Manifolds defined by points: Parameterizing and Boundary Detection

C. W. Gear*, E. Chiavazzo† and I. G. Kevrekidis*

*Princeton University, Princeton, NJ 08544, USA
†Politecnico di Torino, ITALY

Abstract. THIS IS A PRELIMINARY VERSION

Keywords: Smooth parameterization, Noise reduction, Manifold boundary
PACS: 02.30.Hq, 02.40.Vh

1. INTRODUCTION

We assume that we have a set of points in n-dimensional (nD) space (the *native space*) that lie on a finite dD manifold that can be smoothly mapped to a dD Euclidean space without holes, and we would like to create a dD parameterization for the manifold. This type of problem occurs in various contexts, for example, parameterizing part of the surface of a 3D object. The problem we are particularly interested in arises in high-dimensional evolutionary differential equations in which the solution rapidly approaches a low-dimensional manifold. By computing the solution starting from different initial values we can collect samples of points on this low-dimensional manifold. Parameterizing the manifold is important for a number of follow-on tasks. If we wish to re-integrate from a slightly different initial values, restricting the solution to the manifold can lead to faster integrations techniques (because the problem is no longer stiff). If we have a stochastic problem that has a low-dimensional energy well and we wish to explore the time to escape from the well (which could take a long computational time with direct simulation) it would be helpful to initialize the computation beyond the currently explored region of the well to find additional points on the manifold, which means that one needs to extrapolate the manifold from its current boundary. Our objective is to develop an method that can be used as the basis for a robust code because it requires no human intervention.

Diffusion maps[2] have proved to be a powerful tool for analyzing point sets in a high-dimensional space for identifying clusters and manifolds, but sometimes they provide a poor parameterization of the manifold near its boundaries, as we will demonstrate. Diffusion maps use a parameter, ε, which is usually the order of the distance between near neighbors. In that case in the limit of a high density of points everywhere on a finite section of the manifold the diffusion map for small ε approaches a solution of the Laplace-Beltrami equation on the manifold with a no-flux boundary condition, so the spectrum of the numerical approximation (which is what a diffusion map returns) approaches the eigenvectors of the Laplace-Beltrami equation. The no-flux boundary condition causes some problems with the parameterization that we will outline. We have found that using a large ε avoids those difficulties, but introduces other problems when the manifold is curved so that points distant to paths in the manifold are close in the native space. We analyze the effect of large ε and suggest ways around this problem so a large ε can be used. This material is covered in Section 2. In Section 3 we consider ways to find the boundary of a manifold defined only by noisy points on it. If the manifold includes a part that looks like a small circle of close adjacent points (or a higher dimensional equivalent) but has a much larger extent attached to the part (a simple 2D example is the Northern part of the surface of a sphere truncated below a latitude line well into the Southern hemisphere), none of the methods we propose in Section 2 will provide a parameterization. In Section 4 we explore a new approach that attempts to replace the interior of the manifold and its boundary by two uniform resistors. It first simulates passing a current (in d different directions) through the boundary so that is can apply the resulting voltages to the interior whose simulation will then provide a parameterization.

2. DIFFUSION MAPS

Diffusion maps operate on a set of *points*, $\{P_i\}, i = 1, \cdots, N$, that have a *distance*, d_{ij}, defined between each pair. The points could represent any set of objects that have a measure of dissimilarity between them (the distance) although we interested in points in an nD Euclidean space with distance defined in the usual way. The diffusion map process starts by forming a dissimilarity function, $w(d_{ij})$, such that $w(0) = 1$ and $w(d)$ rapidly, monotonically approaches zero as d gets large compared to the typical spacing between near neighbors (an important parameter in the process). We will use

$$w(d) = \exp(-\left[\frac{d}{\varepsilon}\right]^2) \tag{1}$$

where ε is of the order of the typical spacing between near neighbors. A Markov matrix, K, is formed from the dissimilarity matrix $W = \{w_{ij}\}$ by defining the elements of K as

$$k_{ij} = \frac{w_{ij}}{\sum_{k=1}^{N} w_{ik}}. \tag{2}$$

Since all row sums of K are 1, $\mathbf{e} = [1, 1, \cdots, 1]^T$ is an eigenvector corresponding to the eigenvalue 1. All other eigenvalues are less than 1 and information about the structure of the point system can be derived from the eigenvectors corresponding to some of the next smaller eigenvalues.

How do we choose a sensible ε? We define the *cut value* of a connected graph with non-negatively weighted edges as the maximum edge weight such that if all longer edges are removed, the graph remains connected. Computing the cut value is fast as it can be done in $O(N^2)$ operations as shown in the Matlab code below. This code creates a *reachable set*, S_R by starting at node 1 and finding the minimum distance needed to reach another node not in S_R. This is added to S_R and the process continues until all nodes are in S_R. In this code, `D(i,j)` is the distance between nodes `i` and `j` (`i` and `j` run from 1 to `N`). `M` is a value larger than the maximum element of `D(i,j)`, `done(i)` is a row vector indicating which nodes have already been reached from the starting node and `best_distance` is a row vector containing the shortest distances to other nodes from any node in S_R.

```
done(1) = 1;
best_distance = D(1,:);
cut_value = 0;
for i = 2:N
    [mn, ix] = min(best_distance + M*done);
    cut_value = max(cut_value, mn);
    done(ix) = 1;
    best_distance = min([best_distance; D(ix,:)]);
end
```

Unless noted otherwise, all diffusion maps in this paper are calculated using ε = cut value. If ε is somewhat smaller than the cut value of a distance graph, the diffusion map for the graph will tend to locate the disjoint components of the graph when all edges a little longer than ε are removed. (This is the reason that diffusion maps are a powerful tool for cluster analysis.) This is illustrated in Figure 1 that plots the diffusion map eigenvector for 1,000 points uniformly randomly distributed on a a straight line of length 1. The cut value for this data set was about 0.008189 and 4 different ε's were used, starting at half the cut value and doubling each time. As can be seen in Figure 1, the smallest ε leads to a jump in the eigenvector parameterization. The next two parameterizations are non-smooth, but the largest value gives an almost smooth result.

While increasing ε generally increases the smoothness of the parameterization, that is not always the case, as shown in Figure 2. Two different uniformly random samples of 1,000 points were run with ε values of 0.06 (about 60 times the average spacing) and 0.12. We see that in the left-hand figure the larger ε gives a smoother result, while the opposite is true in the right-hand figure.

The “roughness" in a diffusion map parameterization occurs when the points are poorly distributed. In some of the examples in this paper we use a *box-controlled random* allocation of points. This only applies to dD rectangles and we first divide the region into a set of smaller, equally sized rectangles, and then assign one point uniformly randomly in each box. In the following example we assigned 1,000 points to a unit line using box controlled randomness. This guarantees that no point is further apart that 0.002. The diffusion maps for $\varepsilon = 0.12 * 2^i$ for $i = 0, \cdots 3$ are shown in

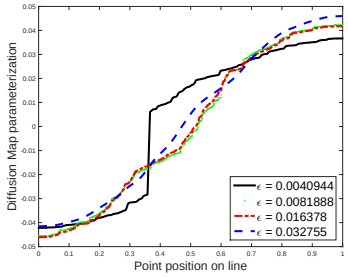

FIGURE 1. Effect of different ε's starting from half the cut value. (All figures are in color in the online version.)

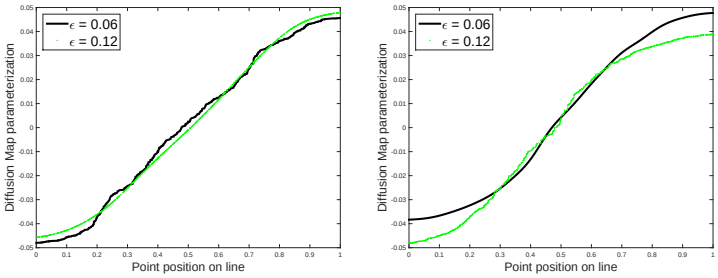

FIGURE 2. Effect of larger ε's on two different samples, one is smoother, the other not. Both data sets are 1,000 uniformly random points on the unit line.

Figure 3. All maps are smooth. The smallest ε value (the thickest line in the graph) has a nearly horizontal slope at the boundary due to the no-flux boundary condition. As ε increases, this affect disappears and the result in a nearly linear parameterization for the largest ε.

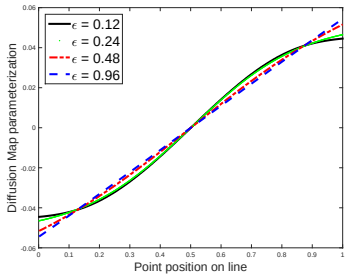

FIGURE 3. Effect of different ε's for 1,000 points uniformly randomly placed in 1,000 equally sized boxes on the unit line.

This suggests that we can get a good parameterization by increasing ε but then other problems can occur. This is illustrated in the next example where we took the data used for Figure 3 and curved the straight line into 7/8 of a circle as shown in the left of Figure 4. The right part of that figure shows the diffusion map for different ε. For the smallest ε we get a unique parameterization, but for the second value (which is around 2/7 of the distance of the end points of the curve) we lose uniqueness, and the effect becomes more noticeable for larger ε.

The no-flux boundary condition for small ε has further impacts in higher-dimensional manifold situations. At any boundary we lose one of the parameterizations. For the 2D manifold shown in Figure 5 (which was created by placing 61 by 61 points regularly on a 2 by 2 square, and removing all points outside a unit disk inscribed in the square) we computed the first two eigenvectors of the diffusion map using $\varepsilon = 1/30$ (the spacing between nearest neighbors, and hence the cut value of the graph) and plotted the lines of constant diffusion map values for both eigenvectors. Either the diffusion coordinate has to have the "cosine" behavior at the boundary (i.e., a slope of zero) or the constant contours have to be perpendicular to the boundary. As we can see in Figure 5 the plotted contours are perpendicular to the boundary.

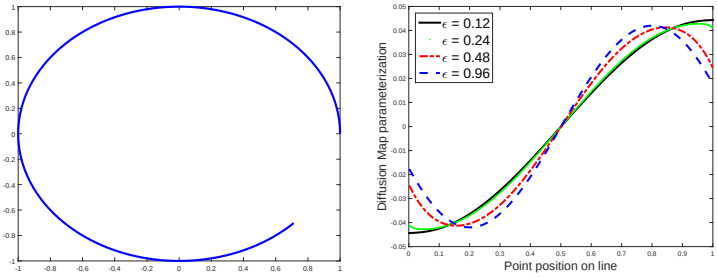

FIGURE 4. Data on 3/4 of a circle and its diffusion map. End points are 0.838 apart.

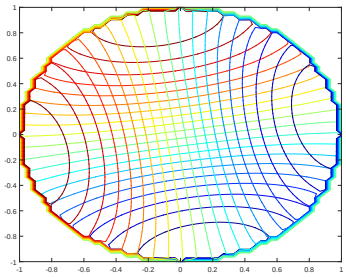

FIGURE 5. Contours of constant diffusion map coordinates for a circle.

If there is a corner in the manifold (a meeting point of all hyper-surfaces of the manifold, such as the corner of a cube) then all diffusion map coordinates must be locally constant at that point since their derivatives are zero in d independent directions (d is the manifold dimension). An illustration of this is given in Figure 6 which shows one diffusion map parameterization of a 2D manifold in terms of two variables that are known to parameterize the manifold. It has three corners, and we can see that the parameterization is flat at the corners.

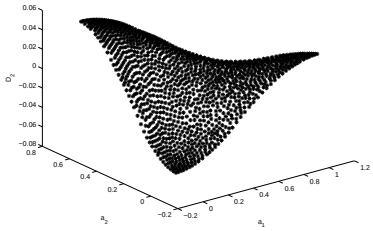

FIGURE 6. One diffusion map coordinate of a 2D manifold with 3 corners.

2.1. Finding the Manifold Dimension

In some applications one may need to estimate the manifold dimension locally (e.g., in data classification problems involving multiple sets of data that might have different dimensionality). In that case one should consider methods such as the one in [4]. In the applications of interest to us, the manifold generally has a constant dimension and it is likely that the additional statistical information available from a global approach will lead to better results.

It has been suggested that diffusion maps can be used to determine the dimension of the manifold, but our experience has not found that to be the best way. If the points are very regularly placed, this is not difficult, as the following example shows. In this, we placed a regular 40 by 40 grid of points on a unit square and then stretched and wrapped it around 3/4 of a circular cylinder of radius 2 and length 1.2. The original data is shown on the left in Figure 7. ϵ

was chosen as the spacing between points around the cylinder (this is the cut value for this graph and is longer than spacing between points along the cylinder). The first non-trivial eigenvector, ξ_2, parameterizes the direction around the cylinder: the right part of Figure 7 shows the angle around the cylinder versus that eigenvector. This graph is a scaled cosine of the angle after translating it and scaling it to lie in the interval $[0, \pi]$ because of the no-flux boundary condition.

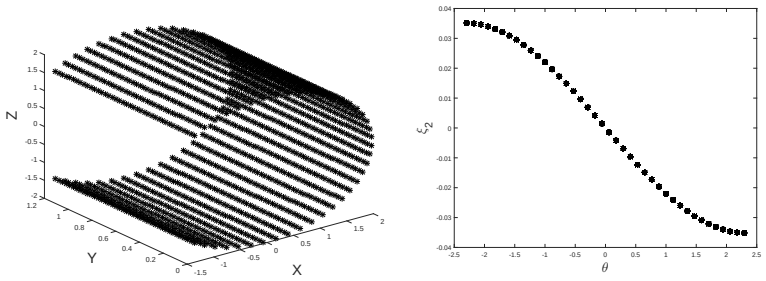

FIGURE 7. Regular data on a cylinder and first diffusion eigenvector.

We can plot subsequent eigenvectors against the first non-trivial one to get the results of the type seen in the left side of Figure 8. This indicates that the ξ_3 is dependent on ξ_2. In fact, because the ξ_2 characterizes the longest direction (around the cylinder), and the length around the cylinder is significantly longer that the length along the cylinder, ξ_3 is a second eigenvector in the same direction - actually proportional to the cosine of twice the angle. This behavior continues in this example until finally ξ_{10} can be seen to be independent of ξ_2 as shown in the right hand figure in Figure 8. This indicates that the manifold dimension is at least 2. From then on, we should plot the these two eigenvectors versus subsequent ones to determine if there is another dimension.

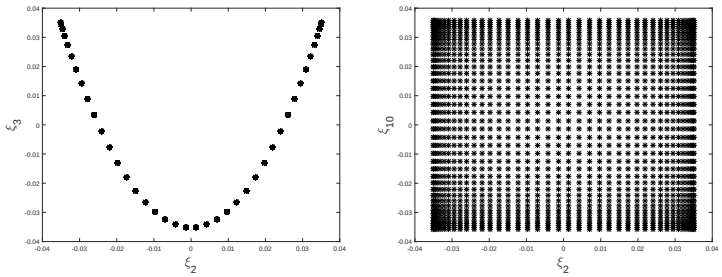

FIGURE 8. Regularly distributed points. ξ_2 plotted against ξ_3 indicating a dependency, which continues until we plot ξ_2 against ξ_{10}.

However, when the points are poorly distributed, the dependency is not so clear, as shown in Figure 9 which shows the ξ_2 plotted and the next 4 eigenvectors for 500 uniformly random points on a 1 x 0.2 rectangle. It is not clear which is the second parameterizing eigenvector, and that would make it hard to determine the dimension of the manifold.

The simplest global method to estimate a manifold dimension is to use a log-log plot to estimate the Correlation Dimension (see, for example, Section 4.3 in [1]). Since we need to compute the $O(N^2)$ point-pair distances, all that need to be done is to sort them (an $O(N^2 \log N)$ operation) and to examine the log-log graph of number of edges of length less than L versus L. To illustrate this a spiral was generated in polar coordinates using $R = \theta/2\pi$ for $\theta \in [0, 6\pi]$. Then 1,000 points were placed along the resulting curve uniformly randomly. The left pane in Figure 10 shows the points with no noise. The right pane shows the points with Gaussian noise along the radius with standard deviation 0.1. (The data was recorded in 3 dimensions, and there was similar noise in the third dimension not shown in the figure.) Figure 11 shows the log-log plot for levels of noise with a standard deviation of σ. The solid curve ($\sigma = 0$) has a slope of 1 between abscissa values of 2×10^{-4} and 1, indicating the 1-dimensionality of the manifold. (The bump in the plot at abscissa 1 is dues to the fact that the spirals are unit distance apart.) As noise is added the early part of the slope jumps to 3 (the noise in 3D space makes it look like a 3D manifold until we are dealing with lengths somewhat larger than σ. At the same time, the section of slope 1 is reduced until at $\sigma = 0.1$ it is too short to get a reliable dimension estimate.

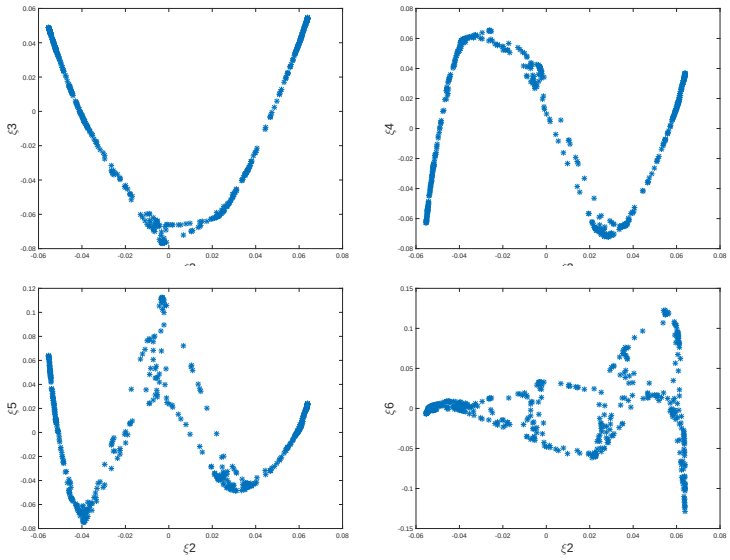

FIGURE 9. Uniformly random points in 1 x 0.2 rectangle. ξ_2 plotted against ξ_i for $i = 3,4,5$ and 6.

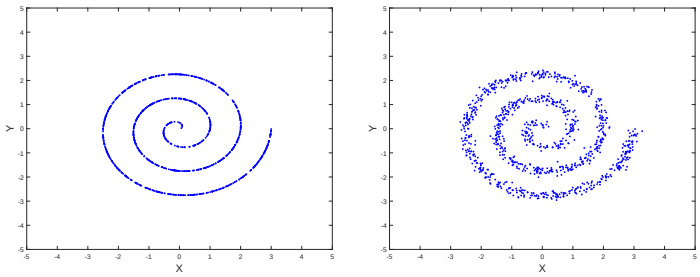

FIGURE 10. Uniform random points on a spiral: No Noise (left) and 0.1 Gaussian noise (right).

2.2. Large ε Diffusion Maps

We saw in Figure 3 that a large ε could avoid the no-flux issue at the boundary and provide a better parameterization. This is particularly apparent in the Figure 12 which is repeat of the data used for Figure 5 but using $\varepsilon = 2$ (the diameter of the circle) rather than the nearest neighbor spacing of 1/30.

The parameterization in Figure 12 is very uniform and avoids the no-flux problem altogether. We will analyze the use of a large ε. Write $\delta = 1/\varepsilon^2$ and assume that we can ignore $O(\delta^2)$ terms. Then, from eq. (1) we have

$$w_{ij} = 1 - d_{ij}^2 \delta.$$

Hence

$$\sum_{k=1}^{N} w_{ik} = N - \sum_{k=1}^{N} d_{ik}\delta$$

so

$$k_{ij} = (1 - (d_{ij}^2 - \sum_{k=1}^{N} d_{ik}/N)\delta)/N$$

or

$$K = (E - F)/N + O(\delta^2)$$

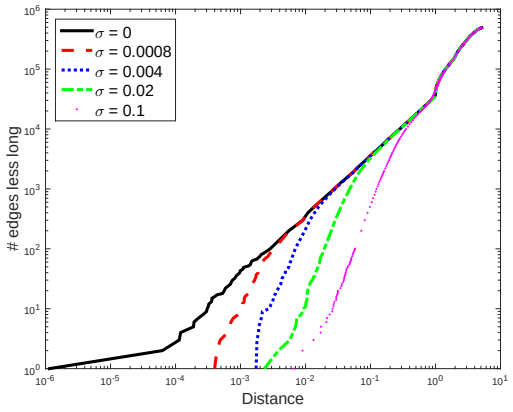

FIGURE 11. Log-log plot of Number of edges less than length vs length.

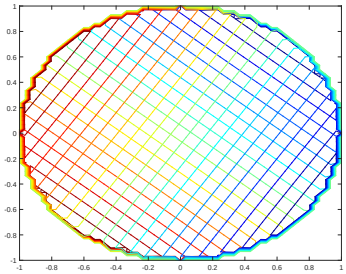

FIGURE 12. Two diffusion map coordinate isobars for $\varepsilon = 2$ (See Figure 5 for the usual ε.)

where $E = \mathbf{e}\mathbf{e}^T$, $F = H(I - E/N)$ and the symmetric matrix H has components $H_{ij} = d_{ij}^2$. $\mathbf{e}$ is a right eigenvector of F and K corresponding to eigenvalues of 0 and 1 respectively.

Consider the symmetric matrix $A = (I - E/N)F = (I - E/N)H(I - E/N)$. It has a complete set of orthogonal eigenvectors. $\mathbf{e}$ is one eigenvector of A corresponding to an eigenvalue of 0, so all other eigenvectors, $\mathbf{e}_i$, are orthogonal to $\mathbf{e}$, or $\mathbf{e}^T\mathbf{e}_i = 0$. Since $E = \mathbf{e}\mathbf{e}^T$ we have $A\mathbf{e}_i = \lambda_i\mathbf{e}_i = (I - E/N)H\mathbf{e}_i = F^T\mathbf{e}_i$. Hence, all eigenvectors of A except for $\mathbf{e}$ are eigenvectors of F^T with the same eigenvalue, and this set of eigenvectors plus $\mathbf{e}$ are mutually orthogonal. An interesting fact about F^T, shown in [3] is that the eigenvectors corresponding to non-zero eigenvalues provide the PCA of the points, in fact, the eigenvectors provide a Cartesian set of coordinates for the points from an origin at the mass center of the points. It is also shown in that reference that if the points are arranged on a regular grid on a d-dimensional rectangular prism, that is also true for the eigenvectors of F itself. An interesting experimental observation is that this also seems to be approximately true when the points are "reasonably" distributed. Since, if an eigenvector of F is orthogonal to $\mathbf{e}$ it is also an eigenvector of K, this may be the reason why a large ε returns fairly good results from the eigenvectors of K (although we have no mathematical arguments at this time).

2.2.1. Modifying the Data to Use a Large ε

Since a large ε gives a much better parameterization, we need a mechanism to increase distances between points that are far apart in the manifold but close in the native space so that their distance is moderately large compared to the ε we plan to use. We have tried two approaches to this:

1. Append a multiple of a function of the diffusion map coordinates for small ε to the native coordinates to enlarge the distance of points distant in the manifold,.
2. Remove all edges longer than a small multiple of the cut value to get a modified graph, G', and replace those edges by the shortest path between points in G'.

We applied the first method to the 1D spiral manifold shown in Figure 13(a). It has 1,000 points placed box-controlled randomly along its length (which is $L = 28.2743$). The cut value of the graph (0.0576) was used as the ε to get the diffusion coordinate shown in Figure 13(b). Then L times the arc cosine of the diffusion map coordinate scaled to lie in [-1, 1] was introduced as a third coordinate shown in Figure 13(c) and the diffusion map coordinate of the 1D manifold was computed from the modified data using $\varepsilon = 2L$. The result is shown in Figure 13(d).

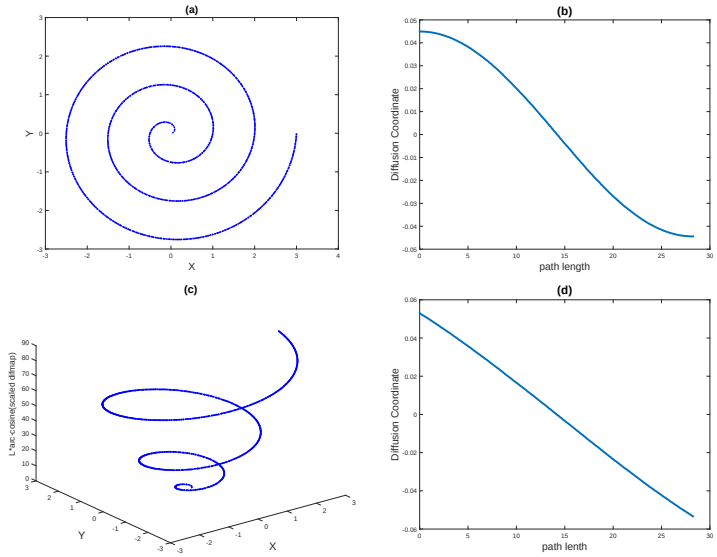

FIGURE 13. Using modified data: including a multiple of the arc cosine of the scaled diffusion coordinate for small ε as an additional coordinate. (a) Spiral data: 1,000 points uniformly random in 1,000 equal length boxes along the spiral of length $L = 28.2743$. (b) Diffusion map with $\varepsilon = 0.0576$ - the cut value for the graph. (c) Modified data with Z axis = L arc-cosine(diffusion map scaled to lie in [-1,1]). (d) Diffusion map of modifeid data with $\varepsilon = 2L$.

The last example gives very good results, but we would get good results for any 1D manifold by simply taking the arc-cosine of the single diffusion coordinate scaled to lie in [-1, 1] (which is what we did, in effect). The example is really given so that we can present the graphic in Figure 13(c). So, why not just take the arc-cosine of the diffusion map coordinates? That doesn't solve the boundary problem. A more challenging example is given in Figure 14 which is a a unit square in the X-Y plane projected up to the 3D paraboloid given by $z = 4a * ((x-0.5)^2 + (y-0.5)^2)$ with $a = 0.9$. The points were generated by selecting them uniformly randomly in the X-Y square, lifting them to the paraboloid, and rejecting any point that was closer than 0.0421 in the 3D space to a retained point. A maximum of 1,000 points were retained, and the process stopped after 6,000 rejections. The sample in this case had 921 points.

The first two diffusion map vectors are shown in Figure 15(a) and (b) plotted against the X and Y coordinates. (Note that the coordinate axes have been rotated separately to give the best view of each diffusion coordinate so that the X-axis is to the left in Figure 15(a) and right in Figure 15(b).) Each map runs corner to corner of the square (the longer directions). There is a slight cosine effect at these corners and some flatness in the other two corners for each map. Figure 15(c) and (d) show the two diffusion map coordinates after appending the arc-cosines of the scaled diffusion map coordinates from the previous step. There is no flatness or cosine effect at the corners. Figure 15(e) and (f) show the results of the second method (removing all long edges and recalculating the distances based on the shortest path in the reduced graph). Although these results look fairly good in the figures, the results are not as smooth because the shortest path may be a poor approximation to the manifold path length.

The first method will not work in some cases because diffusion maps identify the "longest" direction in the figure as the first diffusion coordinate. For example, if we change the a in the paraboloid $z = 4a * ((x-0.5)^2 + (y-0.5)^2)$ used above to 6 rather than 0.9 we get the data shown in Figure 16(a). Its first diffusion coordinate is shown in Figure 16(b). It does not give a unique parameterization.

Neither method will work for the manifold shown in Figure 17. It is the surface of a sphere with everything removed south of latitude 67.5^oS. The boundary points are shown as circles. The same figure shows the diffusion map coordinates for the lines of constant latitude (plotted every 10.5^o). The boundary is the thick line in the plot. Because of the closeness of the points on the boundary and its shortness compared to the rest of the data, a diffusion

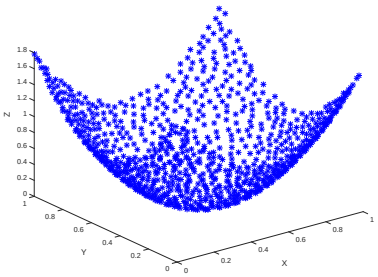

FIGURE 14. 2D Paraboloidal random manifold

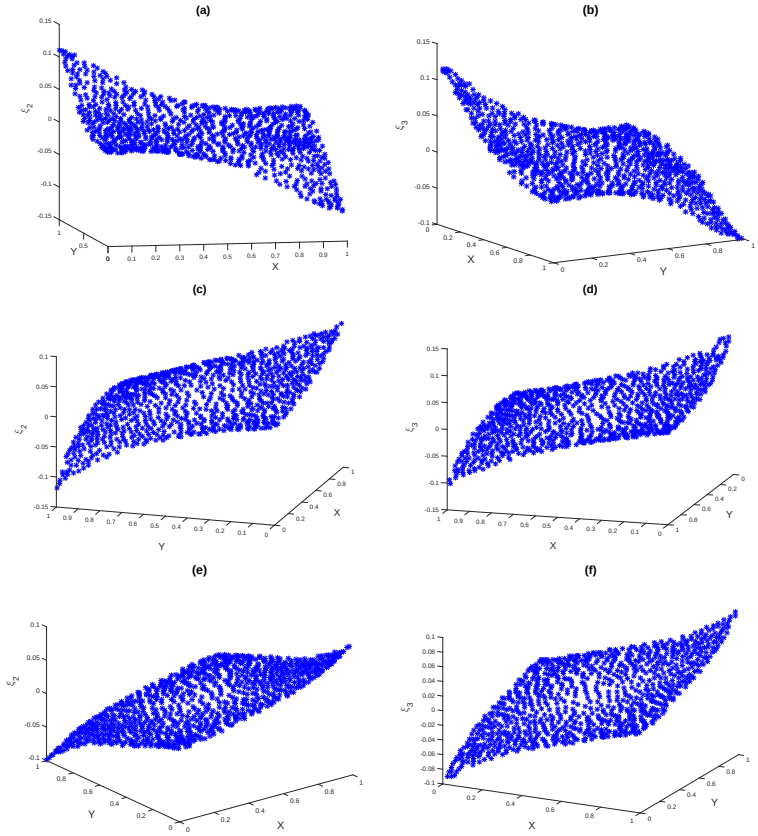

FIGURE 15. (a) Fist Diffusion Map vector. (b) Second vector. (c) First large ε Diffusion Map vector of extended coordinates. (d) Second vector. (e) First large ε Diffusion Map vector of modified distance matrix. (f) Second vector.

map will map the boundary into the interior of the output data, so cannot be used to extend the coordinate system. Discarding long edges will not work either because the boundary edges will not be changed in this example. Section 4 will propose a possible method that might solve this type of problem.

3. FINDING THE MANIFOLD BOUNDARY

We assume that we have already determined the dimension, d of the manifold, and that the density of points is large enough so that we can compute a reasonable approximation to a tangent plane to the manifold at any given point by taking the first d components of a PCA of all neighbors within a given distance that must be larger enough compared to

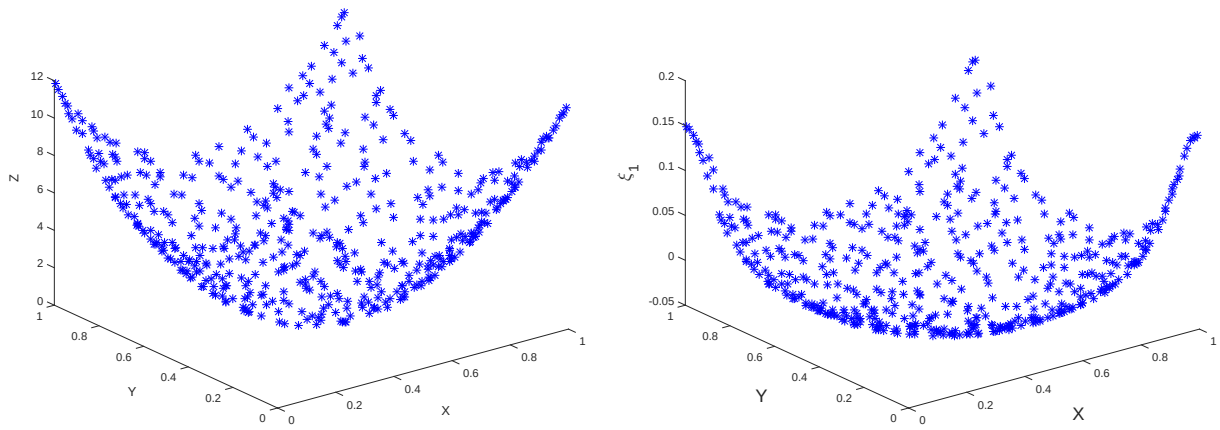

FIGURE 16. Paraboloid with greater height (left). Its first diffusion coordinate (right).

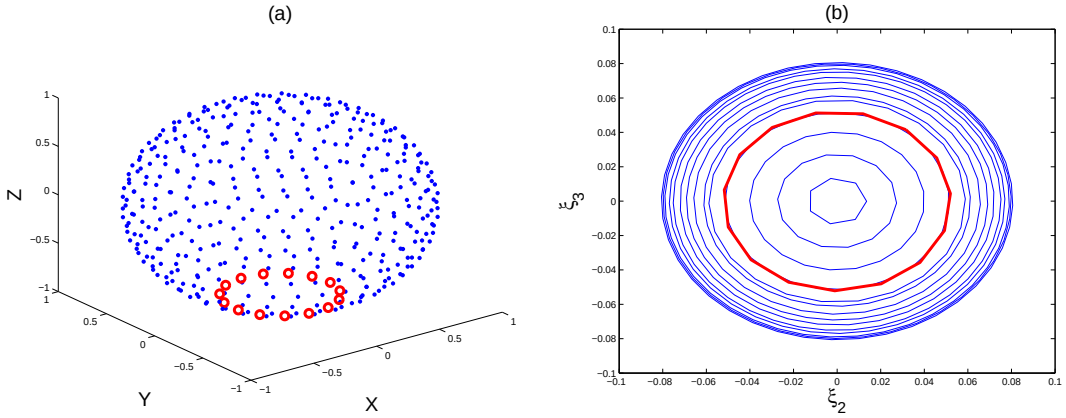

FIGURE 17. The truncated surface of a sphere - everything south of latitude 67.5^oS has been removed and the boundary points are indicated with circles. The diffusion map contours of the lines of constant latitude every 10.5^o. The thick line is the map of the boundary.

the noise level and small enough compared to the local curvature. Then we can work locally in the linear PCA space.

We first assign to every point a *boundary propensity value* (BPV) which is a real number that if positive means that we accept it as a boundary point, but the more negative it is, the less likely we are to accept it as a boundary point. We will give two methods for assigning this value below (the first is faster and probably less reliable).

For many applications we need a reasonable density of boundary points. After assigning a BPV to each point we mark all points with a positive value as boundary points. (If there are none, we declare the point with the largest BPV to be in the boundary set. Then, for each point in the boundary set we compute the fraction of near neighbors that are boundary points. The acceptable fraction depends on the manifold dimension and the desired density of points on the boundary. For example, if we had a 3D manifold, a sphere of radius r around a boundary point would roughly include a hemisphere of the manifold, or a volume of $2\pi r^3/3$. The surface would roughly include a disk and if it had thickness b its volume is $2b\pi r^2$, so if the points were uniformly dense in the interior and the boundary, we would expect a fraction of about $3b/r$ to be boundary points. If the region has fewer, we add those points with the largest BPV.

3.1. Assigning the BVP

In both methods we find a set of nearest neighbors of the point under consideration, P, and compute the best dD PCA approximation to the set of neighbors and P. (If one has an *a priori* estimate of the noise in the data then one should use a radius sufficiently larger than the noise level to get an acceptable PCA. However, there have to be a sufficient number of points in the neighborhood. (It seems likely that one could use information from the residual of the PCA to estimate the effect of noise and curvature and information from the PCA's of neighboring points to estimate effects of the curvature, but we have not examined that yet.) From then on, both methods work with the projection of the points in the PCA space.

Method 1 The mass center, M, of the neighbors of P is computed, and the hyperplane through P orthogonal to the vector from M to P "constructed" (there is no need to formally construct this except for explanatory purposes). The side of the hyperplane that does not contain M is considered to be the *outside* for P. (If P and M are so close, relative to the extent of the neighborhood, a maximum negative BPV is assigned.) Next we compute the signed distances of all neighbors from the hyperplane using negative values for those on the outside. The BPV assigned to P is the minimum of these signed distances. If it is positive, there are no points on the outside, and if it is negative, the BPV is the worst case distance on the outside. This computation is straightforward. After computing M we compute the vector to P, find its length, L, and normalize it to get a vector $\mathbf{v}$. Now, for each point, Q, in the neighborhood we form the vector $\mathbf{v}_q$ from M to Q and the BPV is given by

$$\text{BPV} = \max_q(L - \mathbf{v}^T\mathbf{v}_q)$$

Method 2 The vector, $\mathbf{v}$, from P is computed that has the largest β such that a cone of rotation with angle β does not contain any neighbors of P. The BPV is $\beta - \pi/2$.

Figure 18 illustrates these two methods.

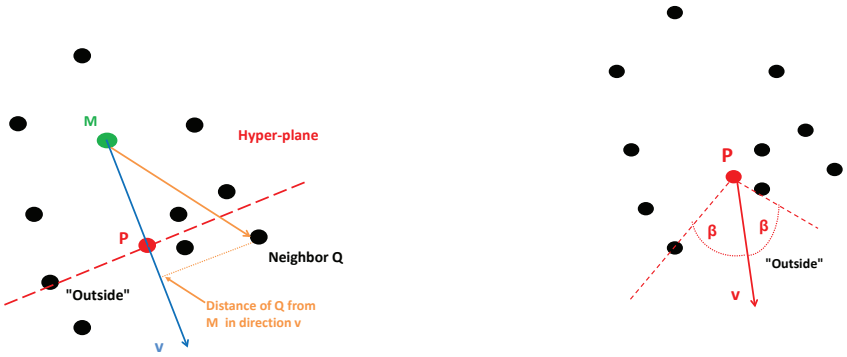

FIGURE 18. Computing the Boundary Propensity Value: Method 1 (left), Method 2 (right).

4. A POSSIBLE METHOD FOR PARAMETERIZATION

As mentioned earlier, neither of the parameterization approaches discussed will handle the type of manifold shown in Figure 17. The difficulty with this problem is that the boundary is composed of fairly closely-spaced points and this forces the diffusion map method to make them relatively close in diffusion map space.

We have been exploring another approach that would address more general problems such as this truncated spherical surface. Suppose we were able to replace the manifold with a uniform resistor of an appropriate dimension (a solid block for a 3D manifold). If a voltage is applied between two points on the manifold boundary it satisfies the Laplace equation, and we could, in principle, compute the solution. Unfortunately the Laplace equation has a no-flux condition at the boundary, so we propose to replace the boundary with a one lower dimensional manifold. This has no boundary (unless the original manifold was 1D, in which case the problem is trivial using this technique). We first find the solution on just the boundary, and then use the resulting voltage values as input to the full manifold.

Unfortunately, we don't know enough about the manifold to replace it with a resistive medium. All we have initially is a set of points close to the manifold. At first we explored replacing the edges in the graph with suitable resistors to approximate a uniform resistive medium, but this does not appear to be possible (or if it is, to be computationally complex). Suppose we had a 6 node graph with resistors R_{ij} between nodes i and j, $0 \leq i, j \leq 5$ and nodes 0 and 5 are pegged to voltages V_0 and V_5 respectively. Then we get a linear equation for the voltages at the other nodes as follows (where we use the conductances $C_{ij} = 1/R_{ij}$ and assume that $C_{ii} = 0$

$$\begin{bmatrix} -\sum_{j=0}^{5} C_{1j} & C_{12} & C_{13} & C_{14} \\ C_{21} & -\sum_{j=0}^{5} C_{2j} & C_{23} & C_{24} \\ C_{31} & C_{32} & -\sum_{j=0}^{5} C_{3j} & C_{34} \\ C_{41} & C_{42} & C_{43} & -\sum_{j=0}^{5} C_{4j} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = -V_0 \begin{bmatrix} C_{10} \\ C_{20} \\ C_{30} \\ C_{40} \end{bmatrix} - V_5 \begin{bmatrix} C_{15} \\ C_{25} \\ C_{35} \\ C_{45} \end{bmatrix} \tag{3}$$

Finding the values of the C_{ij} that would make this equation satisfy the Laplace-Beltrami equation on the manifold seems like an impossible task since we have little information about the manifold until we parameterize it. Instead for each point on the manifold we construct the dD PCA for that point and a set of nearest neighbors and try to choose C_{ij} such that the Laplace equation is satisfied in the linear PCA space. Even that is challenging (and may have no solution). Instead we note that since our objective it to create a set of linear equations that provide an approximation to the Laplace equation for each node in its linear PCA space, it is not necessary that the matrix as in eq. (3) be symmetric since we do not need to be able to physically implement the circuit. To explain the technique suppose that we have a 2D manifold and that the PCA coordinates for the PCA of point P_0 and its q nearest neighbors, $P_i, i = 1, \cdots, q$ are (x_i, y_i) where we have translated the origin so that P_0 is at (0, 0). If the voltage at P_i is v_i then we have

$$v_i = v_0 + v_x x_i + v_y y_i + v_{xy} x_i y_i + v_{xx} x_i^2/2 + v_{yy} y_i^2 + \mathrm{O}(r_i^3).$$

If we chose a set of coefficients $\{C_{0i}\}$ so that

$$\sum_{j=1}^{q} C_{0j} v_j - v_0 \sum_{j=1}^{q} C_{0j} = v_{xx} + v_{yy} \tag{4}$$

then the voltage will satisfy the Laplace equation on the linear manifold to first order accuracy. For a 2D manifold this requires q to be at least 5, and for d dimensions, $q \geq d(d+3)/2$. This prescription can lead to negative conductances (meaning that the matrix is no longer diagonally dominant which can lead to numerical errors) so if we fail to get positive conductances with the q first used, we try increasing it and re-solve. After some maximum number of points has been tried without finding positive conductances, we currently accept the last attempt.

There are some arrangements of points for which it is impossible to get positive conductances. For example, in 1D, if we only have points on one side of the point, P_0 under consideration, it is clear that there is no way to approximate the second derivative with positive conductances. To reduce the probability of this situation, we only find the Laplace approximants for the interior points, using the boundary points in the equations if they are among the nearest neighbors.

Before we compute the Laplace equations for the interior points, the boundary is handled and the points adjusted for smoothness. Think of a 2D manifold, so the boundary is 1D. Since the points are randomly placed, the boundary points are likely to form a "rough" edge to the manifold. If we solved a Laplace equation on these points (actually on a local PCA space for them) we might compute a smooth voltage but because some points are much closer to the interiors points than others, the roughness of the boundary would be reflected in the calculated voltage in the interior near the boundary. For each boundary point we currently compute a PCA with a set of nearest neighbor boundary points and then project its new position onto the PCA of a set of nearest neighbor interior points. We take this projection as the new position of the boundary point. (It may now be off the manifold, but if the point density is large, it wont be very far off.) Then we compute the conductance equations for the points

The computation steps are as follows:

1. Determine whether each point is a boundary or an interior point.
2. Assume each boundary point lies on a manifold of dimension 1 less. Adjust the position of the boundary point as described in the previous paragraph.
3. Compute the conductance equations of each boundary point to get linear equation set L1.
4. For all interior points compute the coefficients C_{ij} for it to satisfy the Laplace equation on the PCA manifold (using boundary points if they are in the nearest neighbor set) to get a linear equation set L2.

5. Form the subgraph, G', consisting of only the boundary points.
6. Remove all edges from G' longer than a small multiple of the cut value of G' and the compute the shortest paths between each point to get a new graph G^{sp}.
7. Pick an arbitrary point in G^{sp}, say P_a.
8. Set the counter $C = 1$
9. Find the furthest node from P_a in G^{sp} and call it P_f
10. Apply voltages +1 to P_a and -1 to P_f and solve linear equations L1.
11. Apply the voltages found for the boundary points in the previous step to the linear equation set L2 and compute the node voltages. This is the C-th parameterization.
12. Find the boundary node with voltages from all previous parameterizations closest to zero and call this node P_a.
13. Increment C and if $C < d$ return to step 9.

In the potential method, the voltage drops very rapidly in the regions around the "anode" and "cathode," leading to the central region being restricted to a small potential range. This is increasingly so in higher-dimensional manifolds. In one way it is the opposite of the boundary effect in the diffusion map that gave little change at the boundary but it can be easily changed by a suitable smooth transformation. In the example below we took the cosine of the voltage scaled to lie in $[0, \pi]$. We applied this method to a manifold similar to the one in Figure 17. That one was generated so that there was a reasonable distribution of points with all points being on a set of 15 discrete latitude lines (so we could easily plot the maps of constant latitude).

In this example, we used the surface of a unit sphere cut below a latitude of 45^oS and generated random points on it. All points placed on the sphere were subjected to an additive Gaussian noise, N(0,0.01), radially. A maximum of N points were placed. The first point was placed at the North pole and all subsequent points were generated uniformly randomly and added if they were not within 0.0655 (this is $\sqrt{Area/N}$) of a previously added point and the process stopped after a maximum of $6N$ attempts to place points. The number of points placed depends on the added noise, and in this example 1,457 points were placed. The raw data is shown in Figure 19. Interior points are black, points initially identified as boundary are green. Blue circles indicate the new positions of the boundary points after smoothing.

Figure 20 left shows the log-log plot of edge lengths indicating that the dimension is 2. Figure 20 right shows the plot of lines of constant latitude in the V1-V2 plane (actually, in the cosine transformation). This was done using linear interpolation, so its non-smoothness may be dues to that. Figure 21 shows the points colored by the cosines of the scaled voltages and the contours of constant voltages.

ACKNOWLEDGMENTS

Any acknowledgements??

REFERENCES

1. F. Camastra, "Data dimensionality estimation mehods: a survey," *Pattern Recognition*, **36**, 2003, pp 2945-2954.
2. R. R. Coifman and S. Lafon, "Diffusion maps," *Appl. and Comp. Harmonic Analysis*, **21**, 2000, pp. 5–30.
3. C. W. Gear, "Parameterization of non-linear manifolds," *physics arXiv* arXiv:1208.5246
4. A. V. Little, Y-M. Jung and M. Maggioni, "Multiscale Estimation of Intrinsic Dimensionality of Data Sets," *Proc AAAI*, 2009. pp 26-33. Available here: http://www.aaai.org/ocs/index.php/FSS/FSS09/paper/viewFile/950/1218

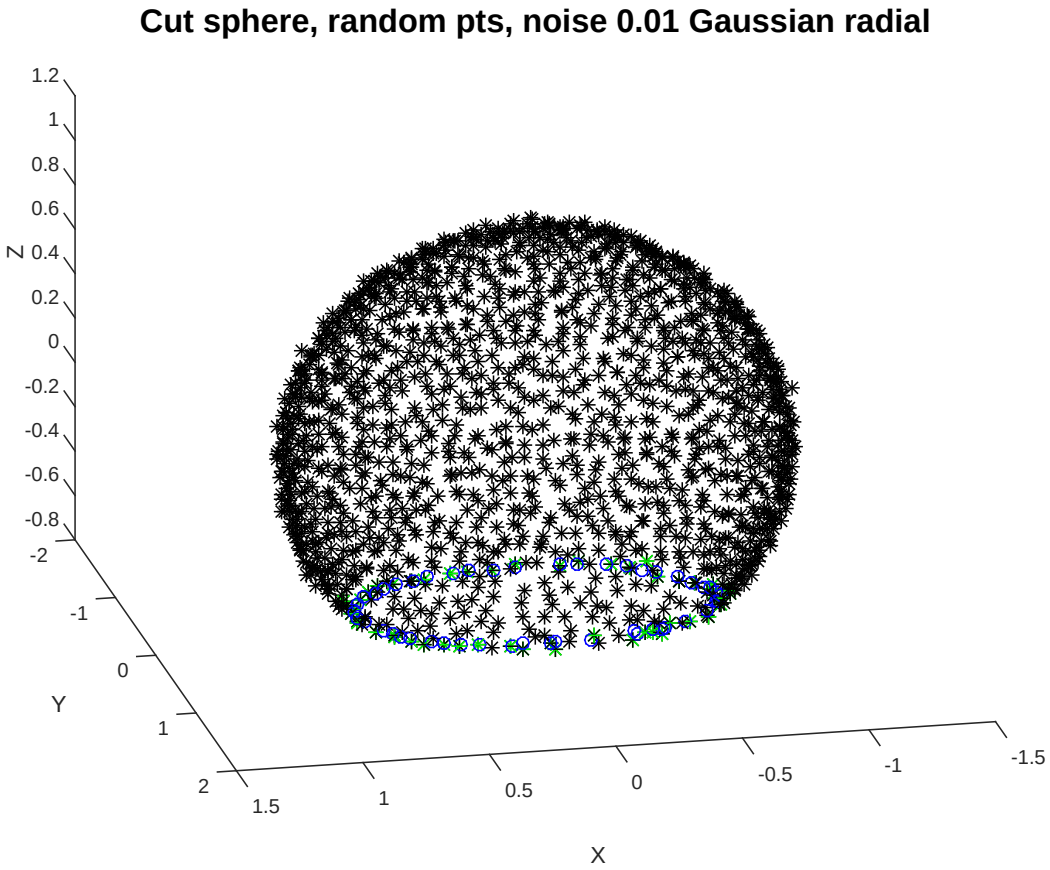

FIGURE 19. Raw data for on cut sphere. Black points have been identified as interior, Green points as boundary. Blue circles are adjusted boundary points (see text).

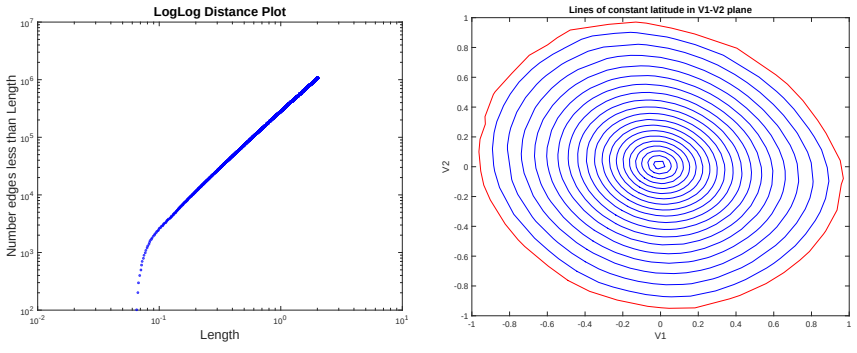

FIGURE 20. (Left): log-log plot of edge lengths. (Right): Plot of lines of constant latitude.

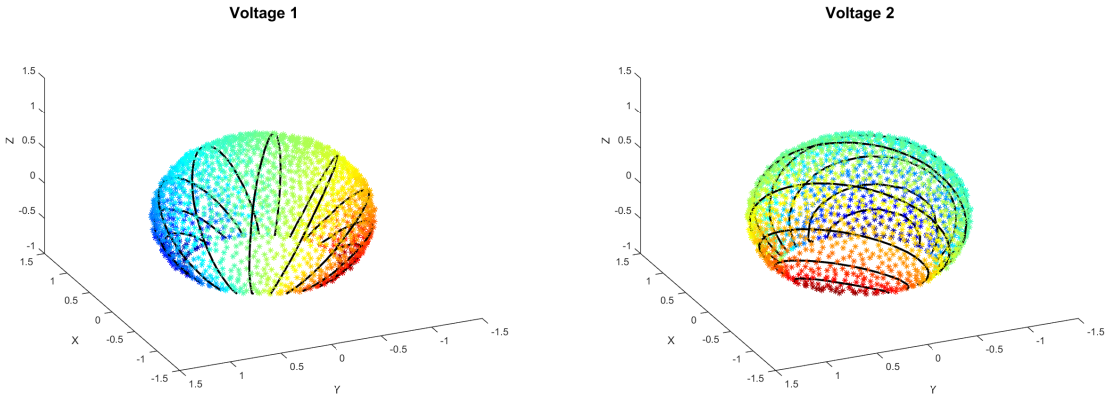

FIGURE 21. The two voltages assigned to each point and their constant contour lines.