UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Implicit, Nonparametric Shape Reconstruction from Unorganized Points Using a Variational Level Set Method

Hong-Kai Zhao
Stanley Osher
Barry Merriman
Myungjoo Kang

February 1998
(Revised February 1999)

CAM Report 98-7

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90095-1555

Implicit, Nonparametric Shape Reconstruction from Unorganized Points Using A Variational Level Set Method‡

by

Hong-Kai Zhao*, Stanley Osher†, Barry Merriman†, and Myungjoo Kang†

Abstract

In this paper we consider a fundamental visualization problem which arises in computer vision, computer graphics and numerical simulation. The problem is to find a curve in two dimensions, or a surface in three dimensions which can be regarded as the shape represented by a set of unorganized points, and/or curves, and/or surface patches. We do not assume any knowledge of the ordering, connectivity or topology of the data sets or of the true shape. Only the location of each point or general Hausdorff distance to the data set is known. The key idea in our approach is to find an implicit nonparametric representation of the curve or surface on a fixed rectangular grid. With this representation of surfaces we can easily (a) find the closest point and distance from any point to the surface (useful in illumination and many other applications), (b) find the intersection curve of two surfaces which is guaranteed to lie on both surfaces in our representation, and (c) perform any Boolean operation (see Figure 17). We use a version of the variational level set method [ZCMO, ZMOW] to interpret the desired shape as one or more elastic membranes attached to the data set. We may initialize our problem with a single large membrane enclosing this set. We then apply a gradient descent algorithm to the energy of this membrane, which results in an interesting and simple motion of the shape involving curvature times

‡ Research supported by NSF and DARPA under grants NSF DMS-9706827, NSF DMS 9706566 and NSF DMS-9615854

* Dept. of Mathematics, Stanford University, Stanford, CA 94305-2125;
e-mail: zhao@math.stanford.edu

† Dept. of Mathematics, University of California, Los Angeles, CA 90095-1555

† e-mails: sjo@math.ucla.edu, barry@math.ucla.edu, mkang@math.ucla.edu

distance to the nearest data point as well as convection. We prove that there exists a local minimum which includes the data points before the shape disappears and also prove that if we get close enough to every data point in 2D in the special case when the data set consists only of discrete points, the final curve will be a piecewise linear interpolant. In 3D the final surface satisfies an interesting relationship involving its curvature and direction of its normal, i.e., it behaves like a weighted minimal surface. This surface looks much more pleasing than the usual polyhedral interpolant when the data points are not very dense or not evenly spaced. Since this is a variational approach it is also quite robust and can be used to denoise surfaces. These two properties can be extremely useful for 3D reconstruction from photographic images instead of from precise but expensive to obtain laser scanning data. We modify our basic functional to handle nonconvexities (which are more difficult to deal with in 2D than 3D). The modification amounts to adding a convection term which involves the weighted distance from the membrane to a collection of nearby data points which do not see the membrane, i.e. the interpolation error. We use finite difference schemes on fixed Cartesian grids to approximate the differential equations of motion. The method allows for the addition of new data points easily. Moreover, the data points and corresponding shape can move in time. As usual, the level set method handles complicated geometries which include topological changes and singularities with no difficulties, and local and global geometric properties of the resulting shape are easily computed. The algorithm is extremely simple and fast if the initial shape is close to each of the data points. This happens, for example, for dynamic problems with points moving in time. Finally, a simple postprocessing may be used to locally smooth the resulting shape.

1. Introduction

We intend to reconstruct a shape corresponding to a set of discrete points when (i) no information is known about the ordering or connections of the points and (ii) the shape is not necessarily simply connected (there may be several disconnected components), (iii) there may be edges and/or corners at unspecified locations and (iv) there may be other information to be taken into account, such as geometric constraints, noise, or uncertainty in the data. This basic problem arises in computer vision, robotic navigation, pattern recognition, data analysis, animation, computer simulation and elsewhere. There is a very large literature on this subject, e.g. [BW, Bo, Br, CG, EM, MT, HSIW].

The approaches to this problem can be characterized as either implicit or explicit. In the explicit approach one often uses a local spline approximation, where the knowledge of the ordering and connections between the data points is crucial. Alternatively one needs a global parameterization. These requirements are quite difficult to enforce under our general assumptions.

We propose to use an implicit representation of the interpolating surface (we use the singular even though our result may consist of several disjoint surfaces) on a structured grid. We regard the surface as the zero level set or surface of a function which is defined on a rectangular grid. The most convenient choice of this function is the signed distance to the surface. This approach has not been commonly used, except for special shapes. In this paper we first construct a variational problem involving an energy which is a surface integral over candidate shapes. We obtain a local minimum (which has a large domain of attraction) which is the interpolating surface. The method involves the variational level set method derived in [ZCMO]. The final numerical output consists of all the values of the distance to the desired surface evaluated within a small neighborhood of the surface, those values whose distance in magnitude is less than $c\Delta x, \Delta x$ the grid size and c typically around 5 or 6. The numerical procedure uses finite difference approximations devised, in part, to solve the equations of level set motion [OSe, OSh, ZCMO].

We consider an elastic membrane attached to a set of data points. This is a shape which is easily recognized by the human eye. We obtain it by thinking of the data as a set of electric charges and we create a potential function that depends only on the locations of the charges. The initial elastic membrane is chosen to enclose all the data points, but is otherwise arbitrary. It is attracted to these charges by minimizing both the electrostatic potential and the elastic energy. The elasticity of the membrane will control the regularity of the surface. Its minimization leads to topological changes whenever appropriate. The final shape involves a balance between elastic and electrostatic forces. We also have a standard property for the model: that the density of data points should increase with the local curvature in order to resolve local features. We can easily add other geometric constraints, either local, such a given direction of the normal or global, such as fixed volume, to our energy minimization. Noise can be controlled using a variational method similar to that of [ROF], used for images. We can also incorporate uncertainties in the data by weighting the attraction each point has to the membrane. After we obtain the signed distance we can easily find the closest point on the surface from any point in space, as well as all geometric properties of the surface – surface area, volume, normal curvature, or number of connected components. We can also easily generate body fitted coordinates with this approach. In the special case when our initial guess is close to the minimizer the method reduces to a simple and very fast advection algorithm. This is important for tracking dynamic motion, e.g. of certain solid/fluid interaction problems[1]. The final shape may be easily smoothed locally via a postprocessing. That is we may move this curve or surface normal to itself with normal velocity $a + b\kappa$ for κ mean curvature. We may do this for a few iterations, where $a(x)$ and $b(x) \geq 0$ are user specified.

2. The Variational Level Set Formulation

2.1 The Level Set Method

Our basic numerical approach is derived from the level set method of Osher and Sethian [OSe]. This is a very versatile and powerful technique for capturing the motion of a moving interface. The essential idea of this method is to (1) define the interface as one particular level set of a scalar function $\varphi(x,t)$, e.g. define the front as the set of x, such that $\varphi(x,t) \equiv 0$ and (2) evolve this "level set function" in time instead of tracking the interface. This procedure allows for the "capturing" of topological changes easily. The extra dimension's worth of calculation is removed by a simple "local level set method" [PMOZK]. This procedure turns a geometric problem into a problem in the numerical solution of a partial differential equation (PDE). Numerically, we represent an arbitrary surface of codimension one by the values, on the grid of a function which changes sign across the interface. Subgrid locations can be obtained accurately by interpolation,

[1] We thank Ronald Fedkiw for pointing this out

used only as a postprocessing of the data. The interpolation does not feed back to the calculation. This method is quite simple, even for the most complicated configurations. All local geometric information concerning the interface can be extracted accurately using divided differences on a fixed, Cartesian grid. Powerful tools developed for PDE's both theoretical and numerical can be applied here, e.g. [ES, ESS, OSe, OSh].

We briefly review the level set method. Consider a closed interface moving with time, $\Gamma(t)$, in R^n of codimension one. Let $\Omega(t)$ be the (generally multiply connected) region enclosed by $\Gamma(t)$. Let $\varphi(x,t)$, be the level set function associated with $\Omega(t)$. The function is Lipschitz continuous and satisfies the following

$$\varphi(x,t) < 0 \text{ in } \Omega(t) \tag{1a}$$

$$\varphi(x,t) = 0 \text{ on } \Gamma(t) \tag{1b}$$

$$\varphi(x,t) > 0 \text{ in } \bar{\Omega}^c(t) \tag{1c}$$

We have thus identified $\Gamma(t)$ as the zero level set of $\varphi(x,t)$. Differentiating the equation (which holds on $\Gamma(t)$):

$$\varphi(x(t),t) \equiv 0, \tag{2}$$

with respect to t, leads to a transport equation for φ which holds on $\Gamma(t)$

$$\varphi_t + \vec{v} \cdot \nabla\varphi = 0 \tag{3}$$

where $\vec{v}$ is the velocity of a particle trajectory on the interface $\Gamma(t)$ moving with velocity $\vec{v} = x(t)$. Let $v_n = \vec{v} \cdot \vec{n} = \vec{v} \cdot \frac{\nabla\varphi}{|\nabla\varphi|}$ be the projection of $\vec{v}$ onto the outwards unit normal direction $\vec{n}$ at $\Gamma(t)$. Here $|x| = (x_1^2 + \cdots x_d^2)^{\frac{1}{2}}$ for $d = 2$ or 3. We arrive at

$$\varphi_t + v_n|\nabla\varphi| = 0. \tag{4}$$

This equation is valid on $\Gamma(t)$. We extend v_n off of the interface in a "natural" way (see discussion below) and assume the equation is valid throughout space, or (in the local level set method) that it is valid in a small neighborhood of the interface. The quantity $v_n(x,t)$

usually depends on (1) the geometric properties of the interface such as its curvature, normal direction, length (area in 3D), see e.g. [MCO, ZCMO] and/or (2) quantities which involve external coupled physics of multiphase problems, obtained by solving a system of partial differential equations. These quantities might include pressure, temperature, etc [CHMO, CMOS, SSO].

There are three improvements over the original level set formulation which are important aids to efficient computation. The first is necessary because of the excessive freedom in the choice of the level set function φ. Left alone, this function might develop very flat or very steep gradients near its zero level set. These are both problematic for calculations, e.g. [CHMO, SSO]. A good choice would be to take φ to be signed distance to the boundary. This means φ satisfies (1) and, additionally, $|\nabla\varphi| \equiv 1$. This gives us a well behaved level set function, which is quite useful for extracting information about the interface. For example, for an arbitrary point x in R^n and φ signed distance, the "projection operator" which takes x to the closest point on the interface, x_p, is simply evaluated by

$$x_p = x - \varphi\nabla\varphi. \tag{5}$$

However, the evolution in (2) will not preserve φ as a distance function, except in the special case $\nabla v_n \cdot \nabla\varphi = 0$ (where v_n is regarded as being defined throughout the region of interest, see below).

A very simple and efficient redistancing algorithm was provided in [SSO]. One merely solves after each time step

$$\varphi_\tau + (\text{sign}(\varphi_0))(|\nabla\varphi| - 1) = 0 \tag{6}$$

$\tau > 0$, $\varphi_0 = \varphi(x,t)$, with φ_0 as initial data for (6).

See [SSO] for more details. In practice a few iterations in τ will suffice.

The second ambiguity comes in the definition of v_n off of the interface. There is often a natural way to extend v_n. For example, for mean curvature motion, such as in (1), one may merely extend v_n as the curvature of each level set of φ, i.e. $v_n = \nabla\left(\frac{\nabla\varphi}{|\nabla\varphi|}\right) = \kappa$.

However, out of necessity, as in [CMOS], and due to its effectiveness and simplicity, we may simply extend v_n so that it is constant normal to each level curve of φ. This means we solve $\nabla v_n \cdot \nabla \varphi = 0$. This was first suggested and analyzed in [ZCMO] and first used effectively in [CMOS]. The numerical method is merely to solve to steady state (again, using only a few iterations):

$$(v_n)_\tau + (\text{sign } \varphi_0)(\nabla \varphi_0 \cdot \nabla v_n) = 0 \tag{7}$$

with $v_n(x,t)$ initial data for the above.

This procedure guarantees that the distance between level contours of φ will not be changed under the evolution (2).

These two properties were shown to be closely related. In fact, redistancing is equivalent to extending v_n so that $\nabla v_n \cdot \nabla \varphi = 0$.

The third simplification involves making the motion defined by (4) local, i.e. we need solve this only for $|\varphi| < \epsilon \approx 6\Delta x$. This gives us a significant speedup. See [PMOZK] for a detailed explanation of this simple localization (which we have used in a number of papers, e.g. [HLOZ, ZMOW, KMOS]).

The advantages of this method can be summarized as follows:

(1) All numerical calculations may be carried out on a fixed rectangular grid. This provides us with a very simple data structure.

(2) Topological changes, complicated geometries, corners, cusps and other singularities can be handled simply.

(3) Geometric quantities such as mean curvature, normal direction, Gaussian curvature, etc can all be computed simply, using divided differences in the rectangular grid.

(4) The formulation is the same in any number of space dimensions.

The success of the level set method depends on:

(1) High order essentially nonoscillatory numerical methods developed in [OSe, OSh]. These were modified from their (more complicated) analogues, use in the numeral approx-

imations of conservation laws, e.g. [ShO] and are used here for the nonlinear first order Hamilton-Jacobi equation one gets when $\vec{v} = v(x, t, \nabla\varphi)$ in (3), or, more generally for any terms of that sort which arise in (3). These schemes effortlessly capture corners and cusps in the solution.

(2) Reinitialization of the level set function (or functions in the multiphase case) to be signed distance near the interface as developed in [SSO]. (3) The use of a fast localized level set method, developed in [PMOZK].

2.2 The Variational Level Set Method

We shall adapt the general framework we developed in [ZCMO]. There we constructed a physically based energy function for some multiphase system. The system moves to its equilibrium state by decreasing the total energy. The main novelty in this method is that the energy functional is expressed in terms of level set functions, as are the constraints. We use the gradient projection method to get an evolution to steady state for the level set functions.

Our goal in this work is to obtain an "appropriate" level set function (which will always be reinitialized to be signed distance) such that all the data points lie on the zero level set. We use the physical model of an elastic membrane which can dynamically bifurcate and "shrink wrap" down to the equilibrium shape which is our desired interpolant. The energy functional includes electrostatic and elastic potential energy and is expressed in terms of the level set function.

Let $S = \{x_\alpha\} \subset \Omega \cdot$ in R^2 or R^3, be the set of points we wish to interpolate. This could be an infinite set, e.g. it could include a subset of a curve or surface, or it might be finite. We begin by computing the potential function $d(x)$ as the (nonnegative) distance from x to S

$$d(x) = \inf_{\alpha \,\varepsilon\, S} |x - x_\alpha|.$$

Using techniques discussed in e.g., [BOS] and more recently in [Ma] this initial computation can be done quickly. Moreover, it need only be computed approximately away from the set S, so that $0 < c < |\nabla d| < C$ outside of a neighborhood of S. Any distance function has kinks at points equidistant from two separate points on S and absolute value types of singularities at S. These cause no problems in our numerical calculation.

Now we define a level set φ as in (1a,b,c). The zero level set of φ is defined to be Γ and this is the curve in R^2 or surface in R^3 which will eventually be our desired interpolant. We define the potential energy of Γ in terms of φ as:

$$E_\Gamma^{(p)} = \left(\int_\Omega d^p(x)\delta(\varphi)|\nabla\varphi|dx\right)^{\frac{1}{p}} = \left\{\int_\Gamma d^p(x)ds\right\}^{\frac{1}{p}} \tag{8}$$

for $p > 1$.

Here $\delta(x)$ is the Dirac delta function and $\delta(\varphi)|\nabla\varphi| = ds$ is the surface element of the zero level set of $\varphi(x)$. See [CHMO] for a justification of this. $E_\Gamma^{(p)}$ is the L^p norm of the potential function $d(x)$ on the elastic membrane Γ. It tells us roughly the distance from Γ to S. (For $p = \infty$ it is just $\max_{x\,\varepsilon\,\Gamma} d(x, S)$).

Using standard calculus of variations, as in [ZCMO], we get the Euler-Lagrange equation for this variational problem

$$\frac{\delta(\varphi)}{p}\left[\int d(x)^p \cdot \delta(\varphi)|\nabla\varphi|\right]^{\frac{1}{p}-1} \nabla\cdot\left(d^p(x)\frac{\nabla\varphi}{|\nabla\varphi|}\right) = 0$$

with Neumann boundary conditions $\left.\frac{\partial\varphi}{\partial n}\right|_{\partial\Omega} \equiv 0$. We cancel out some quantities and obtain

$$\delta(\varphi)\left[\nabla d(x)\cdot\left(\frac{\nabla\varphi}{|\nabla\varphi|}\right) + \frac{1}{p}d(x)\nabla\cdot\frac{\nabla\varphi}{|\nabla\varphi|}\right] = 0. \tag{9}$$

We note that $\nabla\cdot\frac{\nabla\varphi}{|\nabla\varphi|}$ is the curvature of the level set of $\varphi(x)$ for fixed x.

We start with an initial membrane defined by the zero level set of $\phi(x, 0)$ which encloses S. As is by now standard, [ZCMO, ZMOW], we follow the gradient descent direction using artificial time, and replace $\delta(\varphi)$ by $|\nabla\varphi|$.

This leads us to

$$\frac{\partial\varphi}{\partial t} = |\nabla\varphi|\left[\frac{d(x)}{\bar{d}^{(p)}}\right]^{p-1}\left[\nabla d(x)\cdot\frac{\nabla\varphi}{|\nabla\varphi|} + \frac{1}{p}d(x)\nabla\cdot\left(\frac{\nabla\varphi}{|\nabla\varphi|}\right)\right]. \tag{10}$$

We note that this evolution is now morphological, [AGLM], i.e. independent of which level set function, φ we choose. This means that if we replace φ by $h(\varphi)$ with $h(0) = 0$ and $h'(\varphi) > 0$, the equation is invariant.

We have redefined

$$\bar{d}^{(p)} = \left(\int_\Omega d^p(x)\delta(\varphi)|\nabla\varphi| dx \right)^{\frac{1}{p}} = E_\Gamma^{(p)}.$$

Clearly, as $p \to \infty$, $\bar{d}^{(p)} \to d_{\max} = \max_{x \,\varepsilon\, \Gamma} d(x, S)$.

The following simple result shows us that the flow defined by (10) diminishes $E_\Gamma^{(p)}$.

Lemma 1. *If $\varphi(x,t)$ satisfies (1) then $\frac{dE_\Gamma}{dt}^{(p)}(\varphi) \le 0$, with equality holding only at a local extrema.*

Proof.

$$\begin{aligned} \frac{d}{dt}E_\Gamma^{(p)}(\varphi) &= -\frac{1}{p}(\bar{d}^{(p)})^{1-p} \int_\Omega \delta(\varphi)\nabla \cdot \left[d^p(x)\frac{\nabla\varphi}{|\nabla\varphi|} \right] \varphi_t dx \\ &= -\frac{1}{p^2}(\bar{d}^{(p)})^{2(1-p)} \int_\Omega \delta(\varphi)|\nabla\varphi| \left[\nabla \cdot d^p(x)\frac{\nabla\varphi}{|\nabla\varphi|} \right]^2 dx. \end{aligned}$$

The scaling factor $\left[\frac{d(x)}{\bar{d}^{(p)}}\right]^{p-1}$ tells us that points on the membrane further away from the target set S move more quickly. As this happens, new points begin to move more quickly. This gives us a reliable, robust and efficient procedure, especially needed when the initial guess is far from the region S.

There is one potential difficulty in our approach. The quantity $\left(\frac{d(x)}{\bar{d}^{(p)}}\right)^{p-1}$ can be huge for points x belonging to level sets of φ whose value is far from zero. This difficulty is removed because our *local* level set method uses only points within a few grid cells of the zero level set and our grid resolution is always taken to be comparable to the density of data points.

The discrete version of equation (10) is the algorithm we shall use initially. It is interesting to analyze the effect of each term. At steady state the interpolating surface satisfies

$$0 = \nabla d(x) \cdot \nabla\varphi + \frac{1}{p}d(x)\nabla \cdot \frac{\nabla\varphi}{|\nabla\varphi|}. \tag{11}$$

One might be tempted to let $p \to \infty$ in the above, and, just replace $\left(\frac{d(x)}{d^p}\right)^{p-1}$ by 1 in equation (10). This leads us to the simple advection equation

$$\frac{\partial \varphi}{\partial t} - \nabla d \cdot \nabla \varphi = 0 \tag{12}$$

to be solved to steady state. However, in 3D the final state is a bit "ragged", as compared to the state we obtain using our algorithm. Moreover, in our more general setting, the simpler method may fail.

Consider, for example, S consisting of two points in R^2, $(0,0)$ and $(1,0)$. Let Γ contain the line segment from $(0,0)$ to $(\frac{1}{2}, b)$ and from $(\frac{1}{2}, b)$ to $(1,0)$ for any b. This means that $d(x,y) = \sqrt{\min[x^2, (x-1)^2] + y^2}$, while $\varphi(x,y) = y - b\min(x, 1-x)$, (which could be reinitialized to be signed distance to Γ). In practice (12) with small curvature regularization can be used as a an efficient way to find a rough approximation.

Then $\nabla d \cdot \nabla \varphi = 0$ on Γ and the curve will not move. If, on the other hand, we add $\frac{1}{p} d \nabla \left(\frac{\nabla \varphi}{\nabla \varphi}\right)$, the curve will move to its proper equilibrium position, which is the line connecting $(0,0)$ to $(1,0)$.

We may regard the motion caused by the convection, $\nabla d \cdot \frac{\nabla \varphi}{|\nabla \varphi|}$, as reducing the electrostatic energy, while the mean curvature motion induced by $\frac{1}{p} d(x) \nabla \cdot \left(\frac{\nabla \varphi}{|\nabla \varphi|}\right)$ reduces the elastic energy, which is proportional to the surface area. When the membrane Γ is far from S, both of these two factors cause γ to move towards S. However near each point in S, $d(x) \approx 0$, while $|\nabla d| = 1$, so the electrostatic potential dominates. Away from the data points the two factors balance as steady state equation (10) is valid. Since $d(x)$ is related to the density of data points and $\nabla \cdot \frac{\nabla \varphi}{|\nabla \varphi|}$ is the curvature of Γ, we have a reasonable resolution condition.

When the membrane actually passes through a data point, x_i, then $\varphi(x_i) = 0 = d(x_i)$ and the vector $\nabla d(x_i) = 0$. Thus $\frac{\partial \varphi}{\partial t}(x_i) = 0$ and there is no force pushing the membrane away from x_i. We shall prove below, that if $p \geq 1$, the membrane will not collapse through any data point. Our numerical calculations bear this out.

We shall prove the following results in the Appendix.

Lemma 2. *Let $\Gamma(t)$ be the level set of φ, moving according to equation (10). If $x_i \varepsilon S$ and $\Gamma(t)$ contains x_i at $t = \bar{t}$ it will contain x_i for $t > \bar{t}$. Hence, if the initial curve contains S, it will not collapse and disappear.*

Lemma 3. *If Γ passes through every point in S and is a local minimum for $E_\Gamma^{(p)}$ (which means equation (11) is satisfied, and, in R^2, if $x_{i'}$ is connected to x_i on Γ with no other data points between $x_{i'}$ and X_i on Γ, then that connection is a straight line.*

Theorem 1. *In R^2, if $\Gamma(0)$ is close enough to S, then $\Gamma(t)$ will converge to a polygon obtained by the above construction, which gives us a piecewise linear interpolation.*

The argument cannot be extended to R^3 to show that the polyhedron with piecewise triangular facets connecting each x_i to its two closest points is in local equilibrium. In 3D we get some kind of weighted minimal surface interpolant. The reason we start with a large membrane (10) enclosing S is because we do not want $\Gamma(t)$ to shrink down and disappear, which is the global minimum of $E^{(p)}$ for $p < \infty$. We also have, when the set S is a smooth curve in R^2 or surface in R^3:

Lemma 4. *Let S be a closed smooth surface. Then there is no other contour within d_0 of S which acts as a minimizer of $E_{d_0}^{(p)}$ where,*

$$d_0 = \frac{1}{2|K|_\infty}$$

where $|K|_\infty$ is the maximum of the absolute value of the curvature of S in R^2, or is the maximum of the absolute value of each principal curvature of S in R^3.

This gives us our principle conjecture, which we have validated by our numerical experiments.

Conjecture. Let S be a discrete data set lying on $\bar{S}$ which is a smooth interpolant of S, with d_0 defined for $\bar{S}$. If any two closest points in the data set have distance less than $4d_0$ and if our initial guess is within $2d_0$ of $\bar{S}$, then $\Gamma(t)$ will converge to a surface interpolating S.

If the true surface $\bar{S}$ is convex, then minimizing $E_\Gamma^{(p)}$ will very efficiently find the piecewise linear interpolant described in Theorem 1 for R^2, and the more interesting interpolant in R^3. If there are many fine features, i.e. $|K|_\infty$ is large, then we probably need a good initial guess. Without this, our membrane may get "stuck". Another important property of our energy is that the surface tension $d(x)$ automatically adapts to the density of data points.

There is, however, a problem with our energy $E_\Gamma^{(p)}$ which allows Γ to occasionally get "stuck". This energy somehow only measures the distance from $\Gamma(t)$, from the membrane to the closest data point. The philosophy of interpolation is generally different. We usually minimize the distance from each x_i in S to the curve (or surface) $\Gamma(t)$. An example of the difficulties comes from a situation as shown in Figure 1. There, a narrow bottleneck screens out the effects of the enclosed charges. The membrane feels no attraction from those interior points. In fact, $E_\Gamma^{(p)}$ will increase if the membrane $\Gamma(t)$ enters the bottleneck since the electrostatic energy is unchanged, while the elastic energy increases.

At the opening of the square, $d(x)$, is determined by the two points at the end of the opening. When the membrane closes in and forms a straight line there, $\nabla d(x) \cdot \nabla \varphi = 0$ and $\kappa = 0$, so we have a equilibrium. Figure 1 shows the result for different values of p. The bigger the p, the more flexible the membrane, but the "screening" difficulty can not be overcome.

In our calculations we observed that $p = 2$ is a good choice, which gives rapid convergence to steady state and to a pleasant appearing shape in 3D. We conclude that for $E_\Gamma^{(p)}$ we have robust and efficient convergence to steady state. The method works well for disconnected regions as in Figure 2, and for interesting shapes as in Figure 3. In fact, it works quite well in three dimensions since the two principal curvatures are both finite – this is related to the well known result that motion by mean curvature in two dimensions shrinks arbitrary curves down to round points, but not in three dimensions [Gr, Se]. The principal deficiency comes in approximating very concave shapes which are essentially two dimensional.

In order to overcome these difficulties we construct another "energy" function $E_\Gamma^{(-q)}$, $q \geq 1$ which measures, in some norm the distance from each $x_i \, \varepsilon \, S$ to $\Gamma(t)$. We use "   " because the functional is not convex; we approximate its gradient descent below by a well-posed evaluation procedure.

Define $d_i(x) = |x - x_i|$, the distance for each x to each $x_i \varepsilon S$. The distance from x_i to $\Gamma(t)$ is $d_i = \inf_{x \, \varepsilon \, \Gamma(t)} |x - x_0|$. We approximate, for $q \geq 1$

$$(13) \qquad d_i \approx d_i^{(-q)} = \left(\int_\Omega d_i^{-q}(x) \delta(\varphi) |\nabla \varphi| \right)^{-\frac{1}{q}} \to d_i \text{ as } q \to \infty.$$

Now we define

$$(14) \qquad E_\Gamma^{(-q)} = \sum_{i=1}^{n} \left[\int_\Omega d_i^{-q}(x) \delta(\varphi) |\nabla \varphi| \right]^{-\frac{1}{q}}$$

(here n is the number of points on S which, for simplicity of exposition, we have taken to be finite).

The rescaled gradient descent flow (i.e., $\delta(\varphi)$ replaced by $|\nabla \varphi|$) for $E_\Gamma^{(-q)}$ is

$$(15) \qquad \frac{\partial \varphi}{\partial t} = |\nabla \varphi| \sum_{i=1}^{n} \left(\frac{d_i^{(-q)}}{d_i(x)} \right)^{q+1} \left[\nabla d_i(x) \cdot \frac{\nabla \varphi}{|\nabla \varphi|} - \frac{1}{q} d_i(x) \; \nabla \cdot \frac{\nabla \varphi}{|\nabla \varphi|} \right]$$

we replace $d_i^{(-q)}$ by its limit as $q \to \infty$, as in equation (13) arriving at

$$(16) \qquad \frac{\partial \varphi}{\partial t} = |\nabla \varphi| \sum_{i=1}^{n} \left(\frac{d_i}{d_i(x)} \right)^{q+1} \left[\nabla d_i \cdot \frac{\nabla \varphi}{|\nabla \varphi|} - \frac{1}{q} d_i(x) \nabla \cdot \frac{\nabla \varphi}{|\nabla \varphi|} \right].$$

The negative coefficient in front of the curvature term above indicates the ill-posedness of this procedure, corresponding to the nonconvexity of $E_\Gamma^{(-q)}$. The factor $\left(\frac{d_i}{d_i(x)} \right)^{q+1}$ indicates that the point x_i attracts the closest point x on the membrane $\Gamma(t)$ most, and once x_i is on the membrane, it has no effect. The membrane will move locally to pass through each x_i. This is the correct interpolation philosophy, if $\Gamma(t)$ is close to S.

However, when this closest point moves quickly, it is even closer and continues to move. Thus the membrane bends (which explains the negative coefficient of curvature). This helps to explain the ill-posedness. Another serious difficulty comes in the very expensive computation of $\int d_i^{-q}(x) \delta(\varphi) |\nabla \varphi|$, for each x_i and each time step. We simplify as follows.

First we let $q \to \infty$ in the coefficient of curvature, so this coefficient disappears, and second we use the fact that $\phi(x,t)$ has been rescaled to be signed distance to Γ, to obtain the advection equation

$$\frac{\partial\varphi}{\partial t} = \sum_{i=1}^{n} \left[\frac{|\phi(x_i)|}{d_i(x)}\right]^{q+1} \nabla d_i(x) \cdot \nabla\varphi. \tag{17}$$

We call this the flow associated with $E_\Gamma^{(-q)}$.

In this new flow each x feels the attraction of all $x_j \varepsilon S$, but each x_i affects the closest point on Γ the most. The computational cost has diminished because there are no integrals to evaluate. Moreover, the PDE has no second order terms so the restriction in Δt is eased considerably. Finally, we can easily parallelize the computations of $d_i(x) = |x - x_i|$ and $\nabla d_i = \frac{x-x_i}{|x-x_i|}$.

It is interesting to note that pure convection of the form[2].

$$\frac{\partial\varphi}{\partial t} - \nabla d_i(x) \cdot \nabla\varphi = 0 \tag{18}$$

has the effect of attracting points on the zero level set of φ towards x_i. This is easily seen, e.g. in 2 dimensions from the method of characteristics in polar coordinates. If $\varphi(r,\theta,0) = \varphi_0(r,\theta)$ is given, with the origin of the coordinate system shifted to the point x_i, then $\varphi(r,\theta,t) = \varphi_0(r+t,\theta)$, so all level sets compress to the origin.

From this point of view, one might neglect the curvature term in (10) and simplify (10) and (17) further to

$$\frac{\partial\varphi}{\partial t} - c_{(p)}\nabla d \cdot \nabla\varphi - c_{(-q)}\tilde{\Sigma}\nabla d_i \cdot \nabla\varphi = 0 \tag{19}$$

where $c_{(p)}$ and $c_{(-q)}$ are nonnegative, adding up to one, and the sum $\tilde{\Sigma}$ is taken over i for which $\varphi(x_i) \neq 0$. This will attract Γ towards S. We have, however, found generally better results with our slightly more complicated algorithm, especially in 3D.

[2] We again thank Ronald Fedkiw for this helpful observation and several relevant discussions.

There are a few interesting variations to this flow. One obvious modification is to weight each x_i according to some measure of uncertainty. Another, quite intriguing modification is to change our initialization strategy. Namely, we start with a small membrane inside the data points and let it grow with the flow defined in (17). Other variations are possible and easy to implement because of the flexibility of our level set formulation. We could use several membranes, some small, some large, etc. We can also add elastic energy as a regularizer, i.e. add $\epsilon(x)\nabla \cdot \frac{\nabla\varphi}{|\nabla\varphi|}$, with $\epsilon(x) > 0$ chosen as needed to remove noise.

If we start with a poor initial guess and use the flow defined by (16), the point of Γ closest to S will be most attracted, while the distant points will be affected the least. The properties of $E_\Gamma^{(p)}$ exactly complement this. We recommend taking some combination of these two energies. We have used the following procedure in this work. First we minimize $E_\Gamma^{(p)}$ which leads us to the flow defined by (10). We flow to steady state, which occurs when the norm of the difference in φ in succeeding time step is small enough. If at steady state $\sum |\varphi(x_i)|$ is small, i.e. every point in S is close to Γ, we are finished. Otherwise, we have a good initial guess which we use to minimize $E_\Gamma^{(-q)}$ via the flow defined by (16).

A fascinating numerical phenomenon we often observe in these experiments is that when we start using (16), points x_i in S which were initially on Γ, i.e., initially $\varphi(x_i) = 0$ will *leave* Γ, in order to better accomodate points in S which were not initially on Γ. Recall, if a point $x \, \varepsilon \, S$ is on Γ, it no longer attracts Γ. At some later stage the membrane will be attracted back to all the points. Finally, after $\varphi(x_i) = 0$ for all $x_i \, \varepsilon \, S$, we can run (10) again to smooth the final surface.

If more data points are added after convergence, or if we need to follow some dynamically changing surface or surfaces, the models handle the situation easily. We have a good initial guess in both cases.

A very interesting and simple multiscale idea can be used together with our approach. The resolution of the surface is determined by both the density of points in S (reflected by the values of $d(x)$ and the underlying grid. For a fixed grid we can cut our computational cost in the following way. If there are many points in S lying within a given grid cell, we rescale the data by using the mass center of the given points with a weight equal to their number. This lowers the cost without losing resolution.

Additionally, we can build a hierarchical set of grids and find the shape successively from the coarsest grid. We can easily do adaptive refinement so that each cell only contains one data points.

We shall report on these multiscale procedures in future work.

3. Numerical Implementation and Results

We use techniques discussed in [ZCMO] in order to solve (10) and (17). We may rewrite (10) as

$$\frac{\partial \varphi}{\partial t} = |\nabla \varphi| \frac{1}{p} (\bar{d}^{(p)})^{1-p} \nabla \cdot \left[d^p(x) \frac{\nabla \varphi}{|\nabla \varphi|} \right] \tag{20}$$

and use a conservative formulation to compute $\nabla \cdot \left(d^p(x) \frac{\nabla \varphi}{|\nabla \varphi|} \right)$. This potential function $d(x)$ must be calculated initially, but only once. There are fast algorithms to do this [Ma, BOS]. For the advection equation (17) we use an upwind ENO or TVD scheme as discussed in [OSe, OSh]. We also normalize the velocity field in (17), i.e. we replace $\vec{v} \cdot \nabla \varphi$ by $\frac{\vec{v}}{1+|\vec{v}|} \cdot \nabla \varphi$. We also avoid division by zero by replacing $d_i(x)$ by $\max(|x - x_i|, \delta)$, for $0 < \delta << \Delta x$. We reduce the number of terms in the summation in (17) by neglecting those points which are sufficiently close to Γ, i.e. replace $\varphi(x_i)$ by 0 if $|\varphi(x_i)| <$ tolerance.

We use distance reinitialization and the local level set method, which is described in [PMOZK], in both cases.

Although we could adapt the time step more carefully, we use $\Delta t \simeq (\Delta x)^2$ in the $E_\Gamma^{(p)}$ formulation and $\Delta t \simeq \Delta x$ in the $E_p^{(-q)}$ formulation, since the first leads to a 2^{nd} order parabolic PDE and the second a first order hyperbolic PDE. We find that the $(\Delta x)^2$ restriction is not serious since the motion by mean curvature term speeds the convergence to steady state. We used the slightly smoothed out approximation to $\delta(\varphi)$ described in [SSO] in the computation of $\bar{d}^{(p)}$.

One other simplification, which we used in the $E_\Gamma^{(p)}$, formulation is to replace (18) by motion by mean curvature for x on Γ, when $d(x) > d_0$, for some preselected positive d_0, then reverting to (18) when $d(x) \leq d_0$. We thus avoid computing $\bar{d}^{(p)}$ until it becomes necessary.

In our figures, the dots correspond to the given data set S. The points are neither evenly spaced, nor are they on the grid. The continuous lines or surfaces are the zero level sets of φ, i.e. they represent $\Gamma(t)$. The quantities p and q are the numbers used in $E_\Gamma^{(p)}$ and $E_\Gamma^{(-q)}$ respectively.

We have seen the same interpolating surface whether we start with a sphere or a rectangular box enclosing S for the $E_\Gamma^{(p)}$ minimum in figures 2 and 3. In figure 4 we start with a rectangle and use $p = 2$ to converge to the simply connected figure which fails to interpolate some of the points. We then use $q = 2$ to converge to the two circles. In figure 5 we converge at $t = 0.36$ to an object that misses many of the inner points of S. Then we flow with $q = 2$ to steady state at $t = .46$. Finally we smooth the result using $p = 2$. It converges at $t = .55$ to a pleasing interpolant. Figure 6 is treated similarly. In the upper left corner we display contours of $d(x)$. We use $p = 2$, converge to the approximate convex hull at $t = .225$. Then we use $q = 2$ and arrive at our interpolant at $t = .575$. Finally we use $p = 2$ and smooth with success. It is amusing to see the effect of the $(-q)$ flow – the curve Γ leaves S completely at $t = .325$, then returns.

Our 3 dimensional results in figures 7,8 and 9 use only $E_\Gamma^{(p)}$ for $p = 2$. Remarkably, we handle all these topological changes with ease, and with sparse data points. Figure 7 was initialized with a rectangular box. The number of data points was 300. For the interlinked tori in figure 8, we initialized with a sphere (not shown) and there are 500 data points. In figure 9, the initial sphere converges to the four separate balls. In this example we continue past convergence (which occurred at $\approx$ 1500 iterations) and the result is unchanged at 4,000 iterations.

In Figure 10, a section of a pipe with 1200 data points is interpolated with a box as initial guess. First we use $p = 2$ and converge after 500 iterations to an approximate convex hull. Since $\varphi(x_i) \neq 0$ for many points, we then proceed 100 iterations with $q = 2$

to convergence. Finally we smooth with 100 iterations for $p = 2$.

Figure 11, a box with a hole in the shape of a hemisphere is to be interpolated. We have 700 data points and initialize with a box. Using $p = 2$, after 100 iterations we obtain the approximate convex hull. We then finish interpolating with 100 iterations and $q = 2$. Finally we smooth with 50 iterations and $p = 2$.

Figure 12 has the same set of data points as figure 11. We begin with a small sphere lying within S. We then use only the q formulation, with $q = 2$, and converge in 100 iterations to remarkably good interpolant.

Figure 13 has 600 data points on a torus. We again begin with a small sphere, this time lying completely within the hole. We then converge to steady state with the q formulation for $q = 2$. This time we do not quite succeed in removing the hole in the torus. Then we apply 100 iterations of the p flow, for $p = 2$, with success.

Figure 14 shows that interpolation of a graph can be treated as a special case in our formulation, i.e. by deforming a plane. Here we have 50×50 points (evenly spaced in the (x, y) cordinates). In this calculation the advection equation (12) with a small curvature regularization is used.

Figure 15 is an interpolation of two cones balanced on top of each other. In this example, the initial data set contains a point (the common vertex), curves (ten circles on each cone), and two discs (bottoms of the cones).

Figure 16 is an interpolation of a sphere where ten parallel circles on the sphere were prescribed.

Figure 17 shows the (trivial to implement) Boolean operations of a ball and a box.

4. Conclusion

A deformable surface model is used to construct a surface (curve in 2D) which interpolates a given (rather sparse) set of data points, curves and surface pieces. Our method is based on an implicit representation using a fixed Cartesian grid. The level set method is used, motivated by variational formulations.

Our method does not need any information about the ordering of the data set and gives a nonparametric representation in fixed grid, as mentioned above. It uses an extremely simple data structure and handles complicated topologies easily. The data set may be of arbitrary codimension.

We use the fast local level set method of [PMOZK] to find the final shape, which is implicitly represented as the zero value of the signed distance to the shape. The smoothness of the final shape is controlled by the elastic energy, which tends to lower the surface area. This regularization can also be used to remove noise. Uncertainty in the data can be reflected by assigning different values at different points in the energy functional. Other information and constraints such as given values of the unit normal, fixing the volume of a component, even the location of a triple junction etc. can be easily added.

After we converge to the level set representation of the shape, we can add new data point easily. Moreover, dynamic motion of the points, as in the motion of a slightly deformable solid object in a gas, can also be easily handled. We have a good initial guess at each time step, namely the interpolation surface Γ through S at the previous time step.

We can easily compute approximations to all geometric quantities such as curvature, normal, surface area, enclosed volume, numbers of connected components, location of closest point on Γ to a given point x, etc.

The formulation can easily be extended to higher dimensional data analysis. A more efficient algorithm can be implemented via a multigrid or multiresolution strategy.

Finally, a simple, local postprocessing, as described in the introduction, can be used to smooth the final result.

Bibliography

[AGLM] L. Alvarez, F. Guichard, P.-L. Lions and J.-M. Morel, *Axioms and fundamental equations of image processing*, Arch. Rat. Mechanics, **123** (1993), pp. 199-257.

[Bo] J. Boissonnat, *Geometric structures for three-dimensional shape representation*, ACM Transactions on Graphics, (1984), pp. 266-286.

[BOS] B. Boots, A. Okabe and K. Sugihara, *Spatial tesselations: concepts and applications of Voranoi diagrams*, John Wiley & Sons, West Sussex (1992).

[Br] E. Brisson, *Representing geometric structures in d dimensions: topology and order*, Discrete and Computational Geometry, (1993), pp. 387-426.

[BW] M. Bloor and M. Wilson, *using partial differential equations to generate free-form surface*, Comput. Aided Des., **22** (1990b), pp. 202-212.

[CG] G. Celniker and D. Gossard, *Deformable curve and surface finite-element for free-form shape design*, Computer Graphics, **25** (1991), pp. 257-266.

[CHMO] Y. Chang, T. Hou, B. Merriman and S. Osher, *A level set formulation of Eulerian interface capturing method for incompressible fluid flows*, J. Comp. Phys., **124** (1996), p. 449.

[CMOS] S. Chen, B. Merriman, S. Osher and P. Smereka, *A simple level set method for solving stefan problems*, J. Comput. Phys., **135** (1997), pp. 8-29.

[EM] Edelsbrunner and E. Mücke, *Three-dimensional alpha shapes*, ACM Transactions on Graphics, (1994), pp. 43-72.

[ES] L.C. Evans and J. Spruck, *Motion of level set by mean curvature I*, J. Differential Geometry, **33** (1991), pp. 635-681.

[ESS] L.C. Evans, H.M. Soner, and P.E. Souganidis, *Phase transitions and generalized motion by mean curvature*, Comm. Pure and Appl. Math., **65** (1992), pp. 1097-1123.

[Gr] M.A. Grayson, *The heat equation shrinks embedded plane curves to round points*, J. Diff. Geom., **26**, pp. 285-319.

[HLOZ] T. Hou, Z. Li, S. Osher and H. Zhao, *A hybrid method for moving interface problems with application to Hele-Shaw problem*, J. Comp. Phy., **134** (1997).

[HSIN] A. Hilton, A.J. Stoddart, J. Illingworth and T. Windeatt, *Implicit surface - based* *geometric fusion*, Comput. Vision and Image Understanding, **69** (1998), pp. 273-291.

[KMOS] M. Kang, B. Merriman, S. Osher and P. Smereka, *A level set approach for the motion of soap bubbles with curvature dependent velocity or acceleration*, UCLA CAM report 96-19, (1996).

[Ma] S. Mauch, *Calculating closest point and distance to a curve*, AMA Caltech preprint, (1998).

[MCO] B. Merriman, R. Caflisch and S. Osher, *Level set methods, with application to modeling the growth of thin films*, Proc. 1997 Congress on Free Boundary Problems, Herakleion, Crete, Greece, (1997), to appear.

[MT] T. McInerney and D. Terzopoulos, *A finite element model for 3D shape reconstruction and nonrigid motion tracking*, Proc. IEEE 4th Int. Conf. on Comp. Vis., (1993), pp. 518-523.

[OSe] S. Osher and J. Sethian, *Fronts propagating with curvature dependent speed, algorithms based on a Hamilton-Jacobi formulation*, J. Comp. Phys., **79** (1988), pp. 12-49.

[OSh] S. Osher and C.-W. Shu, *High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations*, SINUM, **28** (1991), pp. 907-992.

[PMOZK] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang, *A pale based fast local level set method*, UCLA CAM report 98-25, (1998, (submitted to J. Comp. Phys.).

[ROF] L. Rudin, S. Osher and E. Fatemi, *Nonlinear total variational based noise removal algorithms*, Physica D, **60** (1992), pp. 255-268.

[Se] J. A. Sethian, *A review of recent numerical algorithms for hypersurfaces moving with curvature dependent speed*, J. of Diff. Geom., **31** (1989), pp. 131-151.

[ShO] C.-W. Shu and S. Osher, *Efficient implementation of essentially nonoscillatory shock capturing schemes II*, J. Comput. Phys., **83** (1989), pp. 32-78.

[SSO] M. Sussman, P. Smereka, and S. Osher, *A level set approach for computing solutions to incompressible two-phase flows*, J. Comp. Phys., **119** (1994), pp. 146-159.

[ZCMO] H. Zhao, T. Chan, B. Merriman and S. Osher, *A variational level set approach to multiphase motion*, J. Comp. Phys., **127** (1996), pp. 179-195.

[ZMOW] H. Zhao, B. Merriman, S. Osher and L. Wang, *Capturing the behavior of bubbles and drops using the variational level set approach*, J. Comp. Phys., **143** (1998), pp. 495-518.

Appendix

In this section we shall prove Lemmas 2,3,4 and Theorem 1. We will also discuss relevant geometric issues.

In two dimensions, polygons composed of straight lines connecting data points form a local equilibrium for $E_\Gamma^{(p)}$. In the neighborhood of a point x_i, $d = |x - x_i|$ and ∇d points along the line from x_i to x. It is easy to see from figure 1 that

$$\nabla d \cdot \nabla \varphi + \frac{1}{p} d \nabla \cdot \frac{\nabla \varphi}{|\nabla \varphi|} = 0.$$

The angle θ between two line segments is determined by the neighbors of x_i in S.

Now we shall show that this polygon is actually a local minimum of $E_\Gamma^{(p)}$. We have two simple observations (i) Any curve that passes through x_i can not have less energy than two straight line segments passing through x_i (see figure 17). (ii) Any two line segments meeting at x_i have the same energy. We shall show that if the curve is deformed a bit from x_i, then the energy will increase. As in figure 18 and 19 we deform Γ to $\tilde{\Gamma}$. We denote the weighted length of Γ by I_L and II_L. (By symmetry, we need consider only the right hand part of each line). We have

$$I_L = \int_0^{\epsilon cot\frac{\theta}{2}} s^p ds = \epsilon^{p+1} \frac{(cot\theta)^{p+1}}{p+1}$$

$$II_L = \int_{\epsilon cot\frac{\theta}{2}}^{\ell} s^p ds.$$

The weighted length of the shifted line segment is

$$\begin{aligned} III_L &= \int_{\epsilon cot\frac{\theta}{2}}^{\ell} d^p ds = \int_{\epsilon cot\frac{\theta}{2}}^{\ell} (s^2 + \epsilon^2)^{\frac{p}{2}} ds \\ &= I_L + \int_{\epsilon cot\frac{\theta}{2}}^{\ell} s^p((1 + \frac{\epsilon^2}{s^2})^{\frac{p}{2}} - 1)ds \\ &\geq II_L + c(\ell, \theta)\epsilon^2 \text{ if } p > 1 \\ &\geq II_L + c(\ell, \theta)\epsilon^2 \ell_n \epsilon \text{ if } p = 1. \end{aligned}$$

Where $0 < c$ depends only on ℓ and θ.

We see that if $p \geq 1$, the slightly perturbed line segment has larger energy than the unperturbed segment. Thus, we have proven Lemmas 2, 3 and Theorem 1.

Next, we consider the idealized case when S is the complete set of curves (or surfaces in R^3) which we wish to interpolate. Then $d(x)$ is the exact distance function (unsigned) to S. Since the points on $\Gamma(t)$ which are furthest from S move most quickly, we expect that $\Gamma(t)$ will move so as to line up with the level contours of $d(x)$. It is possible, for example, if S is not simply connected, that the energy $E_{\Gamma}^{(p)}$ can have a local extrema off of S, i.e. for Γ aligned with a nonzero level contour S. One such example occurs whenever S consists of two separate closed curves S_1, S_2 or (surfaces in R^3). if Γ lies on S_1, its energy is zero. If we then deform Γ from S_1 to S_2 along level contours of d, $E_{\Gamma}^{(p)}$ first increases, then decreases to zero again. Thus it has a local maximum somewhere.

Let $\Gamma(d)$ be a level contour of d and $L(d)$ be its length (or surface area). Then

$$E_{\Gamma}^{(p)}(\Gamma(d)) = (\int_{\Gamma_d} d^p ds)^{\frac{1}{p}} = dL^{\frac{1}{p}}(d).$$

Let $F(d) = dL^{\frac{1}{p}}(d)$. Then $F(0) = 0$ is a minimum and

$$F'(d) = L^{\frac{1}{p}}(d) + \frac{d}{p} L^{\frac{1}{p}-1}(d) L'(d).$$

An interesting question is whether there is an isolated local minimum of $F(d)$ near $d = 0$. We shall carry out our argument by assuming $0 < d$ is the signed distance function to S. The argument for Γ lying within S, i.e. when the signed distance function is negative, is similar.

Suppose $S = \Gamma(0)$ is C^2 and, $|K|_\infty$ is the maximum of the norm of curvature along S. If $d|K_\infty| < 1$ and $\Gamma(d)$ is a C^2 diffeomorphism, we have

$$L(d) = \int_{\Gamma(d)} ds = \int_{\Gamma(0)} (1 + dK) ds \tag{21}$$

so

$$L'(d) = \int_{\Gamma(d)} K ds$$

and

$$\begin{aligned} F'(d) &= L^{\frac{1}{p}-1}(L + \frac{d}{p} L'(d)) \\ &= L^{\frac{1}{p}-1}\left(\int_{\Gamma(0)} (1 + d(1 + \frac{1}{p})K)\right) \\ &\geq L^{\frac{1}{p}-1}\left(\int_{\Gamma(0)} (1 - d(1 + \frac{1}{p})|K|)\right) \\ &> L^{\frac{1}{p}-1} \int (1 - 2d|K|) > 0. \end{aligned}$$

We thus have Lemma 4.

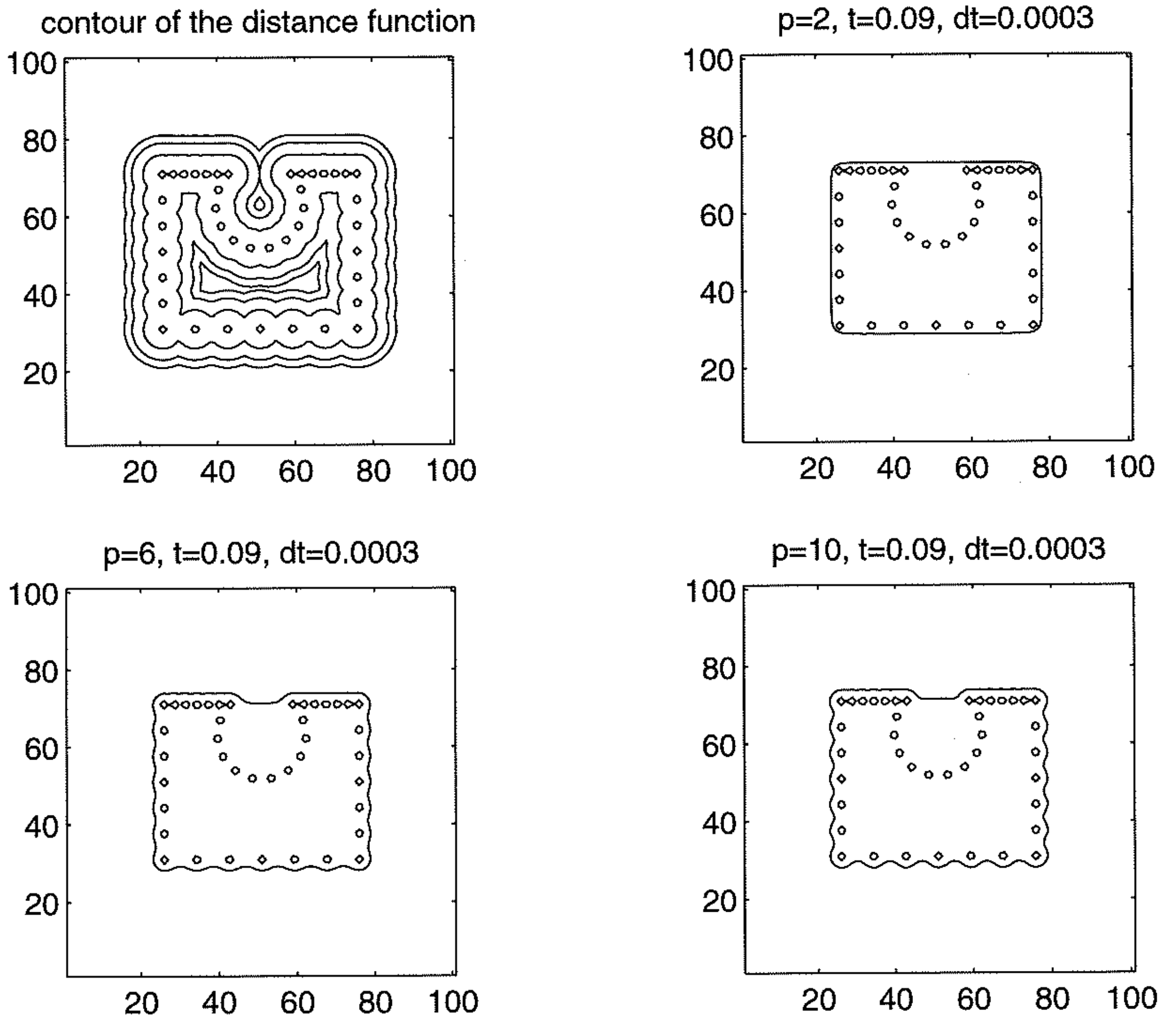

contour of the distance function
p=2, t=0.09, dt=0.0003
p=6, t=0.09, dt=0.0003
p=10, t=0.09, dt=0.0003
100
80
60
40
20
20
40
60
80
100

Figure 1

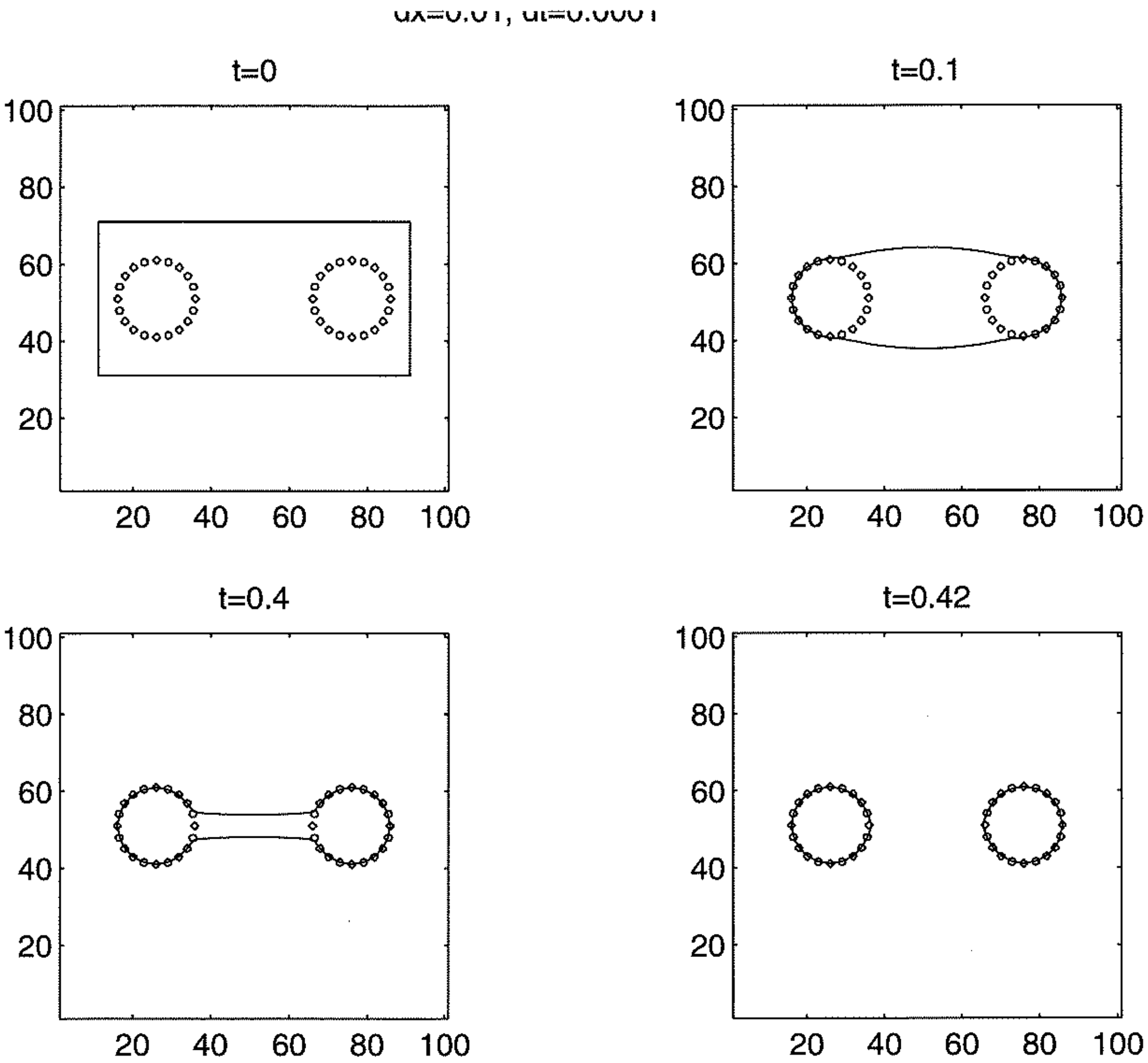


Figure 2

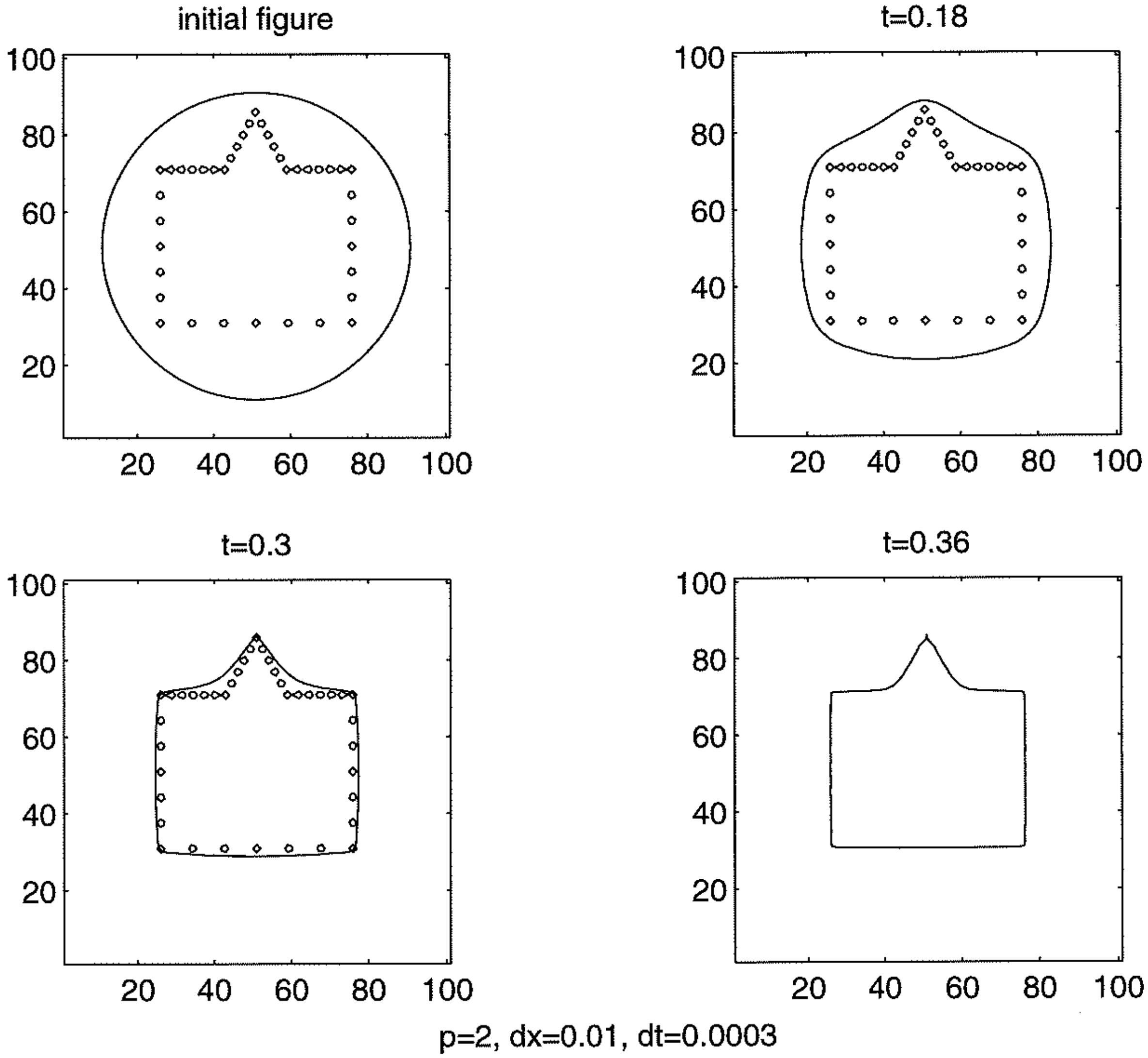

initial figure
t=0.18
t=0.3
t=0.36
100
80
60
40
20
20
40
60
80
100
p=2, dx=0.01, dt=0.0003

Figure 3

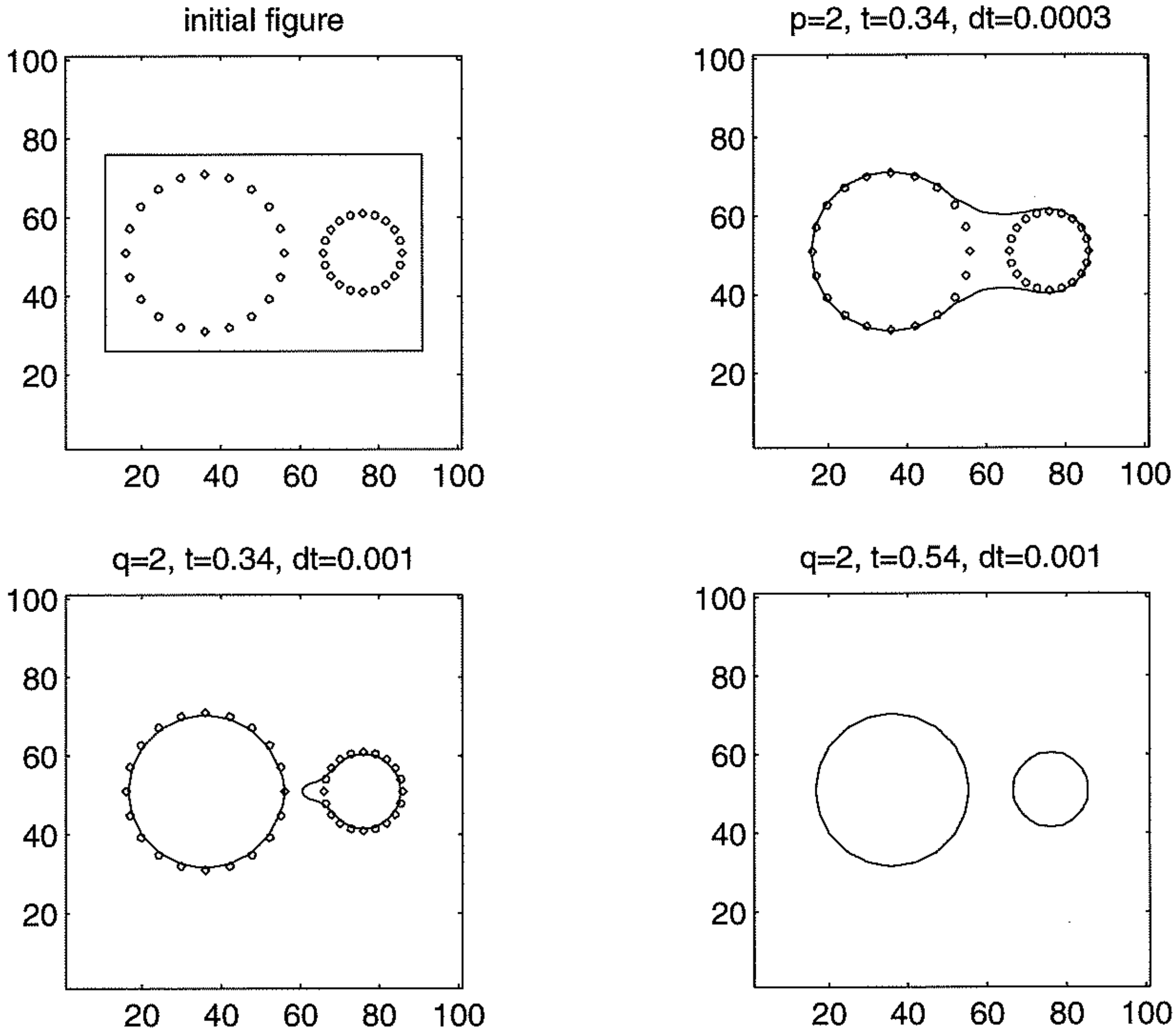

initial figure
p=2, t=0.34, dt=0.0003
q=2, t=0.34, dt=0.001
q=2, t=0.54, dt=0.001
100
80
60
40
20
20
40
60
80
100

Figure 4

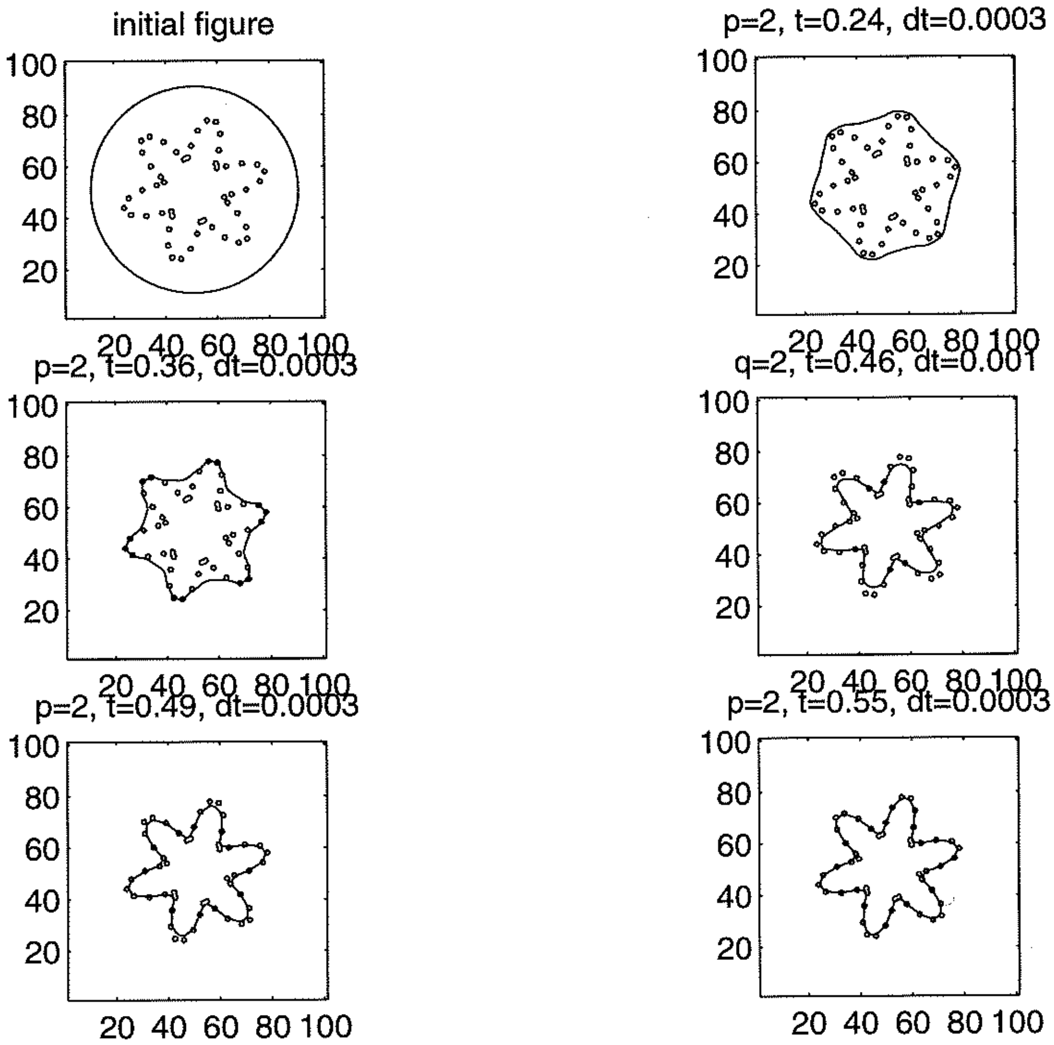

initial figure
p=2, t=0.24, dt=0.0003
p=2, t=0.36, dt=0.0003
q=2, t=0.46, dt=0.001
p=2, t=0.49, dt=0.0003
p=2, t=0.55, dt=0.0003
100
80
60
40
20
20 40 60 80 100

Figure 5

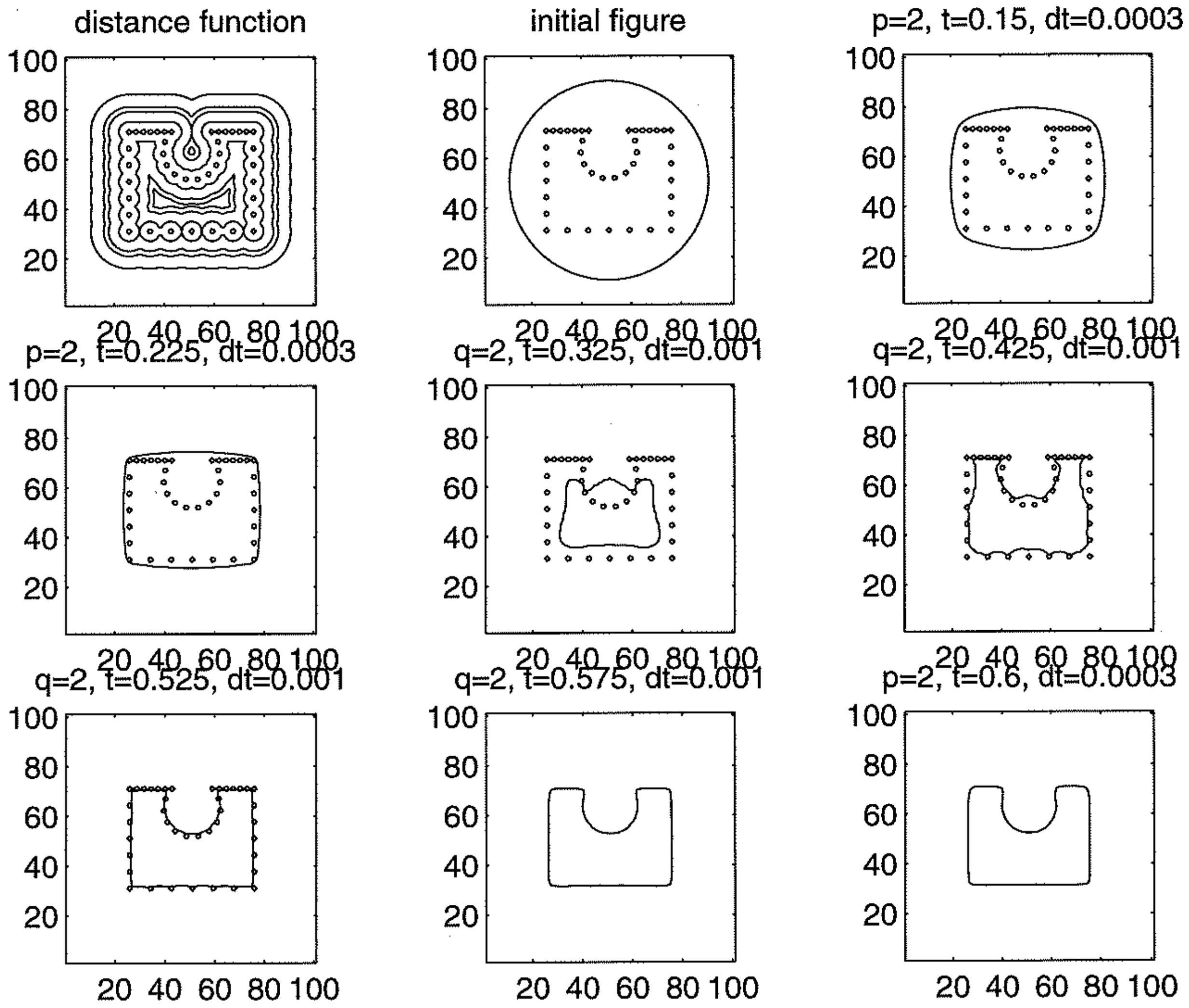
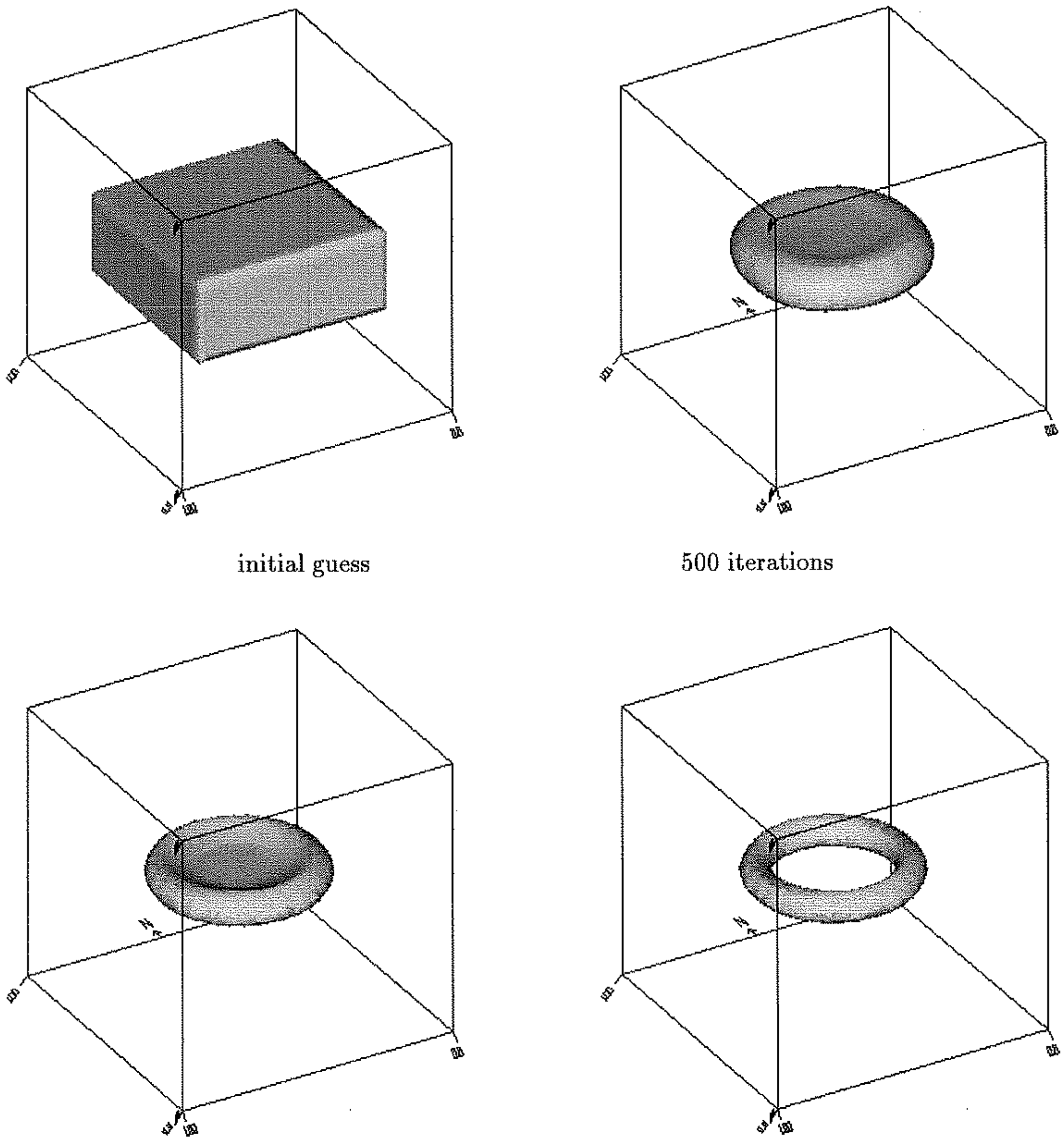

distance function
initial figure
p=2, t=0.15, dt=0.0003
p=2, t=0.225, dt=0.0003
q=2, t=0.325, dt=0.001
q=2, t=0.425, dt=0.001
q=2, t=0.525, dt=0.001
q=2, t=0.575, dt=0.001
p=2, t=0.6, dt=0.0003
100
80
60
40
20
20 40 60 80 100

Figure 6

Interpolation of a Torus, $R = 0.25,\ r = 0.06$



initial guess

500 iterations

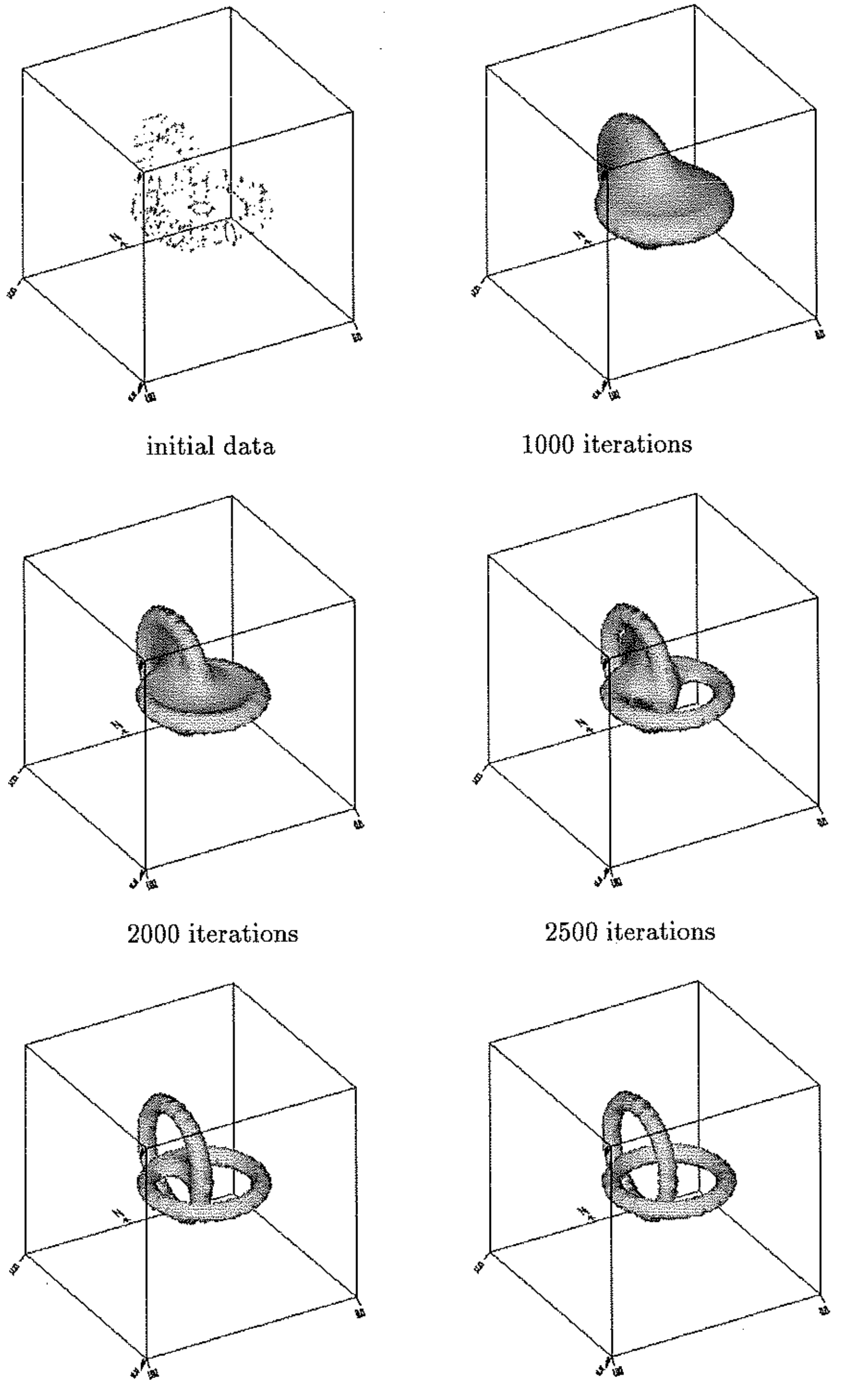
1000 iterations

1500 iterations

number of data points $= 30 \times 10,\ dx = 0.02, dt = 0.0002$

Figure 7

Interpolation of Two Linked Tori, $R = 0.24,\ r = 0.05$



initial data

1000 iterations

2000 iterations

2500 iterations

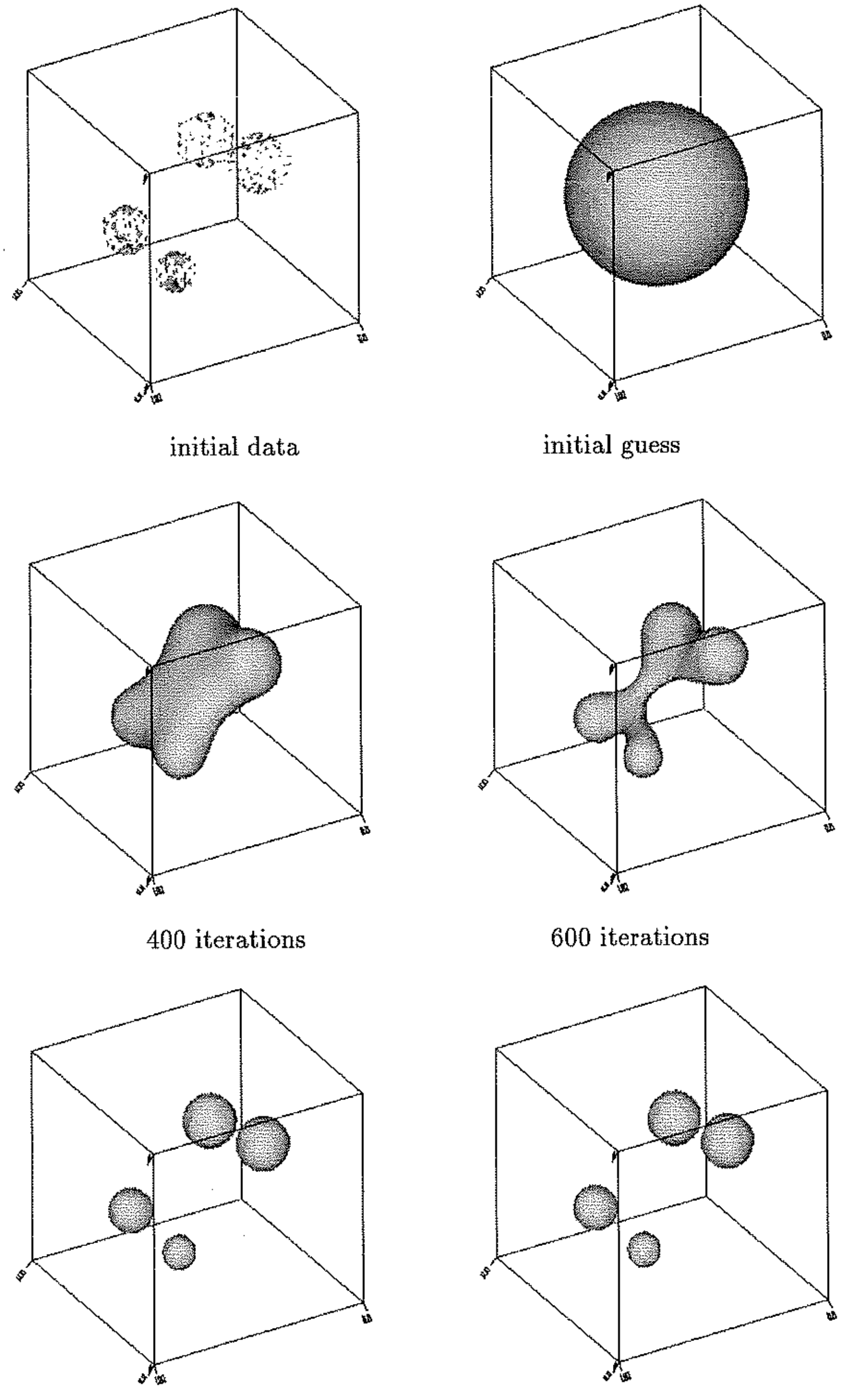
3000 iterations

3500 iterations

number of data points $= 25 \times 10 \times 2,\ dx = 0.02, dt = 0.0002$

Figure 8

Interpolation of Four Separate Spheres, $r = 0.12, 0.12, 0.1, 0.08$



initial data

initial guess

400 iterations

600 iterations

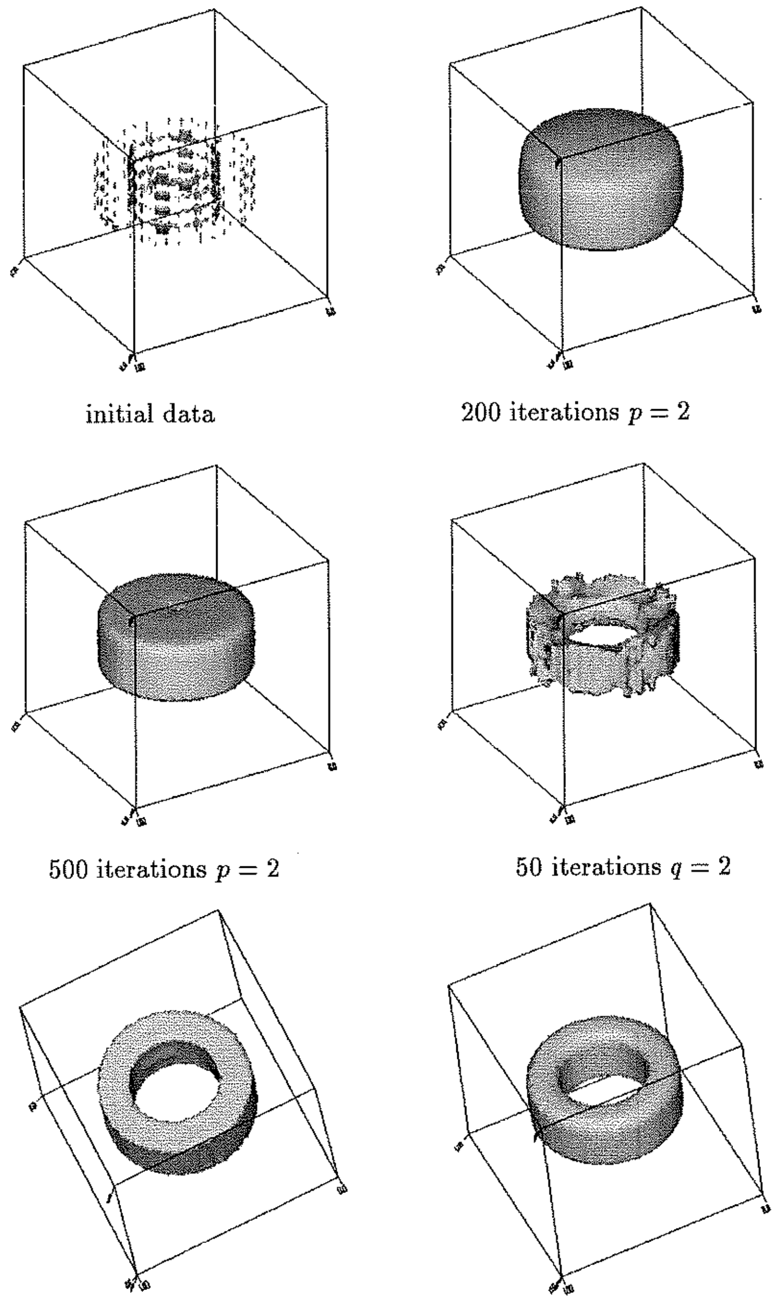
700 iterations

1200 iterations

number of data points = 100×4, $dx = 0.02, dt = 0.0002$

Figure 9

Interpolation of a pipe, $R = 0.35,\ r = 0.2,\ h = 0.3$



initial data

200 iterations $p = 2$

500 iterations $p = 2$

50 iterations $q = 2$

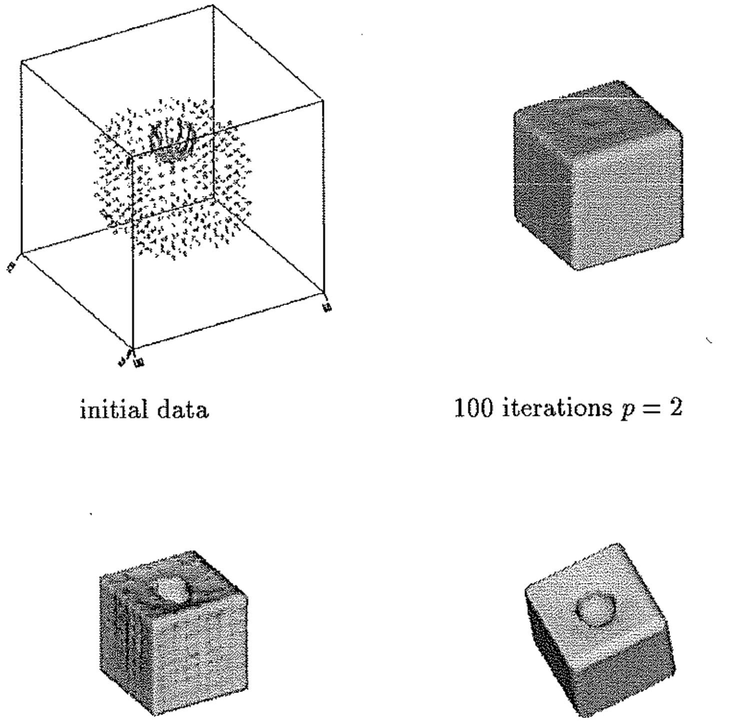
100 iterations $q = 2$

100 iterations $p = 2$

number of data points = 1200, $dx = 0.02$

Figure 10

Interpolation of a box with a rounded cave starting from a big box



initial data

100 iterations $p = 2$

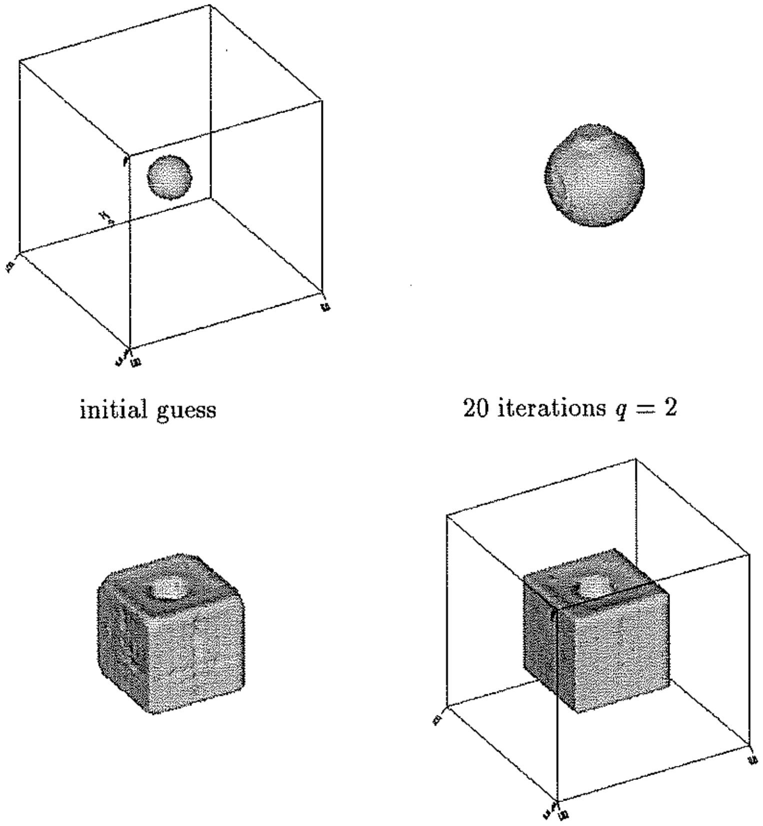
100 iterations $q = 2$

50 iterations $p = 2$

number of data points = 700, $dx = 0.02$

Figure 11

Interpolation of a box with a rounded cave starting from a sphere inside



initial guess

20 iterations $q = 2$

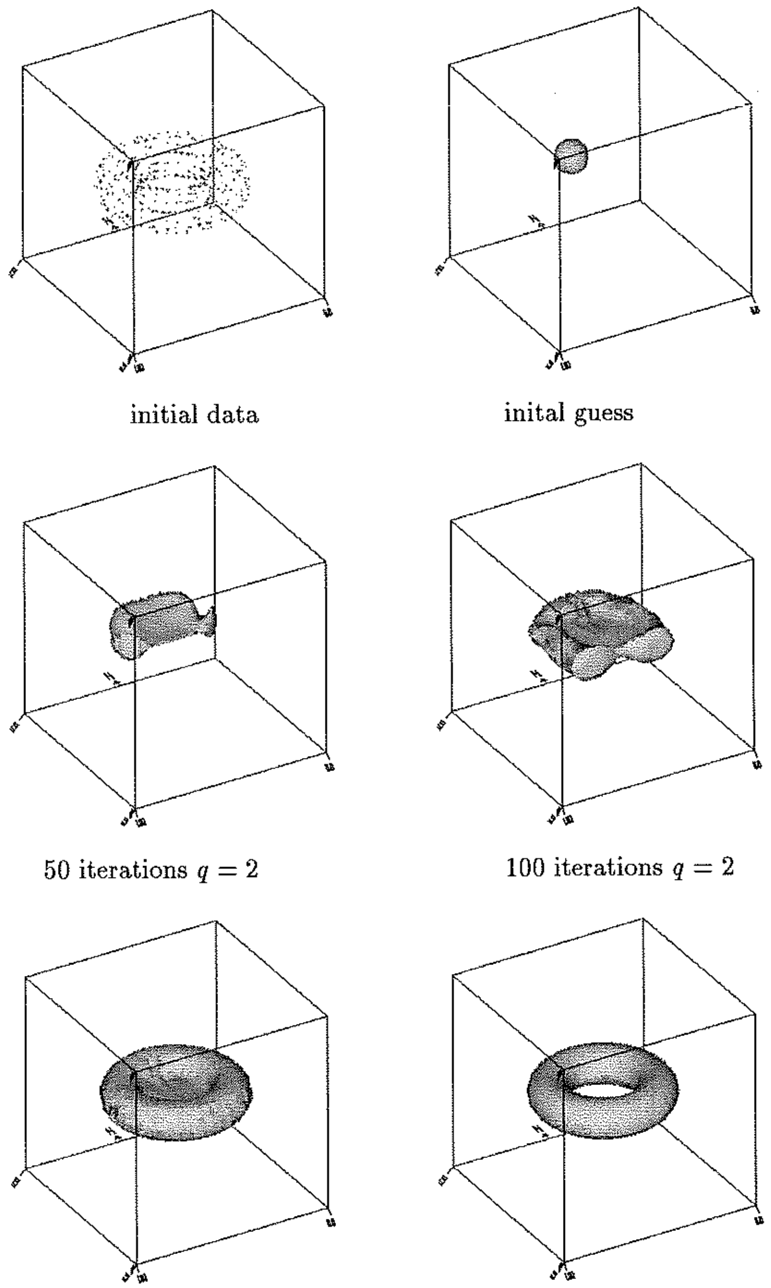
50 iterations $q = 2$

100 iterations $q = 2$

number of data points = 700, $dx = 0.02$

Figure 12

Interpolation of One Torus from a Small Sphere Inside $R = 0.25, r = 0.1$



initial data

inital guess

50 iterations $q = 2$

100 iterations $q = 2$

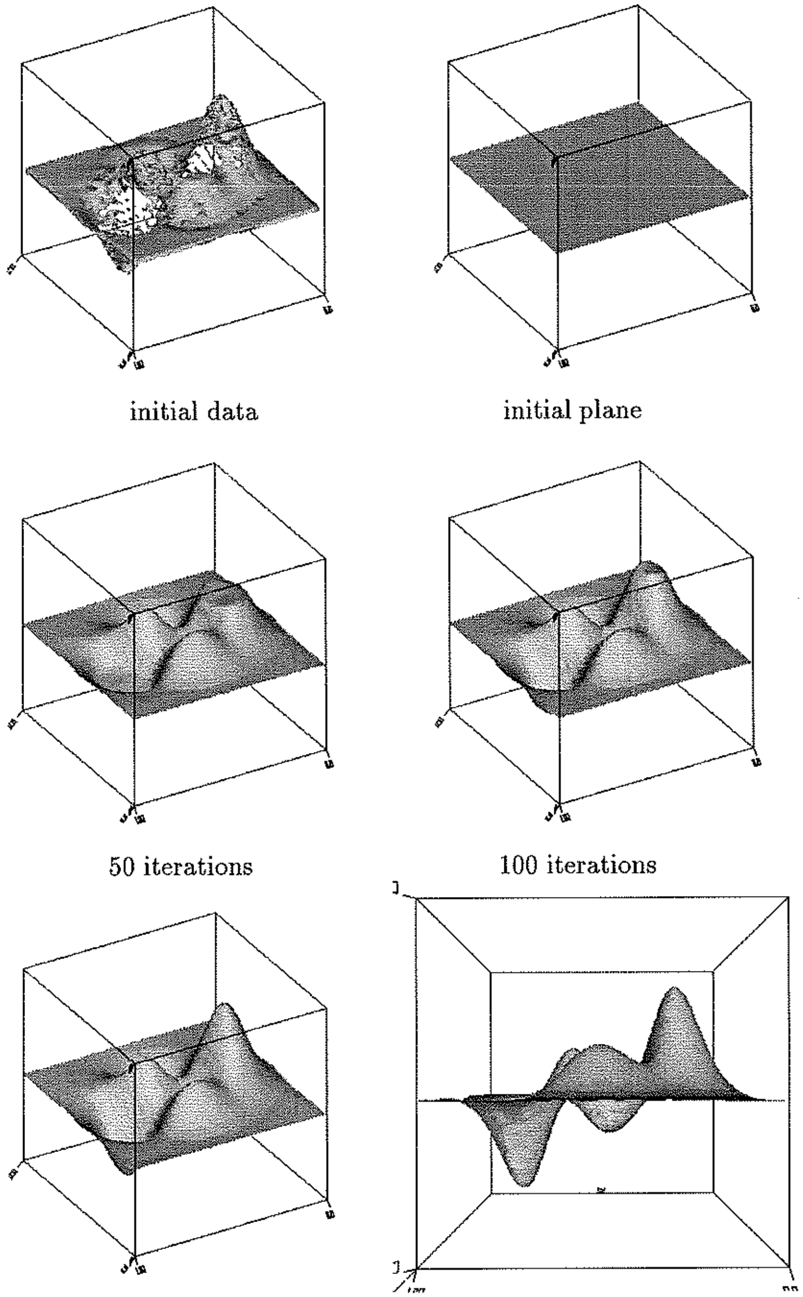
150 iterations $q = 2$

100 iterations $p = 2$

number of data points = 40×15, $dx = 0.02$

Figure 13

Interpolation the Graph of Peaks



initial data

initial plane

50 iterations

100 iterations

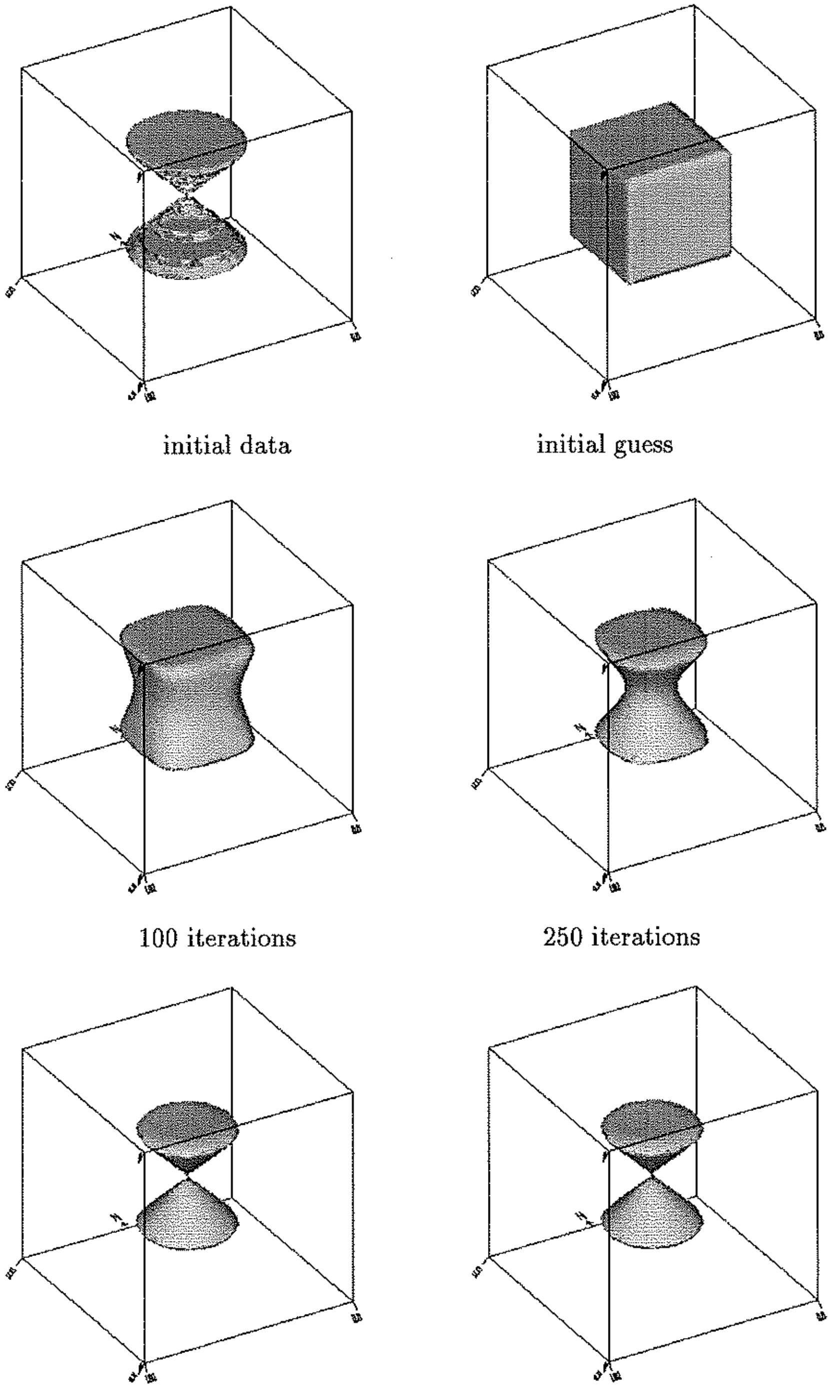
150 iterations

perspective view

number of data points = 50×50, $dx = 0.02$

Figure 14

Interpolation of Two Cones, $r = h = 0.25$



initial data

initial guess

100 iterations

250 iterations

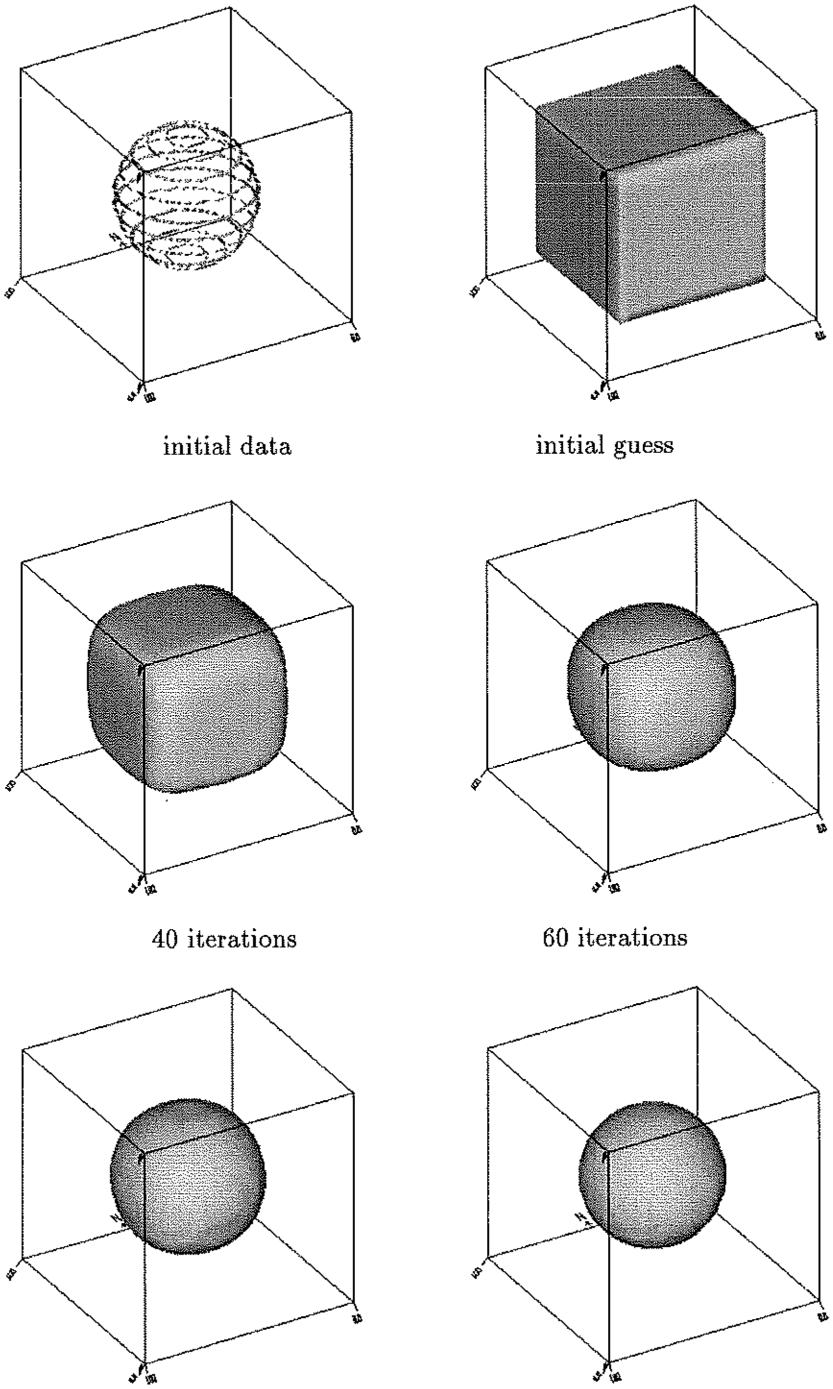
300 iterations

400 iterations

$dx = 0.02, dt = 0.0004$

Figure 15

Interpolation of One Sphere From Circles, $r = 0.3$



initial data

initial guess

40 iterations

60 iterations

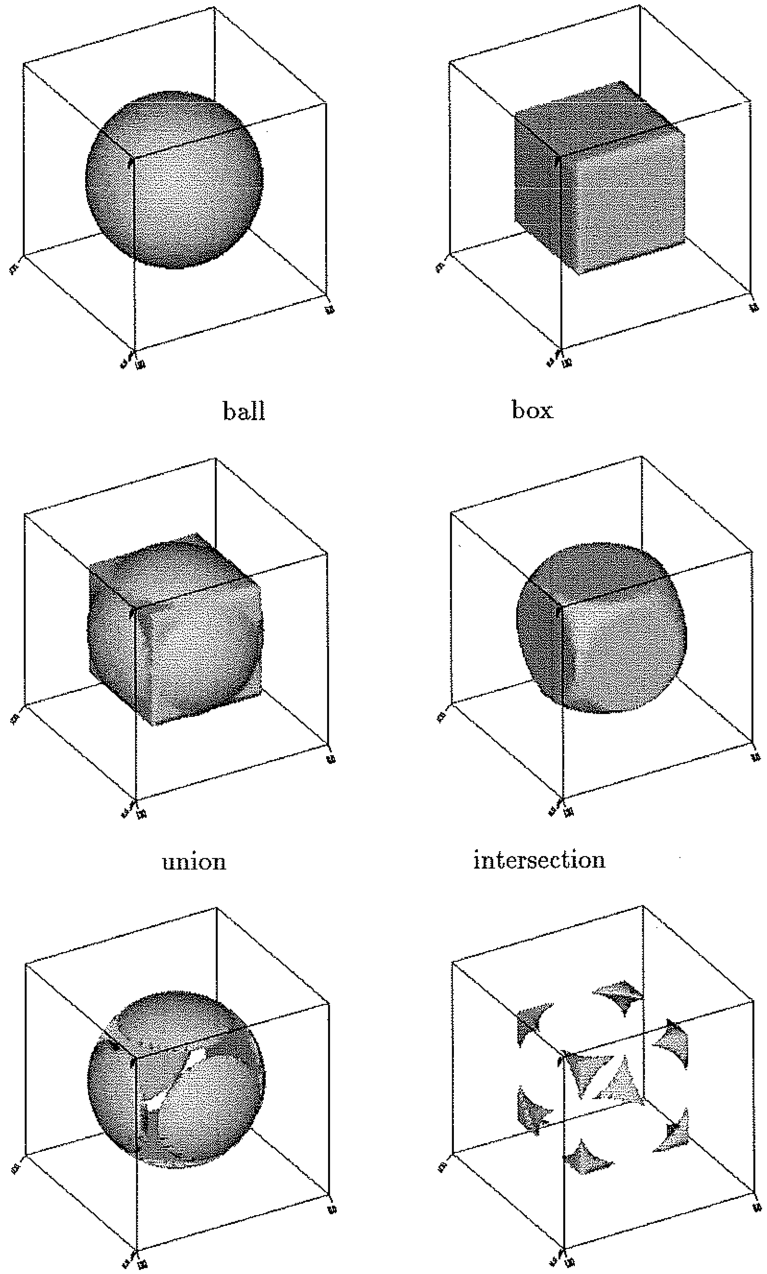
80 iterations

100 iterations

number of circles = 10, $dx = 0.04, dt = dx \times dx$

Figure 16

Boolean Operation of a Ball and a Box



ball

box

union

intersection

ball-box

box-ball

Figure 17

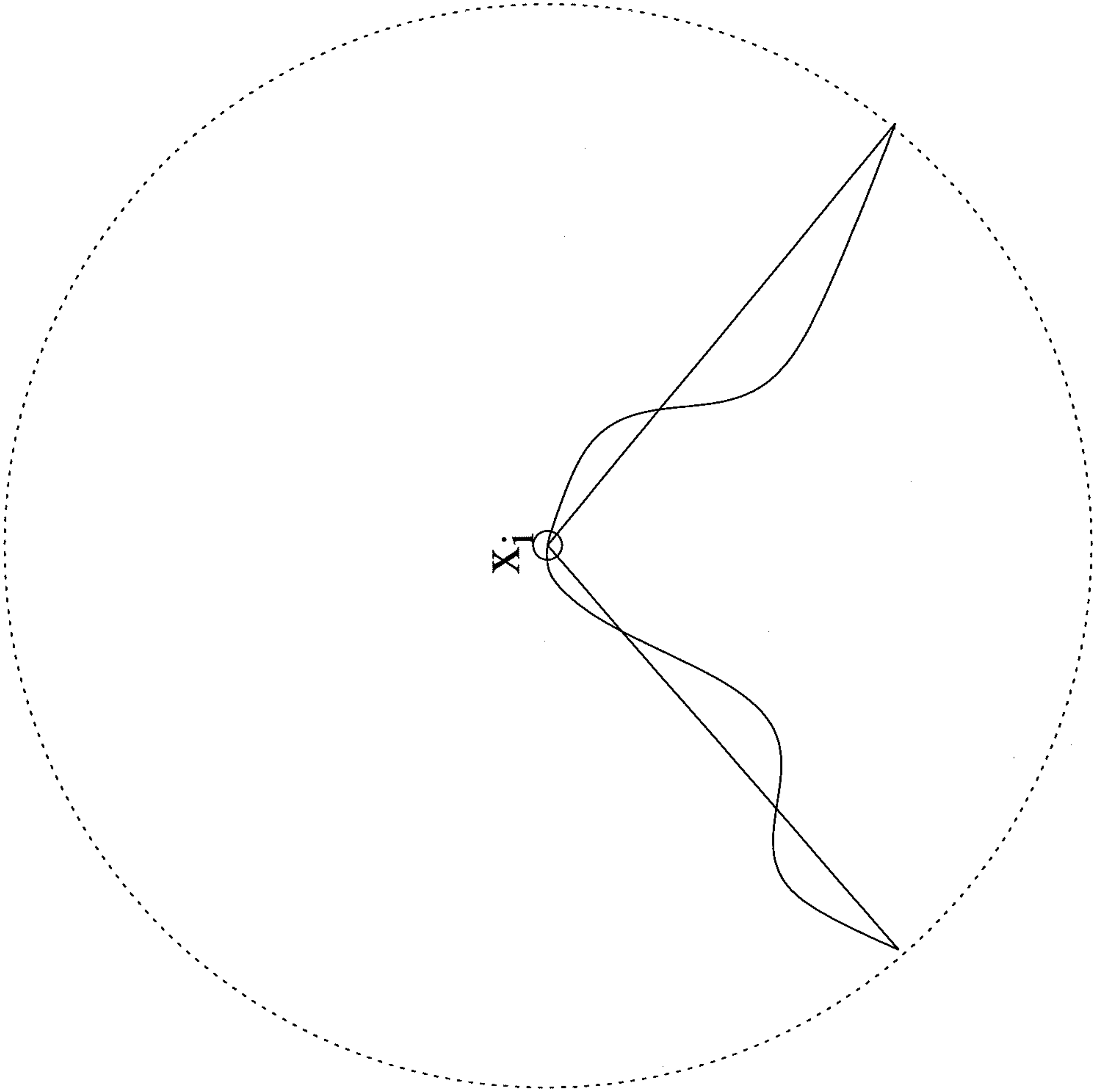


Figure 18

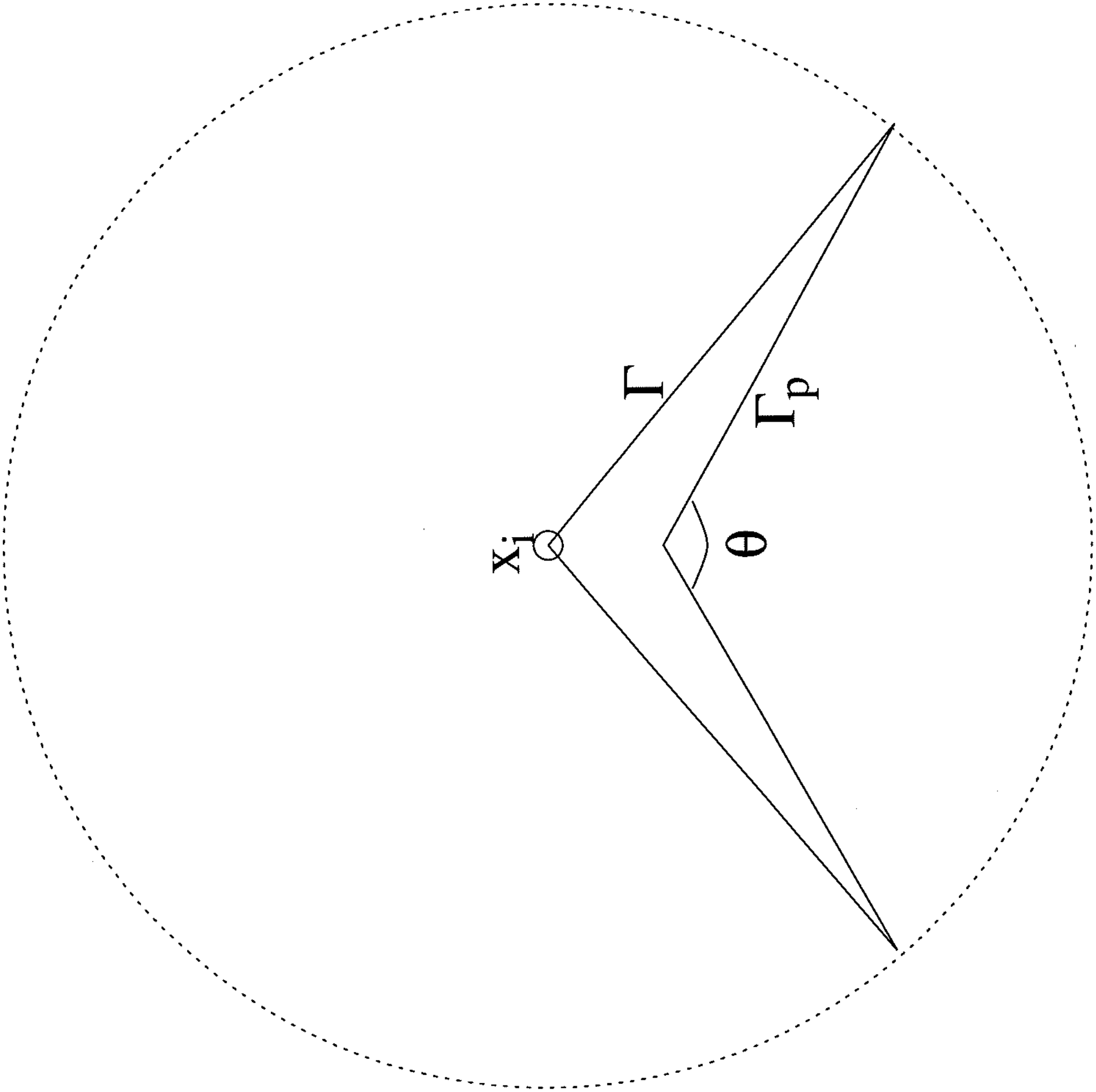


Figure 19

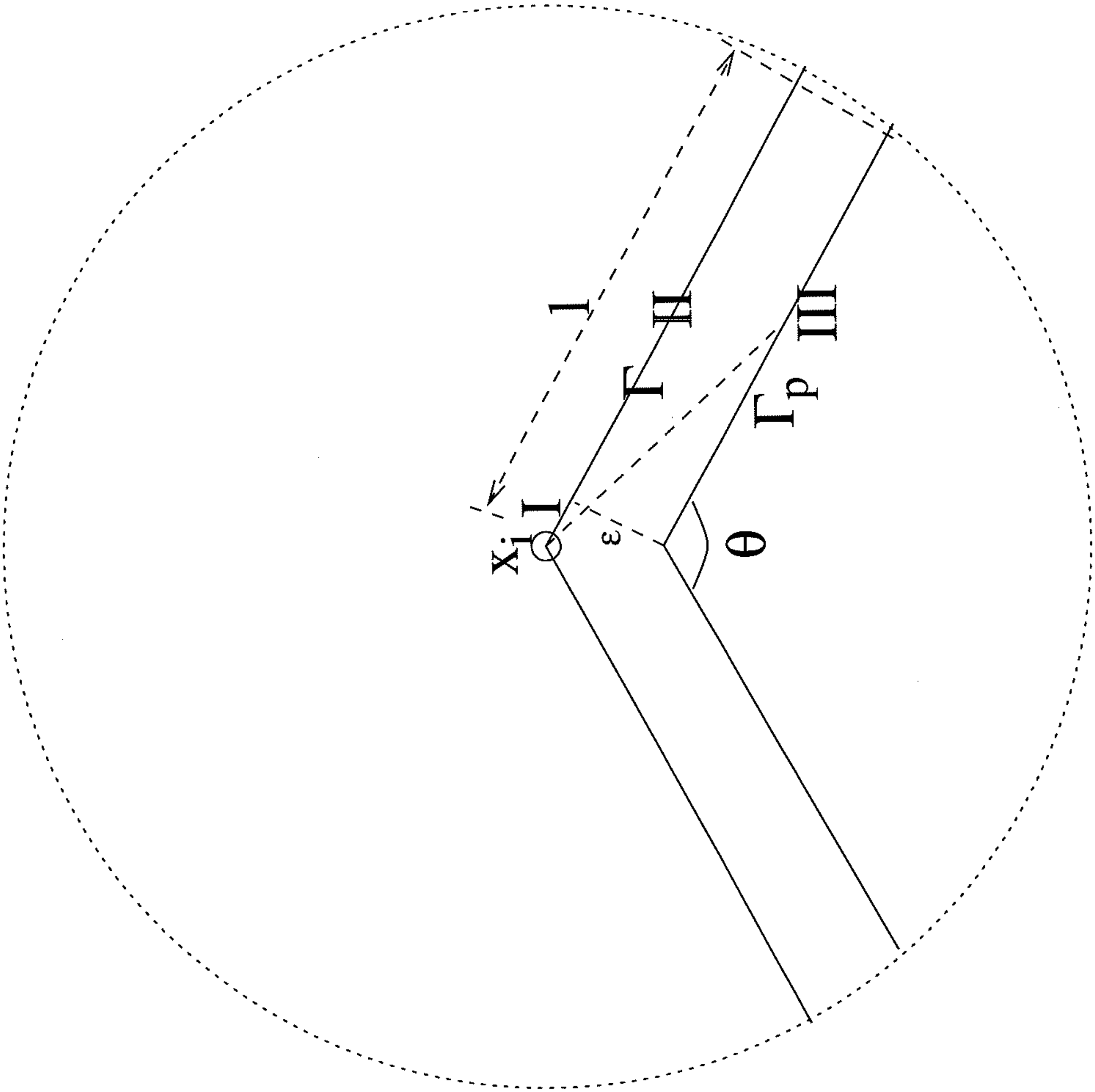


Figure 20