Understanding the Mechanism of Pattern Formation in Passion Flower

Agastya Prakash Bhati

A dissertation submitted for the partial fulfilment of
BS-MS dual degree in Science

Indian Institute of Science Education and Research Mohali
April 2014

Certificate of Examination

This is to certify that the dissertation titled **"Understanding the Mechanism of Pattern Formation in Passion Flower"** submitted by **Mr. Agastya Prakash Bhati** (Reg. No. MS09010) for the partial fulfilment of BS-MS dual degree programme of the Institute, has been examined by the thesis committee duly appointed by the Institute. The committee finds the work done by the candidate satisfactory and recommends that the report be accepted.

Dr. Balanarayan	Dr. Abhishek Chaudhuri	Professor N. Sathyamurthy
		(Thesis supervisor)

Dated: April 25, 2014

कर्मण्येवाधिकारस्ते मा फलेषु कदाचन ।
मा कर्मफलहेतुर्भूर्मा ते सङ्गोऽस्त्वकर्मणि ॥

Dedication

I dedicate this thesis to my parents Om Prakash and Praveena Bhati. This is another milestone on the way. With your blessings, I hope to continue similarly and fulfil your dreams.

Declaration

The work presented in this dissertation has been carried out by me under the guidance of Professor N. Sathyamurthy at the Indian Institute of Science Education and Research Mohali.

This work has not been submitted in part or in full for a degree, a diploma, or a fellowship to any other university or institute. Whenever contributions of others are involved, every effort is made to indicate this clearly, with due acknowledgement of collaborative research and discussions. This thesis is a bonafide record of original work done by me and all sources listed within have been detailed in the bibliography.

Agastya Prakash Bhati
Dated: April 25, 2014

In my capacity as the supervisor of the candidate's project work, I certify that the above statements by the candidate are true to the best of my knowledge.

Professor N. Sathyamurthy
(Thesis supervisor)

Acknowledgment

My first and foremost thanks is to my thesis supervisor Professor N. Sathyamurthy. It was a great experience to work in his guidance on a nice challenging problem. I very much appreciate his unique way of handling a student's doubt by providing him/her the correct line of thought and then allowing one to figure out by oneself, as it helped me learning many new things during the process. Along with his significant inputs, he also provided me enough freedom to approach the problem differently by encouraging my ideas. Moreover, he is a very nice and down-to-earth person.

I am grateful to Dr. Abhishek Chaudhuri and Dr. Rajeev Kapri for sharing their expertise with me and driving me out from difficult situations. I learnt several technicalities from them without which my progress would have been much slow. I am thankful to Professor K. S. Viswanathan and Dr. P. Balanarayan for their fruitful discussions and valuable comments.

I owe sincere thanks to Vivek Kohar for sparing time for me and sharing his programming experience with me. I would also like to thank Dr. Trilochan Bagarati for his important inputs about RD theory and help in Mathematica code.

I cannot forget to thank my group member Vikash Dhindhwal and all my friends at IISER and otherwise, who created a joyful environment for me to work peacefully and enjoy in my spare time. My special appreciation to Anuj Jakhar for helping me in some mathematical aspects of my work and always supporting me throughout my stay at IISER.

Nothing is possible without financial resources. I would like to sincerely acknowledge DST, Government of India for providing me KVPY fellowship. I am much thankful to IISER Mohali for providing me infrastructure and Computer Centre for all the technical support.

No acknowledgement would ever adequately express my gratitude to my family. I would like to give a special mention to Dr. Om Prakash, Praveena and Aditya Bhati for always believing in me. Their moral support has always boosted my confidence and motivated me to achieve and their teachings have enabled me for it. It is their love and affection which gives me a reason to succeed.

Agastya Prakash Bhati
MS09010
IISER Mohali

List of Figures

1.1 Exemplary patterns in nature; Source: Internet 2

1.2 Spatio-temporal patterns in the Belousov-Zhabotinskii reaction[1] 3

1.3 Experimentally observed Turing patterns in chemical reactions 5

1.4 Some of the biological patterns computed by Murray using Thomas kinetics in RD equations[2] 7

1.5 *Example of computed patterns by Gierer-Meinhardt kinetics in RD equations*: **(A-E)** Experimental gene expression for tentacle formation in Hydra; **(F-I)** Computed patterns of the same gene expression (brown).[3] 8

3.1 **(a)** Plot of $h(k^2)$ defined by eq. (32) for typical kinetics. When $d > d_c$, $h(k^2) < 0$ for a finite range of $k^2 > 0$. **(b)** Plot of the largest of the eigenvalues $\lambda(k^2)$ using eq. (32) as a function of k^2. When $d > d_c$, there is a range of wave numbers $k_1^2 < k^2 < k_2^2$ which are linearly unstable. Reproduced from Murray, J. D. *Mathematical Biology* 2002; Vol. II. . . 23

3.2 Schematic illustration of the two qualitatively different cases of diffusion driven instability. **(a)** self-activating u also activates v, which inhibits both the reactants. The resulting initially growing pattern is shown in **(c)**. **(b)** Here the self-activating u inhibits v but is itself activated by v with the resulting pattern illustrated in **(d)**. The matrices give the signs of f_u , f_v , g_u , g_v evaluated at the steady state. **(e)** and **(f)** The reaction phase planes near the steady state. The arrows indicate the direction of change due to reaction (in the absence of diffusion). Case (e) corresponds to the interactions illustrated in (a) and (c), while that in (f) corresponds to the interactions illustrated in (b) and (d). Reproduced from Murray, J. D. *Mathematical Biology* 2002; Vol. II. . . 26

3.3 Typical two-dimensional spatial patterns indicated by the linearly unstable solution (45) when various wave numbers are in the unstable range. The shaded regions are where $u > u_0$, the uniform steady state. Reproduced from Murray, J. D. *Mathematical Biology* 2002; Vol. II. . . 27

4.1 Picture of the Passion flower (top view) 29
4.2 Alternation of white and violet colour bands in a Passion flower 31

6.1 Reproduction of the known pattern 38
6.2 Concentration pattern in extended array of cells 39
6.3 Effect of varying the ratio of diffusion coefficients of v to u 40
6.4 Concentration pattern with locally increased source density at one end while otherwise uniform; concentrations u and v are taken to be uniform everywhere 40
6.5 Evolution of final pattern initiated by local small increment in source density at one end. 41
6.6 Pattern initiated by local perturbation in activator concentration at one end in space from otherwise uniform steady state 41
6.7 Evolution of final pattern initiated by local small increment in activator concentration at one end. 42
6.8 Patterns by initially perturbing activator concentration at different points in space and their combinations. 42
6.9 Known patterns reproduced 43
6.10 Effect of parameter γ on the pattern formation 44
6.11 Fourier Analysis of the measured pattern 45

A.1 Discretised rectangular domain 55
A.2 Sketch for the FTCS scheme 58
A.3 Sketch for the Crank-Nicholson scheme 60

B.1 Comparison of the analytical (solid line) and the numerical solution (*) of the 2D heat equation. 67

List of Tables

4.1 Measured pattern in Passion flower; W: White, V: Violet 30

Acronyms

RD : Reaction-Diffusion

PDE: Partial Differential Equation

TDSE: Time-Dependent Schrödinger Equation

DDI: Diffusion-Driven Instability

BZ: Belousov-Zhabotinskii

CIMA: Chlorite-Iodide-Malonic Acid

FIS: Ferrocyanide-Iodate-Suphite

FD: Finite Difference

FTCS: Forward-Time-Central-Space

BTCS: Backward-Time-Central-Space

DCT: Discrete Cosine Transform

Contents

List of Figures **vii**

List of Tables **ix**

Acronyms **x**

Abstract **xiii**

1 Introduction **1**

2 Fluid dynamics: Introduction to the diffusion equation **9**

3 Reaction-Diffusion theory for biological pattern formation **13**

3.1 Basic theory from origin 13

3.2 Different non-linear kinetics 14

3.2.1 Thomas kinetics 15

3.2.2 Meinhardt kinetics 15

3.2.3 Schnakenberg kinetics 16

3.3 Dimensionless RD system 17

3.4 Conditions for Diffusion-Driven Instability: Linear Stability Analysis . 19

3.5 The range of unstable modes 24

3.6 An illustration of pattern generation 25

4 Measured colour pattern of Passion flower **29**

5 Methods and Tools **33**

5.1 Meinhardt's model 33

5.2 Murray's model 35

5.3 Numerical solution of the heat equation 35

6 Results and Discussion **37**

6.1 Meinhardt's model 37

6.2 Murray's model 41

6.3 Fourier Analysis of the measured pattern 43

7 Conclusions and Outlook **47**

Bibliography **49**

Appendices **55**

A Finite Difference methods **55**

A.1 Forward Time and Central Space (FTCS) Scheme 57

A.2 Backward Time Central Space (BTCS) scheme 58

A.3 Crank-Nicholson 59

A.4 Thomas algorithm for tridiagonal system of equations 61

A.5 Alternate Direction Implicit method 61

B C-programs **65**

B.1 ADI program benchmarked 65

B.2 FTCS scheme on Meinhardt's model-1d 70

B.3 ADI scheme on Murray's model-2d 72

B.4 Discrete Cosine Transform 76

Abstract

The evolution of spatial patterns is a central issue in developmental biology. Turing's chemical theory of morphogenesis was a seminal contribution. We describe briefly some of the interesting mathematical aspects of Turing's Reaction-Diffusion (RD) mechanism and give an overview of a few of the popular reaction models incorporated into it. The conditions on kinetic and diffusion parameter values under which pattern formation takes place are derived. We utilize our understanding of Turing's RD mechanism to study pattern formation in *Passiflora Incarnata* (Passion Flower), which has a pattern of alternate bands of white and violet colours on each of its fibrils with a unique feature of non-uniform widths of the bands. We study systematically the effect of various kinetic and diffusion parameters on the generated patterns using the two different reaction models.

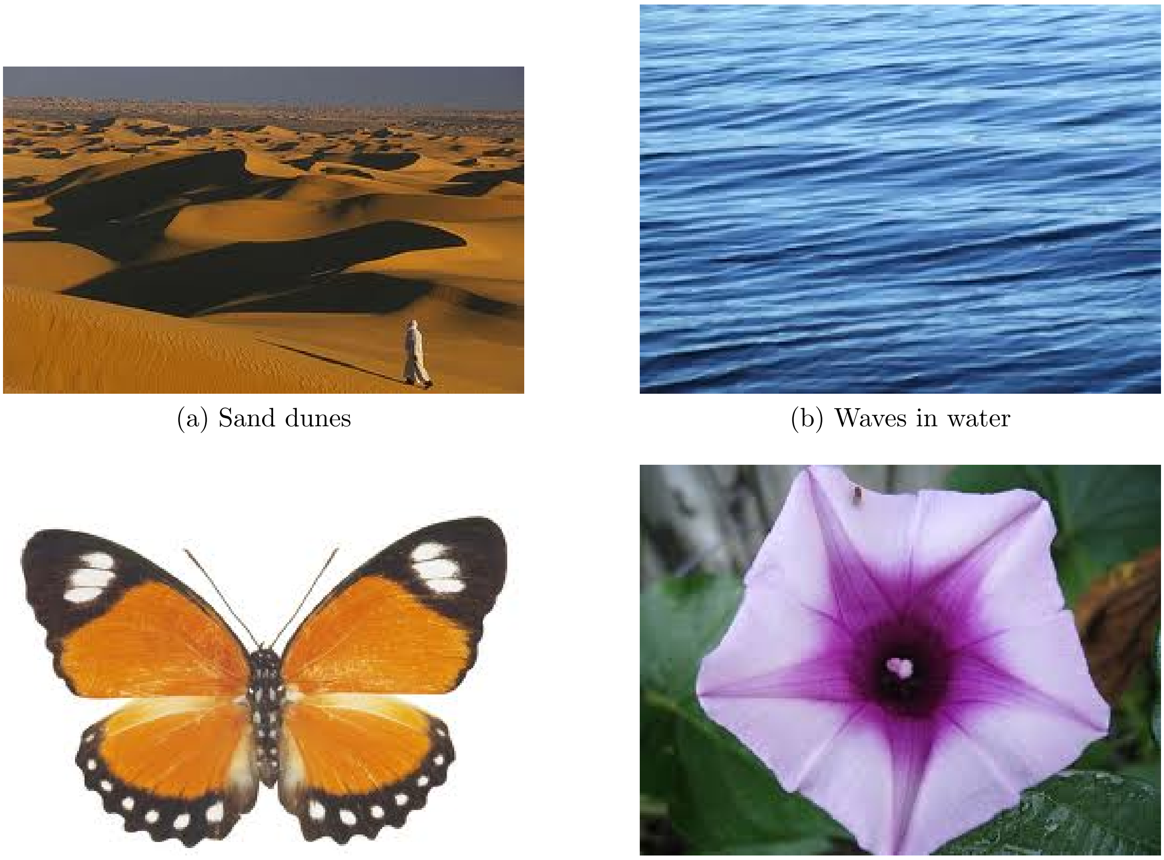
Chapter 1

Introduction

In a glance around oneself, one comes across several beautiful patterns in nature. It may be a living organism or a non-living object. Most of the naturally occurring things seem to have some pattern or another - simple or complex, due to colour, shapes or several other variations one can think of. Spatial and spatio-temporal patterns occur widely in chemistry too. The best known oscillatory reaction is perhaps the Belousov-Zhabotinskii reaction,[4] in which bromate ions oxidise malonic acid in a reaction catalysed by cerium (Ce^{3+}/Ce^{4+}). Sustained periodic oscillations are observed in the concentration of cerium ions. If, instead, one uses the catalyst Fe^{2+}/Fe^{3+} and phenanthroline, the periodic oscillations are visualised as colour changes between reddish-orange and blue.[5] This system can also exhibit a number of different types of wave structures such as propagating fronts, spiral waves, target patterns and toroidal scrolls.[6–11] It is quite natural to ask the question how these patterns are formed in nature.

One of the major issues in developmental biology is understanding morphogenesis, i.e., the emergence of structure and form from an almost uniform mass of dividing cells that constitutes the early embryo. Although genes play a key role, genetics says nothing about the actual mechanism that is responsible for the pattern formation. What exactly happens during embryogenesis which leads to an organism's shape? How do living organisms convert the detailed one-dimensional genetic information into a three-dimensional map, the shape of the living organism?

Many models of how different processes can come together to produce patterns have been proposed and analysed. They range from gradient-type models involving


(a) Sand dunes

(b) Waves in water

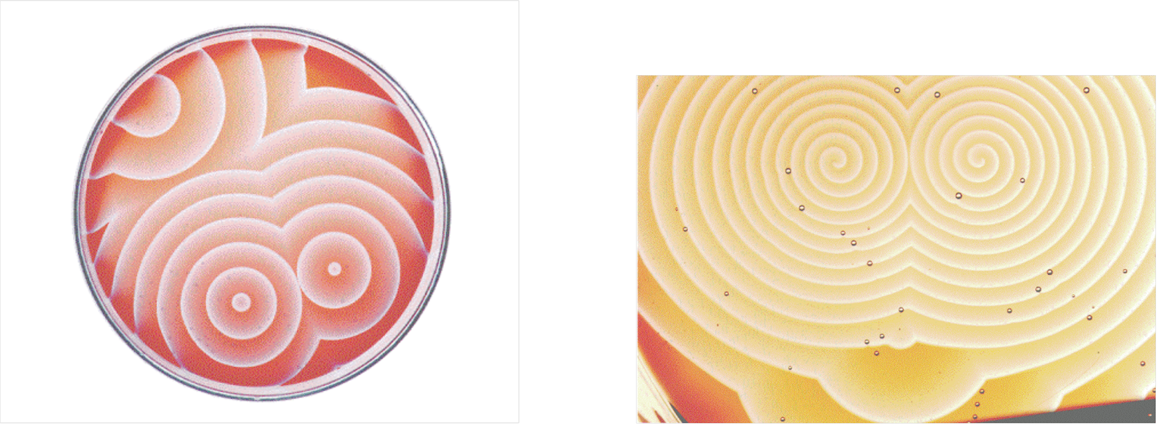
(c) Butterfly wings

(d) Symmertry in flower

Figure 1.1: Exemplary patterns in nature; Source: Internet

a simple source-sink mechanism[12] to cellular automata models in which the tissue is discretised and rules are introduced as to how different elements interact with each other[13] to more complicated models which incorporate more sophisticated chemistry and biology. Here we shall focus on the model from the latter category.

Flowing liquids, for instance in a river, can generate spatial forms that remain constant over time. They owe their existence to the shape of the river bottom, but their expressed form cannot be explained without considering also the dynamics of the flow. In the laboratory, very orderly patterns can be generated in dishes of liquid undergoing convective flow. For biological development, the kinetic paradigm is that the form of living pattern is often generated by the flow of material through numerous biochemical reacting systems of life. This kinetic preconception was put on a firm mathematical basis by Alan Turing in 1952, in a paper entitled 'The chemical basis of morphogenesis'.[14]



(a) Target patterns

(b) Spiral patterns

Figure 1.2: Spatio-temporal patterns in the Belousov-Zhabotinskii reaction[1]

Turing proposed a mathematical model for the mechanism of morphogenesis from a mixture of chemicals (morphogens) which react together and diffuse through the tissues. A spatial pattern in these chemicals may be set up which in turn determines cell differentiation. Thus the pattern or structure that we observe is due to the underlying pre-pattern in morphogens. This model is named Reaction-Diffusion. Turing demonstrated theoretically that a homogeneous mixture of such chemicals could lead to spatial patterns of their concentrations triggered by random disturbances. He showed that diffusion could drive a chemical system to instability, leading to spatial patterns where no prior patterns existed. More specifically, he considered a system of an activator and an inhibitor. The activator stimulated and enhanced the production of the inhibitor, while the inhibitor depleted or inhibited the formation of the activator. He showed that if the diffusion of the inhibitor was greater than that of the activator, then diffusion-driven instability could result. It is worth mentioning here that mixtures of small molecules and ions in aqueous solution may generate travelling waves (a well-known example is Belousov-Zhabotinskii reaction[6–11]), but do not produce stationary patterns, because all the solutes have similar diffusivities. Experimental success in producing undoubted Turing patterns in non-living chemical systems has arisen from the use of polymeric substances which can differentially slow down some of the diffusing reactants.

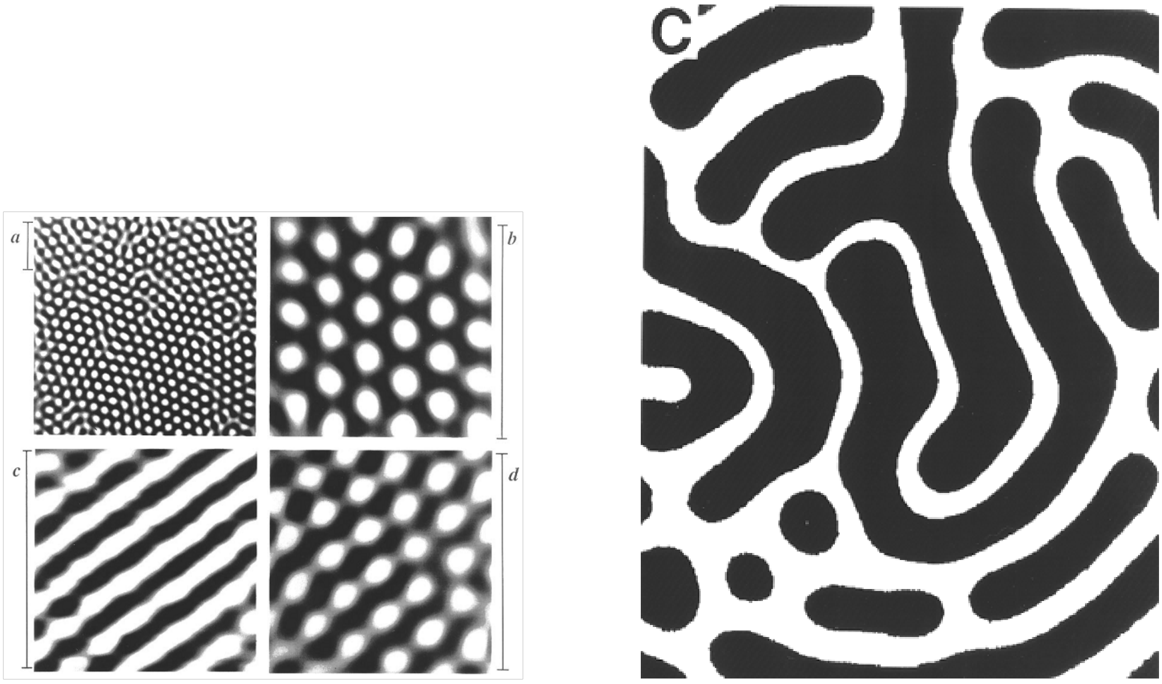
A more general and refined form of the reaction diffusion equation, derived from its original form proposed by Turing is

$$\frac{\partial \boldsymbol{u}}{\partial t} = \boldsymbol{D}\nabla^2 \boldsymbol{u} + \boldsymbol{f}(u, p), \tag{1.1}$$

where $\boldsymbol{u}$ is a vector of chemical concentrations, $\boldsymbol{D}$ is the matrix of diffusion coefficients and $\boldsymbol{f}$ represents chemical coupling with kinetic parameters p.

In late 1960s, Prigogine and co-workers[15–17] pointed out clearly that the direct spontaneous transition from a uniform state to a stationary patterned state (the Turing bifurcation) requires that the system involves at least one positive (e.g. autocatalysis) and one negative feedback (inhibition) process, includes chemical species with appropriately different diffusion coefficients and, last but not the least, operates far from thermodynamic equilibrium. Zhabotinskii *et al.* and Winfree published their experimental observations of long-lasting wave trains in batch solutions of BZ reaction.[6,7,18,19] Turing's approach to pattern formation was theoretically further extended in the 1970s and 1980s by various people, of which the works of Hans Meinhardt[20] and James Murray[2,21,22] are particularly very popular. Several reviews on the subject have been published already.[23–25] The first indisputable demonstration of Turing's prediction in chemical systems was provided by the 'Non-Linear Dynamic Systems' group in Bordeaux, at the end of 1989, nearly forty years after it was proposed.[26] This breakthrough was made possible by operating a newly discovered reaction - the chlorite-iodide-malonic acid (CIMA) reaction[27] - in newly invented open spatial reactors of different geometries. Soon after the observation of these stationary patterns, static and spatio-temporal patterns were observed in the ferrocyanide-iodate-sulphite (FIS) reaction.[28] Thereafter many experimental instances have been found for the RD theory of pattern formation.[29]

A careful look at eq. (1.1) tells us that the general RD equation is basically a set of coupled partial differential equations with first order temporal derivatives on the left hand side and second order spatial derivatives on the right hand side. This general form of RD equation may have any number of concentration variables. Many researchers confine themselves to just two concentration variables, as originally proposed by Turing, out of which one is the activator and the other is the inhibitor. Therefore, hereafter, we will refer to the following equation for RD theory, which



(a) Stationary chemical patterns in the CIMA reaction (bars represent 1mm)[30]

(b) Time asymptotic pattern for the FIS reaction[31]

Figure 1.3: Experimentally observed Turing patterns in chemical reactions

contains just two variables $u = u_1$ and $v = u_2$:

$$\frac{\partial u}{\partial t} = D_u \nabla^2 u + f(u, v) \tag{1.2a}$$

$$\frac{\partial v}{\partial t} = D_v \nabla^2 v + g(u, v) \tag{1.2b}$$

where $D_u = D_1$, $D_v = D_2$, $f_1(u_1, u_2) = f(u, v)$ and $f_2(u_1, u_2) = g(u, v)$.

The reaction part of eq. (1.2) is the most amenable for modification and debate, as is evident from the history of RD theory. Turing proposed a linear form of kinetics. Several non-linear forms have been proposed subsequently. Following are a few exemplary non-linear versions of reaction kinetics used in RD equations:
Schnakenberg kinetics:[32]

$$f(u, v) = k_1 - k_2 u + k_3 u^2 v \tag{1.3a}$$

$$g(u, v) = k_4 - k_3 u^2 v \tag{1.3b}$$

Thomas kinetics (adopted by Murray):[33]

$$f(u,v) = k_1 - k_2u - h(u,v) \tag{1.4a}$$

$$g(u,v) = k_3 - k_4v - h(u,v) \tag{1.4b}$$

$$h(u,v) = \frac{k_5uv}{k_6 + k_7u + k_8u^2} \tag{1.4c}$$

Meinhardt kinetics:[34]

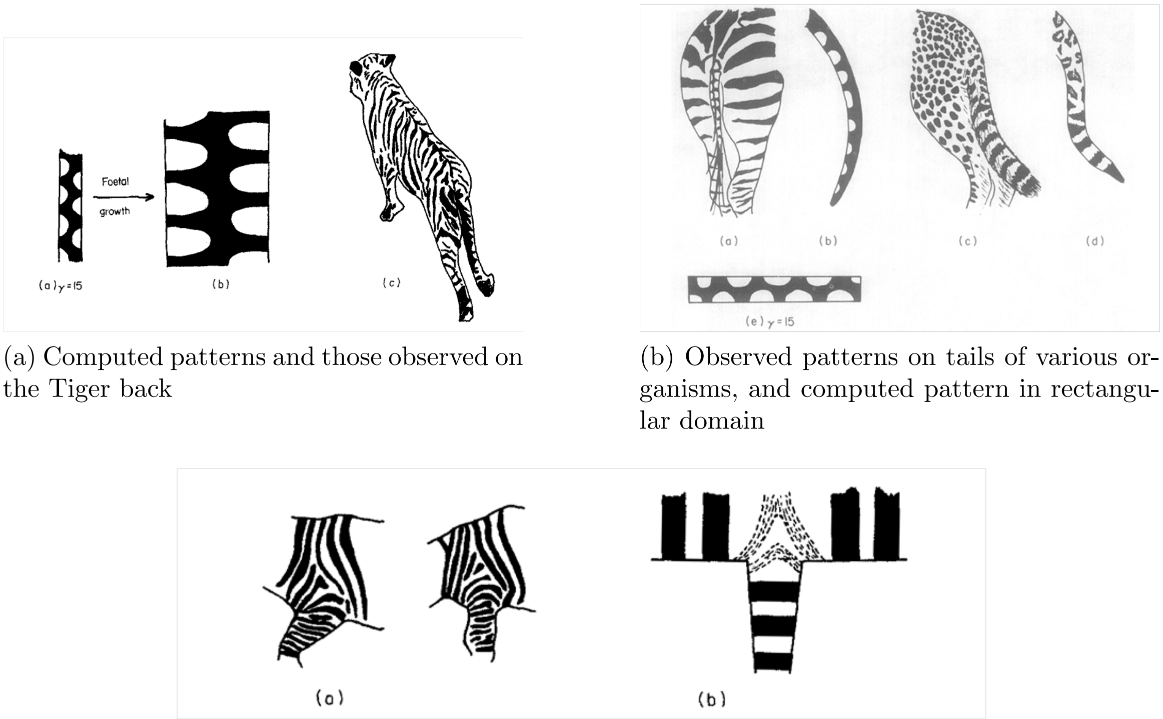
$$f(u,v) = k_1 - k_2u + \frac{k_3u^2}{v} \tag{1.5a}$$

$$g(u,v) = k_4u^2 - k_5v \tag{1.5b}$$

The above equations are three different sets of kinetic equations employed in the RD theory for biological pattern generation. Out of these, kinetics (1.4) and (1.5) are very popular and extensively applied to generate well-known biological shapes and patterns numerically. Both these versions of kinetic equations have been used successfully to understand the mechanism at the molecular level, of various biological phenomena and patterns such as, cartilage formation in vertebrate limbs, coloured patterns of butterfly wings, alligator teeth, head regeneration in Hydra, spots of Cheetah, stripes of Zebra and many more.[2,20,21] By their enormous contribution to the development of the RD theory, Murray and Meinhardt have popularized this field very much and widened the domain of its application.

For a researcher who is trying to understand the mechanism of biological pattern formation using the RD theory, the prime goal is to solve the above set of partial differential equations (1.2) and hence see the evolution of the desired concentrations (of morphogens) with time and space, finally leading to the desired pattern. Here the kinetic terms $f(u,v)$ and $g(u,v)$ are basically the coupling terms of PDEs. With the non-linear coupling terms such as the ones mentioned above (equations (1.3), (1.4) and (1.5)), the set of PDEs (eq. (1.2)) is not analytically solvable. In the absence of an analytic solution, we try to solve them numerically. It is worth mentioning here that the RD equation is analogous to the TDSE, except for the imaginary part in the latter.

The general procedure to generate patterns using RD eq. (1.2) is the following. First of all, a uniform steady state is found. It is the solution of $f(u,v) = 0$ and $g(u,v) = 0$



(a) Computed patterns and those observed on the Tiger back

(b) Observed patterns on tails of various organisms, and computed pattern in rectangular domain

(c) Typical examples of stripes on the foreleg of zebra, and expected mathematical pattern for intersecting striped domains

Figure 1.4: Some of the biological patterns computed by Murray using Thomas kinetics in RD equations[2]

yielding u_0 and v_0, which are constant in space and time. Under certain conditions of kinetic parameters and diffusion coefficients, such a steady state, which is stable in the absence of diffusion, could be made unstable in the presence of diffusion and evolve to a spatially patterned state. Turing was interested in morphogenesis. Therefore, he proposed that this chemical pattern could serve as a pre-pattern to which cells would respond in such a way that a spatial structure would be formed. For example, if one of the chemicals was a plant growth hormone, then the pre-pattern could result in a patterned growth. More generally, these chemicals (termed morphogens) could trigger a genetic switch causing cell differentiation.

Passion flower (*Passiflora incarnata*) exhibits a beautiful pattern on its "fibrils" formed by alternate bands of violet and white colours. Each fibril has such a pattern of colours that collectively all the fibrils give rise to concentric circles of colour bands due to their radially symmetric position on the flower. In this project we focus on understanding the mechanism of the occurrence of this pattern using RD theory. We

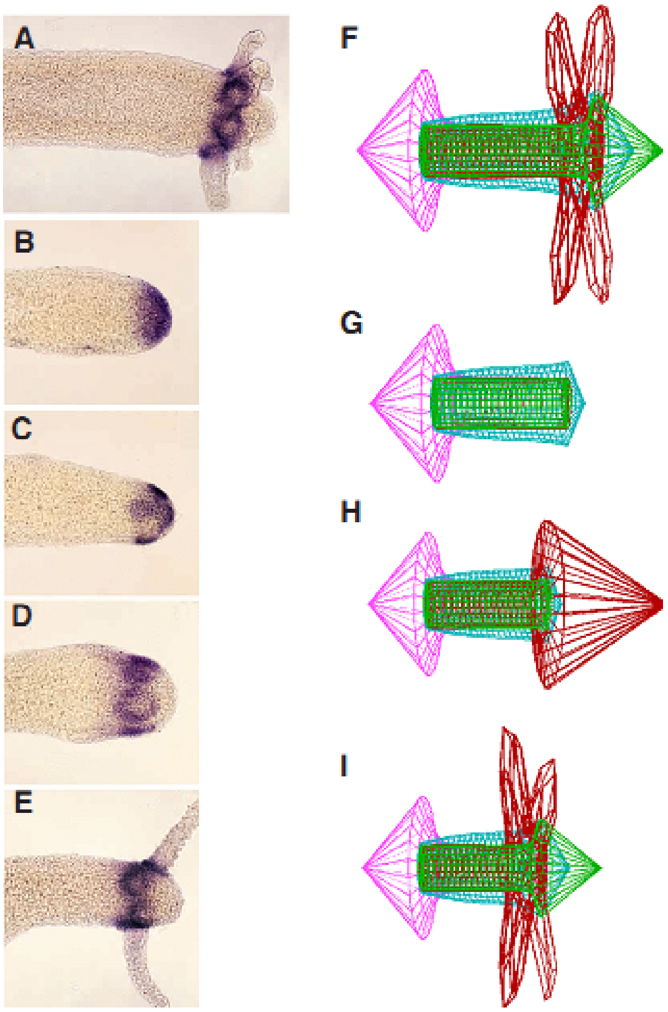


Figure 1.5: *Example of computed patterns by Gierer-Meinhardt kinetics in RD equations*: **(A-E)** Experimental gene expression for tentacle formation in Hydra; **(F-I)** Computed patterns of the same gene expression (brown).[3]

have started with a one-dimensional domain of cells/tissues of a single fibril. The idea is to increase the complexity in steps and increase the number of dimensions. The RD equations have been solved numerically for both Murray and Meinhardt kinetics to generate the desired patterns. Finite difference methods have been employed to obtain numerical solutions of the equations.

Chapter 2

Fluid dynamics: Introduction to the diffusion equation

The problem of flowing fluid is a standard one and applicable to various real life cases. A spatial non-uniformity in concentration of a fluid drives it to flow. In other words, a non uniform distribution of the constituent particles of fluid make them diffuse. The constituent particles may also interact with each other (via. some chemical reactions for example). The mathematical equation which is used to handle such a process is derived below.[35,36]

Let R^n with $n > 1$ be the n^{th} dimensional real space. In particular, we are interested in the cases of $n = 2$ and 3. We assume that Ω is small region in this space. Let *P(t, r)* be the density function of the constituent particles, where t is the time, and $r \in \Omega$ is a point in the n^{th} dimensional space. The dimension of density is number of particles per unit area (if $n = 2$) or unit volume (if $n = 3$).

We need to study the change in function *P(t, r)* with time t and location r. It can happen in two ways: one being diffusion, that is the individual particles move, and the second is production of new particles and/or consumption of existing particles. This may happen due to several reasons, for example, chemical reaction. We model both of these possibilities separately.

We first deal with diffusion. The amount of a substance which passes through a point in space per unit area per unit time is called its **flux** at that point. According to Fick's law of diffusion,[37] the flux of density function *P(t, r)* is a vector pointing from high density region to low density region and with its magnitude proportional to density gradient. Mathematically it can be represented as below:

$$\boldsymbol{J(t,r)} = -D(r)\nabla_r P(t,r), \tag{2.1}$$

where $\boldsymbol{J}$ is the flux of P, *D(r)* is the **diffusion constant** at r, ∇_r is the gradient operator $\nabla_r f(r) = \left(\frac{\partial f}{\partial r_1}, \frac{\partial f}{\partial r_2}, ..., \frac{\partial f}{\partial r_n}\right)$.

The production and/or consumption of constituent particles at any point per unit time, that is the rate of change of the density function, which may occur due to reasons like physical transformation or chemical reactions, is assumed to be given by *f(t, r, P)*, which is called the **reaction rate**. Let O be region in space, then the total number of constituent particles in O is $\int_O P(t,r)dr$, where dr is the infinitesimal volume element. Thus, the rate of change of the total number of particles is

$$\frac{d}{dt}\int_O P(t,r)dr. \tag{2.2}$$

Now we apply the law of mass conservation on this system to derive the Reaction-Diffusion equation. The net production of particles inside the region O is

$$\int_O f(t,r,P(t,r))dr \tag{2.3}$$

and the total out-flux is

$$\int_{\delta O} \boldsymbol{J}(t,r).\boldsymbol{n}(r)dS, \tag{2.4}$$

where δO is the boundary of O and $\boldsymbol{n}(r)$ is the outer normal direction at r. Therefore on conserving the total number of particles we get

$$\frac{d}{dt}\int_O P(t,r)dr = -\int_{\delta O} \boldsymbol{J}(t,r).\boldsymbol{n}(r)dS + \int_O f(t,r,P(t,r))dr. \tag{2.5}$$

From the Divergence Theorem in multi-variable calculus we have

$$\int_{\delta O} \boldsymbol{J}(t,r).\boldsymbol{n}(r)dS = \int_O \boldsymbol{\nabla}.(\boldsymbol{J}(t,r))dr. \tag{2.6}$$

Combining eq. (2.1), (2.5) and (2.6), and interchanging the order of differentiation and integration we obtain

$$\int_O \frac{\partial P(t,r)}{\partial t} dr = \int_0 [\boldsymbol{\nabla}.(D(r)\nabla_r P(t,r)) + f(t,r,P(t,r))] dr \tag{2.7}$$

Since the choice of region O is arbitrary, the differential equation

$$\frac{\partial P(t,r)}{\partial t} = \boldsymbol{\nabla}.(D(r)\nabla_r P(t,r)) + f(t,r,P(t,r)) \tag{2.8}$$

holds for any (t,r). The equation (2.8) is called a **reaction diffusion equation**. Here, $\boldsymbol{\nabla}.(D(r)\nabla_r P(t,r)$ is the diffusion term, which describes the movement of the particles under their density gradient and $f(t,r,P(t,r))$ is the reaction term which describes the reaction occurring in the domain.

The diffusion coefficient *D(r)* may not be a constant always as many systems are heterogeneous. But when the region of the diffusion is approximately homogeneous, we can assume that *D(r)* $\equiv D$, then eq. (2.8) can be simplified to

$$\frac{\partial P}{\partial t} = D \bigtriangleup P + f(t,r,P), \tag{2.9}$$

where $\bigtriangleup P = \boldsymbol{\nabla}.(\nabla_r P) = \sum_{i=1}^{n} \frac{\partial^2 P}{\partial r_i^2}$ is the Laplacian operator. When there is no reaction, the equation is the **diffusion equation**, as follows

$$\frac{\partial P}{\partial t} = D \bigtriangleup P. \tag{2.10}$$

In classical mathematical physics, the equation $T_t = \bigtriangleup T$ is called the **heat equation**, where T is the temperature function. Conduction of heat can be considered as a form of diffusion of heat.

Chapter 3

Reaction-Diffusion theory for biological pattern formation

3.1 Basic theory from origin

In this chapter we will discuss briefly the theory of Reaction-Diffusion of morphogenesis/pattern formation in living organisms. This theory was first proposed by Turing in 1952 in his seminal paper “The Chemical basis of Morphogenesis”.[14] He addressed the issue how an embryo from its blastula stage, when it is a spherically symmetric mass of cells, gives rise to an organism such as horse which is not spherically symmetric. He proposed that the chemicals, called morphogens, react and diffuse through the cells/tissues. The key idea is that these reacting and diffusing set of chemicals within the uniform homogeneous mass of cells set up a chemical pre-pattern, and the cells then differentiate following this pre-pattern, leading to the patterns/forms in the initial uniform mass of cells. A similar explanation is valid for any kind of biological patterns observed in nature. Turing gave a mathematical form to this idea. Following are the equations he used in his original paper for studying RD of two morphogens X and Y in a linear array of N cells:

$$\frac{dx_r}{dt} = ax_r + by_r + \mu(x_{r+1} - 2x_r + x_{r-1}) \tag{3.1a}$$

$$\frac{dy_r}{dt} = cx_r + dy_r + \nu(y_{r+1} - 2y_r + y_{r-1}) \tag{3.1b}$$

where x_r and y_r are perturbations in the steady state concentrations of X and Y in the r^{th} cell, $a,\ b,\ c,\ d$ are ‘marginal reaction rates’, μ and ν are cell to cell diffusion constants for X and Y, respectively, $r = 1$ to N.

Using the above set of equations, he demonstrated that, under certain conditions of parameters, an initially homogeneous distribution of concentrations of X and Y at their steady state could lead to spatially heterogeneous patterns of concentrations stable with time, in response to random perturbations about the steady state. In the absence of diffusion, the system would be resistant to such random perturbations about the steady state. Therefore, basically it was the counter intuitive concept of **diffusion driven instability**, introduced by Turing. A careful analysis of the conditions of parameters leading to such an instability suggests that it is a system of two chemicals, an activator and an inhibitor, and that the inhibitor has a higher diffusion rate than the activator.

To make the problem mathematically tractable, Turing assumed that the system never deviated far from the original homogeneous condition. This assumption was called **linearity assumption** because it permitted the replacement of the general reaction rate functions by linear ones. He believed that the chemical pre-pattern is formed during the early stages of embryogenesis when such an assumption is valid. He mentioned in his paper, “Its justification lies in the fact that the patterns produced in the early stages when it is valid may be expected to have strong qualitative similarity to those prevailing in the later stages when it is not.” (Turing, 1952, p.66) But this linearization led to some stability problems. The equations with non-linear reaction parts are not analytically solvable. Turing had also mentioned the possibility of numerically solving non-linear equations using digital computers.

We have derived the general form of the RD eq. (2.9) in the previous chapter. In our approach we have morphogens flowing through the cells/tissues and produced or consumed due to reaction with each other. We employ eq. (2.9) with the concentrations of morphogens as the variables in our RD theory. The diffusion part is the same and the reaction part is the reaction of morphogens. There are various forms of the reaction terms which have been proposed. Each has its own advantage as well as disadvantage, and yields different results. We will discuss a few of them in the following chapter.

3.2 Different non-linear kinetics

As was mentioned earlier, Turing had assumed linear kinetic terms in RD equations as they are analytically solvable. Since the publication of Turing’s paper, several RD models have been considered with non-linear kinetics terms. These models are more realistic in that their solutions evolve to bounded values (unlike linear ones where solutions tend to exponentially grow in time as we will see later), and are derived essentially in three different ways: (i) empirically, (ii) phenomenologically and (iii) through a hypothetical reaction. An example of (i) is the Thomas model which is based on a specific reaction. The model of Geirer and Meinhardt is an example of (ii) type, while the Schnakenberg model belongs to the (iii) category. We will now discuss each of them one by one.

3.2.1 Thomas kinetics

In empirical models, the kinetics are fitted to experimental data. The immobilized-enzyme substrate-inhibition mechanism of Thomas[33] involves the reaction of uric acid (concentration u) with oxygen (concentration v). Both reactants diffuse from a reservoir maintained at constant concentrations u_0 and v_0, respectively, on to a membrane containing the immobilized enzyme uricase. They react in the presence of the enzyme at the empirical rate $\dfrac{k_5uv}{k_6+k_7u+k_8u^2}$ so that

$$f(u,v) = \alpha(u_0-u) - \frac{k_5uv}{k_6+k_7u+k_8u^2} \tag{3.2a}$$

$$g(u,v) = \beta(v_0-v) - \frac{k_5uv}{k_6+k_7u+k_8u^2}, \tag{3.2b}$$

where $k_2=\alpha$, $k_4=\beta$, $k_1=\alpha u_0$, $k_3=\beta v_0$, k_5, k_6, k_7 and k_8 are positive constants.

3.2.2 Meinhardt kinetics

In phenomenological models, the kinetics is chosen such that one of the chemicals is an activator and the other an inhibitor. In the Gierer-Meinhardt model[34] as shown below, u is the activator; it is produced by autocatalysis and it activates the production of v, which is the inhibitor, inhibiting the production of u:

$$f(u,v) = \underbrace{k_1}_{\text{source}} - \underbrace{k_2u}_{\text{linear degradation}} + \underbrace{\frac{k_3u^2}{v}}_{\text{autocatalysis in } u/\text{inhibition from } v} \tag{3.3a}$$

$$g(u,v) = \underbrace{k_4u^2}_{\text{activation by } v} - \underbrace{k_5v}_{\text{linear degradation}} \tag{3.3b}$$

3.2.3 Schnakenberg kinetics

Schnakenberg (1979)[32] proposed a series of trimolecular autocatalytic reactions involving two chemicals as follows:

$$X \underset{\alpha_1}{\overset{k_2}{\rightleftharpoons}} A, \qquad B \xrightarrow{\beta_1} Y, \qquad 2X + Y \xrightarrow{k_3} 3X$$

Using the Law of Mass Action, which states that the rate of a reaction is directly proportional to the product of the active concentrations of the reactants, and denoting the concentrations of X, Y, A and B by u, v, α and β, respectively, we have

$$f(u,v) = \alpha_1\alpha - k_2 u + k_3 u^2 v \tag{3.4a}$$

$$g(u,v) = \beta_1\beta - k_3 u^2 v, \tag{3.4b}$$

where α_1, β_1, k_2 and k_3 are (positive) rate constants, $k_1 = \alpha_1\alpha$ and $k_4 = \beta_1\beta$. Assuming that there is an abundance of A and B, α and β can be considered to be approximately constant, and so are k_1 and k_4.

Remark: To verify Turing structures, a variation in diffusion coefficients is essentially required. For a general two-species RD system as shown below, the ratio may be changed as follows:[38]

$$\begin{aligned} \frac{\partial u}{\partial t} &= D_u \nabla^2 u + f(u,v) \\ \frac{\partial v}{\partial t} &= D_v \nabla^2 v + g(u,v). \end{aligned}$$

We additionally assume that the activator is involved in a reaction of the form:

$$U + S \underset{r_2}{\overset{r_1}{\rightleftharpoons}} C\cdot$$

Assuming that both S and C are immobile, the RD system is now modified to:

$$\begin{aligned} \frac{\partial u}{\partial t} &= D_u \nabla^2 u + f(u,v) - r_1 us + r_2 c \\ \frac{\partial v}{\partial t} &= D_v \nabla^2 v + g(u,v) \\ \frac{\partial c}{\partial t} &= r_1 us - r_2 c, \end{aligned}$$

where s and c are the concentrations of S and C, respectively, and r_1, r_2 are rate constants. If r_1 and r_2 are large, then using a singular perturbation, c can be approximated in terms of u by $c \equiv ru$, where $r = \dfrac{s_0 r_1}{r_2}$ and we have assumed that the concentration of S remains close to its initial value, s_0.

$$c = ru \qquad \Rightarrow \qquad \frac{\partial c}{\partial t} = r\frac{\partial u}{\partial t}.$$

On the addition of the first and fourth equations above, we obtain the following equation for the activator:

$$(1+r)\frac{\partial u}{\partial t} = D_u \nabla^2 u + f(u,v).$$

Thus when $r >> 1$ the diffusion of the activator is greatly reduced.

This demonstrates one way of reducing the effective diffusion rate of the chemical activator by the formation of an immobile complex. This approach was used by Lengyel and Epstein[39,40] to demonstrate Turing structures in the CIMA reaction. In this case, starch forms a stable complex with triiodide ions via the reaction

$$S + \mathrm{I}^- + \mathrm{I}_2 \rightleftharpoons S\mathrm{I}_3^-$$

and the high molecular weight of the complex reduces the rate of diffusion.

3.3 Dimensionless RD system

To reduce the number of parameters in a model, appropriate dimensionless quantities may be used. On introducing L as a typical length scale in the Schnakenberg model (3.4) and setting,

$$\begin{array}{cccc} x^* = \dfrac{x}{L}, & t^* = \dfrac{D_u t}{L^2}, & u^* = u\sqrt{\dfrac{k_4}{k_1}}, & v^* = v\sqrt{\dfrac{k_4}{k_1}}, \\ a = \alpha\dfrac{k_2}{k_1}\sqrt{\dfrac{k_4}{k_1}}, & b = \beta\dfrac{k_3}{k_1}\sqrt{\dfrac{k_4}{k_1}}, & d = \dfrac{D_v}{D_u}, & \gamma = \dfrac{L^2 k_1}{D_u} \end{array}$$

The dimensionless RD system becomes (dropping * for notational convenience),

$$\frac{\partial u}{\partial t} = \gamma(a - u + u^2 v) + \nabla^2 u \tag{3.5a}$$

$$\frac{\partial v}{\partial t} = \gamma(b - u^2 v) + d\nabla^2 v. \tag{3.5b}$$

The number of parameters has been reduced from eight to four in this case. Of course, the choice of dimensionless variables is not unique. An alternate version is to set $\gamma = 1$; that is, $L = \sqrt{\frac{D_u}{k_1}}$. This is equivalent to setting $x^* = \gamma^{1/2} x$ and $t^* = \gamma t$ so as to incorporate γ into new length and time scales. This reduces the number of parameters to three. But we retain the former dimensionless form of the RD equations due to its advantages, which will become clear in the following text.

An appropriate dimensionless kinetics in eq. (3.2) yields,

$$\begin{aligned} f(u,v) &= a - u - h(u,v) \\ g(u,v) &= \alpha(b - v) - h(u,v) \\ h(u,v) &= \frac{\rho u v}{1 + u + K u^2} \end{aligned}$$

and that of the kinetics in eq. (3.3) yields

$$\begin{aligned} f(u,v) &= a - bu + \frac{u^2}{v} \\ g(u,v) &= u^2 - v \end{aligned}$$

In general, any such RD system can be dimensionless and scaled to take the general form[21]

$$\frac{\partial u}{\partial t} = \gamma f(u,v) + \nabla^2 u \tag{3.6a}$$

$$\frac{\partial v}{\partial t} = \gamma g(u,v) + d\nabla^2 v, \tag{3.6b}$$

where d is the ratio of diffusion coefficients. Based on its definition and appearance in the dimensionless equations, γ can have any of the following interpretations.[21]

1. $\gamma^{1/2}$ is proportional to the *linear* size of the spatial domain in one dimension. In two dimensions γ is proportional to the area. That is, it can be used as a handle to increase or decrease the size/volume of the domain.

2. γ represents the relative strength of the reaction terms. This means, for example, that an increase in γ may represent an increase in the activity of some rate-limiting step in the reaction sequence.

3.4 Conditions for Diffusion-Driven Instability: Linear Stability Analysis

Definition: A uniform steady state is a state $(u, v) = (u_0, v_0)$ where u_0 and v_0 are constants in time and space, satisfying eq. (3.6) and the boundary conditions. We take zero flux boundary conditions.
Zero flux boundary conditions are satisfied by any (u_0, v_0), and eq. (3.6) are satisfied by

$$f(u_0, v_0) = g(u_0, v_0) = 0.$$

As u and v represent chemical concentrations, we consider only non-negative solutions to these equations.

Definition: Diffusion-driven instability (DDI), sometimes called Turing instability, occurs when a uniform steady state is stable to small perturbations in the absence of diffusion, but becomes unstable to small spatial perturbations when diffusion is present.

For any steady state to exhibit DDI, there are certain conditions which must be satisfied. We now carry out a linear stability analysis to derive the conditions under which a DDI can arise.[21]

We first determine the conditions for the uniform steady state to be linearly stable in the absence of any spatial variation. The equations in this case become as follows:

$$\frac{\partial u}{\partial t} = \gamma f(u, v) \tag{3.7a}$$

$$\frac{\partial v}{\partial t} = \gamma g(u, v) \tag{3.7b}$$

Linearising about the steady state (u_0, v_0), we set

$$w = \begin{pmatrix} u - u_0 \\ v - v_0 \end{pmatrix} \tag{3.8}$$

and for small $|w|$, eq. (3.7) becomes

$$\frac{\partial \boldsymbol{w}}{\partial t} = \gamma A \boldsymbol{w}, \qquad A = \begin{pmatrix} f_u & f_v \\ g_u & g_v \end{pmatrix}_{u_0, v_0}, \tag{3.9}$$

where A is the stability matrix and $x_i = \dfrac{\partial x}{\partial i}$ for $x =f$, g and $i = u$, v. Hereafter we assume that the partial derivatives of f and g are evaluated at the steady state unless stated otherwise. We take the solutions as follows

$$\boldsymbol{w} \propto e^{\lambda t}, \tag{3.10}$$

where λ is the eigenvalue. If $Re\lambda < 0$ then $\boldsymbol{w} \to 0$ as $t \to \infty$ and the steady state $\boldsymbol{w} = 0$ is linearly stable. We determine the eigenvalues λ as the solutions of the equation formed by substitution of eq. (3.10) into eq. (3.9) as follows

$$\begin{gathered} |\gamma A - \lambda I| = \begin{vmatrix} \gamma f_u - \lambda & \gamma f_v \\ \gamma g_u & \gamma g_v - \lambda \end{vmatrix} = 0 \\ \Rightarrow \lambda^2 - \gamma(f_u + g_v)\lambda + \gamma^2(f_u g_v - f_v g_u) = 0. \end{gathered} \tag{3.11}$$

Therefore,

$$\lambda_1, \lambda_2 = \frac{1}{2}\gamma \left[(f_u + g_v) \pm \left\{ (f_u + g_v)^2 - 4(f_u g_v - f_v g_u) \right\}^{1/2} \right] \tag{3.12}$$

The condition of linear stability, that is, $Re\lambda < 0$, is surely fulfilled if

$$trA = f_u + g_v < 0, \qquad |A| = f_u g_v - f_v g_u > 0. \tag{3.13}$$

Since (u_0, v_0) are functions of the parameters of the kinetics, these inequalities impose certain constraints on the parameters.

Now let us consider the full RD system given in eq. (3.6). Once again we linearize about the steady state, $\boldsymbol{w} = 0$, to get

$$\frac{\partial \boldsymbol{w}}{\partial t} = \gamma A \boldsymbol{w} + D\nabla^2 \boldsymbol{w}, \qquad A = \begin{pmatrix} f_u & f_v \\ g_u & g_v \end{pmatrix}_{u_0,v_0}, \qquad D = \begin{pmatrix} 1 & 0 \\ 0 & d \end{pmatrix}. \tag{3.14}$$

To solve this system of equations subject to the zero flux boundary conditions we define $\boldsymbol{W(r)}$ to be time-independent solution of the spatial eigenvalue problem defined as follows

$$\nabla^2 \boldsymbol{W} + k^2 \boldsymbol{W} = 0, \qquad (\boldsymbol{n} \cdot \nabla)\boldsymbol{W} = 0 \quad \text{for} \quad \boldsymbol{r} \text{ on boundary,} \tag{3.15}$$

where k is the eigenvalue and $\boldsymbol{W_k(r)}$ is the eigenfunction corresponding to the wave number k. Each W_k satisfies zero flux boundary conditions. We now assume solutions $\boldsymbol{w}(\boldsymbol{r}, t)$ of eq. (3.14) of the following form

$$\boldsymbol{w}(\boldsymbol{r}, t) = \sum_k c_k e^{\lambda t} \boldsymbol{W_k}(\mathbf{r}), \tag{3.16}$$

where the constants c_k can be determined by a Fourier expansion of the initial conditions in terms of $\boldsymbol{W_k}(\boldsymbol{r})$. λ is the eigenvalue which determines temporal growth. Substituting this form into eq. (3.14) with eq. (3.15) and cancelling $e^{\lambda t}$, we get, for each k,

$$\begin{aligned} \lambda \boldsymbol{W_k} &= \gamma A \boldsymbol{W_k} + D\nabla^2 \boldsymbol{W_k} \\ &= \gamma A \boldsymbol{W_k} - Dk^2 \boldsymbol{W_k} \end{aligned}$$

For nontrivial solutions for $\boldsymbol{W_k}$, the λ values are determined by the roots of the characteristic polynomial

$$\left|\lambda I - \gamma A + Dk^2\right| = 0.$$

On evaluating the above determinant with A and D taken from eq. (3.14), we get the eigenvalues $\lambda(k)$ as functions of the wave number k as the roots of

$$\begin{aligned} \lambda^2 + \lambda\Big[k^2(1+d) - \gamma(f_u + g_v)\Big] + h(k^2) = 0, \\ h(k^2) = dk^4 - \gamma(df_u + g_v)k^2 + \gamma^2|A|. \end{aligned} \tag{3.17}$$

We had earlier imposed the constraints that the steady state is stable in the absence of any spatial effects; that is, $\mathrm{Re}\lambda(k^2 = 0) < 0$, and found the required conditions (3.13). Now, in the presence of diffusion, the steady state would be unstable to spatial disturbances when additionally $\mathrm{Re}\lambda(k) > 0$ for some $k \neq 0$. For this to happen either the coefficient of λ in eq. (3.17) should be negative, or $h(k^2)$ should be less than 0 for some $k \neq 0$. Since $(f_u + g_v) < 0$ from conditions (3.13) and $k^2(1 + d) > 0$ for all $k \neq 0$ the coefficient of λ, namely,

$$\Big[k^2(1+d) - \gamma(f_u + g_v)\Big] > 0,$$

The only way Re $\lambda(k^2)$ can be positive is if $h(k^2) < 0$ for some k. This becomes clear immediately from the solutions of eq. (3.17), namely,

$$\lambda = -\frac{1}{2}\Big[k^2(1+d) - \gamma(f_u + g_v)\Big] \pm \Big\{\big[k^2(1+d) - \gamma(f_u + g_v)\big]^2 - 4h(k^2)\Big\}^{1/2}.$$

Since we required the determinant $|A| > 0$ from eq. (3.13), the only possibility for $h(k^2)$ in eq. (3.17) to be negative is if $(df_u + g_v) > 0$. Since $(f_u + g_v) < 0$ from eq. (3.13), this implies that $d \neq 1$ and f_u and g_v must have opposite signs. So, a further requirement to those in eq. (3.13) is

$$df_u + g_v > 0 \qquad \Rightarrow \qquad d \neq 1. \tag{3.18}$$

For the reaction kinetics mentioned in the previous chapter we can show that $f_u > 0$ and $g_v < 0$, so the first condition in eq. (3.13) and the last inequality in eq. (3.18) require that the diffusion coefficient ratio $d > 1$.

The inequality (3.18) is necessary but not sufficient for $\mathrm{Re}\lambda > 0$. For $h(k^2)$ to be negative for some nonzero k, the minimum h_{min} must be negative. Differentiating (3.17) with respect to k^2 shows that

$$h_{min} = \gamma^2\left[|A| - \frac{(df_u + g_v)^2}{4d}\right], \qquad k^2 = k^2_{min} = \gamma\frac{df_u + g_v}{2d}. \tag{3.19}$$

Thus the condition that $h(k^2) < 0$ for some $k^2 \neq 0$ is

$$\frac{(df_u + g_v)^2}{4d} > |A|. \tag{3.20}$$

We recollect below all the four required conditions for the DDI to occur. It is to be noted that all the derivatives are evaluated at the steady state (u_0, v_0).

$$\begin{aligned} &f_u + g_v < 0, \qquad && f_u g_v - f_v g_u > 0, \\ &df_u + g_v > 0, \qquad && (df_u + g_v)^2 - 4d(f_u g_v - f_v g_u) > 0. \end{aligned} \tag{3.21}$$

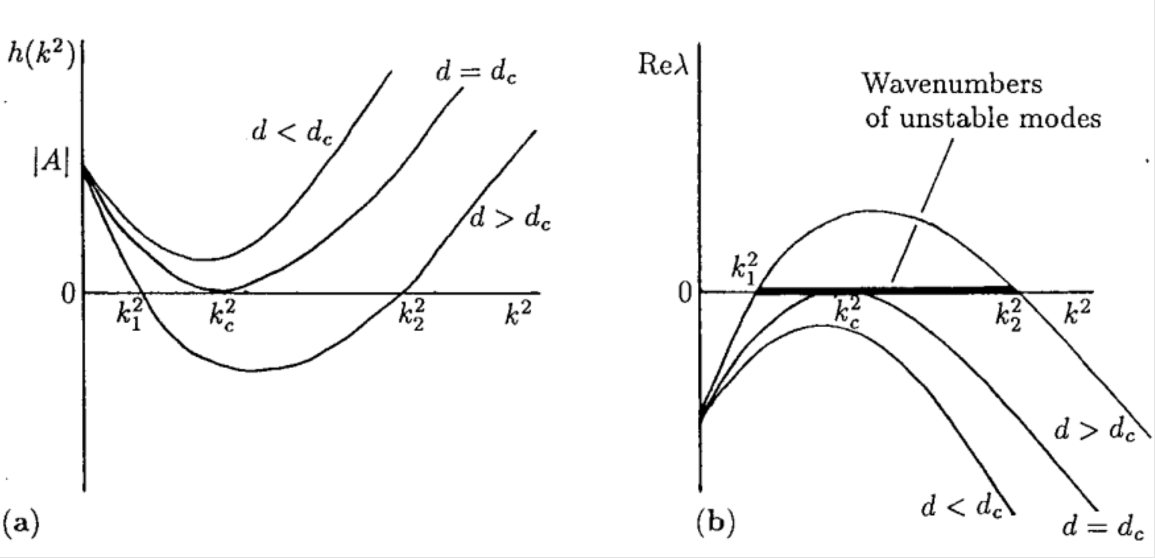


Figure 3.1: **(a)** Plot of $h(k^2)$ defined by eq. (32) for typical kinetics. When $d > d_c$, $h(k^2) < 0$ for a finite range of $k^2 > 0$. **(b)** Plot of the largest of the eigenvalues $\lambda(k^2)$ using eq. (32) as a function of k^2. When $d > d_c$, there is a range of wave numbers $k_1^2 < k^2 < k_2^2$ which are linearly unstable. Reproduced from Murray, J. D. *Mathematical Biology* 2002; Vol. II.

Remark: The onset of instability is called *bifurcation point*, when $h_{min} = 0$. $h(k^2)$ is a quadratic function of k^2 (a parabola). At this point, the discriminant is zero, that is, we have $|A| = (df_u + g_v)^2/4d$ and so for fixed kinetic parameters this defines a critical coefficient ratio $d_c(> 1)$ as the appropriate root of

$$d_c^2 f_u^2 + 2(2f_v g_u - f_u g_v)d_c + g_v^2 = 0.$$

The critical wave number k_c is then given by

$$k_c^2 = \gamma \frac{df_u + g_v}{2d} = \gamma \left[\frac{|A|}{d_c}\right]^{1/2} = \gamma \left[\frac{f_u g_v - f_v g_u}{d_c}\right]^{1/2}$$

For $d > d_c$ there are values of k for which $h(k^2) < 0$ and hence for which $\mathrm{Re}\lambda > 0$. Figure 3.1 shows the variation of $h(k^2)$ and $\mathrm{Re}\lambda$ with k^2 for various values of d.

3.5 The range of unstable modes

Whenever $h(k^2) < 0$ for a range of wave numbers k, eq. (3.17) has a positive solution λ for the same range of k. From eq. (3.17) for $d > d_c$ the range of unstable wave numbers, given by $k_1^2 < k^2 < k_2^2$, is obtained from the zeros k_1^2 and k_2^2 of $h(k^2) = 0$ as

$$\begin{aligned} k_1^2 &= \frac{\gamma}{2d}\Big[(df_u + g_v) - \big\{(df_u + g_v)^2 - 4d|A|\big\}^{1/2}\Big] < k^2 \\ &< \frac{\gamma}{2d}\Big[(df_u + g_v) + \big\{(df_u + g_v)^2 - 4d|A|\big\}^{1/2}\Big] = k_2^2. \end{aligned} \tag{3.22}$$

The expression $\lambda = \lambda(k^2)$ is called a *dispersion relation.*

Let us now examine the effect of γ on the whole process. Rewriting eq. (3.22), when $d > d_c$, as follows

$$\begin{aligned} k_1^2 &= \gamma L < k^2 < \gamma M = k_2^2, \\ L &= \frac{(df_u + g_v) - \big\{(df_u + g_v)^2 - 4d|A|\big\}^{1/2}}{2d}, \\ M &= \frac{(df_u + g_v) + \big\{(df_u + g_v)^2 - 4d|A|\big\}^{1/2}}{2d}. \end{aligned}$$

We can easily see that if we increase or decrease γ, we shift the range of possible unstable wave numbers. In finite domain there is a discrete set of wave numbers. For a fixed set of kinetic and diffusion parameters (a, b, d) lying within the Turing space, if γ is sufficiently small, then there may be no wave number k which lies in the unstable range and hence no eigenfunction will be driven unstable. On the other hand, if γ is large enough, several wave numbers may fall into the unstable range.

If we consider the solution $\boldsymbol{w}$ given by eq. (3.16), the dominant contributions as t increases are those modes for which $\mathrm{Re}\lambda(k^2) > 0$ since all other modes tend to zero exponentially. We determine the range of unstable modes $k_1^2 < k^2 < k_2^2$ as shown above, and so from eq. (3.16)

$$\boldsymbol{w}(\boldsymbol{r},t) \sim \sum_{k_1}^{k_2} c_k e^{\lambda(k^2)t} \boldsymbol{W_k}(\boldsymbol{r}) \qquad \text{for large } t \tag{3.23}$$

The fastest growing mode in this summation will be for a wave number k_m where $\mathrm{Re}\lambda(k_m^2)$ takes the largest possible value. This wave number can be located by first finding the value of k where $\mathrm{Re}\lambda(k^2)$ attains its maximum (or equivalently, where $h(k^2)$ attains its minimum), and then finding the wave number k_m closest to this value of k. Although these eigenfunctions, or spatial patterns, in eq. (3.23) are linearly unstable and grow exponentially with time (which is biologically unrealistic!!), what actually happens is that these eigenfunctions are eventually bounded by the nonlinear terms in the RD system found in the kinetics, given by the functions f and g. Hence a steady state spatially non-uniform solution (i.e. a spatial pattern) emerges. We expect the fastest growing mode to dominate and hence give the realized spatial pattern.

Remark: The conditions eq. (3.21) for DDI yield that $f_u > 0$ and $g_v < 0$. This further implies that there are two possible cases for f_v and g_u since the only restriction on these terms is that $f_v g_u < 0$. So, we can either have $f_v < 0$ and $g_u > 0$ or the other way round. These correspond to qualitatively different reactions. In the former case, u is the activator, and is also self-activating, while v is inhibitor, which inhibits both u and itself. In the latter case, u is the inhibitor, but is self-activating, while, v is the activator, and self-inhibiting. In both cases, v diffuses more quickly. Another notable point is that in the former case, concentrations of the two species are in phase, that is, both are at high or low density in the same region as the pattern grows, while in the latter case, they are out of phase, that is, u is at a high density where v is low and vice-versa. Figure 3.2 below illustrates these features. An example of the first case is the Geirer-Meinhardt kinetics.

3.6 An illustration of pattern generation

To illustrate the pattern formation here,[21] we take the simplest RD mechanism (3.5). The uniform positive steady state (u_0, v_0) is found to be

$$u_0 = a + b, \qquad v_0 = \frac{b}{(a+b)^2}, \qquad b > 0, \qquad a + b > 0 \tag{3.24}$$

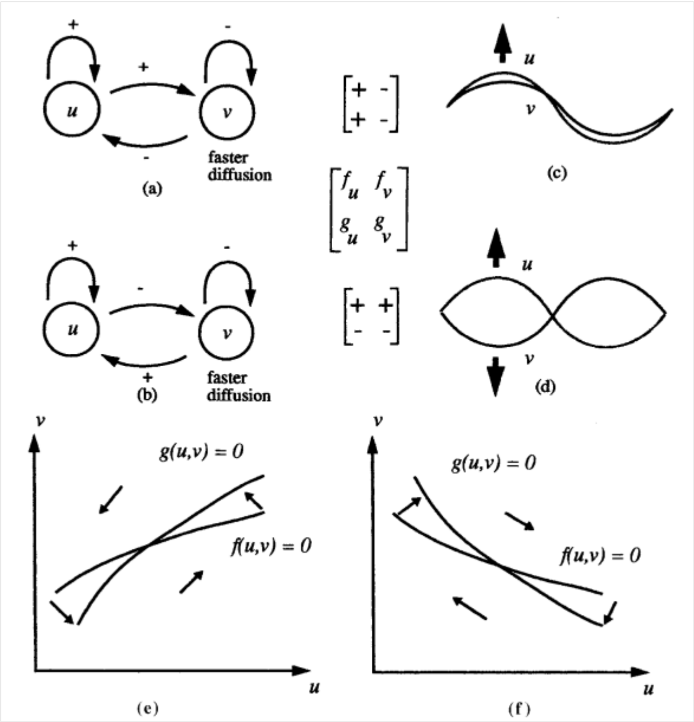


Figure 3.2: Schematic illustration of the two qualitatively different cases of diffusion driven instability. **(a)** self-activating u also activates v, which inhibits both the reactants. The resulting initially growing pattern is shown in **(c)**. **(b)** Here the self-activating u inhibits v but is itself activated by v with the resulting pattern illustrated in **(d)**. The matrices give the signs of f_u , f_v , g_u , g_v evaluated at the steady state. **(e)** and **(f)** The reaction phase planes near the steady state. The arrows indicate the direction of change due to reaction (in the absence of diffusion). Case (e) corresponds to the interactions illustrated in (a) and (c), while that in (f) corresponds to the interactions illustrated in (b) and (d). Reproduced from Murray, J. D. *Mathematical Biology* 2002; Vol. II.

and at the steady state,

$$\begin{aligned} f_u &= \frac{b-a}{a+b}, \qquad f_v = (a+b)^2, \qquad g_u = \frac{-2b}{a+b} < 0, \\ g_v &= -(a+b)^2 < 0, \qquad f_u g_v - f_v g_u = (a+b)^2 > 0. \end{aligned} \tag{3.25}$$

The condition that f_u and g_v have opposite signs implies that $b > a$. With these expressions, conditions (3.21) require

$$\begin{aligned} f_u + g_v < 0 \quad &\Rightarrow \quad 0 < b - a < (a+b)^3, \\ f_u g_v - f_v g_u > 0 \quad &\Rightarrow \quad (a+b)^2 > 0, \\ df_u + g_v > 0 \quad &\Rightarrow \quad d(a-b) > (a+b)^3, \\ (df_u + g_v)^2 - 4d(f_u g_v - f_v g_u) > 0 & \\ &\Rightarrow \quad \big[d(b-a) - (a+b)^3\big]^2 > 4d(a+b)^4. \end{aligned} \tag{3.26}$$

The above inequalities define a domain in (a, b, d) parameter space, within which some of the spatial modes of this mechanism are unstable to spatial disturbances. It is called the pattern formation space (or Turing space). We will now determine the range of wave numbers k for unstable modes.

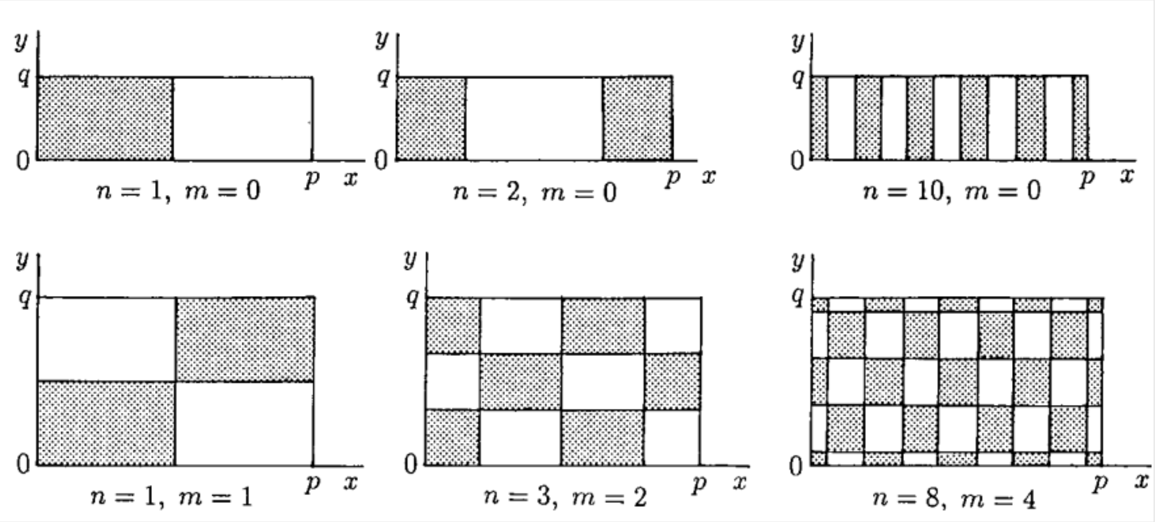


Figure 3.3: Typical two-dimensional spatial patterns indicated by the linearly unstable solution (45) when various wave numbers are in the unstable range. The shaded regions are where $u > u_0$, the uniform steady state. Reproduced from Murray, J. D. *Mathematical Biology* 2002; Vol. II.

Let us consider a two-dimensional domain defined by $0 < x < p$, $0 < y < q$. The spatial eigenvalue problem is

$$\nabla^2 \boldsymbol{W} + k^2 \boldsymbol{W} = 0, \qquad (\boldsymbol{n} \cdot \nabla)\boldsymbol{W} = 0 \qquad \text{for } (x,\, y) \text{ on } \delta B \tag{3.27}$$

where δB is the boundary of the above rectangular domain.

The eigenfunctions of the above problem are

$$\boldsymbol{W}_{p,q}(x,y) = C_{n,m}\cos\frac{n\pi x}{p}\cos\frac{m\pi y}{q}, \qquad k^2 = \pi^2\Big(\frac{n^2}{p^2}+\frac{m^2}{q^2}\Big), \tag{3.28}$$

where n and m are integers. Whenever conditions (3.26) are satisfied and a range of wave numbers k^2 (as defined in eq. (3.28)) lie within the bounds defined by eq. (3.22), then the corresponding eigenfunctions $\boldsymbol{W_{p,q}}$ are linearly unstable and they grow exponentially with time. The range of unstable wave numbers in this case is calculated from eq. (3.22), with eq. (3.25), and gives

$$\begin{aligned} &\gamma L(a,b,d) = k_1^2 < k^2 = \pi^2\Big(\frac{n^2}{p^2}+\frac{m^2}{q^2}\Big) < k_2^2 = \gamma M(a,b,d) \\ L &= \frac{[d(b-a)-(a+b)^3] - \big\{[d(b-a)-(a+b)^3]^2 - 4d(a+b)^4\big\}^{1/2}}{2d(a+b)}, \\ M &= \frac{[d(b-a)-(a+b)^3] + \big\{[d(b-a)-(a+b)^3]^2 - 4d(a+b)^4\big\}^{1/2}}{2d(a+b)}. \end{aligned} \tag{3.29}$$

Now the unstable spatially patterned solution is given by eq. (3.23) with eq. (3.28) as

$$\boldsymbol{w}(x,y,t) \sim \sum_{n,m} C_{n,m} e^{\lambda(k^2)t}\cos\frac{n\pi x}{p}\cos\frac{m\pi y}{q}, \tag{3.30}$$

where (n,m) are all those pairs which satisfy the inequality (3.29) and $\lambda(k^2)$ is the positive solution of eq. (3.17) along with eq. (3.25). A spatial pattern, initially made up of the modes in (3.30), evolves with time. Typical patterns are shown in Figure 3.3.

Chapter 4

Measured colour pattern of Passion flower

In this project we have employed RD theory, as discussed above, to understand the mechanism of pattern formation in *Passiflora incarnata*, commonly known as Passion flower. This flower, as shown in Figure 4.1 (top view), has a large number of beautiful coloured fibrils. Each fibril has alternate white and violet bands. From the top view, it can be seen that all the fibrils collectively generate concentric circles of alternate colours. But we limit ourselves to the pattern of colour bands on a single fibril.

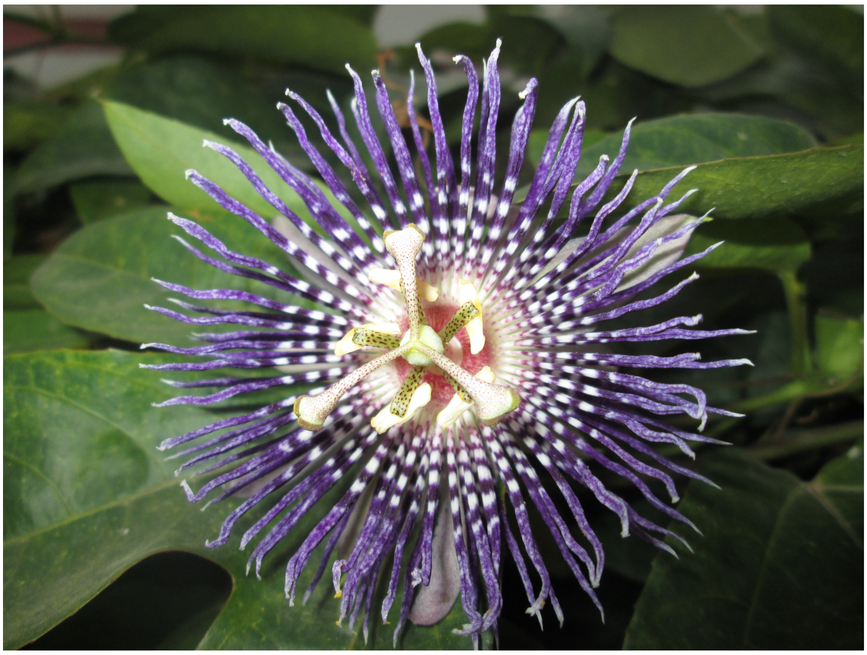


Figure 4.1: Picture of the Passion flower (top view)

Table 4.1: Measured pattern in Passion flower; W: White, V: Violet

Tip of the colour band	Distance from the centre (cm)	Width of the band (cm)
Red	1.0	1.0
W1	1.6	0.6
V1	1.8	0.2
W2	1.9	0.1
V2	2.1	0.2
W3	2.3	0.2
V3	2.6	0.3
W4	2.9	0.3
V4	3.2	0.3
W5	3.6	0.4
V5	4.1	0.5
W6	4.4	0.3
V6	5.0	0.6
W7	5.4	0.4
V7	8.8	3.4

We took a high resolution picture of a Passion flower in IISER Mohali campus (see Figure 4.1). From the coloured print out of this picture we measured the width of each successive colour band in the pattern until the tip of the fibril, using a centimeter ruler (least count = 0.1 cm), for 21 fibrils. The average width of each colour band was plotted to get the pattern shown in Figure 4.2. We also measured the actual length of 10 fibrils and calculated the average length of the fibrils so as to know the scaling factor of the measured pattern. The average actual length of a fibril is 3.1 cm. The measured average length of a fibril in the printout is 8.8 cm. Measured data are listed in Table 4.1 and a plot of the measured data is given in Figure 4.2.

$$\text{Scaling factor} = \frac{\text{Actual length}}{\text{Measured length}} = \frac{3.1}{8.8} = 0.35$$

The measured length should be multiplied by 0.35 to get the actual length of the pattern/width of bands.

An important feature of this pattern is that it is non-uniform. The width of the violet band keeps on increasing on moving towards the tip of the fibril. The width of the white bands increases initially, then decreases and eventually becomes zero on moving towards the tip of the fibril. Beyond a point, only the violet colour sustains on the fibril. It is worth mentioning here that the pattern has been measured only

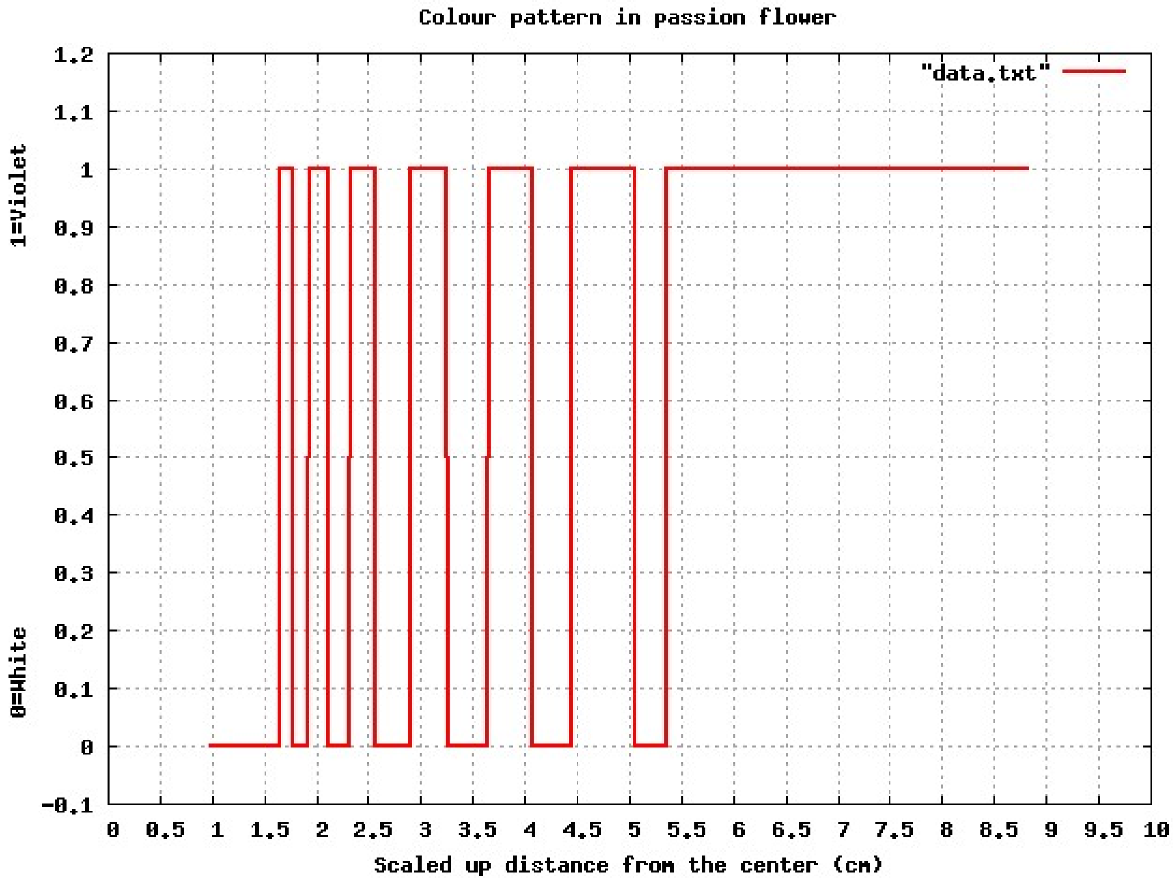


Figure 4.2: Alternation of white and violet colour bands in a Passion flower

along one dimension as it varies only along the length of the fibril. It is uniform along the angular direction.

Chapter 5

Methods and Tools

We approached our problem of understanding the mechanism of colour pattern in Passion flower, as detailed in the previous chapter, using two models: Meinhardt's model and Murray's model. We will describe both of them below one by one.

5.1 Meinhardt's model

We consider a linear array of N cells (along x-direction) in a single fibril of the Passion flower. This array of cells is assumed to be a part of its embryo, during some stage of its embryogenesis, when the chemical pre-pattern responsible for the colour bands of the fibril is to be set up through interactions (reaction and diffusion) of morphogens. We assume that there are two morphogens which diffuse through this array of cells and react with each other. It is assumed that they react according to the Gierer-Meinhardt kinetics (3.3). Thus one of them is the activator and the other is the inhibitor and their concentrations are denoted by u and v respectively. As per the RD theory, we need to evolve these concentrations with time along this array of cells, starting from a uniform steady state, and find out the conditions under which the measured pattern could be produced. Therefore, we need to solve the following set of coupled partial differential equations in one-dimension for *sufficiently* long time:[20]

$$\frac{\partial u(x,t)}{\partial t} = \frac{\rho u^2(x,t)}{v(x,t)} - \mu u(x,t) + D_u \frac{\partial^2 u(x,t)}{\partial x^2} + \rho_0 \tag{5.1a}$$

$$\frac{\partial v(x,t)}{\partial t} = \rho u^2(x,t) - \nu v(x,t) + D_v \frac{\partial^2 v(x,t)}{\partial x^2} \tag{5.1b}$$

where $\mu = k_2$, $\nu = k_5$, D_u, D_v, $\rho_0 = k_1$ and $\rho = k_3 = k_4$ are positive constants. ρ is called the source density. We determine the uniform steady state concentrations for (5.1) by analytically solving its reaction part as follows:

$$\frac{\rho u^2}{v} - \mu u + \rho_0 = 0 = \rho u^2 - \nu v \quad \Rightarrow \quad u = \frac{\nu}{\mu} + \frac{\rho_0}{\mu}, \quad v = \frac{\rho u^2}{\nu}. \tag{5.2}$$

We can infer from eq. (5.1) that if the concentrations of morphogens remain constant throughout the space in all the cells, then the change in concentrations with time, due to diffusion would be zero. Moreover, if that constant value of concentration is the steady state value, then there will be no change due to reaction part also. Thus the concentrations would not change with time.

We solve eq. (5.1) using zero flux (Neumann) boundary conditions and the initial values of concentrations $u(x, 0)$ and $v(x, 0)$ taken to be either set to steady state values given by (5.2) except for a local small perturbation about it for any one of the two morphogens at any one of the cells in the array, or as small random fluctuations about the steady state value for all the cells in the array. In both the cases a similar spatially heterogeneous pattern of concentrations is generated with time. In the former case, the pattern starts from the point of perturbation and keeps on spreading in space with time. In the latter case, a similar pattern emerges with time. However it is to be noted that the time required to get the pattern in the latter case is relatively less than in the former case. The details and the effect of various parameters on the pattern are discussed in the next chapter.

The term ρ in eq. (5.1), called source density, indicates the sensitivity towards activation and production of the activator or the inhibitor in response to their existing concentrations. In biological terms, it may be seen as certain prerequisites necessary for the synthesis of the activator and inhibitor, for instance, particular messengers or enzymes, ribosomes, ATP, or the presence of certain cell type to which the synthesis is possibly restricted. As all these properties can be non-uniform or graded in an organism giving rise to "polarity" in it, we can take source density to be spatially non-uniform or graded. If we set the initial values of concentrations $u(x, 0)$ and $v(x, 0)$ equal to the steady state value uniformly in space and choose a graded source density, spatially heterogeneous concentration pattern, similar to the previous cases, emerges. Graded source density can also be taken as random fluctuation about a constant positive value ρ.

5.2 Murray's model

We assume that two morphogens react according to the Thomas kinetics (3.2) (see also the nondimensional form in chapter 3.3). Following the RD theory, we evolve these concentrations in space with time, starting from a uniform steady state, and try to find out the values of parameters for which the measured pattern could be produced. We solve the following set of coupled partial differential equations in two-dimensions for *sufficiently* long time:[2,21]

$$\frac{\partial u}{\partial t} = \gamma\left(a - u - \frac{\rho uv}{1 + u + Ku^2}\right) + \nabla^2 u \tag{5.3a}$$

$$\frac{\partial v}{\partial t} = \gamma\left(\alpha(b - v) - \frac{\rho uv}{1 + u + Ku^2}\right) + d\nabla^2 v \tag{5.3b}$$

where γ, d, α, a, b, ρ and K are positive constants. The steady state solution of the above equation, which is obtained by equating the reaction part of the equation to zero, cannot be found analytically. So we determine it numerically using Mathematica 9.

We solve eq. (5.3) using zero flux (Neumann) boundary conditions and the initial values of concentrations $u(x, 0)$ and $v(x, 0)$ taken as small random fluctuations about the steady state value throughout the array of cells. Spatially heterogeneous concentration patterns are generated by this procedure. Varying various parameters affects the patterns. The effect of varying different parameters on the pattern and other details are discussed in the following chapter.

5.3 Numerical solution of the heat equation

As mentioned above, we have to solve eq. (5.1) and eq. (5.3) to generate the desired pattern. These equations are same as the heat equation, mentioned as eq. (2.10) in chapter 2, except for the additional coupling terms. The standard heat equation is analytically solvable with known solutions for Dirichlet, Neumann as well as mixed boundary conditions. But due to the non-linear coupling terms in our equations, they are not analytically solvable. So we resort to numerical methods for solving the equations. Out of several methods available for numerically solving the heat or diffusion eq. (2.10) we chose the Finite Difference method. A description of FD methods is provided in appendix A. We used FTCS method to solve eq. (5.1) (Meinhardt's model) and ADI method to solve eq. (5.3) (Murray's model). We wrote our own C-programs for the purpose. Refer to appendix B for C-programs.

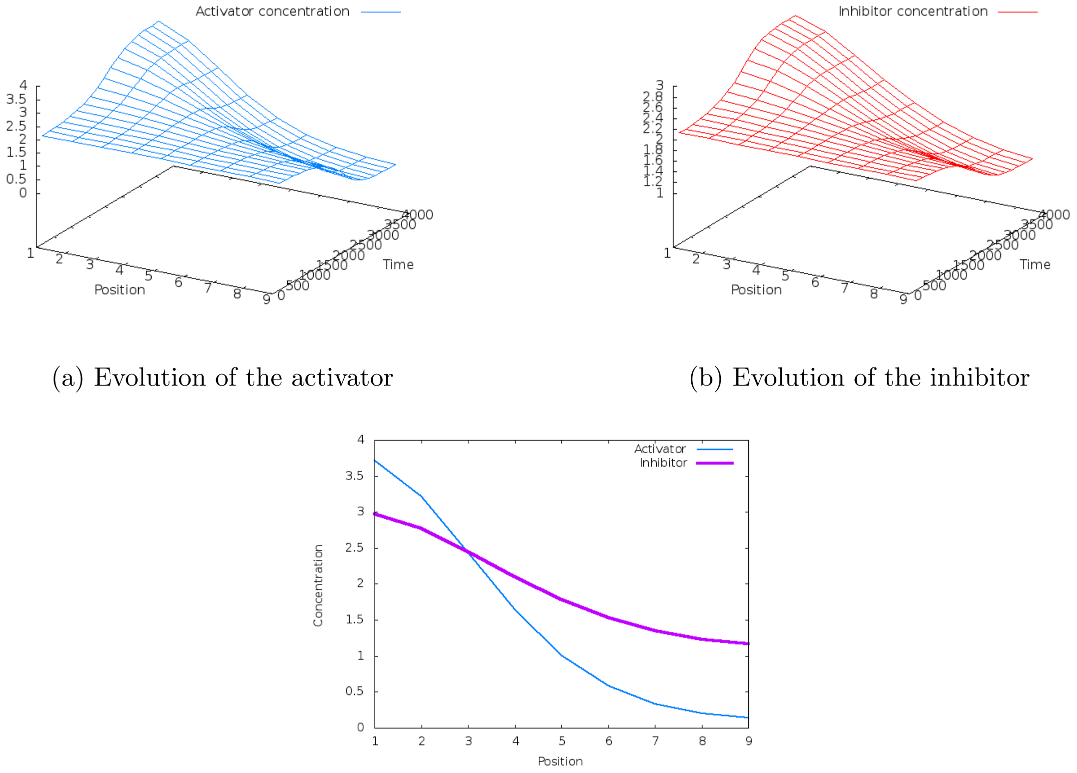
Chapter 6

Results and Discussion

6.1 Meinhardt's model

We started with reproducing a known pattern from Meinhardt's book,[20] Pg-182. We plugged in exactly the same values of parameters in our program as given there. The values of step sizes in time and space were set to 1, that is, $\triangle x = 1 = \triangle t$. The total number of timesteps is 3600. The number of cells is 9. The resultant pattern is shown in Figure 6.1, which is exactly the same as given there. We did this exercise to validate our program for further calculations. In this case, random fluctuations in the source densities along the position axis lead to heterogeneous pattern of concentrations from the initial uniform steady state.

We then studied the effect of various parameters one by one on the produced concentration pattern. We used 9 cells in the above case. When we increase the length of our array of cells to 100 keeping all other parameters same, we get the pattern shown in Figure 6.2. As we can see, on increasing the size of our domain, we get a pattern repeating along the position axis with a period of 16 cells. Thus we have a periodic pattern with a definite *wavelength.* On comparing it with the previous case, where all the parameters were the same except the number of cells, we conclude that the number of cells does not affect the wavelength. Even in the former case the wavelength is the same with half of its cycle being completed in about 8 cells. We can say that the range of unstable spatial modes is not affected by the number of cells. Nevertheless pattern can be varied by varying the number of cells with a larger number of concentration peaks appearing in the case of increased number of cells.



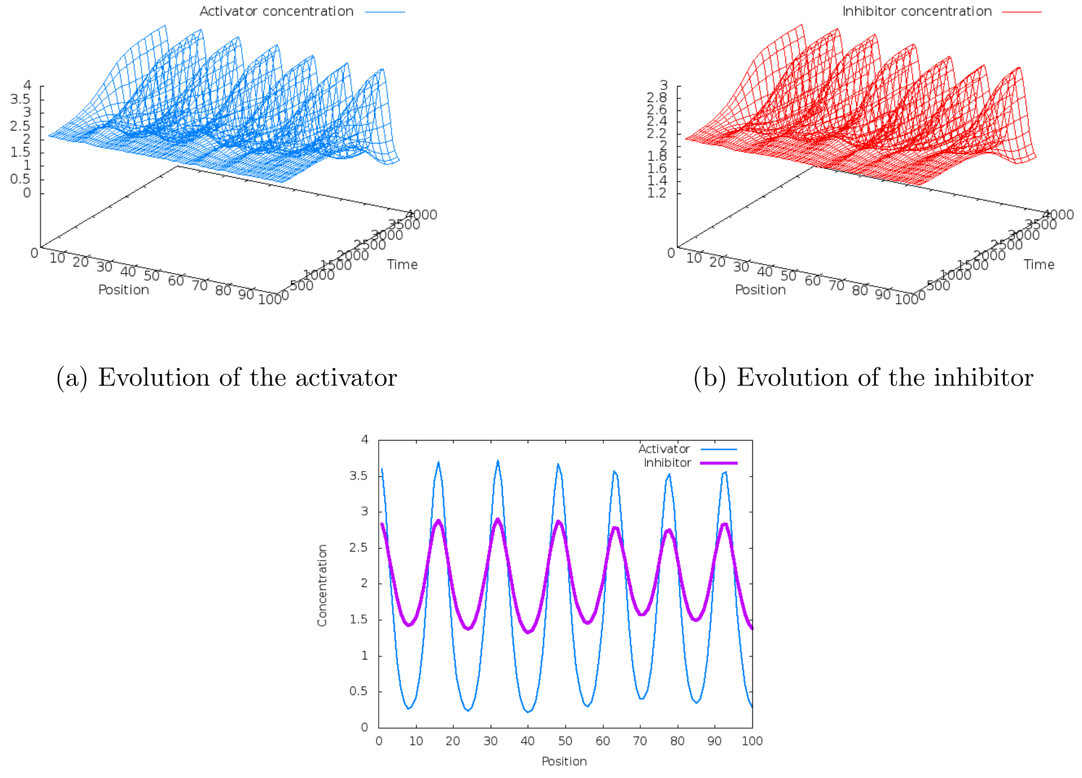
(a) Evolution of the activator

(b) Evolution of the inhibitor

(c) Final concentrations

Figure 6.1: Reproduction of the known pattern

An essential condition of pattern formation proposed by Turing[14] is that the inhibitor should diffuse faster than the activator, that is, the ratio of diffusion coefficients of inhibitor to activator should be greater than 1. The effect of this ratio on the pattern generated is worth studying. In the Meinhardt's theory of local activation and long range inhibition, a small local increase in activator concentration in a homogeneous mixture of such substances, due to random fluctuations, gets amplified by activator autocatalysis, leading to increased concentrations of both the activator and the inhibitor. Due to a quick diffusion of the inhibitor, it suppresses the activator production outside the activation centre. Thus the activator concentration increases at the activation centre upto a maximum (until stabilized by a limiting factor) and is limited to the normal level outside the activated centre, leading to stable concentration pattern in space. In the case shown in Figure 6.1, this ratio is taken to be 20. From the theory described above we expect that the higher the value of this ratio, sharper and more localized would be the peaks of activator and inhibitor concentrations. Final concentration patterns for the diffusion ratio values 40 and 80 are



(a) Evolution of the activator

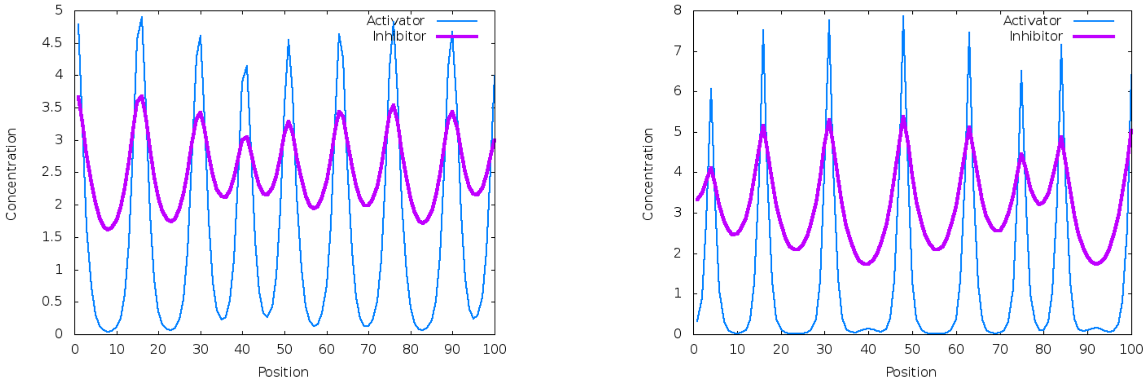
(b) Evolution of the inhibitor

(c) Final concentrations

Figure 6.2: Concentration pattern in extended array of cells

shown in Figure 6.3. On comparing them with those of Figure 6.1 we find that higher diffusion ratios lead to sharper and more localized concentration peaks.

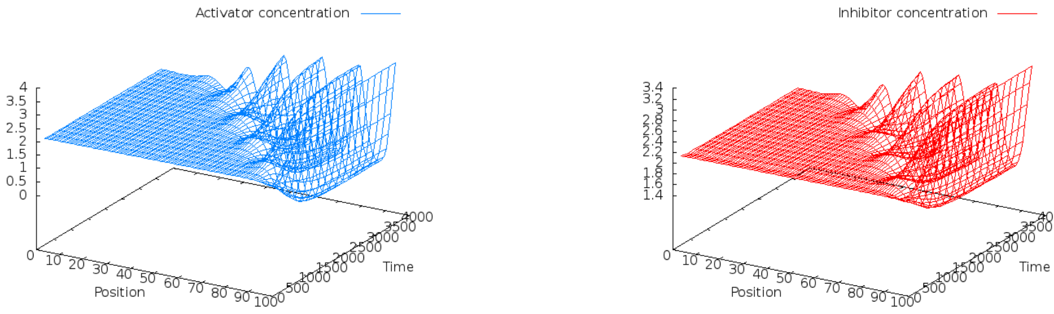
In all the cases discussed until now, the initiation of pattern from the uniform steady state occurred by random fluctuations in source densities along the position. We now discuss other ways in which the pattern formation can be initiated from uniform steady state. We take source density to be uniform all over the space and then manually increase its value at one end of the array of cells. In this present case, we set source density to 0.0102 in all the cells and then increase it by 0.0020 in cell number 100. The computed patterns are shown in Figure 6.4. We see that the pattern gets initiated by the local small increment in the source density and spreads from there towards other cells with time. Figure 6.5 shows the final concentrations after different number of time steps. We can see that the pattern keeps on spreading with time.



(a) D_v/D_u=40

(b) D_v/D_u=80

Figure 6.3: Effect of varying the ratio of diffusion coefficients of v to u

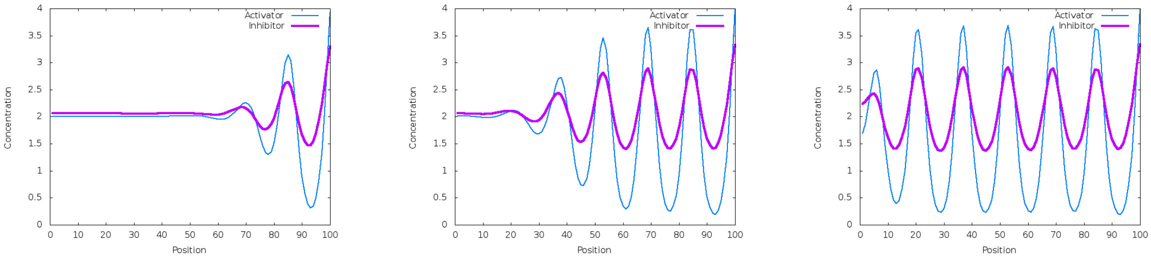


(a) Evolution of the activator

(b) Evolution of the inhibitor

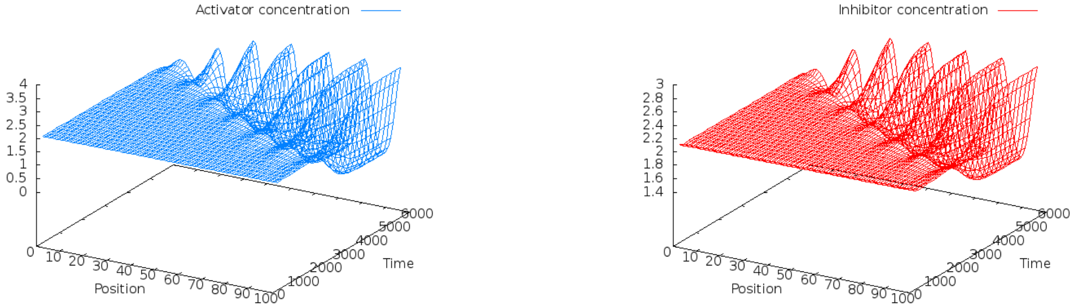
Figure 6.4: Concentration pattern with locally increased source density at one end while otherwise uniform; concentrations u and v are taken to be uniform everywhere

Another way of initiating pattern formation from the uniform steady state concentrations is to locally increase one of the concentrations in one of the cells. If we take both the activator and inhibitor concentrations as uniform all over the space and then locally increase one of them by a small amount, the pattern formation gets initiated and the pattern spreads from the point of disturbance with time. In the case shown in Figure 6.6, we have taken the source density and inhibitor concentration to be uniform all over the space, set equal to 0.0102 and 2.02005 (steady state value) respectively. The activator concentration is set to 2.01 (steady state value) except at cell number 100 where it is increased by 0.02. We see that the pattern originates at the site of local perturbation in concentration and spreads in space from that point with time. Figure 6.7 shows the time evolution of final concentration patterns in this case. We



(a) Timesteps=1600 (b) Timesteps=3600 (c) Timesteps=5600

Figure 6.5: Evolution of final pattern initiated by local small increment in source density at one end.



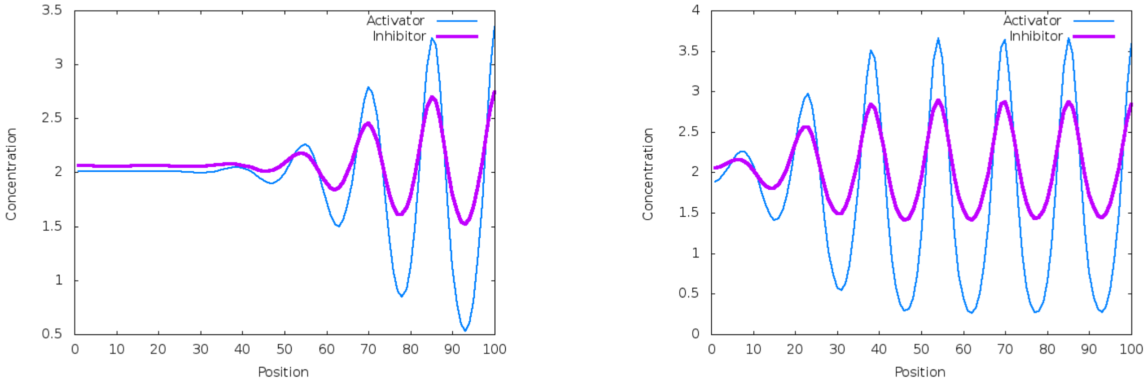
(a) Evolution of the activator (b) Evolution of the inhibitor

Figure 6.6: Pattern initiated by local perturbation in activator concentration at one end in space from otherwise uniform steady state

observe that if the activator concentration is perturbed at the middle point in the array of cells instead of at the end point, then the pattern spreads in both directions as shown in Figure 6.8a. When it is perturbed at both ends then the pattern changes from both sides towards the center and finally overlaps as shown in Figure 6.8b.

6.2 Murray's model

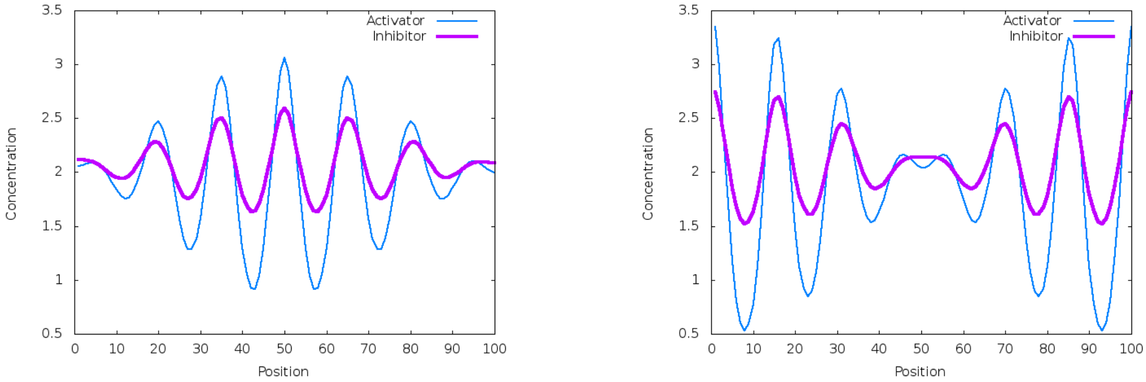
Using this model, we first of all reproduced the known pattern published by Murray,[2] Pg-172, so as to validate our program for further analysis. We chose a rectangular domain with x varying from 0 to 3 and y from 0 to 1. Grid size along both the directions was taken to be 0.01, that is $\triangle x = 0.01 = \triangle y$. Timestep $\triangle t$ was taken to be 0.001. All other parameters were exactly the same as chosen by Murray. Steady state concentrations were found numerically using Mathematica 9. Initial conditions



(a) Timesteps=3600

(b) Timesteps=5600

Figure 6.7: Evolution of final pattern initiated by local small increment in activator concentration at one end.



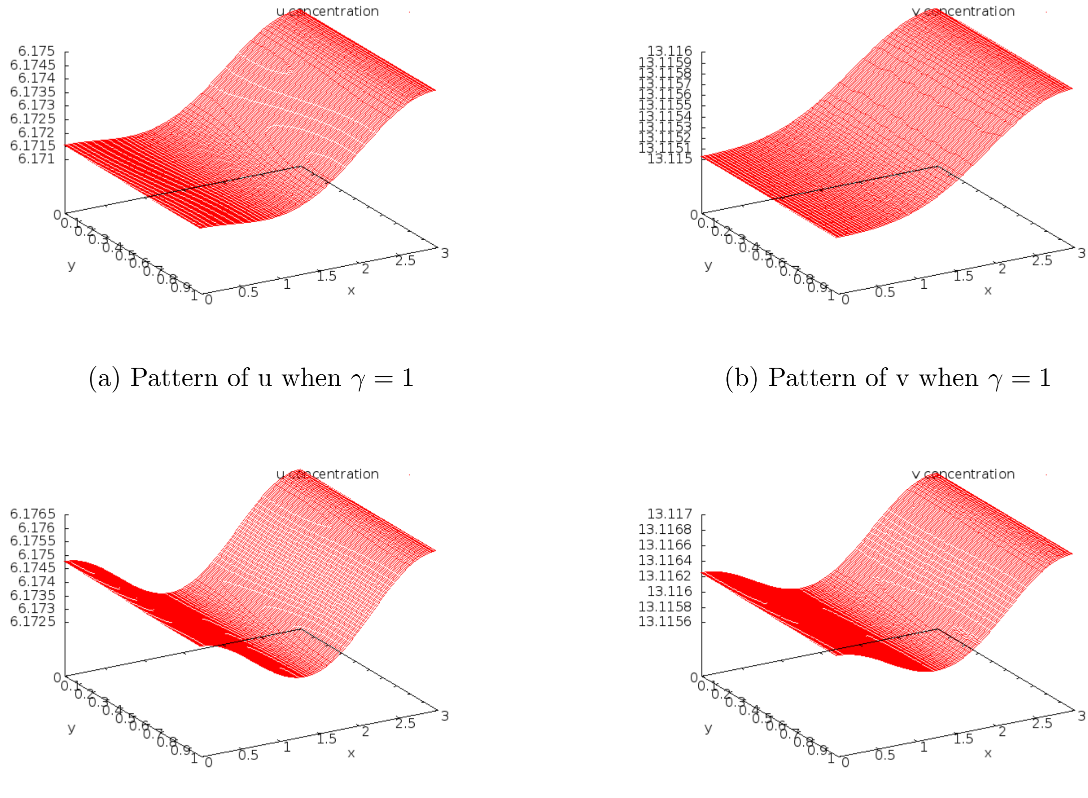
(a) Perturbation at cell number 50

(b) Perturbation at cell number 1 and 100

Figure 6.8: Patterns by initially perturbing activator concentration at different points in space and their combinations.

of concentrations were taken to be random perturbations (±0.5) about the steady state values for both u and v. Figure 6.9 shows the pattern produced by our program for these conditions, which match the published patterns.

The effect of the parameter γ on pattern formation is illustrated here. Keeping all other parameters constant, we vary γ. Figure 6.10 shows the computed patterns for different values of γ. As we increase γ the range of unstable modes increases as dictated by eq. (3.22). Therefore, more and more modes become unstable and the final concentration pattern is a result of their combination as is evident in Figure 6.10.



(a) Pattern of u when $\gamma = 1$

(b) Pattern of v when $\gamma = 1$

(c) Pattern of u when $\gamma = 4$

(d) Pattern of v when $\gamma = 4$

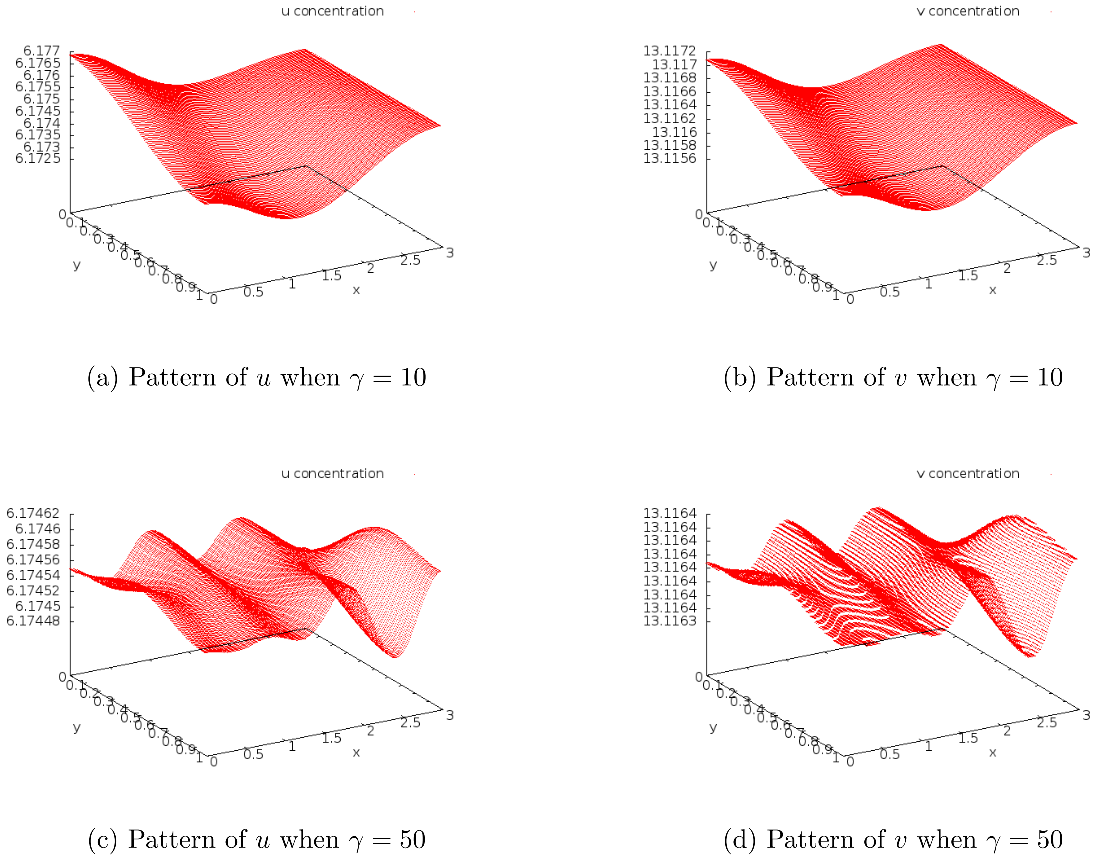
Figure 6.9: Known patterns reproduced

6.3 Fourier Analysis of the measured pattern

We performed Fourier Analysis of the measured pattern as shown in Figure 4.2 to get an idea of the frequencies which have major contribution in our pattern. Due to zero flux boundary conditions used to solve the equations, our solutions are always of the form of cosine functions or their linear combinations. So we found the Discrete Cosine Transform of the measured pattern. The average length of a fibril is 8.8 cm. We discretised this dimension with $\triangle x$ = 0.01 cm. Thus $x_j = j \triangle x$, where j = 1 to N (N=880). We assign the function $f(x_j)$ defining the pattern as follows:

$$f_{violet/white}(x_j) = \begin{cases} 1 & : \text{colour of petal is violet/white at } x_j \\ 0 & : \text{otherwise} \end{cases} \tag{6.1}$$

For the above discrete pattern $f(x_j)$, the coefficients of various frequencies are



(a) Pattern of u when $\gamma = 10$

(b) Pattern of v when $\gamma = 10$

(c) Pattern of u when $\gamma = 50$

(d) Pattern of v when $\gamma = 50$

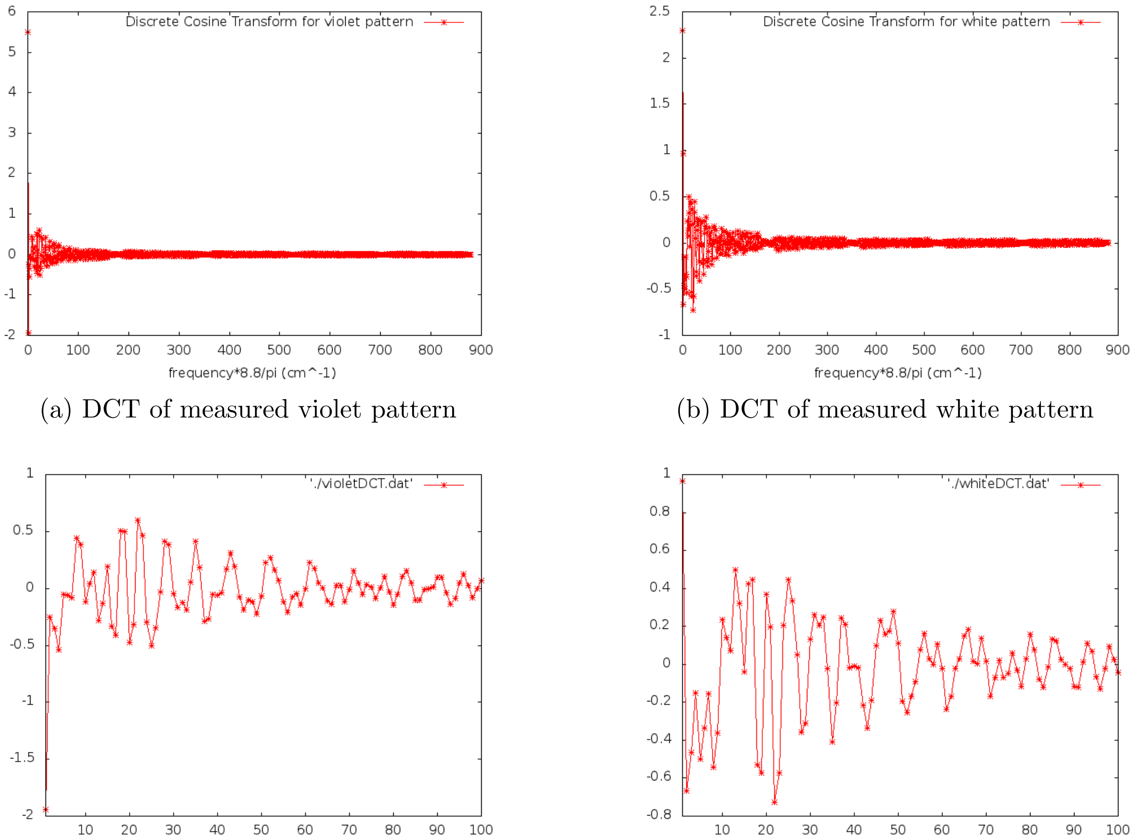
Figure 6.10: Effect of parameter γ on the pattern formation

calculated as follows:

$$F_k = \triangle x \sum_{j=1}^{N} f(x_j) \cos\left(\frac{\pi(j-\frac{1}{2})k}{N}\right) \tag{6.2}$$

for $k = 0$ to N-1. F_k is the coefficient of frequency $\omega_k = \dfrac{\pi k}{N \triangle x} = \dfrac{\pi k}{8.8}$.
The C-program used to find the DCT is given in appendix B. Following are the DCTs of both white and violet patterns:

The frequencies are obtained by multiplying x-coordinates in Figure 6.11 by $\dfrac{\pi}{8.8} cm^{-1}$. We can see that the zero frequency in both the DCTs above have the largest contributions. There are 19 frequencies in case of violet pattern and 23 frequencies in that of white pattern, with k in the range $0 \leq k \leq 43$ and $|F_k| \geq 0.3$. We now need to find a set of parameter values such that the range of unstable modes given by eq. (3.22) encompasses the above set of frequencies.



(a) DCT of measured violet pattern

(b) DCT of measured white pattern

(c) Zoomed out DCT of violet pattern

(d) Zoomed out DCT of white pattern

Figure 6.11: Fourier Analysis of the measured pattern

Chapter 7

Conclusions and Outlook

We discussed in detail two different models of reaction kinetics to be employed in the RD equations. Both the models have been successfully applied to generate patterns observed in nature. We also applied both the models on our system to get some patterns. We have not yet been able to generate our measured pattern as it is not a trivial one. It consists of stripes of two different colours but with non-uniform widths, which is a feature not encountered in the literature so far. We performed Fourier analysis on our measured pattern and as expected it turned out to be a combination of a large number of cosine functions with closely spaced frequencies. Since it is a non-trivial pattern, we must think of a non-trivial way to address the problem.

We have not yet dealt with time/space varying kinetic and diffusion parameters. Based on our experience so far this is one of the possible solutions to generate striped patterns of non-uniform width. For instance, we saw in the previous chapter that on increasing the value of the ratio of diffusion coefficients the concentration peaks become more localized and sharp. So if we vary this parameter, then the widths of concentration peaks are expected to vary in space. Similarly we can try varying some kinetic parameters like the relative strengths of activation and inhibition or the source density. In the previous chapter, we saw in Murray's case that on increasing γ the computed pattern was a combination of several modes, the reason being that the range of unstable frequencies was expanded to an extent that several normal modes fell into it. Thus another possibility to generate non-uniform pattern is to choose the parameter values such that the range of unstable frequencies is sufficiently wide to encompass the required normal modes. We saw earlier that γ can be interpreted as a measure of the size of the domain. Thus we can study the patterns in growing domain by taking γ varying with time. This is another method which could yield non-uniform pattern.

We plan to continue our work to explore the above mentioned possibilities and unravel the untouched part of RD theory. We first need to properly analyse their effects and come up with a mathematical framework for the same. Then using it we hope to be able to logically choose the parameter values and vary them accordingly with time and/or space so as to arrive at our desired pattern.

Bibliography

[1] Johnson, B. R.; Scott, S. K. *Chem. Soc. Rev.* **1996**, *265*.

[2] Murray, J. D. *J. Theo. Bio.* **1981**, *88*, 161–199.

[3] Meinhardt, H. *Int. J. Dev. Bio.* **2012**, *56*, 447–462.

[4] Field, R. J., Burger, M., Eds. *Oscillations and Travelling Waves in Chemical Systems*; Wiley, New York, 1985.

[5] Field, R. J.; Körös, E.; Noyes, R. M. *J. Am. Chem. Soc.* **1972**, *94*, 8649.

[6] Zaikin, A. N.; Zhabotinskii, A. M. *Nature* **1970**, *225*, 535–537.

[7] Winfree, A. T. *Science* **1972**, *175*, 634–636.

[8] Winfree, A. T. *Scientific American* **1974**, *230*, 82–95.

[9] Müller, S. C.; Plesser, T.; Hess, B. *Science* **1985**, *230*, 661–663.

[10] Welsh, B. J.; Gomatam, J.; Burgess, A. E. *Nature* **1983**, *340*, 611–614.

[11] Zykov, V. S. *Simulation of Wave Processes in Excitable Media*; Manchester University Press, 1987.

[12] Wolpert, L. *J. Theo. Bio.* **1969**, *25*, 1–47.

[13] Bard, J. *J. Theo. Bio.* **1981**, *93*, 363–385.

[14] Turing, A. M. *Phil. Trans. of Roy. Soc.(London) B* **1952**, *295*, 451.

[15] Prigogine, I.; Lefever, R. *J. Chem. Phys.* **1968**, *48*, 1695–1700.

[16] Glandsdorff, P.; Prigogine, I. *Thermodynamic theory of structure, stability and fluctuations*; Wiley, New York, 1971.

[17] Nicolis, G.; Prigogine, I. *Self-organization in nonequilibrium systems*; Wiley, New York, 1977.

[18] Zhabotinskii, A. M. In *Oscillating processes in biological and chemical systems*; Frank, G. M., Ed.; Science Publications, Moscow, 1962.

[19] Winfree, A. T. *Science* **1973**, *181*, 937–939.

[20] Meinhardt, H. *Models of biological pattern formation*; Academic Press, New York, 1982.

[21] Murray, J. D. *Mathematical biology*; Springer, Berlin, Heidelberg, 2002; Vol. II.

[22] Kernevez, J. P.; Joly, G.; Duban, M. C.; Bunow, B.; Thomas, D. *J. Math. Bio.* **1979**, *7*, 41–56.

[23] Murray, J. D. *Bull. Math. Bio.* **1990**, *52*, 119–152.

[24] Maini, P. K. *C. R. Biologies* **2004**, *327*.

[25] Murray, J. D. *Interface Focus* **2012**,

[26] Castets, V.; Dulos, E.; Boissonade, J.; De Kepper, P. *Phys. Rev. Letters* **1990**, *64*, 2953–2956.

[27] De Kepper, P.; Epstein, I. R.; Kustin, K.; Orbàn, M. *J. Phys. Chem.* **1982**, *86*, 170–171.

[28] Lee, K. J.; McCormick, D.; Ouyang, Q.; Swinney, H. L. *Science* **1993**, *261*, 192–194.

[29] Szalai, I.; Cuiñas, D.; Takàcs, N.; Horvàth,; De Kepper, P. *Interface Focus* **2012**, *2*, 417–432.

[30] Ouyang, Q.; Swinney, H. L. *Nature* **1991**, *352*, 610.

[31] Gunaratne, G. H.; Ouyang, Q.; Swinney, H. L. *Phys. Rev. E* **1994**, *50*, 2802.

[32] Schnackenberg, J. *J. Theo. Bio.* **1979**, *81*, 389–400.

[33] Thomas, D., Kernezev, J. P., Eds. *Analysis and Control of Immobilized Enzyme Systems*; Springer-Verlag, Berlin-Heidelberg-New York, 1975.

[34] Gierer, A.; Meinhardt, H. *Kybernetik* **1972**, *12*, 30–39.

[35] Murray, J. D. *Mathematical Biology*; Springer, Berlin, Heidelberg, 2002; Vol. I.

[36] `http://www.resnet.wm.edu/~jxshix/math490/lecture-chap1.pdf`.

[37] Fick, A. *Phil. Mag.* **1855**, *10*, 30.

[38] Kapral, R. *Physica D* **1995**, *86*, 149.

[39] Lengyel, I.; Epstein, I. R. *Proc. Natl. Acad. Sci. USA* **1992**, *89*, 3977.

[40] Lengyel, I.; Epstein, I. R. *Science* **1991**, *251*, 650.

Appendices

Appendix A

Finite Difference methods

Consider a rectangular domain D: $0 \leq x \leq a$ and $0 \leq y \leq b$. Draw straight lines parallel to x-axis and y-axis as shown in the figure A.1 such that $x_i = i * \triangle x$ for $i =$ 1, 2, 3, $\cdots$ n-1 and $y_j = j * \triangle y$ for $j =$ 1, 2, 3, $\cdots$ m-1 where $\triangle x$ and $\triangle y$ are small positive steplengths obtained by $\triangle x = \dfrac{a}{n}$ and $\triangle y = \dfrac{b}{m}$.

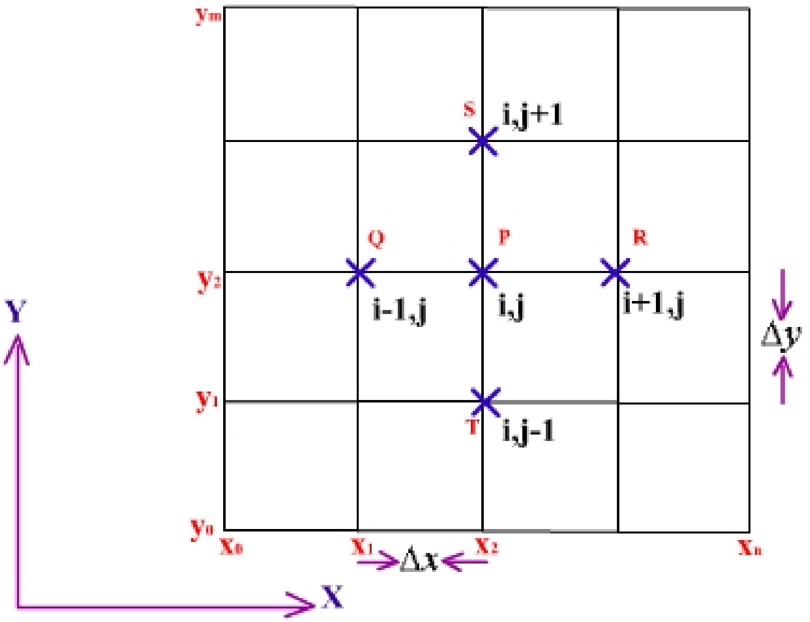


Figure A.1: Discretised rectangular domain

Let $P_{i,j} = P(x_i, y_j)$ be any point in the region D then the co-ordinates x_i and y_j can be obtained by

$$\begin{aligned} x_i &= x_0 + i * \triangle x \\ y_j &= y_0 + j * \triangle y, \end{aligned} \tag{A.1}$$

where (x_0, y_0) are the coordinates of the left bottom most point of the rectangle, that is (0,0) in the present case. If $u(x, y)$ is any continuous function with all necessary derivatives existing in D then

$$u_{i\pm1,j} = u(x_i \pm \triangle x, y_j) = u_{i,j} \pm \triangle x \frac{\partial u_{i,j}}{\partial x} + \frac{\triangle x^2}{2!} \frac{\partial^2 u_{i,j}}{\partial x^2} \pm \cdots\cdots \tag{A.2a}$$

$$u_{i,j\pm1} = u(x_i, y_j \pm \triangle y) = u_{i,j} \pm \triangle y \frac{\partial u_{i,j}}{\partial y} + \frac{\triangle y^2}{2!} \frac{\partial^2 u_{i,j}}{\partial y^2} \pm \cdots\cdots \tag{A.2b}$$

From eq. (A.2), partial derivatives can be approximated as follows:

$$\begin{aligned}
\frac{\partial u_{i,j}}{\partial x} &= \frac{u_{i+1,j} - u_{i-1,j}}{2 \triangle x} + O(\triangle x^2), \qquad \text{(Central difference)} \\
&= \frac{u_{i+1,j} - u_{i,j}}{\triangle x} + O(\triangle x), \qquad \text{(Forward difference)} \\
&= \frac{u_{i,j} - u_{i-1,j}}{\triangle x} + O(\triangle x). \qquad \text{(Backward difference)} \\
\frac{\partial u_{i,j}}{\partial y} &= \frac{u_{i,j+1} - u_{i,j-1}}{2 \triangle y} + O(\triangle y^2), \qquad \text{(Central difference)} \\
&= \frac{u_{i,j+1} - u_{i,j}}{\triangle y} + O(\triangle y), \qquad \text{(Forward difference)} \\
&= \frac{u_{i,j} - u_{i,j-1}}{\triangle y} + O(\triangle y). \qquad \text{(Backward difference)} \\
\frac{\partial^2 u_{i,j}}{\partial x^2} &= \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\triangle x^2} + O(\triangle x^4). \\
\frac{\partial^2 u_{i,j}}{\partial y^2} &= \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\triangle y^2} + O(\triangle y^4).
\end{aligned} \tag{A.3}$$

Making use of these approximations to replace partial derivatives, the partial differential equations are converted into difference equations and the resultant system of algebraic equations are solved using any direct or iterative methods. Since the analytical methods for finding solution of second order partial differential equations depend on the type of PDE, the numerical schemes also depend on the type of PDE. The one dimensional heat equation as shown below is an example of a parabolic PDE.

$$\frac{\partial u}{\partial t} = C \frac{\partial^2 u}{\partial x^2} \tag{A.4}$$

A.1 Forward Time and Central Space (FTCS) Scheme

In this method the time derivative term in the one-dimensional heat eq. (A.4) is approximated with forward difference and space derivatives are approximated with second order central differences. This gives

$$\frac{u_i^{n+1} - u_i^n}{\triangle t} = C\frac{u_{i-1}^n - 2u_i^n + u_{i+1}^n}{\triangle x^2} \tag{A.5}$$

where $x_i = i \triangle x$ $(i = 0,\ 1,\ 2,\ 3,\ \cdots\ N)$ and $t_n = n \triangle t$ $(n = 0,\ 1,\ 2,\ 3,\ \cdots\cdots)$. To distinguish between space and time coordinates superscript index n is used for the time coordinate whereas a subscript i is used to represent the space position along x direction. N is the number of points along the x-direction excluding zeroth point.

At any typical node $(i,\ n)$, the finite difference eq. (A.5) can be rearranged as

$$u_i^{n+1} = u_i^n + r(u_{i-1}^n - 2u_i^n + u_{i+1}^n) \tag{A.6}$$

where $r = \dfrac{c \triangle t}{\triangle x^2}$. It gives a formula to compute the unknown concentrations in the domain at various positions at various times. For n=1, the unknown u is first calculated using the initial conditions at t=0 and boundary values at x=0 and x=L (where L is the length of the domain). Once the solution at time step 1 is obtained, the solution at n=2 is calculated in the same manner by making use of the solution at n=1 and the boundary conditions at x=0 and x=L. The same procedure is repeated until the solution reaches a steady state or until the desired time step.

Since eq. (A.6) has only one unknown for any i and n, it is called an explicit scheme. The FTCS scheme is illustrated in Figure A.2.

This method is known to be numerically stable and convergent only when $r \leq 0.5$. The numerical errors are proportional to the size of the time step and the square of the space step: $\triangle u = O(\triangle t) + O(\triangle x^2)$. Thus it is said to be first order accurate in time and second order accurate in space.

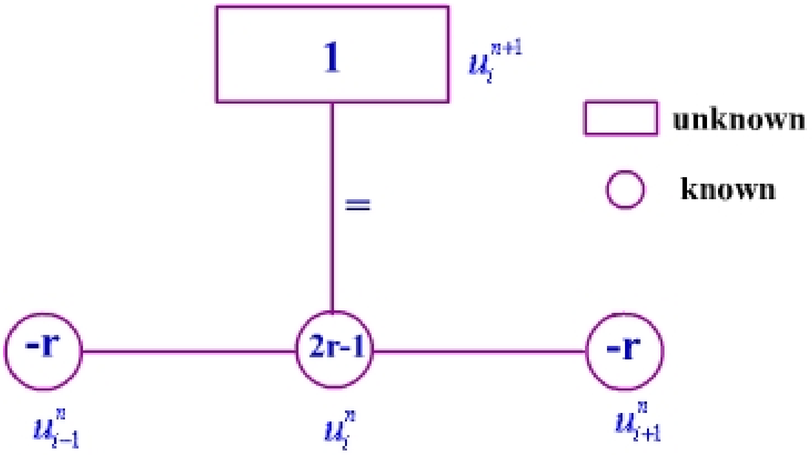


Figure A.2: Sketch for the FTCS scheme

A.2 Backward Time Central Space (BTCS) scheme

If the forward difference approximation for time derivative in the one dimensional heat eq. (A.4) is replaced with the backward difference and the central difference approximation for space derivative is used, then eq. (A.4) can be written as

$$\frac{u_i^n - u_i^{n-1}}{\triangle t} = C\frac{u_{i-1}^n - 2u_i^n + u_{i+1}^n}{\triangle x^2} \tag{A.7}$$

where $i = 1, 2, 3, \cdots N$ and $n = 1, 2, 3, \cdots\cdots$
Alternatively,

$$\frac{u_i^{n+1} - u_i^n}{\triangle t} = C\frac{u_{i-1}^{n+1} - 2u_i^{n+1} + u_{i+1}^{n+1}}{\triangle x^2} \tag{A.8}$$

for $i = 1, 2, 3, \cdots N$ and $n = 0, 1, 2, \cdots\cdots$
Rearranging the above equation, we obtain

$$\begin{aligned} u_i^{n+1} - u_i^n &= r(u_{i-1}^{n+1} - 2u_i^{n+1} + u_{i+1}^{n+1}) \\ \Rightarrow \quad -ru_{i-1}^{n+1} + (1+2r)u_i^{n+1} - ru_{i+1}^{n+1} &= u_i^n \end{aligned} \tag{A.9}$$

for $i = 1, 2, 3, \cdots N$ and $n = 0, 1, 2, \cdots\cdots$

Since there are three unknown terms in eq. (A.9), the scheme so obtained is referred to as an implicit method. The main drawback of having more than one unknown coefficient in any equation, unlike FTCS method, is that the value of the dependent variable at any typical node say $(i,\, n)$ cannot be obtained from a single finite difference equation of the node $(i,\, n)$, and one has to generate a system of equations for each time step separately by varying i. Then for each time step there will be a system of equations equivalent to the number of unknowns in that time step (say N in the present case). This linear system of algebraic equations in N unknowns has to be solved to obtain the solution for each time step . This process has to be repeated until the desired time step is reached. The scheme (A.9) is called the fully implicit method.

This scheme is unconditionally numerically stable and convergent. The errors are linear in the time step size and quadratic in space step: $\triangle u = O(\triangle t) + O(\triangle x^2)$. Therefore, it is first order accurate in time and second order accurate in space.

A.3 Crank-Nicholson

Schemes (A.5) and (A.8) are two different methods to solve the one dimensional heat equation (A.4). Crank-Nicholson scheme is obtained by taking an average of these two schemes, that is

$$\frac{u_i^{n+1} - u_i^n}{\triangle t} = \frac{C}{2}\left[\frac{u_{i-1}^{n+1} - 2u_i^{n+1} + u_{i+1}^{n+1}}{\triangle x^2} + \frac{u_{i-1}^{n} - 2u_i^{n} + u_{i+1}^{n}}{\triangle x^2}\right]$$

for $i = 1,\, 2,\, 3,\, \cdots\, N$ and $n = 0,\, 1,\, 2,\, \cdots\cdots$

$$\Rightarrow \quad -\frac{r}{2}u_{i-1}^{n+1} + (1+r)u_i^{n+1} - \frac{r}{2}u_{i+1}^{n+1} = \frac{r}{2}u_{i-1}^{n} + (1-r)u_i^{n} + \frac{r}{2}u_{i+1}^{n} \tag{A.10}$$

Since more than one unknown is involved for each i in eq. (A.10) Crank-Nicholson scheme is also an implicit scheme. Therefore, one has to solve a system of linear algebraic equations for every time step to get the field variable u. The Crank-Nicholson scheme is illustrated in Figure A.3.

The linear algebraic system of equations generated by the Crank-Nicholson method for the time step t^{n+1} are sparse because the finite difference equation obtained at any space node, say i and at time step t^{n+1} has only three unknown coefficients involving space nodes i-1 , i and i+1 at t^{n+1}. In matrix notation, these equations can be written as **AU=B** , where **U** is the unknown vector of order N at any time level t^{n+1} , **B**

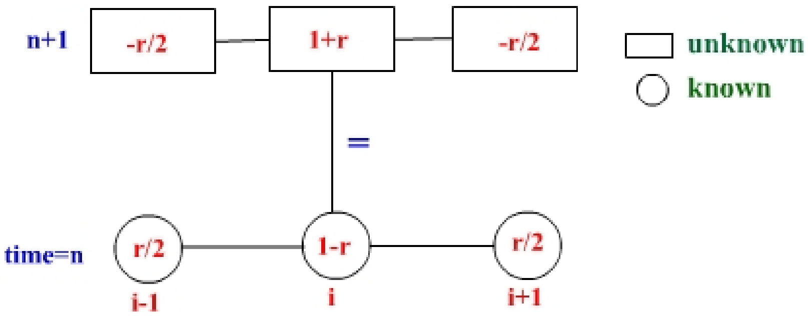


Figure A.3: Sketch for the Crank-Nicholson scheme

is the known vector of order N, which involves the values of **U** at the n^{th} time step and **A** is the coefficient square matrix of order N × N with a tri-diagonal structure as follows:

$$\mathbf{A} = \begin{bmatrix} b_1 & c_1 & 0 & 0 & \cdots & 0 & 0 \\ a_2 & b_2 & c_2 & 0 & \cdots & 0 & 0 \\ 0 & a_3 & b_3 & c_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & a_{N-1} & b_{N-1} & c_{N-1} \\ 0 & 0 & \cdots & 0 & 0 & a_N & b_N \end{bmatrix}$$

Such a matrix is called a tri-diagonal matrix and the system of equations with tridiagonal coefficient matrix is called tridiagonal system. Though direct solvers like Gauss elimination and LU decomposition can be used to solve these systems there are some special schemes available to solve tridiagonal systems. One of them is Thomas algorithm which exploits the tridiagonal nature of the coefficient matrix. Thomas algorithm is similar to Gauss elimination. However, the novelty in the method is that the forward elimination and back substitution parts of Gauss elimination are used only for the non-zero positions of the system **AU=B**.

This scheme is unconditionally numerically stable and convergent. The errors are quadratic over both the time step and the space step: $\triangle u = O(\triangle t^2) + O(\triangle x^2)$. Thus it is second order accurate in both time and space.

A.4 Thomas algorithm for tridiagonal system of equations

Let us call the three non-zero diagonals of the above coefficient matrix **A** as a, b and c, where b is the element of the principal diagonal, a is the element of the diagonal before the principal diagonal with zero as the first element and c is the element of the diagonal that lies after the principal diagonal with a zero as the last element. Then the order of a, b and c is equal to the number of unknowns for any time step with the known vector B (with elements d_i). Then the Thomas algorithm can be written as:

```
Do $i = 2$ to $N$ (if $N$ is the number of unknowns)
        $b_i = b_i - a_i \frac{c_{i-1}}{b_{i-1}}$
        $d_i = d_i - a_i \frac{d_{i-1}}{b_{i-1}}$
end Do
        $u_N = \frac{d_N}{b_N}$
    Do $i = N$-1 to 1
        $u_i = \frac{d_i - c_i u_{i+1}}{b_i}$
    end Do
```

A.5 Alternate Direction Implicit method

The heat equation in two dimensions is as follows:

$$\frac{\partial u}{\partial t} = C\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{A.11}$$

Applying Crank-Nicholson scheme to the above equation we get:

$$\begin{aligned}
\frac{u_{i,j}^{n+1}-u_{i,j}^{n}}{\triangle t} &= \frac{C}{2}\left[\frac{u_{i-1,j}^{n+1}-2u_{i,j}^{n+1}+u_{i+1,j}^{n+1}}{\triangle x^2}+\frac{u_{i-1,j}^{n}-2u_{i,j}^{n}+u_{i+1,j}^{n}}{\triangle x^2}\right]\\
&+\frac{C}{2}\left[\frac{u_{i,j-1}^{n+1}-2u_{i,j}^{n+1}+u_{i,j+1}^{n+1}}{\triangle y^2}+\frac{u_{i,j-1}^{n}-2u_{i,j}^{n}+u_{i,j+1}^{n}}{\triangle y^2}\right]\\
\Rightarrow \quad & -\frac{r_1}{2}\Big(u_{i-1,j}^{n+1}+u_{i+1,j}^{n+1}\Big)+(1+r_1+r_2)u_{i,j}^{n+1}-\frac{r_2}{2}\Big(u_{i,j-1}^{n+1}+u_{i,j+1}^{n+1}\Big)\\
&=\frac{r_1}{2}\Big(u_{i-1,j}^{n}+u_{i+1,j}^{n}\Big)+(1+r_1+r_2)u_{i,j}^{n}-\frac{r_2}{2}\Big(u_{i,j-1}^{n}+u_{i,j+1}^{n}\Big)
\end{aligned}$$

where $r_1=\dfrac{C\triangle t}{\triangle x^2}$ and $r_2=\dfrac{C\triangle t}{\triangle y^2}$,
for $i=1,\,2,\,3,\,\cdots\,N$, $j=1,\,2,\,3,\,\cdots\,M$ and $n=0,\,1,\,2,\,\cdots\cdots$

This is certainly a viable scheme; the problem arises in solving the coupled linear equations. Whereas in one dimension the system was tridiagonal, that is no longer true, though the matrix is still very sparse.

Alternate Direction Implicit (ADI) provides a slightly different way of generalizing the Crank-Nicholson algorithm. It is still second-order accurate in time and space, and unconditionally stable, but the equations are easier to solve than in the above case. Here, the idea is to divide each timestep into two steps of size $\dfrac{\triangle t}{2}$. In each substep, a different dimension is treated implicitly. The equations can be written as follows:
First half time step: implicit along x direction

$$\frac{u_{i,j}^{n+\frac{1}{2}}-u_{i,j}^{n}}{\triangle t/2}=C\left[\frac{u_{i-1,j}^{n+\frac{1}{2}}-2u_{i,j}^{n+\frac{1}{2}}+u_{i+1,j}^{n+\frac{1}{2}}}{\triangle x^2}+\frac{u_{i,j-1}^{n}-2u_{i,j}^{n}+u_{i,j+1}^{n}}{\triangle y^2}\right] \qquad (A.12)$$

Second half time step: implicit along y direction

$$\frac{u_{i,j}^{n+1}-u_{i,j}^{n+\frac{1}{2}}}{\triangle t/2}=C\left[\frac{u_{i-1,j}^{n+\frac{1}{2}}-2u_{i,j}^{n+\frac{1}{2}}+u_{i+1,j}^{n+\frac{1}{2}}}{\triangle x^2}+\frac{u_{i,j-1}^{n+1}-2u_{i,j}^{n+1}+u_{i,j+1}^{n+1}}{\triangle y^2}\right] \qquad (A.13)$$

for $i = 1, 2, 3, \cdots N$, $j = 1, 2, 3, \cdots M$ and $n = 0, 1, 2, \cdots\cdots$
Rewriting eq. (A.12):

$$-\frac{r_1}{2}u_{i-1,j}^{n+\frac{1}{2}} + (1+r_1)u_{i,j}^{n+\frac{1}{2}} - \frac{r_1}{2}u_{i+1,j}^{n+\frac{1}{2}} = \frac{r_2}{2}u_{i,j-1}^{n} + (1-r_2)u_{i,j}^{n} + \frac{r_2}{2}u_{i,j+1}^{n} \qquad \text{(A.14)}$$

This is a tridiagonal system which can be solved using Thomas algorithm for the unknown $u_{i,j}$ at the time step $n+\frac{1}{2}$. Similarly eq. (A.13) can be rewritten as:

$$-\frac{r_2}{2}u_{i,j-1}^{n+1} + (1+r_2)u_{i,j}^{n+1} - \frac{r_2}{2}u_{i,j+1}^{n+1} = \frac{r_1}{2}u_{i-1,j}^{n+\frac{1}{2}} + (1-r_1)u_{i,j}^{n+\frac{1}{2}} + \frac{r_1}{2}u_{i+1,j}^{n+\frac{1}{2}} \qquad \text{(A.15)}$$

Since u terms on the right hand side of eq. (A.15) have already been calculated by solving eq. (A.14), eq. (A.15) is again a tridiagonal system which can also be solved using Thomas algorithm for $u_{i,j}$ at time step n+1. This completes one iteration in time direction and the same is repeated until the desired time step is reached. The advantage of this method is that each substep requires only the solution of a simple tridiagonal system.

Appendix B

C-programs

In this chapter we list our C-programs used for solving the RD equation numerically.

B.1 ADI program benchmarked

To benchmark our program of ADI method for Neumann boundary conditions we first solved the standard 2D diffusion equation (as shown below) numerically and then compared our numerical results with the analytical solution of the equation.

$$\frac{\partial u}{\partial t} = \alpha\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \quad \text{for} \quad 0 \le x \le 1, 0 \le y \le 1,$$

with Neumann boundary conditions

$$\begin{aligned}\frac{\partial u}{\partial x} &= 0 \quad \text{on } x = 0{,}1\\ \frac{\partial u}{\partial y} &= 0 \quad \text{on } y = 0{,}1\end{aligned}$$

We compared the L^2 norm of the analytical and numerical solutions over time. To the find analytical solution, we consider separation of variables:

$$\begin{aligned}u(t,x,y) &= T(t)\phi(x,y)\\ T'\phi &= \alpha T\nabla^2\phi\\ \frac{T'}{T} &= \alpha\frac{\nabla^2\phi}{\phi} = -\lambda\end{aligned}$$

This yields two separate equations,

$$\begin{aligned} T' + \lambda T &= 0 \\ \alpha \nabla^2 \phi + \lambda \phi &= 0 \end{aligned}$$

Applying the separation of variables again to the second equation above:

$$\begin{aligned} \phi(x, y) &= X(x)Y(y) \\ \alpha X''Y + \alpha XY'' + \lambda XY &= 0 \\ \frac{\alpha X''}{X} + \frac{\alpha Y''}{Y} + \lambda &= 0 \\ \frac{-\alpha Y'' - \lambda Y}{Y} = \frac{\alpha X''}{X} &= -k \end{aligned}$$

This gives two separate equations,

$$\begin{aligned} \alpha X'' + kX &= 0 \\ \alpha Y'' + (\lambda - k)Y &= 0 \end{aligned}$$

For the sake of simplicity, we take $\alpha = 1$. The solution of the first equation above can be written as:

$$X(x) = a\cos(x\sqrt{k}) + b\sin(x\sqrt{k})$$

Applying the boundary conditions, we find:

$$\begin{aligned} &b = 0 \\ &k = (n\pi)^2, \qquad n = 1,\, 2,\, \cdots \end{aligned}$$

Similarly, the solution of the second equation can be written as:

$$Y(y) = p\cos(y\sqrt{\lambda - k}) + q\sin(y\sqrt{\lambda - k})$$

Applying the boundary conditions we find:

$$\begin{aligned} p &= 0 \\ \lambda - k &= (m\pi)^2, \qquad m = 1,\, 2,\, \cdots \\ \lambda &= (n\pi)^2 + (m\pi)^2 \end{aligned}$$

Therefore, the solution of the 2D heat equation is:

$$u(t,x,y) = Ae^{-((n\pi)^2+(m\pi)^2)t}\cos(n\pi x)\cos(m\pi y)$$

Setting α to 1 and given the initial conditions $u(x,y,0) = \cos(2\pi x)\cos(\pi y)$, the L^2 norm of the analytical solution is

$$|u(t,x,y)| = \sqrt{\int_0^1\int_0^1 u^2(t,x,y)dxdy} = \frac{1}{2}e^{-5\pi^2 t}$$

Figure B.1 shows a comparison of the analytical solution and the numerical solution obtained using ADI. The fact that the dots lie on the solid line suggests that the numerical and analytical solutions are in good agreement with each other.

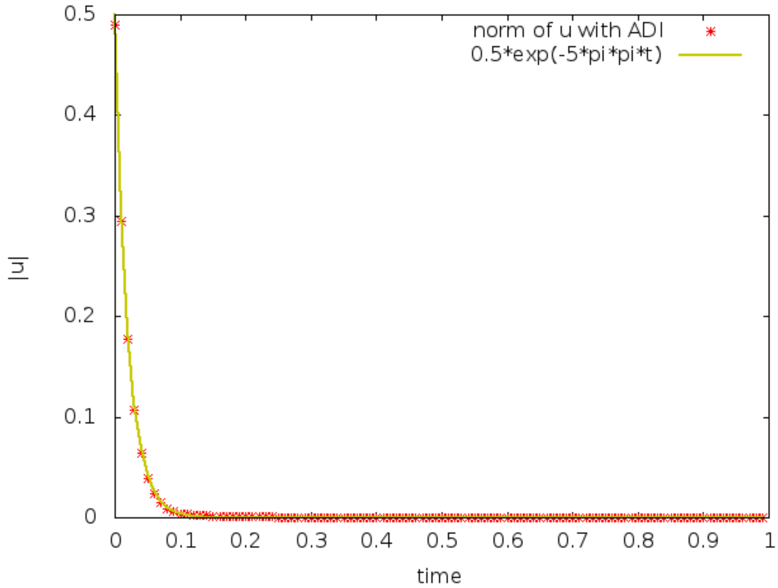


Figure B.1: Comparison of the analytical (solid line) and the numerical solution (*) of the 2D heat equation.

Following is the program for numerical solution of 2D heat equation using the ADI method:

ADI_neumann.C

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#define nx 100
#define ny 100
#define pi 3.14159265358979
#define dx 0.01
#define dy 0.01
void tridag(double a[], double b[], double c[], double d[], double u[], int n);
void neumannbc(double a[nx+2][ny+2]);
double norm2(double a[nx+2][ny+2]);
int main(void)
{
  /*defining required parameters and variables*/
  int i, j;
  double d, unorm, analyticalnorm, error;
  double u[nx+2][ny+2], temp[nx][ny];
  double a1[nx],b1[nx],c1[nx], d1[nx],temp1[nx]; //Tridiagonal coefficients along x
  double a2[ny],b2[ny],c2[ny], d2[ny],temp2[ny]; //Tridiagonal coefficients along y
  double t, dt, tfin, r1, r2, r3, r4, x, y;
  FILE *fp, *fpo;
  /*initializing the parameters and variables*/
  d = 1.0;
  t = 0.0;
  dt = 0.01;
  tfin = 1.0;
  r1 = d*dt/(dx*dx); r2 = d*dt/(dy*dy);
  r3 = d*dt/(dx*dx); r4 = d*dt/(dy*dy);
  /*Initialize concentration arrays and coupling terms*/
for(i=1; i<nx+1; i++)
   {
    for(j=1; j<ny+1; j++)
     {
        u[i][j] = cos(2*pi*i*dx)*cos(pi*j*dy);
        temp[i-1][j-1] = 0.0;
     }
   }
  /*initializing boundary values of concentrations using boundary conditions*/
  neumannbc(u);
   fpo=fopen("norm.dat","w");
        if(fpo==NULL)
          {
            puts("cannot open file");
            exit(1);
          }
  /*Begin time loop*/
  while(t < tfin)
   {
    unorm = sqrt(norm2(u));
    analyticalnorm = 0.5*exp(-5*pi*pi*t);
    error = fabs(unorm-analyticalnorm);
    fprintf(fpo,"%lf\t%lf\t%lf\t%lf\n",t,unorm,analyticalnorm,error);
  /*First half time step: implicit along x*/
  /*Thomas algorithm to update concentration values*/
   for(j=1; j<ny+1; j++)
    {
      /*Initializing tridiagonal coefficients*/
      for(i=0; i<nx; i++)
       {
        a1[i] = (-r1/2.);
        b1[i] = 1. + r1;
        c1[i] = (-r1/2.);
        d1[i] = 0.5*r2*u[i+1][j-1] + (1-r2)*u[i+1][j] + 0.5*r2*u[i+1][j+1];
       }
     /*Update coefficients as per Neumann boundary conditions*/
```

```
        c1[0] = -r1;
        a1[nx-1] = -r1;
     tridag(a1,b1,c1,d1,temp1,nx); // Solving for concentrations using Thomas algorithm
      if(temp1[4]>=1000.0)
       {
        printf("Error 4\n");
        exit(1);
       }
      for(i=0;i<nx;i++)
         temp[i][j-1] = temp1[i];
     }
   /*End of first half time step*/
  /*Update concentrations and boundary values*/
  for(i=1; i<nx+1; i++)
    for(j=1; j<ny+1; j++)
       u[i][j] = temp[i-1][j-1];
   neumannbc(u);
   /*Second half time step: implicit along y*/
  /*Thomas algorithm to update concentration values*/
   for(i=1; i<nx+1; i++)
    {
      /*Initializing tridiagonal coefficients*/
      for(j=0; j<ny; j++)
       {
        a2[j] = (-r2/2.);
        b2[j] = 1. + r2;
        c2[j] = (-r2/2);
        d2[j] = 0.5*r1*u[i-1][j+1] + (1. - r1)*u[i][j+1] + 0.5*r1*u[i+1][j+1];
       }
/*update boundary coefficients as per neumann bc*/
       c2[0] = -r2;
       a2[ny-1] = -r2;
      tridag(a2,b2,c2,d2,temp2,ny); //Solving for concentrations using Thomas algorithm
    if(temp2[4]>=1000.0)
       printf("Error 5\n");
    for(j=0; j<ny; j++)
       temp[i-1][j] = temp2[j];
   }
       /*End of second half time step*/
 /*Update concentrations and boundary values*/
    for(i=1; i<nx+1; i++)
      for(j=1; j<ny+1; j++)
         u[i][j] = temp[i-1][j-1];
       neumannbc(u);
   t +=dt;
  /*End of one complete time step*/
  }
  fclose(fpo);
  fpo=fopen("final.dat","w");
     if(fpo==NULL)
          {
            puts("cannot open file");
            exit(1);
          }
   fprintf(fpo,"Final concentration of u\n\n");
   x=0.0;
       for(i=1; i<nx+1; i++)
         {
           y=0.0;
           for(j=1; j<ny+1; j++)
             {
               fprintf(fpo,"%lf\t%lf\t%lf\n",x,y,u[i][j]);
               y += dy;
             }
           x += dx;
         }
   fclose(fpo);
  return 0;
}
```

```c

/*Function for solving tridiagonal system using Thomas algorithm*/
void tridag(double a[], double b[], double c[], double d[], double u[], int n)
{
  int j;
      double bet, gam[n];
      if (b[0] == 0.0) printf("Error 1 in tridag\n");
  u[0]=d[0]/(bet=b[0]);
  for (j=1;j<n;j++) {
    gam[j]=c[j-1]/bet;
    bet=b[j]-a[j]*gam[j];
    if (bet <= 1.0) printf("Error 2 in tridag\n");
    u[j]=(d[j]-a[j]*u[j-1])/bet;
      }
      for (j=(n-2);j>=0;j--)
    u[j] -= gam[j+1]*u[j+1];
}
/*Function for applying neumann boundary conditions*/
void neumannbc(double a[nx+2][ny+2])
{
  int i;
     for(i=1; i<nx+1; i++)
    {
       a[i][0]=a[i][2];
       a[i][ny+1]=a[i][ny-1];
    }
    for(i=1; i<ny+1; i++)
    {
       a[0][i]=a[2][i];
       a[nx+1][i]=a[nx-1][i];
    }
      a[0][0]=a[2][2];
      a[nx+1][0]=a[nx-1][2];
      a[0][ny+1]=a[2][ny-1];
      a[nx+1][ny+1]=a[nx-1][ny-1];
}
/*Function for double integration over 2d space using trapeziod rule*/
double norm2(double a[nx+2][ny+2])
{
  int i,j;
  double f[nx], F;
  F=0.0;
  for(i=1; i<nx+1; i++)
    {
       f[i-1]=0.0;
       for(j=1; j<ny; j++)
           f[i-1] += 0.5*(a[i][j+1]*a[i][j+1] + a[i][j]*a[i][j])*dy;
    }
   for(i=0; i<nx-1; i++)
        F += 0.5*(f[i+1] + f[i])*dx;
 return (F);
}
```

B.2 FTCS scheme on Meinhardt's model-1d

Following is the program for numerical solution of one-dimensional Meinhardt's model using Forward Time Central Space scheme.

explicit_1d.C

```c
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#define Da 0.0200
#define Db 0.4000
```

```c
#define nx 100
#define timeSteps 3600
void fprintmatrix (double p[nx+2], FILE *f);
int main(void)
{
  /*defining required parameters and variables*/
  int i, n;
  double ba, bb, ra, rb, a0, b0, qc, qb, q;// gamma, a, b, alpha, u0, v0;
  double a[nx+2], b[nx+2], dela[nx], delb[nx], s[nx];
  FILE *fp, *fpo;
  /*initializing the parameters and variables*/
  ba = 0.0001; //rho0 in equation 46
  bb = 0.0000;
  ra = 0.0100; //mu in equation 46
  rb = 0.0200; //nu in equation 46
  qc = 0.0200; //magnitude of random fluctuations about rho in source density
  qb = 1.0000; //100 times rho
  a0 = rb/ra + ba/ra;
  b0 = (a0*a0*0.01)/rb;
  /*Initialize concentration arrays and coupling terms*/
  for(i=1; i<nx+1; i++)
   {
       s[i-1] = 0.01*(qb + qc*((double) rand()/RAND_MAX)); //source density
       a[i] = a0;
       b[i] = b0;
   }
/*initializing boundary values of concentrations using neumann boundary conditions*/
 a[0] = a[1]; b[0] = b[1];
 a[nx+1] = a[nx]; b[nx+1] = b[nx];
  fp=fopen("evolve_a.dat","w");
        if(fp==NULL)
          {
            puts("cannot open file");
            exit(1);
          }
  fpo=fopen("evolve_b.dat","w");
        if(fpo==NULL)
          {
            puts("cannot open file");
            exit(1);
          }
  /*Begin time loop*/
  for (n=1; n<=timeSteps; n++)
   {
   /*Printing concentrations at intermediate times*/
     if(n%200==0)
      {
          for(i=1; i<nx+1; i++)
              fprintf(fp,"%d\t%d\t%lf\n",i,n,a[i]);
        fprintf(fp,"\n");
          for(i=1; i<nx+1; i++)
              fprintf(fpo,"%d\t%d\t%lf\n",i,n,b[i]);
        fprintf(fpo,"\n");
      }
/*Calculating changes in concentrations explicitly*/
     for(i=1; i<nx+1; i++)
       {
         q = s[i-1]*a[i]*a[i];
         dela[i-1] = q/b[i] - ra*a[i] + ba + Da*(a[i-1] - 2*a[i] + a[i+1]);
         delb[i-1] = q - rb*b[i] + bb + Db*(b[i-1] - 2*b[i] + b[i+1]);
       }
/*Updating concentrations by adding the changes*/
     for(i=1; i<nx+1; i++)
       {
         a[i] += dela[i-1];
         b[i] += delb[i-1];
       }
 /*Update boundary values*/
      a[0] = a[1]; b[0] = b[1];
      a[nx+1] = a[nx]; b[nx+1] = b[nx];
  }
  fclose(fp);
  fclose(fpo);
  fpo=fopen("finala.dat","w");
     if(fpo==NULL)
          {
            puts("cannot open file");
            exit(1);
          }
  fp=fopen("finalb.dat","w");
     if(fp==NULL)
          {
            puts("cannot open file");
            exit(1);
          }
//Printing final concentrations
   fprintmatrix(a,fpo);
   fprintmatrix(b,fp);
  fprintf(fp,"\n\n");
  for(i=0; i<nx; i++)
    fprintf(fp,"%d\t%lf\n",i,s[i]);
   fclose(fpo);
   fclose(fp);
  return 0;
}
/*Function for printing matrix into file*/
void fprintmatrix (double p[nx+2], FILE *f)
{
 int i;
       for(i=1; i<nx+1; i++)
           fprintf(f,"%d\t%lf\n",i,p[i]);
}
```

B.3 ADI scheme on Murray's model-2d

Following is the program for numerical solution of two-dimensional Murray's model using Alternate Direction Implicit scheme.

ADI_thomas.C

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#define d 5.0
#define nx 300
#define ny 100
#define dx 0.01
#define dy 0.01
#define dt 0.001
#define timeSteps 500
void fprintmatrix (double p[nx+2][ny+2], FILE *f);
void tridag(double a[], double b[], double c[], double r[], double u[], int n);
void neumannbc(double a[nx+2][ny+2]);
double r1, r2, r3, r4;
int main(void)
{
  /*defining required parameters and variables*/
  int i, j, n;
  double K, rho, gamma, a, b, alpha, u0, v0; //usual notations as in equation 48
  double u[nx+2][ny+2], v[nx+2][ny+2], tempu[nx][ny], tempv[nx][ny], f[nx][ny], g[nx][ny], h[nx][ny];
```

```
 double a1[nx], b1[nx], c1[nx], d1[nx], t1[nx]; //Tridiagonal coefficients along x
 double a2[ny], b2[ny], c2[ny], d2[ny], t2[ny]; //Tridiagonal coefficients along y
 double x, y, t;
 FILE *fp, *fpo;
 /*initializing the parameters and variables*/
 K = 0.1;
 rho = 13.0;
 gamma = 4.0;
 a = 102.0;
 b = 77.0;
 alpha = 1.5;
 u0 = 6.17455;
 v0 = 13.1144;
 r1 = dt/(dx*dx); r2 = dt/(dy*dy);
 r3 = d*dt/(dx*dx); r4 = d*dt/(dy*dy);
/*Initialize concentration arrays and coupling terms*/
 for(i=1; i<nx+1; i++)
  {
   for(j=1; j<ny+1; j++)
    {
      u[i][j] = u0 + 0.5*((double) rand()/(RAND_MAX/2) - 1.0);
      v[i][j] = v0 + 0.5*((double) rand()/(RAND_MAX/2) - 1.0);
      h[i-1][j-1] = rho*u[i][j]*v[i][j]/(1+u[i][j]+K*u[i][j]*u[i][j]);
      f[i-1][j-1] = gamma*(a - u[i][j] - h[i-1][j-1]);
      g[i-1][j-1] = gamma*(alpha*(b - v[i][j]) - h[i-1][j-1]);
      tempu[i-1][j-1] = 0.0;
      tempv[i-1][j-1] = 0.0;
    }
  }
 /*initializing boundary values of concentrations using neumann boundary conditions*/
neumannbc(u);
neumannbc(v);
/*Begin time loop*/
 for (n=1; n<=timeSteps; n++)
  {
   t = n*dt;
 /*First half time step: implicit along x*/
 /*Thomas algorithm to update concentration values*/
  for(j=1; j<ny+1; j++)
   {
    /*Initializing tridiagonal coefficients*/
     for(i=0; i<nx; i++)
      {
       a1[i] = -0.5*r1;
       b1[i] = 1 + r1;
       c1[i] = -0.5*r1;
       d1[i] = 0.5*r2*u[i+1][j-1] + (1-r2)*u[i+1][j] + 0.5*r2*u[i+1][j+1] + 0.5*dt*f[i][j-1];
      }
  /*update boundary coefficients for neumann bc*/
       c1[0] = -r1;
       a1[nx-1] = -r1;
      /*Solving the tridiagonal matrix equations*/
     tridag(a1,b1,c1,d1,t1,nx);
     /*Back substitution into temporary matrix*/
     for(i=0; i<nx; i++)
        tempu[i][j-1] = t1[i];
    /*repeating the above procedure for concentration v*/
     for(i=0; i<nx; i++)
      {
       a1[i] = -0.5*r3;
       b1[i] = 1 + r3;
       c1[i] = -0.5*r3;
       d1[i] = 0.5*r4*v[i+1][j-1] + (1-r4)*v[i+1][j] + 0.5*r4*v[i+1][j+1] + 0.5*dt*g[i][j-1];
      }
      c1[0] = -r3;
      a1[nx-1] = -r3;
     tridag(a1,b1,c1,d1,t1,nx);
```

```
      /*Back substitution: Updating concentrations u*/
      for(i=0; i<nx; i++)
         tempv[i][j-1] = t1[i];
     }
   /*End of first half time step*/
/*Update concentrations and boundary values*/
  for(i=1; i<nx+1; i++)
    for(j=1; j<ny+1; j++)
      {
       u[i][j] = tempu[i-1][j-1];
       v[i][j] = tempv[i-1][j-1];
      }
  neumannbc(u);
  neumannbc(v);
  /*Update coupling terms*/
      for(i=1; i<nx+1; i++)
       {
         for(j=1; j<ny+1; j++)
           {
             h[i-1][j-1] = rho*u[i][j]*v[i][j]/(1+u[i][j]+K*u[i][j]*u[i][j]);
             f[i-1][j-1] = gamma*(a - u[i][j] - h[i-1][j-1]);
             g[i-1][j-1] = gamma*(alpha*(b - v[i][j]) - h[i-1][j-1]);
           }
       }
   /*Second half time step: implicit along y*/
  /*Thomas algorithm to update concentration values*/
  for(i=1; i<nx+1; i++)
    {
      /*Initializing tridiagonal coefficients*/
      for(j=0; j<ny; j++)
       {
        a2[j] = -0.5*r2;
        b2[j] = 1 + r2;
        c2[j] = -0.5*r2;
        d2[j] = 0.5*r1*u[i-1][j+1] + (1-r1)*u[i][j+1] + 0.5*r1*u[i+1][j+1] + 0.5*dt*f[i-1][j];
       }
   /*update boundary coefficients for neumann bc*/
       c2[0] = -r2;
       a2[ny-1] = -r2;
       /*Solving tridiagonal matrix equation*/
      tridag(a2,b2,c2,d2,t2,ny);
       /*Back substitution into temporary matrix*/
      for(j=0; j<ny; j++)
        tempu[i-1][j] = t2[j];
      /*repeating the above procedure for concentration v*/
      for(j=0; j<ny; j++)
       {
        a2[j] = -0.5*r4;
        b2[j] = 1 + r4;
        c2[j] = -0.5*r4;
        d2[j] = 0.5*r3*v[i-1][j+1] + (1-r3)*v[i][j+1] + 0.5*r3*v[i+1][j+1] + 0.5*dt*g[i-1][j];
       }
       c2[0] = -r4;
       a2[ny-1] = -r4;
      tridag(a2,b2,c2,d2,t2,ny);
      for(j=0; j<ny; j++)
       tempv[i-1][j] = t2[j];
    }
       /*End of second half time step*/
  /*Update concentrations and boundary values*/
  for(i=1; i<nx+1; i++)
    for(j=1; j<ny+1; j++)
      {
       u[i][j] = tempu[i-1][j-1];
       v[i][j] = tempu[i-1][j-1];
      }
  neumannbc(u);
  neumannbc(v);
```

```
    /*Update coupling terms*/
      for(i=1; i<nx+1; i++)
       {
         for(j=1; j<ny+1; j++)
           {
             h[i-1][j-1] = rho*u[i][j]*v[i][j]/(1+u[i][j]+K*u[i][j]*u[i][j]);
             f[i-1][j-1] = gamma*(a - u[i][j] - h[i-1][j-1]);
             g[i-1][j-1] = gamma*(alpha*(b - v[i][j]) - h[i-1][j-1]);
           }
       }
 /*End of one complete time step*/
  }
  fpo=fopen("finalu.dat","w");
     if(fpo==NULL)
          {
            puts("cannot open file");
            exit(1);
          }
  fp=fopen("finalv.dat","w");
     if(fp==NULL)
          {
            puts("cannot open file");
            exit(1);
          }
   fprintmatrix(u,fpo);
   fprintmatrix(v,fp);
   fclose(fpo);
   fclose(fp);
  return 0;
}
/*Function for printing concentrations along with their spatial coordinates*/
void fprintmatrix (double p[nx+2][ny+2], FILE *f)
{
 int i,j;
 double x, y;
      x = 0.0;
       for(i=1; i<nx+1; i++)
         {
           y=0.0;
           for(j=1; j<ny+1; j++)
             {
               fprintf(f,"%lf\t%lf\t%lf\n",x,y,p[i][j]);
               y += dy;
             }
            x += dx;
         }
}
/*Function for solving tridiagonal system using Thomas algorithm*/
void tridag(double a[], double b[], double c[], double r[], double u[], int n)
{
  int j;
      double bet, gam[n];
      if (b[0] == 0.0) printf("Error 1 in tridag\n");
  u[0]=r[0]/(bet=b[0]);
  for (j=1;j<n;j++) {
    gam[j]=c[j-1]/bet;
    bet=b[j]-a[j]*gam[j];
    if (bet == 0.0) printf("Error 2 in tridag\n");
    u[j]=(r[j]-a[j]*u[j-1])/bet;
      }
      for (j=(n-2);j>=0;j--)
    u[j] -= gam[j+1]*u[j+1];
}
/*FUnction for applying Neumann boundary conditions*/
void neumannbc(double a[nx+2][ny+2])
{
  int i;
     for(i=1; i<nx+1; i++)
    {
       a[i][0]=a[i][2];
       a[i][ny+1]=a[i][ny-1];
    }
    for(i=1; i<ny+1; i++)
    {
       a[0][i]=a[2][i];
       a[nx+1][i]=a[nx-1][i];
    }
      a[0][0]=a[2][2];
      a[nx+1][0]=a[nx-1][2];
      a[0][ny+1]=a[2][ny-1];
      a[nx+1][ny+1]=a[nx-1][ny-1];
}
```

B.4 Discrete Cosine Transform

Following is the program used to find the Discrete Cosine Transform of the measured patterns.

DCT.C

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define pi 3.14159265358979
#define N 880
#define dx 0.01
double DCT (double a[], int k);
int main()
{
   int k;
   double x[N], f[N];
   FILE *fp;
  fp=fopen("white.txt","r");
     if(fp==NULL)
          {
            puts("cannot open file");
            exit(1);
          }
   for(k=0; k<N; k++)
       fscanf(fp,"%lf",&x[k]);
  fclose(fp);
   fp=fopen("whiteDCT.dat","w");
     if(fp==NULL)
          {
            puts("cannot open file");
            exit(1);
          }
   for(k=0; k<N; k++)
      {
         f[k] = DCT(x,k);
         fprintf(fp,"%d\t%lf\n",k,f[k]);
      }
   fclose(fp);
  fp=fopen("violet.txt","r");
     if(fp==NULL)
          {
            puts("cannot open file");
            exit(1);
          }
   for(k=0; k<N; k++)
       fscanf(fp,"%lf",&x[k]);
  fclose(fp);
   fp=fopen("violetDCT.dat","w");
     if(fp==NULL)
          {
            puts("cannot open file");
            exit(1);
          }
   for(k=0; k<N; k++)
      {
         f[k] = DCT(x,k);
         fprintf(fp,"%d\t%lf\n",k,f[k]);
      }
   fclose(fp);
  for(k=0; k<N; k++)
    x[k] = cos(200.0*k*dx);
 fp = fopen("benchmark.dat", "w");
  if(fp==NULL)
   {
    puts("cannot open file");
    exit(1);
   }
  for(k=0; k<N; k++)
   {
    f[k] = DCT(x,k);
    fprintf(fp,"%d\t%lf\n",k,f[k]);
   }
  fclose(fp);
 return 0;
}
double DCT (double a[], int k)
{
 int j;
 double y =0;
 for(j=0; j<N; j++)
    y += a[j]*cos(pi*(j+0.5)*k/N);
 return y*dx;
}
```