Bayesian Analysis (0000) **00**, Number 0, pp. 1

Bayesian Inference over the Stiefel Manifold via the Givens Representation

Arya A. Pourzanjani*, Richard M. Jiang*, Brian Mitchell*, Paul J. Atzberger† and Linda R. Petzold *

Abstract. We introduce a routine and flexible approach to posterior inference for statistical models with orthogonal matrix parameters that is based on the Givens representation. While many common statistical models such as factor models and probabilistic principal component analysis (PPCA) are often parameterized in terms of orthogonal matrices, existing approaches to inference for these models can be insufficiently general or challenging to implement. By appealing to the Givens representation, densities over the Stiefel manifold can be transformed into densities over Euclidean space. This allows for the application of tools such as Stan, Edward, or PyMC3 to models with orthogonal matrix parameters. We address several of the challenges to using this approach in practice. We show how to inexpensively compute the change-of-measure term necessary for the density transformation. We also describe two topological pathologies that arise when mapping densities over the Stiefel manifold to densities over Euclidean space. We introduce an approach using auxiliary parameters to overcome the first pathology, and we show that the second pathology has minimal impact on inference using an analytic argument and numerical examples. In addition, we discuss how the alternative parameterization introduced by the Givens representation can be used to define new distributions over the space of orthogonal matrices. We illustrate our approach on several practical examples using Stan, and we compare to existing methods.

MSC 2010 subject classifications: Primary 60K35, 60K35; secondary 60K35.

Keywords: sample, LaTeX 2ε.

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).

*Computer Science Department, University of California, Santa Barbara arya@ucsb.edu
†Mathematics Department, University of California, Santa Barbara

 DOI: 0000

Despite their ubiquity, routine and flexible posterior inference in Bayesian models with orthogonal matrix parameters remains a challenge. Given a user-specified posterior density function, modern probabilistic programming frameworks, such as Stan, Edward, and PyMC3 (Carpenter et al., 2016; Tran et al., 2016; Salvatier et al., 2016) can automatically generate samples from the associated posterior distribution with no further input from the user. Unfortunately, none of these frameworks currently offer support for density functions specified in terms of orthogonal matrices. While methods specifically tailored to handle such densities have been introduced, these existing methods are either insufficiently general or too complicated to implement and tune in isolation, as we discuss in the next section.

One appealing approach to this problem is to parameterize the space of orthogonal matrices in terms of unconstrained Euclidean parameters, then use this parameterization to transform a posterior density on the space of orthogonal matrices to a posterior density on Euclidean space. The resulting density could then be used to sample from the posterior distribution using a probabilistic programming framework such as Stan. However, 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) that are used in software like Stan typically require densities that are continuous and differentiable. Furthermore, any such transformation of a random variable usually requires computing a change-of-measure adjustment term that is often unknown or expensive to compute. Another complication is that the Stiefel manifold has a fundamentally different topology than Euclidean space, which causes pathologies when transforming densities between the two spaces. 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 an approach to posterior inference for statistical models with orthogonal matrix parameters that uses the Givens representation of orthogonal matrices and addresses several of the concerns of using a transformation-based approach. We derive a simple analytic form for the change-of-measure adjustment term that is efficient to compute and does not require computing expensive matrix products or determinants. We also describe and address two topological issues that occur near the boundaries of parameter space when using the Givens representation to transform densities. We introduce a parameter expansion scheme to resolve the first issue, which relates to multi-modalities that may be introduced when transforming a density using the Givens representation. We also discuss the second issue which relates to how the Givens representation pathologically transforms densities near "poles" of the Stiefel manifold, and we show both theoretically and practically how this does not pose a problem in practice. In addition, we briefly discuss how the alternative parameterization introduced by the Givens representation can be used to define new distributions over the Stiefel manifold. Our approach enables the application of standard sampling methods such as the No-U-Turn Sampler (NUTS) (Hoffman and Gelman, 2014) to arbitrary densities specified

in terms of orthogonal matrices. 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 common software frameworks such as Stan, Edward, or PyMC3.

In Section 2, we discuss existing methods for Bayesian inference over the Stiefel manifold and the difficulty in implementing these methods in existing Bayesian software frameworks. 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 how to practically apply the Givens representation to sample from posterior densities that are specified in terms of orthogonal matrices 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

While several methods have been proposed for sampling distributions over the Stiefel manifold, most are either insufficiently general in the types of distributions they can sample, or they rely on algorithmic modifications to inference methods which makes them difficult to use in a routine manner.

Hoff (2009) introduces a Gibbs sampling approach to sample densities specified in terms of orthogonal matrices when the orthogonal matrix parameter, conditioned on all other model parameters, follows a distribution in the Bingham-von Mises-Fisher family. In practice, this limits the flexibility of the approach to a specific class of models and may not offer the same sampling efficiency as modern algorithms such as HMC.

More recently, several authors have proposed HMC-based methods for sampling distributions over the Stiefel manifold, but these methods often rely on specialized HMC update rules, making them difficult to implement and tune in practice. In particular, these methods sample distributions with orthogonal matrix parameters by using different HMC update rules for constrained and unconstrained parameters, requiring additional book-keeping in software 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. Without the automatic tuning of inference algorithm parameters offered in common probabilistic programming frameworks, these algorithms must be manually tuned in practice as they usually do not offer an algorithm to choose tuning parameters automatically. Furthermore, the specialization of these methods to HMC makes them difficult to generalize to other inference algorithms based on variational inference (VI) or optimization, which would be more straight-forward with a transformation based approach.

One example of such a method is by Brubaker et al. (2012) who propose a modified HMC algorithm 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 function 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 Geodesic Monte Carlo (GMC) algorithm. Like the method of Brubaker et al. (2012), the use of separate update rules in GMC makes the algorithm difficult to implement in more general settings.

More recently, the work of Jauch et al. (2018), tries to address these difficulties by using a transformation-based approach based on the Cayley Transform. While the work is still under development, to the best of our knowledge, it is unclear whether their approach provides an efficient way to compute the change-of-measure term without relying on matrix products or determinants or a way to address the topological difficulties of mapping the Stiefel manifold to Euclidean space.

3 The Givens Representation of Orthogonal Matrices

In this section, we describe the Givens reduction algorithm of numerical analysis, then describe the connection between the algorithm and the geometric aspects of the Stiefel manifold. Finally, we describe the Givens representation of orthogonal matrices.

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 $A = QR$. As we explain, the algorithm works by successively applying a series of Givens rotation matrices so as to "zero-out" the elements $\{A_{ij} : i > j\}$ of A. These Givens rotation matrices 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 jth element, u_j, of a vector u, by first using the arctan function to find the angle, θ_{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).

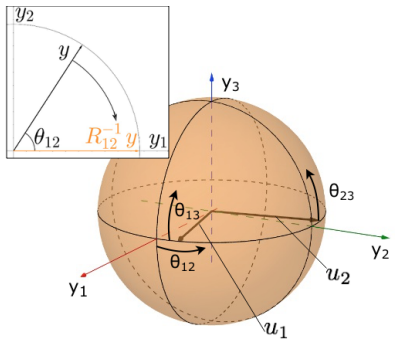

Figure 1: (Inset) Givens rotations can be used to rotate a vector so as to eliminate its component in a certain direction. (Main Figure) A p-frame on the Stiefel manifold can be visualized as a set of rigidly connected orthogonal basis vectors, u_1 and u_2, shown here in black. One can move about the 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 together in the (1,2)-plane, which by our convention is the (x, y)-plane. Similarly, simultaneously applying $R_{13}(\theta_{13})$ 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.

In the Givens reduction algorithm, these rotation matrices are applied one-by-one to A in this way to zero-out all elements below the (i, i) elements of the matrix. 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 jth 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 first 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

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

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

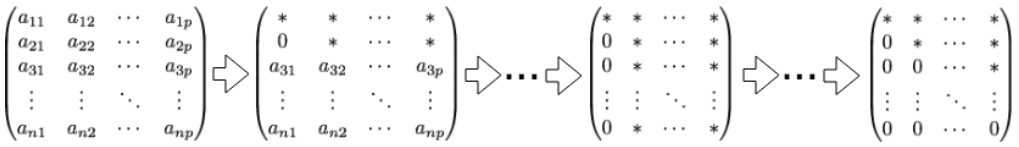

Figure 2: The Givens reduction eliminates lower diagonal elements of an $n \times p$ matrix one column at a time. Because each rotation, $R_{ij}(\theta_{ij})$, only affects rows i and j, previously zeroed out elements do not change.

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

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.

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.

3.2 The Geometry of Orthogonal Matrices

The elements of the Stiefel manifold, $V_{p,n}$, are known as p-frames. A p-frame is an orthogonal set 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

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

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 with our 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, θ_{12} and θ_{13}, that represent how much to rotate the first vector, and a third angle, θ_{23} that controls how much the second vector is rotated about the first (Figure 1). Mathematically this can be represented as the 3×2 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 $I_{n,p}$ 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. Therefore, because the second column must be zero below its second element, it must be e_2 after these $n - 2$ rotations are applied. 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 Θ (Shepard et al., 2015), and lines up with our geometric insight discussed in the previous subsection. We also note that the number of angles in the Givens representation corresponds exactly to the inherent dimensionality, d, of the Stiefel manifold.

4 Using the Givens Representation to Sample Distributions Over the Stiefel Manifold

Using the Givens representation in practice to sample distributions over the Stiefel manifold requires solving several practical challenges. In addition to the standard change-of-measure term required in any transformation of a random variable, care must be taken to address certain pathologies that occur due to the differing topologies of the Stiefel manifold and Euclidean space. We further describe these challenges and how we overcome them in practice. We also briefly remark on how the Givens representation can be leveraged to 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 GMC.

4.1 Transformation of Measure Under the Givens Representation

As is usual in any transformation of random variables, care must be taken to include an extra term in the transformed density to account for a change-of-measure under the transformation. Formally, 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 term accounts for how probability measures are distorted by the transformation (Figure 3). Usually this term is calculated by simply taking the determinant of the Jacobian of the transformation. 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. Thus its Jacobian is non-square and the determinant of the Jacobian is undefined.

One way to compute the change-of-measure term analogous to the Jacobian determinant, is to appeal to the algebra of differential forms. Denote the product of $n \times n$ rotation matrices in the Givens representation by G, i.e.

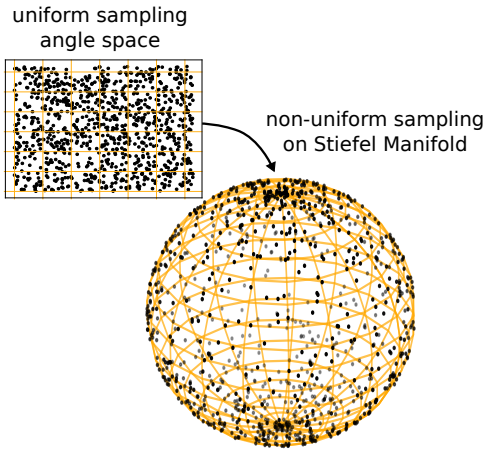

Figure 3: Uniform sampling in the Givens representation coordinates does not necessarily lead to uniform sampling over the Stiefel manifold without the proper measure adjustment term. Under the mapping, regions near the pole are shrunk to regions on the sphere with little area, as opposed to regions 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.

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

and denote its jth 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 ith 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, which would otherwise be expensive to compute, can be analytically simplified to the following simple-to-compute product whose absolute value serves as our measure adjustment term:

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

4.2 Implementation of Angle Coordinates

Two issues related to the topology of the Stiefel manifold arise when using the Givens representation to map densities over the Stiefel manifold to densities over Euclidean space. We let $\theta_{12}, \theta_{23}, \cdots \theta_{p,p+1}$ range from $-\pi$ to π and we refer to these specific coordinates as the latitudinal 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. Formally, this choice of intervals defines a coordinate chart from Euclidean space to the Stiefel manifold, i.e. a mapping between the two spaces. Although this choice of intervals allows us to represent the entire Stiefel manifold in the Givens representation, because the topology of these two space differ, certain connectedness properties of the Stiefel manifold can not be accurately represented in the Givens representation. For example, when representing $V_{1,3}$ in Euclidean space using the Givens representation, contiguous regions of the manifold on either side of the sliver corresponding to $\theta_{12} = \pi$ are disconnected (Figure 4). As we show, this can lead to samples that are far from representative of the correct distribution when applying sampling methods such as HMC to sample densities over the Stiefel manifold using the Givens representation. To address this, we introduce auxiliary parameters to the Givens representation to better represent the connectedness of the Stiefel manifold in Euclidean space.

In addition to these disconnected regions, the coordinate chart will also contain singularities where the measure adjustment term (Equation 4.5) approaches zero near the "poles" of the Stiefel manifold i.e. where the longitudinal coordinates equal zero. This means that a density over the Stiefel manifold that is transformed to a density over Euclidean space using the Givens representation will equal zero at the poles of the Stiefel manifold regardless of whether the original density assigns zero to those points. On the sphere, this happens at the North and South poles where the longitudinal coordinates become exactly $-\pi/2$ or $\pi/2$ (Figure 4). In practice, this prevents algorithms such as HMC from obtaining samples in a small region near these points even when there is positive probability mass in these regions under the original density. The reason is that the

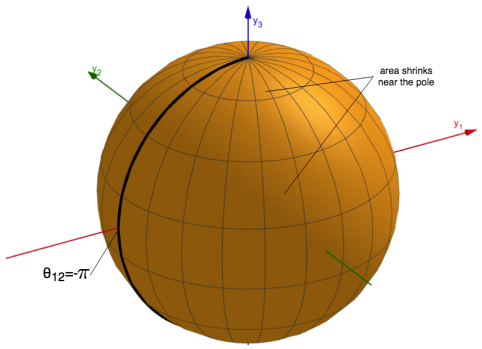

Figure 4: The angular coordinate chart has an infinitesimal sliver of measure zero lying at $\theta_{12} = -\pi = \pi$ that separates two otherwise connected parts of the sphere. Trajectories, $Y(t)$, over the Stiefel manifold that cross this sliver have no equivalent representation, $\Theta(t)$ in the coordinates of the Givens representation. This can become particularly problematic when there is significant probability mass on both sides of the sliver. The grid over the sphere reveals how the Givens representation maps areas that are the same size in the Θ coordinates to smaller and smaller regions on the sphere the closer they are to the poles. Thus the measure adjustment term (Equation 4.5), which measures how the transform changes the area of these infinitesimal regions, goes to zero near the poles, making the transformed density at these points zero regardless of whether they were assigned zero by the original density.

acceptance ratio used by algorithms such as HMC will always equal zero at these points for finite densities. Therefore proposals at these points and will always be rejected. Because of finite numerical precision, this also holds true for points on the Stiefel manifold that are numerically near these poles. Fortunately, as we illustrate both theoretically and experimentally, this generally has a negligible effect on numerical expectations of samples obtained by using the Givens representation in practice.

Introducing Auxiliary Parameters to Address Connectedness

Simply allowing the latitudinal coordinates of the Givens representation to range from $-\pi$ to π leaves regions of parameter space that should otherwise be connected, disconnected in the Givens representation. For a distribution over the Stiefel manifold that is not sufficiently concentrated away from these disconnected regions, this can lead to highly non-representative samples when naively applying the Givens representation to sample (Figure 5, upper).

To handle this multimodality introduced by reparameterizing in terms of Givens angles, we introduce for each angle parameter, θ_{ij}, an independent auxiliary parameter, r_{ij}. We then transform the density to sample over the x_{ij}, y_{ij}-space via the transform $x_{ij} = r_{ij} \cos \theta_{ij}$ and $y_{ij} = r_{ij} \sin \theta_{ij}$. In the transformed space the two ends of the inter-

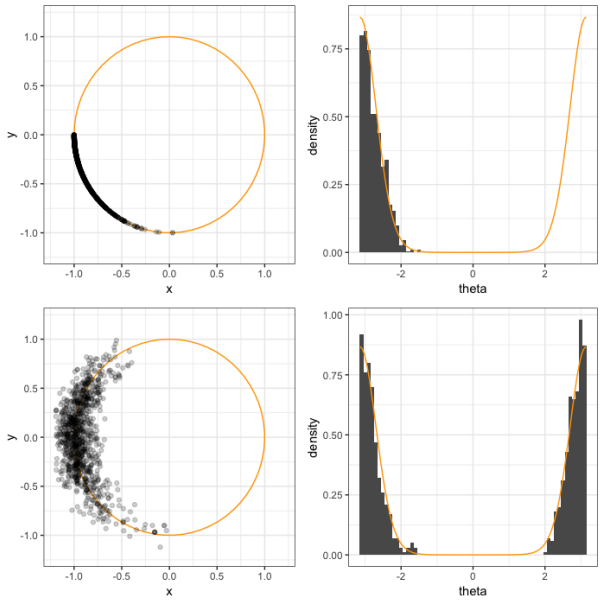

Figure 5: (Upper) 1,000 samples from a Von Mises distribution with parameters $\mu = -\pi$ and $\kappa = 5$ sampled over the space $\theta \in [-\pi, \pi]$ using Stan. Most of the mass of the distribution is concentrated at the ends of the interval while little mass is concentrated towards the middle. Because these two ends of the interval are disconnected in this representation, the sampler gets "stuck" in the mode corresponding to the $-\pi$ side of the interval rather than the π side. (Lower) 1,000 samples from the equivalent distribution sampled over the (x, y)-space. By introducing an auxiliary coordinate, one can effectively replicate the topology of a circle, effectively "wrapping" the two ends of the interval so that the sampler avoids getting stuck in one region.

val are connected, producing samples that are distributed more evenly across the two disconnected regions (Figure 5, lower). Formally, we assign to r a marginal distribution with density $p_r(r)$ so that θ and r are independent and the marginal distribution of θ is left untouched by the introduction of the auxiliary parameter. This leads to the joint density $p_{\theta,r}(\theta, r) = p_\theta(\theta)p_r(r)$ which we then transform to the equivalent density over the unconstrained (x, y)-space by the simple change-of-variables formula between two-dimensional polar coordinates and two-dimensional Euclidean coordinates:

$$p_{x,y}(x, y) \quad = \quad p_{\theta,r}(\arctan\left(y/x\right), \sqrt{x^2 + y^2}\right) \frac{1}{r}. \tag{4.6}$$

This again leaves the marginal distribution of θ unaffected, however, in the new space, paths $\theta(t)$ that cross the region of parameter space at $\theta = \pi$ can actually be represented.

In practice, we set $p_r(r)$ to a normal density with mean one and standard deviation 0.1. Although r_{ij} does not necessarily need to be set to this particular distribution to achieve the correct marginal distribution over θ, this choice helps avoid the region of parameter space where $r = 0$ and the transformed density is ill-defined.

Quantifying Pathologies of the Transformed Density Near Poles

Even with the usual change of variables formula and the measure adjustment term (Equation 4.5), a density over the Stiefel manifold that is transformed to a density over Euclidean space using the Givens representation will not be completely equivalent to the original density. In particular, when any of the longitudinal angles has absolute value equal to $\pi/2$, Equation 4.5 will equal zero. Thus the transformed density will equal zero at these points even when the original density is non-zero. Because of finite numerical precision, in practice, this creates a region of the Stiefel manifold that can not be sampled by algorithms such as HMC despite having positive a positive probability mass under the original distribution. Specifically, because a computed numerical density will equal zero at values numerically near the poles, the acceptance ratio in HMC will always equal zero at these points so that proposals in the region will always be rejected. This effectively blocks off a portion of parameter space by limiting all longitudinal angles to the region $[-\pi/2 + \epsilon, \pi/2 - \epsilon]$ where ϵ is a small value on the order of numerical precision. Despite this concern, as we show both theoretically and experimentally, the exceedingly small volume of this region justifies the use of the Givens representation in practice.

We show that for general n and p, the volume of this blocked off region is $\mathcal{O}(p\epsilon^2)$. First note that the uniform density over $V_{p,n}$ in the Givens representation is simply a constant times the absolute value of Equation 4.5. However, since this density factors into a product of independent terms, the probability that at least one longitudinal angle falls within the ϵ-region is simply the sum of the individual probability of each angle falling within the region. Each of these individual probabilities is proportional to $\cos^{j-i-1}\theta_{ij}$, which for small ϵ can be bounded by ϵ^{j-i-1} over the interval $[\pi/2-\epsilon, \pi/2]$. Thus the probability of falling within the ϵ-region is bounded by a constant times the following quantity:

$$\sum_{i=1}^{p}\sum_{j=i+2}^{n} 2\int_{\pi/2-\epsilon}^{\pi/2} \epsilon^{j-i-1} d\theta_{ij} = \sum_{i=1}^{p}\sum_{j=i+2}^{n} 2\epsilon^{j-i} = \sum_{i=1}^{p} \mathcal{O}(\epsilon^2) = \mathcal{O}(p\epsilon^2). \tag{4.7}$$

Because this quantity falls off with the square of ϵ, even for modestly small ϵ, the probability of a uniformly sampled point falling within the ϵ-region is small. Empirical results further illustrate this. For various values of n, p, and ϵ we drew 100,000 samples uniformly form the Stiefel manifold by sampling the elements of an $n \times p$ matrix from a standard normal distribution, then taking the QR factorization of this matrix, a common technique for uniformly sampling the Stiefel manifold (Muirhead, 2009). We then took these samples, converted them into their Givens representation, and calculated the

p	n	$\epsilon = 0.1$	$\epsilon = 0.05$	$\epsilon = 0.025$	$\epsilon = 0.0125$	$\epsilon = 1e-5$
1	10	490	114	22	4	0
1	20	499	118	25	4	0
1	50	570	148	32	6	0
3	10	1,612	381	79	15	0
3	20	1,665	398	78	19	0
3	50	1,712	416	100	24	0
10	10	4,260	1,071	258	59	0
10	20	5,342	1,336	357	91	0
10	50	5,266	1,368	334	90	0

Table 1: The number of uniform samples out of 100,000 that fell within the ϵ region for various values of n, p, and ϵ. Samples are taken uniformly from the Stiefel manifold using the QR factorization method. As the theoretical bound suggests, the number of samples falling in this region increases negligibly for fixed p and increasing n, it increases linearly with p, and it decreases quadratically as ϵ decreases. In particular, whenever ϵ is halved, the number of sample falling within the region decrease by about a fourth. We also note that for $\epsilon = 1e-5$, the value we used for most of our experiments, the number of samples falling within the ϵ region is zero for all settings.

number of samples that had any longitudinal angle within the region $[-\pi/2, -\pi/2+\epsilon]$ or the region $[\pi/2-\epsilon, \pi/2]$. The results are closely explained by Equation 4.7. In particular, the proportion of samples that fell within this region does not change much for fixed p and increasing n. Furthermore, the proportion increases linearly with p, and it decreases quadratically with ϵ (Table 1).

For non-uniform distributions with a probability density $p(\Theta)$ that is finite in the ϵ-region, the probability of any of the longitudinal angles falling within the ϵ region can again be bounded by a constant times $\mathcal{O}(p\epsilon^2)$. We took 100,000 samples from the von Mises Fisher distribution over $V_{1,3}$ with parameters $\mu = (0, 0, 1)$ and $\kappa = 1, 10, 100$, and 1000 using the simulation method of Wood (1994) as implemented in the R package Rfast. For fixed κ the probability of a sample falling in the ϵ-region drops off with the square of ϵ as the bound would suggest. This holds true even when probability mass is highly concentrated near these regions (Table 2).

Because the probability of samples falling within the ϵ falls with the square of ϵ, even for modestly small ϵ, the distribution of derived quantities of $Y(\Theta)$ remains largely unaffected when sampling using the Givens representation with small enough ϵ. We sampled the von Mises Fisher distribution with the same values of κ using the Givens representation in Stan but with the longitudinal angles deliberately limited to the interval $[-\pi/2 + \epsilon, -\pi/2 + \epsilon]$ with $\epsilon = 0.1, 0.05, 0.025, 0.0125$, and $1e-5$. We then examined histograms and expectations of the principal angle $\arccos(\mu^T Y)$ which represents the angle between the sample and the direction at the pole which lies directly in the middle of the ϵ-region. For large ϵ and large κ the lack of samples from the ϵ-region is evident when compared to samples using the method of Wood (1994) since the sample can not get close enough to μ. However, for any fixed κ as ϵ is decreased, the number of samples that fall within the ϵ-region decreases rapidly as the bound would suggest (Figure 6).

κ	$\epsilon = 0.1$	$\epsilon = 0.05$	$\epsilon = 0.025$	$\epsilon = 0.0125$	$\epsilon = 1e-5$
1	630	163	38	9	0
10	4,839	1,220	317	81	0
100	39,287	11,658	3,086	764	0
1,000	99,295	71,066	26,643	7,473	0

Table 2: The number of samples from a von Mises Fisher distribution with $\mu = (0, 0, 1)$ and $\kappa = 1, 10, 100$ and 1000 that fell within the ϵ region for various values of n, p, and ϵ. For each value of κ 100,000 total samples were taken. As the theoretical bound suggests, the number of samples that fill within the ϵ region fall with the square of ϵ so that even for a modestly small value of $\epsilon = 1e-5$, none of the 100,000 samples fall within this region even in the highly concentrated case ($\kappa = 1,000$).

	Givens					Wood
κ	$\epsilon = 0.1$	$\epsilon = 0.05$	$\epsilon = 0.025$	$\epsilon = 0.0125$	$\epsilon = 1e-5$	
1	1.2027	1.2042	1.2008	1.1995	1.1986	1.2012
10	0.4181	0.4065	0.4031	0.4012	0.4019	0.4015
100	0.1657	0.1377	0.1290	0.1258	0.1261	0.1255
1,000	0.1092	0.0657	0.0483	0.0422	0.0396	0.0398

Table 3: The empirical expectation of the principal angle, $\arccos(\mu^T Y)$, sampled under the von Mises Fisher distribution with $\mu = (0, 0, 1)$ and $\kappa = 1, 10, 100$ and 1000 using the Givens representation in Stan with various sizes of the ϵ area and using the method of Wood (1994). As ϵ decreases, the empirical expectation computed using the Givens representation become much closer to those taken via the method of Wood (1994). For small κ the expectations do not differ much even for large ϵ because much less mass concentrates near the ϵ regions.

Table 3 illustrates this effect numerically using the expectation of the principal angle.

4.3 New Distributions Using the Givens Representation

So far, we have focused on transforming densities defined in terms of the canonical orthogonal matrix coordinates, Y, into densities specified in terms of the angles, Θ, of the Givens representation. However, the angles of the Givens representation can also be used to define new distributions over the Stiefel manifold that may be useful in modeling. In fact, using the intuition described in Section 3.2, the sign and magnitude of the angle θ_{ij} of the Givens representation, roughly corresponds to the sign and magnitude of the $i - j$ element of Y. Thus one can create, for example, sparse priors over the Stiefel manifold by placing sparse priors over the angles of the Givens representation.

Cron and West (2016) utilize sparsity promoting priors over the coordinates of the Givens representation to produce a prior distribution over covariance matrices that favors sparse matrices. Specifically, they describe a model for multivariate Gaussian observations with an unknown covariance matrix. They parameterize the covariance matrix in terms of its eigen-decomposition $Y \Lambda Y^T$ then parameterize the orthogonal matrix Y

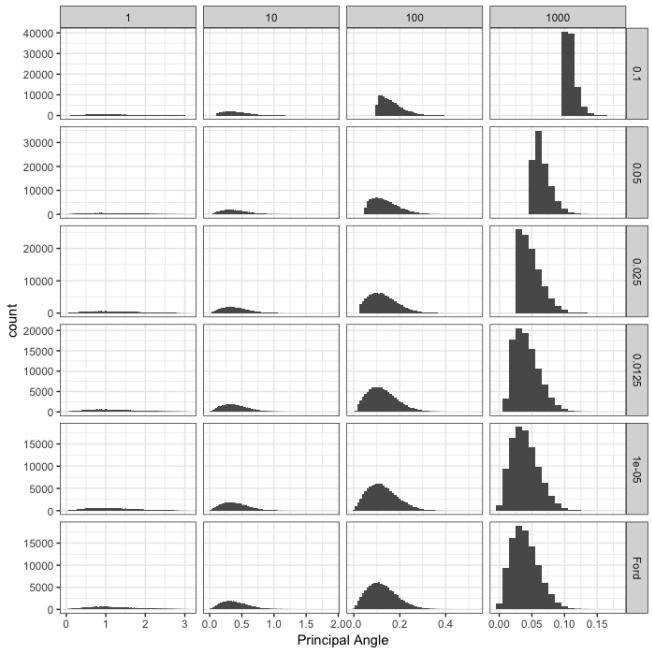

Figure 6: Histograms of the principal angle, $\arccos(\mu^T Y)$, sampled under the von Mises Fisher distribution with $\mu = (0, 0, 1)$ and $\kappa = 1, 10, 100$ and 1000 using the Givens representation in Stan with various sizes of the ϵ area and using the method of Wood (1994). For small ϵ and large κ, the lack of samples from the ϵ-region in the latter method is evident in the histograms. In particular, despite the large amount of mass near zero, the samples are never higher than a bound higher than zero that is dictated by ϵ. As ϵ decreases, the bound rapidly becomes negligible because of the quadratic relationship between ϵ and the volume of the ϵ area. Thus for $\epsilon = 1e - 5$, the histograms of the Givens representation method and the Wood (1994) method become indistinguishable.

using the Givens representation. They then place spike-and-slab mixture priors over the angles of the Givens representation, placing significant prior mass on orthogonal matrices whose Givens angles are mostly zero and thus sparse in the canonical coordinates. They describe a custom reversible jump-based method for sampling the resulting posterior distribution.

Our Givens representation approach provides a routine and flexible way to sample the posterior distribution associated with this model and other more general models using common probabilistic modeling frameworks. In Section 5, we illustrate this with a sparse PPCA example motivated by Cron and West (2016) that places truncated horseshoe priors on the angles of the Givens representation.

4.4 Computational Scaling of the Givens Representation

The primary computational cost in computing 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, GMC involves an orthogonalization of an $n \times p$ matrix which scales as $\mathcal{O}(np^2)$ and a matrix exponential computation that scales as $\mathcal{O}(p^3)$.

Input: θ
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.

We note, however, that this comparison is somewhat obfuscated by the gradient of the log probability computation that is required by both methods. When using the Givens representation in a framework such as Stan, this gradient is computed internally using backward automatic differentiation. Meanwhile, GMC also requires computing gradients of the log density, but if these are analytically known, then this step will usually be faster in GMC. However, when backward automatic differentiation must be used in GMC because the gradients are not analytically known, computing gradients may actually be faster when the density is specified in terms of the Givens representation. This is because backward automatic differentiation of a scalar function such as log probability typically scales with the number of inputs to the function. In this case, the inputs correspond to the unknown parameters of the model. The Givens representation has d inputs (one for each angle not including auxiliary parameters) which is fewer than the np inputs that come with using the canonical matrix representation. Including the auxiliary parameters adds an additional p inputs (one for each latitudinal angle).

In practice, sampling efficiency will depend on several factors, including the size of the orthogonal matrix being sampled, making it difficult to generally recommend one method over the other in terms of efficiency. For uniform sampling of the Stiefel manifold, we find that GMC scales better when p is much smaller than n, whereas the

		GMC		Givens	
p	n	$\hat{R}$	n_{eff}	$\hat{R}$	n_{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 4: $\hat{R}$ and n_{eff} values averaged over all elements of the matrix parameter Y.

Givens representation scales better when p is large and closer to n. We present benchmarks comparing the two methods on orthogonal matrices of various sizes in Section 5.

5 Experiments

We demonstrate the use of the Givens representation for uniformly sampling the Stiefel manifold as well as several statistical examples. 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. Presence of divergences suggests that the sampler may not be visiting areas of the posterior distribution that contain positive mass (Betancourt and Girolami, 2015), while $\hat{R}$ tests for convergence of the Markov chain to the stationary distribution (Gelman et al., 1992). 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 GMC using 500 post-warmup samples from each method (Table 4). We chose the step size tuning parameter of GMC by manually trying several possible values, then selecting the specific value that produced the highest number of effective samples per second over 500 samples.

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 density. Meanwhile, uniform sampling over the Stiefel manifold is achieved in GMC simply using a constant density because the method uses the canonical matrix coordinates. However, as mentioned in section 4.4, this comes at the cost of an expensive HMC update to ensure the updated parameter still satisfies the constraints. In practice, we find that GMC scales better as n is increased, although the approach using the Givens representation in Stan remains competitive (Figure 7).

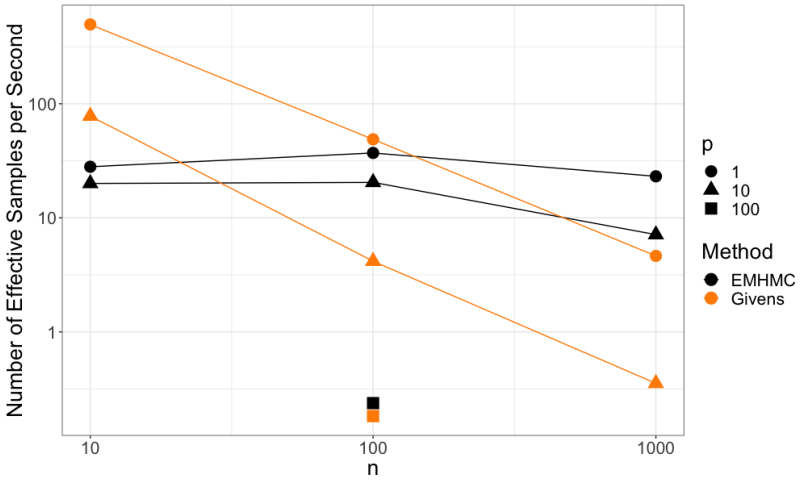

Figure 7: For small values of n the Givens representation approach in Stan produces more effective samplers per second, while for larger values the GMC scales better since the primary cost of the matrix exponential remains constant.

5.2 PPCA

Factor Analysis (FA) and 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, the orthogonal matrix corresponding to this linear function can be thought of as a 2-frame that aligns with this cloud (Figure 8). 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 Λ is a diagonal matrix with positive, ordered elements. Because $\mathbf{x}_i$ is a linear transformation of a multivariate Gaussian, $\mathbf{z}_i$ can be integrated out of the model 5.1 to yield the simplified formulation

$$\mathbf{x}_i | W, \Lambda, \sigma^2 \sim \mathcal{N}_n(0, \mathbf{C}). \tag{5.2}$$

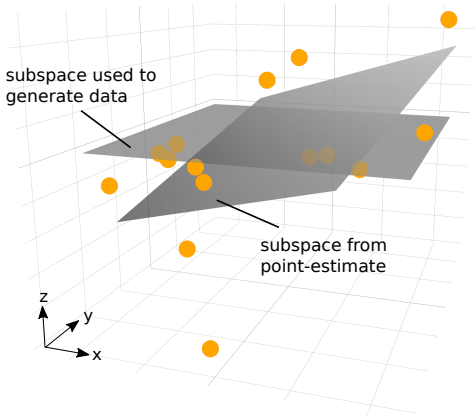

Figure 8: The maximum-a-posteriori (MAP) estimate of PPCA corresponds to a single orthogonal matrix in the Stiefel Manifold that is closest, in terms of average squared distance, to the set of points. When there are few data points relative to the size of the matrix, this point estimate can often have high variance.

where $\mathbf{C} := W\Lambda^2 W^T + \sigma^2 \, \mathrm{I}$ (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

$$p(\mathbf{x}_1, \cdots, \mathbf{x}_N | W, \Lambda, \sigma^2) = -\frac{N}{2} \ln |\mathbf{C}| - \frac{1}{2} \sum_{i=1}^{N} \mathbf{x}_i^T \mathbf{C}^{-1} \mathbf{x}_i \tag{5.3}$$

$$= -\frac{N}{2} \ln |\mathbf{C}| - \frac{N}{2} \mathrm{tr}(\mathbf{C}^{-1} \hat{\Sigma}). \tag{5.4}$$

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. Sampling the posterior of a model both provides a measure of uncertainty for parameter estimates and is possible even for more elaborate models.

We used the Givens representation in Stan to sample the posterior distribution of the parameters in model 5.2 from a simulated dataset with $n = 50$ and $p = 3$. For Λ and σ^2 we chose uniform priors over the positive real line and for W we chose a uniform prior over the Stiefel manifold yielding the unnormalized posterior density

$$p(W, \Lambda, \sigma^2 | \mathbf{x}_1, \cdots, \mathbf{x}_N) \propto p(\mathbf{x}_1, \cdots, \mathbf{x}_N | W, \Lambda, \sigma^2) \tag{5.5}$$

or in the Givens representation

Parameter	2.5%	25%	50%	75%	97.5%	$\hat{R}$	n_{eff}
Λ_1^2	3.98	4.87	5.46	6.14	7.74	1.0	3,313
Λ_2^2	2.48	3.16	3.61	4.09	5.10	1.0	848
Λ_2^2	1.21	1.76	2.07	2.44	3.20	1.0	1,340
σ^2	0.99	1.01	1.03	1.04	1.07	1.0	5,374

Table 5: Posterior quantiles, $\hat{R}$, and n_{eff} values for Λ^2 and σ^2 computed over 10,000 posterior draws.

$$p(\Theta, \Lambda, \sigma^2 | \mathbf{x}_1, \cdots, \mathbf{x}_N) \propto p(\mathbf{x}_1, \cdots, \mathbf{x}_N | W(\Theta), \Lambda, \sigma^2) \, |J_{Y(\Theta)}(\Theta)| \tag{5.6}$$

where the latter term comes from Equation 4.5.

For the true value of the parameters we used the settings used by Jauch et al. (2018) in their experiments section: $\Lambda^2 = \text{diag}(5, 3, 1.5)$, $\sigma^2 = 1$, and W drawn uniformly from $V_{3,50}$. We took 10,000 samples using Stan's NUTS algorithm with default settings. Table 5 shows the posterior quantiles along with $\hat{R}$ and n_{eff} values for Λ^2 and σ^2. Like Jauch et al. (2018), we plot histograms of the principal angle,

$$\phi_j = \arccos(E_j^T W_j), \; j = 1, 2, 3 \tag{5.7}$$

between the columns, W_j, of posterior draws of W and the columns of the first three eigenvectors of $\hat{\Sigma}$, E_j (Figure 9).

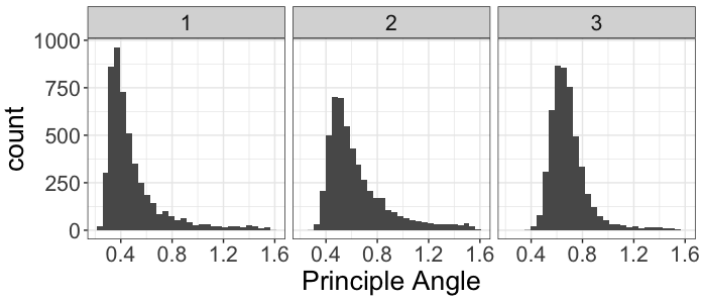

Figure 9: Histograms of the principal angles between posterior samples of W and the first three eigenvectors of $\hat{\Sigma}$.

We also plot the true values of W used in the simulation along with the 80% credible intervals, computed from posterior samples, of the marginal posterior distributions of the elements of W (Figure 10).

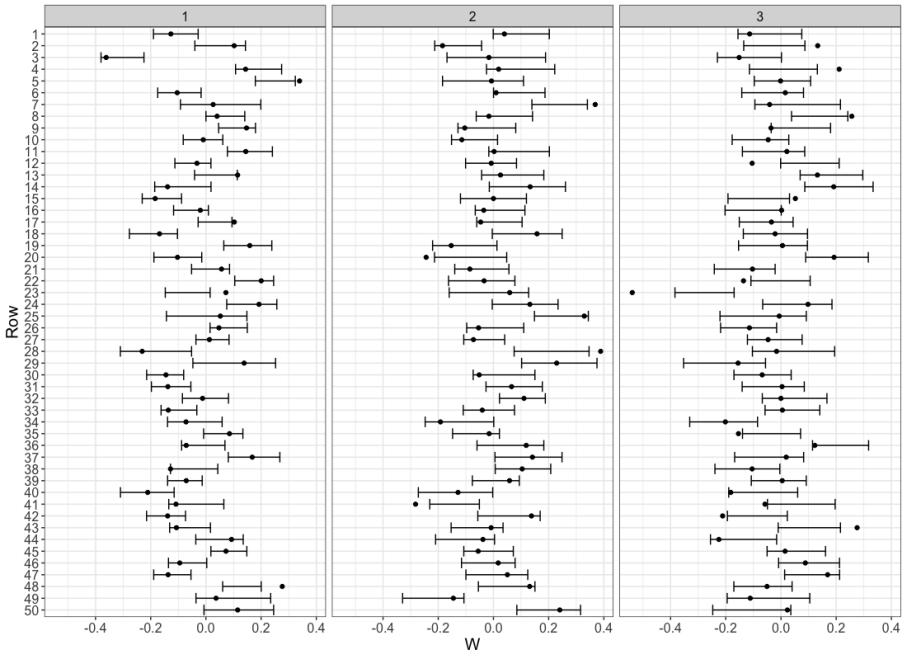

Figure 10: True values of W used in the simulation along with 80% credible intervals computed using draws of the posterior. Each facet corresponds to one of the three columns of W

5.3 Sparse PPCA

To illustrate the utility of placing priors over the angle parameters, Θ, of the Givens representation, we fit a PPCA model with sparse priors over Θ to simulated data generated from 5.1 with the same parameter settings as in the previous example, but with W replaced with a sparse matrix. To generate a sparse orthogonal matrix, we drew an orthogonal matrix uniformly from $V_{3,50}$, converted the result to the Givens representation, randomly set each angle to zero with probability 0.8, then converted the result back to the canonical representation. The result was an orthogonal $V_{3,50}$ matrix, W, with 85% of its elements equal to zero.

For our model, we used the standard PPCA likelihood 5.4 with uniform priors over Λ^2 and σ^2, but rather than placing a prior over Θ was uniform over the Stiefel manifold, we set Θ to follow the regularized horseshoe prior of Piironen et al. (2017), a commonly used sparsity-inducing prior. Formally, we set

$$\theta_{ij} \sim \text{TruncatedNormal}(0, \tau^2\tilde{\lambda}_{ij}^2),\ \tilde{\lambda}_{ij}^2 = \frac{c^2\lambda_{ij}^2}{c^2 + \tau^2\lambda_{ij}^2} \tag{5.8}$$

Model	Parameter	2.5%	25%	50%	75%	97.5%	$\hat{R}$	n_{eff}
Non-Sparse	Λ_1^2	3.69	4.48	4.95	5.52	6.88	1.0	3,938
Non-Sparse	Λ_2^2	2.66	3.38	3.80	4.25	5.16	1.0	1,828
Non-Sparse	Λ_2^2	0.18	0.94	1.27	1.62	2.39	1.0	366
Non-Sparse	σ^2	0.97	1.00	1.01	1.03	1.06	1.0	1,421
Sparse	Λ_1^2	3.71	4.48	4.97	5.58	6.91	1.0	5,425
Sparse	Λ_2^2	2.67	3.31	3.70	4.13	5.01	1.0	4,952
Sparse	Λ_2^2	0.78	1.15	1.38	1.63	2.24	1.0	4,779
Sparse	σ^2	0.97	1.00	1.01	1.03	1.05	1.0	7,191

Table 6: $\hat{R}$ and n_{eff} values averaged over all elements of the matrix parameter Y.

$$\begin{aligned} \lambda_{ij} &\sim \text{HalfCauchy}(0, 1) \\ \tau &\sim \text{HalfCauchy}(0, \tau_0) \\ c^2 &\sim \text{InverseGamma}(\nu/2, \nu s^2/2) \end{aligned}$$

with hyper-parameters set to $\tau_0 = 0.01$, $\nu = 10$, and $s = \pi/4$ following the guidelines of Piironen et al. (2017). We took 10,000 posterior draws in Stan using the resulting unnormalized posterior density

$$p(\Theta, \Lambda, \sigma^2 | \mathbf{x}_1, \cdots, \mathbf{x}_N) \propto p(\mathbf{x}_1, \cdots, \mathbf{x}_N | W(\Theta), \Lambda, \sigma^2)\, p(\Theta, \lambda, \tau, c^2). \tag{5.9}$$

Table 6 shows a posterior summary for the sparse model versus the non-sparse corresponding to the density 5.6. While the marginal posterior distributions of Λ^2 and σ^2 are similar for both models, the sparse model expectedly results in a much sparser posterior distribution over Θ and thus W (Figure 11). In particular, for elements of W that are truly zero, the marginal posterior distributions of the sparse model tends to concentrate much closer to zero, while for truly non-zero elements, the sparse model is able to concentrate posterior mass away from zero.

5.4 The Network Eigenmodel

We used the Givens representation to sample from the posterior distribution of the network eigenmodel of Hoff (2009) which was also illustrated in Byrne and Girolami (2013). We compared the results obtained from using the Givens representation in Stan to results obtained from GMC. The data used in those works and originally described in Butland et al. (2005) consists of a symmetric graph matrix, Y, of dimension 270×270. However, for our experiments we use a smaller 230×230 version of the dataset as we were unable to find access to the larger version. The version we used is freely available in the R package *eigenmodel*. For our GMC experiments we used the same tuning parameters for GMC as Byrne and Girolami (2013).

The graph matrix encodes whether the proteins in a protein network of size $n = 230$ interact with one another. The probability of a connection between all combinations

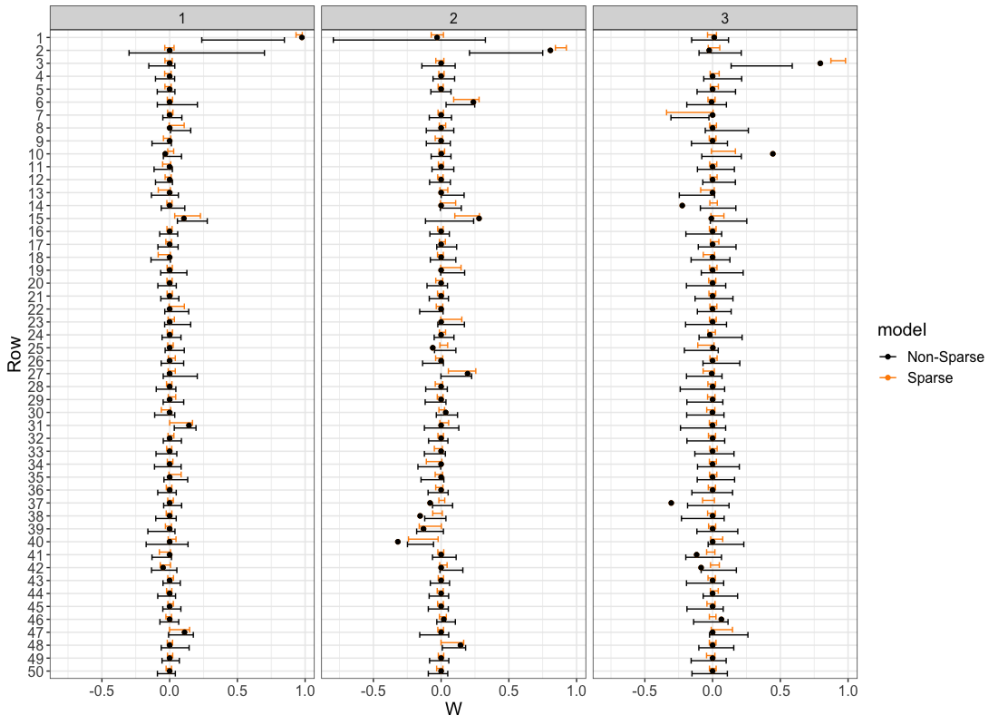

Figure 11: True values of W used in the simulation along with 80% posterior credible intervals computed using 10,000 draws of the posterior from the sparse and non-sparse models, respectively. The posterior distribution of the sparse model places much more posterior mass closer to zero for values that are truly zero while concentrating mass away from zero for truly non-zero values.

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 Λ, and a scalar c, then letting $\Phi(\cdot)$ represent the probit link function, the model is described as follows:

$$\begin{aligned} c &\sim \mathcal{N}(0, 10^2) && (5.10)\\ \Lambda_i &\sim \mathcal{N}(0, n),\ \forall i && (5.11)\\ Y_{ij} &\sim \text{Bernoulli}\left(\Phi([U\Lambda U^T]_{ij} + c)\right),\ \forall i > j. && (5.12) \end{aligned}$$

For U we specified a uniform prior over the Stiefel manifold, which again in the Givens representation corresponds to a prior density that is the absolute value of the change-of-measure term.

The Stan implementation using the Givens representation took approximately 300 seconds to collect 1000 samples, 500 of which were warmup. In contrast, GMC took 812 seconds to run the same 1000 samples using the hyperparameter values specified in Byrne and Girolami (2013). Figure 12 compares traceplots for c, Λ, and the elements of the top row U for the 500 post warmup samples from each sampler. Computed $\hat{R}$ and n_{eff} for these parameters are shown in Table 7.

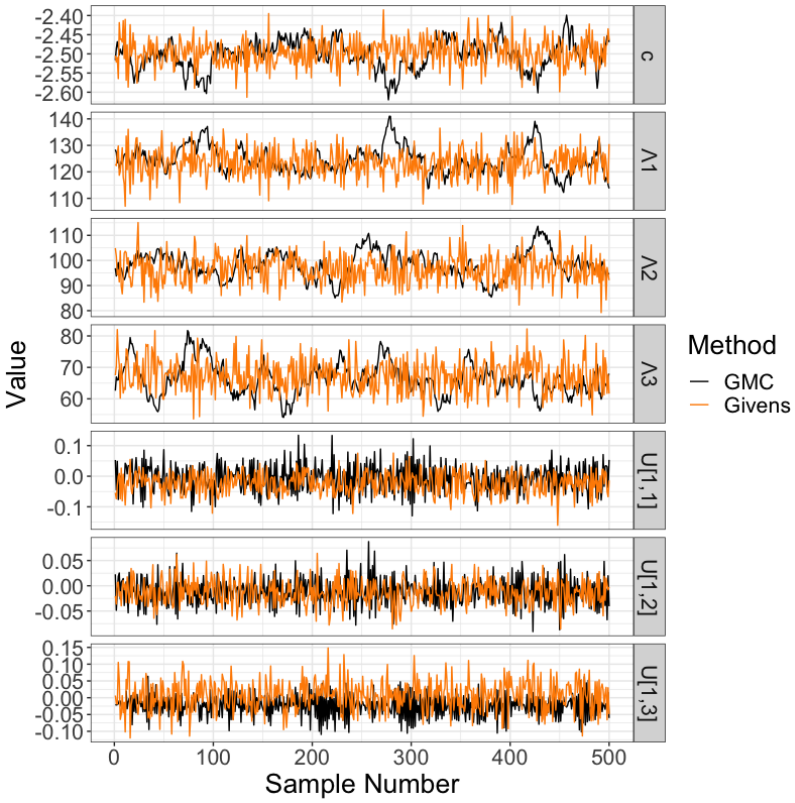

Figure 12: Traceplots of samples from the Givens representation implementation in Stan and GMC. For brevity, only the top three elements of U are shown.

6 Discussion

We introduced an approach based on the Givens representation for sampling distributions specified in terms of orthogonal matrices. The approach is applicable to statistical models with orthogonal matrix parameters. Our approach addresses several of the practical concerns of using a transformation-based approach such as efficient computation of the measure adjustment term used in the density and topological issues that arise

	GMC		Givens	
Parameter	$\hat{R}$	n_{eff}	$\hat{R}$	n_{eff}
c	1.00	22	1.00	496
Λ_1	1.00	19	1.00	500
Λ_2	1.00	23	1.00	500
Λ_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 7: $\hat{R}$ and n_{eff} values for the parameters in the network eigenmodel. For brevity, only three of the matrix parameters are shown.

when mapping the Stiefel manifold to Euclidean space. We also provided an intuitive explanation behind the Givens representation that is accessible to statisticians. Lastly, we illustrated the use of the Givens representation on several practical examples, including a sparse PPCA example that illustrates how the Givens representation can be used to specify new distributions over the Stiefel manifold. We anticipate that one possible future direction of this work may be further exploring distributions over the Stiefel manifold that arise by placing certain priors over the Givens representation.

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

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

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

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

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 $D_{ij} := \partial_{ij} R_{ij}$,

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

(A.22)

which is zero up to the ith 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 $R_i e_i$:

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

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

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

Combining this with Expression A.12 yields finally

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

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.

Betancourt, M. and Girolami, M. (2015). "Hamiltonian Monte Carlo for hierarchical models." *Current trends in Bayesian methodology with applications*, 79: 30.

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.

Gelman, A., Rubin, D. B., et al. (1992). "Inference from iterative simulation using multiple sequences." *Statistical science*, 7(4): 457–472.

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*.

Jauch, M., Hoff, P. D., and Dunson, D. B. (2018). "Random orthogonal matrices and the Cayley transform." *arXiv preprint arXiv:1810.02881*.

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.

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.

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.

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*.

Piironen, J., Vehtari, A., et al. (2017). "Sparsity information and regularization in the horseshoe and other shrinkage priors." *Electronic Journal of Statistics*, 11(2): 5018–5051.

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*.

Wood, A. T. (1994). "Simulation of the von Mises Fisher distribution." *Communications in statistics-simulation and computation*, 23(1): 157–164.

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.