Maintaining the Point Correspondence in the Level Set Framework

J.-P. Pons [a], G. Hermosillo [b], R. Keriven [a] and O. Faugeras [c]

[a] *Odyssée Laboratory, ENPC, Marne-la-Vallée, France*

[b] *Siemens Medical Solutions, Malvern, USA*

[c] *Odyssée Laboratory, INRIA, Sophia-Antipolis, France*

Abstract

In this paper, we propose a completely Eulerian approach to maintain a point correspondence during a level set evolution. Our work is in the spirit of some recent methods (Adalsteinsson & Sethian, J. Comp. Phys. 2003; Xu & Zhao, J. Sci. Comp. 2003) for handling interfacial data on moving level set interfaces. Our approach maintains an explicit backward correspondence from the evolving interface to the initial one, by advecting the initial point coordinates with the same velocity as the level set function. It leads to a system of coupled Eulerian partial differential equations. We describe in detail a robust numerical implementation of our approach, in accordance with the narrow band methodology. We show in a variety of numerical experiments that it can handle both normal and tangential velocities, large deformations, shocks, rarefactions and topological changes. The possible applications of our approach include scientific visualization, computer graphics and image processing.

Key words: level set method, point correspondence, interfacial data, tangential velocity.

1 Introduction

The level set method, introduced by Osher and Sethian in [1] (a similar work in the area of fluid mechanics [2,3] has recently surfaced), is an established technique to represent moving interfaces in two or more dimensions. Basically, it consists in representing the interface as the zero level set of a higher-dimensional scalar function. The movement of the interface can be cast as an evolution of the embedding level set function. We refer the reader to [4,5] for all the details about the theory, the recent developments, the implementation and the applications of the level set method.

On the one hand, this approach has several advantages over an explicit Lagrangian representation of the interface as a parameterized curve or surface: no parameterization is needed, the Eulerian PDE formulation brings strong mathematical proofs as well as robust numerical schemes, topology changes are handled automatically, and intrinsic geometric properties such as normal or curvature can be computed easily from the level set function.

On the other hand, the higher dimensional embedding makes the level set method much more expensive computationally than traditional explicit representations. Much effort has been done to alleviate this drawback, leading to the narrow band methodology [6] and more recently to the PDE-based fast local level set method [7].

Another serious shortcoming of the level set method, which has been studied much more recently, is intimately related to the implicit point of view and to the absence of parameterization: in the traditional level set framework, it is not possible to handle some data associated with the moving interface, for short some *interfacial data*. This is not a surprise, since the level set function conveys a purely geometric description of the interface.

The earliest Eulerian approach related to the handling of some interfacial data in the level set framework appears in [8]. This method allows to track a region on a deforming level set interface. But it cannot handle some arbitrary interfacial data, so it has a rather limited range of applications. Also related is a method to evolve a curve in three dimensions with a level set approach [9]. More recently, some Eulerian methods have been proposed to evolve an interfacial material quantity in the level set framework [10,11]. The data are simultaneously advected, scaled by the local compression/expansion of the interface as a result of mass conservation, and diffused along the interface.

However, in some specific applications, it is also necessary to know how each point or each part of the interface moves, in other words to maintain an explicit point-wise correspondence during the evolution, a mapping between the initial interface and the final one, which all the above methods cannot achieve.

Some hybrid Lagrangian-Eulerian methods have been proposed to circumvent this limitation in some applications such as the unfolding of the surface of the cerebral cortex [12], the construction of transverse lines in grid generation [4], and image registration with level sets [13]. Basically, in these works, the level set equation is complemented with a set of Lagrangian ODEs which track the points during the evolution. This particle-based approach leads to well-known numerical difficulties. Most of the advantages of using an implicit representation are lost. For sake of numerical stability and topology independence, a completely Eulerian approach must be preferred.

In this paper, we propose an Eulerian method to maintain an explicit point

correspondence during a level set evolution. Our approach maintains a mapping from the evolving interface to the initial one, by advecting the initial point coordinates with the same velocity as the level set function.

The rest of this paper is organized as follows. In Section 2, we analyze the previous work on level sets with some interfacial data and we elaborate a novel method for handling an explicit point correspondence, based on a system of coupled Eulerian partial differential equations. Section 3 describes in detail a robust numerical implementation of our approach, in accordance with the narrow band methodology. In Section 4, we report on some numerical experiments that demonstrate the effectiveness of our method in a wide range of situations including normal and tangential velocity fields, large deformations, shocks, rarefactions and topological changes. Finally, Section 5 discusses some possible applications of our approach.

2 Methods

In the following, we note $\Gamma(t), t \in \mathbb{R}^+$ a moving closed and embedded hypersurface in $\mathbb{R}^n$. Γ is represented by a level set function $\phi : \mathbb{R}^n \times \mathbb{R}^+ \to \mathbb{R}$ such that:

$$\begin{cases} \phi(\mathbf{x}, t) < 0 \text{ if } \mathbf{x} \text{ is inside } \Gamma(t), \\ \phi(\mathbf{x}, t) = 0 \text{ if } \mathbf{x} \in \Gamma(t), \\ \phi(\mathbf{x}, t) > 0 \text{ if } \mathbf{x} \text{ is outside } \Gamma(t). \end{cases} \tag{1}$$

A deformation of Γ with a velocity field $\mathbf{v}$ has a direct counterpart in the level set representation:

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0. \tag{2}$$

It is well known that the geometry of the interface is only affected by the normal component of the velocity. The corresponding property in the level set formulation is that the tangential component of the velocity cancels in (2). That is the reason why the velocity is often taken normal to the interface: $\mathbf{v} = \beta \mathbf{N}$, where $\mathbf{N}$ denotes the outward normal. This can be done without any loss of generality when geometry only is of interest. Given that $\mathbf{N} = \nabla\phi / \|\nabla\phi\|$, we get the classical form of the level set evolution equation:

$$\frac{\partial \phi}{\partial t} + \beta \|\nabla \phi\| = 0 \, . \tag{3}$$

A tangential velocity does not affect the geometry, but it does affect the point correspondence and the data associated with the interface. Hence, in this paper, it would be erroneous to restrict to a normal velocity field.

2.1 Previous work on region tracking

In [8], the problem of tracking a region in Γ during the level set evolution is addressed. The boundary of the region of interest is represented as the intersection of Γ with the interior of an auxiliary hypersurface $\hat{\Gamma}$ defined as the zero level set of a function $\hat{\phi}$. The two functions ϕ and $\hat{\phi}$ are evolved according to the following system of coupled Eulerian PDEs:

$$\begin{cases} \dfrac{\partial \phi}{\partial t} + \beta \left\| \nabla \phi \right\| = 0 \ , \\ \dfrac{\partial \hat{\phi}}{\partial t} + \left(\beta \dfrac{\nabla \phi}{\left\| \nabla \phi \right\|} \cdot \dfrac{\nabla \hat{\phi}}{\left\| \nabla \hat{\phi} \right\|} \right) \left\| \nabla \hat{\phi} \right\| = 0 \ . \end{cases} \tag{4}$$

where β is again the magnitude of the outward normal speed of the interface. Rewriting (4) using a form similar to (2) allows an easier interpretation:

$$\begin{cases} \mathbf{v} = \beta \dfrac{\nabla \phi}{\left\| \nabla \phi \right\|} \ , & (5a) \\ \dfrac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0 \ , & (5b) \\ \dfrac{\partial \hat{\phi}}{\partial t} + \mathbf{v} \cdot \nabla \hat{\phi} = 0 \ . & (5c) \end{cases} \tag{5}$$

So (5b) and (5c) simply state that Γ and $\hat{\Gamma}$ move with the same velocity field $\mathbf{v}$. In [8], only a normal velocity is considered. But a tangential component is perfectly possible. Again, we emphasize that a tangential velocity does not affect the geometry of the interface, but it does affect the motion of $\hat{\Gamma}$ and hence of the region of interest.

This method is closely related to the work of [9] on the motion of a curve in three spatial dimensions with a level set approach. The curve is represented as the intersection of Γ and $\hat{\Gamma}$ and its motion can be expressed as the evolution of the two level set functions ϕ and $\hat{\phi}$ according to (5b) and (5c). But in contrast with region tracking and (5a), the velocity then depends on the geometric properties of the curve (e.g. tangent, normal, binormal, curvature, torsion), and not only on Γ.

2.2 Previous work on transport and diffusion of a material quantity

In [10,11], two similar methods are proposed to model the evolution of a material quantity f along a moving interface in the level set framework. Besides being passively advected, the interfacial data are also scaled due to the local compression/expansion of the interface, as a result of mass conservation, and diffused along the interface. This is achieved by the following system of coupled Eulerian PDEs:

$$\begin{cases} \dfrac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0 \ , \\ \dfrac{\partial f}{\partial t} + \mathrm{div}_\Gamma(f\mathbf{v}) = \Delta_\Gamma f \ , \end{cases} \tag{6}$$

where div_Γ is the intrinsic divergence operator and Δ_Γ is the intrinsic Laplacian operator on the interface, often called the Laplace-Beltrami operator.

This method requires to define the material quantity on the whole space. Just like the definition of the level set function off Γ is arbitrary, any extension f which agrees with the data on the interface can be considered. Such an embedding of the data was previously used to solve variational problems and PDEs on *fixed* implicit interfaces [14]. In some applications, the data may have a natural extension off the interface. In other applications, one may build an extension f by some numerical procedure, that we discuss in Section 3.

The mass conservation behavior, responsible for the divergence form in (6), and the diffusion behavior are both related to the physical interpretation of the data as concentrations. However, this interpretation is not relevant to all types of interface data. For example, mass conservation does not make sense in the case of an evolving textured surface, in computer graphics. In the next subsection, we study this slightly different case of passively advected interfacial data, by taking the region tracking method of Subsection 2.1 as a source of inspiration.

2.3 LSID: Level sets with some interfacial data

If we go back to Subsection 2.1 and take a closer look at (5), we notice that the zero level set of $\hat{\phi}$ does not play a particular role. All the level sets of $\hat{\phi}$ evolve according to $\mathbf{v}$. Actually, the evolution equation for $\hat{\phi}$ is nothing but a passive advection equation with an extrinsic velocity field.

Hence we can go beyond the interpretation of $\hat{\phi}$ as the level set function of a hypersurface and substitute to it a general scalar or vector-valued function f

coding for some interfacial data. This slight change immediately generalizes the tracking method of [8] to the evolution of any quantity passively advected with a moving level set interface. The corresponding PDEs are

$$\begin{cases} \dfrac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0 \ , & \text{(7a)} \\ \dfrac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla f = 0 \ . & \text{(7b)} \end{cases} \tag{7}$$

In the sequel, we will refer to this approach as *level sets with some interfacial data* (LSID). We can now regard (4) and (5) as a particular case of LSID with f restricted to a scalar function, the sign of which tags the region of interest. Interestingly, the above method can be used to upgrade a traditional level set evolution without any modification of the existing PDE.

At first sight, the level set function and the data play symmetric roles in (7) and the two sub-equations are decoupled. Actually, this is only the case when the velocity field is given *a priori*. In most problems, the velocity field depends on the geometric properties of Γ (e.g. normal and curvature) and hence on ϕ. In some problems, the velocity field may also depend on the values of the interfacial data.

2.4 LSPC: Level sets with a point correspondence

LSID is versatile but it suffers from several limitations. First, it does not provide an explicit point correspondence, which makes it inadequate for some applications. Second, it requires to solve a PDE for each scalar component of the data, which may be prohibitive in some applications.

In order to overcome these problems we propose to maintain, rather than the interfacial data themselves, some unambiguous coordinates of the points of the interface. This implies a choice of a coordinate system, for example a global parameterization of the interface. However, obtaining a global parameterization of a complex shape is difficult. Moreover, critical points and periodic conditions would be tricky to handle. Finally, reintroducing a parameterization is quite unnatural in the implicit framework.

A very convenient alternative is to use the initial Cartesian coordinates of the points of the interface in the embedding space $\mathbb{R}^n$. We regard them as vector-valued interfacial data that we evolve with LSID. This is equivalent to considering a function $\psi : \mathbb{R}^n \times \mathbb{R}^+ \mapsto \mathbb{R}^n$ such as

$$\psi(\mathbf{x}, 0) = \mathbf{x} \tag{8}$$

and

$$\begin{cases} \dfrac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0 \ , & \text{(9a)} \\ \dfrac{\partial \psi}{\partial t} + D\psi \, \mathbf{v} = 0 \ , & \text{(9b)} \end{cases} \tag{9}$$

where $D\psi$ stands for the Jacobian matrix of ψ. For each point $\mathbf{x}$ of the interface at time t, $\psi(\mathbf{x}, t)$ holds the position that this point was occupying at time $t = 0$. In other words, $\psi(., t)$ provides an explicit backward point correspondence from the current interface $\Gamma(t)$ to the initial one $\Gamma(0)$. Off the interface, the point correspondence is driven by the extension of the velocity. In most applications, the latter is arbitrary, so the values of ψ off $\Gamma(t)$ do not have a physical meaning.

Once the point correspondence is available, the evolution of any other passively advected interfacial data with (7b) can be bypassed. We build any such data by composition of the initial data f_0 with the correspondence function ψ. As a matter of fact, $f = f_0 \circ \psi$ formally satisfies (7b):

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla f = (\nabla f_0 \circ \psi) \cdot \left(\frac{\partial \psi}{\partial t} + D\psi \, \mathbf{v} \right) = 0 \ . \tag{10}$$

In the sequel, we will refer to this approach as *level sets with a point correspondence* (LSPC). Equation (9b) is the Eulerian counterpart of the Lagrangian ODE which gives the forward point correspondence. The latter can be represented by a function $\hat{\psi} : \mathbb{R}^n \times \mathbb{R}^+ \mapsto \mathbb{R}^n$ such as

$$\begin{cases} \hat{\psi}(\mathbf{x}, 0) = \mathbf{x} \ , \\ \dfrac{\partial \hat{\psi}}{\partial t} = \mathbf{v} \circ \hat{\psi} \ . \end{cases} \tag{11}$$

This Lagrangian approach is used in [4,12,13] to circumvent the loss of the point correspondence in the level set method. But the Eulerian PDE (9b) has two important advantages over the Lagrangian ODE (11). First, it is numerically more stable since the computations are performed on a fixed grid. More importantly, it performs automatic deleting of merging characteristics, whereas this task requires intricate delooping algorithms in the Lagrangian approach. Moreover, a forward correspondence may not exist if the evolution forms shocks; the interface may even collapse and merely disappear. In such cases, (11) is not relevant. That is the reason why LSPC only focuses on the backward point correspondence. In some applications however, the forward correspondence is needed and it is necessary to invert the ψ map at a post-processing stage.

If the velocity field is sufficiently smooth [15,16], (9b) generates a one-parameter family of diffeomorphisms. So does (11) and by the chain rule we get $\psi(.,t) \circ \hat{\psi}(.,t) = \hat{\psi}(.,t) \circ \psi(.,t) = \mathrm{Id},\ \forall t$. This is not true in general, as illustrated in some of the numerical experiments that we report in Section 4. Typically, ψ fails to be surjective and develops a discontinuity in the presence of a shock (see Experiments 2 and 4) and fails to be injective in the presence of a rarefaction (see Experiment 3).

3 Numerical Algorithms

In this section, we describe in detail a numerical implementation of LSID and LSPC.

3.1 Level set reinitialization and data extension

On the one hand, the definition of the level set function and of the data off the interface is arbitrary in the continuous formulation. But on the other hand, from a numerical point of view, flat and/or steep regions that develop in the level set function and in the data during the evolution can dramatically decrease the accuracy of the computed solution. This motivates the use of the signed distance function to the interface as the level set function.

An interesting approach to counteract the loss of resolution of the level set function is to use a particular extension of the velocity that maintains the signed distance property [17,18]. However, this approach does not have a counterpart for the data function. A more common approach is to occasionally apply a reinitialization procedure which restores the signed distance property. This can be done either by applying a Fast Marching technique [19] or by considering the steady state solution to the following PDE [20]:

$$\frac{\partial \phi}{\partial \tau} + sign(\phi_0)\,(\|\nabla \phi\| - 1) = 0\ . \tag{12}$$

Similarly, as discussed in [14,21], an extension of the data constant along the normal to the isolevels of ϕ, i.e. of the form

$$\nabla f \cdot \nabla \phi = 0 \tag{13}$$

is numerically advantageous. Such an extension can be obtained either in combination with a reinitialization by Fast Marching [17] or by running the

following PDE [22]:

$$\frac{\partial f}{\partial \tau} + sign(\phi)\,(\nabla f \cdot \nabla \phi) = 0\ . \tag{14}$$

Note that in the case of PDEs on *fixed* implicit interfaces, pioneered by [14], an alternative was recently proposed in [21], in order to naturally retain the desirable property (13) along time, then removing the need for a recurrent extension of the data. This method is potentially interesting, but the case of moving interfaces is not addressed yet.

Independently of numerical accuracy, a reinitialization procedure and an extension procedure are required when using a localized version of the level set method, such as the narrow band methodology [6] or the PDE-based fast local level set method [7]. Indeed, when the narrow band is rebuilt, proper values must be assigned to the level set function and to the data at the new active grid points.

In our implementation, we perform both the reinitialization and the extension with a single pass of Fast Marching, as proposed in [17]. In this work, the authors build an extension of the velocity and a temporary signed distance function as a byproduct, at each iteration. In our case, we run the Fast Marching procedure only occasionally: when the narrow band needs rebuilding, and otherwise periodically to keep the interface and the data well-resolved.

3.2 Keeping the point correspondence onto the initial interface

This subsection specifically applies to LSPC. In this case, the interfacial data are the coordinates of the points of the initial interface. Moreover, due to the extension procedure described in Subsection 3.1, the values of ψ in the whole domain belong to $\Gamma(0)$.

We show that if the initial correspondences belong to the zero level set of ϕ_0, this remains true throughout the evolution. We note $\nu = \phi_0 \circ \psi$. We obtain from (9b)

$$\frac{\partial \nu}{\partial t} + \mathbf{v} \cdot \nabla \nu = 0\ . \tag{15}$$

So if $\nu(.,0) \equiv 0$, we have $\nu(.,t) \equiv 0$, $\forall t$. However, a numerical scheme for (9b) will in general move ψ outside of $\Gamma(0)$. We propose two adaptations to counteract this numerical drift. We take our inspiration in the work of [23] on solving PDEs mapping into an implicitly-defined target manifold.

The first adaptation consists in projecting the variation of ψ on the tangent plane of the initial interface. Without loss of generality, we can assume that ϕ_0 is a signed distance function. In this case, the projection operator can be written in a very simple form. Thus, (9b) is replaced with

$$\frac{\partial \psi}{\partial t} + \Big[\mathrm{I} - (\nabla \phi_0 \circ \psi) \, (\nabla \phi_0 \circ \psi)^T \Big] \, D\psi \, \mathbf{v} = 0 \; . \tag{16}$$

Eq. (16) is mathematically equivalent to (9b), but it turns out to be numerically advantageous. In particular, the preservation of ν is enforced more directly than in (15) since we now have

$$\frac{\partial \nu}{\partial t} = (\nabla \phi_0 \circ \psi) \cdot \frac{\partial \psi}{\partial t} = 0 \; . \tag{17}$$

The second adaptation consists in projecting the correspondences on the initial interface, in other words replacing each values of ψ by the closest point on Γ_0.

$$\psi \; \leftarrow \; P_{\Gamma_0} \circ \psi \; . \tag{18}$$

Generally, a closest point algorithm must be used to compute the projection operator P_{Γ_0}. However, if ϕ_0 is a signed distance function, a simple expression can be used outside of the skeleton of Γ_0:

$$P_{\Gamma_0}(\mathbf{x}) = \mathbf{x} - \phi_0(\mathbf{x}) \, \nabla \phi_0(\mathbf{x}) \; . \tag{19}$$

In our implementation of LSPC, we use these two adaptations in combination. We apply the modified PDE (16) at each iteration, and we reproject the values of the correspondence function with (19) before starting each reinitialization/extension procedure.

3.3 Finite-difference schemes

On the one hand, the proper numerical scheme for the level set evolution equation (7a) and (9a) depends on the properties of the velocity field. We do not discuss this here. All the schemes needed in the numerical experiments of Section 4 can be found in [1,4]. On the other hand, an advection scheme can be adopted for (7b) and (9b) as soon as the velocity field does not depend either on the interfacial data or on the correspondence, in other words as soon as $\mathbf{v}$ is extrinsic to f and ψ.

Moreover, to estimate one-sided space derivatives we adopt the third-order weighted essentially non-oscillatory (WENO) scheme derived in [24,25]. Fi-

nally, with respect to time differencing, a simple Euler scheme has proven sufficient in our experiments. Of course, we could resort to higher-order schemes, such as the total-variation diminishing Runge-Kutta scheme of third order devised in [26].

3.4 Overview of the algorithm

For sake of clarity, we give an overview of our implementation of LSPC in Algorithm 1 below.

Algorithm 1 Algorithm for level sets with a point correspondence

```
for all iterations do
  Compute the velocity field.
  Evolve the level set function by (9a).
  Evolve the correspondence function by (16).
  if the narrow band need rebuilding, otherwise periodically then
    Reproject the correspondence function by (19).
    Run the reinitialization/extension procedure.
  end if
end for
```

4 Experimental Results

In this section, we report on some numerical experiments that demonstrate the robustness and accuracy of LSID and LSPC in a wide range of situations, including normal and tangential velocity fields, large deformations, shocks, rarefactions and topological changes. Whenever the exact solution for the position of the interface, for the data and for the correspondence are known, we measure and we discuss the numerical error.

4.1 Definition of the error measures

The computation of the numerical error is not straightforward. In particular, a direct comparison of the computed data and correspondence with the exact solution is not possible since the computed interface and the exact interface differ. In all our experiments, the error is measured as follows.

To get the error on the position of the interface, we extract the zero isocontour of the computed level set function using the marching cubes algorithm [27].

At the resulting vertices, we compute the distance to the exact interface. Then we accumulate the point-wise errors along the contour to get the mean error (L_1 norm), the mean square error (L_2 norm), and the maximum error (L_∞ norm).

To get the error of LSID on the interfacial data, we sample the exact interface and we interpolate the computed data at these points using cubic spline interpolation. Due to the extension procedure (Subsection 3.1), this is equivalent to evaluating the computed data at the closest points on the computed interface. Then we substract the exact data and we compute the L_1, L_2 and L_∞ norms. We proceed similarly to get the error of LSPC on the correspondence.

4.2 2D test cases

Throughout this subsection, the spatial domain is the unit square $[0,1] \times [0,1]$ and we use a time step of $\Delta t = \Delta x/10$ and a $6\Delta x$-thick narrow band. The reinitialization/extension procedure is started when the distance of the interface to the borders of the narrow band gets below $3\Delta x$. In all five experiments, the initial data are a function of the central angle θ: $f_0(\theta) = \sin(3\theta)$.

Figure 1 demonstrates LSID and LSPC in several 2D test cases. The curves displayed in this figure are obtained from the computed level set function using a marching squares algorithm [27]. At the resulting vertices, we sample the computed data and correspondence function using cubic spline interpolation.

Each row shows a different experiment. In each experiment, column A shows the initial interface colored with the initial data. Column B shows the interface at $t = 0.2$ colored with the data computed with LSID. The last two columns show the results of LSPC at the same time instant. In column C, the correspondence function is plotted at one out of ten vertices. The initial and the current interfaces are plotted too, with a dotted line and a dashed line respectively. Column D shows the interface colored with the transformed data $f_0 \circ \psi$.

The errors for the different experiments at $t = 0.2$ for different grid sizes (50^2, 100^2, 200^2) are given in Tables 1-5. In the next paragraphs, we describe in detail the purpose, the setting and the results of each experiment.

Experiment 1: a rotating and shrinking circle. The initial interface is a circle with radius 0.4 centered at $(0.5, 0.5)$ and the velocity field is a combination of a unit inward normal speed and of an extrinsic rigid rotation of angular velocity 5. This example demonstrates that both normal and tangential velocities can be handled.

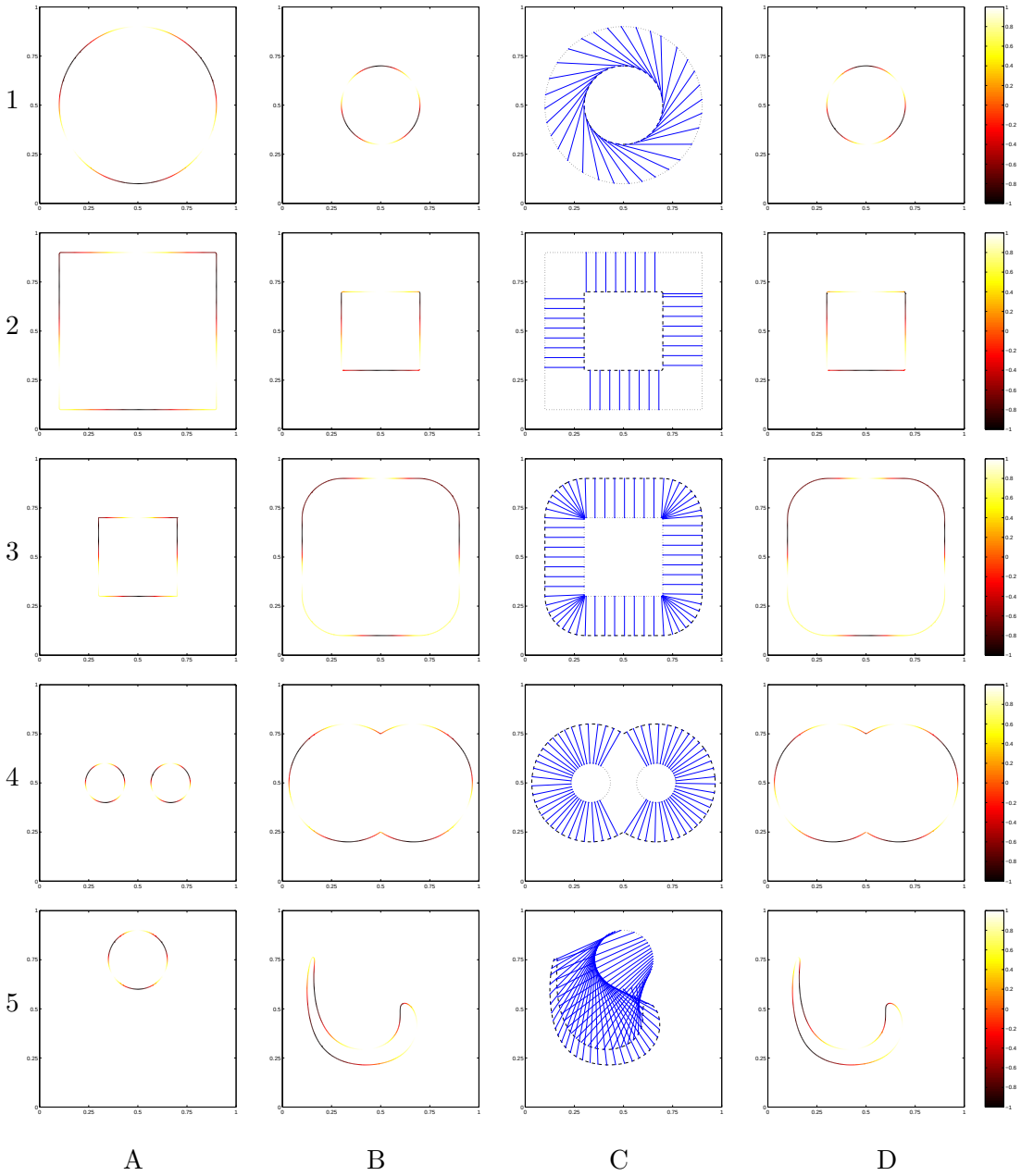


Fig. 1. Results of LSID (column B) and LSPC (columns C,D) in several 2D test cases.

Experiment 2: a shrinking square. The initial interface is a square with side length 0.4 centered at $(0.5, 0.5)$ and the velocity field is a unit inward normal speed. The evolution forms shocks at the angles of the square. Discontinuities develop in the data and in the correspondence. Figure 2,left provides a detailed view of Figure 1,2C with a denser representation for the correspondence. It shows that the expected discontinuities are fairly well recovered by LSPC.

Experiment 3: an expanding square. The initial interface is a square with

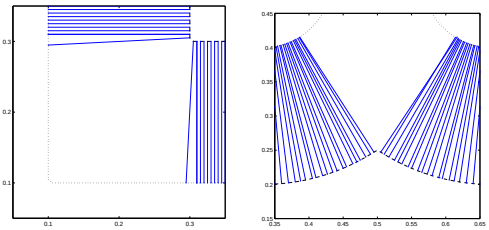


Fig. 2. Some detailed views of Figure 1-C2 and Figure 1,4C.

side length 0.2 centered at $(0.5, 0.5)$ and the velocity field is a unit outward normal speed. The evolution forms rarefactions cones at the angles of the square. The many-to-one correspondence is successfully recovered by LSPC.

Experiment 4: the merging of two expanding circles. The initial interface is composed of two circles with radius 0.1 centered at $(1/3, 0.5)$ and $(2/3, 0.5)$ and the velocity field is a unit outward normal speed. The two circles merge at $t = 2/15$ and two discontinuities develop in the data and in the correspondence. With no surprise, the topological change is handled automatically by the level set method. The discontinuities in the correspondence are also recovered by LSPC, as illustrated in Figure 2,right.

Experiment 5: a circle in a vortex velocity field. The initial interface is a circle with radius 0.15 centered at $(0.5, 0.75)$ and the evolution is driven by a non-constant vorticity velocity field that varies sinusoidally in time. It is defined by

$$\mathbf{v}(x, y) = 4 \cos \frac{\pi t}{T} \begin{pmatrix} \sin^2(\pi x) \sin(2\pi y) \\ -\sin(2\pi x) \sin^2(\pi y) \end{pmatrix}. \tag{20}$$

This velocity field at $t = 0$ is shown in Figure 3. This experiment is numer-

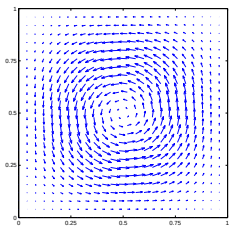


Fig. 3. Velocity field used in Experiment 5.

ically more challenging than the previous ones because the flow considerably stretches the interface. The exact solution is not available at all time, so at first sight we cannot measure the error. But the velocity reverses at time $T/2$, so the initial interface and the initial data should be recovered at time T. This provides a convenient way to evaluate the accuracy of our results. We take $T = 0.4$. The errors at $t = 0.4$ are given in Table 5.

4.3 3D experiments

We now demonstrate our method in three dimensions. In this subsection, the spatial domain is the unit cube $[0,1] \times [0,1] \times [0,1]$ and we use a time step of $\Delta t = \Delta x/5$. In order to test the robustness to large deformations, we consider the following incompressible velocity field proposed by LeVeque [28]:

$$\mathbf{v}(x,y,z) = \cos\frac{\pi t}{T} \begin{pmatrix} 2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z) \\ -\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z) \\ -\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z) \end{pmatrix}. \tag{21}$$

This flow is a surimposition of a deformation in the xy plane with a deformation in the xz plane. As in Experiment 5, the initial interface and the initial data should be recovered at time T. In the next paragraphs, we apply this velocity field to two different interfaces.

Experiment 6: a deforming plane. The initial interface is the plane $x = 0.5$ and the initial data are given by

$$f_0(x,y,z) = \sin(10\pi y)\,\sin(10\pi z)\,. \tag{22}$$

In this experiment, we take $T = 0.8$. Figure 4 shows the computed interface, the data and the correspondence at t=0, 0.2 and 0.4. Rather than plotting the point correspondence which is difficult to visualize in three dimensions, we color the interface with a checkerboard texture obtained by composition with ψ. The errors at $t = 0.8$ for different grid sizes (50^3, 100^3 and 200^3) are given in Table 6.

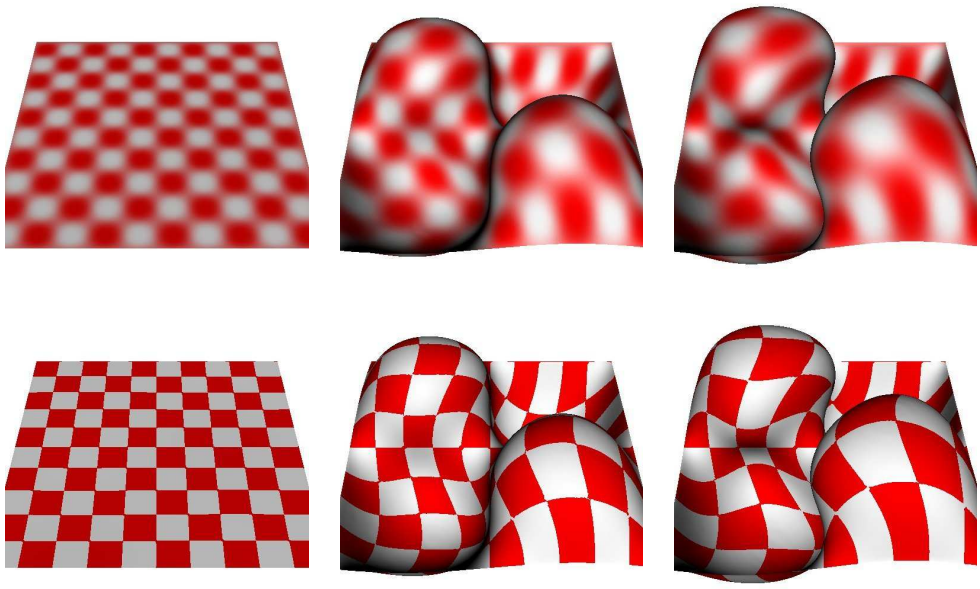
Experiment 7: a deforming sphere. The initial interface is a sphere with radius 0.15 centered at $(0.35, 0.35, 0.35)$ and the initial data are given by

$$f_0(x,y,z) = (x-0.35)\,(y-0.35)\,(z-0.35)\,/\,0.15^3\,. \tag{23}$$

In this experiment, we take $T = 1.6$. Figure 5 shows the results at t=0, 0.4 and 0.8. The errors at $t = 1.6$ are given in Table 7.

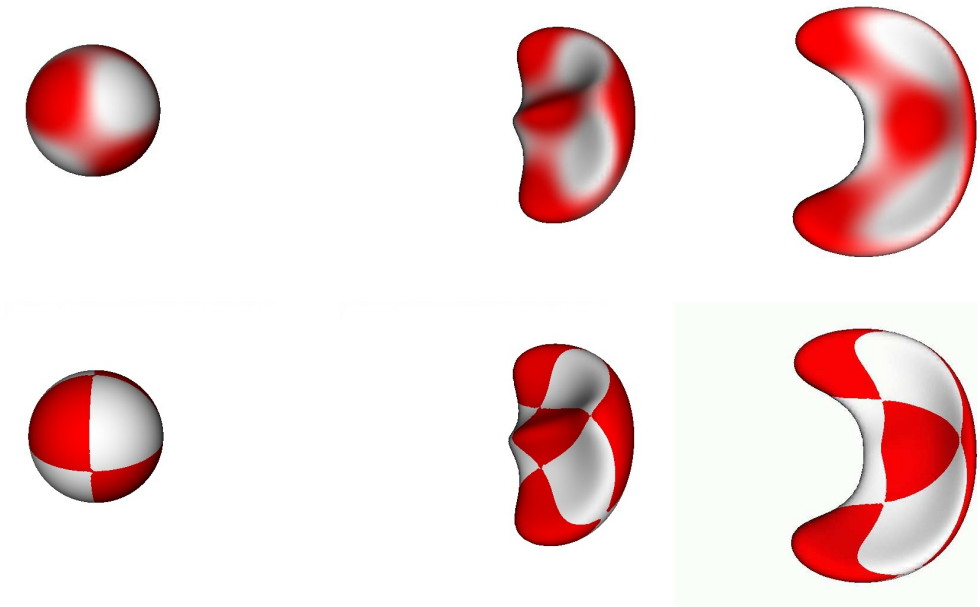
4.4 Comments on the errors

In Tables 1-7, we observe a regular decrease of the L_1, L_2 and L_∞ norms of the errors of LSID and LSPC when the dimension of the computational



$t = 0$ $t = 0.2$ $t = 0.4$

Fig. 4. Results of LSID (top) and LSPC (bottom) in Experiment 6 at different times.



$t = 0$ $t = 0.4$ $t = 0.8$

Fig. 5. Results of LSID (top) and LSPC (bottom) in Experiment 7 at different times.

grid increases. There are a few exceptions to this behavior, shown in bold fonts. They can be explained easily: in Experiment 2 and Experiment 4, the discontinuities that develop in the data and in the correspondence cause a stagnation of the L_∞ norm of the error.

5 Applications

5.1 Coupling of thin shells with fluids

The modeling of the deformation of thin shells [29] is traditionally done in a Lagrangian formulation, because it requires to track the material properties (e.g. the flexural rigidity) associated to the material points. Thus, the definition of elastic forces uses a mapping between the deformed and the undeformed configurations.

Our approach allows an Eulerian modeling of thin shells, which can be numerically advantageous in the case of the two-way coupling with an Eulerian fluid. It would remove the need for special techniques to reconcile the Eulerian and the Lagrangian viewpoints, like the immersed interface method [30], the ghost-fluid method [31] and its many variants (*cf* for example [32]).

Nevertheless, at the time of writing of this paper, we have not further investigated these interesting possibilities.

5.2 Scientific visualization

Our method can be used for visualizing some material properties of an interface with a very complex geometry, on a simpler avatar surface. This technique may be needed in many areas of science and engineering.

For instance, in medical imaging, building unfolded representations of the cerebral cortex, the thin layer of gray matter at the surface of the brain, has become an important area of research (see [33,34] and references therein). Indeed, on a simplified geometry, it becomes easier for physicians to visualize and analyze functional or structural properties of the cortex.

A straightforward level set implementation of cortex unfolding is not feasible, due to the loss of the point correspondence. The latter is essential to connect the observations made on the avatar to the actual surface. By using our approach, in [34], we have been able to compute simplified representations of the cortex from real brain data, while enjoying the numerical stability of the Eulerian formulation. In Figure 6, we show some of our results. The correspondence is displayed indirectly with a texture representing the mean curvature of the initial surface, which provides good landmarks for visualization. By applying to the cortical surface a mean curvature motion complemented with an area-preserving tangential motion, we have built a multi-scale shape description, while maintaining an area-preserving mapping to the original surface.

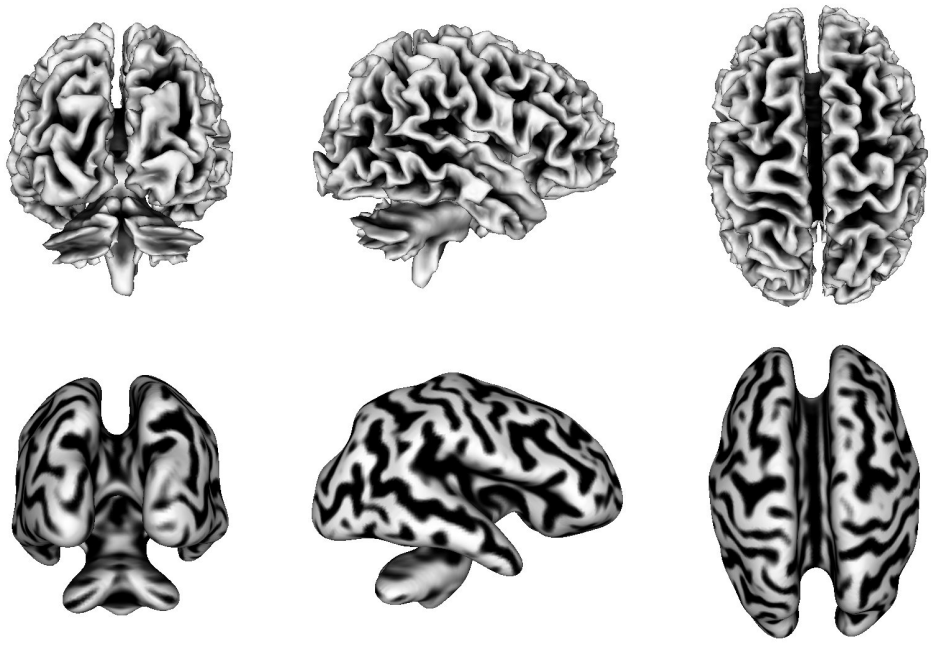


Fig. 6. Several views of the initial surface (top) and of the inflated representation (bottom). The correspondence is displayed indirectly with a texture representing the mean curvature of the initial surface.

6 Conclusion

In this paper, we have extended the applicability of the level set method to a specific class of problems requiring an explicit point correspondence during the evolution. We have proposed a system of coupled Eulerian PDEs to maintain a backward mapping from the current interface to the initial one. We have described in detail a robust numerical implementation of our approach: a procedure for reinitializing the level set function and for extending the interfacial data and the point correspondence, two adaptations to keep the point correspondence onto the initial interface, and the adequate numerical finite-difference schemes for the different PDEs. We have successfully tested our approach, in 2D and in 3D, in a wide range of situations, including normal and tangential velocities, large deformations, shocks, rarefactions and topology changes. In all our experiments, we have computed the numerical error and we have commented its variation with respect to the grid size.

References

[1] S. Osher, J. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations, Journal of Computational Physics 79 (1) (1988) 12–49.

[2] A. Dervieux, F. Thomasset, A finite element method for the simulation of Rayleigh-Taylor instability, Lecture Notes in Mathematics 771 (1979) 145–159.

[3] A. Dervieux, F. Thomasset, Multifluid incompressible flows by a finite element method, Lecture Notes in Physics 11 (1981) 158–163.

[4] J. Sethian, Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Sciences, Cambridge Monograph on Applied and Computational Mathematics, Cambridge University Press, 1999.

[5] S. Osher, R. Fedkiw, The Level Set Method and Dynamic Implicit Surfaces, Springer-Verlag, 2002.

[6] D. Adalsteinsson, J. Sethian, A fast level set method for propagating interfaces, Journal of Computational Physics 118 (2) (1995) 269–277.

[7] D. Peng, B. Merriman, S. Osher, H.-K. Zhao, M. Kang, A PDE-based fast local level set method, Journal of Computational Physics 155 (2) (1999) 410–438.

[8] M. Bertalmío, G. Sapiro, G. Randall, Region tracking on level-sets methods, IEEE Transactions on Medical Imaging 18 (5) (1999) 448–451.

[9] P. Burchard, L.-T. Cheng, B. Merriman, S. Osher, Motion of curves in three spatial dimensions using a level set approach, Journal of Computational Physics 170 (2) (2001) 720–741.

[10] J.-J. Xu, H.-K. Zhao, An Eulerian formulation for solving partial differential equations along a moving interface, Journal of Scientific Computing 19 (2003) 573–594.

[11] D. Adalsteinsson, J. Sethian, Transport and diffusion of material quantities on propagating interfaces via level set methods, Journal of Computational Physics 185 (1) (2003) 271–288.

[12] G. Hermosillo, O. Faugeras, J. Gomes, Unfolding the cerebral cortex using level set methods, in: Scale-Space Theories in Computer Vision, 1999, pp. 58–69.

[13] B. Vemuri, J. Ye, Y. Chen, L. C.M., A level-set based approach to image registration, in: IEEE Workshop on Mathematical Methods in Biomedical Image Analysis, 2000, pp. 86–93.

[14] M. Bertalmío, L. Cheng, S. Osher, G. Sapiro, Variational problems and partial differential equations on implicit surfaces, Journal of Computational Physics 174 (2) (2001) 759–780.

[15] P. Dupuis, U. Grenander, M. Miller, Variational problems on flows of diffeomorphisms for image matching, Quarterly of Applied Math. 56 (1998) 587–600.

[16] A. Trouvé, Diffeomorphisms groups and pattern matching in image analysis, The International Journal of Computer Vision 28 (3) (1998) 213–21.

[17] D. Adalsteinsson, J. Sethian, The fast construction of extension velocities in level set methods, Journal of Computational Physics 148 (1) (1999) 2–22.

[18] J. Gomes, O. Faugeras, Reconciling distance functions and level sets, Journal of Visual Communication and Image Representation 11 (2) (2000) 209–223.

[19] J. Sethian, A fast marching sevel set method for monotonically advancing fronts, in: Proceedings of the National Academy of Sciences, Vol. 93, 1996, pp. 1591–1694.

[20] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, Journal of Computational Physics 114 (1) (1994) 146–159.

[21] J. Greer, An improvement of a recent Eulerian method for solving PDEs on general geometries, Journal of Scientific ComputingTo appear.

[22] H.-K. Zhao, T. Chan, B. Merriman, S. Osher, A variational level set approach to multiphase motion, Journal of Computational Physics 127 (1) (1996) 179–195.

[23] F. Mémoli, G. Sapiro, S. Osher, Solving variational problems and partial differential equations mapping into general target manifolds, Journal of Computational Physics 195 (1) (2004) 263–292.

[24] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, Journal of Computational Physics 126 (1) (1996) 202–228.

[25] G.-S. Jiang, D. Peng, Weighted ENO schemes for Hamilton-Jacobi equations, SIAM Journal of Scientific Computing 21 (6) (2000) 2126–2143.

[26] C.-W. Shu, Total-variation-diminishing time discretizations, SIAM Journal of Scientific and Statistical Computing 9 (6) (1988) 1073–1084.

[27] W. Lorensen, H. Cline, Marching cubes: A high-resolution 3D surface reconstruction algorithm, ACM Computer Graphics 21 (4) (1987) 163–170.

[28] R. LeVeque, High-resolution conservative algorithms for advection in incompressible flow, SIAM Journal of Numerical Analysis 33 (2) (1996) 627–665.

[29] P. Ciarlet, Mathematical Elasticity, Vol. 29 of Studies in Mathematics and its Applications, Amsterdam, 2000, Theory of shells.

[30] C. Peskin, The immersed boundary method, Acta Numerica 11 (2002) 479–517.

[31] R. Fedkiw, Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with the ghost fluid method, Journal of Computational Physics 175 (1) (2002) 200–224.

[32] F. Cirak, R. Radovitzky, A Lagrangian-Eulerian shell-fluid coupling algorithm based on level sets, Computers and Structures 83 (2005) 491–498.

[33] B. Fischl, M. Sereno, A. Dale, Cortical surface-based analysis II : Inflation, flattening, and a surface-based coordinate system, Neuroimage 9 (2) (1999) 195–207.

[34] J.-P. Pons, R. Keriven, O. Faugeras, Area preserving cortex unfolding, in: International Conference on Medical Image Computing and Computer Assisted Intervention, Vol. 1, 2004, pp. 376–383.

Grid size		50×50	100×100	200×200
Surface error	L_1	2.6e-4	1.6e-4	1.1e-4
	L_2	2.7e-4	1.7e-4	1.2e-4
	L_∞	3.9e-4	3.0e-4	1.9e-4
LSID Data error	L_1	1.8e-2	1.2e-2	6.3e-3
	L_2	2.1e-2	1.4e-2	7.3e-3
	L_∞	3.9e-2	2.1e-2	1.2e-2
LSPC Correspondence error	L_1	7.3e-4	4.7e-4	2.7e-4
	L_2	8.5e-4	5.6e-4	3.3e-4
	L_∞	1.4e-3	1.3e-3	6.4e-4

Table 1
Errors for Experiment 1 at $t = 0.2$.

Grid size		50×50	100×100	200×200
Surface error	L_1	2.0e-5	6.3e-6	3.6e-6
	L_2	3.4e-5	1.7e-5	7.4e-6
	L_∞	1.0e-4	6.7e-5	4.0e-5
LSID Data error	L_1	2.4e-2	1.3e-2	6.5e-3
	L_2	1.1e-1	7.7e-2	5.5e-2
	L_∞	**7.4e-1**	**6.7e-1**	**6.5e-1**
LSPC Correspondence error	L_1	7.1e-3	3.2e-3	1.6e-3
	L_2	3.1e-2	2.0e-2	1.4e-2
	L_∞	**1.6e-1**	**1.4e-1**	**1.4e-1**

Table 2
Errors for Experiment 2 at $t = 0.2$.

Grid size		50×50	100×100	200×200
Surface error	L_1	3.1e-3	1.5e-3	7.2e-4
	L_2	4.7e-3	2.3e-3	1.1e-3
	L_∞	9.1e-3	4.4e-3	2.1e-3
LSID Data error	L_1	1.9e-2	1.0e-2	5.4e-3
	L_2	3.1e-2	1.8e-2	9.6e-3
	L_∞	1.0e-1	6.6e-2	4.3e-2
LSPC Correspondence error	L_1	5.8e-3	2.8e-3	1.4e-3
	L_2	8.2e-3	4.3e-3	2.3e-3
	L_∞	1.9e-2	1.3e-2	8.3e-3

Table 3
Errors for Experiment 3 at $t = 0.2$.

Grid size		50×50	100×100	200×200
Surface error	L_1	2.6e-4	7.7e-5	3.8e-5
	L_2	3.8e-4	1.4e-4	6.5e-5
	L_∞	1.3e-3	1.3e-3	6.8e-4
LSID Data error	L_1	6.7e-2	1.8e-2	5.6e-3
	L_2	8.3e-2	5.4e-2	2.7e-2
	L_∞	**3.7e-1**	**5.2e-1**	**3.9e-1**
LSPC Correspondence error	L_1	4.3e-3	1.6e-3	8.1e-4
	L_2	2.6e-2	1.4e-2	9.8e-3
	L_∞	**2.1e-1**	**1.9e-1**	**1.9e-1**

Table 4
Errors for Experiment 4 at $t = 0.2$.

Grid size		50×50	100×100	200×200
Surface error	L_1	6.3e-3	2.2e-3	1.1e-3
	L_2	7.0e-3	2.6e-3	1.3e-3
	L_∞	1.1e-2	5.3e-3	3.1e-3
LSID Data error	L_1	2.7e-1	8.5e-2	3.1e-2
	L_2	3.4e-1	1.3e-1	5.3e-2
	L_∞	7.2e-1	3.7e-1	1.9e-1
LSPC Correspondence error	L_1	1.4e-2	4.2e-3	2.2e-3
	L_2	2.2e-2	6.7e-3	3.3e-3
	L_∞	5.7e-2	2.0e-2	9.8e-3

Table 5
Errors for Experiment 5 at $t = 0.4$.

Grid size		50^3	100^3	200^3
Surface error	L_1	5.5e-3	2.9e-3	1.5e-3
	L_2	7.4e-3	3.9e-3	2.1e-3
	L_∞	1.9e-2	9.5e-3	5.5e-3
LSID Data error	L_1	1.3e-1	5.6e-2	2.4e-2
	L_2	1.9e-1	7.3e-2	3.1e-2
	L_∞	5.6e-1	1.9e-1	1.1e-1
LSPC Correspondence error	L_1	2.1e-3	1.6e-3	9.7e-4
	L_2	2.5e-3	1.9e-3	1.1e-3
	L_∞	6.3e-3	6.2e-3	3.9e-3

Table 6
Errors for Experiment 6 at $t = 0.8$.

Grid size		50^3	100^3	200^3
Surface error	L_1	7.2e-3	3.4e-3	1.9e-3
	L_2	1.1e-2	5.2e-3	2.8e-3
	L_∞	4.4e-2	2.4e-2	1.2e-2
LSID Data error	L_1	4.5e-2	1.7e-2	7.1e-3
	L_2	5.6e-2	2.1e-2	9.1e-3
	L_∞	1.5e-1	8.3e-2	3.6e-2
LSPC Correspondence error	L_1	1.5e-2	6.3e-3	3.4e-3
	L_2	1.8e-2	7.2e-3	4.0e-3
	L_∞	4.4e-2	2.0e-2	1.2e-2

Table 7
Errors for Experiment 7 at $t = 1.6$.