

ON THE NUMERICAL SOLUTION OF ELLIPTIC AND PARABOLIC PDE IN THE REAL PROJECTIVE PLANE*

Mahdieh Sattari and Jukka Tuomela

Department of Physics and Mathematics, University of Eastern Finland,

P.O. Box 111, FI-80101 Joensuu, Finland

Email: mahdieh.sattari@uef.fi, jukka.tuomela@uef.fi

Abstract

Some models dealing with fibers and liquid crystals can be formulated probabilistically in terms of orientation distributions. Since the orientation of a thin object can be specified by a point in a real projective plane this approach leads to elliptic and parabolic problems in the real projective plane. In most previous works these kind of problems have been considered on the unit sphere which is a double cover of the real projective plane. However, numerically this is inefficient because the resulting systems of equations are unnecessarily big. We formulate the problem directly in the real projective plane using a certain parametrization with three coordinate domains. After reducing the computations to the coordinate domains we can then use finite elements almost in a standard way. In particular the standard error estimates with usual Sobolev spaces remain valid in this setting. We consider both elliptic and parabolic cases, and demonstrate the validity of our approach.

Mathematics subject classification: 65M60, 35Q84, 35Q35, 58J32

Key words: Smoluchowski equation, Real projective plane, Finite elements, Orientation distribution.

1. Introduction

In this article we are interested in computing the solutions to elliptic and parabolic problems in the real projective plane using finite element methods. These kind of problems arise in various applications: dilute suspension of wood fibres [7, 19, 21], liquid crystals [8, 17] and even the analysis of images [4]. In all cases the unknown function is a probability distribution of the orientation of the fibers or liquid crystals or other objects.¹⁾ Recall that the orientation of a rigid body can be specified by a point in $\mathbb{SO}(3)$ which is diffeomorphic to the real projective space \mathbb{RP}^3 . Now in the applications described in the references cited above one may assume that the object is in fact a very thin rod so it is reasonable to ignore its rotation around its long axis. Hence the orientation can be specified by a point in the real projective plane \mathbb{RP}^2 .

In all previous publications that we are aware of the relevant equations have been analyzed on the unit sphere S^2 instead of \mathbb{RP}^2 . This is possible since S^2 is the 2 sheeted covering space of \mathbb{RP}^2 . Hence one obtains correct results if one assumes that all relevant functions satisfy $f(p) = f(-p)$ and vector fields satisfy $w(p) = -w(-p)$. Anyway it would appear more natural to do the analysis directly on \mathbb{RP}^2 .

There has also been some interest in the numerical solution of the models discussed above, see for example [2, 5, 6, 11, 12] and references therein. In these numerical studies the problem has

* Received March 26, 2014 / Revised version received May 21, 2015 / Accepted October 8, 2015 /
Published online January 18, 2016 /

¹⁾ The relevant equation is sometimes called Smoluchowski equation or Fokker-Planck equation.

also been analyzed in S^2 . Numerically this is not anymore equivalent to doing the computations in \mathbb{RP}^2 . In the discretization there are in this case twice as many unknowns than necessary, and hence the numerical cost of solving the corresponding linear system is typically 4 to 8 times more expensive. In particular in time dependent problems where a linear system must be solved at each time step, the amount of unnecessary computation can be quite substantial.

In previous numerical studies typically S^2 has been regarded as a submanifold of \mathbb{R}^3 . Hence one part of the discretization error is that one approximates S^2 by some other structure. This approach would be somewhat involved with \mathbb{RP}^2 because it cannot in any case be embedded in \mathbb{R}^3 . However, such an embedding is unnecessary and we will below show how to introduce some convenient parametrizations of \mathbb{RP}^2 which allows us to do computations directly on \mathbb{RP}^2 .

Note also that S^2 and \mathbb{RP}^2 cannot be parametrized using a single coordinate patch. For example using the spherical coordinates, like in [2], to parametrize S^2 creates artificial singularities at the poles. While it is possible to mitigate the effects of these singularities by appropriate numerical tricks this nevertheless has an adverse influence on the stability and accuracy of the computations. In our approach we cover \mathbb{RP}^2 with 3 coordinate patches and hence there are no singularities due to parametrization.

The content of the article is as follows. In Section 2 we recall some background material from Riemannian geometry and PDE theory. In Section 3 we describe the discretization of the problem which reduces the computations to the standard finite element setting. In Sections 4 and 5 we present the numerical results in some elliptic and parabolic test cases and finally in section 6 we give some conclusions and perspectives for future work.

2. Preliminaries

We start by recalling some facts from the theory of PDEs and differential geometry. More details can be found in [9, 13, 14, 18, 23].

2.1. Differential geometry

Let g be a Riemannian metric on some smooth manifold M . The components of g in the coordinate system are denoted by g_{ij} , and the resulting matrix is G . The components of G^{-1} are denoted by g^{ij} . Let M be a Riemannian manifold with boundary ∂M . The canonical volume form of M is denoted by ω_M and the volume of M is thus $\text{vol}(M) = \int_M \omega_M$. The induced volume form on ∂M is denoted by $\omega_{\partial M}$. In case of nonorientable manifolds ω_M is interpreted as Riemannian density.

Then if w is some vector field on M we can define the gradient of u by the formula

$$g(\text{grad}(u), w) = du w = w(u).$$

The standard gradient (resp. divergence and Laplacian) operator in Euclidean spaces is denoted by ∇ (resp. $\nabla \cdot$ and Δ) as usual. In a coordinate system we can write

$$\text{grad}(u) = G^{-1} \nabla u = \sum_{i=1}^n \sum_{j=1}^n g^{ij} \frac{\partial u}{\partial x_j} \frac{\partial}{\partial x_i},$$

$$g(\text{grad}(u), \text{grad}(v)) = \langle \nabla u, G^{-1} \nabla v \rangle.$$

The divergence of a vector field $w = \sum_i w_i \partial / \partial x_i$ is given in coordinates by

$$\operatorname{div}(w) = \frac{1}{\sqrt{\det(G)}} \sum_i \frac{\partial}{\partial x_i} (\sqrt{\det(G)} w_i).$$

Finally the Laplacian of u is given by

$$\Delta_M u = \operatorname{div}(\operatorname{grad}(u))$$

and the coordinate expression is obtained by combining the formulas for divergence and gradient. Note that the coordinate formulas for divergence and Laplacian are not in fact needed in our computations. All these formulas as well as the following divergence theorem remain valid also in the case of nonorientable manifold.

Theorem 2.1. *Let M be a compact Riemannian manifold, w a vector field on M and ν the outer unit normal field of the boundary ∂M . Then*

$$\int_M \operatorname{div}(w) \omega_M = \int_{\partial M} g(w, \nu) \omega_{\partial M}.$$

If M is a manifold without boundary the right hand side of the above formula vanishes. The following formulas follow immediately. If u and v are functions and w is a vector field then

$$\int_M v \Delta_M u \omega_M + \int_M g(\operatorname{grad}(u), \operatorname{grad}(v)) \omega_M = \int_{\partial M} v g(\operatorname{grad}(u), \nu) \omega_{\partial M}, \quad (2.1)$$

$$\int_M g(\operatorname{grad}(u), w) \omega_M + \int_M u \operatorname{div}(w) \omega_M = \int_{\partial M} u g(w, \nu) \omega_{\partial M}.$$

One can define the L^p and Sobolev spaces on M using local charts and we denote the corresponding spaces by $L^p(M)$ and $H^s(M)$ as usual.

2.2. PDE

Let M be a compact Riemannian manifold without boundary. Consider the following elliptic PDE

$$-\Delta_M u + w(u) + b u = f, \quad (2.2)$$

where w is a vector field and b and f are some known functions. We suppose that b and the components of w are in $L^\infty(M)$ and $f \in H^{-1}(M)$. In these circumstances we have

Theorem 2.2. *If $w = 0$ and $b > 0$ the problem (2.2) has a unique solution in $H^1(M)$. If $w \neq 0$ the problem is Fredholm with index zero.*

We will need to consider also the associated parabolic PDE

$$\begin{cases} u_t - \Delta_M u + w(u) + b u = f, \\ u(x, 0) = u_0(x). \end{cases} \quad (2.3)$$

Here w , b and f may a priori depend on t . Let $I = [0, T]$ be some fixed interval. We now suppose that b and the components of w are in $L^\infty(M \times I)$ and that $u_0 \in L^2(M)$. Now we can consider u and f as curves in suitable spaces. Let us set

$$V = \left\{ u \in L^2(I; H^1(M)) \mid \frac{du}{dt} \in L^2(I; H^{-1}(M)) \right\}.$$

Theorem 2.3. *Suppose that $f \in L^2(I; H^{-1}(M))$. Then the problem (2.3) has a unique solution in V .*

Note that $V \subset C(I; L^2(M))$ so the initial condition makes sense. Note that here we have uniqueness even if b is negative, unlike in the elliptic case. We will also need the following version of the parabolic maximum principle.

Theorem 2.4. *Let u be the solution of (2.3).*

- (i) *If $f \geq 0$ and $u_0 \geq 0$ then $u \geq 0$, and*
- (ii) *if $b \geq 0$ and $f = 0$ then $\|u\|_{L^\infty(M)} \leq \|u_0\|_{L^\infty(M)}$.*

The following simple result is essential in the application that we have in mind.

Lemma 2.1. *Let u be the solution of (2.3) where $f = 0$ and $b = \operatorname{div}(w)$. Then $\int_M u \omega_M$ is constant.*

Proof. In this case we can write $u_t - \operatorname{div}(\operatorname{grad}(u) - u w) = 0$ and the result follows from Theorem 2.1. \square

2.3. Variational formulation

In order to solve (2.2) and (2.3) numerically we write them as variational problems. Using the formula (2.1) we can write the problem (2.2) as follows.

find $u \in H^1(M)$ such that

$$a(u, v) = \int_M f v \omega_M \quad \text{for all } v \in H^1(M), \quad (2.4)$$

where a is given by

$$a(u, v) = \int_M \left(g(\operatorname{grad}(u), \operatorname{grad}(v)) + g(w, \operatorname{grad}(u))v + b u v \right) \omega_M.$$

Let us next write (2.3) as a variational problem. Using again the formula (2.1) we can write for given $u, v \in H^1(M)$

$$a(t; u, v) = \int_M \left(g(\operatorname{grad}(u), \operatorname{grad}(v)) + g(w(t), \operatorname{grad}(u))v + b(t) u v \right) \omega_M.$$

This leads to the following problem.

find $u \in V$ such that $u(0) = u_0 \in L^2(M)$ and

$$\frac{d}{dt} \int_M u(t) v \omega_M + a(t; u(t), v) = \int_M f(t) v \omega_M \quad \text{for all } v \in H^1(M), \quad (2.5)$$

where equality is understood in the sense of distributions.

Our goal is thus to solve numerically problems (2.4) and (2.5) when $M = \mathbb{RP}^2$.

3. Discrete Variational Formulation

3.1. Composition of domains

The real projective plane \mathbb{RP}^2 is a compact smooth nonorientable manifold whose Euler characteristic is $\chi(\mathbb{RP}^2) = 1$. Topologically one could construct \mathbb{RP}^2 by identifying appropriate sides of the rectangle in \mathbb{R}^2 as in Fig. 3.1. However, this construction gives a “wrong” (flat) Riemannian metric to \mathbb{RP}^2 . The “right” metric in our context comes by identifying the opposite points of the unit sphere S^2 . Recall that we try to model orientation of a thin object. If we now consider orientation distribution then the appropriate area form should be uniform in all directions. But the standard area form on the sphere has this property and the identification of opposite points preserves this property. This construction thus induces a “round” metric on \mathbb{RP}^2 . Hence we should find some appropriate maps which parametrize \mathbb{RP}^2 and give the round metric.



Fig. 3.1. Topological construction of the projective plane.

One possibility to proceed is as follows. Let us define $D_1 = (-1, 1) \times (-1, 1)$, $D_2 = (1, 3) \times (-1, 1)$ and $D_3 = (-1, 1) \times (1, 3)$. Then we define the maps $\varphi_j : D_j \rightarrow \mathbb{R}^3$ as follows:

$$\varphi_1(z) = \gamma_1^{-1/2} \begin{pmatrix} z_1 \\ z_2 \\ 1 \end{pmatrix}, \quad \varphi_2(z) = \gamma_2^{-1/2} \begin{pmatrix} 1 \\ z_2 \\ 2 - z_1 \end{pmatrix}, \quad \varphi_3(z) = \gamma_3^{-1/2} \begin{pmatrix} z_1 \\ 1 \\ 2 - z_2 \end{pmatrix}, \quad (3.1)$$

where $\gamma_1 = 1 + |z|^2$, $\gamma_2 = 1 + (z_1 - 2)^2 + z_2^2$ and $\gamma_3 = 1 + z_1^2 + (z_2 - 2)^2$.

Then we compute the components of the Riemannian metric in each subdomain:¹⁾

$$G_j = d\varphi_j^T d\varphi_j, \quad \det(G_j) = \gamma_j^{-3}$$

Let us now join the subdomains together: $D = \bar{D}_1 \cup \bar{D}_2 \cup \bar{D}_3$. Note that by construction we have $\varphi_2(s, 1) = \varphi_3(1, s)$. This gives already one identification on the boundary of D . Other identifications are given by

$$\begin{aligned} \varphi_1(-1, s) &\sim \varphi_2(3, -s), & \varphi_1(s, -1) &\sim \varphi_3(-s, 3), \\ \varphi_2(s, -1) &\sim \varphi_3(-1, 4 - s). \end{aligned}$$

The resulting structure is illustrated in Fig. 3.2. One readily verifies that this is topologically the same as Fig. 3.1. Note that joining the maps φ_j together produces a certain map $\varphi : D \rightarrow S^2$.

¹⁾ Here and in subsequent computations we have used MAPLE 16. In particular the *Differential Geometry package* was very convenient.

This map is only continuous across the subdomain boundaries. This has however no effect on the accuracy of our computations.

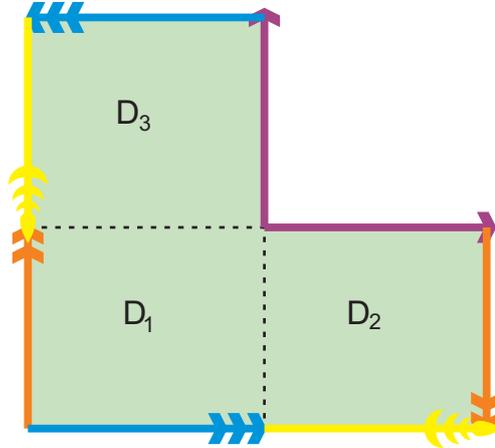


Fig. 3.2. Construction of the projective plane using the parametrizations given in (3.1).

The above approach could be called the composition of domains because the idea is precisely the opposite of what is done in decomposition of domains.

Having the maps φ_j at our disposal we can compute the maps in (2.4) and (2.5) as follows. Let us define the following bilinear maps:

$$a_j(u, v) = \int_{D_j} \left(\langle \nabla u, G_j^{-1} \nabla v \rangle + \langle \nabla u, w \rangle v + b u v \right) \gamma_j^{-3/2} dz_1 dz_2.$$

Let $u : \mathbb{RP}^2 \rightarrow \mathbb{R}$ be some map and let $u_j = u \circ \varphi_j$. Then we can write

$$\begin{aligned} a(u, v) &= \int_{\mathbb{RP}^2} \left(g(\mathbf{grad}(u), \mathbf{grad}(v)) + g(w, \mathbf{grad}(u))v + b u v \right) \omega_{\mathbb{RP}^2} \\ &= a_1(u_1, v_1) + a_2(u_2, v_2) + a_3(u_3, v_3). \end{aligned} \tag{3.2}$$

Similarly if $L_j v = \int_{D_j} f v \gamma_j^{-3/2} dz_1 dz_2$, then

$$L v = \int_{\mathbb{RP}^2} f v \omega_{\mathbb{RP}^2} = L_1 v_1 + L_2 v_2 + L_3 v_3. \tag{3.3}$$

Formulas (3.2) and (3.3) reduce all computations on \mathbb{RP}^2 to the standard computations in the subdomains D_j .

3.2. Triangulation

Our next task is to triangulate \mathbb{RP}^2 in an appropriate way. Since we have already reduced the computations to subdomains D_j it is thus sufficient to triangulate each D_j . Now in order that these separate triangulations represent a global triangulation in \mathbb{RP}^2 the triangulations in the subdomains should be compatible. This is relatively easy to achieve: we simply specify the locations of the vertices at the subdomain boundaries in such a way that the vertices (and hence the edges) match according to the identifications introduced above.

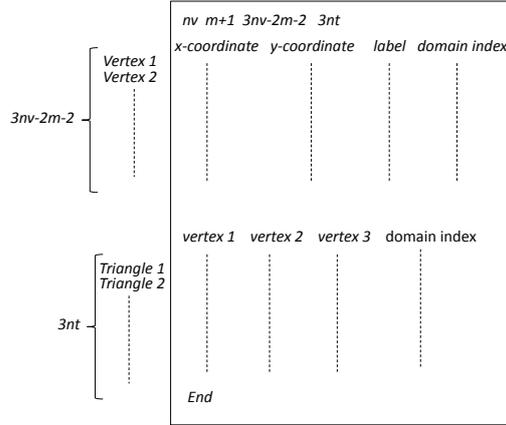


Fig. 3.3. Mesh data structure.

Of course if we use the Euclidean metric in \mathbb{R}^2 the resulting mesh would not be uniform in the round metric of \mathbb{RP}^2 . This would mean that the areas of the triangles would be distorted at most by the factor

$$\max_{z \in D_j} \frac{1}{\sqrt{\det(G_j)}} = 3^{3/2} \approx 5.2.$$

However, we can use the matrices G_j to give the information about the metric to the meshing algorithm. This gives a nice mesh in the correct metric [15]. In Fig. 4.1 below there is an example of such an adapted mesh. The triangles in the subdomains D_j are smaller near the center than near the boundary in the Euclidean metric. In the round metric this triangulation is approximately uniform.

Now there have been studies about how to triangulate directly the projective plane [1]. However, as far as we know these studies could be described as topological or combinatorial in the sense that the metric properties of the projective plane are not taken into account. Since in our application the correct Riemannian metric is essential we had to proceed in another way.

3.3. Data structure and implementation

To describe the algorithm, some notation will be introduced. First each subdomain D_j is triangulated with FREEFEM [3]. These triangulations must be compatible with the identifications indicated in Fig. 3.2. This means that not only the whole boundaries are identified, but also the vertices and edges on the boundaries must be appropriately identified. This is quite easy to arrange with FREEFEM.

Let us now for simplicity of exposition assume that the triangulations of subdomains are identical, i.e. the triangulation in a given subdomain is obtained by an appropriate translation of the triangulation in another subdomain. Further let us suppose that each side of the subdomain is divided into m equally sized edges. Let n_v (resp. n_e and n_t) be the number of vertices (resp. edges and triangles) in each subdomain D_j . Then still with FREEFEM we can put these triangulations together which gives a triangulation of D with $3n_v - 2m - 2$ vertices, $3n_e - 2m$ edges and of course $3n_t$ triangles.

Based on this data structure we must next construct an appropriate mesh data file for D which implements the identifications according to Fig. 3.2. It is straightforward to check that

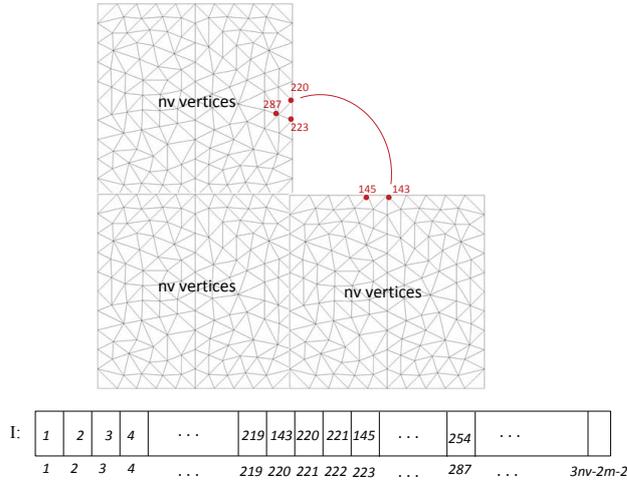


Fig. 3.4. To make the identifications according to Fig. 3.2 the vertex indices are renumbered and saved in the array **I**.

after identifications we have $N_v = 3n_v - 6m - 2$ vertices and $N_e = 3n_e - 6m$ edges. Note that this gives correctly

$$\chi(\mathbb{R}P^2) = N_v - N_e + N_t = 3(n_v - n_e + n_t) - 2 = 1.$$

The structure of the initial data file is shown in Fig. 3.3. Using this data file and we construct an array **I** of size $3n_v - 2m - 2$ containing the new vertex indices after identifications. We initialize **I** by assigning the original values $1, \dots, 3n_v - 2m - 2$ to its elements. To get the new vertex indices, the boundary vertices must be identified according to Fig. 3.2.

Let us for example consider two purple boundaries in Fig. 3.2. We need to renumber m vertices because the common vertex is not being renumbered. Two arrays of size m , called **A**₂ and **A**₃ are set to store the vertex indices on the purple boundary of D_2 and D_3 respectively. The array **A**₂ (resp. **A**₃) is sorted in ascending order according to the z_1 coordinate (resp. z_2 coordinate) of the vertices. After this the vertex index **A**₃[j] is identified with the vertex index **A**₂[j] and we can write

$$\mathbf{I}[\mathbf{A}_3[j]] = \mathbf{A}_2[j].$$

Then we must update also other vertex indices of subdomain D_3 . For instance, let us assume that the triangle in D_3 has vertex indices 220, 223 and 287 and that the vertices of the corresponding edge in D_2 have indices 143 and 145. Let us suppose that the vertex 287 is renumbered as 254 after all identifications. In this case the array **I** is schematically as in Fig. 3.4. All other identifications are done in the same way. In the end the value of **I**[j] represents the new index number of vertex j in the final stage.

In the case of using P_2 element, we also need information about midpoints of the edges. First the boundaries are identified and the new indices of all vertices are computed as above. It is only at this stage that we number the midpoints (more precisely: the degrees of freedom associated to midpoints). In other words there is no need for the initial numbering of midpoints so they receive directly the final correct index. Hence in two different triangles there can be midpoints with the same indices but different coordinates. This works because we only need to associate a certain integral (associated to the midpoint) to a certain index and we do not

need to know what are the coordinates associated to a certain index of midpoint. To number the midpoints we set up the array \mathbf{E} of size N_e where we store the indices of the midpoints, starting with value $N_v + 1$ and increase one by one while moving from edge to edge. We have chosen to number first the midpoints on the boundaries.

The next task is to assemble the system matrix. For simplicity of notation we describe only the P_1 case. Let us consider triangle T_s which belongs to subdomain D_j and let k_i^s , $i = 1, 2, 3$, be the original vertex indices of T_s . Let us further denote by $\psi_{k_i^s}$ the local basis function associated to k_i^s . Then we compute the local matrices M_s , $1 \leq s \leq 3n_t$ whose entries are given by

$$m_{i\ell}^s = \int_{T_s} \left(\langle \nabla \psi_{k_i^s}, G_j^{-1} \nabla \psi_{k_\ell^s} \rangle + \langle \nabla \psi_{k_i^s}, w \rangle \psi_{k_\ell^s} + b \psi_{k_i^s} \psi_{k_\ell^s} \right) \gamma_j^{-3/2} dz_1 dz_2.$$

We have used the Gauss-Kronrod quadrature formula which is exact for polynomials of degree five to compute these integrals [10].

To obtain the global matrix M of size $N_v \times N_v$ we use the array \mathbf{I} to set or update the entries of M appropriately. We first initialize M as a zero matrix. Then for each triangle we set

$$M_{ru} := M_{ru} + m_{i\ell}^s, \quad \text{where } r = \mathbf{I}[k_i^s] \quad \text{and} \quad u = \mathbf{I}[k_\ell^s].$$

In the P_2 case we also have to use the array \mathbf{E} to get the system matrix M . Of course practical details are a bit more complicated but the basic idea is the same as in the P_1 case.

To summarize we have thus first used FreeFem to triangulate each D_j and then D . Then using this initial mesh data file we have implemented the rest in C++. Additionally, we used the UMFPAK²⁾ package in our code to solve the relevant linear systems.

3.4. Error estimates

Once we have the final system matrix for the elliptic problem we can use the method of lines in a standard way for the time dependent problems. The standard error estimates are also valid in our context. In fact one could say that our case is easier than the traditional setting because the projective plane has no boundary. For completeness we recall some basic estimates. Let u_e (resp. u_p) be the numerical approximation to the problem (2.4) (resp. (2.5)). Let δt be the time step and k the order of the time discretization. Then under appropriate smoothness assumptions we have the following estimates in the P_1 case [13] :

$$\|u - u_e\|_{L^2(\mathbb{RP}^2)} = O(h^2), \tag{3.4a}$$

$$\|u - u_e\|_{H^1(\mathbb{RP}^2)} = O(h), \tag{3.4b}$$

$$\|u(t) - u_p(t)\|_{L^2(\mathbb{RP}^2)} = O(h^2 + \delta t^k). \tag{3.4c}$$

Similarly in the P_2 case we have

$$\|u - u_e\|_{L^2(\mathbb{RP}^2)} = O(h^3), \tag{3.5a}$$

$$\|u - u_e\|_{H^1(\mathbb{RP}^2)} = O(h^2), \tag{3.5b}$$

$$\|u(t) - u_p(t)\|_{L^2(\mathbb{RP}^2)} = O(h^3 + \delta t^k). \tag{3.5c}$$

²⁾ <http://www.cise.ufl.edu/research/sparse/umfpack/>

4. Validation of the Code

In the following we test our code in some simple situations where the solution is known. We use standard P_1 and P_2 elements for space discretization and BDF2 method for time discretization. In each example we start the triangulation by choosing some convenient number of edges m for each side of D_j . We then study the convergence in the standard way by splitting each edge into k edges in a such way that each triangle is split into k^2 triangles. Accordingly in the tables below km means that the initial mesh had m edges on the side and km refers to the mesh obtained by splitting each edge to k pieces. In this way the discretization parameter h in (3.4) and (3.5) for a given mesh can be taken to be $1/(km)$. Note finally that it is natural to compare the results with P_1 elements using mesh size $2m$ to results with P_2 elements using the mesh size m . To visualize our results we have used MEDIT package¹⁾.

4.1. Elliptic case

It is perhaps more convenient to describe the test problems extrinsically using the coordinates of \mathbb{R}^3 .

Recall that the spherical harmonics provide eigenfunctions of the Laplacian also in the projective spaces. One simply takes those spherical harmonics which are invariant by the map $p \mapsto -p$. Hence starting from Legendre polynomial

$$P_{6,2} = (1 - x^2)(33x^4 - 18x^2 + 1),$$

we get the associated projective harmonic which is given in ambient coordinates by

$$f_{6,2} = x_1 x_2 (x_1^4 + 2x_1^2 x_2^2 + x_2^4 + 16x_3^4 - 16x_1^2 x_3^2 - 16x_2^2 x_3^2). \quad (4.1)$$

The associated eigenvalue is 42 so $f_{6,2}$ is the unique solution to the problem

$$-\Delta_{\mathbb{R}\mathbb{P}^2} u + u = 43f_{6,2}.$$

Let us then consider the following vector field:

$$w(x) = Kx = \hat{k} \times x, \quad \text{where} \quad K = \begin{pmatrix} 0 & -k_3 & k_2 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{pmatrix} \quad \text{and} \quad \hat{k} = \begin{pmatrix} k_1 \\ k_2 \\ k_3 \end{pmatrix}. \quad (4.2)$$

Evidently w is tangent to S^2 . Moreover since $w(x) = -w(-x)$ this gives also a well defined vector field on $\mathbb{R}\mathbb{P}^2$. Let us then define the following function

$$f = x_1^6 + x_1^4 x_2^2 - 16x_1^4 x_3^2 - x_1^2 x_2^4 + 16x_1^2 x_3^4 - x_2^6 + 16x_2^4 x_3^2 - 16x_2^2 x_3^4.$$

Now choosing $\hat{k} = (0, 0, 10)$ we compute that $f_{6,2}$ is a unique solution to the problem

$$-\Delta_{\mathbb{R}\mathbb{P}^2} u + w(u) + u = 43f_{6,2} + 10f. \quad (4.3)$$

The solution is plotted in Fig. 4.2. For visualizing the solution it is convenient to plot it on S^2 . In Table 4.1 we have shown the relative errors using $m = 10$. The L^2 and H^1 errors behave as estimates (3.4) and (3.5) predict.

¹⁾ <http://www.ljll.math.upmc.fr/~frey/software.html>

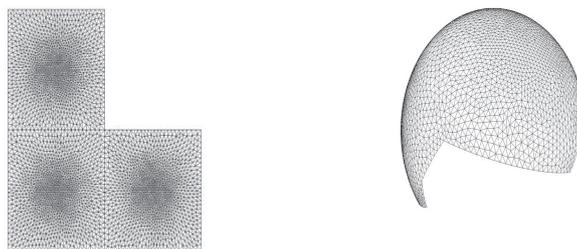


Fig. 4.1. An example of mesh. Note that the triangles are not uniform in the flat metric but they are approximately uniform in the round metric.

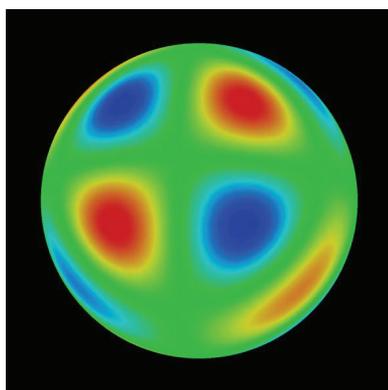


Fig. 4.2. Solution to (4.3) given by a projective harmonic.

Table 4.1: The relative error L^2 and H^1 norms for problem (4.3).

Element		m	$2m$	$4m$	$8m$	$16m$
P_1	H^1		7.13e-2	2.03e-2	5.50e-3	1.45e-3
	L^2		4.64e-2	1.19e-2	3.03e-3	6.27e-4
P_2	H^1	5.13e-2	1.09e-2	1.90e-3	3.53e-4	
	L^2	2.02e-2	2.06e-3	1.62e-4	1.41e-5	

4.2. Parabolic case

Let w be again a rotational vector field as in (4.2) and let $f_{6,2}$ be as before. Consider the problems

$$\begin{cases} u_t - \frac{1}{420} \Delta_{\mathbb{R}\mathbb{P}^2} u + w(u) = 0, \\ u(x, 0) = 1 + f_{6,2}(x), \end{cases} \quad (4.4)$$

$$\begin{cases} v_t - \frac{1}{420} \Delta_{\mathbb{R}\mathbb{P}^2} v = 0, \\ v(x, 0) = 1 + f_{6,2}(x), \end{cases} \quad (4.5)$$

Obviously the solution to the problem (4.5) is given by

$$v(x, t) = 1 + f_{6,2}(x)e^{-t/10}.$$

Then we have

Lemma 4.1. *The solution to the problem (4.4) is $u(x, t) = v(e^{-Kt}x, t)$. In particular $u(t) \rightarrow 1$ when $t \rightarrow \infty$.*

Proof. Note that e^{-Kt} is orthogonal, hence it induces an isometric map to $\mathbb{R}\mathbb{P}^2$. But if $\phi : \mathbb{R}\mathbb{P}^2 \rightarrow \mathbb{R}\mathbb{P}^2$ is any isometry and $q = \phi(p)$, then

$$\Delta_{\mathbb{R}\mathbb{P}^2}u(p) = \Delta_{\mathbb{R}\mathbb{P}^2}v(q).$$

On the other hand $v_t = u_t + w(u)$. This gives the result. □

Now

$$e^{-Kt} = \frac{1}{|\hat{k}|^2} \begin{pmatrix} c_{123} & d_{312}^+ & d_{231}^- \\ d_{312}^- & c_{231} & d_{123}^+ \\ d_{231}^+ & d_{123}^- & c_{312} \end{pmatrix}, \quad \text{where}$$

$$c_{ij\ell} = k_i^2 + (k_j^2 + k_\ell^2) \cos(|\hat{k}|t) \quad \text{and} \quad d_{ij\ell}^\pm = k_j k_\ell (1 - \cos(|\hat{k}|t)) \pm k_i |\hat{k}| \sin(|\hat{k}|t).$$

We chose $\hat{k} = (1, 0, 0)$ and computed the solution with various time steps. We used the exact solution for u to get the value of u at $t = \delta t$ to start BDF2. We obtained the expected result using an appropriate mesh size and time step. Some of the results are presented in Tables 4.2 and 4.2 for both P_1 and P_2 elements using $m = 5$. Table 4.2 shows that L_2 error is almost proportional to δt^2 as expected for BDF2 method and Table 4.2 shows that L_2 error is proportional to h^2 (resp. h^3) using P_1 (resp. P_2) elements.

5. Fiber Orientation

Finally let us consider the problem which was the original motivation for our study. We will only briefly indicate how we arrive at the relevant physical model and refer to [7, 19, 21, 22] for more details. In particular our article [22] is intended as a complement to the present article. Here we give sufficient mathematical details so that the numerical results and the implementation become clear. Hence below we consider a somewhat idealistic situation where the relevant flow field is known. On the other hand the cases in [22] are closer to realistic

Table 4.2: The relative error in L^2 norm at $t = 0.2$ for different time steps using P_1 and P_2 elements for problem (4.4).

		$\delta t = 0.04$	$\delta t = 0.02$	$\delta t = 0.01$
P_1	$6m$	6.81e-3	1.52e-3	4.07e-4
P_2	$3m$	6.76e-3	1.50e-3	4.04e-4

Table 4.3: The relative error in L^2 norm at $t = 0.2$ for different mesh sizes using P_1 and P_2 elements and $\delta t = 0.01$ for problem (4.4).

	m	$2m$	$4m$	$8m$
P_1		1.93e-3	4.86e-4	1.16e-4
P_2	1.88e-3	2.28e-4	3.97e-5	

models in engineering applications, and there were refer to the present article for mathematical details.

Let Ω be some 3 dimensional domain. We consider the situation where there is a fluid flow in Ω and in this ambient fluid there are some fibers whose orientations interest us. In the stationary case this fiber orientation distribution function can be then thought as a function $\tilde{u} : \Omega \times \mathbb{RP}^2 \rightarrow \mathbb{R}$ which is positive and whose integral over \mathbb{RP}^2 is one. Note that this is a 5 dimensional problem and as far as we know nobody has tried to solve the problem in this form.

One possibility to reduce the dimensionality of the problem is to consider streamlines. So let v be the solution of stationary Navier-Stokes equation in Ω , let α be a streamline of v and let $u = \tilde{u} \circ \alpha$. Hence given some streamline u is a map $\mathbb{RP}^2 \times \mathbb{R} \rightarrow \mathbb{R}$ for which we get a parabolic equation of the following form.

$$u_t - c\Delta_{\mathbb{RP}^2}u + \operatorname{div}(wu) = 0.$$

Note that t is not time variable, but the parameter along the streamline. The parameter c is some known positive constant and the vector field w represents the effect of the flow v on u . Since u is interpreted as a probability distribution it should satisfy

- (i) $u(t) \geq 0$ for all t and
- (ii) $\int_{\mathbb{RP}^2} u(t) \omega_{\mathbb{RP}^2} = 1$ for all t .

These properties follow immediately from Theorem 2.4 and Lemma 2.1, if the initial condition satisfies the above requirements.

To specify an appropriate vector field w we first introduce the following quantities

$$\varepsilon = \frac{1}{2}(\nabla v + (\nabla v)^T), \quad \omega = \frac{1}{2}(\nabla v - (\nabla v)^T).$$

Here ε is the strain rate tensor and ω is vorticity tensor. The experimental studies have suggested the following form for w

$$w = \omega x + \lambda \varepsilon x - \lambda \langle x, \varepsilon x \rangle x. \quad (5.1)$$

Here $x \in S^2$ and $0 < \lambda < 1$ is some parameter which in practice is very close to one. It is easy to check that w is tangent to the unit sphere S^2 . Since in addition $w(x) = -w(-x)$ this gives a well defined vector field also on \mathbb{RP}^2 .

Let us briefly analyze the nature of the vector field w . If we define $A = \omega + \lambda \varepsilon$ we can write w as

$$w = Ax - \langle x, Ax \rangle x.$$

From the mathematical point of view one could say that as a model w is as simple as possible. Let $x \in S^2$; then the simplest vector field is of course linear, hence the term Ax . But $Ax \notin T_x S^2$ so this does not give a vector field on S^2 . Then the most natural thing would be to project Ax orthogonally to $T_x S^2$. But this is precisely the geometric meaning of the term $\langle x, Ax \rangle x$. It is also evident that the equilibrium points of w are the eigenvectors (or more precisely eigenspaces) of A , and consequently w has either 1 or 3 equilibrium points in the generic case.

Now let us consider a simple situation which however is quite relevant for the intended application. Consider the following 2 dimensional domain

$$\Omega = \{y \in \mathbb{R}^2 \mid |y_2| < \beta |y_1|, \beta_0 < y_1 < \beta_1 < 0\}. \quad (5.2)$$

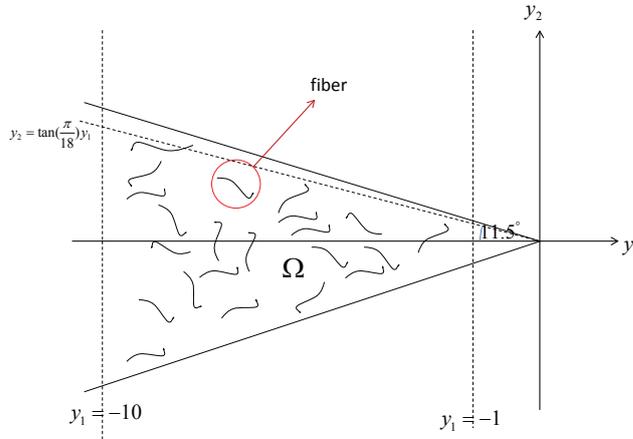


Fig. 5.1. Domain Ω in (5.2) when $\beta_0 = -10$, $\beta_1 = -1$ and $\beta = \tan(0.2)$.

In this case it is well known that there is a solution to stationary Navier-Stokes equation which can be expressed in the polar coordinates as [16]

$$v = -\frac{f(\theta - \pi)}{r} \frac{\partial}{\partial r}.$$

If μ is the viscosity of the fluid then f is a solution of the differential equation

$$\mu f''' - 2ff' + 4\mu f' = 0,$$

with boundary conditions $f(0) = f_0$, $f'(0) = 0$ and $f(\theta_0) = 0$. With our domain Ω we have $\beta = \tan(\theta_0)$ and then the solution is extended to negative values as an even function. There is no closed form solution but of course this is easy to solve numerically. We have taken $\theta_0 = 0.2 \text{ rad} \approx 11.5^\circ$ which is reasonable in present context. For other parameters we have chosen $c = 0.1$, $\beta_0 = -10$, $\beta_1 = -1$, $\mu = 1$, $\lambda = 1$ and $f_0 = 10$ rather arbitrarily. So in this setting the suspension is flowing from left to right in two dimensional channel as in Fig. 5.1.

In the present case the streamlines are thus straight lines converging to the origin. Now simple computations show that

$$\varepsilon = \frac{1}{|y|^4} \begin{pmatrix} (y_1^2 - y_2^2)f(\theta) + y_1y_2f'(\theta) & 4y_1y_2f(\theta) - (y_1^2 - y_2^2)f'(\theta) \\ 4y_1y_2f(\theta) - (y_1^2 - y_2^2)f'(\theta) & -(y_1^2 - y_2^2)f(\theta) - y_1y_2f'(\theta) \end{pmatrix},$$

$$\omega = \frac{1}{|y|^2} \begin{pmatrix} 0 & -f'(\theta) \\ f'(\theta) & 0 \end{pmatrix}.$$

Note that although our domain Ω is 2 dimensional we must in the formula (5.1) think of ε and ω as 3×3 matrices with the vector field v zero in y_3 direction. Recall that by a 2 dimensional flow one really means a 3 dimensional flow which is invariant in one direction.

In this 2 dimensional situation we compute that if $\delta = \lambda^2(\varepsilon_{11}^2 + \varepsilon_{12}^2) - \omega_{12}^2 > 0$ then the vector field w has 3 equilibrium points and only one if $\delta < 0$. Obviously in this 2 dimensional case there is always the equilibrium point $(0, 0, 1)$.

Note that since $f'(0) = 0$ we have $\omega = 0$ when $y_2 = 0$. On the other hand f' is quite big near the boundary. For test computations we have chosen 2 values of θ : $\theta_1 = 0$ and $\theta_2 = 10^\circ$.

Table 5.1: The relative error of integral at $t = 1$ for different step sizes and for P_1 elements for fiber model.

	m	$2m$
$\delta t = 0.01$	4.6e-6	1.4e-8
$\delta t = 0.02$	2.23e-6	6.54e-7
$\delta t = 0.04$	4.55e-6	1.51e-6

Hence $f(\theta_1) = 10$ and $f'(\theta_1) = 0$ and

$$f(\theta_2) \approx 2.44, \quad f'(\theta_2) \approx -88.8.$$

Consequently the streamline corresponding to θ_1 (denoted by α_1) is in the middle of the domain and the streamline corresponding to θ_2 (denoted by α_2) is relatively near the boundary. Consequently one expects that the solutions behave somewhat differently in both cases. To study this let us consider the maps $g_j : \mathbb{RP}^2 \rightarrow \mathbb{R}$ given in extrinsic coordinates by

$$g_j(x) = 17x_j^{16}, \quad j = 1, 2, 3.$$

One easily checks that $\int_{\mathbb{RP}^2} g_j \omega_{\mathbb{RP}^2} = 1$. Hence g_j can be interpreted as probability distributions, and consequently as initial conditions for our problem. When $j = 1$ the fibres are thus initially mostly oriented along the flow, while in cases $j = 2$ and $j = 3$ they are mostly orthogonal to the flow, but in a different way. We use these as initial conditions at the inflow boundary at $y_1 = -10$ of our domain. We then compute the solution along the streamline until we reach the outflow boundary at $y_1 = -1$. The parametrization was chosen such that we reach the outflow boundary when $t = 1$.

In the following computations we used $m = 5$ for the mesh and $\delta t = 0.1$ for the time step. We computed with both P_1 and P_2 elements. However, just by looking at pictures there was no difference. Also choosing some other m or δt did not qualitatively change the solutions. In Table 5 we give some representative values for the error in the integral in the P_1 case. The error in the integral was almost the same for P_2 elements.

So we have 6 different cases: the streamlines α_1 and α_2 with initial conditions g_j , $j = 1, 2, 3$. The results are illustrated in Figs. 5.2-5.4. To interpret the orientation of the pictures recall that we have denoted by y_i the coordinates of the physical domain and by x_i the coordinates of the extrinsic coordinates for S^2 and \mathbb{RP}^2 . So we think that at each point of the streamline, and hence at each point of physical domain, we attach x coordinate system. Now we can then identify the y_j and x_j directions at each point. In all following Figures the $+y_1$ axis points out of the page, the $+y_2$ points to the right and the $+y_3$ points upward.

In the case of initial condition $u(x, 0) = g_1(x)$ the fibres are concentrated along the flow and it seems natural to suppose that they will remain so. There is not much difference in the solutions along different streamlines. Moreover the fibres become more and more aligned when one moves along the streamlines.

When the initial condition is given by g_2 one can expect qualitatively different solutions in the 2 cases. For α_1 there are 2 symmetric ways for the fibres to align with flow and hence in Fig. 5.2 we see that initial distribution splits into 2 parts which then eventually merge together along the flow. For α_2 the flow does not act symmetrically on the fibres and so in Fig. 5.3 the distribution does not split.

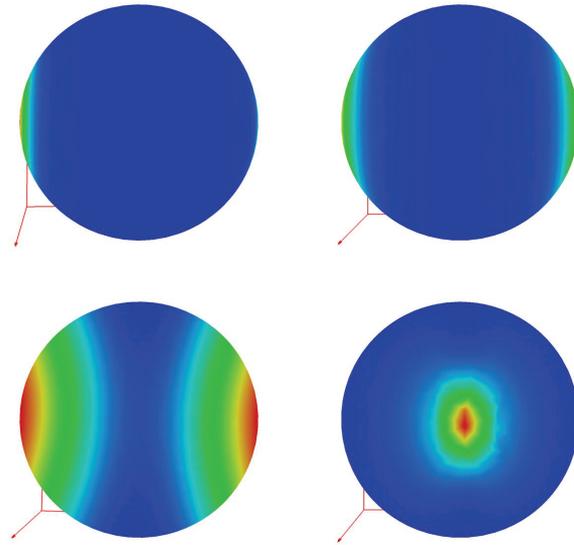


Fig. 5.2. The solution at $t = 0.1, 0.4, 0.8, 1$ on the streamline α_1 with initial condition g_2 .

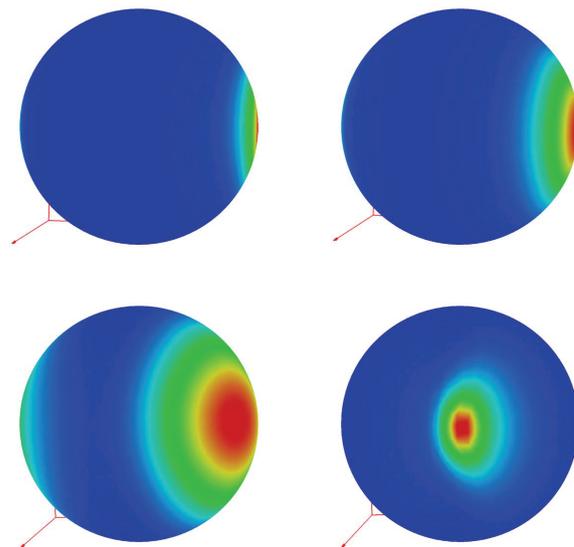


Fig. 5.3. The solution at $t = 0.1, 0.4, 0.8, 1$ on the streamline α_2 with initial condition g_2 .

Finally when the initial condition is given by g_3 the situation is now symmetric for both streamlines. Hence in Fig. 5.4 we see that the distribution first splits and then merges along the flow.

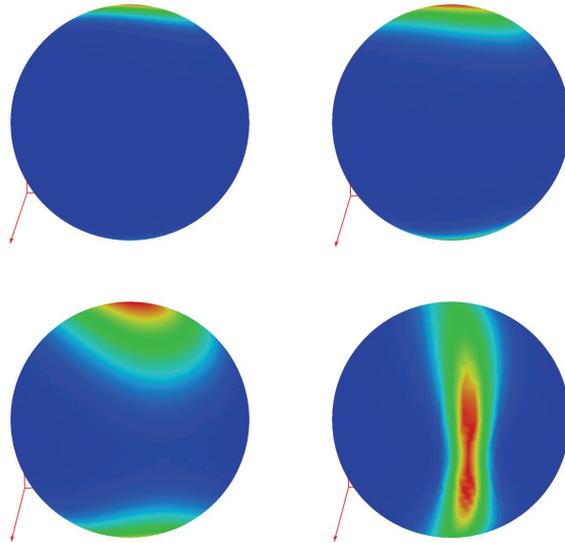


Fig. 5.4. The solution at $t = 0.1, 0.4, 0.8, 1$ on the streamline α_2 with initial condition g_3 .

6. Conclusion

We have shown above how to solve elliptic and parabolic problems in \mathbb{RP}^2 by reducing the computations to appropriate coordinate domains. The numerical results obtained validate the approach taken here. We implemented our method for P_1 and P_2 elements, but in a similar fashion any other element could be used. In some contexts one is really interested in solving PDEs on the sphere. Our approach works also in that case. Then it would be natural to use 6 subdomains to cover the sphere and to proceed otherwise as above with the projective plane.

The motivation for this article came from the problems related to dilute suspensions of wood fibres. Since problems involving orientations arise also in other applications as mentioned in the introduction the results obtained here are of interest also in other contexts. In particular our code can also solve more general parabolic type problems and it could also be easily adapted to solve hyperbolic, wave like problems.

Since our primary interest was to solve PDEs we did not produce an explicit data structure for the triangulation of the projective plane. In some applications such a data structure could be desirable. However, our data structures would still be very useful as a preliminary step in producing these other structures.

Finally this approach can be used to study PDEs on more complicated surfaces because all compact surfaces can be obtained from simple plane figures using appropriate identifications [20].

Acknowledgments. We would like to thank Frédéric Hecht and Pascal Frey for their help with FREEFEM.

References

- [1] M. Aanjaneya and M. Teillaud, Triangulating the real projective plane, INRIA, Sophia Antipolis, *Research Report*, **6296** (2007).

- [2] T. Apel and C. Pester, Clement-type interpolation on spherical domains—interpolation error estimates and application to a posteriori error estimation, *IMA J. Numer. Anal.* **25**:2 (2005), 310–336.
- [3] S. Auliac, A. Le Hyaric, J. Morice, F. Hecht, K. Ohtsuka and O. Pironneau, FREEFEM++. 3rd Edition, Version 3.31-2, 2014. <http://www.freefem.org/ff++/ftp/freefem++doc.pdf>
- [4] M. Bertalmio, L. Cheng, S. Osher and G. Sapiro, Variational problems and partial differential equations on implicit surfaces, *J. Comput. Phys.*, **144**:2 (2001), 759–780.
- [5] D. Calhoun, C. Helzel and R. LeVeque, Logically rectangular grids and finite volume methods for PDEs in circular and spherical domains, *SIAM Rev.*, **50**:4 (2008), 723–752.
- [6] D. Calhoun and C. Helzel, A finite volume method for solving parabolic equations on logically Cartesian curved surface meshes, *SIAM J. Sci. Comput.*, **31**:6 (2009/10), 4066–4099.
- [7] F. Chinesta, G. Chaidron and A. Poitou, On the solution of Fokker-Planck equations in steady recirculating flows involving short fiber suspensions, *J. Non-Newtonian Fluid Mech.*, **113**(2003), 97–125.
- [8] P. Constantin, I.G. Kevrekidis and E.S. Titi, Asymptotic states of a Smoluchowski equation, *Arch. Ration. Mech. Anal.*, **3**(2004), 365–384.
- [9] R. Dautray and J.-L. Lions. Mathematical analysis and numerical methods for science and technology. Springer-Verlag, Berlin, **5**(1992).
- [10] P. Davis and P. Rabinowitz, *Methods of numerical integration*, 2nd ed., Computer Science and Applied Mathematics, Academic Press, Orlando, FL, 1984.
- [11] A. Demlow and G. Dziuk, An adaptive finite element method for the Laplace-Beltrami operator on implicitly defined surfaces, *SIAM J. Numer. Anal.*, **45**:1 (2007), 421–442 (electronic).
- [12] G. Dziuk and C.M. Elliott, Surface finite elements for parabolic equations, *J. Comput. Math.*, **25**:4 (2007), 385–407.
- [13] A. Ern and J.-L. Guermond, Theory and practice of finite elements, *Applied Mathematical Sciences*, Vol. 159, Springer-Verlag, New York, 2004.
- [14] L. Evans, Partial differential equations, *Graduate Studies in Mathematics*, Vol. 19, American Mathematical Society, Providence, RI, 2010.
- [15] P. Frey and P.-L. George, Mesh generation, *Application to finite elements*, ISTE, London, 2008.
- [16] P. Germain, Mécanique, tome 2, *Ellipses*, Edition Marketing, Paris, 1986.
- [17] C. Helzel and F. Otto, Multiscale simulations for suspensions of rod-like molecules, *J. Comput. Phys.*, **216**:1 (2006), 52–75.
- [18] J.M. Lee, Introduction to smooth manifolds, *Graduate Texts in Mathematics*, Springer-Verlag, New York, **218**(2003).
- [19] J.Z. Lin, Z.Y. Gao, K. Zhou and T.L. Chan, Mathematical modeling of turbulent fiber suspension and successive iteration solution in the channel flow, *Applied Mathematical Modelling*, **30**(2006), 1010–1020.
- [20] W. Massey, A basic course in algebraic topology, *Graduate Texts in Mathematics*, Springer-Verlag, New York, **127**(1991).
- [21] H. Niskanen, H. Eloranta, J. Tuomela and J. Hämäläinen, On the orientation probability distribution of flexible fibres in a contracting channel flow, *Int. J. Multiphase Flow*, **27**(2011), 336–345.
- [22] M. Sattari, J. Tuomela, H. Niskanen, and J. Hämäläinen, Coupled simulation of the spherical angles of rigid fibres by using a fibre orientation probability distribution model, *Int. J. Multiphase Flow* **65**(2014), 61–67.
- [23] M. Taylor, Partial differential equations I, *Appl. Math. Sci.*, Springer-Verlag, New York, **115** (1996).