SIAM J. SCI. COMPUT. © 2006 Society for Industrial and Applied Mathematics
Vol. 28, No. 4, pp. 1490–1508

# MOVING MESH METHODS FOR SINGULAR PROBLEMS ON A SPHERE USING PERTURBED HARMONIC MAPPINGS*

YANA DI†, RUO LI†, TAO TANG‡, AND PINGWEN ZHANG†

**Abstract.** This work is concerned with developing moving mesh strategies for solving problems defined on a sphere. To construct mappings between the physical domain and the logical domain, it has been demonstrated that harmonic mapping approaches are useful for a general class of solution domains. However, it is known that the curvature of the sphere is positive, which makes the harmonic mapping on a sphere not unique. To fix the uniqueness issue, we follow Sacks and Uhlenbeck [*Ann. of Math.* (2), 113 (1981), pp. 1–24] to use a perturbed harmonic mapping in mesh generation. A detailed moving mesh strategy including mesh redistribution and solution updating on a sphere will be presented. The moving mesh scheme based on the perturbed harmonic mapping is then applied to the moving steep front problem and the Fokker–Planck equations with high potential intensities on a sphere. The numerical experiments show that with a moderate number of grid points our proposed moving mesh algorithm can accurately resolve detailed features of singular problems on a sphere.

**Key words.** moving mesh methods, singularity, harmonic mapping, perturbed harmonic mapping, spherical domain

**AMS subject classifications.** 65M50, 65N22, 58E20

**DOI.** 10.1137/050642514

## 1. Introduction.

In this work, we will consider efficient moving mesh methods for solving singular problems on a sphere. Partial differential equations (PDEs) defined on a sphere are seen in many practical cases such as polymer rheological analysis, ocean modeling, and global climate simulation. The moving mesh technique is one of the natural choices for solving problems with localized moving singularities. Examples include moving internal boundaries or interfaces separating different regions which are present in many problems in nature, such as interfaces separating two different fluids in a multiphase flow, and between solid and liquid regions in a solidification process. Such interfaces are in general dynamic, i.e., they evolve in time. Moving mesh methods have been proved powerful for solving time-dependent PDEs in two-dimensional domains and simple three-dimensional domains [2, 3, 4, 5, 24, 30, 31, 34].

In general, a moving mesh method always involves a logical domain and a physical domain, together with a transformation between them. In [12], Dvinsky suggests that harmonic function theory may provide a general framework for developing useful mesh generators. His method can be viewed as a generalization and extension of Winslow's method [33]. However, unlike most other generalizations which add terms

*Received by the editors October 12, 2005; accepted for publication (in revised form) March 21, 2006; published electronically September 15, 2006.
http://www.siam.org/journals/sisc/28-4/64251.html

†LMAM & School of Mathematical Sciences, Peking University, Beijing 100871, People's Republic of China (dyn@math.pku.edu.cn, rli@math.pku.edu.cn, pzhang@math.pku.edu.cn). The research of the first two authors was supported in part by the Joint Applied Mathematics Research Institute between Peking University and Hong Kong Baptist University. The research of the fourth author was partially supported by the special funds for Major State Research Projects (2005CB1704) and the National Science Foundation of China for Distinguished Young Scholars (10225103).

‡Mathematics Department, Hong Kong Baptist University, Kowloon Tong, Hong Kong (ttang@math.hkbu.edu.hk). The research of this author was supported by CERG grants of Hong Kong Research Grant Council, FRG grants of Hong Kong Baptist University, NSAF grant 10476032 of the National Science Foundation of China, and by the International Research Team on Complex System of Chinese Academy of Sciences.

or functionals to the basic Winslow grid generator, his approach uses a single functional to accomplish the adaptive mapping. The critical points of this functional are *harmonic maps*. Meshes obtained by Dvinsky's method enjoy desirable properties of harmonic maps, particularly regularity, or smoothness [15, 27]. Motivated by the work of Dvinsky, a moving mesh finite element strategy based on harmonic mapping was proposed and studied by the authors in [21]. The key idea of this strategy is to construct the harmonic map between the physical space and a parameter space by an iteration procedure.

For two-dimensional domains or three-dimensional cubic domains as considered in [21, 22], the metric on the logical domain is Euclidean. Consequently, the construction of the harmonic mapping only solves some linear equations. However, this is not the case for spherical problems. In fact, if the physical domain is a sphere, then it is natural to set the logical domain as a sphere rather than a convex polygon as used in [21]. In this case, the fact that the curvature of a sphere is positive makes harmonic mappings on spheres not unique. Another difficulty is that the metric on the unit sphere is not an identity anymore, which is different from the cases considered before; see, e.g., [21]. Our goal is to construct a transformation which preserves the good properties of harmonic mappings such as smoothness and uniqueness. Moreover, a scheme for solving the nonlinear equation induced by the metric on a sphere will be designed.

Motivated by the work of Sacks and Uhlenbeck [26], we will use the so-called perturbed harmonic mapping to construct a unique mapping between the physical and logical spaces. It will be seen that perturbed harmonic mappings are useful for handling spherical domains. The nonlinear Euler–Lagrange equations from minimizing the mesh energy functional will be solved by a local Gauss–Seidel-like iterative method. The numerical experiments indicate that a satisfactory approximation to the perturbed harmonic mapping can be obtained using only a few iteration steps. Furthermore, a key step for the moving mesh method is to obtain an improved new mesh based on the underlying numerical solutions and the old meshes. With the spherical geometry, the mesh redistribution has to be designed carefully: efforts have to be made to ensure that the mesh movement occurs only on the sphere, and that all the grid points stay on the sphere during the whole process of mesh redistribution. In this work, appropriate mesh redistribution and solution updating strategies will be investigated.

The rest of the paper is arranged as follows. In section 2, the theory of the perturbed harmonic mapping on a sphere will be briefly reviewed. Section 3 describes the general framework of the moving mesh method on a sphere, including some details of the mesh-redistribution and solution-updating strategies. In section 4, the proposed moving mesh scheme will be applied to solve moving shocks on a sphere and the Fokker–Planck equations derived from liquid crystal simulations. For the latter problem, the solutions of the problem tend to a delta function when the strength of the excluded volume potential is large (see [20]). With a uniform mesh, an extremely fine mesh is required in order to obtain reasonable resolution, but the efficiency can be enhanced significantly by using moving mesh methods. Some concluding remarks will be given in the final section.

## 2. Perturbed harmonic mapping.

Let $\Omega$ and $\Omega_c$ be compact Riemannian manifolds of dimension $n$ with metric tensors $d_{ij}$ and $r_{\alpha\beta}$ in some local coordinates $\vec{x}$ and $\vec{\xi}$, respectively. The energy density of a mapping $\vec{\xi} : \vec{x} \mapsto \vec{\xi}$ is defined by

$$e(\vec{\xi}) = G^{ij} g_{\alpha\beta} \frac{\partial \xi^\alpha}{\partial x^i} \frac{\partial \xi^\beta}{\partial x^j}\,, \tag{2.1}$$

where the standard summation convention is assumed. The corresponding energy functional is then defined by

$$E(\vec{\xi}) = \int_\Omega e(\vec{\xi})dx. \tag{2.2}$$

The harmonic mapping is the mapping which minimizes the above energy functional.

Existence and uniqueness of the harmonic map are guaranteed, provided the Riemannian curvature of $\Omega_c$ is nonpositive and its boundary convex (see Hamilton [15] and Schoen and Yau [27]). However, this result does not apply to the case when the physical and logical domains are spheres, i.e., $\Omega = \Omega_c = \mathcal{S}^2$; in this case, the Riemannian curvature of $\Omega_c$ is positive. In fact, the energy functional (2.2) has the conformal invariance [26], which leads to nonuniqueness of the harmonic mappings. To recover the uniqueness, we follow the work of Sacks and Uhlenbeck [26] to employ a so-called perturbed harmonic mapping approach. More precisely, let

$$E_\alpha(\vec{\xi}) = \int_\Omega (1 + e(\vec{\xi}))^\alpha dx, \tag{2.3}$$

with $\alpha > 1$. It is noted that when $\alpha = 1$ the functionals (2.3) and (2.2) are equivalent. It is known that the Sobolev space of mappings,

$$L_1^{2\alpha}(M, N) = \{\xi \in L_1^{2\alpha}(M, R^k) : \xi(x) \in N\} \subset C^0(M, N), \tag{2.4}$$

is a $C^2$ separable Banach manifold for $\alpha > 1$ [25]. In particular, it is shown in [26] that if $\alpha > 1$, then the critical mappings of $E_\alpha$ in $L_1^{2\alpha}(M, N)$ are $C^\infty$ and are unique in the sense of isometric transformation. Such regularity and uniqueness results provide a sound theoretical background for the moving mesh method based on the perturbed harmonic mappings.

The Euler–Lagrange equation of the energy functional (2.3) is given by

$$-\Delta_{\text{L}-\text{B}}\xi^l - (\alpha - 1)\frac{\nabla|\nabla\xi|^2 \cdot \nabla\xi^l}{|\nabla\xi|^2} + [A(\xi)(d\xi, d\xi)]^l = 0, \tag{2.5}$$

where $\Delta_{\text{L}-\text{B}}$ stands for the Laplace–Beltrami operator,

$$\Delta_{\text{L}-\text{B}} f = \frac{1}{\sqrt{G}} \sum_{i,j} \frac{\partial}{\partial x^i}\left(\sqrt{G}G^{ij}\frac{\partial f}{\partial x^j}\right),$$

and $A(\xi)$ is the second fundamental form of $\Omega_c$. For a sphere, it is known that

$$A(\xi)(d\xi, d\xi) = -|\nabla\xi|^2\xi. \tag{2.6}$$

Moreover, the derivatives in curvilinear coordinates are given by

$$\begin{aligned}(\nabla\xi^m)^j &= G^{ij}\frac{\partial\xi^m}{\partial x^i},\\ \left(\nabla|\nabla\xi|^2\right)^j &= G^{ij}\frac{\partial}{\partial x^i}\left(\sum_t G^{kl}\frac{\partial\xi^t}{\partial x^k}\frac{\partial\xi^t}{\partial x^l}\right).\end{aligned}$$

Consequently, the Euler–Lagrange equation (2.5) can be written in the following divergence form:

$$-\nabla\cdot\left(\alpha|\nabla\xi|^{2(\alpha-1)}\nabla\xi^l\right) + \alpha|\nabla\xi|^{2(\alpha-1)}[A(\xi)(d\xi, d\xi)]^l = 0. \tag{2.7}$$

Using the Stokes law, we can get the weak form of the above equation:

$$\int_{\mathcal{S}^2} \left\{ \sum_l |\nabla\xi|^{2(\alpha-1)} \nabla\xi^l \cdot \nabla\lambda^l - \sum_l |\nabla\xi|^{2\alpha} \xi^l \lambda^l \right\} dV = 0 \qquad \forall\lambda \in L_1^{2\alpha}(M,N). \tag{2.8}$$

The way of obtaining the mapping for the perturbed equation (2.7) is similar to that used for obtaining the conventional harmonic mappings described in [21, 22]. It is shown that with the harmonic mappings the metric distortion can be minimized [13, 26]. Moreover, it is expected that the mesh associated with the mapping (2.7) has certain smoothness and uniqueness.

In this work, the perturbed harmonic mapping will be used to generate adaptive meshes for unstructured triangles in spherical domains. We point out that there are several other successful ways for adaptive remeshing also by minimizing the mesh energy function that depends on the local physical scales; see, e.g., Anderson, Zheng, and Cristini [1], Liao et al. [23], Sun and Huang [17], and Wang and Wang [32].

## 3. Moving mesh methods for spherical domain.

Consider the time-dependent PDEs on a sphere $\Omega = \mathcal{S}^2$,

$$\Psi_t = \mathcal{L}(\Psi) \qquad \text{in } \Omega \times (0,T], \tag{3.1}$$

where $\mathcal{L}$ is some spatial partial differential operator, with initial value

$$\Psi|_{t=0} = \Psi_0 \qquad \text{in } \Omega. \tag{3.2}$$

Let us consider a moving mesh method based on the perturbed harmonic mappings to solve this time-dependent problem. We triangulate the domain $\Omega$ into a mesh $\mathcal{T}_h$ with spherical triangular elements. Equation (3.1) is discretized with finite element approximation on the mesh $\mathcal{T}_h$. Assume a PDE solver is given so that the numerical solution at time level $t_{n+1}$ can be obtained by using the numerical approximation at $t = t_n$. Since our mesh varies with respect to time, we denote the mesh at time step $t_n$ as $\mathcal{T}_h^{(n)}$. The procedure to combine the mesh motion and the PDE solutions is illustrated below:

- *Step* 0. *Initialization.* Set $n = 0$ and $t_0 = 0$. Project the initial function on the initial mesh $\mathcal{T}_h^{(0)}$, which gives $\Psi_h^{(0)}$.
- *Step* 1. *Solution evolution.* Obtain the numerical solutions at $t = t_{n+1}$ on the mesh $\mathcal{T}_h^{(n)}$ by an appropriate PDE solver to obtain $\Psi_h^{(n+1)}$.
- *Step* 2. *Mesh quality evaluation.* Check the mesh quality by certain criteria; if satisfactory, set $\mathcal{T}_h^{(n+1)} = \mathcal{T}_h^{(n)}$, $n = n + 1$, and go to Step 1.
- *Step* 3. *Mesh quality enhancement.* Obtain a more appropriate mesh by some mesh moving techniques, as described later in this section.
- *Step* 4. *Solution updating.* Update the numerical solution $\Psi_h^{(n+1)}$ on the new mesh obtained in Step 3 and go to Step 2.

We point out that the mesh quality (mentioned in Step 2 above) takes into account the shape, alignment, and adaptation features of the elements. For example, the minimum angle, the maximum angle, and the aspect ratio have been used widely to characterize the mesh quality. More details can be found in Huang [18].

### 3.1. Finite element space on sphere.

The sphere is triangulated into a mesh with spherical triangular elements. The Voronoi spherical geodesic grid (see, e.g., [11]) is used in our work. Here we give a finite element space for the discretization

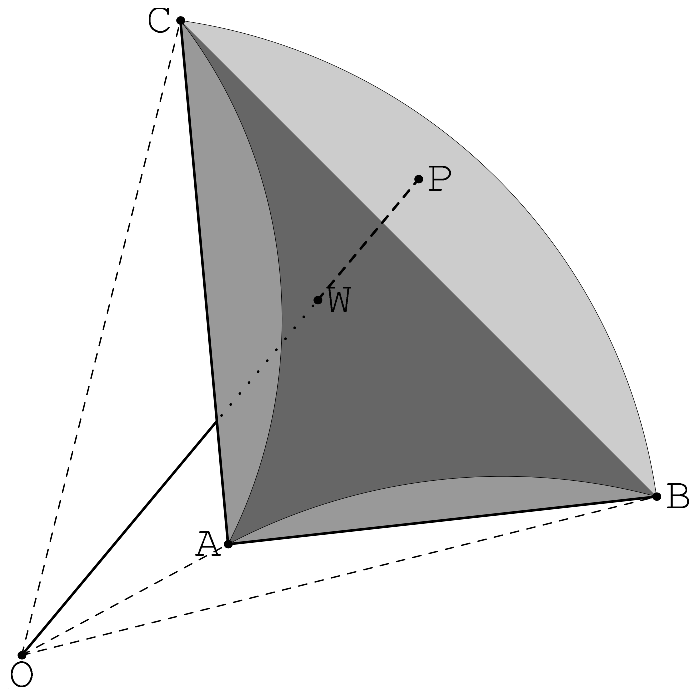


FIG. 1. *The shape function defined on a spherical triangle.*

of the problem (3.1)-(3.2). We will project the linear basis functions of the triangles on polyhedron onto the spherical geodesic triangles. The vertices (counter-clockwise) of a spherical geodesic triangle $K$ are denoted by $A$, $B$, and $C$. The plane triangle $K_p$ has the same vertices. For a point $P \in K$, the intersection point of line $\overrightarrow{OP}$ and $K_p$ is denoted by $W$ (see Figure 1). The three basis functions on spherical geodesic triangle $K$ are defined by

$$\begin{aligned} \lambda_1(\vec{x}) &= \frac{1}{D}(\vec{w} \cdot \vec{a}_2 \times \vec{a}_3), \\ \lambda_2(\vec{x}) &= \frac{1}{D}(\vec{w} \cdot \vec{a}_3 \times \vec{a}_1), \\ \lambda_3(\vec{x}) &= \frac{1}{D}(\vec{w} \cdot \vec{a}_1 \times \vec{a}_2), \end{aligned} \tag{3.3}$$

where

$$\begin{aligned} &\overrightarrow{OA} = \vec{a}_1, \;\; \overrightarrow{OB} = \vec{a}_2, \;\; \overrightarrow{OC} = \vec{a}_3, \;\; \overrightarrow{OP} = \vec{x}, \\ &\overrightarrow{OW} = \vec{w}, \;\; D = (\vec{a}_1 \cdot \vec{a}_2 \times \vec{a}_3), \;\; \vec{w}(\vec{x}) = D\frac{\vec{x}}{\vec{x} \cdot \vec{d}}, \\ &\vec{d} = \vec{a}_1 \times \vec{a}_2 + \vec{a}_2 \times \vec{a}_3 + \vec{a}_3 \times \vec{a}_1. \end{aligned}$$

With this notation, a finite element space on sphere is presented:

$$H_h \triangleq \left\{ \Psi_h \in C^0(\Omega) : \; \Psi_h|_K \in \operatorname{span}\{\lambda_1, \lambda_2, \lambda_3\} \, \forall K \right\}. \tag{3.4}$$

**3.2. Mesh redistribution on sphere.** A key step for the moving mesh method is to obtain an improved new mesh based on the underlying numerical approximations. In our computations, we simply choose the given physical domain, i.e., the sphere, as the logical domain, and we use the Voronoi spherical geodesic grid (see, e.g., [11]) to generate a quasi-uniform mesh with triangular elements on the logical domain. Obviously, this also gives the initial partition for the physical domain. The nodes in the physical mesh are denoted by $X_i$, $1 \leq i \leq N$, and the nodes in the logical mesh by $\Xi_i$, $1 \leq i \leq N$. Initially, since we choose the logical mesh the same as the physical mesh, we have $\Xi_i = X_i$. The physical mesh will be redistributed at each time step, while the logical mesh will be kept unchanged during the entire process of the numerical computations. For each time step, by numerically solving the Euler–Lagrange equation (2.7), we can obtain a discrete transformation between the physical domain $\vec{x}_h$ and the logical domain $\vec{\xi}_h$. The logical mesh is then given such that $\Xi_i = \vec{\xi}_h(X_i)$. The metric on the physical domain is given according to the given PDEs and their numerical solution, which is called the *monitor function*.

On the sphere, there are no local coordinates to cover the whole domain $\mathcal{S}^2$. Therefore, we have to choose suitable local coordinates. The metric for the logical domain is the natural metric of sphere as a submanifold embedded in $\mathbb{R}^3$. Since the Euler–Lagrange equation (2.7) is nonlinear, it will be solved numerically by a point-by-point iterative approach. Assume that we are considering a node in the physical mesh, $X_k$, whose corresponding node in the logical mesh is denoted by $\Xi_k$. Denote the set of triangles in the physical domain with $X_k$ as a vertex by $\mathbf{S}_k$ and the corresponding set in the physical domain by $\mathbf{S}_{c,k}$. We will set the local coordinate around $X_k$ in the physical domain and the local coordinate around $\Xi_k$ in the logical domain. It is natural to eliminate one component whose absolute value is the largest among the three components. This will give two-dimensional local coordinates. For example, for $X_k = (x_k^1, x_k^2, x_k^3)$ set, if $|x_k^1| \geq |x_k^2|$ and $|x_k^1| \geq |x_k^3|$, then we will let $(x_k^1, x_k^2) \leftarrow (x_k^2, x_k^3)$ to be the local coordinate around $X_k$. The same strategy is employed to choose the local coordinate around $\Xi_k$. We then have the Euler–Lagrange equation (2.7) around the node $\Xi_k$:

$$\begin{aligned} \nabla \cdot \left( |\nabla \xi|^{2(\alpha-1)} \nabla \xi^l \right) + |\nabla \xi|^{2\alpha} \xi^l = 0 \quad \text{for } l = 1,\, 2,\, 3, \\ \xi|_{\partial \mathbf{S}_k} = \xi_b. \end{aligned} \tag{3.5}$$

In the physical domain, around $X_k$ the patch of surface on the unit sphere can be expressed as

$$\left(x^3\right)^2 = 1 - \left(x^1\right)^2 - \left(x^2\right)^2 , \tag{3.6}$$

and the metric of the physical domain is taken as a submanifold embedded in $\mathbb{R}^3$:

$$\tilde{G}^{ij}_{i,j=1,2} = \begin{pmatrix} 1 - \left(x^1\right)^2 & x^1 x^2 \\ x^1 x^2 & 1 - \left(x^2\right)^2 \end{pmatrix}. \tag{3.7}$$

The metric $G^{ij}$ on the physical domain around $X_k$ is then set as

$$G^{ij} = m(u_h) \tilde{G}^{ij}, \tag{3.8}$$

where $m(u_h)$ is the monitor function given for the problem under consideration. The choice of the monitor function $m$ is very subtle, which is an indicator of the solution singularity and is problem dependent. A general discussion of the monitor function can be found in [6, 29].

Now on the patch of triangles $\mathbf{S}_k$, we will solve a Dirichlet boundary-value problem for (3.5), with linear boundary mapping segmentwise on $\partial\mathbf{S}_k$ determined by the vertex mapping $\Xi_n = \vec{\xi}(X_n)\ \forall X_n \in \mathbf{S}_k, n \neq k$. The mapping in $\mathbf{S}_k$ is approximated by a piecewise linear mapping on every triangle in $\mathbf{S}_k$. Thus the mapping can be given only if $\vec{\xi}(X_k)$ is given. Using the standard Galerkin approximation for (3.5) in $\mathbf{S}_k$, we have only one test function, which is the pyramid-like basis function located at $X_k$. This gives

$$\int_{\mathbf{S}_k} \left\{ |\nabla\xi|^{2(\alpha-1)}\nabla\xi^l \cdot \nabla\lambda^k - |\nabla\xi|^{2\alpha}\xi^l\lambda^k \right\} dV = 0 \quad \text{for } l = 1,\, 2,\, 3, \tag{3.9}$$

where $\lambda^k$ is the test function for $X_k$, which is an algebraic system with order $2\alpha + 1$ for three unknowns. We will solve (3.9) approximately. On $\mathbf{S}_k$, we denote $\vec{\xi}_h$ by

$$\vec{\xi}_h = \sum_n \Xi_n \lambda_h^n + \Xi_k \lambda^k\,, \tag{3.10}$$

where the summation for $n$ is taken over all vertices in $\mathbf{S}_k$ except $X_k$. By taking $|\nabla\xi|$ in (3.9) explicitly, i.e., $\xi_h$ in $|\nabla\xi|$ takes the currently available $\Xi_k$ as its value at $X_k$, we then get an explicit approximation for (3.9):

$$\Xi_k^l \approx \frac{\int_{\mathbf{S}_k} \sum_n |\nabla\xi|^{2\alpha}\Xi_n^l \lambda^l \lambda^k dV - \sum_n \int_{\mathbf{S}_k} |\nabla\xi|^{2(\alpha-1)} G^{ij} \Xi_n^l \frac{\partial\lambda^n}{\partial x^i}\frac{\partial\lambda^k}{\partial x^j} dV}{\int_{\mathbf{S}_k} \big( - |\nabla\xi|^{2\alpha}\lambda^k\lambda^k + |\nabla\xi|^{2(\alpha-1)} G^{ij} \frac{\partial\lambda^k}{\partial x^i}\frac{\partial\lambda^k}{\partial x^j}\big) dV}. \tag{3.11}$$

Below we will describe the algorithm for mesh redistribution, which is similar to the one proposed in [21]. In the first step, we iterate for $k$ from 0 to $N$, using (3.11) to update all logical nodes $\Xi_k$ until convergence (i.e., the difference between the obtained $\Xi_k$ and initial (fixed) $\Xi_k^{(0)}$ is smaller than a given tolerance). The tolerance for the convergence criteria is of the order $\mathcal{O}(h_l)$, where $h_l$ is the typical element size in the logical mesh. The converged result is denoted by $\Xi_k^\star$, and we then get the difference between $\Xi_k^\star$ and the initial logical mesh as

$$\delta\Xi_k = \Xi_k^{(0)} - \Xi_k^\star. \tag{3.12}$$

In the second step, we use the above difference to obtain the mesh redistribution information in the physical mesh difference. Again we will choose the same local coordinate as described above. For a node $X_k$ in the physical mesh, we first compute the mesh motion vector:

$$\delta X_k^l = \frac{\sum_{K\in\mathbf{S}_k} |E| \left.\frac{\partial x^l}{\partial\vec{\xi}}\right|_K \delta\Xi_k}{\sum_{K\in\mathbf{S}_k} |E|} \qquad \text{for } l = 1, 2, \tag{3.13}$$

where the Jacobian matrix $\partial\vec{x}/\partial\vec{\xi}\,\big|_K$ is computed by using the fact that the linear element is used (see also [21]):

(3.14)

$$\begin{pmatrix} \Xi_{E_1}^1 - \Xi_{E_0}^1 & \Xi_{E_2}^1 - \Xi_{E_0}^1 \\ \Xi_{E_1}^2 - \Xi_{E_0}^2 & \Xi_{E_2}^2 - \Xi_{E_0}^2 \end{pmatrix} \begin{pmatrix} \dfrac{\partial x^1}{\partial\xi^1} & \dfrac{\partial x^1}{\partial\xi^2} \\ \dfrac{\partial x^2}{\partial\xi^1} & \dfrac{\partial x^2}{\partial\xi^2} \end{pmatrix} = \begin{pmatrix} X_{E_1}^1 - X_{E_0}^1 & X_{E_2}^1 - X_{E_0}^1 \\ X_{E_1}^2 - X_{E_0}^2 & X_{E_2}^2 - X_{E_0}^2 \end{pmatrix};$$

and $E_0$, $E_1$, and $E_2$ are the subscripts of the three vertices of the element $K$ in the physical mesh. The coordinate in the third dimension is excluded, as explained earlier (i.e., with the way of choosing the local coordinate). On the other hand, to ensure that the mesh movement occurs only on the sphere, the third component can be recovered by using

$$\delta X_k^3 \to \operatorname{sgn}(X_k^3)\sqrt{1-(X_k^1+\delta X_k^2)^2-(X_k^2+\delta X_k^2)^2}-X_k^3. \tag{3.15}$$

In the third step, we will control the magnitude of each redistribution to avoid mesh tangling. For the element $K \in \mathcal{T}_h$ with vertices $E_0$, $E_1$, and $E_2$, we solve the cubic equation (in terms of $\mu$)

$$\det\begin{pmatrix} X_{E_0}+\mu\cdot\delta X_{E_0} & X_{E_1}+\mu\cdot\delta X_{E_1} & X_{E_2}+\mu\cdot\delta X_{E_2}\end{pmatrix}=0, \tag{3.16}$$

where $\delta X_{E_k}$ is determined by (3.13). Denote the smallest positive root of (3.16) by $\mu_K$. The mesh motion parameter is then set as

$$\mu^\star = \min_{K\in\mathcal{T}_h}\{1;\ \mu_K/2\}. \tag{3.17}$$

It is obvious that the mesh tangling can be avoided if we move the mesh using the formula $X_k \leftarrow X_k + \mu^\star\delta X_k$.

In the last step, we update the mesh moving direction by

$$\delta X_k \leftarrow \frac{X_k+\mu^\star\delta X_k}{|X_k+\mu^\star\delta X_k|}-X_k\,. \tag{3.18}$$

This is to make sure that all the grid points are moving along the sphere. The new mesh nodes are then given by $X_k+\delta X_k$. It can be verified that the new nodes are all on the sphere.

**3.3. Solution interpolation on the new mesh.** After redistributing the mesh in the physical domain $\Omega$, we need to transfer the solution information on the old mesh to the new mesh. Let $\delta\vec{x} := \sum_{i=1}^N \delta X_i\lambda^i$, where $\lambda^i$ is the basis function of $H_h$. Assume that a finite element solution $\Psi_h^{(n)} = \sum_{i=1}^N \Psi_i^{(n)}\lambda^i$ is given at $t=t_n$. Let $\tau$ be a virtual time variable. We will regard the mesh motion from the old mesh to the new mesh as a linear homotopy for $\tau$ goes from 0 to 1. Consequently, the mesh flow with the virtual time $\tau$ is $\vec{x}(\tau) \triangleq \vec{x}_{\text{old}}+\tau\delta\vec{x}$. For an arbitrary point $\vec{x}_0 \in \mathcal{T}_h$, the basis function $\lambda^i(\vec{x})$ associated with $\vec{x}_0$ will be varying linearly with $\tau$. Since $\lambda^i(\vec{x})$ is defined on the physical mesh, it is denoted by $\lambda^i(\vec{x};\tau)$. Direct calculation shows that

$$\frac{d\lambda^i(\vec{x};\tau)}{d\tau} = \delta\vec{x}\cdot\nabla_{\vec{x}}\lambda^i(\vec{x};\tau). \tag{3.19}$$

Assume that the numerical solution on mesh $\vec{x}(\tau)$ is of the form

$$\Psi_h(\vec{x};\tau) = \sum_{i=1}^N \Psi_i(\tau)\lambda^i(\vec{x};\tau), \tag{3.20}$$

which takes the starting value $\Psi_h(\cdot;0)=\Psi_h^{(n)}$. The main idea of the solution updating is to let the variation of $\Psi_h(\cdot;\tau)$ with respect to $\tau$ vanish in the test function space:

$$\left\langle \frac{d\Psi_h(\cdot;\tau)}{d\tau}, \Phi_h\right\rangle = 0 \quad \forall\Phi_h\in H_h. \tag{3.21}$$

Using (3.19) and (3.20) gives

$$\sum_{i=1}^{N}\int_{\Omega}\left\{\frac{d\Psi_i}{d\tau}\lambda^i(\cdot;\tau)\lambda^j(\cdot;\tau)+\Psi_i\delta\vec{x}\cdot\nabla_{\vec{x}}\lambda^i(\cdot;\tau)\lambda^j(\cdot;\tau)\right\}dx=0,\quad 1\le j\le N. \tag{3.22}$$

The above ODE system can be solved by some appropriate ODE solver, e.g., a three-step Runge–Kutta method [21, 22].

**4. Numerical experiments.** We first discuss the perturbation parameter $\alpha$ in (3.5): in the numerical computations reported in this section it is chosen as 1.5. It is known [25] that $\alpha$ must be larger than 1 in order to ensure the uniqueness of the mapping. If $\alpha$ is close to 1, the critical mapping of $E_\alpha$ is close to a harmonic mapping. This implies that $\alpha$ should be chosen to be approximately 1. On the other hand, it is found that larger values of $\alpha$ increase the solution efficiency for (2.8). In fact, larger values of $\alpha$ can speed up the convergence of the mesh movement: note that the nonlinear Euler–Lagrange equations are solved by a local Gauss–Seidel-like iterative method. With a few numerical tests, it is found that one of the good choices for the parameter $\alpha$ should be approximately 1.5.

*Example* 4.1 (moving steep front on a sphere). Our first example is to compute a moving steep front along the direction $\vec{b} := (1,1,1)^T$ governed by the equation

$$\partial\Psi_t+\Psi\vec{b}\cdot\nabla^*\Psi=\epsilon\Delta_{L-B}\Psi, \tag{4.1}$$

which is defined on the unit sphere, where $\nabla^*$ is the surface gradient. The initial condition is given by

$$\Psi(x,y,z;0)=\left(1+\exp\left(\frac{x+y+z+1}{2\epsilon}\right)\right)^{-1}. \tag{4.2}$$

We first briefly discuss the PDE algorithm to be used. For this example, a simple finite element discretization as described in section 3.1 is employed. The weak formulation of (4.1) reads as follows: Find $\Psi\in H^1(\Omega)$ such that

$$\int_\Omega\frac{\partial\Psi}{\partial t}vdx+\int_\Omega\Psi\vec{b}\cdot\nabla^*\Psi vdx=-\epsilon\int_\Omega\nabla^*\Psi\cdot\nabla^*vdx\quad\forall v\in H^1(\Omega). \tag{4.3}$$

In the finite element space $H_h$, the above weak form can be approximated as follows: Find $\Psi_h\in H_h$, such that

$$\int_\Omega\frac{\partial\Psi_h}{\partial t}v_hdx+\int_\Omega\Psi_h\vec{b}\cdot\nabla^*\Psi_hv_hdx=-\epsilon\int_\Omega\nabla^*\Psi_h\cdot\nabla^*v_hdx\quad\forall v_h\in H_h(\Omega). \tag{4.4}$$

For the temporal approximation, an explicit three-step Runge–Kutta scheme is applied; see, e.g., [7].

In our computations, the viscosity coefficient $\epsilon$ is chosen as 0.01. For the convection dominant problem, more mesh points are required around the wave front, which makes the use of the moving mesh methods meaningful. The monitor function $m$ defined in (3.8) is taken as the standard gradient-based one, i.e.,

$$m(\Psi)=\sqrt{1+\beta|\nabla\Psi|^2}. \tag{4.5}$$

For problems with a moving steep front, the parameter $\beta$ has to be set as a large number, say between $10^2$ to $10^3$; see, e.g., [6]. A large value of $\beta$ can usually cluster

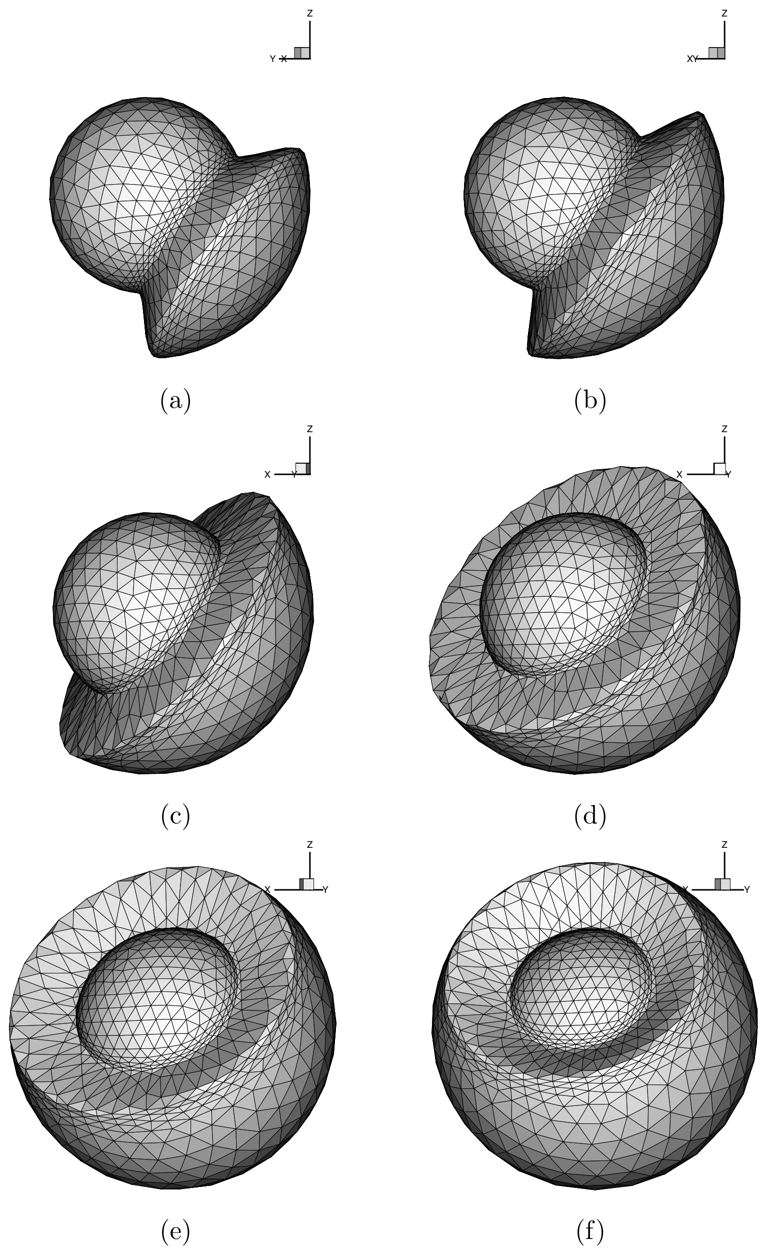


FIG. 2. *Example* 4.1. *The adaptive meshes on a sphere with* 800 *nodes from* $t = 0$ *to* $t = 1.5$.

more grid points around the steep front. On the other hand, if $|\nabla\Psi|$ in (4.5) is replaced by $|\nabla\Psi|/\max_\Omega|\nabla\Psi|$, then the parameter $\beta$ can be chosen as an $\mathcal{O}(1)$ number; see, e.g., [7]. In our computations, the parameter $\beta$ in (4.5) is chosen as $10^3$.

In Figure 2, the moving meshes obtained by using our moving mesh scheme with 800 nodes (about 1600 triangular elements) are plotted. Together in this figure are the numerical solutions for $\Psi$ at various time levels. Figure 2(a) shows the initial

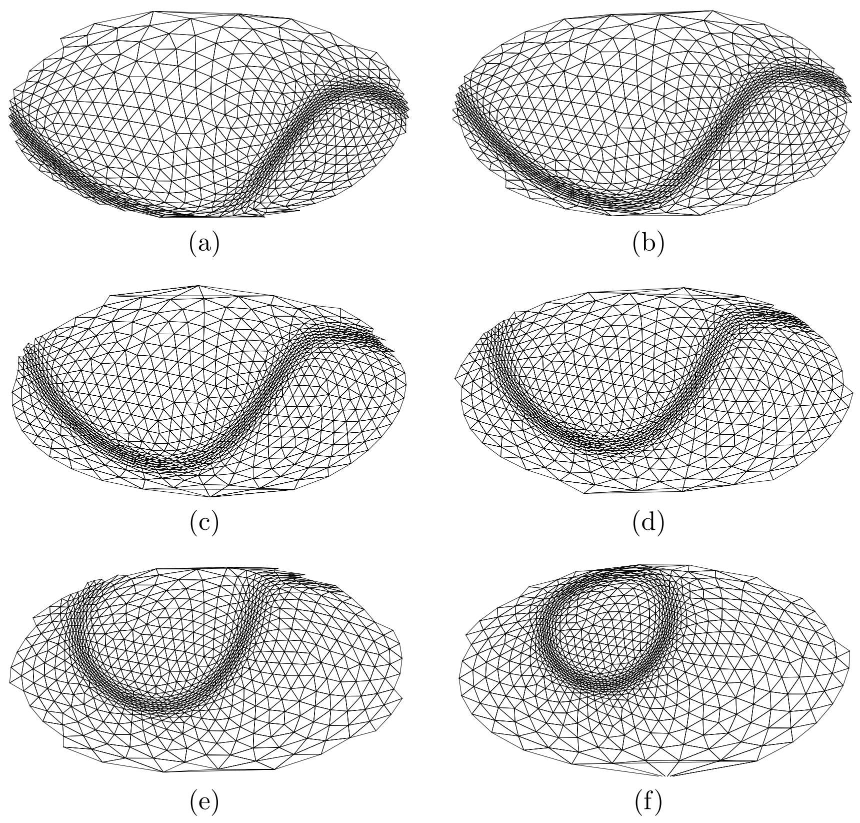


FIG. 3. *Example* 4.1. *The adaptive meshes on planisphere using* 800 *nodes from* $t = 0$ *to* $t = 1.5$.

state: the initial front is at $x + y + z + 1 = 0$, where a stage can be seen. The solution is 0 on the top left side of the front and 1 on the bottom right side. It can be seen that the front moves along the direction $(1, 1, 1)^T$ on the sphere. In Figures 2(b)–(f), the area with solution value 1 becomes bigger, and the area of value 0 becomes smaller as time increases. At $t = 1.5$, the solution is mostly 1 on the sphere and is 0 only on a small concave area. In order to display the numerical solution more clearly, the figures are plotted from various visual angles at different time levels. Moreover, in order to have a better idea of the mesh distribution the meshes are projected to a planisphere and are plotted in Figure 3. It is seen that our moving mesh scheme adapts the mesh extremely well in the regions with large solution gradients.

Figure 4 shows the meshes and solutions obtained by the proposed moving mesh algorithm with 800 nodes (left) and by using a uniform mesh approach with 3200 nodes (right) at $t = 0.6$. It is observed from the figure that the moving mesh method with fewer nodes resolves the sharp front much more accurately than the corresponding uniform mesh approach with more nodes. Furthermore, the numerical oscillation in the uniform mesh solution is not seen in the moving mesh solution. In fact, it is found in our numerical computations that it requires more than 7000 nodes in a uniform mesh in order to obtain the same resolution of the moving mesh solution with 800 nodes.

*Example* 4.2 (solution of the Fokker–Planck equation). The second example is

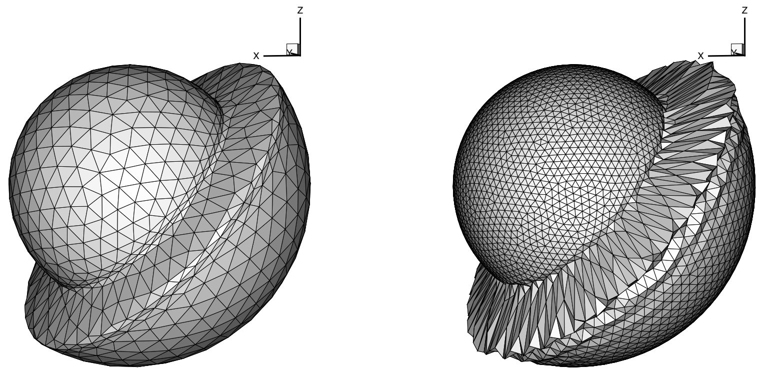


FIG. 4. *Example* 4.1. *The adaptive mesh on sphere with* 800 *nodes (left) and uniform mesh with* 3200 *nodes (right) at* $t = 0.6$.

concerned with the Doi model [9, 16], which is commonly used in studying rod-like polymer fluids. In this model, a homogeneous population of equal rods is described by an orientational distribution function $\Psi(\vec{x}, t)$. The evolution of $\Psi(\vec{x}, t)$ is modeled with the Smoluchowski equation or the Fokker–Planck equation [10]:

$$\frac{\partial \Psi}{\partial t} = \frac{1}{De}\mathcal{R}\cdot(\mathcal{R}\Psi + \Psi\mathcal{R}V_{ev}) \quad \text{on } \Omega = \mathcal{S}^2, \tag{4.6}$$

where $De$ is the Deborah number, $\mathcal{R}$ is the rotational operator $\mathcal{R} \equiv \vec{x} \times \partial/\partial\vec{x}$, and $V_{ev}$ is an effective excluded-volume potential which takes the Maier–Saupe form:

$$V_{ev}(\vec{x}) = \frac{3}{2}U\int_\Omega |\vec{x}\times\vec{x'}|^2\Psi(\vec{x'})dx'. \tag{4.7}$$

In (4.7), $U$ is the nondimensional potential intensity which is proportional to the concentration of the polymers. The initial distribution function is taken as

$$\Psi(\vec{x}, 0) = \frac{1}{4\pi} + \frac{1}{16}\sqrt{\frac{5}{\pi}}\Big(3(\vec{x}_0\cdot\vec{x})^2 - 1\Big),$$

where $\vec{x}_0$ is a constant unit vector.

As the potential intensity increases, the orientational distribution function becomes sharply peaked and behaves like a $\delta$-function. To resolve the solutions in this case, it is necessary to use some adaptive grid methods. Here we will use the framework introduced in the last section, with a few additional details described below.

*Weak formulation.* The weak formulation of (4.6) reads as follows: Find $\Psi \in H^1(\Omega)$, such that

$$\int_\Omega \frac{\partial\Psi}{\partial t}vdx + \frac{1}{De}\int_\Omega \mathcal{R}\Psi\cdot\mathcal{R}vdx = -\frac{1}{De}\int_\Omega \Psi\mathcal{R}V(\Psi)\cdot\mathcal{R}vdx \quad \forall v \in H^1(\Omega). \tag{4.8}$$

In the finite element space $H_h$, the above weak form can be approximated as follows: Find $\Psi_h \in H_h$, such that

$$\int_\Omega \frac{\partial\Psi_h}{\partial t}v_hdx + \frac{1}{De}\int_\Omega \mathcal{R}\Psi_h\cdot\mathcal{R}v_hdx = -\frac{1}{De}\int_\Omega \Psi_h\mathcal{R}V(\Psi_h)\cdot\mathcal{R}v_hdx \quad \forall v_h \in H_h(\Omega). \tag{4.9}$$

For the temporal approximation, similar to [7] we use a semi-implicit three-step Runge–Kutta scheme to solve (4.9): Find $\Psi_h^{n+1} \in H_h$, such that the following hold.

- *Step* 1.

$$\begin{aligned}&\int_\Omega \frac{\Psi_h^{(1)}-\Psi_h^n}{\triangle t/3} v_h dx + \frac{1}{De}\int_\Omega \mathcal{R}\Psi_h^{(1)}\cdot \mathcal{R}v_h dx\\ &= -\frac{1}{De}\int_\Omega \Psi_h^n \mathcal{R}V(\Psi_h^n)\cdot \mathcal{R}v_h dx \quad \forall v_h\in H_h\,.\end{aligned} \tag{4.10}$$

- *Step* 2.

$$\begin{aligned}&\int_\Omega \frac{\Psi_h^{(2)}-\Psi_h^n}{\triangle t/2} v_h dx + \frac{1}{De}\int_\Omega \mathcal{R}\Psi_h^{(2)}\cdot \mathcal{R}v_h dx\\ &= -\frac{1}{De}\int_\Omega \Psi_h^{(1)} \mathcal{R}V(\Psi_h^{(1)})\cdot \mathcal{R}v_h dx \quad \forall v_h\in H_h\,.\end{aligned} \tag{4.11}$$

- *Step* 3.

$$\begin{aligned}&\int_\Omega \frac{\Psi_h^{n+1}-\Psi_h^n}{\triangle t} v_h dx + \frac{1}{De}\int_\Omega \mathcal{R}\Psi_h^{n+1}\cdot \mathcal{R}v_h dx\\ &= -\frac{1}{De}\int_\Omega \Psi_h^{(2)} \mathcal{R}V(\Psi_h^{(2)})\cdot \mathcal{R}v_h dx \quad \forall v_h\in H_h\,.\end{aligned} \tag{4.12}$$

Note that in each step above the diffusion term is treated implicitly while the nonlinear term is treated explicitly. With this treatment, a standard stability criterion of CFL type applies; and in our computations a uniform time step is used. By doing this, it is well known that the moving mesh method does not gain much in time-stepping, since the time steps are controlled by the smallest size of the finite elements. One possible way of further speed-up in temporal discretization is to use the local time-stepping technique; see, e.g., [28].

*Monitor functions.* For a piecewise linear approximation $\phi_h$ to a function $\phi$, the following formula is adopted to approximate the computational error:

$$|\phi-\phi_h|_{1,\Omega} \sim \sqrt{\sum_{l\,:\,\text{interior edge}} \int_l \left[\mathcal{R}\phi_h\cdot \vec{t_l}\right]^2 dl}, \tag{4.13}$$

where $\vec{t_l}$ is the unit tangential vector of the edge $l$ and $[\cdot]$ is the jump along the edge $l$; see, e.g., [7, 19]. With this a posteriori interpolant error estimator, a monitor function is proposed as

$$m(\Psi_h)|_K = \frac{1}{|K|}\sum_{l\subset\partial K} |l| \int_l \left[\mathcal{R}\Psi_h\cdot \vec{t_l}\right]^2 dl \quad \forall K\in \mathcal{T}_h\,. \tag{4.14}$$

The instantaneous average molecular orientation is quantified by the order parameter tensor

$$\mathbf{S} = \langle \mathbf{xx}\rangle - \frac{1}{3}\mathbf{I}. \tag{4.15}$$

The scalar order parameter $S=\frac{3}{2}\lambda$ represents the degree of molecular alignment, where $\lambda$ is the eigenvalue with the largest absolute value for $\mathbf{S}$. It lies in the range $0\sim 1$ with $S=0$ for an isotropic phase and $S=1$ for rods that are all aligned in the same direction. Figure 5 shows the evolution of the scalar order parameter $S$

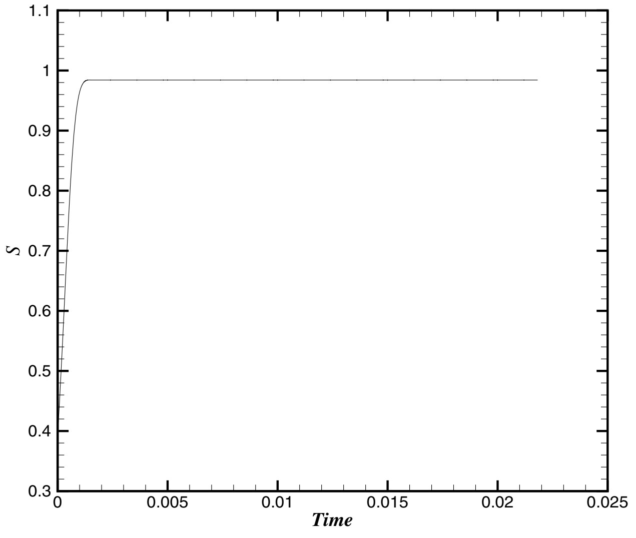


FIG. 5. *Example* 4.2. *Evolution of the scalar order parameter $S$ with $U = 1000$.*

with $U = 100$. In this case, the stationary scalar order parameter $S$ is very close to 1, indicating that all the molecular directions converge to one direction and the distribution function is very singular.

Figure 6 shows the meshes and the corresponding orientational distribution functions with $U = 100$ and 1000. To better demonstrate the singular behaviors of the solutions, the sphere is projected into a planisphere. The peak value of the distribution function with $U = 1000$ reaches $\Psi \sim 200$. Since the integral of the distribution function equals to 1, the area of nonzero $\Psi$ is about $\frac{1}{200}$, where the solution has large gradients. Moreover, since the area of the unit sphere is equal to $4\pi$, the compact support occupies only $\frac{1}{200}/(4\pi) \sim \frac{1}{2000}$ portion of the sphere. In order to put 25 $(= 5 \times 5)$ mesh points in such a small area, a uniform mesh requires a total of 50000 mesh points, while our moving mesh approach needs only 1600 points. By a close-up look of the mesh structure in Figure 7, it is seen that with the moving mesh approach more grid points are pushed into the (small) blow-up region.

More comparisons are made for Example 4.2 with $U = 1000$. In Figure 8, we compare the numerical solutions obtained by using the uniform mesh (nodes 12568, 25542, and 50266, respectively) and by using the adaptive moving mesh (1600 nodes). It is seen that the moving mesh solution with 1600 nodes is more accurate than the uniform mesh solution with 25542 nodes and is comparable to the uniform mesh solution with 50266 nodes. This is quite consistent with our estimate above.

Finally, it is necessary to compare the total CPU time usage for these computations. Table 1 gives the relevant CPU time usages with various numbers of nodes. The time step used is $\Delta t = 10^{-6}$ and the total time is $T = 2 \times 10^{-3}$. For this particular computation, the memory savings with the moving mesh method is about 1:30 and the speed-up is over 15.

## 5. Conclusions.

In this paper, an effective moving mesh method is proposed to resolve steep solutions generated by PDEs on a sphere. The method exploits harmonic maps and invokes a regularization procedure to overcome the lack of convexity in the setup. Some special techniques for handling local coordinates on a sphere are also

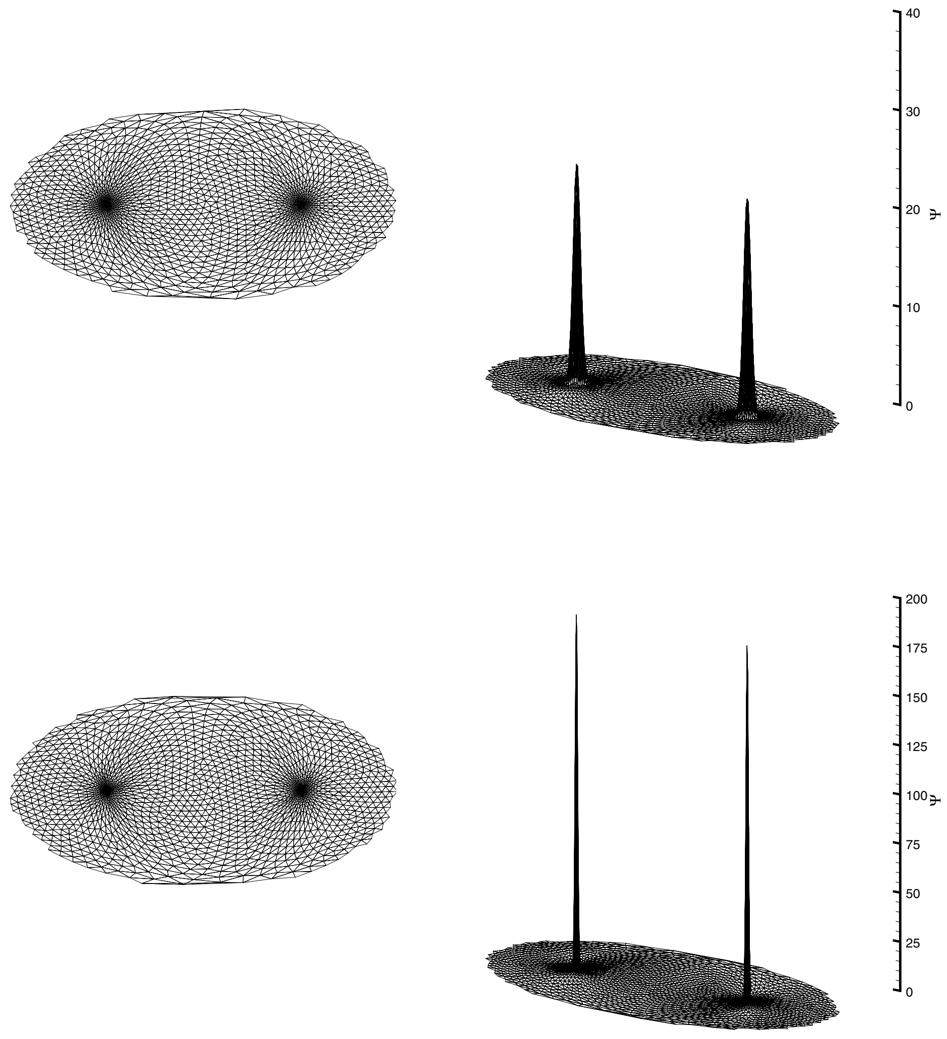


Fig. 6. *Example* 4.2. *The orientational distribution function obtained by using* 1600 *nodes and* 3196 *elements for* $U = 100$ *(top) and* $U = 1000$ *(bottom). The mesh on the sphere is projected to a disk (left) and the corresponding solution is shown on the right.*

designed. Efforts have been made to ensure that the mesh movement occurs only on the sphere and all the grid points remain on the sphere during the entire process of mesh redistribution. Two examples are given, one for a convection-dominated problem and the other for the Fokker–Planck equation on the sphere.

It is demonstrated that the moving mesh methods can not only enhance the solution *speed*, but can also substantially reduce the computational *storage*. The importance of increasing efficiency is obvious, but the latter part (of reducing storage) is also important for many practical problems, in particular higher-dimensional prob-

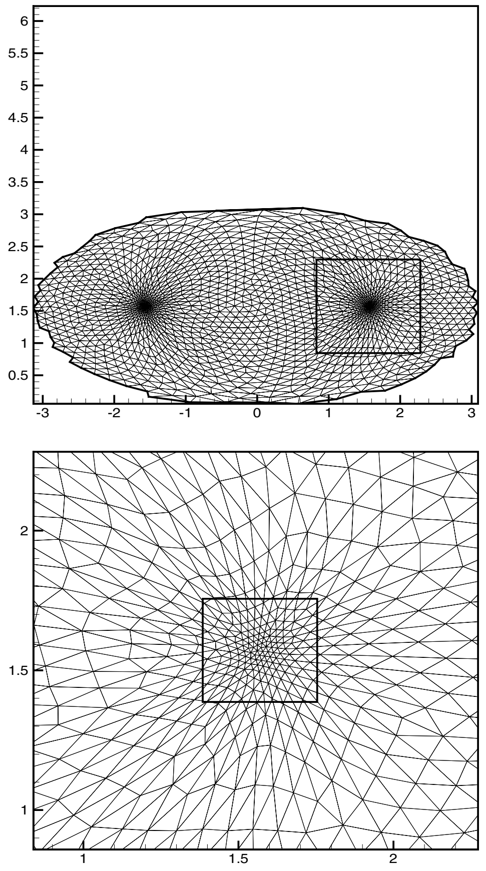
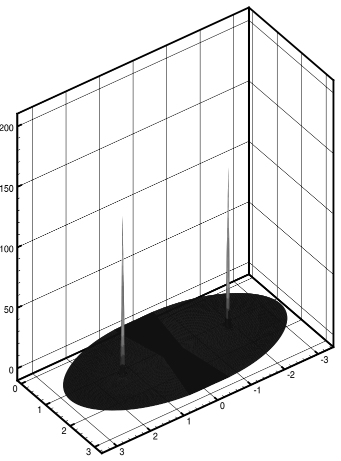


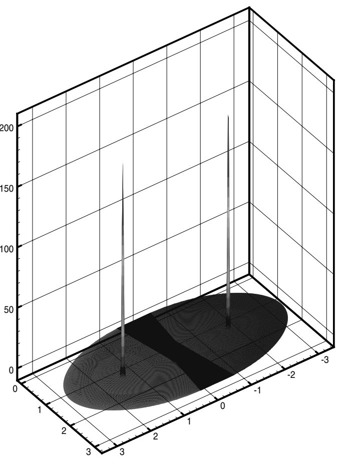
Fig. 7. *Example* 4.2. *Close-up of the adaptive mesh for* $U = 1000$ *(bottom figure of Figure* 6*).*

lems. For example, a large class of problems in complex fluid simulations are posed on spheres, for which Example 4.2 is a simple case. For the microscopic model of the polymer fluid, a well-known model is the rigid dumbbell model of Doi [9, 16]. In the Doi model, the orientation of a rod is determined by a pseudovector $\mathbf{u}$ on the unit sphere. The existing numerical schemes include a spectral-type method with spherical harmonic functions as basis functions (spherical harmonic expansion method); see, e.g., [14]; and a standard finite element approach; see, e.g., [20]. Since the underlying equations for the complex fluid models (such as the Fokker–Planck type) involve several dimensions in both physical and phase spaces, the required storage for the classical methods may be unrealistically large. It is seen that the moving mesh approach can use much less storage to obtain the required resolution. The numerical technique developed in this work is currently being used to study the rheological behavior of rod-like polymers governed by the Doi model; see, e.g., [8].

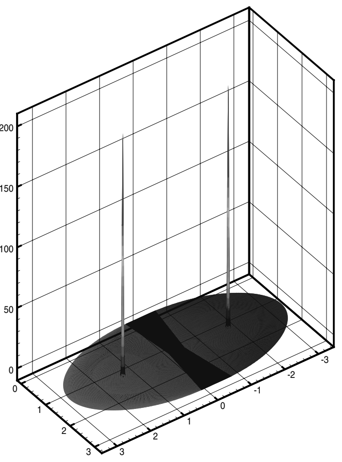
Finally, we point out that many important practical problems have been posed in spherical geometries, such as climate modeling and ocean modeling. One of our future projects is to extend the numerical techniques developed in this work to solve some practical problems defined on spheres with solution singularity.



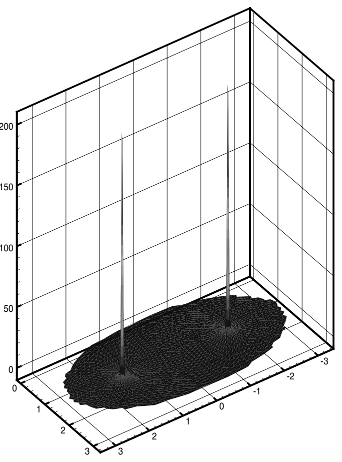
(a)



(b)



(c)



(d)

Fig. 8. *Example* 4.2. *The orientational distribution function obtained by using a uniform mesh with* (a) 12568 *nodes,* (b) 25542 *nodes,* (c) 50266 *nodes; and* (d) *by using a moving mesh* 1600 *nodes.* $U = 1000$.

Table 1
*Example* 4.2. *CPU time (seconds) used for uniform and moving mesh computations.*

| | # of nodes | CPU time used |
|---|---:|---:|
| Uniform mesh | 12568 | 11977 |
| Uniform mesh | 25542 | 24324 |
| Uniform mesh | 50266 | 48473 |
| Moving mesh | 1600 | 3044 |
| Moving mesh | 3200 | 5981 |

## REFERENCES

[1] A. Anderson, X. Zheng, and V. Cristini, *Adaptive unstructured volume remeshing.* I. *The method*, J. Comput. Phys., 208 (2005), pp. 616–625.

[2] B. N. Azarenok and T. Tang, *Second-order Godunov-type scheme for reactive flow calculations on moving meshes*, J. Comput. Phys., 106 (2005), pp. 48–80.

[3] M. J. Baines, M. E. Hubbard, and P. K. Jimack, *A moving mesh finite element algorithm for the adaptive solution of time-dependent partial differential equations with moving boundaries*, Appl. Numer. Math., 54 (2005), pp. 450–469.

[4] G. Beckett, J. A. Mackenzie, and M. L. Robertson, *A moving mesh finite element method for the solution of two-dimensional Stephan problems*, J. Comput. Phys., 168 (2001), pp. 500–518.

[5] W. M. Cao, W. Z. Huang, and R. D. Russell, *An r-adaptive finite element method based upon moving mesh PDEs*, J. Comput. Phys., 149 (1999), pp. 221–244.

[6] W. M. Cao, W. Z. Huang, and R. D. Russell *A study of monitor functions for two-dimensional adaptive mesh generation*, SIAM J. Sci. Comput., 20 (1999), pp. 1978–1994.

[7] Y. Di, R. Li, T. Tang, and P. W. Zhang, *Moving mesh finite element methods for the incompressible Navier–Stokes equations*, SIAM J. Sci. Comput., 26 (2005), pp. 1036–1056.

[8] Y. Di and P. W. Zhang, *Moving mesh kinetic simulation for sheared rodlike polymers with high potential intensities*, Commun. Comput. Phys., 1 (2006), pp. 859–873.

[9] M. Doi, *Molecular dynamics and rheolgical properties of concentrated solutions of rodlike polymers in isotropic and liquid crystalline phases*, J. Polym. Sci., 19 (1981), pp. 229–243.

[10] M. Doi and S. F. Edwards, *The Theory of Polymer Dynamics*, Oxford University Press, New York, 1986.

[11] Q. Du and M. Gunzburger, *Grid generation and optimization based on centroidal Voronoi tessellations*, Appl. Math. Comput., 113 (2002), pp. 591–607.

[12] A. S. Dvinsky, *Adaptive grid generation from harmonic maps on Riemannian manifolds*, J. Comput. Phys., 95 (1991), pp. 450–476.

[13] J. Eell and J. H. Sampson, *Harmonic mapping of Riemannian manifolds*, Amer. J. Math., 86 (1964), pp. 109–160.

[14] M. G. Forest, Q. Wang, and R. Zhou, *The flow-phase diagram of Doi-Hess theory for sheared nematic polymers* II: *Finite shear rates*, Rheological Acta, 44 (2004), pp. 80–95.

[15] R. Hamilton, *Harmonic Maps of Manifolds with Boundary*, Lecture Notes in Math. 471, Springer-Verlag, New York, 1975.

[16] S. Z. Hess, *Fokker-Planck-equation approach to flow alignment in liquid crystals*, Z. Naturforsch. A, 31 (1976), pp. 1034–1037.

[17] W. Huang and W. Sun, *Variational mesh adaptation* II: *Error estimates and monitor functions*, J. Comput. Phys., 184 (2003), pp. 619–648.

[18] W. Z. Huang, *Mathematical principles of anisotropic mesh adaptation*, Commun. Comput. Phys., 1 (2006), pp. 276–310.

[19] R. Li, W.-B. Liu, H.-P. Ma, and T. Tang, *Adaptive finite element approximation for distributed elliptic optimal control problems*, SIAM J. Control Optim., 41 (2002), pp. 1321–1349.

[20] R. Li, C. Luo, and P.-W. Zhang, *Numerical simulation of Doi model of polymeric fluids*, in Advances in Scientific Computing and Applications, Y. Lu, W. Sun, and T. Tang, eds., Science Press, Beijing, 2004, pp. 258–273.

[21] R. Li, T. Tang, and P.-W. Zhang, *Moving mesh methods in multiple dimensions based on harmonic maps*, J. Comput. Phys., 170 (2001), pp. 562–588.

[22] R. Li, T. Tang, and P.-W. Zhang, *A moving mesh finite element algorithm for singular problems in two and three space dimensions*, J. Comput. Phys., 177 (2002), pp. 365–393.

[23] G. Liao, F. Liu, G. D. Pena, D. Peng, and S. Osher, *Level-set-based deformation methods for adaptive grids*, J. Comput. Phys., 159 (2000), pp. 103–122.

[24] K. Lipnikov and M. Shashkov, *The error-minimization-based strategy for moving mesh methods*, Commun. Comput. Phys., 1 (2006), pp. 53–80.

[25] R. S. Palais, *Foundations of Global Nonlinear Analysis*, Benjamin, New York, 1968.

[26] J. Sacks and K. Uhlenbeck, *The existence of minimal immersions of* 2*-spheres*, Ann. of Math. (2), 113 (1981), pp. 1–24.

[27] R. Schoen and S.-T. Yau, *On univalent harmonic maps between surfaces*, Invent. Math., 44 (1978), pp. 265–278.

[28] Z.-J. Tan, Z.-R. Zhang, Y.-Q. Huang, and T. Tang, *Moving mesh methods with locally varying time steps*, J. Comput. Phys., 200 (2004), pp. 347–367.

[29] H.-Z. Tang, *A moving mesh method for the Euler flow calculations using a directional monitor function*, Commun. Comput. Phys., 1 (2006), pp. 656–676.

[30] H. Z. Tang and T. Tang, *Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws*, SIAM J. Numer. Anal., 41 (2003), pp. 487–515.

[31] M. A. Walkley, P. H. Jimack, P. K. Kelmanson, and J. L. Summers, *Finite element simulation of three-dimensional free-surface flow problems*, J. Sci. Comput., 24 (2005), pp. 147–162.

[32] D. S. Wang and X. P. Wang, *A three-dimensional adaptive method based on the iterative grid redistribution*, J. Comput. Phys., 199 (2004), pp. 423–426.
[33] A. Winslow, *Numerical solution of the quasi-linear Poisson equation*, J. Comput. Phys., 1 (1967), pp. 149–172.
[34] P. A. Zegeling, *On resistive mhd models with adaptive moving meshes*, J. Sci. Comput., 24 (2005), pp. 263–284.