Projective geometric algebra as an enabler of FWI with sparse acquisitions and targeted updating

Kris Innanen

ABSTRACT

Geometrica Algebra and Projective Geometric Algebra (PGA) are alterations of our common vector algebra and its theoretical and computational elements. In principle, a PGA formulation opens the adjoint-state gradient calculation to interpretation and understanding which is strongly geometric in nature, which may on its own be valuable. But, more pragmatically, the approach encourages us to compute and think in terms of the row-spaces of the linear systems within the method. Given computational systems able to efficiently compute wedge products, this may permit us to capitalize on a lot of redundant computations normally involved in doing FWI – if our inverse problems are based on sparse acquisition, and targeted towards fractions of the total elements of a model vector.

INTRODUCTION

Geometric algebra (it is probably not an exaggeration to say) is taking large parts of the world of theoretical science by storm. It is an alternative form of linear algebra, which corrects many of the shortcomings of standard vector algebra, based on several non-standard algebra formulations due to Grassmann and Clifford. Two very readable introductions to the subject are by Doran and Lasenby (2007) and Macdonald (2023). It has been of particular value in image processing and graphics, but it seems to be on the path to becoming many physicists' favourite mathematical language within which to formulate any highly geometric or visualizable theory.

Here we will consider its use in the context of Projective Geometric Algebra (PGA), which is a slight extension permitting objects like lines, points, planes, and hyperplanes to be encoded within it and manipulated. To the extent that our adjoint-state treatments of gradient calculations in full waveform inversion have geometric roots, we might wonder what happens when we attempt a treatment of FWI in PGA terms.

My initial conclusions are mostly related to computational repetitiveness in FWI, and the possibility that, if we can access the operations of PGA within computational systems / architectures, we might be able to capitalize on its innate ability to separate out contributions to solutions of linear equations, and avoid repetitive calculations to our advantage. This is, of course, only a very early conclusion and still must be investigated much more intensively.

AN INCOMPLETE REVIEW OF PROJECTIVE GEOMETRIC ALGEBRA

Geometric Algebra

Let the orthogonal unit vectors $\mathbf{e}_i$, $i = 1, ..., N$ be a "canonical basis" for $\mathbb{R}^N$. Any vector in $\mathbb{R}^N$ can then be written as

$$\mathbf{v} = v_i\mathbf{e}_i = v_1\mathbf{e}_1 + v_2\mathbf{e}_2 + v_3\mathbf{e}_3, \tag{1}$$

where summation over repeated indices is implied, and where we have included the $N = 3$ case as an example, which we will continue to do when examples are useful. The dot product between two vectors $\mathbf{u}$ and $\mathbf{v}$ is

$$\mathbf{u} \cdot \mathbf{v} = (u_i\mathbf{e}_i) \cdot (v_j\mathbf{e}_j) = (u_1\mathbf{e}_1 + u_2\mathbf{e}_2 + u_3\mathbf{e}_3) \cdot (v_1\mathbf{e}_1 + v_2\mathbf{e}_2 + v_3\mathbf{e}_3). \tag{2}$$

Distributing, and using the rules $\mathbf{e}_i \cdot \mathbf{e}_j = \mathbf{e}_j \cdot \mathbf{e}_i = \delta_{ij}$, where δ_{ij} is the Kronecker delta, we obtain the familiar form

$$\mathbf{u} \cdot \mathbf{v} = u_iv_i = u_1v_1 + u_2v_2 + u_3v_3. \tag{3}$$

Geometrically we have that the dot product is a scalar of size

$$|\mathbf{u}||\mathbf{v}| \cos\theta, \tag{4}$$

where θ is the angle between the two vectors. We now add to the dot product a second product called the *wedge product*:

$$\mathbf{u} \wedge \mathbf{v} = (u_i\mathbf{e}_i) \wedge (v_j\mathbf{e}_j) = (u_1\mathbf{e}_1 + u_2\mathbf{e}_2 + u_3\mathbf{e}_3) \wedge (v_1\mathbf{e}_1 + v_2\mathbf{e}_2 + v_3\mathbf{e}_3). \tag{5}$$

The wedge product between the canonical basis vectors satisfies

$$\mathbf{e}_i \wedge \mathbf{e}_j = \begin{cases} -\mathbf{e}_i \wedge \mathbf{e}_j & i \neq j \\ 0 & i = j \end{cases}, \tag{6}$$

so, upon distributing, we have

$$\mathbf{u} \wedge \mathbf{v} = (u_iv_j - u_jv_i)\mathbf{e}_i \wedge \mathbf{e}_j, \tag{7}$$

over the restricted index values $i = 1, ..., N - 1$, and $j = i + 1, ..., N$. In the $N = 3$ case this is

$$\mathbf{u} \wedge \mathbf{v} = (u_1v_2 - u_2v_1)\mathbf{e}_1 \wedge \mathbf{e}_2 + (u_1v_3 - u_3v_1)\mathbf{e}_1 \wedge \mathbf{e}_3 + (u_2v_3 - u_3v_2)\mathbf{e}_2 \wedge \mathbf{e}_3. \tag{8}$$

The quantity $\mathbf{u} \wedge \mathbf{v}$ is not a vector, but a particular kind of *bivector*. It is characterized by its magnitude, which is the area of the parallelogram formed by the two vectors within the product:

$$|\mathbf{u} \wedge \mathbf{v}| = |\mathbf{u}||\mathbf{v}| \sin\theta, \tag{9}$$

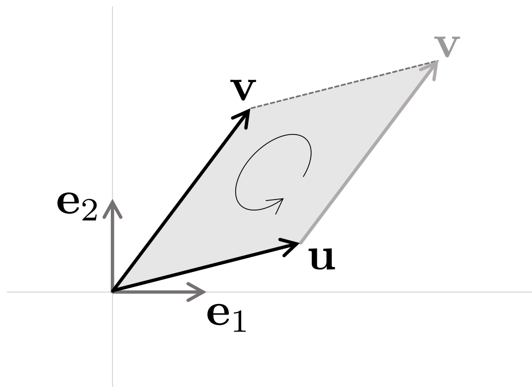

FIG. 1. Geometric interpretation of the wedge product.

and by its sign, which reflects the sense of the two vectors "around" the parallelogram (Figure 1). Notice that the algebraic rule $\mathbf{e}_i \wedge \mathbf{e}_j = -\mathbf{e}_j \wedge \mathbf{e}_i$, which is transferred to general vectors $\mathbf{u} \wedge \mathbf{v} = -\mathbf{v} \wedge \mathbf{u}$ gives the assignment of positive or negative to the object strong justification (if $\mathbf{u}$ and $\mathbf{v}$ are assembled in reverse order to form the parallelogram in Figure 1, the sense reverses also).

The key step in GA is to unite the dot and wedge products into a single operation, which is a third type of multiplication or composition between elements of the GA, referred to as the *geometric product*:

$$\mathbf{uv} = \mathbf{u} \cdot \mathbf{v} + \mathbf{u} \wedge \mathbf{v}. \tag{10}$$

This form tends to inspire discomfort, as it seems less than consistent to add a scalar to a bivector. The originators of geometric algebra acknowledge this, and encourage us to think of the sum similarly to how we think of a complex number, $z = a + ib$, which solves a wide range of important mathematical problems, but, like the geometric product, appears at first blush to be the sum of objects that are not commensurable.

In our current paper, the geometric product actually will not play too important a role, though as the general idea of applying GA and PGA in FWI and applied geophysics continues to be explored, it must be expected to begin to play a central role. For the moment, we can see some immediate consequences when it is applied to the canonical basis:

$$\mathbf{e}_i\mathbf{e}_j = \mathbf{e}_i \cdot \mathbf{e}_j + \mathbf{e}_i \wedge \mathbf{e}_j = \begin{cases} \mathbf{e}_i \wedge \mathbf{e}_j, & i \neq j \\ 1, & i = j \end{cases} \tag{11}$$

For our purposes in this paper, which will mainly be repeatedly computing wedge products, it means we are free to replace $\mathbf{e}_i \wedge \mathbf{e}_j$ with the geometric product $\mathbf{e}_i\mathbf{e}_j$. In fact, for notational convenience, we will adopt the following convention:

$$\mathbf{e}_{ij} = \mathbf{e}_i\mathbf{e}_j = \mathbf{e}_i \wedge \mathbf{e}_j. \tag{12}$$

By virtue of the elegance of GA, it turns out that

$$\mathbf{e}_{ij},\ i = 1, ..., N-1,\ j = i+1, ..., N, \tag{13}$$

is a basis for bivectors $\mathbf{u}\wedge\mathbf{v}$, if the $\mathbf{e}_i$ are a basis for the underlying vectors $\mathbf{u}$ and $\mathbf{v}$. That is, equation (8) can be considered a general expansion of the bivector $\mathbf{u}\wedge\mathbf{v}$ over the bivector basis $\mathbf{e}_1\mathbf{e}_2$, $\mathbf{e}_1\mathbf{e}_3$, and $\mathbf{e}_2\mathbf{e}_3$.

The process can continue upward in dimensionality. A *trivector* is the sequential wedge product of three vectors:

$$\mathbf{u}\wedge\mathbf{v}\wedge\mathbf{w} = (u_i\mathbf{e}_i)\wedge(v_j\mathbf{e}_j)\wedge(w_k\mathbf{e}_k). \tag{14}$$

Expanding the vectors, distributing, and applying the dot and wedge product rules to the canonical basis vectors, we find in the 3D case (though not in higher dimensions) that all but one of them are zero:

$$\mathbf{u}\wedge\mathbf{v}\wedge\mathbf{w} = \big[w_1(u_2v_3 - u_3v_2 - w_2(u_1v_3 - u_3v_1) + w_3(u_1v_2 - u_2v_1)\big]\mathbf{e}_{123}, \tag{15}$$

where we have continued the notational practice of setting $\mathbf{e}_{ijk} = \mathbf{e}_i\mathbf{e}_j\mathbf{e}_k$. Undertaking the process of distributing the components and the bases, and examining the combinations that emerge, we find that rules applying to the notationally simple $\mathbf{e}_{ijk}$ suggest themselves. Specifically, any repeated index, e.g., within which $\mathbf{e}_{113}$, produces a zero contribution, and any two indices may be switched at the cost of a minus sign, e.g., $\mathbf{e}_{123} = -\mathbf{e}_{132}$.

This process of sequential computation of wedge products can be continued, until the number of vectors being so combined equals the dimension of the canonical vector space. In the example in (15) being three 3D vectors, we have reached the end of the line. In 3D, the trivector is also called the *pseudoscalar*, because once the dimension (3) has been reached, the result has a single component, and bears some resemblance to a scalar.

In 3D, there is, therefore, an expanded set of basis objects. Scalars (via the first term in the geometric product) are admitted, and so in total we have:

$$\begin{aligned} &1: \quad \text{scalar} \\ &\mathbf{e}_1,\ \mathbf{e}_2,\ \mathbf{e}_3: \quad \text{vector} \\ &\mathbf{e}_{12},\ \mathbf{e}_{13},\ \mathbf{e}_{23}: \quad \text{bivector} \\ &\mathbf{e}_{123}: \quad \text{pseudoscalar.} \end{aligned} \tag{16}$$

The same rules applied to spaces generated by vector spaces of higher dimension produce more complex sets of basis elements.

Projective Geometric Algebra

The above review covers a negligible fraction of the subject of GA, but the key aspects allowing the work in this paper to proceed are in place. We might next ask a few questions about applying GA to linear systems of equations, which to be sure are very concerned with geometry; Strang (1988), for instance, introduces the subject of linear algebra by discussing the vector spaces associated with the rows and columns of matrices, and how they can be used to understand solutions of linear systems.

For this purpose, a special type of GA, called Projective Geometric Algebra (PGA), appears to be of particular use. It offers the opportunity for us to in fact go against Strang's clear admonition to focus on the column spaces of linear systems, and instead pursue insights deriving from the row picture. Strang's suggestion derives from the interpretability of the column picture; however, if PGA offers a similar intuitive/interpretive scheme applicable to the row picture, then it may be worth pursuing.

The geometry of the row picture of a linear system

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{17}$$

is the geometry of linear equations. Suppose that $\mathbf{x}$ is a vector in a space $\mathbb{R}^N$. Absent any of the information in the system (17), all points in this N-dimensional space are valid possibilities. The first row of (17), which is a linear equation in $x_1, x_2, ..., x_N$, defines an $N-1$ subspace of $\mathbb{R}^N$. We can call this a *hyperplane*, since if $N = 3$ the locus of points satisfying that single linear equation is a plane in 3D. All rows map similarly to hyperplanes. Each additional equation further reduces the possible N-tuplets that could be the solution we seek. We can understand this reduction in terms of the intersection of hyperplanes. In the $N = 3$ case, the first two equations in a 3×3 system correspond to two planes, whose line of intersection comprises the possible triplets (x_1, x_2, x_3) which solve the full system. The intersection of all of these hyperplanes is a unique point in $\mathbb{R}^N$, corresponding to the solution. In the $N = 3$ case, the line comprising the first two equations pierces the plane associated with the 3rd equation at the solution.

The difficulty in applying a GA to this problem lies in discovering a meaningful map between vectors (and bivectors, etc.) to lines, planes, and hyperplanes. Let us consider the $N = 3$ case specifically. A mapping from 3D GA to planes and lines cannot be as simple as identifying a 3D vector or bivector to a line, or to a plane. This is because vectors (and bivectors etc.) are insensitive to translations around in their respective spaces*. However, lines and planes are sensitive to such translations: the idea of a y-intercept very obviously distinguishes between two lines that have been translated relative to one another.

PGA overcomes this difficulty, and allows spaces describing lines, planes, and their N-dimensional counterparts (i.e., hyperplanes) by the elegant step of introducing an additional dimension. An $N+1$ dimensional geometry, properly formulated, turns out to be sufficient for this purpose. The requisite PGA starts with N canonical basis vectors

$$\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_N, \tag{18}$$

satisfying the usual conditions $\mathbf{e}_i \cdot \mathbf{e}_j = \delta_{ij}$, and then tacks on a special $N+1$'th basis vector

$$\mathbf{e}_{N+1}, \tag{19}$$

with the special attribute that $\mathbf{e}_i \cdot \mathbf{e}_{N+1} = 0$, including the case $i = N+1$. Apart from this oddity, the new canonical basis

$$\mathbf{e}_i, \;\; i = 1, 2, ..., N+1, \tag{20}$$

*This freedom is how we actually create the parallelograms needed to compute vector sums, for instance.

then taken through sequential wedge products, generates a perfectly valid $N + 1$ dimensional algebra.

Let us see with an example how the geometry of the PGA description plays out. Suppose we have the 2×2 system

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}. \tag{21}$$

We can quickly write out the solution of this system:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{1}{D} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \tag{22}$$

or

$$\begin{aligned} Dx_1 &= a_{22}b_1 - a_{12}b_2 \\ Dx_2 &= a_{11}b_2 - a_{21}b_1, \end{aligned} \tag{23}$$

where $D = a_{11}a_{22} - a_{12}a_{21}$ is the determinant. Knowing that the point (x_1, x_2) represents the intersection of the 2D hyperplanes[†] $a_{11}x_1 + a_{12}x_2 = b_1$ and $a_{21}x_1 + a_{22}x_2 = b_2$, let us see how PGA treats the problem. Construct 2 vectors in the $N + 1 = 3$ dimensional space with the basis $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$ as follows:

$$\begin{aligned} \mathbf{A}_1 &= a_{11}\mathbf{e}_1 + a_{12}\mathbf{e}_2 - b_1\mathbf{e}_3 \\ \mathbf{A}_2 &= a_{21}\mathbf{e}_1 + a_{22}\mathbf{e}_2 - b_2\mathbf{e}_3, \end{aligned} \tag{24}$$

i.e., with the elements of the rows of the matrix $\mathbf{A}$, and the rows of $\mathbf{b}$, forming the components of the vectors. In PGA, these $N+1$ dimensional vectors represent $N-1$ dimensional hyperplanes, which live in the N-dimensional space.

In PGA, the intersections of objects like $\mathbf{A}_1$ and $\mathbf{A}_2$ are called *meets*. They are computed through wedge products in this specialized PGA. The wedge product of $\mathbf{A}_1$ and $\mathbf{A}_2$ should, then, according to this logic, be connected with the solution. Let us check:

$$\begin{aligned} \mathbf{A}_1 \wedge \mathbf{A}_2 &= (a_{11}\mathbf{e}_1 + a_{12}\mathbf{e}_2 - b_1\mathbf{e}_3) \wedge (a_{21}\mathbf{e}_1 + a_{22}\mathbf{e}_2 - b_2\mathbf{e}_3) \\ &= (a_{11}a_{22} - a_{12}a_{21})\mathbf{e}_{12} + (a_{21}b_1 - a_{11}b_2)\mathbf{e}_{13} + (a_{22}b_1 - a_{12}b_2)\mathbf{e}_{23} \\ &= D\mathbf{e}_{12} - x_2\mathbf{e}_{13} + x_1\mathbf{e}_{23}. \end{aligned} \tag{25}$$

The solution emerges in an unusual but easily-characterizable way. The basis vector *absent* the $N+1$th index, i.e., $\mathbf{e}_{12}$ has as a coefficient the determinant of the matrix. The two basis vectors which *do* contain the $N + 1$th index contain the terms from the adjugate matrix. The solution is organized such that the index of the solution (e.g., the 1 in x_1), is the index missing from the basis element (e.g., $\mathbf{e}_{23}$) it is associated with. Sporadic negative signs can appear as well, which must be accounted for when setting up computational schemes along these lines.

[†]Otherwise known as lines.

This generalizes, with general wedge products in $N+1$ dimensional PGA,

$$\mathbf{A}_i \wedge \mathbf{A}_j, \tag{26}$$

corresponding to the *meet* or intersection of the two objects. This also applies to wedge products between any other PGA objects. For instance, if $\mathbf{A}_i \wedge \mathbf{A}_j$ represents a hyperplane in $N+1$ dimensional PGA, and $\mathbf{A}_k$ represents another of different dimension, then

$$(\mathbf{A}_i \wedge \mathbf{A}_j) \wedge \mathbf{A}_k = \mathbf{A}_i \wedge \mathbf{A}_j \wedge \mathbf{A}_k \tag{27}$$

represents the intersection or meet of those two generalized objects. Thus,

$$\mathbf{A}_1 \wedge \mathbf{A}_2 \wedge ... \wedge \mathbf{A}_N \tag{28}$$

contains, with a structure similar to (25), the solution of the N dimensional system $\mathbf{Ax} = \mathbf{b}$, with the $\mathbf{A}_i$ being

$$\mathbf{A}_i = a_{i1}\mathbf{e}_1 + ... + a_{iN}\mathbf{e}_N - b_i\mathbf{e}_{N+1}. \tag{29}$$

Though (like in the GA discussion) we have only set out the most basic features of PGA, we do now have enough to discuss its impact on FWI and gradient calculations.

COMPUTATION OF THE GRADIENT VIA THE ADJOINT STATE METHOD

Let ϕ be the scalar objective function derived from the squared error between computed ($\mathbf{d}$) and observed ($\mathbf{d}^o$) data,

$$\phi = \frac{1}{2}||\mathbf{d} - \mathbf{d}^o||_2^2. \tag{30}$$

This is typically summed over some set of experimental variables *ev*, such as source position and frequency. For simplicity, we proceed in terms of this ϕ, which can be thought of as the full objective function per unit *ev*. Let the data be samples of the field $\mathbf{u}$:

$$\mathbf{Ru} = \mathbf{d}, \tag{31}$$

which (in frequency-domain FWI) itself satisfies a matrix equation derived from discretization of the Helmholtz equation,

$$\mathbf{Su} = \mathbf{f}, \tag{32}$$

where $\mathbf{S}$ is the impedance matrix. The adjoint-state method gives the vector gradient $\mathbf{g}$ of ϕ, the core of the updating procedure in FWI, as

$$\mathbf{g} = \frac{d\phi}{d\mathbf{m}} = -\boldsymbol{\lambda}^T \frac{\partial \mathbf{S}}{\partial \mathbf{m}}\mathbf{u}. \tag{33}$$

The wave field solution vector $\mathbf{u}$, which is of length N, contains what in adjoint-state terminology are referred to as the "state variables"; we will refer to it as the *forward state*, and $\boldsymbol{\lambda}$ as the *adjoint state*. For convenience we distinguish between two similar systems:

$$\mathbf{Au} = \mathbf{f}, \tag{34}$$

where $\mathbf{A} = \mathbf{S}$, and

$$\mathbf{B}\boldsymbol{\lambda} = \mathbf{r}, \tag{35}$$

where $\mathbf{B} = \mathbf{S}^T$, and where $\mathbf{r} = \mathbf{d} - \mathbf{d}^o$ is the vector of residuals, or differences between the predicted and observed data.

The vectors $\mathbf{u}$, $\boldsymbol{\lambda}$, $\mathbf{f}$, and $\mathbf{r}$ are all elements of $\mathbb{C}^N$, the "wavefield space", and $\mathbf{m} \in \mathbb{R}^M$, the "model space". Let

$$\mathbf{x}_i, \ i = 1, ..., N, \tag{36}$$

be an orthonormal basis of $\mathbb{C}^N$, and

$$\mathbf{y}_i, \ i = 1, ..., M, \tag{37}$$

be an orthonormal basis of $\mathbb{R}^M$. Then the vectors associated with each space can be expanded in these bases, i.e.,

$$\mathbf{u} = u_i\mathbf{x}_i = u_1\mathbf{x}_1 + ... + u_N\mathbf{x}_N, \tag{38}$$

and likewise for $\boldsymbol{\lambda} = \lambda_i\mathbf{x}_i$, $\mathbf{f} = f_i\mathbf{x}_i$, and $\mathbf{r} = r_i\mathbf{x}_i$; note we have here made use of the summation convention for repeated indices, which we shall continue to do from now on. In the same vein we can define the components of the model vector in terms of the basis $\mathbf{y}_i$:

$$\mathbf{m} = m_\alpha\mathbf{y}_\alpha = m_1\mathbf{y}_1 + ... + m_M\mathbf{y}_M. \tag{39}$$

If we next introduce a tensor product $\mathbf{x}_i\mathbf{x}_j$ defined such that

$$\begin{aligned} &\mathbf{x}_k \cdot (\mathbf{x}_i\mathbf{x}_j) = \delta_{ki}\mathbf{x}_j, \text{ and} \\ &(\mathbf{x}_i\mathbf{x}_j) \cdot \mathbf{x}_k = \mathbf{x}_i\delta_{jk}, \end{aligned} \tag{40}$$

then $\mathbf{S}$ can be expressed in terms of components s_{ij}, $i, j = 1, ..., N$, as

$$\mathbf{S} = s_{ij}\mathbf{x}_i\mathbf{x}_j. \tag{41}$$

This is then also true for $\mathbf{A} = a_{ij}\mathbf{x}_i\mathbf{x}_j$ and $\mathbf{B} = b_{ij}\mathbf{x}_i\mathbf{x}_j$. In general the s_{ij} depend on the model $\mathbf{m}$, and so we can define a derivative matrix for each element of $\mathbf{m}$:

$$\frac{\partial s_{ij}}{\partial m_\alpha} = s_{ij,\alpha}. \tag{42}$$

Putting these quantities together, we can re-write the standard form of the gradient derived from the adjoint-state method as

$$g_\alpha = -(\lambda_i\mathbf{x}_i) \cdot (s_{jk,\alpha}\mathbf{x}_j\mathbf{x}_k) \cdot (u_l\mathbf{x}_l). \tag{43}$$

SOLVING FOR THE FORWARD STATE WITHIN A PGA

We now set up a projective geometric algebra $\mathbb{P}^{N+1}$, starting with a canonical vector basis

$$\mathbf{e}_i,\ i = 1, ..., N + 1. \tag{44}$$

Let us identify with the first N of these $\mathbf{e}_i$ the orthonormal set $\mathbf{x}_i$ used in the previous section: $\mathbf{e}_i = \mathbf{x}_i$, $i = 1, ..., N$. The single remaining $N + 1$th basis vector, $\mathbf{e}_{N+1}$, is specially defined, following standard PGA theory, such that

$$\mathbf{e}_{N+1} \cdot \mathbf{e}_{N+1} = 0. \tag{45}$$

This basis allows us now to define N vectors in $\mathbb{P}^{N+1}$, chosen to match with the N linear equations embodied in $\mathbf{Au} = \mathbf{f}$. Specifically, let

$$\mathbf{A}_i = a_{i1}\mathbf{e}_1 + a_{i2}\mathbf{e}_2 + ... + a_{iN}\mathbf{e}_N - f_i\mathbf{e}_{N+1}, \tag{46}$$

for $i = 1, ..., N$. The first N coefficients are the elements of the ith row of $\mathbf{A}$, and the $N + 1$th coefficient is the negative of the ith element of $\mathbf{f}$. It will soon be convenient to hide the distinction between the matrix coefficients a and the source components f in these expansions. Let $A_{ij} = a_{ij}$ for $j = 1, ..., N$, and $A_{ij} = -f_j$ for $j = N + 1$:

$$\mathbf{A}_i = A_{i1}\mathbf{e}_1 + A_{i2}\mathbf{e}_2 + ... + A_{iN}\mathbf{e}_N + A_{iN+1}\mathbf{e}_{N+1} = A_{ij}\mathbf{e}_j \tag{47}$$

These vectors, each containing information from a single constraint equation on an N dimensional space, represent hyperplanes which are subspaces of $\mathbb{C}^N$ of dimension $N - 1$.

We can then combine them in wedge products, recalling that the wedge product of two objects in PGA generates their *meet*, or intersection. Let the meet between the hyperplanes $\mathbf{A}_i$ and $\mathbf{A}_j$ be $\mathbf{A}_{ij}$, where

$$\mathbf{A}_{ij} = \mathbf{A}_i \wedge \mathbf{A}_j = (A_{i1}\mathbf{e}_1 + ...A_{iN+1}\mathbf{e}_{N+1}) \wedge (A_{j1}\mathbf{e}_1 + ... + A_{jN+1}\mathbf{e}_{N+1}). \tag{48}$$

Distributing terms in this product, and making use of $\mathbf{e}_i \wedge \mathbf{e}_i = 0$, and $\mathbf{e}_i \wedge \mathbf{e}_j = -\mathbf{e}_j \wedge \mathbf{e}_i$, we obtain

$$\mathbf{A}_{ij} = A^{ij}_{12}\mathbf{e}_{12} + A_{ij}A_{13}\mathbf{e}_{13} + ... + A^{ij}_{NN+1}\mathbf{e}_{NN+1} = A^{ij}_{kl}\mathbf{e}_{kl}, \tag{49}$$

where $\mathbf{e}_{kl} = \mathbf{e}_k \wedge \mathbf{e}_l$ and

$$A^{ij}_{kl} = A_{ik}A_{jl} - A_{il}A_{jk}. \tag{50}$$

There are as many terms in this expansion as there are pairs of indices ranging from 1 to $N + 1$ in which the second index is greater than the first, which is $(N + 1)(N + 2)/2$. This is because the wedge product contribution from terms with equal indices is zero, and the contributions from pairs in which the second is greater can have their indices switched using $\mathbf{e}_i \wedge \mathbf{e}_j = -\mathbf{e}_j \wedge \mathbf{e}_i$. This is why each expansion coefficient in (50) has one positive and one negative contribution. The

$$\mathbf{e}_{ij},\ i = 1, ..., N,\ j = i, ..., N + 1, \tag{51}$$

form a basis for the bivectors of this PGA. The bivector $\mathbf{A}_{ij}$ has assembled the information contained in the ith and jth linear equations in the system $\mathbf{Au} = \mathbf{f}$, by forming their meet. It represents a lower-dimensional hyperplane, within which we know the solution $\mathbf{u}$ must lie.

This process, in which row information is combined through computation of the meets of PGA elements, can continue in many different possible ways from this point. The bivectors can be combined with individual vectors, i.e., rows, to form trivectors; or, other pairs of rows realized as bivectors can be combined, producing 4-vectors, etc. When all of the vectors have been combined into a single $N + 1$-vector, by PGA rules we have obtained:

$$\mathbf{A}_{ij...} = A_1\mathbf{E}_1 + ... + A_N\mathbf{E}_N + D\mathbf{E}_{N+1}, \tag{52}$$

where the $\mathbf{E}_i$, $i = 1, ..., N + 1$ are a compact form for the $(N + 1)$-vector basis, corresponding to wedge products of a subset of N of the $N+1$ canonical basis vectors $\mathbf{e}_j$. Thus, in any one of the $\mathbf{E}_i$, one and only one of the possible $\mathbf{e}_j$ vectors is missing. We find it convenient to label the $\mathbf{E}_i$ such that i represents the *missing* index, rather than the long list of indices which are present. For instance,

$$\begin{aligned} \mathbf{E}_1 &= \mathbf{e}_2 \wedge \mathbf{e}_3 \wedge ... \wedge \mathbf{e}_N \wedge \mathbf{e}_{N+1} = \mathbf{e}_{2345...}, \\ \mathbf{E}_2 &= \mathbf{e}_1 \wedge \mathbf{e}_3 \wedge ... \wedge \mathbf{e}_N \wedge \mathbf{e}_{N+1} = \mathbf{e}_{1345...}, \end{aligned} \tag{53}$$

etc. Since we are combining (in ascending order of indices as before) $N + 1$ basis vectors, in combinations that can be at most N elements in length, there are exactly $N + 1$ such combinations, which defines the number of components of the "final" $(N + 1)$-vector.

According to PGA theory, the coefficients A_i in (52) correspond with the solutions of the system $\mathbf{Au} = \mathbf{f}$. This follows geometrically. We understand from elementary linear algebra that the solution $\mathbf{u}$ is the point of intersection in $\mathbb{C}^N$ of all of the hyperplanes defined by the rows in the matrix equation. Thus in PGA the meet of all of the vectors (hyperplanes) simultaneously must correspond to the solution. However, the solution vector $\mathbf{u}$ has N components, and (we find) the number of components of $\mathbf{A}$ in (52) is $N + 1$; indeed, if we carry these products out numerically we also find that none of the components so produced seem to match with the solution $\mathbf{u}$.

The coefficient for the combination

$$\mathbf{E}_{N+1} = \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3 \wedge ... \wedge \mathbf{e}_{N-1} \wedge \mathbf{e}_N, \tag{54}$$

which we have called D, is the "odd element out". In fact, D is the determinant of the matrix $\mathbf{A}$. And, the remaining elements of the object $\mathbf{A}_{ij...}$, when divided by D, do then match with the values of $\mathbf{u}$.

Conclusion: by the time the full set of PGA vectors $\mathbf{A}_i$ have been combined to determine the meet of all of the original hyper-planes, we have solved the matrix equation $\mathbf{Au} = \mathbf{f}$. Referring to the gradient calculation, we have, in other words, found a PGA framework for determining the u_l in equation (43).

A 3D example

To give an explicit example, consider the 3×3 system

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}. \tag{55}$$

We can quickly generate the solution for $\mathbf{u}$ in this small system:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \frac{1}{D} \begin{bmatrix} a'_{11} & a'_{12} & a'_{13} \\ a'_{21} & a'_{22} & a'_{23} \\ a'_{31} & a'_{32} & a'_{33} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}, \tag{56}$$

or, organizing in terms of column contributions,

$$D \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = f_1 \begin{bmatrix} a'_{11} \\ a'_{21} \\ a'_{31} \end{bmatrix} + f_2 \begin{bmatrix} a'_{12} \\ a'_{22} \\ a'_{32} \end{bmatrix} + f_3 \begin{bmatrix} a'_{13} \\ a'_{23} \\ a'_{33} \end{bmatrix}, \tag{57}$$

where the a' are the elements of the adjugate matrix, and D is the determinant:

$$D = a_{11}(a_{22}a_{33} - a_{21}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31}). \tag{58}$$

The adjugate elements are, explicitly,

$$\begin{aligned} a'_{11} &= a_{22}a_{33} - a_{23}a_{32}, \; a'_{12} = a_{13}a_{32} - a_{12}a_{33}, \; a'_{13} = a_{12}a_{23} - a_{13}a_{22}, \\ a'_{21} &= a_{23}a_{31} - a_{21}a_{33}, \; a'_{22} = a_{11}a_{33} - a_{13}a_{31}, \; a'_{23} = a_{13}a_{21} - a_{11}a_{23}, \\ a'_{31} &= a_{21}a_{32} - a_{22}a_{31}, \; a'_{32} = a_{12}a_{31} - a_{11}a_{32}, \; a'_{33} = a_{11}a_{22} - a_{12}a_{21}. \end{aligned} \tag{59}$$

To further simplify the calculations, suppose that only the first of the source components, f_1, were non-zero:

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} f_1 \\ 0 \\ 0 \end{bmatrix}. \tag{60}$$

Then,

$$D \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = f_1 \begin{bmatrix} a'_{11} \\ a'_{21} \\ a'_{31} \end{bmatrix} = f_1 \begin{bmatrix} a_{22}a_{33} - a_{23}a_{32} \\ a_{23}a_{31} - a_{21}a_{33} \\ a_{21}a_{32} - a_{22}a_{31} \end{bmatrix}. \tag{61}$$

With these results in mind, let us construct the PGA vectors per the previous section:

$$\mathbf{A}_1 = a_{11}\mathbf{e}_1 + a_{12}\mathbf{e}_2 + a_{13}\mathbf{e}_3 - f_1\mathbf{e}_4, \tag{62}$$

being the only vector with an $\mathbf{e}_4$ component, and

$$\begin{aligned}\mathbf{A}_2 &= a_{21}\mathbf{e}_1 + a_{22}\mathbf{e}_2 + a_{23}\mathbf{e}_3 \\ \mathbf{A}_3 &= a_{31}\mathbf{e}_1 + a_{32}\mathbf{e}_2 + a_{33}\mathbf{e}_3.\end{aligned} \tag{63}$$

These quantities, being PGA vectors in the 4D linear space of planes, each correspond to a plane in the 3D space of the field $\mathbf{u}$. Taking the wedge product of the second two PGA vectors,

$$\mathbf{A}_{23} = \mathbf{A}_2 \wedge \mathbf{A}_3 = A_{12}^{23}\mathbf{e}_{12} + A_{13}^{23}\mathbf{e}_{13} + A_{23}^{23}\mathbf{e}_{23}, \tag{64}$$

where

$$\begin{aligned}A_{12}^{23} &= a_{21}a_{32} - a_{22}a_{31}, \\ A_{13}^{23} &= a_{21}a_{33} - a_{23}a_{31}, \text{ and} \\ A_{23}^{23} &= a_{22}a_{33} - a_{23}a_{32},\end{aligned} \tag{65}$$

we obtain, $\mathbf{A}_{23}$, the meet of two of the planes, which is a line in 3D. The significance of the absence of $\mathbf{e}_4$ components in these two planes is that each of the planes passes through the origin $(u_1, u_2, u_3) = (0, 0, 0)$, and so does their line of intersection.

We next form the meet of the line $\mathbf{A}_{23}$ and the remaining plane, which is a unique point and must represent the solution. We have

$$\mathbf{A}_{123} = \mathbf{A}_1 \wedge \mathbf{A}_{23} = (a_{11}\mathbf{e}_1 + a_{12}\mathbf{e}_2 + a_{13}\mathbf{e}_3 - f_1\mathbf{e}_4) \wedge (A_{12}^{23}\mathbf{e}_{12} + A_{13}^{23}\mathbf{e}_{13} + A_{23}^{23}\mathbf{e}_{23}).$$

To speed up the distribution of these products, we note that the rules for wedge products quickly eliminate any combination for which there are repeated indices, for instance, the first contribution $a_{11}\mathbf{e}_1 \wedge A_{12}^{23}\mathbf{e}_{12} = 0$ because $\mathbf{e}_1 \wedge \mathbf{e}_{12} = \mathbf{e}_1 \wedge \mathbf{e}_1 \wedge \mathbf{e}_2$, and in this construction the leftmost wedge product is zero. This fact alone leaves only a small number of nonzero contributions:

$$\mathbf{A}_{123} = \left(a_{11}A_{23}^{23} - a_{12}A_{13}^{23} + a_{13}A_{12}^{23}\right)\mathbf{e}_{123} + \left(f_1A_{12}^{23}\right)\mathbf{e}_{124} + \left(f_1A_{13}^{23}\right)\mathbf{e}_{134} + \left(f_1A_{23}^{23}\right)\mathbf{e}_{234}$$

Using (65) we recognize that the coefficient of $\mathbf{e}_{123}$ is D, the determinant, and that the other coefficients are elements of the adjugate column in (61), though re-ordered and with sporadic minus signs incorporated:

$$\mathbf{A}_{123} = D\mathbf{e}_{123} + (u_3)\mathbf{e}_{124} + (-u_2)\mathbf{e}_{134} + (u_1)\mathbf{e}_{234}. \tag{66}$$

Thus, the solution is obtained through progressive calculation of meets. We observe:

1. Although the components of the solution u_i are re-ordered, the re-ordering is not random. Each trivector basis has one value between 1 and 3 missing, and that missing value corresponds to the component represented.

2. The minus signs are characteristic of the PGA approach, which (similarly to eigenvector calculations) does not distinguish between a basis element and its negative. The pattern of $\pm$'s is identical for products of multivectors, and so a computational framework only needs to "calibrate" in mapping meets to solutions once.

3. Using the low-to-high ordering we have adopted here, the re-ordering can also be seen to be precisely a reversal of the ordering of the u_i: 3, 2, then 1.

4. Carrying out the products in a different order impacts the sign of the result; it can be quickly confirmed, that, e.g., $\mathbf{A}_{123} = -\mathbf{A}_{132}$, etc. However, this has no impact on the solution (which is the ratio of the remaining coefficients and the coefficient associated with the determinant, since the latter changes sign also).

Finally, moving to the capital-E terminology for the bases, expanding in terms of the original matrix elements, and re-ordering for convenience, we have

$$\mathbf{A}_{123} = f_1(a_{22}a_{33} - a_{23}a_{32})\mathbf{E}_1 + f_1(a_{21}a_{33} - a_{23}a_{31})\mathbf{E}_2 + f_1(a_{21}a_{32} - a_{22}a_{31})\mathbf{E}_3 + D\mathbf{E}_4,$$

where $\mathbf{E}_1 = \mathbf{e}_{234}$, $\mathbf{E}_2 = \mathbf{e}_{134}$, $\mathbf{E}_3 = \mathbf{e}_{124}$, and $\mathbf{E}_4 = \mathbf{e}_{123}$, i.e., the single index refers to the missing index in the associated trivector basis.

A COMPUTATIONAL SCHEME FOR PGA SOLUTION OF LINEAR SYSTEMS

We have set up a numerical scheme in Matlab for low dimensional versions of these computations to be examined. In this section we will validate the solution for the 8×8 system with a random matrix equation plotted in Figure 2. Following the PGA procedure outlined in the previous sections, we define 8 vectors in the 9 dimensional PGA space, from the rows of the matrix and vector plotted in Figure 2. These are plotted in Figure 3. We next begin to compute pairwise wedge products, each time producing the meet of the two input elements, or their intersection. The 4 $\mathbf{A}_{ij}$ bivectors are first, plotted in Figure 4. Next, the 4 bivectors are likewise combined in pairs to generate 2 4-vectors $\mathbf{A}_{ijkl}$, which are plotted in Figure 5, and these two are again engaged in a wedge product to generate a single 8-vector (i.e., $\mathbf{A}_{ijklmnop}$), as plotted in Figure 6. This 8-vector is (like the input vectors), of length $8 + 1 = 9$. In accordance with the PGA approach outlined above, the first element of this vector is the determinant D of the original matrix, and the remaining 8 components are equal to $D\times$ the solution, specially ordered and with a known set of '-' signs on some of them. Re-ordering the elements of this product, and dividing the last 8 elements by the first, we compare the result with the numerically computed $\mathbf{u} = \mathbf{S}^{-1}\mathbf{f}$ in Figure 7, confirming their correctness.

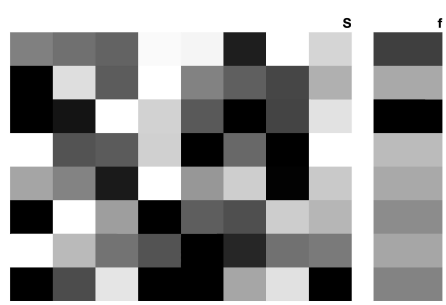

FIG. 2. Random matrix $\mathbf{S}$ and vector $\mathbf{f}$ to be solved for $\mathbf{u}$ wherein $\mathbf{S}\mathbf{u} = \mathbf{f}$.

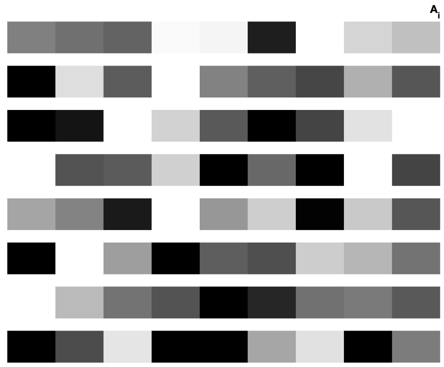

FIG. 3. $N = 8$ vectors of length $N+1 = 9$ are extracted from the matrix $\mathbf{S}$ and vector $\mathbf{f}$, representing the 8 hyperplanes which intersect at the solution.

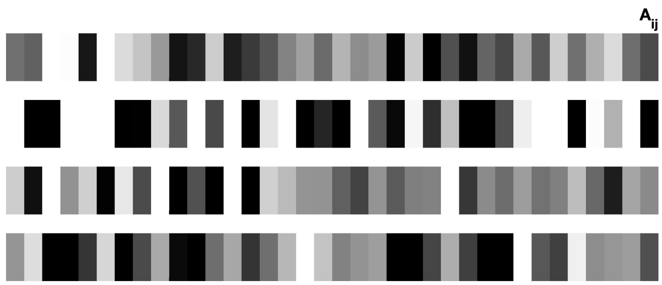

FIG. 4. The 8 vectors are combined pair-wise in wedge products to produce 4 bivectors, which represent the 4 intersections or meets of the 8 hyperplanes.

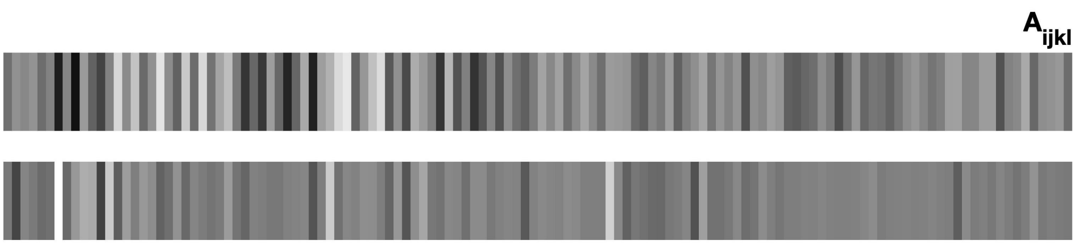

FIG. 5. The 4 bivectors are again engaged pair-wise in wedge products, producing two 4-vectors.

FIG. 6. The 2 4-vectors are, finally, combined in a wedge product to produce a single 8-vector.

FIG. 7. Re-ordering the last 8 elements of the final meet calculation, scaling them by the first element, and adjusting for several negative signs, we confirm that the product matches the numerical solution for $\mathbf{u} = \mathbf{S}^{-1}\mathbf{f}$.

EXPLOITING SPARSE ACQUISITION AND TARGETED UPDATING VIA PGA

The FWI gradient, being computed within the adjoint-state method via

$$\mathbf{g} = \frac{d\phi}{d\mathbf{m}} = -\boldsymbol{\lambda}^T \frac{\partial \mathbf{S}}{\partial \mathbf{m}} \mathbf{u}, \tag{67}$$

or in indicial notation as

$$g_\alpha = -(\lambda_i \mathbf{x}_i) \cdot (s_{jk,\alpha}\mathbf{x}_j\mathbf{x}_k) \cdot (u_l \mathbf{x}_l), \tag{68}$$

can unsurprisingly also be treated within the PGA framework, since the $\mathbf{u}$ and the $\boldsymbol{\lambda}$ are solutions to the systems $\mathbf{Su} = \mathbf{f}$ and $\mathbf{S}^T\boldsymbol{\lambda} = \mathbf{r}$, both of which can be so treated. Still, let us write the PGA form for the gradient out in full, using the terminology established in and around equations (53)-(54). We have

$$cg_\alpha = (B_i \mathbf{E}_i) \cdot (S_{jk,\alpha}\mathbf{E}_j\mathbf{E}_k) \cdot (A_l \mathbf{E}_l), \tag{69}$$

where we have *not* divided through by determinants, or included the negative sign, but allowed these to be absorbed within a constant c which we place for convenience on the left had side. The matrix elements $S_{jk,\alpha}$ have had to be adjusted slightly from those of $s_{jk,\alpha}$ in equation (68): they have been re-ordered to match with the ordering of the N-vectors $B_i\mathbf{E}_i$ and $A_l\mathbf{E}_l$, and extra zero rows and columns have been added to the top and left of the matrix respectively, in order to suppress the extra first elements of the N-vectors.

The key question we now wish to ask is: does the fact that we have sparse acquisition (i.e., few nonzero source or residual entries), and/or targeted updating in mind (i.e., few rows of the matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{S}$ which change from one iteration to another, or one inversion to another), mean that some or many of the ingredients in (69) can be computed *once*, and then re-used many times?

A PGA solution for the forward/adjoint states and the FWI gradient

Let us start by moving through the PGA gradient calculation. In Figures 8a-b two 8×8 systems involving a Helmholtz matrix $\mathbf{S}$ and source and residual vectors, matching the systems $\mathbf{Au} = \mathbf{f}$ and $\mathbf{B}\boldsymbol{\lambda} = \mathbf{r}$, where $\mathbf{A} = \mathbf{S}$ and $\mathbf{B} = \mathbf{S}^T$. In Figures 9a-b we see the rows extracted to form the basic 8 vectors of the scheme. In Figures (10)-(11) the established sequence of wedge products, or meets, are computed, leading to the two 8-vector outcomes. In Figure 12 the results are combined per equation (69) to compute the gradient.

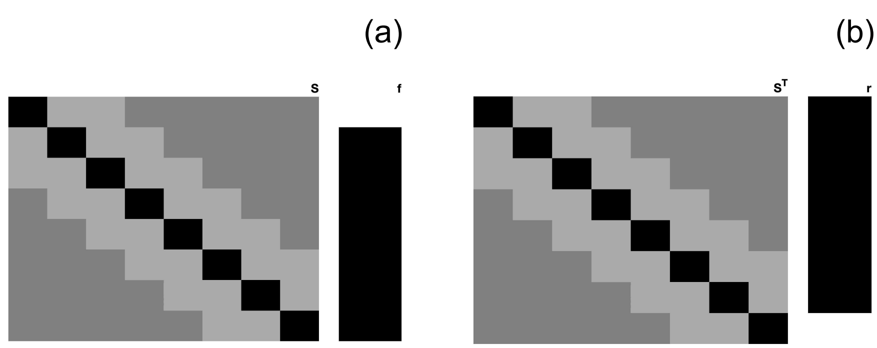

FIG. 8. Small (8×8) but perfectly valid Helmholtz systems $\mathbf{Su} = \mathbf{f}$ (a) and $\mathbf{S}^T\boldsymbol{\lambda} = \mathbf{r}$; matrix and source vector shown.

Compartmentalizing n-vectors based on acquisition and targeted updating

We can now mimic the situation where the source has changed in value, the residual has changed in value, and a targeted subset of model parameters have changed in value. Let us

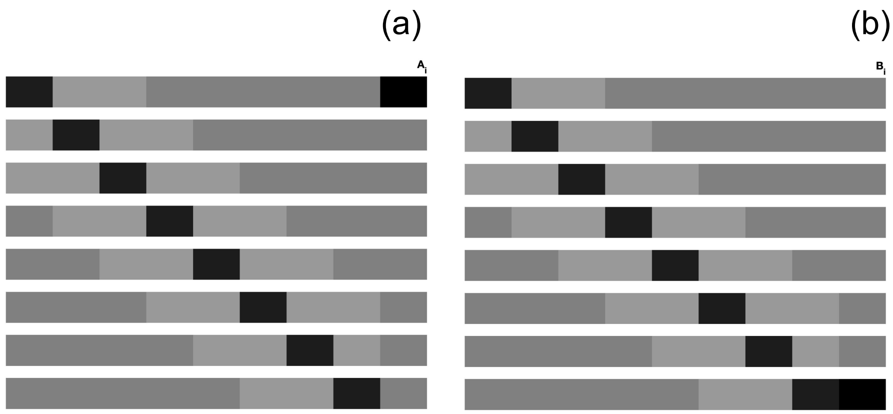

FIG. 9. Extractions to produce the basic PGA vectors for the two gradient systems.

FIG. 10. Final 8-vector for the $\mathbf{u}$ system, with re-ordered $\mathbf{u}$ solution included for comparison.

FIG. 11. Final 8-vector for the $\boldsymbol{\lambda}$ system, with re-ordered $\boldsymbol{\lambda}$ solution included for comparison.

FIG. 12. Merged to form the gradient.

perturb the inputs per Figure 13. Extracting the PGA vectors as before, and looking at their perturbations, we observe that only a small number of these undergo change. In fact, since the row vectors can be re-ordered an any desired way, with at most a cost of a minus sign incurred, we can re-order the rows such that all perturbations sit close together. Carrying this out and then forming the 2-vectors, we stop and take stock of what we have (see Figures 14-15. Evidently, roughly 1/4 of each half of the adjoint state gradient calculation are impacted when 1/8 of the model parameters are focused on, for a total of 1/8 of a change. The pattern is suggestive of generalization - with sparse acquisition, a targeting of P% of the model parameters evidently leads to at most P% of the resulting PGA quantities requiring recalculation. That is, at least (100-P)% of the quantities used in the gradient computation are straightforwardly re-usable.

CONCLUSIONS

The PGA formulation opens the adjoint-state gradient calculation to interpretation and understanding which is strongly geometric in nature, which may on its own be valuable. But, more pragmatically, the approach encourages us to compute and think in terms of the

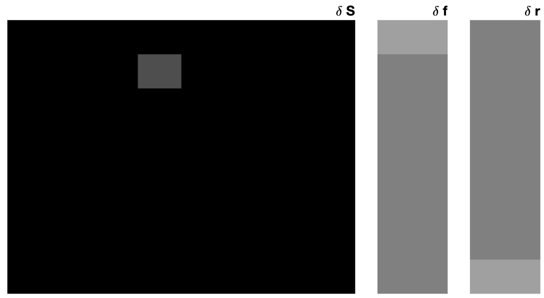

FIG. 13. Changes in the systems ($\mathbf{S}$, $\mathbf{f}$ and $\mathbf{r}$) mimicking targeting and sparseness.

FIG. 14. Changes due to sparseness in 1/4 of computed elements.

FIG. 15. Changes due to sparseness in 1/4 of computed elements.

row-spaces of the linear systems within the method. Given computational systems able to efficiently compute wedge products, this may permit us to capitalize on a lot of redundant computations normally involved in doing FWI – if our inverse problems are based on sparse acquisition, and targeted towards fractions of the total elements of a model vector.

ACKNOWLEDGEMENTS

The sponsors of CREWES are gratefully thanked for continued support. This work was funded by CREWES industrial sponsors, NSERC (Natural Science and Engineering Research Council of Canada) through the grant CRDPJ 543578-19, and in part by an NSERC-DG.

REFERENCES

Doran, C., and Lasenby, A., 2007, Geometric Algebra for Physicists: Cambridge University Press, 1st edn.

Macdonald, A., 2023, Linear and Geometric Algebra: Luther College, 1st edn.

Strang, G., 1988, Linear Algebra and its Applications: Harcourt Brace & Company, 3rd edn.