BRINGING ENGINEERS TO GEOMETRIC ALGEBRA (AND VICE VERSA)

D. Storti[a]

[a]Department of Mechanical Engineering
University of Washington-Seattle
Seattle, WA, USA
storti@uw.edu

Summary of the Abstract

This paper describes a strategic plan aimed at achieving significant adoption of geometric algebra in engineering, particularly in mechanical engineering (ME), at the university level. The target engineering audience includes three significant sub-groups: faculty, graduate students, and undergraduate students. The plan includes creation and presentation of material aimed at each sub-group: a specially-designed introductory seminar aimed at generating faculty interest, a project-based class aimed at graduate students entering the research environment, and a re-imagined version of introductory mechanics (statics & dynamics) courses aimed at students in the early undergraduate years. In each case, the goal is to open with an amazing demonstration of the power of GA sufficient to inspire the willingness to listen to and engage with a brief introduction to GA basics and functionality aimed at showing how GA makes the amazing entirely plausible. At that point, the audience is hopefully ready to move on to more significant engagement with GA. This paper focuses on plans for engaging audiences of faculty and graduate students, and the plans for getting undergraduates to engage with geometric algebra will be the subject of future works. The discussion presented here includes some personal experiences on the road to developing this plan for bringing geometric algebra to MEs.

Introduction

This paper aims to address the challenge of teaching and preaching geometric algebra (GA) to a particular audience: mechanical engineers (MEs) in a university setting. While this sounds like a very specific task, there are actually three separate subtasks associated with different audience segments including undergraduate students, graduate students, and faculty. In each case, the working supposition is that getting any substantial portion of the specific target audience to adopt GA requires connecting to that target audience where they are. In other words, bringing MEs to GA depends on bringing GA to them. In the remainder of the paper, we discuss suitable plans for connecting GA to the audience subgroups with emphasis on faculty and graduate students.

Advanced Audiences: Faculty & Graduate Students

Getting through to academic colleagues presents the most crucial and most challenging task: "most challenging" because members of the faculty typically have years of experience with traditional linear algebra, are quite comfortable with matrix-vector formulations, and are all busily engaged in their own research and teaching specialties; and "most essential" because if faculty are not persuaded to incorporate GA, then students will not see GA throughout the curriculum and will not see GA as an inherent and essential component of the design and analysis tasks at the core of engineering. Because of the time constraints, winning typical engineering faculty over to GA requires bringing GA to them and doing so in a very efficient manner and as part of an activity in which they are already engaged so they do not feel like they are going out of their way. Based on my experience with faculty schedules, this means that the delivery of GA must be compact enough that it can be packaged as a single one-hour seminar. Of course, full coverage cannot be achieved on such a short time scale, but it should be feasible to present the following 5-step plan:

(1) Demonstrate GA in action doing something the audience can immediately recognize, but in a way that appears amazing enough to capture their attention. The opening example needs to be simultaneously familiar and magical. [1] (2) Provide quick context (historical and technical) for GA. (3) Illustrate the basics of GA with necessary equations and as many pictures (or, better yet, animations) as possible until the opening "magic" seems plausible. (4) Present a list of areas where GA has already been applied to achieve advances. (5) Provide access to materials that enable the target audience to efficiently learn GA and how to incorporate GA into the curriculum.

Such a seminar plays a dual role of "preaching" to faculty and advertising to graduate students. The hope is to recruit graduate students to a course focused on GA (where meaningful "teaching" can occur) based on the students' interest or, better yet, at the behest of a faculty advisor which produces a 2-for-1 bonus: while the student engages actively in the course, the faculty advisor can also engage with GA vicariously as part of their ongoing role as research supervisor. In my experience, having a student advisee working on a project is a tried-and-true way to maintain faculty engagement.

Organizing & Preparing the Seminar Materials

Choosing a GA Flavor

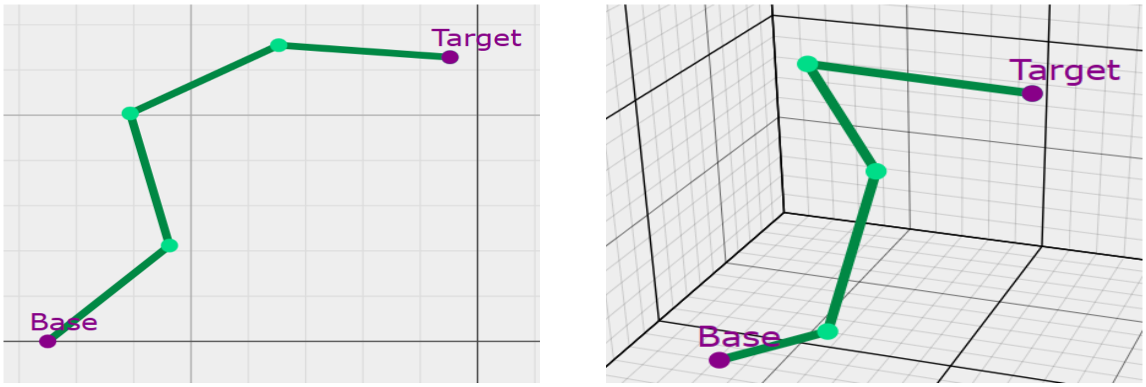
One of the big challenges when getting into GA is sorting through all of the different flavors of GA and deciding where to focus attention. Euclidean geometric algebra (EGA)[7] is attractive based on its relative simplicity and intuitiveness. However, for ME applications, EGA's lack of support for modeling translations is a deal-breaker. Conformal geometric algebra (CGA)[6] is attractive based on its ability to directly model circles and spheres, geometries that arise frequently in engineering applications (especially in some of my personal interest areas including computer-aided design and geometric skeletons). However, computational efficiency is also of significant importance in those areas,

[1]Here "magical" is used in the sense of Arthur C. Clarke's third law that "Any sufficiently advanced technology is indistinguishable from magic."

and the 2-up nature of CGA (i.e. the inclusion of 2 additional independent geometric numbers) creates issues related to efficient implementation in terms of both computation and storage. For 3D CGA, the additional geometric numbers cause the GA multivector basis dimension to increase by a factor of 4: from 8 for 3D EGA to 32 for 3D CGA. There are tricks available to enhance efficiency[8], but the combination of computational cost and a less intuitive relationship between vectors and associated geometric entities makes CGA seem not quite the right choice. There are also choices involving space-time algebra that are well-suited for electrodynamics[5] (which is relevant to the work done by a large subset of MEs), again it seems too far removed from the intuitive algebra-geometry relationship of EGA to be the right entry point for most MEs. After spending a fair amount of time exploring the possible choices, I see Plane-based Geometric Algebra (PGA)[1] as likely the best entry point for MEs. There is a disadvantage in that the algebra-geometry relationship is not quite as intuitive as for EGA, but that disadvantage is minor since using what some consider the "dual representation" (i.e. that a vector corresponds not to its line but to the normal hyperplane) is not too foreign and involves a relatively painless adjustment of intuition. Moreover, the slight disadvantage is heavily outweighed by the advantage that PGA includes both rotations and translations, thus supporting the essentials of statics and dynamics that form the foundation of mechanics.

Finding a "Magical" Opening Demonstration

Having identified PGA as the target flavor, the first task of seminar preparation is to find an appropriate "familiar yet magical" example. The outstanding resource for this task is Steven De Keninck's `ganja.js` website which offers a number of excellent candidates. For initial familiarity to the ME audience, the inverse kinematics sample application from the 2D PGA ($\mathbb{R}_{2,0,1}$) selection is difficult to beat. (A screen shot produced by the 2D inverse kinematics sample is shown in Fig.1a.) The problem of determining the configuration of a multi-link robot that positions the end effector at a target location is recognizable, relevant, and requires minimal explanation. The inverse kinematics sample application is also outstanding in terms of its simple, concise code. The listing is 43 lines long, but most are comments; the actual code comprises fewer than 20 lines. While most of the ME audience may be unfamiliar with JavaScript, the code is short and simple enough to explain what is happening line-by-line, even to an audience completely unfamiliar with the language. That covers the issue of familiarity, but what about the "magic"? Having explained the basics of the problem (which is familiar and easy to think about in 2D), the set-up for the magic is to ask the audience to think about how they would implement the inverse kinematics sample for a robot with spherical joints in a 3D setting. After a bit of thought and/or discussion, it is time to perform the "magic" by changing a single character in the code and producing a working implementation of inverse kinematics for the 3D robot. The only required change is to replace 2 with 3 in the first line which becomes `Algebra(3,0,1,()=>{`. Following this simple change, the 3D version runs and produces output shown in Fig.1b. The still image in Fig.1b may not be amazing, but seeing the 3D robot model perform real-time tracking of a target moving through space after changing a single character in the 2D implementation is perceived as "magical" and does get the attention of the audience.



(a) 2D inverse kinematics

(b) 3D version changing 1 character.

Figure 1: Inverse kinematics of a multi-link robot arm from the `ganja.js` sandbox.

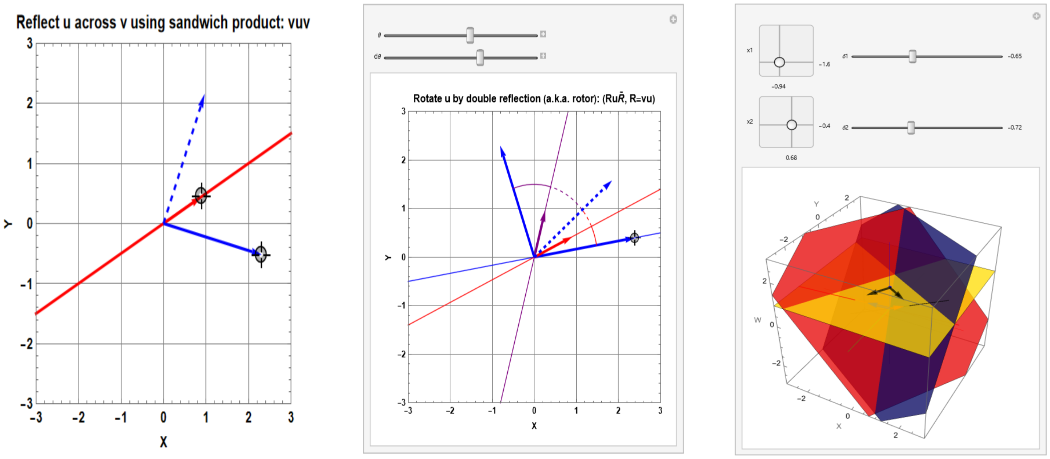
Conveying the Basis of PGA

Introduction of Geometric Numbers, Geometric Product and EGA

The best hook I have seen for geometric numbers is in Martin Roelfs' videos (see //bivector.net). He introduces geometric numbers by an alternative view of the quadratic $x^2 + 1 = 0$. Most MEs go straight to complex algebraic manipulation; i.e. $x^2 + 1 = 0 \rightarrow x^2 = -1 \rightarrow x = \sqrt{-1} = \pm i$. The alternative considers the quadratic to be about areas of square regions. While you can associate the terms 1 and 0 with areas of squares with real sides, no real is the side of a square with area -1. This requires something new, but it need not be i. Instead, a geometric number, e_-, is introduced as the side of the square with area -1. Geometric numbers, e_+ and e_0, are also introduced that square to 1 and 0. These numbers work enough like cartesian unit vectors that it is not a big leap to vectors as linear combinations of geometric numbers, and the stage is set for EGA. [2]
We introduce algebraic numbers, e_1 and e_2, with $e_1^2 = 1, e_2^2 = 1$ and form GA vectors as linear combinations of e_1 and e_2): $u = u_1e_1 + u_2e_2, v = v_1e_1 + v_2e_2$. We then compute u^2 and impose the Contraction Axiom (geometric numbers square to reals):
$u^2 = (u_1e_1 + u_2e_2)^2 = u_1^2\cancel{e_1e_1} + u_2^2\cancel{e_2e_2} + u_1u_2(e_1e_2 + e_2e_1) = u_1^2 + u_2^2 + u_1u_2(e_1e_2 + e_2e_1)$
For u^2 real, the parentheses must vanish giving $e_2e_1 = -e_1e_2$ which leaves $u^2 = u_1^2 + u_2^2$.
This simple exercise leads to 4 important realizations:
(1) Switching the u's in uu does nothing, so parallel vectors commute.
(2) Switching e_1e_2 to e_2e_1 introduces a minus sign, so orthogonal vectors anticommute.
(3) The magnitude squared is $|u|^2 = u_1^2 + u_2^2$, so the Pythagorean Theorem is built-in.
(4) GA vectors have inverses ($uu = |u|^2$ so $u^{-1} = u/|u|^2$), and we can divide GA vectors.
Now consider the product of different vectors:
$uv = (u_1e_1 + u_2e_2)^2 = u_1v_1 + u_2v_2 + (u_1v_2 - u_2v_1)e_1e_2$
$vu = (u_1e_1 + u_2e_2)^2 = u_1v_1 + u_2v_2 - (u_1v_2 - u_2v_1)e_1e_2$
The results include a scalar and a part involving $e_{12} := e_1e_2$, the product of geometric numbers, or equivalently, the product of vectors.[3] This introduces bivectors and enables tracking all of the pieces (scalars, vectors, & bivectors) as they intermingle during oper-

[2] We stick to 2D EGA to keep both equations and graphics simple.

[3] Linear combinations of these product equations can isolate terms y symmetry and write the geometric product of vectors as the sum of the dot product and the wedge product: $uv = u \cdot v + u \wedge v$.



(a) (b) (c)

Figure 2: Images from interactive GA visualizations. (a) 2D EGA reflection: Control elements (indicated by locator "crosshairs") include input vector u (solid arrow) and mirror vector v (solid line). Reflected output from sandwich product vuv shown as dashed arrow. (b) 2D EGA rotation: Control elements include input vector u (solid arrow with locator) and mirror vectors v (arrow with solid line) and w (arrow with dashed line). Rotated output from sandwich product $wvuvw = Ru\widetilde{R}$ shown as solid arrow (without locator). (c) 2D PGA: Planes defined by vectors with e_0 components meeting 2D plane to define lines that meet at a point. Enables visualization of parallel lines intersecting at an infinity and motor transformations (omitted here for clarity).

ations. Thus, we arrive at 2D EGA as a vector space with basis $\{1, e_1, e_2, e_{12}\}$. We seem to pay a price by working in 4D space to model 2D geometry, but there are benefits to be gained in terms of ease of transformations. In traditional matrix-vector algebra, transformations rely on more complicated objects (matrices) from outside the vector space. In GA, transformations are performed by GA multiplication using the sandwich product. To reflect input vector u in the mirror along normalized vector v, we compute the transformed vector u' using the sandwich product, $u' = vuv$, as shown in Fig.2a.[4] Rotations are accomplished as the product of reflections (about mirrors aligned with v and w) which adds a layer to the sandwich. We compute the rotated vector as $u'' = wvuvw$. See Fig.2b. To set up the jump to PGA, a similar interactive tool is available (but not included here due to space) for the plane-based version (where a vector represents its normal plane). [5] Having shown that GA supports rotations as bireflections in vector subspaces, it is time to deal with the EGA limitation that that vector subspaces pass through the origin, so there is no way to translate or rotate about other points. To support those operations, we extend to 2D PGA where we embed the 2D plane in a 3D space by introducing an additional coordinate direction and using a geometric number e_0 with $e_0^2 = 0$. This is where interactive graphics become more complicated but more essential to conveying the properties of PGA including: (1) adding an e_0 vector compo-

[4]The figure shows a still image from an interactive visualization, and the GA computations are performed using the `clifford.m` package[4].

[5]We note the useful terminology that groups the product of mirror vectors as the rotor $R = wv$ so rotation becomes $u'' = Ru\widetilde{R}$. The tilde indicates the reverse: $R = uv \iff \widetilde{R} = vu$.The idea of rotor as exponentiated invariant could come in here, but it likely makes more sense when discussing PGA.

nent allows tilting the plane through the 3D origin so that the intersection line with the 2D space is displaced from the 2D origin; (2) implementing geometric operations ("join" and "meet") using the wedge product and duality[6]; (3) the sandwich product again implements transformations, but now rotations are performed about arbitrary points and translations are included[7]; (4) generalization of rotors to motors; and (5) generating a motor by exponentiating its invariant. These aspcets are non-trivial for newcomers, but it is possible to get the ideas across with well-planned interactive demonstrations. Figure2c shows a still from an interactive display of 2D PGA hyperplanes defining lines in the 2D plane that meet in a point. This sets up for illustrating PGA transformations, but that aspect is omitted here because interactivity is needed for it to make sense.

Seminar Finale and Conclusion

To wrap up, another display of dimensional "magic" is in order that hints at what is ahead: classical physics; i.e. "weighted" multivectors and a rate relation between motors and forques (combining forces and torques). This is the gist of what must be distilled (ongoing work) as the basis for updated introductory statics & dynamics classes. This example (again illustrating amazing dimensional independence) is presented as a motivational aid without getting into details. An appropriate choice is the dynamic simulation of a box on a spring subject to gravity and damping[3]. The code is explainable, < 30 lines, and does the "magic" of adapting from 2D to 3D with the change of 1 character providing a nice conclusion for both the seminar plan and this paper.

References

[1] Leo Dorst & Steven De Keninck, Guided Tour to Plane-based Geometric Algebra (PGA), Lecture Notes, SIGGRAPH, 2020. bivector.net/PGA4CS.html

[2] L. Dorst & S. De Keninck, Plane-based GA for CS. bivector.net/PGA4CS.html

[3] L. Dorst & S. De Keninck, May the Forque be with you. //bivector.net/PGADYN.html, enki.ws/ganja.js/examples/pga_dyn.html (code)

[4] J.-L. Aragon et al., Clifford Algebra with *Mathematica*, in *Proceedings of the 29th International Conference on Applied Mathematics* Budapest, 2015.

[5] Peter Joot, *Geometric Algebra for Electrical Engineers*, 2023.

[6] Hestenes et al, in G. Sommer (ed.), *Geometric Computing with Clifford Algebra.* Springer Verlag, 2000. ISBN 3-540-41198-4

[7] Alan Macdonald, *Linear and Geometric Algebra*, Decorah, IA, 2010.

[8] D. Hildenbrand, Using Gaalop for High-Performance Geometric Algebra Computing. *Geometry and Computing*, vol 8. Springer, Berlin, 2013.

[6] This is a perfect occasion for a first discussion of duality in GA.

[7] A short aside is appropriate at this point on the robustness of projective intersection tests and the interpretation of translation as rotation about an infinite point.