Domain Mapping using Nonlinear Finite Element Formulation

Tangudu Srinivas Patro, Hari K. Voruganti, Bhaskar Dasgupta[*] and Sumit Basu
Department of Mathematical Engineering, Indian Institute of Technology Kanpur— 208016, India.

Abstract

Domain mapping is a bijective transformation of one domain to another, usually from a complicated general domain to a chosen convex domain. This is directly useful in many application problems like shape modeling, morphing, texture mapping, shape matching, remeshing, path planning etc. A new approach considering the domain as made up of structural elements, like membranes or trusses, is developed and implemented using the nonlinear finite element formulation. The mapping is performed in two stages, boundary mapping and inside mapping. The boundary of the 3-D domain is mapped to the surface of a convex domain (in this case, a sphere) in the first stage and then the displacement/distortion of this boundary is used as boundary conditions for mapping the interior of the domain in the second stage. This is a general method and it develops a bijective mapping in all cases with judicious choice of material properties and finite element analysis. The consistent global parameterization produced by this method for an arbitrary genus zero closed surface is useful in shape modeling. Results are convincing to accept this finite element structural approach for domain mapping as a good method for many purposes.

Key Words: Domain mapping, parameterization, atlas, finite element method.

1. Introduction

Domain mapping is the task of mapping one region to another. Complicated general domains are mapped to chosen well-shaped domains where one can perform the required task easily. For example, planning an optimal path between two points in a general (non-convex) domain is a complicated problem, since segment(s) of a straight line path joining a pair of points may fall outside the domain. But the same task is trivial in a convex domain. So, naturally, one would ask for a mapping of the original domain to a convex one, so as to find the path and invert it back to the original domain. This mapping is called *domain mapping*. Besides path planning, domain mapping has applications in many fields of science and

[*] Corresponding author:
Email: *dasgupta@iitk.ac.in*, Tel: +91 512 259 7995, Fax: +91 512 259 7408.

engineering. Morphing, where the task is to transform one domain to another (e.g. dog face to cat face), is regularly needed in animation. Surface approximation, scattered data fitting, reparameterization, remeshing are regular applications related to computer graphics and geometric modeling. Molecular modeling, protein docking etc are some of its potential applications related to computational biology.

Domain mapping is also known as shape blending, shape transformation and metamorphosis. Suryawanshi et al. [1] considered the given domain *D* as composed of triangular elastic rubber sheets sewn together along their edges, and the boundary of the domain *D* is stretched over the boundary of a polygon/polyhedron *P* according to a mapping *h*. If '*h*' is a mapping from one manifold to another and if '*E(h)*' is the energy associated with it, then any deformation of '*h*' (i.e., topological irregularity) due to folds and wrinkles will increase the energy '*E(h)*'. The domain mapping is established through minimization of the total energy '*E(h)*'. Such a mapping is bound to be bijective, because such an assembly of material objects would indeed seek a least-energy configuration and overlapping of material particles is physically ruled out. Even though not essential, this physical analogy helps in appreciating the following essential characteristics associated with mapping.

- *Existence*: Mapping will always exist as long as two regions are topologically equivalent.
- *Non-uniqueness*: Such a mapping is not unique as final equilibrium configuration achieved depends on physical properties also.
- *Bijectivity*: The particles will never overlap through elastic deformation though they can come closer.

Yuan et al. [2] showed a new direction for the correspondence problem in shape transformation. They developed new algorithms for geometric transformations between two genus zero *3-D* polyhedral models by generating two new models, which possess the common topology and allow transformations from one to another to be easily computed. During morphing from a *3-D* shape to the target shape, the original shape has to be parametrized onto a sphere. This *spherical parametrization* for *3-D* case is attempted by many people. Kent et al. [3] proposed a spherical parametrization scheme for general cases. They simulate a balloon inflation process, but are not able to guarantee a bijective map. More recently, Sheffer et al. [4] formulated this spherical parameterization problem as an optimization problem by deriving the necessary and sufficient conditions for mapping using spherical geometry. Praun et al. [5] also proposed a spherical parameterization and remeshing approach without cutting the domain. Alexa [6] used a spring-like relaxation process. The relaxation solutions may collapse to a point, or experience fold-overs, depending on the starting state. He demonstrated several heuristics that help the solution converge to valid maps. Grimm [7] partitioned the surface into six charts, and mapped these to faces of a cube, and from there to a sphere. Schemes based on *a priori chart partitions* constrain the spherical parametrization. Gotsman et al. [8] showed a nice relationship between spectral graph theory and spherical parametrization, and embedded simple meshes on the sphere by solving a quadratic system. Quicken et al. [9] parametrized the surface of a voxel volume onto a sphere. Their nonlinear objective function exploits the uniform quadrilateral structure of the voxel surface and seeks to equalize areas and preserve right-angles of surface elements. Their scheme is not applicable to general triangular meshes. Khodakovsky et al. [10] designed a global parameterization approach by optimizing the patch layout subject to metric distortion criteria etc. But choosing the base domain is cumbersome and, moreover, this is not efficient for large unstructured meshes. Floater [11] developed a method based on graph theory for creating global parameterizations for surface fitting. The parameterizations are the solutions of linear systems based on convex combinations. Voruganti et al. [12] illustrated a general method of mapping planar non-convex region to a convex region by using artificial potential field. The potential filed is developed using harmonic functions which are free of *local minima*. The potential value along with the angle made by streamline at the center are two unique parameters for domain mapping. In the methodology proposed by Suryawanshi et al. [1] for *3-D* domain mapping, the *3-D* non-convex region is mapped onto *3-D* convex region in two steps: boundary mapping and inside mapping. First, the outer boundary is mapped layer-wise onto a sphere, then the inside nodes are interpolated to the interior of the sphere using finite element formulation and the obstacles/voids are mapped to points in the target domain. For boundary mapping, the outer boundary is divided into a number of layers. But some non-convex regions cannot be divided easily into layers. For such type of regions, roadblocks are encountered during boundary mapping. The present work is an improvement of the above method, using a different method for mapping the outer boundary onto a *2-D* atlas by dissecting it into four parts. The major part of the boundary is mapped by minimization of the virtual work (using nonlinear finite element formulation). The algorithmic details of the present work are presented in the next section followed by results in the third section. Finally, the paper is concluded with a brief summary and some remarks on future avenues of research in this direction.

2. Methodology

Domain Mapping is carried out in two stages, boundary mapping and inside mapping. The set of triangles on the outer surface of a domain constitutes its boundary which will be mapped first. Here, we follow the analogy of making a latitude-longitude (φ –θ) atlas from the globe. The non-convex outer boundary undergoes distortion to get mapped onto a sphere. Using this distortion as boundary condition, the inside mapping is achieved. The domain is discretized to start with. This can be accomplished using the standard solid modeling softwares or a triangulation algorithm [13] if the domain is defined by a point cloud. We used 4-noded tetrahedron elements for discretization. This produces triangular faces on the boundary surface. This triangulation is, in a way, a piecewise linear approximation of a smooth surface with triangles.

2.1. Boundary mapping

For mapping the boundary of the domain to the surface of a sphere, which is our target convex domain, the boundary triangles are to be determined using the ‘sharing test’[1]. This mapping is the same as parameterizing the outer surface onto *2-D* atlas. The aim is to prepare a (φ –θ) atlas of the outer boundary. This can be done by stretching or flattening the outer boundary to a *2-D* (parametric) plane, but it can not be performed directly for closed surfaces. So we dissect the boundary to make it an open surface and then map it to *2-D* plane, appropriately. This procedure is elaborated below.

2.1.1. Dissection of boundary

Inheriting the procedure of making the ‘atlas’ from the ‘globe’, the same naming convention has been followed. Two extreme points of the domain are named as north pole (N_p) and south pole (S_p) [2]. The set of triangles neighboring the south pole and the north pole are termed as south cap and north cap, respectively. The path peel is the set of triangles neighboring a ‘path’ lying on the outer boundary and connecting south pole to north pole. As the outer boundary surface is closed, it is cut into four parts viz., south cap, north cap, path peel and cut-out shell. This gives the major part of the closed boundary surface, normally the cut-out shell, as an open surface, as shown in Fig. 1.

Identification of south cap and north cap

South pole and north pole are two extreme points on the domain which can be found out by sorting one of the coordinates of all vertices. For forming the south cap and north cap, we first search for the triangles to which the north pole and south pole belong. Next, the immediate neighbors to the triangles containing the south pole and north pole are identified. This is performed using the ‘sharing’ test again, but this time it is sharing of an edge. These triangles form the first layer of cap. Again, searching for the neighbors to the first layer triangles gives the second layer. This is continued up to three layers[3]. The three layers at south pole constitute south cap. A similar process is carried out for north cap also. These caps are then separated from the domain.

[1] If a triangular face is shared by only one tetrahedron then it is a part of the boundary.

[2] Unlike usual atlas, latitude angle varies from 0 to π here with zero at S_p and π at N_p.

[3] With too less number of layers, the cut-out shell would have ill-conditioning on its boundary. On the other hand, with too many layers, the shape of the south/north cap itself would become complex.

Formation of the path peel

Here, the path is a curve connecting south pole and north pole and lying on the boundary of the domain. A shortest path is determined using the Dijkstra algorithm [14] so as to have minimum distortion in the final mapping. This path will be mapped to the arc of zero degree longitude on the target sphere. This path is defined by a set of nodes on the boundary of the domain. The set of triangles sharing these nodes will form the 'path peel' as shown in Fig. 1.

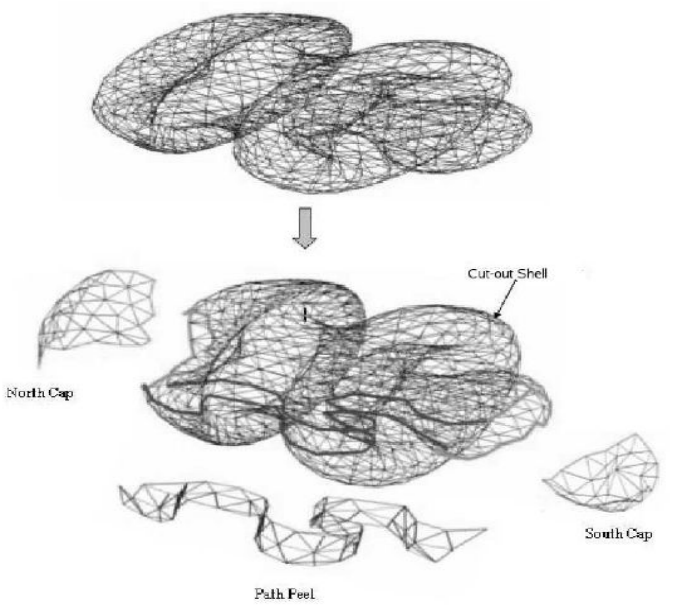


Fig. 1. Exploded view of dissected boundary of domain.

This path peel is also separated from the domain. After removing south cap, north cap and path peel, the left out part of the domain boundary is termed as the cut-out shell.

> The essence of this exercise is to obtain the major part of the surface, the cut-out shell, as an open surface, which lends itself easily to a parameterization over a rectangular domain of parameters.

2.1.2. Mapping to the atlas

Now the outer boundary of the domain is in the form of four pieces. We map all of them to the (φ –θ) atlas. In other words, we need to determine unique values of θ and φ for each node. Before mapping, the radius of the target sphere is determined using the following relation

surface area of sphere = λ (surface area of domain), i.e.

$$4\pi r^2 = \lambda A. \tag{1}$$

where r is the radius of sphere, A is the surface area of the domain and λ is a constant. South pole and north pole are assigned 0 and π as latitude values[4], respectively. For the rest of the nodes on both the caps, we use the method proposed by Suryawanshi et al.[1]. This involves mapping cap layers (LVS1, LVS2, LVN1, LVN2 in Fig. 2) one by one to the sphere. Each layer of a cap contains a node belonging to the path which is considered as the starting node for the determination of θ. The path itself is mapped to zero degree longitude (θ) arc on the sphere. The latitude (φ) for each node on the path is determined in proportion to the segment length as shown schematically in Fig. 2.

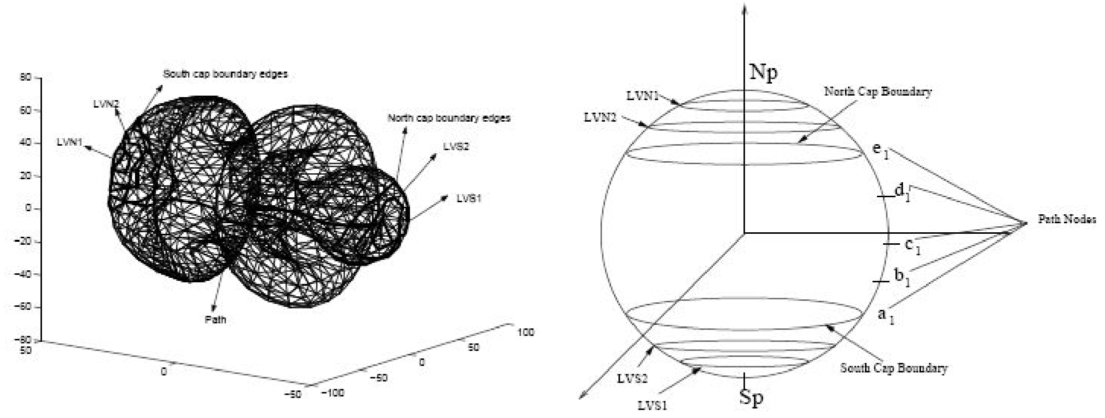


Fig. 2. [a]. Domain with path and layers at both the caps. [b] Caps and path nodes on the mapped sphere.

The cut-out shell boundary has four parts, viz. 'south cap boundary' on the left, 'north cap boundary' on the right, 'path bottom' in the bottom and 'path top' in the top. These are to be mapped to corresponding sides of a plane sheet of rectangular shape[5]. After stretching, there should be no folds on the surface and the mapping should be bijective, i.e. when a ray is passed normal to the sheet at an arbitrary point, it should intersect the mapped plane exactly once.

Here, the cut-out shell is considered as a thin membrane structure made of rubber. This is to be stretched on to sphere. The deformations required for this stretching at every node of discretized cut-out shell itself are the boundary conditions of finite element (FE) problem. Hence, there are no force boundary conditions. This mapping is performed through a nonlinear finite element formulation, considering each face as an incompressible triangular membrane element with hyper-elastic material (rubber) properties [17]. This finite element problem is solved by large-deformation formulation with an updated Lagrangian technique. The material of the membrane is assumed to obey a Mooney-Rivlin constitutive law though any other hyper-elastic law will also suffice. Considering the complexity of domain, geometric nonlinearity is also included in the formulation. This is implemented effectively in the FEA software,

[4] Longitude value is not defined at a pole.

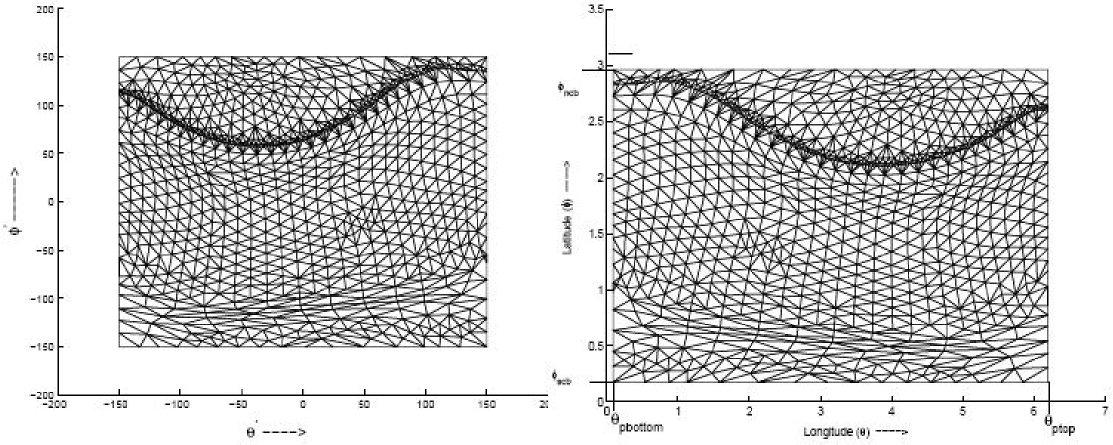
[5] Size of the rectangle is to be found based on the size of the domain and number of iterations of FE analysis

ABAQUS, which is used for the nonlinear Mooney-Rivlin membrane analysis. Formulation details related to Mooney-Rivilin analysis can be found in a standard text-book or in [16].

Atlas of the cut-out shell

The displacement of each node on the cut-out shell is available after the finite element analysis solution. Adding these displacement values to the original coordinates gives the position of all nodes on the target sphere. Now, by projecting the final shape of the cut-out shell onto the plane of stretch, we develop a chart (say *S*) as shown in Fig. 3(a). Now this chart is scaled to get the final atlas. The horizontal axis, or longitude (θ), is restricted between θ_{ptop} and $\theta_{pbottom}$ whereas the vertical axis, or latitude (φ), is restricted between φ_{scb} and φ_{ncb} as shown in Fig. 3(b). Here θ_{ptop} and $\theta_{pbottom}$ are the longitude values of the top node and bottom node on the path peel and similarly for φ_{scb} and φ_{ncb}. With this, we have the complete atlas of the entire domain boundary. The coordinates of the points on the mapped sphere are calculated using

$$x' = r \sin\varphi \cos\theta, \quad y' = r \sin\varphi \sin\theta, \quad z' = r \cos\varphi. \tag{2}$$



(a) (b)

Fig. 3. [a]: Projection of deformed cut-out shell onto the plane of stretch. [b] (φ−θ) atlas of the cut-out shell.

2.2. Inside mapping

The set of triangles of the domain which do not fall on the boundary represent the interior of the domain. For completing the mapping of the domain, we map this part to the interior of the target sphere and hence find the coordinates for all nodes. This step is achieved by solving a simple finite element problem. Here, the domain is visualized as *3-D* structural assemblage of many 'trusses'. Each edge of a tetrahedron is considered as a truss element. As we already have the displacement of boundary nodes from the previous boundary mapping, we use these values as boundary conditions and solve for inside nodes. Now, the problem is stated as, “when boundary nodes of the original domain are mapped to the surface of the sphere, what are the locations of inside nodes?” This is relatively less complicated than boundary mapping. Using the standard procedure [15], we calculate stiffness matrix for each truss and assemble them all to form the global stiffness matrix. Then we apply boundary conditions and solve the resulting linear system of equations. This procedure is elaborated below.

2.2.1. Determination of element stiffness matrix [K$_e$]

In the present problem, the truss is a 2-noded element in *3-D* reference frame. The finite element problem is solved by minimization of strain energy using weighted residual method. For a uniform bar of length *l*, cross-sectional area *A*, and Young’s modulus of elasticity *E*, the force-displacement relation is

$$\{F_i\} = [K]\,\{u_i\} \tag{3}$$

where, $\{F_i\} = [F_{1x}\, F_{1y}\, F_{1z}\, F_{2x}\, F_{2y}\, F_{2z}\,]^T$ = vector representing force components acting on the element at its nodes, and $\{u_i\} = [u_1\ v_1\ w_1\ u_2\ v_2\ w_2]^T$ = vector representing displacement components of the nodes of an element. The force displacement relation derived above is with respect to local coordinate system (X, Y, Z). It has to be transformed to global coordinate system (X_0, Y_0, Z_0) before assembling into the global stiffness matrix. Hence, the final force-displacement relation with respect to global coordinate system is

$$\{F_i^0\} = [T]^T [K][T]\{u_i^0\}, \tag{4}$$

where [T] is the transformation matrix[6]. The final form of the element stiffness matrix in global coordinates is given by

$$[K^e] = (AE/l)\ [M] \tag{5}$$

where, *A* is the cross-sectional area, *E* is Young’s modulus of elasticity and *l* is the length of uniform bar and

$$[M] = \begin{bmatrix} [C] & -[C] \\ -[C] & [C] \end{bmatrix}, \tag{6}$$

$C_{mn} = C_{xm}C_{xn}$; $m = x^0, y^0, z^0$; $n = x^0, y^0, z^0$;

$$l = \sqrt{(x_j^0 - x_i^0)^2 + (y_j^0 - y_i^0)^2 + (z_j^0 - z_i^0)^2},$$

$C_{x,y0}$ = direction cosine of OX with respect to OY^0 = $(y_j^0 - y_i^0)^2 / l$ etc.

[6] The transformation matrix T is orthogonal, hence $[T]^T = [T]^{-1}$.

2.2.2. Assembly and boundary conditions

Using the connectivity information, all the element stiffness matrices are assembled into a single matrix called the *global stiffness matrix* [K_{glob}]. The global force-displacement relation is

$$\{F_{glob}\} = [K_{glob}]\ \{u_{glob}\}. \tag{7}$$

The vector differences between the coordinates of the nodes on the original *3-D* domain and the coordinates of the corresponding nodes on the sphere is calculated, which give the displacement boundary conditions for the finite element problem to solve for the displacement of the internal boundary nodes. As there is no external force, the force vector $\{F_{glob}\}$ is zero.

2.2.3. Solution of the system of equations

The matrices [K_{glob}] and $\{F_{glob}\}$ will get modified, after applying the boundary conditions. Boundary conditions can be treated using either penalty approach or elimination approach [15]. We used elimination approach which involves modifying the above linear system by eliminating the rows and columns corresponding to the nodes at which boundary conditions are specified. This will reduce the size of the linear system. The force-displacement equation (Eq. 7) is then solved by using Gauss-elimination method to obtain the nodal displacement vector, {U}. By adding this vector to the vector of initial positions of the nodes in the original region, we get the positions of these nodes in the mapped region. This completes the mapping of the domain.

Since this mapping has been arrived at through the minimization of a positive definite energy function, it turns out to be bijective. As a consequence, in particular, it guarantees that the inside nodes of the original domain are mapped to the interior of the sphere.

3. Results

The entire method is implemented in MATLAB programming environment except the Mooney-Rivlin analysis part which is carried out using the standard FEA solver, ABAQUS. For all cases, the material properties used are same. To demonstrate the utility of the domain mapping, we also applied this method for path planning. Even though we tested the method for many cases, only three of them are reported here. The following results show the effectiveness of the method. Unlike the existing works, we show the resulting *mapping* on *latitude–longitude* chart of the target sphere instead of the sphere itself. Though the mapping shown on a sphere is more appealing it is not visible on the entire domain since a complete sphere is never visible on the paper. This atlas representation followed here clearly shows the quality of mapping over the entire domain.

3.1. Domain I

The region shown in the Fig. 4(a) is a surface of revolution and it is simply connected.

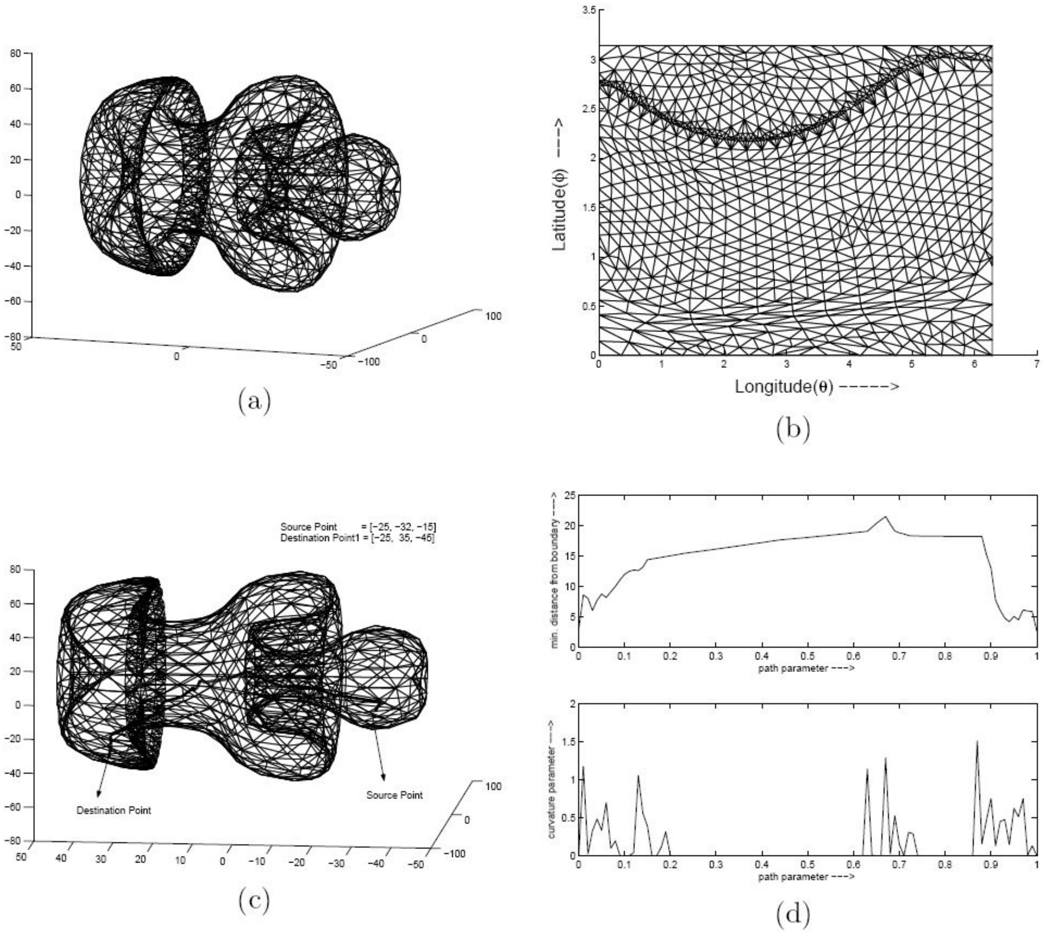
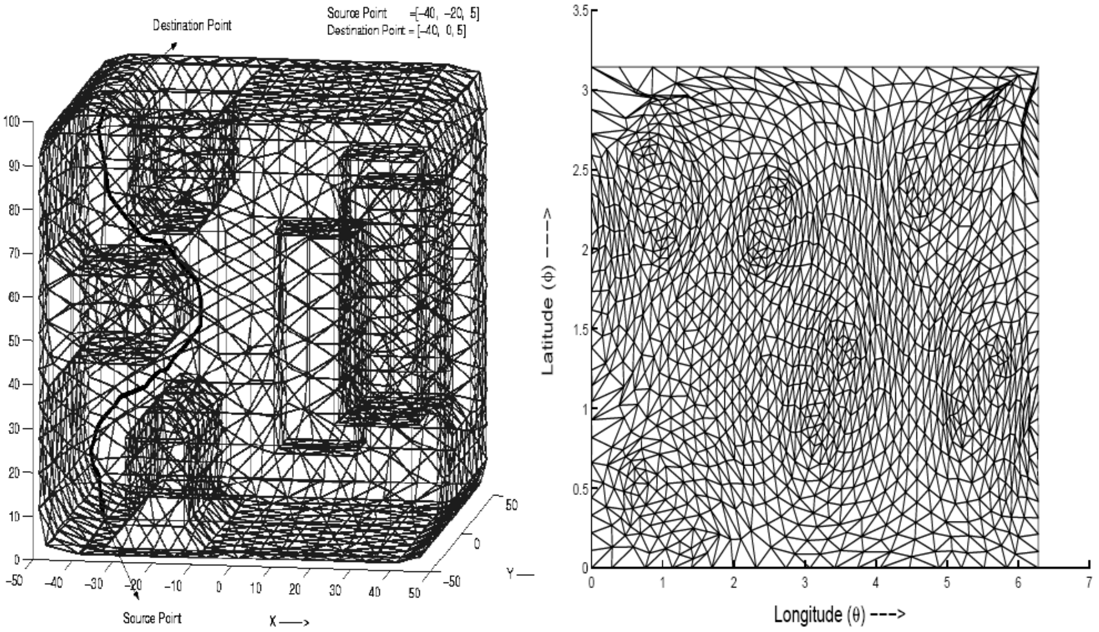


Fig. 4. Domain I: [a] Discretized region. [b] Latitude(φ)-longitude(θ) atlas of boundary of the region. [c] Path between source and destination points. [d] Obstacle avoidance and curvature parameter of the path.

This has been constructed by making a curve using the free spline in the XY- plane and then rotating it about the Y-axis. This region is mapped to the sphere and the atlas is prepared as shown in Fig. 4(b). The mapping obtained is bijective, without any folds. As an application of this method, we planned a few paths in the domain. One such path is shown in Fig. 4(c). Since it is a simply connected region, difficulty is posed by boundary only. As we can see, the paths planned are away from boundary. This is verified by calculating the distance between the boundary and path which is shown in Fig. 4(d). Whenever the source or destination points are close to boundary, the path bends to avoid collision, otherwise it is almost straight. This is measured by calculating the curvature of the planned path. Except when the path is close to the boundary, the curvature value is almost zero along the path can be seen in Fig. 4(d).

3.2. Domain II

The second region illustrated here is a 3-D maze. This case is much more complicated.



(a) (b)

The region is constructed by drawing a cube, then extruding several rectangular blocks into the cube and thereby making the rectangular blind holes by Boolean operations. The domain is discretized using tetrahedral elements as shown in Fig. 5(a). Using the same hyper-elastic material properties, the resulting FE problem is solved to get the final atlas of the domain as shown in Fig. 5(b). Computationally, this case was found more expensive than the previous one as it is more complicated. An iterative approach is followed in solving the nonlinear FE problem instead of flattening the cut-out shell in a single stretch. The mapping is bijective. To demonstrate this, an optimal path is planned between two points which are close to the boundary as shown in Fig. 5(a). Because of the cavities present on the outer boundary, the layer based approach of Suryawanshi et al.[1] fails in this case.

3.3. Domain III

This domain is a *3-D* star shaped one. To construct this region, first an ellipse is drawn, then a copy of the same ellipse is revolved through 180° about Z-axis. Again, another copy of the resulting ellipsoid is generated and revolved through 180° about the Y-axis. The union of the three ellipsoids gives the shape of the region. The complete discretized region and (φ−θ) atlas of its boundary are shown in Fig. 6(a) and

6(b), respectively. This region is also mapped to a sphere successfully without any folds, thereby providing the bijectivity.

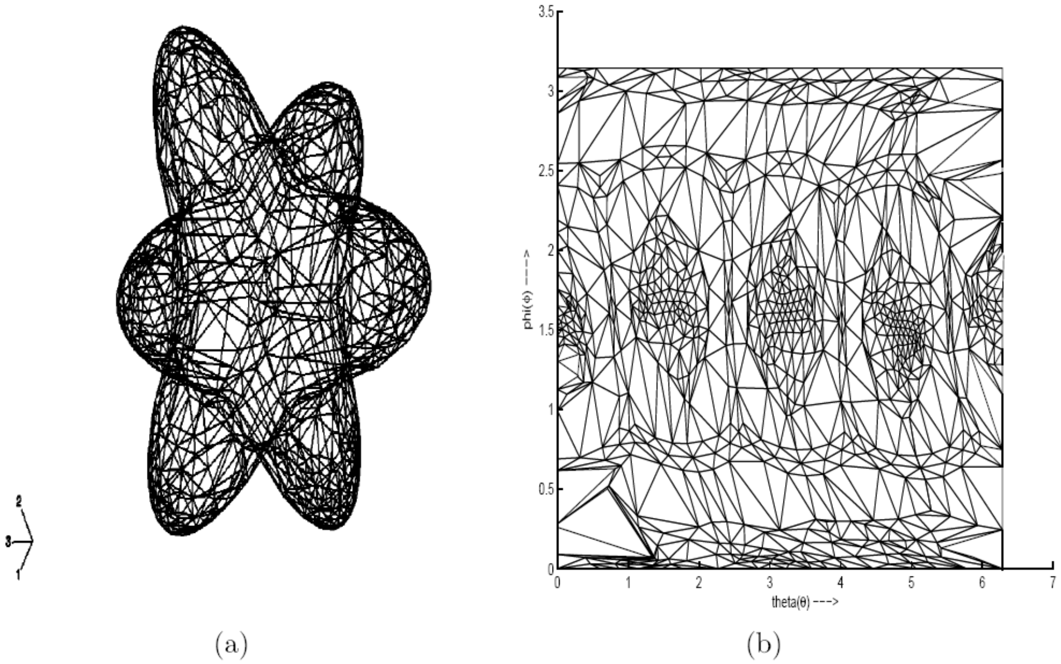


Fig. 6. Domain III: [a] Discretized region. [b] Latitude(φ)- longitude(θ) atlas of boundary of the region.

Mapping is performed for many other cases which are not presented here. In all cases, mapping was always bijective. Number of iterations taken by FE solver was dependent on the complexity of the domain.

4. Conclusions

The domain mapping method is an effective general method for reducing a complicated *3-D* domain to a simple one, thereby simplifying the problem. The effectiveness of the method is demonstrated using several complicated domains. The (φ−θ) atlas produced is a spherical parameterization of the original complicated domain.

There are several possible continuations of the present work. In principle, bijectivity is always guaranteed by this method. As one can see, the principle of this method can be extended to any dimension. In implementation, as mentioned earlier, choosing the appropriate material properties and its behavior is an important aspect. The bijectivity is lost if the chosen material does not allow the stretching required for the mapping. Such cases can be handled by some modifications like taking different material properties for different parts of domain, solving the finite element problem in more iterations etc. An inherent advantage of this method is that the loss of bijectivity is *localized* in the cases where bijective mapping is not obtained. This means that some local corrective strategy is enough for ensuring bijectivity. There is no need to restart the method afresh as in some other methods. Hence this method has strong potential for many application problems.

References

1. A. B. Suryawanshi, M. B. Joshi, B. Dasgupta, A. Biswas, Domain mapping as an expeditionary strategy for fast path planning, Mechanism and Machine Theory 38 (2003) 1237–1256.
2. Y. T. Yuan, Y. S. Ching, New algorithms for fixed and elastic geometric transformations models, IEEE Transactions on Image Processing 3 (4) (1994) 355–366.
3. J. R. Kent, W. E. Carson, R. E. Parent, Shape transformation for polyhedral objects, Computer Graphics 26 (2) (1992) 47–54.
4. A. Sheffer, C. Gotsman, N. Dyn, Robust spherical parameterization of triangular meshes, Computing 72 (2004) 185–193.
5. E. Praun, H. Hoppe, Spherical parametrization and remeshing, Proceedings of ACM SIGGRAPH, 2003, 340–349.
6. M. Alexa, Recent advances in mesh morphing, Computer Graphics Forum 21 (2) (2002) 173–196.
7. C. Grimm, Simple manifolds for surface modeling and parametrization, Proceedings of the Shape Modeling International 2002, 2002, pp. 237–244.
8. C. Gotsman, X. Gu, A. Sheffer, Fundamentals of spherical parameterization for 3-D meshes, Proceedings of ACM SIGGRAPH, 2003, 237–244.
9. M. Quicken, C. Brechbuhler, J. Hug, H. Blatmann, G. Szekely, Parametrization of closed surfaces for parametric surface description, Proceedings of IEEE Computer Society Conference on Computer Vison and Pattern Recognition, 2000, 354–360.
10. A. Khodakovsky, N. Litke, P. Schr¨oder, Globally smooth parameterizations with low distortion, ACM Transactions on Graphics 22 (3) (2003) 350 – 357.
11. M. S. Floater, Parameterization and approximation of surface triangulation, Computer Aided Geometric Design 14 (3) (1997) 231–250.
12. H. K. Voruganti, B. Dasgupta, G. Hommel, Harmonic function based domain mapping method for general domains, WSEAS Transactions on Computers 5 (10) (2006) 2495–2502.
13. W. E. Lorensen, H. E. Cline, Marching cubes: A high resolution 3-D surface construction algorithm, Proceedings of the 14th annual conference on Computer graphics and interactive techniques, 1987, 163–169.

14. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to algorithms, 2nd Edition, MIT Press and McGraw-Hill, 2001.

15. C. T. F. Ross, Finite element methods in structural mechanics, Ellis Horwood Ltd, 1985.

16. Membrane elements, Abaqus theory manual, V 6.4.

17. G. K. Klute, B. Hannaford, Accounting for elastic energy storage in mckibben artificial muscle actuators, ASME Journal of Dynamic Systems, Measurement and Control 122 (2) (2000) 368–388.

Tangudu Srinivas Patro is currently working as Senior Software Engineer at MSC Software, India. He received his bachelor degree in Mechanical Engineering from University College of Engineering, Burla in 2001 and masters in "Solid Mechanics and Design" from Indian Institute of Technology Kanpur, India in 2004. He is currently a method expert to provide CAE solution to varied industries (like Aero, Automobile and Manufacturing) in different domains like Safety, Structural and Closure analyses.

Hari K. Voruganti is currently pursuing his Ph.D in Mechanical Engineering at the Indian Institute of Technology Kanpur, India. He was a visiting scholar in the Technical University, Germany in 2006. He obtained his Master of Engineering in Automation and Robotics from Osmania University, Hyderabad in 2003. His current research interests include geometric modeling for CAD, optimization, co mputational biology and robotics.

Bhaskar Dasgupta received his bachelor's degree in Mech Engg from Patna University, Patna, India (1991), master's degree in Mech Engg from Indian Institute of Science, Bangalore, India (1993) and Ph.D. from Indian Institute of Science, Bangalore, India (1997). Since 1997, he is teaching at Indian Institute of Technology, Kanpur, India in the Department of Mechanical Engineering, where currently he is holding the position of Associate Professor. His teaching and research interests include robotics, CAD, optimization, applied mathematics and computational biology.

Sumit Basu obtained the Ph.D degree from the Indian Institute of Science in 1999. He was a post doctoral researcher in the Delft University of Technology and The University of Groningen in the Netherlands. He joined the Mechanical Engineering Department of the Indian Institute of Technology Kanpur as an Assistant Professor in 2002. His current research interests include studying, primarily through computational methods, structure property correlations at various length scales in thin films, polymers, composites and other heterogeneous materials.