U.S. Department of Commerce
National Oceanic and Atmospheric Administration
National Weather Service
National Centers for Environmental Prediction
5200 Auth Road
Camp Springs, MD 20746-4304

Office Note 447

A GEOMETRICAL APPROACH TO THE SYNTHESIS OF SMOOTH ANISOTROPIC COVARIANCE OPERATORS FOR DATA ASSIMILATION

R. James Purser*
Science Applications International Corp., Beltsville, Maryland

December 6, 2005

* email: jim.purser@noaa.gov

Abstract

This article describes a fairly general approach to the synthesis of smooth spatial covariance operators, based on the application of certain geometrical properties of regular lattices. The work is motivated by the need, in data assimilation, to be able to convolve an arbitrary distribution of gridded data by a smooth covariance function, which we may assume locally to be the superposition of at most a small set of bell-shaped quasi-Gaussians. This operation must be carried out repeatedly during the course of the iterative solution to the optimization problem that constitutes the major computational task of objectively assimilating new data. Since it tends to be highly non-local in all spatial dimensions, it becomes the main computational bottle-neck of the whole process. It is therefore imperative that the combination of numerical operations that collectively synthesize the covariance convolution are constructed to be as efficient as possible. With efficiency as one of the main objectives, we find that, once we have established the Gaussian form as the versatile building-block for the additive synthesis of more general covariance shapes, we are able to synthesize the Gaussian covariance profile itself via only a modest number of orientations of parallel line-smoothing filters applied sequentially.

An earlier publication supplied a cursory outline of the most basic forms of these algorithms, referred to respectively as the Triad and Hexad methods in two and three dimensions. However, a great deal more about the underlying geometry and symmetry principles of these techniques has been discovered recently. This allows us to extend the methods systematically to achieve a greater degree of spatial consistency in the case of the markedly inhomogeneous statistics so typical of meteorological or oceanic data analysis. The modified blended versions of these methods utilize a few additional directions of filtering and a systematic reconstruction of the collective smoothing parameters to ensure a smoother result overall.

We describe here some of these formal geometrical insights and the new algorithms they have inspired and we briefly touch upon the construction of four-dimensional fully anisotropic covariances by the analogous extension of this generic approach to covariance synthesis in dimensions greater than three.

1. INTRODUCTION

The synthesis of covariances in an efficient manner is one of the crucial components of a modern optimal data assimilation procedure in atmospheric or oceanic analysis (Thiébaux and Pedder 1987, Daley 1991, Ghil et al. 1997, Lewis et al. 2006). The simulated covariance shapes must be reasonably close to those that describe the statistical uncertainty of the background error field, yet they must be of a form that permits rapid execution on a computer since, in almost all the standard optimization techniques typically considered for problems of this kind, the convolution operation involving the covariance kernel must be repeated many times (about 100, for example) in the course of the iterations used. Since this part of the process often comprises the main bottle-neck, it is very important to make the synthesis of the covariance as efficient as possible, subject to retaining a sufficient degree of fidelity to what is presumed to be the appropriate structure of background error.

The geophysical data assimilation problem is generally a multivariate one since dynamical considerations force us to take explicit account of the strong statistical correlations that occur, for example, between wind and mass fields. Other important correlations exist among the thermodynamic variables and the various tracers in the atmospheric case, or between temperature and salinity in the oceanic case. These aspects of variability can usually be separated out by the adoption of appropriate new variables that combine different families of the characteristic dynamical *modes*. For example, a stream function and mass combination that corresponds to the geostrophically *balanced* structures usually possesses substantially greater variance than the remaining component of *unbalanced* stream function, expressed in a common measure such as an energy norm. Similarly, (unbalanced) horizontal divergence is inhibited, though not absolutely prohibited, by assigning a relatively small amplitude of covariance to this quantity's background error. Except in boundary layers where surface friction effects are felt, the correlations between these balanced and unbalanced parts can be neglected without detriment to the resulting analysis. In this way, a multivariate problem is conveniently separated into univariate (scalar) components that are then only indirectly linked in the optimization problem through the observations.

Having reduced a multivariate covariance into separated scalar covariances, we model each scalar's covariance using a particular characteristic spatial profile. Historically, such covariances have been artificially restricted to a geographically uniform *homogeneous* form. Also, there has been an almost universal tendency to model the covariances of the derived balanced and unbalanced scalar variables as having isotropic functional forms, as if the natural kinematic stretching effects of deformational flows, or the well developed horizontal and vertical thermal gradients in frontal zones and jets could be neglected. In the light of recent studies such as Otte et al. (2001), we find such assumptions less easy to justify, especially now that the computational tools are being formulated that make the treatment of more general covariances computationally feasible. The proper shape for a covariance is not easy to define or to assess from even vast stores of data, especially when these shapes are permitted to be adaptive to geography and ambient flow characteristics. However, if, as is usually the case, the scalar analysis variables can be chosen such that their covariances are predominantly characterized by bell-shaped spatial profiles, then it is reasonable to suppose that the following principle holds[†]:

- The best bell-shaped isotropic and homogeneous covariance model of background error tends to possess fatter tails and sharper peaks than a typical example of the best spatially adaptive model of the same background error.

The justification for this idea, which we may refer to as the **Kurtosis Principle**, is that the isotropic and homogeneous model must be taken as an average, or **mixture** of a great variety of the best adaptive profiles that have different degrees and orientations of spatial stretching and a wide range of characteristic spatial scales. But the averaging of such a mixture inevitably tends to increase the **kurtosis**. The kurtosis is a measure of the size of the fourth central moment of a distribution normalized by the square of the second central moment and, informally, quantifies the degree of fat-tailedness of the distribution. While valid distributions, including some which formally qualify as candidates for covariances, may have a kurtosis less

[†] I thank Dr. David Parrish for the discussions on this topic that stimulated the work presented in appendix A.

than that of a Gaussian (for which the value is $\kappa = 3$), it is hard to find distributions as *natural* as the Gaussian (in terms of smoothness, for example) that possess a lower kurtosis. We may use this as an argument to claim that, in an adaptive-covariance assimilation scheme it is more justifiable to use the Gaussian model than it is in schemes where, owing to technical limitations, the covariance has to be artificially restricted to being isotropic and homogeneous. Some of these fundamental properties of kurtosis are dealt with in appendix A for the one-dimensional case but, as shown in Purser 2004, the concept has a generalization in higher dimensions also. Even when the Gaussian assumption by itself is invalid, it is often possible to capture the main features of a covariance through a linear superposition of just a few Gaussian contributions (e.g. Wu et al. 2002). For all these reasons, we shall proceed under the assumption that the goal of our covariance synthesis is primarily to produce an anisotropic quasi-Gaussian distribution as efficiently as possible.

The Gaussian model enjoys several important analytical properties, some of which have been exploited in data assimilation. This is true not just for statistical analyses, but also for the empirical *successive corrections* variety, as in the Barnes (1964) or in the earliest recursive filtering analysis scheme of Purser and McQuigg (1982) and Hayden and Purser (1995), all essentially variants of the methods first pioneered by Bergthórsson and Döös (1955) and Cressman (1959). First we note that the result of a simulated diffusive process applied for a finite *pseudo-time* results in an effective Gaussian convolution. For essentially isotropic diffusion this property was exploited by Derber and Rosati (1989) in their ocean analysis. More recently Weaver and Courtier (2001) have extended this technique quite effectively to produce anisotropic covariances for an ocean assimilation, and Weaver and Ricci (2004) show that the restriction to Gaussians may be relaxed somewhat by the strategy of using a limited number of implicit solver iterations in the diffusion process. As Courtier (1997) has discussed, linear optimal data assimilation methods may be solved equivalently either by a linear inversion in model space or, following the tradition established by the classical *optimum interpolation* approach of Gandin (1963), in the space of measurements. The equivalence of these dual methods extends to the numerical condition numbers of the two methods. Using the framework of the Physical Space Analysis System (PSAS) of da Silva et al. (1995), Gaspari and Cohn (1998, 1999) have shown how to construct compact-support approximations to Gaussians from rational functions of relative displacement. In this way, the covariances may be evaluated efficiently without the supporting structure of a numerical grid, making this covariance choice very attractive for measurement-space analysis. Given the widespread preference for Gaussian-based covariance models it is worthwhile to examine some of the possible reasons for this preference.

A notable property of the Gaussian family is that it enjoys *closure* under linear deformation of the space in which it is embedded. Since the Fourier transformation (which is the spatial power spectrum) is also Gaussian, it too inherits this property.

There is another well known property of the Gaussian which is crucial to the systematic construction of covariances by the means of what we have called **Triad** and **Hexad** methods in two and three dimensions respectively: Gaussians convolved with each other produce a Gaussian. That this remains true even when the combining factors are degenerate Gaussian distributions confined to lines (regardless of orientation) makes it possible to build up any homogeneous two-dimensional anisotropic Gaussian from a triad of (three) line-smoothers, and to generate a three-dimensional one from a hexad of (six) line-smoothers. The lines referred to

are generalized grid lines that thread the grid possibly at oblique (non-Cartesian) orientations.

Note that the number of line-smoothers required is equal to the number of the independent components, in the relevant dimensionality, of a symmetric **aspect tensor** (a term that generalizes the concept of an aspect ratio). This tensor represents the centered second-moment of the response of the full product filter, normalized by its zeroth moment. The triad and hexad algorithms outlined in Purser et al. (2003b) efficiently determine the line orientations and associated smoothing coefficients that correspond to a given aspect tensor at each point of an analysis grid.

At NCEP, we favor the *recursive filter* technique (for example, Purser et al. 2003a) to provide the quasi-Gaussian line-smoothers for the new anisotropic analysis schemes under development, because of this technique's inherent computational efficiency. However, the triad and hexad methods can equally well be combined with other styles of line-filtering, including simulated iterative diffusion, direct convolution and multigrid-based smoothing strategies.

It is to be expected that, when the characteristic scale over which the covariance parameters vary is comparable with the covariance scale itself, then several iterations of the sequential line-smoothers may be needed to ensure a result devoid of numerical artifacts that betray the line-by-line construction process. However, we have found that, even when the changes in the covariance parameters are very gradual, the transition from one triad or hexad to another is sometimes accompanied by a numerical **dislocation** in the result which is not sufficiently mitigated merely by increasing the number of iterations in the construction process.

The problem of numerical dislocations occurs because, although the line-filter second moments, or **weights** prescribed by the triad or hexad algorithms for the oblique orientations are continuous when they approach zero, they do not approach the zero value sufficiently smoothly. Much of this article deals with a generalization of these procedures which, by including a larger set of line-smoothers at each geographical point, enables the smoothing weights that vanish to do so more gradually, so the aforementioned numerical artifacts may be completely and elegantly eliminated. The modified methods will be referred to as the **blended triads** and **blended hexads** algorithms. While the geometrical analysis required to develop the capability to perform these algorithms is not trivial, the application of them remains a relatively straight-forward procedure.

In order to establish the geometrical framework, we introduce the so-called **aspect space** of covariance shapes in the next section and describe its various metrical characterizations. We particularly examine how the line smoothers on the two-dimensional lattice ('Z^2') are interpreted geometrically in the corresponding aspect space, since the geometrical principles established here form the foundation for the various smoothing algorithms described in the later sections. Section 3 describes the construction and geometrical interpretation of the basic triad algorithm, the simplest of our algorithms to handle anisotropic shapes, which applies to the 2D lattice. Section 4 extends this analysis and description to the basic hexad algorithm for the 3D lattice and devotes separate subsections to the topics of symmetries and Galois fields, which have been of crucial significance in constructing geometrically elegant and versatile forms of this algorithm. Roughly, the second half of this paper is devoted to the refinements of the basic triad and hexad algorithms to their corresponding *blended* forms. This has been found necessary, or at least desirable, in cases of pronounced spatial inhomogeneity in the aspect tensors, in order to avoid artificial unsightly numerical effects wherever the triads or hexads of

line smoothers change to a new configuration. Section 5 treats the *blended triads* algorithm and shows idealized examples of its performance compared to the basic triad method. Section 6 treats the considerably more complex case of the *blended hexads* method for the 3D lattice. The geometrical manipulations here involve up to six dimensions, and the systematic reduction of the problems to manageable pieces is discussed within several specialized subsections. Section 7 discusses (without any great detail) the prospects for extending these methods to four dimensions. Here, the complexity of the geometrical problems increase again by orders of magnitude. While the *basic* form of the 4D algorithm has been successfully determined, the extension to a *blended* form of the algorithm remains an unfinished project at this time. A further discussion of the relevance and future prospects of these methods is provided in the concluding section 8. Some of the more technical topics are relegated to the appendices.

2. Aspect space

(*a*) *Geometry of the aspect space associated with* R^n

We characterize the multidimensional Gaussian shapes,

$$B(\mathbf{x}) = \exp(-\mathbf{x}^T \mathsf{A}^{-1} \mathbf{x}/2), \tag{2.1}$$

through the geometry of their aspect tensors, A. In two real dimensions (R^2) these tensors, being symmetric, possess three independent components, A_{xx}, A_{yy} and A_{xy}.

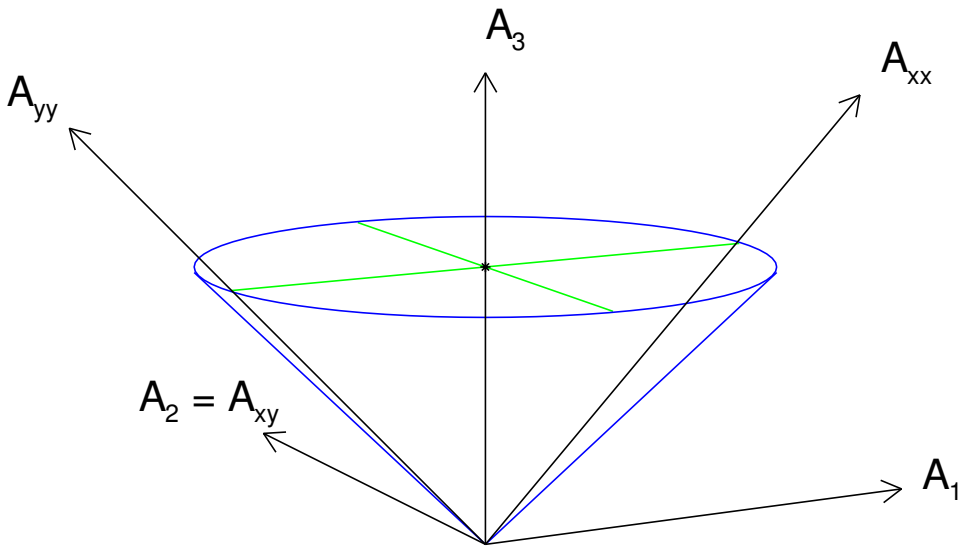

Figure 1. Transformation of the aspect tensor components to form a 3-vector whose region of validity is the interior of an infinite circular 'aspect cone'.

As shown in Purser (2004), this vector of three independent components may be conveniently rotated to a standard form, the **aspect vector**, $\mathbf{A}$, whose components are:

$$A_1 \;=\; (A_{xx} - A_{yy})/2, \tag{2.2a}$$

$$A_2 = A_{xy}, \tag{2.2b}$$
$$A_3 = (A_{xx} + A_{yy})/2, \tag{2.2c}$$

as sketched in Fig. 1.

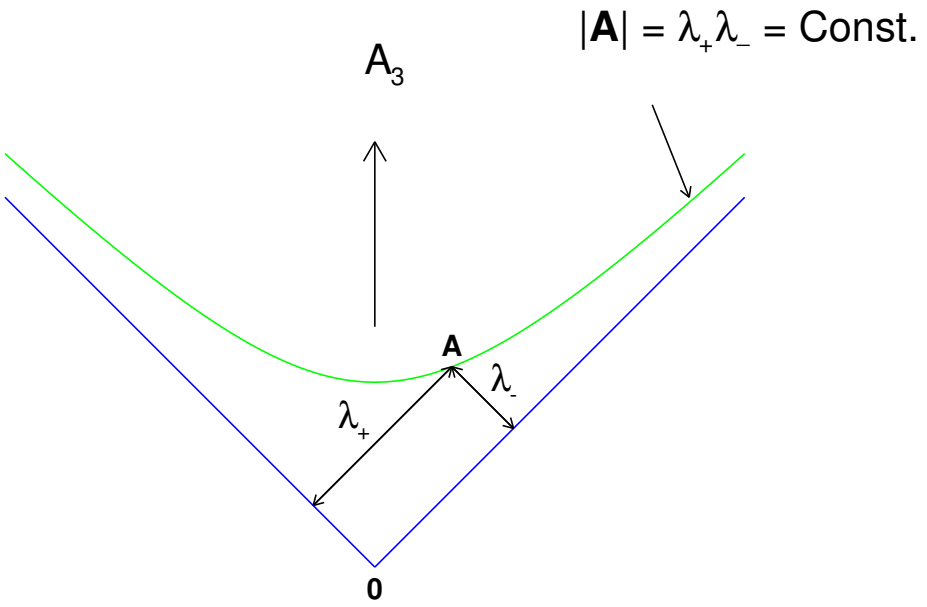

Figure 2. Schematic cross-section bisecting the aspect cone in a plane containing **A**, with a geometrical interpretation of the eigenvalues of the aspect tensor as the pair of orthogonal distances to the boundary of the cone. The locus of aspect tensors corresponding to any constant value of the determinant forms a convex hyperboloid. Central projection (i.e., by a ray passing through the apex) onto the surface of this kind associated with unit determinant provides one way of standardizing the scale of the aspect tensor while preserving its shape.

The component, A_3, is aligned with the axis of the **aspect cone** occupying the region:

$$(A_1^2 + A_2^2) \leq A_3^2, \tag{2.3a}$$
$$A_3 \geq 0, \tag{2.3b}$$

which constitutes the entirety of the feasible region where the eigenvalues,

$$\lambda_\pm = A_3 \pm (A_1^2 + A_2^2)^{1/2}, \tag{2.4}$$

of A are non-negative, as required for a distribution of the form (2.1) to make sense as the profile of a smoothing kernel. When the apex angle of the cone is, by the choice of metric, made a right-angle, the eigenvalues λ_+ and λ_- can be thought of as the orthogonal distances to the boundary of the cone, as shown schematically in Fig. 2.

In any number n of physical dimensions (i.e., $\mathbf{x} \in R^n$), a natural measure of separation for aspect tensor A and an infinitesimal perturbation to it, $\mathsf{A} + \mathsf{dA}$, is obtained by taking $\sqrt{2}$ times the smallest Frobenius norm (e.g. Golub and Van Loan 1989) among infinitesimal spatial deformation operators dT that transform A into $\mathsf{A} + \mathsf{dA}$ according to the **congruence**:

$$\mathsf{A} + \mathsf{dA} \equiv (\mathsf{I} + \mathsf{dT})\mathsf{A}(\mathsf{I} + \mathsf{dT})^T. \tag{2.5}$$

The square of this natural distance measure is then:

$$\|\mathsf{A},\, \mathsf{A}+\mathsf{dA}\|^2 = \frac{1}{2}\mathrm{trace}[(\mathsf{A}^{-1}\mathsf{dA})^2]. \tag{2.6}$$

In terms of this metric, any affine transformation of the spatial domain, such as a shear, rotation, contraction, or a combination of these, which transforms the aspect tensor according to the congruence:

$$\mathsf{A}' = \mathsf{U}\mathsf{A}\mathsf{U}^T, \tag{2.7}$$

for an arbitrary nonsingular operator U, induces an **isometry** in aspect space, in the sense that

$$\|\mathsf{A}',\, \mathsf{A}'+\mathsf{dA}'\| = \|\mathsf{A},\, \mathsf{A}+\mathsf{dA}\|. \tag{2.8}$$

Using this result to transform one of an arbitrary pair of aspect tensors, A_1, A_2, to the identity, I, it is readily shown that the formula for an infinitesimal separation integrates to the following formula for a finite separation:

$$\|\mathsf{A}_1,\, \mathsf{A}_2\|^2 = \frac{1}{2}\sum_{i=1}^{n}(\log \Lambda_i)^2, \tag{2.9}$$

where Λ_i are the eigenvalues of $(\mathsf{A}_1^{-1}\mathsf{A}_2)$. Equivalently,

$$\|\mathsf{A}_1,\, \mathsf{A}_2\|^2 = \frac{1}{2}\mathrm{trace}\{[\log(\mathsf{A}_1^{-1}\mathsf{A}_2)]^2\}, \tag{2.10}$$

where the logarithm function has been extended in the natural way to tensor arguments. Transformations U that map the grid into itself have matrix representations that have integer components and determinants of either ± 1. The transformation of tensor A is therefore of a kind that preserves its determinant. Equivalently, such a transformation confines the image of each vector $\mathbf{A}$ to a manifold of dimension $(n-1)(n+2)/2$ in the aspect cone everywhere normal to the vector $\mathbf{A}$ itself in the sense implied by the metric (2.6). We can choose a representative manifold to correspond to aspect tensors having unit determinant. When distances within this manifold are measured by the induced metric (2.6) we find that, for each $n > 1$, we are dealing with a particular variety of non-Euclidean geometry of negative intrinsic curvature.

(*b*) *Geometry of the aspect space associated with* Z^2

Returning to our two-dimensional ($n = 2$) example, the surfaces of constant determinant, $D = \lambda_+\lambda_-$, form concentric hyperboloids:

$$A_3^2 - (A_1^2 + A_2^2) = D, \tag{2.11}$$

(an example of which is sketched in Fig. 2) upon each of which the natural metric (2.6) induces the classical hyperbolic geometry with negative-unit Gaussian curvature. (It is the choice of the otherwise arbitrary constant term, $\frac{1}{2}$ in (2.6) that gives us this convenient magnitude for the curvature.) It is useful to centrally project all aspect vectors onto the representative $D = 1$ member of this family of surfaces for the purposes of visualization. In effect, this normalizes

the *scale* of the covariance without altering its *shape*. A **map** of this surface is obtained by a projection of rays about a focus at the origin, $\mathbf{A} = [0, 0, 0]^T$, through the plane normal to the cone's axis, $A_3 = 1$. This leads to a **gnomonic** mapping in which the space is projected to the interior of the unit disk and the shortest paths with respect to metric (2.6), or **geodesics**, of aspect space map to straight lines segments within this circle. This is traditionally referred to as the **Klein** representation of classical hyperbolic geometry (Hilbert and Cohn-Vossen (1999). Visualizing the outer regions of the map is often made easier by adopting the alternative **stereographic** ray projection instead. In this case, the projection focus is placed at $\mathbf{A} = [0, 0, -1]^T$ and the mapping plane at $A_3 = 0$ in order to map the geometry once more to the interior of the unit disk. This yields what is traditionally referred to as the **Poincaré** representation of hyperbolic geometry, with geodesics now projecting to circular arcs that intersect the limiting circle normally.

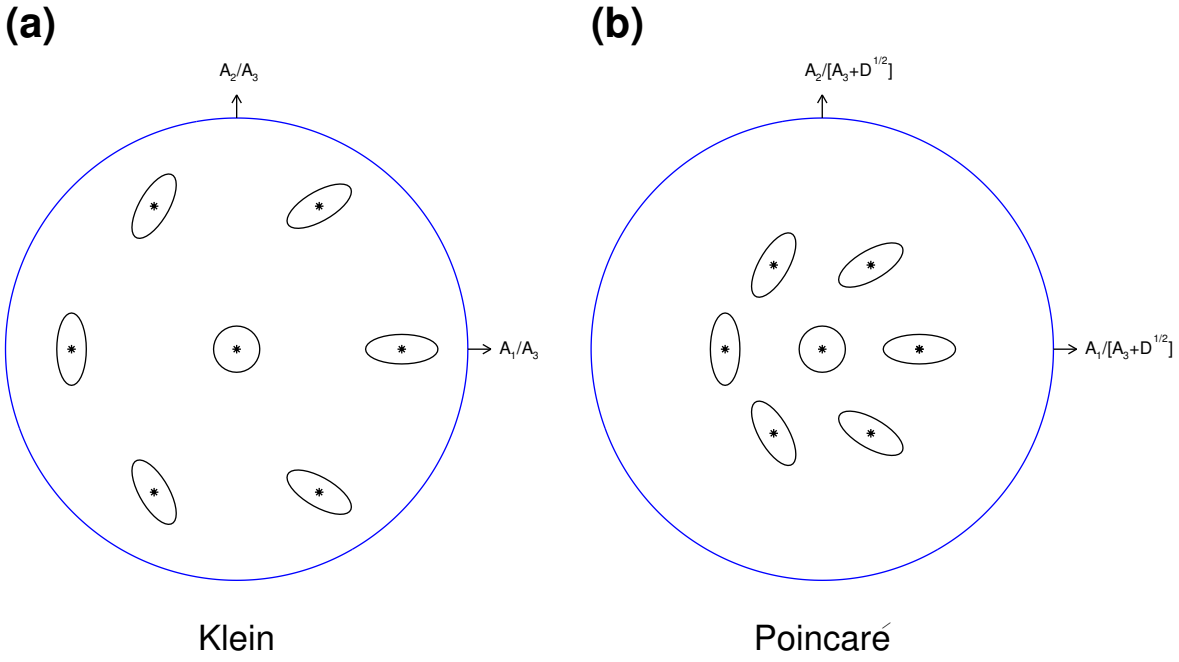

Figure 3. Representations of aspect space corresponding to two-dimensional aspect tensors, normalized by two commonly used map projections. (a) the 'Klein', or gnomonic, representation; (b) the Poincaré, or stereographic, representation. In both cases, the shapes of the corresponding covariance profiles are depicted at selected points by a single contour. The same set of aspect tensors is depicted in both maps.

In order to see the way shapes are mapped by these two standard representations of the hyperbolic aspect space, Fig. 3 shows the Klein (a) and Poincaré (b) representations side by side with a configuration of covariance shapes (delineated by a single representative contour) centered at their proper positions within the respective maps. We may also examine how sets of grid line orientations appear in these maps. A lattice **generator** consists of a non-vanishing and irreducible integer-vector. The position vectors of all those unit-lattice points that are *visible* (unobscured by other lattice points) from the origin are collectively equivalent to the set

of generators. A **generator image** in aspect space is the aspect vector equivalent to the rank-one tensor obtained as the outer product of the lattice generator with itself. Thus, in the context of Z^2, the generators $\mathbf{g}_1 = [1, 0]^T$ and $\mathbf{g}_2 = [0, 1]^T$ define the Cartesian 'x' and 'y' directions of the lattice. As a tensor, the generator image of $\mathbf{g}_1$ has components $A_{xx} = 1$, $A_{yy} = 0$, $A_{xy} = 0$. Likewise, the tensorial generator image of $\mathbf{g}_2$ has components, $A_{xx} = 0$, $A_{yy} = 1$, $A_{xy} = 0$. Their respective vectorial generator images are therefore: $\mathbf{A}_1 = [1/2, 0, 1/2]^T$, $\mathbf{A}_2 = [-1/2, 0, 1/2]^T$. The image of the negative of any generator is clearly identical to the image of the generator itself, since the transformation involved is homogeneous of second degree. Note that all rank-one aspect tensors map to the boundary of the aspect cone and, if obtained from integer-component generator vectors, twice their image vector components, i.e., $[2A_1, 2A_2, 2A_3]$, always forms a (possibly degenerate) Pythagorean integer triple[‡].

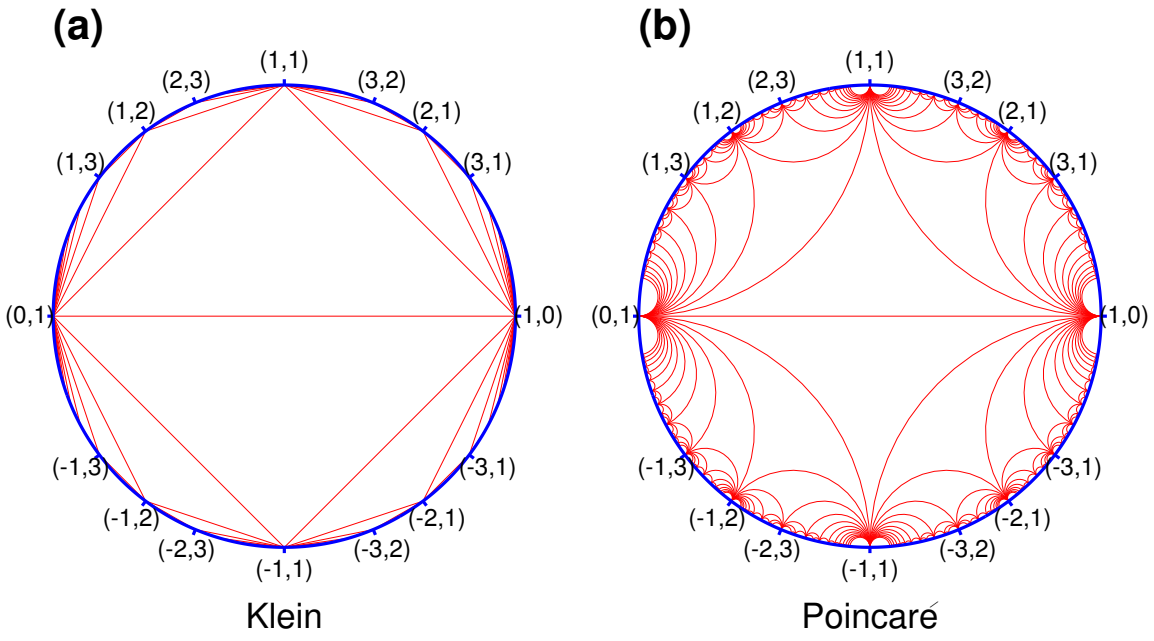

Figure 4. The honeycomb of triads of aspect space as they appear in (a) the Klein representation; (b) the Poincaré representation. A few of the generators of the two-dimensional unit lattice are marked at the appropriate locations around the limit circles of these maps.

Fig. 4 shows the Klein (a) and Poincaré (b) map representations of the images of some of the integer lattice generators, as indicated, together with connecting geodesics that divide the hyperbolic space into the regions associated with triads of these generators. These connecting lines are the projections of the edges of the polyhedral shell forming the (lower) surface of the convex hull of the aspect space images of all the grid generators. Given that the negative of a generator is, for all practical purposes, the equivalent of that generator, we can exploit that choice of sign in expressing each member of the feasible triad of generators written in the

[‡] In fact, *all* the Pythagorean triples, once simplified by division of common factors, are expressible in this way.

standard tableau form:

$$\begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{g}_3 \end{bmatrix}, \tag{2.12}$$

so that,

$$\mathbf{g}_1 + \mathbf{g}_2 + \mathbf{g}_3 = 0. \tag{2.13}$$

Another property that every feasible triad is found to possess is that:

$$\det\{\mathbf{g}_1; \mathbf{g}_2\} = \pm \det\{\mathbf{g}_2; \mathbf{g}_3\} = \pm \det\{\mathbf{g}_3; \mathbf{g}_1\} = \pm 1. \tag{2.14}$$

Although, by inverting the order of some of the triad generators, we could insist that this determinant is always $+1$, the triad algorithm can be put into a more symmetrical form when the sign of this determinant alternates from one triad to the next. The aspect space triangles associated with the triads of line generators tile the hyperbolic domain without gaps in a completely symmetrical configuration when interpreted in terms of the natural metric of this space. Thus, despite the distorting effects of the map projections, every triangle is actually equivalent in shape and size. A celebrated theorem of Gauss relates the area of a polygon inscribed by geodesics in a surface of constant curvature to its summed exterior angle excess, defined as the sum of exterior angles, minus 2π. For example, in the Euclidean plane, an equilateral triangle has three identical exterior angles (the angle through which one turns as one traces out the outline of the polygon) of $2\pi/3$, giving zero excess, which holds for all other simply-connected polygons too. In a uniformly curved surface the angular excess is minus the product of the curvature and the area. Thus, for the triangular **tiles** of the hyperbolic aspect space, where the interior angles are degenerate and where the curvature is minus-unity, the angular excess, and hence each tile's area, is π. However, the largest *diameters* of the tile are infinite, so toward the three vertices, each tile tapers exponentially, as interpreted according to the natural metric (2.6).

Any valid aspect vector maps, by central projection, into either: (i) the interior of precisely one triad or, (ii) to a point on the interface between two adjacent triads. The projection takes the form,

$$\mathsf{A} = \sum_{i=1}^{3} w_i (\mathbf{g}_i \mathbf{g}_i^T) \tag{2.15}$$

In the case (i), the three **weights** w_i by which the generator images are modulated to recover the target aspect vector are all *positive*; in the case (ii), two of the weights, associated with the two generators common to both adjoining triads, are positive but a *zero* weight is assigned to the third, ambiguous member of either of the two possible triads. Thus, the triad decomposition is essentially unique, and the weights are always in proportion to the barycentric coordinates of the central projection of the target aspect vector into the plane containing the three generator images.

3. Geometry of the Basic Triad algorithm for Z^2

The basic **triad algorithm** initially takes a feasible trial triad of three generators satisfying (2.13) and (2.14), and carries out the projection (2.15) to determine, from the signs of the three w_i, whether this is *the* valid triad for the given aspect tensor. When one w_i is negative, the triad is *not* valid and the offending generator associated with this negative weight must be replaced by a better candidate. There is only one feasible alternative (apart from a trivial sign reversal of all three triad members) at each step that satisfies (2.13). One of the most symmetric prescriptions for what we shall refer to as the **transition rules** asks that, when the ith generator of the tableau is replaced, the new tableau is given by the rule, $R(i)$, defined for each i as follows:

$$R(1) = \begin{bmatrix} \mathbf{g}_2 - \mathbf{g}_3 \\ -\mathbf{g}_2 \\ \mathbf{g}_3 \end{bmatrix}, \qquad R(2) = \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_3 - \mathbf{g}_1 \\ -\mathbf{g}_3 \end{bmatrix}, \qquad R(3) = \begin{bmatrix} -\mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{g}_1 - \mathbf{g}_2 \end{bmatrix}. \tag{3.1}$$

It is readily verified that these rules preserve the condition (2.13) but switch the sign of the determinant in (2.14). Performing a transition, then reversing it according to the same rule exactly restores the original tableau. Each iteration finds a feasible triad closer to satisfying the positive-weights criterion and never fails to converge if the aspect tensor itself is a valid one. This iterative process is essentially equivalent to a special variant of the **simplex algorithm** of linear programming (e.g. Dantzig 1963), the connection being revealed most directly by applying a Legendre-Fenchel transformation (Rockafellar, 1996) to the geometrical picture of the aspect-cone structures we have developed here. The search for the valid triad amounts to a progression of steps along the infinite binary-tree formed by all the connecting triads. For the hexad, we shall see in the next section that the geometry and topology become considerably more complicated. Nevertheless, many of the same underlying principles will continue to apply.

(*a*) *Efficient scheduling of line filters*

Even assuming we have succeeded in decomposing the given field of aspect tensors into its triads of component line smoothing operators, a practical problem arises when the time comes to apply these line filters efficiently. With processing occurring in parallel the line filters must be partitioned amonst the available processors and the work synchronized in a way that ensures that the filtering operation along one line segment is not interrupted by one of the other line filtering operations on a second segment that intersects the first.

A solution to this problem of line filter scheduling involves the adoption of a 'color coding' of all the generalized grid line orientations using three colors such that, for every valid triad, the three line orientations are associated with the three colors. Such a color coding of the two dimensional lattice of line orientations is shown in Fig. 5a, where the origin is shown in solid black. A 2×2 pattern containing the three colors (and one uncolored element) is repeated in both coordinate directions to tile the lattice in a way that ensures that: (i) every integer vector corresponding to a ray generator has a color; (ii) every valid triad possesses all three colors. The second property, though perhaps less obvious, is easily proved inductively. Fig. 5b shows the Poincaré map of triads with some of the generators' colors, consistent with panel (a), indicated by the pointers arranged around the limiting circle.

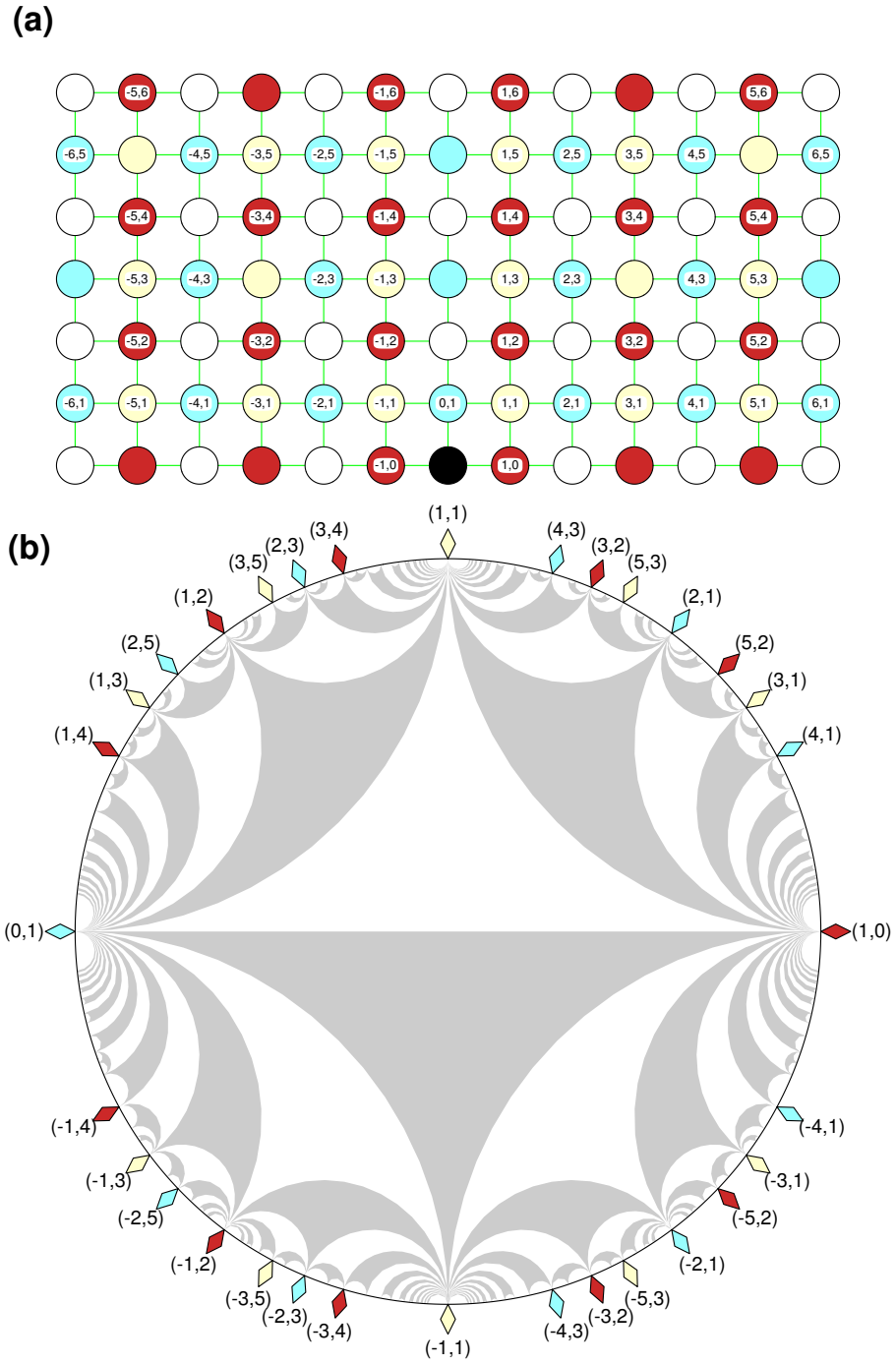

Figure 5. Panel (a) displays a coloring of the lattice of grid generators. Precisely one of the three colors occurs on, and thereby characterizes, any given generalized grid ray extending from the origin (marked solid black). The pattern is doubly periodic, repeating a basic 2×2 block. That this simple arrangement suffices to confer upon each valid triad the complete palette is corroborated by the corresponding Poincaré map of panel (b). The alternating gray and white triads, which match the alternating signs of their determinants in (2.14), serve to make the triad regions visible.

By performing all line operations synchronized by color, in three distinct stages, we necessarily avoid any possibility of a clash of intersecting line operators. We shall see in later sections that more elaborate color codings are sometimes necessary, as when we generalize the present basic triad algorithm to its blended form, or pass to lattices in higher dimensions. We defer discussion of the abstract algebraic tools which serve to guide these generalizations until we consider the three-dimensional examples, where the power and elegance of these algebraic techniques becomes self-evident.

4. Geometry of the Basic Hexad algorithm for Z^3

(*a*) *Symmetries*

Hexad generators of an idealized three-dimensional unit-spaced lattice Z^3 are sets of six integer 3-vectors whose images in the corresponding 6-dimensional aspect space form the vertices of the *tiles* that collectively make up the five-dimensional boundary of the convex hull of these image aspect vectors. Again, the tiles are congruent to one another, interpreted in terms of the natural metric (2.10) of this aspect space, but the configuration is less regular than in the triad case owing to fact that each hexad tile is *not* internally symmetric under all the possible permutations of its vertices.

As described in Purser et al. (2003b), the configuration of a hexad's six grid-line generators and their negatives lie at the vertices of a linearly transformed **cuboctahedron**. The geometrical reasons why this particular shape is selected are discussed in appendix B. Not counting the chirality-reversing *improper* rotations, which are undetectable in aspect space, the relevant symmetry group of this configuration, in the most isotropic representation of the geometry, would be referred to as the **octahedral group**. It permutes the diameters linking the centers of the four opposing pairs of triangles of this configuration, and in this sense is a representation of the abstract '**symmetric group**' ('S_4') whose order (number of elements) is $4! = 24$. We have found it more convenient to adopt a different convention for the labeling and grouping of standard generators than was used in Purser et al. (2003b); we shall adopt the convention that, for the integer Cartesian grid, Z^3, the first three standard generators, $\mathbf{g}_1$, $\mathbf{g}_2$, $\mathbf{g}_3$, when formed into a matrix, give positive unit determinant,

$$\det\{\mathbf{g}_1; \mathbf{g}_2; \mathbf{g}_3\} = +1, \tag{4.1}$$

(i.e., they form a right-handed basis for the grid). Also, interpreted as position vectors, they form one of the triangular facets of the linearly-deformed cuboctahedral configuration. The remaining three of the set of six standard generators then obey the 3-cyclic pattern:

$$\mathbf{g}_4 = \mathbf{g}_3 - \mathbf{g}_2, \tag{4.2a}$$

$$\mathbf{g}_5 = \mathbf{g}_1 - \mathbf{g}_3, \tag{4.2b}$$

$$\mathbf{g}_6 = \mathbf{g}_2 - \mathbf{g}_1. \tag{4.2c}$$

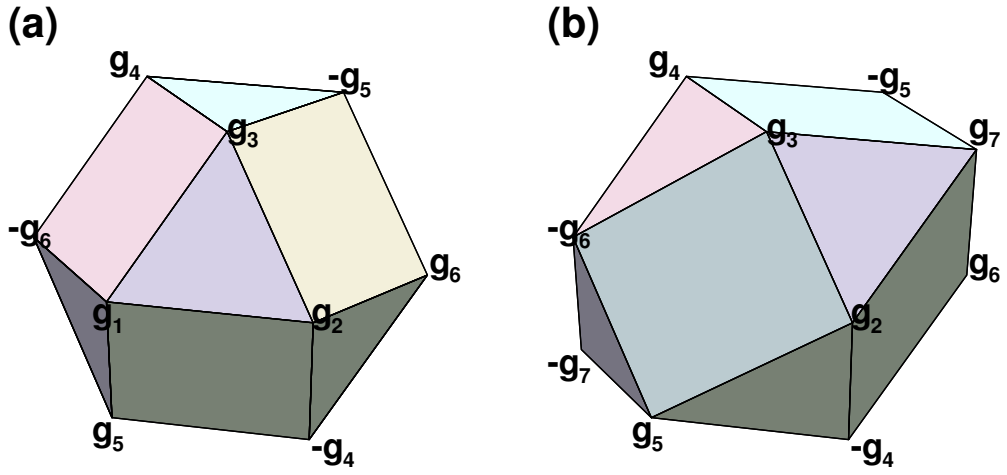

Figure 6. Schematic depiction of hexads of generators on a cubic lattice outlined by their associated convex hulls that form linearly-deformed cuboctahedrons. (a) A hexad configuration involving the six generators, $\mathbf{g}_1$–$\mathbf{g}_6$ and their negatives. (b) The configuration evolved from that of (a) by replacing generator pair, $\mathbf{g}_1$ and $-\mathbf{g}_1$ by the only valid alternatives, which we call, $\mathbf{g}_7$ and $-\mathbf{g}_7$. In practice, the generators of (b) would now need to be re-labeled so as to conform to the standard tableau, (4.3) and the rules, (4.1) and (4.2a)–(4.2c).

This puts $\mathbf{g}_1$ and $\mathbf{g}_4$ at opposite corners of a quadrilateral facet, and likewise for the other two associated pairs, $\mathbf{g}_2$ and $\mathbf{g}_5$, and $\mathbf{g}_3$ and $\mathbf{g}_6$ (see Fig. 6a). To highlight the significance of this pairing of generators that share the same quadrilateral, we shall often adopt the convention of tabulating the generators of the hexad in the matrix tableau generalizing the one used for the triad and which, for the hexad defined above, would be:

$$\left[\begin{array}{cc} \mathbf{g}_1 & \mathbf{g}_4 \\ \mathbf{g}_2 & \mathbf{g}_5 \\ \mathbf{g}_3 & \mathbf{g}_6 \end{array}\right] . \tag{4.3}$$

As a short-hand, it is also convenient sometimes, in large lists or tabulations of single generators, to drop the '$\mathbf{g}$' and simply write the algebraic sign and the generator's identifying subscript. Thus, we can explicitly list *all* the 24 valid ways of choosing from the complete set 12 generators, $\{\pm\mathbf{g}_i\}$, $i = 1, \ldots, 6$, those patterns of six that would, according to the relabeling implied by the tableau (4.3), also conforming to the rules (4.1)–(4.2c). But we do this in Table 1 by setting down the two columns of threes of the standard tableau into a combined row of six in each row and in each of the three major columns of the table. Here, the three rearrangements implied by the entries in each row of this table are related simply by 3-cyclic permutations of these entries.

The iterative algorithm for finding the valid hexad that has all non-negative weights w_i is very like the triad algorithm. The generator associated with a negative weight is replaced; but, in contrast to the triad algorithm, where valid aspect tensors cannot make more than one weight of a trial triad negative, in the case of the hexad algorithm, as many as three of the weights can be negative in an erroneous choice of structurally feasible trial hexad. It is customary to choose the identity of the most negative w_i as the criterion for selecting the direction to be replaced. Again, there is then only one feasible alternative replacement possible

TABLE 1. SET OF ALL POSSIBLE CONSISTENT EQUIVALENT LABELINGS OF THE HEXAD DEFINED BY (4.3), WHOSE TWO COLUMNS ARE HERE LISTED AS CONSECUTIVE ROWS OF THREE EACH SEPARATED BY A SEMICOLON. THE THREE LABELINGS IN EACH ROW OF THE TABLE ARE 3-CYCLIC ROTATIONS OF EACH OTHER IN WHICH THE PRIMARY TRIANGULAR FACET OF THE CUBOCTAHEDRON CORRESPONDING TO THE FIRST THREE LABELS IS THE SAME, BUT ROTATED TO ITS THREE DIFFERENT ORIENTATION RELATIVE TO THE LABELING STENCIL. THE PRIMARY TRIANGLES IMPLIED BY THE FOUR ROWS OF SECOND HALF OF THE TABLE ARE THE RESPECTIVE OPPOSITES OF THOSE OF THE FOUR ROWS OF THE FIRST HALF OF THE TABLE.

+1,	+2,	+3;	+4,	+5,	+6	+2,	+3,	+1;	+5,	+6,	+4	+3,	+1,	+2;	+6,	+4,	+5
−1,	−5,	+6;	−4,	−2,	+3	−5,	+6,	−1;	−2,	+3,	−4	+6,	−1,	−5;	+3,	−4,	−2
+4,	−2,	−6;	+1,	−5,	−3	−2,	−6,	+4;	−5,	−3,	+1	−6,	+4,	−2;	−3,	+1,	−5
−4,	+5,	−3;	−1,	+2,	−6	+5,	−3,	−4;	+2,	−6,	−1	−3,	−4,	+5;	−6,	−1,	+2
−1,	−3,	−2;	+4,	+6,	+5	−3,	−2,	−1;	+6,	+5,	+4	−2,	−1,	−3;	+5,	+4,	+6
+1,	−6,	+5;	−4,	+3,	−2	−6,	+5,	+1;	+3,	−2,	−4	+5,	+1,	−6;	−2,	−4,	+3
−4,	+6,	+2;	+1,	−3,	−5	+6,	+2,	−4;	−3,	−5,	+1	+2,	−4,	+6;	−5,	+1,	−3
+4,	+3,	−5;	−1,	−6,	+2	+3,	−5,	+4;	−6,	+2,	−1	−5,	+4,	+3;	+2,	−1,	−6

once the reject has been selected. However, as we have seen from its intrinsic symmetries, there are 24 different valid ways of labeling the generators of a given hexad and, since there are six ways in which a transition from one hexad to a neighbor can potentially occur, there are, in principle, $24^6 = 191{,}102{,}976$ distinct and equally correct ways of choosing the complete list of transition rules of the hexad algorithm. We greatly narrow down the choice by imposing some reasonable conditions of symmetry on the algorithm. In addition to endowing the method with a satisfying aesthetic form, the imposition of some symmetries will also lead later to some substantial practical advantages in the *blended* form of the algorithm, which we discuss in a section 6.

The first symmetry we shall impose on the transition rules is that, under simultaneous and equivalent cyclic rotations of the labels $\{1, 2, 3\}$ and $\{4, 5, 6\}$ of the generators, the *same* rules are retained. It is then sufficient to list the rules $R(1)$ for the arrangement into the tableau slots of (4.3) resulting from the replacement of $\mathbf{g}_1$, and the rules $R(4)$ for the case of a replacement of $\mathbf{g}_4$; the rules $R(2)$ and $R(3)$ can be recovered from $R(1)$, while rules $R(5)$ and $R(6)$ can be recovered uniquely from the form of $R(4)$. This reduces the number of combinations to $24^2 = 576$. The other condition, which becomes especially desirable in the case of the *blended hexads* method, is to ask that, if possible, the algorithm automatically chooses labeling orientations at each stage such that, on taking the hexad algorithm around any **circuit** of successive hexads, starting and ending at the same one, the initial and final labelings are always the same. (There are no nontrivial circuits in the triad case since the entire connected set of triads forms an unending **binary tree** rather than the multiply-connected **network** that characterizes the collection of hexads; however, note that even in the triad case, we fashioned the transition rules in order to preserve the original labeling under reversal of a transition.)

For the hexad, the **2-circuit** is what occurs when the second step undoes the first. When we further restrict the transition rules of our 576 3-cyclic possibilities to those that at least yield an invariance of labels under all six of the possible 2-circuits, we find that we are left with only 16 possible choices. These are given in Table 2, where just the $R(1)$ and $R(4)$ rules are explicitly listed, each 'unrolled' as in the format of Table 1, but the '7' in the table now referring to a new replacement generator, '$\mathbf{g}_7$'. For the cyclic rotations of the transition rules

to the other positions, we would also need to define '$\mathbf{g}_8$' and '$\mathbf{g}_9$'. For later reference we define all three here in terms of the existing six:

$$\mathbf{g}_7 = \mathbf{g}_3 + \mathbf{g}_6, \tag{4.4a}$$

$$\mathbf{g}_8 = \mathbf{g}_1 + \mathbf{g}_4, \tag{4.4b}$$

$$\mathbf{g}_9 = \mathbf{g}_2 + \mathbf{g}_5, \tag{4.4c}$$

and note that the transition rules are now invariant to synchronized cyclic permutations of the three triples, $\{\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3\}$, $\{\mathbf{g}_4, \mathbf{g}_5, \mathbf{g}_6\}$ and $\{\mathbf{g}_7, \mathbf{g}_8, \mathbf{g}_9\}$. Figure 6b shows the result of the transition from the hexad shown in Fig. 6a when the generator pair, $\pm\mathbf{g}_1$, is deleted and the pair, $\pm\mathbf{g}_7$, is added.

TABLE 2. Set of all the representative transition rules $R(1)$ and $R(4)$ for schemes invariant to 3-cyclic label rotations and that produce labelings invariant to transitions through 2-circuits. Bullets denote those that are also 3-circuit and 6-circuit invariant.

Index	**R(1)**	**R(4)**	**3-circuit consistent**	**6-circuit consistent**
1	+3, +4, −6; +5, +7, −2	−5, +3, +7; +6, −2, +1	•	•
2	+4, −6, +3; +7, −2, +5	+7, −5, +3; +1, +6, −2	•	
3	−6, +3, +4; −2, +5, +7	+3, +7, −5; −2, +1, +6	•	
4	−3, +6, −4; +5, −2, +7	−2, −7, −6; +3, −1, +5	•	
5	+6, −4, −3; −2, +7, +5	−6, −2, −7; +5, +3, −1	•	
6	−4, −3, +6; +7, +5, −2	−7, −6, −2; −1, +5, +3	•	
7	+5, −4, +2; +3, −7, +6	+2, +6, +7; +3, +5, −1	•	
8	−4, +2, +5; −7, +6, +3	+7, +2, +6; −1, +3, +5	•	
9	+2, +5, −4; +6, +3, −7	+6, +7, +2; +5, −1, +3	•	•
10	−5, −2, +4; +3, +6, −7	+5, −7, −3; +6, +1, −2	•	
11	−2, +4, −5; +6, −7, +3	−3, +5, −7; −2, +6, +1	•	
12	+4, −5, −2; −7, +3, +6	−7, −3, +5; +1, −2, +6	•	
13	+7, +3, +2; −4, −5, −6	−1, −2, −5; +7, −3, −6		
14	+7, +3, +2; −4, −5, −6	−1, +6, −3; −7, −5, +2		
15	−7, −2, −3; −4, −6, −5	−1, −2, −5; +7, −3, −6		
16	−7, −2, −3; −4, −6, −5	−1, +6, −3; −7, −5, +2		

A glimpse of the topology of the network of connections corresponding to adjacency of hexads reveals that there exist nontrivial 3-circuits, 6-circuits, and circuits of every larger size. For example, the two alternative hexads obtained by the algorithm when $\mathbf{g}_1$ is replaced, or when $\mathbf{g}_4$ is replaced, are themselves mutual neighbors, therefore forming a 3-circuit with the original hexad. Our set of 16 transition rules of Table 2 is further narrowed down to only the first 12 of this list when we impose the restriction of label invariance with respect to excursions around the 3-circuits, as indicated in the table. When we take excursions around the 6-circuits, we are left with only two possible contenders. It turns out that these two schemes, indexed in the table as schemes '1' and '9', do indeed possess the universal label invariance that we seek, but we shall need more powerful geometrical and algebraic techniques to prove this. The

formal mathematical tools we need include the abstract algebraic structures called **Galois fields** (Dickson 1958). Since these will also figure prominently in the development of efficient ways to schedule the various line-filters that both the triad and hexad methods imply, it is worth devoting a subsection to a description of the fundamentals of this algebra, and to the properties that pertain to the hexad algorithm. This will set the stage for the proof that our two winning transition rules do indeed lead to unconditional label invariance.

(*b*) *Galois fields*

A Galois field possesses p^n elements where p is prime and n is a natural number. Addition and commutative multiplication are defined and the usual distributive and associative rules of algebra hold. For *addition*, the elements form an n-dimensional vector space over integers modulo-p, while the $p^n - 1$ non-null elements can be shown always to be capable of being put into an arrangement that conforms to the cyclic group of this order under *multiplication*. The **prime field**, $GF(p)$, is just the field of integers $\{0, \ldots, p-1\}$ where addition and multiplication are carried out modulo-p. The elements of a Galois **extension field** $GF(p^n)$ can be thought of as the field of the equivalence classes of the polynomials having the elements of the associated prime field, $GF(p)$ [which is referred to as the so-called **base field** of this $GF(p^n)$], as coefficients, and where the resulting polynomials are taken as equivalent, modulo some appropriate **primitive polynomial** of degree n. Thus, by the prescribed equivalence, every polynomial can be reduced to an equivalent one of degree not exceeding $n-1$ and having coefficients not exceeding $p-1$, giving the total of p^n possible nonequivalent elements of the resulting extension field. All the constructions of Galois fields of a given order according to this prescription turn out to be mutually isomorphic – that is, the Galois field of a given order is essentially unique up to isomorphisms. It is always possible to find a **primitive element** of the field that generates **all** nonzero elements by taking its successive powers – hence the identification with the cyclic group structure under multiplication. For our purposes, it is the Galois fields whose n is the dimensionality of the lattice, Z^n, that are relevant. For example, consider the following construction of $GF(8) \equiv GF(2^3)$ using the primitive polynomial,

$$P(\lambda) = 1 + \lambda + \lambda^3,$$

of degree $n = 3$ and the primitive element,

$$q(\lambda) = \lambda,$$

multiplying under modulo-p ($=2$) rules. We shall denote the null element, c_0, and generate the 7-cycle group (under multiplication) of remaining elements, $c_1, \ldots, c_7$. Starting with $c_1 = 1$, these remaining elements may be produced by repeated multiplication by the primitive element,

$c_2 \equiv q(\lambda)$. The complete list of the elements in this representation of $GF(2^3)$ is therefore:

$$\left.\begin{array}{rcl} c_0 &=& 0, \\ c_1 &=& 1, \\ c_2 &=& \lambda, \\ c_3 &=& \lambda^2, \\ c_4 &=& \lambda^3 \equiv 1 + \lambda, \\ c_5 &=& \lambda + \lambda^2, \\ c_6 &=& \lambda^2 + \lambda^3 \equiv 1 + \lambda + \lambda^2, \\ c_7 &=& \lambda + \lambda^2 + \lambda^3 \equiv 1 + \lambda^2. \end{array}\right\} \tag{4.5}$$

The relevant ordered-triples of coefficients that define this representation are conveniently listed in Table 3.

TABLE 3. A REPRESENTATION OF THE ELEMENTS OF THE GALOIS FIELD, $GF(2^3)$

ELEMENT OF GF(8)	INTEGER-VECTOR COEFFICIENTS		
c_0	0	0	0
c_1	1	0	0
c_2	0	1	0
c_3	0	0	1
c_4	1	1	0
c_5	0	1	1
c_6	1	1	1
c_7	1	0	1

In order to relate the geometrical properties of lattices to Galois fields, consider the regular n-dimensional lattice of integer displacement vectors $\mathbf{x}$ generated by a basis set $\mathbf{B}$ multiplying an integer n-vector, $\mathbf{v} \in \mathrm{Z}^n$:

$$\mathbf{x} = \mathbf{B}\mathbf{v}. \tag{4.6}$$

Then we can identify each lattice location $\mathbf{x}$ with an element c_i of $GF(p^n)$ whenever the ordered coordinates $\mathbf{v}$ are congruent, modulo-p, with the corresponding coefficients of the polynomial representation of this c_i. When the components of a non-vanishing $\mathbf{v}$ have no common factor, this induced mapping is always to a non-null member of the Galois field. We can therefore think of the nonzero elements of $GF(p^n)$ as a way to 'color', with a palette of $p^n - 1$, the *nodes* of the lattice that are *visible* (that is, not obscured by intervening lattice nodes) from the origin. However, for $p > 2$ we find that each *ray* of the lattice would associate with $(p-1)$ of the colors, namely, a subset C_i of the non-null elements of $GF(p^n)$ of the form $\{c_r^k c_i\}, \quad k = 0, \ldots p-2$ where c_r is one of the elements that satisfy $c_r^{p-1} = c_1$ (the root of the multiplicative *unit* element of the field). Thus, the colors associated with rays are more consistently identified with the various sets C_i rather than the elements c_i themselves. In this way, there are $(p^n - 1)/(p-1)$ colors in the palette for rays associated with the scheme using $GF(p^n)$. (This convention assumes the colors attributed to rays are invariant to a reversal of direction; if we wish to color

oriented rays and distinguish between the two orientations of a line, then we may double the palette of colors when $p > 2$). It should be no surprise that the coloring of the two-dimensional lattice of generators and the associated triads of Fig. 5 can now be seen as an application of this Galois field identification for the case of $GF(4)$, implying three colors. A 13-color identification is of considerable importance to the efficient construction of the blended hexads algorithm, as we shall discover. Table 4 contains one possible representation of this field.

TABLE 4. A REPRESENTATION OF ELEMENTS OF $GF(3^3)$ AS TRIPLES OF THE SUCCESSIVE COEFFICIENTS OF POLYNOMIALS WHOSE NONZERO INSTANCES ARE GENERATED USING MODULO-3 ARITHMETIC WITH A PRIMITIVE ELEMENT, λ, AND A PRIMITIVE POLYNOMIAL, $1 + 2\lambda + \lambda^3$. 13 'COLORS', C_i, ARE ASSOCIATED WITH ROWS OF THE TABLE CONTAINING THE NON-NULL ELEMENTS.

COLOR	ELEMENT OF GF(27)	INTEGER-VECTOR COEFFICIENTS	
	c_0	0 0 0	
C_1	$c_1,\ c_{14}$	1 0 0	2 0 0
C_2	$c_2,\ c_{15}$	0 1 0	0 2 0
C_3	$c_3,\ c_{16}$	0 0 1	0 0 2
C_4	$c_4,\ c_{17}$	2 1 0	1 2 0
C_5	$c_5,\ c_{18}$	0 2 1	0 1 2
C_6	$c_6,\ c_{19}$	2 1 2	1 2 1
C_7	$c_7,\ c_{20}$	1 1 1	2 2 2
C_8	$c_8,\ c_{21}$	2 2 1	1 1 2
C_9	$c_9,\ c_{22}$	2 0 2	1 0 1
C_{10}	$c_{10},\ c_{23}$	1 1 0	2 2 0
C_{11}	$c_{11},\ c_{24}$	0 1 1	0 2 2
C_{12}	$c_{12},\ c_{25}$	2 1 1	1 2 2
C_{13}	$c_{13},\ c_{26}$	2 0 1	1 0 2

Although the matrix **B** of basis vectors in (4.6) is not the only one we can choose to define the lattice, and alternatives lead to different mappings between the Galois field elements and the lattice points or lines, one property that does *not* change with the different choices of **B** is the classification of the subsets of the elements of the Galois field that correspond to the lattice *planes* through the origin. In our example of the representation $GF(2^3)$, we find that, in addition to c_0, the lattice planes through the origin involve the following seven triples: (c_1, c_2, c_4), (c_2, c_3, c_5), (c_3, c_4, c_6), (c_4, c_5, c_7), (c_5, c_6, c_1), (c_6, c_7, c_2), (c_7, c_1, c_3). This incidence pattern is nicely encapsulated in the diagram of Fig. 7a, where the numbers are the indices of the Galois elements. This may be interpreted as the specification of the **Fano plane** (Fano 1892, Coxeter 1968), the simplest example of the class of **finite projective planes**, where the numbered elements are the *points* and the connecting straight lines and the circle identify the seven *lines* of this geometry. Each pair of lines intersect at a single point and each pair of points defines a single line in a projective plane. Similar constructions of finite projective planes result from the incidence relations among lattice planes and lines when the coloring of lattice points derives from the mapping from the three-dimensional lattice to the elements of any other $GF(p^3)$ [with $(1 + p + p^2)$ colors] while the equal number of inequivalent lattice planes acquire subsets of $(1 + p)$ colors in accordance with representations of $GF(p^2)$ associated with each of

these planes. In each case, the palette of $(1+p+p^2)$ colors forms a cycle, the *points* of the cyclic projective plane, within which the incidence pattern defining the *lines* is simply the same basic stencil rotated coherently through all $(1+p+p^2)$ of the possible positions. For the seven-color example with $p=2$ the stencil is *always* in the shape (or its reverse) shown in Fig. 7b where it is picked out for the case of the *line* of points indexed, 1, 2, 4. The relative locations, $(0, 1, 3)$, define a **perfect difference set** since, modulo-7, the index differences produce all natural numbers up to 6, each in a unique way. The other cyclic projective planes of this general family also produce perfect difference sets whose indices $d_0, \ldots, d_p$ yield differences, modulo-$(1+p+p^2)$, of all natural numbers up to $p+p^2$. However, the case $p=2$ seems to be unique in allowing only one stencil pattern (within reversal). This feature helps to show that the two forms of the basic algorithm, represented by rows 1 and 9 of Table 2 are indeed solutions satisfying the symmetry requirements we mandated above. The cyclic properties of finite projective geometries have been extensively studied, notably by Singer (1938). The practical importance of this subject has increased enormously in recent times owing to the development of applications in efficient error-correcting codes for electronic communication and storage. A highly technical mathematical survey of the subject is the book by Hirschfeld (1998).

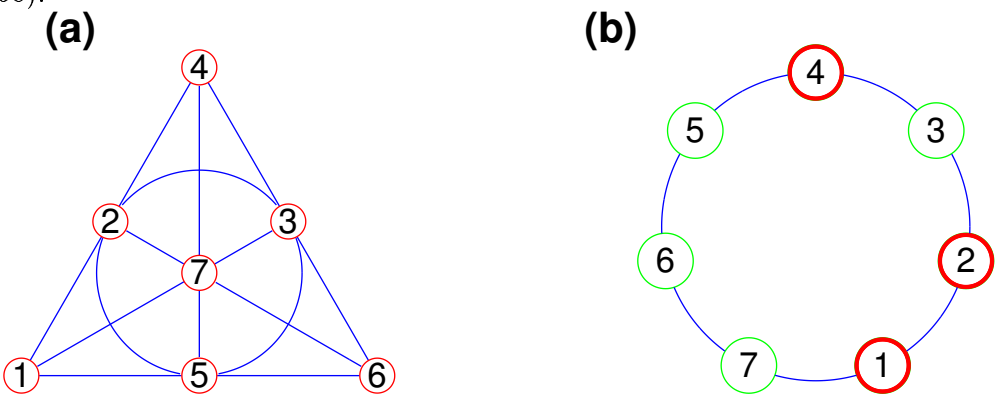

Figure 7. (a) The so-called 'Fano plane' finite projective geometry of seven '*points*' and '*lines*'. (b) The cyclic pattern formed by the points of the Fano plane and the generic incidence pattern of a line (for the case exemplified by line comprising points, 1, 2 and 4) marked by the bold circles. The other six lines are picked out by the cyclic rotation of this stencil to other positions of the cycle of points.

(*c*) *The two basic hexad formulations favored by symmetries*

From the seven colors associated with the $GF(2^3)$ lattice mapping, each hexad possesses six of them. Hence, we can unambiguously associate the missing seventh color with the hexad itself. In an iterative step of the hexad algorithm a generator of one color is replaced by a new generator that becomes the same color as that of the original hexad, while the color of the new hexad is correspondingly the color of the generator replaced. These facts suggest the following way of ensuring that the same alignment of any given hexad's indices is arrived at every time the search algorithm passes through this particular hexad. For any hexad of a given color, seek to establish a *standard* orientation of it such that the six remaining colors are assigned to the index tableau slots of (4.3) in the *same* pattern every time. The chosen pattern of colors

leads to no ambiguity in the signs of the generators represented in the standard index tableau because the chirality condition (4.1) is consistent with only *one* sign of each of the six tableau entries.

The two distinguished entries of Table 2 each have the property that they will place each one of the seven colors just once in each position of the tableau when we consider some initial starting hexad together with the six replacement hexads one transition step away from it. This suggests that we seek a cyclic color assignment such that a transition step from any one hexad to any neighboring one is associated with a *multiplication* of all the initial tableau's implied $GF(2^3)$ elements (or 'colors') by one of the nonzero and non-unit elements of $GF(2^3)$, this latter element being determined according to which slot in the initial tableau the replaced generator for this particular transition step is to be found. For this to work, the cyclic pattern of incidence relations between the points and lines of the Fano plane as depicted in Fig. 7b must also appear in all those subsets of the cyclic elements ('colors') that define the seven inequivalent planes through the origin of our 3D lattice. Four of these planes through the origin always contain generator triplets, namely the triplets $\{\mathbf{g}_1, \mathbf{g}_5, \mathbf{g}_3\}$, $\{\mathbf{g}_2, \mathbf{g}_6, \mathbf{g}_1\}$, $\{\mathbf{g}_3, \mathbf{g}_4, \mathbf{g}_2\}$ and $\{\mathbf{g}_4, \mathbf{g}_5, \mathbf{g}_6\}$ of each given hexad, in its standard tableau, (4.3). The remaining three inequivalent planes through the origin contain only the generator pairs, $\{\mathbf{g}_1, \mathbf{g}_4\}$, $\{\mathbf{g}_2, \mathbf{g}_5\}$ and $\{\mathbf{g}_3, \mathbf{g}_6\}$. The same coloring assignment cannot fit both schemes and still enjoy these cyclic properties, but each can be mapped separately in this desirable way. Then the color patterns of the seven possible versions of the standard tableau for each of the two schemes can indeed be derived by Galois field *multiplication* of the the elements of any one of the seven tableaux by the appropriate nonzero c_i.

Let us rename our two realizations of the hexad algorithm: **Scheme A** (row 1 of Table 2) and **Scheme B** (row 9 of Table 2). Scheme A's transition rules for the six generator replacement options are listed below, where, for brevity, only the sign of each '$\mathbf{g}$' and its index are tabulated, and the particular generator of the original set $\{\mathbf{g}_i\}$ of (4.3) replaced is indicated by the notation '$R(i) \equiv$' prefixing each corresponding replacement tableau:

$$R(1) \equiv \begin{bmatrix} +3 & +5 \\ +4 & +7 \\ -6 & -2 \end{bmatrix}, \quad R(2) \equiv \begin{bmatrix} -4 & -3 \\ +1 & +6 \\ +5 & +8 \end{bmatrix}, \quad R(3) \equiv \begin{bmatrix} +6 & +9 \\ -5 & -1 \\ +2 & +4 \end{bmatrix}, \tag{4.7a}$$

$$R(4) \equiv \begin{bmatrix} -5 & +6 \\ +3 & -2 \\ +7 & +1 \end{bmatrix}, \quad R(5) \equiv \begin{bmatrix} +8 & +2 \\ -6 & +4 \\ +1 & -3 \end{bmatrix}, \quad R(6) \equiv \begin{bmatrix} +2 & -1 \\ +9 & +3 \\ -4 & +5 \end{bmatrix}. \tag{4.7b}$$

This is consistent with a representation of the Galois field mapping that assigns field elements (and hence, colors) according to the identifications:

$$\left.\begin{aligned} \pm\mathbf{g}_6 &\mapsto c_1, \\ \pm\mathbf{g}_5 &\mapsto c_2, \\ \pm\mathbf{g}_3 &\mapsto c_3, \\ \pm\mathbf{g}_4 &\mapsto c_4, \\ \pm\mathbf{g}_1 &\mapsto c_5, \\ \pm\mathbf{g}_2 &\mapsto c_6, \\ \pm\{\mathbf{g}_7, \mathbf{g}_8, \mathbf{g}_9\} &\mapsto c_7, \end{aligned}\right\} \tag{4.8}$$

whereupon the standard tableau (4.3), whose hexad color is c_7, is endowed with the colors:

$$\begin{bmatrix} c_5 & c_4 \\ c_6 & c_2 \\ c_3 & c_1 \end{bmatrix}, \tag{4.9}$$

and, of course, the colors of the substitution tableaux (4.7a) and (4.7b) are just the appropriate modulo-7 rotations of these.

The alternative scheme, Scheme B, is:

$$R(1) \equiv \begin{bmatrix} +2 & +6 \\ +5 & +3 \\ -4 & -7 \end{bmatrix}, \quad R(2) \equiv \begin{bmatrix} -5 & -8 \\ +3 & +4 \\ +6 & +1 \end{bmatrix}, \quad R(3) \equiv \begin{bmatrix} +4 & +2 \\ -6 & -9 \\ +1 & +5 \end{bmatrix}, \tag{4.10a}$$

$$R(4) \equiv \begin{bmatrix} +6 & +5 \\ +7 & -1 \\ +2 & +3 \end{bmatrix}, \quad R(5) \equiv \begin{bmatrix} +3 & +1 \\ +4 & +6 \\ +8 & -2 \end{bmatrix}, \quad R(6) \equiv \begin{bmatrix} +9 & -3 \\ +1 & +2 \\ +5 & +4 \end{bmatrix}. \tag{4.10b}$$

This is consistent with an assignment of colors:

$$\left.\begin{array}{rcl} \pm\mathbf{g}_4 & \mapsto & c_1, \\ \pm\mathbf{g}_5 & \mapsto & c_2, \\ \pm\mathbf{g}_1 & \mapsto & c_3, \\ \pm\mathbf{g}_6 & \mapsto & c_4, \\ \pm\mathbf{g}_3 & \mapsto & c_5, \\ \pm\mathbf{g}_2 & \mapsto & c_6, \\ \pm\{\mathbf{g}_7, \mathbf{g}_8, \mathbf{g}_9\} & \mapsto & c_7, \end{array}\right\} \tag{4.11}$$

whereupon the standard tableau (4.3) has the colors:

$$\begin{bmatrix} c_3 & c_1 \\ c_6 & c_2 \\ c_5 & c_4 \end{bmatrix}, \tag{4.12}$$

while the colors of the entries in (4.10a) and (4.10b) are all modulo-7 rotations of this one.

A close examination reveals that the schemes A and B are not actually completely independent, but chiral reflections of each other. To recapitulate, we have noted that, owing to the chirality convention, a given hexad can only be oriented, or labeled, in one way (at most) to put the colors of its six generators into correspondence with a preset pattern of colors assigned to the tableau locations of (4.3). We have also shown that there exist only two successful contenders for transition rules possessing both 3-cyclic symmetry and label-invariance for 2-circuits, 3-circuits and 6-circuits through the network of contiguous hexads. Finally, we have shown that, for both these selected schemes, A and B, a $GF(8)$-based coloring scheme can be made consistent with the idea that each possible transition from one hexad to its neighbor is associated with a 7-cyclic permutation of the colors that occur in the tableau together with that complementary color attributed to the hexad itself. Taking these properties together, we may

therefore surmise that, regardless of the path of transition steps taken to go from one initial hexad to *any* other (however distant from the initial one), and provided that either scheme A or scheme B is consistently used throughout the intermediate steps, then the hexad labels at the destination hexad will be fixed and independent of the intermediate path taken. Thus, of the 3-cyclic transition schemes, A and B, and only these schemes, have the universal labeling invariance property we sought.

5. Constructing the Blended Triads scheme

A practical defect of the *basic* triad and hexad algorithms is that, when the aspect tensor changes smoothly across the domain and the triads or hexads consequently change, there is a small but noticeable roughness that appears in the filtered fields at the interfaces between adjacent triads or hexads. The problem occurs because the triad weight approaches zero at the interface at too fast a rate for the filters to keep up with. Fig. 8a shows an example of a transect passing straight through the aspect cone, as it is seen in the Klein mapping. The two end points of the transect are the aspect tensors, $\mathbf{e}_1\mathbf{e}_1^T$ and $\mathbf{e}_2\mathbf{e}_2^T$, where

$$\begin{aligned} \mathbf{e}_1^T &= \left[\cos(-45^\circ), \sin(-45^\circ)\right], \\ \mathbf{e}_2^T &= \left[\cos(30^\circ), \sin(30^\circ)\right]. \end{aligned}$$

Fig. 8b shows a graph of the weights returned by the basic triad scheme along this transect. Note that, since only three weights are ever simultaneously non-zero, the three-color assignment will suffice to schedule filtering operations that prevents conflicts. However, the weights, while continuous, are clearly not smooth functions of position along the transect. Since each weight is proportional to the *square* of the corresponding smoothing scale, we may anticipate that a problem will arise where the weights go to zero at a rate that is linear; it implies that, at some point, the smoothing range associated with the orientation corresponding to the weight that goes to zero must exceed the distance to the locus of transition where *no* smoothing along this orientation should occur. Before discussing the remedy it is instructive to illustrate the actual *effects* of the unsmoothed basic triad method in an idealized model of an inhomogeneous covariance.

These effects are illustrated in Fig. 9 where the aspect tensor in two horizontal dimensions projects onto one triad on the left of the figure, but projects onto another on the right. An array of nine sample source point impulses are smoothed, in this illustrative example, using the basic triad algorithm to supply the three respective weights at every point. In this case, to make the defects of the filtering more obvious we have *not* symmetrized the sequence of line smoothers (as one would do in practice for a real covariance synthesis), and we have used explicit Gaussian filters (instead of recursive, or simulated diffusion filters) along each line, which also serves to expose the defects of the synthesis.

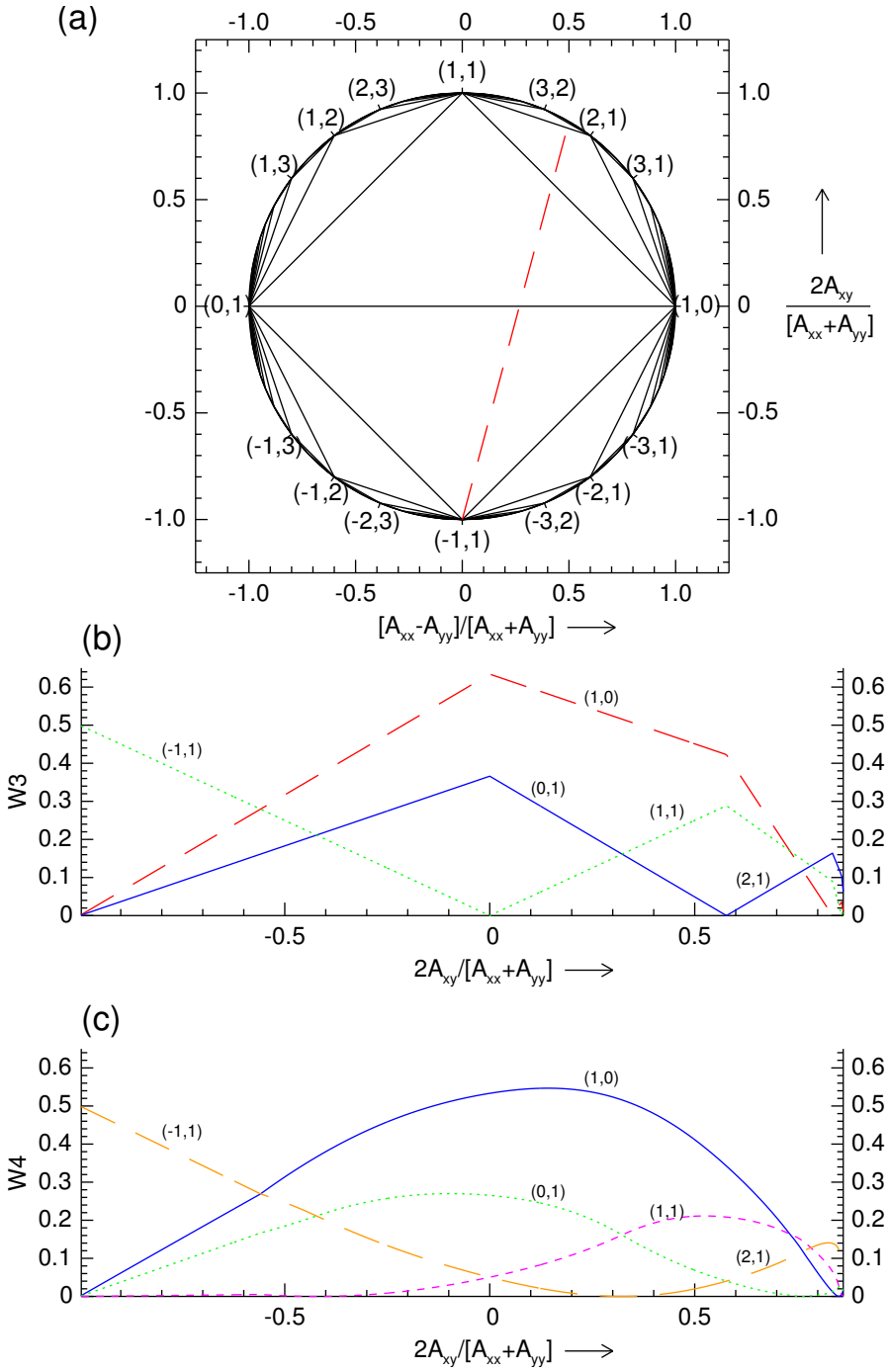

Figure 8. (a) Klein map of triads with transect (dashed line). (b) Color coded basic triad weights (W3) on this transect. (c) Weights (W4) from the blended triads algorithm.

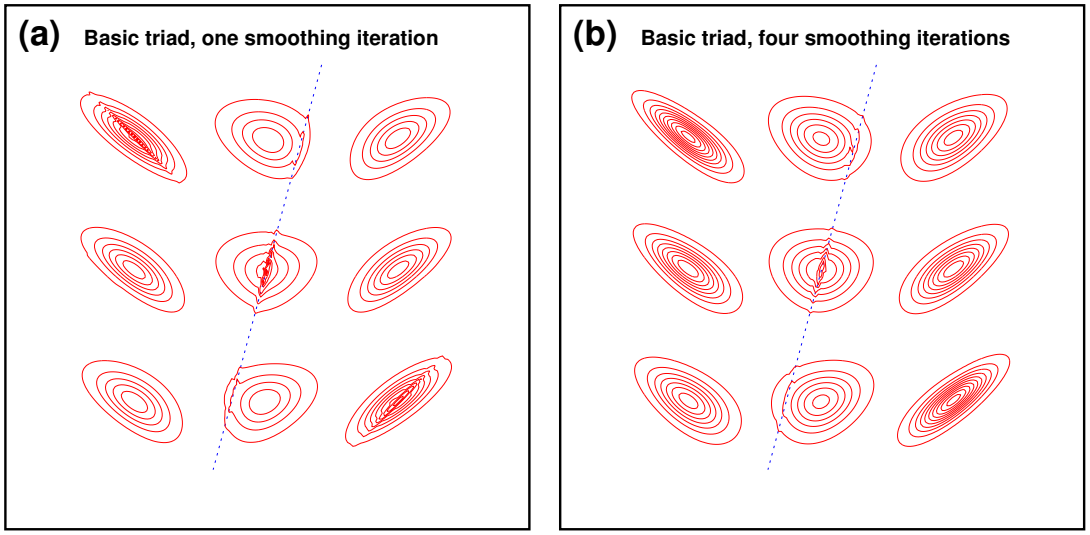

Figure 9. A smooth change of aspect tensor components can lead to a discrete change in the triad, which is shown here by the line of demarcation running from the upper right to the lower left in each panel. The aspect tensors are taken along a transect at $a_1 = A_1/A_3 = 0.2$, with a smooth tanh-profile symmetrical about the point of transition at $a_2 = A2/A3 = 0$. (a) Although the weights corresponding to the triad lines that are being replaced go continuously to zero at the transition interface, they cannot go to zero fast enough in the basic triad algorithm to prevent the unsightly numerical artifacts, 'dislocations', that we see here at the spatial locus of this interface. In this example a single application of the unsymmetrized smoother is applied. Panel (b) shows that, even when replacing the single iteration with four correspondingly weaker applications of the filter that are calculated to possess the same overall aspect tensors, the dislocations are still present, and with amplitudes only slightly diminished.

Fig. 9a shows the result of using the basic triad method with one iteration of each line smoother. The line of transition that separates the regions associated with the two active triads is shown as the dotted line. The aspect tensor components change smoothly across the zone of transition straddling this line, with a profile proportional to the 'tanh' function, and this profile remains constant along lines parallel to the transition line. In this case the changing aspect tensor corresponds to a transect in the aspect cone at $a_1 = 0.2$, where a_1 is the 'horizontal' coordinate, A_1/A_3 of Fig. 3a. The physical domain inside the square shown is gridded with 200 points on each side. The response function contours clearly show a severe numerical *dislocation* wherever the contours intersect the line of transition. Figure 9b shows that replacing the single iteration with four (each with a smoothing scale one half of that used previously, so as to preserve the theoretical overall dispersion) only serves to reduce the amplitude of the spurious disturbance somewhat, but comes nowhere close to eliminating it.

If we want to be sure of a smooth transition it is necessary at least to constrain the way the weights go to zero so that their graphs make *tangential contact* with zero, rather than cutting the zero axis abruptly. (Of course, this not a sufficient condition since, even with the smoothest of tangential contacts, it will always be possible to engineer continuous changes of the aspect tensor with distance that are simply too large to be adequately recovered by a synthesis based on a handful of line smoothers.) This necessary condition in turn implies that we must consider a generalization of the triad and hexad procedures so that, in the vicinity of a transition point, we are effectively blending weights as if from a symmetrically dispersed distribution of aspect

space points rather than taking the weights as if from the single aspect point. By dispersing the aspect point to a distribution of finite radius in aspect space it is, in principle, possible to prevent the weights from going steeply to zero. This is because, in the vicinity of a transition, the blending distribution surrounding the target aspect point can be made to overlap the triads or hexads on either side of any transition interface in aspect space and the finite radius of this averaging distribution then effectively smooths out the filtering weights without altering the implied reconstructed aspect tensor. The implied price is the need at some points to apply more than the minimum of *three* line filters that the basic triad is automatically limited to. But, with careful control of the range of averaging implied by the aspect-space smoothing kernel, we can still keep the number of line smoothers at each point restricted to no more than four.

In the case of the triad algorithm, where the junctions between adjacent tiles that cover the standardized aspect space are relatively simple interfaces involving only two adjacent triads at a time, this averaging can be achieved implicitly by the following purely geometrical procedure. We find the triad to which the aspect point belongs (by the regular basic triad algorithm), rotating the labels to associate the third direction with the smallest weight. This implies geometrically that the aspect point is closer to the edge joining the first two generator images of the triad than to either of the other two edges. We perform the transformations like (2.2a)—(2.2c), that convert the aspect tensor into a 3-vector, but in a framework where the three generators of our triad have aspect vectors, $[1/2, 0, 1/2]^T$, $[-1/2, 0, 1/2]^T$ and $[0, 1, 1]^T$. A fourth generator, formed by the difference of the first two, will supply a fourth aspect vector in this framework, $[0, -1, 1]^T$, which completes the nearest neighboring triad to the one in which the target aspect point lies. Next, we centrally-project our transformed aspect vector $\mathbf{A}$ to form its two Klein-map Cartesian coordinates:

$$a_1 = A_1/A_3, \tag{5.1a}$$

$$a_2 = A_2/A_3. \tag{5.1b}$$

Define

$$d = a_2/(2 - a_2), \tag{5.2}$$

and

$$d_L = \frac{1 - |a_1|}{3 + |a_1|}. \tag{5.3}$$

The locus of a_2 formed by $d = d_L$ within this triad and the equivalent locus within its neighbor together form the rhombus shown as the bold dashed lines in the Klein-map of this situation, shown in Fig. 10. This rhombic region occupies the respective thirds of these two tiles nearest to their common edge (in the Klein projection, the proportion looks larger owing to the projection's emphasis on the middle of this map). Note that, for *any* target aspect point in this rhombic region, the quartet of generator images formed by both triads is the nearest such quartet to that chosen point (in a sense that is unambiguous); points outside the rhombic region are closer to *other* quartets of generator images. The next step is to use d and d_L to reintroduce a third coordinate and also to amplify the Klein-map coordinates to place the image of the aspect vector in the interior of the convex hull of the aspect vectors of the four generators in such a way that the locus of this image forms a smooth surface as a function of the original aspect tensor, even as the second component, A_2, of the transformed aspect vector

passes through zero. In effect, we are smoothing the dihedral angle formed by the two adjacent triad tiles with a saddle-shaped surface that spans the rhombic region outlined in Fig. 10. This is accomplished by setting:

$$a_3' = \begin{cases} (2 + d_L + d^2/d_L)/4 & : \quad d < d_L, \\ (1 + d)/2 & : \quad d \geq d_L. \end{cases} \tag{5.4}$$

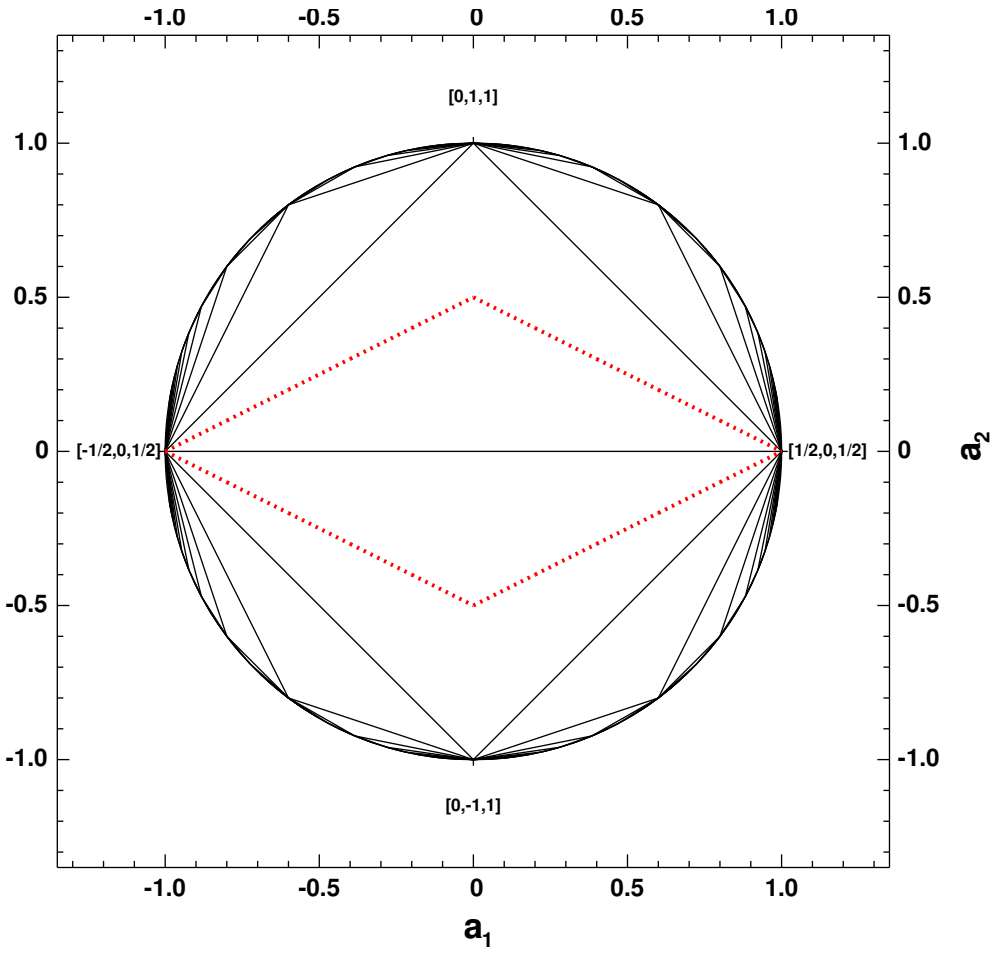

Figure 10. A Klein-map whose focus is a rhombic region straddling a pair of adjacent triads such that: an aspect point mapping to the interior of this rhombus is closer to the common edge (the horizontal diameter of the map) than to any other triad edge. Such points can be assigned weights associated with the *four* generators whose aspect images form the inscribed square of the Klein-map with the vertices labelled in this case by the three-dimensional coordinates, A_i of their vertex images.

Then a corresponding amplification of what had originally been the rotated Klein-map coordinates to:

$$a_1' = a_1 a_3', \tag{5.5a}$$
$$a_2' = a_2 a_3', \tag{5.5b}$$

gives a new location in the aspect cone whose Klein-map position is unchanged, but which is now positioned on a curved, approximately saddle-shaped surface, that meets the two triad tiles that it partially straddles with tangential contacts at the rhombic boundary.

The four barycentric coordinates of this projected aspect vector relative to the four surrounding generator image aspect vectors associated with the four active generators are easily found from the vector $\mathbf{a}' = [a'_1, a'_2, a'_3]^T$:

$$\mathbf{w}' = \mathbf{w}'_c + \mathsf{W}\mathbf{a}', \tag{5.6}$$

where

$$\mathbf{w}'_c = \left[\begin{array}{cccc} 1, & 1, & -1/2, & -1/2 \end{array} \right]^T, \tag{5.7}$$

and

$$\mathsf{W} = \left[\begin{array}{ccc} 1, & 0, & -1 \\ -1, & 0, & -1 \\ 0, & 1/2, & 1 \\ 0, & -1/2, & 1 \end{array} \right]. \tag{5.8}$$

Finally, we must further amplify the weights according to the ratio of the magnitudes of the original A_3 to the projected a'_3:

$$\mathbf{w} = \mathbf{w}' A_3 / a'_3. \tag{5.9}$$

The direct result of this effective blending of two adjacent triads can be seen in the graphs of the weights, of which there are now up to four that are simulataneously non-vanishing, plotted along the same transect illustrated in Fig. 8a. Figure 8c shows the new weights that result from the same transect and provides a visual confirmation that, in the interior, the new weights that go to zero do so tangentially. We note that, strictly speaking, the saddle-shaped function we have used to bridge adjacent triads is still not free of discontinuities in derivatives of weights away from zero since such a discontinuity exists where $a_1 = 0$. This is the part of one of the symmetry axes of a triad, marking the locus where the two dominant weights from either the basic or blended triad algorithms become equal. The subtle effect of this lack of smoothness is visible as a very slight kink in the graphs of the weights assigned to generators $(-1, 1)$ and $(1, 0)$ of Fig. 8c where they intersect. However, this feature does not appear to translate into discernable artifacts in the fields smoothed in accordance with this blended triads prescription.

Clearly, the 3-color scheduling of line filters discussed in section 3 is no longer an adequate guide to conflict-free line filtering in the blended triads method because, by blending pairs of adjacent triads, we necessarily involve two lines of the *same* color in the resulting quartet of line smoothers at each grid point. It is the coloring associated with the Galois field, $GF(3^2)$ that comes to the rescue and provides us with the appropriate 4-color assigment in this case. Since every triad under this coloring regime possesses three of the four colors, it is natural to confer upon the triad itself the color that its own line generators do *not* possess. An illustration, in the same style as Fig. 5, shows both the color assignment on the generator lattice, Fig. 11a, and the corresponding Poincaré map, consistently colored as described above, in Fig. 11b. A glance will confirm that the quartet of generators belonging to pairs of triads that are adjacent (and therefore blended) has the full complement of four colors, so that a four-stage scheduling of the line operators according to this color assignment guarantees the avoidance of numerical conflicts.

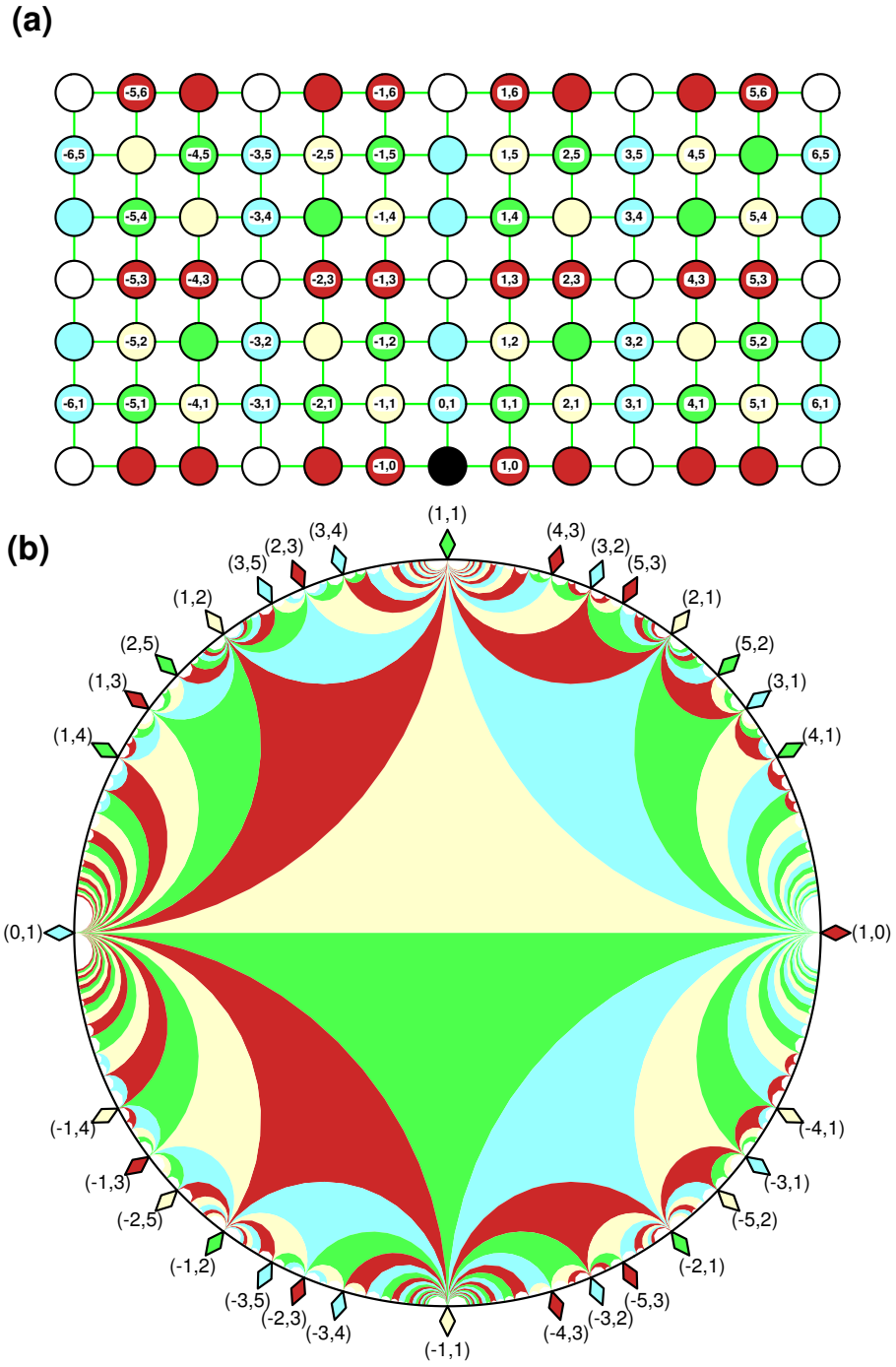

Figure 11. (a) A mapping of the two-dimensional lattice of grid generators to four colors consistent with the Galois field, $GF(9)$. (b) The corresponding Poincaré map, with each triad accorded the color missing from its generators.

Figure 12 shows the same smoothing portrayed in Fig. 9, but performed now according to

the blended triads scheme. Even with only one iteration of smoothing (Fig. 12a), the dislocation corresponding to the transition has been entirely removed and the only remaining distortion of the contours (a slight tremor in the contours on the right side of the middle distribution of the bottom row) is completely cleaned up by the application of four iterations, as shown by Fig. 12b.

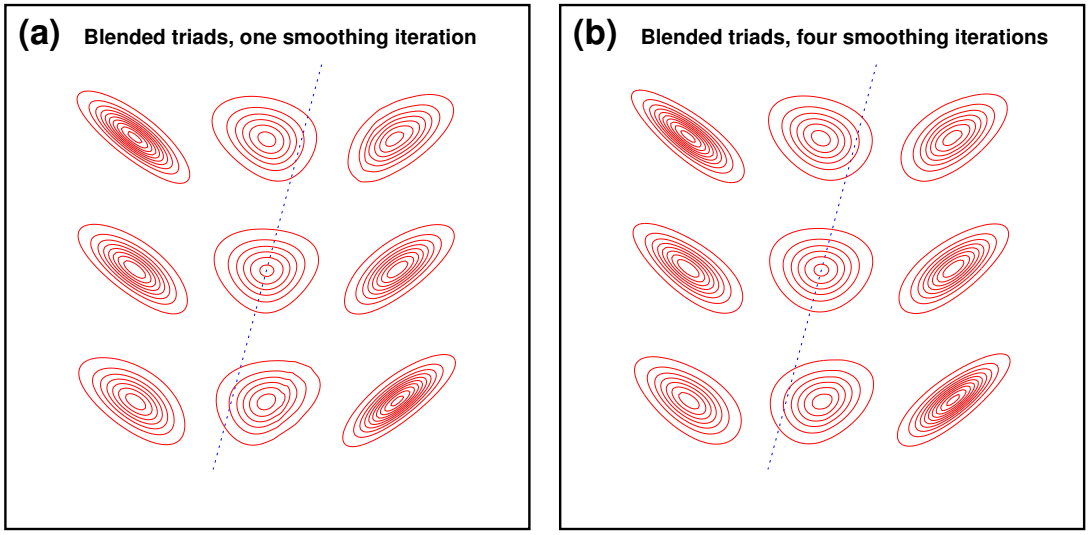

Figure 12. (a) A major improvement in the quality of the response function is achieved by adopting a *blended triads* method, which compares very favorably to Fig. 9a, and shows *no* dislocations on the line of transition itself. (b) Using four iterations of the blended triads method, no visible roughness in the resulting contours occurs anywhere, in contrast to the poor results still seen after four iterations of the basic triad, shown in Fig. 9b.

6. Constructing the Blended Hexads scheme

The method of using projections onto a smooth surface in order to construct the *blended triads* method was feasible because the **junctions** between adjacent triads in the aspect cone are geometrically simple; each junction only involves a pair of triads and is therefore a plane interface of **dimension** two within the full 3D aspect space. The complementary **co-dimension** of the triad junctions is one — the maximum number of triad weights that can vanish simultaneously and still leave an aspect tensor that lies in the proper interior of the aspect cone. In general, in order to reveal and describe the essential geometrical characteristics of a junction it is sufficient to restrict consideration only to a section transverse to that junction and sharing the junction's co-dimensionality. Projections, like the Klein and Poincaré mappings, which leave the important geometrical relationships among the regions separated by junctions intact, reduce the mapped junctions' dimensions, but not their co-dimensions; in this sense, it is always the junction's *co-dimension* that is the intrinsically more revealing index.

Junctions among hexads in the interior of the aspect cone are of higher co-dimensionality than the trivial interfaces that separate the triad tiles. This is because it is possible for up to three, but not more than three, of the weights, w_i, of a hexad to vanish simultaneously without the aspect tensor becoming singular. This implies that the co-dimension of the most singular

generic junction, and therefore the minimal dimension of an affine subspace in which the full complexity of the junction is still retained, is three.

Given this extra complexity, we will not even attempt to solve the problem of blending hexads by a direct projective manipulation analogous to that used to effectively blend triads. Instead, we will construct the vector of smoothing weights by additively combining the basic hexad weights that, for each basic hexad, correspond to a point in a suitably weighted 'cloud', or **averaging kernel**, of such points arranged symmetrically around the target aspect point of interest. The averaging kernel is symmetrically distributed so that integration of the aspect points over the kernel (whose own weight integrates to unity) must produce exactly the same aspect tensor as that of the target point about which the kernel is symmetrically distributed. We must be sure, when applying the averaging kernel, that it does not overlap more than *one* generic junction of co-dimension three, otherwise the number of smoothing directions involved can become very large. This requires us to know both qualitatively and quantitatively enough about the geometry of the **honeycomb** of hexads in aspect space to intelligently judge the appropriate radius of the averaging kernel. We also need to select a radial profile for it that leads to a tractable quadrature. Moreover, it will not normally be feasible to carry out these geometrical investigations and the resulting quadratures for every individual assimilation point's aspect tensor; instead, we must be pragmatic and store the relevant information encoded in suitable tables and provide efficient interpolation software to use these tabulated data effectively. The tabulation problem itself is not trivial since the aspect space is six dimensional. We shall treat this problem in subsection 6b. Meanwhile, we shall describe the hexad honeycomb geometry in sufficient detail to allow us to formulate an adequate kernel-averaging strategy.

(*a*) *Geometrical structure of the honeycomb of hexads*

For a canonical example of such a junction in the case of the unit cubic lattice, Z^3, take $\mathbf{u}_1 = [1, 0, 0]^T$, $\mathbf{u}_2 = [0, 1, 0]^T$, $\mathbf{u}_3 = [0, 0, 1]^T$. All three of these $\mathbf{u}_i$, and their negatives, are found amongst the generators of the hexads whose $\mathbf{g}_1$, $\mathbf{g}_2$, $\mathbf{g}_3$ are of one of the following forms:

$$\{\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3\} = \{s_1\mathbf{u}_1, s_2\mathbf{u}_2, s_3\mathbf{u}_3\}, \tag{6.1}$$

or

$$\{\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3\} = \{s_i\mathbf{u}_i, s_k\mathbf{u}_k, s_i\mathbf{u}_i - s_j\mathbf{u}_j\}, \quad [i, j, k] \in \{[1, 2, 3], [2, 3, 1], [3, 1, 2]\}, \tag{6.2}$$

where, in each expression an odd number of the three *sign* terms, s, are equal to '+' leaving an even number that are '−', in keeping with the chiral convention (4.1). This leads to a total of 16 distinct possible hexads that contain $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$, no other hexads being possible. We note that the 16 hexads of this **cluster** involve all generators belonging to the outer 26 points of the $2 \times 2 \times 2$ lattice cube of points $[v_1, v_2, v_3]^T$ with $v_i \in \{-1, 0, 1\}$, with $|\mathbf{v}| \neq \mathbf{0}$. Since any hexad may be brought into coincidence with any other by an appropriate linear transformation of the lattice into itself, then, in general, a cluster is associated with an origin-centered lattice parallelepiped of the same volume.

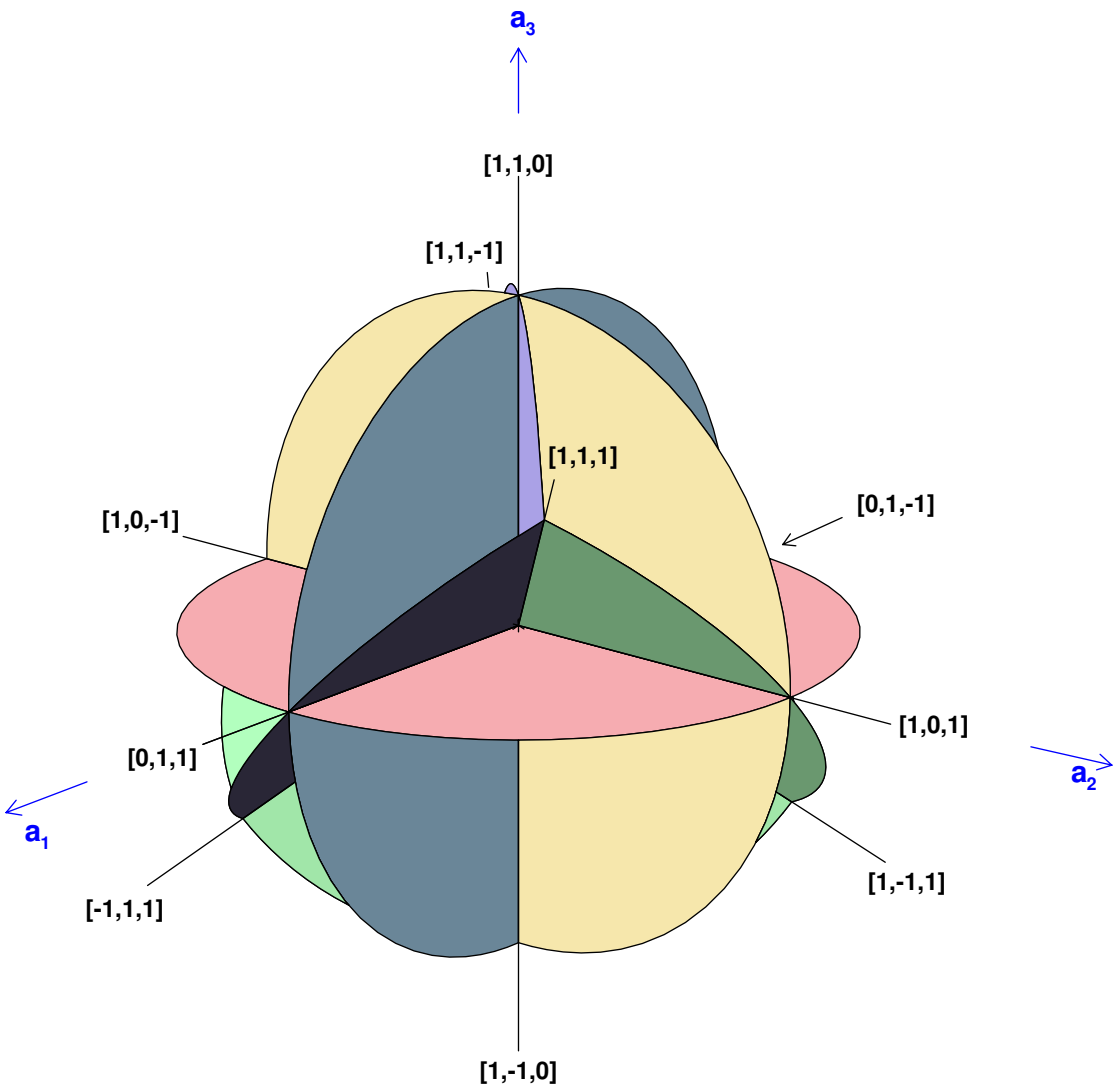

Figure 13. Schematic depiction of the sectors surrounding a typical junction among a cluster of 16 mutually touching hexads. In this example, all hexads include the generators, [1, 0, 0], [0, 1, 0] and [0, 0, 1]. The 10 rays are labeled by additional generators of this cluster. Thus, each hexad of this cluster shares the three generators common to *all* the hexads, together with the three generators of the 10 that specify the *particular* hexad.

On the same unit cubic lattice, the aspect tensor $\mathsf{A} = \mathsf{D}$ lies on the junction of our canonical cluster for D any diagonal matrix. Thus, the junction itself is the intersection of a 3D linear subspace (containing the origin) with the interior of the aspect cone and, using rescaling arguments, we understand the shape of the configuration of hexads that surround this singularity to be qualitatively the same at any point of it that we choose to look at. We choose the particular example, $\mathsf{A}_0 = \mathsf{I}$, and look for neighboring aspect tensors A that explore a maximal 3D affine subspace containing A_0 but oriented transversally to the singularity. The simplest construction is to choose the three local coordinates $\{a_1, a_2, a_3\}$ of this subspace (whose three dimensions correspond to the complementary three co-dimensions characterizing the singularity) to be the

off-diagonal elements of an aspect tensor, A, defined to be:

$$\mathsf{A} = \begin{bmatrix} 1, & a_3, & a_2 \\ a_3, & 1, & a_1 \\ a_2, & a_1, & 1 \end{bmatrix} . \tag{6.3}$$

Note that A reduces to $\mathsf{A}_0 = \mathsf{I}$ when the three components a_i are simultaneously zero. If we divide up this space of A in the neighborhood of A_0 into octants (like the division into eight corner pieces of a radar reflector) separated by the three planes, $a_1 = 0$, $a_2 = 0$, $a_3 = 0$, we find that the hexad (6.1) with all the s equivalent to '+' occupies the nearby portion of the octant, $a_1 < 0$, $a_2 < 0$, $a_3 < 0$; the other three octants obtained by reversing *pairs* of these inequalities contain the other three hexads of (6.1). The remaining four octants are each further partitioned into three narrow corners by the portions of the appropriate three planes of the kind: $|a_2| = |a_3|$; $|a_3| = |a_1|$; $|a_1| = |a_2|$; that join each edge of the octant to its interior axis of three-fold symmetry. The resulting total of 12 new regions contain the 12 hexads defined by the combinations (6.2). An illustration of the geometry of this 3D projection of the generic junction of hexads is sketched in Fig. 13.

The complete set of all possible generators of hexads in a given cluster have been shown to form an origin-centered $2 \times 2 \times 2$ parallelepiped in the lattice of generators. With this identification, we can ask: for a given hexad, what is the *complete* set of clusters which contain this particular central hexad as a member? It turns out that 16 clusters contain any given hexad. Since the given central hexad has its most symmetrical geometrical form in the so called 'face-centered cubic' (fcc) lattice (Ziman 1979), whereupon the opposing pairs of generators of the hexad then form the 12 corners of a geometrically true (undeformed) cuboctahedron, we shall find the most symmetrical geometrical representation of this set of 16 clusters in this fcc lattice.

One convenient numerical representation of the fcc lattice of generators is defined using integer 3-vectors, $[v_1, v_2, v_3]^T$, where $v_1 + v_2 + v_3$ is even. Twelve generators (six opposing pairs) forming the vertices of the central hexad of this lattice are those with all $|v_i| \leq 1$. The parallelepipeds of 12 of the clusters look like the one with corners, in this representation, at $\pm[3, 0, 1]^T$, $\pm[-1, -2, -1]^T$, $\pm[-1, 2, -1]^T$, $\pm[-1, 0, 1]^T$. The other 11 of this dozen are obtained from it by application of some of the 24 proper rotations about the origin which map the lattice into itself. The parallelepipeds of the remaining four clusters look like the one with corners at $\pm[2, 2, 2]^T$, $\pm[-2, 0, 0]^T$, $\pm[0, -2, 0]^T$, $\pm[0, 0, -2]^T$. The other three of this quartet are again obtained from this prototypical example via the rigid rotations which map the lattice into itself. To give the configuration of 16 clusters associated with the central hexad a name, we call it the hexad's **starburst**. The generic starburst configuration involves 37 pairs of opposing generators and, as an aid to appreciating the geometrical relationships involved, the shape formed by the union of the 16 parallelepipeds of the component clusters is sketched as a non-convex polyhedron in Fig. 14. The innermost 12 vertices of this polyhedron (of which only three are visible in this sketch) are the vertices of the central hexad's cuboctahedron.

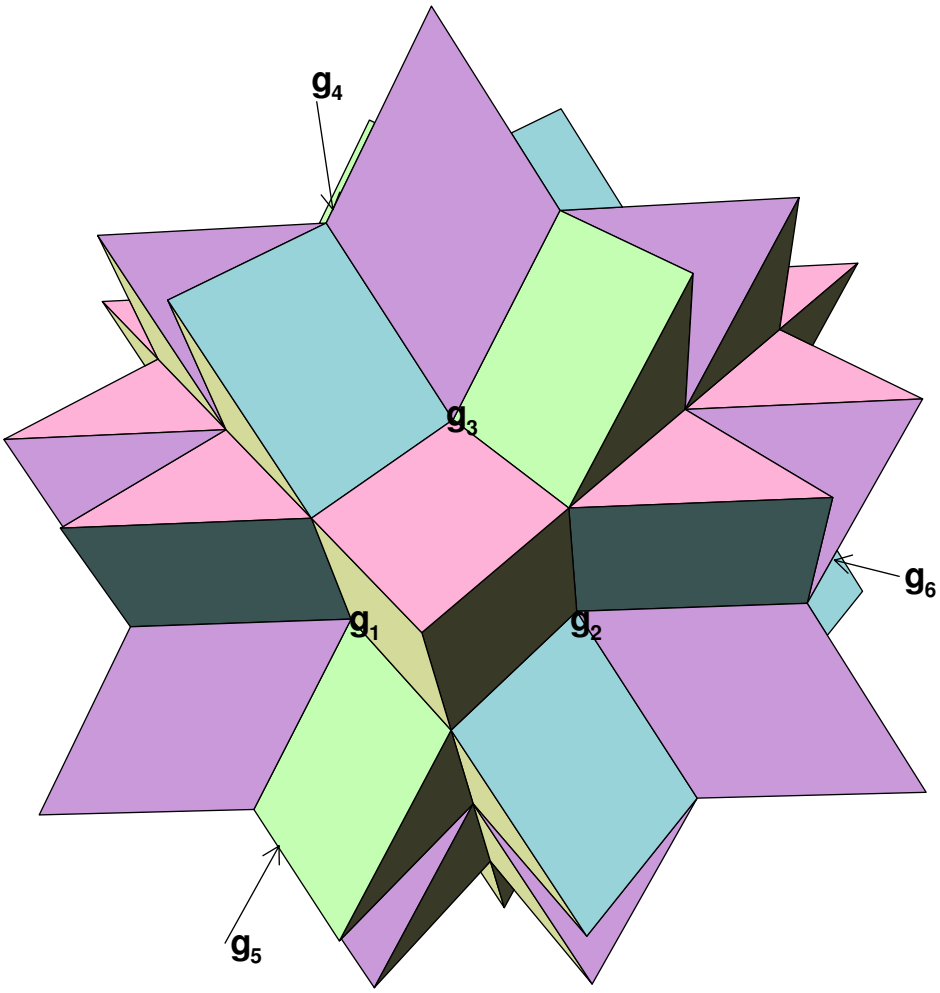

Figure 14. The 'starburst' configuration of clusters and their generators in the case of a face-centered cubic generator lattice. Each cluster is here represented as a particular elongated parallelepiped having two generalized grid spaces along each of its edges and is characterized by its unique pair of outermost vertices. The outermost 32 points of the starburst account for these extreme pairs of characterizing vertices of each of the 16 clusters.

The kernel for the blending operation can have any radius so long as it can be contained within a single cluster. Then it will be possible to color all lattice rays used at this particular point by the 13-color scheme associated with $GF(3^3)$. The larger the kernel radius, the more likely that we are to have a truly smooth covariance operator. However, we cannot use one constant radius everywhere because, as we explicitly saw for the triad case and as also applies to the hexad, these tiles tend to thin exponentially toward their extremes. We therefore seek a non-constant radius, as a function of position in a representative hexad tile, such that a smoothing kernel *just* stays within the bounds of a cluster everywhere. We might also require that this radius is a *smooth* function of position; otherwise we would be risking a fresh source of numerical roughness through the kernel averaging process itself. The question of *which* cluster the kernel must be confined within can be answered by a geometrical analysis that involves consideration

of the relevant starburst whose central hexad contains the particular target aspect point of interest. The **critical radius** at such a target point is found as follows. Having identified (by the basic hexad algorithm) the hexad containing the target, and hence the starburst whose central hexad this is, then for each of the 16 clusters of this starburst, calculate the shortest distance to the boundary of the cluster. Of these 16 shortest distances, choose the largest. This *is* the critical radius. Then the corresponding cluster is nominated to be the one that supplies the 13 line directions that will be used for the line-smoothers at this geographical point. The actual radius of the smoothing kernel must never exceed the local critical radius; it can be a little less, in order to allow this critical radius to be rendered a smooth function of aspect space location, although this additional requirement did not seem to be critical for the success of the blended triads algorithm and, in fact, is not rigorously satisfied by the construction for that case described in section 5.

We have referred to the *critical radius* without reference to the metric by which this distance is measured. Consistency between points in one hexad and points in a neighboring one requires that we adopt a metric that is smoothly defined over aspect space and is invariant to the symmetries that transform one hexad to another and to the group of 24 symmetries we have already identified that transform one hexad into itself. In most respects, the ideal contender is the natural Riemannian metric implied by (2.6), except this choice makes the requisite geometrical calculations exceedingly complicated. A more pragmatic alternative, which we shall refer to as the **tangent metric**, enables all the relevant geometrical manipulations to be carried out in a Euclidean geometry of only five dimensions. If $\mathbf{A}_0$ is the **target** aspect vector then there is a unique 5D **tangent plane** through $\mathbf{A}_0$ which, at this target, is normal to the vector $\mathbf{A}_0$ itself with respect to the natural metric (2.6). This natural Riemannian metric now induces a unique *Euclidean* metric on the tangent plane by virtue of the fact that aspect vectors already form a linear space and, on the tangent plane, only one choice from the range of possible Euclidean metrics that we can confer on this plane is consistent, at $\mathbf{A}_0$, with the restriction of the natural metric (2.6) to this 5D plane. The tangent plane intersects all the hexads in the aspect cone. In fact, the hexads and their associated aspect vectors map to this plane in the higher dimensional equivalent of the *Klein* or gnomonic representation, but having the orientation of $\mathbf{A}_0$ as the central projection axis. The hexads each map to a 5D *simplex* whose 4D boundary elements are themselves *flat* in the induced tangent metric. Each *cluster*, regarded as the union of the simplexes of the 16 constituent hexads, is a non-convex **polytope** (the term generalizing *polyhedron* to arbitrary dimensions) whose boundary can be analyzed in terms of the bounding elements of the 16 constituent hexads. It can be shown that the boundary of the generic cluster in this five-dimensional projection consists of 48 4D (i.e., five-pointed) simplexes. Since the projected locations in the tangent plane of all the vertices are readily determined, it is also a relatively straightforward calculation to find the Euclidean shortest distance to the boundary of a particular cluster that contains $\mathbf{A}_0$. Some details are provided in appendix C. The calculation is repeated for all 16 of the clusters that contain $\mathbf{A}_0$ in order to find the selected cluster that has its boundary farthest away and to record this tangent-metric distance.

(*b*) *Gridding a simplex*

As we have remarked, the distance we obtain from the procedure above gives us an upper bound to the radius of the averaging kernel we are permitted to use. However, this upper bound

is a continuous, but not a smooth function, of the location $\mathbf{A}_0$ in aspect space. In order to choose a practical kernel radius that *is* smooth, then some smoothing and rescaling of this distribution of upper bounds must be performed numerically, which requires that these quantities first be collected in an appropriate regular gridding over a 5D projection of a nonredundant representative portion of a generic hexad. Likewise, to tabulate the subsequently calculated weights attributed to each resulting line smoother, we are obliged to devise a regular gridding and tabulation strategy for such a representative portion of the generic hexad.

The simplest way to grid a simplex in any number of dimensions is to uniformly subdivide the simplex in intersecting families of (hyper)planes parallel to the boundaries of the simplex. A simple inductive argument shows that the number N of grid points resulting from such a decomposition of the simplex in n dimensions subdivided by planes that partition each edge into M segments is:

$$N = \binom{n+M}{M}, \tag{6.4}$$

or, in our example of $n = 5$,

$$N = (M+1)(M+2)(M+3)(M+4)(M+5)/120. \tag{6.5}$$

The size N of this tabulation is clearly very large for any reasonable resolution M (for example, with $M = 40$, $N = 1221759$), but the intrinsic symmetries of the generic hexad can be exploited to reduce the size of the minimal representative portion of this table by a factor that approaches 24 as M increases, provided transformation software is available to expand efficiently from this minimal portion of recorded data back to the full tabulation. This gridding allows the smoothing of kernel-radius upper bounds, if desired, by a numerical simulation of *isotropic* and *homogeneous* diffusion, as these concepts are interpreted within the context of the local tangent metric. Symmetries that allow us to map one hexad into any other also allow us to define the appropriate boundary conditions of our smoothing filter's diffusion operator on the representative hexad. The same gridding of the non-redundant portion of the representative hexad is then used to store the weights that emerge from the blended hexads, but we defer detailed discussion of this tabulation until subsection 6f.

(*c*) *Averaging kernels and Abel transforms*

Having obtained a tabulated field of kernel radii, we must next determine the radial profile of the averaging kernel that will be used in the hexad blending and a numerical procedure by which the necessary quadratures can be carried out. Scaling symmetry allows us to trivially rescale any aspect vector so that, for example, it lies on the plane which contains the six aspect images of the generators of the hexad in which the aspect tensor resides. This scaled aspect vector is therefore expressible as a convex mixture of these generator images. This is a convenient way to reduce the dimensionality of the tabulation from six to five, since the original scaling is easily undone once the standardized blending weights are obtained for the aspect vector normalized in this way. The averaging kernel centered at this normalized aspect point will intersect up to all 16 hexads of the nominated cluster associated with this target aspect point and, in each of the hexad regions of the target point's tangent plane, the fragment of the kernel that occupies each projected hexad has a centroid. Owing to the linearity of aspect tensors, the integrated effect of

the aspect vectors weighted by the averaging kernel within one of the cluster's hexad regions is equivalent to the effect of that centroid, weighted by the amount of the kernel that occupies the hexad regions. The weights, w_i, that we associate with the six generator images of each hexad are linear functions of aspect space position $\mathbf{A}$ throughout the region of aspect space occupied by that hexad. We found at the beginning of section 6 that the co-dimension of the hexad junctions is three, which, with the aforementioned linearity, implies that the actual quadratures needed to identify the centroid of each kernel fragment can be reduced to only the three dimensions of the affine subspace that, with respect to the tangent metric, intersects the junction normally and contains the projected target aspect point $\mathbf{A}_0$. If the kernel only intersects a simpler junction of co-dimension two or one, then the dimensionality of the quadrature can be correspondingly reduced. However, the reduction of quadrature dimensionality is accompanied by a change in the radial profile of the kernel effective in the subspace. An original radial profile, $f_n(r)$ in n dimensions with $r \leq R$ is replaced at a dimension $n-1$ by the corresponding **marginal** profile, $f_{n-1}(r)$, according to the **Abel transform**:

$$f_{n-1}(r) = 2\int_r^R \frac{f_n(q)q}{(q^2-r^2)^{1/2}}\,dq \tag{6.6}$$

which we can iterate to successively lower dimensions when required. Statisticians are familiar with families of distribution profiles for both finite and infinite R for which repeated marginalization is particularly simple. For finite R the simplest family have radial density profiles that relate to the symmetric **beta** distributions (Abramowitz and Stegun 1970) on the interval $[-R, R]$, which we shall now describe.

(*d*) *Kernels derived from symmetric beta distributions*

Consider the radial density profile in n dimensions to have the form:

$$f_{n,a}(r) = T_{n,a}\left(1-\frac{r^2}{R^2}\right)^{a-1}, \tag{6.7}$$

where $T_{n,a}$ is a normalizing constant. In order that this integrates to unity, that is,

$$\int_{\text{domain}} f_{n,a}(|\mathbf{A}-\mathbf{A}_0|)\,dA_1\,\ldots\,dA_n = 1, \tag{6.8}$$

we need

$$\int_0^R f_{n,a}(r)S_n(r)\,dr = 1, \tag{6.9}$$

where

$$S_n(r) = \frac{n\pi^{n/2}r^{n-1}}{(n/2)!}, \tag{6.10}$$

is the surface measure of the sphere of radius r in n dimensions. Thus,

$$\begin{aligned}\int_0^R T_{n,a}\left(1-\frac{r^2}{R^2}\right)^{a-1}\frac{n\pi^{1/2}r^{n-1}}{(n/2)!}dr &\equiv \\ \frac{T_{n,a}nR^n\pi^{n/2}}{2(n/2)!}\int_0^1(1-z)^{a-1}z^{\frac{n}{2}-1}\,dz &= 1.\end{aligned} \tag{6.11}$$

But the integral defines a beta function:

$$\int_0^1 (1-z)^{a-1} z^{\frac{n}{2}-1}\, dz \equiv B(a, n/2) = \frac{(a-1)!(\frac{n}{2}-1)!}{(a+\frac{n}{2}-1)!}. \tag{6.12}$$

Therefore,

$$\begin{aligned} T_{n,a} &= \frac{2(n/2)!}{nR^n\pi^{n/2}} \frac{(a+\frac{n}{2}-1)!}{(a-1)!(\frac{n}{2}-1)!}, \\ &= \frac{1}{R^n\pi^{n/2}} \frac{(a+\frac{n}{2}-1)!}{(a-1)!}. \end{aligned} \tag{6.13}$$

Now consider the radial profile of the Abel transform of $f_{n,a}$.

$$g_{n,a}(r) = 2T_{n,a} \int_r^R \left(1 - \frac{q^2}{R^2}\right)^{a-1} \frac{q}{(q^2-r^2)^{1/2}} dq. \tag{6.14}$$

We solve this integration by the change of variables,

$$w = \frac{q^2-r^2}{R^2-r^2}, \tag{6.15a}$$

$$dw = \frac{2q\, dq}{R^2-r^2}, \tag{6.15b}$$

whereupon,

$$\begin{aligned} g_{n,a}(r) &= T_{n,a} R \left(1 - \frac{r^2}{R^2}\right)^{a-1/2} \int_0^1 (1-w)^{a-1} w^{-1/2}\, dw, \\ &= T_{n,a} R \left(1 - \frac{r^2}{R^2}\right)^{a-1/2} \frac{(a-1)!(-\frac{1}{2})!}{(a-\frac{1}{2})!}, \\ &= \frac{1}{R^{n-1}\pi^{\frac{n-1}{2}}} \frac{(a+\frac{1}{2}+\frac{n-1}{2}-1)!}{(a-\frac{1}{2})!} \left(1 - \frac{r^2}{R^2}\right)^{a-1/2}, \\ &= T_{n-1,a+1/2} \left(1 - \frac{r^2}{R^2}\right)^{a-1/2}, \\ &= f_{n-1,a+1/2}(r). \end{aligned} \tag{6.16}$$

Thus the marginalization to a dimension smaller by one changes the radial profile to the symmetric beta distribution whose shape parameter (a) is incremented by a half.

The application of the *inverse* Abel transform allows us to invert the marginalization and find the radial profile of the kernel in a *higher* dimension that projects down to a chosen member of this beta family in a lower dimension. In this way, we can extend the definition of the symmetric beta function to negative shape parameters a that would not be considered as valid choices in statistical applications, but which continue to have sensible interpretations as **generalized functions** in the present context. Thus, in order for the smoothing to be

qualitatively consistent with the blending carried out for the triad case, we would choose the kernel marginalized to one dimension to have the form of the **rectangle function**, that is,

$$f_{1,1}(r) = \begin{cases} \frac{1}{2R} & : \quad r \leq R, \\ 0 & : \quad r > R. \end{cases} \tag{6.17}$$

Inverting the Abel transform twice, we find that the corresponding kernel profile at the $n = 3$ dimensions at which the numerical quadratures need to be carried out is:

$$f_{3,0}(r) = \frac{1}{4\pi R^2}\delta(r - R), \tag{6.18}$$

where the delta function here describes an impulsive spherical shell. We could apply further inverse Abel transforms to obtain explicit expressions for the higher dimensional equivalent kernels, $f_{4,-1/2}$ and $f_{5,-1}$, which involve more complicated generalized functions, but there is no practical need here to do this §.

(*e*) *Numerical quadrature for kernel averaging*

It actually turns out to be fortuitous that the chosen kernel profile in three dimensions has the form it does since, by having its weight confined to a spherical shell, the averaging reduces to only two-dimensional quadratures. The domain of quadrature required to calculate the centroid and weight of each kernel fragment associated with one of the 16 sectors of the junction is the intersection of the spherical shell of radius R with the three-walled corner-sector that constitutes the three-dimensional projection of the relevant portion of the associated hexad. The 16 sectors of the configuration shown in Fig. 13 provide the generally correct picture except that, in terms of the tangent metric and a target aspect point $\mathbf{A}_0$ situated off the junction (as is expected in the generic case) this configuration will appear linearly distorted relative to the most symmetrical arrangement illustrated in Fig. 13. Nevertheless, the geometry of the intersection of each corner sector with the spherical shell is reduceable to one of a small number of distinct cases.

Fig. 15 shows configurations that involve intersections of the spherical shell and one or more of the edges of the corner sector. We label the three edges, 1, 2 and 3, and assign the same labels to the three planes of the corner sector that are transversal to the respective edges (and therefore contain the angled interfaces opposite to the same-labeled edges). Where the edge pierces the sphere passing inwards, we denote the intersection one of type 'G', and where it passes outwards, type 'H'. Then a notation for the directed arc from, say, G-type intersection on edge, a, along the plane b to H-type intersection on edge c can be denoted, $[G_a H_c]_b$. The panels a–d of Fig. 15 show the standard set of structurally stable (with respect to arbitrary small perturbations of geometrical parameters) configurations involving respectively two, four, six and three arcs defining the boundary of the intersected spherical shell that intersects edges. Naturally, Figs. 15a and 15b may both appear rotated to the two other edge labelings under cyclic permutation. In addition, Fig. 15a may include a second boundary consisting of a full circle on the plane opposite edge 1, and other configurations with boundaries composed *only* of one, two, or thee such circles are also possible. However, the weight and moment calculations for the unbroken spherical caps are very easily dealt with; the interesting calculation concerns

§ These profiles characterize the shapes of idealized radially symmetric weak shocks in various dimensions.

the weight and moments of the portions of the spherical shell bounded by the strings of arcs shown in Fig. 15 and summarized in our **arc notation** in Table 5.

TABLE 5. SUMMARY OF THE SET OF CIRCULAR ARCS BOUNDING THE REGIONS OF THE SPHERICAL SHELL CONTAINED IN A CORNER SECTOR AS DEPICTED IN THE PANELS OF FIG. 15.

No. of arcs	Panel	Bounding arcs, in the notation defined in the text
2	Fig. 15a	$[G_1 H_1]_2, \quad [H_1 G_1]_3$
4	Fig. 15b	$[G_1 H_1]_2, \quad [H_1 H_2]_3, \quad [H_2 G_2]_1, \quad [G_2 G_1]_3$
6	Fig. 15c	$[G_1 G_3]_2, \quad [G_2 G_1]_3, \quad [G_3 G_2]_1,$ $[H_1 H_2]_3, \quad [H_2 H_3]_1, \quad [H_3 H_1]_2$
3	Fig. 15d	$[H_1 H_2]_3, \quad [H_2 H_3]_1, \quad [H_3 H_1]_2$

The key to solving this part of the problem is to use Green's theorem on stereographic maps, as we shall now describe. Suppose we stereographically project the spherical surface using a projection axis normal to one of these planes and scale the map to make the *equator* parallel to this projection plane a unit circle. The circular arcs shown in Fig. 15 that form the boundary of the region of the shell contained in the corner sector project to circular arcs in this mapping and applications of Green's theorem enable the integrated weight and the first moment in the direction of the projection axis to be computed as contour integrals around the sequence of arcs in the mapping plane. Recall that Green's theorem states that a vector field $\mathbf{v} \equiv (v_1, v_2)^T$ as a function of two coordinates (x_1, x_2) in a domain $\mathcal{D}$ of boundary $\partial\mathcal{D}$ is such that its directed path integral around $\partial\mathcal{D}$ is equal to the surface integral within $\mathcal{D}$ of the curl of $\mathbf{v}$. Thus, parameterizing the boundary curve by t:

$$\oint_{\partial\mathcal{D}} \left(v_1 \frac{dx_1}{dt} + v_2 \frac{dx_2}{dt} \right) \, dt = \iint_{\mathcal{D}} \left(\frac{\partial v_2}{\partial x_1} - \frac{\partial v_1}{\partial x_2} \right) dx_1 \, dx_2. \tag{6.19}$$

It is straightforward to show that a rotationally-symmetric and tangentially-oriented vector field $\mathbf{v}$ of magnitude,

$$|\mathbf{v}| = \frac{r}{2\pi(1+r^2)}, \tag{6.20}$$

at map radius r, integrated around the circle at this radius, yields the proportion of the sphere's area that maps to the interior of this circle. Similarly, the rotationally-symmetric and tangentially-oriented vector field $\mathbf{v}'$ of magnitude,

$$|\mathbf{v}'| = \frac{r}{2\pi(1+r^2)^2}, \tag{6.21}$$

at map radius r, integrated around the circle at this radius, yields the proportionate first moment about the sphere's center, and in the direction normal to the map-plane, in units of the sphere's radius, of the spherical cap that maps to the interior of the circle of integration. But, by the rotational symmetry of these vector fields, these area-integrating and moment-integrating properties must extend to arbitrary regions. Based upon these vector fields, and using stereographic mappings aligned with the normals of each of the three planes of the corner region in turn, we may therefore integrate the directed components of each vector field along

the mapped circular arcs bounding the regions depicted in Fig. 15 to find proportional areas and moments, and, by a little geometry, the desired centroids. While it might be possible, in principle, to evaluate these path integrals analytically, it is expedient numerically simply to employ Gauss-Legendre quadratures, which are found to provide efficient evaluations with truncation errors comparable to double-precision round-off.

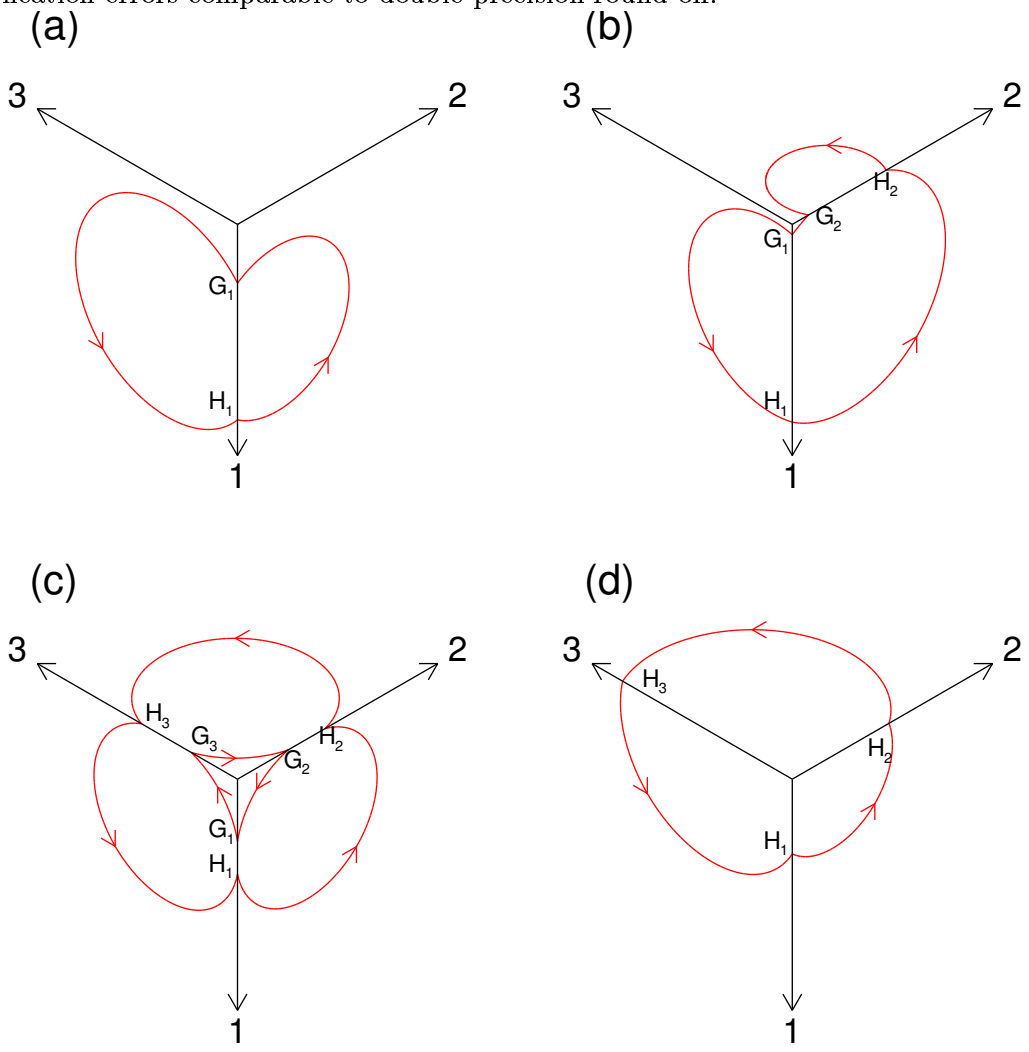

Figure 15. The main configurations of arcs that mark the intersections of a spherical shell with one of the corner regions of the generic junction of hexads in the propoer interior of the aspect cone. The view in each example is as it would appear from inside the corner region projected into the three nontrivial co-dimensions of the junction.

(*f*) *Tabulating and retrieving data for the blended hexad method*

We have already mentioned how smoothly varying data over a simplex-shaped domain in n dimensions can be conveniently stored as a regularly gridded **simplex table**. The smooth data of relevance here are the 13 smoothing weights associated with the lattice line orientations

that go into the *blend* of hexads so that, by applying the line-smoothers sequentially with these weights in the respective directions, one recovers a smoother with the standardized aspect tensor that corresponds to this tabulated point of the canonical hexad. The canonical hexad is taken to be the one of the form (4.3), satisfying the standard relations, (4.2c), and with

$$\mathbf{g}_1 = \begin{bmatrix} 1, & 0, & 0 \end{bmatrix}^T, \tag{6.22a}$$

$$\mathbf{g}_2 = \begin{bmatrix} 0, & 1, & 0 \end{bmatrix}^T, \tag{6.22b}$$

$$\mathbf{g}_2 = \begin{bmatrix} 0, & 0, & 1 \end{bmatrix}^T. \tag{6.22c}$$

The basic weights,

$$\mathbf{w} = [w_1, w_2, w_3, w_4, w_5, w_6]^T, \tag{6.23}$$

of the tabulated point of the canonical hexad are regarded as the *coordinates* for locating positions within the simplex table and the standardization is such that $\sum_{i=1}^{6} w_i = 1$. In other words, the weights form the barycentric coordinates of the canonically transformed aspect tensor, which is also projected onto the corresponding *tile* of the convex shell bounding the convex hull of the lattice generator images. The continuous data stored at each of the simplex table grid points is the vector of 13 *blended hexads* weights, $\mathbf{w}'$, organized by color:

$$\mathbf{w}' = [w'_1, \ldots, w'_{13}]^T. \tag{6.24}$$

While every tabulated point's 13 weights, $\mathbf{w}'$, provide the proper amounts of smoothing associated with the point's nominated *cluster* so that line smoothers applied sequentially along the directions prescribed by the cluster will produce the corresponding standardized aspect tensor, the nominated cluster for such tabulated points is only one of 16 that are associated with the whole simplex making up the complete table. Thus, for each point in the table, it is also necessary to provide an indicator that prescribes precisely which cluster this location is associated with. The discrete transitions from one cluster to another within the interior of the table would seem, at first sight, to invalidate any interpolations to standardized aspect tensors that lie in between the nodes of this canonical simplex tabulation. However, it appears in practice that, at a place in the interior of the table that corresponds to a transition between one nominated cluster and another, the pair (or pairs) of line orientations of equivalent color undergoing the transitions have associated with them weights that conveniently go smoothly to zero on both sides of the transition. A smooth interpolation of the weights $\mathbf{w}'$ therefore *does* make practical sense and truncation errors incurred are acceptably small provided these weights are recorded *color-consistently* for all 16 of the clusters associated with the basic canonical hexad that the complete simplex table is associated with.

Storing the data of a simplex table efficiently requires a non-standard data structure (unlike the *typical* tabulation of n-dimensional data where a Cartesian array is most common). The necessary indirect addressing that is required introduces a very minor inefficiency in the procedures for looking up tabulated values or interpolating from them. However, since this is already unavoidable, there is relatively little further inconvenience in exploiting the hexad symmetries which, as we recall, allow a reduction of data quantity by a factor that approaches 24 asymptotically.

We can retain that portion of the simplex table that occupies the region determined by the following inequalities (and verify that the size of this region asymptotically approaches 1/24 of the total as the resolution index M of the table increases):

$$\min(w_1, w_4) \leq \min(w_2, w_5) \leq \min(w_3, w_6), \tag{6.25}$$

$$w_1 \leq w_4, \tag{6.26}$$

$$w_2 \leq w_5. \tag{6.27}$$

The relabeling symmetries that bring about conformity to the inequalities (6.25) essentially fold the original simplex into 1/6 of its original *volume*, and the remaining two, (6.26) and (6.27), further reduce the non-redundant portions by successive factors of 1/2. [Note: we *cannot* enforce what might seem a further natural inequality, $w_3 \leq w_6$, in general because, half the time, this would violate the chirality condition (4.1).]

The symmetry transformations that rotate the standardized aspect point into the non-redundant part of the simplex must be accompanied by corresponding transformations of the definitions of the lattice generators and their colors. To enable interpolations to be carried out, the retained portion of the table must extend a little way beyond the geometrical symmetry interfaces by which the canonical simplex is dissected. However, these are all relatively straightforward technical issues that are easily taken care of.

The interpolation from the tabulated values to some generic target aspect point presents new problems. It is not appropriate to employ the 5D Cartesian-product of 1D interpolations like the linear, quadratic, cubic, and so on. For one thing, the expense of applying the aforementioned schemes to each scalar fields is in proportion, respectively, to $2^5 = 32$, $3^5 = 243$, $4^5 = 1024$, operations, which, in each example, is much more expensive than is formally justified by the order of accuracy achieved. For another thing, the boundaries of the simplex table are not aligned in the rectilinear Cartesian orientations that make such interpolations natural and symmetric with respect to interchanges of the the six vertex labels of the table's simplex. However, it is possible to chop the simplex into six symmetrical chunks, each of which can, by a projective transformation, be mapped into a 5D *hypercube*. In this highly deformed mapping, Cartesian coordinates *are* very natural to apply. This geometrical trick is useful for other numerical operations (such as efficient quadratures), but we will not pursue the subject further here.

A more natural interpolation, at a given order of accuracy, is to use a local stencil of table nodes that is itself in the form of a lattice simplex. For the linear interpolations, a sub-simplex of only six lattice points (much less than 32) is all that is required. If we should wish to achieve a higher order of formal accuracy, quadratic interpolation in 5D is achieved using a simplex stencil of 21 points (much less than 243); cubic interpolation requires 56 points (much less than 1024). In each of these interpolations, the number of stencil points is exactly the same as the number of coefficients of the polynomial, in five coordinate variables, of the required degree. Thus, these schemes are optimally efficient. However, the dissection of the hexad table's simplex into the requisite sub-simplex blocks of sizes appropriate to the chosen order of accuracy is not a trivial problem in itself. Evenly dividing a simplex in two dimensions, that is, a triangle, presents no difficulty or ambiguity, but even at three dimensions, one finds that, in order not

to create new vertices lying in between the original grid points, care must be taken in choosing the planes along which the sub-simplex blocks are separated.

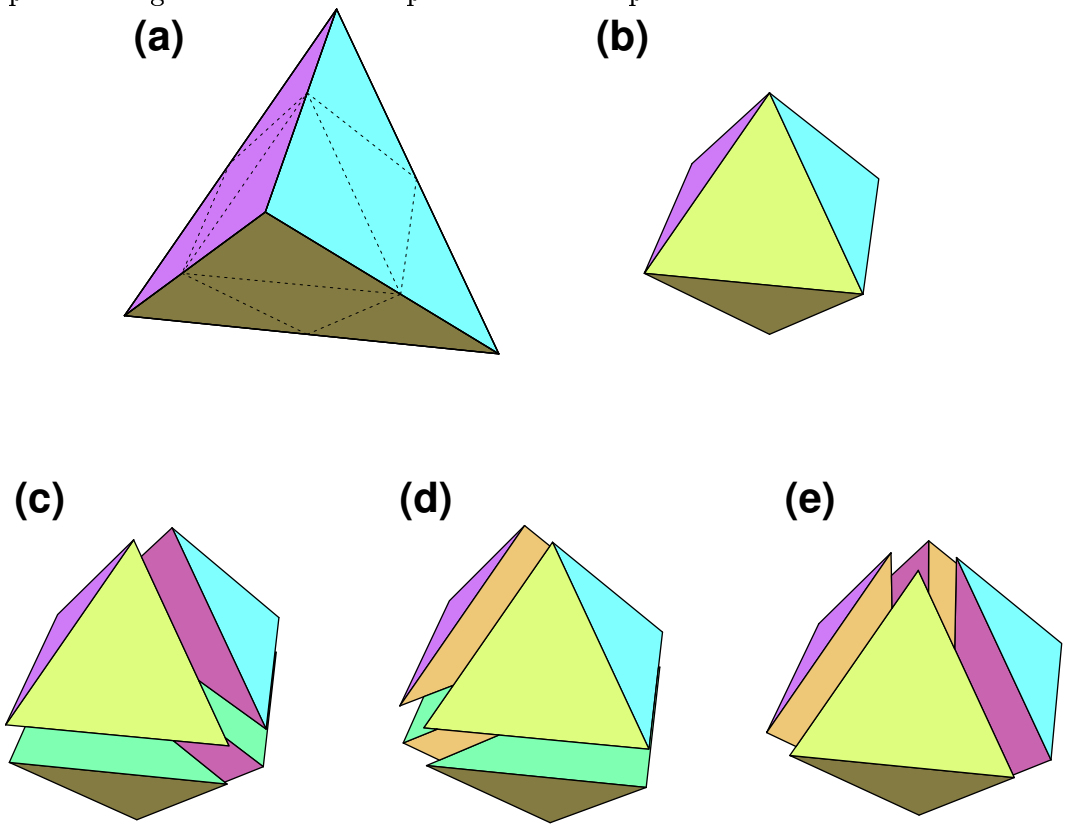

Figure 16. An illustration of the ambiguity inherent in the dissection of a lattice simplex in dimensions of three or greater. Panel (a) shows a 'simplex' (tetrahedron), imagined to be gridded by a lattice whose steps are one half those of the dimensions of region as a whole, and for which we seek a regular dissection into elementary sub-simplexes (in this case, eight of them). The four planes of the original simplex comprise the set of orientations of the mandatory cuts by which the elementary sub-simplexes at the corners are separated to leave an octahedral core, (b). The subsequent dissection of this core into the remaining four inner elements, in a way that creates no new vertices that do not belong to the lattice, can be carried out in three distinct ways, illustrated in panels (c), (d) and (e), requiring two (only) of the available three allowed cuts to be chosen. In a 5D gridded simplex, the ambiguity involves a more daunting choice among 60 valid combinations of selected discretionary cuts in order to reduce the gridded simplex table to its 120 distinct varieties of sub-simplex elements.

The simplex in 3D is, of course, a tetrahedron and the following simple example illustrates the general problem in this case. Suppose we have the tetrahedron gridded with only $M = 2$ intervals along each edge, making a total of 10 grid points in the entire table, and we wish to partition the tetrahedron containing this grid into $8 = 2^3$ sub-simplexes of equal volume. Obviously, four of the dissecting planes need to be parallel to the sides of the large tetrahedron, but after applying these cuts, the remaining core is an octahedron [see Fig. 16, panels (a) and (b)] whose lattice points are all on its surface. Two further cuts are required to reduce this octahedron into the four inner sub-tetrahedra, but three bisecting planes contend for this honor. This is shown in the sketches of Figs. 16c–d. Thus, in the 3D case, we find a three-fold ambiguity in the choice of orientations of the two discretionary planes (those not including the

four mandatory ones parallel to the original tetrahedron's sides). In 4D, five cuts are mandated by the orientations of the large simplex, and an additional five discretionary cuts must be selected, for a total of ten; the total pool of lattice planes from which it is feasible to choose these ten is now $2^4 - 1 = 15$, and the number of combinations that succeed is now 12. Finally, in the 5D case, six cuts are mandated, nine are discretionary (for a total of 15), and the pool of feasible planes are 31 in number. Here we find that 60 combinations satisfy the requirement of dissecting the larger simplex into equal-sized sub-simplexes.

Appendix D describes a general strategy for producing sets of the discretionary dissection planes for a gridded simplex in any number of dimensions. There, it is shown that, by adopting the Scheme A or Scheme B hexad orientation conventions of section 4c, based on the symmetries encapsulated by the $GF(8)$ algebra, we can seek those particularly desirable ways of dissecting each hexad table that cause the 4D patterns of cuts at the boundaries of the adjacent 5D simplexes of the honeycomb to match perfectly. Thus, although based on the seven color hexad classification imposed by $GF(8)$, we could, in principle define 60^7 different dissection strategies, the appendix notes that only 360 of these combinations of dissections have this property that the interpolations remain continuous across hexad boundaries. Moreover, it is noted that just three of these continuity-preserving combinations enjoy the additional especially convenient property that, oriented in conformance with their standard tableaux of section 4c, the hexads' dissection patterns are *independent* of hexad color. Once again, it is the cyclic multiplicative property of the elements of $GF(8)$ that provides the key to establishing this result, as elaborated in appendix D.

We can now review the steps that occur when the blended hexad scheme is presented with a new target aspect tensor. The appropriate hexad containing this point, and the associated six weights $\mathbf{w}$, are found by iterations of the basic hexad algorithm. The aspect point is rescaled ('*standardized*'), by normalizing the six weights so that they sum to one (but recording the normalization factor that this involves). Next, it is necessary to identify the sub-simplex of the tabulation that contains this standardized target point. This is done in two stages. We consider the case where a linear interpolation to the target is sufficient, so that the *source points* of the interpolation are the six vertices of one of the multitude of elementary lattice simplexes, which we must identity. First, regarding the simplex table's lattice points as forming integer-valued 5-vectors, a basic unit-sided lattice parallelepiped containing the target point is found. Second, the location within this particular parallelepiped is refined to determine which of the 120 standard elementaty sub-simplexes that it splits into (via the combination of 'mandatory' and 'discretionary' cuts chosen for the dissection of the standardized hexad's simplex) contains the target. Likewise, if a quadratic order of interpolation is desired, the parallelepiped and subsequent sub-simplex in the procedure above are both of sides two units instead of one, and the 21 lattice points of the sub-simplex and associated weights are then found. It is at this point that the *octahedral-group* of hexad symmetry operations is invoked to transform each of the source-nodes to the portion of the table that satisfies the inequalities, (6.25)–(6.27). The 13 weights $\mathbf{w}'$ can now be interpolated, making due allowance for the permutation of the $GF(3^3)$-based colors that accompanies the octahedral-group symmetry operations alluded to above. The 13 line-smoothing directions of the blended hexads are interpreted by reference to their directions expressed as linear combinations of the first three generators $(\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3)$ of the basic hexad and the 13 weights $\mathbf{w}'$ associated with these blended hexads directions are finally

un-normalized to make the magnitude of the reconstructed aspect tensor correct.

Once this blended hexads table-look-up has been performed for all the points of the analysis grid, it becomes possible to construct continuous line segments threading through the analysis grid, in each of the 13 colors, and, for each color, i, to pass on the weights $\mathbf{w}_i'$ of all the grid points along such a line so that the recursive filter coefficients for this line segment may be computed by the methods described in Purser et al. (2003b). Applying the anisotropic smoother is then a matter of taking each of the 13 colors in turn and performing the appropriate smoothing operations along all the designated line segments associated with this color. Self-adjointness is usually a requirement of the complete smoother, so it is normal to follow one set of 13 smoothing steps with the adjoints of each, but now taken in the reverse order. (Naturally, allowance is made to adjust the aspect tensor passed to the blended hexads procedure to anticipate the factor of two scaling of the aspect tensor that this symmetrization implies).

7. Four dimensions

(*a*) *Generalizations in the style of the triad/hexad approach*

The generalization of an anisotropic smoother to four dimensions involves a further order of magnitude in the complexity of the relevant combinatorial geometry. The number of independent elements to a symmetric aspect tensor in four dimensions is 10, suggesting that we should seek an appropriate **decad** generalization to the 3D *hexad* or 2D *triad* algorithms. As before, we can standardize the aspect tensors within their cone by a suitable rescaling to put them on a 9D surface. However, one serious extra complication here is that, if we are projecting them onto the the 9D polyhedral shell boundary of the convex hull of the aspect vectors (in 10 dimensions) associated with grid generators, we find that the flat pieces that make up this piece-wise flat surface now come in two kinds. The 10-point form is a simplex whose associated 10 opposing pairs of 4D generators form the 20 vertices of the polytope formed as the convolution of a 4D minimal lattice simplex with its invertion, or 'involution'. This is exactly analogous to the construction of triads and hexads in the lower dimensions. But the other flat piece is *not* a simplex but rather a 12-pointed polytope. This is bounded by 64 *sides* which *are* simplexes (in eight dimensions). These 64 sides are themselves of two types, 48 of one, 16 of the other. The problem of constructing a systematic and symmetrical *decad* algorithm is most simply resolved by artificially introducing an additional pseudo vertex at the exact center of each of the 12-point objects. These additions will be referred to as the **hub** points. Symmetry requires that the hub point is attributed a weight of one twelfth of each of the surrounding true generator images, but, in a geometrical division of the 9D convex shell, it will be assumed that each hub acts like a true vertex in most respects. Like slices of a nine-dimensional pie, we can now divide the 12-point object by linking the hub to each of the 64 simplex-shaped boundary pieces, so that the slices created by forming the convex hulls of the hub with each boundary piece *are* 9D simplexes with the requisite 10 vertices. The true tiles of ten points will be referred to as **type-1** tiles; the 48 similar slices of the 12-point objects will be referred to henceforth as **type-2** tiles; the remaining 16 slices of the 12-point object will be referred to as **type-3** tiles. In this way we have once again tiled a generic section of the aspect cone using only simplexes.

The connectivity is as follows:

(i) Type-1 tiles touch type-2 tiles on all 10 sides.

(ii) Type-2 tiles touch one type-1 tile (opposite the hub), 6 type-2 tiles and 3 type-3 tiles.

(iii) Type-3 tiles touch one type-3 tile (opposite the hub) and 9 type-2 tiles.

From the connectivity we infer that the numbers of tiles of each type, 1,2,3, within a sufficiently large ball in aspect space approach the ratio, 3 : 30 : 10. Using a Monte Carlo method, we find that the probabilities of a randomly chosen aspect tensor falling within the three types of tile are approximately in the ratio, 0.263 : 0.579 : 0.158. Therefore, the ratios of the *volumes* of each type of tile are in the approximate ratio: 1.000 : 0.220 : 0.180, respectively.

We have already remarked that the type-1 tiles map back into the 4D lattice as the convolution of a minimal lattice simplex with its involution. The 4D polytope this implies is, in some ways, a natural generalization of the now familiar cuboctahedron; in the most symmetrical linear transformation of the object, we find it to be a 30-sided polytope made up of 10 regular tetrahedra and 20 square-sided triangular prisms. The tetrahedra interface with the ends of the prisms and a square side of each prism interfaces with another prism with their two axes orthogonal. The polytope admits five bisecting hyperplanes in which each of the vertices present are configured as the vertices of the familiar 3D cuboctahedron.

The 12-point object (i.e., the union of the 48 type-2 and 16 type-3 tiles of a co-planar set) maps back to the 4D lattice to form the vertices of a linearly deformed regular polytope known as the **24-cell** (Coxeter 1973, 1999). This is a self-dual configuration of 24 regular octahedra meeting at vertices six at a time. The regular 24-cell has a very high degree of symmetry. Among other things the symmetry of this object makes it natural to group the 12 non-collinear generators into a tableau of three columns of four:

$$\begin{bmatrix} \mathbf{g}_1 & \mathbf{g}_5 & \mathbf{g}_9 \\ \mathbf{g}_2 & \mathbf{g}_6 & \mathbf{g}_{10} \\ \mathbf{g}_3 & \mathbf{g}_7 & \mathbf{g}_{11} \\ \mathbf{g}_4 & \mathbf{g}_8 & \mathbf{g}_{12} \end{bmatrix}, \tag{7.1}$$

where member generators within each *column* are mutually orthogonal in the form of the regular polytope, and those of each *row* are at 120 degrees. With such an arrangement, the 64 sides of the 12-point polytope in 9D are the 9-point simplexes formed by 3 points from column-1, 3 points from column-2 and 3 points from column-3 of the tableau. The set of 16 are those for which the associated 3 generators *not* represented are the linearly dependent ones - the other 48 do not have this property. We can always adopt the convention that, in the standard configuration of a type-3 tile, it is the elements $\mathbf{g}_1$, $\mathbf{g}_5$, and $\mathbf{g}_9$ (i.e., all the ones on row-1) of the 12 that are left out, while for a type-2 sub-tile rotated into its standard configuration, we omit the elements $\mathbf{g}_2$, $\mathbf{g}_8$ and $\mathbf{g}_{11}$ (i.e., all on *different* rows and columns and *not* on row-1).

The basic *decad* algorithm proceeds similarly to the hexad except that, when the standardized aspect tensor is found to lie within a type-2 or type-3 tile, the weight attributed to the artificial hub point must be re-apportioned uniformly over the 12 true generators. Thus, even without any kind of further blending we might wish to add for cosmetic reasons, the basic algorithm in this most simple and symmetrical form will require one to smooth along 12 generalized grid orientations in general.

The basic decad algorithm lends itself to a scheduling of the smoothing operations based on a *color coding* provided by the Galois field $GF(2^4)$. This uses a palette of 15 colors. It is natural to ask whether it is possible to formulate a practical *blended* extension, analogous to

those we have described for the triad and hexad methods. The geometrical complexity involved would be enormous. For one thing, a tabulation of the weights over a representative portion of each of the tile types would involve simplex tables of nine dimensions, so storage requirements would be huge for any moderate resolution of the table. Even for a very modest resolution, the kernel averaging over the *cluster* surrounding a generic junction would be a difficult task. The co-dimension of the generic junction between tiles in the interior of the aspect cone is now six. Quadratures in dimensions as large as this, over irregular shaped sectors that meet at the junction, do not seem to be amenable to such nice numerical tricks as those of sections 6d, 6e, that we were able to use in the hexad case. However, Monte Carlo integration methods, or some quasi-regular variants, might just be feasible.

As an indication of the kind of geometrical complexity one would need to deal with near each junction, it seems that 200 type-1 tiles and 104 of the 12-point objects, including at least some of the type-2 and type-3 sub-tiles they contain, must meet at a generic junction. Also, the hope that the *cluster* of tiles possesses only generators that are accommodated by the 40-color scheme associated with $GF(3^4)$ is dampened by the fact that, of the 104 12-point objects, surrounding the junction associated with the identity aspect tensor in a 4D Cartesian lattice, 32 of them contain generators with a component ± 2 that would *not* be contained within the fundamental cubic array of $3 \times 3 \times 3 \times 3$ that this Galois field could distinguish. It might be possible to divide up the original 12-point tiles in some alternative, more complicated way, in order to solve this difficulty. Failing that, the use of the next higher-order of Galois field, $GF(5^4)$, while technically possible, would be unappealing owing to the fact that this implies the huge palette of 156 colors. The number of non-collinear generators involved in the *cluster* associated with the junction that we have described is 72. It is not clear whether any adequate alternative division of the original 12-point tiles could reduce this to the 40 that would fit within the scope of the $GF(3^4)$ scheme, but this is a subject which merits further investigation, since the ability to apply a *blended decads* algorithm in 4D assimilations would be of great value.

(*b*) *Generalizations employing 'Cholesky flow'*

An alternative strategy for augmenting an anisotropic quasi-diffusive smoother from its existing n dimensions to $n+1$ dimensions is possible in the form of what we shall refer to as the **Cholesky flows** method. It essentially assumes a splitting of the aspect tensor and its associated simulated diffusion operator into parts residing purely in each of the n-dimensional lattice's hyperplanes transversal to the $(n+1)$th dimension of the domain, plus a part representing one-dimensional diffusion along quite generally oblique, but smoothly continuous, **flow lines** transversally intersecting all the aforementioned hyperplanes. Here, the flow lines do *not* link lattice points from one hyperplane to the next in the way that all our previously discussed algorithms do, so the most direct numerical implementation of the 1D diffusion along these flow lines must include n-dimensional interpolation operators as each hyperplane is intersected, in order to bring the smoothed values back onto the computational lattice. The accuracy of the interpolations must also be sufficient (essentially third-order or better) to minimize additional spurious dispersion. The method possesses some features in common with semi-Lagrangian advection schemes, so, not surprisingly, it can benefit from some of the same numerical techniques, such as the method of **cascade** interpolation suggested by Purser and Leslie (1991). Also, since the method is generic and works in $(n+1)$ dimensions, there is no reason why it should not

be recursively extended back to the n-dimensional part of the decomposition, and so on. Thus, the Cholesky flows method may be used in combination with the triad or hexad approaches to extend the anisotropic covariances into the time domain, or used in its purest form as a competing alternative to these schemes. While, at the present time, the computational cost and complexity of the high-order interpolations that the Cholesky flows schemes require make them unattractive as competitors with the efficient triad and hexad schemes for two or three dimensions, the Cholesky flows combined with the blended hexads method is almost certainly the best way we have at present for the 4D synthesis of generally anisotropic quasi-Gaussian covariance operators. This situation may even remain unchanged regardless of future developments in the *blended decads* approach owing to the onerous numerical burden that the best possible of these latter methods must impose associated with their expected 40-orientation smoothing sequence. Therefore, although a detailed experimental investigation of Cholesky flows methods is beyond the scope of this article, we provide an abbreviated technical summary of this approach in appendix E.

8. Conclusion

It has not been possible (or desirable) to delve into the complete details of every part of the algorithms we have described. Rather, it has been the intention to provide a scientific and geometrical overview of the main techniques that have gone into the construction of them so that the various major tasks carried out by the computer code itself are no longer particularly mysterious or cryptic. At the present time, the codes for the basic triad and hexad algorithms and the new alorithms for the blended triads and blended hexads, are written, tested and considered to be at a stage of maturity where they can be relied upon. Further developments of these codes will most likely be in the form of very minor refinements ('tweaking') rather than any major overhauls.

The four-dimensional code is not yet at such a mature stage of development. The algorithm described for the *basic decad* method, in which the 12-point objects are split into simplex pieces through the intervention of artificial *hub* points, seems to be the simplest way to form a *basic* version of the algorithm but, as we have discussed in the previous section, this approach does not lend itself to an efficient construction of the *blended decads* extension analogous to those successfully developed in the much simpler triad and hexad cases. It is anticipated that, before a mature blended decads algorithm becomes a reality, we will need to replace the *pie-slice* dissection of the 12-point aspect space regions by some other reduction to simplexes, or even by some entirely different geometrical partitioning, in order that the blended algorithm's smoothing directions at any generic analysis grid point can all be uniquely *colored* by the 40 non-null element-pairs of the relatively modest Galois field, $GF(3^4)$. The alternative, blending the decads that involve the central *hub* points of the 12-point objects, can be done in principle, but only lends itself to automatic scheduling if we are prepared to tolerate the complexity and inefficiency implied by using the 156 colors associated with the unwieldy Galois field, $GF(5^4)$. A smooth and reliable anisotropic filtering scheme in four dimensions will become inportant in the future if *weak model constraint* 4DVAR, such as has been pioneeered by Bennett et al.(1996, 1997), becomes a serious contender for the operational data assimilation schemes of the future.

While the emphasis of this note has been applications to anisotropic quasi-diffusive *smoothing* operators on regular grids, we note that there is a complementary family of numerical operators which could also be constructed according to the principles embodied in the triad and hexad approaches. These are the self-adjoint second-order elliptic operators, such as occur in the diffusion equation. Formally, the logarithm of the linear operator representing the effect of finite-time inhomogeneous and anisotropic diffusion *is* such an elliptic operator, but the useful applications of elliptic operators of this form are not confined merely to the diffusion equation but occur (in the meteorological or oceanic context) in semi-geostrophic theory (Hoskins and Bretherton 1972, Schubert 1985), for example. For these elliptic operators, the triad and hexad methods provide a way to synthesize these operators as *additive* combinations of simple second-order differential line-operators. Exact numerical self-adjointness is also brought about here by *adding* the adjoint of the unadjusted operator (and dividing the result by two, of course). This approach has the advantage that the differencing stencil is assured not to possess any spurious *negative* numbers, which are notoriously problematic in numerical treatments of anisotropic elliptic operators.

Acknowledgment

This study benefitted greatly from the many stimulating discussions I had with Drs. David F. Parrish here at NOAA/NCEP and Dezső Dévényi of NOAA/FSL, Boulder. The work was partially supported by the NSF/NOAA Joint Grants Program of the US Weather Research Program. This research is also in response to requirements and funding by the Federal Aviation Administration (FAA). The views expressed are those of the author and do not necessarily represent the official policy or position of the FAA.

Appendix A

Kurtosis relations

The kurtosis of a distribution described by the function $f(x)$ symmetric about the origin ($f(x) = -f(-x)$) and which is normalized to unit integral, $\int f(x)\,dx = 1$, is defined to be the fourth central moment divided by the square of the second central moment. Thus, defining $\sigma = \int f(x)\,x^2\,dx$ and $\tau = \int f(x)\,x^4\,dx$ the kurtosis of f is

$$\kappa = \tau/\sigma^2. \tag{A.1}$$

We observe that the kurtosis in unchanged by a change in horizontal scale.

The kurtosis for a pair of equal impulses is clearly 1 and we find that, for more continuous distributions, $\kappa > 1$. Informative examples include the square- and triangular *hat* functions, $f_s(x) = 1$, $|x| < 1/2$ and $f_t(x) = 1 - |x|$, $|x| < 1$, where the support in each case is the range indicated. Their kurtosis is easily verified to be $\kappa_s = 9/5$ and $\kappa_t = 12/5$ respectively. An example of a less 'boxy' distribution is the exponential, $f_e(x) = \frac{1}{2}\exp(-|x|)$ whose kurtosis is $\kappa_e = 6$. But the *standard* against which we tend to compare other distributions' kurtosis is the Gaussian, $f_g(x) = (2\pi)^{-1/2}\exp(-x^2/2)$, whose kurtosis is $\kappa_g = 3$. The central limit theorem of classical statistics formalizes the notion that convolving two functions produces another that is, in well defined ways, *more Gaussian* than its progenitors. In the context of kurtosis it is

not surprising that we corroborate this notion; by the use of the moment generating functions (Fourier transform) of distributions f_1 and f_2, and an application of the convolution theorem (Bracewell 1965, Champeney 1973) we may expand the product moment generating function as a series:

$$\begin{aligned} \tilde{f}_3(k) &= 1 - \frac{1}{2}\sigma_3 k^2 + \frac{1}{24}\tau_3 k^4 + \ldots \\ &= \tilde{f}_1(k).\tilde{f}_2(k). \end{aligned} \tag{A.2}$$

With similar expansions for factors $\tilde{f}_1(k)$ and $\tilde{f}_2(k)$ and employing the relations coupling κ, σ and τ, we find that,

$$\begin{aligned} \sigma_3^2\kappa_3 &= \sigma_1^2\kappa_1 + \sigma_2^2\kappa_2 + 6\sigma_1\sigma_2, \\ &\equiv \sigma_1^2\kappa_1 + \sigma_2^2\kappa_2 + (2\sigma_1\sigma_2)\kappa_g, \end{aligned} \tag{A.3}$$

where, since $\sigma_3 = \sigma_1 + \sigma_2$,

$$\sigma_3^2 = \sigma_1^2 + \sigma_2^2 + 2\sigma_1\sigma_2.$$

Thus we see that, since the κ_3 is a weighted average of κ_1, κ_2 and κ_g, then the only possible stable fixed-point for the kurtosis of functions combined under convolution is $\kappa = \kappa_g = 3$. Note that the functions f_t and f_s above are also related by convolution, $f_t = f_s * f_s$, being the first two of a the series of basis functions on unit grids of the so-called **B-splines** (de Boor 1978) which are formed by successive convolution with f_s and which all therefore have kurtosis approaching, but less than that of the Gaussian.

Next, we consider the effect on kurtosis of *mixing* distributions. Let $f_3 = w_1 f_1 + w_2 f_2$, with $w_1 + w_2 = 1$. First we note that

$$\begin{aligned} \sigma_3 &= w_1\sigma_1 + w_2\sigma_2, \\ \tau_3 &= w_1\tau_1 + w_2\tau_2. \end{aligned} \tag{A.4}$$

Then insertion of these into the kurtosis definition reveals that

$$\kappa_3 = \bar{\kappa}(1 + g), \tag{A.5}$$

where $\bar{\kappa}$ is a generalized average:

$$\bar{\kappa} = \frac{w_1\sigma_1^2\kappa_1 + w_2\sigma_2^2\kappa_2}{w_1\sigma_1^2 + w_2\sigma_2^2},$$

and g is the non-negative quantity:

$$g = \frac{w_1 w_2(\sigma_1 - \sigma_2)^2}{\sigma_3^2}, \tag{A.6}$$

which becomes strictly positive when the second moments differ. Therefore, when mixing distributions the resulting kurtosis is never less than a generalized average of the components' kurtoses and tends to be increased significantly when the components have widely differing

spreads. It is this result, and its generalization to higher dimensions, that justifies the principle that low-kurtosis (thin tailed) distributions such as the Gaussian make more reasonable models of covariances in an adaptive assimilation than in a non-adaptive one.

Appendix B

Tiling property of hexads

In the case of the 3D regular lattice we have asserted that the configurations of grid generators that form linearly deformed cuboctahedrons have corresponding generator images in aspect space that define, via their convex hulls, the natural *tiles* covering the interior of the aspect cone. Here we justify this assertion using an argument based on *convexity*. The geometry takes its simplest form when the representative hexad is maximally symmetric, which occurs when we choose the lattice to be of face-centered cubic (fcc) variety, (as also defined in subsection 6a). For example, the position vectors of the lattice points comprise the integer triples that sum to an even number. Then the six opposing pairs of generators for the most symmetrical hexad are $\pm$ the columns of:

$$\pm \begin{bmatrix} 0, & 1, & 1, & 0, & -1, & 1 \\ 1, & 0, & 1, & 1, & 1, & -1 \\ 1, & 1, & 0, & -1, & 1, & 0 \end{bmatrix}, \tag{B.1}$$

with generator images forming the columns of:

$$\begin{bmatrix} 0, & 1, & 1, & 0, & 1, & 1 \\ 1, & 0, & 1, & 1, & 0, & 1 \\ 1, & 1, & 0, & 1, & 1, & 0 \\ 1, & 0, & 0, & -1, & 0, & 0 \\ 0, & 1, & 0, & 0, & -1, & 0 \\ 0, & 0, & 1, & 0, & 0, & -1 \end{bmatrix}, \tag{B.2}$$

when each generator image of $\mathbf{g} = [g_1, g_2, g_3]^T$ is arranged into the column,

$$\mathbf{A} = [g_1 g_1, g_2 g_2, g_3 g_3, g_2 g_3, g_3 g_1, g_1 g_2]^T . \tag{B.3}$$

Let

$$\mathbf{n} = [1, 1, 1, 0, 0, 0]^T . \tag{B.4}$$

Then the dot product

$$\mathbf{n}^T \cdot \mathbf{A} = 2, \tag{B.5}$$

defines a 5D affine subspace in which all six of the generator images in (B.2) clearly reside. For *any* generator, $\mathbf{g}$, with image, $\mathbf{A}$, it is also clear that,

$$\mathbf{n}^T \cdot \mathbf{A} \equiv \mathbf{g}^T \cdot \mathbf{g} \geq 2, \tag{B.6}$$

and that strict inequality pertains for all generators of the lattice *other* than the 12 defined by (B.1). Thus, while all generator images fall into the half-space (B.6) only the six of (B.2) fall

onto its boundary. The rank of the matrix (B.2) is five, so these columns define a 5D simplex. We have already shown how following the hexad transition rules leads to a new analogous hexad having the five unchanged vertices in common. Thus, by symmetry, this new hexad's generator images must also form a 5D simplex in aspect space meeting the old hexad's image simplex across the *side* corresponding to the five generator images the two hexads have in common. This argument applies to all six of the possible transitions, ensuring that each simplex tile of aspect space is surrounded on all six sides by another simplex tile just like it. Cuboctahedral hexads are therefore *not* arbitrary, but are the uniquely natural geometrical forms associated with aspect space decompositions related to a regular 3D lattice.

The same general principles are brought to bear in the case of the 4D lattice, where the convexity arguments confirm that two species of polytopes are implied by the tiling formed by the boundary elements of the convex hull of all the generator images.

Appendix C

Distance from a point to a simplex

A simplex S_n with $n+1$ vertices in n dimensions is the convex hull of these vertices and has the property that its boundary comprises $n+1$ simplexes each of n vertices in $n-1$ dimensions. This principle extends recursively to successively lower dimensions, to triangles, line segments and points. The **affine hull** of a set of points is the affine subspace (like a linear subspace, but not necessarily passing through the origin) of smallest dimension that contains these points (van Tiel 1984). For an m-dimensional simplex S_m whose $m+1$ vertices $X_i, i = 0, \ldots m$ in an n-dimensional Cartesian coordinate system are $\mathbf{x}_i, \quad i = 0, \ldots m$, the m vectors, $\mathbf{x}_i - \mathbf{x}_0, \quad i = 1, \ldots m$, are independent and lie in the linear space parallel to the affine hull of S_m A Gram-Schmidt orthogonalization (e.g. Golub and Van Loan 1989) of these vectors supplies a Cartesian basis, $\mathbf{v}_i, \quad i = 1, m$, which we can combine, as mutually orthogonal columns, into a matrix, $\mathbf{V}$, such that any point X in the affine hull, with coordinates $\mathbf{x}$, is also defined,

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{V}\mathbf{y}, \tag{C.1}$$

for some m-vector $\mathbf{y}$ of alternative Cartesian coordinates associated with the basis $\mathbf{V}$.

In order to calculate the Euclidean distance in n dimensions between a simplex S_m of dimension $m > 0$ and a point P represented by the position vector $\mathbf{p}$ we proceed as follows. Drop a perpendicular from P to a point Y in the affine hull of S_m whose coordinates with respect to the basis $\mathbf{V}$ are:

$$\mathbf{y} = \mathbf{V}^T(\mathbf{p} - \mathbf{x}_0). \tag{C.2}$$

Express $\mathbf{y}$ as barycentric coordinates, $(w_0, \ldots, w_m)$, with respect to the $m+1$ vertices of S_m, which, with respect to the basis $\mathbf{V}$, we take to be at $\mathbf{y}_i, \quad i = 0, \ldots, m$ (with $\mathbf{y}_0 = \mathbf{0}$). If we collect all but $\mathbf{y}_0$ of the $\mathbf{y}_i$ into the matrix $\mathbf{Y}$ and all but w_0 into the m-vector, $\mathbf{w}$, then $\mathbf{w}$ is found by inverting:

$$\mathbf{y} = \mathbf{Y}\mathbf{w}, \tag{C.3}$$

and the missing component recovered using, $w_0 = 1 - \sum_{i=1}^{m} w_i$. If these barycentric coordinates are all non-negative, Q lies in (or on) the simplex, S_m, and its distance to P is simply D, where,

$$D^2 = (\mathbf{p} - \mathbf{x}_0)^T(\mathbf{p} - \mathbf{x}_0) - \mathbf{y}^T\mathbf{y}. \tag{C.4}$$

Otherwise, the distance to S_m is the minimum of the $m+1$ distances to the simplexes of dimension $m-1$ that make up the boundary of S_m. From this fact and the aforementioned construction of any simplex from simplex parts of lower dimensions, we can recursively determine the distance between P and the original simplex S_n of dimension n, since the recursions are bound to terminate with some definite nearest subsimplex (even if this is merely a point) that *contains* the perpendicular dropped into its affine hull from P.

Appendix D

Simplex grid dissections

The problem we initially address is to find a characterization of all the possible ways to decompose a gridded simplex by families of parallel and regularly spaced cuts so that the resulting fragments are '**minimal simplexes**' possessing only the original grid points as their vertices and having no other grid points in their interiors. We state a solution to this problem and sketch an outline of the proof required to demonstrate its correctness. We then show how an application of these dissections combined with the insights provided by Galois field theory lead us to a simple and convenient specification of the dissection structure of the generic hexad's simplex table that guarantees continuity of interpolated values across the hexad's boundary.

In 2D the problem is obviously trivial as the simplex (triangle) admits only one decomposition that employs the three families of cuts parallel to the three sides. In higher dimensions, the decomposition is no longer unique. For example, in three dimensions, where the simplex is a tetrahedron, in addition to the four mandatory families of cuts parallel to the simplex faces, two additional families must be specified, but, as we have noted and illustrated in Fig. 16, there are three ways in which this can be done.

We shall suppose that the original simplex occupies the subset of $\mathbf{x} \in R^n$ defined by the $n+1$ inequalities:

$$\sum_{i=1}^{n} x_i \quad \leq \quad M, \tag{D.1}$$

$$x_i \quad \geq \quad 0, \tag{D.2}$$

where integer M defines the resolution of the simplex table whose grid points here form a piece of a unit Cartesian grid. Each family of valid cuts must comprise the hyperplanes defined by constraints of the form:

$$\mathbf{x}^T \cdot \mathbf{b}^{(k)} = q, \qquad q = 0, \ldots, M, \tag{D.3}$$

where $\mathbf{b}^{(k)}$ is a non-null vector, normal to the planes it constrains, whose components are all either ones or zeroes (otherwise, fractional grid intervals would be formed along edges of the original simplex). The set $\mathbf{N} = \{\mathbf{b}^{(k)}\}$ certainly includes each of the vectors with a single 'one' and the vector of all 'ones', corresponding to planes that are parallel to the sides of the original simplex. In addition, for there to be no interior points of intersection (vertices of the minimal simplexes) that are *not* integer-component vectors, we require that the determinant of any $n \times n$ matrix formed from the vectors $\mathbf{b}^{(k)}$ taken n at a time is one of $\{-1, 0, 1\}$, and not some other integer. It is useful to attribute to a region of a lattice hyperplane a **surface**

measure defined by the maximum of the n Euclidean measures of the orthogonal projections of this region onto the n mutually orthogonal coordinate hyperplanes. The surface measure of each single $(n-1)$-dimensional face of a minimal simplex is always $1/(n-1)!$ units, making the total surface measure of the simplex $(n+1)/(n-1)!$ units. Generically, every family of cuts can be regarded as creating precisely two units (one on either side of the cut) of surface measure per basic parallelepiped. Since $n!$ minimal simplexes occupy each generic basic grid-unit hypercube or parallelepiped, it therefore requires $K(n)$ distinct families of parallel cuts to dissect the simplex table into its minimal simplexes, where:

$$K(n) = \binom{n+1}{2}. \tag{D.4}$$

We assert that we can form the set $\{\mathbf{b}^{(k)}\}$, $k = 1, \ldots K(n)$ with the requisite properties in the following way. First, define a **staircase basis B**, to be a square matrix whose n column vectors are such that, by appropriately permuting the indices labeling the components of these vectors (i.e., the *rows* of the matrix), one obtains a $\mathbf{B}$ in the standard form:

$$B_{i,j} = \begin{cases} 0 & : \quad i < j, \\ 1 & : \quad i \geq j. \end{cases} \tag{D.5}$$

We may take $b_i^{(k)} = B_{ik}, \quad k \leq n$ and augment such a basis with additional columns to complete the set $\mathbf{N}$, with the additional columns $\mathbf{b}^{(k)}, \quad k > n$, that comprise all the differences $b_i^{(k)} = B_{i,j} - B_{i,j'}$, with $j < j'$. For the 3D example,

$$\mathbf{N} = \left\{ \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \right\} \tag{D.6}$$

In general, this construction produces all the finite n-vectors whose non-vanishing components form a contiguous string of 'ones'.

The resulting augmented set of $K(n)$ vectors possesses the determinant properties we require. To prove this, note that if two of a subset of any n of these vectors chosen to form a trial matrix have the same largest row index where a 'one' appears, then, without changing the determinant, we can replace the vector with the larger number of 'ones' with the vector formed by subtracting the other vector from this one. An iteration of this process ensures that, eventually, all the columns of the matrix will have a different row-position for their last 'one', or will contain zero columns. The determinant is therefore one of $\{-1, 0, 1\}$ as asserted.

The standard form of the staircase basis specifies a particular permutation of the vertices of the original simplex: the vertex occupying the origin followed, in strict order, by the vertices 'pointed to' by the successive row labels. However, amongst the columns of $\mathbf{N}$ are the n which also form a staircase basis, sharing the same $\mathbf{N}$, but with the row-index ordering reversed; other row-index permutations do **not** lead to a valid staircase basis, so, the total number of distinct $\mathbf{N}$ for $n \geq 2$ is $n!/2$. If we affinely map the large simplex and an associated dissection encoded by $\mathbf{N}$ to the same region defined by (D.1), (D.2), we find that the mapped families of cuts will

also conform to a pair of staircase basis constructions, in which the new implied permutations of the vertices are simply the two cyclic permutations of the previous ones that put the new origin-occupying vertex in first position. Thus, each of the $n!/2$ schemes of cuts constructed via staircase bases are associated with one of the $n!/2$ distinct direction-indifferent circuits through the $n+1$ vertices of the original simplex.

We can now use the identification of a valid simplex dissection with the cyclic pattern of its vertices to characterize a complete solution to the problem of matching the boundary patterns of cuts of adjacent hexads color coded by the seven-color $GF(8)$ scheme. The trick is to observe that the color assignments of vertices and hexads in the $GF(8)$ scheme is exactly what would be observed if we were to 'roll' a 6D (seven cornered) simplex from one neighboring hexad to another, 'printing' the seven different vertex colors and hexad colors as we proceed. There are $6!/2 = 360$ different valid ways of dissecting the 6D simplex, which induce corresponding valid ways of dissecting the seven different colored hexads. If we are prepared to specify a *different* way of dissecting the hexad according to its particular color assignment, then each of the 360 ways of dissecting the 6D simplex leads to a distinct hexad dissection scheme of this kind. However, an enormous simplification is achieved if we demand that each hexad is dissected in the *same* way, once it is oriented by the appropriate standardizing *multiplicative* operation amongst the non-null elements of $GF(8)$ in the manner described in section 4c. All we need to do is to choose the circuit through the seven 'colors' that define the dissection of the 6D simplex to also be a constant-stride circuit around the cycle of the corresponding multiplicative group of the seven non-null elements of $GF(8)$. Noting that strides one and six give equivalent direction-indifferent circuits, as do strides two and five, and strides three and four, then there are precisely these three distinct ways (out of the original 360) which enjoy this elegant property of requiring only a single standard dissection scheme for *all* the hexads, that guarantees a perfect match of the patterns of cuts at hexad boundaries, and that therefore allow perfect continuity of interpolations from the simplex tables as interpolation targets cross hexad boundaries.

Appendix E

Cholesky flows

Suppose we wish to extend the capability of a given anisotropic quasi-Gaussian smoother in n dimensions to enable it to work in $n+1$ dimensions. The Cholesky flows method we describe here represents one possible strategy for solving this problem. Let us suppose that an existing n-dimensional smoother acts in each grid hyperplane transverse to the $(n+1)$th dimension, t, dispersing or diffusing their input data in a way quantitatively compatible with the field of n-dimensional aspect tensors, A, provided to it on each hyperplane. The additional dimension we have denoted 't', while conveniently spoken of here as 'time', need not necessarily by the actual temporal coordinate in this general abstract approach. Also, threading transversally to these hyperplanes, suppose we specify a '**Cholesky flow**' in the form of a smooth field of n-vectors, $\mathbf{u}$. To the extent that the 'time' interpretation is valid, this vector $\mathbf{u}$ is an effective advecting velocity which allows us to split the $(n+1)$-dimensional smoothing task into two parts applied one after the other. First, we may apply the n-dimensional smoother with partial aspect tensor A at *all* 'times'. Second, we apply to the output of the first smoother another that

acts primarily in the 'time' dimension, except in the 'Lagrangian' sense of diffusing along the 'trajectories' implied by the Cholesky flow vector field $\mathbf{u}$. This second smoother is essentially one-dimensional so the aspect tensor has just the single component, a. The square-root of a has the interpretation of being the coherence 'time' scale along the Cholesky flow trajectories.

Provided the changes in the tensor field A, the vector field $\mathbf{u}$ and the scalar a are not too abrupt in comparison to the intrinsic dispersion scales throughout the $(n+1)$-dimensional domain that field a and the the components of A imply, the resultant dispersion of the combined smoothers, quantified by the central second moments of the $(n+1)$-dimensional aspect tensor, which we shall call $\bar{\mathsf{A}}$ will, to an approximation whose validity is as good as the aforementioned assumption of smoothness, satisfy:

$$\bar{\mathsf{A}} = \mathsf{U}.\mathsf{D}.\mathsf{U}^T, \tag{E.1}$$

where the structures of the matrices on the right are as follows.

$$\mathsf{U} \equiv \begin{bmatrix} \mathsf{I}, & \mathbf{u} \\ \mathbf{0}^T, & 1 \end{bmatrix}, \tag{E.2}$$

and,

$$\mathsf{D} \equiv \begin{bmatrix} \mathsf{A}, & \mathbf{0} \\ \mathbf{0}^T, & a \end{bmatrix}. \tag{E.3}$$

This is formally a modification of the classical block-matrix variant of the Cholesky factorization of a symmetric matrix into mutually-transpose block-triangular factors. The modification, which avoids explicit square-root operations, is the intermediate block-diagonal factor, D. The second row and column 'blocks' are actually just the single added dimension, t. In practice, of course, we start with $\bar{\mathsf{A}}$ and factorize it in this fashion in order to obtain the A, the $\mathbf{u}$ and the a that enables the two-stage construction of the complete smoother to proceed.

Practical implementation of a diffusive process along the Cholesky flow lines would involve the constant intervention of n-dimensional interpolations, similar to those used in semi-Lagrangian advection schemes, at each successive step from one hyperplane (pseudo-time level) to the next. Probably the most efficient way to carry out the 'time' smoothing part of this procedure is therefore to adopt the double-sweep (forward, then backwards) recursive filter methods to execute the dispersive component of the problem and to apply the simplest form of the *Cascade* interpolation procedure (Purser and Leslie 1991) to regain a foothold on the computational grid after each grid interval of 'time', t.

There is no obstacle, in principle, to chaining a sequence of such Cholesky flows procedures to build up the smoothed result at successive increments of the problem's dimensions. Such a recursive application constitutes an alternative methodology to the use of the triad or hexad methods. However, the considerable computational burden of the multiple interpolations appear to make this grand recursion uncompetitive; only when the stage is reached where we need to go from three dimensions to four does the Cholesky flows approach seem to enjoy the advantage.

References

Abramowitz, M., and I. A. Stegun 1970 *Handbook of Mathematical Functions*, Dover, New York. 1046 pp.

Barnes, S. L. 1964 A technique for maximizing details in numerical weather map analysis. *J. Appl. Meteor.,* **3,** 396–409.

Bennett, A. F., B. S. Chua, and L. M. Leslie 1996 Generalized inversion of a global numerical weather prediction model. *Meteor. Atmos. Phys.,* **60,** 165–178.

Bennett, A. F., B. S. Chua, and L. M. Leslie 1997 Generalized inversion of a global numerical weather prediction model, II: Analysis and implementation. *Meteor. Atmos. Phys.,* **62,** 129–140.

Bergthórsson, P., and B. R. Döös 1955 Numerical weather map analysis. *Tellus* **7,** 329–340.

Bracewell, R. 1965 *The Fourier Transform and its Applications.* McGraw-Hill, New York. 444 pp.

Champeney, D. C. 1973 *Fourier Transforms and their Physical Applications.* Academic Press, New York. 256 pp.

Courtier, P. 1997 Dual formulations of four-dimensional variational assimilation. *Quart. J. Roy. Meteor. Soc.,* **123,** 2449–2461.

Coxeter, H. S. M. 1963 *Regular Polytopes*, Dover, New York. 321 pp.

Coxeter, H. S. M. 1968 *The Beauty of Geometry* Dover, New York,274 pp.

Cressman, G. P. 1959 An operational analysis scheme. *Quart. Mon. Wea. Rev.,* **87,** 367–374.

Da Silva, A., J. Pfaendtner, J. Guo, M. Sienkiewicz, and S. Cohn 1995 Assessing the effects of data selection with DAO's physical-space statistical analysis system. *Second International Symposium on Assimilation of Observations in Meteorology and Oceanography*, Tokyo, 13–17 March 1995. WMO/TD., **651,** 273–278.

Daley, R. A. 1991 *Atmospheric Data Assimilation.* Cambridge University Press, 457 pp.

Dantzig, G. B. 1963 *Linear Programming and Extensions*, Princeton, New Jersey. 648 pp.

de Boor, C. 1978 A Practical Guide to Splines. Springer, New York. 392 pp.

Derber, J. C., and A. Rosati 1989 A global ocean data assimilation system. *J. Phys. Ocean.,* **19,** 1333–1347.

Dickson, L. E. 1958 *Linear Groups (With an Exposition of the Galois Field theory).* Dover, New York, 336 pp.

Fano, G. 1892 Sui postulati fondamentali della geometria proiettiva in uno spazio lineare a un numero qualunque di dimensioni. *Giornale di Matematiche,* **30,** 106–132.

Gandin, L. S. 1963 *Objective Analysis of Meteorological Fields.* Gidrometeorologicheskoe Izdatel'stvo, 242 pp. Translated by Isreal Program for Scientific Translations.

Gaspari, G., and S. E. Cohn 1998 Construction of correlation functions in two and three dimensions. Office Note Series on Global Modeling and Data Assimilation, DAO Office Note 96-03R1, DAO, GSFC, 53 pp.

Gaspari, G., and S. E. Cohn 1999 Construction of correlation functions in two and three dimensions. *Quart. J. Roy. Meteor. Soc.,* **125,** 723–757.

Ghil, M., K. Ide, A. Bennett, P. Courtier, M. Kimono, M. Nagata, M. Saiki, and N. Sato	1997	*Data Assimilation in Meteorology and Oceanography: Theory and Practice* Meteorological Society of Japan. 386 pp.
Golub, G. H., and C. F. Van Loan	1989	*Matrix Computations.* Johns Hopkins, 642 pp.
Hayden, C. M., and R. J. Purser	1995	Recursive filter objective analysis of meteorological fields: applications to NESDIS operational processing. *J. Appl. Meteor.,* **34,** 3–15.
Hilbert, D., and S. Cohn-Vossen	1999	*Geometry and the Imagination.* American Mathematical Society, 357 pp.
Hirschfeld, J. W. P.	1998	*Projective geometries over finite fields (Second Ed.)* Oxford University Press. 576 pp.
Hoskins, B. J., and F. P. Bretherton	1972	Atmospheric frontogenesis models: mathematical foundation and solution. *J. Atmos. Sci.,* **29,** 11–37.
Lewis, J. M., S. Lakshmivarahan, and S. Dhall	2006	*Dynamic Data Assimilation.* Cambridge University Press (to appear).
Otte, T. L., N. L. Seaman and D. R. Stauffer	2001	A heuristic study on the importance of anisotropic error distributions in data assimilation. *Mon. Wea. Rev.,* **129,** 766–783.
Purser, R. J.	2004	A scheme for the characterization and synthesis of anisotropic background error covariances suitable for adaptive variational assimilation. *(Preprint) AMS 20th Conference on Weather Analysis and Forecasting and 16th Conference on Numerical Weather Prediction,* 12th–15th January 2004, Seattle, WA.
Purser, R. J., and L. M. Leslie	1991	An efficient interpolation procedure for high-order three-dimensional semi-Lagrangian models. *Mon. Wea. Rev.,* **119,** 2492–2498.
Purser, R. J., and R. McQuigg	1982	A successive correction analysis scheme using recursive numerical filters. Met. O 11 Tech. Note, No. 154, British Meteorological Office. 17 pp.
Purser, R. J., W.-S. Wu, D. F. Parrish, and N. M. Robert	2003a	Numerical aspects of the application of recursive filters to variational statistical analysis. Part I: Spatially homogeneous and isotropic Gaussian covariances. *Mon. Wea. Rev.,* **131,** 1524–1535.
Purser, R. J., W.-S. Wu, D. F. Parrish, and N. M. Roberts	2003b	Numerical aspects of the application of recursive filters to variational statistical analysis. Part II: Spatially inhomogeneous and anisotropic general covariances. *Mon. Wea. Rev.,* **131,** 1536–1548.
Rockafellar, R. T.	1996	*Convex Analysis*, Princeton, New Jersey. 469 pp.
Schubert, W. H.	1985	Semigeostrophic theory. *J. Atmos. Sci.,* **42,** 1770–1772.
Singer, J.	1938	A theorem in finite projective geometry and some applications to number theory. *Trans. Amer. Math. Soc.,* **43,** 377–385.
Thiébaux, H. J., and M. A. Pedder	1987	*Spatial Objective Analysis.* Academic Press, London.
van Tiel, J.	1984	*Convex Analysis; An Introductory Text.* John Wiley and sons.
Weaver, A., and P. Courtier	2001	Correlation modelling on the sphere using a generalized diffusion equation. *Quart. J. Roy. Meteor. Soc.,* **127,** 1815–1846.

Weaver, A. T., and S. Ricci 2004 Constructing a background-error correlation model using a generalized diffusion operator. In *Proceedings of the ECMWF Seminar Series on "Recent developments in atmospheric and ocean data assimilation."* ECMWF, Reading, U.K., 8–12 September 2003, pp. 327–340.

Wu, W.-S., R. J. Purser, and D. F. Parrish 2002 Three-dimensional variational analysis with spatially inhomogeneous covariances. *Mon. Wea. Rev.,* **130,** 2905–2916.

Ziman, J. M. 1979 *Principles of the Theory of Solids* Cambridge, 448 pp.