Vol. 00 (0000) 1
DOI: 0000

# General Bayesian Inference over the Stiefel Manifold via the Givens Representation

**Arya A. Pourzanjani[1], Richard M. Jiang[1], Brian Mitchell[1], Paul J. Atzberger[2] and Linda R. Petzold [1]**

[1] *Compute Science Department, University of California, Santa Barbara e-mail:* arya@ucsb.edu

[2] *Math Department, University of California, Santa Barbara*

**Abstract:** We introduce an approach based on the Givens representation that allows for a routine, reliable, and flexible way to infer Bayesian models with orthogonal matrix parameters. This class of models most notably includes models from multivariate statistics such factor models and probabilistic principal component analysis (PPCA). Our approach overcomes several of the practical barriers to using the Givens representation in a general Bayesian inference framework. In particular, we show how to inexpensively compute the change-of-measure term necessary for transformations of random variables. We also show how to overcome specific topological pathologies that arise when representing circular random variables in an unconstrained space. In addition, we discuss how the alternative parameterization can be used to define new distributions over orthogonal matrices as well as to constrain parameter space to eliminate superfluous posterior modes in models such as PPCA. While previous inference approaches to this problem involved specialized updates to the orthogonal matrix parameters, our approach lets us represent these constrained parameters in an unconstrained form. Unlike previous approaches, this allows for the inference of models with orthogonal matrix parameters using any modern inference algorithm including those available in modern Bayesian modeling frameworks such as Stan, Edward, or PyMC3. We illustrate with examples how our approach can be used in practice in Stan to infer models with orthogonal matrix parameters, and we compare to existing methods.

**MSC 2010 subject classifications:** Primary 60K35, 60K35; secondary 60K35.
**Keywords and phrases:** sample, LaTeX $2_\varepsilon$.

## 1. Introduction

Statistical models parameterized in terms of orthogonal matrices are ubiquitous, particularly in the treatment of multivariate data. This class of models includes certain multivariate time series models (Brockwell et al., 2002), factor models (Johnson and Wichern, 2004), and a swath of recently developed probabilistic dimensionality reduction models such as Probabilistic PCA (PPCA), Exponential Family PPCA (BXPCA), mixture of PPCA (Ghahramani et al., 1996), and Canonical Correlation Analysis (CCA) (Murphy, 2012, Chapt. 12.5).

These sorts of models have not only enjoyed extensive use in fields such as psychology (Ford et al., 1986), but are also gaining traction in diverse applications including biology (Hamelryck et al., 2006), finance (Lee et al., 2007), materials science (Oh et al., 2017), and robotics (Lu and Milios, 1997).

Despite their ubiquity, their remains no quick, routine, and flexible options for fitting models with orthogonal matrix parameters. Existing methods for inferring these models are either insufficiently general or too complicated to implement and tune in isolation. Modern probabilistic programming frameworks, such as Stan, Edward, and PyMC3 (Carpenter et al., 2016; Tran et al., 2016; Salvatier et al., 2016), try to abstract their users away from the details of inference and implementation, but none offer support for orthogonal matrix parameters. The reason is that rather than using a specialized inference algorithm for orthogonal matrices which existing approaches do, these software frameworks typically handle constrained parameters such as orthogonal matrices by transforming them to an unconstrained space (Carpenter et al., 2016; Kucukelbir et al., 2014). For example, if a model contains a parameter $\sigma > 0$ that is constrained to be positive, these frameworks typically take the log of this parameter and conduct inference over $\tilde{\sigma} = \log \sigma$, which is unconstrained.

An unconstrained parameterization of orthogonal matrices would allow for general Bayesian inference in any software framework without having to change its inner-workings, but because of the complexities in dealing with the space of orthogonal matrices, otherwise known as the Stiefel manifold, several challenges remain in the way of this approach. While many parameterizations of orthogonal matrices exist (Anderson et al., 1987; Shepard et al., 2015), only smooth representations, such as the Givens representation, can be practically considered, as inference methods such as Hamiltonian Monte Carlo (HMC) typically require continuous and differentiable likelihoods. Furthermore, any such transformation of a random variable typically requires computing a change-of-measure adjustment term that is often unknown or expensive to compute. A further complication is that the Stiefel manifold has a fundamentally different topology than Euclidean space, which can lead to biased inference if particular care is not taken in implementation. Lastly, while not strictly necessary, any representation would ideally have an intuitive interpretation that would allow practitioners to work with and even define useful distributions in terms of the new representation.

We introduce a general approach to the posterior inference of statistical models with orthogonal matrix parameters based on the Givens representation of orthogonal matrices. We address several practical implementation issues such as computation of the change-of-measure adjustment, as well as proper handling of transformed coordinates to ensure unbiased samples. Our approach enables the application of any general inference algorithm to models containing orthogonal matrix parameters, allowing inference of these models by any commonly available inference algorithm such as HMC Neal et al. (2011), the No-U-Turn Sampler (NUTS) (Hoffman and Gelman, 2014), Automatic Differentiation Variational Inference (ADVI) (Kucukelbir et al., 2014) or Black Box Variational Inference (Ranganath et al., 2014). Unlike existing approaches, our approach

is easy to implement and does not require any specialized inference algorithms or modifications to existing algorithms or software. This allows users to rapidly build and prototype complex probabilistic models with orthogonal matrix parameters in any common software framework such as Stan, Edward, or PyMC3 without the worry of messy implementation details.
In Section 2 we discuss existing methods for Bayesian inference over the Stiefel manifold and the difficulty in implementing these methods in a general Bayesian inference framework. In Section 3 we describe the Givens representation by first introducing the Givens reduction algorithm and then connecting it to a geometric perspective of the Stiefel manifold, providing an approachable intuition to the transform. We go on to describe practical solutions for using the Givens representation in a general Bayesian inference setting in Section 4. In Section 5 we illustrate with statistical examples the use of the Givens representation and how it compares to existing methods in practice. Lastly, we conclude with a brief discussion in Section 6 where we summarize our contributions.

## 2. Related Work

Hoff (2009) introduces a Gibbs sampling approach to update unknown orthogonal matrix parameters from a collection of known conditional distributions. Unfortunately, this requires that the conditional distribution of the orthogonal matrix parameter given other model parameters belongs to a known parametric distribution that is easy to sample. In practice, this limits the approach to a specific class of models.
More general HMC methods have been devised, but their use of specialized update rules makes them difficult to implement and tune in practice. In particular, these methods infer orthogonal matrix parameters by using different HMC update rules for constrained and unconstrained parameters. This separation of constrained and unconstrained parameters requires additional book-keeping to know which update rules to use on which parameter. Unfortunately, many probabilistic programming languages do not keep track of this as they treat parameters agnostically by transforming to an unconstrained space. The specialization of these methods to HMC also makes them difficult to generalize to other inference algorithms based on VI or optimization which an unconstrained parameterization approach would have no trouble with.
Specifically, Brubaker et al. (2012) proposed a modified HMC, which uses a different update rule for constrained parameters based on the symplectic SHAKE integrator (Leimkuhler and Reich, 2004). For unconstrained parameters, the method uses a standard Leapfrog update rule. For constrained parameters, the method first takes a Leapfrog step which usually moves the parameter to a value that does not obey constraints. The method then uses Newton's method to "project" the parameter value back down to the manifold where the desired constraints are satisfied.
Byrne and Girolami (2013) as well as Holbrook et al. (2016) also utilize a separate HMC update rule to deal with constrained parameters. Specifically, they

utilize analytic results and the matrix exponential to update the parameters in such a way that guarantees constraints are still satisfied in the embedded matrix coordinates. More precisely, they use the fact that analytic solutions for the geodesic equations on the Stiefel manifold in the embedded coordinates are known. This gives rise to their Embedded Manifold HMC (EMHMC) algorithm. Like the method of Brubaker et al. (2012), the use of separate update rules in EMHMC makes the algorithm difficult to implement in more general settings.

## 3. The Givens Representation of Orthogonal Matrices

We motivate then introduce the Givens representation by first describing the related Givens reduction algorithm of numerical analysis then tying this to the geometric aspects of the Stiefel manifold.

### *3.1. Givens Rotations and Reductions*

Given any $n \times p$ matrix, $A$, the Givens reduction algorithm is a numerical algorithm for finding the $QR$-factorization of $A$, i.e. an $n \times p$ orthogonal matrix $Q$ and an upper-triangular $p \times p$ matrix $R$ such that $A = QR$. The algorithm works by successively applying a series of Givens rotations so as to "zero-out" elements of $A$ below the diagonal. These Givens rotations are simply $n \times n$ matrices, $R_{ij}(\theta_{ij})$, that take the form of an identity matrix except for the $(i,i)$ and $(j,j)$ positions which are replaced by $\cos\theta_{ij}$ and the $(i,j)$ and $(j,i)$ positions which are replaced by $-\sin\theta_{ij}$ and $\sin\theta_{ij}$ respectively.

When applied to a vector, $R_{ij}(\theta_{ij})$ has the effect of rotating the vector counter-clockwise in the $(i,j)$-plane, while leaving other elements fixed. Intuitively, its inverse, $R_{ij}^{-1}(\theta_{ij})$, has the same effect, but clockwise. Thus one can "zero-out" the $j$th element, $u_j$, of a vector $u$, by first using the arctan function to find the angle $\theta_{ij}$ formed in the $(i,j)$-plane by $u_i$ and $u_j$, and then multiplying by the matrix $R_{ij}^{-1}(\theta_{ij})$ (Figure 1, inset).

In the Givens reduction algorithm, these rotation matrices are applied one-by-one to $A$ in this way to eliminate all elements below the diagonal. First, all elements in the first column below the first row are eliminated by successively applying the rotation matrices $R_{12}^{-1}(\theta_{12}), R_{13}^{-1}(\theta_{13}), \cdots, R_{1n}^{-1}(\theta_{1n})$ (Figure 2). Because multiplication by $R_{ij}(\theta_{ij})$ only affects elements $i$ and $j$ of a vector, once the $j$th element is zeroed out, the subsequent rotations, $R_{13}^{-1}(\theta_{13}), \cdots, R_{1n}^{-1}(\theta_{1n})$, will leave the initial changes unaffected. Similarly, once the first column of $A$ is zeroed out below the first element, the subsequent rotations, which do not involve the first element will leave the column unaffected. The rotations $R_{23}^{-1}(\theta_{23}), \cdots, R_{2n}^{-1}(\theta_{2n})$ can thus be applied to zero out the second column, while leaving the first column unaffected. This results in the upper triangular matrix

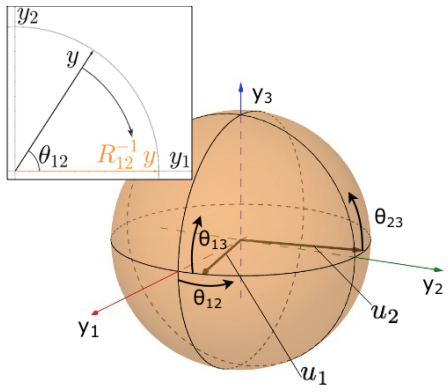


FIG 1. *(Inset) Givens rotations can be used to rotate a vector so as to eliminate its component in a certain direction. (Main Figure) A p-frame on the Stiefel manifold can be visualized as a set of rigidly connected orthogonal basis vectors, $u_1$ and $u_2$, shown here in black. One can move about the Stiefel manifold and describe any p-frame by simultaneously applying rotations matrices of a prescribed angle to these basis vectors. Applying the rotation matrix $R_{12}(\theta_{12})$ corresponds to rotating the two basis vectors toegher in the (1,2)-plane, which by our convention is the $(x, y)$-plane. Similarly, simultaneously apply $R_{13}(\theta_{13})$ corresponds to a rotation of the 2-frame in the $(1, 3)$ or $(x, z)$-plane, while $R_{23}(\theta_{23})$ corresponds to rotating $u_2$ about $u_1$.*

$$R_* := \underbrace{R_{pn}^{-1}(\theta_{pn}) \cdots R_{p,p+1}^{-1}(\theta_{p,p+1}) \cdots R_{2n}^{-1}(\theta_{2n}) \cdots R_{23}^{-1}(\theta_{23}) \cdots R_{1n}^{-1}(\theta_{1n}) \cdots R_{12}^{-1}(\theta_{12})}_{Q_*^{-1}} A. \tag{3.1}$$

Crucially, the product of rotations, which we call $Q_*^{-1}$, is orthogonal since it is simply the product of rotation matrices which are themselves orthogonal. Thus its inverse can be applied to both sides of Equation 3.1 to obtain

$$Q_* R_* = A. \tag{3.2}$$

The familiar $QR$ form can be obtained by setting $Q$ equal to the first $p$ columns of $Q_*$ and setting $R$ equal to the first $p$ rows of $R_*$. The Givens reduction is summarized in Algorithm 1.

### *3.2. The Geometry of Orthogonal Matrices*

The Stiefel manifold, $V_{p,n}$, consists of $p$-frames: ordered sets of $p$ $n$-dimensional unit-length vectors, where $p \leq n$. $p$-frames naturally correspond to $n \times p$ orthogonal matrices which can be used to define the Stiefel manifold succinctly as

$$V_{p,n} := \{Y \in \mathbb{R}^{n \times p} : Y^T Y = I\}. \tag{3.3}$$

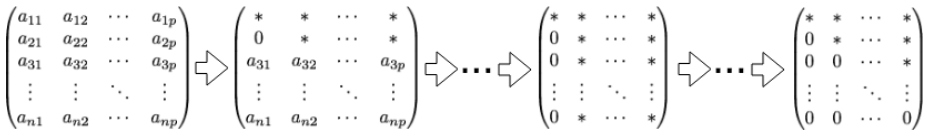

FIG 2. *The Givens reduction eliminates lower diagonal elements of an $n \times p$ matrix one column at a time. Because each rotation, $R_{ij}(\theta_{ij})$, only affects rows $i$ and $j$, previously zeroed out elements do not change.*

**Input:** $A$
**Result:** $Q, R$
$Q_*^{-1} = I$ $R_* = A$
**for** *i in 1:p* **do**
  **for** *j in (i+1):n* **do**
    $\theta_{ij} = \arctan(Y[j,i]/Y[i,i])$
    $Q_*^{-1} = R_{ij}^{-1}(\theta_{ij})Q_*^{-1}$
    $R_* = R_{ij}^{-1}(\theta_{ij})R_*$
  **end**
**end**
return $Q_*[, 1:p], R_*[1:p, 1:p]$

**Algorithm 1:** Psuedo-code for the Givens reduction algorithm for obtaining the $QR$ factorization of a matrix $A$.

Geometrically, an element of the Stiefel manifold can be pictured as a set of orthogonal, unit-length vectors that are rigidly connected to one another. A simple case is $V_{1,3}$, which consists of a single vector, $u_1$, on the unit sphere. This single vector can be represented by two polar coordinates that we naturally think of as longitude and latitude, but can also be thought of simply as subsequent rotations of the standard basis vector $e_1 := (1,0,0)^T$ in the $(x,y)$ and $(x,z)$ planes, which we refer to as the $(1,2)$ and $(1,3)$ planes for generality. In mathematical terms, $u_1$ can be represented as $u_1 = R_{12}(\theta_{12})R_{13}(\theta_{13})e_1$ (Figure 1).
Continuing without geometric interpretation, $V_{2,3}$ can be pictured as a vector in $V_{1,3}$ that has a second orthogonal vector, $u_2$, that is rigidly attached to it as it moves about the unit sphere. Because this second vector is constrained to be orthogonal to the first, its position can be described by a single rotation about the first vector. Thus elements of $V_{2,3}$ can be represented by three angles: two angles, $\theta_{12}$ and $\theta_{13}$, that represent how much to rotate the first vector, and a third angle, $\theta_{23}$ that controls how much the second vector is rotated about the first (Figure 1). Mathematically this can be represented as the $3 \times 2$ orthogonal matrix $R_{12}(\theta_{12})R_{13}(\theta_{13})R_{23}(\theta_{23})(e_1, e_2)$.
Although elements of the Stiefel manifold can be represented by $n \times p$ matrices, their inherent dimension is less than $np$ because of the constraints that the matrices must satisfy. The first column must satisfy a single constraint: the unit-length constraint. The second column must satisfy two constraints: not only must it be unit length, but it must also be orthogonal to the first column.

The third column must additionally be orthogonal to the second column, giving it a total of three constraints. Continuing in this way reveals the inherent dimensionality of the Stiefel manifold to be

$$d := np - 1 - 2 - \cdots p = np - \frac{p(p+1)}{2}. \tag{3.4}$$

### 3.3. Obtaining the Givens Representation

The Givens reduction applied to an orthogonal matrix gives rise to a representation of the Stiefel manifold that generalizes the intuitive geometric interpretation described above. When applied to an $n \times p$ orthogonal matrix $Y$, the Givens reduction yields

$$R_{pn}^{-1}(\theta_{pn}) \cdots R_{p,p+1}^{-1}(\theta_{p,p+1}) \cdots R_{2n}^{-1}(\theta_{2n}) \cdots R_{23}^{-1}(\theta_{23}) \cdots R_{1n}^{-1}(\theta_{1n}) \cdots R_{12}^{-1}(\theta_{12}) Y = I_{n,p} \tag{3.5}$$

where $I_{p,n}$ is defined to be the first $p$ columns of the $n \times n$ identity matrix, i.e. the matrix consisting of the first $p$ standard basis vectors $e_1, \cdots, e_p$. The first $n-1$ rotations transform the first column into $e_1$, since it zeros out all elements below the first and the orthogonal rotations do not affect the length of the vector which by hypothesis is unit length. Similarly, the next $n-2$ rotations will leave the length of the second column and its orthogonality to the first column intact because again, the rotation matrices are orthogonal. Because the second column must be zero below its second element it must be $e_2$. Continuing in this way explains the relationship in Equation 3.5.

Because $Y$ was taken to be an arbitrary orthogonal matrix, then it is clear from Equation 3.5 that any orthogonal matrix $Y$ can be factored as

$$Y = R_{12}(\theta_{12}) \cdots R_{1n}(\theta_{1n}) \cdots R_{23}(\theta_{23}) \cdots R_{2n}(\theta_{2n}) \cdots R_{p,p+1}(\theta_{p,p+1}) \cdots R_{pn}(\theta_{pn}) I_{n,p}. \tag{3.6}$$

Defining $\Theta := (\theta_{12} \cdots \theta_{1n} \cdots \theta_{23} \cdots \theta_{2n} \theta_{p,p+1} \cdots \theta_{pn})$ we can consider any orthogonal matrix as a function, $Y(\Theta)$, of these angles, effectively parameterizing the Stiefel manifold and yielding the Givens representation. The Givens representation is a smooth representation with respect to the angles $\Theta$ (Shepard et al., 2015), and lines up with our geometric insight discussed in the previous subsection.

## 4. The Givens Representation for Bayesian Inference of Orthogonal Matrix Parameters

Practical use of the Givens representation in a general Bayesian inference framework involves solving several practical challenges. In addition to the standard change of measure term required in any transformation of a random variable, careful care must be taken to address certain pathological cases of the Givens

representation that occur due to the different topologies of the Stiefel manifold and Euclidean space. We further describe these challenges and explain how we overcome them in practice. We also briefly remark on how the Givens representation can be leveraged in practice to solve issues with identifiability and define new and useful distributions over the Stiefel manifold. We conclude the section by describing how the computation of the Givens representation scales in theory, particularly in comparison to EMHMC.

### *4.1. Transformation of Measure Under the Givens Representation*

As is usual in any transformation of random variables, careful care must be taken to include a Jacobian determinant term in the transformed density to account for a change of measure under the transformation. For a posterior density over orthogonal matrices that takes the form $p_Y(Y)$, the proper density over the transformed random variable, $\Theta(Y)$, takes the form $p_\Theta(\theta) = p_Y(Y(\Theta))|J_{Y(\Theta)}(\Theta)|$ (Keener, 2011). Intuitively, this extra Jacobian determinant term accounts for how probability measures are distorted by the transformation (Figure 3). Unfortunately, the Givens representation, $Y(\Theta)$, is map from a space of dimension $d := np - p(p+1)/2$ to a space of dimension $np$. Hence the determinant is non-square and thus undefined.

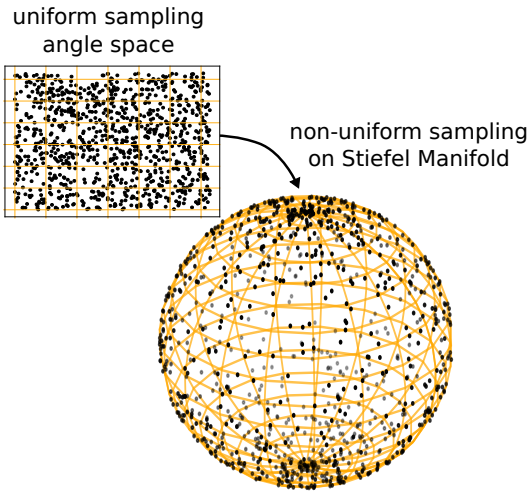


FIG 3. *Uniform sampling in the Givens representation coordinates does not necessarily lead to uniform sampling over the Stiefel manifold without the proper measure adjustment term. Under the mapping, regions near the pole are shrunk to regions on the sphere with little area, as opposed to regions near to the equator which the transform maps to much larger areas on the sphere. Intuitively, the change-of-measure term quantifies this proportion of shrinkage in area.*

To compute the change of measure term analogous to the Jacobian determinant,

one must appeal to the algebra of differential forms. Denote the product of $n \times n$ rotation matrices in the Givens representation by $G$, i.e.

$$G := R_{12}(\theta_{12}) \cdots R_{1n}(\theta_{1n}) \cdots R_{23}(\theta_{23}) \cdots R_{pn}(\theta_{pn}) \cdots R_{p,p+1}(\theta_{p,p+1}) \cdots R_{pn}(\theta_{pn}), \tag{4.1}$$

and denote its $j$th column by $G_j$. Muirhead (2009) shows that the proper measure form for a signed surface element of $V_{p,n}$ is the differential form

$$\bigwedge_{i=1}^{p} \bigwedge_{j=i+1}^{n} G_j^T \, dY_i. \tag{4.2}$$

Letting $J_{Y_i(\Theta)}(\Theta)$ be the Jacobian of the $i$th column of $Y$ with respect to the angle coordinates of the Givens representation, this differential form can be written in the coordinates of the Givens representation as

$$\bigwedge_{i=1}^{p} \bigwedge_{j=i+1}^{n} G_j^T \, J_{Y_i(\Theta)}(\Theta) d\Theta. \tag{4.3}$$

Because this is a wedge product of $d$ $d$-dimensional elements, Equation 4.3 can be conveniently written as the determinant of the $d \times d$ matrix

$$\begin{pmatrix} G_{2:n}^T J_{Y_1(\Theta)}(\Theta) \\ G_{3:n}^T J_{Y_2(\Theta)}(\Theta) \\ \vdots \\ G_{p:n}^T J_{Y_p(\Theta)}(\Theta) \end{pmatrix} \tag{4.4}$$

where $G_{k:l}$ denote columns $k$ through $l$ of $G$. As we show in the appendix, this term can be analytically simplified to the following simple product whose absolute value serves as our measure adjustment term:

$$\prod_{i=1}^{p} \prod_{j=i+1}^{n} \cos^{j-i-1} \theta_{ij}. \tag{4.5}$$

### *4.2. Implementation of Angle Coordinates*

When using the Givens representation for general Bayesian inference in practice, care must be taken to properly account for pathologies that arise when mapping the Stiefel manifold to Euclidean space. We let $\theta_{12}, \theta_{23}, \cdots \theta_{p,p+1}$ range from $-\pi$ to $\pi$ and we refer to these specific coordinates as the latidinal coordinates to evoke the analogy for the simple spherical case. Similarly, we let the remaining coordinates range from $-\pi/2$ to $\pi/2$ and we refer to these coordinates as longitudinal coordinates. This choice of intervals defines a coordinate chart from Euclidean space to the Stiefel manifold, i.e. a mapping between the two spaces. As is inevitable with any coordinate chart between differing topological spaces, there is a subset of the Stiefel manifold of measure zero that

the Givens representation will be unable to represent because the topologies of the Stiefel manifold and Euclidean space differ. For $V_{1,3}$ this corresponds to a sliver of the sphere (Figure 4). Furthermore, the coordinate chart will contain singularities where the adjustment term (Equation 4.5) becomes zero, possibly biasing any distributional calculations. On the sphere, this corresponds to areas on the unconstrained space being mapped to smaller and smaller areas near the pole (Figure 4). We further discuss these pathologies and introduce techniques to overcome them in practice.

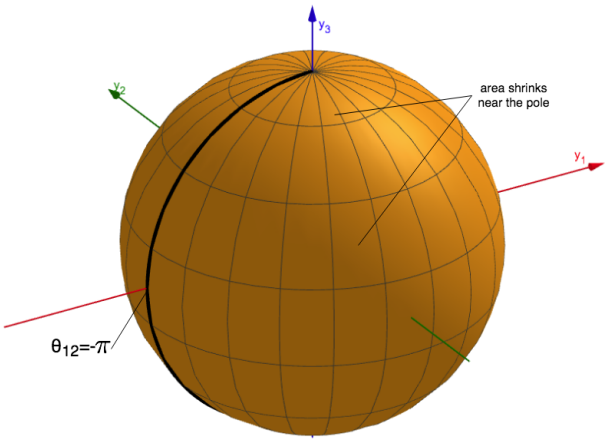


FIG 4. *The angular coordinates chart has an infinitesimal sliver of measure zero that lies between $\theta_{12} = -\pi$ and $\theta_{12} = \pi$ that separates the two parts of the sphere in the Givens representation. The grid over the sphere reveals how the Givens representation maps areas that are the same size in the $\Theta$ coordinates to smaller and smaller regions on the sphere the closer they are to the poles.*

As is routinely done in practice, a logistic transform can be used to map the interval $[-\pi, \pi]$ to the unconstrained interval $(-\infty, \infty)$. Unfortunately, this leaves regions of parameter space that should otherwise be connected, disconnected by the aforementioned set of measure zero. In practice, this can lead to biased sampling where regions of parameter space with equal mass are not visited for equal amounts of time if the posterior is not sufficiently concentrated (Figure 5, upper).

To overcome this, we create for each longitudinal angle, $\theta$, a pair of coordinates $x$ and $y$ then set $\theta = \arctan(y/x)$. Introducing this auxilary dimension connects otherwise separate regions of parameter space. Furthermore, we let $r = \sqrt{x^2 + y^2} \sim \mathcal{N}(1, 0.1)$. This helps in practice to avoid regions of parameter space where arctan is ill-defined, while leaving the marginal distribution of $\theta$ untouched (Figure 5, lower).

For the latitudinal angles, we can use the standard technique of constraining the parameters over an interval, then using the logistic transform. However, to avoid

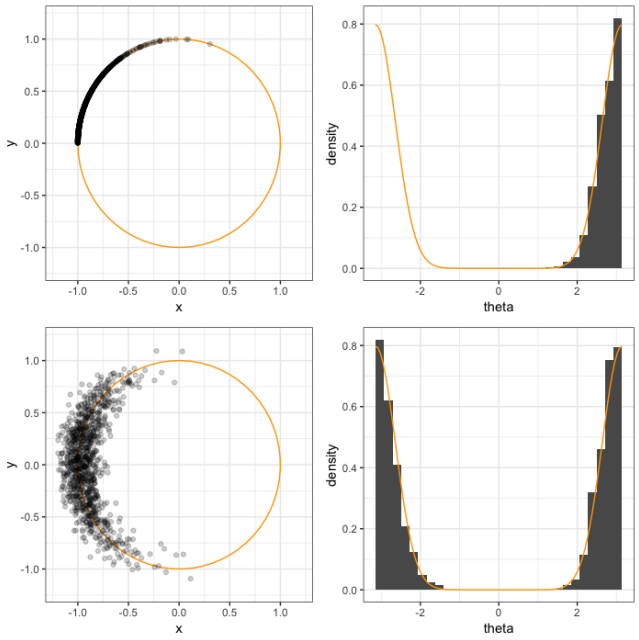


FIG 5. *(Upper) When posterior mass is not sufficiently bounded away from the edges of the interval, regions of posterior mass that are separated by massless regions may arise. Because these relatively massless regions are difficult for a sampler to traverse, bias sampling can often occur as the sampler is only able to visit one mode of the posterior distribution. (Lower) By introducing an auxiliary coordinate, one can effecitvely replicate the topology of a circle, effectively "wrapping" the two ends of the interval, leading to unbiased sampling.*

singularities in the measure adjustment term, we set the interval to the slightly smaller interval $[-\pi/2 + \epsilon, \pi/2 - \epsilon]$ rather than the full interval $[-\pi/2, \pi/2]$. Here $\epsilon$ is a small value (on the order of $10^{-5}$ in our experiments) that effectively blocks off a small portion of parameter space surrounding the singularities of the change of measure term. In the spherical case, this is equivalent to a small patch on either pole that is blocked off. In practice, blocking off this small region avoid issues such as divergences that occur in HMC in such regions of high curvature, while not meaningfully affecting the results of posterior inference.

### *4.3. Coordinate Charts and Identifiability*

For certain applications such as PPCA, it may be desirable to further limit parameter space to avoid symmetries that lead to identifiability issues in the posterior. In the Givens representation coordinates this is simply a matter of constraining the range of the longitudinal angles to the interval $[-\pi/2, \pi/2]$ (Figure 6). Unfortunately, as in the case of the full interval, this can lead to biased sampling due to regions of low mass separating regions of high mass in

parameter space. However, this issue can be resolved by carefully connecting these regions via a simple mirroring technique which we describe.

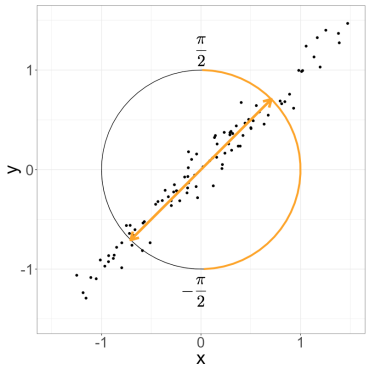


FIG 6. *PPCA seeks to find the best lower dimensional p-frame to describe a high-dimensional set of points. For $n = 2$ and $p = 1$, this corresponds to the vector that most closely describes a set of two-dimensional points that lie close to flat line. Since a p-frame and its negative can describe the data equally well, a multi-modal posterior over the Stiefel manifold results. By limiting the longitudinal angle to lie in the interval $[-\pi/2, \pi/2]$ the sampler does not consider this redundant mode.*

We can allow the original longitudinal and latitudinal coordinates, $\theta_{\text{lon}}$ and $\theta_{\text{lat}}$ to freely roam the Stiefel manifold using the aforementioned approach then define the new transformed parameters $\theta^*_{\text{lon}}$ and $\theta^*_{\text{lat}}$ to essentially be mirrored versions of these original coordinates. Specifically, we can define

$$\theta^*_{\text{lon}} = \begin{cases} \theta_{\text{lon}}, & |\theta_{\text{lon}}| \leq \frac{\pi}{2} \\ -\frac{\pi}{2} + (\theta_{\text{lon}} - \frac{\pi}{2}), & \theta_{\text{lon}} > \frac{\pi}{2} \\ \frac{\pi}{2} + (\theta_{\text{lon}} + \frac{\pi}{2}), & \theta_{\text{lon}} < -\frac{\pi}{2} \end{cases} \tag{4.6}$$

and

$$\theta^*_{\text{lat}} = \begin{cases} \theta_{\text{lat}}, & |\theta_{\text{lon}}| \leq \frac{\pi}{2} \\ -\theta_{\text{lat}}, & |\theta_{\text{lon}}| > \frac{\pi}{2}. \end{cases} \tag{4.7}$$

These transformed coordinates essentially mirror and reflect the original coordinates so that once the hemisphere is crossed, the path taken continues on the opposite side of the Stiefel manifold where there would naturally be an area of high posterior mass (Figure 7). In fact, one can check that the PPCA likelihood (Equation 5.3) is continuous with respect to these new coordinates, allowing for efficient sampling even when there is appreciable posterior mass near the edge of the hemisphere.

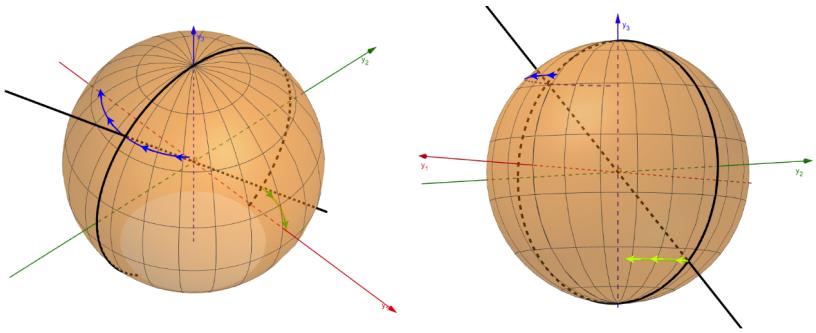

FIG 7. *Occasionally, the direction that best describes a high-dimensional dataset in PPCA (black line) is near the boundary of the longitudinal coordinate (thick black border). In this case, its negative will have an appreciable probability mass near it and the same density. Because of this continuity in the density, these areas of parameter space can be smoothly connected. Specifically, once the border is crossed (blue path) the coordinates now describe a point on the opposite end of the Stiefel manifold (green path).*

### *4.4. New Distributions Using the Givens Representation*

Rather than placing priors over standard orthogonal matrix coordinates, $Y$, one can place priors over the coordinates of the Givens representation $\Theta$. In practice this leads to new classes of possible distributions. Cron and West (2016) utilize sparsity promoting priors over the coordinates of the Givens representation to produce a distribution over the Stiefel manifold that favors sparse matrices. They apply this distribution to the estimation of normal mixture classification probabilities. León et al. (2006) make use of a different parameterization of orthogonal matrices to define a distribution over orthogonal matrices.

### *4.5. Computational Scaling of the Givens Representation*

The primary computational cost in using the Givens representation, is the series of $d$ $n \times n$ matrix multiplications applied to $I_{n,p}$ in Equation 3.6. Fortunately, unlike dense matrix multiplication, applying a Givens rotation to an $n \times p$ matrix only involves two vector additions of size $p$ (Algorithm 2). Thus since $d$ scales on the order of $np$, computation of the Givens representation in aggregate scales as $\mathcal{O}(np^2)$.

In comparison, EMHMC involves an orthogonalization of an $n \times p$ matrix which scales as $\mathcal{O}(np^2)$ and a matrix exponential computation that scales as $\mathcal{O}(p^3)$. In practice, we find that EMHMC scales better when $p$ is much smaller than $n$, whereas the Givens representation scales better when $p$ is large and closer to $n$. We present benchmarks in Section 4.5.

**Input:** $\theta$
**Result:** $Y$
$Y = I_{n,p}$; idx $= d$
**for** *i in p:1* **do**
  **for** *j in n:(i+1)* **do**
    $Y_i = \cos(\theta_{\text{idx}})Y[i,] - \sin(\theta_{\text{idx}})Y[j,]$
    $Y_j = \sin(\theta_{\text{idx}})Y[i,] + \cos(\theta_{\text{idx}})Y[j,]$
    $Y[i,] = Y_i$
    $Y[j,] = Y_j$
    $idx = idx - 1$
    log density $+= (j - i - 1)\log\cos\theta_{\text{idx}}$
  **end**
**end**
return $Y$

**Algorithm 2:** Psuedo-code for obtaining the orthogonal matrix $Y$ from the Givens Representation as well as appropriately adjusting the log of the posterior density.

| | | EMHMC | | Givens | |
|---|---|---|---|---|---|
| $p$ | $n$ | $\hat{R}$ | $n_{\text{eff}}$ | $\hat{R}$ | $n_{\text{eff}}$ |
| 1 | 10 | 1.00 | 231 | 1.00 | 496 |
| 1 | 100 | 1.00 | 317 | 1.00 | 488 |
| 1 | 1000 | 1.00 | 238 | 1.00 | 487 |
| 10 | 10 | 1.00 | 408 | 1.00 | 390 |
| 10 | 100 | 1.00 | 473 | 1.00 | 487 |
| 10 | 1000 | 1.00 | 454 | 1.00 | 488 |
| 100 | 100 | 1.00 | 484 | 1.00 | 479 |

TABLE 1
*$\hat{R}$ and $n_{\text{eff}}$ values averaged over all elements of the matrix parameter $Y$.*

## 5. Results and Examples

We demonstrate the use of the Givens representation and compare it with EMHMC for three common statistical examples from the literature. All Givens representation experiments were conducted in Stan using Stan's automatic warm-up and tuning options. For all Stan experiments we ensured that there were no divergences during post-warmup sampling and that all $\hat{R}$ were 1.01 or below. All timing experiments were conducted on a 2016 Macbook Pro.

### *5.1. Uniform Sampling on the Stiefel Manifold*

We sample uniformly from the Stiefel manifold of various sizes to assess the practical scalability of the Givens representation. We compare its sampling efficiency and $\hat{R}$ values to EMHMC on 500 post-warmup samples from each method (Table 1).
As mentioned in section 4.1, to uniformly samples the Stiefel manifold in the Givens representations, the change of measure term, Equation 4.5, must be computed as part of the likelihood. Meanwhile, uniform sampling over the Stiefel manifold is achieved in EMHMC simply using a constant likelihood because the

method uses the original matrix coordinates. However, as mentioned in section 4.5, this comes at the cost of an expensive HMC update to ensure the updated parameter still satisfies the constraints.. In practice, we find that EMHMC scales better as $n$ is increased, although the approach using the Givens representation in Stan remains competitive (Figure 8).

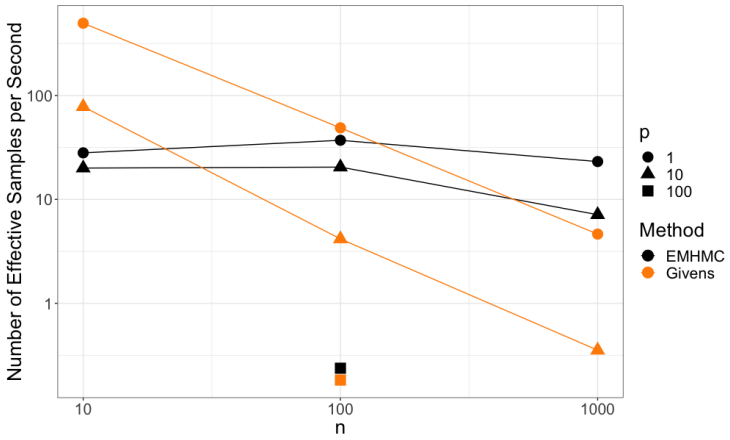


FIG 8. *For small values of $n$ the Givens representation approach in Stan produces more effective samplers per second while for larger values the EMHMC scales better since the primary cost of the matrix exponential remains constant.*

### *5.2. Probabilistic PCA (PPCA)*

Factor Analysis (FA) and Probabilistic PCA (PPCA) (Tipping and Bishop, 1999) posit a probabilistic generative model where high-dimensional data is determined by a linear function of some low-dimensional latent state (Murphy, 2012, Chapt. 12). Geometrically, for a three-dimensional set of points forming a flat pancake-like cloud, PCA can be thought of as finding the best 2-frame that aligns with this cloud (Figure 9). Formally, PPCA posits the following generative process for how a sequence of high-dimensional data vectors $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1, \cdots, N$ arise from some low dimensional latent representations $\mathbf{z}_i \in \mathbb{R}^p$ $(p < n)$:

$$\begin{aligned} \mathbf{z}_i &\sim \mathcal{N}_p(0, I) \\ \mathbf{x}_i | \mathbf{z}_i, W, \Lambda, \sigma^2 &\sim \mathcal{N}_n(W\Lambda\mathbf{z}_i, \sigma^2 I). \end{aligned} \tag{5.1}$$

To ensure identifiability $W$ is constrained to be an orthogonal $n \times p$ matrix while $\Lambda$ is a diagonal matrix with positive, ordered elements. Because $\mathbf{x}_i$ is a linear transformation of a multivariate Gaussian, its distribution is also multivariate

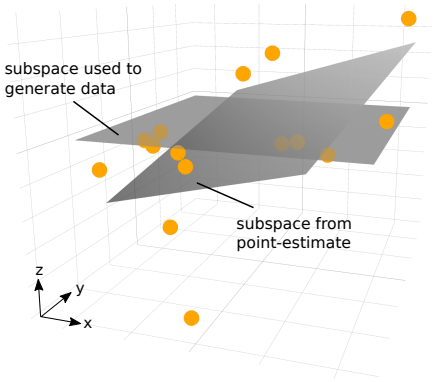


FIG 9. *PCA finds a single orthogonal matrix in the Stiefel Manifold that is closest, in terms of average squared distance, to the set of points. This point estimate can often mislead us from the true subspace, which in this case is the horizontal $(x, y)$-plane which was used to generate the noisy data. The data shown here is within the three-dimensional space parameterized by $x$, $y$, and $z$. Alternatively, in Probabilistic PCA (PPCA) a posterior distribution is used to estimate the approximating subspace and also to quantify the uncertainty of the result.*

Gaussian with mean zero and covariance $\mathbf{C} := W\lambda^2 W^T + \sigma^2$ (Murphy, 2012). Letting $\hat{\Sigma} := (1/N)\sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^T$ denote the empirical covariance matrix this gives us the simplified PPCA likelihood

$$p(\mathbf{x}_1, \cdots, \mathbf{x}_N | W, \sigma^2) = -\frac{N}{2} \ln |\mathbf{C}| - \frac{1}{2} \sum_{i=1}^{N} \mathbf{x}_i^T \mathbf{C}^{-1} \mathbf{x}_i \tag{5.2}$$

$$= -\frac{N}{2} \ln |\mathbf{C}| - \frac{N}{2} \mathrm{tr}(\mathbf{C}^{-1} \hat{\Sigma}). \tag{5.3}$$

Traditional PCA corresponds to the closed-form maximum likelihood estimator for $W$ in the limit as $\sigma^2 \to 0$, providing no measure of uncertainty for this point-estimate. Furthermore, for more elaborate models, the analytical form of the maximum-likelihood estimator is rarely known.

We used the Givens representation to infer this model using simulated data. Specifically, we generated a three-dimensional dataset that lies on a two-dimensional plane with $N = 15$ observations according to the above generative process. The data is plotted in Figure 9). We chose $\mathrm{diag}(\Lambda) = \mathrm{diag}(2, 1)$, $\sigma^2 = 1$, and $W$ to be $I_{3,2}$, which in the Givens representation corresponds to $\theta_{12} = \theta_{13} = \theta_{23} = 0$ i.e. the horizontal plane. We point out how this horizontal plane differs from the slanted plane obtained from the classical PCA maximum likelihood estimate (Figure 9). In this case, the advantage of the full posterior estimate that the Bayesian framework affords is clear. Posterior samples of $\theta_{13}$, which if we recall from Figure 1 is the Givens representation angle that controls the upwards tilt of the plane, reveal a wide posterior which cautions us against the

spurious maximum likelihood estimate of $\hat{\theta}_{13} = -0.15$ (Figure 10, right). Note also that by naturally constraining the set of angles considered as in Section 4.3, the superfluous modes that EMHMC visits are avoided. Likewise, posterior distributions of $\Lambda$ are more informative that point estimates for quantifying the inherent dimensionality of the data (Figure 10, left).

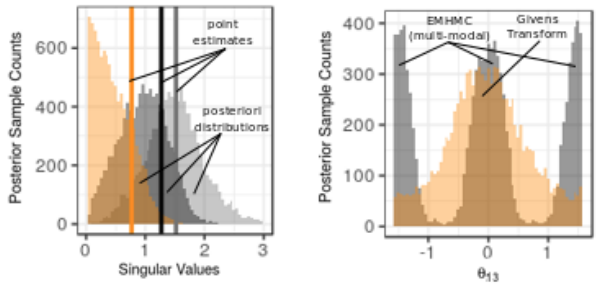


FIG 10. *PPCA inference for three-dimensional synthetic data. (Left) Posterior draws of the $\Lambda$ parameter are more informative in dimensionality selection than point-estimates. The posterior distributions for $\Lambda_1$ and $\Lambda_2$ (dark grey, and grey) contain almost no mass near zero, suggesting that the data probably contains significant variation in those directions. Meanwhile, the posterior $\Lambda_3$ (orange) has it posterior mode at zero, suggesting there is a high probability that this parameter is close to zero, which the point estimate by itself neglects to convey. (Right) By limiting the angles of rotation in the Givens Transform, we can further avoid unidentifiability in our problem and eliminate multi-modal posteriors that show up in other methods such as EMHMC.*

### *5.3. The Network Eigenmodel*

To illustrate the Givens representation on a more elaborate model with orthogonal matrix parameters, we used it to infer the network eigenmodel of Hoff (2009) on real data and compared it to EMHMC. The same model was inferred using EMHMC by Byrne and Girolami (2013). The data, which was originally described in Butland et al. (2005) and freely available in the R package *eigenmodel*, consists of a symmetric $230 \times 230$ graph matrix, $Y$, which encodes whether the proteins in a protein network of size $n = 230$ interact with one another.
The probability of a connection between all combinations of proteins can be described by the lower-triangular portion of a symmetric matrix of probabilities, however the network eigenmodel uses a much lower dimensional representation to represent this connectivity matrix. Specifically, given an orthogonal matrix $U$, a diagonal matrix $\Lambda$, and a scalar $c$, then letting $\Phi(\cdot)$ represent the probit link function, the model is described as follows:

$$c \sim \mathcal{N}(0, 10^2) \tag{5.4}$$
$$\Lambda_i \sim \mathcal{N}(0, n), \forall i \tag{5.5}$$
$$Y_{ij} \sim \text{Bernoulli}\left(\Phi([U\Lambda U^T]_{ij} + c)\right), \forall i > j. \tag{5.6}$$

The Stan implementation using the Givens representation took approximately 300 seconds to collect 1000 samples, 500 of which were warmup. In contrast, EMHMC took 812 seconds to run the same 1000 samples using the hyperparameter values specified in Byrne and Girolami (2013). Figure 11 compares traceplots for $c, \Lambda$, and the elements of the top row $U$ for the 500 post warmup samples from each sampler. As mentioned in Byrne and Girolami (2013) the non-ordering of the $\Lambda$ parameters results in a multimodality in the posterior whereby values of $\Lambda$ can be "flipped". Computed $\hat{R}$ and $n_{\text{eff}}$ for these parameters are shown in Table 2.

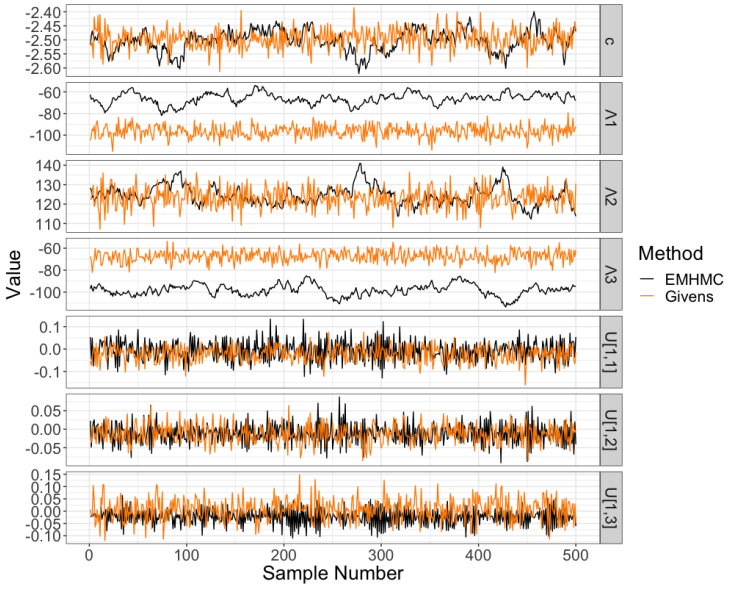


FIG 11. *Traceplots of samples from the Givens representation implementation in Stan and EMHMC reveal the multimodality in the elements of* $\Lambda$. *For brevity, only the top three elements of* $U$ *are shown.*

## 6. Discussion

We have introduced a systematic approach to incorporating the Givens representation into a general Bayesian inference framework for the purpose of inferring general Bayesian model with orthogonal matrix parameters. Our approach

| | EMHMC | | Givens | |
|---|---|---|---|---|
| Parameter | $\hat{R}$ | $n_{\text{eff}}$ | $\hat{R}$ | $n_{\text{eff}}$ |
| $c$ | 1.00 | 22 | 1.00 | 496 |
| $\Lambda_1$ | 1.00 | 19 | 1.00 | 500 |
| $\Lambda_2$ | 1.00 | 23 | 1.00 | 500 |
| $\Lambda_3$ | 1.10 | 18 | 1.00 | 500 |
| $U[1,1]$ | 1.01 | 500 | 1.00 | 500 |
| $U[2,1]$ | 1.00 | 500 | 1.00 | 500 |
| $U[3,1]$ | 1.02 | 500 | 1.00 | 500 |

TABLE 2

*$\hat{R}$ and $n_{\text{eff}}$ values for the parameters in the network eigenmodel. For brevity, only three of the matrix parameters are shown.*

overcomes practical barriers to using the Givens representation in such a setting including having to efficiently compute the measure adjustment term and dealing with singularities caused by differences in topology. Furthermore, we also provided an intuitive explanation behind the Givens representation that is accessible to statisticians and followed with practical examples for which we provide code. We expect our approach can be used quite widely in practice by a variety of practitioners.

## Appendix A: Deriving the Change of Measure Term

We derive the simplified form (Expression 4.5) of the differential form (Expression 4.2). We point out that Khatri and Mardia (1977) provide a similar expression for a slightly different representation, but do not offer a derivation. We start with the determinant of the matrix form of the change of measure term from Expression 4.4 (reproduced below):

$$\begin{pmatrix} G_{2:n}^T J_{Y_1(\Theta)}(\Theta) \\ G_{3:n}^T J_{Y_2(\Theta)}(\Theta) \\ \vdots \\ G_{p:n}^T J_{Y_p(\Theta)}(\Theta) \end{pmatrix} \tag{A.1}$$

For $l = 1, \cdots, n$, let us define the following shorthand notation

$$\partial_{i,i+l} Y_k := \frac{\partial}{\partial \theta_{i,i+l}} Y_k \tag{A.2}$$

and

$$\partial_i Y_k := \begin{pmatrix} \partial_{i,i+1} Y_k & \partial_{i,i+2} Y_k & \cdots & \partial_{in} Y_k. \end{pmatrix} \tag{A.3}$$

In the new notation Equation can be written in the following block matrix form:

$$\begin{pmatrix} G_{2:n}^T \partial_1 Y_1 & G_{2:n}^T \partial_2 Y_1 & \cdots & G_{2:n}^T \partial_p Y_1 \\ G_{3:n}^T \partial_1 Y_2 & G_{3:n}^T \partial_2 Y_2 & \cdots & G_{3:n}^T \partial_p Y_2 \\ \vdots & \vdots & \ddots & \vdots \\ G_{p:n}^T \partial_1 Y_p & G_{p:n}^T \partial_2 Y_p & \cdots & G_{p:n}^T \partial_p Y_p \end{pmatrix}. \tag{A.4}$$

First note that the block matrices above the diagonal are all zero. This can be seen by noting that the rotations in the Givens representation involving elements greater than $i$ will not affect $e_i$, i.e. letting $R_i := R_{i,i+1} \cdots R_{in}$,

$$Y_i = R_1 R_2 \cdots R_p e_i = R_1 \cdots R_i e_i. \tag{A.5}$$

Thus for $j > i$, $\partial_j Y_i = 0$ and the determinant of Expression A.4 simplifies to the product of the determinant of the matrices on the diagonal i.e. the following expression:

$$\prod_{i=1}^{p} \det\left(G_{i+1:n}^T \partial_i Y_i\right). \tag{A.6}$$

### *A.1. Simplifying Diagonal Block Terms*

Let $I_i$ denote the first $i$ columns of the $n \times n$ identity matrix and let $I_{-i}$ represent the last $n - i$ columns. The term $G_{i+1:n}^T$ in Expression A.6 can be written as

$$G_{i+1:n}^T = I_{-i}^T G^T = I_{-i}^T R_p^T \cdots R_1^T. \tag{A.7}$$

To simplify the diagonal block determinant terms in Expression A.6 we take advantage of the following fact

$$\det\left(G_{i+1:n}^T \partial_i Y_i\right) \quad = \quad \det\left(I_{-i}^T R_p^T \cdots R_1^T\right) = \det\left(I_{-i}^T R_i^T \cdots R_1^T \partial_i Y_i\right). \tag{A.8}$$

In other words, the terms $R_p^T \cdots R_{i+1}^T$ have no effect on the determinant. This can by first separating terms so that

$$\begin{aligned} \det\left(G_{i+1:n}^T \partial_i Y_i\right) &= \det\left(\underbrace{I_{-i}^T}_{(n-i)\times n} R_p^T \cdots R_1^T \underbrace{\partial_i Y_i}_{n\times(n-i)}\right) && \text{(A.9)} \\ &= \det\left(I_{-i}^T \left[R_p^T \cdots R_{i+1}^T\right] \left[R_i^T \cdots R_1^T \partial_i Y_i\right]\right) && \text{(A.10)} \end{aligned}$$

then noticing that $R_{i+1} \cdots R_p$ only effects the first $i$ columns of the identity matrix so

$$I_{-i}^T \left[R_p^T \cdots R_{i+1}^T\right] \quad = \quad \left(R_{i+1} \cdots R_p \, I_{-i}\right)^T = \left(I_{-i}\right)^T. \tag{A.11}$$

Thus Expression A.6 is equivalent to

$$\prod_{i=1}^{p} \det\left(I_{-i}^T R_i^T \cdots R_1^T \partial_i Y_i\right). \tag{A.12}$$

Now consider the $k, l$ element of the $(n-i)\times(n-i)$ block matrix $I_{-i}^T R_i^T \cdots R_1^T \partial_i Y_i$. This can be written as

$$\begin{aligned} e_{i+k}^T R_i^T \cdots R_1^T \partial_{i,i+l} Y_i &= e_{i+k}^T R_i^T \cdots R_1^T \partial_{i,i+l} (R_1 \cdots R_i e_i) \\ &= e_{i+k}^T R_i^T \cdots R_1^T R_1 \cdots R_{i-1} (\partial_{i,i+l} R_i e_i) \\ &= e_{i+k}^T R_i^T (\partial_{i,i+l} R_i e_i). \end{aligned} \tag{A.13}$$

Since $e_{i+k}^T R_i^T R_i e_i = 0$, taking the derivatives of both sides gives and applying the product rule yields

$$\begin{aligned} & \partial_{i,i+l}(e_{i+k}^T R_i^T R_i e_i) = \partial_{i,i+l} 0 \\ \Rightarrow \quad & (\partial_{i,i+l} e_{i+k}^T R_i^T) R_i e_i + e_{i+k}^T R_i^T (\partial_{i,i+l} R_i e_i) = 0 \\ \Rightarrow \quad & e_{i+k}^T R_i^T (\partial_{i,i+l} R_i e_i) = -(\partial_{i,i+l} e_{i+k}^T R_i^T) R_i e_i. \end{aligned} \tag{A.14}$$

Combining this fact with Expression A.13, the expression for the $k, l$ element of $I_{-i}^T R_i^T \cdots R_1^T \partial_i Y_i$ becomes $-(\partial_{i,i+l} e_{i+k}^T R_i^T) R_i e_i$.
However, note that

$$e_{i+k}^T R_i^T = e_{i+k}^T R_{in}^T \cdots R_{i,i+1}^T = e_{i+k}^T R_{i,i+k}^T \cdots R_{i,i+1}^T, \tag{A.15}$$

and the partial derivative of this expression with respect to $i, i+l$ is zero when $k > l$. Thus it is apparent that $I_{-i}^T R_i^T \cdots R_1^T \partial_i Y_i$ contains zeros above the diagonal and that $\det\left(I_{-i}^T R_i^T \cdots R_1^T \partial_i Y_i\right)$ is simply the product of the diagonal elements of the matrix.

### *A.2. Diagonal Elements of the Block Matrices*

To obtain the diagonal terms of the block matrices we directly compute $-\partial_{i,i+l} e_{i+k}^T R_i^T$ for $l = k$, $R_i e_i$, and their inner-product. Defining $D_{ij} := \partial_{ij} R_{ij}$,

$$
\begin{aligned}
-\partial_{i,i+k} R_i e_{i+k} &= -\partial_{i,i+k}(R_{i,i+1}\cdots R_{i,i+k}e_{i+k}) && \text{(A.16)}\\
&= -R_{i,i+1}\cdots R_{i,i+k-1} D_{i,i+k} e_{i+k} && \text{(A.17)}\\
& && \text{(A.18)}\\
&= R_{i,i+1}\cdots R_{i,i+k-1} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \cos\theta_{i,i+k} \\ 0 \\ \vdots \\ 0 \\ \sin\theta_{i,i+k} \\ 0 \\ \vdots \\ 0 \end{pmatrix} && \text{(A.19)}\\
&= R_{i,i+1}\cdots R_{i,i+k-2} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \cos\theta_{i,i+k-1}\cos\theta_{i,i+k} \\ 0 \\ \vdots \\ 0 \\ \sin\theta_{i,i+k-1}\cos\theta_{i,i+k} \\ \sin\theta_{i,i+k} \\ 0 \\ \vdots \\ 0 \end{pmatrix} && \text{(A.20)}\\
&= \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \cos\theta_{i,i+1}\cos\theta_{i,i+2}\cdots\cos\theta_{i,i+k-1}\cos\theta_{i,i+k} \\ \sin\theta_{i,i+1}\cos\theta_{i,i+2}\cdots\cos\theta_{i,i+k-1}\cos\theta_{i,i+k} \\ \vdots \\ \sin\theta_{i,i+k-1}\cos\theta_{i,i+k} \\ \sin\theta_{i,i+k} \\ 0 \\ \vdots \\ 0 \end{pmatrix} && \text{(A.21)}\\
& && \text{(A.22)}
\end{aligned}
$$

which is zero up to the $i$th spot and after the $i+k$th spot.

$$\begin{aligned} R_i e_i &= R_{i,i+1} \cdots R_{in} e_i && \text{(A.23)} \\ & && \text{(A.24)} \\ &= \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \cos\theta_{i,i+1}\cos\theta_{i,i+2}\cdots\cos\theta_{i,n-1}\cos\theta_{in} \\ \sin\theta_{i,i+1}\cos\theta_{i,i+2}\cdots\cos\theta_{i,n-1}\cos\theta_{in} \\ \vdots \\ \sin\theta_{i,n-1}\cos\theta_{in} \\ \sin\theta_{in} \end{pmatrix}. && \text{(A.25)} \end{aligned}$$

Finally, directly computing the inner-product of $-\partial_{i,i+l} e_{i+k}^T R_i^T$ and $R_i e_i$:

$$\begin{aligned} -(\partial_{i,i+l} e_{i+k}^T R_i^T)(R_i e_i) &= \cos^2\theta_{i,i+1}\cos^2\theta_{i,i+2}\cdots\cos^2\theta_{i,i+k}\cos\theta_{i,i+k+1}\cdots\cos\theta_{in} \\ &+ \sin^2\theta_{i,i+1}\cos^2\theta_{i,i+2}\cdots\cos^2\theta_{i,i+k}\cos\theta_{i,i+k+1}\cdots\cos\theta_{in} \\ &+ \sin^2\theta_{i,i+2}\cos^2\theta_{i,i+3}\cdots\cos^2\theta_{i,i+k}\cos\theta_{i,i+k+1}\cdots\cos\theta_{in} \\ &\vdots \\ &+ \sin^2\theta_{i,i+k}\cos\theta_{i,i+k+1}\cdots\cos\theta_{in} \\ &= \cos^2\theta_{i,i+2}\cos^2\theta_{i,i+3}\cdots\cos^2\theta_{i,i+k}\cos\theta_{i,i+k+1}\cdots\cos\theta_{in} \\ &+ \sin^2\theta_{i,i+2}\cos^2\theta_{i,i+3}\cdots\cos^2\theta_{i,i+k}\cos\theta_{i,i+k+1}\cdots\cos\theta_{in} \\ &\vdots \\ &+ \sin^2\theta_{i,i+k}\cos\theta_{i,i+k+1}\cdots\cos\theta_{in} \\ &= \cdots \\ &= \cos\theta_{i,i+k+1}\cdots\cos\theta_{in} \\ &= \prod_{k=i+1}^{n}\cos\theta_{ik}. && \text{(A.26)} \end{aligned}$$

Thus the determinant of the entire block matrix $I_{-i}^T R_i^T \cdots R_1^T \partial_i Y_i$ simplifies to

$$\prod_{k=i+1}^{n}\left(\prod_{j=k+1}^{n}\cos\theta_{ik}\right) = \prod_{j=i+1}^{n}\cos^{j-i-1}\theta_{ij}. \qquad \text{(A.27)}$$

Combining this with Expression A.12 yields finally

$$\prod_{i=1}^{p}\det\left(I_{-i}^T R_i^T \cdots R_1^T \partial_i Y_i\right) = \prod_{i=1}^{p}\prod_{j=i+1}^{n}\cos^{j-i-1}\theta_{ij}. \qquad \text{(A.28)}$$

## References


Anderson, T. W., Olkin, I., and Underhill, L. G. (1987). "Generation of random orthogonal matrices." *SIAM Journal on Scientific and Statistical Computing*, 8(4): 625–629.

Brockwell, P. J., Davis, R. A., and Calder, M. V. (2002). *Introduction to time series and forecasting*, volume 2. Springer.

Brubaker, M., Salzmann, M., and Urtasun, R. (2012). "A family of MCMC methods on implicitly defined manifolds." In *Artificial Intelligence and Statistics*, 161–172.

Butland, G., Peregrín-Alvarez, J. M., Li, J., Yang, W., Yang, X., Canadien, V., Starostine, A., Richards, D., Beattie, B., Krogan, N., et al. (2005). "Interaction network containing conserved and essential protein complexes in Escherichia coli." *Nature*, 433(7025): 531.

Byrne, S. and Girolami, M. (2013). "Geodesic Monte Carlo on embedded manifolds." *Scandinavian Journal of Statistics*, 40(4): 825–845.

Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M. A., Guo, J., Li, P., and Riddell, A. (2016). "Stan: A probabilistic programming language." *Journal of Statistical Software*, 20.

Cron, A. and West, M. (2016). "Models of random sparse eigenmatrices and Bayesian analysis of multivariate structure." In *Statistical Analysis for High-Dimensional Data*, 125–153. Springer.

Ford, J. K., MacCallum, R. C., and Tait, M. (1986). "The application of exploratory factor analysis in applied psychology: A critical review and analysis." *Personnel psychology*, 39(2): 291–314.

Ghahramani, Z., Hinton, G. E., et al. (1996). "The EM algorithm for mixtures of factor analyzers." Technical report, Technical Report CRG-TR-96-1, University of Toronto.

Hamelryck, T., Kent, J. T., and Krogh, A. (2006). "Sampling realistic protein conformations using local structural bias." *PLoS Computational Biology*, 2(9): e131.

Hoff, P. D. (2009). "Simulation of the matrix Bingham–von Mises–Fisher distribution, with applications to multivariate and relational data." *Journal of Computational and Graphical Statistics*, 18(2): 438–456.

Hoffman, M. D. and Gelman, A. (2014). "The No-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo." *Journal of Machine Learning Research*, 15(1): 1593–1623.

Holbrook, A., Vandenberg-Rodes, A., and Shahbaba, B. (2016). "Bayesian Inference on Matrix Manifolds for Linear Dimensionality Reduction." *arXiv preprint arXiv:1606.04478*.

Johnson, R. A. and Wichern, D. W. (2004). "Multivariate analysis." *Encyclopedia of Statistical Sciences*, 8.

Keener, R. W. (2011). *Theoretical statistics: Topics for a core course*. Springer.

Khatri, C. and Mardia, K. (1977). "The von Mises-Fisher matrix distribution in orientation statistics." *Journal of the Royal Statistical Society. Series B (Methodological)*, 95–106.

Kucukelbir, A., Ranganath, R., Gelman, A., and Blei, D. (2014). "Fully automatic variational inference of differentiable probability models." In *NIPS Workshop on Probabilistic Programming*.

Lee, S.-Y., Poon, W.-Y., and Song, X.-Y. (2007). "Bayesian analysis of the factor model with finance applications." *Quantitative Finance*, 7(3): 343–356.

Leimkuhler, B. and Reich, S. (2004). *Simulating hamiltonian dynamics*, volume 14. Cambridge University Press.

León, C. A., Massé, J.-C., and Rivest, L.-P. (2006). "A statistical model for random rotations." *Journal of Multivariate Analysis*, 97(2): 412–430.

Lu, F. and Milios, E. (1997). "Robot pose estimation in unknown environments by matching 2d range scans." *Journal of Intelligent and Robotic systems*, 18(3): 249–275.

Muirhead, R. J. (2009). *Aspects of multivariate statistical theory*, volume 197. John Wiley & Sons.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Neal, R. M. et al. (2011). "MCMC using Hamiltonian dynamics." *Handbook of Markov Chain Monte Carlo*, 2(11).

Oh, S.-H., Staveley-Smith, L., Spekkens, K., Kamphuis, P., and Koribalski, B. S. (2017). "2D Bayesian automated tilted-ring fitting of disk galaxies in large Hi galaxy surveys: 2dbat." *Monthly Notices of the Royal Astronomical Society*.

Ranganath, R., Gerrish, S., and Blei, D. (2014). "Black box variational inference." In *Artificial Intelligence and Statistics*, 814–822.

Salvatier, J., Wiecki, T. V., and Fonnesbeck, C. (2016). "Probabilistic programming in Python using PyMC3." *PeerJ Computer Science*, 2: e55.

Shepard, R., Brozell, S. R., and Gidofalvi, G. (2015). "The representation and parametrization of orthogonal matrices." *The Journal of Physical Chemistry A*, 119(28): 7924–7939.

Tipping, M. E. and Bishop, C. M. (1999). "Probabilistic principal component analysis." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3): 611–622.

Tran, D., Kucukelbir, A., Dieng, A. B., Rudolph, M., Liang, D., and Blei, D. M. (2016). "Edward: A library for probabilistic modeling, inference, and criticism." *arXiv preprint arXiv:1610.09787*.

Research reported in this publication was performed by the Systems Biology Coagulopathy of Trauma Program of the US Army Medical Research and Materiel Command under award number W911QY-15-C-0026. The author P.J.A acknowledges support from research grant DOE ASCR CM4 de-sc0009254 and NSF DMS - 1616353.