University of Reading
School of Mathematical and Physical Sciences

The Closest Point Method

theory and applications

Tudor Ciochina

August 2011

This dissertation is submitted to the Department of Mathematics and Statistics in partial fulfillment of the requirements for the degree of Master of Science

Corrigendum

An error in that the signed distance function for the ellipse has since been discovered, with the following consequences:

- the property "the extension is constant along the normal direction" is NOT kept

- the results obtained by using representation (3) from the article do NOT possess the property of the closest point method

Declaration

I confirm that this is my own work, and the use of all material from other sources has been properly and fully acknowledged.

Tudor Ciochina

Abstract

Partial differential equations posed on a surface (or more generally a manifold) arise in a natural way in fields ranging from continuum mechanics to the more recent computational neuroscience. In most cases (if not all) the analytic approach fails due to the high degree of complexity of the differential equation or the underlying physical or mathematical phenomena. Therefore the computational approach is chosen. One of the traditional numerical methods is the level set method introduced by Osher and Sethian in 1988 which in turn gave rise to the closest point method of Ruuth and Merriman in 2008. The closest point method is also called an embedding method since the quantities defined only on the surface (intrinsic quantities) are embedded into the surrounding domain. In this thesis the closest point method is explained and applied to relevant examples.

Acknowledgements

I would like to thank my supervisors Prof. Michael J. Baines and Dr. Peter K. Sweby for their guidance and support. I also would like to thank Dr. Colin Macdonald of the Oxford Centre for Collaborative Applied Mathematics (OCCAM) of Oxford University.

Contents

List of Acronyms

ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
MBE	Molecular Beam Epitaxy
CPM	Closest Point Method
TVD	Total Variation Diminishing
SSP	Strong Stability Preserving

List of Figures

1 Introduction

The partial differential equation (PDE) is central to pure mathematics but also to the applied sciences, ranging from the geodesic flow in geometry, the classical continuum mechanics [MC04, OM97, CGM$^+$98, GRM$^+$98] to the more recent computational neuroscience. Other fields include physics [GRM$^+$98], biology and even computer graphics [MCC02, WK91]. A notable example would be the prediction of pattern formation in an fMRI scan or the prediction of pattern formation of the coating of a mammal. In both cases the problem is nothing else than the evolution of one or more curves. Associated to every PDE is a domain, initial condition and a boundary condition. Without these ingredients, the solving of the PDE (analytical or numerical) would be impossible. The domain together with its boundary holds a special place, because in most cases it is obtained from the physical setting of the phenomena. If the physical phenomena exists (or it is wanted) only on the surface of the domain (boundary or more generally, a manifold), this in turn will become the domain of the PDE. Some examples include tensor equations posed on an elastic boundary (the modelling of blood vessels), medical imaging/image processing (removing the noise of fMRI scans), the visualization of vector fields, solidification, silica precipitation in the manufacturing of microprocessors (thin films), multi-phase problems [MBO94] and many more.

There are many techniques, either analytical or numerical, when dealing with PDEs posed on a surface. In order to introduce gradually the closest point method, a small review of the related methods must be made.

The first technique is to apply a parametrization [FH05] to the surface (for example switching to polar, cylindrical or parabolic cylindrical coordinates) and to transform the PDE accordingly. If the analytical approach fails, numerical techniques are applied. In most cases the technique of “patching” is applied to different portions of the parametrized surface in order to “connect” them properly. However, when dealing with real - life phenomena, the parametrization in most cases becomes difficult, impractical or impossible to construct.

The second technique is to apply a triangulation [BCOS01, OPB$^+$03, Gre06] (generation of polygons) of the surface and directly solve the PDE on this domain, by some grid generation method. This is a very widely used approach, but several important issues persist. While this gives satisfactory results, the geometric quantities, such as the surface normal or the curvature are not properly defined and the final result will be a very cumbersome numerical computation and the quantities defined through a projection operator [BCOS01] will pose difficulties.

The convergence aspect of the problem, related to the numerical methods, in both cases, is not well understood [Gre06]. The convergence of the numerical method, in a parametrization or triangulation, is less understood compared to the standard Cartesian space.

The third technique is the level set method. The surface PDE is "embedded" on the entire domain, rather than only on the surface.

The level set methods [OS88] provide the means, the level set function, for dealing with surfaces or more generally, geometrical objects of codimension two or higher. Embedding methods, which in turn use the level set method, posses several limitations. The first one would be that the embedding PDE is posed on entire domain. However, if it is required to build a narrow band around the surface (in order to minimize the number of points and therefore to optimize the method), appropriate boundary conditions must be imposed. These boundary conditions are dependent on the geometry and in general lead to the degradation of the solution.

Bertalmio [BCOS01] introduced a new approach for solving these types of problems by using an implicit representation of the surface. The representation is based on the zero level set of a specific function ϕ, and because of this, the surface PDE can be recast in "Eulerian" coordinates. The numerical computation of the intrinsic quantities is performed by embedding the surface into a higher dimensional space. The "Eulerian" (embedded) version of the PDE is derived by the use of a projection operator. This operator is applied to the derivatives from the embedding space, and in this way the surface derivatives (intrinsic quantities) are equivalent to the standard (embedded) derivatives. The discretisation is on the standard fixed Cartesian grid and the numerical method is based on finite differences. Also, by choosing this approach, parametrization and triangulations can be avoided. As a side note, triangulated surfaces can still be used by transforming them to implicit surfaces via polynomials [EH96]. The geometric quantities, such as the normal or the curvature can obtained directly from the derivatives of the function ϕ.

The surface deformation is modelled by a so - called embedding function deformation and for every curve a Lagrangian system is attached to it. By using the deformation function, the Lagrangian system is transformed to a Eulerian system. This generates the Euler - Lagrange equations related to the velocity of deformation of the curve will be used, hence the name of a level set variational problem. The level set variational problem is then defined (embedded) in the higher dimensional space by using the energy integral and the delta function of the theory of distributions.

The theory of harmonic maps [EDD$^+$95, EL78, EL88, TSC00, PW99] plays a fundamental role since the energy is first defined between two manifolds. The manifolds are then chosen to be for example the threedimensional Euclidean space and the first variation is then applied to the energy integral to obtain the desired form. The next step, after the variational problem defined on the surface is obtained, is to use the zero level set representation of the surface (of a higher dimensional function). The gradient descent is then computed and the Cartesian coordinate system of the higher dimensional space are used. The technique can be extended to a more general partial differentian equation by the use of the implicit representation. Since the surface is fixed in the higher dimensional embedding space, the effect of deformation is achieved by the actual values that are computed. Therefore, by extending back to the surface the values at the grid points of the entire domain, the evolution of the deformation is accomplished.

Although the Eulerian approach is very flexible and solves a number of important problems it does not perform well regarding to the diffusion equation posed on surfaces. The PDE is embedded into a domain one dimension larger than the surface, i.e. a one-dimensional surface PDE will become a two-dimension PDE defined on the entire domain. In an attempt to compensate for the increase in dimension a narrow band around the surface is typically chosen and boundary conditions must be applied. These boundary conditions do not affect the analytical solution but will influence the numerical solution at later times. Therefore, the details regarding this phenomena remain unclear, and in practice a strategy based on trial - and - error is typically applied. Numerical experiments using Dirichlet boundary conditions showed that the numerical solution at the boundary possesses a jump and will lead to large errors at later times. The homogenous Neumann boundary condition has also been applied and the numerical solution also possesses a jump and will lead to large errors.

Together with the embedding of the surface PDE, the initial condition must also be embedded. The process is called the extension of the initial condition to the embedding space. Since the finite difference method is used, and the actual finite difference equations use the values at grid points, the extension process must be performed in a specific way. The extension is done by imposing that the initial condition must be constant in the normal direction. The effect is the minimization of oscillations of the solution (which is not on the level set) and therefore keeping the errors of discretisation under control. On the other hand, the solution of the embedded PDE, by using the Eulerian method, is not constant along the normal direction. Therefore the solution must be extended on the entire domain after each time step evolution.

By embedding the diffusion equation using the Eulerian approach a degenerate diffusion equation is obtained posed on the entire space. This degenerate equation has the property that there is no diffusion normal to the surface. The degeneracy also affects the discretisations and also the convergence of the numerical methods.

Greer [Gre06] modified the Eulerian method such that the extension step (after each time iteration) can be omitted. The boundary conditions are now imposed naturally and the effect, in the case of the diffusion equation, is that the resulting equation will be non - degenerate. To be more specific, the main modification is that the initial condition is constant in the direction perpendicular to the surface (the solution will also possess this property as time evolves).

1.1 A review of the level set method

Let a bounded, connected, open set $\Omega \subset \mathbb{R}^n$ and let Γ be a time dependent interface of codimension one such that $\Omega \subset \Gamma$. The goal is to find the position of the interface Γ influenced by a velocity field $\mathbf{v}$. The velocity field may depend on the coordinates of the interface, on time, on the geometry, or even on the underlying physical process. Osher and Sethian [OS88] created a method, the level set method, for the numerical computation of the interface Γ. The main property of the level set method is the smooth function $\varphi(x,t)$ through which the zero level set is defined as $\varphi(x,t) = 0$.

The mapping (function) φ is called a level set function [OS88] and possesses the following properties

$$\begin{cases} \varphi(x,t) > 0 & x \in \Omega \\ \varphi(x,t) < 0 & x \notin \bar{\Omega} \\ \varphi(x,t) = 0 & x \in \partial\Omega \end{cases}$$

where $x = (x_1, ..., x_n) \in \mathbb{R}^n$ and $\partial\Omega$ is a time dependent boundary (interface), i.e. $\partial\Omega = \Gamma(t)$.

Therefore, in order to “locate” the time-dependent interface Γ the property $\varphi(x,t) = 0$ will be used (the sets on which φ vanishes). The main topological consequence is that the numerical method can handle difficult situations involving two or more interfaces, or phenomena such as interface merging, interface separation or “pinching” [Hou95]. All of this is accomplished without prior knowledge of these regions or points.

Using the fact that φ is a level set function of the interface Γ and that the vector field $\mathbf{v}$ is used to convect the motion, the following differential equation can be used for the determination of φ

$$\frac{\partial\varphi}{\partial t} + \mathbf{v} \cdot \nabla\varphi = 0 \tag{1}$$

From the point of view of geometry, the interface motion depends only on the normal component of the velocity $\mathbf{v}$. Rewriting (1)

$$\frac{\partial\varphi}{\partial t} + \mathbf{v}_N|\nabla\varphi| = 0 \tag{2}$$

where $v_N = \mathbf{v} \cdot \dfrac{\nabla\varphi}{|\nabla\varphi|}$ is the normal component of the velocity and $|\cdot|$ is the norm.

If the underlying physical process is, for example, crystal growth [OM97] or Uniform Density Island Dynamics [GRM$^+$98], (1) will still depend on the normal, but written as

$$\frac{\partial\varphi}{\partial t} + |\nabla\varphi|\gamma(\mathbf{n}) = 0 \tag{3}$$

where $\gamma(\mathbf{n})$ is a function which depends on the normal $\mathbf{n}$ of the interface and $\mathbf{n} = \dfrac{\nabla\varphi}{|\nabla\varphi|}$ and $|\cdot|$ is the norm.

1.2 The geometry of curves from the point of view of level sets

Osher and Sethian [OS88] created the level set method for dealing with surfaces belonging to a two-dimensional space or a three-dimensional space, $\mathbb{R}^2$ and respectively $\mathbb{R}^3$. Extension to higher dimensions (high codimension surfaces) were firstly attempted by Ambrosio and Soner [AS96], where the goal was to “track” a surface which moves because of its curvature. The level set function was chosen to be the squared distance to the curve in the Euclidean space (using the Euclidean norm), therefore identifying the zero level sets associated with the curve. The evolution in time was performed by the numerical solution of a PDE which depends on the level set function. The first problem that was encountered was the inability to manage the merging or “pinching” of curves, which is in essence the main advantage of using level sets.

The level set method was also applied in front tracking [Hou95]. In this field, a parametrization of the curve is chosen and discretized on a grid. The goal here is to find the discrete points where the phenomena of merging and "pinching" happen. When these points are found, a new parametrization will be applied to the curve. Ambrosio and Soner [AS96] also investigated this problem by the use of two level set functions ϕ and ψ. The curve is specified by the intersection of the level sets, which translates to the equality between the level set functions, $\varphi = \psi = 0$. This approach created more problems than it solved because the analytical theory became very difficult, although the geometric quantities can still be found [PLBS01]:

$$\begin{cases} \text{the tangent } \mathbf{t} = \dfrac{\nabla\varphi \times \nabla\psi}{|\nabla\varphi \times \nabla\psi|} \\ \text{the curvature } k\mathbf{n} = \nabla\mathbf{t}\cdot\mathbf{t} \\ \text{the torsion } \tau\mathbf{n} = -\nabla\mathbf{b}\cdot\mathbf{t} \end{cases}$$

where $\mathbf{n}$ is the normal, $\mathbf{b}$ is the binormal to the curve and $|\cdot|$ is the norm.

Even though two level set functions are used, the motion of the curve can still be found by an appropriate system of partial differential equations

$$\begin{cases} \dfrac{\partial\varphi}{\partial t} = -\mathbf{v}\cdot\nabla\varphi \\ \dfrac{\partial\psi}{\partial t} = -\mathbf{v}\cdot\nabla\psi \end{cases} \tag{4}$$

where $\mathbf{v}$ is the velocity which can depend on a underlying physical process or on the specific geometry of the curve.

To give an example of a velocity dependent curve [kZCMO96], let $\mathbf{v} = k\mathbf{n}$, where $\mathbf{n}$ is the normal to the curve. The approach for this kind of problem is to use the variational level set method in which a minimization process is used, based on a gradient descent type algorithm. Hence, minimize:

$$L(\varphi,\psi) = \int_{\mathbb{R}^3} |\nabla\varphi \times \nabla\psi|\delta(\varphi)\delta(\psi)d\mathbf{x} \tag{5}$$

where $\delta(\varphi)$, $\delta(\psi)$ are delta functions in the sense of distribution theory and $|\cdot|$ is the norm.

In a comparison to the approach of Ambrosio and Soner [AS96], the variational level set method [ZMOW98] manages automatically the phenomena of merging and "pinching".

To summarize, if Γ is a sufficiently smooth curve, then the geometric quantities associated to it can be expressed by the use of one or more level set functions.

The level set function provides the zero level set of the curve. Choosing a three-dimensional space and two level set functions

$$\begin{cases} \varphi(x, y, z, t) = 0 \\ \psi(x, y, z, t) = 0 \end{cases} \tag{6}$$

provides the members (points that belong on the curve and therefore satisfy the equations) of the zero level sets of the curve.

The level set method has also been applied to the more unusual fields such as

- heteroepitaxy of (notched) materials [Sch99a]
- Lagrangian simulations [Sch99b]

1.3 The tracked point function and the closest point function

Steinhoff in [SFW00] (and later it's predecessor [RMO99]) introduced the Dynamic Surface Extension (DSE) in order to compute the evolution of a time-dependent interface. The underlying grid is chosen to be fixed and the simulation also included the self - intersecting (wavefronts) phenomena found in geometric optics. The main difficulty is that the wavefronts can pass through each other. The method called ray-tracing has also been applied, although as time evolves the associated markers exhibit divergence, which in turn lead to the degradation of the solution.

Ruuth and Osher extended the DSE [RMO99] method in order to accurately capture this type of phenomena. In this case, from a computational point of view, the interface Γ does not have an “inside” or an “outside”. A function, which is a distance, is used such that it is constant on the tangential direction to the surface. Let the domain be $\mathbb{R}^n$ and let a point $\mathbf{x} = (x_1, ..., x_n) \in \mathbb{R}^n$. The DSE algorithm is as follows:

- let the initial interface be Γ_0 and $TP(\mathbf{x}) = CP(\mathbf{x})$ on Γ_0
- let $\mathbf{n} = \mathbf{n}_{\Gamma_0}$ and $\mathbf{v}_{\Gamma_0} = \mathbf{v}(TP(\mathbf{x}))$
- evolve $TP(\mathbf{x})$ using $\dfrac{d}{dt}TP(\mathbf{x}) = \mathbf{v}(TP(\mathbf{x}))$
- for each time step evolution extend the newly found interface Γ_{new} such that $TP(\mathbf{x}) = CP(\mathbf{x})$ on Γ_{new}
- $\mathbf{n}(\mathbf{x}) = TP(\mathbf{x})$

where

- $\mathbf{n}$ is the normal of the "analytic" interface (surface) and $\mathbf{n}_{\Gamma_0}$ is the normal to the initial interface
- Γ_0 is the initial interface, i.e. initial coordinates in the discretized grid of the interface
- $TP(\mathbf{x})$ is a function which returns the "tracked" points, i.e. the cloud of points which will be used to find the interface at each time step evolution
- $CP(\mathbf{x})$ is a function which return the closest point in the domain to a point on the interface and satisfies specific conditions (depending on the algorithm and/or the method that is chosen)
- $\dfrac{d}{dt}TP(\mathbf{x}) = \mathbf{v}(TP(\mathbf{x}))$ is a formal representation of the time evolution of the interface, i.e. the evolution in time of the interface depends on the velocity $\mathbf{v}$, which in turn depends on the "tracked" points.

Description of the algorithm

Obtain (or pick) the initial interface Γ_0 and obtain the "tracked" points $TP(\mathbf{x})$. Because at this (initial) time step every quantity is known, the closest points $CP(\mathbf{x})$ to the surface are exactly the "tracked" points. The normal of the "analytical" surface becomes the normal of the "discretized" version Γ_0 and the velocity at the surface Γ_0 is extended into the surrounding domain by using the "tracked" points function TP. Evolution in time (time-stepping) is accomplished by the numerical method chosen, for example, the strong stability preserving Runge-Kutta method. After each iteration of time-stepping is completed, and the new "tracked" points are obtained, they are extended in order to find the new closest points $CP(\mathbf{x})$. In practice, the extension is an interpolation step, and finally obtain the new normal to the new found interface.

The flexibility of this approach enables the capture of wavefronts which pass through each other by letting the level sets, generated at each time step, to pass through each other. Also, the interface can change its dimension unexpectedly.

The variational approach can also be used, in the sense that a minimization problem [RMO99] can be associated to the "tracked" points. Let $TP(\mathbf{x})$ be the function which returns the "tracked" points and let $MP(\mathbf{x})$ be the function which returns the closest point to the surface but from a ray - tracing point of view. To be more specific:

$$MP(\mathbf{x}) = \min_{x_\Gamma \in \Gamma} \beta |(x - x_\Gamma) \cdot \mathbf{n}(x_\Gamma)| + ||x - x_\Gamma||_2$$

The ray - tracing problem is actually a minimal time arrival problem, in the sense that the goal is to find a point x near the surface such that an ray emanating from that point will reach the interface in the shortest time possible. In an Euclidean domain, the shortest time is equivalent to the Euclidean norm, since the perpendicularity argument gives the shortest distance.

Choosing a more complicated interface problem, such as the dependence of the curvature on the motion of the interface, De Giorgi [EDG06] proposed the use of the closest point function $CP(\mathbf{x})$. For this kind of surface, the following geometric quantities will be used

$$\begin{cases} k\mathbf{n} = -\Delta\nabla(d^2/2) \\ d\nabla d = \mathbf{x} - CP(\mathbf{x}) \end{cases} \tag{7}$$

where

- k is the (local) mean curvature
- d is the distance (function) to the surface.

Combining the two equations from (7), a new formula for the mean curvature vector is obtained

$$k\mathbf{n} = -\Delta(\mathbf{x} - CP(\mathbf{x})) = \Delta CP(\mathbf{x}) \tag{8}$$

The velocity vector can now be rewritten using (8)

$$\mathbf{v} = f(\underbrace{\Delta CP(\mathbf{x})}_{\text{evaluated at } TP(\mathbf{x})} \cdot\mathbf{n})\mathbf{n} \tag{9}$$

The research in this area is very much active and efforts are being made to produce better methods [RMO99]. These methods will be used on problems such as interface merging. Extension to complicated topologies and geometries such interfaces with boundaries ("cut" surfaces) or filament(s) attached to a sheet are found in [kZCMO96] and [MOS92]. Diffusion in codimension one has also been experimented with in [SSD05].

1.4 The level set method with an underlying physical process

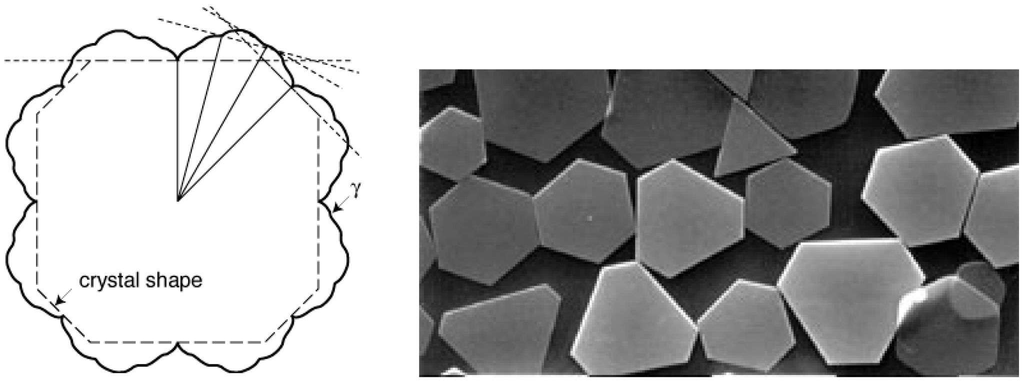
The problem of interface motion, which is influenced by a physical process has been extensively studied. To name only a few areas of research, compressible flow, incompressible flow, Stefan problems, thin films, island dynamics, crystal growth, fMRI image segmentation and many more. Notable methods include:

- front tracking [UT92, Hou95]
- phase - field methods [Kob93, NPV91]
- volume of fluid [BKZ92]

The level set method has been successfully applied to the well studied field of computational fluid dynamics [MOS92, Kar96, Kar94], but also to the kinetic crystal growth and epitaxial growth of thin films. These two areas of research deserve a special place in this review, since the continuum approach has been successfully applied to atomistic processes.

1.4.1 Kinetic Crystal Growth

Let a crystal belonging to $\mathbb{R}^n$ and let a function denote the initial configuration (shape) of the crystal. The movement of the crystal (growth) is accomplished by a normal dependent velocity field at the surface, i.e. $\mathbf{v}(\mathbf{n}) > 0$. This type of construction is called the Wulff construction (figure 1) and was conjectured first by Gross in 1918, also the crystal configuration is a minimization of a surface integral of the normal dependent velocity.



(a) The advected function $\gamma(\mathbf{n})$ of (3).

(b) Crystals taking a preferred shape.

Figure 1: The Wulff construction.

The level set method has been successfully applied to this type of interface problem in [OM97] and [Sor94]. The authors used the Hamilton-Jacobi equation, the Hopf-Bellman formula and the Brunn-Minkowski inequality in a level set framework. The result concerned the evolution of an interface, subject to a motion described by equation (3). The interface (surface) was taken to be convex, and based on this argument, the quantity (a ratio) that is needed to be minimized decreased to a minimum as time evolved. The whole article provided a proof that the generalized isoperimetric problem can be solved by the Wulff construction. As a welcome side-effect, the use of the Hamilton-Jacobi equations provided a bridge between the theory of hyperbolic partial differential equations and the energy minimization problem. The extension to anisotropic kinetic crystal growth is provided in [POMZ98, CGM$^+$98, GRM$^+$98].

Choosing a two-dimensional Wulff shape (interface), the authors in [POMZ98] gave an approach based on the Riemann problem of a scalar conservation law, in the sense that, “contact discontinuities correspond to jumps in the angle of the normal to the shape” [POMZ98]. The “non-flat” portions of the interface are equivalent to rarefaction waves and the “flat” portions are equivalent to constant states. Because of the flux function of the differential equation, the constant states will have “kinks”, which have been also found in the convex version of the Wulff energy.

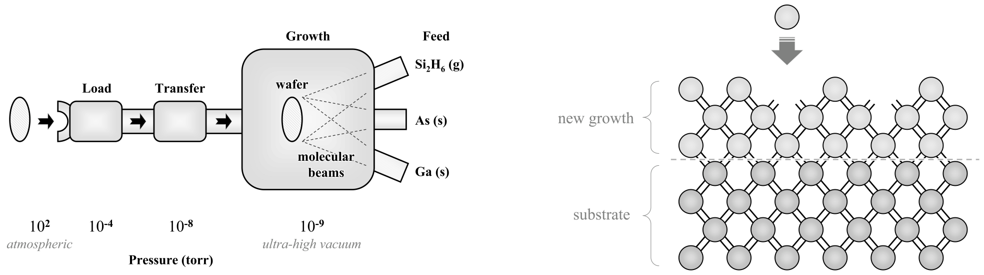
1.4.2 Epitaxial Growth of Thin Films

Thin films is a general term for describing from a theoretical and practical point of view the creation of very thin (nano scale) substrates [BCF51a]. The Molecular Beam Epitaxy (MBE) method provides a practical way of “growing” (precipitating) the substrate (figure 2). The method can be summarized as follows:

- create vacuum conditions and choose an appropriate surface for deposit
- create a steady and controllable flux of atoms which will be deposited on the desired surface
- by regulating the rate of growth, either in length or width, a monolayer is created over a period of a few seconds

Based on experiments, the contact motion between the atoms and the surface is not elastic, i.e. the atoms stick (weakly), and based on this, “surfaces” (interfaces) are created [BCF51a]. The atoms at the surface are called adatoms and are able to move from one part of the lattice to another. When an adatom falls in a neighbourhood where atoms of the same energy level are found, the bonding effect occurs, thus the adatom will stick and remain at that particular site. This of course can be exploited practically to control the “growing” process. If the neighbourhood is an incomplete monolayer, the adatom will become a part of that site.

A particularly important phenomena can occur when two adatoms colide with each other. By the process of nucleation, an incomplete monolayer is created which has the form of an island, hence the term “island formation”. This particular site will



(a) The MBE growth chamber.

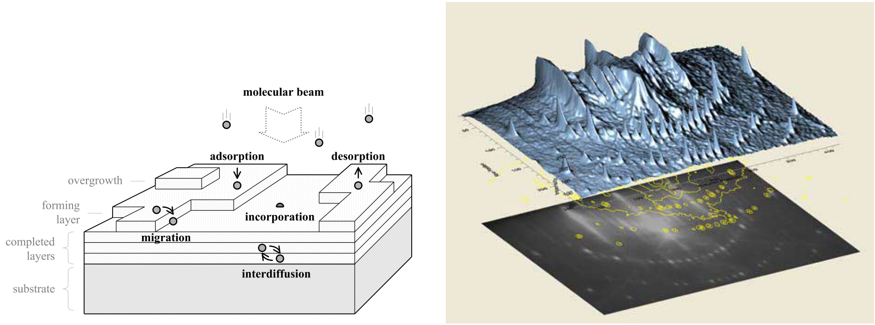
(b) Adatoms being deposited and the forming of a monolayer.

Figure 2: A general schematic of Molecular Beam Epitaxy process.

continue to "trap" adatoms at the edge, and therefore contributing to the growth of the island.

Evolving the process in time, atoms will continue to be "put" or deposited and the site will continue to grow. Islands will merge or fill in the gaps of the desired monolayer.

Taking into account that the flux of atoms keeps running, the "wedding cake" effect will occur. This phenomena happens when adatoms are being "grabbed" by the tip of an island. Islands (figure 3) can form on top of islands and the underlying monolayer will be incomplete.



(a) The formation of islands and the grabbing of adatoms.

(b) The "wedding cake" effect.

Figure 3: Islands dynamics.

The dynamics of island formation is based on a continuum model [MCO98]. This approach is chosen because it favours the analysis and is able to obtain results about the dynamics and interaction of the monolayers, since the most important quantities are considered to be longer (and on a larger scale) than an individual atom. To name only a few of the quantities:

• the effect of noise on growth, i.e. the effect of the "lack of accuracy" of the deposit process on the (existing or not) monolayer, on the islands or on the entire substrate

• the evolution and scaling between the islands, i.e main and mean island area, the steepness of the island, the step size of the steepness of the island

• the types of growth, i.e. precoalescence, coalescence

For more details about the growing of materials refer to [BCF51b].

The level set approach in the field of island dynamics is given by [MCO98]. The authors managed to include:

• the effects of a Stefan problem

• the "wedding cake" effect

• the use of multiple level sets generated by a single level set function

• the nucleation of islands

• numerical method (implicit) related optimization

The model [MCO98] considers that the islands possess a length of unit one but a continuous step descent. This is in agreement with the continuum approach, since the height of the substrate is strictly smaller than the length, hence a two-dimensional coordinate system will be used. The deposit of atoms will be modelled by a continuous function, called the adatom density function $\rho(x, y, t)$.

The interface is time-dependent and is associated with every layer of an island

$$\Gamma_i(t), \quad \text{for every layer number } i = 1, 2, ..., N$$

The main equation used is a diffusive equation [MCO98]

$$\frac{\partial \rho}{\partial t} = \nabla \cdot (D \nabla \rho) + F \quad \text{on} \quad \Omega = \bigcup_i \Gamma_i \tag{10}$$

where

• the function D is the deposition flux

• the function F is given, and usually a constant

The boundary conditions (11) are posed in such a way that irreversible aggregation is accomplished, in the sense that the adatoms are deposited on the step edges, which in turn leaves the surrounding regions of the island without any atoms.

$$\rho \text{ on } \Gamma = 0 \tag{11}$$

Robin type boundary conditions (12) occur when effects like the detachment of a atom (adatom) or energy barriers are incorporated into the model

$$\left[A\rho + B\frac{\partial \rho}{\partial n} \right] = C \tag{12}$$

where C is a given constant and the square brackets indicate the jump condition across the boundary.

The solution ρ of equation (10) can present discontinuities and will posses a jump across the boundary [MCO98]. The physical interpretation of this is that the atoms (adatoms) are likely to be assimilated into steps at the edge of the island rather than go into the lower levels.

The velocity of the boundary Γ_i is given by

$$\mathbf{v} = v_n \mathbf{n} \tag{13}$$

where $\mathbf{n}$ and v_n is the normal component of the velocity.

Since an atom is considered indivisible (in this case), a principle of conservation of atoms can be stated, which in turn will be used to find the normal component of the velocity v_n

$$\begin{bmatrix} \text{total amount of atoms, from the flux,} \\ \text{deposited to both boundaries} \\ \times \\ a^2 \\ \text{(area per atom)} \end{bmatrix} = \begin{bmatrix} \text{local rate} \\ \text{of growth} \\ \text{of the boundary} \\ v_n \end{bmatrix} \tag{14}$$

Applying the principle (14), the normal component of the velocity becomes:

$$v_n = -a^2[\mathbf{q} \cdot \mathbf{n}] \tag{15}$$

where $\mathbf{q}$ is the surface flux (adatoms) of the interface Γ_i and $\mathbf{n}$ is the outward normal.

The Kinetic Monte Carlo technique can also be incorporated [RSS03]. Diffusive transport, attachment and detachment probabilities can be used to express the flux $\mathbf{q}$ (linked, of course, to the Kinetic Monte Carlo).

The expression for the flux $\mathbf{q}$ can be simplified by using the irreversible aggregation model, in which case it becomes

$$\mathbf{q} = -D\nabla\rho \tag{16}$$

The deterministic approach falls short when dealing with the “collision” of islands (the nucleation). The probabilistic approach is chosen. If the islands nucleate as a consequence of a random collision between two atoms (adatoms) then a “new” island is formed, at site with coordinates (x, y) and at time t and the probability density function has a form of

$$P[dx, dy, dz] = \varepsilon\rho(t, x, y)^2 dtdxdy \tag{17}$$

A further simplification to the nucleation of islands can still be made and (17) can be avoided [MCO98]. Let $n(t)$ be a continuous time-dependent function which denotes the number of new islands being created

$$\frac{\partial n}{\partial t} = \underbrace{\langle \varepsilon\rho(t, x, y)^2 \rangle}_{\text{probabilistic rates at time } t} \tag{18}$$

where $\langle \cdot \rangle$ is the spatial average.

Equation (18) expresses the fact that an island will be created with a probability ρ at a site of the substrate.

The model can be extended to include even more complex thin film phenomena. Continuum equations, such as (18), can be attached to (10), to model how the density of kink sites are being influenced by the deposit of adatoms [CGM$^+$98].

The deposit flux function D can tend to infinity in order to model the case when the density of atoms being deposited is uniform on the entire substrate:

$$\frac{\partial \rho}{\partial t} = F - \lambda L\rho \tag{19}$$

where

- $\rho(t, x, y) = \rho(t)$
- L is the total length of the boundaries of all the islands.

In the case of (19), the normal componenet of the velocity takes the form of

$$v_n = v_n(\mathbf{n}) \tag{20}$$

and expresses the fact that islands grow very fast and will turn into the “Wulff shape” [OM97].

2 The closest point method

The closest point method [RM08] belongs to the class of embedding methods, which in turn are based on the level set method [OS88]. The first key ability of the method is the use of the closest point function to represent the surface. The second is the evolution in time, in the sense that the PDE is solved one time-step at a time. The third represents the flexibility of choosing any kind of surface, either smooth or triangulated (codimension of one or higher). If it is required, a narrow band can be chosen around the surface without degradation of the solution (as opposed to the level set method/level set function approach [Gre06]). The boundary conditions are posed in a natural way, for example to be periodic.

2.1 The closest point function

To embed the surface PDE into the surrounding domain, the intrinsic differential operators will be replaced by their standard $\mathbb{R}^d$ counterparts. As $t \to \infty$ the solution of the surface PDE will not agree to the solution of the embedded PDE, therefore a new operator will be introduced. This operator will extend the functions defined on the surface to functions defined on the whole domain (or the domain created by the narrow band). The intrinsic differential operators are constant along the directions normal to the surface [RM08]. Consider a time-dependent surface PDE of the form:

$$\begin{cases} \dfrac{\partial u_s}{\partial t} + \left|\dfrac{\partial u_s}{\partial s}\right| = 0 \\ u(0,s) = f(s) \end{cases} \tag{21}$$

where

- the initial condition f depends on the arclength s and is defined only on the surface of the domain
- u_s is the solution of the equation (which exists only on the surface)
- $\dfrac{\partial u_s}{\partial s}$ is the intrinsic quantity defined only on the surface
- $|\cdot|$ is the norm

The next step is to extend (embed) equation (21) into the whole domain. This is accomplished by replacing the intrinsic gradients with their standard Cartesian counterparts, taking a two dimensional example for simplicity. In this case equation (21) becomes:

$$\begin{cases} \dfrac{\partial u}{\partial t} + |\nabla u| = 0 \\ u(0,x,y) = g(x,y) \end{cases} \tag{22}$$

where

- the initial condition g is defined on the entire domain and represents the embedding of the initial condition f
- u is the solution of the equation defined on the whole domain
- ∇u is the Cartesian gradient (a scalar or vector field, depending on the underlying mathematical or physical phenomena)
- $|\cdot|$ is the norm

As noted above, equations (21) and (22) agree only at short-times. Further more, the equations must agree only at the surface, in the sense that the solutions are the same. In order to accomplish this, the extension step operator E, which depends on u_S, $E[u_S]$, must have gradients along the surface direction, but not normal to the surface.

A practical method to achieve this kind of result (constant in the normal direction) is to take advantage of the closest point representation of the surface. For any $x \in \mathbb{R}^d$, the closest point function, denoted by $cp(x)$ will return the closest point, which belongs to the surface, closest to x.

Lemma

Let $S \subset \mathbb{R}^d$ be a surface and let cp denote the closest point function. For every point $x \in \mathbb{R}^d$, $cp(x) \in S$ is the closest point to x in Euclidean distance:

$$cp(x) = \text{argmin}_{cp(x) \in S} ||x - cp(x)||_2 \tag{23}$$

and the link between the operator E and $cp(x)$ is given by

$$E[u_S](x) = u_S(cp(x)). \tag{24}$$

where $||\cdot||$ is the Euclidean norm.

Remark

The closest point representation technique gives the possibility of using surfaces like the Klein bottle, the Mobius strip, algebraic curves and even triangulated surfaces such as Annie Hui's pig or the Brusselator [MR08, MR09, TMR09, MBR11].

2.2 The determination of the closest point function

There are two ways of actual determining the closest point function:

- the analytical approach, based on the signed distance function
- the numerical optimisation approach, based on the "least-squares" problem, which in turn is a minimization problem [Str99, Mau03]

In this dissertation only the analytical approach will be used.

Definition

A function $d : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ is defined as a distance function if:

$$d(x) = \min ||x - x_C|| \quad \forall x_C \in \partial\Omega \tag{25}$$

where $\partial\Omega$ is boundary of the domain Ω.

The geometric interpretation of the definition for a three-dimensional domain is given by:

$$\text{choose a point } x \in \Omega \subset \mathbb{R}^3 \text{ and find } cp(x) \in \partial\Omega \text{ closest to } x \tag{26}$$

In the case when Ω belongs to the Euclidean space:

$$d(x) = \min ||x - x_C||_2 \quad \forall x_C \in \partial\Omega \tag{27}$$

where $|| \cdot ||_2$ is the Euclidean norm.

An important property of the distance function is

$$||\nabla d||_2 = 1 \tag{28}$$

Component-wise, for the three-dimensional case, (28) becomes

$$\sqrt{\left(\frac{\partial d}{\partial x}\right)^2 + \left(\frac{\partial d}{\partial y}\right)^2 + \left(\frac{\partial d}{\partial z}\right)^2} = 1 \quad \forall\ (x, y, z) \in \Omega \tag{29}$$

Remark

Equation (28) is only valid in a general sense, using the fact that there is a unique closest point. In practice, because of standard discretisations techniques or triangulations, uniqueness fails.

Definition

A function $\phi : \Omega \in \mathbb{R}^n \to \mathbb{R}$ is defined as the signed distance function if:

$$|\phi(x)| = d(x) \quad \forall x \in \Omega \tag{30}$$

The signed distance function (30) has the following properties:

$$\begin{cases} 1. \quad \phi(x) = d(x) = 0 \quad \forall x \in \partial\Omega \\ 2. \quad \phi(x) = -d(x) \quad \forall x \in \Omega^- \\ 3. \quad \phi(x) = d(x) \quad \forall x \in \Omega^+ \\ 4. \quad ||\nabla\phi|| = 1 \\ 5. \quad \text{signed distance functions are monotonic across interfaces} \end{cases}$$

The closest point [EDG06, MR07, MR08] is found by using the signed distance function (30):

$$\mathbf{x}_C = \mathbf{x} - \phi(\mathbf{x})\mathbf{N} \quad \text{where } \mathbf{N} = \frac{\nabla\phi}{||\nabla\phi||}. \tag{31}$$

Component-wise for a generic point $P = (x_C, y_C, z_C)$ belonging to a surface in a three-dimensional space

$$\begin{cases} x_C = x - \phi(x,y,z)\dfrac{\dfrac{\partial\phi}{\partial x}}{\sqrt{\left(\dfrac{\partial\phi}{\partial x}\right)^2 + \left(\dfrac{\partial\phi}{\partial y}\right)^2 + \left(\dfrac{\partial\phi}{\partial z}\right)^2}} \\ y_C = y - \phi(x,y,z)\dfrac{\dfrac{\partial\phi}{\partial y}}{\sqrt{\left(\dfrac{\partial\phi}{\partial x}\right)^2 + \left(\dfrac{\partial\phi}{\partial y}\right)^2 + \left(\dfrac{\partial\phi}{\partial z}\right)^2}} \\ z_C = z - \phi(x,y,z)\dfrac{\dfrac{\partial\phi}{\partial z}}{\sqrt{\left(\dfrac{\partial\phi}{\partial x}\right)^2 + \left(\dfrac{\partial\phi}{\partial y}\right)^2 + \left(\dfrac{\partial\phi}{\partial z}\right)^2}} \end{cases}$$

Equation (31) can be applied directly to find the coordinates of a closest point (x_C, y_C) based on a generic point (x, y). For a two-dimensional circle $x^2 + y^2 = R$ the signed distance function becomes $\phi(x, y) = \sqrt{x^2 + y^2} - R$ and the closest points are given by:

$$(x_C, y_C) = \left(\frac{Rx}{\sqrt{x^2+y^2}}, \frac{Ry}{\sqrt{x^2+y^2}}\right) \tag{32}$$

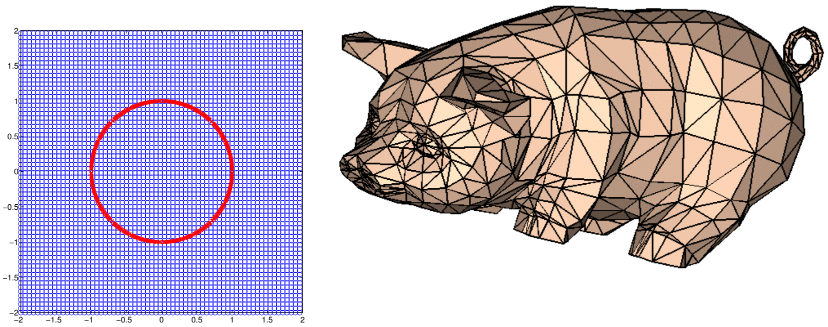
In the same manner, for the two-dimensional ellipse $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$ the signed distance function becomes $\phi(x, y) = \sqrt{\frac{x^2}{a^2} + \frac{y^2}{b^2}} - 1$ and the closest points are given by:

$$(x_C, y_C) = \left(\frac{\frac{x}{a^2}}{\sqrt{\frac{x^2}{a^4} + \frac{y^2}{b^4}}}, \frac{\frac{y}{b^2}}{\sqrt{\frac{x^2}{a^4} + \frac{y^2}{b^4}}} \right)$$

The preliminary steps, prior to the actual application of the closest point method, are as follows [RM08]:

• choose a surface, either smooth or triangulated

• based on the equation of the surface derive the closest point function by using (31) (in the case of a triangulated surface as in figure 4b, the array of closest points will be found by solving the "least-squares" problem)



(a) A standard computational grid [RM08].

(b) A triangulated (polygonal) surface [TMR09].

Figure 4: Examples of different domains on which the closest point method has already been applied.

• discretise the continuous domain Ω into its discrete counterpart Ω_Δ

• choose a narrow band around the surface (banding)

• derive the embedded PDE using the relationships between the intrinsic quantities and their Cartesian counterparts

• embed the initial condition, i.e. apply the closest point function to the initial condition of the surface PDE

• apply a high-order finite difference scheme, in time and/or space

• after each evolution in time apply the closest point extension, i.e. in the case of a Runge-Kutta type method, the closest point extension is applied after each stage

Remark

The closest point extension represents an interpolation process. In this thesis polynomial based interpolation is used. As a side note, the extension to n-dimensions is done in a dimension-by-dimension fashion. The general algorithm mentioned above can be illustrated more clearly by the following examples [RM08] based on a generic PDE.

Example

Let S be a generic surface belonging to a domain Ω and let the general form of a time-dependent surface PDE, containing terms which use the gradient and the Laplacian (for example reaction, advection and diffusive transport):

$$\begin{cases} \dfrac{\partial u_s}{\partial t} = F(x, u_s, \nabla_s u_s, \Delta_s u_s) \quad \text{on } S \\ u(s, 0) = g(\cdot) \end{cases} \tag{33}$$

where

• u_s is the time-dependent solution defined only on the surface of the domain

• $F(x, u_s, \nabla_s u_s, \Delta_s u_s)$ is a generic function which depends on the intrinsic quantities defined only on the surface of the domain

• $u(s, 0)$ is the initial condition which depends on the arclength s

• $g(\cdot)$ is a generic function depending on one or more arguments or it may very well be constant

Time discretisation is based on the first order forward Euler method and the following surface semi-discrete differential equation is obtained (since the space is still continuous):

$$\begin{cases} u_s^{n+1} = u_s^n + \Delta t F(x, u_s^n, \nabla_s u_s^n, \Delta_s u_s^n) \quad \text{on } S \\ u_s^0 = g(\cdot) \end{cases} \tag{34}$$

where

• u_s^n is the temporal discretisation at time n of the solution u_s defined only on the surface of the domain

• $F(x, u_s^n, \nabla_s u_s^n, \Delta_s u_s^n)$ is the temporal discretisation at step n of the generic function which depends on the temporal discretised intrinsic quantities defined only on the surface

- u_s^0 is the discretised initial condition at time $n = 0$ which depends on the arclength s
- $g(\cdot)$ is a generic function depending on one or more arguments or it may very well be constant

The next step is the to use the closest point function to embed the equation into the whole domain:

$$\begin{cases} u^{n+1} = u^n(cp) + \Delta t F(cp, u^n(cp), \nabla u^n(cp), \Delta u^n(cp)) & \text{on } \Omega \\ u^0 = g(\cdot) \end{cases} \tag{35}$$

where

- $u^n(cp)$ is the temporal discretised solution defined on the entire embedded space by using the closest point function
- $F(cp, u^n(cp), \nabla u^n(cp), \Delta u^n(cp))$ is the temporal discretisation at step n of the generic function which depends on the temporal discretised embedded intrinsic quantities defined on the entire domain by using the closest point function
- u^0 is the discretised initial condition at time $n = 0$ which is defined on the entire domain by using the closest point function

The right hand side of equation (35) is then discretised in space using standard finite difference operators. The purpose of the closest point function, in this case, is to update and map the values of the numerical solution at every time step.

Example

Let the general form of a time-dependent surface PDE, containing terms which use norms of gradients or the divergence of the curvature (for example in fields such as image processing, contour sharpening, segmentation, noise removal or the computation of the geodesics)

$$\begin{cases} \dfrac{\partial u_s}{\partial t} = F\left(x, u_s, ||\nabla_s u_s||, div\left(\dfrac{\nabla_s u_s}{||\nabla_s u_s||}\right)\right) & \text{on } S \\ u(s,0) = g(\cdot) \end{cases} \tag{36}$$

where

- u_s is the time-dependent solution defined only on the surface of the domain
- $F\left(x, u_s, ||\nabla_s u_s||, div\left(\dfrac{\nabla_s u_s}{||\nabla_s u_s||}\right)\right)$ is a generic curvature dependent function which depends on the intrinsic quantities defined only on the surface of the domain
- $u(s, 0)$ is the initial condition which depends on the arclength s

- $g(\cdot)$ is a generic function depending on one or more arguments or it may very well be constant
- $||\cdot||$ is the norm

The forward Euler time-stepping is used and the following surface semi-discrete differential equation is obtained (since the space is still continuous):

$$\begin{cases} u^{n+1} = u^n + \Delta t F\left((x, u_s^n, ||\nabla_s u_s^n||, div\left(\frac{\nabla_s u_s^n}{||\nabla_s u_s^n||}\right)\right) & \text{on } S \\ u^0 = g(\cdot) \end{cases} \tag{37}$$

where

- u_s^n is the temporal discretisation at time n of the solution u_s defined only on the surface of the domain
- $F\left((x, u_s^n, ||\nabla_s u_s^n||, div\left(\frac{\nabla_s u_s^n}{||\nabla_s u_s^n||}\right)\right)$ is the temporal discretisation at step n of the generic curvature dependent function which depends on the temporal discretised intrinsic quantities defined only on the surface of the domain
- u_s^0 is the discretised initial condition at time $n = 0$ which depends on the arclength s

As before, the closest point function is applied to embed the surface PDE into the entire domain

$$\begin{cases} u^{n+1} = u^n(cp) + \Delta t F\left((cp, u^n(cp), ||\nabla_s u^n(cp)||, div\left(\frac{\nabla_s u^n(cp)}{||\nabla_s u^n(cp)||}\right)\right) & \text{on } \Omega \\ u^0 = g(\cdot) \end{cases} \tag{38}$$

2.3 The intrinsic quantities and the closest point function

This analysis, which is based on the intrinsic quantities, is used to prove that the embedded PDE is equivalent to the surface PDE, by the use of the closest point function. The intrinsic quantities are:

- the intrinsic gradient, denoted by $\nabla_S u_S(x)$, $\forall x \in S \subset \partial\Omega$
- the intrinsic divergence, denoted by $div_S u_S(x)$, $\forall x \in S \subset \partial\Omega$
- the intrinsic Laplacian, denoted by $\Delta_S u_S(x)$, $\forall x \in S \subset \partial\Omega$ where $\partial\Omega$ is the boundary of the domain Ω.

The intrinsic quantities are linked to the closest point function through the following fundamental properties [RM08, MR08] in a three-dimensional domain:

- Let a scalar field be $u : \mathbb{R}^3 \rightarrow \mathbb{R}$, and let u be constant along the directions normal to the surface. Then:

$$\nabla_S u = \nabla u \quad \text{on } S \tag{39}$$

- Let the vector field be $u : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ and let u be constant along the directions normal to the surface. Then:

$$div_S\ u = div\ u \quad \text{on } S \tag{40}$$

As a remark, the first property can be considered an existence argument of the closest point function.

Let $u : S \rightarrow \mathbb{R}$ and let $u(cp)$ be constant along the directions normal to S, then by using (39):

$$\nabla_S u = \nabla u(cp) \quad \text{on } \Omega \tag{41}$$

Let $u : S \rightarrow \mathbb{R}$ and let $u(cp)$ be constant along the directions normal to S, then by using (40):

$$div_S\ u = div\ u(cp) \quad \text{on } \Omega \tag{42}$$

In the same manner, the extension to a surface diffusion flow (heat operator) can be made. To this end, let the operator be defined as:

$$div_S(a(x)\nabla_S u(x)) \quad \text{with} \quad \text{on } S \tag{43}$$

and by applying (41) and (42), the intrinsic operator (43) becomes:

$$div_S(a(x)\nabla_S u(x)) = div(a(cp(x))\nabla u(cp(x))) \quad \text{on } \Omega \tag{44}$$

where $a(cp(x))\nabla u(cp(x))$ is tangent to the level surfaces of S.
This approach can be used for an ever more general function a of the form

$$a = a(x, u_S, \nabla_S u_S) \quad \text{on } S \tag{45}$$

and the extension is given by

$$div_S(a(x, u_S, \nabla_S u_S)\nabla_S u(x)) = div(a(cp(x), u(cp(x)), \nabla u(cp(x)))\nabla u(cp(x))) \quad \text{on } \Omega \tag{46}$$

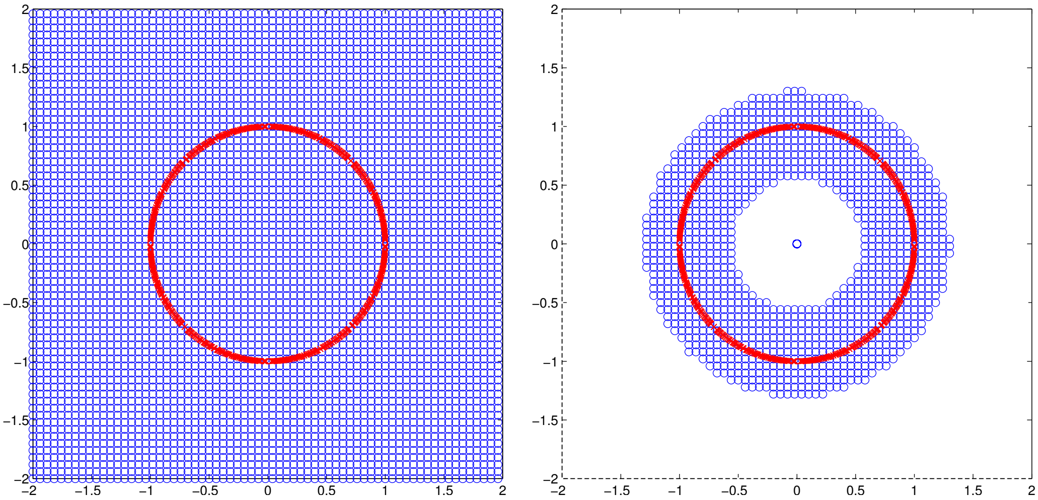
2.4 Banding

From the point of view of the domain, the closest point method uses the closest point function to change the domain of the surface PDE into the standard Cartesian domain. Since it is a numerical method, the aspect of efficiency and number of operations is of most importance.

As mentioned earlier, the technique of using a narrow band around the surface is used. The computational domain becomes

$$\Omega_{\Delta} = \{x : ||x - cp(x)||_2 < \lambda\} \tag{47}$$

where λ is the width of the band (figure 5).



(a) The computational grid and the unit circle constructed via (39)

(b) The application of banding.

Figure 5: The computational grid.

Remark

The other embedding methods [Gre06, BCOS01], because of their specific design, can not manage the banding process as easily as the closest point method. To be more specific, the previous embedding methods embed the surface PDE in the entire space (for example, by using a projection operator) and solve for *all* time t, therefore, when a narrow band is used, for efficiency, a number of problems are encountered:

- when the boundaries of the band are reached, artificial boundary conditions must be imposed; because of this degradation of the solution occurs
- the selection of the artificial boundaries influence the order of accuracy and therefore the solution

• there is little analytical theory when choosing λ; in practice, λ is chosen based on trial-and-error

• regularity at the boundaries is imposed by using some numerical or operator based methods which extends intrinsic quantities from the surface to the band

Time-stepping (the evolution in time) of the closest point method plays a fundamental role. The PDE, defined on the entire domain, has the same solution with the surface PDE only when the values (away from the surface) correspond to a constant normal extension of the values from the surface. In this case, the embedded PDE does not give a true solution for all time $t > 0$. However, in the case when $t = 0$, which corresponds to the posing of the initial condition, the embedded PDE does agree with the surface PDE, exactly. Choosing explicit time-stepping and applying the interpolation process, the data is found such that it has a constant normal extension. Therefore the algorithm can proceed to the next step. Because the algorithm possesses this kind of separation of stages, banding can be implemented without complications and no artificial boundaries are introduced.

The case of a banded computation is the same as the case of using the entire domain, provided the width λ of the band is chosen accordingly.

In order to use an appropriate λ a short analysis must be made and for generality we choose a d-dimensional space. The interpolation uses, for example, Lagrange polynomials of degree p for the one dimensional case. This is extended to the d-dimensional case in a "dimension-by-dimension" fashion, i.e. interpolate in the x direction, interpolate in the y direction, ... , interpolate in the d direction. A bound for λ can be derived using the following guidelines:

• pick a point x belonging to the surface

• each of the polynomial used in the interpolation will need $(p+1)^d$ points

• each point will depend on the neighboring points

• the width of the neighborhood, in Euclidean distance, gives the bandwidth of the band.

2.5 Time Stepping and the Strong Stability Preserving (SSP) Runge - Kutta

Upon the spatial discretisation of a PDE, a large system of ODEs is created and the next step is to choose an appropriate time-discretisation strategy. One of the most important is the strong stability preserving (SSP) time-discretisation.

From the point of view of a single ODE, one can choose from numerous numerical methods. To name a few, the predictor-corrector method, the Runge-Kutta method or the multistep method (which is nothing else than the natural generalization). However, when the ODE is derived from a special class of problems, specific solvers are employed.

The strong stability preserving methods are also solvers, hence the traditional methods used for analysis can be applied (for example the von Neumann stability analysis).

The point of departure represents the class of PDEs called hyperbolic conservation laws. For the one dimensional case:

$$\begin{cases} u_t + f(u)_x = 0, \quad (t,x) \in \mathbb{R} \times \mathbb{R} \\ u(0,x) = g(x) \quad x \in \mathbb{R} \end{cases} \tag{48}$$

where g is the initial condition.

The solution of (48) in most cases has discontinuities. Therefore the numerical method must manage (or must be created) to obtain a valid solution in spite of this considerable difficulty.

The first step in applying a numerical method is to discretise in space and obtain a semi-discrete form:

$$\begin{cases} u_t = F(u_i), \quad (t,x_i) \in \mathbb{R} \times \Omega_{\Delta x} \\ u(0,x_i) = g(x_i) \end{cases} \tag{49}$$

where

- $\Omega_{\Delta x} = \{x_i : a = x_0 < x_1 < x_2 < ... < x_n = b, x_i = a + i\Delta x\}$
- $F(u)$ represents a generic representation of a function which depends on the function u or on a linear or nonlinear combination of the function u
- $g(x_i)$ is the initial condition

The right hand side of (49) contains the space step, which is a small constant that depends on the mesh size, in the case of a finite difference scheme. In general terms:

$$F(u_i) = -f(u_i)_x + O(h^k) \tag{50}$$

where

- h^k is the space step
- k is called the order of the scheme

The time-discretisation now comes into play. The question regarding the numerical method being stable is raised when the system of ODEs becomes very large as a consequence of the spatial step being small (which is often the case).

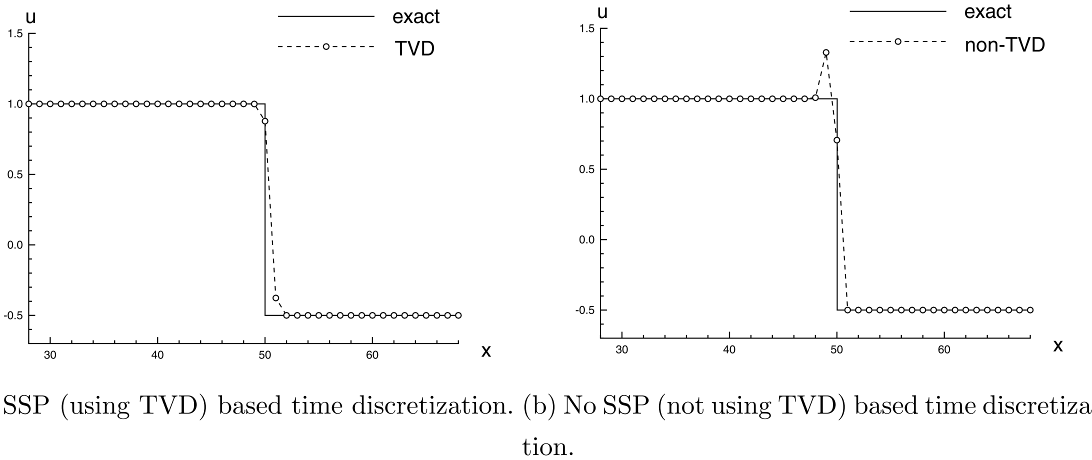
Going back to the original PDE (48), stability is measured in a certain norm, for example, the total variation norm [Shu88, SO88] or the maximum stability norm. When the discretisation becomes complete, the first equation of (49) will have a form of:

$$u_i^{n+1} = u_i^n + \Delta t F(u_i^n) \quad \text{or} \quad \frac{u_i^{n+1} - u_i^n}{\Delta t} = F(u_i^n) \tag{51}$$

with an order of 1 in time and is nothing else than the first order forward Euler method.

In the case when stability analysis is applied to hyperbolic equations, a bound represented by the ratio between the time step and the space step is derived. Globally, equation (51) has order 1 and does not depend on the spatial discretisation (no matter how high the order is). Therefore, the goal is to achieve high order accuracy also in time, while preserving stability.

The standard approach when dealing with a time dependent partial differential equation is to first discretise the spatial variables to obtain a semi - discrete system of ODEs in the time variable. Afterwards, depending on the type of problem, time discretisation is done using an explicit or implicit method. Of course, in this whole process of discretisation (spatial and/or temporal) the question of stability remains and how to deal with oscillations (figure 6).



(a) SSP (using TVD) based time discretization. (b) No SSP (not using TVD) based time discretization.

Figure 6: An example of oscillations (shock forming) based on Burgers equation with a discontinuous initial condition.

2.5.1 The Strong Stability Preserving (SSP) time stepping

The starting point in developing the strong stability preserving (SSP) time discretisation is to assume that the first order forward Euler time discretisation is strongly stable under a certain norm, when Δt has an appropriate bound. The next step is to find a high order method (i.e. Runge-Kutta) which maintains the SSP property under the same norm (but, for example, with a different time step). The norm chosen for this analysis is the total variation norm. This was first used in the development of the total variation diminishing (TVD) high order schemes [Shu88, SO88].

Returning to equation (48), consider the time-dependent one-dimensional hyperbolic conservation law:

$$\begin{cases} u_t + f(u)_x = 0 & (t, x) \in \mathbb{R} \times \mathbb{R} \\ u(0, x) = g(x) & x \in \mathbb{R} \end{cases} \tag{52}$$

where g is the initial condition.
Rewriting and discretising the spatial derivative, a system of ODEs is obtained

$$u_t = -f(u)_x \quad \Rightarrow \quad \frac{du}{dt} = (-f(u)_x)_i \quad \Rightarrow \quad \frac{du}{dt} = F(u_i) \tag{53}$$

In practice $f(u)_x$ is discretised by a TVD high order scheme or by finite elements.

Using the first order forward Euler time stepping, and the bound for time obtained by applying the Courand-Friedrichs-Levy condition [GST01]:

$$u_i^{n+1} = u_i^n + F(u_i^n) \quad \text{and} \quad \Delta t \leq \Delta t_{FE} \tag{54}$$

leads to the TVD property for time stepping (55) and the inequality for bounded growth (56):

$$TV(u^{n+1}) \leq TV(u^n) \tag{55}$$

where

$$TV(u_i^n) = \sum_i |u_{i+1}^n - u_i^n| \text{ and } u^n = \sum_i u_i^n 1_{\{x_{i-1/2} \leq x \leq x_{i+1/2}\}}$$

$$||u^{n+1}|| = ||u^n + F(u^n)|| \leq ||u_i^n|| \tag{56}$$

or by introducing a small parameter $\varepsilon = (1 + O(\Delta t)) \approx 1$, the inequality (56) becomes the "softer" or "relaxed" version:

$$||u^n + F(u^n)|| \leq \varepsilon ||u^n||$$

Thus, the concept of strong stability refers to *no* temporal (time-stepping) growth in inequality (56), as opposed to the bounded temporal growth. The extension to higher dimensions is done in a dimension-by-dimension fashion and for every dimension inequality (56) must hold.

The next step is to use the TVD property to obtain a high order scheme, such as the SSP Runge-Kutta method.

2.5.2 SSP Runge-Kutta methods

An m-th stage forward in time Runge-Kutta method [GST01] takes the form of

$$\begin{cases} u^{(0)} = u^n \\ u^{(i)} = \sum_{k=1}^{i-1}(\alpha_{i,k}u^k + \Delta t\beta_{i,k}L(u^{(k)})) \\ u^{n+1} = u^m \end{cases} \tag{57}$$

Lemma [GST01]

If the first order forward Euler method satisfies (56), under the CFL condition (54), then the m-th stage Runge-Kutta method (57) is SSP with the CFL condition given by:

$$\Delta t \leq c\Delta t_{FE} \text{ and } c = \min_{i.k} \frac{\alpha_{i.k}}{\beta_{i,k}} \tag{58}$$

If some of the coefficients $\beta_{i,k}$ are negative then the time-stepping is done backwards in time. The first order Euler time-stepping becomes

$$u^{n+1} = u^n - \Delta t\bar{F}(u^n) \tag{59}$$

where $\bar{F}(u^n)$ denotes a generic expression involving the semi-discrete solution u^n and in this case it is called the backward in time operator.

This kind of formulation appears frequently in applications. Modifying the sign of (48) (and using the same notation):

$$\begin{cases} u_t - f(u)_x = 0 \\ u(0,x) = g(x) \end{cases} \Rightarrow \begin{cases} u_t = f(u)_x \\ u(0,x) = g(x) \end{cases} \tag{60}$$

Lemma [Shu88]

If the first order forward Euler method, using the operator F, satisfies $||u^n + F(u^n)|| \leq ||u^n||$ (using the CFL condition) and if the first order backwards Euler, using the operator $\bar{F}$, satisfies $||u^n - \bar{F}(u^n)|| \leq ||u^n||$ (using the CFL condition), then the m-th stage Runge-Kutta methods satisfies $||u^{n+1}|| \leq ||u^n||$ with:

$$\Delta t \leq c\Delta t_{FE} \text{ and } c = \min_{i.k} \frac{\alpha_{i.k}}{|\beta_{i,k}|} \tag{61}$$

where $|\beta_{i,k}|\bar{L}$ replaces $\beta_{i,k}L$ whenever $\beta_{i,k} < 0$.

Taking m=2 and respectively m=3, second order and third order SSP Runge-Kutta method are obtained from (57). In both cases the CFL condition coefficient $c = 1$. Both methods are optimal and considered as low-storage.

$$\text{2nd order SSP Runge-Kutta} \Rightarrow \begin{cases} u^{(1)} = u^n + \Delta t L(u^n) \\ u^{n+1} = \frac{1}{2}u^n + \frac{1}{2}u^{(1)} + \frac{1}{2}\Delta t L(u^{(1)}) \end{cases} \tag{62}$$

$$\text{3rd order SSP Runge-Kutta} \Rightarrow \begin{cases} u^{(1)} = u^n + \Delta t L(u^n) \\ u^{(2)} = \frac{3}{4}u^n + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t L(u^{(1)}) \\ u^{n+1} = \frac{1}{3}u^n + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t L(u^{(2)}) \end{cases} \tag{63}$$

2.6 Interpolation

One of the key steps in the closest point method is the extension step. In this stage the values of the solution u are found on the closest points by means of interpolation using the surrounding points. Therefore, the interpolation technique and its inherent properties are crucial. A comprehensive account for the one dimensional theory of interpolation can be found in [MM10, Phi03] and also in [MB11].

2.6.1 Multivariate Algebraic Interpolation

Multivariate interpolation, as the name suggests, is the mathematical technique of interpolating a function of two or more variables. Comparing with the univariate interpolation, for the values of a function f at distinct points, $x_0, x_1, ..., x_n$ and a monomial basis of $\mathbb{P}_n$, $1, x, x^2, ..., x^n$, a polynomial p_n is found. This polynomial approximates the function f at chosen points, within the domain. The basis was chosen as the fundamental polynomial or the Newton polynomial which in turn gave rise to the Lagrange polynomial or the divided difference approach or the Hermite polynomial.

The extension to the multidimensional case is accomplished, as a first step, by choosing the Euclidean space $\mathbb{R}^d$ with elements denoted by $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_N$. The function f is now defined as $f : \mathbb{R}^d \to \mathbb{R}$. Let the space $C[\mathbb{R}^d]$ be the space of all linear independent continuous functions $\phi_1, \phi_2, ..., \phi_N$ with $\phi_i : \mathbb{R}^d \to \mathbb{R}$. Also, let the set of all linear combinations of ϕ_i's be $S_\phi \subset [\mathbb{R}^d]$, which is nothing else then the span of $\phi_1, \phi_2, ..., \phi_N$, and $C[\mathbb{R}^d] \cap S_\phi = \emptyset$.

The problem of interpolation, for the multidimensional case, for $1 \leq j \leq N$:

$$\text{find } a_1, a_2, ..., a_N \in \mathbb{R} \text{ such that } a_1\phi_1(\mathbf{x_1}) + a_2\phi_2(\mathbf{x_2}) + ... + a_N\phi_N(\mathbf{x_j}) = f(\mathbf{x_j}) \tag{64}$$

Expanding and writing in matrix form, (64) becomes

$$\underbrace{\begin{pmatrix} \phi_1(\mathbf{x_1}) & \phi_2(\mathbf{x_1}) & \cdots & \phi_N(\mathbf{x_1}) \\ \phi_1(\mathbf{x_2}) & \phi_2(\mathbf{x_2}) & \cdots & \phi_N(\mathbf{x_2}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x_N}) & \phi_2(\mathbf{x_N}) & \cdots & \phi_N(\mathbf{x_N}) \end{pmatrix}}_{\mathbf{A}} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{pmatrix} = \begin{pmatrix} f(\mathbf{x_1}) \\ f(\mathbf{x_2}) \\ \vdots \\ f(\mathbf{x_N}) \end{pmatrix} \tag{65}$$

for which the matrix $\mathbf{A}$ must be nonsingular in order to have a unique solution. In a three-dimensional space, a surface has an equation of the form

$$z = f(x, y) \tag{66}$$

in the sense that any point (x, y, z) belonging to the surface will satisfy (66).

The goal now is to extend the concepts from the univariate interpolation to the three-dimensional case. Taking a fixed value x_i from the x-axis and setting $x = x_i$, the function will now become

$$z = f(x_i, y) \tag{67}$$

where the implicit representation (67) is a curve and represents the intersection of the plane $x = x_i$ and the surface defined as in (66).

The approach of taking a fixed value x_i can be extended by discretizing the x-axis, $x_0, x_1, ..., x_m$, but keeping the y-axis in the continuum (for now). The next step is to use the fundamental (Lagrangian) polynomial $L_i(x) = \prod_{j \neq i} \dfrac{x - x_j}{x_i - x_j}$ [Phi03] and the implicit semi-discrete curve representation (67):

$$\xi(f; x, y) \equiv \xi(x, y) = \sum_{i=0}^{m} f(x_i, y) L_i(x) \tag{68}$$

where the function defined as in (68) is called a *blending function* [Gor71]. The blending function, because of the property of $L_i(x) = \begin{cases} 1, x = x_i \\ 0, x \neq x_i \end{cases}$, at the intersection points of the plane $x = x_i$ and the surface $z = f(x, y)$, implies:

$$\xi(f; x, y) = f(x, y) \tag{69}$$

Taking a fixed i and applying the same approach to the variable y, equation (68) will now be fully discretised and the blending function becomes:

$$\eta(f; x, y) \equiv \eta(x, y) = \sum_{j=0}^{n} f(x_i, y_j) M_j(y) \quad \text{with } i \text{ fixed} \tag{70}$$

where $M_j(y) = \prod_{j \neq i} \frac{y - y_j}{y_i - y_j}$ is the fundamental (Lagrangian) polynomial.

Combining the blending functions from (68) and (70), a two-dimensional interpolation polynomial is obtained:

$$p(x, y) = \sum_{i=0}^{m} \left(\sum_{j=0}^{n} f(x_i, y_j) L_i(x) \right) M_j(y) = \sum_{i=0}^{m} \sum_{j=0}^{n} f(x_i, y_j) L_i(x) M_j(y). \tag{71}$$

Generation based on divided difference

The technique used for deriving the one-dimensional Newton interpolation can be extended to the two-dimensional case. The multivariate polynomial (70) will take a form based on the following divided difference operator [Phi03]:

$$[x_0, x_1, ..., x_i]_x f \tag{72}$$

To be more explicit, (72) expanded takes the form of

$$[x_0, x_1]_x f = \frac{f(x_1, y) - f(x_0, y)}{x_1 - x_0} \tag{73}$$

Using the same technique as in the fundamental polynomial based approach, the Newton polynomial is used first for the x-axis:

$$\xi(f; x, y) = \sum_{i=0}^{m} \pi_i(x) [x_0, x_1, ..., x_i]_x f \tag{74}$$

Using (74), the interpolation for the y-axis is constructed:

$$\eta(f;x,y)=\sum_{j=0}^{n}\pi_j(y)[y_0,y_1,...,y_j]_y\xi \tag{75}$$

Combining equation (74) and equation (75) the desired polynomial is found

$$p(x,y)=\sum_{j=0}^{n}\pi_j(y)[y_0,y_1,...,y_j]\left(\sum_{i=0}^{m}\pi_i(x)[x_0,x_1,...,x_i]f\right) \tag{76}$$

Rewriting (76) a Newton based two-dimensional interpolation polynomial is obtained:

$$p(x,y)=\sum_{i=0}^{m}\sum_{j=0}^{n}\pi_i(x)\pi_j(y)[x_0,x_1,...,x_i][y_0,y_1,...,y_j]f. \tag{77}$$

3 Applications

The diffusion equation, the advection equation and the advection-diffusion equation will serve as numerical convergence tests for the closest point method. The previous chapters provide all the necessary theory in order to fully implement the method. The boundary conditions are taken, for each case, as periodic. The extension step is implemented by the use of a third order and respectively a fourth order polynomial derived by using the theory presented in Chapter 2, Subchapter 2.6. Spatial discretization is based on central finite differences with integer points and temporal discretization is based on the first order forward Euler and respectively on the SSP (3,3) Runge-Kutta. Although the examples are given in two dimensions, the extension to three dimensions (or more) is accomplished in a dimension by dimension fashion.

3.1 Diffusion on the unit circle and its derivation

The time-dependent, two-dimensional diffusion (heat) equation is given by:

$$\begin{cases} \dfrac{\partial u}{\partial t} = \Delta u = \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} \quad (t,x,y) \in \mathbb{R} \times \mathbb{R}^2 = \Omega \\ u(0,x,y) = f(x,y) \quad (x,y) \in \mathbb{R} \times \mathbb{R} \end{cases} \tag{78}$$

where f is the initial condition defined on Cartesian coordinates.
The closest point method relies on intrinsic quantities. That is, quantities (infinitesimal or not) defined only on the surface. The most straightforward method that can be used in order to obtain an equivalent of the standard Cartesian Laplacian is to apply a parametrization. This approach will modify the Cartesian domain into a polar domain, in this case a circle. The parametrization is given by:

$$x = r\sin(\theta), \quad y = r\cos(\theta), \quad 0 < r \leq R,\ 0 \leq \theta \leq 2\pi \tag{79}$$

where R is the upper bound (constant) for the polar variable r.
The arguments of the solution u and the initial condition f will now depend on the polar variables (r,θ). The parametrization (79) is applied to equation (78):

$$\begin{cases} \dfrac{\partial u}{\partial t} = \dfrac{1}{r^2}\dfrac{\partial^2 u}{\partial \theta^2} + \dfrac{1}{r}\dfrac{\partial}{\partial r}\left(r\dfrac{\partial u}{\partial r}\right) \quad \text{on } \Omega \\ u(0,r,\theta) = g(r,\theta) \end{cases} \tag{80}$$

where g is the initial condition defined on the polar domain Ω.

The intrinsic quantities, by definition, are defined only on the surface. Taking $r = R$ as a constant, the effect of "snapping" on the surface is obtained. Indeed the curvilinear Laplacian is defined (takes values) only on the surface, i.e. the boundary. Equation (80) now takes the form of:

$$\begin{cases} \dfrac{\partial u}{\partial t} = \dfrac{1}{R^2}\dfrac{\partial^2 u}{\partial \theta^2} + \underbrace{\dfrac{1}{r}\dfrac{\partial}{\partial r}\left(r\dfrac{\partial u}{\partial r}\right)}_{=0} = \dfrac{1}{R^2}\dfrac{\partial^2 u}{\partial \theta^2} \quad \text{on } \partial\Omega \\ \\ u(0,\theta) = g(\theta) \end{cases} \tag{81}$$

where the initial condition g depends now only on the variable θ and $\partial\Omega$ is the boundary of the domain Ω.

Since the radius is a constant, it may very well have the value of 1, hence the unit circle is obtained. Equation (81) can further be modified. Using the relantionship between the angle θ, the radius R and the arc length s ($s = R\theta$), the angle θ is represented as a function of the arc length s:

$$\theta(s) = \frac{1}{R}s \tag{82}$$

The solution of (81) depends the radius $R = 1$ and the angle θ. Using (82) and the chain rule:

$$\begin{cases} \dfrac{\partial u}{\partial s} = \dfrac{1}{R}\dfrac{\partial u}{\partial \theta} \underbrace{=}_{R=1} \dfrac{\partial u}{\partial \theta} \\ \dfrac{\partial^2 u}{\partial s^2} = \dfrac{1}{R^2}\dfrac{\partial^2 u}{\partial \theta^2} \underbrace{=}_{R=1} \dfrac{\partial^2 u}{\partial \theta^2} \end{cases} \tag{83}$$

Therefore, by using (83), the PDE (81) becomes a PDE defined only on the surface of a circle of radius 1:

$$\begin{cases} \dfrac{\partial u}{\partial t} = \dfrac{\partial^2 u}{\partial \theta^2} \\ u(0,\theta) = g(\theta) \end{cases} \Leftrightarrow \begin{cases} \dfrac{\partial u}{\partial t} = \dfrac{\partial^2 u}{\partial s^2} \quad \text{on } \partial\Omega \\ u(0,s) = g(s) \end{cases} \tag{84}$$

where g is the initial condition which depends on the arclength s defined only on the boundary $\partial\Omega$ of the domain, i.e. the unit circle.

Following the algorithm of the closest point method, equation (84) will have to be embedded. Since (84) is a time-dependent one-dimensional surface PDE, a two-dimensional Cartesian domain will serve as the embedding domain. The closest points will depend on the signed distance function, a property inherited from the level set method. For a unit circle defined as $x^2+y^2=1$, the signed distance function is:

$$\phi(x,y)=\sqrt{x^2+y^2}-1 \tag{85}$$

The determination of the closest points are based on (85) and on the equation defined as:

$$\mathbf{x}_C=\mathbf{x}-\phi(\mathbf{x})\mathbf{n} \tag{86}$$

where $\mathbf{n}=\dfrac{\nabla\phi}{||\nabla\phi||}$ and $||\cdot||$ is the Euclidean norm.

Expanding (86) into its components and applying (85), the relationship between all the points in the embedding domain and the points belonging only to the surface is obtained (figure 7):

$$\begin{cases} x_C=x-\phi(x,y)\dfrac{\dfrac{\partial\phi}{\partial x}}{\sqrt{\left(\dfrac{\partial\phi}{\partial x}\right)^2+\left(\dfrac{\partial\phi}{\partial y}\right)^2}} \\ y_C=y-\phi(x,y)\dfrac{\dfrac{\partial\phi}{\partial y}}{\sqrt{\left(\dfrac{\partial\phi}{\partial x}\right)^2+\left(\dfrac{\partial\phi}{\partial y}\right)^2}} \end{cases} \Leftrightarrow \begin{cases} x_C=\dfrac{x}{\sqrt{x^2+y^2}} \\ y_C=\dfrac{y}{\sqrt{x^2+y^2}} \end{cases} \tag{87}$$

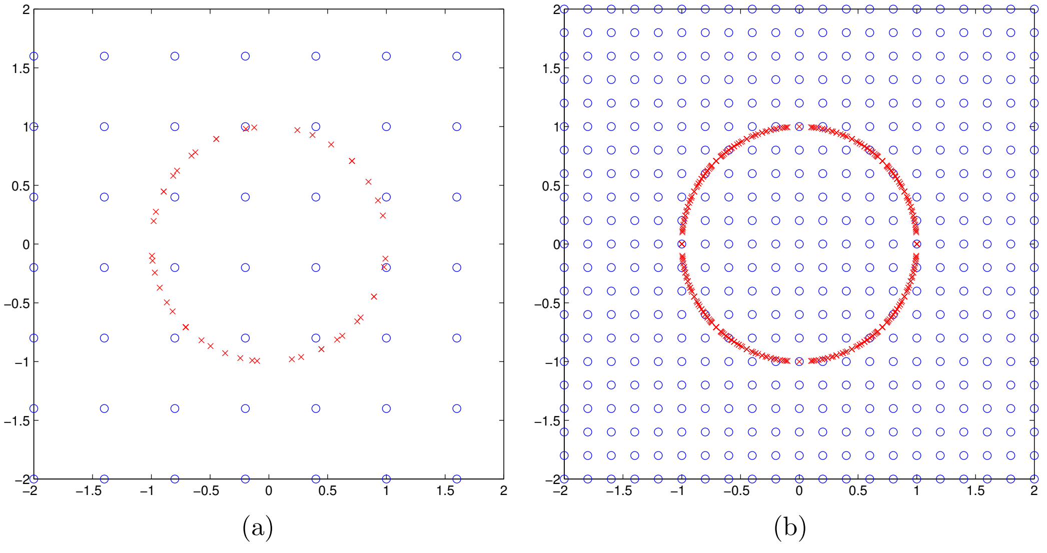


Figure 7: The closest point transformation (87) for the unit circle applied to a computational grid.

The coordinate transformation (87) represents a transformation of the domain of the surface PDE (84). The domain of the embedded PDE will have points of the form of (87). In this way, the surface domain is equivalent to the embedded domain.

The closest points, defined as in (87), are linked to the polar variable θ. For every angle θ a point on the surface is associated, hence (figure 8):

$$\forall\theta \in [0, 2\pi) \text{ there is } (x_C, y_C) \text{ belonging to the surface} \tag{88}$$

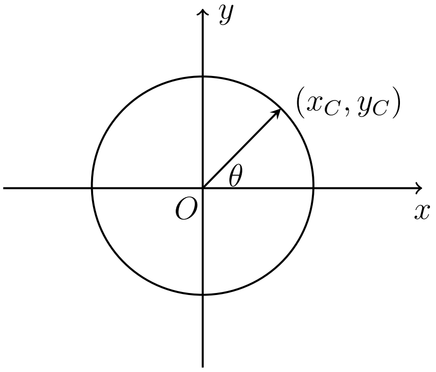


Figure 8: A generic point on the surface (x_C, y_C) and the corresponding angle θ.

Principle (88) provides a bridge between the Cartesian based numerical solution, which has values in every point of the domain, and the polar based numerical solution. The transformation between the two is nothing else than the usual transformation from polar coordinates to Cartesian coordinates.

Description of plots

The plots from figures 9 - 12 are made by using the equation (84) with the initial condition given by $u(0, \theta) = \sin(\theta)$ and the analytic solution given by $u(t, \theta) = \exp(-t)\sin(\theta)$.

In the case of the diffusion equation the time step is $\Delta t = 0.01$ and the space step is $\Delta x = 0.1$. The blue circles represent the analytic solution and the red circles represent the numerical solution. Both of them are evaluated in the same grid points. As to be expected the initial condition extended using the closest point method agrees exactly with the analytic solution at $t = 0$. The evolution in time of the numerical solution (figure 9) is smooth and oscillations do not appear, even though the time stepping is not SSP. A small degradation of the numerical solution appears at time $t = 0.08$ due to interpolation errors (figure 9). To fix this problem, the space step is taken sufficiently small such that the domain (the unit circle) becomes very smooth (due to the transformation (87)). Hence, when the solution is re-extended back on the domain (the unit circle) the effects of discontinuities diminish. Same results are obtained when banding is applied to the domain (the unit circle). Limitations do exist because of machine error, and in this case, the limit of memory in MATLAB and because of this OCTAVE has been used. The associated contour plot of figure 9 is given in figure 10. This type of representation is useful since small errors are seen more clearly (time $t = 0.08$).

Taking a larger time step $\Delta t = 0.1$, the phenomena of oscillations appears in figure 11. In this case, the difference in interpolation between time steps becomes to big and large errors are being produced, destroying the solution. The same effect can be seen in the contour representation in figure 12.

The conclusion is that the time-step based on the first order forward Euler method must be carefully chosen together with the space-step and the degree of the interpolation polynomial.

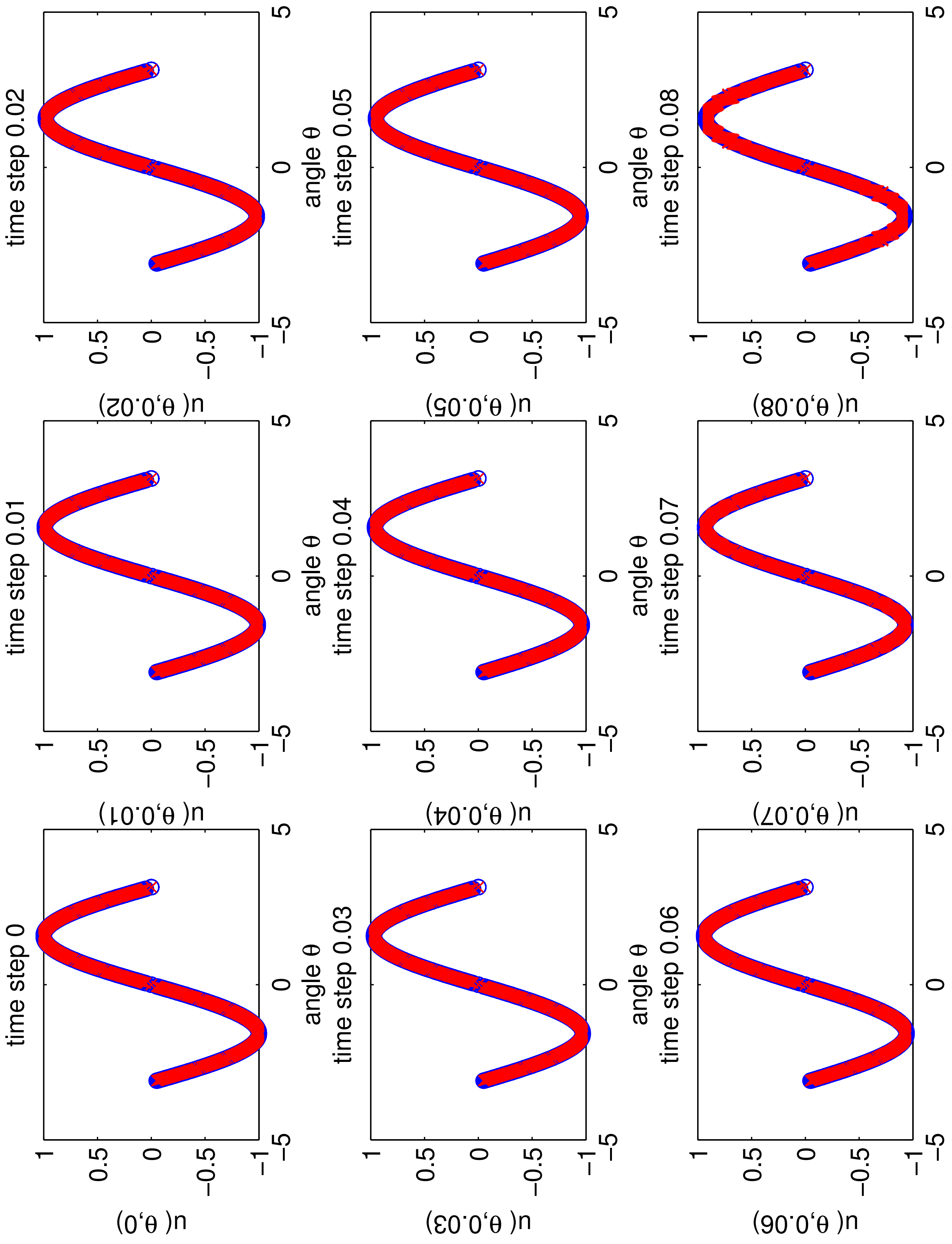


Figure 9: (The Unit Circle) The numeric solution (red) versus the analytic solution (blue) of equation (84) using the initial condition $u(0, \theta) = \sin(\theta)$ with $\Delta t = 0.01$ and $\Delta x = 0.1$.

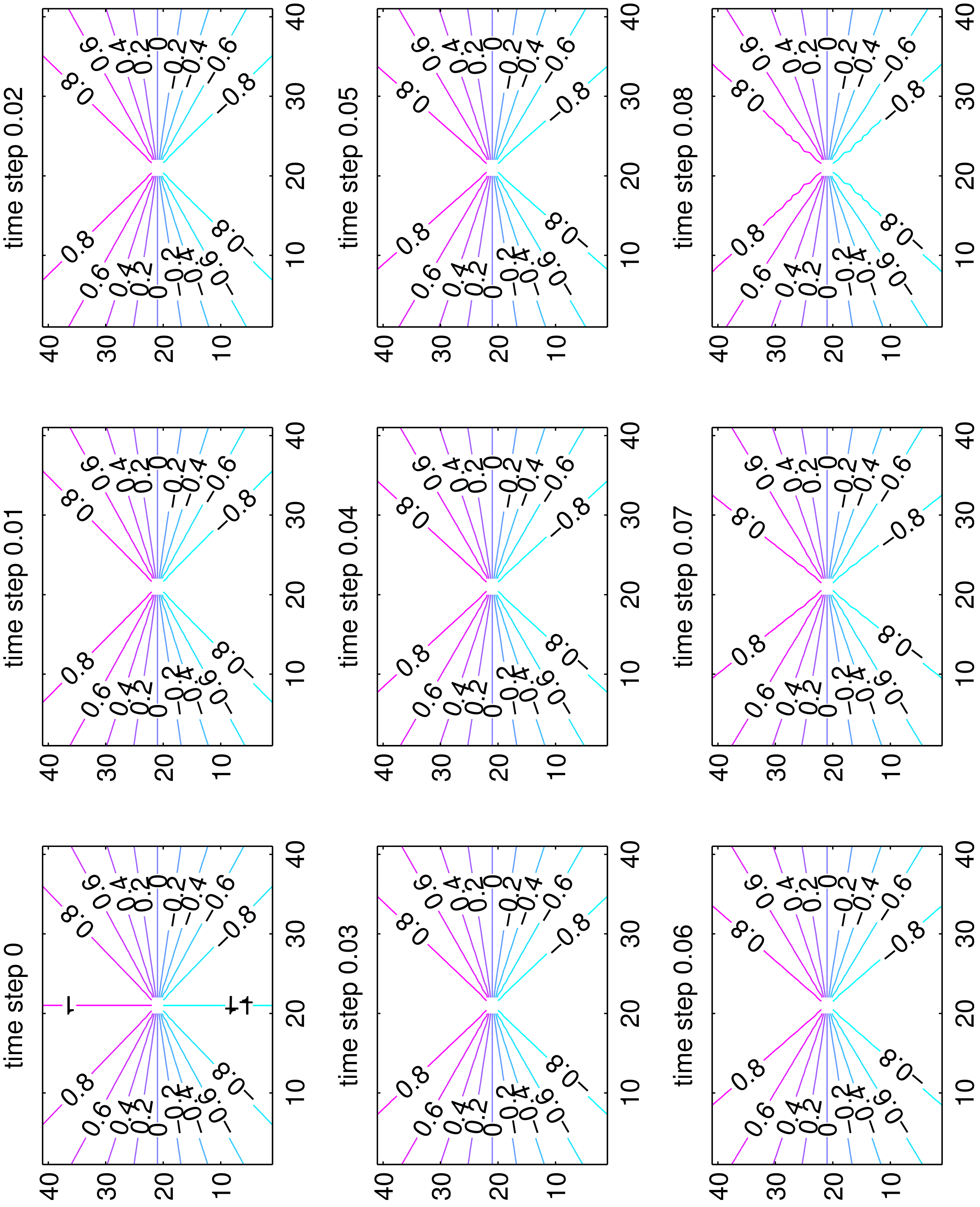


Figure 10: Contour plot: The numeric solution of equation (84) with $\Delta t = 0.01$ and $\Delta x = 0.1$.

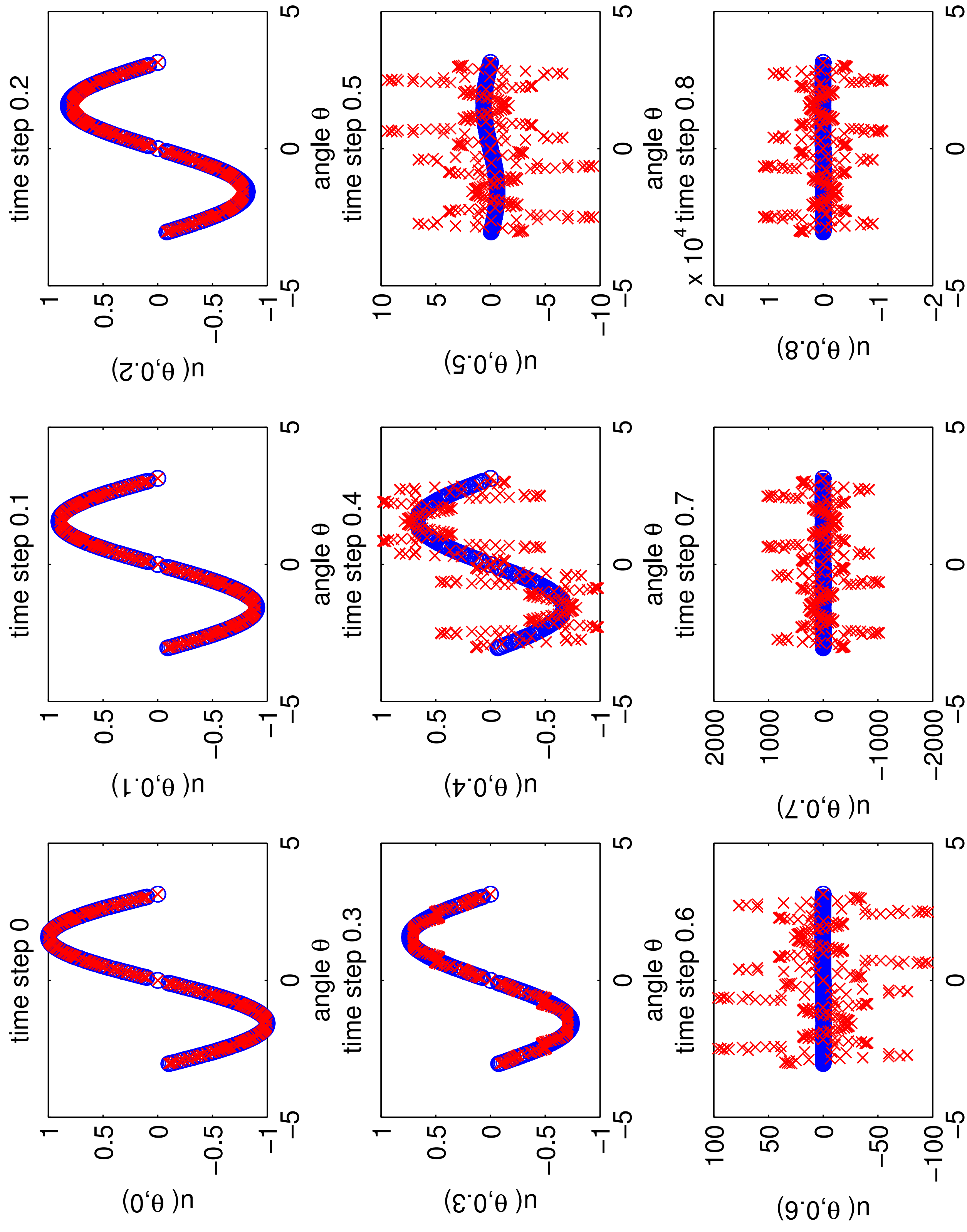


Figure 11: The numeric solution (red) versus the analytic solution (blue) of equation (84) with $\Delta t = 0.1$ and $\Delta x = 0.1$.

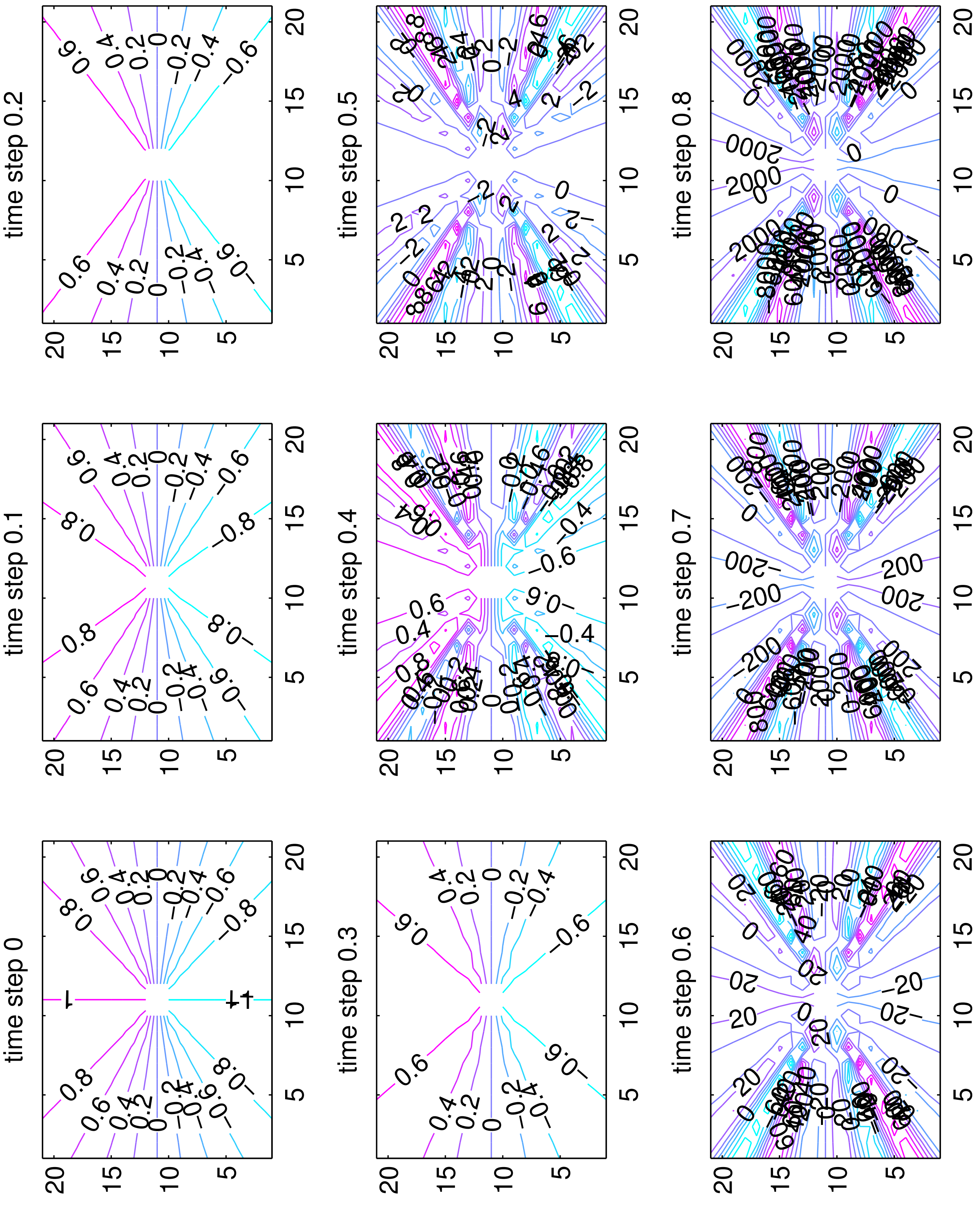


Figure 12: Contour plot: The numeric solution of equation (84) with $\Delta t = 0.1$ and $\Delta x = 0.1$.

3.2 Advection on the ellipse and its derivation

The advection equation governs the motion of a scalar field (in this case) under the influence (advection) of a velocity vector field. The velocity will become the unit tangent and the scalar field will be advected along the surface of an ellipse.

The time-dependent, two-dimensional advection equation is:

$$\begin{cases} \dfrac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u = 0 \quad (t, x, y) \in \mathbb{R} \times \mathbb{R}^2 = \Omega \\ u(0, x, y) = f(x, y) \end{cases} \tag{89}$$

where f is the initial condition defined on Cartesian coordinates.
Using the components of the velocity vector field $\mathbf{v} = (v_x, v_y)$, equation (89) can be written in component form as:

$$\begin{cases} \dfrac{\partial u}{\partial t} + v_x \dfrac{\partial u}{\partial x} + v_y \dfrac{\partial u}{\partial y} = 0 \quad \text{on } \Omega \\ u(0, x, y) = f(x, y) \end{cases} \tag{90}$$

In the case of the diffusion equation, a parametrization using polar coordinates was applied in order to obtain the desired surface PDE. In this case a different approach is taken, for diversity. The goal is to obtain a vector velocity field defined on points which belong only to the surface. Therefore the vector velocity field $\mathbf{v}$ can be taken as an equivalent to the tangent vector field $\mathbf{t}$ at the surface. Equation (90) becomes:

$$\begin{cases} \dfrac{\partial u}{\partial t} + \mathbf{t} \cdot \nabla u = 0 \\ u(0, x, y) = f(x, y) \end{cases} \quad \Leftrightarrow \quad \begin{cases} \dfrac{\partial u}{\partial t} + t_x \dfrac{\partial u}{\partial x} + t_y \dfrac{\partial u}{\partial y} = 0 \\ u(0, x, y) = f(x, y) \end{cases} \tag{91}$$

where f is defined on the entire domain Ω (which does include its boundary $\partial\Omega$).
Equation (91) can further be modified by using $\mathbf{t} \cdot \nabla u = \dfrac{\partial u}{\partial s}$:

$$\begin{cases} \dfrac{\partial u}{\partial t} + \dfrac{\partial u}{\partial s} = 0 \\ u(0, s) = g(s) \end{cases} \tag{92}$$

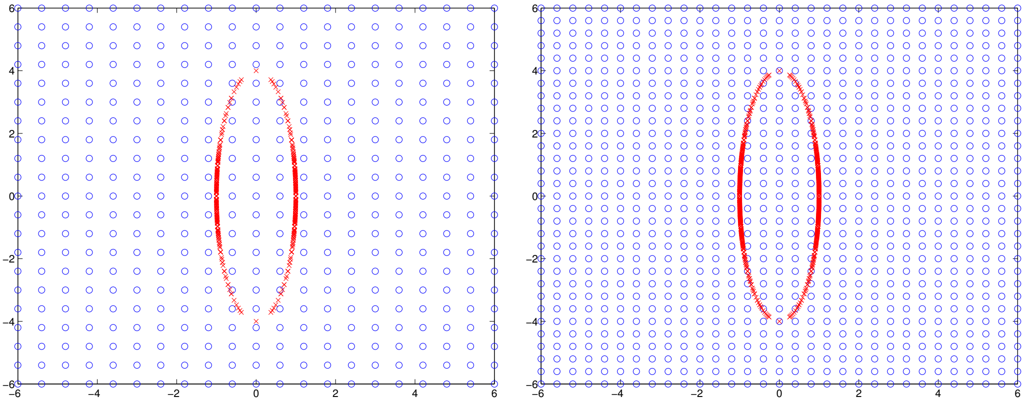
where g is the initial condition defined only on the surface, i.e. the boundary $\partial\Omega$. Equation (92) has a domain defined by all the points belonging to the surface (boundary) of the domain.

Following the same approach as in the case of the diffusion equation, the domain is now taken as the unit ellipse defined as $\dfrac{x^2}{a^2} + \dfrac{y^2}{b^2} = 1$. The closest points will be determined by using the signed distance associated to the unit ellipse, and is defined as:

$$\phi(x,y) = \sqrt{\frac{x^2}{a^2} + \frac{y^2}{b^2}} - 1 \tag{93}$$

Equation (93) is used in (86) in order to obtain the relationship between the embedding domain and the surface of the ellipse. The analytic expression for the closest points is (figure 13):

$$(x_C, y_C) = \left(\frac{\dfrac{x}{a^2}}{\sqrt{\dfrac{x^2}{a^4} + \dfrac{y^2}{b^4}}}, \frac{\dfrac{y}{b^2}}{\sqrt{\dfrac{x^2}{a^4} + \dfrac{y^2}{b^4}}} \right) \tag{94}$$



(a)

(b)

Figure 13: The closest point transformation (94) for the unit ellipse applied to a computational grid.

The tangent is defined, as a function, at every points on the surface of the ellipse, hence on every closest point. Therefore $\mathbf{t} = \mathbf{t}(x_C, y_C)$. Using the closest points as the components of the tangent, equation (91) becomes:

$$\begin{cases} \dfrac{\partial u}{\partial t} + \left(\dfrac{\dfrac{x}{a^2}}{\sqrt{\dfrac{x^2}{a^4} + \dfrac{y^2}{b^4}}} \right) \dfrac{\partial u}{\partial x} + \left(\dfrac{\dfrac{-y}{b^2}}{\sqrt{\dfrac{x^2}{a^4} + \dfrac{y^2}{b^4}}} \right) \dfrac{\partial u}{\partial y} = 0 \quad \text{on } \Omega \\ u(0, x_C, y_C) = f(x_C, y_C) \end{cases} \tag{95}$$

where f is the initial condition defined only on the closest points (x_C, y_C) by the transformation (94) (hence only on the surface of the ellipse).

Description of the plots

The plots from figures 14 - 17 are made by using equation (95) with the initial condition given by $u(0,s) = \cos^2\left(\frac{2\pi s}{L}\right)$ and the analytical solution given by $u(t,s) = \cos^2\left(\frac{2\pi(s-t)}{L}\right)$.

In the case of the advection equation the time step is $\Delta t = 0.01$ and the space step is $\Delta x = 0.1$. The blue circles represent the analytic solution and the red circles represent the numerical solution and both of them are evaluated in the same grid points. The initial condition agrees perfectly with the extension of the closest point method at $t = 0$. The evolution in time in figure 14, as with the diffusion equation, is smooth and oscillations do not appear. However in this case the strong stability preserving Runge-Kutta (3,3) method (which satisfies the total variation diminishing property) is used for time-stepping. The associated contour plot is smooth and given in figure 15.

Even though the problem with oscillations has been fixed by the TVD property of the SSP Runge-Kutta method, another problem appears, due to the shape of the domain (the unit ellipse). The closest point representation in this case needs more points to produce a smoother ellipse profile (hence removing discontinuities) as seen in the example from figure 13a. By taking a space step $\Delta x = 0.4$, the interpolation process (even if it is applied after each stage of the method) generates errors and destroys the numerical solution from figure 16 from time $t = 0.03$ to time $t = 0.08$. This can be compared to the evolution at the same time in figure 14 where an appropriate space-step has been used.

The conclusion is that the shape of the domain plays a very important part, even if the closest point transformation (94) gives satisfactory results. In this case, a careful balance between the space step and the degree (cubic) of the interpolation polynomial must be maintained and supersedes the time-step method.

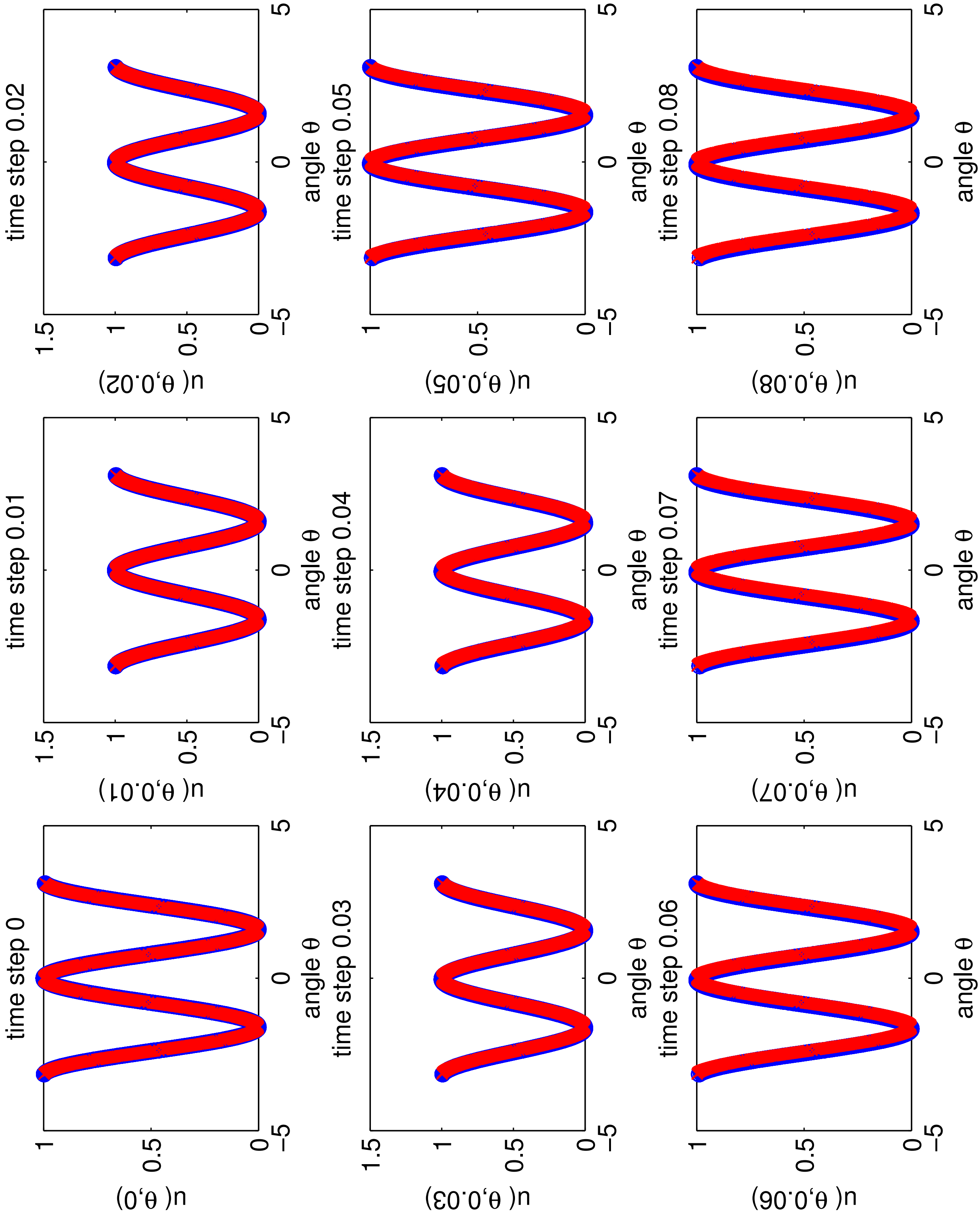


Figure 14: (The Unit Ellipse) The numeric solution (red) versus the analytic solution (blue) of equation (95) using the initial condition $u(0, s) = \cos^2\left(\frac{2\pi s}{L}\right)$ with $\Delta t = 0.01$ and $\Delta x = 0.1$.

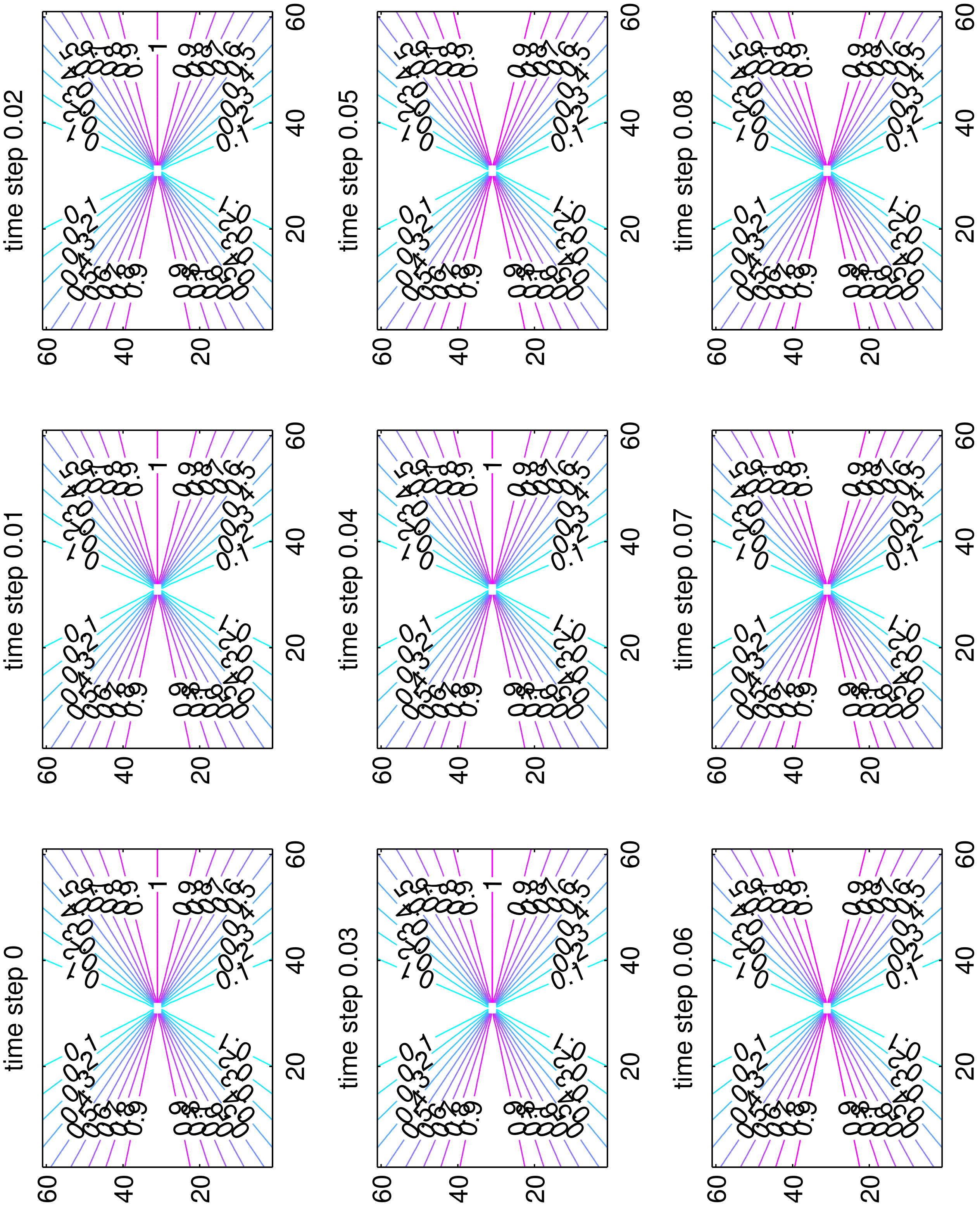


Figure 15: Contour plot: The numeric solution of equation (95) with $\Delta t = 0.01$ and $\Delta x = 0.1$.

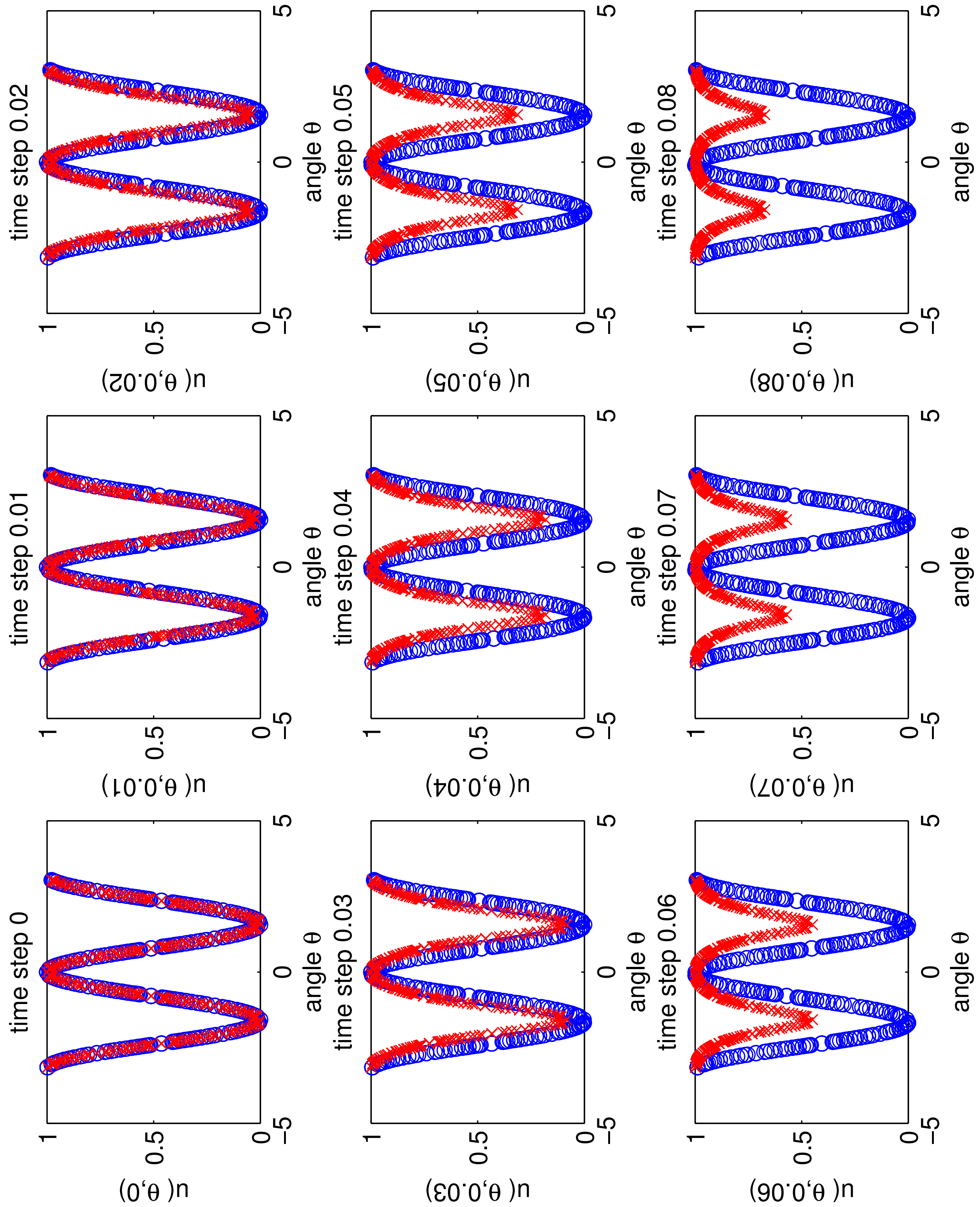


Figure 16: The numeric solution (red) versus the analytic solution (blue) of equation (95) with $\Delta t = 0.1$ and $\Delta x = 0.4$.

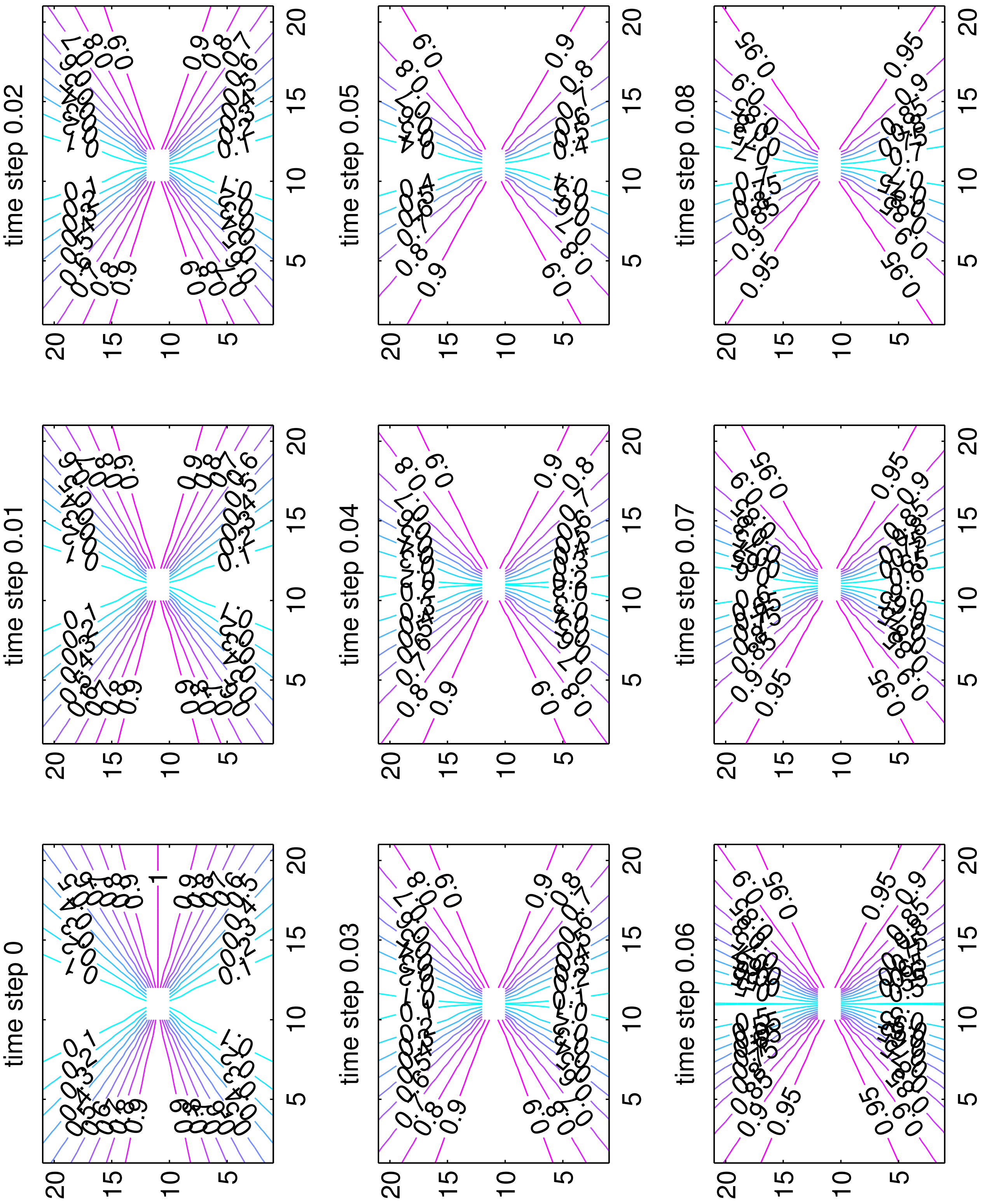


Figure 17: Contour plot: The numeric solution of equation (95) with $\Delta t = 0.1$ and $\Delta x = 0.4$.

3.3 Advection and diffusion on the ellipse

In this case, the partial differential equation will become a surface partial differential equation (96) and will be posed directly on the domain, the unit ellipse. A derivation leading to the intrinsic form (96) can be performed by combining the approach from the case of the diffusion equation and from the case of the advection equation. Therefore:

$$\begin{cases} \dfrac{\partial u}{\partial t} + \dfrac{\partial u}{\partial s} = \dfrac{\partial^2 u}{\partial s^2} \quad \text{on } S \\ u(0,s) = f(s) \end{cases} \tag{96}$$

where f is the initial condition defined only on the surface S of a domain Ω.

Equation (96) will have to be embedded into a two-dimensional domain. Using the results from the diffusion equation and the advection equation, (96) becomes:

$$\begin{cases} \dfrac{\partial u}{\partial t} + \mathbf{t} \cdot \nabla u = \Delta u \quad \text{on } \Omega \\ u(0,x,y) = g(x,y) \end{cases} \tag{97}$$

where the initial condition g is the embedding (in the entire domain Ω) of the initial condition f by using the closest point extension.

Equation (97) is the equivalent of equation (96) and is nothing else than a time-dependent advection and diffusion of a field, in this case the tangent. By using the closest point representation (94), the domain Ω of equation (97) is the unit ellipse, as in figure 13. Using the same reasoning as in equation (95), equation (97) becomes:

$$\begin{cases} \dfrac{\partial u}{\partial t} + \left(\dfrac{\dfrac{x}{a^2}}{\sqrt{\dfrac{x^2}{a^4} + \dfrac{y^2}{b^4}}} \right) \dfrac{\partial u}{\partial x} + \left(\dfrac{\dfrac{-y}{b^2}}{\sqrt{\dfrac{x^2}{a^4} + \dfrac{y^2}{b^4}}} \right) \dfrac{\partial u}{\partial y} = \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} \quad \text{on } \Omega \\ u(0, x_C, y_C) = g(x_C, y_C) \end{cases} \tag{98}$$

where f is the initial condition defined only on the closest points (x_C, y_C) by the transformation (94) (hence only on the surface of the ellipse).

Description of the plots

The plots from figures 18 - 22 are made by using equation (98) with the initial condition given by $u(0,s) = \sin^2\left(\dfrac{2\pi s}{L}\right)$ and the analytical solution given by $u(t,s) = \exp(-4t)\sin^2\left(\dfrac{2\pi(s-t)}{L}\right)$.

In the case of the advection-diffusion equation the time step is $\Delta t = 0.01$ and the space step is $\Delta x = 0.1$. The blue circles represent the analytic solution and the red circles represent the numerical solution and both of them are evaluated in the same grid points. The initial condition agrees perfectly with the extension of the closest point method at $t = 0$. The evolution in time in figure 18, as with the advection equation, is smooth and oscillations do not appear. In this case also the strong stability preserving Runge-Kutta (3,3) method (which satisfies the total variation diminishing property) is used for time-stepping. The associated contour plot is smooth and given in figure 19. The problem that appears is the "weakness" of interpolation if a cubic polynomial is used, in contrast with a simple advection, where a third order interpolating polynomial is sufficient. This effect can be seen even in figure 18 at time $t = 0.08$. The numeric solution simply breaks apart from interpolation errors in figure 20 and seen even better in the contour version in figure 21. When a space step $\Delta x = 0.1$ is taken as in figure 22 an interesting phenomena appears. The numeric solution still breaks apart, however because of the total variation diminishing property in time, the solution is "preserved", time $t = 0.08$.

The conclusion is that when the equation increases in complexity (even in a simple case like this one), the interpolation polynomial must be chosen appropriately. Macdonald in [MR08] used a WENO based interpolation to increase the accuracy. Time stepping is still important, but in certain "problematic" domains (as the unit ellipse) a different strategy, based on approximation theory must be chosen.

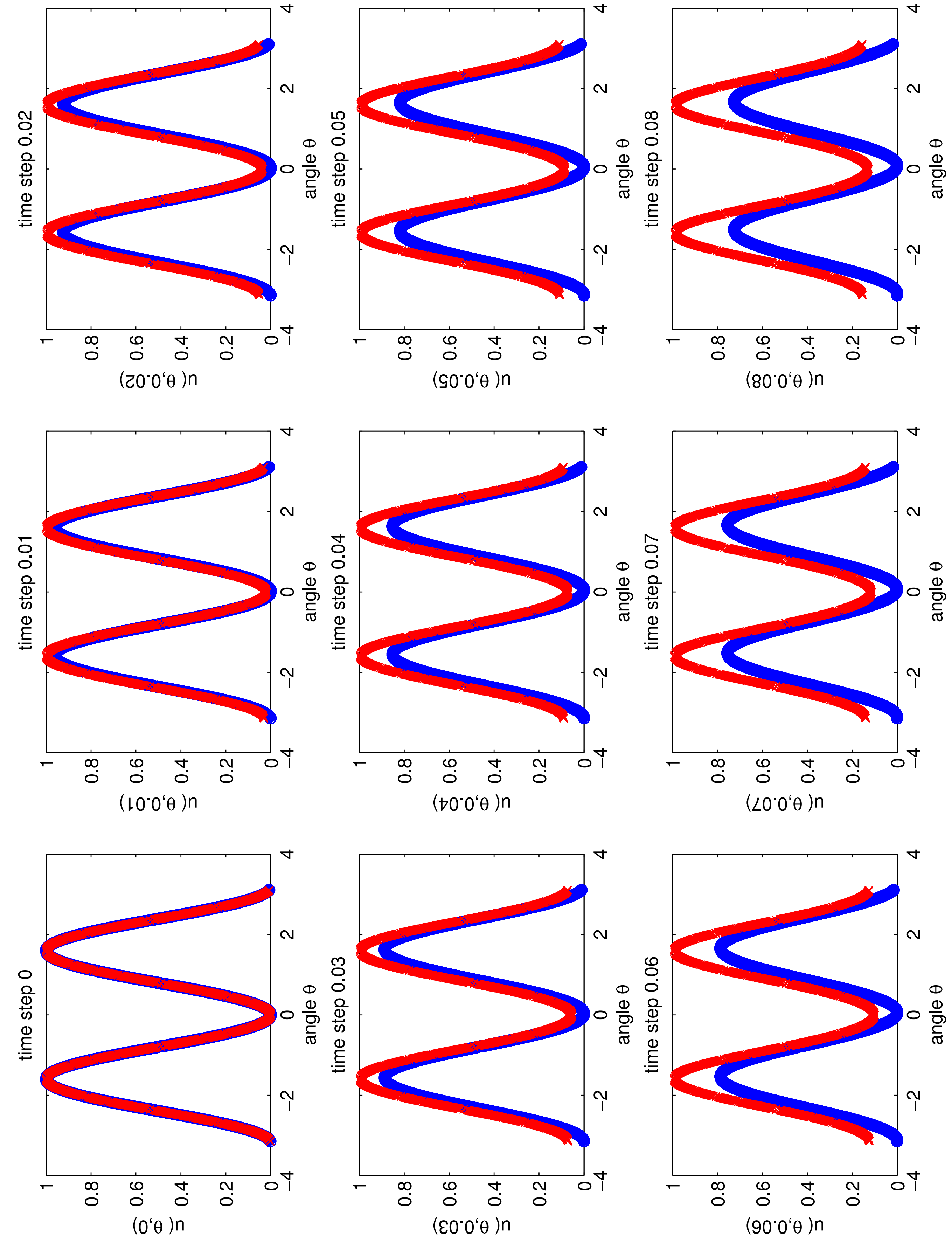


Figure 18: (The Unit Ellipse) The numeric solution (red) versus the analytic solution (blue) of equation (98) using the initial condition $u(0,s) = \sin^2\left(\frac{2\pi s}{L}\right)$ with $\Delta t = 0.01$ and $\Delta x = 0.1$.

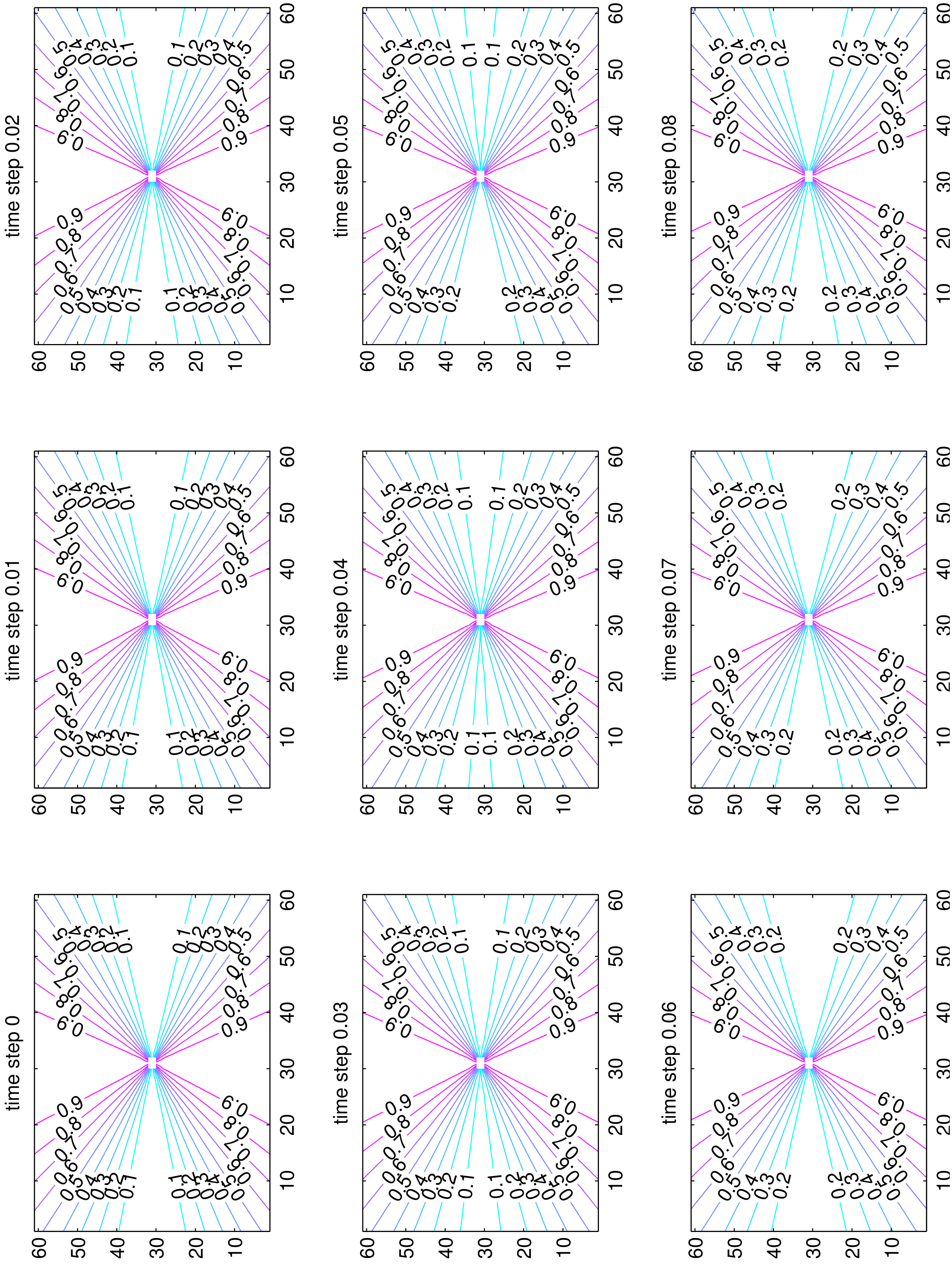


Figure 19: Contour plot: The numeric solution of equation (98) with $\Delta t = 0.01$ and $\Delta x = 0.1$.

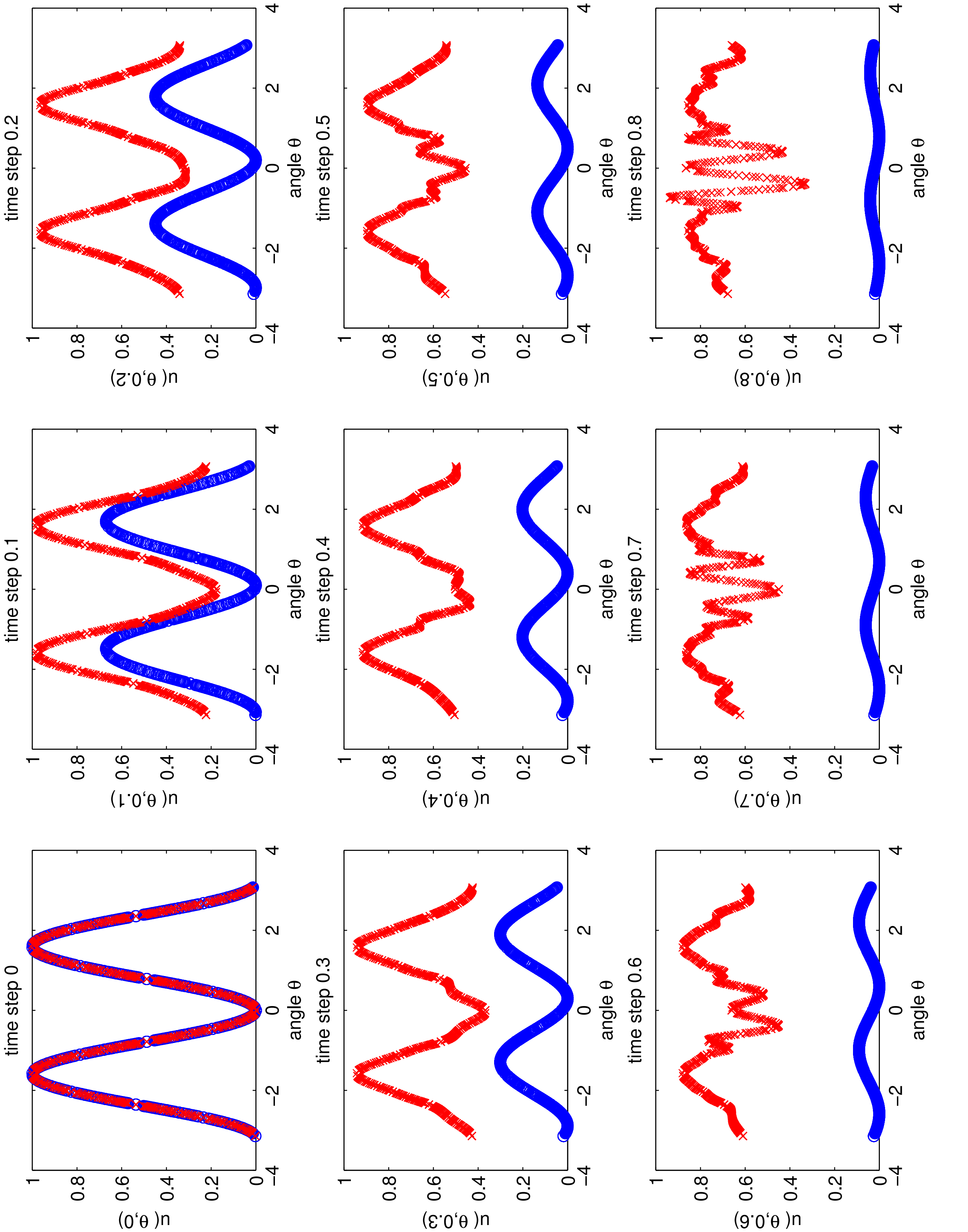


Figure 20: The numeric solution (red) versus the analytic solution (blue) of equation (98) with $\Delta t = 0.1$ and $\Delta x = 0.4$.

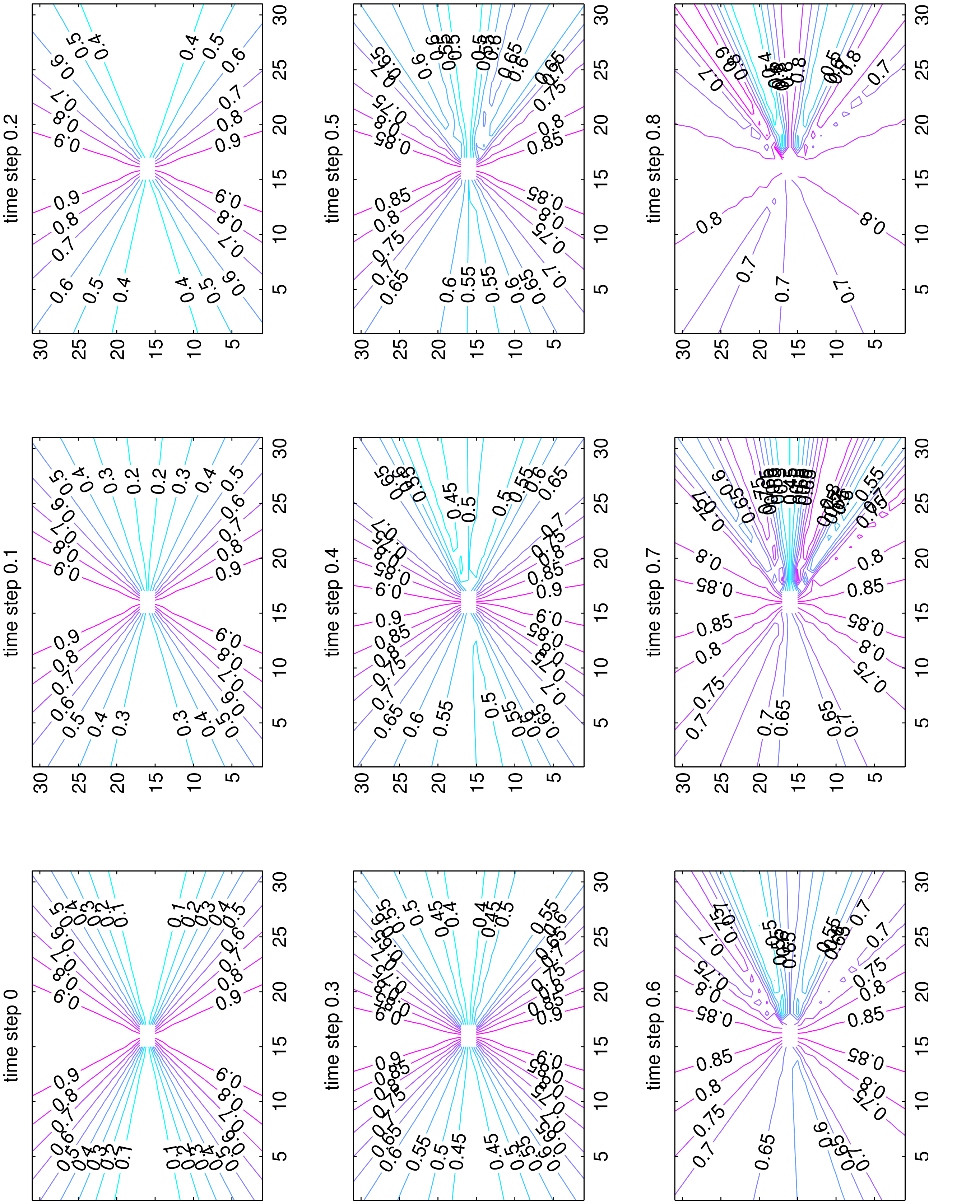


Figure 21: Contour plot: The numeric solution of equation (98) with $\Delta t = 0.1$ and $\Delta x = 0.4$.

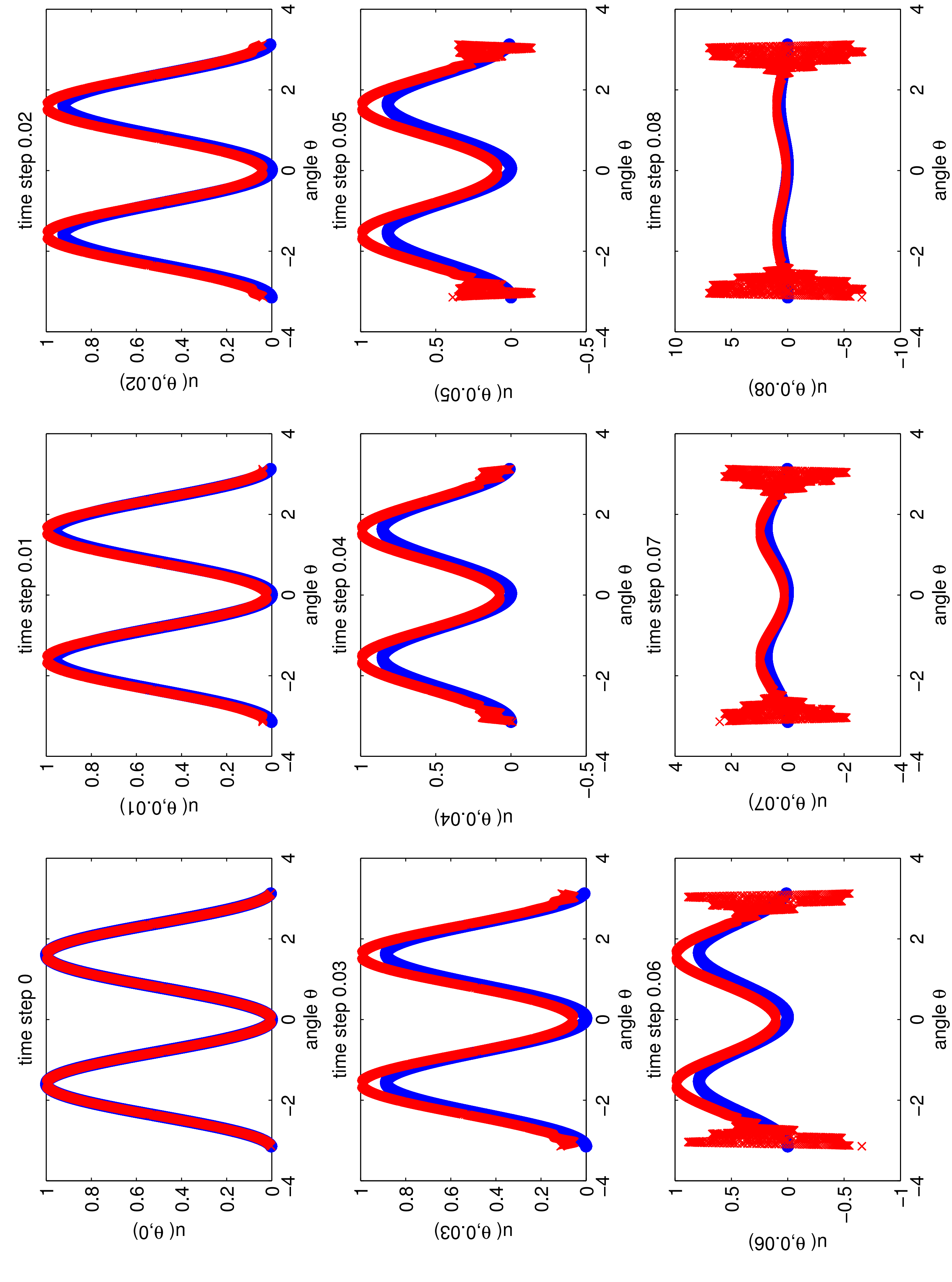


Figure 22: The effects of interpolation error in contrast with the TVD property of equation (98) with $\Delta t = 0.01$ and $\Delta x = 0.1$.

4 Conclusions and further developments

In this dissertation the closest point method has been explained in detail and applied to a specific class of partial differential equations defined only on the surface of a domain. The closest point method is also called an embedding method and belongs to a higher class of numerical methods called level set methods. These in turn, have also been reviewed, together with two significant application in physics.

The design of the closest point method relies on the closest point representation of the surface, which is based on the signed distance function. The implicit representation of a surface (smooth or polygonal) gives the appropriate setting for deriving the appropriate transformation (mapping) between all the points in the domain and the desired surface. Complex geometries can be handled by using the signed distance function, but other problems appear (even in the case of the ellipse), and a relevant analysis of the interpolation polynomial must be made. The technique of banding can be applied but interpolation or time-stepping problems still remain, since it is only a programming technique to optimize the computational overhead.

Numerical experiments have also been made. Examples have been given for the derivation of the surface PDE and the closest point transformation (mapping) in the case of the unit circle and the ellipse. The extension to other geometries (shapes of the surfaces) is straightforward ***while*** a suitable implicit representation can be made. Taking for example the Cardioid, the Cartesian representation is given by $(x^2+y^2-a^2)^2-4a^2((x-a)^2+y^2) = 0$, hence an algebraic manipulation must be made in order to obtain a valid closest point representation based on the associated signed distance function. Even a more complex example can be given, in which one can anticipate significant problems due to singularities. Consider the Deltoid curve with the Cartesian representation given by $(x^2+y^2)^2+18a^2(x^2+y^2)-27a^4 = 8a(x^3-3xy^2)$. In this case also, the signed distance function must also be used to obtain a valid representation. Since the closest point method relies into generating the desired shape by using every point from the embedded domain, the difficulties described for the two examples given above can not be ignored.

If a polygonal surface is chosen, the closest point representation ceases to be analytic and a least-squares problem must be implemented and solved.

On the other hand, the simplicity of the closest point method gives the possibility for improvement. The internal stages of the method, the interpolation and the time-stepping, can be improved to produce more accurate results.

In the case of the interpolation stage, WENO based interpolation has already been applied to a three-dimensional problem in [MR08]. Depending on the underlying geometry (in the sense of smoothness) other building blocks for interpolants can be chosen. To mention only a few, the convex ENO [LL07] or the trigonometric WENO [ZQ10]. The traditional Legendre, Chebyshev, Jacobi, Hermite and Laguerre polynomials can also be used as the basis for the blending function and a new interpolant can be derived for the multivariate case.

In the case of the time-stepping stage, the strong stability preserving approach is preferred, since there is a control of oscillations, given by the total variation diminishing property. A 8 stage SSP Runge-Kutta method has been obtained in [GST01]. The implicit approach has also been investigated in [MR09], but the overall method becomes more difficult because a stabilization technique must be applied. The review for high-order finite difference schemes given in [CJST98] provide a starting point for developing a new and more accurate time-stepping method.

In most of articles or books cited in this dissertation, only deterministic partial differential equations have been used. The stochastic approach can also be applied. The following is a simple derivation from Cartesian coordinates to polar coordinates in the sense of stochastic differential equations (the Fokker-Plank equation):

$$d\mathbf{X_t} = \underbrace{\mathbf{W_t}}_{\text{Wiener process}} \quad \Leftrightarrow \quad \underbrace{\frac{\partial f}{\partial t} = \frac{1}{2}\left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}\right)}_{\text{The Fokker-Plank equation with 0 drift and 1/2 diffusion}} \tag{99}$$

Transforming from Cartesian to polar coordinates ($x = R\cos(\theta), y = R\sin(\theta)$) and taking the surface as the unit circle (R=1), the stochastic differential equation (99) becomes:

$$\frac{\partial f}{\partial t} = \frac{1}{2}\left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}\right) \Leftrightarrow \frac{\partial f}{\partial t} = \frac{1}{2}\left(\underbrace{\frac{\partial^2 f}{\partial r^2}}_{=0} + \underbrace{\frac{1}{r}\frac{\partial f}{\partial r}}_{=0} + \underbrace{\frac{1}{r^2}}_{=1}\frac{\partial f^2}{\partial \theta^2}\right) = \frac{1}{2}\left(\frac{\partial f^2}{\partial \theta^2}\right) \tag{100}$$

Using the archlength relation $s = \theta$ (since $R = 1$), equation (100) becomes a surface stochastic differential equation defined only on the boundary of the unit circle:

$$\frac{\partial f}{\partial t} = \frac{1}{2}\left(\frac{\partial f^2}{\partial \theta^2}\right) = \frac{1}{2}\left(\frac{\partial f^2}{\partial s^2}\right) \tag{101}$$

An initial condition must be prescribed to equation (101) (for example, the standard delta function) together with appropriate boundary conditions. In this case specialized boundary conditions can be imposed, for example reflective boundary conditions and/or even attach a first passage escape problem. The time-stepping is accomplished by an appropriate Brownian dynamics numerical method. The extension (interpolation) stage should be done after each stage of the time-stepping scheme and in this way mimic the behavior related to the SSP Runge-Kutta type methods.

References

[AS96] Luigi Ambrosio and Halil Mete Soner, *Level set approach to mean curvature flow in arbitrary codimension*, J. Differential Geom. **43** (1996), no. 4, 693–737. MR 1412682 (97k:58047)

[BCF51a] W. K. Burton, N. Cabrera, and F. C. Frank, *The growth of crystals and the equilibrium structure of their surfaces*, Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences **243** (1951), no. 866, pp. 299–358 (English).

[BCF51b] ———, *The growth of crystals and the equilibrium structure of their surfaces*, Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences **243** (1951), no. 866, 299–358.

[BCOS01] Marcelo Bertalmio, Li-Tien Cheng, Stanley Osher, and Guillermo Sapiro, *Variational problems and partial differential equations on implicit surfaces*, Journal of Computational Physics **174** (2001), no. 2, 759 – 780.

[BKZ92] J. U. Brackbill, D. B. Kothe, and C. Zemach, *A continuum method for modeling surface tension*, J. Comput. Phys. **100** (1992), 335–354.

[CGM$^+$98] R. E. Caflisch, M. Gyure, B. Merriman, S. Osher, C. Ratsch, D. Vvedensky, and J. Zinck, *Island dynamics and the level set method for epitaxial growth.*

[CJST98] B. Cockburn, C. Johnson, C. W. Shu, and E. Tadmor, *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, Springer, 1998.

[EDD$^+$95] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle, *Multiresolution analysis of arbitrary meshes*, 173–182.

[EDG06] Ennio De Giorgi, *Ennio De Giorgi: Selected Papers*, Springer, 2006.

[EH96] Matthias Eck and Hugues Hoppe, *Automatic reconstruction of b-spline surfaces of arbitrary topological type*, Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, SIGGRAPH '96, ACM, 1996, pp. 325–334.

[EL78] J. Eells and L. Lemaire, *A report on harmonic maps*, Bulletin of the London Mathematical Society **10** (1978), no. 1, 1–68.

[EL88] J Eells and L Lemaire, *Another report on harmonic maps*, Bulletin of the London Mathematical Society **20** (1988), no. 5, 385–524.

[FH05] Michael S. Floater and Kai Hormann, *Surface parameterization: a tutorial and survey*, 157–186.

[Gor71] William J. Gordon, *Blending-function methods of bivariate and multivariate interpolation and approximation*, SIAM Journal on Numerical Analysis **8** (1971), no. 1, pp. 158–177 (English).

[Gre06] John B. Greer, *An improvement of a recent eulerian method for solving pdes on general geometries*, J. Sci. Comput. **29** (2006), 321–352.

[GRM$^+$98] Mark F. Gyure, Christian Ratsch, Barry Merriman, Russel E. Caflisch, Stanley Osher, Jennifer J. Zinck, and Dimitri D. Vvedensky, *Level-set methods for the simulation of epitaxial phenomena*, Phys. Rev. E **58** (1998), no. 6, R6927–R6930.

[GST01] Sigal Gottlieb, Chi-Wang Shu, and Eitan Tadmor, *Strong stability-preserving high-order time discretization methods*, SIAM Rev. **43** (2001), 89–112.

[Hou95] Thomas Y. Hou, *Numerical solutions to free boundary problems*, Acta Numerica **4** (1995), 335–415.

[Kar94] S. Karni, *Multicomponent flow calculations by a consistent primitive algorithm*, J. Comput. Phys. **112** (1994), 31–43.

[Kar96] ———, *Hybrid multifluid algorithms*, no. 5, 1019–1039.

[Kob93] Ryo Kobayashi, *Modeling and numerical simulations of dendritic crystal growth*, Physica D: Nonlinear Phenomena **63** (1993), no. 3-4, 410 – 423.

[kZCMO96] Hong kai Zhao, T. Chan, B. Merriman, and S. Osher, *A variational level set approach to multiphase motion.*

[LL07] Chi-Tien Lin and Xu-Dong Liu, *Convex eno schemes for hamilton-jacobi equations*, J. Sci. Comput. **31** (2007), 195–211.

[Mau03] Sean Mauch, *Efficient algorithms for solving static hamilton-jacobi equations.*

[MB11] Robin McLeod and Louisa Baart, *Geometry and interpolation of curves and surfaces*, Cambridge University Press, 2011.

[MBO94] Barry Merriman, James K. Bence, and Stanley J. Osher, *Motion of multiple junctions: a level set approach*, J. Comput. Phys. **112** (1994), 334–363.

[MBR11] Colin B. Macdonald, Jeremy Brandman, and Steven J. Ruuth, *Solving eigenvalue problems on curved surfaces using the Closest Point Method*, To appear in Journal of Computational Physics.

[MC04] T.G. Myers and J.P.F. Charpin, *A mathematical model for atmospheric ice accretion and water flow on a cold surface*, International Journal of Heat and Mass Transfer **47** (2004), no. 25, 5483 – 5500.

[MCC02] T. G. Myers, J. P. F. Charpin, and S. J. Chapman, *The flow and solidification of a thin fluid film on an arbitrary three-dimensional surface*, no. 8, 2788–2803.

[MCO98] Barry Merriman, Russel Caflisch, and Stanley Osher, *Level set methods, with an application to modeling the growth of thin films*, 51–70.

[MM10] Giuseppe Mastroianni and Gradimir Milovanovic, *Interpolation processes: Basic theory and applications*, Springer, 2010.

[MOS92] W. Mulder, S. Osher, and James A. Sethian, *Computing interface motion in compressible gas dynamics*, J. Comput. Phys. **100** (1992), 209–228.

[MR07] Barry Merriman and Steven J. Ruuth, *Diffusion generated motion of curves on surfaces*, Journal of Computational Physics **225** (2007), no. 2, 2267 – 2282.

[MR08] Colin B. Macdonald and Steven J. Ruuth, *Level set equations on surfaces via the Closest Point Method*, J. Sci. Comput. **35** (2008), no. 2–3, 219–240, doi:10.1007/s10915-008-9196-6.

[MR09] ———, *The implicit Closest Point Method for the numerical solution of partial differential equations on surfaces*, SIAM J. Sci. Comput. **31** (2009), no. 6, 4330–4350, doi:10.1137/080740003.

[NPV91] R. H. Nochetto, M. Paolini, and C. Verdi, *An adaptive finite element method for two-phase stefan problems in two space dimensions. part ii.: implementation and numerical experiments*, SIAM J. Sci. Stat. Comput. **12** (1991), 1207–1244.

[OM97] S. Osher and B. Merriman, *The wulff shape as the asymptotic limit of a growing crystalline interface*, Asian J. Math **1** (1997), 560–571.

[OPB$^+$03] Stanley Osher, Nikos Paragios, Marcelo Bertalmio, Facundo Memoli, Li-Tien Cheng, Guillermo Sapiro, and Stanley Osher, *Variational problems and partial differential equations on implicit surfaces: Bye bye triangulated surfaces?*, Geometric Level Set Methods in Imaging, Vision, and Graphics, Springer New York, 2003, pp. 381–397.

[OS88] Stanley Osher and James A. Sethian, *Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations*, JOURNAL OF COMPUTATIONAL PHYSICS **79** (1988), no. 1, 12–49.

[Phi03] George M. Phillips, *Interpolation and approximation by polynomials*, Springer, 2003.

[PLBS01] Burchard P., Cheng L.T., Merriman B., and Osher S., *Motion of curves in three spatial dimensions using a level set approach*, Journal of Computational Physics **170** (2001), no. 2, 720–741.

[POMZ98] D. Peng, S. Osher, B. Merriman, and H. K. Zhao, *The geometry of wulff crystal shapes and its relations with riemann problems*, 251–303.

[PW99] P. Perona and A. Why, *Orientation diffusions*, IEEE Trans. Image Processing **7** (1999), 457–467.

[RM08] Steven J. Ruuth and Barry Merriman, *A simple embedding method for solving partial differential equations on surfaces*, Journal of Computational Physics **227** (2008), no. 3, 1943 – 1961.

[RMO99] Steven J. Ruuth, Barry Merriman, and Stanley Osher, *A fixed grid method for capturing the motion of self-intersecting interfaces and related pdes.*

[RSS03] G Russo, L M Sander, and P Smereka, *A hybrid monte carlo method for surface growth simulations*, Physical Review B **69** (2003), no. 12, 4.

[Sch99a] K. W. Schwarz, *Simulation of dislocations on the mesoscopic scale. i. methods and examples*, no. 1, 108–119.

[Sch99b] ______, *Simulation of dislocations on the mesoscopic scale. ii. application to strained-layer relaxation*, no. 1, 120–129.

[SFW00] John Steinhoff, Meng Fan, and Lesong Wang, *A new eulerian method for the computation of propagating short acoustic and electromagnetic pulses*, J. Comput. Phys. **157** (2000), 683–706.

[Shu88] Chi-Wang Shu, *Total-variation-diminishing time discretizations*, SIAM J. Sci. Stat. Comput. **9** (1988), 1073–1084.

[SO88] Chi-Wang Shu and Stanley Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, Journal of Computational Physics **77** (1988), no. 2, 439 – 471.

[Sor94] Pierpaolo Soravia, *Generalized motion of a front propagating along its normal direction: a differential games approach*, Nonlinear Anal. **22** (1994), 1247–1262.

[SSD05] David Shafrir, Nir Sochen, and Rachid Deriche, *Regularization of mappings between implicit manifolds of arbitrary dimension and codimension*, Variational, Geometric, and Level Set Methods in Computer Vision (Nikos Paragios, Olivier Faugeras, Tony Chan, and Christoph Schnorr, eds.), Lecture Notes in Computer Science, vol. 3752, Springer Berlin - Heidelberg, 2005, pp. 344–355.

[Str99] John Strain, *Fast tree-based redistancing for level set computations.*

[TMR09] Li (Luke) Tian, Colin B. Macdonald, and Steven J. Ruuth, *Segmentation on surfaces with the Closest Point Method*, Proc. ICIP09, 16th IEEE International Conference on Image Processing (Cairo, Egypt), 2009, doi:10.1109/ICIP.2009.5414447, pp. 3009–3012.

[TSC00] Bei Tang, Guillermo Sapiro, and Vicent Caselles, *Diffusion of general data on non-flat manifolds via harmonic maps theory: The direction diffusion case*, Journal Computer Vision **36** (2000), 149–161.

[UT92] S. O. Unverdi and G. Tryggvason, *A front-tracking method for viscous, incompressible, multi-fluid flows*, J. Comput. Phys. **100** (1992), 25–37.

[WK91] Andrew Witkin and Michael Kass, *Reaction-diffusion textures*, 299–308.

[ZMOW98] Hong-Kai Zhao, Barry Merriman, Stanley Osher, and Lihe Wang, *Capturing the behavior of bubbles and drops using the variational level set approach*.

[ZQ10] J. Zhu and J. Qiu, *Trigonometric weno schemes for hyperbolic conservation laws and highly oscillatory problems*, Communications in Computational Physics **8** (2010), 1242–1263.