Spherical Gridding in Seismic Imaging

Igor Ravve* and Zvi Koren, Paradigm Geophysical

Summary

We introduce a new gridding method to interpolate the image function on the spherical or ellipsoidal surface. The function is specified at discrete control points, and the proposed method allows one to obtain a continuous distribution of the image function through the curved surface. To solve the problem, we simulate elastic bending of a thin shell reinforced by springs. The springs are located at the control points. We do not require an exact match between the interpolated function and the given values at the control points, and this makes it possible to get an optimum balance between accuracy and smoothness. The method is a key component in angle domain imaging, in particular in cases where the angle domain is not fully illuminated.

Introduction

The LAD (Local Angle Domain) display is a suitable tool to visualize the data at a given image point. The four LAD angles (Koren et al., 2007) constitute two imaging systems: a direction system where the image function is defined vs. the dip and azimuth angles of the normal to the reflection surface element, and a reflection system where the function is defined vs. the opening angle and the opening azimuth. Thus, for each image point two angle domain imaging systems can be created: directional and reflection. Both imaging systems are defined on a unit sphere. A point on the spherical surface is defined by two components of the polar angle: zenith and azimuth. In case of the direction system, each point corresponds to a certain direction of the inward normal to the reflection element, with all possible reflection angles and their azimuths summed up. In case of the reflection system, each point on the spherical surface corresponds to a certain opening angle and opening azimuth, with all possible directions of the normal summed up. The image function is given at a number of control points which may be irregularly spaced on the spherical surface. There is an array of input control points, and at each control point the three values are specified. Two of them are components of the polar angle: zenith and azimuth. In case of the direction system, each point corresponds to a certain direction of the inward normal to the reflection element, with all possible reflection angles and their azimuths summed up. In case of the reflection system, each point on the spherical surface corresponds to a certain opening angle and opening azimuth, with all possible directions of the normal summed up. The image function is given at a number of control points which may be irregularly spaced on the spherical surface. There is an array of input control points, and at each control point the three values are specified. Two of them are components of the polar angle that defines the location of the point on the sphere, and the third is the value of the image function. In this paper we introduce a new gridding technique to interpolate the image function at any location on the spherical surface. We relax the requirement of exact data fit at the observation (control) points, thus compromising between the accuracy and continuity of gridding. Sometimes, when the control points are biased to the in-line or cross-line direction, there is an option to “squeeze” the sphere, so that it becomes a spheroid or a general ellipsoid with all three axes distinct, and the gridding is being done over the ellipsoid surface.

Gridding Method

To solve the problem, we apply a mechanical analogy. Variations of the target function on a curved surface are modeled by bending displacements of a thin elastic shell. If the control values had to be fit exactly, we would specify the normal displacements at these points. Since the exact fit is not needed (and sometimes unwanted), we attach the pre-stretched springs to the shell at the control points locations. These springs exert normal forces. When the normal displacement of the shell at the control point is zero, the spring load corresponds to a pre-set control value of the image function. Varying the stiffness of the springs, as compared to the shell stiffness, we can reach the optimum smoothing, while keeping the reasonable accuracy. A similar approach was applied by Lancaster and Salkauskas (1986) and by Koren and Ravve (2006) for a planar 2D grid with irregularly spaced control data, where a thin elastic Kirchhoff plate is used.

Stiffness of Spherical Shell

A real mechanical shell exerts two kinds of stiffness: the bending stiffness (out of the shell plane), and the membrane stiffness (in plane). Normally, the membrane stiffness prevails, but in case of gridding it causes an unwanted effect. If we compress the true spherical shell at the poles, it protrudes at the equator due to the membrane stiffness.

Figure 1. Elastic deformation of spherical shell

Therefore, we account for the bending stiffness (or shell stiffness) only, searching the surface of a minimum curvature.

Finite Element Approach

A Finite Element approach is applied to study the elastic bending of a spherical/ellipsoidal shell resting on the springs. It includes the following stages:

- Finite Element mesh generation.
- Building the stiffness matrices of the shell elements, in their local frames of reference.
- Building the nodal and non-nodal spring matrices in the local frames.
- Rotation of elements' stiffness matrices from the local frames to global
- Assembling the element matrices into a global matrix
- Building the load vector, corresponding to the stretch of the springs. Reduction of the non-nodal forces (resulting from the non-nodal location of control points) to the nodes of the mesh.
- Solving equation set for one or several cases of the load. In the latter case we take into account that the stiffness matrix of an elastic construction is symmetric and positive definite, and thus it can be split into a product of the lower and upper triangular matrices. Such decomposition is to be done only once for all loads. Then for each load we have to solve two sets: with the upper triangular and with the lower triangular matrix. Note that the coefficient matrix is defined by the locations of the control points, while the load vector is defined by the values of the control function
- Interpolation of solution inside the triangular elements for the required non-nodal points.

Mesh Generation
There is no way to tile the sphere densely with a regular mesh. To generate the spherical mesh, we start from a regular icosahedron inscribed into a sphere (Sword, C., J. Claerbout, and N. Sleep, 1986, Finite-element propagation of acoustic waves on a spherical shell, SEP-50, pp. 43-78). This regular polyhedron shown in Figure 2 has 12 vertices, 20 faces and 30 edges.

![Figure 2. Regular Icosahedron](image)

We replace the icosahedron faces by “primary” spherical triangles. For this, the edges of flat triangles are replaced by geodesic curves (arcs of great circles). We can achieve a sufficiently dense tiling by subsequent dividing of each spherical triangle by medians into four smaller (derivative) spherical triangles as shown in Figure 3.

![Figure 3. Split of primary triangle into four derivatives](image)

This denser tiling will no longer produce a regular mesh. The distances between the neighbor points will depend on the positions of these points on the sphere. However, the irregularity is moderate and does not exceed a few per cents. Twenty primary spherical triangles, corresponding to the icosahedron faces, are equilateral. When we split a primary triangle into four derivatives, we get three peripheral isosceles triangles and a central equilateral (central). Each peripheral triangle occupies 23.79 % of area of the primary triangle, and the central triangle 28.63 % of area. Further split ruins even isosceles triangles. The lack of symmetry is caused by decrease of the area of triangles after split. The sum of angles of a spherical triangle depends directly on its area. Split is continued until the desired level of recursion. For a recursion $k$, the number of vertices is $10 \times 4^k + 2$, number of faces is $20 \times 4^k$, and number of edges is $30 \times 4^k$. The nodes are numerated consequently by vertical slices (or levels), from the top vertex to the bottom vertex. There are $3 \times 2^k + 1$ vertical levels, and the optimum numeration of the nodes leads to the minimum possible band width of the stiffness matrix, per degree of freedom at a node, $B = 5 \times 2^k + 2$. A shell in bending has three nodal degrees of freedom: a translatory normal displacement and two rotations about the axes in the shell plane. The mesh includes 12 primary nodes (icosahedron vertices), while the other nodes are derivative. Five triangular faces meet at each primary node, and six faces meet at each derivative node. This rule can be used to check the correctness of the mesh. The proposed recursive mesh is suitable for implementation: each face object includes, in particular, four pointers to derivative faces. Triangular shell elements at the highest level of recursion
are small and thus shallow. These curved surfaces can be approximated by planar triangular plates with the same location of vertices.

**Parametric Ellipsoidal Surface**

The mesh is first generated on a unit sphere. When the mesh is ready, we stretch the unit sphere to a general (scalenol) ellipsoid with semi-axes \(A, B, C\).

\[
\frac{x^2}{A^2} + \frac{y^2}{B^2} + \frac{z^2}{C^2} = 1
\]  
(1)

The Cartesian coordinates are related to polar angles by

\[
x = r \cdot \sin \theta \cos \phi, \quad y = r \cdot \sin \theta \sin \phi, \quad z = r \cdot \cos \theta
\]  
(2)

where \(\theta\) and \(\phi\) are zenith and azimuth, respectively, and \(r\) is the radius of ellipsoid at the current point,

\[
r = \frac{A \cdot B \cdot C}{\sqrt{B^2 C^2 \sin^2 \theta \cos^2 \phi + A^2 C^2 \sin^2 \theta \sin^2 \phi + A^2 B^2 \cos^2 \theta}}
\]

**Coordinate frames**

We introduce three coordinate frames: a local frame, related to specific triangular element, a global frame, related to specific shell node, and a unique reference frame, related to the whole ellipsoidal surface. Axis \(z\) of the vertical frame is vertical, from the south pole to the north. Global axis \(z\) is normal to the elliptic surface at the given node, and global axis \(y\) points to the north (except for the poles). Orientation of local and global frames can be both specified vs. the reference frame. Alternatively, orientation of the local frame can be set vs. the corresponding global frame.

**Degrees of Freedom**

Plate (or shell) bending is described by bi-harmonic equation and requires continuity of displacements and its first derivatives. Degrees of freedom (DOF) are nodal displacements and rotations. Small rotations correspond to the components of displacement gradient,

\[A_x = -\frac{\partial w}{\partial x}, \quad A_y = +\frac{\partial w}{\partial y}\]

(4)

Displacements are interpolated by a complete cubic polynomial, which includes ten terms. Thus, each finite element has ten DOF. There are nine nodal DOF at the vertices of the triangle, and in addition, the last DOF is the normal translation at the centroid of the triangle,

\[w(x, y) = C_0 + C_1 x + C_2 y + C_3 x^2 + C_4 y^2 + C_5 x y + C_6 x^3 + C_7 y^3 + C_8 x^2 y + C_9 x y^2\]

(5)

At each point of the element, the displacement vector comprises three components,

\[\overrightarrow{V}(x, y) = \{A_x, A_y, w\}\]

(6)

Given the ten nodal displacements, the interpolation coefficients \(C_i\) can be found.

**Strain Energy**

The specific strain energy \(U_S\) per unit area is given by

\[
\frac{2U_S}{D} = \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2}\right)^2 - 2(1 - \nu) \left(\frac{\partial^2 w}{\partial x \partial y} - \frac{\partial^2 w}{\partial x^2} \frac{\partial^2 w}{\partial y^2}\right)
\]

The first term is the Laplacian squared, and the second term is the Gaussian component. Parameter \(D\) is the flexural rigidity of the plate,

\[
D = \frac{h^3}{12} \frac{E}{1 - \nu^2}
\]

(8)

where \(h\) is the plate thickness, \(E\) is the Young modulus and \(\nu\) is the Poisson ratio. Since we do not solve the true mechanical problem of shell bending, the Gaussian term can be omitted. Its effect is small, and not necessarily it improves the results of gridding. We may assume \(D = 1\), and the method reduces to the search of the minimum curvature of the surface, while satisfying the control values, at least approximately. The equation for the specific energy reduces to

\[
U_S = \frac{1}{2} \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2}\right)^2
\]

(9)

The elastic energy of a finite element is obtained by integration through the triangle area \(\Omega\),

\[
U = \frac{1}{2} \int_{\Omega} U_S \, d\Omega
\]

(10)

The specific energy depends on the coordinates of a given point and on the nodal displacements, and the full energy is a quadratic form of ten nodal displacements only. This makes it possible to establish the element stiffness matrix of dimension \(10 \times 10\).

**Spring Matrix**

The spring stiffness is a scalar value, but the control points are generally located inside the elements, and their locations do not necessarily coincide with the mesh nodes. Energy of a spring is considered proportional to the displacement at the control point squared. The control point displacement, in turn, depends on the location of the control point and on the ten nodal displacements of the finite element. This makes it possible to present the spring energy as a quadratic form of the nodal displacements, and this yields the spring stiffness matrix \(10 \times 10\). The total element stiffness matrix is the sum of the shell matrix and all spring matrices inside of this element. If the spring is located on the boundary between two adjacent elements (but not at the grid node), then we take it with the weight \(1/2\) for each triangular element. The nodal springs are accounted directly.
Load at Control Point

The spring load is a scalar normal force applied to the shell at the non-nodal location of a control point. This force is to be reduced to the nodal forces and moments. To establish the nodal load vector, equivalent to the non-nodal point force, we equate the two works: the work done by the point force and the work done by the nodal forces and moments. The displacement at the non-nodal point under the spring is expressed through the nodal displacements. Thus, the non-nodal force is reduced to a load vector of length 10.

Reduction of Stiffness Matrix

The element has nodes of different types: three vertex nodes with three DOF each and a centroid node with single DOF. This is inconvenient and impractical, and in addition, existence of the centroid node increases the bandwidth of the global stiffness matrix of the whole shell. Therefore we apply the reduction of the centroid DOF and convert the $10 \times 10$ matrix into $9 \times 9$ matrix with vertex DOF only. Assume that there is no external force applied to the centroid node, then the element equilibrium equation reads

$$
\begin{bmatrix}
K_A & K_B \\
K_B^T & K_C
\end{bmatrix}
\begin{bmatrix}
V_A \\
V_C
\end{bmatrix} = \begin{bmatrix}
F \\
0
\end{bmatrix}
$$

(11)

where the matrix is split into blocks of different dimensions, $V_A$ are displacements at the vertices, $V_C$ is the centroid displacement, and $F$ is the load (forces and moments) applied to the vertices. Expand the block-matrix equilibrium equation, and eliminate the centroid translation. This yields the reduced $9 \times 9$ element stiffness matrix,

$$
K = K_A - K_B \cdot K_C^{-1} \cdot K_B^T
$$

(12)

We reduce also the load vectors of length 10 to vectors of length 9 (Bazeley, G.P., Y.K. Cheung, B.M. Irons, and O.C. Zienkiewicz, 1965, Triangular elements in plate bending – Conforming and nonconforming solutions, Proceedings First Conference on Matrix Methods in Structural Mechanics, pp. AFFDL-TR-66-80, Air Force Institute of Technology, Dayton, Ohio, pp. 547-576). We keep the vertex loads, and split the centroid force. Let $i=1,2,3$ be indices of the triangle vertices, and $D$ - the centroid, whose location is

$$
3 x_D = \sum_{i=1}^3 x_i, \quad 3 y_D = \sum_{i=1}^3 y_i
$$

(13)

The additional loads at the vertices caused by the centroid force $F_D$ are

$$
M_{y}^{y} = \frac{y_{D} - y_{i}}{6} \cdot F_{D}, \quad M_{x}^{y} = \frac{x_{D} - x_{i}}{6} \cdot F_{D}, \quad F_{i} = F_{D} / 3
$$

After the problem is solved, the nodal translations and rotations are established, and we use the interpolation equation 5 to find the displacements at the control points and any other non-nodal points of the shell. Since the method works with the rotational DOF in addition to the translations, the control point may include meridian and latitude derivatives, in addition or instead of the function values. Thus, the gradients can be taken into account.

Figure 4 shows an example of spherical gridding. Green and red colors correspond to different ranges of output values of the interpolated function, and the dots show the locations of the control points.

Conclusions

A new spherical gridding method has been developed to interpolate the image function values through the surface of a spherical or an ellipsoidal shell. Gridding is based on a mechanical analogy and simulates elastic bending of a thin Kirchhoff shell resting on springs. The values of the image function and/or its derivatives can be set at control points. The requirement to fit the control points is relaxed, and one can compromise between the required accuracy and continuity (smoothing).