Journal of Statistical Distributions and Applications

METHODOLOGY Open Access

An R package for modeling and simulating generalized spherical and related distributions

John P. Nolan

Correspondence:
jpnolan@american.edu
Department of Mathematics and Statistics, American University, Washington, DC, USA

Abstract

A flexible class of multivariate generalized spherical distributions with star-shaped level sets is developed. To work in dimension above two requires tools from computational geometry and multivariate numerical integration. An algorithm to approximately simulate from these star-shaped distributions is developed; it also works for simulating from more general tessellations. These techniques are implemented in the R package `gensphere`.

Mathematics Subject Classification (2000): 62H11, 68U05, 65D30

Keywords: Generalized spherical distributions, Star-shaped distributions, Computational geometry

1 Introduction

There is a need for tractable models for multivariate data with nonstandard dependence structures. Our motivation here was to be able to flexibly model distributions with star-shaped level sets. The R package `gensphere` has been developed that allows one to work with these classes of distributions: specifying flexible shapes for the level sets, computing densities, and simulating. A deliberate goal in this process is to have methods and programs that work in dimension $d \geq 2$, and this requires some methods from computational geometry. While the original intent focused on star-shaped regions, some of the tools developed here are useful for other problems, e.g. sampling from more general sets.

Fernández et al. (1995) proposed defining multivariate distributions for which the level sets are scaled versions of a contour $\mathcal{C}$ (a simple closed curve/surface in $\mathbb{R}^d$). We will specify a contour by a function $c : \mathbb{S} \to [0, \infty)$:

$$\mathcal{C} = \{c(\mathbf{s})\, \mathbf{s} \; : \; \mathbf{s} \in \mathbb{S}\}.$$

Here $\mathbb{S} = \left\{\mathbf{s} \in \mathbb{R}^d : |\mathbf{s}| = 1\right\}$ is the unit sphere in the Euclidean norm $| \cdot |$, a $(d-1)$-dimensional surface. We assume throughout that $c(\mathbf{s})$ is a piecewise continuous function, so measurability issues are automatically satisfied. Figure 1 shows a 2-dimensional example and Fig. 4 shows a 3-dimensional example of such contours.

A motivating example for this work is to model fragment dispersion from an explosion. In such problems, the fragments disperse in three dimensions in patterns like those of Fig. 4. The ability to easily specify different contour functions by adding together multiple terms as in Section 2.1 is of practical importance for describing different types of explosive

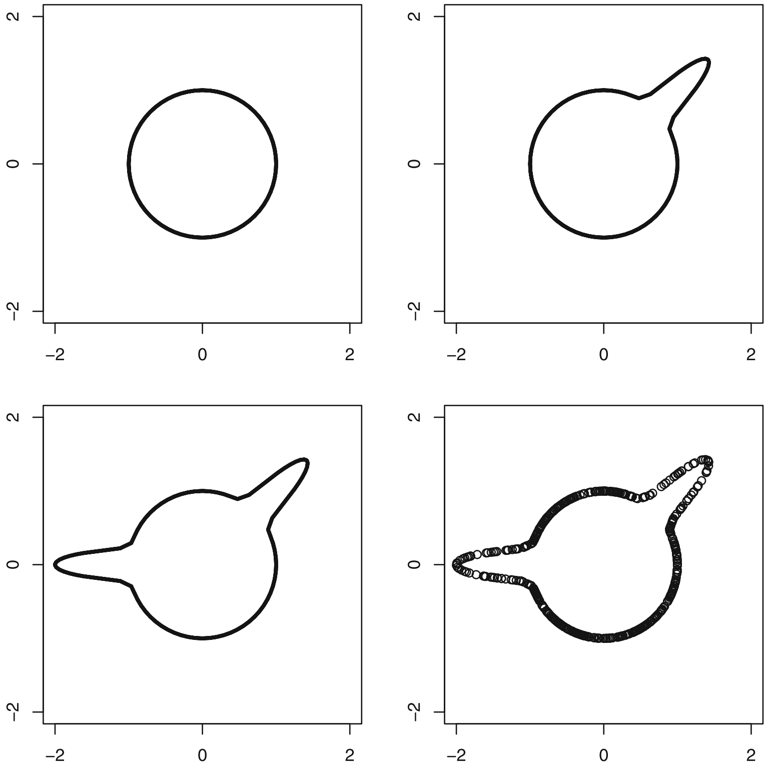

Fig. 1 Constructing a 2-dimensional contour. The *top left* plot shows a base of type 1, a circle of radius 1. The top right shows the base with one Gaussian bump of type 3 in direction $\left(\sqrt{2}/2, \sqrt{2}/2\right)$, the bottom left shows the final contour with another Gaussian bump in direction (-1,0). The *bottom right* plot shows a sample of size $n = 1000$ from this contour using the method described in Section 3

devices. The goal of this modeling is to design better body and vehicle armor to protect people.

Let $g : [0, \infty) \to [0, \infty)$ be a nonnegative function and define

$$f(\mathbf{x}) = \begin{cases} g\left(\frac{|\mathbf{x}|}{c(\mathbf{x}/|\mathbf{x}|)}\right) & |\mathbf{x}| > 0 \\ g(0) & |\mathbf{x}| = 0 \end{cases} \tag{1}$$

Under integrability conditions discussed below, this will give a probability density function on $\mathbb{R}^d$, and the level sets of such a distribution are scalar multiples of $\mathcal{C}$. Such distributions are also called *homothetic*, see Balkema and Nolde (2010), Section 3.1 or Simon and Blume (1994), Section 20.4. We will call $c(\cdot)$ the *contour function* and $g(\cdot)$ the *radial decay function* of the distribution.

Our approach differs from Fernández et al. (1995) where they start with a function $\nu : \mathbb{R}^d \to [0, \infty)$ that is homogeneous: $\nu(a\mathbf{x}) = |a|\nu(\mathbf{x})$. Such functions are called *gauge functions* or *Minkowski functionals*, and are well studied in convex analysis and functional analysis. The relationship between their ν function and our contour function is $\nu(\mathbf{x}) = |\mathbf{x}|/c(\mathbf{x}/|\mathbf{x}|)$. If $c(\mathbf{s}) = 1$, then $\mathcal{C}$ is the unit sphere and $\nu(\mathbf{x}) = |\mathbf{x}|$, so the resulting classes of distributions are the spherical/isotropic distributions. If $\nu(\cdot)$ is convex, then $\nu(\cdot)$ is a norm on $\mathbb{R}^d$ and $\mathcal{C}$ is the unit sphere in that norm, hence the name ν-spherical distributions. When $\nu(\cdot)$ is not convex, e.g. the ℓ_p quasi-norm with $p < 1$, $\nu(\mathbf{x})$ does not give a norm, so $\mathcal{C}$ is not strictly speaking a unit sphere, but we will still call the resulting distributions ν-spherical.

The purpose of this paper is to describe a method of defining a flexible class of generalized spherical distributions in any dimension $d \geq 2$, and to describe an R package `gensphere` that implements this method. The package gives the ability to

1. Define a flexible set of contours
2. Carefully tessellate a contour
3. Sample from a tessellation
4. Use a contour and a radial function $g(\cdot)$ to define a generalized spherical distribution
5. Compute the density $f(\cdot)$ given by (1)
6. Approximately simulate from a distribution with density $f(\cdot)$

The third step above also provides a way to simulate from paths and surfaces unrelated to generalized spherical laws, giving new classes of probability distributions on paths and surfaces.

Other references on generalized spherical laws are Arnold et al. (2008), Kamiya et al. (2008), Rattihalli and Basugade (2009), Rattihalli and Patil (2010), and Balkema and Nolde (2010). These papers develop the idea of generalized spherical distributions, but do not provide general purpose software for working with these distributions and do not cover techniques for working with higher dimensional models. Richter (2014) gives a rigorous investigation of p-generalized elliptically contoured distributions, with a detailed analysis of the surface measure and a polar disintegration of the laws.

2 Generalized spherical distributions

For (1) to be a proper density, it is required that (see equations (4) and (5) of Fernández et al. (1995))

$$k_{\mathcal{C}}^{-1} := \int_{\mathbb{S}} c^d(\mathbf{s}) d\mathbf{s} \in (0, \infty) \tag{2}$$

and

$$\int_0^\infty r^{d-1} g(r) dr = k_{\mathcal{C}}. \tag{3}$$

We will assume $c(\cdot)$ is continuous on $\mathbb{S}$ and that $c(\mathbf{s}) \leq c_0$. This guarantees (2) is finite, though evaluating it may be difficult, especially when $d > 2$. Section 3 discusses an approach to this problem that improves the accuracy of this computation for the types of contours considered here. Given any univariate probability density $h(\cdot)$ on the positive axis, the function $g(r) = k_{\mathcal{C}} r^{1-d} h(r)$ is a valid radial decay function. This is the approach used in the rest of this paper and in the associated package.

To simulate values for a generalized spherical random vector, we are interested in a stochastic representation of the form

$$\mathbf{X} \overset{d}{=} R\mathbf{Z}. \tag{4}$$

Choosing $\mathbf{Z}$ uniformly distributed (proportional to surface area) on the contour does not work in general. Richter (2014) shows this works in special circumstances, e.g. if the contour $\mathcal{C}$ is an ℓ_2 ball, ℓ_1 ball, or ℓ_∞ ball. In Section 3 we develop a way to approximately simulate a wider class of distributions by using a piecewise linear approach: approximate the contour $\mathcal{C}$ by a simplicial tessellation and use (4) on each piece.

2.1 Specification of a contour function

For modeling purposes, we want a flexible family of functions that can be used in a variety of problems.

To be able to include the distributions discussed by the authors cited above, we allow contour functions of the form

$$c(\mathbf{s}) = \sum_{j=1}^{N_1} c_j r_j(\mathbf{s}) + \frac{1}{\sum_{j=1}^{N_2} c_j^* r_j^*(\mathbf{s})},$$

where $c_j > 0$, $c_j^* > 0$, and $r_j(\cdot)$ and $r^*(\cdot)$ are one of the cases discussed below. N_1 and N_2 are non-negative integers telling how many terms of each type are used.

1. $r(\mathbf{s}) = 1$, which makes $\mathcal{C}$ the Euclidean ball. Any isotropic/radially symmetric distribution can be modeled by using just this term in a contour function and the appropriate radial decay function.
2. $r(\mathbf{s}) = c(\mathbf{s}|\boldsymbol{\mu}, \theta)$ is a cone with peak 1 at center $\boldsymbol{\mu} \in \mathbb{S}$ and height 0 at the base given by the circle $\{\mathbf{x} \in \mathbb{S} : \boldsymbol{\mu} \cdot \mathbf{x} = \cos\theta\}$. It is assumed that $|\theta| \leq \pi/2$.
3. $r(\mathbf{s}) = c(\mathbf{s}|\boldsymbol{\mu}, \sigma) = \exp(-t(\mathbf{s})^2/(2\sigma^2))$ is a Gaussian bump centered at location $\boldsymbol{\mu} \in \mathbb{S}$ and "standard deviation" $\sigma > 0$. Here $t(\mathbf{s})$ is the distance between $\boldsymbol{\mu}$ and the projection of $\mathbf{s} \in \mathbb{S}$ linearly onto the plane tangent to $\mathbb{S}$ at $\boldsymbol{\mu}$.
4. $r^*(\mathbf{s}) = ||\mathbf{s}||_{\ell^p(\mathbb{R}^d)}, p > 0$.
5. $r^*(\mathbf{s}) = ||A\mathbf{s}||_{\ell^p(\mathbb{R}^m)}, p > 0$, A an $(m \times d)$ matrix. This allows a generalized p-norm. If A is $d \times d$ and orthogonal, then the resulting contour will be a rotation of the standard unit ball in ℓ^p. If A is $d \times d$ and not orthogonal, then the contour will be sheared. If $m > d$, it will give the ℓ^p norm on $\mathbb{R}^m$ of $A\mathbf{s}$.
6. $r^*(\mathbf{s}) = (\mathbf{s}^\top A\mathbf{s})^{1/2}$, where A is a positive definite $(d \times d)$ matrix. Then the level curves of the distribution are ellipses. Any elliptically contoured distribution can be modeled by using just this term in a contour function and the appropriate radial decay function.

Sums of the first three types allow us to describe star-shaped contours, see Fig. 1. Inverses of sums of the last three types allow us to consider contours that are familiar unit balls, or generalized unit balls, or sums of such shapes. Specifying a radial decay function $g(\cdot)$ defines a density $f(\mathbf{x})$ by (1) as in Fig. 2. An implementation of this construction is given in the R package `gensphere`. The R statements used in this example are given in the Appendix.

It is relatively easy to add new types of terms to this list if other contours are of interest. However this set of basic shapes can model a wide range of shapes, including contours supported on a cone. Figure 3 shows nine examples. The top row shows ℓ_p balls with $p = 1/2$, $p = 1$, and $p = 5$. The middle row starts with a contour made up of an ℓ_p ball with a $p = 0.3$ and a copy of that rotated by $\pi/4$, the rotation done by using a generalized ℓ_p norm with A a rotation matrix. The next two plots show generalized ℓ_p balls with $A = (1, 1; 1, -4; 1, 3; 5, -3)$ and $p = 1/2$ (middle) and $p = 1.1$ (right). The last row shows contours supported on a cone. The left plot is the sum of three Gaussian bumps of type 3, each centered at $(\cos\theta, \sin\theta)$, $\theta = \pi/4, \pi/2, 3\pi/2$ and $\sigma = 0.3$. The middle plot has two type 2 cones, at angles $-\pi/6$ and $-\pi/3$ with $\sigma = 0.4$. The last graph also has two cones, centered at $\pi/6$ and $\pi/3$, with $\sigma = 0.25$. Any of the contours that have a corner or cusp on a ray will generate a density surface with a ridge along that ray. A more complicated three

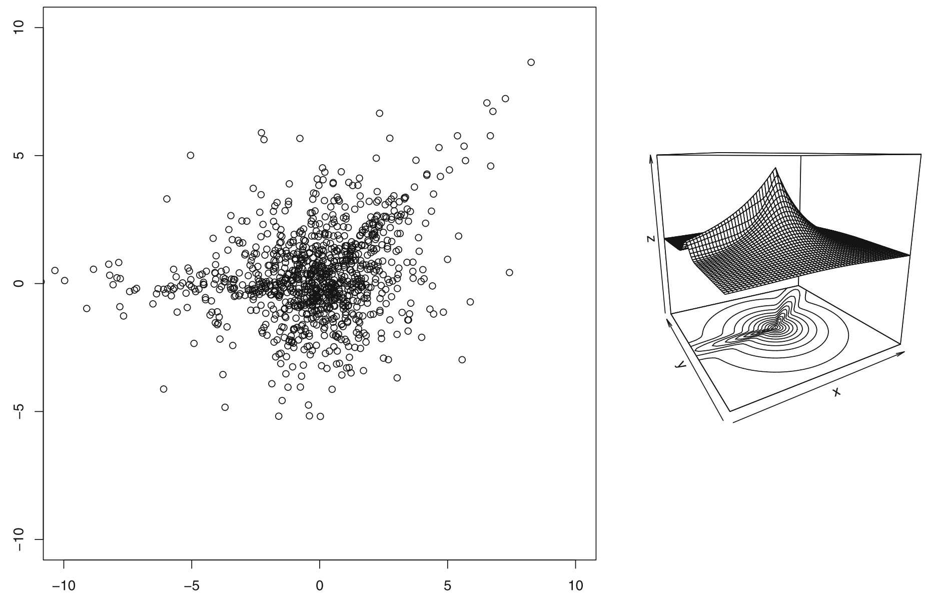

Fig. 2 On the *left* is a sample of size $n = 1000$ from the generalized spherical distribution based on the contour in Fig. 1 and a $\Gamma(2,1)$ radial term. On the *right*, is a surface plot of the corresponding bivariate density $f(x, y)$

dimensional example with 11 terms in the definition of $c(\cdot)$ is given in Fig. 4: an elliptical base of type 6 and 10 cones of type 2.

2.2 Choice of *R*

In general, $g(r)$ can be any nonnegative integrable function. The radial decay of R determines the decay of $f(\cdot)$ on $\mathbb{R}^d$. In most applications one wants $0 < g(0) < \infty$ and $g(r)$

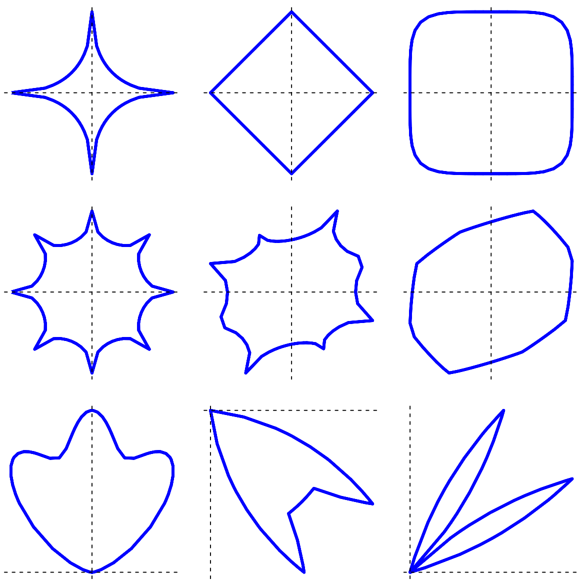

Fig. 3 A selection of contours made from the different types of terms. See the text for a description

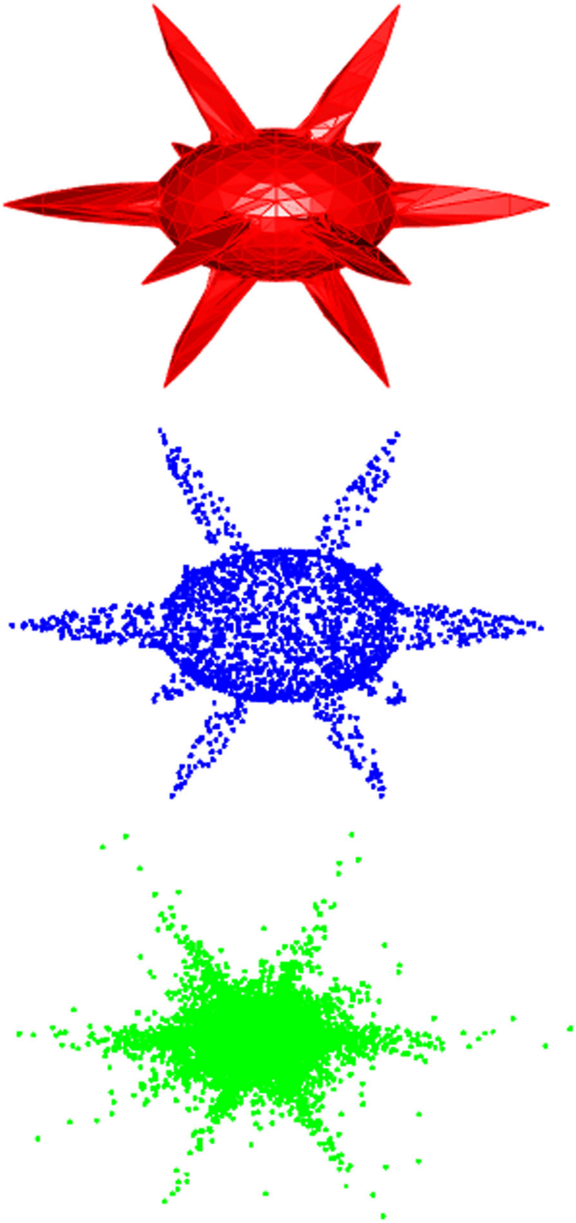

Fig. 4 A 3D star-shaped region with one term of type 6 and 10 terms of type 2. The *top* plot shows the contour, the *middle* shows a sample of size 2500 from the contour, the *bottom* shows a sample of size 10000 from the generalized spherical distribution given by this contour and a $\Gamma(3, 1)$ radial term R

decreasing for $r > 0$, but other possibilities may be of interest. If $g(0) = 0$, the density surface given by (1) will have a "well" at the origin; if $g(0) = +\infty$, then the density blows up at the origin. If $g(\cdot)$ oscillates, then the density surface will have radial "waves" emanating out from the origin. If R has bounded support, then $\mathbf{X}$ will have bounded support.

The gamma distributions give a family of distributions that can be used to get generalized spherical distributions with light tails. If a $\Gamma(d, 1)$ law is used for R, then $h(r) = \Gamma(d)^{-1}r^{d-1}\exp(-r)$, so $g(r) = k_{\mathcal{C}}r^{1-d}h(r) = (k_{\mathcal{C}}/\Gamma(d))\exp(-r)$, which is finite at the origin and monotonically decreasing. If one wants heavy tails for $\mathbf{X}$, then some possibilities for R are Fréchet, Pareto and multivariate stable amplitude. (The latter is defined in Nolan (2013) by $R = |\mathbf{Z}|$, where $\mathbf{Z}$ is radially symmetric/isotropic α-stable in d-dimensions. Numerical methods to calculate the density $h(r)$ of R and simple ways to simulate are given in the reference).

Figure 5 shows the effect that the choice of R has. In all cases, the base contour is the unit ball in ℓ_1, a diamond shape. At the upper left, R is a uniform r.v. on (0, 1). In this case, $g(0) = +\infty$ and the density has a spike at the origin and bounded support on the diamond. At the top right, $R \sim \Gamma(2, 1)$, so $g(0) = 1$ and the distribution has unbounded support with light tails. At the lower left, R is the $\alpha = 1$ stable amplitude in $d = 2$ dimensions; here $g(0)$ is finite and the distribution has heavy tails. The bottom right plot is with $R \sim \Gamma(5, 1)$, so $g(0) = 0$ and the distribution has a well at the origin and unbounded support with light tails.

3 Contours: tessellating, integrating and simulating

A large part of the technical complexity of working with generalized spherical laws is in representing the contours and evaluating the norming constant k_C in (2) and simulating

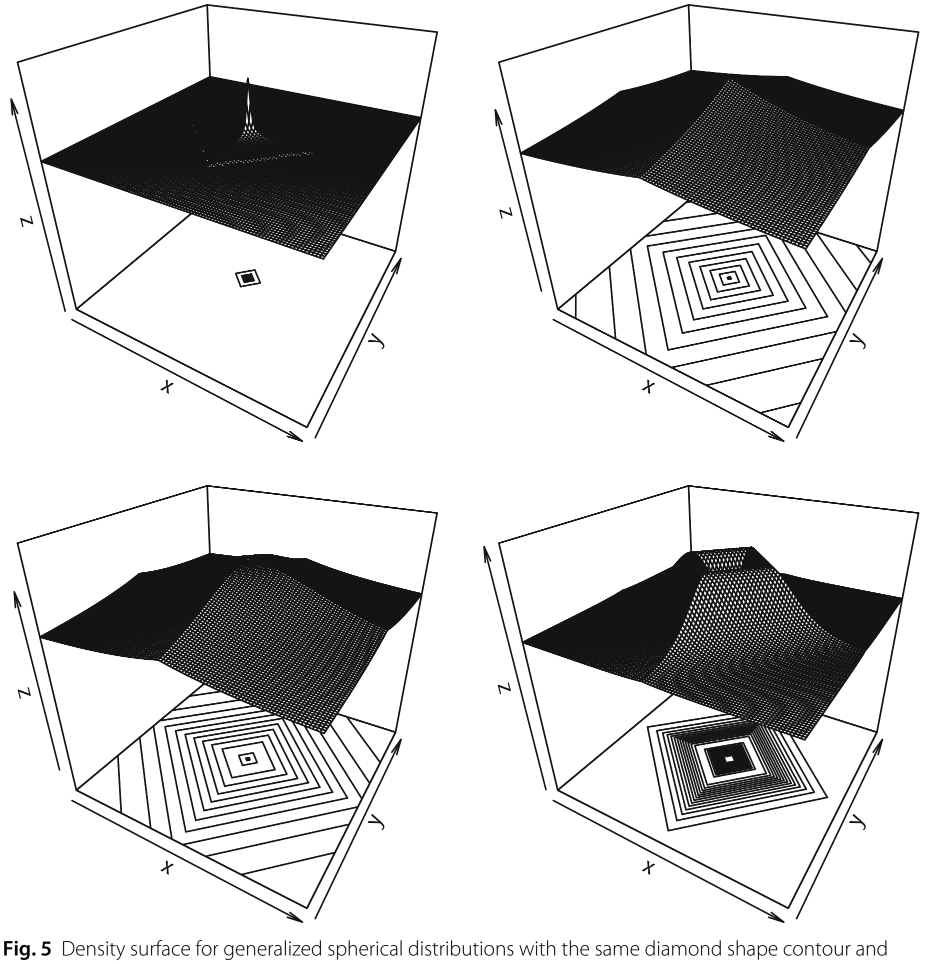

Fig. 5 Density surface for generalized spherical distributions with the same diamond shape contour and different radial term R as specified in the text

from the contour $\mathcal{C}$. The gensphere package uses two other recent R packages for these problems: SphericalCubature Nolan (2015b) and mvmesh Nolan (2015a).

SphericalCubature numerically integrates a function on a d-dimensional sphere. Given a tessellation of the sphere in $\mathbb{R}^d$, it uses adaptive integration to integrate over the $(d-1)$-dimensional surface to evaluate $k_{\mathcal{C}}$. If the integrand function is smooth and the tessellation is reasonable, then the numerical integration is accurate in modest dimensions, say $d = 2, 3, 4, 5, 6$. However, when the integrand function has abrupt changes, numerical techniques can miss parts of the integral. This is even a problem in dimension 2, where the integration is a one dimensional problem. One way to deal with this is to work with tessellations that focus on the places where the integrand is not smooth. In complete generality, this is hard to do. However, in evaluating integral (2) for one of the contours described above, we have an implicit description of where the contour changes abruptly.

The mvmesh package is used to define multivariate meshes, e.g. a collection of vertices and grouping information that specify a list of simplices that approximate a contour. The first place where mvmesh is used in gensphere is to give a grid on the sphere $\mathbb{S}$ in d-dimensions, e.g. the top left plot in Fig. 1. mvmesh has a function UnitSphere that computes an approximately equal surface area approximation to a hypersphere in dimension d. It takes a parameter k to say how many recursive subdivisions are used in each octant; increasing this value will give a finer tessellation of the sphere. Then this tessellation is refined by adding points to the sphere centered on the places where the contour has bumps, e.g. the cone and Gaussian bumps (type 2 and 3). Then the new points are combined with the original tessellation of the sphere to get a refined tessellation of the sphere that includes these key points.

It is at this point that the SphericalCubature package is used to evaluate the integral (2). This is difficult to accurately evaluate in dimension greater than three if the contour is not smooth. In addition to the estimate of the integral, we use an option in the adaptive integration routine to return the partition used in the multivariate cubature, along with the estimated integral over each simplex. The reasoning is that the integration routine is subdividing regions where the integrand is changing quickly to get a better estimate of the integrand. This subdivision should make the tessellation more closely approximate the contour. We now have the final tessellation of the unit sphere, an estimate of the integral (2) over each of the simplices, and an estimate of the norming constant, e.g. sum of these just mentioned values.

Now the tessellation of the contour is defined by deforming the tessellation of the sphere to the contour: each partition point $\mathbf{s} \in \mathbb{S}$ gets mapped to $c(\mathbf{s})\mathbf{s}$ on the contour. The grouping information from the spherical tessellation is inherited by the contour tessellation. This tessellation is returned as an S3 object of class “mvmesh”. This object contains the vertices, the grouping information, and a list of all the simplices $S_1, S_2, \ldots, S_k$ in the tessellation. One advantage of this is that the plot method from the mvmesh package can plot the contours in 2 and 3 dimensions. This process of refining the tessellation has two purposes: (a) get a more accurate estimate of the norming constant by focusing the numerical integration routine on regions where the integrand changes rapidly and (b) get a more accurate tessellation of the contour. Each step of this process can add more simplices, with the goal of capturing key features of the contour. For example, the contour in Fig. 4 started with 512 simplices in the tessellation of the sphere in $\mathbb{R}^3$ with $k = 3$, adding the points on the cones brought the number up to 888 simplices, and after the adaptive cubature routine subdivision there were 2284 simplices.

Exact simulation from a surface is a challenging problem and general methods are difficult to apply for complicated contours like our star-shaped regions. We now describe an approximate method based on the above tessellation. Recall that the above process gives us a list of simplices $S_1, \ldots, S_m$ and associated weights $w_1, \ldots, w_m$, with w_j an estimate of the surface area of the contour approximated by simplex S_j.

The simulation routine to sample from the tessellation is straightforward:

1. Select an index $j \in \{1, \ldots, m\}$ with probability proportional to w_j.
2. Simulate a point $\mathbf{u}$ that is uniformly distributed on the unit simplex in d-dimensions. This is standard: simulate $\mathbf{u}$ from a Dirichlet distribution with parameter $\boldsymbol{\alpha} = (1, 1, \ldots, 1)$, e.g. let $E_1, \ldots, E_d$ i.i.d. standard exponential random variates and set $\mathbf{u} = (E_1, \ldots, E_d)/\left(\sum_{i=1}^d E_i\right)$.
3. Map the point $\mathbf{u}$ to the simplex S_j using the coordinates of $\mathbf{u}$ as barycentric coordinates: $\mathbf{Z} = \mathbf{u}^\top S_j$.
4. Simulate R from the radial distribution with density $h(r)$.
5. Return the value $\mathbf{X} = R\mathbf{Z}$.

This method works in any dimension and the first three steps are adaptable to a wide variety of shapes, more than just the contours described above. This gives a way to define distributions on paths and surfaces. Figure 6 illustrates some examples with different shapes and weights. In all cases the points $\mathbf{Z}$ are sampled from the approximating simplex faces; to work well the tessellation should be fine enough to closely approximate the shape of the surface of interest. This is controlled by the parameter k described above. The trefoil knot in the upper left plot is approximated by 101 line segments; for simulation, a line

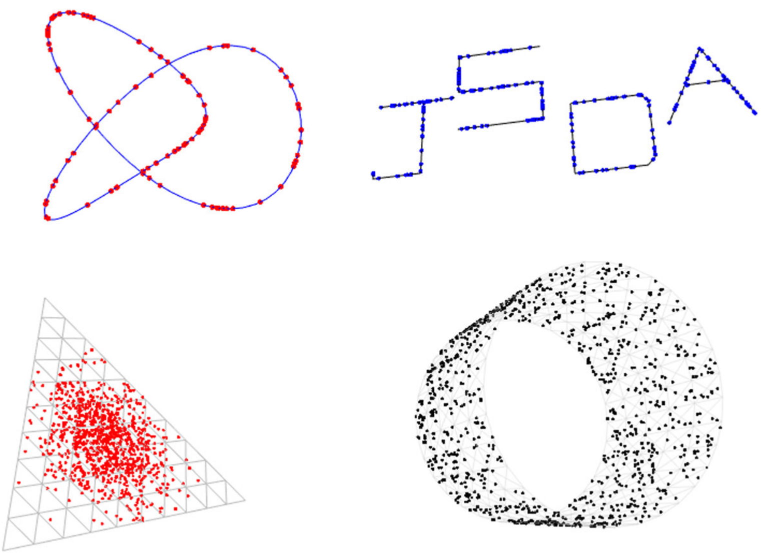

Fig. 6 Approximate simulation from general sets; details are given in the text. At top left is a trefoil knot with points sampled uniformly from the path. *Top right* has the letters JSDA constructed from line segments, then embedded in $\mathbb{R}^3$. Points are then sampled uniformly according to lengths of the line segments. *Bottom left* has points sampled on the unit simplex according to a density $\exp\left(-20|\mathbf{x} - \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)|^2\right)$. The last plot shows points sampled uniformly from a hollow tube

segment is sampled uniformly ($w_j = 1/101$) and then a point is picked randomly along that segment. In the second plot, the letters JSDA are constructed out of straight line segments, then embedded in $\mathbb{R}^3$. A line segment is selected with weight proportional to the lengths of the line segments making up the letters, and then a point is sampled uniformly along that segment. The bottom left plot subdivides the unit simplex $x_1 + x_2 + x_3 = 1$, $x_1 \geq 0$, $x_2 \geq 0$, $x_3 \geq 0$ into 100 triangles of equal area (a $k = 10$ edge subdivision) and weights are assigned to each triangle with weights proportional to $w_j =$ average of the density $\exp\left(-20|\mathbf{x} - \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)|^2\right)$ at the vertices of simplex j. The last plot shows a hollow tube approximated by 160 rectangles (5 subdivisions along the axis and 32 subdivisions around the cylinder) with rectangles sampled uniformly and points sampled uniformly from that rectangle.

The subdivision process, including the numerical cubature is the slowest part of the process. This is done in the R function `cfunc.finish`, which finishes the definition of a contour by performing the above calculations and saving the results in an object of class "contour.function". For the example the 3-dimensional example in Fig. 4 took about half an hour[1] to complete the construction.

In contrast, once the tessellation is produced, density calculations and simulations are quite fast: to evaluate a density at 10,000 points takes less than a second and to simulate 100,000 random vectors takes less than a second for this example.

In principle, the methods described here work in any dimension; in practice the numerical challenges, particularly evaluating the integral in (2) and the time needed to work limit us as the dimension increases. At the current time, these methods are useful for low dimension d =2, 3, or 4.

Endnote

[1] Times are for an Intel i5-4460 CPU at 3.20 GHz.

Appendix

Here are the R statements used to produce Figs. 1 and 2 from the R package `gensphere`.

```
# define a new contour function (cfunc)
cfunc <- cfunc.new(d=2)
cfunc <- cfunc.add.term( cfunc,"constant",k=1)
cfunc <- cfunc.add.term( cfunc,"proj.normal",
             k=c( 1, sqrt(2)/2, sqrt(2)/2, 0.1) )
cfunc <- cfunc.add.term( cfunc,"proj.normal",
             k=c( 1, -1,0, 0.1) )
cfunc <- cfunc.finish( cfunc, nsubdiv=4 )

# simulate from the tessellation of the contour
x <- rmvmesh( n=500, cfunc$tessellation,
                     cfunc$tessellation.weights )
 dev.new(height=4,width=8); par(mfrow=c(1,2))
 plot(cfunc); plot(x)

# define a generalized spherical distribution with
# the above contour and a gamma radial component
rradial <- function( n ){ rgamma( n, shape=2 ) }
dradial <- function( x ) { dgamma( x, shape=2 ) }
```

```
dist <- gensphere( cfunc, dradial, rradial, g0=1 )

# simulate from the generalized spherical distribution
x <- rgensphere( 1000, dist )
plot(t(x))

# compute the density
xx <- matrix(1:8,nrow=2)
dgensphere( xx, dist )
```

Additional file 1 contains the R commands to generate the other figures in this paper.

Additional file

Additional file 1: R commands to generate Figs. 3, 4, 5 and 6. (R 11.2 kb)

Acknowledgements

The author is grateful to the referees and associate editor who provided valuable suggestions on improving the paper and additional references.

Supported by contract W911NF-12-1-0385 from the Army Research Office.

Competing interests

I confirm that I have read SpringerOpen’s guidance on competing interests and have no competing interests in the manuscript.

Received: 22 January 2016 Accepted: 13 October 2016

Published online: 28 October 2016

References

Arnold, BC, Castillo, E, Sarabia, JM: Multivariate distributions defined in terms of contours. J. Stat. Plan. Inf. **138**, 4158–4171 (2008)

Balkema, G, Nolde, N: Asymptotic independence for unimodal densities. Adv. Appl. Prob. **42**, 411–432 (2010)

Fernández, C, Osiewalski, J, Steel, MFJ: Modeling and inference with ν-spherical distributions. J. Amer. Stat. Assoc. **90**, 1331–1340 (1995)

Kamiya, H, Takemura, A, Kuriki, S: Star-shaped distributions and their generalizations. J. Stat. Plan. Inf. **138**, 3429–3447 (2008)

Nolan, JP: Multivariate elliptically contoured stable distributions: theory and estimation. Comp. Stat. **28**, 2067–2089 (2013)

Nolan, JP: mvmesh: Multivariate Meshes and Histograms in Arbitrary Dimensions. R package version 1.1, on CRAN (2015a). https://CRAN.R-project.org/package=mvmesh. Accessed 16 May 2016

Nolan, JP: SphericalCubature: Numerical Integration over Spheres and Balls in n-Dimensions. R package version 1.1, on CRAN (2015b). https://CRAN.R-project.org/package=SphericalCubature. Accessed 24 July 2016

Rattihalli, RN, Basugade, AB: Generation of densities using contour transformations. J. Indian Stat. Assoc. **47**, 63–90 (2009)

Rattihalli, RN, Patil, PY: Generalized ν-spherical densities. Comm. Stat. Theory Methods. **39**, 3568–3583 (2010)

Richter, WD: Geometric disintegration and star-shaped distributions. J. Stat. Distrib. Appl. **1**, 20 (2014). doi:10.1186/s40488-014-0020-6

Simon, C, Blume, L: Mathematics for Economists. Norton, New York (1994)