

M. Tech. (Computer Science) Dissertation Series

Pattern Analysis using Level Set Method

a dissertation submitted in partial fulfillment of the
requirements for the M. Tech. (Computer Science)
degree of the Indian Statistical Institute

by

Amit Chattopadhyay

under the supervision of

Dr. D. P. Mukherjee

Associate Professor

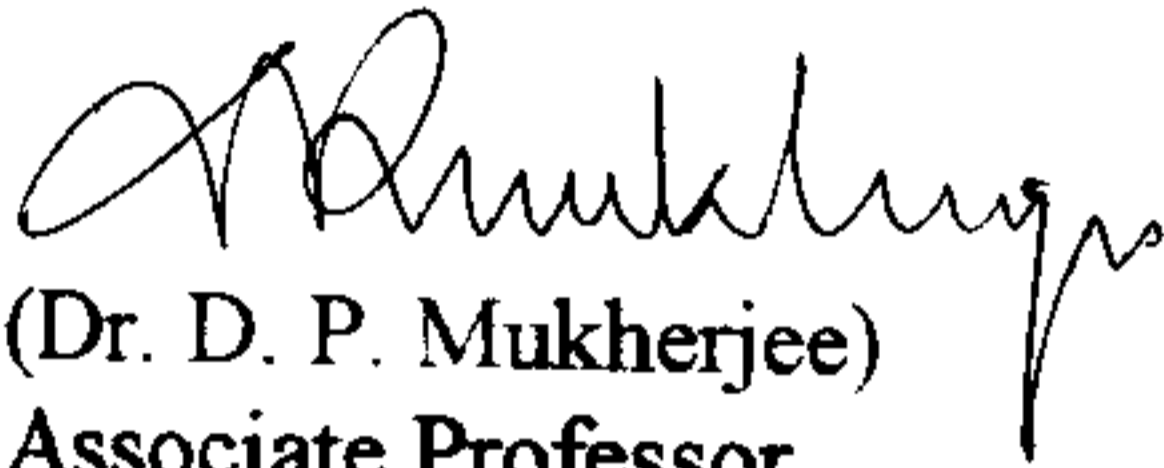
Electronics and Communication Sciences Unit



INDIAN STATISTICAL INSTITUTE
203, Barrackpore Trunk Road
Kolkata-700108.

Certificate of Approval

This is to certify that the thesis titled *Pattern Analysis using Level Set Method* submitted by Amit Chattopadhyay, towards partial fulfillment of the requirements for the degree of M.Tech in Computer Science at the Indian Statistical Institute, Kolkata, embodies the work done under my supervision. His work is satisfactory.



(Dr. D. P. Mukherjee)

Associate Professor

Electronics and Communication Sciences Unit

Indian Statistical Institute, Kolkata

Acknowledgement

This is with affection and appreciation that I acknowledge my indebtedness to Dr. Dipti Prasad Mukherjee, Associate Professor, Indian Statistical Institute, Kolkata, for his advice, enthusiasm, and criticism throughout the course of this dissertation. It would have been very difficult to complete the dissertation without his able guidance. I am thankful to him.

Amit Chattopadhyay
M. Tech (Computer Science)
Indian Statistical Institute, Kolkata

Pattern Analysis using Level Set Method

Contents

Chapter 1: Introduction	2
Chapter 2: Pattern Generation Methods	4
2.1: Structural Optimization Method	4
2.2: Reaction-Diffusion Method	8
Chapter 3: Curve Evolution Methods	13
3.1: Basic Concepts of Curve Evolution	13
3.2: Level Set Method	14
Chapter 4: Level Set Evolution Based Pattern Generation Model	20
4.1: Level Set Evolution Based on Shape Optimization	20
4.2: Level Set Evolution Based on Reaction-Diffusion	28
4.3: Intensity Generation Model	30
4.4: Pattern Disocclusion	31
Chapter 5: Conclusion	33
References	34

Chapter 1: Introduction

Patterns generated in nature often enchant us. Since reproducing the realism of the physical world is a major goal of computer graphics, patterns are important for rendering synthetic images and animations. However, patterns are so diverse that it is difficult to describe and generate them in a single framework. This motivates computer graphics people to propose different pattern generation models. Patterns are basically of two types periodic and quasi-periodic. Patterns, which occur in a regular manner or periodically, are called regular patterns. Spots of cheetah, coat pattern of zebra, spots of giraffe etc. are examples of periodic patterns. Patterns, which are repeated after long period or in other words, which occur in a quasi-periodic way, are called quasi-periodic patterns. An example of quasi-periodic pattern is human fingerprint.

There are many pattern generation models in literature [1,3,4]. We address two models, which emulate patterns in physical world. Physical phenomena like compliance or stiffness¹ of a body, is optimized to create patterns in real world. This is also known as pattern generation through structural topology optimization. The term 'topology optimization' will be explained in the proper context. The underlying goal of this model is to get the optimal material distribution of the structure, given certain forces and constraints applying on the structure. This is as if a goldsmith is giving different shapes to a piece of gold for creating different patterns. Other phenomenon of creation of patterns is biological phenomenon. These are based on some chemical reactions, which control pattern generation in biological world. For example the biological phenomena by which spots of cheetah, stripes of zebra, patterns of giraffe, spots of leopard etc are formed. The reaction-diffusion model proposed by Turing [1] was the pioneering work in this context. Among other models, Meinherdt's stripe-formation model [3], netlike structure generation model [3], models proposed by J.D. Murray [4] are well known.

The motivation behind our work is to study pattern generation models and to design a similar model using level set framework [8]. Level set is a curve evolution method. Here evolution of the curve is traced by embedding the curve into a higher dimensional surface. We generate patterns by evolving a curve under different conditions, motivated by physical and biological phenomena. The main difference of our model with previous models is that level set based model of pattern generation is more efficient. Using level set method, the change in connectivity or topology of an evolving curve can be traced efficiently. This is important for pattern generation process. Reaction diffusion model of Turing is computationally expensive But the method what we have proposed faster than reaction-diffusion model.

In Chapter 2, we discuss topology optimization model and reaction-diffusion model of pattern generation. In Chapter 3, curve evolution theory and level set model is described. In Chapter 4, we propose our model of pattern generation based on level set

¹ Stiffness of a material body is measured by work done by the external forces on the body.

framework. Also we apply our model of pattern generation to regenerate some occluded patterns. In Chapter 5, we conclude by discussing about possible future work of the proposed approach.

In the next chapter we discuss how patterns are generated using shape optimization and reaction-diffusion approach.

Chapter 2: Pattern Generation Methods

Shape optimization approach generates optimized structure of a piece of material under specific physical constraints. The optimized structure is often visualized as aesthetically attractive graphical pattern. In Section 2.1 we discuss pattern generation using structural topology optimization. Here we describe the method with an example of a rectangular structure. In Section 2.2 we describe pattern generation by reaction-diffusion method.

Section 2.1: Structural Optimization Method

In the pattern generation model based on optimization of structural topology, the objective is that given a structure, we apply forces and constraints on it to generate a new pattern. So the aim of this model of pattern generation is to find out the stable configurations of the structure under the application of certain forces and constraints. The stable configuration of the structure is an optimal material distribution of the structure where stability of the structure is ensured.

The physical property of the structure, which is optimized, is the stiffness or strength of the structure. This is measured in terms of compliance which is the work done by the external forces on the body. Therefore, the objective is to maximize the stiffness of the body or minimize the compliance with certain constraint on the volume of the total material.

Consider a design layout as shown in Figure 2.1(a). The body is subject to a force at the right end and supported at the left end. The shape of the body is allowed to vary, under the condition that its boundary is within the design domain. The optimal layout is found by discretizing the design domain. Figure 2.1(b) shows the optimization structure with a specific pattern. With the variation of load, support and material characterization different patterns can be generated. Finite element method (FEM) [16] is used to find the stresses and displacements of the elements of the body and the method of moving asymptotes (MMA) [17] is used to find the optimal material distribution.

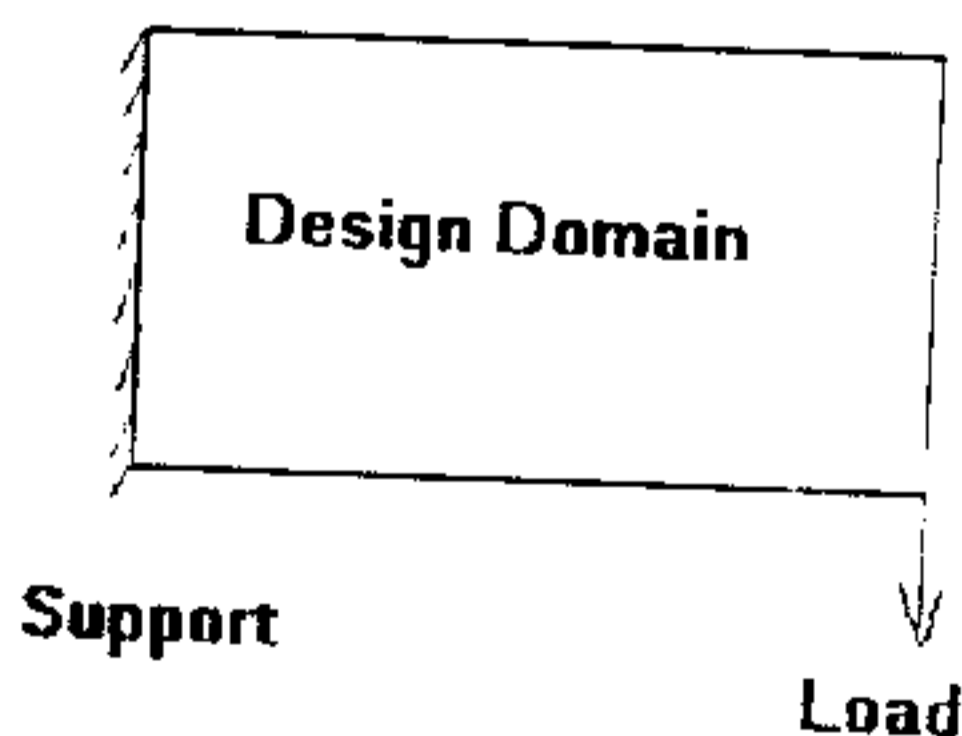


Figure 2.1(a) Design Layout



Figure 2.1(b) Desired Pattern

Material Model

The design domain is discretized into $N = nelx \times nely$ number of square elements or cells where $nelx$ is the number of rows and $nely$ is the number of columns of the design domain. Vertices of each square cell are called nodes. So there are total $(nelx+1)$ nodes in a row and $(nely+1)$ nodes in a column. We take upper left corner of our design domain as origin. We can arrange the displacements u_i and the forces F_i on every node in the design domain in vectors U and F respectively. So the compliance C can be expressed as,

$$C = F^T U, \quad (2.1)$$

by the definition of the work as the force times displacement. Using finite element methods we get a system of linear equations,

$$F = KU, \quad (2.2)$$

where K is the global stiffness matrix and U the displacement vector. Since K is symmetric, i.e. $K^T = K$, the compliance may be written as

$$C = F^T U = U^T K U. \quad (2.3)$$

Design Variables

The geometry of the body is specified by some design variables, the densities of material in each cell, x_e . The index e refers to cell e . The constraint is to use a fixed volume of material, so the design variables must fulfill

$$V = x_1 v_1 + x_2 v_2 + \dots + x_N v_N, \quad (2.4)$$

where N is the total number of cells, x is the vector with components x_i and v is the vector of cell volumes v_i .

The total stiffness matrix of the body obtained by finite element method can be written as a sum

$$K = x_1 k_0 + x_2 k_0 + \dots + x_N k_0, \quad (2.5)$$

where k_0 is the element stiffness matrix determining the relations between the stresses and displacements in a single cell. The stiffness of each cell is assumed to be proportional to the density of the material it contains.

To reduce the gain from overlapping nodes one may assign

$$K = x_1^p k_0 + x_2^p k_0 + \dots + x_N^p k_0, \quad (2.6)$$

that is, the densities are raised to the power of p . For example, when $p=3$ the benefit from a density of 0.5 will be 0.125. p is called penalization power.

To sum up, the topology optimization problem where the objective is to minimize the compliance can be written as,

$$\begin{aligned} \min_x : C(x) = U^T K U &= \sum_{e=1}^N (x_e)^p u_e^T k_0 u_e \\ \text{subject to: } \frac{V(x)}{V_0} &= f, \\ &: KU = F, \\ &: 0 < x_{\min} \leq x \leq 1, \end{aligned} \quad (2.7)$$

where U and F are the global displacements and the force vectors, respectively, K is the global stiffness matrix, u_e and k_0 are the element displacement vector and stiffness matrix, respectively, x is the vector of design variables, x_{\min} is a vector of minimum relative densities (non-zero to avoid singularity), $N (= nelx \times nely)$ is the number of elements used to discretize the design domain, p is the penalization power, $V(x)$ and V_0 are the material volume and design domain volume, respectively and f is the prescribed volume fraction.

The optimization problem (2.7) could be solved using different approaches such as Optimality Criteria (OC) method [7] or the Method of Moving Asymptotes (MMA) [17]. For simplicity we describe a standard OC-method.

Following [7,8] a heuristic updating scheme for the design variables can be formulated as

$$x_e^{new} = \left. \begin{aligned} \max(x_{\min}, x_e - m) & \quad \text{if } x_e B_e^\eta \leq \max(x_{\min}, x_e - m) \\ x_e B_e^\eta & \quad \text{if } \max(x_{\min}, x_e - m) < x_e B_e^\eta < \min(1, x_e + m) \\ \min(1, x_e + m) & \quad \text{if } \min(1, x_e + m) \leq x_e B_e^\eta \end{aligned} \right\} \quad (2.8)$$

where x_e^{new} is the updated design vector, m is a positive constant which is the limit of change of the design vector, η is a numerical damping coefficient and B_e is found from the optimality condition as follows:

$$B_e = \frac{-\frac{\partial C}{\partial x_e}}{\lambda \frac{\partial V}{\partial x_e}}, \quad (2.9)$$

where λ is a Lagrangian multiplier that can be found by a well-known bi-sectioning algorithm. The element sensitivity (i.e. change in compliance with respect to change in design variable) of the objective function is found as

$$\frac{\partial C}{\partial x_e} = -p(x_e)^{p-1} u_e^T k_0 u_e. \quad (2.10)$$

In order to ensure existence of solutions to the topology optimization problem (2.7), some sort of restriction on the resulting design must be introduced. Often a filtering technique [18] is used. This filter works by modifying the element sensitivities as follows:

$$\frac{\partial \hat{C}}{\partial x_e} = \frac{1}{x_e \sum_{f=1}^N \hat{H}_f} \sum_{f=1}^N \hat{H}_f x_f \frac{\partial C}{\partial x_f}, \quad (2.11)$$

$$H_f = r_{\min} - \text{dist}(e, f), \quad F = \{f \in N \mid \text{dist}(e, f) \leq r_{\min}\}, \quad e = 1, 2, \dots, N, \quad (2.12)$$

where the operator $\text{dist}(e, f)$ is defined as the distance between centre of element e and the centre of element f . The convolution operator \hat{H}_f is zero outside the filter area (F). The convolution operator decays linearly with the distance from element f . r_{\min} is the filter size. Instead of the original sensitivities as in (2.10), the modified sensitivities of (2.11) are used in the optimality criteria update in (2.8).

Figure 2.2 shows an example of the pattern generated by topology optimization of a 32×20 structure whose left side is fixed and an unit force is applied at the position (30,20), initial volume fraction is taken as 0.5 and penalization power is taken as 3.

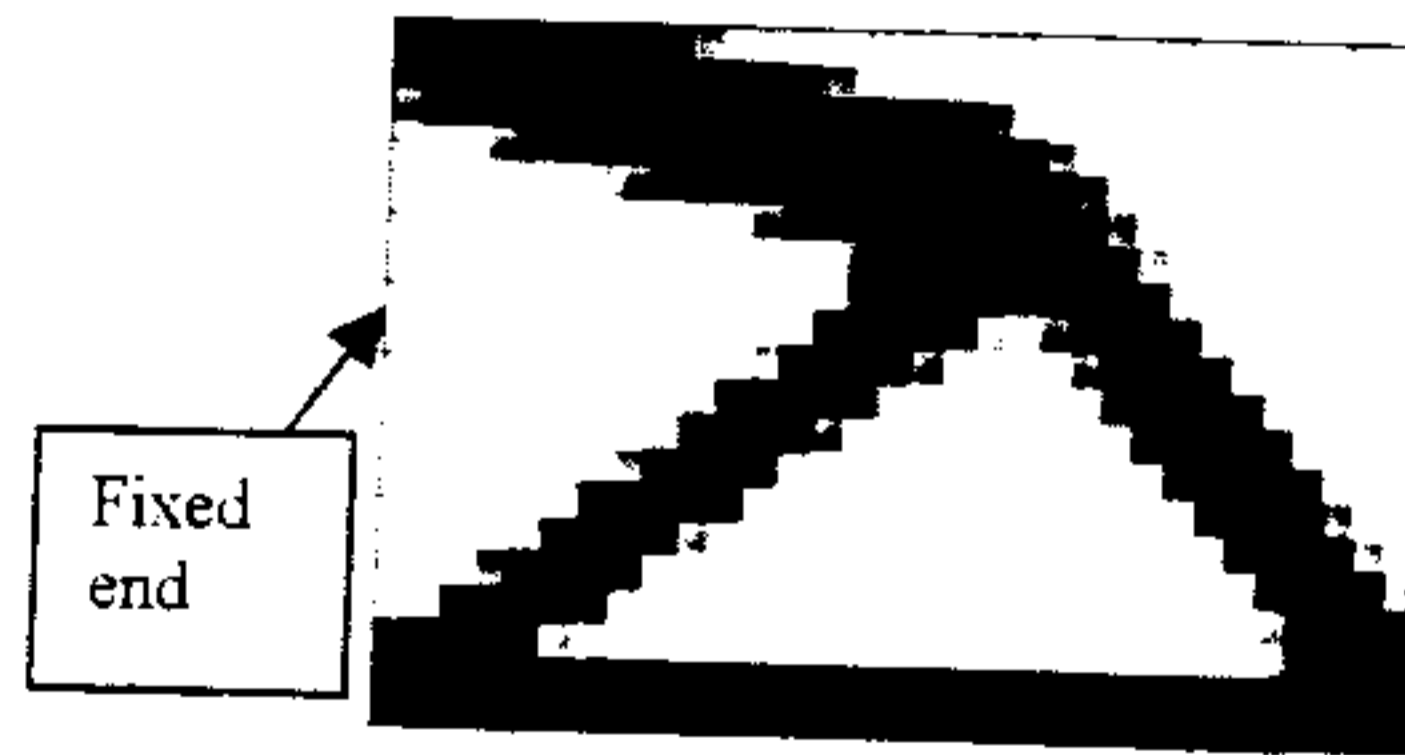


Figure 2.2: Pattern generated by topology optimization

For biologically motivated models, partial differential equations governing reaction diffusion generate various patterns. This is discussed next.

Section 2.2: Reaction-Diffusion Method

Human has a long time fascination for patterns generated through biological process, for example, patterns of zebra, jaguar, leopards etc. Alan Turing is the first to articulate an explanation of how the patterns of animals, like leopards, jaguars and zebras are determined [1]. Turing observed that patterns could arise as a result of instabilities in the diffusion of morphogenetic chemicals in the animals' skins during the embryonic stage of development. The basic form of a simple reaction-diffusion system is to have two chemicals (call them a and b) that diffuse through the embryo at different rates and that react with each other to either build up or break down the chemicals a and b . These systems can be explored in different dimensions. For example, we might use a one-dimensional system to look at segment formation in worms, or we could look at reaction-diffusion on a surface for spot formation. Following are the equations showing the general form of a two chemical reaction-diffusion system:

$$\frac{\partial a}{\partial t} = F(a, b) + D_a \nabla^2 a \quad (2.13a) \quad \text{and}$$

$$\frac{\partial b}{\partial t} = G(a, b) + D_b \nabla^2 b \quad (2.13b)$$

The equation (2.13a) says that the change of concentration of a at a given time depends on the sum of the function $F(a, b)$ of the local concentrations of a and b and the diffusion of a from places nearby. The constant D_a defines how fast a is diffusing, and the Laplacian $\nabla^2 a$ is a measure of how high the concentration of a is at one location with respect to concentration of a nearby in a local region. If nearby places have a higher concentration of a , then $\nabla^2 a$ is positive and a diffuses towards the center position of the local region. If nearby places have lower concentrations, then $\nabla^2 a$ is negative and a diffuses away from the center of the local region.

The key to pattern formation based on reaction-diffusion is that an initial small amount of variation in the concentrations of chemicals can cause the system to be unstable initially and to be driven to a stable state in which the concentrations of a and b vary across a surface. A set of equations that Turing has proposed for generating patterns in one dimension provides a specific example of reaction-diffusion [2]:

$$\Delta a_i = s(16 - a_i b_i) + D_a (a_{i+1} + a_{i-1} - 2a_i) \quad (2.14a)$$

$$\Delta b_i = s(a_i b_i - b_i - \beta_i) + D_b (b_{i+1} + b_{i-1} - 2b_i). \quad (2.14b)$$

Equations (2.14a) and (2.14b) are given for a discrete model, where a_i is the concentration of chemical a in the i -th "cell" in a line of cells and the concentrations of neighbors of i -th cell are a_{i-1} and a_{i+1} . The values for β_i are the sources of slight irregularities in chemical concentrations across the line of cells. Figure 2.3 illustrates the progress of concentration of chemical b across a line of 60 cells as its concentration varies over time. Initially the values of a_i and b_i are set to 4 for all cells along the line.

The value of β_i cluster around 12, with the values varying randomly by ± 0.05 . The diffusion constants are set to $D_a = .25$ and $D_b = .0625$, which means that a diffuses more rapidly than b , and we take parameter s as 0.03125.

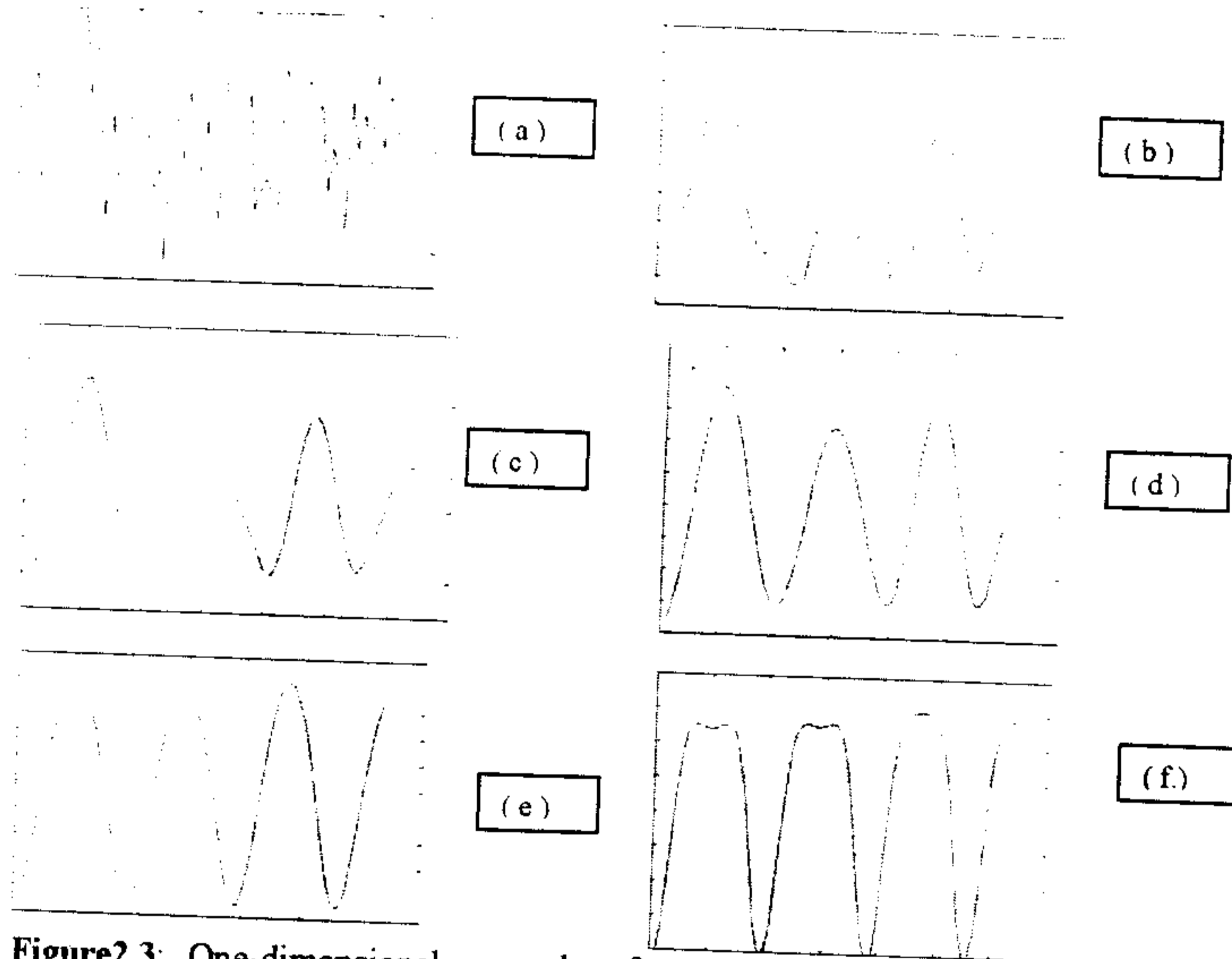


Figure 2.3: One-dimensional example of reaction-diffusion. From figures (a) to (f) concentration of chemical 'b' is shown in intervals of 4000 time steps.

Reaction Diffusion on a Two-dimensional Grid:

The reaction-diffusion system given in the previous example can also be simulated on a two-dimensional field of cells. The most common form for such a simulation is to have each cell as a square in a regular grid. The two-dimensional extension of equations (2.14a) and (2.14b) are:

$$\Delta a_y = s(16 - a_y b_y) + D_a (a_{i+1,j} + a_{i-1,j} + a_{i,j+1} + a_{i,j-1} - 4a_{i,j}) \quad (2.15a)$$

$$\Delta b_y = s(a_y b_y - b_{i,j} - \beta_{i,j}) + D_b (b_{i+1,j} + b_{i-1,j} + b_{i,j+1} + b_{i,j-1} - 4b_{i,j}). \quad (2.15b)$$

In this form, the value of Laplacian $\nabla^2 a$ (in equations (2.15a) and (2.15b)) at a cell is found by summing each of the four neighbouring values of a and subtracting four times the value of a at the cell. Each of the neighboring values for a are given the same weight in this computation because the length of the shared edge between any two cells is always the same on a square grid.

We simulate equations (2.15a) and (2.15b) on a 64×64 grid of cells. Notice that the valleys of concentration in b take the form of spots in two dimensions. It is the nature of this system that have high concentrations for a in this spot regions where b is low. Chemical a is called an inhibitor because high values for a in the spot region prevent other spots from forming nearby. In two-chemical reaction-diffusion systems the inhibitor is always the chemical that diffuses more rapidly. We can create spots of different sizes by changing the values of the constant s for the system. Small values for s cause the reaction part of the system to proceed more slowly relative to the diffusion and this creates larger spots. Larger values of s produce smaller spots. Figure 2.4 shows the simulation where initial values of a_i and b_i are set to 4 for all cells of the grid and $\beta = 12 \pm 0.1$. The diffusion constants are set to $D_a = 25$ and $D_b = 0.0625$, which means that a diffuses more rapidly than b , and we take parameter s as 0.05.

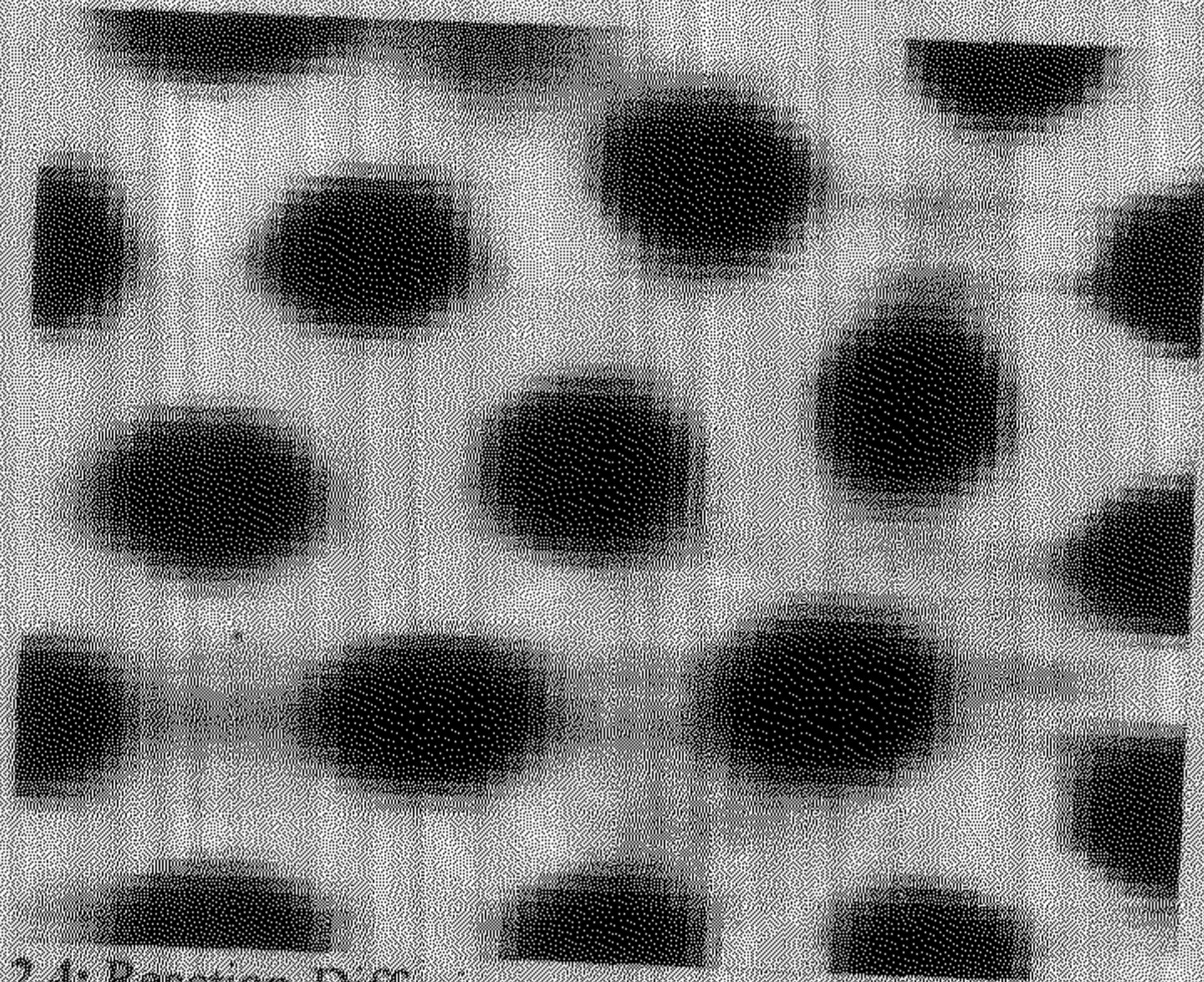


Figure 2.4: Reaction-Diffusion on a square grid. Formation of spots of cheetah.

Reaction-Diffusion for more than two chemicals:

Reaction-diffusion need not be restricted to two-chemical systems. For the generation of stripe patterns, Meinhardt had proposed a system involving five chemicals that react with one another [3]. The equations of Meinhardt's system with five chemicals are as follows:

$$\frac{\partial g_1}{\partial t} = \frac{cs_2 g_1^2}{r} - \alpha g_1 + D_s \frac{\partial^2 g_1}{\partial x^2} + \rho_0$$

$$\frac{\partial g_2}{\partial t} = \frac{cs_1 g_2^2}{r} - \alpha g_2 + D_s \frac{\partial^2 g_2}{\partial x^2} + \rho_0$$

$$\frac{\partial r}{\partial t} = cs_2 g_1^2 - cs_1 g_2^2 - \beta r$$

$$\begin{aligned}\frac{\partial s_1}{\partial t} &= \gamma(g_1 - s_1) + D_s \frac{\partial^2 s_1}{\partial x^2} + \rho_1 \\ \frac{\partial s_2}{\partial t} &= \gamma(g_2 - s_2) + D_s \frac{\partial^2 s_2}{\partial x^2} + \rho_1.\end{aligned}\quad (2.16)$$

In this system, the chemicals g_1 and g_2 indicate the presence of one or the other stripe color (white or black, for instance). The concentration of r is used to make sure that only one of g_1 and g_2 are present at any one location. Chemicals s_1 and s_2 assure that the regions of g_1 and g_2 are limited in width.

Figure 2.5 shows stripe pattern, which is formed by using equation (2.16) taking the parameter values as $\alpha=0.1$, $D_g=0.005$, $D_r=0.2$, $\rho_0=0.001$, $\rho_1=0.001$, $\gamma=0.04$, $g_1=g_2=s_1=s_2=\sqrt{1-\alpha}$, $\alpha=0.3$, $r=1$. and a small perturbation is given at the center of 80×80 grid.

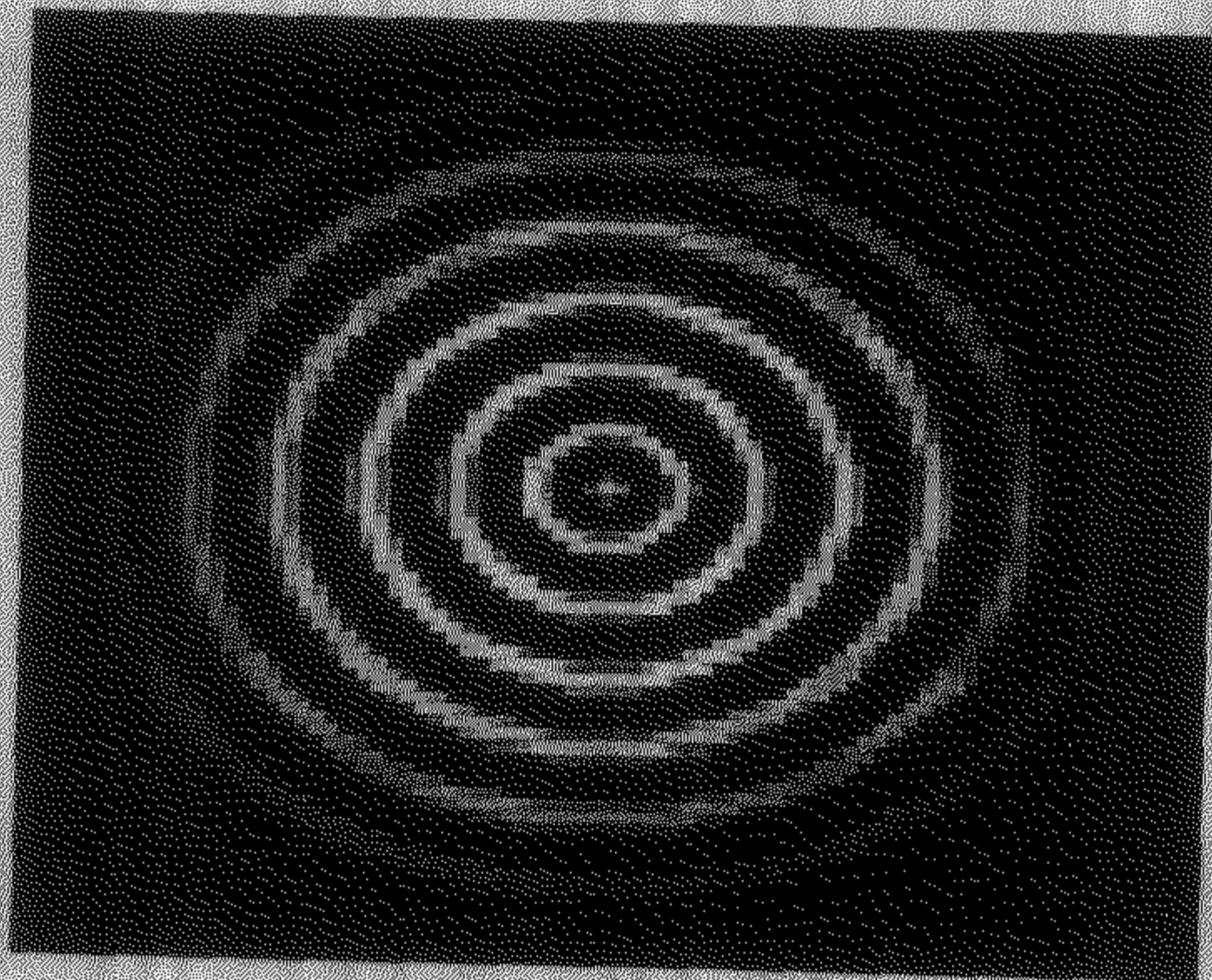


Figure 2.5: Reaction-Diffusion with five chemicals. Stripe pattern formation.

In a similar model as in Figure 2.5 and same parameter settings if we start with a random noise in the whole domain of 80×80 grid we get pattern as in Figure 2.6.



Figure 2.6 Random Stripe pattern

The challenge in pattern generation is to implement the compliance model and also the spot and stripe formation through level set framework. In the next chapter we discuss the basic concept of curve and surface evolution using level set analysis [8].

Chapter 3: Curve Evolution Models

Section 3.1: Basic concepts of Curve evolution:

The basic concepts of the theory of curve evolution are now presented. These concepts can be extended for surfaces as well.

We consider curves to be deforming in time. Let $C(\tilde{p}, t) : S^1 \times [0, T) \rightarrow R^2$ denotes a family of closed curves where t parameterizes the family and \tilde{p} parameterizes the curve. Assume that this family of curves obeys the following partial differential equation (PDE):

$$\frac{\partial C(\tilde{p}, t)}{\partial t} = \alpha(\tilde{p}, t) \vec{T}(\tilde{p}, t) + \beta(\tilde{p}, t) \vec{N}(\tilde{p}, t), \quad (3.1)$$

with $C_0(\tilde{p})$ as the initial condition. Here \vec{N} stands for the outward unit normal and \vec{T} denotes the unit vector along tangential direction. This equation (3.1) has general form and means that the curve is deforming with α velocity in the tangential direction and β velocity in the normal direction.

Note that for a general velocity \vec{V} , $\alpha = \langle \vec{V}, \vec{T} \rangle$ and $\beta = \langle \vec{V}, \vec{N} \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the vector 'dot' product. If we are interested in the geometry of the deformation, but not in its parameterization this flow can be further simplified following the result of Epstein and Gage [19]:

Lemma 3.1: If β does not depend on parameterization, meaning β is a geometrical intrinsic characteristic of the curve, then the image of $C(\tilde{p}, t)$ that satisfies equation (3.1) is identical to the image of the family of curves $C(p, t)$ that satisfies

$$\frac{\partial C(p, t)}{\partial t} = \beta(p, t) \vec{N}(p, t) \quad (3.2)$$

In other words, the tangential velocity does not influence the geometry of the deformation, just its parameterization. Basically, with equation (3.2) we do not know exactly where each point $C(p)$ is moving; this depends on the parameterization. We just know how the whole curve, as a geometric object, deforms. The art in curve evolution then becomes the search of the function β that solves a given problem. Figure 3.1 shows the independence of the geometry of the curve under the effect of tangential component of velocity pictorially.

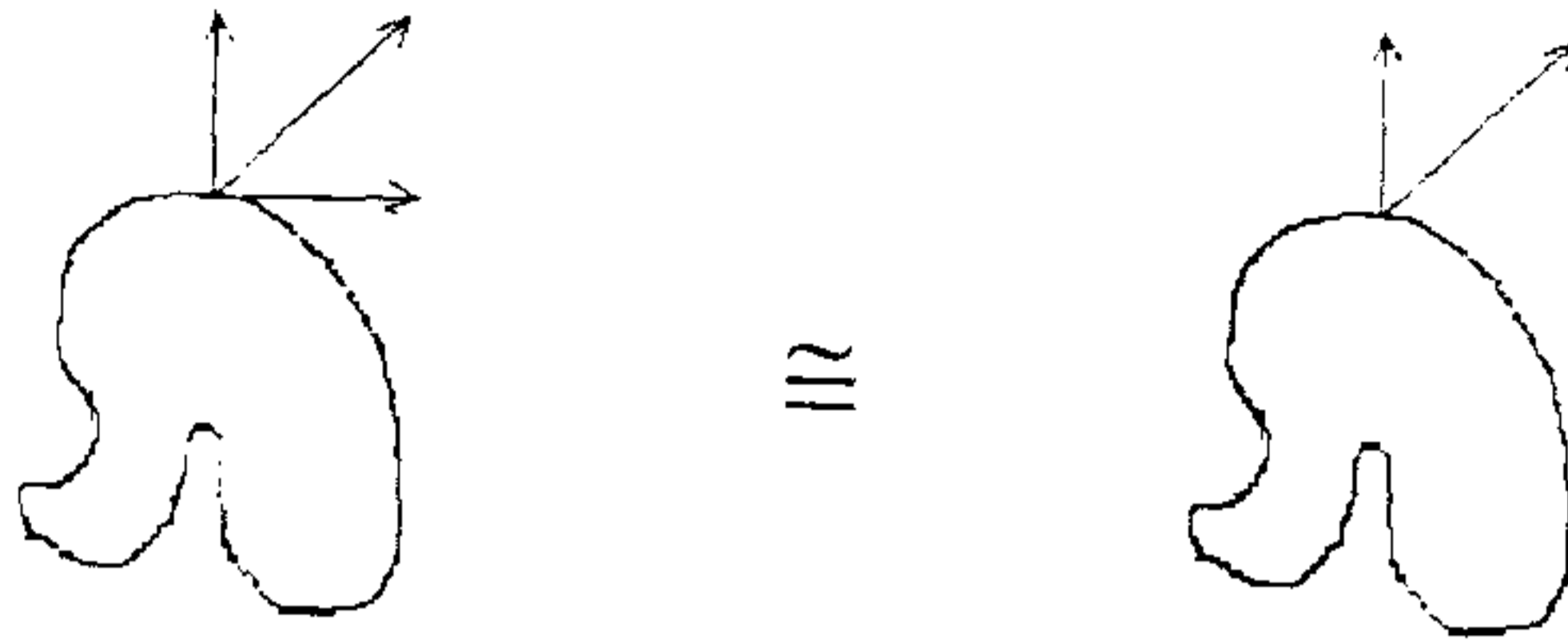


Figure 3.1: General form of a curve evolution. The velocity is decomposed into its tangential and normal components, the former not affecting the geometry of the flow, just its parameterization.

In the next section we describe how curve evolution can be embedded in level set framework.

Section 3.2: Level Set Method:

A number of problems need to be solved when we are implementing curve evolution such as equation (3.2):

1. Accuracy and stability. The numerical algorithm must approximate the continuous flow, and the algorithm must be robust. A Lagrangian approximation of the curve, based on moving particles along the curve requires an impractically small time step to achieve stability. The basic problem is that the marker particles² on the evolving shape can come very close together or very far away during the deformation. These can be solved by a frequent redistribution of markers, altering the motion of curve in a non-obvious way.

2. Developments of singularities. If, for example $\beta = 1$ in equation (3.2), singularities may develop. The question is how to continue the evolution after singularities appear. The solution is that we need a scheme that finds the right weak solution when singularities are present in the flow.

3. Topological Changes. When a curve is deforming, its topology can change (split or merge). Tracking the possible topological changes with marker particles is an almost impossible task, or at least incredibly hard to implement and computationally demanding at run time.

These problems lead to development of level set techniques, which are presented now. The technique for 2D deformable curves is described; the extension to higher-dimensional surfaces is straightforward.

² In marker particle technique, the parameterization of the curve is discretized into a set of marker particles whose positions at any time are used to reconstruct the front

Level set method is based on the following observation due to Osher and Sethian [8]: "A curve can be represented implicitly as the zero level set of a function in higher dimension". For a 2D deformable curve, zero level set is the intersection of the higher dimensional (3D) function and the $z = 0$ plane. In the following paragraph the method of embedding of the curve in level set surface is described mathematically.

Let, as before, $C(p, t) : S^1 \times [0, T) \rightarrow R^2$ denotes a family of closed (embedded) curves where time t parameterizes the family and p parameterizes the curve. At different instant of time t we get different curve of the family. Assuming that this family of curves obeys the following PDE:

$$\frac{\partial C(p, t)}{\partial t} = \vec{V} = \beta(p, t) \vec{N}(p, t), \quad (3.3)$$

with $C_0(p)$ as the initial condition.

Let us represent a curve at time t , which is satisfying the differential equation (3.3), as the zero level set of an embedding function $\Phi(x, y, t) : R^2 \times [0, T) \rightarrow R$ as

$$L(x, y, t) = \{(x, y) \in R^2 : \Phi(x, y, t) = 0\}. \quad (3.4)$$

The initial curve C_0 is represented by the intersection of an initial function $\Phi_0(x, y)$ and x - y plane. For example, we can consider the signed distance function $d(x, y)$ from a point on the plane to the curve C_0 (negative in the interior and positive in the exterior of C_0), and $\Phi_0 = d(x, y)$ as the initial level set function. Figure 3.2 shows a two dimensional curve and its level set function which is taken as signed distance function. Zero level set, which is the intersection of the level set function and the x - y plane, is shown in the figure by red curve.

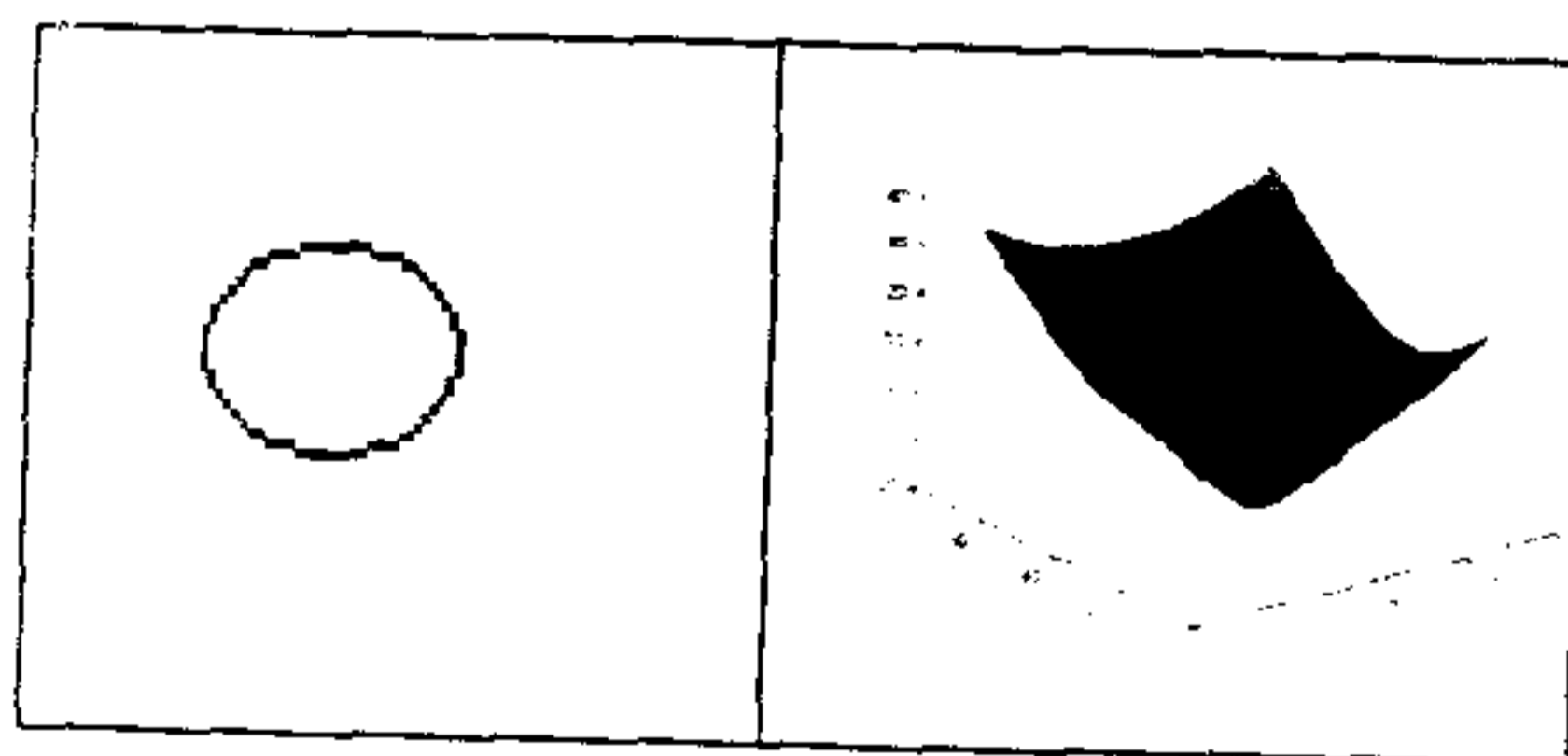


Figure 3.2:(a) The original curve lies in x - y plane. (b) Corresponding level set function.

We now have to find the evolution of Φ such that $C(x, y, t) = L(x, y, t)$. that is, the evolution of C coincides with the evolution of the level sets of Φ . By differentiating (3.4) with respect to t , we have

$$\nabla\Phi(L, t)\frac{\partial L}{\partial t} + \frac{\partial\Phi}{\partial t}(L, t) = 0. \quad (3.5)$$

We have $\frac{\nabla\Phi}{\|\nabla\Phi\|} = \vec{N}$, (3.6)

where \vec{N} is the normal to the level set L (the sign depends on the assumed convention for the direction of the normal). This equation combines information from the function Φ (left hand side) with information from the planar curve (right hand side).

To have $C=L$, we must have

$$\frac{\partial L}{\partial t} = \vec{V} = \beta\vec{N}. \quad (3.7)$$

Combining the equations (3.5), (3.6) and (3.7) we have,

$$\begin{aligned} 0 &= \nabla\Phi.\vec{V} + \Phi_t \\ &= \nabla\Phi.\beta\vec{N} + \Phi_t \\ &= \nabla\Phi.\beta\left(\frac{\nabla\Phi}{\|\nabla\Phi\|}\right) + \Phi_t \\ &= \beta\|\nabla\Phi\| + \Phi_t. \end{aligned}$$

Alternatively,

$$\frac{\partial\Phi}{\partial t} = -\beta\|\nabla\Phi\|. \quad (3.8)$$

Recapping, when a function is moving according to equation (3.8), its level sets, all of them, are moving according to equation (3.3). A number of comments are worth mentioning regarding this formulation:

1. We see once again that this time, because of the dot product in the derivation leading to (3.8), only the normal component of the velocity affects the flow.
2. As evolution of level set depends on the normal component of the velocity, the level set formulation is a parameterization free formulation, it is written in a fixed (x, y) coordinate system. For this reason level set formulation is called Eulerian formulation.
3. A number of questions must be asked when the above derivation is introduced: (i) Is the deformation independent of the initial embedding Φ_0 ? (ii) What happens when the curve is not smooth, and classical derivatives cannot be computed? Fortunately, these

questions have been answered in the literature [8]. For a large class of initial embeddings, the evolution is independent of them. The classical solution of equation (3.8), if it exists, coincides with the classical solution of equation (3.3). In addition, when singularities are developed, then the theory of viscosity solutions is used. This automatically gives a generalization of the of the curve flows of equation (3.3) when the curve becomes singular and notions such as normal are not well defined.

4. Embedding the curve in a higher-dimensional function automatically solves topological problems. Although the curve can change its topology, the topology of Φ is fixed; so there is no need to track topological changes, the evolution of Φ is implemented, and the changes in topology of the curve are derived when the corresponding zero level set is computed.

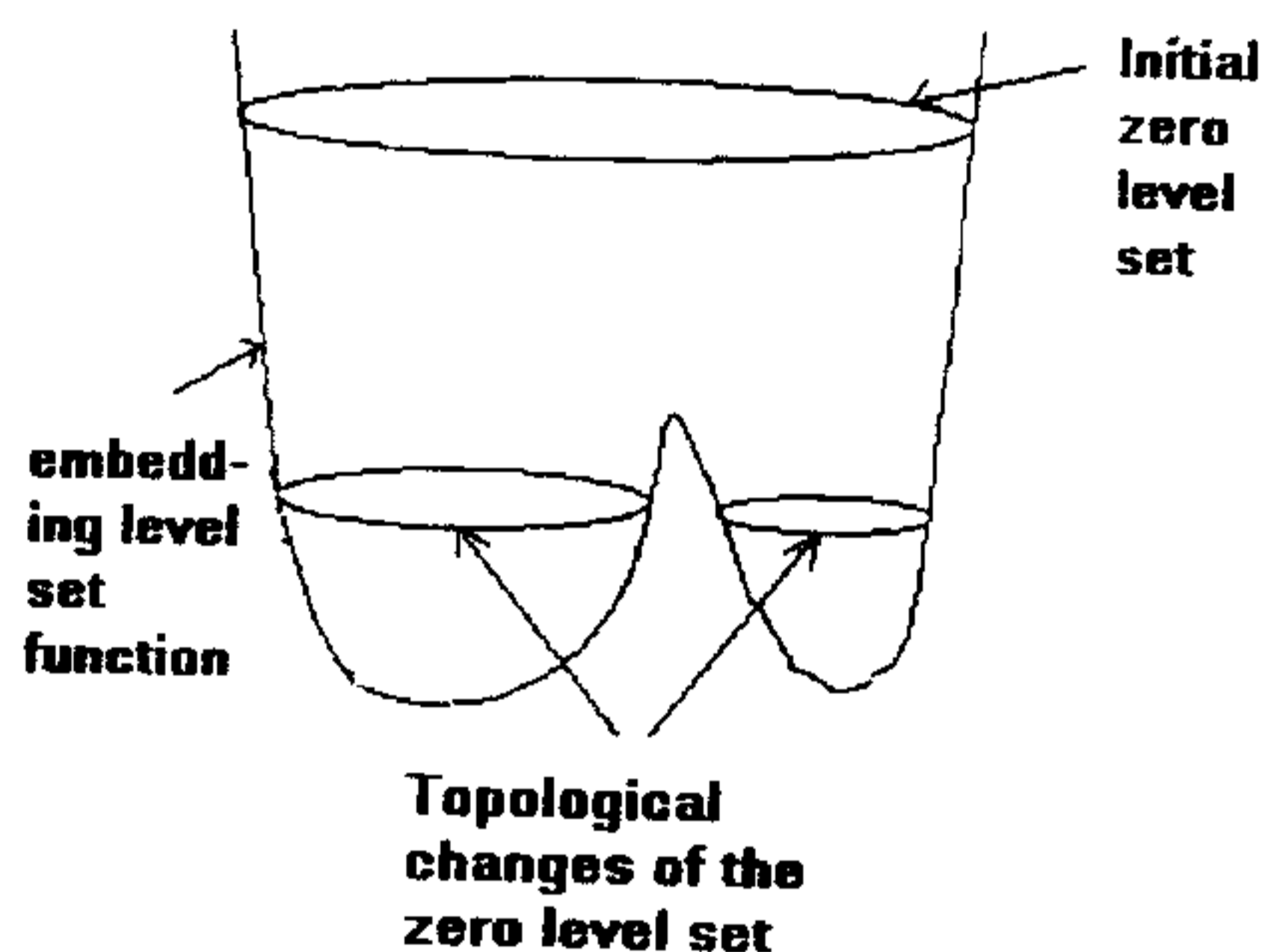


Figure 3.3: Topological changes of the zero level set

Section 3.3: Numerical Implementation of Level Set Flows:

To complete the picture, we need to show how to numerically solve equation (3.8). This is not a straightforward case, as for example, we have to make sure that in the case of singularities the correct weak solution is picked up by the numerical implementation. A brief description of the basic numerical technique is now presented:

To solve numerically we discretize the domain of level set function into a uniform grid of spacing h . At grid node (i, j) we use the notation Φ_{ij}^n to approximate the solution $\Phi(ih, jh, n\Delta t)$, where Δt is the time step. Using this notation one might write equation (3.8) in discrete form as follows:

$$\frac{\Phi_{ij}^{n+1} - \Phi_{ij}^n}{\Delta t} + \beta \left| \nabla_{ij} \Phi_{ij}^n \right| = 0. \quad (3.9)$$

Here, a forward difference scheme in time has been used, and $\left| \nabla_{ij} \Phi_{ij}^n \right|$ represents some appropriate finite difference operator for the spatial derivative. An explicit finite difference approach is described in the next section.

Up-Wind Computation Schemes:

The discrete solution to the Hamilton-Jacobi equation (3.8) is computed using finite differences over discrete time steps and on a discrete grid over the level set function. A highly robust and accurate computational method was developed by Osher and Sethian [8]. Based on the notion of weak solutions and entropy limits [8], a so called "up-wind scheme" is proposed to solve (3.8) with the following update equation

$$\Phi_{ijk}^{n+1} = \Phi_{ijk}^n - [\max(\beta_{ijk}, 0) \nabla^+ + \min(\beta_{ijk}, 0) \nabla^-], \quad (3.10)$$

with

$$\nabla^+ = \left[\begin{array}{c} \max(D_{ijk}^{-x}, 0)^2 + \min(D_{ijk}^{+x}, 0)^2 + \\ \max(D_{ijk}^{-y}, 0)^2 + \min(D_{ijk}^{+y}, 0)^2 + \\ \max(D_{ijk}^{-z}, 0)^2 + \min(D_{ijk}^{+z}, 0)^2 \end{array} \right]^{1/2} \quad (3.11)$$

$$\text{and } \nabla^- = \left[\begin{array}{c} \max(D_{ijk}^{+x}, 0)^2 + \min(D_{ijk}^{-x}, 0)^2 + \\ \max(D_{ijk}^{+y}, 0)^2 + \min(D_{ijk}^{-y}, 0)^2 + \\ \max(D_{ijk}^{+z}, 0)^2 + \min(D_{ijk}^{-z}, 0)^2 \end{array} \right]^{1/2} \quad (3.12)$$

Here, Δt is the time step, and $D_{ijk}^{\pm x}$, $D_{ijk}^{\pm y}$ and $D_{ijk}^{\pm z}$ are the respective forward and backward difference operators in the three dimensions of $x \in \mathbb{R}^3$ separately for a general 3D solid. In addition, the time step Δt must be limited to ensure the stability of the up-wind scheme (3.8). The Courant-Friedrichs-Lewy [8] condition requires Δt satisfying

$$\Delta t \max \left| \beta_{ijk} \right| \leq \Delta_{\min}, \quad (3.9)$$

where $\Delta_{\min} = \min(\Delta x, \Delta y, \Delta z)$ stands for the minimum grid space among the three dimensions [8].

Local Schemes of Level Set Computation:

The up-wind solutions produce the motion of level set models over the entire range of embedding, i.e. for all values of Φ in (3.8). Since the curve under consideration is defined to be, as zero level set, the calculation of solutions over the entire range of iso-

values is unnecessary. This forms the basis for “narrow-band” schemes that solves (3.6) in a narrow band of the grid nodes that surround the level set of interest [8]. While the up-wind scheme makes the level set method numerically robust, the narrow-band scheme makes its computational complexity proportional to the boundary area of the structure being optimized rather than the size of the volume in which it is embedded.

In the next chapter we introduce a pattern generation model under level set framework.

Chapter 4: Level Set Evolution Based Pattern Generation Model

After studying two pattern generation models in Chapter 2 and level set model in Chapter 3 our idea is to develop a similar model of pattern generation in level set framework. In this chapter we show how various interesting patterns can be generated using level set based model. Briefly speaking, our plan of pattern generation is as follows.

To evolve a level set influenced by some physical or biological phenomena for pattern generation. The shape optimization and reaction-diffusion based methods are utilized in this respect. The *stable* zero level set provides the desired pattern. Visualization of a rendered pattern also needs assignment of intensity values. This is further explored on the derived patterns using interpolation model. The main issue of pattern generation using level set framework is to find the velocity field with which the level set is evolving. Corresponding to each velocity field we get a new pattern.

In section 4.1 we show that shape optimization can be interpreted within level set framework to generate graphical patterns. We find a velocity field, which is obtained by minimizing the compliance as described in equation (2.7). The method of 'Shape Derivative' [15] is applied to find the velocity field. We first derive energy functional for shape optimization. Optimization of this energy function along gradient descent direction gives the velocity field of level set evolution.

In section 4.2, we derive a velocity field motivated by reaction-diffusion model of pattern generation. We first construct an energy functional corresponding to reaction-diffusion system. Again, the gradient descent based optimization provides evolution of level set leading to pattern generation.

Integration of pattern and intensity through interpolation is discussed in section 4.3.

We apply proposed model to regenerate discluded patterns in section 4.4.

Section 4.1: Level Set Evolution Based On Shape Optimization

In this section we first derive a velocity field due to structural optimization. We apply a method of 'Shape Derivative' [15] to get this velocity field. Then we use this velocity field to generate various patterns in level set framework. We first write the compliance minimization problem of (2.7), analytically.

In linear elasticity setting (i.e. stress strain relation of the material is linear), let $\Omega \subset R^2$ be a bounded open set occupied by a linear isotropic elastic material (i.e. elastic properties are independent of the orientation of the axes of coordinates) with elasticity coefficient A . For simplicity we assume that there is no volume forces but only surface

loadings g . The boundary of Ω is made of three disjoint parts $\partial\Omega = \Gamma \cup \Gamma_N \cup \Gamma_D$ with Dirichlet boundary conditions on Γ_D and Neumann boundary conditions on $\Gamma \cup \Gamma_N$. The boundary parts Γ_D and Γ_N are defined and only Γ is allowed to vary in the optimization process.

The displacement field u in Ω is the unique solution of the linearized elasticity system:

$$-div(Ae(u)) = 0 \quad \text{in } \Omega \quad (4.1a)$$

$$u = u_0 \quad \text{on } \Gamma_D \quad (4.1b)$$

$$(Ae(u))n = g \quad \text{on } \Gamma \cup \Gamma_N, \quad (4.1c)$$

with strain tensor $e(u)$ is given as $e(u) = \frac{1}{2}(\nabla u + \nabla^t u)$, t denotes the transpose operator.

$Ae(u)$ is actually the stress tensor and is denoted by $\sigma(u)$. The prescribed initial value of u on Γ_D is u_0 . The unit normal direction to boundary $\partial\Omega$ is n . The objective function is denoted by $J(\Omega)$ and is defined by,

$$J(\Omega) = \int_{\Gamma \cup \Gamma_N} g u ds = \int_{\Omega} Ae(u) e(u) dx. \quad (4.2)$$

To take into account the weight of the structure, we introduce a positive Lagrange multiplier l and we minimize

$$\inf_{\Omega} J(\Omega) + l \int_{\Omega} dx. \quad (4.3)$$

In general (4.3) admits a minimizer (i.e. the problem is well-posed) only if some geometrical and topological restrictions on the shape are enforced [20].

In the next part we define shape derivative and state some results of shape derivative proved by Murat and Simon [15]. We compute shape derivative of the objective function (4.2) in a continuous framework to get the required velocity field.

Framework of Murat-Simon:

Let Ω_0 be a reference domain. Consider its variation $\Omega = (Id + \theta)\Omega_0$ with $\theta \in W^{1,\infty}(R^2; R^2)$. $W^{1,\infty}(R^2; R^2)$ is the space of all mappings from R^2 to R^2 which are differentiable infinitely many times and Id is the identity mapping in $W^{1,\infty}(R^2; R^2)$.

The set $\Omega = (Id + \theta)\Omega_0$ is defined by, $\Omega = \{x + \theta(x) | x \in \Omega_0\}$. The vector field $\theta(x)$ is the displacement of Ω_0 . We consider the following definition of shape derivative as the Frechet derivative [14]:

Definition 4.1: Let T be an operator on a normed space X into another normed space Y . Given $x \in X$, if a linear operator $dT(x) \in B[X, Y]$ exists such that

$$\lim_{\|h\| \rightarrow 0} \frac{\|T(x+h) - T(x) - dT(x)h\|}{\|h\|} = 0,$$

then $dT(x)$ is said to be the Frechet derivative of T at x , and T is said to be Frechet differentiable at x . $B[X, Y]$ is the space of bounded linear operators on a normed space X into another normed space Y .

The operator $dT : X \rightarrow B[X, Y]$, which assigns $dT(x)$ to x is called the Frechet derivative of T .

Definition 4.2: The shape derivative of $J(\Omega)$ at Ω_0 is the Frechet derivative of in $W^{1,\infty}(R^2; R^2)$ of $\theta \rightarrow J((Id + \theta)\Omega_0)$ at 0.

$$\lim_{\|\theta\| \rightarrow 0} \frac{\|J((Id + \theta)(\Omega)) - J(\Omega) - J'(\Omega)\theta\|}{\|\theta\|} = 0$$

We apply the following results of shape derivative [15].

Result (4a): If $J_1(\Omega) = \int_{\Omega} f(x) dx$, then shape derivative of $J_1(\Omega)$ at Ω_0 is given by,

$$J_1'(\Omega_0)(\theta) = \int_{\Omega_0} \text{div}(\theta(x) f(x)) dx = \int_{\partial\Omega_0} \theta(x) \cdot n(x) f(x) ds$$

where $n(x)$ is the unit normal vector to $\partial\Omega_0$ (boundary of Ω_0) and for any $\theta \in W^{1,\infty}(R^2; R^2)$.

Result (4b): If $J_2(\Omega) = \int_{\partial\Omega} f(x) ds$, then shape derivative of $J_2(\Omega)$ at Ω_0 is given by,

$$J_2'(\Omega_0)(\theta) = \int_{\partial\Omega_0} \theta(x) \cdot n(x) \left(\frac{\partial f}{\partial n} + H f \right) ds,$$

where H is the mean curvature of $\partial\Omega_0$ which is defined by, $H = \text{div}(n(x))$. Applying results (4a) and (4b), we get shape derivative of compliance (4.2) is

$$J'(\Omega_0)(\theta) = \int_{\Gamma} \theta(x) \cdot n(x) \left(2 \left[\frac{\partial(g \cdot u)}{\partial n} + H(g \cdot u) \right] - Ae(u)e(u) \right) ds, \quad (4.4)$$

where Γ is the variable part of the boundary of the reference domain Ω_0 . $n(x)$ is the normal unit vector to Γ . H is the curvature of Γ and u is the solution of (4.1) in Ω_0 .

We now have all the necessary theoretical ingredients to describe a gradient method for the minimization of an objective function $J(\Omega)$. As we have seen, the general form of shape derivative is

$$J'(\Omega_0)(\theta) = \int_{\partial\Omega_0} \theta(x) \cdot n(x) v_0 ds, \quad (4.5)$$

where v_0 is given by,

$$v_0 = 2 \left[\frac{\partial(g \cdot u)}{\partial n} + H(g \cdot u) \right] - Ae(u)e(u). \quad (4.6)$$

By Cauchy-Schwartz inequality we find a gradient descent field, which minimizes the objective function as,

$$\theta = -v_0 n, \quad (4.7)$$

and then update the shape as

$$\Omega_t = (Id + \Delta t \theta) \Omega, \quad (4.8)$$

where $\Delta t > 0$ is a small descent step. We use vector field (4.7) in the normal direction of the boundary as our required velocity field for evolutions of the level set function. We generate various patterns by controlling the boundary conditions (4.1). Pattern generation technique can be written in an algorithm form as follows:

Step 1: Initialize the embedding level set function $\Phi(x,0)$ at $t = 0$ by the distance mapping of the boundary of the initial structure Ω . So $\Phi(x,0) = 0$ on $\partial\Omega$, greater than 0 inside $\partial\Omega$ and less than 0 outside $\partial\Omega$.

Step 2: The boundary conditions (4.1) are solved to find the displacement u .

Step 3: Calculate the speed function $v_0 = 2 \left[\frac{\partial(gu)}{\partial n} + H(gu) \right] - Ae(u)e(u)$ that defines the speed of propagation of all level sets of the embedding function $\Phi(x,t)$.

Step 4: Solve the following standard level set equation to update the embedding function $\Phi(x,t)$:

$$\frac{\partial\Phi}{\partial t} = v_0 \|\nabla\Phi\|.$$

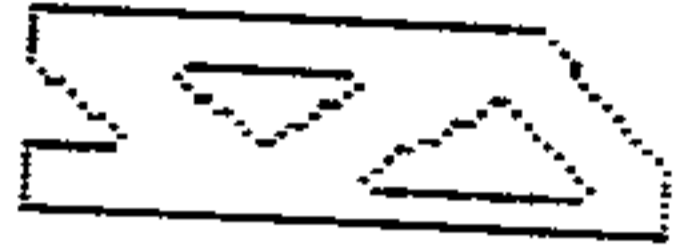

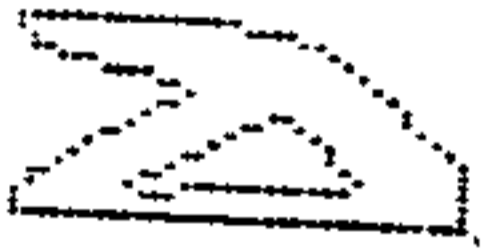


For implementation we use numerical technique proposed in Chapter 3.

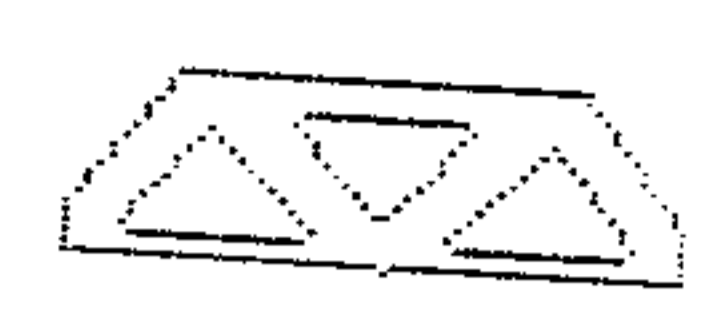


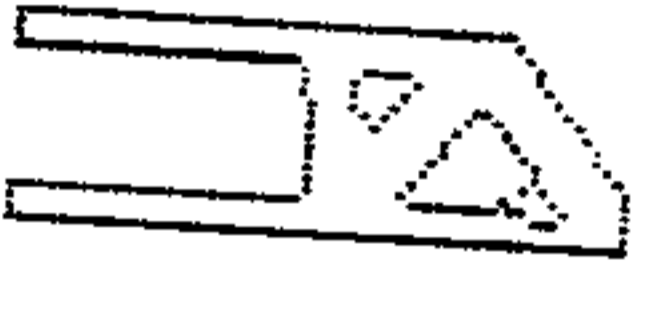

Step 5: Stop when we get a stable pattern. This condition is checked by the change in volume fraction of the structure in consecutive iteration. If change is negligible we assume stability condition has been arrived.

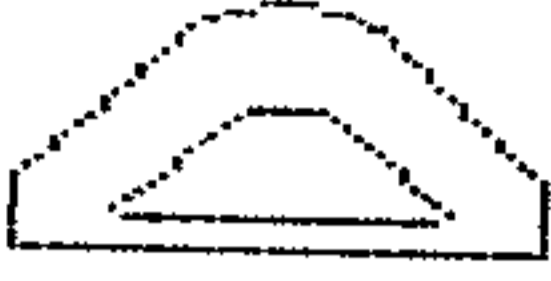
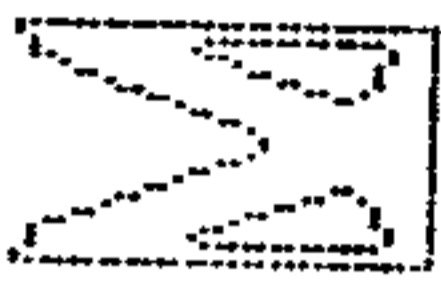


Examples of Patterns:

In Figure 4.1 we present several examples of patterns obtained with the proposed level set based pattern generation model. In left hand side of each pattern we write the

parameter values of the model, which we set to generate this pattern. We start with a rectangular pattern. $nelx$ is denoted as the total number of elements along a row and $nely$ is denoted as the total number of elements in a column. Vertices of every element are nodes. Both nodes and elements are numbered column wise from left to right. Furthermore, each node has two degrees of freedom (horizontal and vertical). Thus $F(2*n,1) = -1$, implies vertical unit force at the node number n , $F(2*n-1,1) = -1$, implies horizontal unit force at the node number n . '*' denotes the multiplication.

<p>(i). $nelx = 60, nely = 20,$</p> <p>Fixed degrees of freedom: $1:2:2*(nely+1),$ $2*(nelx+1)*(nely+1),$</p> <p>Load: $F(2,1) = -1;$</p>	
<p>(ii). $nelx = 60, nely = 20,$</p> <p>Fixed degrees of freedom: $1:2:2*(nely+1),$ $2*(nelx)*(nely+1): 2: 2*(nelx+1)*(nely+1)$</p> <p>Load: $F(2,1) = -1;$</p>	
<p>(iii). $nelx = 32, nely = 20,$</p> <p>Fixed degrees of freedom: $1:2*(nely+1),$</p> <p>Load: $F(2*(nelx+1)*(nely+1),1) = -1;$</p>	
<p>(iv). $nelx = 32, nely = 20,$</p> <p>Fixed degrees of freedom: $2*(nely+1),$ $2*(nelx+1)*(nely+1),$</p> <p>Load: $F((nelx+1)*(nely+1),1) = -1;$</p>	
<p>(v). $nelx = 61, nely = 31.$</p> <p>Fixed degrees of freedom: $1:2*(nely+1),$</p> <p>Load: $F(2*(nelx+1)*(nely+1)-(nely+1),1) = -1;$</p>	

<p>(vi). $nelx = 60, nely = 20,$</p> <p>Fixed degrees of freedom: $2*(nely+1)-1, 2*(nely+1), 2*(nelx+1)*(nely+1),$</p> <p>Load: $F((nelx+1)*(nely+1)+(nely+1), 1) = 1;$</p>	
<p>(vii). $nelx = 60, nely = 20,$</p> <p>Fixed degrees of freedom: $2*(nely+1)-1: 2*(nely+1), 2*(nely+1)*(nelx+1),$</p> <p>Load: $F((nelx)*(nely+1)+2, 1) = -1;$</p>	
<p>(viii). $nelx = 60, nely = 20,$</p> <p>Fixed degrees of freedom: $1: 2*(nely+1), 2*(nelx)*(nely+1)+1: 2*(nelx+1)*(nely+1)$</p> <p>Load: $F((nelx)*(nely+1)+2, 1) = -1;$ $F((nelx+1)*(nely+1)+(nely+1), 2) = 1;$</p>	
<p>(ix). $nelx = 60, nely = 20,$</p> <p>Fixed degrees of freedom: $1: 2: 2*(nely+1), 2*(nelx+1)*(nely+1),$</p> <p>Load: $F((nelx)*(nely+1)+2, 1) = -1;$ $F((nelx+1)*(nely+1)+(nely+1), 2) = 1;$</p>	
<p>(x). $nelx = 60, nely = 20,$</p> <p>Fixed degrees of freedom: $1: 2*(nely+1), 2*(nelx+1)*(nely+1),$</p> <p>Load: $F((nelx)*(nely+1)+2, 1) = -1;$ $F((nelx+1)*(nely+1)+(nely+1), 2) = 1$</p>	

<p>(xi). $nelx = 45, nely = 45,$</p> <p>Fixed degrees of freedom: $2*(nely+1), 2*(nelx+1)*(nely+1);$</p> <p>Load: $F((nelx)*(nely+1)+2,1)=-1;$ $F((nelx+1)*(nely+1)+(nely+1),2)=1$</p>	
<p>(xii). $nelx = 30, nely = 30,$</p> <p>Fixed degrees of freedom: $1:2*(nely+1),$</p> <p>Load: $F(2*(nelx)*(nely+1)+2,1)=1;$ $F(2*(nelx+1)*(nely+1)+(nely+1),2)=-1$</p>	
<p>(xiii). $nelx = 30, nely = 30,$</p> <p>Fixed degrees of freedom: $1:2*(nely+1),$</p> <p>Load: $F(2*(nelx)*(nely+1)+2,1)=1;$ $F(2*(nelx+1)*(nely+1)+(nely+1),2)=-2$</p>	
<p>(xiv). $nelx = 60, nely = 20,$</p> <p>Fixed degrees of freedom: $2*(nely+1), 2*(nelx+1)*(nely+1);$</p> <p>Load: $F((nelx+1)*(nely+1)+(nely+1)+1,1)=-1;$ $F((nelx+1)*(nely+1)+(nely+1)-10,2)=-1;$</p>	

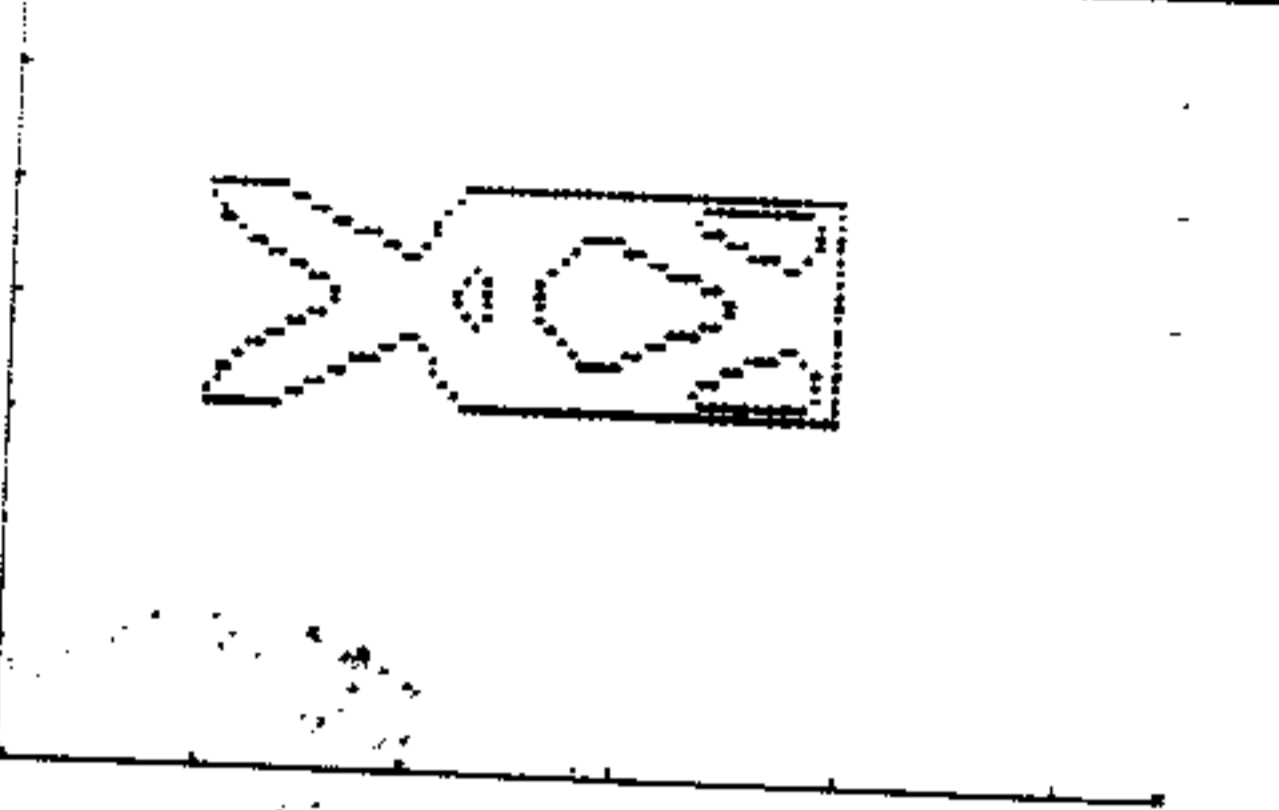
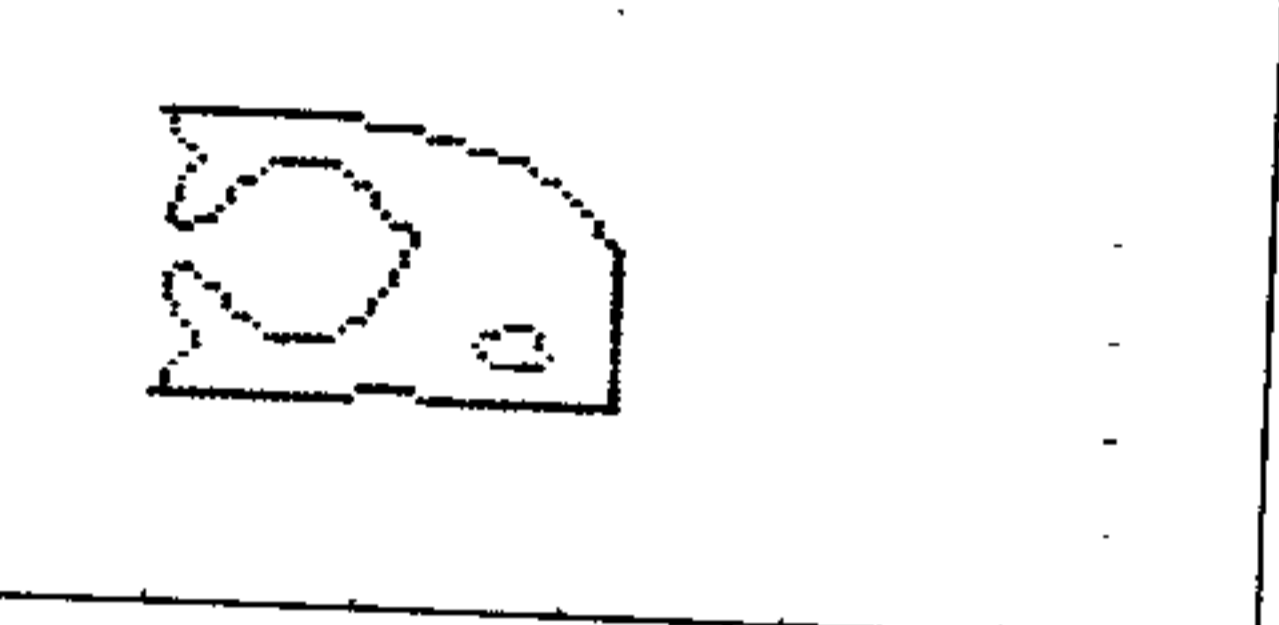
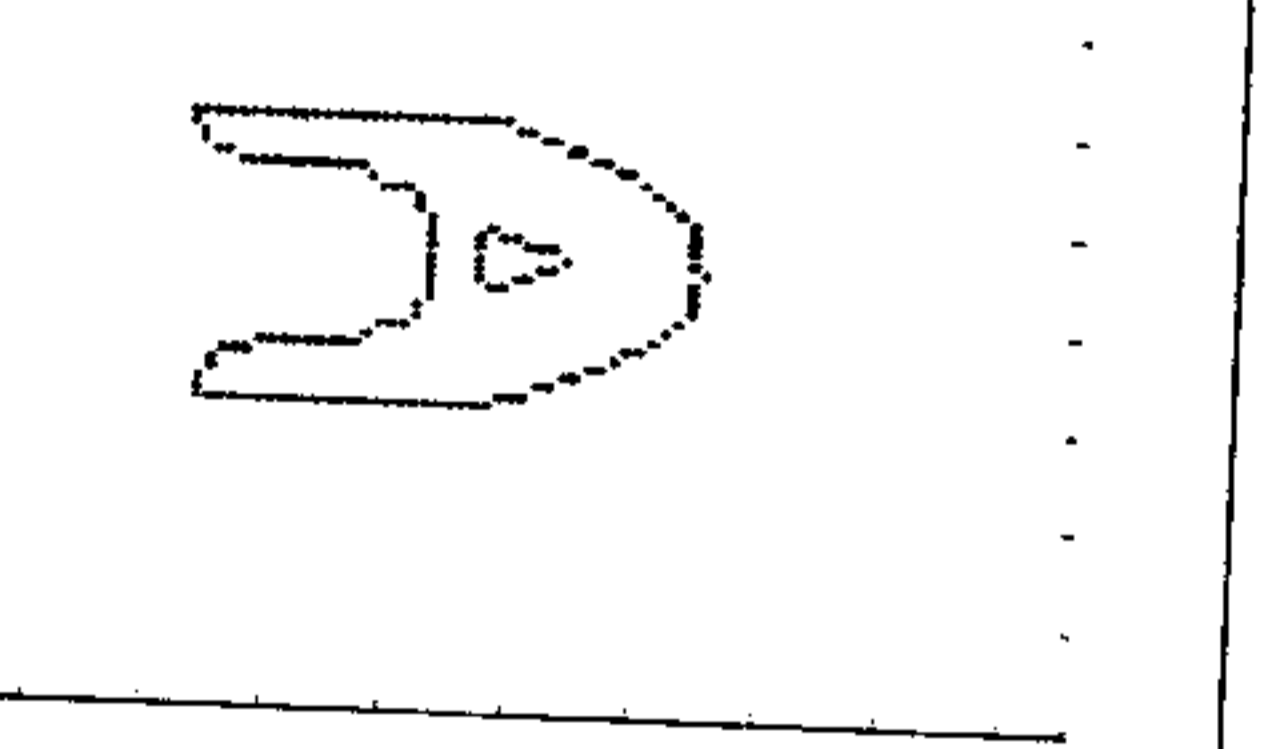
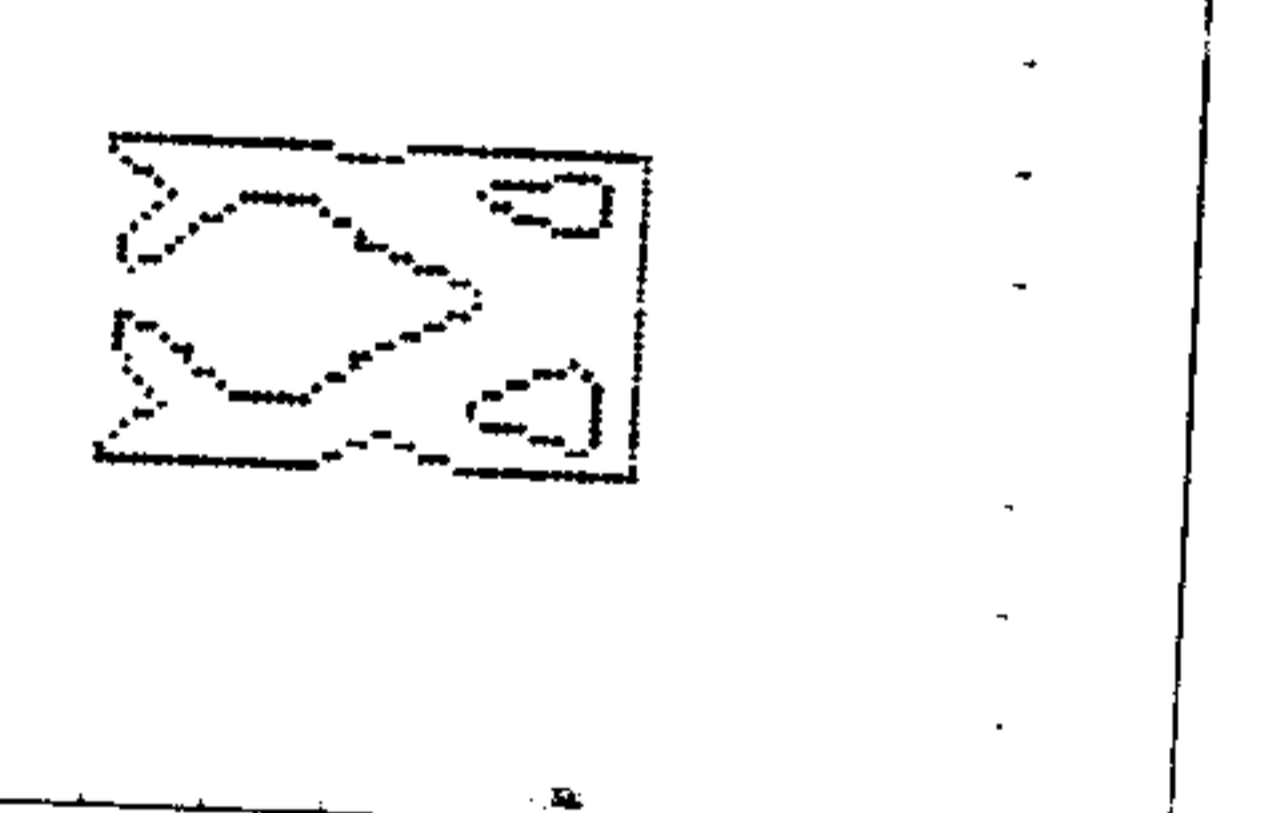
<p>(xv). $n_{elx} = 60, n_{ely} = 20,$</p> <p>Fixed degrees of freedom: $1:2*(n_{ely}+1), 2*(n_{elx})*(n_{ely}+1)+2, 2*(n_{elx}+1)*(n_{ely}+1)$</p> <p>Load: $F((n_{elx})*(n_{ely}+1)+2, 1) = 1;$ $F((n_{elx}+1)*(n_{ely}+1)+(n_{ely}+1), 2) = 1;$</p>	
<p>(xvi). $n_{elx} = 45, n_{ely} = 30,$ with a hole with center at $(n_{elx}/3, n_{ely}/2)$ and radius $n_{ely}/3.$</p> <p>Fixed degrees of freedom: $1:2*(n_{ely}+1),$</p> <p>Load: $F(2*(n_{elx}+1)*(n_{ely}+1), 1) = -1;$</p>	
<p>(xvii). $n_{elx} = 45, n_{ely} = 30,$ with a hole with center at $(n_{elx}/3, n_{ely}/2)$ and radius $n_{ely}/3.$</p> <p>Fixed degrees of freedom: $1:2*(n_{ely}+1),$</p> <p>Load: $F(2*(n_{elx})*(n_{ely}+1)+(n_{ely}+1)) = -1;$</p>	
<p>(xviii). $n_{elx} = 45, n_{ely} = 30,$ with a hole with center at $(n_{elx}/3, n_{ely}/2)$ and radius $n_{ely}/3.$</p> <p>Fixed degrees of freedom: $1:2*(n_{ely}+1),$</p> <p>Load: $F(2*(n_{elx})*(n_{ely}+1)+2, 1) = 1;$ $F(2*(n_{elx}+1)*(n_{ely}+1), 2) = -1;$</p>	

Figure 4.1 Patterns generated by level set model

Section 4.2: Level Set Evolution Based On Reaction-Diffusion

In this section we derive a velocity field to minimize the energy of a reaction-diffusion system of two chemicals. Let us consider a mechanical system made up of two unstable components. These components are chemicals having different levels of density distribution. The problem is to describe stable configurations and to characterize the interface between the two chemicals while the system reaches to a stable state. We characterize the interface between the chemicals by zero level set. This interface gives the boundary of pattern to be generated. We evolve the level set function with a velocity, which makes the system stable. A system is stable when the energy corresponding to the system is minimum. So to generate the pattern we have to find out the position of the zero level set in the stable configuration of the system.

The energy term corresponding to a reaction-diffusion system of two chemicals with densities a and b can be as [4],

$$E(t) = \int_B \|\nabla w\|^2 dx, \quad (4.9)$$

where the norm $\|\nabla w\|^2 = |\nabla a|^2 + |\nabla b|^2$ and B is the domain of reference. The initial boundary condition is given as:

$$(n \cdot \nabla)w = 0 \quad \text{on} \quad \partial B \quad (4.10a)$$

$$w(x,0) = w_0(x) \quad \text{on} \quad B. \quad (4.10b)$$

To find the gradient descent direction so that energy defined in (4.9) is minimized. Taking the first variation of the energy term with respect to time we get,

$$\begin{aligned} \frac{\partial E}{\partial t} &= \frac{\partial}{\partial t} \int_B (|\nabla a|^2 + |\nabla b|^2) dx \\ &= \int_B 2|\nabla a| \frac{\nabla a \cdot \nabla a_t}{|\nabla a|} dx + \int_B 2|\nabla b| \frac{\nabla b \cdot \nabla b_t}{|\nabla b|} dx \\ &= \int_{\partial B} \nabla a \cdot a_t \cdot \vec{n} ds - \int_B \text{div}(\nabla a) a_t dx + \int_{\partial B} \nabla b \cdot b_t \cdot \vec{n} ds - \int_B \text{div}(\nabla b) b_t dx \\ &= - \int_{\partial B} \text{div}(\nabla a) a_t dx - \int_{\partial B} \text{div}(\nabla b) b_t dx \\ &= \int_{\partial B} \langle -\text{div}(\nabla a), a_t \rangle dx - \int_{\partial B} \langle -\text{div}(\nabla b), b_t \rangle dx \end{aligned}$$

[Using boundary condition (4.10a)]

By Cauchy-Schwarz inequality the field for which energy $E(t)$ decreases most rapidly is given by,

$$\frac{\partial a}{\partial t} = \text{div}(\nabla a), \quad (4.11a) \text{ and}$$

$$\frac{\partial b}{\partial t} = \text{div}(\nabla b). \quad (4.11b)$$

We can take either of the above fields in the normal to the surface of concentration as our required velocity field of level set evolution.

So we write the pattern generation algorithm as follows:

Step 1: Initialize the embedding level set function $\Phi(x,0)$ at $t=0$ by the distance mapping of any closed curve in the domain B . So $\Phi(x,0)=0$ on ∂B , > 0 inside ∂B and < 0 outside ∂B .

Step 2: Calculate the speed function $\beta = \text{div}(\nabla a)$. This defines the speed of propagation of all level sets of the embedding function $\Phi(x,t)$.

Step 3: Solve the following standard level set equation to update the embedding function $\Phi(x,t)$:

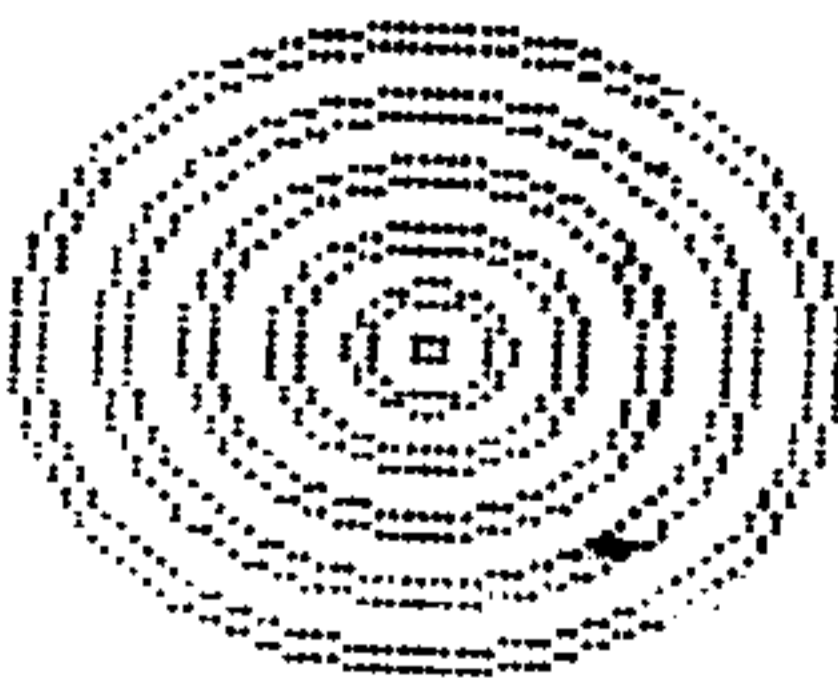
$$\frac{\partial \Phi}{\partial t} = -\beta \|\nabla \Phi\|.$$

For implementation we use numerical techniques proposed in Chapter 3.

Step 4: Stop when we get a stable pattern.

Examples of Patterns:

In Figure 4.2 we present some examples of patterns obtained by the proposed level set based pattern generation model.

<p>(i) Stripe Pattern using level set model: Iteration number = 1000</p>	
--	---

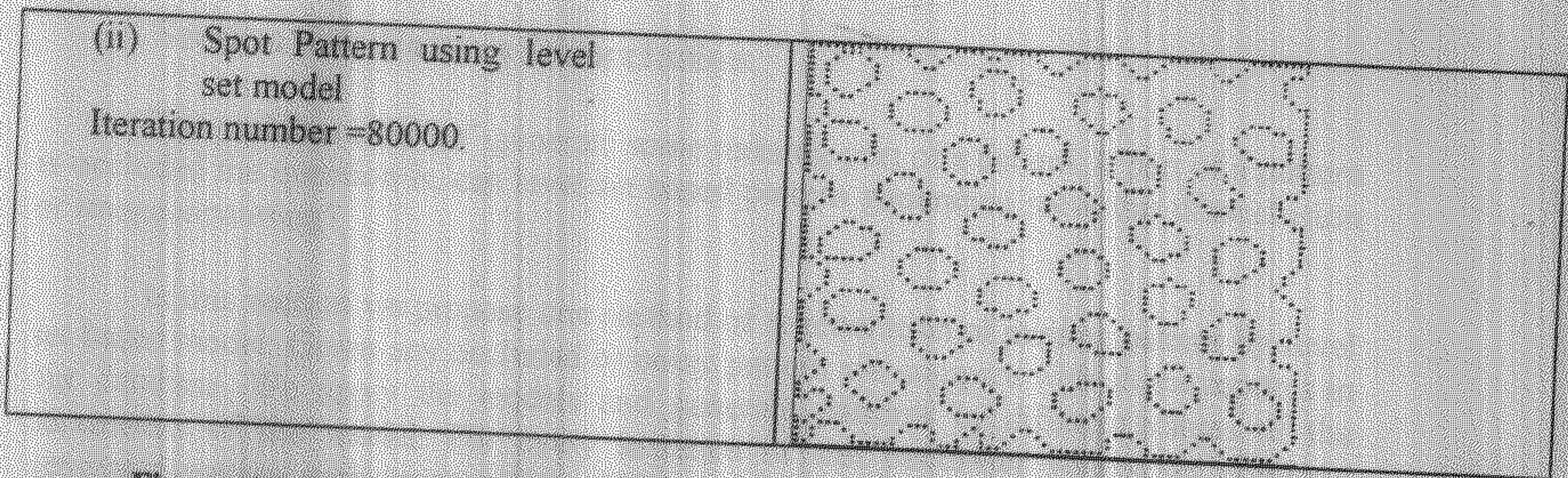


Figure 4.2 Patterns generated by level set model.

Section 4.3: Intensity Generation Model

The major point of pattern generation using level set model is how to handle the gray values for the pattern. In level set model of pattern generation we get boundary of a pattern. After getting this boundary we can apply different interpolation models to generate the gray values inside the boundary of the pattern. We can also associate a gray value to each point of the final level set surface, which is generated by the pattern generation algorithm. In figure 4.3 we apply a level set surface based gray value generation method to generate gray values in stripe pattern.

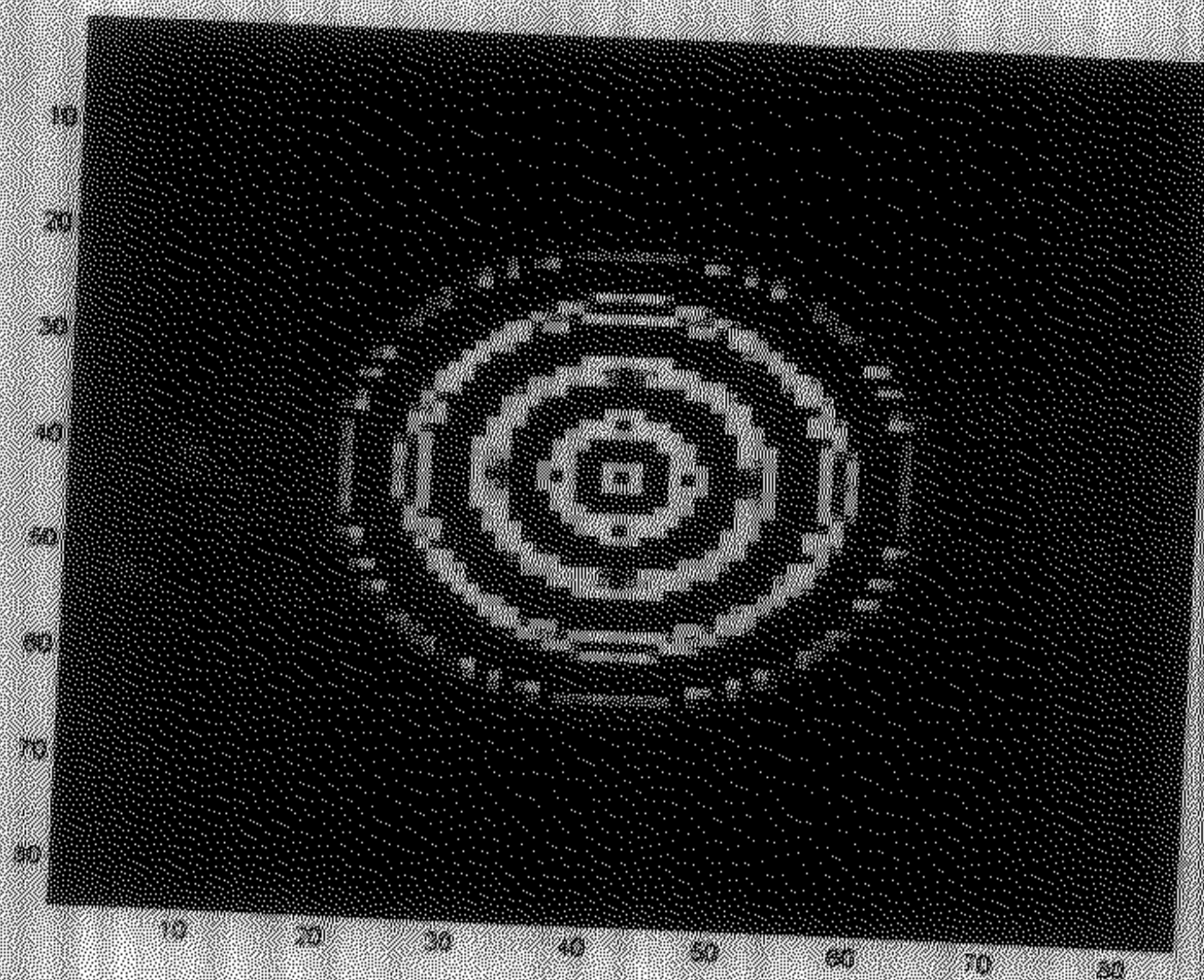
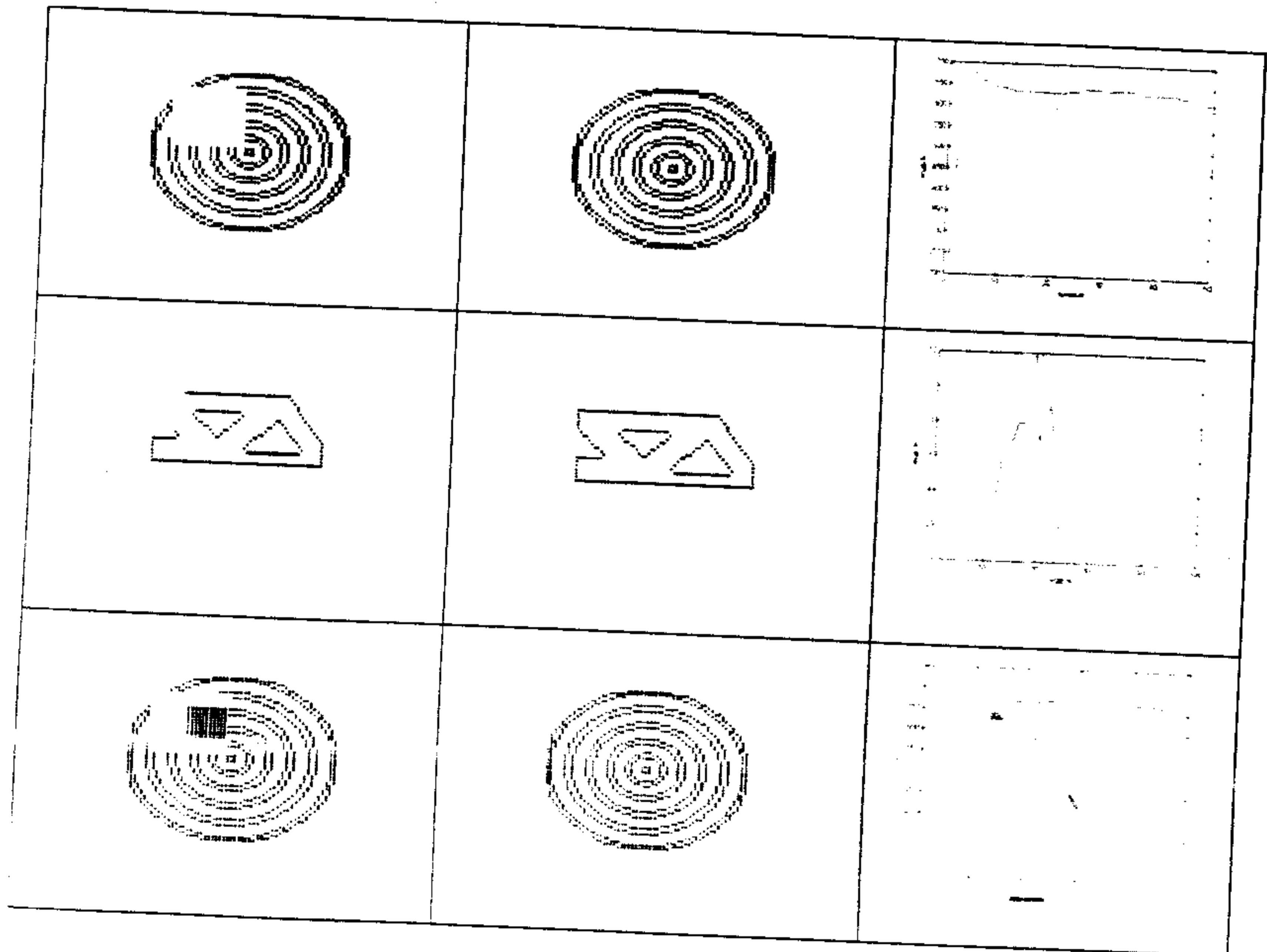


Figure 4.3: Gray value generation inside the stripe pattern

Section 4.4: Pattern Disocclusion

The problem is to "re"-generate some missing part of a pattern. If part of a pattern is occluded the question is, can level set based pattern generation solve this disocclusion or image inpainting?

In this section we propose one method of regeneration of missing pattern by level set based model. Given any occluded or noisy pattern we compare with all the models in our database. We apply the models and generate patterns by different values of the parameters of the model. For every instance of the parameter values we compare generated pattern with given pattern. We compare the patterns by finding a similarity measure between them. We choose those parameter values for which similarity measure is maximum. We claim that the pattern with these parameter values is the reconstructed pattern of the given pattern. In Figure 4.4 we give some examples of Pattern reconstruction, here number of iteration is taken as parameter which is varied for getting the most similar pattern corresponding to the occluded pattern. The similarity measure has been taken as sum of the matching portion of the curve. For binary pattern matrices we point wise multiply the matrices and then count the total number of matching points to get the similarity measure. We plot the similarity measure-iteration graph in each case. And show that each of these graphs has unique maxima.



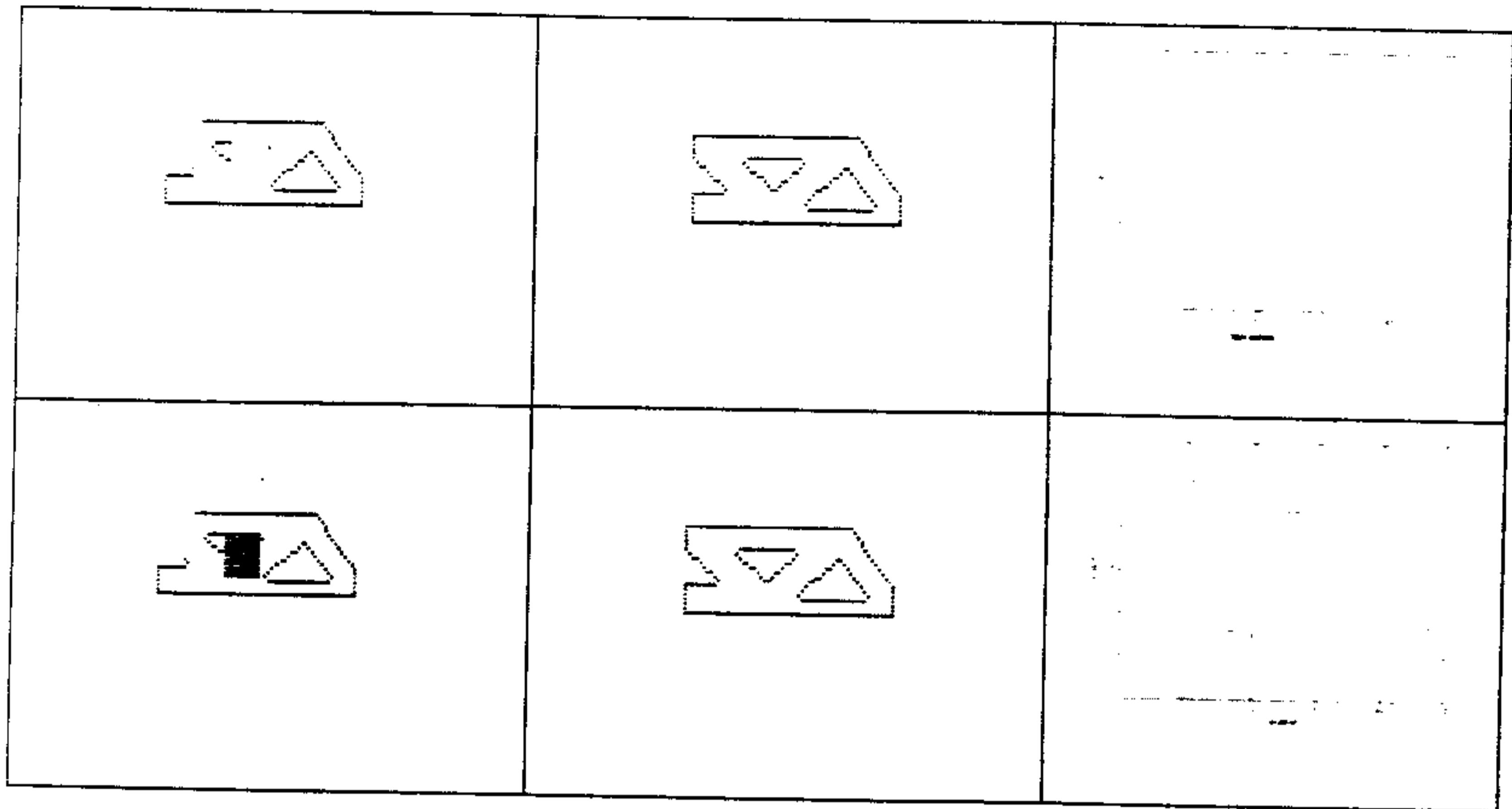


Figure 4.4: First column shows occluded patterns. Second column shows the original model. Third column shows the iteration-match plot.

Chapter 5: Conclusion

We have proposed a level set based pattern generation technique. Also we have discussed a pattern disocclusion technique. There is lot of scopes of exploring level set based technique for generating textured images and quasi-periodic patterns like human fingerprint. Intensity interpolation method can be modified so that generated patterns become more realistic. Level set based pattern generation model can also be applied in the area of surface mapping.

References

1. Turing, A, "The chemical basis of morphogenesis", *Phil. Trans B*, 237, 37-72, 1952.
2. "Computer Graphics", Volume 25, Number 4, July 1991
3. Hans Meinhardt, *Models of Biological Pattern Formation*, Academic press
4. Murray, J.D, *Mathematical Biology*, 3rd Ed, 2002.
5. O. Sigmund, *Struct. Multidisc. Optim*, 21, 120-127, Springer-Verlag, 2001.
6. M.P. Bendsoe, *Optimization of Structural Topology, Shape Material*, Springer, Berlin, 1997.
7. M.P. Bendsoe, "Optimal shape design as a material distribution problem", *Structural Optimization*, 1 193-202, 1989.
8. J.A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry. Fluid Mechanics, Computer Vision. and Material Sciences*, Cambridge University Press, 1999.
9. Guillermo Sapiro, *Geometrical Partial Differential Equations and Image Analysis*, Cambridge University Press, 1st Ed , 2001.
10. J.A. Sethian, A Weimann, "Structural boundary design via level set and immersed interface methods", *Journal of Computational Physics*, 163 (2) 489-528, 2000
11. Gregoire Allaire, Francois Jouve, Anca-Maria Toader, "A level set method for shape optimization", *C.R. Acad. Sci. Paris*, 2002
12. Michel Yu Wang, Xiaoming Wang, Dongming Guo, "A level set method for structural topology optimization", *Comput. Methods Appl. Mech. Engrg.* 192 (2003) 227-246
13. Sokolnikov, *Theory of Elasticity*
14. Siddiqi, a. h., *Functional Analysis with Applications*.
15. Murat, F. Simon, S. "Etdes de problems d'optimal design", *Lecture Notes in Computer Science*, 41, 54-62, Springer Verlag, Berlin, 1976.
16. Zienkiewicz OC, *Finite Element Method*. 3rd, Tata McGrow Hill, 1979.
17. <http://www.mai.liu.se/~anrie/top/>, Method of Moving Asymptote.
18. O. Sigmund, "On the design of Compliant Mechanisms using topology optimization", *Mechanics of structures and machines*, 25(4) 495-526, 1997
19. C.L.Epstein, Gage, "The Curve Shortening Flow", in *Wave Motion: Theoty Modelling and Computation*, Springer-Verlag, 1987
20. Chenais D., "On the existence of a solution in a domain identification problem", *J. Math. Anal. Appl.* 52, pp. 189-289, 1975
21. G. Aubert, P. Kornprobst, *Applied Mathematical Sciences*, Vol-147, Springer.