Theory and Computational Methods for Dynamic Projections in High-Dimensional Data Visualization

ANDREAS BUJA[1] DIANNE COOK[2],
DANIEL ASIMOV[3], CATHERINE HURLEY[4]

March 31, 2004

Projections are a common tool for dimension reduction and visualization of high-dimensional data. For example, any projection from $\mathbb{R}^p$ down to a 2-D plane produces a 2-D view of p-D data.

We are concerned with continuous one-parameter families of projections for the purpose of producing *animated views* of data; we therefore call such one-parameter families *dynamic projections*. A special case of dynamic projections are "rotations" of 3-D data. This paper gives a treatment of the visualization, the mathematics and the numerics of dynamic projections.

We start with a discussion of rendering methods for projections of data. We then examine the kinematics of dynamic projections in terms of invariant Riemannian metrics on Stiefel manifolds. We finally describe several algorithms for dynamic projections based on continuous interpolation, the generalization of polylines to the Stiefel manifold. These algorithms lend themselves for the implementation of interactive visual exploration tools of high-dimensional data, such as so-called grand tours, guided tours and manual tours.

1 Introduction

Motion graphics for data analysis have long been almost synonymous with 3-D data rotations. The intuitive appeal of 3-D rotations is due to the power of 3-D perception as well as the natural controls they afford. To perform 3-D rotations, one selects triplets of data variables, spins the resulting 3-D pointclouds, and presents a 2-D projection thereof to the viewer of a computer screen. Human interfaces for controlling

[1]AT&T Labs — Research, Shannon Laboratory, 180 Park Avenue, Florham Park, NJ 07932-0971; andreas@research.att.com, http://www.research.att.com/~andreas/

[2]Dept of Statistics, Iowa State University, Ames, IA 50011; dicook@iastate.edu, http://www.public.iastate.edu/~dicook/

[3]Mathematics Department, University of California, Berkeley, CA 94720; asimov@msri.org.

[4]Mathematics Department, National University of Ireland, Maynooth Co. Kildare, Ireland; churley@maths.may.ie, http://www.maths.may.ie/staff/churley/churley.html

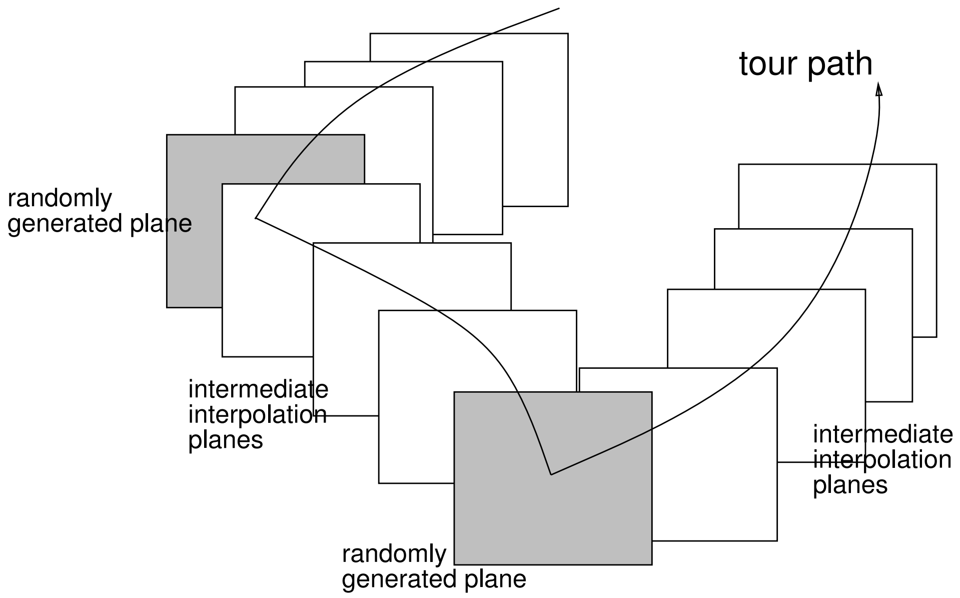
tour path
randomly generated plane
intermediate interpolation planes
randomly generated plane
intermediate interpolation planes

Figure 1: *Schematic depiction of a path of projection planes that interpolates a sequence of randomly generated planes. This scheme is an implementation of a grand tour as used in XGobi (Swayne et al. 1998).*

3-D data rotations follow natural mental models: Thinking of the pointcloud as sitting inside a globe, one would enable the user to rotate the globe around its north-south axis, for example, or to pull an axis into oblique viewing angles, or to push the globe with a sweeping motion of the hand (the mouse, that is) into continuous motion.

The mental model behind these actions proved natural to such an extent that an often asked question became vexing and almost unanswerable: How would one generalize 3-D rotations to higher dimensions? If 3-D space presents the viewer with one hidden backdimension, the viewer of $p > 3$ dimensions would face $p - 2 > 1$ backdimensions and wouldn't know how to use them to move the p-dimensional pointcloud!

Related is the fact that in 3 dimensions we can describe a rotation in terms of an axis around which the rotation takes place, while in higher than 3 dimensions the notion of a rotation axis is generally not useful: If the rotation takes place in a 2-D plane, the "axis" is a $(p-2)$-dimensional subspace of fixed points; but if the rotation is more general, the "axis" of fixed points can be of dimensions $p-4$, $p-6$, $\ldots$, and such "axes" do not determine a unique 1-parameter family of rotations. A similar point was made by J. W. Tukey (1987, section 8).

The apparent complexity raised by these issues was answered in a radical way by Asimov's notion of a grand tour (Asimov 1985): Similar to 3-D data rotations, a grand tour exposes viewers to dynamic low-dimensional data projections, but unlike 3-D rotations, the grand tour gives up user control and presents the viewer with an

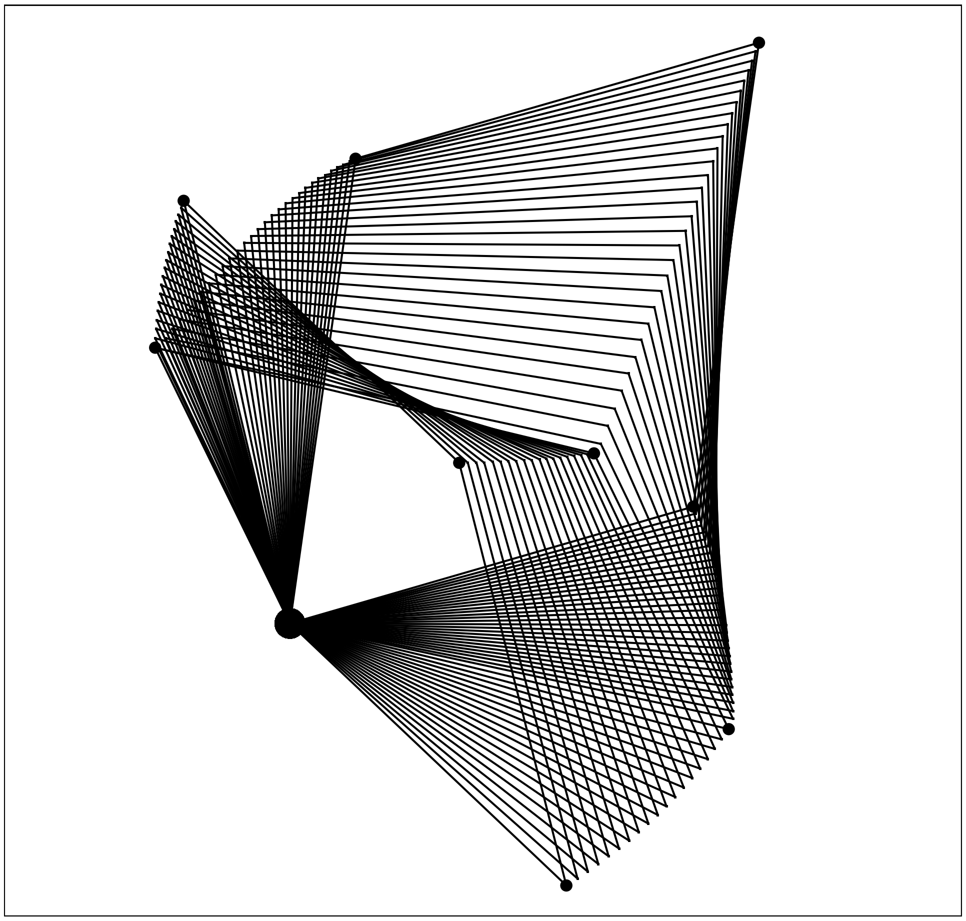

Figure 2: *XGobi looking at XGobi: This figure shows a random projection of a path of projection planes that interpolates three randomly generated projections. The planes are represented as squares spanned by their projection frames. The origin is marked by a fat dot, and the three random planes by small dots. The figure was created in XGobi's tour module with data generated from a sequence of XGobi projection frames.*

automatic movie of projections. Since Asimov's original paper, much of our work on tours has gone into reclaiming the interactive possibilities of 3-D rotations and extending them in new directions. At the root of these efforts is a flexible paradigm for constructing several varieties of tours: The paradigm is based on interpolation of projections. The idea is to provide a computational substrate that allows viewers to visit sequences of projections along smoothly interpolating paths of dynamic projections. The selection of such sequences is deferred to high-level modules built on top of the substrate. See figure 1 for a schematic depiction of the substrate, and figure 2 for a realistic depiction.

We have mentioned some basic ideas of interpolation of projections in several places (Buja and Asimov 1986, Buja et al. 1988, Hurley and Buja 1990), but we have not yet given a comprehensive presentation of the underlying geometry and associated computational techniques. This situation will be remedied in the present work.

In order to fully appreciate the power of interpolation schemes, it is necessary to have an idea of the range of high-level applications they can support. Although this is not the topic of this paper, below is a list of such applications, most of which are available in the XGobi software (Swayne et al. 1998). Sequences of projections to be visited by interpolating paths can be constructed as follows:

- **Random choice:** If the goal is to "look at the data from all sides," that is, to get an overview of the shape of a p-dimensional pointcloud, it may be useful to pick a random sequence of projections. This yields indeed a particular implementation of the **grand tour** (Asimov 1985), defined as an infinite path of projections that becomes dense in the set of all projections.
- **Precomputed choice:** At times one would like to design paths of data projections for specific purposes. An example is the **little tour** (McDonald 1982) which provides a path connecting all 2-D projections onto pairs of variables. Another example is what may be called the **packed tour**: One places a fixed number of projections as far as possible from each other — that is, one finds a packing of fixed size of the Grassmannian manifold — and places a shortest (Hamiltonian) path through these projections. Such Grassmann packings have been computed by Conway, Hardin and Sloane (1995). This is essentially an improvement over the random-choice-based grand tour under the stipulation that one is only willing to watch the tour for a fixed amount of time (Asimov 1985, remark R1, p. 130).
- **Data-driven choice:** This results in methods summarily called **guided tours** because the paths are partly or completely guided by the data. A particular guided tour method uses *projection pursuit*, a technique for finding data projections that are most structured according to a criterion of interest, such as a clustering criterion or a spread criterion. Our implementation of interactive projection pursuit follows hill-climbing paths on the Grassmanian by interpolating gradient steps of various user-selected projection pursuit criteria (Cook et al. 1995). Alternation with the grand tour provides random restarts for the hill-climbing paths. — Another guided tour method uses multivariate analysis rather than projection pursuit: One restricts paths of projections for example to principal component and canonical variate subspaces (Hurley and Buja 1990).
- **Manual choice:** Contrary to common belief, it is possible to manually direct projections from higher than three dimensions. Intuitively, this can be done by "pulling variables in and out of a projection." Generally, any direction in p-space can serve as a "lever" for pulling or pushing a projection. The notion is thus to move a projection plane with regard to immediately accessible levers such as variable directions in data space. Unlike customary 3-D controls, this notion applies in arbitrarily many dimensions, resulting in what we may call a **manual tour**. See Cook and Buja (1996) for more details and for an implementation

in the XGobi software. A related technique has been proposed by Duffin and Barrett (1994).

In the present work we attempt to lay the foundations for thinking about the exploration of spaces of more than three dimensions with moving projections. Because the concepts we introduce are so different from anything taught in conventional 3-D computer graphics, the groundwork to be laid is extensive. We start with some preliminaries (section 2), followed by a discussion of the problem of rendering of projections (section 3). We take a very general view and allow for the possibility that data are rendered by complex methods onto arbitrary dimensions. Thus, a 2-dimensional projection may be rendered by a plain scatterplot as in XGobi, but a 5-dimensional projection may be rendered by a parallel coordinate plot with five axes; in the extreme, the plot may render a fully p-dimensional rotation of all of data space (Wegman 1991). In section 3 we examine an incomplete list of rendering methods in relation to the projection dimension.

For interpretation of a view, rendering the projected data alone is not sufficient — it is equally necessary to convey information about the position of the projection in data space. Graphical methods for displaying projection frames are described in section 4.

From rendering considerations we find that the notion of projection is ambiguous. Is the result of a projection an oriented image, or are all orientations of a projection image equivalent? This question leads to the distinction between projection onto a frame — which imposes a particular orientation — and projection onto a plane — which abstracts from particular orientations. The distinction is important in practice: For some types of renderings, orientation is essential (parallel coordinate plots), while for others it is arguably irrelevant (scatterplots). These issues and some of their implications for interpolation of projections are discussed in sections 5 and 6.

The distinction between projecting onto frames and projecting onto planes becomes even more important when we consider the kinematics of dynamic projections. We will find that frame motion can be decomposed into plane motion — which rotates the plane straight away from itself — and within-plane spin ("whip" spin for short) — which rotates the plane within itself. We analyze the relation between these two types of motion in section 7. In the case of renderings for which orientation is irrelevant, whip spin is generally undesirable, and optimal plane motion must completely avoid whip spin. For full-dimensional data rotations, however, whip spin is essential.

The ideas of plane motion and whip spin help us to analyze the information content in motion. It turns out that motion displays additional data projections, although not as effectively as location. The exact meaning of this statement is explained in section 8.

In the most mathematical part of this work, sections 9 and 10, we examine the notions of distance and speed of frames and planes. Speed is related to distance because it is just distance traveled per time unit. A priori it is not clear how to measure distance of frames and planes, although proposals can be made up easily.

We systematize the question by considering all possible (Riemannian) metrics on projection frames, introducing some intuitive invariance requirements, and showing that essentially only a one-parameter family of distance measures qualifies. In concrete terms, the result says that there is essentially only one way to measure speed of plane motion and only one way to measure speed of whip spin, but one can choose how to combine the two speed components into an overall speed measure of frames. The resulting ambiguity has consequences: Depending on the choice of speed measure, one obtains different optimal (straightest, geodesic) paths connecting two frames. "Optimal" means roughly that at fixed speed the shortest time is taken. Fortunately, the ambiguity does not exist for paths of frames that consist of pure plane motion or of pure whip spin, which, it turns out, are the two most important cases for tour construction.

The remaining sections are dedicated to computational issues of dynamic interpolation of projections. Section 11 gives an algorithmic framework for computer implementations, independent of the particular interpolating algorithms. Section 12 describes optimal interpolation of planes with zero whip spin. Section 13 gives several alternative methods for interpolation of frames, one of which is optimal for pure whip spin. The methods are based on familiar tools from numerical linear algebra: complex eigendecompositions, Givens rotations, and Householder transformations.

In the appendix (section 15) we develop the differential geometry that underlies the theorems used in sections 7 and 9.

A word on the relation of this work to the original grand tour paper by Asimov (1985) is in order: That paper coined the term "grand tour" and devised the most frequently implemented grand tour algorithm, the so-called "torus method." This algorithm parametrizes projection frames, constructs a space-filling path in parameter space, and maps this path to the space of frames. The resulting space-filling path of frames has some desirable properties, such as smoothness, reversibility and ease of speed control. There are two reasons, however, why some of us now prefer interpolation methods:

- The paths generated by the torus method can be highly non-uniformly distributed in the manifold of projection frames. This shows visually quite clearly when the tour lingers among projection planes that are near some of the coordinate axes but far from others. Such non-uniformities are unpredictable and depend on subtle design choices in the parametrization. Although the paths are uniformly distributed in parameter space, their images in the manifold of frames are not, and the nature of the non-uniformities is hard to analyze. By contrast, tours based on interpolation of uniformly sampled projection frames are uniformly distributed by construction. The torus method poses interesting research problems that are not currently solved.
- In interactive visualization systems, for which the grand tour is meant, users have a need to frequently change the set of active variables that are being viewed.

When such a change occurs, a prototype implementation of a tour simply discontinues the current tour, jumps to the new subspace, and starts a new tour. The consequence is discontinuity. In our experience with DataViewer (Buja et al. 1988) and XGobi (Swayne et al. 1998) we found this very undesirable. In order to fully grant the benefits of continuity — such as visual object persistence — a tour should remain continuous at all times, even when the subspace is being changed. Therefore, when a user changes variables in XGobi, the tour interpolates to the new viewing space in a continuous fashion. The advantages of interpolation methods were thus impressed on us by needs at the level of user interaction.

Most numerical methods described in this paper have been implemented and tested in the tour modules of the XGobi software (Swayne et al. 1998), which can be freely obtained from the following web sites for Unix® and Linux operating systems:

http://www.research.att.com/areas/stat/xgobi/

A version that runs under MS Windows$^{\text{TM}}$ using a commercial X$^{\text{TM}}$ emulator has been kindly provided by Brian Ripley. It is freely available from the following web site:

http://www.stats.ox.ac.uk/pub/SWin/

This paper is about mathematical aspects of dynamic projections. As such it will leave those readers unsatisfied who would like to see dynamic projection methods in use for actual data analysis. We can satisfy this desire partly by providing pointers to other literature: A few of our own papers give illustrations (Buja, Cook and Swayne, 1996; Cook, Buja, Cabrera and Hurley, 1995; Furnas and Buja, 1994; Hurley and Buja, 1990). Important authors in this area are Wegman and Carr (Wegman and Carr, 1993; Wegman, 1991). Three-D data rotations are used extensively in the context of regression by Cook and Weisberg (1994). Tierney's (1990, chapter 10) Lisp-Stat system also contains a grand tour implementation. It should be kept in mind, though, that the printed paper has never been a satisfactory medium for conveying intuitions about motion graphics. Nothing replaces hands-on experience and live or taped demonstrations.

2 Preliminaries

Conventions: We use the term "projection plane" or simply "plane" in a very general sense, meaning a subspace of arbitrary dimension d in p-dimensional data space. We consider projection dimensions d anywhere between 1 and the dimension p of data space. In XGobi, $d = 2$ and $d = 1$ are the typical dimensions, but for Wegman (1991) $d = p$ is generic. The "data dimension" p really denotes the number of currently active variables, as opposed to the number of *all* variables. This distinction is necessary in interactive systems such as XGobi, where arbitrary subsets of the variables can be interactively chosen for viewing with dynamic projections.

Remark on metrics in data space: We will always assume that data space is equipped with a canonical inner product that determines a length unit across variables and assumes variable directions to be orthogonal. We are not concerned with the problem of choosing an inner product but simply assume that some choice has been made. It should be up to the data visualization system to provide various choices by allowing viewers to choose different scalings of the data. In XGobi, for example, viewers can choose the half-range, the standard deviation, or the median absolute deviation (MAD) as the unit length.

Notation: A d-dimensional projection plane is generated by an orthonormal basis which we collect in a $p \times d$-matrix $F = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_d)$ called a ***d*-frame**. Note that $F^T F = I_d$ due to orthonormality of the columns. We denote the plane by span(F). Any plane has of course infinitely many frames that span it. The projection of a high-dimensional data vector $\boldsymbol{x}_i$ onto the direction $\boldsymbol{f}_j$ is given by the scalar product $\boldsymbol{f}_j^T \boldsymbol{x}_i$. The projection of $\boldsymbol{x}_i$ onto the orthonormal frame F is the d-dimensional vector

$$View_i = F^T \boldsymbol{x}_i \ .$$

In dynamic graphics, the d-frame is a function of a time parameter t:

$$F(t) = (\boldsymbol{f}_1(t), \ldots, \boldsymbol{f}_d(t)) \ ,$$

and so is the projection

$$View_i(t) = F(t)^T \boldsymbol{x}_i$$

3 Graphical Rendering of a Projection

A rendering method is a way of creating a graphical scene from a d-dimensional projection. In this sense, a scatterplot is a rendering of a 2-D projection, but a parallel coordinate plot with d axes is equally a rendering of a d-dimensional projection.

Rendering methodology is a wide area, covering much of statistical data visualization; see Wegman and Carr (1993) for an excellent introduction. In what follows we describe a few methods that are suitable for rendering real-time dynamic projections of multivariate data given widely available display technology. Most have seen actual use in dynamic graphics systems. We limit ourselves to inexpensive graphical scenes that consist of points and lines, such as pointclouds, polygons, curves and wireframes. This graphical vocabulary is sufficiently rich to permit renderings of data projections as scatterplots, time series plots, simple geometric surfaces and geometric bodies and graphs (see Littman et al. 1992 for visualizing high-dimensional graph layouts).

We are not concerned with the typical rendering questions that are at the heart of most of 3-D computer graphics, such as lighting models and hidden line removal, much of which is dependent on the presence of a single back-dimension. See again Wegman and Carr (1993) for such issues. For some interesting higher-dimensional rendering attempts, see Hanson and Heng (1991) who propose a 4-D shading method, and Young and Rheingans (1991) who experiment with high-dimensional depth-cuing.

One type of complex rendering, however, should be mentioned: Scott (1992, 1995) reminds us that density displays in 2 and 3 dimensions are valuable for large data sets that cause extensive overplotting in conventional 2-D scatterplots and obscuring in 3-D point scatters. Similarly, Carr (Carr et al. 1986, Carr 1991) has argued in favor of binned displays for large data samples. In a different context, Miller and Wegman (1991) explore density enhancements for parallel coordinate plots. Some of this is now in ExploreN with hardware support based on alpha-blending techniques (Carr et al. 1996).

We discuss rendering methods for projected multivariate data in the order of the projection dimension.

3.1 Rendering of 1-dimensional projections

The obvious method of displaying a 1-D projection as points along a line is not useful due to the problem of overstrike. One way out of this problem is to make meaningful use of the second screen dimension. Below we describe several options, all of them illustrated in figure 3. Other possibilities, not discussed here, are aggregating displays such as histograms and boxplots.

- **Jitter plots:** A simple method to alleviate the overstrike problem is to spread out the points in the second screen dimension, for example with random "jitters", by plotting the 1-D projection against random numbers:

  $$Hor_i = S_H \cdot Random_i + C_H \ , \qquad Vert_i = S_V \cdot \boldsymbol{f}_1^T \boldsymbol{x}_i + C_V \ ,$$

  where S_H, S_V, C_H, C_V are horizontal and vertical scale and centering constants, respectively. Perceptionally superior to simple random jitters is a sophisticated method by Tukey and Tukey (1990) called **textured dot plots**. They create jitters that are slightly more regular so as to avoid the random clumps that come with the use of simple random numbers. Textured dot plots are computationally more expensive than random number jitters because the jitters are computed based on the distribution of the 1-D projection. For dynamic projections this implies continuous updating as the projection moves in time.

- **Density plots:** Another use of the second screen dimension is by plotting an estimated density function $\hat{p}$ against the projection:

  $$Hor_i = S_H \cdot \boldsymbol{f}_1^T \boldsymbol{x}_i + C_H \ , \qquad Vert_i = S_V \cdot \hat{p}(\boldsymbol{f}_1^T \boldsymbol{x}_i) + C_V \ .$$

  An early use of dynamic density plots appeared in the PRIM-H system (Donoho, Huber, Ramos and Thoma 1982). Tierney (1990) demonstrated a histogram

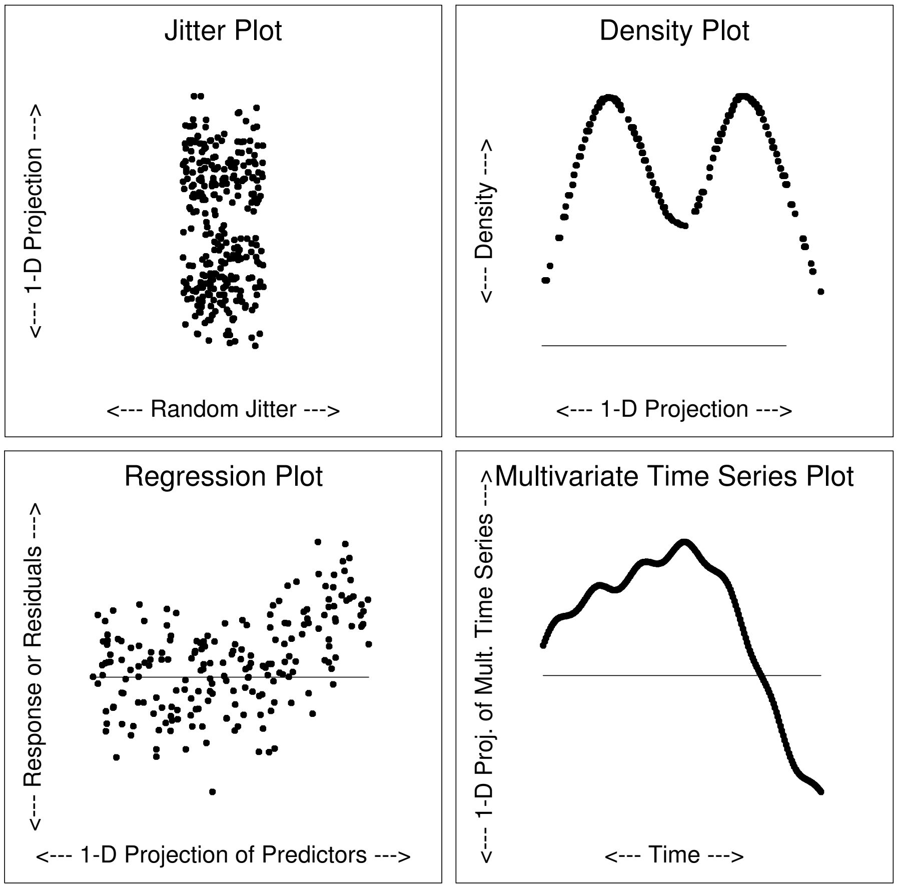

Figure 3: *Examples of 1-D rendering methods: jitter plots, density plots, regression plots, and multivariate time series plots.*

tour in his Lisp-Stat system. In Hurley and Buja (1990) we used average shifted histograms for computationally fast density estimation (Scott 1985). Both for speed and for the ability to point at individual cases in the data, we only plotted the density values at the projected data points. Even if overstrike occurs, the presence of high density areas in the projection is conveyed graphically.

- **Regression plots:** A useful application of dynamic 1-D projections is to predictor variables in regression problems. We plot a fixed variable y_i, such as a response or (partial) residual, against the dynamic 1-dimensional projection of some of the predictors $\boldsymbol{x}_i$:

$$Hor_i = S_H \cdot \boldsymbol{f}_1^T \boldsymbol{x}_i + C_H \ , \qquad Vert_i = S_V \cdot y_i + C_V \ .$$

A thorough search over $\boldsymbol{f}_1 = \boldsymbol{f}_1(t)$ may lead to the discovery of nonlinearities, heterogeneous variance, or extreme response values (Cook and Weisberg 1994).

- **Time series plots:** We have also found it useful to plot the dynamic projection $\boldsymbol{f}_1^T \boldsymbol{x}_i$ vertically and use an auxiliary fixed variable x_i as the horizontal axis:

$$Hor_i = S_H \cdot x_i + C_H \ , \qquad Vert_i = S_V \cdot \boldsymbol{f}_1^T \boldsymbol{x}_i + C_V \ .$$

  We have applied this technique to multivariate time series data $\boldsymbol{x}_i$, in which case it is sensible to use time as the horizontal auxiliary variable x_i. (Time refers here to observed time in the data, not the time parameter t of the dynamic projection vector $\boldsymbol{f}_1(t)$.)

The last two rendering techniques can also be interpreted as particular 2-D projections of the data, but the dynamic part is only a 1-D projection. Both techniques were tried out in the DataViewer system (Buja et al. 1988) as special cases of a broader class of methods called "correlation tours." These are now available in XGobi.

An application of 1-D tours to multi-spectral images is described in Wegman, Solka and Poston (1996): They use the linear combination produced by a restricted 1-D tour for dynamic grey scale coding of 6-dimensional images.

3.2 Rendering of 2-dimensional projections

The usual way of mapping a projection 2-plane to a scene is by assigning the vectors of the 2-frame $F = (\boldsymbol{f}_1, \boldsymbol{f}_2)$ to the horizontal and vertical screen dimensions, respectively:

$$Hor_i = S_H \cdot \boldsymbol{f}_1^T \boldsymbol{x}_i + C_H \ , \qquad Vert_i = S_V \cdot \boldsymbol{f}_2^T \boldsymbol{x}_i + C_V \ ,$$

which results in the usual Cartesian scatterplot, possibly enhanced with lines for interpretability. It may be sensible to choose S_H and S_V such that one length unit in the projection plane translates to the same physical dimensions horizontally and vertically on the computer screen. This amounts to $S_V = -S_H$ if the physical dimensions of the pixels are identical horizontally and vertically, and if the hardware uses the conventional axis orientation (left to right and top to bottom).

3.3 Rendering of 3-dimensional projections

For a 3-dimensional projection with corresponding 3-frame $(\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3)$, three common viewing methods are: Stereo views, depth cues, and 3-D rotations. The most powerful 3-D views take advantage of all three methods simultaneously. In the following we assign the frame vectors to horizontal, vertical, and depth directions, respectively.

- **Stereo views**, in the simplest case, can be obtained by generating two sets of screen coordinates that differ slightly in their 2-dimensional views, one view per eye:

$$Hor_i = S_H \cdot (\cos\eta \cdot \boldsymbol{f}_1 + \sin\eta \cdot \boldsymbol{f}_3)^T \boldsymbol{x}_i + C_H \ , \qquad Vert_i = S_V \cdot \boldsymbol{f}_2^T \boldsymbol{x}_i + C_V \ ,$$

where the exposure angle η may be $-1°$ for the left eye and $+1°$ for the right eye, if we interpret the $\boldsymbol{f}_3$-direction as pointing away from the viewer (see Wegman and Carr (1993), section 6, for a detailed discussion). Separate exposure for each eye can be achieved by several techniques: 1) special purpose screens that generate two overlaid images in different polarizations, combined with suitable glasses for the viewer to achieve separate exposure of the eyes to the two images; 2) special purpose mirrors that match two distinct screen windows with separate projections to the left and the right eye; 3) alternating exposure of the two eyes to separate images at rapid speed; 4) general purpose color screens on which two overlaid images are generated in, say, blue and red colors, combined with red-blue glasses to achieve separate eye exposure. Among these possibilities, the last is low-tech and does not require extra hardware besides cheap red-blue glasses. It is limiting, however, in that it confines the implementor to one specific use of color. The above simple formulae are sufficient for creating the stereo effect but if desired one could follow standard computer graphics techniques and implement sophisticated 3-dimensional perspective viewing with finite eye position.

- **Depth cues** or, strictly speaking, monocular depth cues (in contrast to binocular depth cues provided by stereo views): These are created by forcing certain viewing parameters of the display objects to be monotone functions of the back projection

  $$Depth_i = S_D \cdot \boldsymbol{f}_3^T \boldsymbol{x}_i + C_D \ .$$

  Examples of such viewing parameters are:

  - Brightness: Show far objects dimmer than near objects,
  - Blue-tint or fog: Show far objects as seen through the atmosphere over a distance,
  - Size: Draw far objects smaller.

  See figure 4 for examples of brightness and size cues.

- **3-D rotations** are generated as dynamic images, in the simplest case of a rotation around the vertical axis, by

  $$Hor_i = S_H \cdot (\cos t \cdot \boldsymbol{f}_1 + \sin t \cdot \boldsymbol{f}_3)^T \boldsymbol{x}_i + C_H \ , \qquad Vert_i = S_V \cdot \boldsymbol{f}_2^T \boldsymbol{x}_i + C_V \ ,$$

  where t is a time parameter.

  It might appear that 3-D rotations are not suitable for rendering dynamic 3-D projections because such rotations are dynamic themselves. However, an early proposal by Tukey and Tukey (1981, p. 272-3) suggests just this: In essence, they would run a grand tour with $d = 3$ at slow speed and spin the 3-D projection space within itself at high speed, a process they call **wabing**. They derive this term from the first stanza in Lewis Carroll's "Jabberwocky" (from "Through the Looking Glass"):

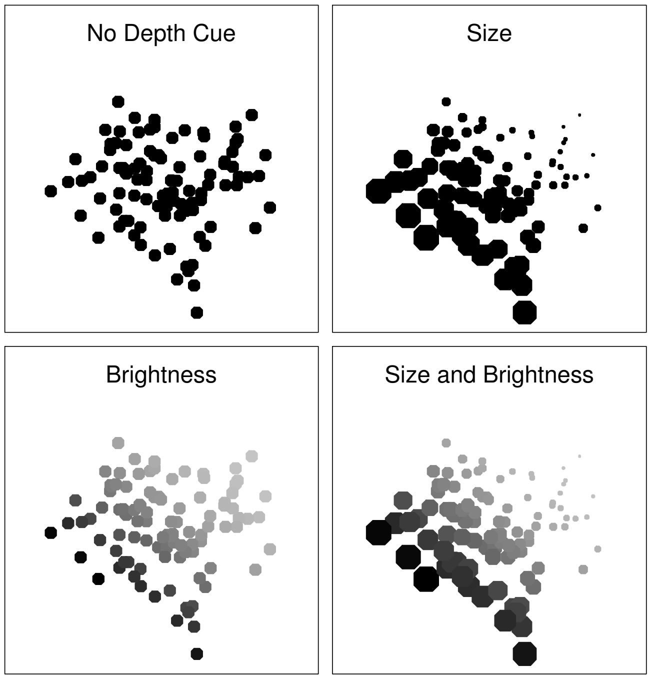

Figure 4: *Examples of depth cues for 3-D rendering methods: size and brightness.*

'Twas brillig, and the slithy toves
Did gyre and gimble in the wabe;
All mimsy were the borogoves,
And the mome raths outgrabe.

Borrowing further terms from "Jabberwocky", they call the fast 3-D rotation **gyring** and the slow tour motion **gimbling**. This would allow one to take advantage of 3-D perception without stereo technology and yet move through higher-dimensional data space at the same time. The drawback of wabing is that the speed of the tour motion has to be relatively slow, but unlike some of the depth cue techniques, there is no interference with the use of color.

An implementation reminiscent of wabing is available in Tierney's (1990) Lisp-Stat system: His 2-D tour carries out a full-dimensional tour but shows only two dimensions while in progress. When the tour is stopped, a third dimension is accessible for interactive 3-D rotation in the familiar manner.

3.4 Rendering of projections onto dimensions greater than 3

We consider projection onto dimensions greater than 3. In the extreme, the projection dimension d can equal the dimension p of data space, in which case the dynamic projection amounts to a full display of all variables in a dynamically rotating coordinate system. Some examples of d-D rendering methods follow and these are illustrated in figure 5:

- **Scatterplot matrices** (called "generalized draughtman's views" by Tukey and Tukey (1981, p. 206f); see also Chambers et al. (1983)): One arranges the scatterplots of all variable pairs in a matrix arrangement. If used as a rendering method of a d-dimensional projection, the (j,k)'th scatterplot of the $d \times d$ matrix arises from the projections onto the frame vectors $\boldsymbol{f}_j$ and $\boldsymbol{f}_k$. This method is used in the ExploreN system (Carr et al. 1996) to render a full-dimensional grand tour with $d = p$.
- **Parallel coordinate plots** (Inselberg 1985, Wegman 1991, Wegman and Luo 1996): One uses parallel lines to represent the d coordinate axes; one then represents a case by a polyline that connects the coordinate values on the axes with straight line segments. This method depends on an ordering of the coordinates, and it generally works best for small numbers of cases. In order to render a projection, the frame vector $\boldsymbol{f}_j$ yields the j'th axis of the display. Wegman (1991) and Wegman and Luo (1996) illustrate the use of parallel coordinates for a full-dimensional tour with $d = p$. It is also implemented in the ExploreN system (Carr et al. 1996).
- **Andrews curves** (Andrews 1972): The description of this technique is often buried under arcane details involving trigonometric functions. The essential idea is this: Each case is represented by a curve that is obtained as a trace of smoothly varying 1-D projections: Case $\boldsymbol{x}_i$ is represented by the curve $s \rightarrow \boldsymbol{f}(s)^T \boldsymbol{x}_i$, where $\boldsymbol{f}(s)$ is a 1-parameter family of projection vectors. Effectively, Andrews curves display 1-D tours as plots of functions of the time parameter, as opposed to the conventional 1-D tour which makes use of the time parameter as physical time. If used as a rendering method of dynamic projections, Andrews curves will be dynamically computed from $View_i(t) = F(t)^T \boldsymbol{x}_i$ rather than $\boldsymbol{x}_i$. As of this writing we do not know of a tour implementation that makes use of Andrews curves for rendering dynamic projections.

Other rendering methods for projections onto d dimensions exist (generally of the glyph type: stars, castles, ...) and could be used for tour rendering as well.

4 Knowing Where We're Looking in High-Dimensional Space: Rendering Frames

A dizzy feeling besets many first-time viewers of high-dimensional data projections, and they rightfully ask "How do I know what I'm looking at?" Crucial for the successful use of data projections is the availability of tools for interpreting views. In geometric terms, the task supported by such tools is that of locating the position of the projection frame in p-space. In numeric terms, the task really boils down to interpreting the numbers contained in the $p \times d$ frame F. In principle, one could simply

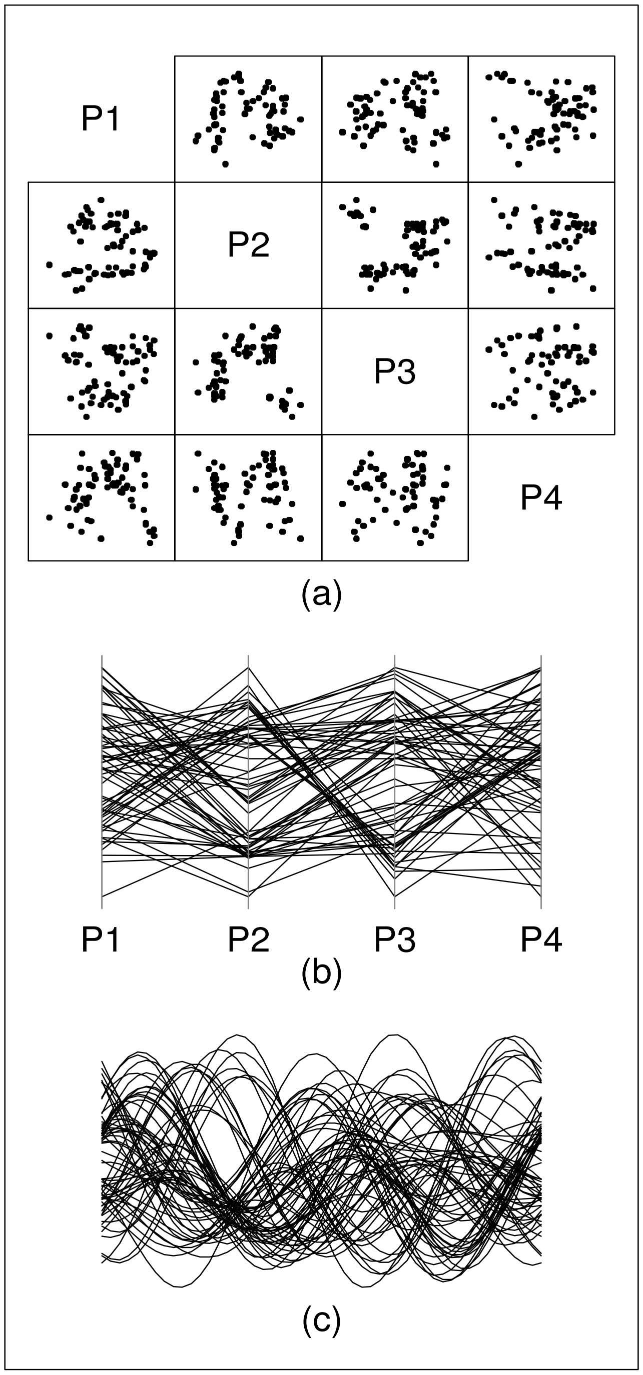

Figure 5: *Examples of d-D rendering methods for $d = 4$: (a) Scatterplot matrix,(b) parallel coordinates, (c) Andrews curves. The data are the first four principal components extracted from some speech data representing 11 vowels, each vowel replicated 6 times ($N = 66$). Thus, 11 groups of size 6 are known to exist, which may or may not show up as clusters in the rendered projections.*

print these numbers as a matrix, and this may be useful at times: The values show how much individual variables "load" on the 1-D projections given by the columns $\boldsymbol{f}_j$ of F. There exist, however, better and more visual ways of conveying this information.

We take a general view and ask how arbitrary rendering methods can be supplemented with tools for visually rendering F. In particular, the principles should apply to all methods discussed in the previous section: 1-D renderings with jitter plots, density plots, regression plots, multivariate time series plots; 2-D renderings with scatterplots; 3-D renderings with stereo views, depth cues, 3-D rotations; p-D renderings with parallel coordinate plots, scatterplot matrices, Andrews curve plots.

Two rendering principles that often lead to reasonable solutions are the following:

1. Render the d column vectors of F individually with graphical renditions of the p numbers in each column. That is, make d times use of something that works for rendering a 1-D projection vector.

2. Interpret the variable unit vectors in p-space as regular data, project them with F like the real data, and render the result also like the real data. That is, render the rows of F as if they were projected data.

An example of rendering a 1-D projection frame is shown in the lower part of figure 6: The p numbers in the 1-frame are plotted below the data area and coded as values on a common scale on vertically shifted axes in order to avoid overplotting. One easily reads off how much and with which sign each variable contributes to the projection. — This method can be easily generalized according to the first principle: In parallel coordinate plots and scatterplot matrices, for example, one could attach a suitably miniaturized version of these 1-D renditions to each of the d projection dimensions, sideways or below the horizontal or vertical axes.

An example of the application of the second principle is shown in figure 7: The traditional tripod of 3-D computer graphics is just an enhanced rendition of the projection of the three variable unit vectors. Similarly, the generalized tripod called "p-pod" shown when touring in XGobi is an enhanced rendition of the p variable unit vectors in p-space. In XGobi, the p-pod is also rendered by a second method in order to avoid overplotting and to provide interactive manipulation of the variables, see the right of the same figure. — The p-pod method works also for 3-D projections rendered with 3-D stereo views and 3-D rotations: The pod will look like a star with p unequal rays in 3-space.

The second principle can be applied to parallel coordinate plots and Andrews curves: Treat the variable unit vectors as if they were data, render them with polylines or curves, and place, mark and label them visually so they are recognized as guide posts rather than data. For an example in terms of Andrews curves, see figure 8.

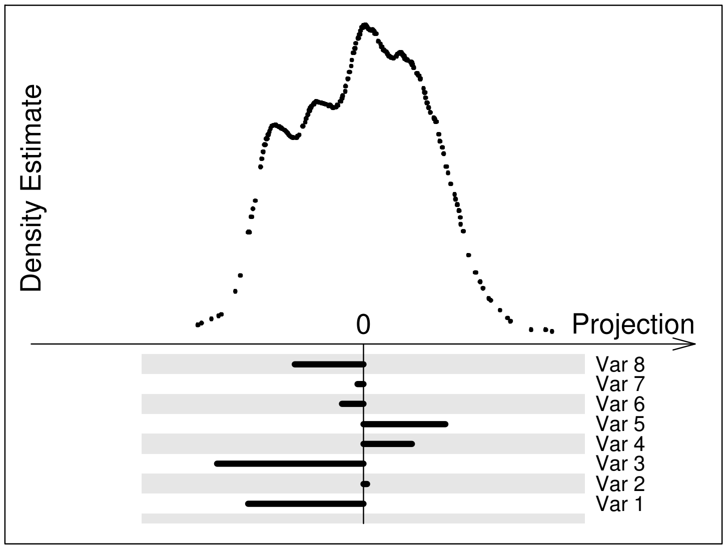

Figure 6: *Rendering of the current frame position for 1-D projections: The example shows $p = 8$. The data are rendered with a density plot, while the position of the projection direction is rendered with horizontal bars below the plotting area.*

5 When do we Project on a Frame, When on a Plane?

5.1 Orientation: Why it matters

Section 3 makes it clear that the notion of projection is quite ambiguous: For $d = 2$ one might assume that the essential substrate of the projection is the plane span(F), but for $d = p$ the projection subspace is trivial, namely all of data space, and the essential substrate of the projection is not the space but its orientation as given by the full $p \times p$ frame F.

The questions then are: When does orientation matter, and when are all orientations of a plane equivalent for visualization purposes? Does it matter whether an axis is used left-to-right or right-to-left, whether a scatterplot is shown 45 or 90 degrees rotated, which among three variables is coded as depth cue in a 3-D scatterplot, whether the axes in a parallel coordinate plot are permuted or replaced by linear combinations of the original variables? Generally, when are a frame F and a rotated version FV thereof equivalent? (V is $d \times d$ and $V^T V = I_d$.)

Some of these questions concern visual perception and should be answered by empirical research. In the absence of such research, we are left with a priori reasonings that hopefully will not be too far off. The role of orientation generally depends on

- the rendering method and
- the type of scenes we expect to see.

In the next section we give our best guesses for the rendering methods described in the previous section for dimensions $d = 1$, 2, 3, and $d > 3$.

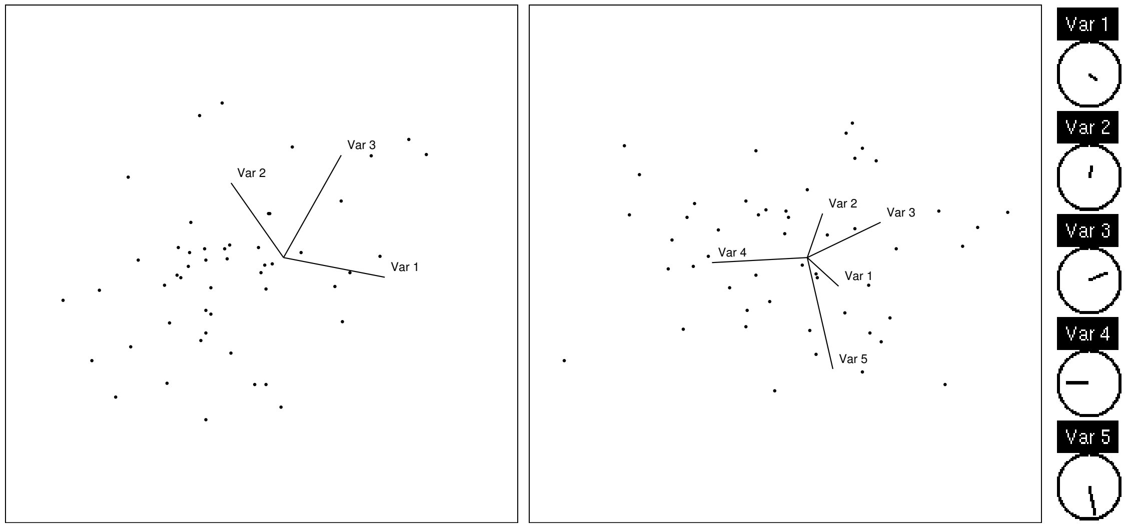

Figure 7: *Rendering of the current frame position for scatterplots of 2-D projections: Left: A conventional tripod representing the projections of the unit vectors of the three variables in 3-D data space.*
Right: A generalization of the tripod to a so-called "p-pod" is shown in the plotting area. In this example $p = 5$. On the right is the XGobi-style rendition of the p-pod: The rays of the p-pod are disentangled and placed in individual variable circles to avoid overplotting and allow manipulation of variables with mouse clicks. The rays again represent the projections of the variable unit vectors onto the current projection plane. Equivalently, the j'th ray represents the j'th row vector of the 2-frame F.

Our interest in the question of orientation stems from the fact that it has fundamental implications for tour interpolations:

- If projections are rendered such that orientation is irrelevant, we need to interpolate planes span(F). It is then possible to establish precise optimality of certain interpolating paths of planes.
- If projections are rendered such that orientation matters, we need to interpolate frames F rather than planes. Curiously, this case is theoretically less tractable, with one exception: In the extreme, when the plane span(F) is at rest and the dynamic frame F is essentially a dynamic basis of span(F), optimal interpolations do exist. This includes the case of full-dimensional tours $p = d$.
- For general dynamic frame motion we will see that even the criteria for optimality are not unique. The reason is that no unique method for measuring speed of frame motion exists (section 9).

5.2 Orientation: When it matters

In what follows, it should be kept in mind that every rendering method depends on roles assigned to the vectors of a basis frame $F = (\boldsymbol{f}_1, \boldsymbol{f}_2, ..., \boldsymbol{f}_d)$. In a 3-D scatterplot,

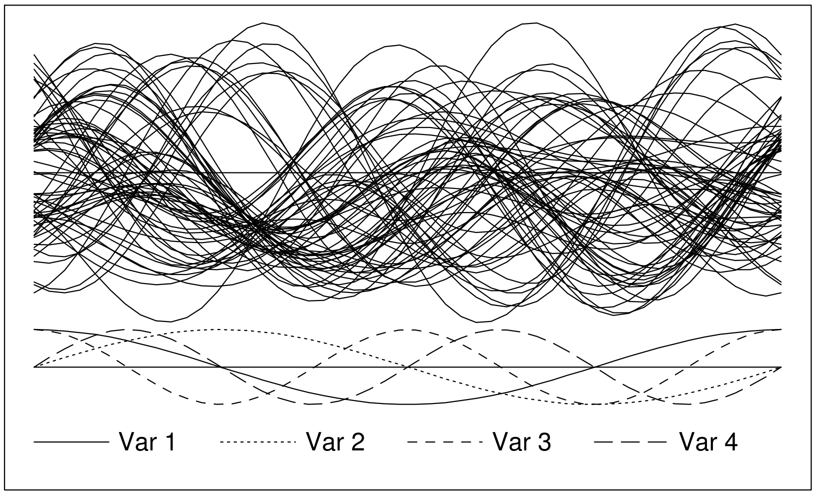

Figure 8: *Rendering of the current frame position for Andrews curves: The upper curves represent the data, the lower and smaller curves represent the variable unit vectors. They express how much each variable contributes at each location of the horizontal axis.*

for example, $\boldsymbol{f}_1$ may be used for the horizontal screen direction, $\boldsymbol{f}_2$ for the vertical direction, and $\boldsymbol{f}_3$ for the depth cues. Here is a discussion of the role of orientation for each of the rendering methods of section 3:

$\boldsymbol{d} = 1$: In all applicable rendering methods, the single projection dimension is used as a horizontal or vertical screen axis. Irrelevance of orientation means that a projection can be equally displayed left-to-right and right-to-left when used horizontally, or bottom-up and top-down when used vertically. When the projection is fixed and has a specific meaning, such as time, orientation may matter. This can be due to the nature of the visual scene: Text, for example, is hard to read when mirror-imaged. In the kinds of scenes we expect from dynamic 1-D projections of multivariate data, however, no specific interpretation of the projection exists other than being a linear combination of variables, and the scenes are not structured in a way that makes a distinction between left-to-right and right-to-left essential. Therefore, orientation seems irrelevant for dynamic 1-D projections.

$\boldsymbol{d} = 2$: The only case we consider is essentially a scatterplot of the two 1-D projections. Irrelevance of orientation means that rotated and axially reflected scatterplots are visually equivalent. As for $d = 1$, there do exist scatterplots where orientation matters because of the nature of the scene and the specific interpretations of the axes. An example are again time series, but so are all plots whose main feature of interest is slope. The kinds of plots we expect from dynamic 2-D projections of multivariate pointclouds, however, would rarely have features whose perception depends on a specific orientation on the screen. Typical structure such as clusters, lines, curves, outliers and combinations thereof can be recognized quite easily without turning a scene around. Therefore, orientation seems irrelevant for dynamic 2-D projections.

$\boldsymbol{d = 3}$**:** In this situation, the rendering methods discussed above show somewhat different behaviors under changes of orientation. Depth cues and stereo views do not provide as much resolution in the back dimension as does location in the two front dimensions. Therefore, orientation matters somewhat. It matters less for 3-D rotations: The depth dimension changes continuously, and what was depth is screen location a moment later after a 90 degree rotation. This is true for Tukey and Tukey's (1981) wabing proposal for rendering a slowly moving p-to-3-D projection with fast moving 3-D rotations. This rotation amounts to a purposeful use of within-plane spin for rendering. It is therefore desirable that this 3-D rotation be unaffected by the motion of the 3-frame, which just means that the moving 3-frame be free of within-plane spin.

$\boldsymbol{d > 3}$**:** In more than 3 dimensions, the three rendering methods discussed above — scatter plot matrices, parallel coordinate displays and Andrews curves — all depend on orientation through their particular use of the projection directions $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_d$. We do not know of any methods for orientation-independent rendering in this general case. For a full-dimensional tour $d = p$, the point of a tour is to expose the viewer to different orientations of full-dimensional data space, as opposed to different low-dimensional projections. Therefore, underlying a full-dimensional tour is the assumption that data space is rendered with essential use of orientation.

The answer to the question of whether to project onto a plane or a frame is primarily a function of the rendering method. The choice of rendering method, however, is largely a function of the projection dimension, hence the question of planes versus frames ends up being a function of the projection dimension d by proxy. In summary:

- When rendering with scatterplot methods onto dimensions 1, 2 and 3, we project onto a *plane*;
- when rendering with scatterplot matrices, parallel coordinate plots or Andrews curves onto dimensions $d > 3$, we project onto a *frame*.

6 Interpolating Frames and Planes

For visualization of data projections it is always necessary to refer to a frame

$$F = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_d)$$

because some orientation of the projection is needed for rendering. This holds even when orientation is irrelevant: Irrelevance only means that renderings that differ in orientation are perceptually equivalent, not identical.

When orientation matters for a rendering method, a frame is the essential substrate: A frame F and a rotated version FV of the same frame do *not* produce equivalent visual scenes, although the planes are the same: $\mathrm{span}(F) = \mathrm{span}(FV)$. [$V$ is $d \times d$ and orthogonal $V^T V = I_d$.]

When orientation does not matter, rendering is still based on a frame F, but rotated versions FV of the frame yield equivalent visual scenes. We therefore need a mathematical substrate that encodes this equivalence. The most plausible substrate of a plane, $\mathrm{span}(F)$, is mathematically inconvenient. More useful is the orthogonal projection matrix P generated by F:

$$P \;=\; FF^T \;.$$

The planes of dimension d are in a 1-1 correspondence with the matrices P characterized by idempotence ($PP = P$), symmetry ($P^T = P$) and rank d. Obviously, F and FV generate the same P.

Convention: *Henceforth the symbols F and P will always represent a frame and its associated orthogonal projection matrix.*

We turn to the problem of interpolating frames and planes. Let

$$F_a = (\boldsymbol{f}_{a,1}, \ldots, \boldsymbol{f}_{a,d}) \quad \text{and} \quad F_z = (\boldsymbol{f}_{z,1}, \ldots, \boldsymbol{f}_{z,d})$$

be a starting frame and a target frame, respectively. Similarly let

$$P_a = F_a {F_a}^T \quad \text{and} \quad P_z = F_z {F_z}^T$$

be the projection matrices that encode the corresponding starting plane and target plane, respectively.

Interpolation of the frames F_a and F_z is done with a path of frames $F(t)$ ($t_a \le t \le t_z$) that satisfies

$$F(t_a) = F_a \quad \text{and} \quad F(t_z) = F_z \;.$$

Interpolation of planes in the abstract means that a path of frames generates a path of projection matrices $P(t) = F(t)F(t)^T$ that runs from P_a to P_z. This definition is not useful in dynamic graphics because the path $F(t)$ needs to continue from where the previous path left off in order to ensure continuity of motion. Thus, the path needs to satisfy

$$F(t_a) = F_a \quad \text{and} \quad P(t_z) = P_z \;.$$

The difference between frame and plane interpolation is that in plane interpolation *any frame in the target plane can serve as a target frame.* This confers a certain measure of freedom to plane interpolations. The question is how to take advantage of this freedom. This is one of the topics of the next section: Through proper choice of the target frame, one can avoid undesirable spin within the projection plane.

7 Kinematics I: Within-Plane Spin ("Whip Spin")

7.1 Motivation

Consider, for illustration purposes, the canonical (1,2)- and (3-4)-frame in 4-space, and the simplest interpolating path $F(t)$:

$$F_a = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} , \quad F_z = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} , \quad F(t) = \begin{pmatrix} c_t & 0 \\ 0 & c_t \\ s_t & 0 \\ 0 & s_t \end{pmatrix} ,$$

where we abbreviated $c_t = \cos(t)$ and $s_t = \sin(t)$. The path of 2-frames $F(t)$ interpolates between F_a and F_z when t runs from 0 to $\pi/2$. Now consider an alternative path:

$$\tilde{F}(t) \ = \ F(t) \begin{pmatrix} c_{vt} & -s_{vt} \\ s_{vt} & c_{vt} \end{pmatrix}$$

For any value of v, the paths $F(t)$ and $\tilde{F}(t)$ generate the same path of planes, interpolating the (1,2)-plane and the (3,4)-plane. The path $\tilde{F}(t)$, however, interpolates the (1,2)-frame and the (3,4)-frame only when v is an integer multiple of 4: $v = 4n$. In an intuitive sense, the most economical interpolation is for $v = 0$. If v is far from zero, the path of frames $\tilde{F}(t)$ rotates within the plane it spans at great speed, similar to propeller blades of an airplane. In data visualization, where the frames are used to propel projections of high-dimensional objects, the viewer of such projections will perceive an image that spins within the computer screen at high speed, while the change in the image due to the motion of the projection plane itself is obscured. This type of spin is clearly undesirable when plane interpolation is wanted. Because of the importance of this notion we introduce the term **within-plane spin** or contracted **whip spin** to refer to this component of moving frames.

7.2 Mathematical definition of whip spin and plane motion

Given a general path of d-frames

$$F = F(t) \ = \ (\boldsymbol{f}_1(t), \ldots, \boldsymbol{f}_d(t)) \ ,$$

denote its time derivative by

$$F' \ = \ F'(t) \ = \ \mathrm{d}F(t)/\mathrm{d}t$$

and the time derivative of its columns by

$$\boldsymbol{f}_j' \ = \ \boldsymbol{f}_j'(t) \ = \ \mathrm{d}\boldsymbol{f}_j(t)/\mathrm{d}t \ .$$

[We usually suppress the time parameter t.]

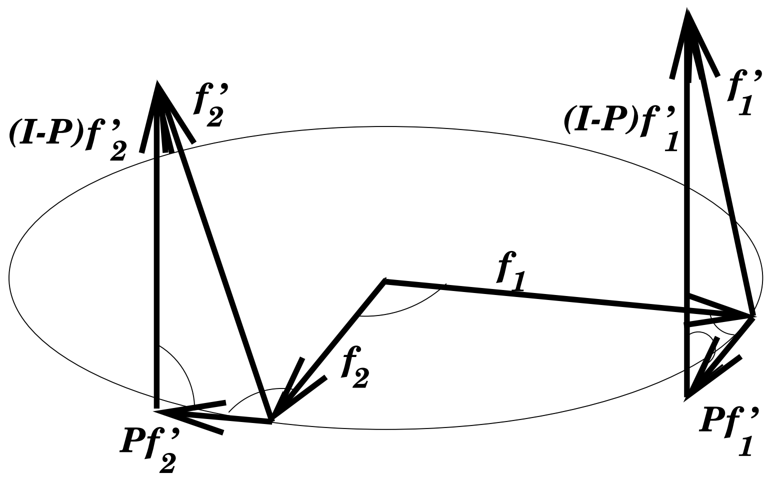

Figure 9: *Decomposition of the derivative F' of an orthonormal 2-frame $F = (\boldsymbol{f}_1, \boldsymbol{f}_2)$ into whip spin PF' and plane motion $(I-P)F'$. The derivatives $\boldsymbol{f}_1'$ and $\boldsymbol{f}_2'$ are the tangent vectors of $\boldsymbol{f}_1 = \boldsymbol{f}_1(t)$ and $\boldsymbol{f}_2 = \boldsymbol{f}_2(t)$, respectively. Note the patterns in the projections: $P\boldsymbol{f}_1' = \alpha \cdot \boldsymbol{f}_2$ and $P\boldsymbol{f}_2' = -\alpha \cdot \boldsymbol{f}_1$, which expresses the geometric meaning of the skew whip spin matrix F^TF' as an infinitesimal rotation in the plane spanned by $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$. (All shown angles are right angles.)*

Whip spin can be quantified by projecting the time derivative of the frame onto the plane spanned by the frame. Plane motion is what's left after removing whip spin. Hence the following terminology:

Convention:
- *The* **whip spin** *or* **within-plane spin** *of F is* PF'.
- *The* **plane motion** *of F is* $(I-P)F'$.

Both are $p \times d$-matrices. Whip spin is the component of F' that takes place within the plane span(F), and plane motion is the component of F' that shoots vertically out of the projection plane.

Because $PF' = F(F^TF')$, all the relevant information about whip spin of F is contained in the $d \times d$-matrix F^TF': It describes whip spin with regard to the basis F of the projection plane. Due to its pervasiveness throughout the rest of the paper we introduce the following:

Convention: *We call F^TF' the* **whip spin matrix** *associated with $F = F(t)$.*

The (j,k)-entry of F^TF' is $\boldsymbol{f}_j^T\boldsymbol{f}_k'$ which represents the projection of $\boldsymbol{f}_k'$ onto the direction $\boldsymbol{f}_j$. The following fact is simple but crucial and used everywhere in what follows:

Fact: *The whip spin matrix is skew-symmetric for orthonormal paths of frames:*

$$(F^T F')^T \;=\; -\,F^T F' \;.$$

This is illustrated in figure 9 for 2-frames in 3-space. The proof is by taking derivatives in the orthonormality condition $F^T F = I_d$.

It is easily checked that the path $F(t)$ in the example of section 7.1 has zero whip spin: $F^T F' = 0$. The path $\tilde{F}(t)$ has zero whip spin iff $v = 0$. It is of interest to generalize this example to arbitrary d-frames as follows: Let $F = F(t)$ be a path of d-frames with zero whip spin; also, let $V = V(t)$ be a path of $d \times d$-rotations. Then the path of frames $\tilde{F} = FV$ generates the same path of planes as F, but its whip spin matrix is:

$$\tilde{F}^T \tilde{F}' = V^T V' \;.$$

That is, the rotations V are the purveyors of whip spin. [To prove the identity use $F^T F = I_d$ and $F^T F' = 0$.] This example will be used repeatedly in the following sections where we consider the extremes of 0% whip spin and 100% whip spin, respectively.

7.3 First extreme: Pure plane motion

Consider the following situation: All the apparent motion is due to motion of the plane. The frame vectors move orthogonally out of the plane. This is characterized by the equivalent conditions

$$F^T F' = 0 \;, \quad \text{and} \quad P F' = 0 \;.$$

For a path $\tilde{F} = FV$ where F has zero whip spin, the condition is equivalent to $V^T V' = 0$. It turns out that any path $\tilde{F}$ can be represented in this way:

Theorem 1. *For every path of frames $\tilde{F}(t)$, there exists another path $F(t)$ that generates the same planes but has zero whip spin. More specifically, there exists a dynamic $d \times d$-rotation $V(t)$ such that*

$$\tilde{F} \;=\; FV \;, \qquad F^T F' = 0 \;, \qquad F(0) = \tilde{F}(0).$$

The proof is given in the Appendix (section 15.8).

The theorem guarantees that if there exists a path of frames that interpolates two planes (as opposed to frames), the path can be chosen with zero whip spin.

In section 12 we construct plane interpolations that have not only zero whip spin but are optimal in the sense that they are geodesic with regard to most reasonable metrics on frames. Metrics will be the topic of section 9.

7.4 Second extreme: Pure whip spin

Now consider the opposite extreme: Motion takes place entirely within the plane spanned by F. The viewer is getting to see the same plane rotated within itself. That is, the plane is at rest although the frame may move. This is characterized by the equivalent conditions

$$P' = 0 \ , \quad \text{and} \quad PF' = F' \ .$$

Equivalence follows from skew-symmetry of the whip spin matrix $F^T F'$.

Pure whip spin is easily generated by superposing a fixed frame F_a with a dynamic $d \times d$-rotation $V(t)$:

$$F(t) \ = \ F_a V(t) \ .$$

Obviously, the projection $P = FF^T = F_a F_a^T$ is fixed, and the whip spin matrix is $F^T F' = V^T V'$.

If a tour consists of paths of pure whip spin, the point is to get to see different orientations of span(F_a). This is the case in particular for full-dimensional tours $d = p$, see Wegman (1991), Wegman and Luo (1996), and Carr, Wegman and Luo (1996). The full-dimensional case is really the generic case: If $d < p$ but the plane is at rest, consider this fixed plane as data space, hence $p = d$ by restriction.

8 What We See When We Watch Moving Projections

If a data vector $\boldsymbol{x}_i$ is projected onto a moving frame $F(t)$, we see some rendering of

$$View_i(t) \ = \ F(t)^T \boldsymbol{x}_i \ .$$

In addition, we see the projection of $\boldsymbol{x}_i$ in motion, that is, we see where the projected point is going, at least infinitesimally. For a feeble attempt at depicting this fact in a static image, see the right hand plots in figure 10. In mathematical terms we can say we "see" the derivative

$$View_i'(t) \ = \ F'(t)^T \boldsymbol{x}_i \ .$$

These two items, $View_i(t)$ and $View_i'(t)$, represent location and velocity of a projected data vector. We can now compare the information they contain:

> *Both location and velocity are projections of the data vector $\boldsymbol{x}_i$, the first onto the frame $F(t)$, the second onto the frame $F'(t)$.*

In contrast to $F(t)$, the derivative frame $F'(t)$ is not usually an orthonormal frame. Nevertheless, $F'(t)\boldsymbol{x}_i$ can be interpreted as an oblique projection.

For example, in the common situation of 2-dimensional data projections, velocity adds in principle another 2 dimensions of information to the projection. Thus, when watching a grand tour of 2-dimensional projections in $p \geq 4$-dimensional space, we "see" the 4-dimensional subspace span$(F(t), F'(t))$! If data space is p=3-dimensional,

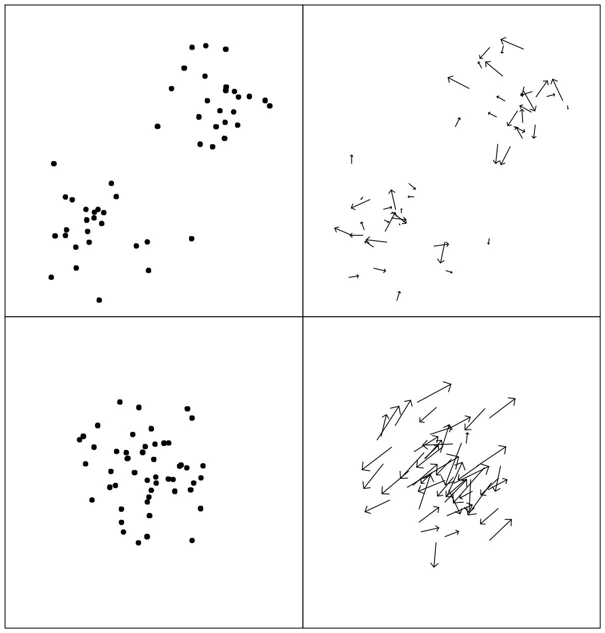

Figure 10: *What we see when we watch moving 2-D projections: A static rendition in terms of scatterplots. The two rows show two projections of a 4-D data set consisting of two well-separated normal clusters. The left column shows location, the right column shows in addition speed represented by arrows. In the top row, the cluster separation is seen in location but not in speed. In the bottom row, the clusters overlap completely in location but are separated in speed: Some arrows point to the upper right, others to the lower left, indicating the presence of two groups.*

the additional information provided by velocity is confined to one additional dimension. This generalizes to higher dimensions: The velocity frame F' can be rank-deficient; in particular when $d > p/2$, its rank must be less than that of F.

We can ask how the location and velocity projections relate to each other. To this end, we use the notion of whip spin. We consider the two extremes of 0% whip spin and 100% whip spin:

- If the frame motion has 0% whip spin, the location frame $F(t)$ and the velocity frame $F'(t)$ produce projections onto orthogonal subspaces. This follows because absence of whip spin, $F^T F' = 0$, is equivalent to orthogonality of span(F) and span(F'). The combined dimensionality "seen" in the dynamic projection is $d + \text{rank}(F')$, which can be as much as $2 \cdot d$.
- If the frame motion has 100% whip spin, then the velocity frame produces a projection onto a subspace of the location subspace: span$(F') \subset$ span(F). The velocity frame F' therefore presents all or part of the information contained in the location frame F, although in a different orientation, and possibly affinely distorted. In a full-dimensional tour with $d = p$, the benefit of motion is that

two different frames F and F' are visible at the same time.

A caveat: We do, of course, not argue that the quality of the information contained in location and velocity are perceptually equivalent. In fact, for the usual rendering methods such as scatterplots, location is vastly superior to velocity. Yet, velocity is not useless: It adds glimpses of additional dimensions that provide a much more holistic view of the geometry of a pointcloud in high-dimensional data space. — As an illustration, we attempt to depict a moving scatterplot of clustered data in figure 10. The point of the figure is to show the cluster structure coded in location but not velocity (top row), and coded in velocity but not location (bottom row), respectively. In practice, the coding mode changes from one moment to the next: What is coded as location now may be coded as speed later, and vice versa.

9 Kinematics II: Speed of Moving Frames and Planes

9.1 Quadratic speed measures

In implementations of dynamic projections, it is desirable to provide viewers with a sense of constancy of motion. This is not just a psychological need but a consideration of practical usefulness: When watching a tour, a viewer is often able to judge the strength of structure by how often it becomes visible. If tour motion is not steady, such judgement is impossible or biased. Erratic motion that changes unpredictably between slow and fast perceived speed is therefore not only disconcerting but potentially misleading.

For tour implementations that are based on interpolation, it is desirable not only to hold speed steady on each leg, but to stitch interpolating paths together in such a way that speed is perceived as about the same on each leg.

The question of how to measure speed does not have a unique answer, even after making strong invariance assumptions and limiting the choices to algebraically simple forms of speed measures.

Reasonable measures of speed are matrix (semi-)norms of the frame derivative F'. In what follows we consider only speed measures that are derived from quadratic forms of F':

$$g_F(F') \;=\; \sum_{ijkl} g_{ijkl}(F)\; {F'}_{ij}{F'}_{kl} \;,$$

where the coefficients $g_{ijkl} = g_{ijkl}(F)$ may depend on the current frame F. Essentially, the derivative F' is strung out as a $p \cdot d$-dimensional vector, and g_F is a quadratic function thereof. The necessity of allowing $g = g_F$ to depend on the current frame F stems from the fact that the space of derivatives F' differs for different F's.

Assuming g_F is non-negative, speed at F is measured by $g_F(F')^{1/2}$. In order to simplify language, we use the term "speed measure" for both $g_F(F')^{1/2}$ and $g_F(F')$, although only $g_F^{1/2}$ is speed in the proper sense.

The simplest example of a quadratic speed measure is the Euclidean squared norm of F' considered as a $p \cdot d$-vector, which is also called the squared Frobenius norm of F':

$$g_F(F') = \|F'\|_{Frob}^2 = \sum_{ij} F'^2_{ij} = \mathrm{trace}(F'^T F') = \mathrm{trace}(F' F'^T) \ .$$

This example is intuitive as we will see below, but it is atypical because the coefficients do not depend on F: $g_{ijkl} = 0$ for $(ij) \neq (kl)$ and $g_{ijij} = 1$.

9.2 Reducing the set of speed measures: Invariance requirements

We need to develop criteria to whittle down the universe of speed measures to an intuitively plausible subset. Criteria that are particularly intuitive and mathematically powerful are invariance requirements under orthogonal transformations of both

- the p-dimensional data space and
- the d-dimensional projection plane.

Requiring invariance is a normative act; it is not something that can be derived from still higher principles. One can, however, make informal arguments: Invariance is desirable because the notion of speed should be devised a priori without looking at the structure of particular data. A priori we do not have knowledge of interesting projections of data space, that is, projections near which it might be worthwhile to slow down and linger. Neither do we have a priori knowledge about how to orient the projection plane to our advantage. The purpose of the two invariance requirements is to level the playing field among dynamic projections and their orientations. While it is true that potentially interesting data projections can be found with multivariate analysis and projection pursuit, this fact is quite irrelevant when trying to gauge speed measures of data projections; the only data that can possibly serve as a gauge are structureless or null data. A suitable set of null data is formed by data that "look the same from all sides," or equivalently, that are rotationally symmetric, such as multivariate standard normal data, which incidentally serve as a null gauge for exploratory projection pursuit as well (Friedman 1987). We will make use of null data in subsection 9.4.

Underlying any invariance requirement with regard to orthogonal transformations is of course a Euclidean metric in data space. We have made use of Euclidean metrics from the moment when we required projection frames to be orthonormal and projections to be orthogonal. The necessity of Euclidean metrics is unquestioned, although no single best metric may exist in any particular data analysis.

9.3 The structure of invariant speed measures

We proceed with the mathematical definition of invariance. Denote by $O(p)$ the group of orthogonal transformations U ($U^T U = I_p$) of p-dimensional data space. Orthogonal transformations comprise proper rotations (determinant $+1$), reflections, and their compositions. Similarly, denote by $O(d)$ the group of orthogonal transformations V ($V^T V = I_d$) of the d-dimensional projection plane.

Definitions: *A speed measure* $g_F(F')$ *is*

- **left-invariant** *if two paths* $F(t)$ *and* $UF(t)$ *have the same speeds:* $g_{UF}(UF') = g_F(F')$ *for all* $U \in O(p)$ *and all* t*;*
- **right-invariant** *if two paths* $F(t)$ *and* $F(t)V$ *have the same speeds:* $g_{FV}(F'V) = g_F(F')$ *for all* $V \in O(d)$ *and all* t*.*

Note that the transformations U and V are fixed: They do *not* vary with time t.

The speed measure based on the Frobenius norm, $g_F(F') = \|F'\|^2_{Frob}$, is both left-invariant and right-invariant, as is easily seen with basic trace manipulations: $\mathrm{trace}((UF'V)^T(UF'V)) = \mathrm{trace}(F'^T F')$.

In preparation of the main result of this subsection, we introduce some definitions: For any speed measure g_F there are two associated component speed measures derived from the decomposition of frame motion F' into whip spin PF' and plane motion $(I-P)F'$:

1. *Frame speed:* $g_F(F')$,
2. *Whip speed:* $g_F(PF')$,
3. *Plane speed:* $g_F((I-P)F')$,

where each is really squared speed. The following theorem states two consequences of invariance: 1) Frame speed has a Pythagorean decomposition into whip speed and plane speed, and 2) whip speed and plane speed are just Frobenius norms up to a scalar factor:

Theorem 2: *Squared speed measures* $g_F(F')$ *that are both left-invariant and right-invariant are of the form*

$$\begin{aligned} g_F(F') &= \alpha_w \cdot \|PF'\|^2_{Frob} \;+\; \alpha_p \cdot \|(I-P)F'\|^2_{Frob} \\ &= \alpha_w \cdot \|F^T F'\|^2_{Frob} \;+\; \alpha_p/2 \cdot \|P'\|^2_{Frob} \end{aligned}$$

for some $0 \le \alpha_w, \alpha_p \le 1$.

The proof is in the Appendix (sections 15.2 and 15.3).

The first equality is the main result. The second equality is an algebraic reformulation with intuitive appeal: Whip speed is just a function of the whip spin matrix, and plane speed is just a function of the path of projections $P = P(t)$.

With theorem 2 we are down to speed measures that are not too different from the basic Frobenius norm: For $\alpha_w = \alpha_p = 1$, we have $g_F(F') = \|F'\|^2_{Frob}$. Because speed measures are obviously equivalent if they differ only by a scale factor, the remaining freedom is that of choosing the relative weighting of whip speed and plane speed, making this essentially a one-parameter family of speed measures.

Before we go into specific choices, here is a short discussion of cases where the choice is irrelevant:

- **Pure plane motion:** These are the preferred paths for tours that use low-dimensional planes, typically $d \leq 3$, with rendering methods for which orientation is unimportant. For such paths there is essentially only one way to measure speed, modulo an irrelevant choice $\alpha_p = 1$. Because $PF' = 0$, we have $(I - P)F' = F'$, hence
$$g_F(F') = \|F'\|^2_{Frob} = 1/2 \cdot \|P'\|^2_{Frob} ,$$
up to an irrelevant choice $\alpha_p = 1$. This is only a function of the underlying plane path P, as it should be.
- **Pure whip spin:** These are the paths used when the plane is at rest as in full-dimensional tours with $d = p$. Again, there is essentially only one way to measure speed: Because $(I - P)F' = 0$, we have
$$g_F(F') = \|F'\|^2_{Frob} = \|F^T F'\|^2_{Frob} ,$$
up to an irrelevant choice $\alpha_w = 1$. In particular this implies that for paths of $p \times p$ frames $U(t) \in O(p)$ the invariant speed measure is essentially unique: $g_U(U') = \|U'\|^2_{Frob}$.

9.4 Choosing among speed measures that are left- and right-invariant

The relative weighting of whip speed and plane speed is not something that can be decided on mathematical grounds as strong as left- and right-invariance. We know of auxiliary arguments in favor of two particular choices:

- $\alpha_w = 1$, $\alpha_p = 1$,
- $\alpha_w = 1$, $\alpha_p = 2$.

Average squared speed of projected null data: For the speed measure defined by $\alpha_w = 1$, $\alpha_p = 1$ ($\|F'\|^2_{Frob}$) there exists an interpretation in terms of average squared speed of null data as seen in the projection. Null data is data with no or little structure in a sense to be specified. It is intuitive to choose null data to gauge overall speed of a moving point cloud on a computer screen: If speed — as quantified by a speed measure — is held constant, the viewer should perceive null data as moving at constant overall speed. In practice this can be used as follows: When viewing real data, perceived variations in overall speed act as auxiliary indicators for the presence of structure.

In order to specify the exact meaning of "null data," it is convenient to use random vectors rather than finite data. Let $\boldsymbol{X}$ be a p-dimensional random vector with existing second moments ($\mathrm{E}[\, \|\boldsymbol{X}\|^2 \,] < \infty$, E denotes expectation). Null data for our purposes are formed by a random vector with spherically symmetric second moments: $\mathrm{E}(\boldsymbol{X}\boldsymbol{X}^T) = \alpha \cdot I_p$. That is, this data "looks the same" from all sides as far as second moments are concerned.

The projection of $\boldsymbol{X}$ onto a moving d-frame $F = F(t)$ is the time-dependent d-dimensional random vector $F^T\boldsymbol{X}$, with velocity $F'^T\boldsymbol{X}$ (another time-dependent random vector) and squared speed $\|F'^T\boldsymbol{X}\|^2$ (a time-dependent random variable; note this is the Euclidean norm in $\mathbb{R}^d$).

Fact: *If the second moment matrix is rotationally symmetric,* $\mathrm{E}(\boldsymbol{X}\boldsymbol{X}^T) = \alpha \cdot I_p$*, then the average squared speed of the projected data is*

$$\mathrm{E}[\, \|F'^T\boldsymbol{X}\|^2 \,] \;=\; \alpha \cdot \|F'\|_{Frob}^2 \;.$$

The proof is elementary, observing that $\|F'^T\boldsymbol{X}\|^2 = \mathrm{trace}(F'F'^T\boldsymbol{X}\boldsymbol{X}^T)$.

An example of second-order null data is the p-dimensional standard normal, but so is any spherically symmetric distribution in $\mathbb{R}^p$. Because we do not require centered variables, "data" consisting of the variable unit vectors form another example: Let $\boldsymbol{X}$ be a random vector that takes on each of the p variable unit vectors with probability $1/p$; the second moment matrix is $1/p \cdot I_p$. Thus, $\|F'\|_{Frob}^2$ can be interpreted as essentially the average squared speed of the projected variable unit vectors.

Average squared speed of rotating null data: For the speed measure defined by $\alpha_w = 1$, $\alpha_p = 2$, there exists an interpretation in terms of the intuitive notion of "data rotation." We implicitly used this speed measure in Asimov and Buja (1994) because it is mathematically the simplest choice (see section 10).

The term "data rotation" suggests rotating data space rather than the projection plane. Hence we fix the projection frame once for all, for example, at the frame E_d consisting of the first d variable unit vectors. Let $U = U(t) \in SO(p)$ be a path of rotations of p-dimensional data space ($SO(p)$ is the group of orthogonal maps on $\mathbb{R}^p$ with determinant $+1$).

We rotate the data with the inverse of U, that is, a data vector $\boldsymbol{x}_i$ gets mapped to $U(t)^T\boldsymbol{x}$. We then project it onto the first d variable directions: $View_i(t) = E_d^T U(t)^T \boldsymbol{x}_i$. In effect, this amounts to a projection onto the frame $F(t) = U(t)E_d$, that is, the first d columns of U. We write $U(t) = (F(t), G(t))$, where G is the complement frame of size $p \times (p-d)$. The velocity vector of the rotating data vector $U(t)^T\boldsymbol{x}_i$ in p-space is $U'(t)^T\boldsymbol{x}_i$, and the squared speed is $\|U'(t)^T\boldsymbol{x}_i\|^2$. Describing null data again by a random vector $\boldsymbol{X}$ with rotation symmetric second moments $\mathrm{E}[\,\boldsymbol{X}\boldsymbol{X}^T\,] = \alpha \cdot I_p$, the average squared speed of the rotating data $U^T\boldsymbol{X}$ is

$$\mathrm{E}[\; \|U'(t)^T\boldsymbol{X}\|^2 \;] \;=\; \alpha \cdot \|U'(t)\|_{Frob}^2 \;,$$

which makes sense because $\|U'\|_{Frob}$ is essentially the only way to measure speed of "full-dimensional frames" $U(t)$.

We now examine $\|U'(t)\|_{Frob}$ as a potential speed measure for the moving frame $F(t) = U(t)E_d$. We start with the following rationale: When measuring speed of a data rotation $U = (F, G)$ as visible in the projection onto F, it should be required that the invisible whip spin in the orthogonal complement does not contribute. Therefore, we ask that the complement frame G have zero whip spin, which is possible by theorem 1. The result is:

Fact: *If a path of d-frames $F(t)$ is augmented with a path of $(p-d)$-frames $G(t)$ that have zero-whip spin to form a path of data rotations $U(t) = (F(t), G(t))$, then the speed of the data rotation is measured by*

$$\|U'\|^2_{Frob} \;=\; \|PF'\|^2_{Frob} \;+\; 2 \cdot \|(I-P)F'\|^2_{Frob} \;.$$

The proof is in section 15.9. As a result, we have an interpretation of the speed measure corresponding to $\alpha_w = 1$ and $\alpha_p = 2$ in terms of speed of most restful rotations of null data with regard to the path of frames $F(t)$.

10 Optimal Paths of Frames

Metrics on paths of frames do not only allow us to measure speed but to find shortest paths between two given frames. The interpretation for data visualization is that such paths carry one data projection onto another data projection in the most restful manner.

As we have learned, there exists a multiplicity of metrics that are reasonable. This poses a quandary: Each metric has its own shortest paths. In two specific situations, however, the quandary does not exist: 1) when using paths of frames that consist of pure plane motion, as in tours of low-dimensional projections where orientation is irrelevant; and 2) when using paths of frames that consist of pure whip spin, as in full-dimensional tours. This is not too surprising because we found that both pure plane motion and pure whip spin are measured in essentially unique ways, the choice arising from combining the two speed components into an overall speed measure.

In what follows, we replace the notion of "shortest path" with that of "geodesic path," which means "locally shortest." This notion is more useful and more convenient on a curved manifold such as the one formed by d-frames in p-space.

In order to introduce geodesic paths of frames, we use a simplification:

Convention: *The starting frame F_a is the unit frame E_d.*

The case of a general starting frame F_a is obtained by mapping E_d with some $U_a \in SO(p)$ to F_a, so that paths $F(t)$ through E_d map to paths $U_a F(t)$ through F_a. Due to left-invariance of speed measures, $F(t)$ and $U_a F(t)$ have the same speed properties, which implies that a geodesic through E_d will be mapped to a geodesic through F_a.

In order to describe geodesic paths of frames, we need some facts about the connection between rotations and their infinitesimal generators via matrix exponentials $\exp(S) = \sum_0^\infty S^n/n!$:

Facts:

- *Any skew matrix S of size $p \times p$ can be block-diagonalized with skew blocks of size 2×2 in a suitable orthonormal coordinate system.*
- *Any rotation $U \in SO(p)$ can be block-diagonalized with blocks consisting of 2×2 rotations (and a trailing +1 if p is odd) in a suitable orthonormal coordinate system.*
- $\exp \begin{pmatrix} 0 & -t \\ t & 0 \end{pmatrix} = \begin{pmatrix} c_t & -s_t \\ s_t & c_t \end{pmatrix}$
- *For all $p \times p$ skew matrices S:* $\exp(S) \in SO(p)$.

The first two facts show how the third fact implies the fourth, all of which is a consequence of the spectral theorem for unitary transformations and its translation to orthogonal transformations (Halmos 1958, section 81 and 82).

Exponentials of skew matrices can be used to generate paths of rotations $U(t)$ by spinning at constant but different speeds in sets of mutually orthogonal 2-planes. This is expressed by $U(t) = \exp(St)$. With this background, we can formulate a class of paths of frames that contain all geodesics for all left- and right-invariant metrics:

$$(*) \qquad F(t) = \exp(St)\, E_d\, \exp(Qt)\,, \quad S \text{ is } p{\times}p,\ Q \text{ is } d{\times}d, \text{ both skew.}$$

The $p \times p$ path $\exp(St)$ transports the starting frame E_d through p-space, while the $d \times d$ path $\exp(Qt)$ applies an additional whip spin to $\exp(St)\, E_d$. This class of paths of frames is quite rich. In order to sort out the members that are geodesic w.r.t. a given metric, we need notation for the canonical blocks of S:

$$S = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} \qquad \begin{array}{l} \} \; d \\ \} \; (p-d) \end{array}$$

Theorem 3: *The geodesic paths of frames with regard to the metric given by α_w and α_p are exactly the paths of the form $(*)$ satisfying $Q = (\frac{\alpha_p}{2\alpha_w} - 1)S_{11}$ and $S_{22} = 0$.*

The proof is in the Appendix (sections 15.5 and 15.6).

This theorem is surprisingly hard to apply when the goal is to interpolate $F_a = E_d$ and an arbitrary target frame F_z. Note that for a given target F_z the problem is to construct a suitable S and its associated Q such that $F_z = \exp(S)\, E_d \exp(Q)$. In Asimov and Buja (1994) we solved the problem for 2-frames with regard to the metric defined by $\alpha_w = 1$ and $\alpha_p = 2$, which is surprisingly the simplest case because Q magically vanishes: $Q = (\frac{\alpha_p}{2\alpha_w} - 1)S_{11} = 0$. The construction of S from F_z is still an open problem for all other metrics, including the straight Frobenius metric defined by $\alpha_w = \alpha_p = 1$.

The situation is not so bleak, though, because two cases of particular interest are readily solved: 1) pure plane motion for low-dimensional tours where orientation is irrelevant, and 2) pure whip spin for full-dimensional tours.

Corollary: *Paths of theorem 3 are universally geodesic for all left- and right-invariant metrics in the following cases:*
- $S_{11} = 0$, *that is, pure plane motion;*
- $S_{12} = 0$, *that is, pure whip spin.*

In the first case $Q = 0$ by necessity, and in the second case Q can be absorbed into S_{11}, hence $Q = 0$ w.l.o.g.; thus $F(t) = \exp(St)\, E_d$. The matrix S is of the following form, respectively:

$$S = \begin{pmatrix} 0 & S_{12} \\ S_{21} & 0 \end{pmatrix}, \qquad S = \begin{pmatrix} S_{11} & 0 \\ 0 & 0 \end{pmatrix}.$$

The first case is steady rotation vertically out of the current plane, and the second case is equally steady rotation within the current plane. In order to fully appreciate this interpretation, it helps to cast S and $\exp(St)$ in canonical form as mentioned in the list of facts at the beginning of the section. Here are two examples:

- Pure plane motion of a 2-frame in 4-space ($d = 2$, $p = 4$, canonical coordinates): The 1- and 2-axis in the starting plane are being rotated towards the 3- and 4-axis at speeds u and v, respectively. The (1,3)-plane and the (2,4)-plane are invariant.

$$S = \begin{pmatrix} 0 & 0 & -u & 0 \\ 0 & 0 & 0 & -v \\ u & 0 & 0 & 0 \\ 0 & v & 0 & 0 \end{pmatrix} : \qquad \exp(St) = \begin{pmatrix} c_{ut} & 0 & -s_{ut} & 0 \\ 0 & c_{vt} & 0 & -s_{vt} \\ s_{ut} & 0 & c_{ut} & 0 \\ 0 & s_{vt} & 0 & c_{vt} \end{pmatrix},$$

- Pure whip spin of a 4-frame in 4-space ($d = p = 4$, canonical coordinates): The (1,2)-plane and the (3,4)-plane are invariant in the full 4-space. The rotation speeds are u and v, respectively.

$$S = \begin{pmatrix} 0 & -u & 0 & 0 \\ u & 0 & 0 & 0 \\ 0 & 0 & 0 & -v \\ 0 & 0 & v & 0 \end{pmatrix} : \qquad \exp(St) = \begin{pmatrix} c_{ut} & -s_{ut} & 0 & 0 \\ s_{ut} & c_{ut} & 0 & 0 \\ 0 & 0 & c_{vt} & -s_{vt} \\ 0 & 0 & s_{vt} & c_{vt} \end{pmatrix}.$$

Note that for $d = p$, the condition $S_{12} = 0$ is trivially satisfied because S_{12} is of size $d \times (p - d) = d \times 0$.

In computer implementations of geodesic interpolation, the problem is in either case to find the canonical coordinates in which a path can be expressed as simply as

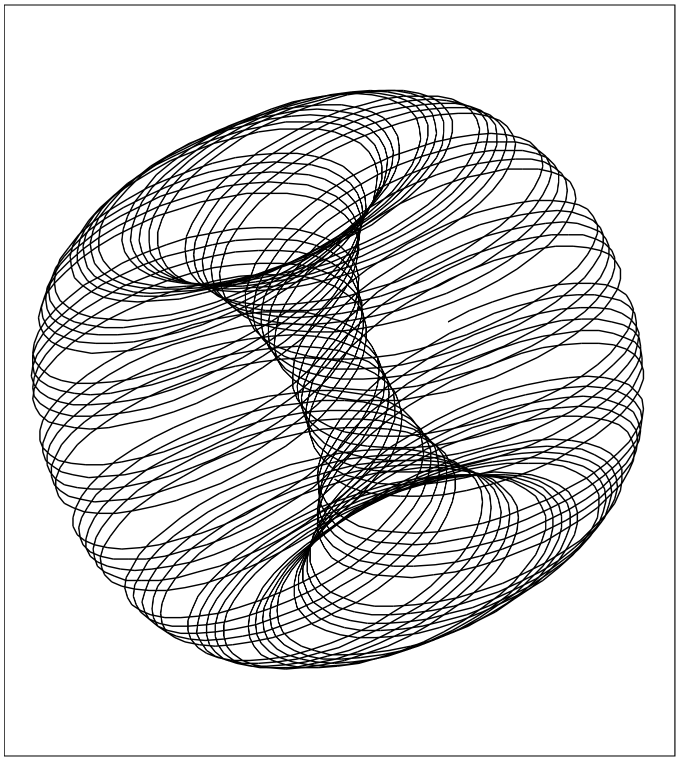

Figure 11: *A random 2-projection of a geodesic path of 2-frames in 4-space, interpreted as a curve in 8-space (or 4-space after eliminating four zeros). The figure shows a finite piece of a single non-intersecting path filling up a surface that is topologically a 2-D torus.*

above. Details are carried out in section 12 for pure plane motion, and in section 13.1 for pure whip spin.

We can gain some intuition into the structure of geodesic paths by visualizing a moving frame with the same means that they are intended to serve: projections from high-dimensional space. Consider a path of 2-frames in 4-space,

$$F(t) = \exp(St)E_d = ((c_{ut}, 0, s_{ut}, 0)^T, (0, c_{vt}, 0, s_{vt})^T) \ ,$$

generated by the above example of pure plane motion. Interpret $F(t)$ as a curve in $\mathbb{R}^8$, eliminate the trivial zeros to make this a curve in $\mathbb{R}^4$ and form 2-D projections of the curve. A random projection from a grand tour in XGobi is shown in figure 11. The particular curve has $u = .1$ and $v = 2^{1/2}$. It should not be too surprising that it seems to be filling up a torus-like surface. The underlying mathematical reason is that the mapping $(u, v) \mapsto (c_u, s_u, c_v, s_v)$ is a parametrization of a 2-D torus in 4-space, and $t \mapsto (c_{ut}, s_{ut}, c_{vt}, s_{vt})$ is a dense curve on the torus iff u/v is irrational.

11 Tools for Constructing Plane and Frame Interpolations: Preprojection and Planar Rotations

We turn from mathematical to algorithmic considerations. The goal is to outline a computational framework that can serve as the base of an interactive system for data visualization with dynamic projections.

11.1 Preprojection

The point of the following considerations is to reduce the problem of path construction to the smallest possible subspace and allow for the necessity of constructing particular bases in which interpolation can be carried out simply.

A path of frames $F(t)$ that interpolates two frames F_a and F_z should be parsimonious in the sense that it should not traverse parts of data space that are unrelated to F_a and F_z. For example, if both these frames exist in the space of variables $x_1, \ldots, x_5$, the path $F(t)$ should not make use of variable x_6. In general terms, $F(t)$ should live in the joint span of F_a and F_z.

This is a handy requirement because it allows preprojection of the frames and the data to the often lower-dimensional subspace spanned by F_a and F_z. The cost of computing the initial preprojection of the data is of course proportional to the dimension p of data space, but thereafter the cost of computing data projections along the path is essentially proportional to the dimension of the preprojection subspace, which can be substantially smaller than p when d is low and p is high.

The preprojection step is as follows: Form an arbitrary orthonormal basis of the joint span of F_a and F_z, for example, by applying Gram-Schmidt to F_z with regard to F_a. If the dimension of span(F_a, F_z) is d_S, the resulting basis frame $B = (\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{d_S})$ has d_S vectors. Note that

$$d_S = 2d - d_I \ , \quad \text{for} \quad d_S = \dim(\text{span}(F_a, F_z)) \ , \quad d_I = \dim(\text{span}(F_a) \cap \text{span}(F_z)) \ .$$

We have $d_S = 2d$ whenever $d_I = 0$, that is, the starting and target plane intersect only at the origin. In the other extreme case, when the two planes are identical, we have $d_S = d_I = d$, which characterizes situations of pure whip spin such as full-dimensional tours. In this latter case preprojection is *not* vacuous: Most interpolation algorithms require particular (for example canonical) bases in order to simplify interpolation.

We can now express the original frames in this basis:

$$F_a \ = \ BW_a \quad \text{and} \quad F_z \ = \ BW_z \ ,$$

where $W_a = B^T F_a$ and $W_z = B^T F_z$ are orthonormal frames of size $d_S \times d$. This represents the preprojection of the frames. The preprojection of the data is given by

$$\boldsymbol{\xi}_i \ = \ B^T \boldsymbol{x}_i \ .$$

The problem is now reduced to the construction of paths of frames $W(t)$ that interpolate the preprojected frames W_a and W_z. The corresponding path in data space is

$$F(t) \;=\; BW(t)$$

and the viewing coordinates of a data vector $\boldsymbol{x}_i$ are

$$View_i(t) \;=\; W(t)^T \boldsymbol{\xi}_i \;.$$

11.2 Planar rotations

The basic building blocks for constructing paths of frames are planar rotations, that is, rotations that have an invariant 2-D plane with action corresponding to

$$\begin{pmatrix} c_\tau & -s_\tau \\ s_\tau & c_\tau \end{pmatrix}$$

and an orthogonal complement of fixed points.

If the action is in the plane of variables i and j, we denote the rotation by $R_{ij}(\tau)$, which is then also called a *Givens rotation*. Note that the order of i and j matters: $R_{ij}(\tau) = R_{ji}(-\tau)$. For efficiency, $R_{ij}(\tau)$ is never stored explicitly; matrix multiplications involving $R_{ij}(\tau)$ are directly computed from i, j, and τ. See Golub and Van Loan (1983), section 3.4, for computational details (note that their $J(i,k,\theta)$ is our $R_{ik}(-\theta)$).

The methods we introduce in the following sections are based on the composition of a number of Givens rotations in a suitable coordinate system. The fundamental step is always the construction of a composition that maps the starting frame onto the target frame. Writing $R_\mu(\tau_\mu)$ for $R_{i_\mu j_\mu}(\tau_\mu)$, this is

$$W_z \;=\; R_m(\tau_m) \;\ldots\; R_2(\tau_2)\; R_1(\tau_1)\; W_a$$

in the preprojection. We arrive at an interpolating path of frames by simply inserting a time parameter t into the formula:

$$W(t) \;=\; R_m(\tau_1 t) \;\ldots\; R_2(\tau_2 t)\; R_1(\tau_1 t)\; W_a \;,$$

where $0 \le t \le 1$. Obviously, $W(0) = W_a$ and $W(1) = W_z$. In compact notation we write

$$R(\boldsymbol{\tau}) \;=\; R_m(\tau_m) \;\ldots\; R_2(\tau_2)\; R_1(\tau_1) \;, \qquad \boldsymbol{\tau} = (\tau_1, \ldots, \tau_m) \;.$$

Interpolating paths based on rotations are never unique, for obvious reasons: If $W_z = R(\boldsymbol{\tau})W_a$, then any other $\tilde{\boldsymbol{\tau}}$ with $\tilde{\tau}_j = \tau_j + k_j \cdot 2\pi$, k_j integer, also satisfies $W_z = R(\tilde{\boldsymbol{\tau}})W_a$. Among all these candidates, one usually selects the $\boldsymbol{\tau}$ that is closest to the vector of zero angles.

11.3 Calculation, implementation and control of speed

Assuming a speed measure for moving frames has been chosen, we can step along paths of frames in such a way that the motion has constant speed with regard to the chosen measure, thus providing a sense of constancy of motion.

For actual calculations, it is convenient to rewrite the speed measures of theorem 2 as follows:

$$g_F(F') \;=\; \alpha_p \cdot \|F'\|^2_{Frob} \;+\; (\alpha_w - \alpha_p) \cdot \|F^T F'\|^2_{Frob}$$

Thus there is only a need to compute the norm of the velocity frame F' and the norm of its whip spin $F^T F'$. For $\alpha_w = \alpha_p = 1$ only $\|F'\|_{Frob}$ has to be computed.

In order to control speed in computer implementations, the following considerations are elementary: Computer animations are generated by updating the display at a fast but *constant* rate (at least 5-10 per second). This implies that animation speed is not usually controlled by varying the update rate but by varying the step size along the motion path: Wider steps produce faster motion.

For dynamic projections, this implies that a path of frames is discretized, and speed is controlled by proper choice of the step size of the discretization. Incrementally, this means that at time t, one steps from $F(t)$ to $F(t+\Delta)$, amounting to a distance $\int_t^{t+\Delta} g_F(F'(\tau))^{1/2} d\tau$. Approximate constancy of speed can be provided as follows. Using the first order approximation

$$\text{StepSize} \;=\; \int_t^{t+\Delta} g_{F(\tau)}(F'(\tau))^{1/2} d\tau \;\approx\; g_{F(t)}(F'(t))^{1/2} \cdot \Delta \;,$$

we can choose the increment Δ as

$$\Delta \;=\; \text{StepSize}/g_{F(t)}(F'(t))^{1/2} \;,$$

where StepSize is a user-chosen speed parameter that expresses something akin to "degrees of motion per update." Typically, the user does not need to know actual values of StepSize because this quantity is controlled interactively through graphical gauges. [As a recommendation to implementors, such gauges should not represent StepSize linearly: At slow speeds it is important to be offered very precise control, while for high speeds it is more important to be offered a large range than high precision.]

We note that all speed calculations can be carried out in the preprojection: Because $B^T B = I_{d_S}$ we have $g_F(F'(t)) = g_W(W'(t))$ for the invariant speed measures of theorem 2.

Abbreviating the concatenated planar rotations of subsection 11.2 and their derivatives by $R = R(\boldsymbol{\tau} t)$ and $R' = d/dt\; R(\boldsymbol{\tau} t)$, respectively, we have $W' = R' W_a$, and the speed measures of theorem 2 in the form given above become

$$g_W(W') \;=\; \alpha_p \cdot \|R' W_a\|^2_{Frob} \;+\; (\alpha_w - \alpha_p) \cdot \|W_a^T R^T R' W_a\|^2_{Frob} \;.$$

When R is a complex concatenation of many planar rotations, it can be impossible or at least messy to get at R' analytically. It may then be advantageous to approximate R' numerically by actually computing the matrix $R(\boldsymbol{\tau}t)$ for two close values of t, and calculating a finite difference quotient,

$$R' \approx (R(\boldsymbol{\tau}(t+\delta)) - R(\boldsymbol{\tau}t))/\delta$$

which can be substituted for R'. — In some cases, such as the optimal paths of sections 12 and 13.1, there exist explicit formulas for speed.

11.4 Outline of an algorithm for interpolation

In the following we give the structure of an interpolation algorithm. Given starting and target frames F_a and F_z, the algorithm assumes that we know how to construct a preprojection basis B and a sequence of planar rotations $R(\boldsymbol{\tau}) = R_m(\tau_m)\ldots R_1(\tau_1)$ that map $W_a = B^T F_a$ to $W_z = B^T F_z$. The construction of B and $R(\boldsymbol{\tau})$ is dependent on the interpolation method, and in the following sections we give several such constructions.

Algorithm:

1. *Given a starting frame F_a, create a new target frame F_z.*
2. *Initialize interpolation:*
   - *Construct a preprojection basis B of* $\mathrm{span}(F_a, F_z)$.
     (B has d_S columns, where $d_S = \dim(\mathrm{span}(F_a, F_z))$; $d_S \geq d$.)
   - *Preproject the frames: $W_a = B^T F_a$, $W_z = B^T F_z$.*
   - *Check: If starting and target plane are the same: $d = d_S$, and if starting and target frame have opposite orientations:* $\det(W_a^T W_z) = -1$, *then interpolation within the span is impossible.*
     Possible remedy: flip one frame vector, $\boldsymbol{f}_{z,d} \leftarrow -\boldsymbol{f}_{z,d}$.
   - *Construct planar rotations in coordinate planes:*
     $R(\boldsymbol{\tau}) = R_m(\tau_m)\ldots R_1(\tau_1)$, $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_m)$,
     such that $W_z = R(\boldsymbol{\tau})W_a$.
   - *Preproject the data: $\boldsymbol{\xi}_i = B^T \boldsymbol{x}_i$, $i = 1, ..., N$.*
   - *Initialize: $t \leftarrow 0$.*
3. *Execute interpolation; iterate the following:*

```
    $t \leftarrow \min(1, t)$
    $W(t) \leftarrow R(\boldsymbol{\tau}t)W_a$
    $View_i(t) \leftarrow W(t)^T \boldsymbol{\xi}_i$, $i = 1, ..., N$    (render data)
    $F(t) \leftarrow BW(t)$                                             (render frame)
    If $t = 1$: break iterations.
    Else: $\Delta \leftarrow \mathrm{StepSize}/g_W(W')^{1/2}$, $t \leftarrow t + \Delta$, do next iteration.
```

4. *Set $F_a \leftarrow F_z$ and go to beginning.*

For interpolation of planes as opposed to frames, the only difference is that the target frame W_z is generally replaced with another target frame that spans the same plane. (In plane interpolation, when the starting plane and the target plane are the same, no interpolation is necessary; hence the problem of opposite orientations of frames is nonexistent.)

In the following sections, we will only specify the construction of B and $R(\boldsymbol{\tau})$ and refer the reader to the above algorithm.

When the algorithm is integrated in a larger interactive visualization system such as XGobi, a number of additional measures have to be taken at each iteration because interactions by the user may have reset some of the parameters:

- Read the speed parameter StepSize.
- Check whether a new subspace of data space has been selected. Most often a new subspace is specified in terms of a new subset of data variables.
- Check whether another method of target generation has been selected.

If one of the last two checks is positive, the current path has to be interrupted and a new path has to be initialized with the proper changes.

The two lines marked "*render*" imply execution of the rendering mechanisms for generating a new graphical scene from the data projection and for giving the viewer feedback about the position of the current projection. See sections 3 and 4.

12 Interpolating Paths of Planes

We describe paths of frames that optimally interpolate planes, that is, they interpolate frames only modulo orientation, but they have zero whip spin and are locally shortest (geodesic) with regard to the metrics discussed in section 9.3. The interpolation scheme based on these paths is appropriate for rendering methods that are visually invariant under changes of orientation (section 6). The construction of the paths is based on the following:

Fact: *Given two planes of dimension d, there exist orthonormal d-frames G_a and G_z that span the planes and for which $G_a^T G_z$ is diagonal.*

Optimal paths of planes essentially rotate the columns of G_a into the corresponding columns of G_z. Denoting the columns of G_a and G_z by $\boldsymbol{g}_{a,j}$ and $\boldsymbol{g}_{z,j}$, respectively, we see that the planes spanned by $\boldsymbol{g}_{a,j}$ and $\boldsymbol{g}_{z,j}$ are mutually orthogonal for $j = 1, 2, ..., d$, and they are either 1-dimensional (if $\boldsymbol{g}_{a,j} = \boldsymbol{g}_{z,j}$) or 2-dimensional (otherwise). The motion is carried out by a moving frame $G(t) = (\boldsymbol{g}_1(t), ..., \boldsymbol{g}_d(t))$ as follows: If $\boldsymbol{g}_{a,j} = \boldsymbol{g}_{z,j}$, then $\boldsymbol{g}_j(t) = \boldsymbol{g}_{a,j}$ is at rest; otherwise $\boldsymbol{g}_j(t)$ rotates from $\boldsymbol{g}_{a,j}$ to $\boldsymbol{g}_{z,j}$ at constant speed proportional to the angle between $\boldsymbol{g}_{a,j}$ and $\boldsymbol{g}_{z,j}$. It is clear that $G(t)$ has the structure described in section 10 and is hence universally geodesic.

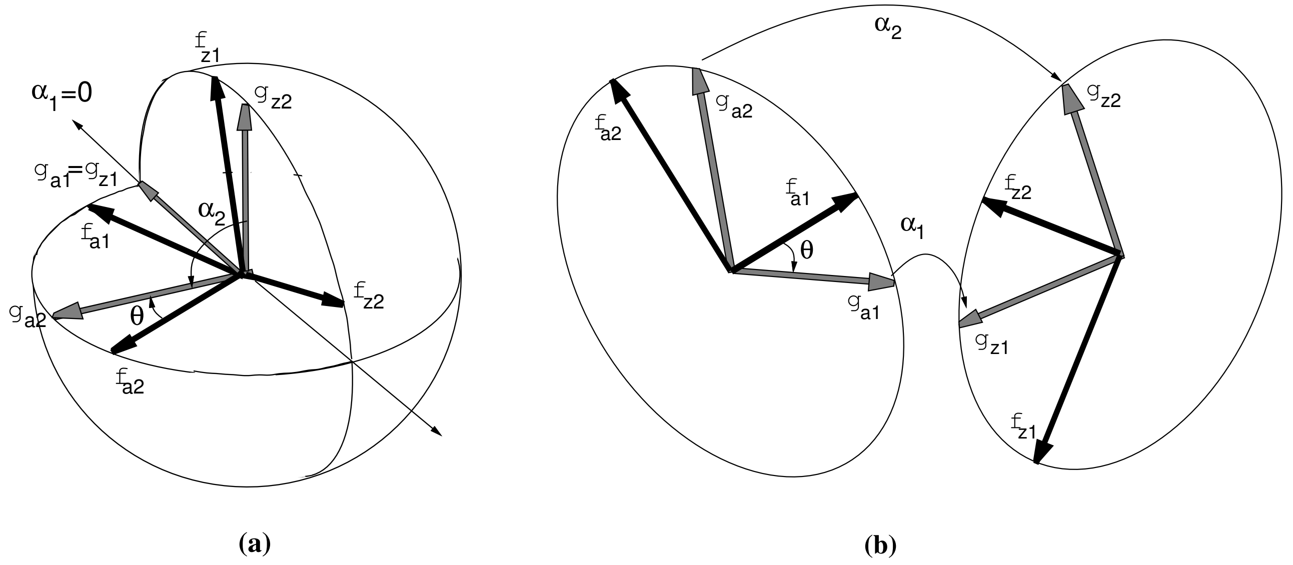

Figure 12: *Relative positions of given orthonormal bases,* $(\boldsymbol{f}_{a,1}, \boldsymbol{f}_{a,2})$, $(\boldsymbol{f}_{z,1}, \boldsymbol{f}_{z,2})$, *and pairs of principal directions,* $(\boldsymbol{g}_{a,1}, \boldsymbol{g}_{a,2})$, $(\boldsymbol{g}_{z,1}, \boldsymbol{g}_{z,2})$, *of the two 2-planes in (a)* 3 *dimensions, (b)* 4 *dimensions (the origin is pulled apart to disentangle the picture).*

The columns of the frames G_a and G_z are called "principal directions" of the pair of planes. Without loss of generality, we can assume that the diagonal elements of $G_a^T G_z$ are 1) non-negative and 2) sorted in descending order: $1 \geq \lambda_j \geq \lambda_{j+1} \geq 0$. (For non-negativity, multiply a column of G_a with -1 if necessary.) The diagonal elements λ_j of $G_a^T G_z$, called "principal cosines", are the stationary values of the cosines of angles between directions in the two planes. In particular, the largest principal cosine describes the smallest angle that can be formed with directions in the two planes. The angles $\alpha_j = \cos^{-1} \lambda_j$ are called principal angles $(0 \leq \alpha_1 \leq \alpha_2 \leq \ldots \leq \alpha_d \leq \pi/2)$. Note that for two planes with a nontrivial intersection of dimension $d_I > 0$, there will be d_I vanishing principal angles: $\alpha_1 = \ldots = \alpha_{d_I} = 0$. In particular, two 2-planes in 3-space always have at least an intersection of dimension $d_I = 1$, and hence $\lambda_1 = 1$ and $\alpha_1 = 0$. The geometry of principal directions is depicted in figure 12.

[The principal angles are the essential invariants for the relative position of two planes to each other. This means technically that two pairs of planes with equal principal angles can be mapped onto each other with a rotation (Halmos 1970).]

Principal directions are easily computed with a singular value decomposition (SVD). Given two arbitrary frames F_a and F_z spanning the respective planes, let

$$F_a^T F_z \;=\; V_a \Lambda V_z^T$$

be the SVD of $F_a^T F_z$, where V_a and V_z are $d \times d$ orthogonal matrices and Λ is diagonal with singular values λ_j in the diagonal. The frames

$$G_a \;=\; F_a V_a \quad \text{and} \quad G_z \;=\; F_z V_z$$

satisfy $G_a^T G_z = \Lambda$. Hence the singular values are just the principal cosines, and the orthogonal transformations V_a and V_z provide the necessary rotations of the initial frames. (See Bjorck and Golub (1973), and Golub and Van Loan (1983) section 12.4.).

The following construction selects an interpolating path with zero whip spin that is not only locally but globally shortest. The path is unique iff there are no principal angles of 90 degrees.

Path Construction:

1. Given a starting frame F_a and a preliminary frame F_z spanning the target plane, compute the SVD
$$F_a^T F_z \;=\; V_a \Lambda V_z^T \;, \quad \Lambda = \mathrm{diag}(\lambda_1 \geq \ldots \geq \lambda_d) \;,$$
and the frames of principal directions:
$$G_a \;=\; F_a V_a \;, \quad G_z \;=\; F_z V_z \;.$$

2. Form a preprojection frame B by, roughly speaking, orthonormalizing (G_a, G_z) with Gram-Schmidt. Due to $G_a^T G_z = \Lambda$, it is sufficient to orthonormalize $\boldsymbol{g}_{z,j}$ with regard to $\boldsymbol{g}_{a,j}$, yielding $\boldsymbol{g}_{*,j}$. This can be done only for $\lambda_j < 1$ because for $\lambda_j = 1$ we have $\boldsymbol{g}_{a,j} = \boldsymbol{g}_{z,j}$, spanning the d_I-dimensional intersection $\mathrm{span}(F_a) \cap \mathrm{span}(F_z)$.
(Numerically, use the criterion $\lambda_j > 1 - \epsilon$ for some small ϵ to decide inclusion in the intersection.)
Form the preprojection basis
$$B \;=\; (\boldsymbol{g}_{a,d}, \boldsymbol{g}_{*,d},\; \boldsymbol{g}_{a,d-1}, \boldsymbol{g}_{*,d-1},\; \ldots, \boldsymbol{g}_{a,d_I+1}, \boldsymbol{g}_{*,d_I+1},\; \boldsymbol{g}_{a,d_I}, \boldsymbol{g}_{a,d_I-1}, \ldots, \boldsymbol{g}_{a,1}) \;.$$
The last d_I vectors together span the intersection of the starting and target plane. The first $d - d_I$ pairs of vectors span each a plane in which we perform a rotation:

3. The sequence of planar rotations $R(\boldsymbol{\tau})$ is composed of the following $d - d_I$ planar rotations:
$$R_{12}(\tau_1) R_{34}(\tau_2) \ldots \;, \quad \text{where} \quad \tau_j = \cos^{-1} \lambda_{d+1-j} \quad \text{for} \quad j = 1, 2, \ldots, d - d_I.$$

The resulting path moves G_a to G_z, and hence $F_a = G_a V_a^T$ to $G_z V_a^T$. The original frame F_z in the target plane is thus replaced with the target frame $G_z V_a^T = F_z V_z V_a^T$.

For these paths, it is possible to give an explicit formula for speed measures, of which there exists essentially only one: In the preprojection basis, the moving frame is of the form
$$W(t) \;=\; R(\boldsymbol{\tau} t) W_a \;=\; \begin{pmatrix} c_{\tau_1 t} & 0 & \ldots \\ s_{\tau_1 t} & 0 & \ldots \\ 0 & c_{\tau_2 t} & \ldots \\ 0 & s_{\tau_2 t} & \ldots \\ \ldots & \ldots & \ldots \end{pmatrix} \;,$$

hence $g_W(W') = \|W'\|^2 = \tau_1^2 + \tau_2^2 + \ldots$, which is constant along the path. It therefore needs to be computed only once at the beginning of the path. — From this representation one immediately reads off the geodesic property: The plane spanned by coordinates 1, 3,... is rotated orthogonally out of itself through planar rotation in the invariant planes spanned by the coordinate pairs (1,2), (3,4), ... (see section 10).

In the most important case of $d = 2$-dimensional projections, the SVD problem is of size 2×2, which can be solved explicitly: The eigenvalues of the 2×2 symmetric matrix $(F_a^T F_z)(F_a^T F_z)^T$ are the squares of the singular values of $F_a^T F_z$ and can be found by solving a quadratic equation. We give the results without proof:

Lemma 1:
The principal directions in a 2-D starting plane are given by

$$\boldsymbol{g}_{a,1} = \cos\theta \cdot \boldsymbol{f}_{a,1} + \sin\theta \cdot \boldsymbol{f}_{a,2} \ , \quad \boldsymbol{g}_{a,2} = -\sin\theta \cdot \boldsymbol{f}_{a,1} + \cos\theta \cdot \boldsymbol{f}_{a,2} \ ,$$

where

$$\tan(2\theta) = 2\frac{(\boldsymbol{f}_{a,1}^T\boldsymbol{f}_{z,1}) \cdot (\boldsymbol{f}_{a,2}^T\boldsymbol{f}_{z,1}) + (\boldsymbol{f}_{a,1}^T\boldsymbol{f}_{z,2}) \cdot (\boldsymbol{f}_{a,2}^T\boldsymbol{f}_{z,2})}{(\boldsymbol{f}_{a,1}^T\boldsymbol{f}_{z,1})^2 + (\boldsymbol{f}_{a,1}^T\boldsymbol{f}_{z,2})^2 - (\boldsymbol{f}_{a,2}^T\boldsymbol{f}_{z,1})^2 - (\boldsymbol{f}_{a,2}^T\boldsymbol{f}_{z,2})^2} \ . \qquad (*)$$

The principal cosine λ_j is obtained by projecting $\boldsymbol{g}_{a,j}$ onto the target plane: $\lambda_j^2 = (\boldsymbol{g}_{a,j}^T\boldsymbol{f}_{z,1})^2 + (\boldsymbol{g}_{a,j}^T\boldsymbol{f}_{z,2})^2$.

Caution is needed when using equation $(*)$: It has four solutions in the interval $0 \le \theta < 2\pi$ spaced by $\pi/2$. Two solutions spaced by π yield the same principal direction up to a sign change. Hence the four solutions correspond essentially to two principal directions.

The denominator of the right hand side of equation $(*)$ is zero when the principal angles of the two planes are identical, in which case all unit vectors in the two planes are principal. One can hence pick F_a as a principal 2-frame in the starting plane.

The principal 2-frame in the target plane should always be computed by projecting the principal 2-frame of the starting plane. If both principal angles are $\pi/2$, any 2-frame in the target plane can be used for interpolation. If only one of the principal angles is $\pi/2$, one obtains $\boldsymbol{g}_{z,2}$ as an orthogonal complement of $\boldsymbol{g}_{z,1}$ in the target plane.

13 Interpolating Paths of Frames

Frame interpolation — as opposed to plane interpolation — is necessary when the orientation of the projection matters, as in full-dimensional tours. In addition, frame interpolation can always be used for plane interpolation when the human cost of implementing paths with zero whip spin is too high. Some frame interpolation schemes are indeed quite simple to implement. They have noticeable whip spin — a fact which XGobi users can confirm by playing with interpolation options in the tour module.

The methods described here are based on 1) decompositions of orthogonal matrices, 2) Givens decompositions, and 3) Householder decompositions. The second and third of these methods do not have any optimality properties, but the first method is optimal for full-dimensional tours in the sense that it yields geodesic paths in $SO(p)$; also, it can be improved so as to yield optimal paths of frames even for lower-dimensional projections, at least for the second of the speed measures of section 9.4.

13.1 Orthogonal matrix paths and optimal paths for full-dimensional tours

The idea of this interpolation technique is to augment the starting frame and the target frame to square orthogonal matrices and solve the interpolation problem in the orthogonal group ($SO(d_S)$, to be precise). The implementation is quite straightforward. The suboptimality of the paths is obvious from the arbitrariness of the way the frames are augmented to square orthogonal matrices. This is why full-dimensional paths are optimal: They do not require any augmentation at all. Strictly speaking, the requirement for optimality is not "full dimensionality" in the sense $d = p$, but simply that the starting plane and the target plane are the same, that is, $d_S = d$, in which case a simple rotation of the resting space takes place. For lower-dimensional frames whose space is not at rest ($d < d_S$), this arbitrariness can be remedied at least for the metric defined by $\alpha_w = 1$ and $\alpha_p = 2$: The trick is to optimize the augmentation (Asimov and Buja 1994). — The method is based on the following:

Fact: *For any orthogonal transformation A with* $\det(A) = +1$*, there exists an orthogonal matrix V such that*

$$A \;=\; VR(\boldsymbol{\tau})V^T \quad \textit{where} \quad R(\boldsymbol{\tau}) \;=\; R_{12}(\tau_1)\; R_{34}(\tau_2)\; \ldots\;.$$

That is, in a suitable coordinate system, every orthogonal transformation with det = +1 is the composition of planar rotations in mutually orthogonal 2-D planes:

$$A(\boldsymbol{v}_1, \boldsymbol{v}_2) = (\boldsymbol{v}_1, \boldsymbol{v}_2) \begin{pmatrix} c_\phi & -s_\phi \\ s_\phi & c_\phi \end{pmatrix} \;,$$

where $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ form an orthonormal basis of an invariant plane. See, for example, Halmos (1958, section 81). This is just a formal statement of the second fact listed in section 10.

Invariant planes can be found with a complex eigendecomposition of A (as implemented, for example, in subroutine "dgeev" in LAPACK, available with ftp from *ftp://netlib.att.com/netlib/lapack*). If $\boldsymbol{v} = \boldsymbol{v}_r + i\boldsymbol{v}_i$ is a complex eigenvector of A with eigenvalue $e^{i\phi}$, then the complex conjugate $\bar{\boldsymbol{v}} = \boldsymbol{v}_r - i\boldsymbol{v}_i$ is an eigenvector with eigenvalue $e^{-i\phi}$, hence

$$A(\boldsymbol{v}_r, \boldsymbol{v}_i) \;=\; (\boldsymbol{v}_r, \boldsymbol{v}_i) \begin{pmatrix} c_\phi & s_\phi \\ -s_\phi & c_\phi \end{pmatrix} \;,$$

which implies that $-\phi$ is the rotation angle in the invariant plane spanned by the frame $(\boldsymbol{v}_r, \boldsymbol{v}_i)$. (The vectors $\boldsymbol{v}_r$ and $\boldsymbol{v}_i$ need to be normalized as they are not usually of unit length when returned by a routine such as "dgeev".)

Path Construction:

1. Form a preliminary basis frame $\tilde{B}$ for preprojection by orthonormalizing the combined frame (F_a, F_z) with Gram-Schmidt. The preprojection of F_a is

$$\tilde{W}_a \;=\; \tilde{B}^T F_a \;=\; E_d = ((1,0,0,\ldots)^T, (0,1,0,\ldots)^T, \ldots) \;.$$

   The target frame $\tilde{W}_z = \tilde{B}^T F_z$ is of a general form.

2. Expand $\tilde{W}_z$ to a full orthogonal matrix A with $\det(A) = +1$, for example, by appending random vectors to $\tilde{W}_z$ and orthonormalizing them with Gram-Schmidt. Flip the last vector to its negative if necessary to ensure $\det(A) = +1$. Note that

$$\tilde{W}_z \;=\; A\tilde{W}_a \;,$$

   because $\tilde{W}_a$ extracts the first d columns from A, which is just $\tilde{W}_z$.

3. Find the canonical decomposition $A = VR(\boldsymbol{\tau})V^T$ according to the above.

4. Change the preprojection basis: $B = \tilde{B}V$, such that $W_a = B^T F_a = V^T\tilde{W}_a$ and $W_z = B^T F_z = V^T\tilde{W}_z$. From the canonical decomposition follows

$$W_z \;=\; R(\boldsymbol{\tau})W_a \;.$$

The above is the only method described in this paper that has not been implemented and tested in XGobi.

13.2 Givens paths

This interpolation method adapts the standard matrix decomposition techniques based on Givens rotations. It relies on the following:

Fact: *In any vector $\boldsymbol{u}$ one can zero out the i'th coordinate with a Givens rotation in the (i,j)-plane for any $j \neq i$. This rotation affects only coordinates i and j and leaves all coordinates $k \neq i, j$ unchanged.*

For example, to zero out coordinate x_2 in the (x_1, x_2)-plane, use a rotation $\begin{pmatrix} c_\tau & -s_\tau \\ s_\tau & c_\tau \end{pmatrix}$ with $c_\tau = x_1/(x_1^2 + x_2^2)^{1/2}$ and $s_\tau = -x_2/(x_1^2 + x_2^2)^{1/2}$. That is, τ is the angle from (x_1, x_2) to $(1, 0)$. See figure 13 for a depiction. For computational details see Golub and Van Loan (1983), section 3.4.

Sequences of Givens rotations can be used to map any orthonormal d-frame F in p-space to the standard d-frame $E_d = ((1,0,0,\ldots)^T, (0,1,0,\ldots)^T, \ldots)$ as follows:

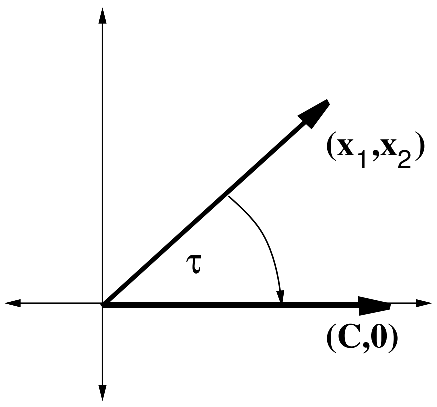

Figure 13: *Rotation to make subvector of matrix coincide with one of the coordinate axes.*

- Apply a sequence of $p-1$ Givens rotations to zero out coordinates $2, \ldots, p$ of the first vector $\boldsymbol{f}_1$. Examples of suitable sequences are rotations in the variables $(1,2),\ (1,3),\ (1,4), \ldots,\ (1,p)$, or $(p-1,p), \ldots,\ (3,4),\ (2,3),\ (1,2)$, where the second variable is the one whose coordinate is being zeroed out. Care should be taken that the last rotation chooses among the suitable angles τ and $\tau+\pi$ the one that results in the first coordinate =+1. The resulting frame will have $\boldsymbol{e}_1 = (1,0,0,\ldots)^T$ as its first vector.
- Apply to the resulting frame a sequence of $p-2$ Givens rotations to zero out the coordinates $3, \ldots, p$. Do not use coordinate 1 in the process, to ensure that the first vector $\boldsymbol{e}_1$ remains unchanged. A suitable sequence of planes could be $(2,3),\ (2,4), \ldots,\ (2,p)$, or else $(p-1,p), \ldots,\ (3,4),\ (2,3)$. Note that the zeros of the first column remain unaffected because $(0,0)$ is a fixed point under all rotations.
- And so on, till a sequence of $(p-1)+(p-2)+\ldots+(p-d) = pd - \begin{pmatrix} d \\ 2 \end{pmatrix}$ Givens rotations is built up whose composition maps F to E_d.

Path Construction:

1. Construct a preprojection basis B by orthonormalizing F_z with regard to F_a with Gram-Schmidt:
$$B \;=\; (F_a, F_*)\ .$$
2. For the preprojected frames
$$W_a \;=\; B^T F_a \;=\; E_d \quad \text{and} \quad W_z \;=\; B^T F_z$$

construct a sequence of Givens rotations that map W_z to W_a:

$$W_a \;=\; R_m(\tau_m)\ldots R_1(\tau_1)\; W_z\ .$$

Then the inverse mapping is obtained by reversing the sequence of rotations with the negative of the angles:

$$R(\boldsymbol{\tau}) \;=\; R_1(-\tau_1)\ldots R_m(-\tau_m)\ , \qquad W_z = R(\boldsymbol{\tau})W_a\ .$$

We made use of Givens rotations to interpolate projection frames. By comparison, the original grand tour implementation proposed in Asimov (1985), called "torus method," makes use of Givens decompositions in a somewhat different way: Asimov parametrizes the Stiefel manifold $V_{2,p}$ of 2-frames with angles $\boldsymbol{\tau} = (\tau_1, \tau_2, ...)$ provided by Givens rotations, and he devises infinite and uniformly distributed paths on the space (the "torus") of angles $\boldsymbol{\tau}$. The mapping of these paths to $V_{2,p}$ results in dense paths of frames, which therefore satisfy the definition of a grand tour. The non-uniformity problem mentioned at the end of the introduction stems from this mapping: The path of frames is not uniformly distributed, although its pre-image in the space of angles is. The non-uniformity may cause the tour to spend more time in some parts of the Stiefel manifold than in others. It would be of interest to better understand the mapping from the torus of angles to the manifold of frames. In particular, it would be interesting to know which of the many ways of constructing Givens decompositions lead to mappings with the best uniformity properties.

13.3 Householder paths

This interpolation method is based on reflections on hyperplanes, also called "Householder transformations" (see, for example, Golub and Van Loan (1983) section 3.3). A reflection at the hyperplane with normal unit vector $\boldsymbol{r}$ is given by

$$H \;=\; I - 2\cdot \boldsymbol{r}\boldsymbol{r}^T\ .$$

The usefulness of reflections stems from the following:

Facts:

1. *Any two distinct vectors of equal length, $\|\boldsymbol{w}\| = \|\boldsymbol{w}_*\|$, $\boldsymbol{w} \neq \boldsymbol{w}_*$, can be mapped onto each other by a uniquely determined reflection H with*

$$\boldsymbol{r} = (\boldsymbol{w} - \boldsymbol{w}_*)/\|\boldsymbol{w} - \boldsymbol{w}_*\|\ .$$

2. *Any vector orthogonal to $\boldsymbol{r}$ is fixed under the reflection.*

3. *The composition of two reflections H_1 and H_2 with vectors $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$, respectively, is a planar rotation in the plane spanned by $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$, with an angle double the angle between $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$.*

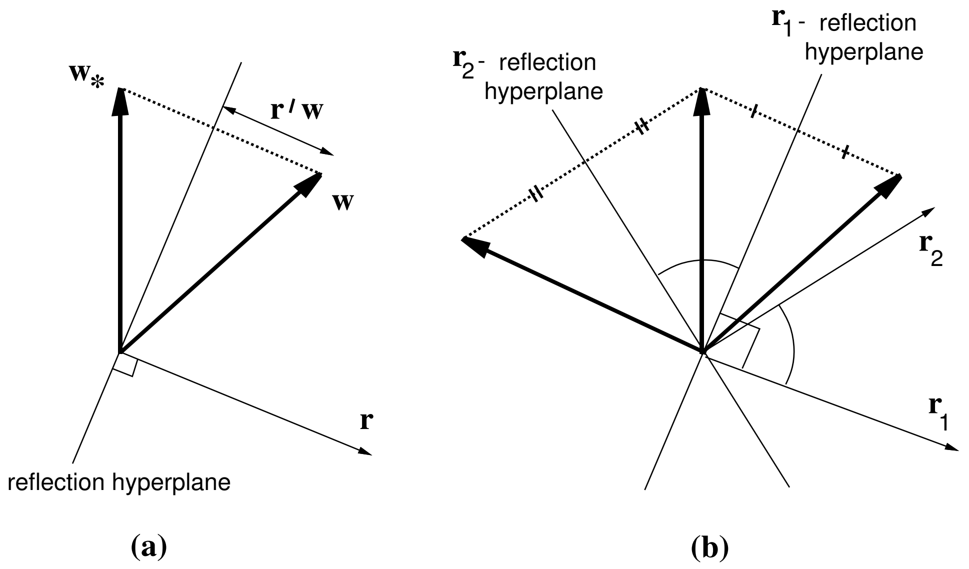

Figure 14: *(a) Reflection of a vector; (b) composition of two reflections to yield a planar rotation.*

See figure 14 for a depiction. We illustrate the technique for $d = 2$. We use a first reflection H_1 to map $\boldsymbol{f}_{a,1}$ to $\boldsymbol{f}_{z,1}$. Subsequently, we use a second reflection H_2 to map $H_1\boldsymbol{f}_{a,2}$ to $\boldsymbol{f}_{z,2}$. Under H_2, the vector $H_1\boldsymbol{f}_{a,1}$ is left fixed because both $H_1\boldsymbol{f}_{a,2}$ and $\boldsymbol{f}_{z,2}$, and hence their difference, are orthogonal to $H_1\boldsymbol{f}_{a,1} = \boldsymbol{f}_{z,1}$. Thus

$$F_z \;=\; H_2 H_1 F_a \;,$$

where H_2H_1 is a planar rotation according to the third fact above. The computational details are somewhat more involved than in numerical analysis applications because we must make sure that we end up with exactly two reflections that amount to a planar rotation:

Path Construction (for $d = 2$):

1. Find a first reflection vector $\boldsymbol{r}_1$ such that $H_1\boldsymbol{f}_{a,1} = \boldsymbol{f}_{z,1}$: If $\boldsymbol{f}_{a,1} \neq \boldsymbol{f}_{z,1}$, use $\boldsymbol{r}_1 = (\boldsymbol{f}_{a,1} - \boldsymbol{f}_{z,1})/\|\boldsymbol{f}_{a,1} - \boldsymbol{f}_{z,1}\|$, and $\boldsymbol{r}_1 \perp \boldsymbol{f}_{a,1}$ otherwise.

2. Map $\boldsymbol{f}_{a,2}$ with this reflection: $\boldsymbol{f}_{a,2+} = H_1\boldsymbol{f}_{a,2} = \boldsymbol{f}_{a,2} - 2\cdot(\boldsymbol{f}_{a,2}^T\boldsymbol{r}_1)\boldsymbol{r}_1$.

3. Find a second reflection vector $\boldsymbol{r}_2$ such that $H_2\boldsymbol{f}_{a,2+} = \boldsymbol{f}_{z,2}$: If $\boldsymbol{f}_{a,2+} \neq \boldsymbol{f}_{z,2}$, use $\boldsymbol{r}_2 = (\boldsymbol{f}_{a,2+} - \boldsymbol{f}_{z,2})/\|\boldsymbol{f}_{a,2+} - \boldsymbol{f}_{z,2}\|$, and $\boldsymbol{r}_2 \perp \boldsymbol{f}_{a,2+},\ \perp \boldsymbol{f}_{z,2}$ otherwise.

4. Form a preprojection basis B: Orthonormalize $\boldsymbol{r}_2$ with regard to $\boldsymbol{r}_1$ to yield $\boldsymbol{r}_*$; expand $(\boldsymbol{r}_1, \boldsymbol{r}_*)$ to an orthonormal basis of span(F_a, F_z).

5. Rotation: $R_{12}(\tau)$ with $\tau = 2\cos^{-1}(\boldsymbol{r}_1^T\boldsymbol{r}_2)$.

The generalization of Householder paths to d-frames for $d > 2$ is quite obvious for d even: The process generates d reflections that can be bundled up into $d/2$ planar rotations. For d odd, some precautions are in order: If $\text{span}(F_a) \neq \text{span}(F_z)$, one has to introduce one additional dummy reflection that leaves F_z fixed, using an $\boldsymbol{r}_{d+1}$ in $\text{span}(F_a, F_z)$ orthogonal to F_z; if $\text{span}(F_a) = \text{span}(F_z)$, the last reflection was not necessary because $H_{d-1}...H_1\boldsymbol{f}_{a,d} = \pm\boldsymbol{f}_{z,d}$ automatically, hence $(d-1)/2$ rotations result. If the spans are identical, the frames have to have the same orientation in order to allow continuous interpolation; hence it may be necessary to change the sign of $\boldsymbol{f}_{z,d}$.

A peculiar aspect of the Householder method is that it uses fewer planar rotations than any of the other methods; as we have seen it transports 2-frames onto each other with a single planar rotation. Yet it does not produce paths that are optimal in any sense that we know of. It would be of interest to better understand the geometry of the Householder method and possibly produce criteria for which it is optimal. For example, we do not even know whether Householder paths are geodesic for one of the metrics introduced in section 10.

14 Conclusions

Dynamic projections form a powerful set of tools for viewing high-dimensional data and mathematically defined high-dimensional objects. Dynamic projections exploit the human eye's natural ability to detect and recognize objects in motion. In addition, dynamic projections exploit the human instinct for playfulness that is most apparent in a child. For graphical data analysis, we can turn a humble static 2-D projection into a rocking image that conveys up to four simultaneous dimensions at any given time.

The goal of this paper was to give a mathematical as well as an algorithmic framework for dynamic projections: We discussed graphical rendering of projections of data space onto planes of dimensions $1, 2, 3$ and > 3. We analyzed the role of orientation of projections. We proposed notions that allow us to decompose motion along paths of projections into two natural components: within-plane spin and plane motion. Pure plane motion is arguably optimal for most types of low-dimensional rendering, but for full-dimensional motion there is no plane motion and all that matters are dynamic changes in orientation. We discussed ways of measuring speed of moving projections and found a wealth of mathematical structure that lead to notions of geodesic motion of projections. We gave a framework for the implementation of paths of projections based on the interpolation of pairs of projections. The notion of steering from target projection to target projection makes for a flexible environment in which grand tours, interactive projection pursuit and manual projection control, can be nicely embedded. Finally, we proposed several numerical techniques for implementing interpolating paths of projections.

15 Appendix: Invariant Metrics for Frames and Planes

The goal of this appendix is to describe invariant Riemannian metrics on the Stiefel manifold $V_{d,p}$ of orthonormal d-frames in p-space. The purpose of Riemannian metrics is to measure speed of moving frames. There are two types of invariance: Left-invariance requires that for fixed $U \in O(p)$, the paths $F(t)$ and $UF(t)$ show the same speed at each time point t. Right-invariance requires that for fixed $V \in O(d)$, the paths $F(t)$ and $F(t)V$ show the same speed at each time point t.

Left-invariance implies a structure theorem. Part of the theorem says that the two components of frame motion — pure whip spin and pure plane motion — are always orthogonal, implying a Pythagorean relation among the two.

If we assume right invariance in addition to left-invariance, then whip speed and plane speed are each essentially measured by Frobenius norms. Nothing is implied, however, about the relative weighting of the two types of speed when merging them into an overall speed measure.

Finally, we characterize Riemannian metrics on the Grassmann manifold $G_{d,p}$ of d-planes in p-space, but we think of these metrics as semi-metrics on $V_{d,p}$ that are invariant under dynamic within-plane spin. More precisely, "Grassmann semi-metrics" are required to assign the same speed properties to a path of frames $F(t)$ and any other path $F(t)V(t)$ that differs from the first only by a dynamic within-plane rotation $V(t)$. Such Grassmann semi-metrics are essentially unique.

Below, we introduce the concepts of tangent space and Riemannian metric of differential geometry, but we immediately apply them to the Stiefel manifold. The presentation is self-contained, and no knowledge of differential geometry is assumed. The material belongs really in the framework of reductive homogeneous spaces (Kobayashi and Nomizu 1969, chap. X), but we found it possible to give an elementary introduction.

A word to the mathematician: Because we adopt invariance with regard to $O(p)$ rather than $SO(p)$, we rule out the curious appearance of non-equivalent invariant metrics on the Grassmann manifold $G_{2,4}$. It seems to us that these metrics are irrelevant for visualization with dynamic projections. Similarly "unnatural" metrics exist on $S_{d,p}$ when $p - d = 2$, and they are equally ruled out.

15.1 Metrics for frames

Definition: *The* **tangent space** T_F *at the frame* $F \in V_{d,p}$ *is the space of all* $p \times d$ *matrices* X *that are derivatives of differentiable paths of frames through* F*:*

$$T_F \;=\; \{X|\; \textit{there exists a path } F(t) \textit{ such that } \; F(0) = F,\; F'(0) = X\;\}$$

Fact: $T_F \;=\; \{X|\; (F^T X)^T = -(F^T X)\;\}$

Proof: The inclusion $\subset$ is easily shown by taking derivatives of the orthonormality condition $F(t)^T F(t) = I_d$, which for each t represents the set of equations that define

$V_{d,p}$. It remains to show that every such X is a derivative matrix. By a change of coordinates, we can achieve that F consists of the first d columns of the identity matrix I_p. The matrix $F^T X$ is then the top $d \times d$ block of X, hence the condition $(F^T X)^T = -(F^T X)$ says that this block is skew. Therefore X can be extended to a skew matrix S of size $p \times p$. (This extension is not unique.) The matrix exponential $U(t) = \exp(St)$ is a path of orthogonal matrices that satisfy $U(0) = I_p$ and $U'(0) = S$. Therefore, the first d columns form a path of frames $F(t)$ with $F(0) = F$ and $F'(0) = X$. $\square$

The matrix $F^T X$ is what we called the whip spin matrix. The above condition says that every $p \times d$ matrix with skew whip spin matrix with regard to F is an element of T_F. Skew symmetry of a $d \times d$ matrix implies $(d+1)d/2$ independent equality constraints, hence:

Fact: $\dim(V_{d,p}) \;=\; \dim(T_F) \;=\; pd - (d+1)d/2$

Definition: *A* **Riemannian metric** *is a family of symmetric bilinear forms $g_F(.,.)$ defined on T_F (and differentiable in F in a suitable sense).*

Riemannian metrics can be thought of as "infinitesimal inner products." The usual definitions of Riemannian metrics include positive definiteness, but we will also consider degenerate bilinear forms, that is, semi-metrics.

Symmetric bilinear forms are uniquely determined by their quadratic forms which, by abuse of notation, we also denote by g_F:

$$g_F(X) = g_F(X,X) \;, \qquad g_F(X,Y) \;=\; (g_F(X+Y) - g_F(X-Y))/4 \;.$$

We therefore use the term "Riemannian metric" both for $g_F(.,.)$ and $g_F(.)$.

15.2 Left-invariant metrics for frames

We turn to left-invariance of Riemannian metrics with regard to $O(p)$, the group of orthogonal $p \times p$ matrices. We use invariance with regard to $O(p)$ as opposed to $SO(p)$, the orthogonal transformations with determinant $+1$. The difference is that $O(p)$ includes reflections besides proper rotations. Although $SO(p)$-invariance is standard in differential geometry, we decided in favor of $O(p)$-invariance for two reasons: 1) We see no reason why a path of frames and a reflected version thereof should have different speed properties. 2) Some unintuitive technical artifacts can be avoided, such as strange invariant metrics on the Grassmannian $G_{2,4}$, peculiar to these dimensions and unrelated to any visualization properties of dynamic projections.

For any $U \in O(p)$, if $F(t)$ is a path of frames, so is $UF(t)$, hence:

Fact: $U \in O(p) \;,\;\; X \in T_F \quad \Longrightarrow \quad UX \in T_{UF} \;.$

Definition: *A Riemannian metric is left-invariant w.r.t. $O(p)$ if it satisfies*

$$g_{UF}(UX, UY) \;=\; g_F(X, Y) \quad \textit{or, equivalently,} \quad g_{UF}(UX) \;=\; g_F(X) \;,$$

for all $U \in O(p)$.

Theorem: *A left-invariant Riemannian metric satisfies*

$$g_F(UX) \;=\; g_F(X) \quad \textit{for all } U \quad \textit{such that} \quad UF = F \;.$$

Conversely, any positive definite quadratic form $g(X)$ defined on a single tangent space T_F satisfying

$$(*) \qquad g(UX) \;=\; g(X) \quad \textit{for all } U \quad \textit{such that} \quad UF = F$$

can be extended to a unique left-invariant Riemannian metric through

$$g_{UF}(UX) \overset{def}{=} g(X) \quad \textit{for all} \quad U \in O(p) \;.$$

Proof: The first part is a trivial specialization of the definition of left-invariance.

For the second part, assume we have two representations of a frame $U_1F = U_2F$ and two representations of a tangent vector $U_1X_1 = U_2X_2 \in T_{U_1F}$ where $X_1, X_2 \in T_F$. We have to show $g_{U_1F}(U_1X_1) = g_{U_2F}(U_2X_2)$ in order to make the above definition consistent.

The defining equations are $g_{U_1F}(U_1X_1) = g(X_1)$ and $g_{U_2F}(U_2X_2) = g(X_2)$, hence we have to show $g(X_1) = g(X_2)$.

From $U_1X_1 = U_2X_2$ we get $X_2 = U_2^TU_1X_1$, and from $U_1F = U_2F$ we get $U_2^TU_1F = F$. Hence, we can apply the assumption $(*)$ with $U = U_2^TU_1$ and get $g(X_2) = g(UX_1) = g(X_1)$. □

Reduction of the problem:

Given the above theorem, we can analyze left-invariant Riemannian metrics by choosing a convenient frame F and by analyzing the structure of symmetric bilinear and quadratic forms g on T_F satisfying property $(*)$.

The most convenient frame is $E_d = (\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d)$, that is, the frame consisting of the first d columns of the identity matrix I_p. From now on we write $g = g_{E_d}$, and we assume that g satisfies condition $(*)$.

We need to characterize the tangent space T_{E_d} and specialize property $(*)$. To this end, for a $p \times d$ matrix X and for $U \in O(p)$ we write

$$X \;=\; \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}, \qquad U \;=\; \begin{pmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{pmatrix},$$

where X_1 is $d \times d$, X_2 is $(p-d) \times d$, U_{11} is $d \times d$, U_{12} is $d \times (p-d)$, and so on.

Fact: $T_{E_d} \;=\; \{X|\; X_1 \textit{ is skew symmetric }\}$

Fact: $UE_d = E_d \quad \Longleftrightarrow \quad U \;=\; \begin{pmatrix} I_d & 0 \\ 0 & U_{22} \end{pmatrix},\;\; U_{22} \in O(p-d)$.

Proofs: $X_1 = E_d^T X$ is the whip spin matrix. The condition $UE_d = E_d$ implies $U_{11} = I_d$ and $U_{21} = 0$, and the rest follows from orthogonality of the columns of U. □

Lemma: $X \;=\; \begin{pmatrix} X_1 \\ 0 \end{pmatrix}$ *and* $Y \;=\; \begin{pmatrix} 0 \\ Y_2 \end{pmatrix} \quad \Longrightarrow \quad g(X,Y) = 0.$

That is, whip spin and plane motion are orthogonal under any left-invariant Riemannian metric. As a consequence:

Corollary: $g(X) \;=\; g(\begin{pmatrix} X_1 \\ 0 \end{pmatrix}) \;+\; g(\begin{pmatrix} 0 \\ X_2 \end{pmatrix})$.

Proof: We use condition $(*)$. Let R be such that $UE_d = E_d$, that is $U_{11} = I_d$, $U_{21} = 0$, $U_{12} = 0$, $U_{22} \in O(p-d)$. From this and $X_2 = 0$ follows $UX = X$. It also follows that $UY = \begin{pmatrix} 0 \\ U_{22}Y_2 \end{pmatrix}$ because $Y_1 = 0$. We now use the conclusion of $(*)$ in the form $g(X,Y) \;=\; g(UX,UY)$:

$$g(X,Y) \;=\; g(UX,UY) \;=\; g(X, \begin{pmatrix} 0 \\ U_{22}Y_2 \end{pmatrix}) \;, \quad \text{for all } U_{22} \in O(p-d) \;.$$

Consider matrices Y_2 in which all but one column is zero: It follows that $g(X,Y)$ is a $O(p-d)$-left-invariant *linear* form of these special matrices. Such left-invariant linear forms are identically zero. The d spaces of such special matrices span the space of all Y_2 matrices, hence $g(X,Y) = 0$, for all Y_2. □

Fact: $g(\begin{pmatrix} X_1 \\ 0 \end{pmatrix})$ *can be any quadratic form of* X_1.

Lemma: $X = \begin{pmatrix} 0 \\ X_2 \end{pmatrix}$ *and* $Y = \begin{pmatrix} 0 \\ Y_2 \end{pmatrix} \quad \Longrightarrow \quad g(X,Y) \;=\; \text{trace}(X_2^T Y_2 C)$ *for some symmetric* $d \times d$ *matrix* C*. The metric is positive definite on* X*'s of the form* $X = \begin{pmatrix} 0 \\ X_2 \end{pmatrix}$ *iff* C *is positive definite.*

Proof: For X and Y as above, any bilinear form $g(X,Y)$ is of the form

$$g(X,Y) \;=\; \sum_{i,j=1\ldots d} \boldsymbol{x}_i^T A_{ij} \boldsymbol{y}_j \;,$$

where $X_2 = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d)$, $Y_2 = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_d)$ and A_{ij} are matrices of size $(p-d)\times(p-d)$.

We consider special matrices X_2 and Y_2 that have only one non-zero column each, which we denote $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$, respectively. We have

$$g(X,Y) \;=\; \boldsymbol{x}_i^T A_{ij} \boldsymbol{y}_j \;.$$

With property $(*)$, we also have for $UE_d = E_d$:

$$g(X,Y) \;=\; g(UX, UY) \;=\; \boldsymbol{x}_i^T U_{22}^T A_{ij} U_{22} \boldsymbol{y}_j \;,$$

from which follows

$$A_{ij} \;=\; U_{22}^T A_{ij} U_{22} \;, \quad \text{for all } U_{22} \in O(p-d) \;.$$

By a) of the corollary in section 15.7, this implies that A_{ij} is multiple of the identity:

$$A_{ij} \;=\; c_{ij} \cdot I_{p-d} \;.$$

It follows that for general X_2 and Y_2 we have

$$g(X,Y) \;=\; \sum_{i,j=1\ldots d} \boldsymbol{x}_i^T A_{ij} \boldsymbol{y}_j \;=\; \sum_{i,j=1\ldots d} \boldsymbol{x}_i^T \boldsymbol{y}_j c_{ij} \;=\; \mathrm{trace}(X_2^T Y_2 C) \;,$$

where $C = (c_{ij})$. Symmetry of C follows from symmetry of $g(.,.)$.

For positive definiteness, let $C \;=\; \sum_i \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^T$ be an eigendecomposition of C. We get

$$g(X) \;=\; \mathrm{trace}(X_2^T X_2 C) \;=\; \sum_i \lambda_i \|X_2 \boldsymbol{u}_i\|^2 \;,$$

where $\|..\|$ is the Euclidean norm on $I\!R^{p-d}$. Thus $g(X)$ is positive definite iff all λ_i are positive, that is, if C is positive definite. □

The following theorem is a summary and reformulation of the above lemmas for an arbitrary frame F. Note that X_1 at $F = E_d$ corresponds to $F^T X$ at an arbitrary F. Thus whip speed can be measured by an arbitrary non-negative quadratic function $g_{whip}(F^T X)$ of the whip spin matrix $F^T X$. As for plane motion, the matrix $X_2^T X_2$ at $F = E_d$ becomes $((I-P)X)^T((I-P)X) = X^T(I-P)X$ at an arbitrary F. We finally translate $\mathrm{trace}(X^T(I-P)XC) = \|\,(I-P)XC^{1/2}\,\|_{Frob}^2$.

Theorem: *Left-invariant Riemannian metrics are of the form*

$$g_F(X) \;=\; g_{whip}(F^T X) \;+\; \|\,(I-P)XC^{1/2}\,\|_{Frob}^2 \;,$$

where $X \in T_F$, g_{whip} is an arbitrary non-negative quadratic function of $d(d-1)/2$ arguments, and C is an arbitrary symmetric non-negative definite $d \times d$ matrix, and as usual $P = FF^T$. In particular, whip spin and plane motion are orthogonal under all left-invariant Riemannian metrics.

15.3 Left- and right-invariant metrics for frames

For any $V \in O(d)$, if $F(t)$ is a path of frames, so is $F(t)V$, hence:

Fact: $V \in O(d)\ ,\quad X \in T_F \quad \Longrightarrow \quad XV \in T_{FV}\ .$

Definition: *A Riemannian metric is right-invariant w.r.t. $O(d)$ if it satisfies*

$$g_{FV}(XV, YV) \;=\; g_F(X,Y) \quad \textit{or, equivalently,} \quad g_{FV}(XV) \;=\; g_F(X)\ ,$$

for all $V \in O(d)$.

Right invariance is not in itself a far reaching property. It is, however, when combined with left-invariance. Therefore we make henceforth the

Assumption: *The metric $g_F(X,Y)$ (equivalently: $g_F(X)$) is both left-invariant w.r.t. $O(p)$ and right-invariant w.r.t. $O(d)$.*

Fact: *At $F = E_d$, we have* $g(\begin{pmatrix} V^T X_1 V \\ X_2 V \end{pmatrix}) = g(\begin{pmatrix} X_1 \\ X_2 \end{pmatrix})$ *for all $V \in O(d)$.*

Proof: Let $U = \begin{pmatrix} V & 0 \\ 0 & I_{p-d} \end{pmatrix}$ and note $E_d V = U E_d$. Hence:

$$g_{E_d}(X) = g_{E_d V}(XV) = g_{UE_d}(XV) = g_{E_d}(U^T XV) = g(\begin{pmatrix} V^T X_1 V \\ X_2 V \end{pmatrix})\ .\ \square$$

The following theorem describes the consequences of left- and right-invariance for whip spin and plane motion separately:

Theorem: $g(\begin{pmatrix} X_1 \\ 0 \end{pmatrix}) \;=\; \alpha_w \cdot \mathrm{trace}(X_1^T X_1)\ ,\quad g(\begin{pmatrix} 0 \\ X_2 \end{pmatrix}) \;=\; \alpha_p \cdot \mathrm{trace}(X_2^T X_2)\ .$

Proof: *a) Whip spin:* The whip spin matrix X_1 is skew symmetric, hence there exists $V \in O(d)$ that brings X_1 to canonical form:

$$V^T X_1 V \;=\; \begin{pmatrix} 0 & -\sigma_1 & 0 & 0 & ... \\ \sigma_1 & 0 & 0 & 0 & ... \\ 0 & 0 & 0 & -\sigma_2 & ... \\ 0 & 0 & \sigma_2 & 0 & ... \\ ... & ... & ... & ... & ... \end{pmatrix}\ ,$$

where $\sigma_i \geq 0$ are the canonical speeds in the invariant 2-D planes (they are also the singular values of X_1). The 2×2 blocks are the infinitesimal rotations that generate whip spin. It is sufficient to consider X_1's in canonical form. Denote with S_{ij} the infinitesimal rotation with unit speed in the i-j-coordinate plane, that is, S_{ij} has action $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$ in this plane. We show that, for example, S_{12} and S_{34} are orthogonal under any left-right-invariant metric: $g(S_{12}, S_{34}) = 0$ (where, by abuse of notation, we ignored the part due to plane motion). To this end, let $V \in O(d)$ be a reflection that maps the 4'th coordinate to its negative, and note that $V^T S_{12} V = S_{12}$ and $V^T S_{34} V = -S_{34}$. It follows

$$g(S_{12}, S_{34}) \;=\; g(V^T S_{12} V, V^T S_{34} V) \;=\; g(S_{12}, -S_{34}) \;=\; -\,g(S_{12}, S_{34})\;,$$

hence $g(S_{12}, S_{34}) = 0$. Therefore, the metric is of the form $g(X_1) = \alpha_1\sigma_1^2 + \alpha_2\sigma_2^2 + \ldots$. The coefficients α_i must be identical because there exists V such that, for example, $V^T S_{12} V = S_{34}$, implying permutation invariance in the σ_i's. Thus,

$$g(X_1) \;=\; 2\alpha_w \cdot \sum_i \sigma_i^2 \;=\; \alpha_w \cdot \mathrm{trace}(X_1^T X_1)\;.\;\;\square$$

b) Plane motion: Left-invariance implies $g(X) = \mathrm{trace}(X_2^T X_2 C)$ for some symmetric C of size $(p-d) \times (p-d)$. We show that right-invariance implies $C = c \cdot I_d$: From the above fact and a simple manipulation of the trace we get

$$\mathrm{trace}(X_2^T X_2 C) \;=\; \mathrm{trace}(V^T X_2^T X_2 V C) \;=\; \mathrm{trace}(X_2^T X_2 V C V^T)\;.$$

Hence $VCV^T = C$ for all $V \in SO(d)$. By part c) of the corollary of section 15.7 it follows $C = c \cdot I_d$. $\square$

Here is a summary, written for arbitrary frames F, figuring in that whip spin and plane motion are orthogonal for left-invariant metrics:

Corollary: *For some $\alpha_w, \alpha_p > 0$, and for $P = FF^T$, we have*

$$\begin{aligned} g_F(F') \;&=\; \alpha_w \cdot \mathrm{trace}(F'^T P F') \;+\; \alpha_p \cdot \mathrm{trace}(F'^T (I-P) F') \\ &=\; \alpha_w \cdot \|F^T F'\|_{Frob}^2 \;+\; \alpha_p/2 \cdot \|P'\|_{Frob}^2\;. \end{aligned}$$

Proof: The first equality is a straight translation from the frame E_d to an arbitrary frame F. The second equality requires a minor calculation at $F = E_d$ and $F'^T = (X_1^T, X_2^T)$:

$$P' \;=\; F'F^T + FF'^T \;=\; \begin{pmatrix} X_1 & 0 \\ X_2 & 0 \end{pmatrix} \;+\; \begin{pmatrix} X_1^T & X_2^T \\ 0 & 0 \end{pmatrix} \;=\; \begin{pmatrix} 0 & X_2^T \\ X_2 & 0 \end{pmatrix}\;,$$

because X_1 is skew. Hence $\|P'\|_{Frob}^2 = 2 \cdot \|X_2\|_{Frob}^2$. $\square$

15.4 Invariant metrics for planes

From what we have proven so far, it is almost trivial that invariant metrics on planes must be the plane speed component of the corollary in the previous section. We should note in what follows that metrics on planes are semi-metrics on frames because they are blind to motion within the plane, by definition. Because frames are the necessary ingredient for visual rendering, though, we prefer to think of plane metrics as semi-metrics on frames, for which we reserve the term "Grassmannian":

Definition: *A left- and right-invariant semi-metric g on frames is called a Grassmann semi-metric if it is invariant under dynamic within-plane rotations. That is, for any fixed frame such as E_d, any path $V = V(t) \in SO(d)$ of whip rotations, and for all t,*

$$g_{E_d V(t)}(E_d V'(t)) \;=\; 0 \;.$$

Because V' can be any skew matrix of size $d \times d$, we must have $g_{whip} = 0$, hence:

Corollary: *There exists essentially only one invariant Grassmann semi-metric:*

$$g_F(F') \propto \|P'\|^2_{Frob} \;.$$

15.5 The equations of geodesics with regard to invariant metrics

We derive the equations for geodesic paths with regard to the invariant metrics of section 15.4. In principle, any textbook of differential geometry, such as Spivak (1979, p. 444 and 451), has the equations for a geodesic in coordinates x_i:

$$\sum_l g_{kl} x_l'' \;=\; \frac{1}{2} \sum_{ij} \left(\frac{\partial}{\partial x_k} g_{ij} - \frac{\partial}{\partial x_i} g_{jk} - \frac{\partial}{\partial x_j} g_{ik} \right) x_i' x_j'$$

This form, however, is extremely unhelpful in our situation: Orthonormal d-frames in p-space form a submanifold of $I\!R^{pd}$ defined in terms of implicit equations ($F^T F = I_d$), endowed with Riemannian metrics that are not inherited from the Euclidean metric of $I\!R^{pd}$ (except when $\alpha_p = \alpha_p$). We re-cast the equations in a form that facilitates application to frames. The new equations will be in vectorized coordinates in order to avoid the tedium of index calculations. To this end, we introduce notation for directional derivatives in coordinates:

$$\partial_Y f(\boldsymbol{x}) \;=\; \sum_i Y_i \frac{\partial}{\partial x_i} f(x_1, x_2, ...) \;,$$

where $\boldsymbol{x} = (x_1, x_2, ...)^T$ are point coordinates and $Y = (Y_1, Y_2, ...)$ are coordinates of a tangent vector at $\boldsymbol{x}$. For a curve $\boldsymbol{x}(t)$ the vector $\boldsymbol{x}'$ also contains tangent vector coordinates. The reason for introducing directional derivatives is their ease of application to vector-algebraic expressions. — For a Riemannian metric

$$g_{\boldsymbol{x}}(X,Z) \;=\; \sum_{ij} g_{ij}(\boldsymbol{x}) X_i Z_j$$

we can now make sense of $\partial_Y g_{\boldsymbol{x}}$:

$$(\partial_Y g_{\boldsymbol{x}})(X,Z) \;=\; \sum_{ij} (\partial_Y g_{ij}(\boldsymbol{x})) X_i Z_j \;.$$

With the aid of arbitrary tangent vectors Y, the equations for a geodesic can now be re-cast, first in indexed coordinates:

$$\sum_{k,l} g_{kl} x''_l Y_k \;=\; \frac{1}{2} \sum_{ijk} \left(\frac{\partial}{\partial x_k} g_{ij} - \frac{\partial}{\partial x_i} g_{jk} - \frac{\partial}{\partial x_j} g_{ik} \right) x'_i x'_j Y_k \;,$$

then in vectorized coordinates:

$$\begin{aligned} g(Y, \boldsymbol{x}'') \;&=\; \frac{1}{2} \left[(\partial_Y g)(\boldsymbol{x}', \boldsymbol{x}') - (\partial_{\boldsymbol{x}'} g)(Y, \boldsymbol{x}') - (\partial_{\boldsymbol{x}'} g)(\boldsymbol{x}', Y) \right] \\ &=\; \frac{1}{2} (\partial_Y g)(\boldsymbol{x}', \boldsymbol{x}') - (\partial_{\boldsymbol{x}'} g)(Y, \boldsymbol{x}') \;, \end{aligned}$$

where we omitted the argument $\boldsymbol{x}$ from $g = g_{\boldsymbol{x}}$. These equations are to hold for all Y denoting tangent vectors at $\boldsymbol{x}$.

From a purist's point of view of differential geometry, this form of the geodesic equations is peculiar: It looks like an invariant formulation but it isn't; for one thing, $\boldsymbol{x}''$ does not denote a tangent vector, yet $g(\boldsymbol{x}'', Y)$ is a well-defined algebraic expression. This form of the equations is just a device for executing messy coordinate calculations in vectorized notation. (In an invariant interpretation, the equations express the invariant geodesic condition $D_X X = 0$ in the equivalent form $g_{\boldsymbol{x}}(Y, D_X X) = 0$ for all tangent vectors Y, where X is the tangent vector denoted by $\boldsymbol{x}'$.)

We now derive the equations for geodesic paths of frames $F = F(t)$ with regard to any of the invariant metrics. For the tangent vectors F' and Y at F, the metric in its bilinear form is

$$g_F(Y, F') \;=\; \alpha_p \cdot \mathrm{trace}(Y^T F') + (\alpha_w - \alpha_p) \cdot \mathrm{trace}(Y^T F F^T F') \;.$$

We calculate $g(Y, F'')$, $(\partial_Y g)(F', F')$ and $(\partial_{F'} g)(Y, F')$ in turn:

$$\begin{aligned} g(Y, F'') \;&=\; \alpha_p \cdot \mathrm{trace}(Y^T F'') + (\alpha_w - \alpha_p) \cdot \mathrm{trace}(Y^T P F'') \\ (\partial_Y g)(F', F') \;&=\; (\alpha_w - \alpha_p) \cdot \left[\mathrm{trace}(F'^T Y F^T F') + \mathrm{trace}(F'^T F Y^T F') \right] \\ &=\; (\alpha_w - \alpha_p) \cdot 2 \cdot \mathrm{trace}(Y^T F' F'^T F) \\ &=\; -\,(\alpha_w - \alpha_p) \cdot 2 \cdot \mathrm{trace}(Y^T F' F^T F') \\ (\partial_{F'} g)(Y, F') \;&=\; (\alpha_w - \alpha_p) \cdot \left[\mathrm{trace}(Y^T F' F^T F') + \mathrm{trace}(Y^T F F'^T F') \right] \\ &=\; (\alpha_w - \alpha_p) \cdot \mathrm{trace}(Y^T [F' F^T F' + F F'^T F']) \end{aligned}$$

We used repeatedly the trace identities

$$\partial_Y \mathrm{trace}(AFF^TB) = \mathrm{trace}(AYF^TB) + \mathrm{trace}(AFY^TB) \ ,$$
$$\mathrm{trace}(A^T) = \mathrm{trace}(A) \ , \quad \mathrm{trace}(AB) = \mathrm{trace}(BA) \ .$$

For the second term we also made use of the fact that F^TF' is skew. The equations for geodesic paths are therefore:

$$\begin{aligned} 0 &= g(Y, F'') - 1/2 \cdot (\partial_Y g)(F', F') + (\partial_{F'} g)(Y, F') \\ &= \mathrm{trace}\left(Y^T \left[\alpha_p \cdot F'' + (\alpha_w - \alpha_p) \cdot \{PF'' + 2F'F^TF' + FF'^TF'\} \right] \right) \end{aligned}$$

This is to hold for all tangent vectors Y at F. In order to flesh out this condition we need a lemma:

Lemma: *For a $p \times d$-matrix Z the following conditions are equivalent:*
1) $\mathrm{trace}(Y^TZ) = 0$ *for all Y for which F^TY is skew.*
2) $(I - P)Z = 0$ and F^TZ is symmetric.

Proof of the lemma: We can assume w.l.o.g. $F = E_d$, in which case

$$Z = \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} \ , \quad Y = \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \ , \quad Y_1 = E_d^TY \ \text{ is skew.}$$

The expression $\mathrm{trace}(Y^TZ)$ is just the Euclidean inner product on $\mathbb{R}^{pd}$. It is zero for all Y with skew Y_1 iff Z_1 is symmetric and $Z_2 = 0$. At an arbitrary frame F, this translates to F^TZ symmetric and $(I - P)Z = 0$. □

We apply the lemma to the equation for geodesic paths of frames: For

$$Z \ = \ \alpha_p \cdot F'' + (\alpha_w - \alpha_p) \cdot \{PF'' + 2F'F^TF' + FF'^TF'\}$$

we must have F^TZ symmetric and $(I - P)Z = 0$:

- The symmetry condition simplifies as follows: 1) $F^TF'F^TF'$ is the square of the skew matrix F^TF', which is symmetric; 2) $F^TFF'^TF' = F'^TF'$, which is also symmetric; 3) $F^TPF'' = F^TF''$. Hence F^TZ is symmetric iff F^TF'' is symmetric.
- For the condition $(I - P)Z = 0$, note that $(I - P)PF''$ and $(I - P)FF'^TF'$ both vanish, hence

$$(I - P)Z \ = \ (I - P)(\alpha_pF'' + (\alpha_w - \alpha_p)\, 2\, F'F^TF')$$

This proves the following:

Theorem: *A path of frames $F(t)$ is geodesic iff F^TF'' is symmetric, and*

$$(I - P)(\alpha_pF'' \ + \ 2\,(\alpha_w - \alpha_p)F'F^TF') \ = \ 0 \ ,$$

where α_p and α_w define a left- and right-invariant metric.

15.6 Construction of geodesics with regard to invariant metrics

The equations for geodesic paths of frames can be used to verify that a particular explicit construction yields geodesics. A dimensional argument will then confirm that this construction yields all geodesics.

Without loss of generality, we consider paths that start at the unit frame E_d, which we denote simply by E due to frequent use in this section. The functional form of paths we consider is

$$(*) \qquad F(t) \;=\; \exp(St)\; E\; \exp(Qt)\,, \qquad S \,\text{skew}\, p{\times}p\,,\;\; Q\,\text{skew}\, d{\times}d\,.$$

Hence $\exp(St) \in SO(p)$ and $\exp(Qt) \in SO(d)$ are both paths of rotations. The former transports the starting frame E through space starting at $F(0) = E$, while the latter modifies the whip spin of the path. It will be useful to have notation for the natural blocks of S:

$$S \;=\; \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix}\,,$$

where S_{11} is $d \times d$, $S_{12} = -S_{21}^T$ is $d \times (p-d)$, and S_{22} is $(p-d) \times (p-d)$. Note that $S_{11} = E^T S E$.

The goal is to apply the equations for geodesics to the paths $F(t)$. To this end the derivatives are:

$$\begin{aligned} F' &= \exp(St)\;(SE + EQ)\;\exp(Qt)\,,\\ F'' &= \exp(St)\;(S^2E + 2SEQ + EQ^2)\;\exp(Qt)\,. \end{aligned}$$

As a first application we obtain the whip spin matrix:

$$F^T F' \;=\; \exp(-Qt)\;(S_{11} + Q)\;\exp(Qt)$$

Thus, the matrix Q can be used to modify the whip spin generated by the transport with $\exp(St)$.

Proposition: *Paths of frames of the form $(*)$ have constant speed:*

$$g_{F(t)}(F'(t)) \;=\; \alpha_w \|S_{11} + Q\|^2_{Frob} \;+\; \alpha_p \|S_{21}\|^2_{Frob}$$

where α_w and α_p define a left- and right-invariant metric.

Proof: Calculate the contributions due to whip spin and plane motion and note that all terms $\exp(St)$ and $\exp(Qt)$ cancel:

$$\begin{aligned} \text{trace}((F^TF')^T(F^TF')) &= \text{trace}((S_{11}+Q)^T(S_{11}+Q)) \;=\; \|S_{11}+Q\|^2_{Frob}\\ \text{trace}(F'^T(I-P)F') &= \text{trace}((SE+EQ)^T(I-EE^T)(SE+EQ))\\ &= \text{trace}(E^TS^T(I-EE^T)SE)\\ &= \text{trace}((S_{11}^T,\; S_{21}^T)\begin{pmatrix} 0 & 0 \\ 0 & I_{p-d}\end{pmatrix}\begin{pmatrix} S_{11} \\ S_{21}\end{pmatrix}\\ &= \text{trace}(S_{21}^TS_{21}) \;=\; \|S_{21}\|^2_{Frob} \qquad \Box \end{aligned}$$

We now investigate the conditions under which the paths $F(t)$ are geodesic for a given left- and right-invariant metric. The first condition is symmetry of $F^T F''$:

$$\begin{aligned} F^T F'' &= \exp(Qt)^T \left(E^T S^2 E + 2E^T SEQ + E^T EQ^2\right) \exp(Qt) \\ &= \exp(Qt)^T \left(-E^T S^T SE + 2S_{11}Q - Q^T Q\right) \exp(Qt) \end{aligned}$$

It follows that $F^T F''$ is symmetric iff $S_{11}Q$ is symmetric:

$$S_{11}Q = (S_{11}Q)^T = Q^T S_{11}^T = QS_{11} \ ,$$

That is, we have symmetry iff S_{11} and Q commute.

We turn to the second condition for geodesic paths which requires calculation of $(I-P)F''$ and $(I-P)F'F^TF'$:

$$\begin{aligned} (I-P)F'' &= \exp(St)\,(I - EE^T)\,(S^2E + 2SEQ + EQ^2)\ \exp(Qt) \\ &= \exp(St)\,(I - EE^T)\ S(SE + 2EQ)\ \exp(Qt) \\ &= \exp(St) \begin{pmatrix} 0 & 0 \\ 0 & I_{p-d} \end{pmatrix} \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} \begin{pmatrix} S_{11} + 2Q \\ S_{21} \end{pmatrix} \exp(Qt) \\ &= \exp(St) \begin{pmatrix} 0 \\ S_{21}(S_{11} + 2Q) + S_{22}S_{21} \end{pmatrix} \exp(Qt) \\ (I-P)F'F^TF' &= \exp(St)\,(I - EE^T)\,(SE + EQ)E^T(SE + EQ)\ \exp(Qt) \\ &= \exp(St)\,(I - EE^T)\ SEE^T(SE + EQ)\ \exp(Qt) \\ &= \exp(St)\,(I - EE^T)\ SE(S_{11} + Q)\ \exp(Qt) \\ &= \exp(St) \begin{pmatrix} 0 & 0 \\ 0 & I_{p-d} \end{pmatrix} \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} \begin{pmatrix} S_{11} + Q \\ 0 \end{pmatrix} \exp(Qt) \\ &= \exp(St) \begin{pmatrix} 0 \\ S_{21}(S_{11} + Q) \end{pmatrix} \exp(Qt) \end{aligned}$$

Combining the two terms, we get the second condition for geodesic paths: We have $(I-P)(\alpha_p F'' + 2\,(\alpha_w - \alpha_p)F'F^TF') \ = \ 0$ iff

$$\alpha_p\,[S_{21}(S_{11} + 2Q) + S_{22}S_{21}] \ + \ 2\,(\alpha_w - \alpha_p)\,S_{21}(S_{11} + Q) \ = \ 0 \ .$$

Rearranging this equation yields:

Theorem: *A path of frames of the form* $(*)$ *is geodesic iff* S_{11} *and* Q *commute and*

$$\alpha_p\,S_{22}S_{21} \ + \ S_{21}\,((2\alpha_w - \alpha_p)\,S_{11} \ + \ 2\alpha_w\,Q) \ = \ 0 \ ,$$

where α_p *and* α_w *define a left- and right-invariant metric.*

Corollary: *The following are sufficient conditions for paths of frames of the form* $(*)$ *to be geodesic:*

- $S_{22} = 0$ *and* $Q = (\frac{\alpha_p}{2\alpha_w} - 1)\, S_{11}$ *, hence:*

$$F(t) = \exp(\begin{pmatrix} S_{11} & S_{12} \\ S_{21} & 0 \end{pmatrix} t) \begin{pmatrix} \exp((\frac{\alpha_p}{2\alpha_w} - 1)\, S_{11} t) \\ 0 \end{pmatrix}$$

  These paths generate all possible geodesics emanating from E for the left- and right-invariant metric defined by α_p and α_w.

- $S_{21} = 0$*, and S_{11} and Q commute; that is, these paths are pure whip spin: $F = E \exp((S_{11} + Q)t)$; they are universally geodesic for all left- and right-invariant metrics.*

- $S_{11} = Q = 0$ *and* $S_{22} = 0$ *, that is, these paths are pure plane motion; they are universally geodesic for all left- and right-invariant metrics.*

The conditions follow immediately from the theorem. The family of geodesics following from the first criterion is complete because the dimension of skew matrices S that satisfy the criterion equals the dimension of the Stiefel manifold $V_{d,p}$, namely, $pd - (d+1)d/2$.

The second and third criteria generate universally geodesic paths because the criteria are independent of α_w and α_p. $\square$

Note a couple of special cases:

- For $\alpha_p = 2,\ \alpha_w = 1$ the first criterion yields $Q = 0$, hence the geodesic paths are of the form $F(t) = \exp(St)E$. The speed measure is $g_F(F') = \|S\|^2_{Frob}$ due to $S_{22} = 0$.

- For $\alpha_p = \alpha_w = 1$ the first criterion yields $Q = -S_{11}/2$, hence the geodesic paths are of the form $F(t) = \exp(St)E \exp(-S_{11}t/2)$. The speed measure is $g_F(F') = \|SE\|^2_{Frob}$ due to $S_{22} = 0$.

15.7 $O(p)$-Invariant real matrices

We prove a theorem from linear algebra that underlies the invariance theorems for Riemannian metrics on Stiefel manifolds.

We consider a p-dimensional *real* vector space with inner product and linear maps from the vector space into itself. In particular, let $\mathcal{U}$ be a set of linear maps and A an individual linear map; assume that A commutes with all elements of $\mathcal{U}$: $AU = UA$ for all $U \in \mathcal{U}$. The question is under what conditions we can infer that A is a multiple of the identity: $A = c \cdot I_p$.

Theorem: *1) If for any two unit vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ there exists $U \in \mathcal{U}$ such that $U\boldsymbol{x} = \boldsymbol{y}$, and if A has a real eigenvalue, then $A = c \cdot I_p$.*
2) If for any two unit vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ there exists $U \in \mathcal{U}$ such that $U\boldsymbol{x} = \boldsymbol{y}$ and $U\boldsymbol{y} = \boldsymbol{x}$, then $A = c \cdot I_p$.

Proof: 1) The map A has a real eigenvalue λ by assumption. Let $\boldsymbol{x}$ be a unit length eigenvector for λ and $\boldsymbol{y}$ an arbitrary unit vector. Again by assumption $\boldsymbol{y} = U\boldsymbol{x}$ for some $U \in \mathcal{U}$, so $A\boldsymbol{y} = AU\boldsymbol{x} = UA\boldsymbol{x} = \lambda U\boldsymbol{x} = \lambda\boldsymbol{y}$.
2) A general linear map A has a complex eigenvalue $\lambda = \lambda_r + i\lambda_i$ over the complexification of the linear space and there exists a complex eigenvector $\boldsymbol{x} = \boldsymbol{x}_r + i\boldsymbol{x}_i$: $A\boldsymbol{x} = \lambda\boldsymbol{x}$. Because A is real, complex conjugation yields another eigenvalue and eigenvector: $A\bar{\boldsymbol{x}} = \bar{\lambda}\bar{\boldsymbol{x}}$. It follows that the real vectors $\boldsymbol{x}_r$ and $\boldsymbol{x}_i$ span an invariant real plane in which A acts as a rotation and dilation:

$$A(\boldsymbol{x}_r, \boldsymbol{x}_i) \;=\; (\boldsymbol{x}_r, \boldsymbol{x}_i)\begin{pmatrix} c_\phi & s_\phi \\ -s_\phi & c_\phi \end{pmatrix} r$$

where $\lambda = (c_\phi + is_\phi)r$ and $\boldsymbol{x}_r$ and $\boldsymbol{x}_i$ are unit length. See Halmos 1958, p. 164 for a derivation in a related case.

By assumption there exists $U \in \mathcal{U}$ such that $U\boldsymbol{x}_r = \boldsymbol{x}_i$ and $U\boldsymbol{x}_i = \boldsymbol{x}_r$. Because U commutes with A we have

$$AU(\boldsymbol{x}_r, \boldsymbol{x}_i) = A(\boldsymbol{x}_i, \boldsymbol{x}_r) = (\boldsymbol{x}_r, \boldsymbol{x}_i)\begin{pmatrix} s_\phi & c_\phi \\ c_\phi & -s_\phi \end{pmatrix} r \;,$$

$$UA(\boldsymbol{x}_r, \boldsymbol{x}_i) = (\boldsymbol{x}_i, \boldsymbol{x}_r)\begin{pmatrix} c_\phi & s_\phi \\ -s_\phi & c_\phi \end{pmatrix} r = (\boldsymbol{x}_r, \boldsymbol{x}_i)\begin{pmatrix} -s_\phi & c_\phi \\ c_\phi & s_\phi \end{pmatrix} r \;.$$

It follows $s_\phi = 0$, hence $\phi = 0$ or $\phi = \pi$, and finally $A = \pm rI_p$. Thus we showed the existence of a real eigenvalue, and part 1) applies. □

Corollary: *The matrix A is a multiple of the identity if one of the following holds:*
a) $p \geq 2$ and A commutes with all elements of $O(p)$;
b) $p \geq 3$ and A commutes with all elements of $SO(p)$.
c) $p \geq 2$ and A is symmetric and commutes with all elements of $SO(p)$.

Proof: a) The set $\mathcal{U} = O(p)$ satisfies the assumption of part 2) of the Theorem because it contains reflections that can map any pair of unit vectors onto each other.
b) In order to apply part 2) of the Theorem to $\mathcal{U} = SO(p)$ one needs $p \geq 3$: For two unit vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ consider the reflection that maps $\boldsymbol{x}$ and $\boldsymbol{y}$ onto each other in the plane spanned by $\boldsymbol{x}$ and $\boldsymbol{y}$; the determinant of this 2×2 reflection is -1. Pick an arbitrary vector $\boldsymbol{z}$ orthogonal to $\boldsymbol{x}$ and $\boldsymbol{y}$ and extend the reflection to a map on the full p-dimensional space by requiring $\boldsymbol{z} \leftarrow -\boldsymbol{z}$ and leaving the space orthogonal to $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$ fixed; the resulting map is orthogonal and has determinant $+1$. Note that a third dimension (represented by $\boldsymbol{z}$) was needed to correct the determinant to $+1$.
c) Symmetric A's have real eigenvalues, so part 1) of the Theorem applies. □

Fact: *For $p = 2$, if A commutes with all elements of $SO(2)$, then $A \in \mathbb{R} \cdot SO(2)$.*

Proof: By part 1) of the Theorem, if A has a real eigenvalue then $A = c \cdot I_2$. If A has a complex eigenvalue, then the first half of the proof of part 2) of the Theorem applies: A is a multiple of a rotation. $\square$

This last fact explains the exceptions that arise in two dimensions if one insists on $SO(2)$-invariance as opposed to $O(2)$-invariance.

15.8 Removing whip spin

We prove theorem 1 of section 7.3 by constructing for a given path of frames $\tilde{F}(t)$ another path $F(t)$ that has zero whip spin and generates the same path of planes: $\tilde{F}(t) = F(t)V(t)$ for some path $V(t) \in SO(d)$.

The proof is by deriving a differential equation for $V(t)$ from the condition of vanishing whip spin for $F(t) = \tilde{F}(t)V(t)^T$:

$$0 \;=\; F^TF' \;=\; V\tilde{F}^T(\tilde{F}V^T)' \;=\; V\tilde{F}^T(\tilde{F}'V^T \;+\; \tilde{F}V'^T) \;=\; VSV^T + VV'^T \;,$$

where $S(t) = \tilde{F}^T\tilde{F}'$ is the whip spin matrix of $\tilde{F}$. Dropping the factor V, transposing, and using skewness of S, we get the matrix differential equation

$$VS \;=\; V' \;,$$

which can be interpreted as a vector differential equation for the rows of V. By basic existence theorems we obtain a matrix path $V(t)$ satisfying the initial conditions $V(0) = I_d$. It remains to show that $V(t) \in SO(d)$:

$$(VV^T)' \;=\; V'V^T + VV'^T \;=\; VSV^T + VS^TV^T \;=\; 0$$

due to skewness of S. Therefore VV^T is constant and equal to $V(0)V(0)^T \;=\; I_d$. $\square$

15.9 Data rotation versus frame motion

We prove the fact mentioned at the end of section 9.4:

$$\|U'\|^2_{Frob} \;=\; \|PF'\|^2_{Frob} \;+\; 2 \cdot \|(I-P)F'\|^2_{Frob} \;,$$

where $U(t) = (F(t), G(t))$ is a path in $SO(p)$, decomposed into a $d \times p$-frame $F(t)$ and a $(p-d) \times p$-frame $G(t)$ with zero-whip spin. The following trivial calculations decompose the (essentially unique) speed measure $\|U'\|^2_{Frob}$ of data rotations:

$$\begin{aligned}
\|U'\|^2_{Frob} \;&=\; \|U^TU'\|^2 \;=\; \left\| \begin{pmatrix} F^TF' & F^TG' \\ G^TF' & G^TG' \end{pmatrix} \right\|^2_{Frob} \\
&=\; \|F^TF'\|^2_{Frob} \;+\; \|F^TG'\|^2_{Frob} \;+\; \|G^TF'\|^2_{Frob} \;+\; \|G^TG'\|^2_{Frob} \\
&=\; \|F^TF'\|^2_{Frob} \;+\; 2 \cdot \|G^TF'\|^2_{Frob} \;+\; \|G^TG'\|^2_{Frob}
\end{aligned}$$

The last equation follows because $U^TU' + U'^TU = 0$, hence $F^TG' = -(G^TF')^T$. We note that $GG^T = I - P$, where as always $P = FF^T$. Therefore

$$\|G^TF'\|_{Frob}^2 \;=\; \mathrm{trace}(F'^TGG^TF') \;=\; \mathrm{trace}(F'^T(I-P)F') \;=\; \|(I-P)F'\|_{Frob}^2,$$

which is just the term for plane speed in theorem 2 of section 9.3. The whip spin matrix G^TG' is zero by assumption. □

References

[1] Andrews, D. F. (1972), "Plots of High-Dimensional Data," *Biometrics* **28**, pp. 125–136.

[2] Asimov, D. (1985), "The grand tour: a tool for viewing multidimensional data," *SIAM J. Sci. Statist. Computing* **6** 1, pp. 128–143.

[3] Asimov, D., and Buja, A. (1994), "The grand tour via geodesic interpolation of 2-frames," in *Visual Data Exploration and Analysis, Symposium on Electronic Imaging Science and Technology*, IS&T/SPIE (Soc. for Imaging Sci. and Technology/Internat. Soc. for Optical Engineering).

[4] Bjorck, A., and Golub, G. H. (1973), "Numerical methods for computing angles between linear subspaces," *Mathematics of Computation* **27** 123, pp. 579–594.

[5] Buja, A., and Asimov, D. (1986), "Grand tour methods: an outline," *Computer Science and Statistics: Proc. of the 17th Symp. on the Interface between Comput. Sci. and Statist.*, Amsterdam: Elsevier, pp.63–67.

[6] Buja, A., Asimov, D., Hurley, C., and McDonald, J. A. (1988), "Elements of a viewing pipeline for data analysis," in *Dynamic Graphics for Statistics*, eds. W. S. Cleveland and M. E. McGill, Belmont, CA: Wadsworth, pp. 277-308.

[7] Buja, A., Hurley, C., and McDonald, J. A. (1986), "A data viewer for multivariate data," *Computer Science and Statistics: Proc. of the 18th Symp. on the Interface between Comput. Sci. and Statist.*, Amsterdam: Elsevier.

[8] Buja, A., Cook, D., and Swayne, D. F. (1996), "Interactive high-dimensional data visualization," *Journal of Computational and Graphical Statistics* **5**, pp. 78–99. A companion video tape can be requested from andreas@research.at.com.

[9] Carr, D. B., Littlefield, R. J., and Nicholson, W. L. (1986), "Scatterplot matrix techniques for large N," *Computer Science and Statistics: Proc. of the 17th Symp. on the Interface between Comput. Sci. and Statist.*, Amsterdam: Elsevier, pp. 297–306.

[10] Carr, D. B. (1991), "Looking at large data sets using binned data plots," in: *Computing and Graphics in Statistics*, eds. A. Buja and P. A. Tukey, pp. 7–39, New York: Springer.

[11] Carr, D. B., Wegman, E. J., Luo, Q. (1996), "ExploreN: Design considerations past and present," Technical Report 129, Center for Computational Statistics, George Mason University, Fairfax, VA 22030.

[12] Chambers, J. M., Cleveland, W. S., Kleiner, B., and Tukey, P. A. (1983), *Graphical Methods for Data Analysis*, Belmont, CA: Wadsworth.

[13] Conway, J. H., Hardin, R. H., and Sloane, N. J. A. (1996), "Packing lines, planes, etc.: Packings in Grassmannian spaces," *Journal of Experimental Mathematics* **5**, pp. 139–159.

[14] Cook, D., and Buja, A. (1996), "Manual controls for high-dimensional data projections," technical report, Iowa State University and AT&T Laboratories.

[15] Cook, D., Buja, A., Cabrera, J., and Hurley, H. (1995), "Grand tour and projection pursuit," *J. of Computational and Graphical Statistics* **2** 3, pp. 225–250.

[16] Cook, D. R., and Weisberg, S. (1994), *An Introduction to Regression Graphics*, New York: Wiley.

[17] Donoho, D. L., Huber, P. J., Ramos, E. and Thoma, M. (1982), "Kinematic Display of Multivariate Data," in *Proc. of the Third Annual Conference and Exposition of the National Computer Graphics Association.*

[18] Duffin, K. L., and Barrett, W. A. (1994), "Spiders: a new user interface for rotation and visualization of N-dimensional point sets," in *Proceedings Visualization '94*, IEEE Computer Society Press, Los Alamitos, California, pp. 205–211.

[19] Friedman, J. H. (1987), "Exploratory projection pursuit," *Journal of the American Statistical Association*, **82**, pp. 249-266.

[20] Furnas G. W., and Buja A. (1994), "Prosection Views: Dimensional Inference through Sections and Projections," *Journal of Computational and Graphical Statistics*, **3**, pp. 323-385.

[21] Golub, G. H., and Van Loan, C. F. (1983), *Matrix Computations*, second edition, Baltimore, Maryland: The Johns Hopkins University Press.

[22] Halmos, P. R. (1958), *Finite-Dimensional Vector Spaces*, New York: Springer.

[23] Halmos, P. R. (1970), "Finite-Dimensional Hilbert Spaces," *The American Mathematical Monthly,* **77** 5, pp. 457–464.

[24] Hanson, A. J., and Heng, P. A. (1991), "Visualizing a fourth dimension using geometry and light," *Proceedings Visualization '91*, IEEE Computer Society Press, Los Alamitos, California, pp. 321–328.

[25] Hurley, C. (1987), *The Data Viewer: A Program for Graphical Data Analysis*, PhD Thesis and Tech. Report, Statistics Dept., University of Washington, Seattle.

[26] Hurley, C., and Buja, A. (1990), "Analyzing high-dimensional data with motion graphics," *SIAM Journal on Scientific and Statistical Computing*, **11** 6, pp. 1193-1211.

[27] Kobayashi, S., and Nomizu, K. (1969), *Foundations of Differential Geometry*, vol. II, New York: Springer.

[28] Inselberg, A. (1985), "The plane with parallel coordinates," The Visual Computer 1, New York: Springer, pp. 69–91.

[29] Littman, M., Swayne, D. F., Dean, N., and Buja, A. (1992), "Visualizing the embedding of objects in Euclidean space," *Computing Science and Statistics: Proc. of the 24th Symp. on the Interface*, Fairfax Station, VA: Interface Foundation of North America, Inc., pp. 208–217.

[30] Miller, J. J., and Wegman, E. J. (1991), "Construction of line densities for parallel coordinate plots," in: *Computing and Graphics in Statistics*, eds. A. Buja and P. A. Tukey, pp. 107–123, New York: Springer.

[31] McDonald, J. A. (1982), "Orion I: Interactive graphics for data analysis," in *Dynamic Graphics for Statistics*, eds. W. S. Cleveland and M. E. McGill, Belmont, CA: Wadsworth.

[32] Scott, D. W. (1985), "Average shifted histograms: effective non-parametric density estimation in several dimensions," *Ann. of Statist.*, **13**, pp. 1024–1040.

[33] Scott, D. W. (1992), *Multivariate Density Estimation: Theory, Practice, and Visualization*, New York, NY: Wiley.

[34] Scott, D. W. (1995), "Incorporating density estimation into other exploratory tools," *ASA 1995 Proceedings of the Section on Statistical Graphics*, pp. 28–35.

[35] Spivak, M. (1979), *Differential Geometry*, volume I, New York, NY: Wiley.

[36] Swayne, D. F., Cook, D., and Buja, A. (1998), "XGobi: Interactive Dynamic Data Visualization in the X Window System," *Journal of Computational and Graphical Statistics*, **7** 1, pp. 113-130.

[37] Tierney, L. (1990), *Lisp-Stat*, New York, NY: Wiley.

[38] Tukey, J. W. (1987), "Comment on 'Dynamic graphics for data analysis' by Becker et al.," *Statistical Science*, **2** 355-395; also in *Dynamic Graphics for Statistics*, eds. W. S. Cleveland and M. E. McGill, Belmont, CA: Wadsworth.

[39] Tukey, J. and Tukey, P. (1981), "Graphical display of data sets in 3 or more dimensions," in: *Interpreting Multivariate Data*, ed. V. Barnett, New York: Wiley.

[40] Tukey, J. and Tukey, P. (1990), "Strips Displaying Empirical Distributions: I. Textured Dot Strips," Bellcore Technical Memorandum.

[41] Wegman, E. J. (1991), "The grand tour in k-dimensions," Technical report No. 68, Center for Computational Statistics, George Mason University.

[42] Wegman, E. J., and Carr, E. B. (1993), "Statistical graphics and visualization," in *Handbook of Statistics*, **9**, pp. 857-958, ed. C. R. Rao, Amsterdam: Elsevier.

[43] Wegman, E. J., and Luo, Q. (1996), "High dimensional clustering using parallel coordinates and the grand tour," Technical report No. 124, Center for Computational Statistics, George Mason University.

[44] Wegman, E. J., Solka, J. L., and Poston, W. L. (1996), "Immersive methods for mine warfare," Technical report No. 123, Center for Computational Statistics, George Mason University.

[45] Wong, Y.-C. (1967), "Differential geometry of Grassmann manifolds," *Proc. of the Nat. Acad. of Sci.*, **57**, 589-594.

[46] Young, F. W., Kent, D. P., and Kuhfeld, W. F. (1988), "Dynamic graphics for exploring multivariate data," in: *Dynamic Graphics for Statistics*, eds. W. S. Cleveland and M. E. McGill, Belmont, CA: Wadsworth.

[47] Young, F. W., and Rheingans, P. (1991), "High-dimensional depth-cuing for guided tours of multivariate data," in: *Computing and Graphics in Statistics*, eds. A. Buja and P. A. Tukey, pp. 239–252, New York: Springer.