國立交通大學

應用數學系

碩 士 論 文

隱式最近點方法求解在變動曲面上的對流擴散方程

An implicit closest point method for solving convection-diffusion equations on a moving surface

研 究 生：成澤仕軒

指導教授：賴明治　教授

中 華 民 國 一百零二 年 六 月

隱式最近點方法求解在變動曲面上的對流擴散方程

An implicit closest point method for solving convection-diffusion equations on a moving surface

研 究 生：成澤仕軒　　Student：Narisawa Shinoki

指導教授：賴明治　　Advisor：Ming-Chih Lai

國 立 交 通 大 學

資 訊 科 學 系

碩 士 論 文

A Thesis

Submitted to Department of Applied Mathematics

College of Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Applied Mathematics

June 2013

Hsinchu, Taiwan, Republic of China

中華民國一百零二年六月

隱式最近點方法求解在變動曲面上的對流擴散方程

學生：成澤仕軒　　指導教授：賴明治

國立交通大學應用數學系（研究所）碩士班

摘　要

本文提出一個數值計算方法去求解變動曲面上的對流擴散方程。利用水平集函數捕捉變動曲面。根據最近點方法，利用最近點將對流擴散方程延拓到曲面附近的小區域，並且在這小區域上用 Crank-Nichoson 方法求解嵌入方程。

An implicit closest point method for solving convection-diffusion equations on a moving surface

student：Narisawa Shinoki　　Advisors：Ming-Chih Lai

Department（Institute）of Applied Mathematics
National Chiao Tung University

ABSTRACT

We propose a numerical method to solving convection-diffusion equation on a moving surface. We use the level set function to capture the deforming surface. Based on the closest point method, we extend the convection-diffusion equation into a small neighborhood of the surface by closest point, and use Crank-Nicolson scheme to solving the embedding PDE on the neighborhood of the surface.

誌　　謝

本論文的完成，首先感謝我的指導教授賴明治老師。在教授的指導下，讓我逐漸熟悉數值偏微分方程及科學計算方法。使我在數值方法及科學計算領域中，找到我的興趣以及研究做出的成就感。老師不只引導我學習應有的態度，並且教導我們做人的道理，相信往後的人生道路上將會受用無窮。並感謝胡偉帆學長與陳冠羽學長，教導我許多知識，在學業上對我有極大的幫忙。

感謝曾經教導過我的教授，你們不僅傳授我數學上的知識，還有學習應有的態度。在論文口試期間，承蒙吳金典教授與黃聰明教授費心審閱並提供許多寶貴的意見，使得本篇論文更加完備，學生銘記在心。

感謝我周遭所有同學兩年來的相處，一起學習，一起玩樂，一起成長。在我低潮的時候，需要幫忙的時候，總是有你們陪伴。我不會的東西很多，許多事情都是你們教我的，感謝有你們的陪伴與包容。你們讓我研究生涯過的多采多姿，充滿樂趣。謝謝你們。

最後，我要感謝我的父母，有你們的支持，我才能順利地完成學業，你們是我最堅強的後盾，讓我無憂無慮的學習。願與所有關心我的人，一同分享此篇論文完成之喜悅與榮耀。

目　錄

1 Introduction

Solving PDEs on a moving and deforming surface is important. There are many applications in fluid dynamics, material science, and the mathematics of images. For example, the convection-diffusion equation on a moving surface can be used to describe the surfactant which is convected by flow and diffused along the surface. It is important for the immiscible fluids problem which has surfactant on the interface between different fluids. The surface tension depends on the surfactant concentration, and it will effect the flow around the interface and deform the interface. To solve this problem, we need to capture the deforming surface. In the literature, there are some methods to capture a deforming surface, including front-tracking method[12], volume-of-fluid method[13], level set method[7], Arbitrary Lagrange-Euler method[14],etc. For solving PDEs on a moving surface, Xu et al. [7] developed a level set method to solve convection-diffusion equation on a moving surface. Dziuk and Elliott [5] solved the same equation on evolving surface by finite element method.

In [1], the authors provided an embedding method to solve the surface PDEs on a surface of fixed shape. The method extend the PDE from the surface to $\mathbb{R}^n$ by closest points, and solve the embedding PDE by finite difference method. Here we use the implicit closest point method to solve the convection-diffusion equations on a deforming surface and use the level set method to capture our surface. In section 2, we describe the mathematical formulation for the problem. In section 3, we introduce the closest point method and its analysis. In section 4, we give the representation for moving surface by level set function. In section 5, we develop the numerical algorithms in detail. In section 6, we present the numerical results.

2 Surfactant concentration equation

Consider a surface $\Sigma(t)$, which is deformable and moves with the incompressible velocity $\mathbf{u}$ as

$$\frac{d\mathbf{x}}{dt} = \mathbf{u} \ , \ \mathbf{x} \in \Sigma(t), \tag{1}$$

Let f denote the mass of the surfactant per unit area defined on the surface and the surfactant remains on the surface and just convects and diffuses along the surface. Based on the conservation of total mass, we obtain the convection-diffusion equation for the surfactant in [8,9,10] given by

$$f_t + \mathbf{u} \cdot \nabla_s f + (\nabla_s \cdot \mathbf{u}) f = \frac{1}{Pe_s} \triangle_s f \tag{2}$$

where ∇_s is surface gradient , $\nabla_s \cdot$ is surface divergence , $\triangle_s$ is surface Laplacian and Pe_s is the Peclet number. The surface gradient is a operator which describes the changing rate of function only along the tangent direction. Let $\mathbf{n}$ be normal vector of the surface. The surface gradient operator can be written as

$$\nabla_s f = \nabla f - \mathbf{n}(\nabla f \cdot \mathbf{n}).$$

The surface divergence represents the value of outward flow along the surface. The surface divergence operator acting on $\mathbf{v}$ can be written as

$$\nabla_s \cdot \mathbf{v} = \nabla \cdot \mathbf{v} - \mathbf{n}^t \cdot \nabla \mathbf{v} \cdot \mathbf{n}$$

and surface Laplace $\Delta_s = \nabla_s \cdot \nabla_s$

3 The closest point method

In this section, we review the closest point method in [1]. The closest point method is a simple embedding method for solving PDEs on the surfaces. The main idea is to construct a embedding PDE on $\mathbb{R}^d$ which is a normal extension of the surface PDE. The embedding PDE involves only the standard Cartesian differential operators and same as surface PDE on the surface, so that we can easily solve the embedding PDE by the finite difference schemes on the regular grid points and approximate the solution of surface PDE by the solution which we solve from embedding PDE.

In order to analyze the closest point method, we state the two fundamental properties for the surface operator:

1. Suppose F is any function defined on $\mathbb{R}^d$ that is constant along the normal direction to the surface, then $\nabla F \cdot \mathbf{n} = 0$ at the surface. From the definition of surface gradient, we have

   $$\nabla F = \nabla_s F$$

   at the surface.

2. For any vector field $\mathbf{v}$ on $\mathbb{R}^d$ that is tangent to Σ, and also tangent to all surface displaced by a fixed distance from Σ(i.e., all surfaces defined as level-sets of the distance function to Σ), then at the surface

   $$\nabla \cdot \mathbf{v} = \nabla_s \cdot \mathbf{v}$$

Details of the proof of property 2 can be found in appendix.

These are obvious statements. For the first property, a function which is constant in the normal direction only varies along the surface. The second property says that a velocity field which is directed along the surface can only spread out within the surface direction.

Consider a general prototype for a PDE describing some physical process on the surface Σ in the form

$$\frac{\partial f}{\partial t}(y) = G(y, f(y), \nabla_s f(y), \triangle_s f(y)) \quad \forall y \in \Sigma. \tag{3}$$

For any x in space, let $cp(x)$ denote the closest point to x in the surface Σ, that means $|x - cp(x)| = \inf_{y \in \Sigma} |x - y|$. Figure 1 shows an example about relation of x and $cp(x)$. Here we note that $cp(x)$ is well-defined if x is in a sufficiently small neighborhood of a smooth surface.

Now, we define an embedding PDE on Ω which is the neighborhood of the surface by closest point and replace surface operators by standard Cartesian operators, as

$$\frac{\partial F}{\partial t}(x) = G(cp(x), F(cp(x)), \nabla F(cp(x)), \triangle F(cp(x))) \quad \forall x \in \Omega. \tag{4}$$

Since the vector $x - cp(x)$ is normal to the surface Σ, the function $F(cp(x))$ is a constant normal extension from the surface. If $x \in \Sigma$, the first property implies that

$$\nabla F(cp(x)) = \nabla_s F(cp(x)) = \nabla_s F(x)).$$

Moreover, $\nabla F(cp(x))$ is always tangent to the level-sets of the distance function, so applying the second property at the surface, we obtain

$$\nabla \cdot (\nabla F(cp(x))) = \nabla_s \cdot (\nabla F(cp(x)) = \nabla_s \cdot (\nabla_s F(x)).$$

According to the two properties, the embedding PDE (4) is equal to the surface PDE (3) on the surface. It means that, if F is a solution of embedding PDE , then the solution on the surface also satisfies surface PDE. Based on this, the closest point method proceeds by the following two steps:

1. Extend the solution off the surface to each grid node on the computational domain around the surface.

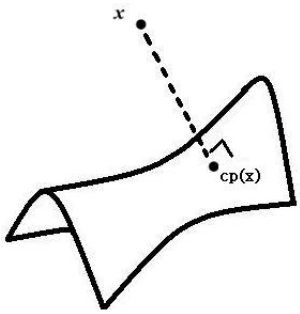


Figure 1: x is any point in space and $cp(x)$ is closest point in surface of x

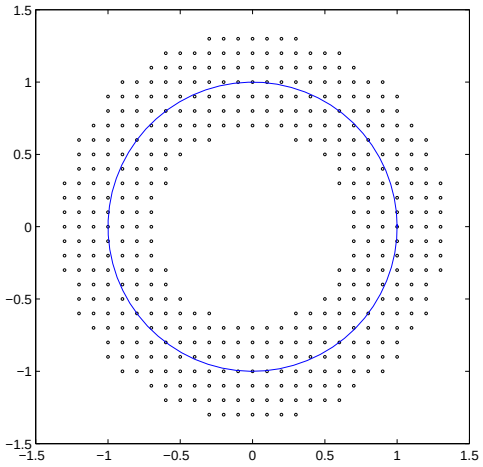


Figure 2: The function F which we solved from embedding PDE is defined on '$\cdot$' around of surface, and we use F to approximate surface function f, which is solution for surface PDE.

2. Compute the solution of the embedding PDE using standard finite differences scheme on a Cartesian mesh of the computation domain.

At every time step, the solution of the surface PDE is approximated by the solution of the embedding PDE at the grid points on the computation domain. For example, we use the forward Euler method to solve the heat equation on the surface

$$f^{n+1} = f^n + \Delta t \cdot \triangle_s f^n .$$

We do not treat this surface equation directly. Instead, we assume $F^n(x) = f^n(cp(x))$ and evolve the equation

$$F^{n+1} = F^n + \Delta t \cdot \triangle F^n$$

on the regular grid points around of surface, and use F^{n+1} to approximate the solution f^{n+1} on the surface.

4 Level set representation

Following [6,7], we represent a moving surface $\Sigma(t)$ in $\mathbb{R}^d$ by zero level set of the signed distance function $\phi(x,t)$. That is

$$\Sigma(t) = \{x : \phi(x,t) = 0\} \tag{5}$$

where $\phi(x,t)$ defined by

$$\phi(x,t) = \begin{cases} -d(x,t) & \text{if } x \in \Sigma(t)^- \\ 0 & \text{if } x \in \Sigma(t) \\ d(x,t) & \text{if } x \in \Sigma(t)^+ \end{cases}$$

where $d(x,t) = \inf_{y \in \Sigma(t)} |x-y|$ is the distance function , $\Sigma(t)^-$ and $\Sigma(t)^+$ represent inside and outside of the surface, respectively. For example, consider a circle surface, the signed distance function of the circle is $\phi = \sqrt{x^2+y^2} - 1$.

However, it is difficult to give an explicit formula for the signed distance function of any surface. As in [6], we reinitialize the level set function ϕ_0 to the signed distance function ϕ by evolving the reinitialization equation

$$\begin{cases} \phi_\tau + S(\phi_0)(|\nabla\phi| - 1) = 0 \\ \phi(\mathbf{x}, 0) = \phi_0 \end{cases} \tag{6}$$

where τ is pseudo time and $S(x)$ is the signed function taken as 1 in $\Sigma(t)^+$, -1 in $\Sigma(t)^-$ and 0 on the surface. By evolving this equation, the function ϕ will be identically equal to zero on Σ, and $|\nabla\phi|$ will converge to 1 on the narrowband of the surface. That means the level set function $\phi(t)$ will converge to the signed distance function. Figure 3 shows the level set function of an ellipse transformed to the signed distance function after reinitialization precess.

By taking the time derivative of $\phi(x,t) = 0$, then we have the Hamilton-Jacobi equation

$$\frac{D\phi}{Dt} = \phi_t + \mathbf{u} \cdot \nabla\phi = 0 \tag{7}$$

where $\mathbf{u}$ is the surface velocity. Since the surface is represented by the zero level set of $\phi(x,t)$, we can move the surface by solving the equation (7) for a given initial condition $\phi(x,0)$.

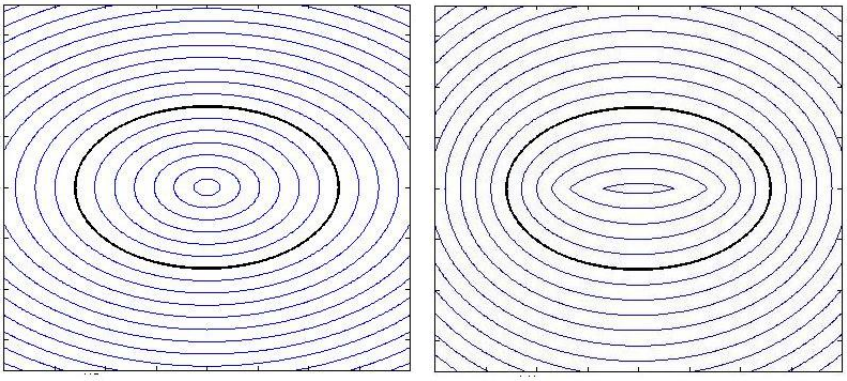


Figure 3: The left figure is the level set of the function $\phi_0 = \frac{x^2}{(1.2)^2} + \frac{y^2}{(0.8)^2} - 1$ to represent an ellipse. The right figure is level set of ϕ which is reinitialized of ϕ_0

5 Numerical method

Based on the closest point method [4,1], we use an implicit scheme to solve the embedding PDE and interpolate the value on the surface by degree-p Lagrange interpolation polynomials. Given the surface function f^n, the signed distance function ϕ^n and the vector field $\mathbf{u}$, we evolve f^{n+1} and ϕ^{n+1} by the following steps:

Step1: Evolve the surface $\Sigma(t_{n+1})$ which represented by zero level set of ϕ_0^{n+1} in velocity field u. We solve the Hamilton-Jacobi equation (7) for one time step to move our surface. For spatial discretization, we use the third-order upwind weighted essentially non-oscillatory(WENO) scheme [7] to discretize $\mathbf{u} \cdot \nabla \phi^n$. At each grid point (x_i, y_j)

$$(\mathbf{u} \cdot \nabla \phi)_{ij} = u^+ D_x^- \phi_{ij} + u^- D_x^+ \phi_{ij} + v^+ D_y^- \phi_{ij} + v^- D^+ y \phi_{ij}$$

where $x^+ = \max(x, 0)$, $x^- = \min(x, 0)$ and $D_x^{\pm}\phi_{ij}$, $D_y^{\pm}\phi_{ij}$ are the one-sided divided differences for the WENO scheme. In third-order WENO scheme [2], the approximation to $\frac{\partial \phi}{\partial x}$ on left-biased stencil $\{x_{i-2}, x_{i-1}, x_i, x_{i+1}\}$ is

$$D_x^- \phi_i = \frac{1}{2\Delta x}\left[(\Delta^+\phi_{i-1} + \Delta^+\phi_i) - \omega_-(\Delta^+\phi_{i-2} - 2\Delta^+\phi_{i-1} + \Delta^+\phi_{i-1})\right]$$

where

$$\omega_- = \frac{1}{1+2r^2} \qquad \text{with} \qquad r = \frac{\varepsilon + (\Delta^+\Delta^-\phi_{i-1})^2}{\varepsilon + (\Delta^+\Delta^-\phi_i)^2}.$$

where the notation $\Delta^+\phi_i = \phi_{i+1} - \phi_i$, $\Delta^-\phi_i = \phi_i - \phi_{i-1}$ are forward and backward difference operators respectively. By symmetry, the approximation to $\frac{\partial\phi}{\partial x}$ on right-biased stencil $\{x_{i-1}, x_i, x_{i+1}, x_{i+2}\}$ is

$$D_x^+\phi_i = \frac{1}{2\Delta x}\left[(\Delta^+\phi_{i-1} + \Delta^+\phi_i) - \omega_+(\Delta^+\phi_{i+1} - 2\Delta^+\phi_i + \Delta^+\phi_{i-1})\right]$$

where

$$\omega_+ = \frac{1}{1+2r^2} \qquad \text{with} \qquad r = \frac{\varepsilon + (\Delta^+\Delta^-\phi_{i+1})^2}{\varepsilon + (\Delta^+\Delta^-\phi_i)^2}.$$

For time discretization, we use the third-order total variation diminishing(TVD) Runge-Kutta scheme in [11]. Consider the time dependent PDE

$$\frac{d\phi}{dt} = H(\phi),$$

We solve the level set function ϕ_0 at the (n+1)th step by

$$\begin{cases} \phi_1 = \phi^n + \Delta t H(\phi^n) \\ \phi_2 = \frac{3}{4}\phi^n + \frac{1}{4}\phi_1 + \frac{\Delta t}{4}H(\phi_1) \\ \phi_0^{n+1} = \frac{1}{3}\phi^n + \frac{2}{3}\phi_2 + \frac{2\Delta t}{3}H(\phi_2) \end{cases}$$

Step2: We reinitialize the level set function ϕ_0^{n+1} to the signed distance function ϕ^{n+1} by evolving the reinitialization equation (6) with initial condition ϕ_0^{n+1}. For time discretization, we also use the third-order total variation diminishing(TVD) Runge-Kutta scheme. For spatial discretization, following in [7] ,we use the spatial discretization for $S(\phi_0)(|\nabla\phi| - 1)$ like WENO scheme

$$\begin{aligned} S(\phi_0)(|\nabla\phi| - 1)_{ij} = {} & s_{ij}^+(\sqrt{(a^+)^2 + (b^-)^2 + (c^+)^2 + (d^-)^2} - 1) \\ & + s_{ij}^-(\sqrt{(a^-)^2 + (b^+)^2 + (c^-)^2 + (d^+)^2} - 1) \end{aligned}$$

where s_{ij} is numerical approximation to signed function of ϕ_0^{n+1} given by

$$s_{ij} = \frac{\phi_0}{\sqrt{\phi_0^2 + \Delta x^2}}$$

and notation a, b, c, d are one-sided divided differences for the WENO scheme as

$$a = D_x^-\phi_{ij} \quad , \quad b = D_x^+\phi_{ij} \quad , \quad c = D_y^-\phi_{ij} \quad , \quad d = D_y^+\phi_{ij}$$

Step3: In this step, we need to find the closest point $cp(x)$, the computational domain Ω_1 and Ω_2 at time step t_{n+1}. Since ϕ is the signed distance function, $\nabla\phi$ will be the unit normal vector $\mathbf{n}$ at surface for each point at surface. So given any point x around of the surface, the closest point $cp(x)$ of x can be written as

$$cp(x) = x - \phi(x)\nabla\phi(cp(x)).$$

Assume that the surface is sufficiently smooth and x is close to the surface, then we have $\nabla\phi(x) = \nabla\phi(cp(x))$. So that, we can approximate the closest point $cp(x)$ of x by

$$cp(x) = x - \phi(x)\nabla\phi(x).$$

Next, we try to find the computation domain Ω_1 which we will solve the embedding on this domain. Note that, the computation domain Ω_1 should contain every grid point which can appear in the interpolation stencil for some point on surface $\Sigma(t_{n+1})$, since we use the solution of embedding PDE defined on Ω_1 to approximate the solution of surface PDE on surface. In [1], the authors provided a way to approximate the computation domain Ω_1 by

$$\Omega_1 = \{x : |x - cp(x)| \leq \lambda\}$$

where the bandwidth λ is given by

$$\lambda = \sqrt{(d-1)\left(\frac{p+1}{2}\right)^2 + \left(1 + \frac{p+1}{2}\right)^2}\Delta x.$$

We write Ω_1 as

$$\Omega_1 = \{x_1, x_2, ..., x_{m_1}\}$$

Let Ω_2 be disjoint from Ω_1 which contains every grid point which will be used in the finite difference scheme as

$$\Omega_2 = \{x_{m_1+1}, x_{m_1+2}, ..., x_{m_1+m_2}\}$$

Figure 4 shows an example of the two set Ω_1 and Ω_2 on a circle with the usage of degree-4 Lagrange interpolation polynomials.

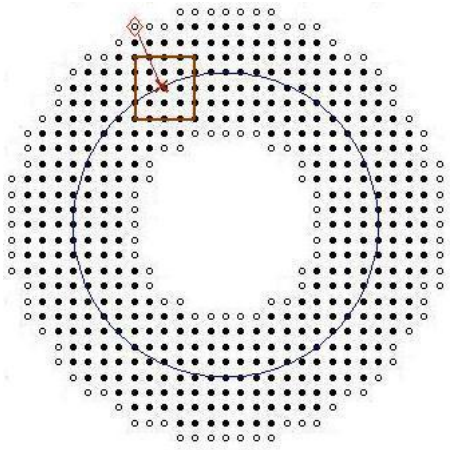


Figure 4: Example of unit circle with degree-4 Lagrange interpolation polynomials. The grid points in Ω_1 denote $\bullet$, and grid point in Ω_2 are denote $\circ$. Grid points $\times$ are closest point of $\diamond$, and the grid points in the square are interpolation stencil of $\times$

Step4: In this step, we use implicit closest point method to solve our problem. Consider the embedding PDE which is defined on Ω_1 for our convection-diffusion equation as

$$F_t(x) + \mathbf{u} \cdot \nabla F(cp(x)) + (\nabla_s \cdot \mathbf{u}) F(cp(x)) = \frac{1}{Pe_s} \triangle F(cp(x)) \tag{8}$$

Let we denote the vectors

$$\mathbf{F} = [F(x_1), F(x_2), ..., F(x_{m_1})]$$
$$\mathbf{F}_1^* = [F(cp(x_1)), F(cp(x_2)), ..., F(cp(x_{m_1}))]$$
$$\mathbf{F}_2^* = [F(cp(x_{m_1+1})), F(cp(x_{m_1+2})), ..., F(cp(x_{m_1+m_2}))]$$

We discretize the differential operators of (8) by central difference. Since $\nabla_s \cdot \mathbf{u} = \nabla \cdot \mathbf{u} - (\nabla \phi)^t \cdot \nabla \mathbf{u} \cdot \nabla \phi$. , we can write formula (8) into a matrix form

$$\frac{\partial}{\partial t} \mathbf{F} = K \begin{pmatrix} \mathbf{F}_1^* \\ \mathbf{F}_2^* \end{pmatrix} \tag{9}$$

where K is m_1 by $m_1 + m_2$ matrix.

From the closest point method, we let F^n be the normal extension from f^n of the surface $\Sigma(t_n)$, that is $F^n(x_i) = f^n(cp(x_i))$. By using the Crank-Nicolson scheme, we

have

$$I_{m_1}\mathbf{F^{n+1}} - \frac{\Delta t}{2}K\begin{pmatrix}\mathbf{F}_1^{*^{n+1}}\\ \mathbf{F}_2^{*^{n+1}}\end{pmatrix} = I_{m_1}\mathbf{F^{n}} + \frac{\Delta t}{2}K\begin{pmatrix}\mathbf{F}_1^{*^{n}}\\ \mathbf{F}_2^{*^{n}}\end{pmatrix}$$
$$= b \tag{10}$$

where I_{m_1} is an m_1 by m_1 identity matrix. Note that, we approximate the value of f^{n+1} at any point in the surface by F^{n+1}. Since $cp(x_i)$ is at the surface, we apply degree-p Lagrange interpolation polynomials and then obtain for any index $1 \le i \le m_1 + m_2$

$$F^{n+1}(cp(x_i)) = f^{n+1}(cp(x_i)) = \sum_{s=1}^{(p+1)^d} l_{i_s} F^{n+1}(x_{i_s})$$

where x_{i_s} in Ω_1 and l_{i_s} is the Lagrange coefficient for $F^{n+1}(x_{i_s})$. Rewrite it in a matrix form

$$\begin{pmatrix}\mathbf{F}_1^{*^{n+1}}\\ \mathbf{F}_2^{*^{n+1}}\end{pmatrix} = E\mathbf{F}^{n+1} \tag{11}$$

where E is a $m_1 + m_2$ by m_1 extension matrix with the entries

$$e_{ij} = \begin{cases} l_{i_s} & \text{if } x_j \text{ is in the interpolation stencil for } cp(x_i) \\ & (\text{ i.e. } j = i_s \text{ for some } 1 \le s \le (p+1)^d\ ) \\ 0 & \text{otherwise} \end{cases}$$

We let $M = KE$, and combine the two formulae (10),(11), we have

$$AF^{n+1} = b \tag{12}$$

where

$$A = I_{m_1} - \frac{\Delta t}{2}M,$$

is an m_1 by m_1 matrix.

Step5: Once we solve F^{n+1}, then we can approximate the value of f^{n+1} at surface. By examining the spectra of M given in Figure.5(left), the matrix M has some eigenvalues with positive real parts. From (9)(11), we have

$$\frac{\partial}{\partial t}\mathbf{F} = M\mathbf{F},$$

these positive eigenvalues will cause the solution an exponential growth and lead to instability. For example, Figure.6(left) shows the oscillatory results for heat equation on a circle using the implicit closest point method with M. Theoretically, it should not occur in our solution, because we have the diffusion term in our equation. So that, we should stabilize the matrix M. Let

$$\widetilde{M} = D + (K - D)E$$

where D is diagonal matrix of K. That means the stabilized form of our system is

$$\begin{aligned}\frac{\partial}{\partial t}F(x_{ij}) = &-\frac{1}{2\Delta x}\mathbf{u}\cdot(F(cp(x_{i+1,j})) - F(cp(x_{i-1,j})), F(cp(x_{i,j+1})) - F(cp(x_{i,j-1})))\\ &-(\nabla_s\cdot\mathbf{u})\,F(x_{i,j}) \qquad (13)\\ &+\frac{1}{Pe_s(\Delta x)^2}(F(cp(x_{i+1,j})) + F(cp(x_{i-1,j})) + F(cp(x_{i,j+1})) + F(cp(x_{i,j-1}))\\ &\qquad -4F(x_{i,j}))\end{aligned}$$

where the only change is that the diagonal entries. This modification will increase the value of diagonal of the matrix. Note that, the system (13) also matches the surface PDE at the surface, because $cp(x) = x$ for any x on the surface. So we can solve F^{n+1} by the system (13) and use F^{n+1} to interpolate the f^{n+1} at the surface $\Sigma(t_{n+1})$.

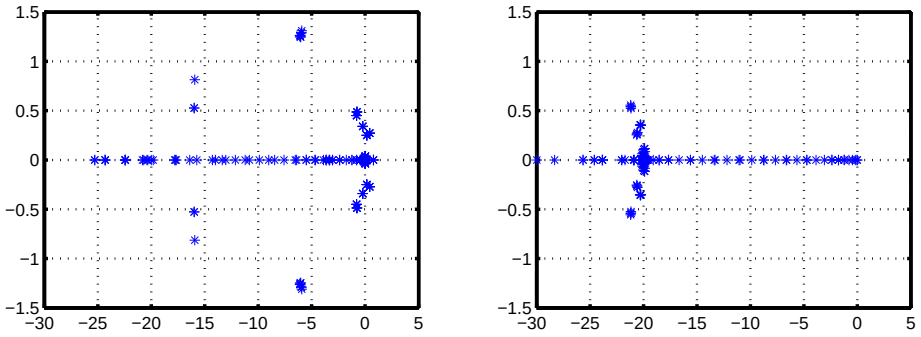


Figure 5: Spectra of the M (left) and $\widetilde{M}$ (right) matrices. The matrix M and $\widetilde{M}$ are derived from a heat equation on a unit circle in 2D with degree-4 Lagrange interpolation polynomial and $\Delta x = 0.1$. Observe that M has eigenvalues in the right half plane, and all eigenvalues of $\widetilde{M}$ have negative real parts

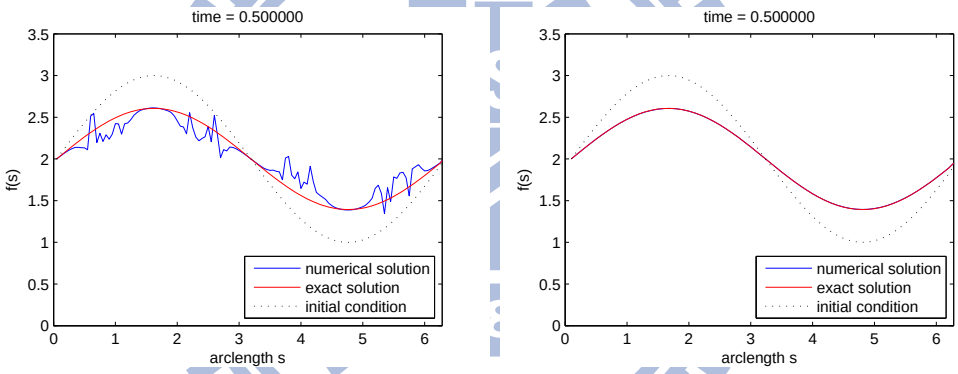


Figure 6: Stable and unstable solutions of the heat equation on a unit circle embedded in 2D with $\Delta x = 0.1$ and degree-4 interpolation

6 Numerical results

In this section, the Peclet number Pe_s is set to be 1, except that in the last example.

Example 1.

Consider the heat equation on a unit circle

$$f_t = \triangle_s f$$

with the initial condition $f_0(x, y) = \sin\theta + 2$ defined on surface where $\theta = \sin^{-1}(\frac{y}{\sqrt{x^2+y^2}})$. We know the surface Laplacian operator on the circle in polar coordinate is

$$\triangle_s f = \frac{1}{r^2}\frac{\partial^2 f}{\partial \theta^2},$$

which implies that the function $f(x, y, t) = e^{-t}\sin\theta + 2$ is the exact solution of heat equation with initial condition f_0. We apply the implicit closest point method to the problem with degree-4 Lagrange interpolation polynomials, and the closest point of (x, y) is given by $\frac{(x,y)}{\sqrt{x^2+y^2}}$. We use the time step-size $\Delta t = \Delta x = \Delta y$ and compute up to final time $T = 1$. In Table 1, the errors at the final time T are computed on Σ with infinity-norm. The results give the rate of convergence about second-order.

Example 2.

Consider an example in 3D. The heat equation on a unit sphere

$$f_t = \triangle_s f$$

Table 1: Heat equation on a circle, $\Delta t = \Delta x$, $T = 1$

Δx	m_1	Error	conv.rate
0.1	440	1.0177e-003	-
0.05	888	2.5663e-004	1.987
0.025	1768	6.3506e-005	2.015
0.0125	3568	1.5900e-005	1.998
0.00625	7128	3.9784e-006	1.999

Table 2: $\Delta t = \Delta x$, $T = 1$

Δx	m_1	Error	conv.rate
0.1	11542	2.1502e-03	-
0.0	44110	5.3535e-04	2.005
0.025	174982	1.3365e-04	2.002
0.0125	698182	3.3403e-05	2.000

with initial condition $f_0(x, y, z) = xy$ defined on surface. The surface Laplacian operator on sphere in spherical coordinate system is

$$\triangle_s f = \frac{1}{r^2 \sin\theta} \frac{\partial}{\partial \theta}(\sin\theta \frac{\partial f}{\partial \theta}) + \frac{1}{r^2 \sin^2\theta} \frac{\partial^2 f}{\partial \phi^2}$$

with $x = r\sin\theta\cos\phi$, $y = r\sin\theta\sin\phi$ and $z = r\cos\theta$. Then the function $f(x, y, z) = e^{-6t}xy$ is an exact solution of diffusion equation with initial condition f_0 on a unit sphere. The closest point of (x, y, z) is given by $\frac{(x,y,z)}{\sqrt{x^2+y^2+z^2}}$ in our method, and we compute up to final time $T = 0.5$ with time step-size $\Delta t = \Delta x = \Delta y = \Delta z$. The result is shown in Table 2. We can observe that the rate of convergence is also second-order.

Example 3.

Following the Example 1, we also use the same initial function $f_0(x, y) = \sin\theta + 2$ on a unit circle Σ where $\theta = \sin^{-1}(\frac{y}{\sqrt{x^2+y^2}})$, and we give the velocity field

$$\mathbf{u} = \frac{(-y, x)}{\sqrt{x^2 + y^2}}.$$

In the Figure 7, the velocity field $\mathbf{u}$ is always tangent to the surface Σ at any point in surface. The surface will rotate counterclockwisely but not change the shape itself. That means the level set function which we use to represent the unit circle does not change, so we do not need to solve the Hamilton-Jacobi equation (6) and (7) in our numerical process. We consider the convection-diffusion equation on Σ as

$$f_t + \mathbf{u} \cdot \nabla_s f = \triangle_s f.$$

If we use the polar coordinates, the equation on the surface Σ can be rewritten as

$$f_t + \frac{\partial f}{\partial \theta} = \frac{\partial^2 f}{\partial \theta^2}.$$

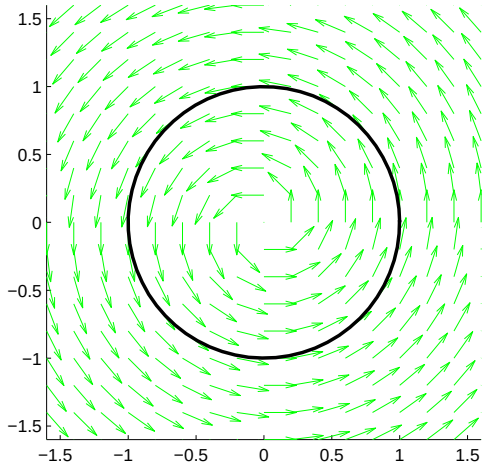


Figure 7: The black curve is unit circle and the green arrow is velocity field. The velocity is given such that the surface rotates counterclockwisely

Table 3: convection diffusion equation on a unit circle, $\Delta t = \Delta x$, $T = 1$

Δx	m_1	Error	conv.rate
0.1	440	5.2753e-004	-
0.05	888	1.3152e-004	2.004
0.025	1768	3.2961e-005	1.996
0.0125	3568	8.2431e-006	2.000
0.00625	7128	2.0611e-006	2.000

Then the function $f(x, y, t) = e^{-t} \sin(\theta - t) + 2$ is an exact solution of convection-diffusion equation with initial condition f_0. We use the time step-size $\Delta t = \Delta x = \Delta y$ and the numerical results at final time $T = 1$ are reported in Table 3.

Example 4.

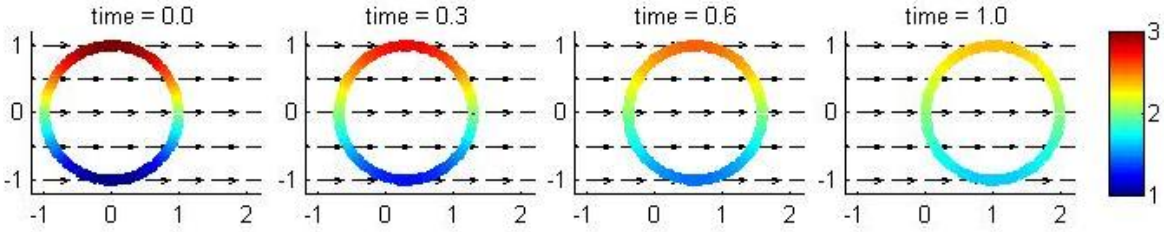
In this example, we give a simple velocity field $\mathbf{u} = (1, 0)$. The surface in the velocity field will move along x-axis with unit speed. We solve the convection-diffusion equation with the same initial function f_0 and initial surface as in Example 1. The function defined on surface

$f(x,y,t) = e^{-t}\sin(\theta) + 2$, where $\theta(t) = \sin^{-1}(\frac{y}{\sqrt{(x-t)^2+y^2}})$ is an exact solution of

$$f_t + \mathbf{u}\cdot\nabla_s f = \triangle_s f$$

with initial condition. In Table 4, we show the result with using signed distance function ϕ to represent the surface $\Sigma(t)$ and exactly closest point at each time.

Figure 8: The moving surface and surfactant concentration at different time



Example 5.

In this example, we consider the problem

$$\begin{cases} f_t + \mathbf{u}\cdot\nabla_s f + (\nabla_s\cdot\mathbf{u})f = \triangle_s f & \text{on } \Sigma(t) \\ f(x,y,0) = f_0(x,y) & \text{on } \Sigma(0) \\ \frac{d\mathbf{x}}{dt} = \mathbf{u} & \mathbf{x}\in\Sigma(t) \end{cases}$$

Let velocity field

$$\mathbf{u} = (0, \frac{x}{2})$$

be a shear flow, the surface in the flow will move and deform. The initial surface is given by the zero level set of $\phi(x,y,0) = \sqrt{(x^2+y^2)} - 1$ which is a unit circle and the initial surfactant concentration on the surface is $f_0(x,y) = \sin(\theta) + 2 = y + 2$.

Table 4: $\Delta t = \Delta x$, $T = 1$

Δx	m_1	Error	conv.rate
0.05	888	3.3504e-003	-
0.025	1768	1.5642e-003	1.10
0.0125	3568	7.5500e-004	1.05
0.00625	7128	3.7079e-004	1.03

Consider the domain of level set function to be $[-1.5, 1.5] \times [-2, 2]$. Compute the problem up to final time $T = 1$ with time step-size $\Delta t = \Delta x$ and take about 10 time steps to reinitialize the level set in our numerical process in each time step. The errors and rates of convergence are shown in Table 5. In this example, the total mass of surfactant on the surface is conserved, and the volume (area) of the interior region enclosed by the moving surface is also conserved since we use a divergence free velocity field. In [6], the total mass of surfactant f on the surface can be written as

$$M = \int_{\mathbb{R}^n} f(\mathbf{x})\delta(\phi(\mathbf{x}))|\nabla\phi(\mathbf{x})| \, d\mathbf{x},$$

and the volume (area) of the interior region is

$$V = \int_{\mathbb{R}^n} 1 - H(\phi(\mathbf{x})) \, d\mathbf{x}.$$

where H is Heaviside function and $\delta(\phi) = H'(\phi)$. We approximate the Heaviside function H and delta function δ by

$$H(\phi) = \begin{cases} 0 & \text{if } \phi < -\varepsilon \\ \frac{1}{2} + \frac{\phi}{\varepsilon} + \frac{1}{2\pi}\sin\frac{\pi\phi}{\varepsilon} & \text{if } -\varepsilon \le \phi \le \varepsilon \\ 1 & \text{if } \varepsilon \le \phi \end{cases}$$

$$\delta(\phi) = \begin{cases} 0 & \text{if } \phi < -\varepsilon \\ \frac{1}{2\varepsilon} + \frac{1}{2\varepsilon}\cos\frac{\pi\phi}{\varepsilon} & \text{if } -\varepsilon \le \phi \le \varepsilon \\ 0 & \text{if } \varepsilon \le \phi \end{cases}$$

with $\varepsilon = 1.5\Delta x$. Figure 9 shows the relative errors for total mass of surfactant on the surface and the area enclosed by the surface, respectively. The relative error for total mass of surfactant is about 10^{-3}, and the relative error of the area is near 10^{-6}. The moving surface and surfactant concentration on the surface at different times are shown in Figure 10.

Example 6.

In this example, we change the velocity field in Example 5 to

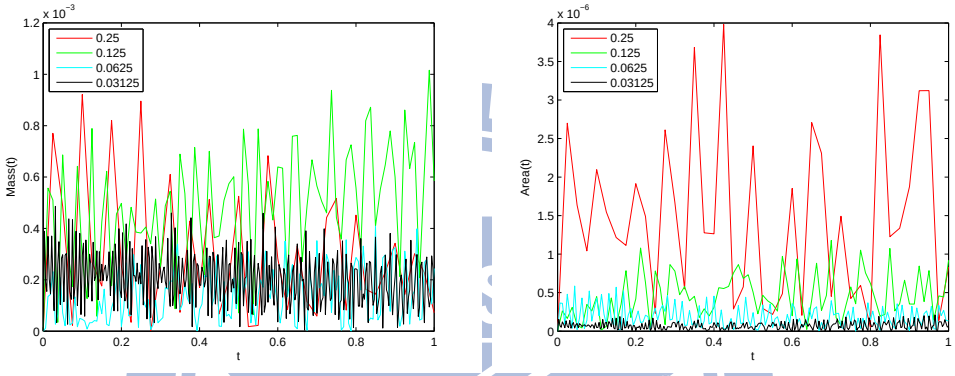
$$\mathbf{u} = \begin{cases} (y^2, 0) & \text{if } y \ge 0 \\ (-y^2, 0) & \text{if } y < 0 \end{cases}$$

which is like a shear flow, and the initial condition for surfactant concentration and surface are same as in Example 5. Consider the domain for level set function to be $[-2, 2] \times [-1.5, 1.5]$.

Table 5: Error = $\| f - f_{ref} \|_\infty$, $\Delta t = \Delta x$, $T = 1$

Δx	m_1	Error	conv.rate
0.05	924	6.4168e-003	-
0.025	1850	3.0279e-003	1.08
0.0125	3714	1.3117e-003	1.21
0.00625	7436	4.3986e-004	1.58
0.003125	14844	-	-

Figure 9: Relative error for total mass (left) and area (right) at each time



We computed the solution up to final time $T = 1$ with time step-size $\Delta t = \Delta x = \Delta y$ and take about 20 time steps to reinitialize the level set in our numerical process in each time step. In order to make sure the stability of solving the Hamilton-Jacobi equation (6), we choose the time step-size $\Delta t = \frac{\Delta x}{4}$. The result is shown in Table 6 Figures 11 shows the relative errors for total mass of surfactant on the surface and the area enclosed by the surface, respectively. The error for total mass of surfactant is still around 10^{-3}, and that of the area enclosed by surface is in the order of 10^{-6}. The moving surface and surfactant concentration on the surface at different time are shown in Figure 12.

Table 6: Error = $\| f - f_{ref} \|_\infty$, $\Delta t = \frac{\Delta x}{4}$, $T = 1$

Δx	m_1	Error	conv.rate
0.0125	4056	3.9977e-003	-
0.00625	8122	1.2875e-003	1.63
0.003125	16268	-	-

Figure 10: The moving surface and surfactant concentration at different time

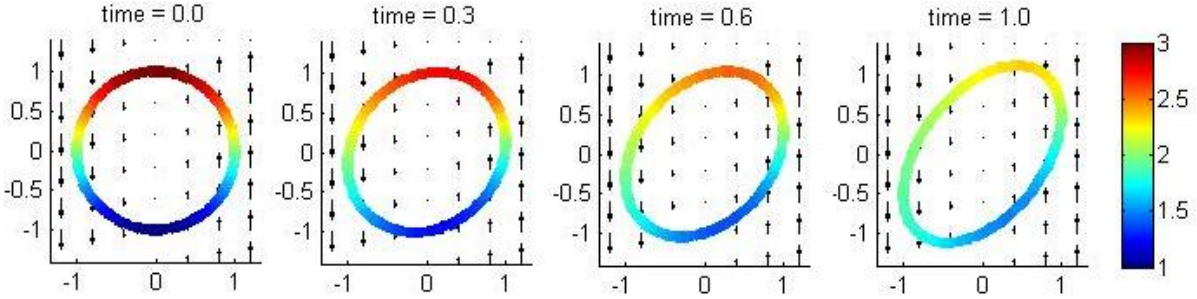
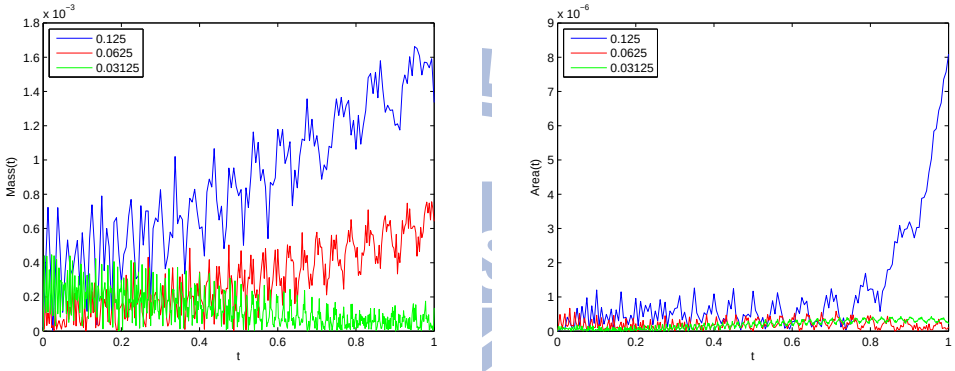


Figure 11: Relative error for total mass (left) and area (right) at each time



Example 7.

In this example, we construct a surface PDE which has an exact solution with given initial condition. Let $f(x, y, t) = e^{-t}xy + 2$ and the velocity field $\mathbf{u} = (0, \frac{x}{2})$, and we add a source term g to the convection diffusion equation which is given by

$$g = f_t + \mathbf{u} \cdot \nabla_s f + (\nabla_s \cdot \mathbf{u})f - \triangle_s f$$

Figure 12: The moving surface and surfactant concentration at different time

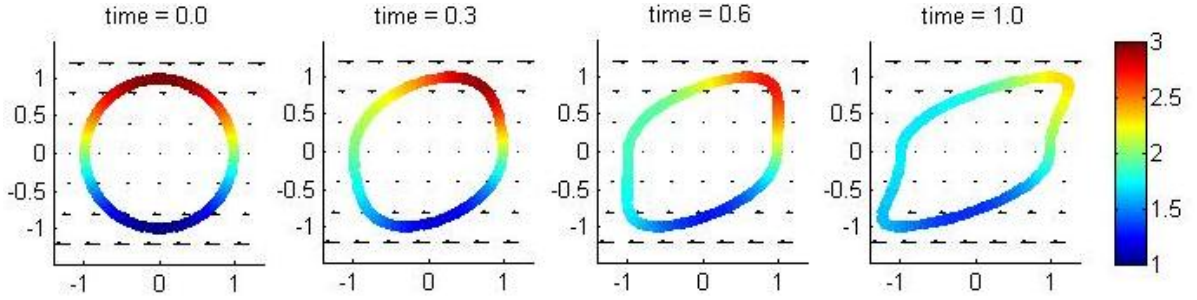


Table 7: $\Delta t = \Delta x$, $T = 1$

Δx	m_1	Error	conv.rate
0.05	924	3.6077e-003	-
0.025	1850	1.7042e-003	1.08
0.0125	3712	8.3113e-004	1.04
0.00625	7436	4.1031e-004	1.02

Consider the problem

$$\begin{cases} f_t + \mathbf{u}\cdot\nabla_s f + (\nabla_s\cdot\mathbf{u})f = \triangle_s f + g & \text{on } \Sigma(t) \\ f(x,y,0) = f_0(x,y) & \text{on } \Sigma(0) \\ \frac{d\mathbf{x}}{dt} = \mathbf{u} & \mathbf{x}\in\Sigma(t) \end{cases}$$

then f is an exact solution of the problem with initial condition $f_0(x,y) = f(x,y,0)$. In our numerical process, the embedding PDE of the problem is

$$f_t(x) + \mathbf{u}\cdot\nabla f(cp(x)) + (\nabla_s\cdot\mathbf{u})f(cp(x)) = \triangle f(cp(x)) + g(cp(x))$$

where g is compute by

$$g = f_t + \mathbf{u}\cdot(\nabla f - (\nabla f\cdot\mathbf{n})\mathbf{n}) + (\nabla_s\cdot\mathbf{u})f - (\triangle f - \mathbf{n}\cdot\nabla(\nabla f)\cdot\mathbf{n})$$

and approximate $\mathbf{n}$ by $\nabla\phi$ where ϕ is signed distance function of the surface $\Sigma(t)$. The computational domain for level set function is a rectangle $[-2,2]\times[-1.5,1.5]$. Compute the solution up to final time $T = 1$ with time step-size $\Delta t = \Delta x = \Delta y$, and take about 10 time steps to reinitialize the level set in our numerical process in every time step. The numerical result at final time T is shown in Table 7.

Example 8.

Consider a 3D example. Give a function $f(x,y,z) = xyz$ defined on a unit sphere and the velocity field $\mathbf{u} = (1,0,0)$ such that the surface in the flow will move along x-axis with unit speed. Similar to previous example, we add a source term g to the convection diffusion equation to make the problem

$$\begin{cases} f_t + \mathbf{u}\cdot\nabla_s f = \triangle_s f + g & \text{on } \Sigma(t) \\ f(x,y,0) = f_0(x,y) & \text{on } \Sigma(0) \\ \frac{d\mathbf{x}}{dt} = \mathbf{u} & \mathbf{x}\in\Sigma(t) \end{cases}$$

Table 8: $\Delta t = \Delta x$, $T = 1$

Δx	m_1	Error	conv.rate
0.1	11542	9.9516e-03	-
0.05	44110	4.8172e-03	1.046
0.025	174982	2.3653e-03	1.026

having an exact solution with initial condition $f_0 = f(x, y, z, 0) = xyz$. Since we know the position of surface at any time step, we do not need to solve the equation about the level set function. The closest point of (x, y, z) is given by $\frac{(x-t,y,z)}{\sqrt{(x-t)^2+y^2+z^2}} + (t, 0, 0)$ at any time step in our process . We compute the solution up to final time $T = 1$ with time step-size $\Delta t = \Delta x = \Delta y = \Delta z$, and the numerical result at final time is shown in Table 8.

Example 9.

In this example, we are interested in the effect of surfactant concentration on the surface under the fixed velocity field but with different Peclet numbers. We set the initial condition $f_0(x, y, z) = 1$, which is constant function defined on the unit sphere, and the velocity field $\mathbf{u} = (\frac{y}{2}, 0, 0)$. Consider the problem

$$\begin{cases} f_t + \mathbf{u} \cdot \nabla_s f + (\nabla_s \cdot \mathbf{u}) f = \frac{1}{Pe_s} \triangle_s f & \text{on } \Sigma(t) \\ f(x, y, z, 0) = f_0(x, y, z) & \text{on } \Sigma(0) \\ \frac{d\mathbf{x}}{dt} = \mathbf{u} & \mathbf{x} \in \Sigma(t) \end{cases}$$

We use different values of Peclet number in our numerical test. The result of surfactant concentration on the surface at different times are shown in Figure 13,14 and 15, and Figure 16 shows the relative error for total mass of surfactant on the surface. We can observe that the flow will move surfactant to the tip points of the surface, and the surfactant concentration is decreasing since the surface area is increasing.

Figure 13: Peclet number = 2

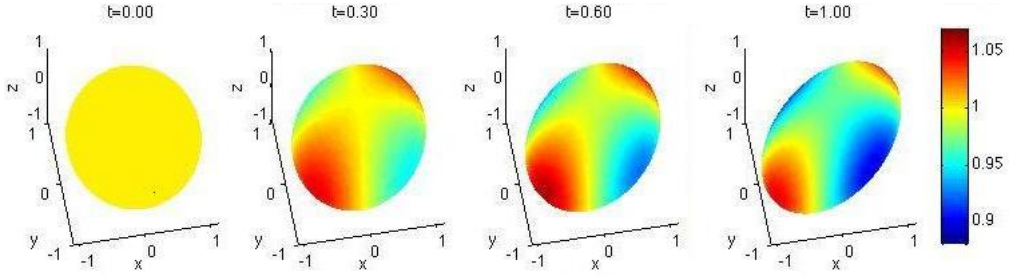


Figure 14: Peclet number = 1

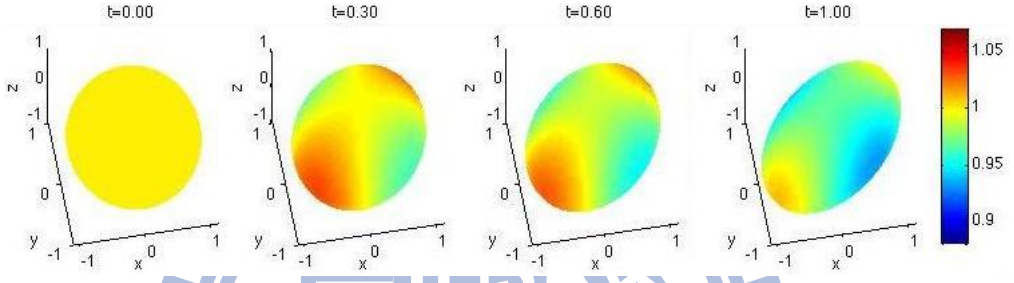


Figure 15: Peclet number = 0.5

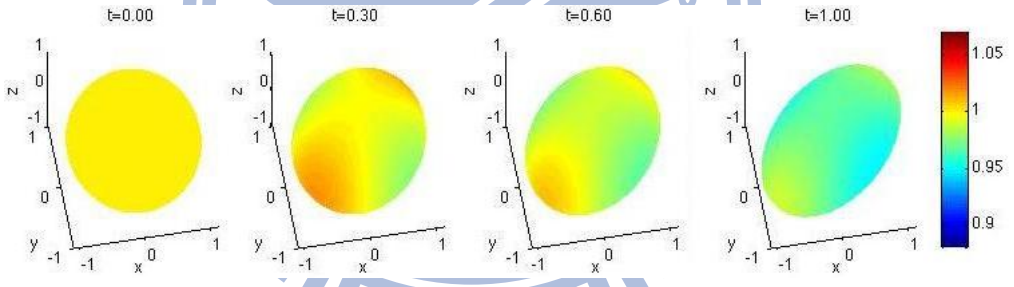
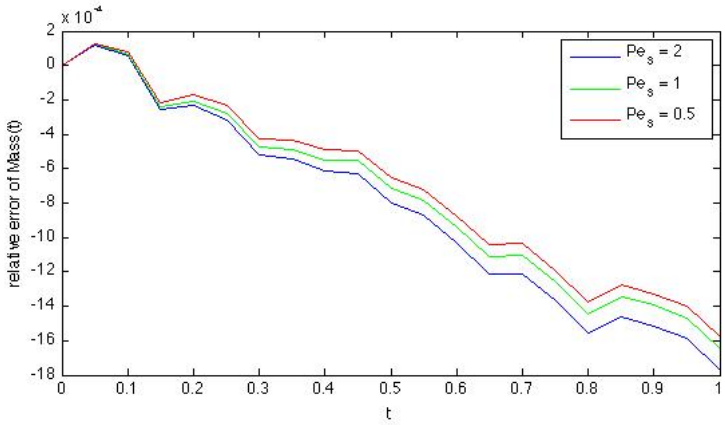


Figure 16: Relative error for total mass at each time, plotted in scale of 10^{-4}



7 Conclusion

In this paper, we apply an implicit closest point method to solve convection-diffusion equation on the moving surface, and use level set function to capture the moving surface. In closest point method, the way to find the closest point is important. Since we represent the surface by signed distance function, we can easily find the closest point in our numerical method. In the numerical test, our algorithm demonstrates good results for the rate of convergence and error of total mass.

Appendix

We prove the fundamental property 2. Assume that the velocity field $\mathbf{u}$ is tangent to the level-sets of the distance function, i.e., $\mathbf{u} \cdot \mathbf{n} = 0$. By the definition of surface divergence operator, we just need to show that $\mathbf{n}^t \cdot \nabla \mathbf{u} \cdot \mathbf{n} = 0$. Take gradient on both sides of $\mathbf{u} \cdot \mathbf{n} = 0$, we have

$$
\begin{aligned}
\mathbf{0} &= \nabla \mathbf{u} \cdot \mathbf{n} \\
&= \nabla (u_1 n_1 + u_2 n_2) \\
&= \begin{bmatrix} \frac{\partial}{\partial x}(u_1 n_1 + u_2 n_2) & \frac{\partial}{\partial y}(u_1 n_1 + u_2 n_2) \end{bmatrix} \\
&= \begin{bmatrix} \frac{\partial}{\partial x}(u_1) n_1 + \frac{\partial}{\partial x}(u_2) n_2 & \frac{\partial}{\partial y}(u_1) n_1 + \frac{\partial}{\partial y}(u_2) n_2 \end{bmatrix} \\
&+ \begin{bmatrix} u_1 \frac{\partial}{\partial x}(n_1) + u_2 \frac{\partial}{\partial x}(n_2) & u_1 \frac{\partial}{\partial y}(n_1) + u_2 \frac{\partial}{\partial y}(n_2) \end{bmatrix} \\
&= \begin{bmatrix} n_1 & n_2 \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x}(u_1) & \frac{\partial}{\partial y}(u_1) \\ \frac{\partial}{\partial x}(u_2) & \frac{\partial}{\partial y}(u_2) \end{bmatrix} + \begin{bmatrix} u_1 & u_2 \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x}(n_1) & \frac{\partial}{\partial y}(n_1) \\ \frac{\partial}{\partial x}(n_2) & \frac{\partial}{\partial y}(n_2) \end{bmatrix} \\
&= \mathbf{n}^t \nabla \mathbf{u} + \mathbf{u}^t \nabla \mathbf{n}
\end{aligned}
$$

Then

$$
\begin{aligned}
(\mathbf{n}^t \cdot \nabla \mathbf{u}) \cdot \mathbf{n} &= -(\mathbf{u}^t \cdot \nabla \mathbf{n}) \cdot \mathbf{n} \\
&= -\mathbf{u}^t \begin{bmatrix} \nabla n_1 \cdot \mathbf{n} \\ \nabla n_2 \cdot \mathbf{n} \end{bmatrix} = 0
\end{aligned}
$$

because n_1 and n_2 only vary along tangent direction. Then we have

$$
\begin{aligned}
\nabla_s u &= \nabla u - \mathbf{n}^t \cdot \nabla \mathbf{u} \cdot \mathbf{n} \\
&= \nabla u
\end{aligned}
$$

References

[1] Steven J. Ruuth, Barry Merriman, A simple embedding method for solving partial differential equations on surfaces, Journal of Computational Physics 227(3)(2008) p.1943-1961.

[2] Guang-Shan Jiang, Danping Peng, Weighted ENO schemes for Hamilton-Jacobi equations, SIAM Journal on Scientific Computing 21(6) p.2126-2143.

[3] Chohong Min, On reinitializing level set functions, Journal of Computational Physics 229(8)(2010) p.2764-2772.

[4] Colin B. Macdonald, Steven J. Ruuth, The implicit closest point mwthod for the numerical solution of partial differential equation on surfaces, SIAM Journal on Scientific Computing 31(6)(2009) p.4330-4350.

[5] G. Dziuk, C. M. Elliott, Finite elements on evolving surfaces, IMA Journal of Numerical Analysis 27(2)(2007) p.262-292.

[6] Stanley Osher, Ronald Fedkiw, Level set methods and dynamic implicit surfaces, Spring(2003).

[7] Jian-Jun Xu, Hong-Kai Zhao, An eulerain formulation for solving partail differenctial equation along a moving interface, Journal of Scientific Computing 19(1-3)(2003) p.573-594.

[8] H. Wong, D. Rumschitzki, and C. Maldarelli, On the surfactant mass balance at a deforming fluid interface, Physics of Fluid 8(11)(1996) p.3203-3204.

[9] H. A. Stone, A simple derivation of the time-dependent convective-diffusion equation for surfactant transport along a deforming interface, Physics of Fluids 2(1)(1990) p.111-112.

[10] Huaxiong Huang, Ming-Chih Lai, Hsiao-Chieh Tseng, A parametric derivation of the surfactant transport equation along a deforming fluid interface, Frontiers of Applied and Computational Mathematics, ed. by D. Blackmore et al.(2008) p.198-205.

[11] Chi-Wang. Shu, Total-variation-diminishing time discretization, SIAM Journal on Scientific and Statistical Computing 9(6)(1988) p.1073-1084.

[12] Ming-Chih Lai, Yu-Hau Tseng, Huaxiong Huang, An immersed boundary method for interfacial flows with insoluble surfactant, Journal of Computational Physics 227(2008) p.7279-7293.

[13] Ashley J. James, John Lowengrub, A surfactant-conserving volume-of-fluid method for interfacial flows with insoluble surfactant, Journal of Computational Physics 201(2004) p.685-722.

[14] Xiaofeng Yang, Ashley J. James, An arbitrary Lagrangian-Eulerian (ALE) method for interfacial flows with insoluble surfactants, Fluid Dynamics and Materials Processing 3(1)(2007) p65-96.