are converted into the rain droplets through the collision processes. Evaporation, however, applies to droplets of any size.

The COSMO model considers this sink term for the rain droplets as:

$$S_{ev} = E q_r^{11/16}, \quad (28)$$

where E is dependant on the temperature and the density of the rain droplets.

This process is also considered in [Rosemeier et al., 2018] as:

$$S_{ev} = (e_1 q_r^{\delta_1} + e_2 q_r^{\delta_2}) \quad (29)$$

where e_i are related to the supersaturation S of the water vapour ($S = \frac{e_v}{e_s} - 1$, where e_v is the partial pressure of water vapour and e_s the saturation water vapour pressure.). However this process is negligible compared to the diffusional growth of smaller cloud droplets and is hence neglected in [Rosemeier and Spichtinger, 2021].

[Wacker, 1992] considers the condensation of the rain particles as a constant term ($S_{ev} = \Theta$).

3.5 Precipitation

When sufficient cloud droplets have coalesced together or combined with a raindrop then under gravity the rain drops can do as expected and fall to the ground.

Both [Wacker, 1992] and [Rosemeier and Spichtinger, 2021] consider the cloud to be one layer and thus consider a mass flux of rain from above (B) and q_r being depleted by gravitational sedimentation. In both papers this is represented by:

$$P_r = B - dq_r^\zeta \quad (30)$$

The operational models must consider the cloud as being several layers thick due to the vertical grid. The COSMO model calculates the precipitation flux as being:

$$P_r = v_T \rho q_r \quad (31)$$

Where v_T is the terminal fall speed of the droplets (approximated empirically as $\approx v_0 D^{1/2}$ where $v_0 = 130$ and D is the droplet diameter) and ρ is the density of the rain droplets. Hence, the COSMO model uses a linear approximation for the gravitational sedimentation of rain droplets.

Lastly, the IFS model has a different parameterisation of precipitation where the precipitation from a cloudy sky and a clear sky are considered separately. This cloud precipitation flux is then parameterised following [Jakob and Klein, 2000] which calculates the fraction of gridbox that is precipitating using a 'maximum-random overlap treatment of cloud fraction'.

The cloudy sky precipitation (rain from sedimentation of q_r) is calculated by integrating the precipitation flux over the proportion of each grid cell containing cloud. The precipitation rate is discussed in [Jakob and Klein, 2000] and has the form:

$$P_r = v_T \rho l_r \quad (32)$$

This formulation for the precipitation rate is almost identical to the form in the COSMO model except that q_r is replaced by l_r which is the specific humidity of q_r . As the specific humidity is a measure of the mass of a species within a unit mass of air [Office, nda] we can consider this proportional to q_r ($P_r = v_T \rho \alpha q_r$ using a constant of proportionality α).

Neither of the operational models directly contain a term for the rain flux from above (B) as they are multilayered models and thus the precipitation from above is determined by the conditions in the grid box directly above.

3.6 Comparison of Warm-Rain Cloud Models

As discussed in Section 1.1 statistical approaches are used when modelling clouds due to their complexity. Unfortunately, there is no generally accepted cloud scheme and thus many different schemes exist ([Rosemeier et al., 2018],[Rosemeier and Spichtinger, 2021]). All of these models show similarities in the way that they parameterise different micro-physical processes. However, they also differ in their complexity and assumptions.

Thus, in this subsection I will firstly compare the two operational cloud models and then compare these to the two research models in [Wacker, 1992] and [Rosemeier and Spichtinger, 2021].

Using the derivations for each of the source and sink terms described above, each of the models can be neatly surmised as follows. (The notation used here for the operational models differs slightly than that in their respective literature in order to make comparing them with the general cloud model easier.) Firstly, the COSMO model can be represented by the diagnostic equations:

$$\frac{\partial q_c}{\partial t} = A_{q_c} + c(q_c - \tilde{q}_c) - a_1 q_c - a_2 q_c q_r^{7/8}, \quad (33a)$$

$$\frac{\partial q_r}{\partial t} = A_{q_r} + a_1 q_c + a_2 q_c q_r^{7/8} - v_T \rho q_r. \quad (33b)$$

Secondly, the equivalent cloud scheme from the IFS model reads:

$$\frac{\partial q_c}{\partial t} = A_{q_c} + (c - \phi) q_c - a_1 q_c^{2.47} - a_2 q_c^{1.15} q_r^{1.15}, \quad (34a)$$

$$\frac{\partial q_r}{\partial t} = A_{q_r} + a_1 q_c^{2.47} + a_2 q_c^{1.15} q_r^{1.15} - v_T \rho \alpha q_r. \quad (34b)$$

Thirdly, the research scheme presented by Wacker is:

$$\frac{\partial q_c}{\partial t} = A_{q_c} + c q_c - a_1 q_c - a_2 q_c q_r^\beta, \quad (35a)$$

$$\frac{\partial q_r}{\partial t} = \Theta + a_1 q_c + a_2 q_c q_r^\beta + B - d q_r^\zeta. \quad (35b)$$

Finally, the general cloud scheme in [Rosemeier and Spichtinger, 2021] is:

$$\frac{\partial q_c}{\partial t} = c q_c - a_1 q_c^\gamma - a_2 q_c^{\beta_c} q_r^{\beta_r}, \quad (36a)$$

$$\frac{\partial q_r}{\partial t} = a_1 q_c^\gamma + a_2 q_c^{\beta_c} q_r^{\beta_r} + B - d q_r^\zeta. \quad (36b)$$

Observation of the four systems above shows the main differences arise in how each of the models parameterise the collision terms. Both models treat the precipitation flux as linear in q_r and proportional to the mean fall velocity of the falling droplet and the droplets density. The only difference in this term is that the IFS model considers the specific humidity of q_r and hence the fractional mass of the parcel made up of rain droplets is instead considered which results in a proportionality constant α . Although the research models allow for a nonlinear precipitation term through the parameter ζ , both papers assume $\zeta = 1$ for most of the analysis. Hence the precipitation term is effectively constant throughout the different systems.

There is only a slight difference in the parameterisation of the condensation-evaporation of q_c . The COSMO model uses a 'saturation adjustment technique' where the saturation of each parcel of air is adjusted back to the saturated state through condensation or evaporation over every time step. Hence the constant c in Equation 33a can be either positive or negative dependant on which process occurs at each timestep. The IFS instead considers two separate constants for condensation (c) and evaporation ($-\phi$) and hence both processes can occur simultaneously here. Both research models consider this term to be linear in q_c .

The main difference between the two operational models is the treatment of the accretion and autoconversion terms. The values of the exponents in the IFS model were derived based on the data analysis in [Khairoutdinov and Kogan, 2000] where a least squares method was applied to derive the values of the exponents. On the other hand, the values of the exponents in the COSMO models collision terms comes from the original derivations in [Kessler, 1969] with autoconversion being linearly dependant on q_c and the accretion term being derived from considering exponentially distributed droplets obeying the continuous collection equation.

Effectively, the terms in IFS model were experimentally derived whilst the COSMO models terms come from analytical derivations. This would suggest that in practicality, the IFS model is more accurate although it is naturally limited by the data that it is derived from.

The general cloud model (shown in Equation 36) is largely based on Wacker's model and has but a few differences. The flexibility of parameters is extended in the more recent model allowing nonlinearity in both cloud species in the accretion term. The only other differences between these two models are that [Rosemeier and Spichtinger, 2021] ignores non-microphysical processes such as advection and the diffusional growth of rain droplets.

As the research models are not part of a massive assimilation of different processes, temperature, pressure, and water vapour concentration (q_v) are considered constant in time. This is obviously not possible in the other operational models as temperature and pressure are two key variables in atmospheric modelling. The analytical derivation of the terms in the COSMO model mean that all of the coefficients in this model are functions of atmospheric parameters (with the exception of a_1). The coefficients in the other models are all considered constant, apart from c .

The research models consider the cloud as a single layer and thus represent rain from above through the term B . [Rosemeier et al., 2018] notes that B cannot be expressed with q_c and q_r as this would link neighbouring cells and take away from the focus of the research models, which is to understand the localised cloud behaviour. Rain from above is still considered in the operational models but simply comes from the cell above's behaviour (i.e. the value of $d q_r^\zeta$ in the cell above)

The operational cloud models (COSMO and IFS) use specific constants as to describe the non-linearity of the accretion and autoconversion terms but the advantage of this general model is the flexibility allowed in these exponents (β , γ and ζ). [Wacker, 1992] notes the 'striking similarity' between the equations in his paper (from which Equations 36 are derived) and those in chemical reactions. However two significant differences are noted. Firstly, the exponents discussed above are not limited to integers in the cloud models (as seen in both the COSMO and IFS schemes). This is not the case in chemical reactions where these exponents are solely integers. Secondly, in the warm-rain clouds that we are considering, both of the collision processes (autoconversion and accretion) are irreversible, unlike many of the reversible processes within chemical reactions. Nevertheless the similarities between this model and chemical reactions are considerable and likely the reason that pattern formation from these cloud models was conceived.

The final set of equations (36) is of the form shown in Section 2 (Equations 1) and thus the addition of diffusion terms could lead to the formation of structures. Furthermore, the work done by [Rosemeier and Spichtinger, 2021] shows that pattern formation through the Turing mechanism under

certain assumptions and simplifications is indeed possible. The flexibility provided by the various constants and exponents also means that studying pattern formation under different assumptions will be possible. For example, a reasonable extension of the work done in [Rosemeier and Spichtinger, 2021] is to discover if the values of the exponents used in the two operational models permit pattern formation (should diffusion terms be added).

4 Stability Analysis

The flexibility of the general cloud model (Equations 36) means that, under certain assumptions, analytical stability analysis can be applied to discover whether pattern formation is possible. This section follows, to a large extent, Section 6 of [Rosemeier and Spichtinger, 2021] with some additional explanations provided.

4.1 No rain from above (B=0)

The number of different exponents in Equations 36 means that stability analysis of these equations is incredibly complex. Therefore, [Rosemeier and Spichtinger, 2021] makes several reasonable simplifications in order to make analytical analysis feasible. Firstly, we consider the case of $\zeta = 1$, which results in a linear downward precipitation flux, bringing the model in line with both operational models. Secondly, as in the COSMO model, the autoconversion is also considered linear ($\gamma = 1$). Thirdly, the exponents in the accretion term are considered to be the same, $\beta_c = \beta_r$, as is the case in the IFS model. Finally, it is assumed that there is no rain from above (B=0) to give the following simplified set of equations:

$$\frac{\partial q_c}{\partial t} = cq_c - a_1 q_c - a_2 q_c^\beta q_r^\beta, \quad (37a)$$

$$\frac{\partial q_r}{\partial t} = a_1 q_c + a_2 q_c^\beta q_r^\beta - dq_r. \quad (37b)$$

Steady states occur when the time derivatives of both equations in 37 equal zero. From the definitions of the source and sink terms, all of the coefficients are positive with the exception of c . The presence of q_c or q_r in every term implies that a trivial steady state obviously exists when both q_c and q_r equal zero. This trivial equilibrium physically represents the physical state where there are no clouds and all of the water content in the air is falling as rain. However, the other non-trivial steady state can be found as shown below. Rearranging 37a gives:

$$cq_c - a_1 q_c - a_2 q_c^\beta q_r^\beta = 0, \quad (38)$$

$$a_2 q_c^\beta q_r^\beta = c - a_1. \quad (39)$$

Substituting this into 37b gives:

$$\begin{aligned} a_1 q_c + (c - a_1) q_c - dq_r &= 0, \\ dq_r &= cq_c, \\ q_{re} &= \frac{c}{d} q_{ce}. \end{aligned}$$

Now taking 38 and dividing through by q_c^β and rearranging, using logarithm laws:

$$\begin{aligned} \frac{c - a_1}{a_2} &= q_c^{\beta-1} q_r^\beta, \\ \ln \left(\frac{c - a_1}{a_2} \right) &= \ln (q_c^{\beta-1} q_r^\beta), \\ \ln \left(\frac{c - a_1}{a_2} \right) &= (\beta - 1) \ln q_c + \beta \ln q_r. \end{aligned}$$

Substituting the expression for q_{re} and again using logarithm laws gives:

$$\begin{aligned}
\ln \left(\frac{c - a_1}{a_2} \right) &= (\beta - 1) \ln q_c + \beta \ln \left(\frac{cq_c}{d} \right), \\
(2\beta - 1) \ln q_c &= \ln \left(\frac{c - a_1}{a_2} \right) - \ln \left(\frac{c^\beta}{d^\beta} \right), \\
\ln q_c &= \frac{1}{2\beta - 1} \left[\ln \left(\frac{c - a_1}{a_2} \right) \div \frac{d^\beta}{c^\beta} \right], \\
&= \ln \left(\frac{c^\beta}{d^\beta} \frac{c - a_1}{a_2} \right)^{\frac{1}{2\beta - 1}}, \\
q_c &= \left(\frac{d^\beta}{c^\beta} \frac{c - a_1}{a_2} \right)^{\frac{1}{2\beta - 1}}. \tag{40}
\end{aligned}$$

The non-trivial steady state thus occurs at:

$$(q_{ce}, q_{re}) = \left(\left(\frac{d^\beta}{c^\beta} \frac{c - a_1}{a_2} \right)^{\frac{1}{2\beta - 1}}, \frac{c}{d} q_c \right). \tag{41}$$

In order to guarantee a physically meaningful equilibrium both q_c and q_r must be positive and thus the first constraint is that $c > a_1$.

To calculate the stability of this equilibrium we evaluate the Jacobian of the system at the steady state (q_c, q_r) . The Jacobian for the system of equations 37 is:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial q_c} & \frac{\partial f_1}{\partial q_r} \\ \frac{\partial f_2}{\partial q_c} & \frac{\partial f_2}{\partial q_r} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} c - a_1 - a_2 \beta q_c^{\beta-1} q_r^\beta & -a_2 \beta q_c^\beta q_r^{\beta-1} \\ a_1 + a_2 \beta q_c^{\beta-1} q_r^\beta & a_2 \beta q_c^\beta q_r^{\beta-1} - d \end{bmatrix}.$$

Evaluating \mathbf{J} at the non-trivial steady state gives the following four entries. This can be evaluated as follows (the algebra behind the simplification of the $q_{ce}^{\beta-1} q_{re}^\beta$ and $q_{ce}^\beta q_{re}^{\beta-1}$ terms can be found in Appendix A):

$$\begin{aligned}
a_{11} &= c - a_1 - a_2 \beta \left(\frac{c - a_1}{a_2} \right), \\
&= c - a_1 - \beta(c - a_1), \\
&= (c - a_1)(1 - \beta).
\end{aligned}$$

$$\begin{aligned}
a_{12} &= -a_2 \beta \frac{d}{c} \left(\frac{c - a_1}{a_2} \right), \\
&= -\beta \frac{d}{c} (c - a_1).
\end{aligned}$$

$$\begin{aligned}
a_{21} &= a_1 + a_2 \beta \left(\frac{c - a_1}{a_2} \right), \\
&= a_1 + \beta(c - a_1).
\end{aligned}$$

$$\begin{aligned} a_{22} &= a_2 \beta \frac{d}{c} \left(\frac{c - a_1}{a_2} \right) - d, \\ &= d \left(\beta \frac{c - a_1}{c} - 1 \right). \end{aligned}$$

The four entries are thus:

$$a_{11} = (c - a_1)(1 - \beta) < 0, \quad (42)$$

$$a_{12} = -\beta \frac{d}{c} (c - a_1) < 0, \quad (43)$$

$$a_{21} = a_1 + \beta(c - a_1) > 0, \quad (44)$$

$$a_{22} = d \left(\beta \frac{c - a_1}{c} - 1 \right). \quad (45)$$

One of the corollaries of the conditions for Turing Instability is that a_{11} and a_{22} must have opposite signs and thus a_{22} must be positive. This occurs for $\beta \frac{c - a_1}{c} > 1$ which is equivalent to the condition: $\beta > 1$.

The first condition of the Turing model was that the system must be stable before the introduction of diffusion, and this is satisfied if the trace of the determinant is negative. The trace can be expressed based on the entries above as:

$$Tr(\mathbf{J}) = (c - a_1)(1 - \beta) + d \left(\beta \frac{c - a_1}{c} - 1 \right). \quad (46)$$

As the first part is guaranteed to be negative for $c > a_1$ and $\beta > 1$ and thus if d is sufficiently small then the first condition is satisfied.

The second condition stipulates that the determinant of the Jacobian be positive. This can be calculated as follows:

$$Det(\mathbf{J}) = (c - a_1)(1 - \beta)d \left(\beta \frac{c - a_1}{c} - 1 \right) - \beta \frac{d}{c}(c - a_1)(a_1 + \beta(c - a_1)) > 0, \quad (47)$$

$$= (c - a_1)d \left[(1 - \beta) \left(\frac{\beta}{c}(c - a_1) - 1 \right) + \frac{\beta}{c}(a_1 + \beta(c - a_1)) \right] > 0, \quad (48)$$

$$= (c - a_1)d \left[\frac{\beta}{c}(c - a_1) - \frac{\beta^2}{c}(c - a_1) - 1 + \beta + \frac{\beta a_1}{c} + \frac{\beta^2}{c}(c - a_1) \right] > 0, \quad (49)$$

$$= (c - a_1)d \left[\beta - \frac{\beta a_1}{c} - 1 + \beta + \frac{\beta a_1}{c} \right] > 0, \quad (50)$$

$$= (c - a_1)d(2\beta - 1) > 0. \quad (51)$$

Given the previous assumptions of $c > a_1$, $d > 0$ and $\beta > 1$, this inequality is always true and hence the second Turing condition is satisfied.

Finally, the third Turing condition can be satisfied for a suitable choice of D_1 and D_2 , with the general requirement that there is a significant order of magnitude between the two so that the negative entry of the Jacobian (a_{11}) is negligible compared to the positive one (a_{22}). This is a general feature of systems that display Turing patterns and was a reason that several decades passed before the Turing model was proven experimentally [Cross and Greenside, 2009].

4.2 $\mathbf{B} \neq 0$

The inclusion of the constant rain from above \mathbf{B} makes the analysis considerably more complex and is not considered as part of the analytical section of this dissertation. However, solving system 36 numerically whilst including the constant \mathbf{B} is more easily achieved and thus considered in the numerical section. Please see Section 6 of [Rosemeier and Spichtinger, 2021] for the analytical derivations of the Turing conditions for the specific case where $B \neq 0$ and $\beta = 2$.

5 Numerical Methodology

[Rosemeier and Spichtinger, 2021] uses a pseudo-spectral numerical method to solve the general cloud model and demonstrate pattern formation through the Turing mechanism. However, as discussed, the aim of this dissertation is to use the finite element numerical methods for the reasons outlined in Section 1.2.

In this section I will describe how finite element methods (FEMs) are used to solve partial differential equations generally and then how they are applied to solve the general cloud model.

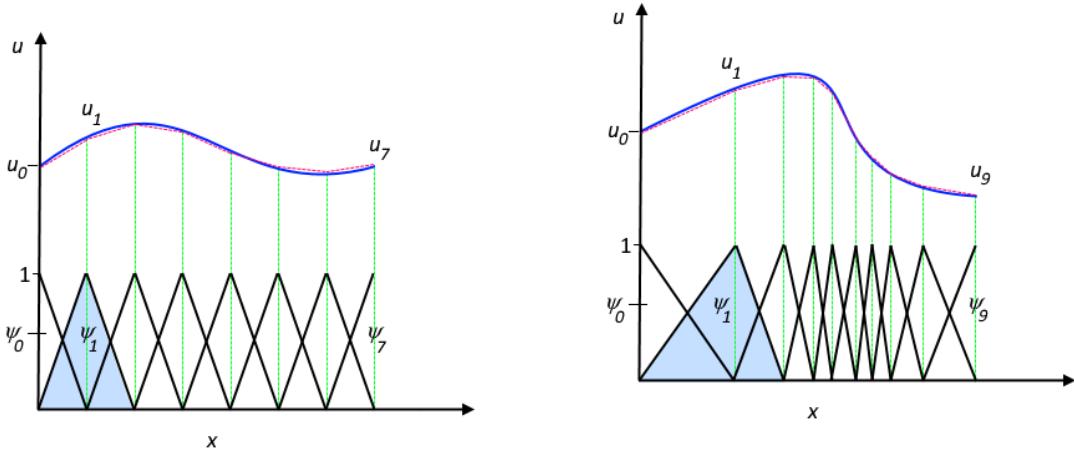
5.1 Finite Element Methods

Finite element methods were first discovered as a useful tool by Richard Courant in 1943 and are a popular way to approximate partial differential equations that are apparent in describing so many different fields of science. In essence the approximation works as follows:

A function is approximated by a sum of basis functions ψ_i (each with a particular coefficient u_i) that are defined on a grid that suits the particular geometry of the problem (i.e. a sphere for atmospheric modelling). Mathematically:

$$u \approx \sum u_i \psi_i. \quad (52)$$

The relevant basis function must also obey the boundary and initial conditions for the defined problem. Figure 6 demonstrates how this works graphically on both a uniform and a non-uniform grid.



(a) A finite element approximation using a uniformly distributed grid.

(b) A finite element approximation using a non-uniform grid. Note the difference in spacings between the vertical green lines.

Figure 6: The analytical solution u (blue line) is approximated (red dashed line) by a linear combination of basis functions ψ_i multiplied by coefficients u_i on a uniformly distributed mesh in subfigure (a) and on a non-uniform grid in subfigure (b) [COMSOL, 2017].

One of the key benefits gained from using FEM is the flexibility of the choice available when choosing the basis functions ψ_i . They can be linear, as seen in 6, or through a combination of higher order polynomials.

In order to apply this methodology to a physical set of equations (say with a solution \mathbf{q} on a domain Ω) the PDE and boundary conditions must be converted into their weak (or variational) formulation. This is obtained by multiplying the PDE by a test function v and integrating over the domain Ω . Formally, if the classical solution \mathbf{q} obeys an equation $L\mathbf{q} = \mathbf{f}$, where \mathbf{f} is a function and L a differential operator. Then the defect is $\mathbf{d}(\mathbf{v}) := L\mathbf{v} - \mathbf{f}$ and \mathbf{v} is the weak solution if $\langle \mathbf{d}(\mathbf{v}), \mathbf{w} \rangle = 0$ for all \mathbf{w} in an infinite dimensional linear Hilbert space [Iserles, 2008].

The advantage of expressing the equation in this form (and the reason behind its name) is that the requirement that each of the terms in the PDE must be well defined at all points is no longer required [COMSOL, 2017]. Therefore, any derivatives of order greater than one can be reduced through integration by parts. For example, a second order PDE must have a solution that can be differentiated at least twice (i.e. nonlinear) whilst the weak representation, once undergoing integration by parts, only requires the solution to be once differentiable [Harish, 2020].

Once the weak formulation is arrived at it can be discretised in space and time according to any number of schemes (e.g. forwards and backwards Euler, Crank-Nicolson or Galerkin methods). Note the difference here to the finite *difference* numerical methodology which discretises PDE's directly from their differential form.

5.2 FEM for the general cloud scheme

Given the conditions for the parameters calculated in the previous Section, the next step to solving this system is to apply the finite element methodology discussed above.

As shown in [Rosemeier and Spichtinger, 2021] and in 4, the simplified generic cloud model described in Equations 37 can create Turing patterns if certain conditions on the parameters are satisfied. Thus, we add a spatial Laplacian to each equation and then apply the finite element methodology.

$$\frac{\partial q_c}{\partial t} = cq_c - a_1 q_c - a_2 q_c^\beta q_r^\beta + D_1 \nabla^2 q_c, \quad (53a)$$

$$\frac{\partial q_r}{\partial t} = a_1 q_c + a_2 q_c^\beta q_r^\beta - dq_r + B + D_2 \nabla^2 q_r. \quad (53b)$$

The initial step is to convert these equations into their weak form. The solution \mathbf{q} is considered in a vector space V and thus to find the weak form we take the inner product of both equations with a test function v and integrate over the domain:

$$\int_{\Omega} v \frac{\partial q_c}{\partial t} dx = \sum_k \int_{e_k} v [cq_c - a_1 q_c - a_2 q_c^\beta q_r^\beta + D_1 \nabla^2 q_c] dx, \quad (54a)$$

$$\int_{\Omega} v \frac{\partial q_r}{\partial t} dx = \sum_k \int_{e_k} v [a_1 q_c + a_2 q_c^\beta q_r^\beta - dq_r + B + D_2 \nabla^2 q_r] dx. \quad (54b)$$

where e_k is the k^{th} element of the mesh. This returns the following formulation, for q_c first:

$$\int_{\Omega} \left[\frac{\partial q_c}{\partial t} v - (cq_c v) + (a_1 q_c v) + (a_2 q_c^\beta q_r^\beta v) - (D_1 v \nabla^2 q_c) \right] dx = 0 \quad (55)$$

The diffusion terms can be integrated by parts as follows:

$$\sum_k \int_{e_k} v \nabla^2 q_c dx = [v \nabla q_c]_{\partial e_k} - \int_{e_k} (\nabla q_c \cdot \nabla v) dx, \quad (56)$$

$$= 0 - \int_{e_k} (\nabla q_c \cdot \nabla v) dx. \quad (57)$$

The term evaluated at the boundaries is equal to zero as it is cancelled by the sum of basis elements [Le Dret and Lucquin, 2016]. This simplifies the weak form to the following:

$$\int_{\Omega} \left[\frac{\partial q_c}{\partial t} v - (cq_c v) + (a_1 q_c v) + (a_2 q_c^\beta q_r^\beta v) - (D_1 \nabla q_c \cdot \nabla v) \right] dx = 0. \quad (58)$$

Similarly for q_r :

$$\int_{\Omega} \left[\frac{\partial q_r}{\partial t} v - (a_1 q_c v) - (a_2 q_c^\beta q_r^\beta v) + (dq_r v) - (B v) - (D_2 \nabla q_r \cdot \nabla v) \right] dx = 0. \quad (59)$$

The next step is to discretise in time using the selected time stepping method. After using forwards Euler successfully for most of the initial models (e.g 1D Heat equation and Burgers equation) it was found that the forwards Euler was too slow as the time-step to maintain stability was very small.

Backwards Euler has the desired property of unconditional stability under any size timestep (dt) [Krishnamurti et al., 1998]. This allowed the model to be run for the required simulation period far quicker than in the forward Euler case.

The Backwards-Euler discretised versions of Equations 58 and 59 are simply:

$$\int_{\Omega} \left[\frac{q_c^{n+1} - q_c^n}{dt} \times v - cq_c^{n+1}v + a_1 q_c^{n+1}v + a_2 v(q_c^{n+1} q_r^{n+1})^\beta - D_1 \nabla q_c^{n+1} \cdot \nabla v \right] dx = 0, \quad (60a)$$

$$\int_{\Omega} \left[\frac{q_r^{n+1} - q_r^n}{dt} \times v - a_1 q_c^{n+1}v - a_2 v(q_c^{n+1} q_r^{n+1})^\beta + dq_r^{n+1}v - Bv - D_2 \nabla q_r^{n+1} \cdot \nabla v \right] dx = 0. \quad (60b)$$

The Turing model suggests that the addition of diffusion to a stable steady state causes instability and the formation of structures. Thus the initial condition for q_c and q_r are simply the non-trivial equilibrium calculated in 4. However, a small perturbation away from this is required to initialise the instability. Therefore, a normally distributed randomly generated perturbation of order 0.1 was added to the initial condition. This then allowed the evolution of q_c and q_r into patterns and structures discussed in the results (Section 6).

Furthermore, a domain length of 50 (in both x and y in the 2D case) was chosen for the numerical simulations to keep consistency with the results in [Rosemeier and Spichtinger, 2021]. The boundary conditions are periodic to maintain the required smoothness.

The code (CD) was written in Python and used the Firedrake system which provides accurate and highly efficient code to solve PDE's using FEM's [Rathgeber et al., 2017].

6 Results and Discussion

The model and numerical solver described above allow for a range of parameter values and various different assumptions to be applied to the model. Hence a great number of results for different parameter values were obtained. In this section I will present the results and findings under each of the different categories of assumptions made. I will then compare my results to those produced using the spectral method from [Rosemeier and Spichtinger, 2021] which can be seen in Appendix B.

6.1 $\beta = 2, B = 0$

The first set of results to be replicated are in one dimension and with $\beta = 2$ and no rain forcing from above ($B=0$). As the model under these assumptions is explicitly analysed theoretically in Section 4 (Section 6 of [Rosemeier and Spichtinger, 2021]) and proven to allow pattern formation, these are the key results to replicate.

Furthermore, this value is between the accretion and autoconversion exponents used in the IFS model (1.15 and 2.47 respectively), which suggests that in practicality 2 is not an unreasonable estimate for β .

6.1.1 1D case

In order to allow for an accurate comparison, the values of the constants (a_1, a_2, c, d and the diffusion constants D_1 and D_2) are taken from Section 7 of [Rosemeier and Spichtinger, 2021]. Respectively:

- $a_1 = 1;$
- $a_2 = 1;$
- $c = 5;$
- $d = 0.1;$
- $D_1 = 1000;$
- $D_2 = 0.1.$

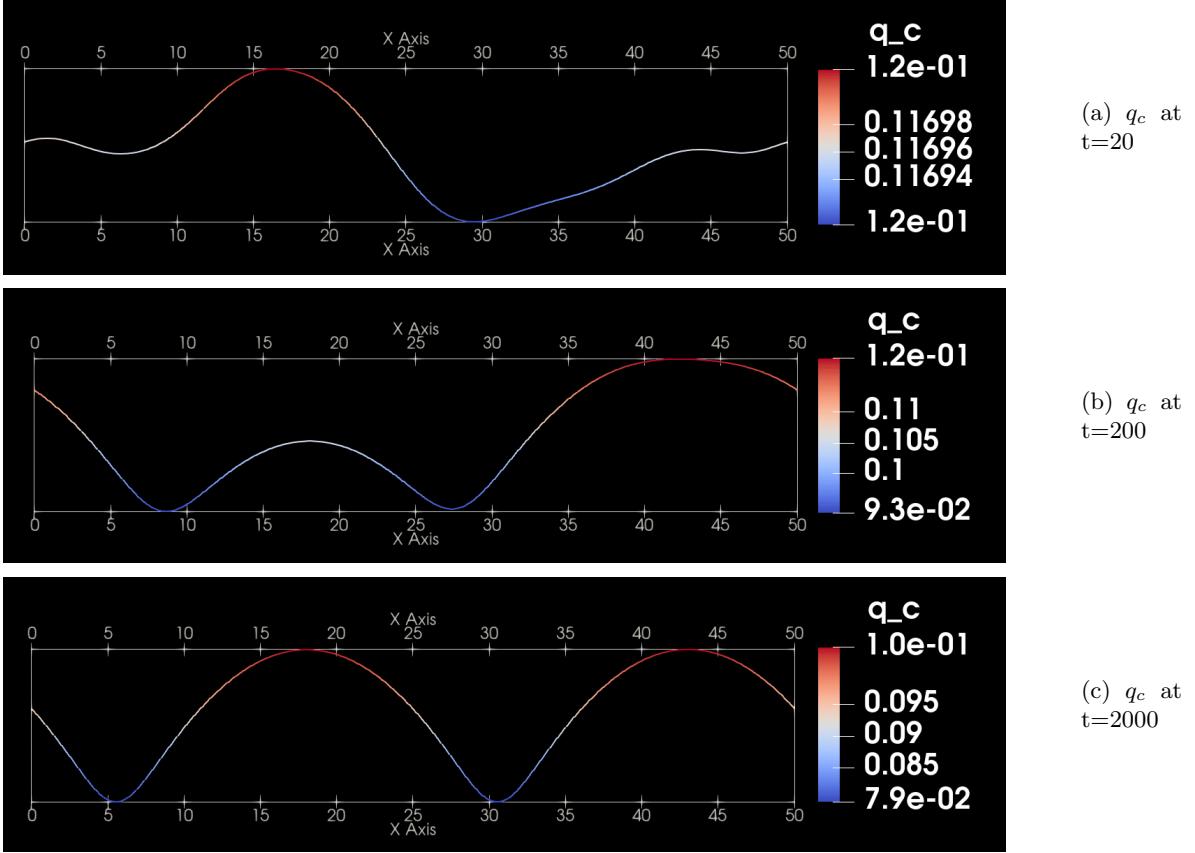


Figure 7: The variation of q_c in one dimensional space at times $t=20$, $t=200$, $t=2000$. (Note the different scales on the y axis). After 20 timesteps the value of q_c is barely changed from the equilibrium value. However, by 200 timesteps a wavelike pattern is becoming more apparent and a clear sinusoidal wave can be seen at 2000 timesteps.

By substituting these values into the derived conditions from Sections 2 and 4 we can see that the three conditions of the Turing model are satisfied ($\beta > 1$, $c > a_1$, D_1 and D_2 sufficiently different in magnitude and d sufficiently small).

The equilibrium state (q_{ce}, q_{re}) can be also be calculated using these values:

$$(q_{ce}, q_{re}) = \left(\left(\frac{d^\beta c - a_1}{c^\beta - a_2} \right)^{\frac{1}{2\beta-1}}, \frac{c}{d} q_c \right), \quad (61)$$

$$= \left(\left(\frac{0.1^2 5 - 1}{5^2 - 1} \right)^{\frac{1}{2\times 2-1}}, \frac{5}{0.1} q_c \right), \quad (62)$$

$$\approx (0.117, 5.848). \quad (63)$$

The results for both the cloud and rain droplets for three points in a simulation of 2000 timesteps are shown in Figures 7 and 8. The results returned what was expected under these parameters; the equilibrium state is restored after the small perturbation before structures grow around this steady state at certain wavenumbers.

These are almost identical to the results seen in [Rosemeier and Spichtinger, 2021] (see 13 in Appendix B). In both sets of results, it can be clearly seen that a wavelike pattern emerges in both q_c and q_r after around 200 timesteps. Furthermore, the distances between the peaks/troughs of these wavelike structures are the same in both simulations (≈ 25). By the end of the simulation the pattern is even more clear and remains in this form for further time.

There are two obvious differences between the two sets of results. Firstly, the graphs at $t=20$ show the largest amount of difference (compared to the other two timeframes shown). Secondly, (and

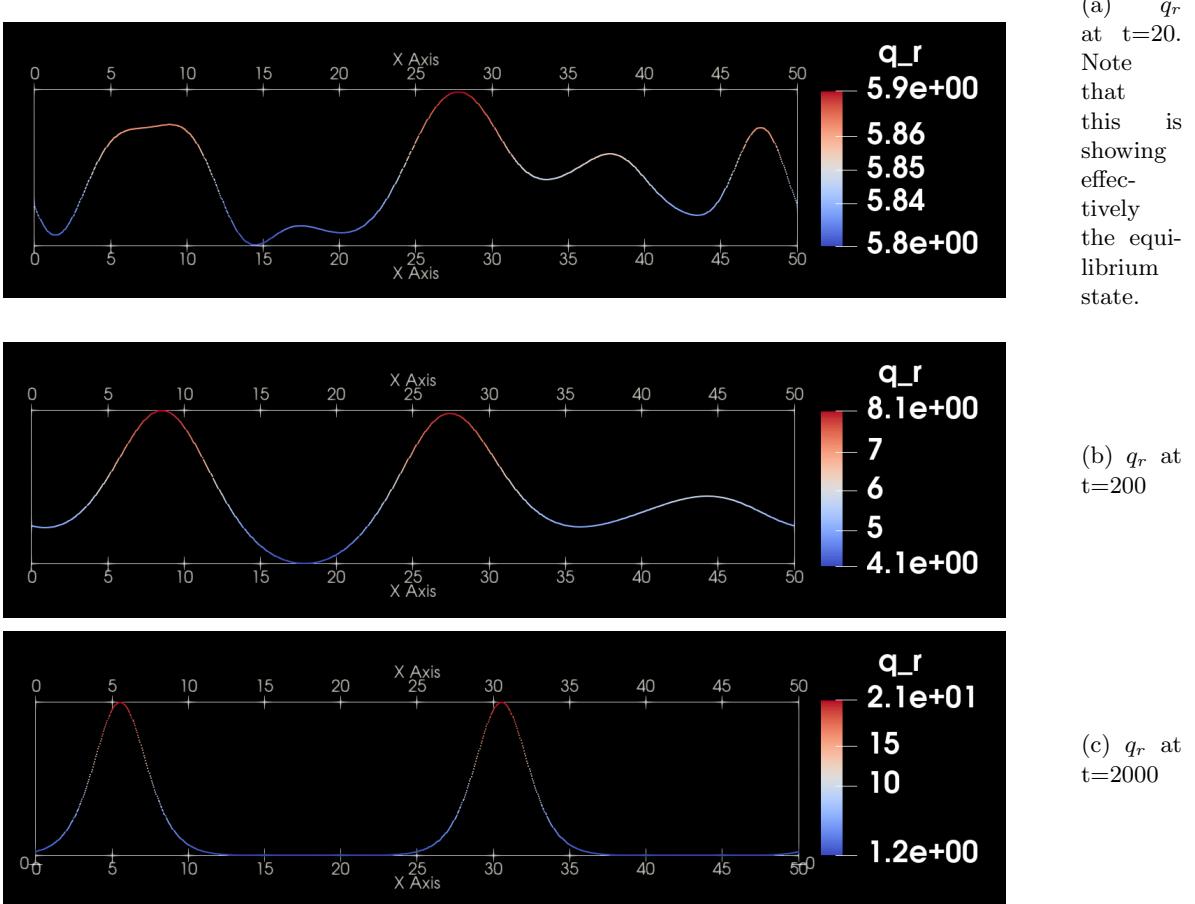


Figure 8: The variation of q_r in one dimensional space at times $t=20$, $t=200$, $t=2000$. (Note the different scales on the y axis). In a similar fashion to the cloud droplets q_c , the rain droplets show barely any variation away from the non-trivial equilibrium after 20 timesteps. Again, by $t=200$ there is a clearer wavelike pattern, which is fully realised by $t=2000$.

more interestingly) the location of the peaks and troughs are in different places. In Figures 7 and 8 the troughs of q_c and the peaks of q_r are seen at approximately 6 and 31. This is in contrast to the respective peaks and troughs from Figure 4 in [Rosemeier and Spichtinger, 2021] which appear around 22.5 and 47.5.

The large shift in location of the peaks is likely influenced by the randomness of the perturbation applied to the initial condition. This could have caused the wavenumbers, for which the growth rate was non-negative, to be different between both sets of results. The difference in the two sets of results at early points in the simulation could be because of this random perturbation also.

Both sets of results were produced using entirely different spatial and temporal discretisation's. The difference in which these methods approximate the true solution mean that a perfect replication of results would be impossible. The difference in numerical approximation is therefore likely to be the leading cause in any discrepancies between results.

Some simple models that permit Turing patterns allow for further analysis into the specific wavenumbers where pattern formation is possible. A further step to the work done here to explain this shift in structure location could be to apply these theories to the cloud scheme used. Due to the complexity of the model this may not be possible without further simplifications.

6.1.2 2D Case

The model was also extended into two dimensions and solved. Here, a square mesh of dimensions 50^2 was used with periodic boundary conditions. Some of the constants were also adjusted slightly to match those in the 2D case of [Rosemeier and Spichtinger, 2021]. The changed constants were:

- $d = 0.13$
- $D_1 = 100$
- $D_2 = 0.025$

All other constants remained the same. This changed the equilibrium point to $(0.139, 5.358)$. The numerical solver produced Figures 9 and 10 for the cloud and rain water for a simulation length of 120.

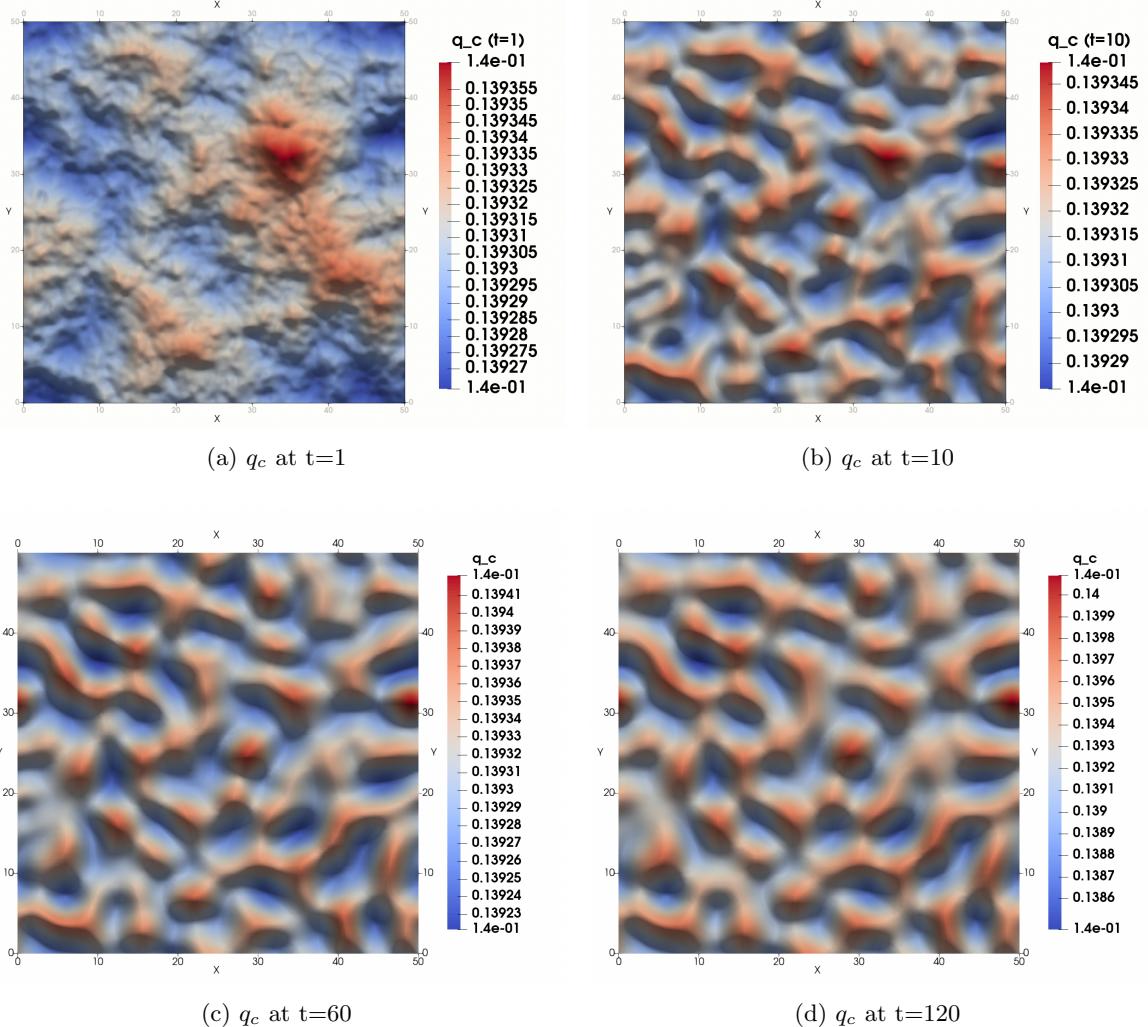


Figure 9: The distribution of cloud water q_c in two dimensional space at different times ($t=1/10/60/120$). Noting the different scales in the colour bars, one can see that by the end of the simulation patterns are beginning to emerge.

Similarly, to the one-dimensional case, the first few results effectively showed no change away from the equilibrium. However, as time progressed, again, structures began to appear, as predicted.

Whilst comparison between the two sets of results in 2 dimensions is naturally more difficult due to the increased complexity, a significant amount can still be gleaned from comparison of the two sets of results.

The largest difference between the two sets of results is the speed at which patterns are formed. [Rosemeier and Spichtinger, 2021] notes that in the two dimensional case, patterns are realised much sooner than in the one dimensional. This is not the case with the results seen here.

This difference can be seen quantitatively by looking at the range of values between the two sets of results at similar timeframes. At $t=120$ the results from my finite element solver showed that the range

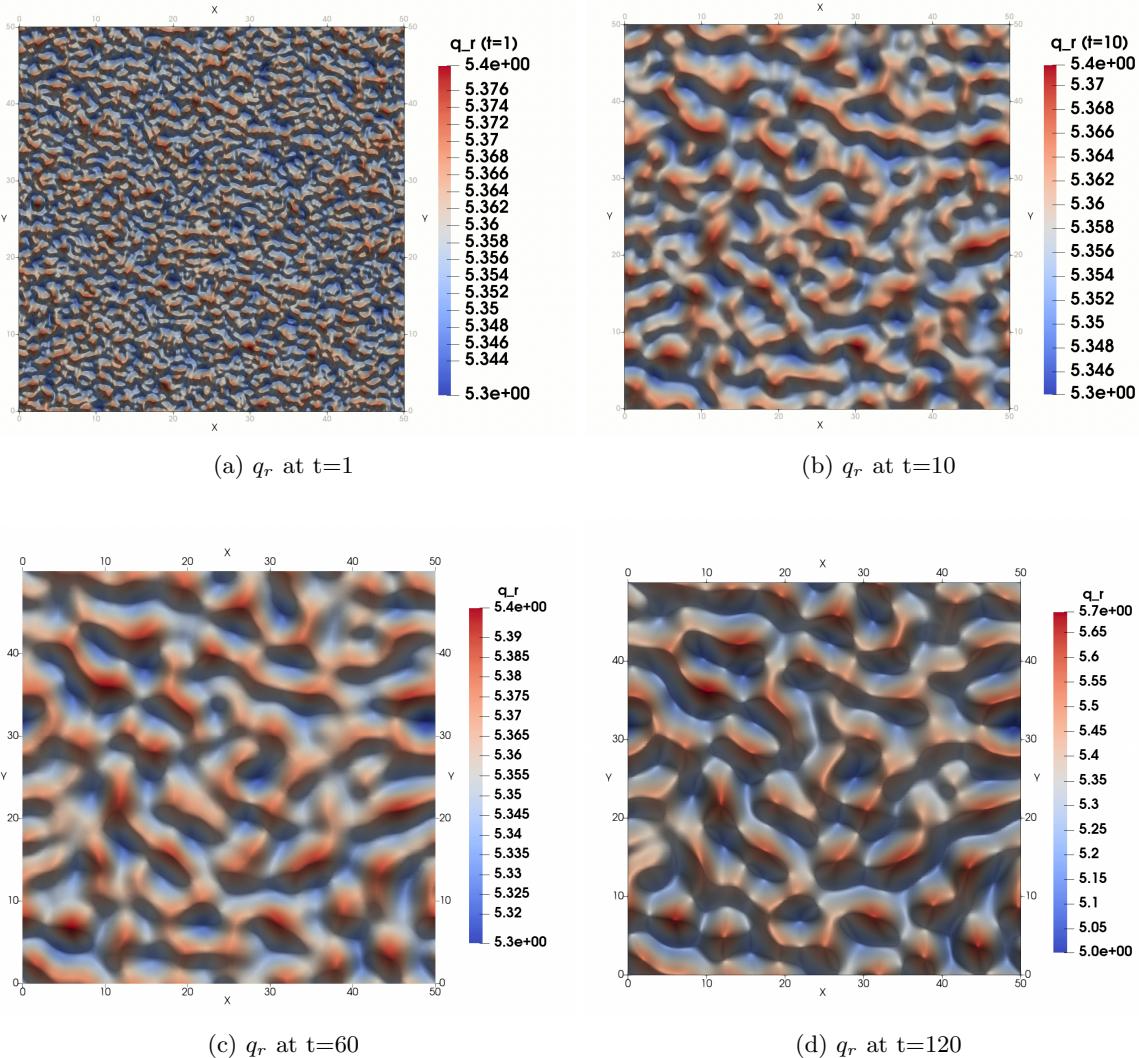


Figure 10: The distribution of rain water q_r in two dimensional space at different times ($t=1/10/60/120$). Here the cloud model goes from being very close to the equilibrium at $t=1$ to wavelike patterns becoming more visible at larger times.

between the largest and smallest values of q_c and q_r were 0.0015 and 0.69 respectively. On the other hand, the results in [Rosemeier and Spichtinger, 2021] showed a range of 0.005 and 7 respectively. This represents an order of magnitude of difference between the range of q_r while my range of q_c was over three times smaller.

The random perturbation (that was again applied in the two-dimensional case) may be a reason for differences in the patterns seen at early points. However, the difference in the speed of pattern formation is most likely because of the different timestepping scheme and spatial discretisation used. Potentially due to the same 'peak-shift' difference as seen in the one-dimensional case it is also difficult to see obvious structures appearing the same locations across both sets of results.

6.2 $\beta = 2, B \neq 0$

The conclusion in [Rosemeier and Spichtinger, 2021] regarding the introduction of the constant rain flux from above is that above a certain threshold value for B no pattern formation is possible. By including this term in the model and applying the numerical solver to it, the same outcome was reached. It can be clearly seen in Figure 11 that after 2000 timesteps with $B=0.1$, the results are very similar to those in the case with $B=0$ in Figures 7c and 8c.

When the model was solved using a value of above the threshold value (0.137) calculated in

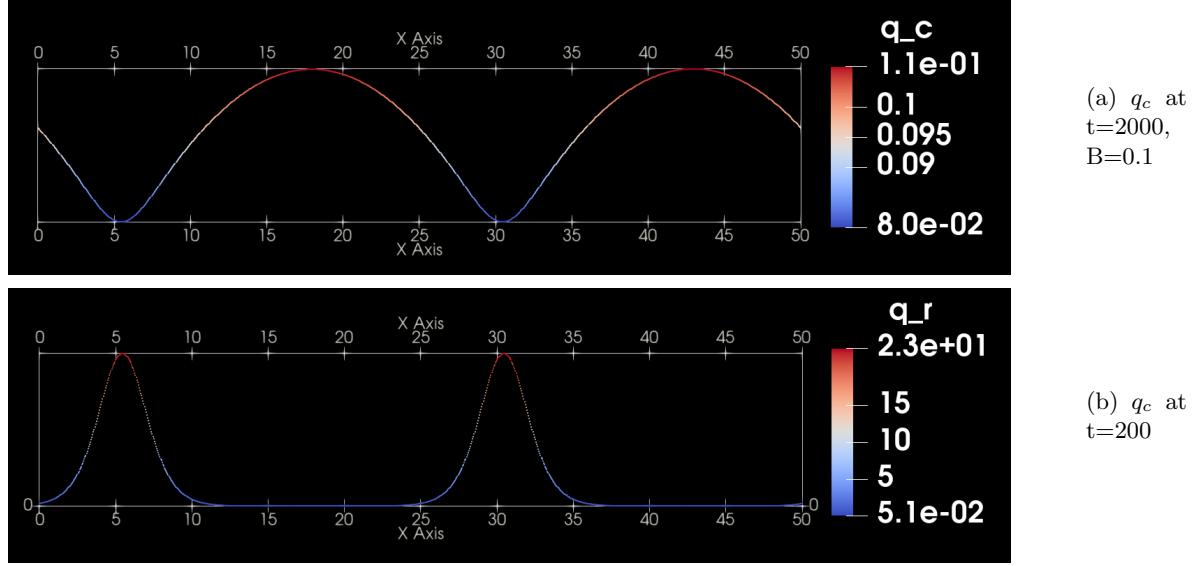


Figure 11: q_c for the case of $B=0.1$ at $t=2000$. Wavelike pattern formation can be seen in both species on a very similar scale to that seen in Figures 7c and 8c.

[Rosemeier and Spichtinger, 2021] then no pattern formation was seen.

6.3 $\beta = 3$, $B=0$

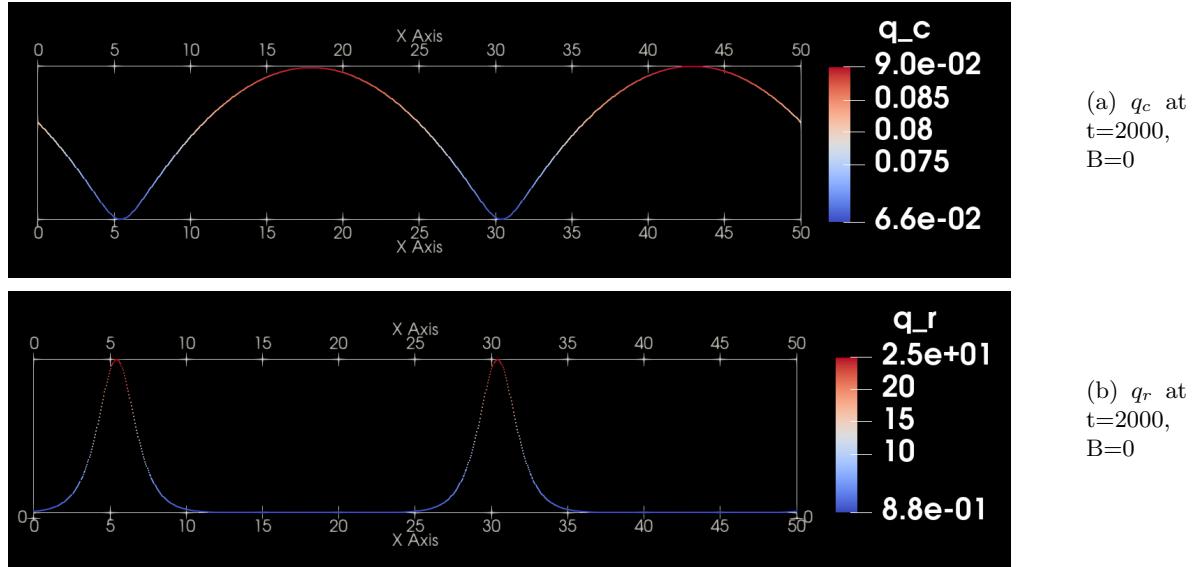


Figure 12: When the value of β is increased to 3 wavelike pattern formation is still apparent in both cloud species.

When the value of β was increased to 3 very little change was observed. The locations of the peaks and troughs remained the same, as did the speed of pattern formation. In fact, the only noticeable difference was in the values of the two cloud species, with the range of q_r becoming significantly smaller.

6.4 Using the IFS exponents

The variation of the exponent between the accretion and autoconversion terms in the equations is something that is very clear in 'real-world' models such as the COSMO and IFS models. However,

in the cases considered in [Rosemeier and Spichtinger, 2021] the exponent is the same for both terms. Hence a relevant extension to the work done in [Rosemeier and Spichtinger, 2021] would be to see if pattern formation was possible under different (non-integer) exponents for the two collision terms.

The value of the accretion exponent in the IFS model discussed in Section 3.6 is 1.15 and the value of the autoconversion exponent is 2.47. Therefore, in order to understand whether pattern formation would be possible using some of the constants already in use in applied forecasting models I changed the exponents in the general cloud model (Equations 53) to integers closest to these values. Unfortunately, the numerical solver was unable to solve the model with non-integer values for the exponents so rounding to the nearest integer was the closest approximation available for this. I also considered $B=0$ for simplicity here.

$$\frac{\partial q_c}{\partial t} = cq_c - a_1 q_c^2 - a_2 q_c q_r + D_1 \nabla^2 q_c, \quad (64a)$$

$$\frac{\partial q_r}{\partial t} = a_1 q_c^2 + a_2 q_c q_r - dq_r + D_2 \nabla^2 q_r.. \quad (64b)$$

Although the model successfully solved this adapted model, no patterns were observed over the simulation period. More analytical/numerical stability analysis for a model of this form would be required to see if the Turing conditions could be met under restrictions on the coefficients.

7 Conclusion

Clouds remain one of the largest sources of uncertainty in atmospheric modelling and one of the reasons is how differently structured clouds interact with radiation from the Sun and Earth. With no first principle set of equations available to represent cloud microphysics, many different schemes exist, including bulk models that analyse the mass fraction of different particles within the clouds. The work done in [Rosemeier and Spichtinger, 2021] shows, both analytically and numerically, that the addition of spatial Laplacian terms to one moment bulk cloud schemes similar to those used in certain operational models could produce pattern formation between two of the droplet species seen in warm-rain clouds. The results show pattern formation in both one and two dimensions if the derived conditions on the model's parameters are met. It is also shown that rainflux from above if greater than a threshold value destroys the potential for pattern formation.

The aim of this dissertation was to recreate the results produced in [Rosemeier and Spichtinger, 2021] using a finite element numerical methodology. There was also scope within the aims to explore different constants in the cloud model to see whether operational models' parameterisation of cloud droplet collision terms would allow for pattern formation if diffusion terms were added. Here I will discuss if the results produced by my numerical solver satisfied the aims and what further work could be undertaken.

The first results in [Rosemeier and Spichtinger, 2021] shows how, once perturbed away from the equilibrium, both the cloud and rain droplets form wavelike patterns over a simulation period of 2000. The results produced when solving the same cloud model with the same parameters, using finite element methods gave an almost exactly the same results as those from the spectral method. The scale of the patterns, in both cloud droplets and rain droplets, were the same, as was the number of structures that appeared and the distance between these structures. The only difference when comparing these two sets of results was in the location of the structures, with the one dimensional results in this dissertation appearing translated along the x axis.

The second set of results that I aimed to replicate were the two-dimensional patterns of the cloud model, using the same form of the cloud scheme as in the one-dimensional case (with slightly different constants). Whilst the finite element solver successfully solved this model my results showed several differences to the results in [Rosemeier and Spichtinger, 2021]. Although by the end of the simulation ($t=120$) my results showed some structural growth, the scale of the patterns was significantly different to the spectral results at the same point in time. My results showed that the size of the patterns away from the equilibrium was even smaller in scale to the one-dimensional case at the same point in time. To see this clearly see the one-dimensional results for the pattern growth of q_c and q_r at $t=120$ in appendix E. On the other hand, the two-dimensional results in [Rosemeier and Spichtinger, 2021] sees patterns grow much faster and by $t=120$ the patterns are at a similar scale to those at $t=2000$ in

the one-dimensional case. The entirely different spatial and temporal discretisation used is likely the reason for this difference between the results.

The final results presented in [Rosemeier and Spichtinger, 2021] show how rain flux from above influences pattern formation. I successfully used my numerical solver to corroborate that pattern formation is possible if the rain flux is below a threshold value.

The variety in the parameterisations of different collision terms between models led me to explore whether pattern formation was possible using different values of the exponents. Specifically, I changed the model's parameters to be like that of the IFS model and discovered that, using integers closest to the exponents in this model, pattern formation did not occur.

One of the drawbacks to the numerical solver I used was that it failed to successfully solve the equations if the values of the exponents were non integers. As both of the operational models discussed in the literature review of different cloud models in Section 3.6 contain terms that have a non-integer exponent, a useful extension of this dissertation would be to adapt the numerical solver to allow for non-integer exponents. This would allow the exact form of the operational models to be inputted in the numerical solver and the structures that emerged over time (if indeed they allow this) to be studied and compared to one another.

In conclusion, this dissertation was successful in recreating the results of pattern formation shown in [Rosemeier and Spichtinger, 2021]. I was particularly successful in replicating the one-dimensional results and in the (one-dimensional) case with rain flux from above. There were some large discrepancies between the two-dimensional results, however both works showed structure formation away from the equilibrium. The differences between the two sets of results were attributed to either the different numerical methods used, or the randomness of the perturbation applied to the initial condition.

I would like to thank Dr Jemma Shipton for her tireless help in understanding and applying finite element methods to the cloud model. I would also like to thank Dr Juliane Rosemeier for her useful explanations of cloud microphysical processes.

References

- [Ball, 2015] Ball, P. (2015). Forging patterns and making waves from biology to geology: A commentary on turing (1952) ‘the chemical basis of morphogenesis’. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 370.
- [Borde and Isaka, 1996] Borde, R. and Isaka, H. (1996). Radiative transfer in multifractal clouds. *Journal of Geophysical Research: Atmospheres*, 101(D23):29461–29478.
- [Caendkoelsch, 2018] Caendkoelsch (2018). What is the difference between finite element method (fem), finite volume method (fvm) and finite difference method (fdm) ? <https://caendkoelsch.wordpress.com/2017/11/04/the-difference-between-fem-fdm-and-fvm/>.
- [Ceppi and Williams, 2022] Ceppi, P. and Williams, R. (2022). Why clouds are the missing piece in the climate change puzzle. <https://theconversation.com/why-clouds-are-the-missing-piece-in-the-climate-change-puzzle-140812>.
- [Collins et al., 2013] Collins, S. N., James, R. S., Ray, P., Chen, K., Lassman, A., and Brownlee, J. (2013). Grids in numerical weather and climate models. In Zhang, Y. and Ray, P., editors, *Climate Change and Regional/Local Responses*, chapter 4. IntechOpen, Rijeka.
- [COMSOL, 2017] COMSOL (2017). The finite element method (fem). <https://www.comsol.com/multiphysics/finite-element-method>.
- [Cross and Greenside, 2009] Cross, M. and Greenside, H. (2009). *Linear instability: applications*, page 96–125. Cambridge University Press.
- [Doms et al., 2011] Doms, G., Förstner, J., HEISE, E., Herzog, H.-J., Mironov, D., Raschendorfer, M., Reinhardt, T., Ritter, B., Schrödin, R., Schulz, J.-P., and et al. (2011). *Part II, Physical Parameterization*, chapter II. DWD.
- [ECMWF, 2017] ECMWF (2017). *IFS Documentation CY43R3 - Part IV: Physical processes*, chapter 7, pages 99–121. Number 4 in IFS Documentation. ECMWF.

- [Grabowski et al., 2019] Grabowski, W. W., Morrison, H., Shima, S.-I., Abade, G. C., Dziekan, P., and Pawlowska, H. (2019). Modeling of cloud microphysics: Can we do better? *Bulletin of the American Meteorological Society*, 100(4).
- [Grabowski and Wang, 2013] Grabowski, W. W. and Wang, L.-P. (2013). Growth of cloud droplets in a turbulent environment. *Annual Review of Fluid Mechanics*, 45(1):293–324.
- [Harish, 2020] Harish, A. (2020). Finite element method - what is it? fem and fea explained. <https://www.simscale.com/blog/what-is-finite-element-method/>.
- [Iserles, 2008] Iserles, A. (2008). *The finite element method*, page 171–204. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2 edition.
- [Jakob and Klein, 2000] Jakob, C. and Klein, S. A. (2000). A parametrization of the effects of cloud and precipitation overlap for use in general-circulation models. *Quarterly Journal of the Royal Meteorological Society*, 126(568):2525–2544.
- [Kessler, 1969] Kessler, E. (1969). *On the Distribution and Continuity of Water Substance in Atmospheric Circulations*, pages 1–84. American Meteorological Society, Boston, MA.
- [Khairoutdinov and Kogan, 2000] Khairoutdinov, M. and Kogan, Y. (2000). A new cloud physics parameterization in a large-eddy simulation model of marine stratocumulus. *Monthly Weather Review*, 128(1):229 – 243.
- [Khouider et al., 2019] Khouider, B., Klein, R., and Smith, L. M. (2019). Moist processes in the atmosphere. *Oberwolfach*, 16(1):465–512.
- [Kondo and Miura, 2010] Kondo, S. and Miura, T. (2010). Reaction-diffusion model as a framework for understanding biological pattern formation. *Science*, 329(5999):1616–1620.
- [Krishnamurti et al., 1998] Krishnamurti, T. N., Bedi, H. S., and Hardiker, V. M. (1998). Time-Differencing Schemes. In *An Introduction to Global Spectral Modeling*. Oxford University Press.
- [Le Dret and Lucquin, 2016] Le Dret, H. and Lucquin, B. (2016). *The Finite Element Method in Dimension Two*, pages 167–218. Springer International Publishing, Cham.
- [LeVeque, 2007] LeVeque, R. J. (2007). Finite difference methods for ordinary and partial differential equations: Steady-state and time-dependent problems.
- [Manton and Cotton, 1977] Manton, M. and Cotton, W. (1977). *Formulation of Approximate Equations for Modeling Moist Deep Convection on the Mesoscale*. Atmospheric science paper. Department of Atmospheric Science, Colorado State University.
- [MetOffice, 2021] MetOffice (2021). The next-generation modelling system reaches a new milestone. <https://www.metoffice.gov.uk/research/news/2021/gungho-and-lfric-10th-anniversary>.
- [MetOffice, nd] MetOffice (n.d.). Lfric - a modelling system fit for future computers. <https://www.metoffice.gov.uk/research/approach/modelling-systems/lfric>.
- [Office, nda] Office, M. (n.d.a). Humidity. <https://www.metoffice.gov.uk/weather/learn-about/weather/types-of-weather/humidity>.
- [Office, ndb] Office, M. (n.d.b). Stratocumulus clouds. <https://www.metoffice.gov.uk/weather/learn-about/weather/types-of-weather/clouds/low-level-clouds/stratocumulus>.
- [Peiro and Sherwin, 2005] Peiro, J. and Sherwin, S. (2005). *Finite Difference, Finite Element and Finite Volume Methods for Partial Differential Equations*, pages 2415–2446. Springer, Dordrecht.
- [Perkins, 2015] Perkins, S. (2015). Mystery solved: How these rocks got their strange hexagonal shape. <https://www.science.org/content/article/mystery-solved-how-these-rocks-got-their-strange-hexagonal-shape>.

- [Rathgeber et al., 2017] Rathgeber, F., Ham, D. A., Mitchell, L., Lange, M., Luporini, F., Mcrae, A. T. T., Bercea, G.-T., Markall, G. R., and Kelly, P. H. J. (2017). Firedrake. *ACM Transactions on Mathematical Software*, 43(3):1–27.
- [Rosemeier et al., 2018] Rosemeier, J., Baumgartner, M., and Spichtinger, P. (2018). Intercomparison of warm-rain bulk microphysics schemes using asymptotics. *Mathematics of Climate and Weather Forecasting*, 4:104–124.
- [Rosemeier and Spichtinger, 2021] Rosemeier, J. and Spichtinger, P. (2021). Pattern formation in clouds via turing instabilities.
- [Spreitzer et al., 2017] Spreitzer, E., Marschalik, M., and Spichtinger, P. (2017). Subvisible cirrus clouds – a dynamical system approach. *Nonlinear Processes in Geophysics*, 24:307–328.
- [Taylor, 2002] Taylor, A. F. (2002). Mechanism and phenomenology of an oscillating chemical reaction. *Progress in Reaction Kinetics and Mechanism*, 27(4):247–326.
- [Turing, 1952] Turing, A. M. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 237(641):37–72.
- [UCAR, 2019] UCAR (2019). Center for science education. <https://scied.ucar.edu/learning-zone/clouds/how-clouds-form>.
- [Wacker, 1992] Wacker, U. (1992). Structural stability in cloud physics using parameterized micro-physics. *Beitrage zur Physik der Atmosphare*, pages 231–242.
- [Williamson and Laprise, 2000] Williamson, D. and Laprise, R. (2000). *Numerical Approximation for Global Atmospheric General Circulation Models*, pages 127–219.
- [Yates, 2022] Yates, P. (2022). Undulatus clouds look like wavy rows. <https://earthsky.org/earth/undulatus-clouds-wavy-rows/>.
- [Zhang et al., 2018] Zhang, F., Yan, J.-R., Li, J., Wu, K., Iwabuchi, H., and Shi, Y.-N. (2018). A new radiative transfer method for solar radiation in a vertically internally inhomogeneous medium. *Journal of the Atmospheric Sciences*, 75(1):41 – 55.

A Evaluating the Jacobian at the non-trivial equilibrium

The non-trivial steady state is given by:

$$(q_{ce}, q_{re}) = \left(\left(\frac{d^\beta c - a_1}{c^\beta a_2} \right)^{\frac{1}{2\beta-1}}, \frac{c}{d} \left(\frac{d^\beta c - a_1}{c^\beta a_2} \right)^{\frac{1}{2\beta-1}} \right) \quad (65)$$

The two terms required to evaluate and simplify the Jacobian are: $q_{ce}^{\beta-1} q_{re}^\beta$ and $q_{ce}^\beta q_{re}^{\beta-1}$.

The first can be calculated as follows:

$$\begin{aligned} q_{ce}^{\beta-1} q_{re}^\beta &= \left(\frac{d^\beta c - a_1}{c^\beta a_2} \right)^{\frac{\beta-1}{2\beta-1}} \times \frac{c^\beta}{d^\beta} \left(\frac{d^\beta c - a_1}{c^\beta a_2} \right)^{\frac{\beta}{2\beta-1}} \\ &= \frac{c^\beta}{d^\beta} \left(\frac{d^\beta c - a_1}{c^\beta a_2} \right)^{\frac{\beta+\beta-1}{2\beta-1}} \\ &= \frac{c^\beta}{d^\beta} \left(\frac{d^\beta c - a_1}{c^\beta a_2} \right) \\ &= \frac{c - a_1}{a_2} \end{aligned}$$

Similarly:

$$\begin{aligned}
q_{ce}^\beta q_{re}^{\beta-1} &= \left(\frac{d^\beta}{c^\beta} \frac{c - a_1}{a_2} \right)^{\frac{\beta}{2\beta-1}} \times \frac{c^{\beta-1}}{d^{\beta-1}} \left(\frac{d^\beta}{c^\beta} \frac{c - a_1}{a_2} \right)^{\frac{\beta-1}{2\beta-1}} \\
&= \frac{c^{\beta-1}}{d^{\beta-1}} \left(\frac{d^\beta}{c^\beta} \frac{c - a_1}{a_2} \right)^{\frac{\beta+\beta-1}{2\beta-1}} \\
&= \frac{dc^\beta}{cd^\beta} \left(\frac{d^\beta}{c^\beta} \frac{c - a_1}{a_2} \right) \\
&= \frac{d}{c} \frac{c - a_1}{a_2}
\end{aligned}$$

These two results are then substituted into the expressions in the Jacobian to give the four entries provided in Section 4.

B Results from the spectral numerical method

The results from [Rosemeier and Spichtinger, 2021] for the generic cloud model (Equations 36) when solved numerically using a spectral method of exponential integrators. The constants for these results are the same as those defined in 6 for both one and two dimensions. Also, note the graphs depict the results after the same simulation times and are thus directly comparable.

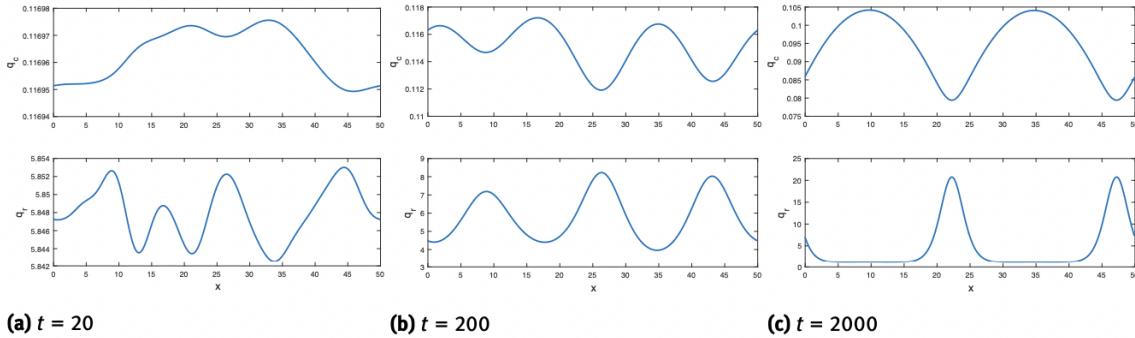
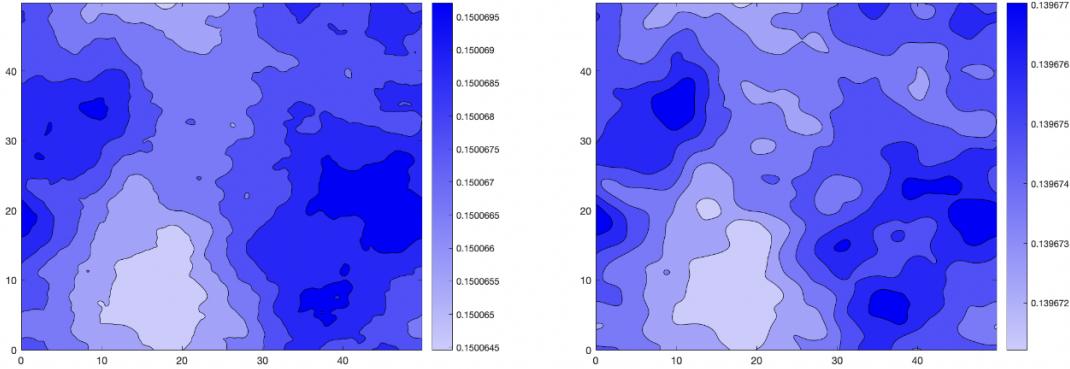
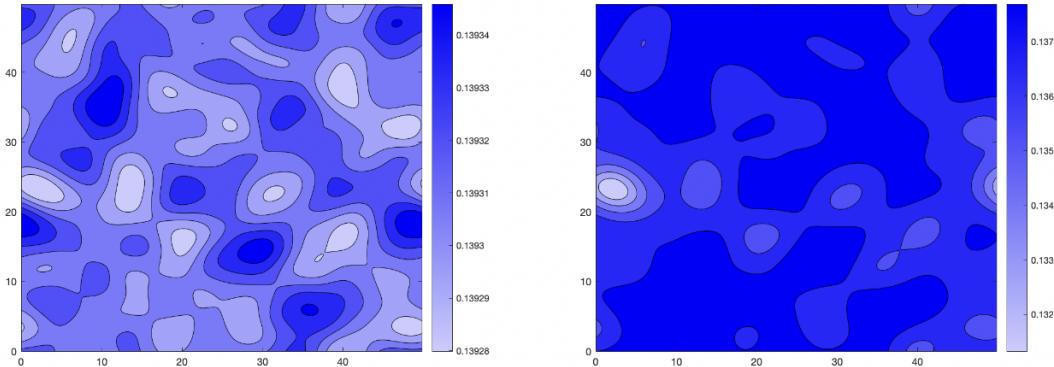


Figure 13: Figure 4 in [Rosemeier and Spichtinger, 2021]. This shows how structures appear in both the cloud and rain droplets over time. This figure is directly comparable with Figures 7 and 8



(a) $t = 1$

(b) $t = 10$



(c) $t = 60$

(d) $t = 120$

Figure 14: Figure 5 in [Rosemeier and Spichtinger, 2021] showing the spatial distribution of the cloud droplets. Patterns can be seen to emerge after a shorter period than in the one dimensional case. This figure is directly comparable with Figure 9.

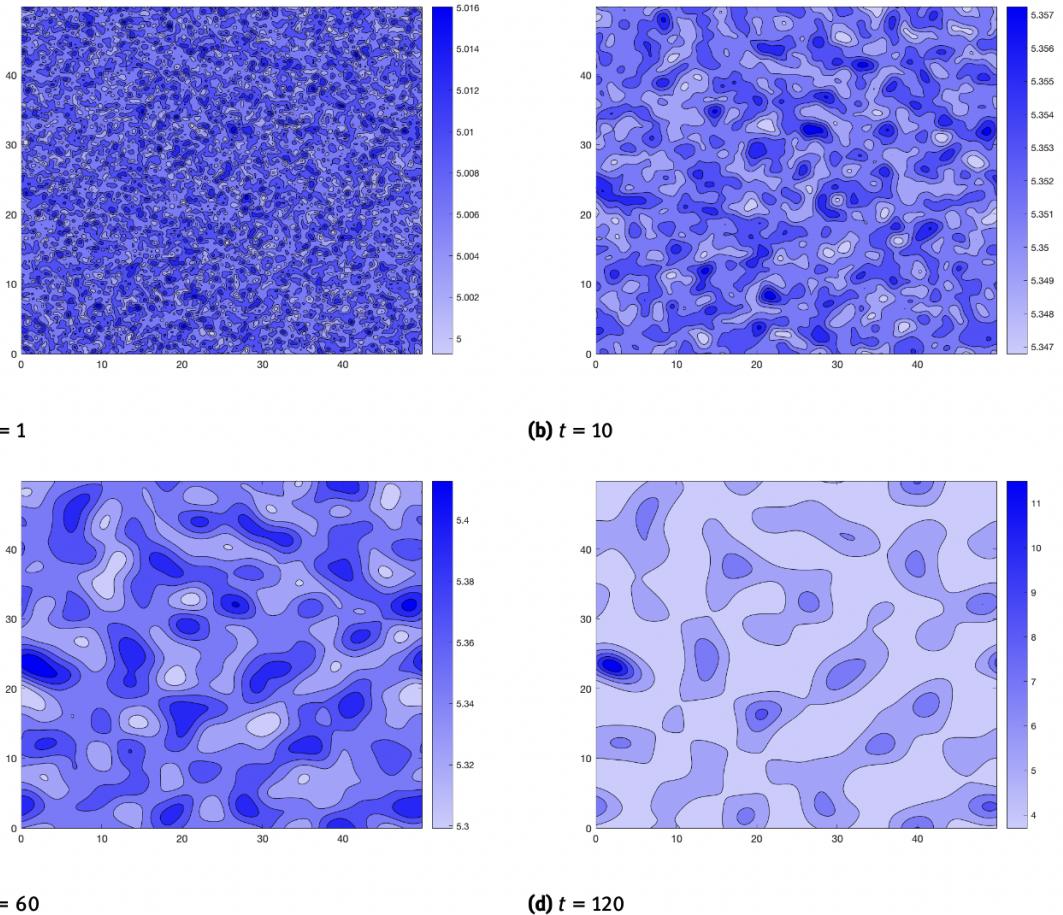


Figure 15: Figure 6 in [Rosemeier and Spichtinger, 2021] showing the spatial distribution of rain droplets. Similarly to the cloud droplets, patterns are seen much sooner than in the one dimensional case. This figure is directly comparable with Figure 10

C 1D Code

The code used to solve the cloud scheme in one dimension and produce the one dimensional results seen in Figures 7, 8, 12 and 11. The parameters were adjusted at the begining of the code for each of the different subsections of 6. Note that the equilibrium calculated is only correct for the case of B=0.

```
from firedrake import *
import math
import matplotlib.pyplot as plt

#constants in the 1D case
a1=1
a2=1
c=-5
d=0.1
D1=1000
D2=0.1
beta1=2
beta2=1
B=0.15

#calculates the equilibrium value.
qc_eq =((d**beta1)*(c-a1))/((c**beta1)*a2) )**((1/(2*beta1 -1))
qr_eq = (c*qc_eq)/d
print(qc_eq, qr_eq)

n = 1000 #number of points in the mesh.
mesh = PeriodicIntervalMesh(n, 50.0) #a mesh of length 50 with 1000 points

V = FunctionSpace(mesh, 'CG',1)
W = V*V

#Defining the variables required for the timestepping scheme.
qc_ = Function(V, name='Cloud')
q1 = Function(V,name='q_c')
qr_ = Function(V, name='CloudR')
q2 = Function(V,name='q_r')

#the test functions that multiply the equations to return the weak form.
v1 = TestFunction(V)
v2 = TestFunction(V)
```

Figure 16: The first part of code used to solve the cloud scheme in one-dimension.

```

x = SpatialCoordinate(mesh) #defining the spatial coordinate based on the mesh defined above.

#generating the random perturbation to apply to the initial condition.
pcg = PCG64(seed=123456789)
rg = RandomGenerator(pcg)

qc_pert = rg.normal(V)
qr_pert = rg.normal(V)

#equilibrium state plus a small random perturbation.
q1.assign(qc_eq + 0.1*qc_pert )
q2.assign(qr_eq + 0.1*qr_pert )

qc_.assign(q1)
qr_.assign(q2)

#timestep.
dt=0.1

#the discretized cloud model equations. Note B is included here to consider rain from above
F1 = v1*(q1- qc_)*dx - dt*( (c-a1)*(q1**beta1)*v1 - a2*(q1**beta2)*v1*(q2**beta2) - D1*inner(grad(q1),grad(v1)) )*dx
F2 = v2*(q2- qr_)*dx - dt*(B*v2+ a1*(q1**beta1)*v2 - d*q2*v2 + a2*(q1**beta2)*v2*(q2**beta2) - D2*inner(grad(q2),grad(v2)) )*dx

#defineing the file where the results are written to.
outfile= File('play.pvd')
outfile.write(q1,q2)

#time loop steps
t=0.0
count = 0
end = 2000
while (t<=end):
    solve(F1==0, q1) #solving the cloud scheme for q_c.
    qc_.assign(q1)
    solve(F2==0, q2) #solving the cloud scheme for q_r.
    qr_.assign(q2)
    t+=dt
    if count % 100 == 0: #reducing the number of files created by only creating a file every 100 timesteps
        print("t = ", t)
        outfile.write(q1,q2)
    count += 1

```

Figure 17: The second part of code used to solve the cloud scheme in one-dimension.

D 2D Code

The code used to solve the cloud scheme in two dimensions and produce the results seen in Figures 9 and 10. The only difference between this and the one dimensional case is the

```

from firedrake import *
import math
import matplotlib.pyplot as plt

#constants from section 7 in the 2D case
a1=1
a2=1
c=5
d=0.13   #note different to the 1D
D1=100   #note different to the 1D
D2=0.025 #note different to the 1D
beta=2

qc_eq =((d**beta)*(c-a1))/((c**beta)*a2) **(1/(2*beta -1)) #calculates the equilibrium value.
qr_eq = (c*qc_eq)/d

print(qc_eq, qr_eq) #checking that the equilibrium state is calculated correctly.

n = 1000 #number of points in the mesh.
mesh = PeriodicSquareMesh(n,n, 50.0) #a 50x50 periodic mesh.

V = FunctionSpace(mesh, 'CG',1)
W = V*V

#Defining the variables required for the timestepping scheme.
qc_ = Function(V, name='Cloud')
q1 = Function(V,name='CloudNext')
qr_ = Function(V, name='CloudR')
q2 = Function(V,name='CloudRNext')

#the test functions that multiply the equations to return the weak form.
v1 = TestFunction(V)
v2 = TestFunction(V)

```

Figure 18: The first part of code used to solve the cloud scheme in two-dimensions.

```

#defineing the spatial coordinate based on the mesh defined above.
x = SpatialCoordinate(mesh)

#defineing the random pertubation to apply to the initial condition.
pcg = PCG64(seed=123456789)
rg = RandomGenerator(pcg)

qc_pert = rg.normal(V)
qr_pert = rg.normal(V)

q1.assign(qc_eq + 0.1*qc_pert ) #equilibrium state plus a small random pertubation.
q2.assign(qr_eq + 0.1*qr_pert )

qc_.assign(q1)
qr_.assign(q2)

#timestep.
dt=0.1

#the discretized cloud model equations.
F1 = v1*(q1 - qc_)*dx - dt*( (c-a1)*q1*v1 - a2*(q1**beta)*v1*q2**beta - D1*inner(grad(q1),grad(v1)) )*dx
F2 = v2*(q2 - qr_)*dx - dt*( a1*q1*v2 - d*q2*v2 + a2*(q1**beta)*v2*q2**beta - D2*inner(grad(q2),grad(v2)) )*dx

#defineing the file where the results are written to.
outfile= File("cloud2D.pvd")
outfile.write(q1,q2)

#time loop steps.
t=0.0
count = 0
end = 120
while (t<=end):
    solve(F1==0, q1) #solving the cloud scheme for q_c.
    qc_.assign(q1)
    solve(F2==0, q2) #solving the cloud scheme for q_r.
    qr_.assign(q2)
    t+=dt
    if count % 20 == 0: #reducing the number of files created by only creating a file every 20 timesteps
        print("t = ", t)
        outfile.write(q1,q2)
    count +=1

```

Figure 19: The second part of code used to solve the cloud scheme in two-dimensions.

E 1D results at t=120

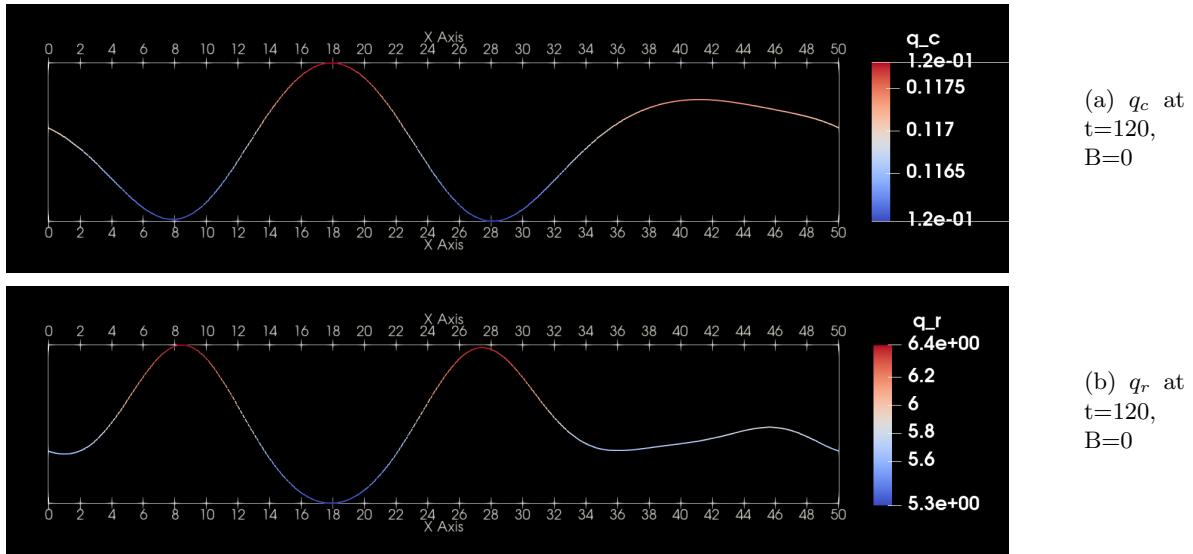


Figure 20: The spatial distribution of q_c and q_r at $t=120$.

The range of q_c and q_r in Figure 20 are 0.002 and 1.1 respectively. The results for the 2D case have a range of 0.0016 and 0.7 respectively. Hence there is a similar order of magnitude between both cases of my results. On the other hand, the 2D results in [Rosemeier and Spichtinger, 2021] sees a range of 0.005 and 7 for q_c and q_r respectively. This is over twice as large when compared to my results for q_c and an order of magnitude greater in the q_r case.