Université Catholique de Louvain

Faculté des Sciences Appliquées

Laboratoire de Télécommunication et de
Télédétection

Segmentation 3-D d'images de résonance magnétique

Mémoire présenté par
Luc Cole

En vue de l'obtention du grade de diplôme d'études complémentaires en sciences appliquées orientation télécommunication

Promoteur
Benoît Macq

Louvain-la-Neuve
Septembre 2004

Acknowledgements

I would like to thank everybody who has contributed to the realization of this thesis. Special thanks go to my supervisor Professor Benoît Macq for giving me the opportunity to realize this dissertation and for his guidance and support all through my studies at the Communications and Remote Sensing Laboratory of the Faculty of Engineering at the Université catholique de Louvain (UCL).

I am also grateful to Vincent Nicolas, whose door was always open to me and my everyday concerns. Owing to his knowledge and experience with VTK programming and medical studio he has offered me numerous insightful suggestions and valuable perspectives for realizing this work.

I gratefully acknowledge Eduardo Romero for his constant willingness to answer my questions mostly over the wire by Emails concerning the work he previously done on the level set methods at UCL.

I want to express my gratitude to Quentin Noirhomme, Annabelle Gouze for their suggestions, advice and ideas throughout this year. They have been particularly helpful and generous with their time and expertise.

I am grateful for the many friends that have supported me; too many to mention them all. My final thanks go to my mother and brother. I reckon growing up in such a lively, stimulating and therefore inspiring family life has been of considerable influence on my conduct in life. My mother always encouraged me and guided me to independence, never trying to limit my aspirations.

Abstract

This work adopts the notion of feature extraction for three-dimensional Magnetic Resonance Images of Multiple Sclerosis lesions. Multiple sclerosis is a progressive disease that requires an evolution study through time. The evolution of the disease can be followed on a patient with a temporal series of examinations that produce three-dimensional images of the brain. The purpose is to segment the Multiple Sclerosis lesions using a sophisticated computerized algorithm based on interface evolution.

Image segmentation is the process of extracting objects from a background thus partitioning the image into distinct regions. There exist three main methods for segmentation: Edge-based methods, pixel-based classification methods and region-based methods. This work gives a review of the mathematical background and the application for a specific region-based method called the level set method introduced by J.A. Sethian from University of California, Berkeley.

The level let method is in fact a numerical technique, which can follow the evolution of interfaces. These interfaces can develop sharp corners, break apart, and merge together. This method is a powerful numerical technique for tracking the evolution of interfaces moving under a variety of complex motions. Moreover, the level let approach is based on computing viscosity solutions to the appropriate equations of motion, using techniques borrowed from hyperbolic conservation laws.

This work considers the evolution of a front propagating along its normal vector field with curvature dependent speed. Furthermore, it defines an "energy-like" quantity of the moving front, the total variation, and it shows a general result relating the growth/decay of this energy to the speed. In addition it studies a front moving with speed dependent on the curvature, and shows that the curvature term plays a smoothing role in the solution. When the speed is constant, the solution is seen to blow up, differentiability is lost, and an entropy condition can be formulated to provide an explicit construction for a weak solution beyond blow up time. The equations of motion are numerically solved and it is also showed that the solution converges to the constructed weak solution as the curvature smoothing term vanishes. Finally, there is a discussion on the difficulties involved in numerically solving such problems and a possible remedy is described.

Table of Contents

Acknowledgements 2
Abstract 3
Introduction 5
Chapter 1 Propagating Fronts 7
1.1 Evolving interface representation 7
1.2 Boundary value problem 10
1.3 Initial value problem 10
1.4 Hamilton-Jacobi general equations 13
Chapter 2 Differential geometry basics 14
2.1 Curve parameterization 14
2.2 Curvature 15
2.3 Total variation 17
Chapter 3 Motion equations 19
3.1 Formulation 19
3.2 Properties of the level set equations 23
Chapter 4 Numerical algorithms 27
4.1 Marker methods 27
4.2 Cell method 33
4.3 Level set numerical algorithm 34
4.3.1 Numerical domain of dependence 35
4.3.2 1D Level set scheme 37
4.3.3 3D level set scheme 38
4.4 Speed function calculation 39
4.5 The Narrow Band method 41
Chapter 5 Software application 43
5.1 Algorithm 43
5.2 Level set scheme 44
Conclusions 47
References 49
Table of Figures 51

Introduction

A crucial step in medical image analysis is image segmentation. Segmentation is a process of isolating shapes from an image. In other words the segmentation technique is used to divide an image to separate zones by collecting adjacent pixels based on some relationship criterion. We shall apply a special scheme to the automatic detection, segmentation and tracking of lesions in Magnetic Resonance images of the brain, in particular, multiple sclerosis lesions.

Multiple sclerosis is a neurological disease primarily affecting the central nervous system. The disease is characterized by the presence of areas which have a destructive lack of *myelin*, an insulating and protective fatty protein that covers nerve cells (neurons) and increases the speed that nerve signals move down the neurons.
The areas which lack of myelin and are characterized by an inflammation around the blood vessels in the brain's white matter is known as multiple sclerosis lesions.

Segmentation of multiple sclerosis lesions from Magnetic Resonance (MR) brain images is important in the diagnosis of the disease. However, manual detection and analysis of these lesions from MR brain images are usually time consuming, expensive and can produce manual mistakes. Automated procedures offer the advantage of producing consistent results in much less operator time. Hence automating the delineation of Multiple sclerosis lesions from MR brain image is a crucial task.

For our purpose, in the multiple sclerosis case, the purpose of segmentation is to identify and extract the boundaries of regions of high grayness intensities, the lesions, which are microscopic appearance of diseased white matter. Figure 1 shows a MRI image of a brain with multiple sclerosis lesions.

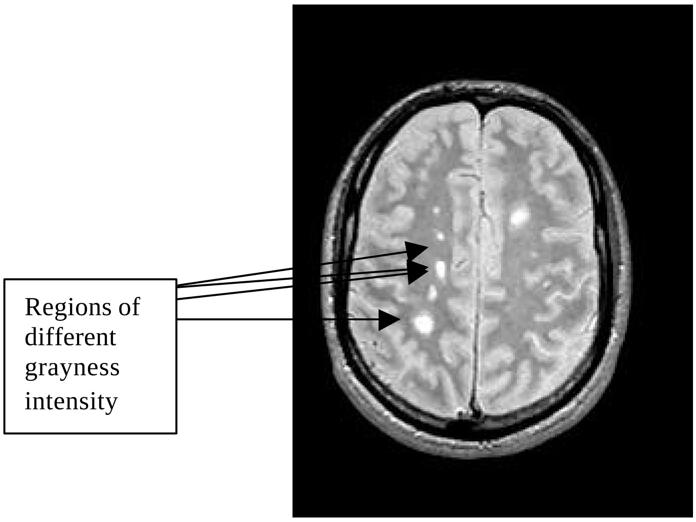

Figure 1 Multiple sclerosis lesions

Image segmentation is divided into three main categories:

- *Pixel-based methods*, in which decisions are based on local pixel information, derived from the histogram statistics of the image. Most of these methods are classification techniques that use thresholding in order to partition the image into regions. These methods are not very resistant to noise.
- *Edge-based methods*, in which the shape boundaries are being analyzed and contour, are being detected. The weakness of these methods in connecting together broken contour lines makes them prone to failure in the presence of blurring.
- *Region-based methods*, in which the main idea is to start with, some initial boundary shape represented in the form of curves, and iteratively modify it by applying various shrink/expansion operations according to some energy function. These active contour methods are in fact curves, which evolve while merging neighborhood pixels till a stopping criterion is satisfied, and segmentation is completed.

In this work one specific region-based method is used called the level set method which provides mathematical and computational tools for naturally tracking evolving interfaces with sharp corners, topological changes, and three-dimensional complications.

Chapter 1
Propagating Fronts

This chapter describes the mathematical formulation for propagating fronts. We shall consider an interface that separates one region from another. This interface could be a curve in two dimensions or a surface in three dimensions. Furthermore, the curve moves in a direction normal to itself with a known speed. This speed may be influenced by several parameters such as local parameters (relate to the geometric properties like curvature), global parameters (relate to the shape of the front like integrals along), and independent parameters relate to external properties like velocity of a fluid that carries the front to a specific direction.

The purpose of this chapter is to find the rule of motion of an interface as it evolves. The central idea is to follow the evolution of a function f *whose zero level set always corresponds to the position of the propagating interface. The motion for this evolving function* f *is determined from a partial differential equation in one higher dimension, which permits cusps, sharp corners, and changes in topology in the zero level set describing the interface.*

1.1 Evolving interface representation

One of the obvious ways to describe a moving front is by a graph of a function that by its definition gives a unique value of the front at any point in space. To illustrate the problem we shall start with a simple example in one dimension. Consider the upper half of a circle with unit radius as an initial front moving in a normal direction with unit speed as shown in Figure 2. The mathematical function of an upper half circle is given by

$$y = \sqrt{1 - x^2}, \quad -1 < x < 1,$$

and thus the derivative of y with respect to x is given by

$$y_x = \frac{-x}{\sqrt{1 - x^2}}.$$

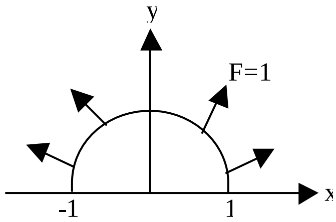

Figure 2 initial front moving in normal direction with speed F = 1

The y coordinate is in fact the height of the propagating function at time t, while the tangent at (x, y) is $(1, y_x)$. Lets also denote by y_t the change (with respect to time) in the height of the function curve. Using these notations Figure 3 illustrates a zoom in on arbitrary point propagation with speed F.

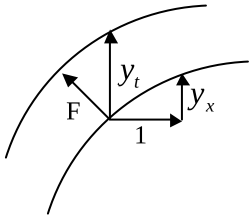

Figure 3 propagating graph parameters

By using simple geometrical rules of congruence of triangles one can derive that the change in height y_t is related to the speed F in the normal direction by

$$\frac{y_t}{F} = \frac{\sqrt{1+y_x^2}}{1}$$

$$\Downarrow$$

$$y_t = F\sqrt{1+y_x^2}$$

and thus given a unit speed while substituting the y derivative with respect to x gives

$$y_t = \sqrt{1 + \left(\frac{-x}{\sqrt{1 - x^2}} \right)^2},$$

$$y_t = \sqrt{\frac{1 - x^2 + x^2}{1 - x^2}} = \frac{1}{\sqrt{1 - x^2}},$$

Solving a simple differential equation with respect to t gives

$$y = \frac{1}{\sqrt{1 - x^2}} t + f(x),$$

where $f(x)$ is a constant depending the initial position x of the point. For example the front position after one second results in a half circle with two units radius as shown in Figure 4.

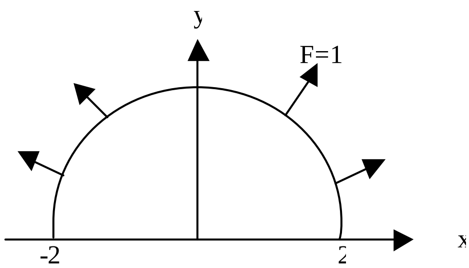

Figure 4 moving front in normal direction with speed F = 1 after one second

One drawback of this method is that it only works for initial curves that can be written as the graph of a function. Numerical algorithms trying to solve the differential equation would have problems to converge. Another limitation here is the use of a coordinate system, though it has nothing to do with the problem, but still severely restrains our options.

Rather than follow the interface itself, the level set approach instead takes the original interface and adds an extra dimension to the problem. This idea which may seem rather peculiar at first glance will prove to miraculously solve the problem related to the changing of the evolving front that result in a curve that stopped being a function.

The next sections describe the level set method formulations of two cases. One approach called the *Boundary value problem* when the speed is strictly positive and the other, more general, called the *initial value problem* when the speed could be any value.

1.2 Boundary value problem

The boundary value problem limits to a case where the speed is always positive, therefore, the curve moves outwards. Lets consider a grid of crossing time values $T(x, y)$ which determines a function in two dimensions. At each grid point, the function evaluates the time at which the front crosses any point in the plain x,y. The initial position of the front is the boundary for the arrival time surface $T(x, y)$ at $T = 0$. An example of the arrival surface $T(x, y)$ for a circular front evolving with unit speed is shown in Figure 5.

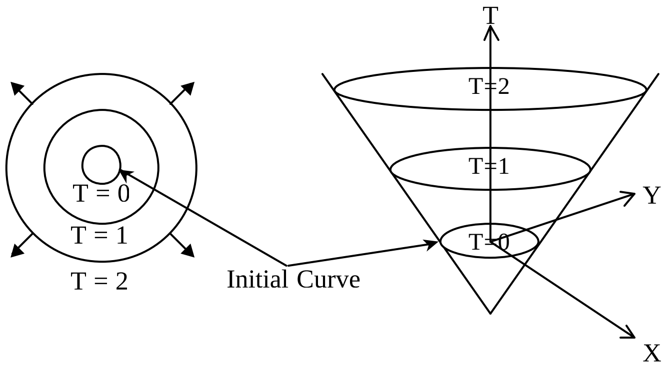

Figure 5 Front motion in higher dimension gives the boundary value problem

Taking into account that speed is proportional to the derivative of space with respect to time then the directional derivative of $T(x, y)$ obeys the following partial differential equation (in two dimensions):

$$\left(\left|\nabla T(x,y)\right| F \equiv\right) \quad \sqrt{\left(\frac{\partial T}{\partial x}\right)^2 + \left(\frac{\partial T}{\partial y}\right)^2} \cdot F = 1$$

A solution to the boundary value problem is given by the fast marching method that starts from the initial curve being the lowest spot on the surface T and iteratively builds the surface that corresponds to the front moving with the speed F.

1.3 Initial value problem

The initial value problem describes the time-evolution of an interface on the basis of an initial state curve, and it is a more general approach to the boundary value problem due to the fact that the speed F can have any value, positive or negative.

In this case the arrival time $T(x, y)$ is not anymore a single-valued function. One way of taking care of this is to embed the initial position of the front as the zero level set of a higher–dimensional function f. This cone-shaped surface f, shown in Figure 6, is a function of space and time that gives the distance from any point to the interface at a certain time t, and has the property that it intersects the xy plane exactly where the propagating front sits.

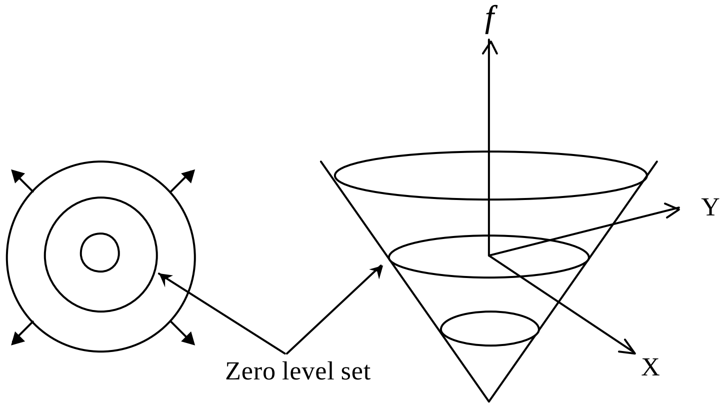

Figure 6 The level set function and the initial curve that lies in xy plane

First, lets define a point coordinate as $(x, y) \equiv \vec{x}$, and furthermore with abuse of notation let us drop the vector sign and note a point by x.
From the definition of the level set function, a point x located on the front must obey the following equation

$$f(x(t), t) = 0,$$

since the propagating interface always corresponds to the place where $f = 0$. By taking the time derivative of both sides an applying the chain rule from calculus one can obtain

$$f_t + \nabla f \cdot x_t = 0.$$

F gives the speed of any point in the direction of the normal

$$x_t \cdot \vec{n} = F,$$

and from differential geometry the normal to the level set surface is given by

$$\vec{n} = \frac{\nabla f}{|\nabla f|}.$$

Combining the last two equations gives

$$x_t \cdot \frac{\nabla f}{|\nabla f|} = F$$
$$\Downarrow$$
$$x_t \cdot \nabla f = F|\nabla f|$$

thus the level set partial differential equation is given by

$$f_t + F|\nabla f| = 0 \text{, when } f(x, t = 0) \text{ is the initial front}$$

The level set function can expand, rise or fall while the propagating front could always be obtain by plotting the zero contour. Figure 7 shows falling of the level set function while the initial front evolution is marked in bold.

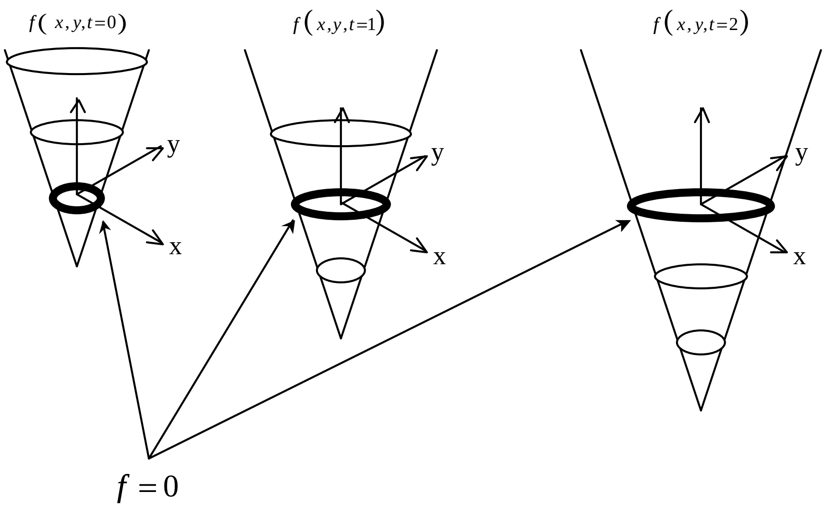

Figure 7 Evolution of the level set function with respect to time

A very strong advantage of the level set method relies on the fact that topological changes in the evolving front are easily handled. The front need not be a single curve but several that can merge and break as the level set cone evolves in time, nevertheless f remains single valued as shown in Figure 8.

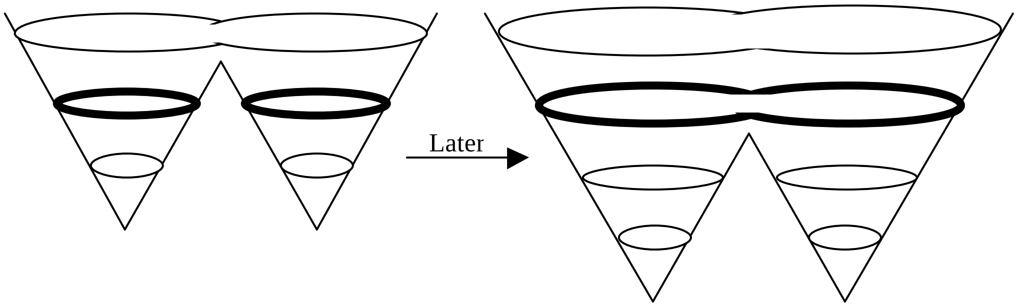

Figure 8 Evolution of two cones of the level set function while the initial front merges

1.4 Hamilton-Jacobi general equations

The level set formulation for the initial value problem is given as

$$f_t + F\left|\nabla f\right| = 0$$

or more precisely as

$$f_t + F\sqrt{f_x^2 + f_y^2 + f_z^2} = 0$$

and it is in fact a special case of the more general *Hamilton-Jacobi* equations given as

$$a f_t + H\left(f_x, f_y, f_z, x, y, z\right) = 0\text{, for } a = 1$$

which has special properties that interest us and will be discussed later.

Furthermore the boundary value problem given as $\left|\nabla T\right|F = 1$ may be also written in a Hamilton-Jacobi form by writing $H \equiv \left|\nabla T\right|F - 1$ and $a = 0$.

Chapter 2
Differential geometry basics

This chapter describes some basic concepts in differential geometry, which are extremely important for the understanding of curve propagation. Notions like parameterization, curvature and total variation are essential for building a computer model of an evolving interface.

For all following sections we shall use the notation of a curve a in a parameterized form by $a \equiv \vec{x}(s,t) \equiv (x(s,t), y(s,t))$, while the derivation of a with respect to the parameter s is simply performing the derivation of each component of the curve. The first and second derivatives of a with respect to s is given by

$$a_s \equiv (x_s(s,t), y_s(s,t)), a_{ss} \equiv (x_{ss}(s,t), y_{ss}(s,t)) \ .$$

2.1 Curve parameterization

An important step towards solving the level set partial differential equation is to move from a coordinate system to parameterization by arclength. The arclength will express how far along the curve we are and it shall be denoted by s.

Let a be a propagating curve in the xy plane, when t is time and $0 \le s \le S$ is the parameterization by arclength. It can be easily seen that if we walk across the curve in increasing direction of s we will eventually arrive to the starting point after walking for a distance of S, which is the arclength, i.e. $\vec{x}(0,t) = \vec{x}(S,t)$. By calculus the arclength of the curve is given by the following formula

$$S = \oint |a_s| ds$$

A positive direction of increasing s is defined to be the direction in which the interior of the curve is on the left side for a simple smooth and close curve as shown in Figure 9.

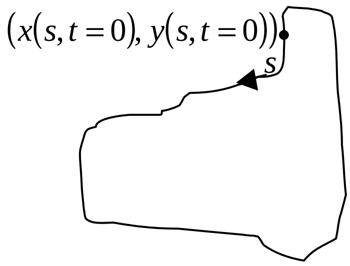

Figure 9 curve parameterization by arclength s

2.2 Curvature

The curvature measures the deviation of a curve from being a line, in other words it determines how much a curve can bend at each point. The curvature concept is extremely important in finding smooth solutions to the level set equation. Motion by curvature gives solutions that are stable and that eventually converge.

The plan is to find an expression for the curvature in a parameterized form. In order to do that it is important to describe some basic notion in the geometry of curves, which is in fact the twisting and turning of the curves. The geometry of all curves may be completely described by understanding how three perpendicular "fingers" attached to the curve vary while moving along the curve. This "fingers" are the *unit tangent vector* T, the *unit normal vector* N and the *binormal vector* B (defined by $B \equiv T \times N$). The three-vector frame is shown in Figure 10.

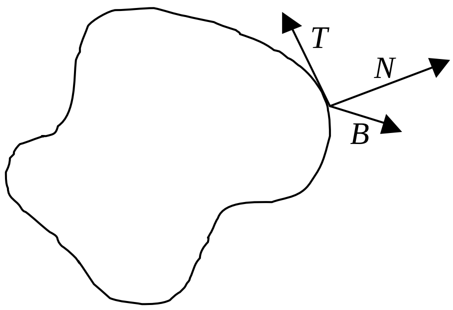

Figure 10 Three-vector frames

This three-vector frame are related together by three differential equations know as the *Frenet Formulas* given as

$$
\begin{aligned}
T_s &= \quad k \cdot n \cdot N \\
N_s &= -k \cdot n \cdot T + t \cdot n \cdot B \\
B_s &= -t \cdot n \cdot N
\end{aligned}
$$

where n is the parameterization stretch, k is the curvature, t is the torsion, that will be discussed shortly.

The unit tangent vector to a curve a is given by derivating a with respect to the parameterization and normalizing to get a unit vector, i.e. $T = \frac{a_s}{|a_s|}$. The normalization factor in the denominator is more often called the parameterization stretch metric and it is given by

$$n \equiv |a_s| = \sqrt{x_s^2 + y_s^2}\ .$$

The curvature $k(s)$ is defined by the absolute value of the derivative of the tangent vector normalized by n, i.e. $\frac{1}{n}|T_s|$. Of course $k \geq 0$ and k increases as a turns more sharply.
By the definition of the curvature $k(s)$ and using Frenet formulas it can be shown that

$$k \equiv |T_s| = \frac{a_s \times a_{ss}}{|a'|^3} \equiv \frac{a_s \times a_{ss}}{n^3} = \frac{y_{ss}x_s - x_{ss}y_s}{\left(x_s^2 + y_s^2\right)^{3/2}}$$

The torsion t measures the deviation of a curve from being contained in a plane. This discussion deals with plane curves therefore the torsion has little significance.

As an example of the curvature concept, lets consider a circle with radius r given by the parametric representation $a(s) = \left(r\cos\frac{s}{r}, r\sin\frac{s}{r}\right)$. The derivative of a is given by

$$a_s = \frac{1}{r}\left(-r\sin\frac{s}{r}, r\cos\frac{s}{r}\right) = \left(-\sin\frac{s}{r}, \cos\frac{s}{r}\right),$$

while the second derivative of a is given by

$$a_{ss} = \frac{1}{r}\left(-\cos\frac{s}{r}, -\sin\frac{s}{r}\right).$$

Note that a has unit parameterization form, i.e. $|a_s| = \sqrt{\left(-\sin\frac{s}{r}\right)^2 + \left(\cos\frac{s}{r}\right)^2} = 1$. Using the definition of the curvature one can obtain that

$$k=\frac{y_{ss}x_s-x_{ss}y_s}{\left(x_s^2+y_s^2\right)^{3/2}}=\frac{\left(\frac{1}{r}\sin\frac{s}{r}\right)\left(\sin\frac{s}{r}\right)+\left(\frac{1}{r}\cos\frac{s}{r}\right)\left(\cos\frac{s}{r}\right)}{\left(\left(-\sin\frac{s}{r}\right)^2+\left(\cos\frac{s}{r}\right)^2\right)^{3/2}}=\frac{1}{r}$$

That is, for a circle of radius r, the curvature is constant and equal to $\frac{1}{r}$. The amount of bending at each point is constant; therefore, for a circle of radius r, the curvature is constant and equal to $\frac{1}{r}$. Furthermore it is inversely proportional to the radius, thus the smaller a circle is the higher it curves, which is intuitive.

2.3 Total variation

An interesting question is what happens to a smooth closed curve as it moves with speed proportional to its curvature. The answer to this is that some parts of the curve move outwards while others move inwards. In the figure below (Figure 11) the sign of the curvature is marked beside the curve and the arrows show the motion direction given that the speed is $F=-k$.

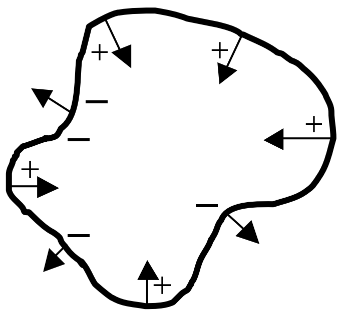

Figure 11 Motion with speed proportional to the curvature

The above curve will eventually smooth out and become circular in a way that the total “oscillation” of the curve decreases. These oscillations called also *total variation* could be mathematically quantified by integrating the curvature absolute value along the arclength

$$Var=\int_0^S |k(s)|n(s)ds$$

The total variation evolves with respect to time so it is normally a function of time. It is important to know if the total variation will decrease or increase with time, i.e. the sign of $\frac{d\,Var(t)}{dt}$. In other words we would like to know if during the propagation the front is going to smooth out or not, this way conditions for convergence of the differential equation of the level set could be obtained.

Given a non-convex initial curve moving along its normal vector with speed $F(k)$, it can be shown that the sign of $\frac{d\,Var(t)}{dt}$ is the same as the sign of $F_k(k=0)$ (i.e. when the curvature changes sign). In other words the front will smooth out if the derivative of the speed with respect to the curvature is negative at points of zero curvature.

For a convex curve the curvature is always positive therefore the absolute sign in the integral of the total variation could be dropped which leaves us with

$$Var = \int_0^S kn ds = 2p$$

and the derivative of the total variation with respect to time is zero which is trivial.

Chapter 3
Motion equations

The level set technique tracks moving interfaces in a wide variety of problems. It relies on the relation between propagating interfaces and propagating shocks. We shall see in this chapter that the equation for a front propagating with curvature dependent speed is linked to a viscous hyperbolic conservation law for the propagating gradients of the fronts. This chapter deals with some of the difficulties involved solving the motion equation in certain cases where corners can develop and create singularities in the solution.

3.1 Formulation

As a first step towards finding solutions to the level set equation and further to the discussion regarding motion of curves parameterization given in Chapter 2 we shall derive the equations of motion by curvature in two dimensions.

Consider a simple closed curve moving along its normal direction $\vec{N}$ with speed F being a scalar function of the curvature. Using the arclength parameterization s discussed in the previous chapter the equations of motion are given by

$$a_t = F \cdot \vec{N} .$$

An illustration of the parameters of the propagating curve is give in the following figure

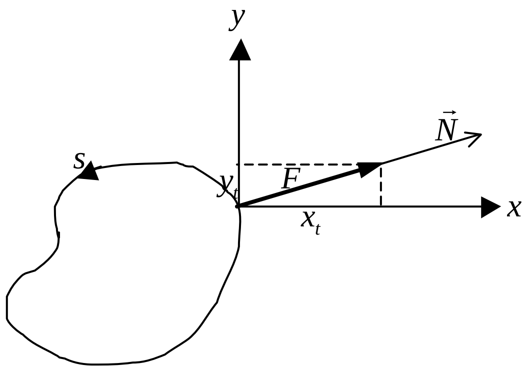

Figure 12 Propagating curve

According to the second equation of Frenet given in the previous chapter the normal to the curve in the parameterized form is given by

$$\vec{N} = \frac{1}{n}\left(y_s, -x_s\right).$$

This form is also referred to as the Lagrangian representation. Therefore an explicit expression of the motion is given by

$$x_t = \quad F(?) \cdot \frac{y_s}{?}$$

$$y_t = -F(?) \cdot \frac{x_s}{?}$$

For example, consider an initial cosine curve given by the following representation

$$a\left(s, t = 0\right) = (1 - s, 1 + \cos(s))$$

moving in the normal direction with unit speed, i.e. F = 1. The front is illustrated in the figure below

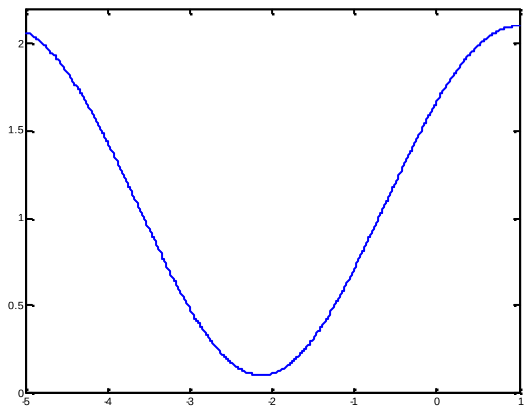

Figure 13 Initial cosine front moving in the normal direction

A solution to the equations of motion is given by performing and integration of each equation of motion with respect to time and substituting F by one, hence

$$x(s,t) = \int \frac{y_s}{?} dt = \left. \frac{y_s}{?} \right|_{t=0} \cdot t + x|_{t=0} = \frac{-\sin(s)}{\sqrt{1+(-\sin(s))^2}} t + 1 - s$$

$$y(s,t) = -\int \frac{x_s}{?} dt = -\left. \frac{x_s}{?} \right|_{t=0} \cdot t + y|_{t=0} = \frac{1}{\sqrt{1+(-\sin(s))^2}} t + 1 + \cos(s)$$

One possible solution is illustrated in the following figure

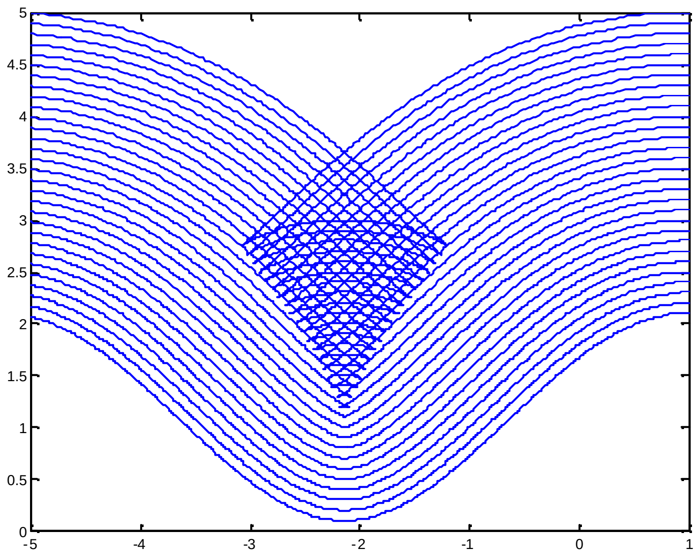

Figure 14 Possible Solution to the moving front

This solution above satisfies an integral form of the motion equations due to the fact that the differential equations were integrated with respect to time, therefore the solution is said to be a *weak solution* to the partial differential equation.

From one hand weak solutions are more flexible in the sense that they allow solutions which are not differentiable, or even continuous, while from the other hand singularities in the curvature of the solution can develop corners that eventually could allow the solution to expand beyond the formation of the shock.

Figure 14 is a typical example of a corner development while starting from an initial front that is completely differentiable in any order. The front evolved while in a certain time t it “braked” and created a corner that split into two moving fronts intersecting each other. Imposing the Entropy condition on the solution to the differential equation could prevent this additional undesired “tail”.

Entropy relates to the amount of information available in a solution. The entropy condition is a certain rule imposed on a solution that prevents new information (referring to the tail) to be created while the front evolves, thus whenever a shock develops it selects the correct solution to stop from being multiple-valued.

In the example above, after forcing the Entropy condition the additional tail in the solution will disappear as shown in the following figure

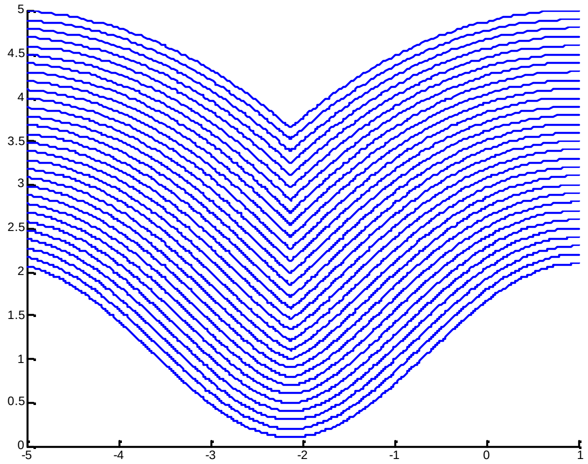

Figure 15 Entropy solution of cosine front moving in the normal direction with F = 1

3.2 Properties of the level set equations

As we have seen before the level set formulation for the initial value problem is given as $f_t + F|\nabla f| = 0$ or more precisely as $f_t + F\sqrt{f_x^2 + f_y^2 + f_z^2} = 0$ is in fact a special case of the more general Hamilton-Jacobi equations given as

$$a\, f_t + H(f_x, f_y, f_z, x, y, z) = 0\,.$$

The goal is to find a weak solution to the level set equations that satisfy the Entropy condition that will allow the solution to remain smooth and prevent singularities.

Furthermore it has been mentioned before that the total variation will decrease when the speed function derivative with respect to the curvature is negative.

Instead of choosing a unit speed we can choose a speed function of the form

$$F = 1 - ek, \quad e > 0$$

which has a constant smoothing term e that adds a diffusion part to the equation that prevents the development of corners, thus giving a negative derivative of the speed function with respect to the curvature that equals to $F_k = -e$.

It can be shown that the choice of a solution that propagates by curvature with a speed function of the form $F = 1 - ek$ when the "smoothness" term e goes to 0 gives the desired entropy solution. This is known as the *viscous limit*.
More formally written, given a solution $f(e)$ to a propagating curve with speed $F = 1 - ek$ then

$$f(e) \xrightarrow{e \to 0} f$$

where f is the solution obtained by a unit speed function (i.e. $F = 1$). The solution $f(e)$ also obeys the rules of uniqueness given certain initial conditions.

Moreover, there is a tight link between the general Hamilton-Jacobi equation solution and a viscosity solution. In fact it can be proven that a smooth solution to the Hamilton-Jacobi equation is a viscosity solution, and the opposite is also true given a differentiable viscosity solution.

Taking the Hamilton-Jacobi equation given as

$$f_t + H(f_x, f_y, f_z) = 0 \text{ , when}$$

$$H(f_x, f_y, f_z) \equiv F\sqrt{f_x^2 + f_y^2 + f_z^2}$$

and focusing on a one-dimensional version, that is,

$$f_t + [H(f)]_x = 0$$

gives a special form which is known as a *hyperbolic conservation law*, which have been studied in considerable detail. As described before we are looking for weak solution for the differential equation. In order to find a weak solution we shall both sides of the equation and obtain

$$\int_a^b \left(f_t + [H(f)]_x\right)dx = 0$$

$$\Downarrow$$

$$\int_a^b f_t dx + \int_a^b [H(f)]_x dx = 0$$

$$\Downarrow$$

$$\frac{d}{dt}\int_a^b f dx = H(f(a)) - H(f(b))$$

which explains the reason why this is a conservation law. In fact there is a balance between the change of f in the interval and the flux $H(f)$ of material into the interval therefore f is conserved as illustrated below

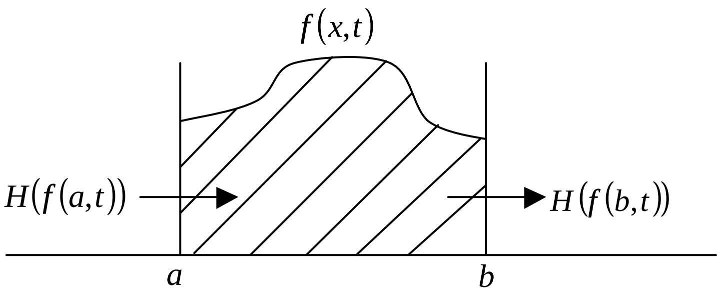

Figure 16 Flux of H in an interval

A simple example of a hyperbolic conservation law is the propagating graph curve given in the previous chapter. As a reminder, the equations of motions of a propagating curve that remains a graph for all time is given by

$$y_t = F\sqrt{1+y_x^2}\ .$$

In order to get a viscosity solution we shall use a speed F that includes a viscosity term e as $F = 1 - ek$.
Using differential geometry rules one can obtain that the curvature k is given by

$$k = \frac{-y_{xx}}{\left(1+y_x^2\right)^{3/2}}\ .$$

Substituting the curvature and the speed function in the motion equation gives

$$y_t = \left(1 + \frac{e y_{xx}}{\left(1+y_x^2\right)^{3/2}}\right)\sqrt{1+y_x^2}$$

$$\Downarrow$$

$$y_t - \sqrt{1+y_x^2} = e\,\frac{y_{xx}}{1+y_x^2}$$

then differentiating with respect to x gives

$$\left(y_t - \sqrt{1+y_x^2}\right)_x = \left(e\,\frac{y_{xx}}{1+y_x^2}\right)_x$$

$$\Downarrow$$

$$(y_x)_t + \left(-\sqrt{1+(y_x)^2}\right)_x = e\left(\frac{(y_x)_x}{1+(y_x)^2}\right)_x$$

that shows a hyperbolic conservation law for the slope y_x with a viscosity term e .

Chapter 4
Numerical algorithms

The goal of this chapter *is t*o show that algorithms based on *direct parameterizations of the moving front face considerable difficulties.* This is because such algorithms *adhere to local properties of the solution, rather than the global structure. Conversel*y, imbedding the surface in a higher-dimensional function preserves the global properties of the motion. *In this setting, the resulting equations of motion can b*e accurately approximated using *numerical techniques borrowed from hyperbolic conservation laws.*

4.1 Marker methods

One of the techniques to solve propagating interfaces equations is marker methods. They divide the evolving curve into a finite number of points equally spaced as shown in the following figure

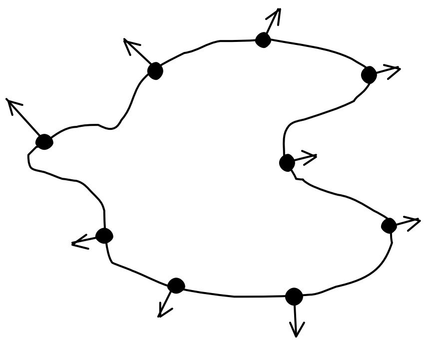

Figure 17 Marker points discretization of curve

We shall the denote the n^{th} time interval by $n\Delta t$, and the i_{th} parameterization interval by $i\Delta s$. The x value of the time interval n and the parameterization interval s will be noted by x_i^n.

From Taylor series we have

$$x_{i+1}^n \cong x_i^n + \frac{dx_i^n}{ds}\frac{\Delta s}{1!} + \frac{d^2x_i^n}{ds^2}\frac{\Delta s^2}{2!} + \frac{d^3x_i^n}{ds^3}\frac{\Delta s^3}{3!} + \cdots$$
$$x_{i-1}^n \cong x_i^n + \frac{dx_i^n}{ds}\frac{(-\Delta s)}{1!} + \frac{d^2x_i^n}{ds^2}\frac{(-\Delta s)^2}{2!} + \frac{d^3x_i^n}{ds^3}\frac{(-\Delta s)^3}{3!} + \cdots$$

by taking the difference between the equations and neglecting values of $O(\Delta s^3)$ one can obtain

$$x_{i+1}^n - x_{i-1}^n \approx 2\frac{dx_i^n}{ds}\frac{\Delta s}{1!}$$
$$\Downarrow$$
$$\frac{dx_i^n}{ds} \approx \frac{x_{i+1}^n - x_{i-1}^n}{2\Delta s}$$

while summing the two equations and neglecting values of $O(\Delta s^4)$ gives

$$x_{i+1}^n + x_{i-1}^n \approx 2x_i^n + 2\frac{d^2x_i^n}{ds^2}\frac{\Delta s^2}{2!}$$
$$\Downarrow$$
$$\frac{d^2x_i^n}{ds^2} \approx \frac{x_{i+1}^n - 2x_i^n + x_{i-1}^n}{\Delta s^2}$$

The same notations are equivalent to any space variable (i.e. y and z). The notation for the time derivative are similarly given by

$$\frac{dx_i^n}{dt} \approx \frac{x_i^{n+1} - x_i^n}{\Delta t}, \quad \frac{dy_i^n}{dt} \approx \frac{y_i^{n+1} - y_i^n}{\Delta t}$$

As shown before the curvature of the front at any point is given by

$$k = \frac{y_{ss}x_s - x_{ss}y_s}{\left(x_s^2 + y_s^2\right)^{3/2}}$$

Therefore substituting the first and second derivative approximations gives

$$k_i^n \approx \frac{\dfrac{y_{i+1}^n - 2y_i^n + y_{i-1}^n}{\Delta s^2} \cdot \dfrac{x_{i+1}^n - x_{i-1}^n}{2\Delta s} - \dfrac{x_{i+1}^n - 2x_i^n + x_{i-1}^n}{\Delta s^2} \cdot \dfrac{y_{i+1}^n - y_{i-1}^n}{2\Delta s}}{\left(\left(\dfrac{x_{i+1}^n - x_{i-1}^n}{2\Delta s}\right)^2 + \left(\dfrac{y_{i+1}^n - y_{i-1}^n}{2\Delta s}\right)^2\right)^{3/2}}$$

$$\Downarrow$$

$$k_i^n \approx 4\frac{\left(y_{i+1}^n - 2y_i^n + y_{i-1}^n\right)\cdot\left(x_{i+1}^n - x_{i-1}^n\right) - \left(x_{i+1}^n - 2x_i^n + x_{i-1}^n\right)\cdot\left(y_{i+1}^n - y_{i-1}^n\right)}{\left(\left(x_{i+1}^n - x_{i-1}^n\right)^2 + \left(y_{i+1}^n - y_{i-1}^n\right)^2\right)^{3/2}}$$

And finally the motion equations $(x_t, y_t) = F(k)\frac{(y_s, -x_s)}{\sqrt{x_s^2 + y_s^2}}$ are given by the approximation

$$\left(\frac{x_i^{n+1} - x_i^n}{\Delta t}, \frac{y_i^{n+1} - y_i^n}{\Delta t}\right) = F\left(k_i^n\right)\frac{\left(\dfrac{y_{i+1}^n - y_{i-1}^n}{2\Delta s}, -\dfrac{x_{i+1}^n - x_{i-1}^n}{2\Delta s}\right)}{\sqrt{\left(\dfrac{x_{i+1}^n - x_{i-1}^n}{2\Delta s}\right)^2 + \left(\dfrac{y_{i+1}^n - y_{i-1}^n}{2\Delta s}\right)^2}}$$

$$\Downarrow$$

$$\left(x_i^{n+1}, y_i^{n+1}\right) = \left(x_i^n, y_i^n\right) + \Delta t\, F\left(k_i^n\right)\frac{\left(y_{i+1}^n - y_{i-1}^n, -\left(x_{i+1}^n - x_{i-1}^n\right)\right)}{\sqrt{\left(x_{i+1}^n - x_{i-1}^n\right)^2 + \left(y_{i+1}^n - y_{i-1}^n\right)^2}}$$

where k_i^n estimate is given above.

This central difference approximation of the motion equations describes a well-defined procedure of updating by iterations the marker position in the next time step.

Given a speed evolution function $F = 1 - ek$ and a positive smoothing factor $e > 0$ there exist a small enough time step Δt that ensures a stable solution for the motions equations given in the Lagrangian approximation form above, therefore the markers techniques can be extremely accurate in the attempt to follow the motions of small perturbations, yet for large, complex motion, several problems soon occur.

First, marker particles come together in regions where the curvature of the propagating front builds, causing numerical instability unless a regridding technique is employed. The regridding mechanism usually contains a error term which resembles diffusion and dominates the real effects of curvature under analysis.

Another basic problem of the markers method is that no time step, no matter how small, can produce a scheme which correctly incorporates the entropy condition, which means that these methods suffer from topological problems; when two regions "burn" together to form a single one, ad-hoc techniques to eliminate parts of the boundary are required to make the algorithm work. This drawback comes from the fact that markers must somehow be removed from the discretization as corners form which is a laborious task for higher-dimensional interface problems.

In order to emphasize the weakness of the markers' methods we shall study an example of a propagating interface in two dimensions. For the simplicity of the discussion let us restrain ourselves to a special case of an emanating interface, a propagating graph with unit speed.

The associated differential equation of a propagating graph discussed in a previous chapter is inherently simpler then the motion equations given above and is given by

$$y_t = \sqrt{1 + y_x^2}\ .$$

Using the central difference approximations used in the formulation for the markers method described above we have the following iterative algorithm

$$y_i^{n+1} = y_i^n + \Delta t \sqrt{1 + \left(\frac{y_{i+1}^n - y_{i-1}^n}{2\Delta x} \right)^2}\ .$$

Lets analyze the method on a simple numerical example of an initial curve given as

$$y\big|_{t=0} = |x|, \quad x = -3,-2,-1,0,1,2,3$$

and shown in the following figure

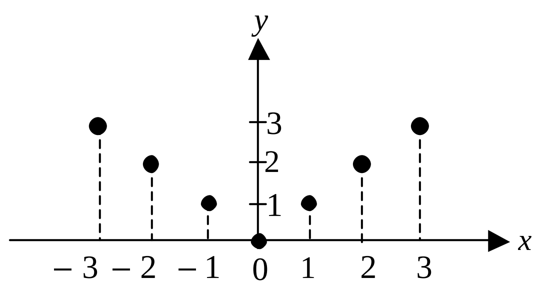

Figure 18 Initial curve for propagating graph with unit speed

We shall use a unit time step, i.e. $\Delta t = 1$. One question rather obvious is what should be done with the points in the edges of the curve (i.e. t=-3,3) having only one neighbor marker while the numerical formulation requires two? A possible answer will be to use a one side derivative approximation only for the edge points which will require only one neighbor, i.e.

$$y_{-3}^{n+1} = y_{-3}^{n} + \Delta t \sqrt{1+\left(\frac{y_{-2}^{n} - y_{-3}^{n}}{\Delta x}\right)^2}, \qquad y_{3}^{n+1} = y_{3}^{n} + \Delta t \sqrt{1+\left(\frac{y_{3}^{n} - y_{2}^{n}}{\Delta x}\right)^2}.$$

The desired exact solution for the differential equation is parallel lines that move up each time step by $\sqrt{2}$, due to the fact that the differential equation results in the following form

$$y_i^{n+1} = y_i^n + 1 \cdot \sqrt{1+\left(\frac{2}{2\cdot 1}\right)^2} = y_i^n + \sqrt{2}$$

The desired exact solution is shown in the following figure

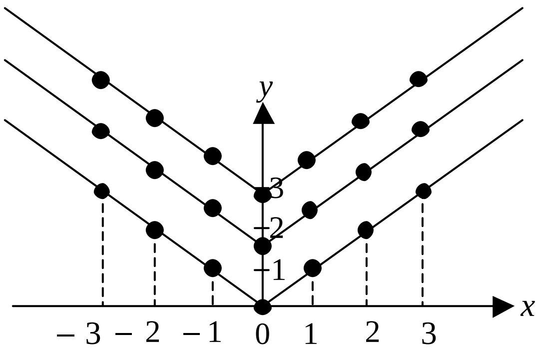

Figure 19 Desired solution

As can be seen from Figure 19 the shock with all markers should moves up in parallel due to the fact that the slope of the curve is –1 from the left of origin and +1 from the right of origin.

By simulating the central differences scheme for the two more time steps gives the following result shown in Figure 20.

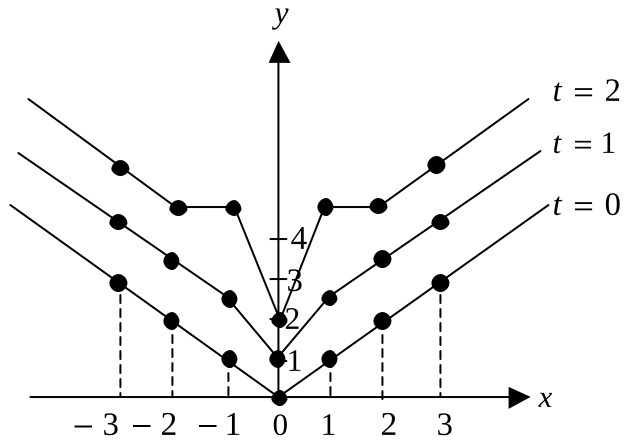

Figure 20 Solution using central differences

Unfortunately the central difference approximation chooses a wrong limiting solution because of the derivative approximation of y_x at $x = 0$. For all points $x \neq 0$ the approximation solves the differential approximation $y_t = \sqrt{2}$ except from $x = 0$ where the approximation is $y_t = 1$, which causes the middle point $x = 0$ to move up slower than the other points. This miscalculation of the slope propagates outward from $x = 0$ causing the code to blowup.

This exercise could be tried with different time steps, nonetheless no matter how small we take the numerical parameters, as long as we use an odd number of evenly spaced markers the approximation will not improve.

If by some way we can decide for each point to take the left or the right derivative according to a well defined criteria then a better approximation of the derivative could be obtained and the solution will remain stable at all time.

Let's note the left derivative approximation of y by

$$D_- y_i^n \equiv \frac{y_i^n - y_{i-1}^n}{\Delta x},$$

while the right derivative approximation shall be noted by

$$D_+ y_i^n \equiv \frac{y_{i+1}^n - y_i^n}{\Delta x}.$$

We shall also define the function g as follows

$$g(a_i, a_{i+1}) \equiv \sqrt{1 + \min(a_i, 0)^2 + \max(a_{i+1}, 0)^2}$$

Consider the following scheme

$$y_i^{n+1} = y_i^n + \Delta t \cdot g\left(D_- y_i^n, D_+ y_i^n\right)$$

or in a more detailed form as

$$y_i^{n+1} = y_i^n + \Delta t \sqrt{1 + \min\left(\frac{y_i^n - y_{i-1}^n}{\Delta x}, 0\right)^2 + \max\left(\frac{y_{i+1}^n - y_i^n}{\Delta x}, 0\right)^2} \; .$$

It can be shown that this scheme has no spurious overshoots nor wiggles near discontinuities and it obeys an entropy condition for limit solutions while miraculously giving a very close approximation to the exact solution.

Furthermore, the more we refine the distance between the markers Δx the sharper the corner remains. This numerical scheme is said to be in conservation form while the function g is called a numerical flux function that will be discussed in details later.

4.2 Cell method

A different technique that satisfies the entropy condition and provides the correct weak solution is the cell method. This technique divides the space into cells and assigns each cell a value, which is the fraction of amount of material inside the interface.

At any iteration of the algorithm the propagating interface passing through some cells splits them into two parts: a fraction of the cell which is inside the curve and a fraction which is outside. Each time step the interface propagates according to a speed function and the cells are updated with new values corresponding to the fraction of the cell inside the interface.

An illustration of an interface evolution with some speed function in a diagonal direction is shown below

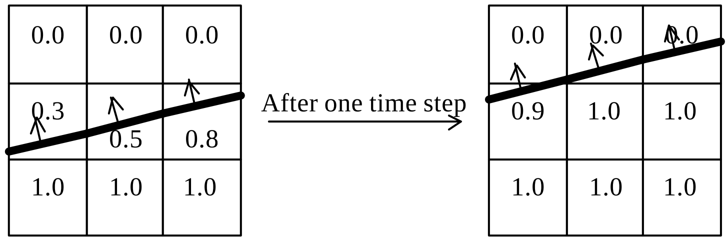

Figure 21 Cell method front evolution of one time step

One advantage of the cell method over the markers method is that curves merging are handled naturally while there is no need to eliminate markers that are a result of a merge between to propagating interfaces.

Yet, the cell technique is inaccurate since there is lose of precision regarding the exact shape of the interface lines crossing a cell due to the fact that only a single fraction number expresses the position of the curve. This inaccuracy propagates to other calculations like curvature and normal direction, which are completely undesired.

This drawbacks gives motivation to use more sophisticated techniques that could incorporate the entropy condition from the weak solutions and could be naturally extended to higher dimensions that will give accurate calculations of curvature with relatively less work.

4.3 Level set numerical algorithm

As mentioned before the level set function $f_t + F|\nabla f| = 0$ or the boundary value formulation $F|\nabla T| = 1$ is a special case of a more general Hamilton-Jacobi family of differential equations $a f_t + H(Df) = 0$.

The idea of this section is to show the formulation of the correct equation of motion for a front propagating with curvature-dependent speed.

This equation in an initial-value Hamilton-Jacobi equation with right-hand-side that depends on curvature effects. By viewing the surface as a level set, topological complexities and changes in the moving front are handled naturally. With these equations as a basis, any number of numerical algorithms may be formulated of arbitrary degree of accuracy, using the technology developed for the solution of hyperbolic conservation laws.

In particular, algorithms can be devised to have the correct limiting entropy-satisfying solution. In fact, the previous algorithms as the markers and cells methods may be viewed as less sophisticated approximations to our equations of motion.

So the aim is to find a basic scheme for the level set formulation that would choose the correct weak solution entropy satisfying corresponding to the viscous limit. We shall see that there are specific numerical schemes that under certain conditions (monotonicity, conservation etc.) give the correct entropy condition.

4.3.1 Numerical domain of dependence

One of the most important issues in the for stability of a scheme, that is, what it does to small errors in the initial data is the numerical domain of dependence. This issue could be well understood from a simple example of the wave equation

$$y_t + y_x = 0, \quad y(x,0) = f(x).$$

Solving this equation gives a solution which is a function $f(x)$ moving in the x direction with speed $F = 1$ i.e.

$$y(x,t) = f(x-t).$$

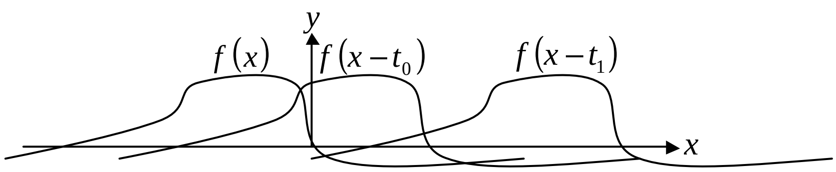

Figure 22 Wave equation evolution

The characteristic lines given by the solution of

$$\begin{aligned} \frac{dt}{ds} &= 1 \quad \Rightarrow \quad t = s + c_1 \\ \frac{dx}{ds} &= 1 \quad \Rightarrow \quad x = s + c_2 \qquad \begin{array}{l} \Rightarrow \quad t - x = const \\ \Rightarrow \quad y = const \end{array} \\ \frac{dy}{ds} &= 0 \quad \Rightarrow \quad y = c_3 \end{aligned}$$

and shown in the $t-x$ plane

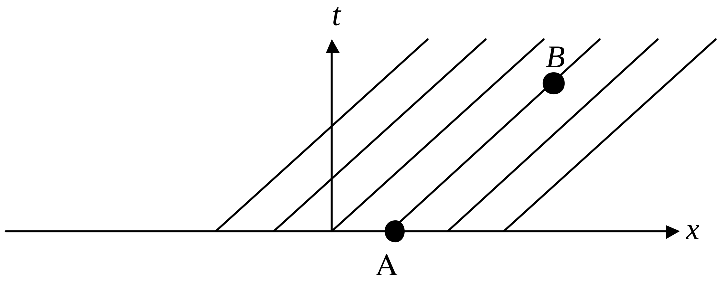

Figure 23 Characteristic curves in t-x plane

The resulting wave solution moves from left to right, therefore A is the *domain of dependence* of B.

While constructing a numerical scheme it is very important to identify the domain of dependence in order to find the correct spatial derivative. The numerical solution for the wave equation is one of the following first order approximations

$$\textit{Forward scheme}: \quad y_i^{n+1} = y_i^n - \Delta t \cdot \frac{y_{i+1}^n - y_i^n}{\Delta x}$$

$$\textit{Backward scheme}: \quad y_i^{n+1} = y_i^n - \Delta t \cdot \frac{y_i^n - y_{i-1}^n}{\Delta x}$$

$$\textit{Centered scheme}: \quad y_i^{n+1} = y_i^n - \Delta t \cdot \frac{y_{i+1}^n - y_{i-1}^n}{2\Delta x}$$

The backward scheme is the most appealing due to the fact that the wave moves to the right and thus the information at point i should be determined by the information at point i and $i-1$. In order words if the wave propagates to the right then so should the information of the derivative approximation and vice versa. This is also called an *upwinding scheme*, which means that for stability of the solution in any given scheme the numerical domain of dependence should contain the mathematical domain of dependence.

Lets consider another problem that will emphasize the use of domain of dependence, the non-constant speed wave equation:

$$y_t + a(x) y_x = 0.$$

The function $a(x)$ is in fact the wave speed because the characteristics equations state that $\frac{dx}{dt} = a(x)$. Therefore the domain of dependence will now depend on the sign of $a(x)$ which could be settled nicely in the following scheme:

$$y_i^{n+1} = y_i^n - \Delta t\left[\min(0,a_i)\frac{y_{i+1}^n - y_i^n}{\Delta x} + \max(0,a_i)\frac{y_i^n - y_{i-1}^n}{\Delta x}\right]$$

If $a(x)$ is positive then the backward scheme is used while when negative the forward is used.

Another example is the *Burgers'* equation given as

$$y_t + yy_x = 0.$$

This time the speed depends directly on the value of y. By analogy to the last wave equation we can use a similar scheme as follows

$$y_i^{n+1} = y_i^n - \Delta t\left[\min(0,y_i^n)\frac{y_{i+1}^n - y_i^n}{\Delta x} + \max(0,y_i^n)\frac{y_i^n - y_{i-1}^n}{\Delta x}\right].$$

The common factor in all last examples of wave equations scheme is the fact that they could all be written in a conservation law, the same as a first dimension level set differential equation could be written.

4.3.2 1D Level set scheme

As defined before the conservation law form is given by

$$f_t + [G(f)]_x = 0.$$

The one dimension level set equation for the initial value problem is a conservation form with $G(f) \equiv f$.

An integration of the conservation law with respect to space gives

$$\frac{d}{dt}\int_a^b f dx = G(f(a,t)) - G(f(b,t)).$$

Armed with the knowledge of upwinding schemes the following scheme can be constructed

$$f_i^{n+1} = f_i^n + \frac{\Delta t}{\Delta x}\left(G_{i-1/2} - G_{i+1/2}\right),$$

$$when \begin{array}{l} G_{i-1/2} \equiv g(f_{i-1}, f_i) \\ G_{i+1/2} \equiv g(f_i, f_{i+1}) \end{array}$$

Therefore

$$f_i^{n+1} = f_i^n + \frac{\Delta t}{\Delta x}\left(g(f_{i-1}, f_i) - g(f_i, f_{i+1})\right).$$

This scheme is called a conservation form and the function g is referred to as a numerical flux function that is subject to the consistency requirement $g(f,f)=G(f)$.

It can be shown that if the conservation form above is a monotone scheme, i.e. f_i^{n+1} is an increasing function of $f_{i-1}^n, f_i^n, f_{i+1}^n$, then it satisfies the entropy condition.

This statement is extremely important for our purpose due to the fact that the only thing to be done in order to obtain the desired entropy solution is to make sure that out scheme is monotone function of all its arguments.

There are a large variety of numerical flux functions, and some of them have been seen in the last section. For example for the non-linear wave equation $y_t + \left[y^2\right]_x = 0$ the numerical flux function is given by

$$g(y_1, y_2) = \left(\max(y_1,0)^2 + \min(y_2,0)^2\right).$$

This equation could be written as

$$y_t + [G(y)]_x = 0, \quad G(y) \equiv y^2$$

while performing an analysis over the characteristics curves shows that

$$y_t + \frac{dG(y)}{dy} y_x = 0 \Rightarrow wave\,speed \equiv \frac{dx}{dt} = \frac{dG(y)}{dy} = 2 \cdot y$$

which gives that the wave speed is proportional to y therefore the numerical flux function associated with this wave equation chooses the derivative direction that corresponds to the wave direction. Thus for the one dimensional unit speed level set function with $G(y) \equiv \sqrt{y^2}$ we shall use a numerical flux function that is given by

$$g(y_1, y_2) = \sqrt{\max(y_1,0)^2 + \min(y_2,0)^2}\ .$$

The numerical flux function is a good approximate for the $G(y)$ function thus the level set function could be approximated by the following scheme

$$y_i^{n+1} = y_i^n - \Delta t\, g\left(\frac{y_i^n - y_{i-1}^n}{\Delta x}, \frac{y_{i+1}^n - y_i^n}{\Delta x}\right).$$

4.3.3 3D level set scheme

The 1D level-set scheme could be naturally extended to a multi-dimensional of the level set function, i.e.

$$f_t + F\sqrt{f_x^2 + f_y^2 + f_z^2} = 0$$

by the following scheme

$$f_{ijk}^{n+1} = f_{ijk}^{n} - \frac{\Delta t}{\Delta x}\left[\max\left(F_{ijk,0}\right)\nabla^{+} + \min\left(F_{ijk,0}\right)\nabla^{-}\right]$$

where,

$$\begin{aligned}\nabla^{+} \equiv [&\max\left(f_{ijk}^{n} - f_{i-1\,jk}^{n},0\right)^2 + \min\left(f_{i+1\,jk}^{n} - f_{ijk}^{n},0\right)^2 + \\ &\max\left(f_{ijk}^{n} - f_{ij-1k}^{n},0\right)^2 + \min\left(f_{ij+1k}^{n} - f_{ijk}^{n},0\right)^2 + \\ &\max\left(f_{ijk}^{n} - f_{ijk-1}^{n},0\right)^2 + \min\left(f_{ijk+1}^{n} - f_{ijk}^{n},0\right)^2]^{1/2} \\ \nabla^{-} \equiv [&\min\left(f_{ijk}^{n} - f_{i-1\,jk}^{n},0\right)^2 + \max\left(f_{i+1\,jk}^{n} - f_{ijk}^{n},0\right)^2 + \\ &\min\left(f_{ijk}^{n} - f_{ij-1k}^{n},0\right)^2 + \max\left(f_{ij+1k}^{n} - f_{ijk}^{n},0\right)^2 + \\ &\min\left(f_{ijk}^{n} - f_{ijk-1}^{n},0\right)^2 + \max\left(f_{ijk+1}^{n} - f_{ijk}^{n},0\right)^2]^{1/2}\end{aligned}$$

4.4 Speed function calculation

A very important stage in image segmentation using the level set method is to find a way to stop the growing of a front at the boundary of the desired object, in our case the Multiple Sclerosis lesions. There are two key components of this technique: first, the ability of our algorithm to naturally execute topological change, and second, a stopping criterion built from the image gradient.

Handling of topological changes is extremely important because in many cases the raw magnetic resonance images contains noise in the form of pixels whose values drift quite far from those of their neighbors. The goal is to create a speed function that could ignore the noise spots and focus on the real boundary of the lesions. Small isolated spots of noise where the image gradient changes substantially should be ignored. The desired behavior is a front propagating around these points, splitting into two fronts, while the ring around the isolated spots closes back in on itself and then disappears. The desired propagation of the initial front is illustrated in a simple example of a two-dimensional propagating front shown in Figure 24, that is supposed to recover the boundary of the large circle while ignore the small noise spots.

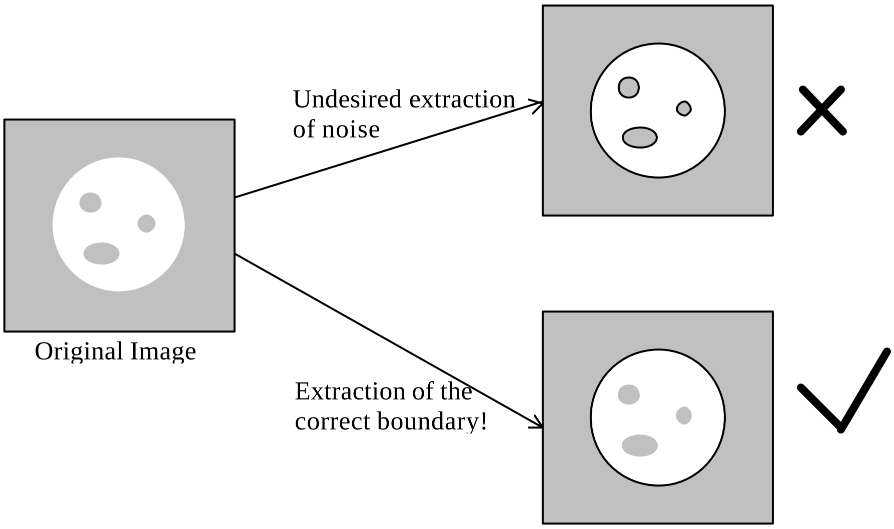

Figure 24 Expanding front stop criteria

The desired speed function would be large far from the boundary and drop to zero near sharp changes in the image gradient. Defining the image brightness function by $I(x, y)$ and some constants C and K, then the following speed function answers the above criteria

$$F_{\exp} = \frac{K}{C + \left|\nabla\left(G_s * I(x, y)\right)\right|}.$$

The G_s term is a Gaussian smoothing filter while in the expression it is convoluted with the image brightness function $I(x, y)$. It can be easily seen that the term $\nabla\left(G_s * I(x, y)\right)$ gets large values close to edges while small values far from edges as desired.

As described before it is a good practice to add a smoothing factor to the speed function that is a function of the curvature, which will choose the entropy solution and prevent the solution from creating corners that could produce singularities. Thus putting it into mathematical formulation gives

$$F_{curv} = F_{\exp}(1 - ek) = \frac{K(1 - ek)}{C + \left|\nabla\left(G_s * I(x, y)\right)\right|}.$$

This force function could be used in the level set numerical scheme described in the previous section.

Another important issue is numerical stability. The level set scheme is restricted to a certain condition for stability called the Courant-Friedrichs-Lewy (or CFL) stability criterion. It requires the front not to cross more than one grid cell each time step, i.e.

$$\max F_{ijk}\Delta t \le \Delta x \quad \forall i, j, k .$$

In practice one shall scan all possible values of the speed and choose the appropriate time step that obeys the CFL condition.

4.5 The Narrow Band method

As further step towards implementation it is a good practice to optimize the numerical scheme for computation speed. The idea that instead of calculating the whole distance function (i.e. the level set function f) each iteration of the algorithm, one needs only to calculate values inside a narrow band close to the zero level set.

The narrow band method has numerous advantages which the most important of all is gain in calculation speed. The calculation in three dimensions is performed in the order of $O(kN^2)$ instead of the order of $O(N^3)$ in the full matrix approach, where N is the number of grid points and k is the number of voxels in the narrow band. Furthermore the narrow band approach speeds up the calculation of the speed function and other parameters needed for the iterative scheme due to the fact that they need to be calculated only near the narrow band. Moreover, according to the CFL condition the time step is a function of the speed, therefore the less values of the speed F to scan the higher the time step Δt could be, which allows a faster scheme.

The narrow band method steps consist of choosing a band width of k points and indicating the boundaries of the narrow band, then solving the level set equation till the boundary is hit and then recalculate a new narrow band. An important issue is the choice of the narrow band width k . If chosen too big then it resembles the full matrix method while if chosen too small it can cause a lot of overhead in re-initializing the distance function. Therefore a compromise should be chosen in order to receive optimal speed. In practice a good narrow band width is $k = 6$ which is a good balance between re-initialization costs and update costs.

In terms of scheme complexity of the level set used with the enhancement of the narrow band method gives a equivalent level of complexity to the marker method and cell techniques discussed before while maintaining the advantages of topological merger and easy extension to high dimensions. Typically the narrow band method is ten times faster than the full matrix method which gives it a tremendous advantage.

Chapter 5
Software application

While a skilled eye can pick out the desired boundaries of multiple sclerosis lesions from a noisy image, even those characterized by slight changes in image intensity, asking a physician to draw by hand outlines on each image is both time-consuming and inaccurate. An automatic procedure could be much more efficient. This chapter gives a view of the software application used to perform level set segmentation of multiple sclerosis lesions using 3D magnetic resonance brain images.

5.1 Algorithm

The approach taken is an implementation of the numerical level set scheme with curvature driven speed. An initial circle is created inside the desired multiple sclerosis lesion. This circle evolves using the level set numerical scheme till the narrow band is hit, then the distance function f is recalculated for a new narrow band and then other iterations of the scheme is proceeded till hit of the new narrow band, and so on and so forth.

The algorithm is show in the following schematic illustration

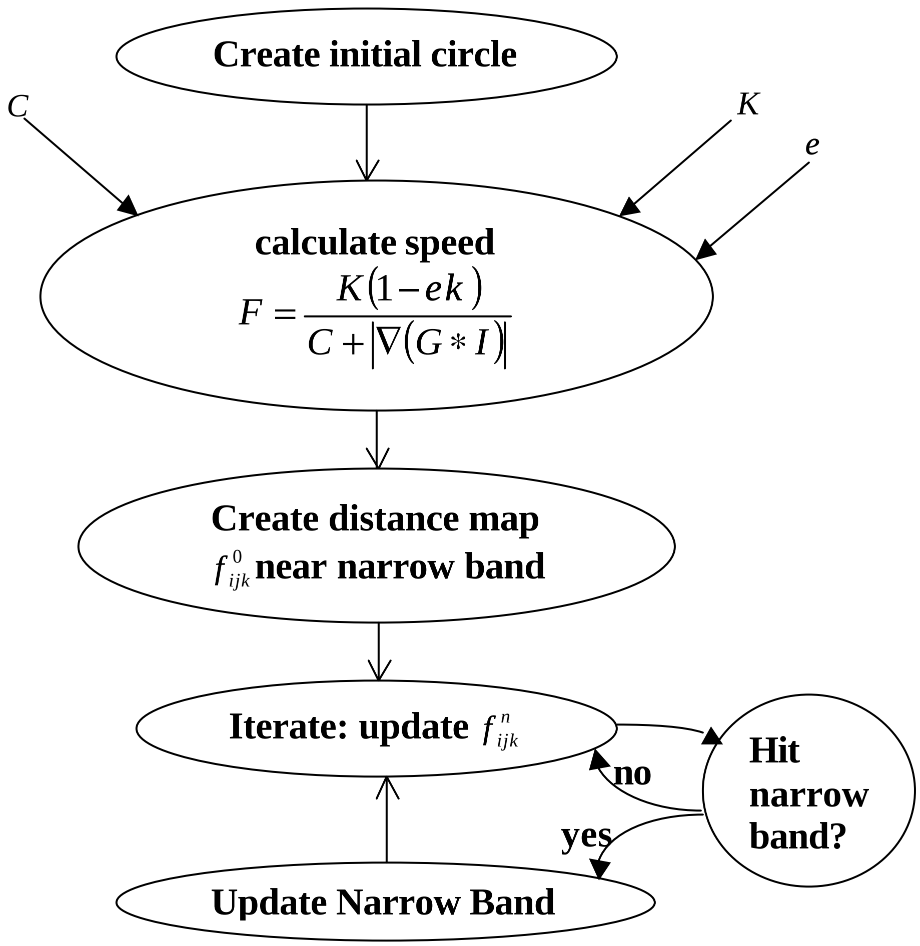

Figure 25 level set algorithm

5.2 Level set scheme

The level set scheme used in the code is the following

$$\phi_{ijk}^{n+1} = \phi_{ijk}^{n} - \frac{\Delta t}{\Delta x}\left[\max\left(F_{ijk},0\right)\nabla^{+} + \min\left(F_{ijk},0\right)\nabla^{-}\right],$$

where,

$$\begin{aligned}\nabla^{+} \equiv [&\max\left(\phi_{ijk}^{n} - \phi_{i-1jk}^{n},0\right)^2 + \min\left(\phi_{i+1jk}^{n} - \phi_{ijk}^{n},0\right)^2 + \\ &\max\left(\phi_{ijk}^{n} - \phi_{ij-1k}^{n},0\right)^2 + \min\left(\phi_{ij+1k}^{n} - \phi_{ijk}^{n},0\right)^2 + \\ &\max\left(\phi_{ijk}^{n} - \phi_{ijk-1}^{n},0\right)^2 + \min\left(\phi_{ijk+1}^{n} - \phi_{ijk}^{n},0\right)^2]^{1/2}\end{aligned}$$

$$\begin{aligned}\nabla^{-} \equiv [&\min\left(\phi_{ijk}^{n} - \phi_{i-1jk}^{n},0\right)^2 + \max\left(\phi_{i+1jk}^{n} - \phi_{ijk}^{n},0\right)^2 + \\ &\min\left(\phi_{ijk}^{n} - \phi_{ij-1k}^{n},0\right)^2 + \max\left(\phi_{ij+1k}^{n} - \phi_{ijk}^{n},0\right)^2 + \\ &\min\left(\phi_{ijk}^{n} - \phi_{ijk-1}^{n},0\right)^2 + \max\left(\phi_{ijk+1}^{n} - \phi_{ijk}^{n},0\right)^2]^{1/2}\end{aligned}$$

while the speed function F_{ijk} is calculated as follows

$$F_{ijk} = \frac{K\left(1-\epsilon k_{ijk}\right)}{C + \left|\nabla\left(G_{ijk} * I_{ijk}\right)\right|}.$$

The time step should obey the CFL condition, therefore

$$\Delta t = \frac{\max F_{ijk}}{\Delta x},$$

while the step Δx is chosen as a parameter.

The only parameter left to calculate is the mean curvature k, which is calculated in 3D as follows

$$k = \frac{\left(\phi_{yy} + \phi_{zz}\right)\phi_x^2 + \left(\phi_{xx} + \phi_{zz}\right)\phi_y^2 + \left(\phi_{xx} + \phi_{yy}\right)\phi_z^2 - 2\left(\phi_x\phi_y\phi_{xy} + \phi_x\phi_z\phi_{xz} + \phi_y\phi_z\phi_{yz}\right)}{\left(\phi_x^2 + \phi_y^2 + \phi_z^2\right)^{3/2}}$$

while the derivative are given as follows:

- First order derivatives:

$$\phi_x \equiv \frac{\phi_{i+1,j,k}^{n} - \phi_{i-1,j,k}^{n}}{2\Delta x}, \phi_y \equiv \frac{\phi_{i,j+1,k}^{n} - \phi_{i,j-1,k}^{n}}{2\Delta y}, \phi_z \equiv \frac{\phi_{i,j,k+1}^{n} - \phi_{i,j,k-1}^{n}}{2\Delta z}$$

- Second order derivatives:

$$\phi_{xx} \equiv \frac{\phi_{i+1,j,k}^{n} - 2\phi_{i,j,k}^{n} + \phi_{i-1,j,k}^{n}}{2\Delta x}, \phi_{yy} \equiv \frac{\phi_{i,j+1,k}^{n} - 2\phi_{i,j,k}^{n} + \phi_{i,j-1,k}^{n}}{2\Delta y}, \phi_{zz} \equiv \frac{\phi_{i,j,k+1}^{n} - 2\phi_{i,j,k}^{n} + \phi_{i,j,k-1}^{n}}{2\Delta z}$$

- Mixed derivatives:

$$f_{xy} \equiv \frac{f^n_{i+1,j+1,k} - f^n_{i+1,j-1,k} - f^n_{i-1,j+1,k} + f^n_{i-1,j-1,k}}{4\Delta x \Delta y},$$

$$f_{xz} \equiv \frac{f^n_{i+1,j,k+1} - f^n_{i+1,j,k-1} - f^n_{i-1,j,k+1} + f^n_{i-1,j,k-1}}{4\Delta x \Delta z},$$

$$f_{yz} \equiv \frac{f^n_{i,j+1,k+1} - f^n_{i,j+1,k-1} - f^n_{i,j-1,k+1} + f^n_{i,j-1,k-1}}{4\Delta y \Delta z}$$

All the time steps $\Delta x, \Delta y, \Delta z$ have the same value and are configurable parameters as mentioned above.

Conclusions

In many physical problems, a key aspect is the motion of a propagating front separating two components. As fundamental as this may be, the development of a numerical algorithm to accurately track the moving front is a difficult problem. In particular tracking multiple sclerosis lesions from magnetic resonance brain images incorporates intensity variations due to RF inhomogenities, motion artifacts and noise that render this problem an arduous task. Educating a piece of software to both ignore noise and avoid introducing non-existent boundaries is quite a challenge.

Some other algorithms of image segmentation are shown like placing marker particles along the front and advancing the position of the particles according to a set of finite difference approximations to the equation of motion. While a different approach divides space into cells and tracks a moving front by updating the cells with new fractions of material inside the interface, which reflects the progress of the front. These schemes are inaccurate and usually go unstable causing the code to blow up as the curvature builds around an edge, since small errors in the position produce large errors in the determination of the curvature.

This work details the level set method and the problems involved in constructing a solution to track the multiple sclerosis lesions. The power of the level set approach lies on its ability to naturally extend to higher dimensions, track sharp corners and cusps, and handle topological changes of merger and breakage. Furthermore a particularly efficient approach has been described, namely the Narrow Band Level Set Method, which yields an extraordinary bust in calculation speed while maintaining the same advantages for interface tracking as the complete level set method.

For future research it is important to point out that far more sophisticated level set schemes exists than the one used in the software application. For example, the scheme used in the code is a first order space convex scheme, while another possibility will be to use a second order scheme that incorporates a switch that turns itself off whenever a shock is detected. Another possibility would be to add other forces to the level set equation that attracts the surface towards the boundary, which have a stabilizing effect, especially when there is a large variation in the image gradient value.

References

[1] Oprea, J., *Differential Geometry and its Applications*, Cleveland State University, Prentice Hall, 1997.

[2] Sethian, J.A., *Level Set Methods and Fast Marching Methods*, Department of Mathematics, University of California, Berkeley, 1999.

[3] Osher, S., and Sethian, J.A., *Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations,* Journal of Computational Physics, 79, pp.12-49, 1988.

[4] Sethian, J.A., *Curvature and the Evolution of Fronts*, Comm. in Math. Phys., 101, pp. 487-499, 1985.

[5] Sethian, J.A., *Numerical Methods for Propagating Fronts*, in Variational Methods for Free Surface Interfaces, Eds. P. Concus and R. Finn, Springer-Verlag, NY, 1987

[6] Sethian, J.A., *A Review of Recent Numerical Algorithms for Hypersurfaces Moving with Curvature-Dependent Speed*, Journal of Differential Geometry, 31, pp. 131-161, 1990.

[7] Adalsteinsson, D., and Sethian, J.A., *A Fast Level Set Method for Propagating Interfaces,* J. Comp. Physics, 118, pp.269-277, 1995.

[8] Sethian, J.A., *A Fast Marching Level Set Method for Monotonically Advancing Fronts,* Proc. Nat. Acad. Sci., 93, 4, pp.1591-1595, 1996.

[9] Malladi, R., and Sethian, J.A., *Image Processing: Flows under Min/Max Curvature and Mean Curvature,* Graphical Models and Image Processing, 58,2, pp. 127-141, 1996.

[10] Malladi, R., and Sethian, J.A., *Level Set Methods for Curvature Flow, Image Enhancement, and Shape Recovery in Medical Images*, Proc. of Conf. on Visualization and Mathematics, June, 1995, Berlin, Germany, Springer-Verlag, Heidelberg, Germany, pp. 329-345, 1997.

[11] Sethian, J.A., *Tracking Interfaces with Level Sets*, American Scientist, pp. 254-263, May-June, 1997.

[12] Sethian, J.A., *Fast Marching Methods and Level Set Methods for Propagating Interfaces*, von Karman Institute Lecture Series, Computational Fluid Mechanics, 1998.

[13] Sethian, J.A., *Fast Marching Methods*, SIAM Review, 41,2, 1999, pp. 199-235.

[14] Levandosky, J., lecture notes on *Partial Differential Equations of Applied Mathematics* course, MATH 220A, School of Engineering, Institute of Computational and Mathematical Engineering, Stanford University, Fall 2002.

Table of Figures

Figure 1 Multiple sclerosis lesions.. 6
Figure 2 initial front moving in normal direction with speed F = 1 8
Figure 3 propagating graph parameters 8
Figure 4 moving front in normal direction with speed F = 1 after one second...................... 9
Figure 5 Front motion in higher dimension gives the boundary value problem.................. 10
Figure 6 The level set function and the initial curve that lies in xy plane 11
Figure 7 Evolution of the level set function with respect to time 12
Figure 8 Evolution of two cones of the level set function while the initial front merges .. 13
Figure 9 curve parameterization by arclength s 15
Figure 10 Three-vector frames 15
Figure 11 Motion with speed proportional to the curvature.. 17
Figure 12 Propagating curve... 20
Figure 13 Initial cosine front moving in the normal direction... 21
Figure 14 Possible Solution to the moving front 22
Figure 15 Entropy solution of cosine front moving in the normal direction with F = 1 23
Figure 16 Flux of H in an interval 25
Figure 17 Marker points discretization of curve 27
Figure 18 Initial curve for propagating graph with unit speed... 31
Figure 19 Desired solution... 31
Figure 20 Solution using central differences 32
Figure 21 Cell method front evolution of one time step.. 34
Figure 22 Wave equation evolution... 35
Figure 23 Characteristic curves in t-x plane 36
Figure 24 Expanding front stop criteria 40
Figure 25 level set algorithm.. 44