GAViewer exercises

February 5, 2008

Introduction

This is a companion set of exercises for "Geometric Algebra for Computer Science" that use GAViewer. The intent is for the reader to do the exercises corresponding to each chapter as they read the book to get a more hands-on feeling for using geometric algebra. To learn more about the book and to download GAViewer, please visit the webpage `www.geometricalgebra.net`

The intent of these exercises is to guide your exploration. After computing something, it usually will be drawn on the screen. Rotate the scene to get a better feel for the 3D relationships. Consider constructing similar objects to see how variations change the results. If you are teaching geometric algebra, these exercises might prove useful as models for homeworks assignments.

If you wish to save the text in the text window, use the `diary` command, which will save the text (both commands and output) since the previous `diary` command (or the start of the session for the first `diary` command).

For many of the exercises, you may find it helpful to label individual items. For an object names `a`, you can draw a label for it using the command `label(a)`.

1 Chapter 1

This set of exercises takes you through the example in Chapter 1; some of the constructions found here won't be explained until later chapters of the book, but hopefully you will get a better feeling for the geometry that is happening. The geometry that results from the commands that follow isn't identical to that appearing in the text, but the general result is the same.

1. If you have been trying other things in GAViewer, exit and restart GAViewer as the easiest way to return to its pristine state.
2. Set the geometry model with `default_model(c3ga)`
3. Construct three points (for convenience, you can use up-arrow keys to get earlier command lines, and edit those by insertion and overwriting):
   - `c1 = c3ga_point(e1)`
   - `c2 = c3ga_point(2*e1+e2)`
   - `c3 = c3ga_point(e3)`
4. Construct a circle `C = c1^c2^c3`
5. Construct a rotor `R = e3*(e1+0.85*e3)`. The circle you see represents the plane of rotation. Note that the '0' in '0.85' is necessary in GAViewer.
6. Apply the rotor to the circle: `RC = R*C/R` and then `RRC = R*RC/R`. You can view the result from different angles by a left-click-and-drag.
7. Construct a line `L = c1^c2^ni`
8. Construct a rotor around the line as `RL = 1 + 0.1 dual(L)` (this will not be drawn).

9. Apply the line rotor to the original circle: `RC = RL*C/RL` and then `RRC = RL*RC/RL`. You can view the result from different angles by a left mouse button drag.

10. Construct a plane `pi1=c3ga_point(e1+e2).(e2^ni)`

11. Reflect the circle, the line, and the points in the plane:
    - `rC = pi1*C/pi1`
    - `rL = pi1*L/pi1`
    - `rc1 = pi1*c1/pi1`, etc.

2 Chapter 2

GAViewer Notes

- Exit and restart GAViewer. To ensure that you are in the correct geometry model for this chapter, type `default_model(e3ga)`.
- If you type `e1`, then a vector representing $\mathbf{e}_1$ is drawn in the drawing window. However, when the next item is drawn, the vector representing $\mathbf{e}_1$ will disappear. For an object to remain on the screen, you must assign it to a new variable, e.g., `a=e1`
- To clear the screen and all assigned variables, use the command `clf()`
- If you ctrl-right-click-and-drag, you can move an object. For vectors, you can move the location of the head of the vector; for bivectors, you will rescale the bivector.
- To delete an individual object, ctrl-left-click on it; then on the right side of the screen, you will see a button saying 'remove this object'. Click that.
- Left-click-and-drag rotates the scene; middle-click-and-drag translates the scene; right-click-and-drag moves things closer/farther away.

2.1 Drills

1. Oriented Area Elements

   (a) Compute and draw the following with GAViewer. Between each exercise, be sure to type `clf()` to clear the previously drawn objects. The `red`, etc., set the color of the object.

   - ```
     a = e1
     b = e1+e2
     c = a^b
     ```
   - ```
     b = e1^e2
     g = green(e2^e3)
     r = red(b+g)
     y = yellow(0.8*b + 0.2*g)
     ```
   - $(\mathbf{e}_1 + \mathbf{e}_2) \wedge (\mathbf{e}_2 + \mathbf{e}_3) \wedge (\mathbf{e}_3 + \mathbf{e}_1)$

   (b) Draw the following bivectors in one figure. They should appear as a set of circular disks.

   - `b=e1^e2`
   - `r=red(e2^e3)`
   - `g=green(e3^e1)`

   (c) Draw the following bivectors in one figure (you will need to assign them to variables to get them all to appear in one figure). They should appear as a set of circular disks. Make sure you understand their relative direction and magnitude.

- $\mathbf{e}_1 \wedge \mathbf{e}_2$
- $\mathbf{e}_1 \wedge (\mathbf{e}_1 + \mathbf{e}_2)$
- $2\mathbf{e}_1 \wedge \mathbf{e}_2$

2. Parallelness

   (a) Let $\mathbf{B} = (\mathbf{e}_1 + 3\mathbf{e}_2) \wedge (\mathbf{e}_2 + 2\mathbf{e}_3)$. Determine by a GAViewer computation using the outer product ^ if the following vectors lie in the plane represented by $\mathbf{B}$

   - $\mathbf{e}_1$
   - $\mathbf{e}_2$
   - $\mathbf{e}_3$
   - $\mathbf{e}_1 + 2\mathbf{e}_3$
   - $\mathbf{e}_2 + 2\mathbf{e}_3$
   - $-\mathbf{e}_1 + 6\mathbf{e}_3$

   (b) Draw a picture the objects in the previous question (i.e., $\mathbf{B}$ and all six of the vectors).

2.2 Advanced GAViewer

1. Draw all of the bivectors in Drill 1b of the previous section using the `dm2` command (for 'drawing method 2'). For example, to draw $\mathbf{e}_1 \wedge \mathbf{e}_2$ you would type `dm2(e1∧e2)`.

2. Draw all of the bivectors in the Drill 1c of the previous section using the `dm2` command. You can change drawing method by using '`dm1`' et cetera, or using the control menu: select the element, and then the 'draw method'.

3. Trivectors.

   (a) Draw the trivector $\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3$.

   (b) Draw the trivector $\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3$ using `dm4`.

4. Draw the bivectors $\mathbf{e}_1 \wedge \mathbf{e}_2$, $\mathbf{e}_2 \wedge \mathbf{e}_3$, $0.2\,\mathbf{e}_3 \wedge \mathbf{e}_1$. Now draw a "dynamic" bivector, first by typing `a=e1`, `b=e2`, and then `dynamic{d=cyan(a^b),}` (Note: the ',' at the end of the dynamic statement is needed). By ctrl-right-click-and-dragging the arrow heads of `a` or `b`, you can change the cyan bivector interactively. Adjust the arrows so that

   (a) The cyan bivector aligns with the small, dark-blue bivector in both orientation and size.

   (b) The cyan bivector aligns with the small, dark-blue bivector, but with the cyan bivector the same size as the two large, dark-blue bivectors.

   To remove dynamic statements, you need to call the `cld()` command (i.e., dynamic statements are not cleared with `clf()`).

5. If you play with the `dm2`, etc., commands, you will notice that they draw canonical representations of objects. E.g., the vectors $\mathbf{a}$ and $\mathbf{b}$ that form the parts of an expression like $\mathbf{a} \wedge \mathbf{b}$ are lost.

   To draw the basic components of a bivector, we need to add the following function to GAViewer:

```
function factored_bivector(e3ga a, e3ga b) {
    set_factor(0, a);
    set_factor(1, b);
    return dm2(a ^b);
}
```

   Now if we have $\mathbf{a} = \mathbf{e}_1 + \mathbf{e}_2$ and $\mathbf{b} = e1$ and we type `factored_bivector(a,b)`, then we see $\mathbf{a}$ and $\mathbf{b}$ as well as the bivector, although the components may be translated. You can even put the `factored_bivector(a,b)` statement inside a dynamic statement and change the components and see the bivector change.

(a) Draw all of the bivectors in Drill 1b of the previous section using `factored_bivector`.

(b) Draw all of the bivectors in the Drill 1c of the previous section `factored_bivector`.

Note: rather than have to retype the code for `factored_bivector(a,b)` each time you start GAViewer, you can store the code in a .g file. Further, the figures.zip file that is available on the book's webpage contains factored.g which has code for drawing both factored bivectors and factored trivectors. See the GAViewer documentation for more details on .g files and how to load them into GAViewer.

3 Chapter 3

GAViewer notes

- The dual of an object `a` can be found using `dual(a)`.
- In GAViewer, the contraction of two elements is formed using the '.' as in `a.B`.

3.1 Drills

1. Geometric Interpretation of the Contraction

   (a) Draw a picture of $\mathbf{B} = \mathbf{e}_1 \wedge \mathbf{e}_2$, $\mathbf{v} = \mathbf{e}_1 + \mathbf{e}_3$ and $\mathbf{v} \rfloor \mathbf{B}$.

   (b) Draw a picture of $\mathbf{B} = \mathbf{e}_1 \wedge \mathbf{e}_2$, $\mathbf{v} = \mathbf{e}_1 + \mathbf{e}_2$ and $\mathbf{v} \rfloor \mathbf{B}$.

2. Duality

   Draw each of the following objects and compute and draw their duals; each part of the question should be drawn in separate figures, although all the objects in one part (and all of their dual) should be drawn in one figure.

   (a) $\mathbf{e}_1$

   (b) $\mathbf{e}_1$, $\mathbf{e}_1 + \mathbf{e}_2$

   (c) $\mathbf{e}_1 \wedge \mathbf{e}_2$

   (d) $\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3$

   (e) 3.0

3. Reciprocal Frames

   Compute the reciprocal frame $\{\mathbf{b}^1, \mathbf{b}^2, \mathbf{b}^3\}$ of $\{\mathbf{b}_1 = \mathbf{e}_1, \mathbf{b}_2 = \mathbf{e}_1 + \mathbf{e}_2, \mathbf{b}_3 = \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3\}$. For the inverse use (3.22), which uses (3.4). You can either code this by hand, or use the reverse in GAViewer, which is '`reverse( )`'. Verify that you have computed the reciprocal frame correctly by computing $\mathbf{b}_i \cdot \mathbf{b}^j$. Then find the coordinates of $\mathbf{v} = \mathbf{e}_1 + 2\mathbf{e}_2 + 3\mathbf{e}_3$ relative to $\{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$.

3.2 Advanced GAViewer

1. Let $\mathbf{B} = \mathbf{e}_1 \wedge \mathbf{e}_2$, $\mathbf{v} = \mathbf{e}_1 + \mathbf{e}_2$. Draw both, and then draw as a dynamic entity $\mathbf{v} \rfloor \mathbf{B}$. What happens to the magnitude of $\mathbf{v} \rfloor \mathbf{B}$ as $\mathbf{v}$ becomes perpendicular to $\mathbf{B}$? What happens to the magnitude of $\mathbf{v} \rfloor \mathbf{B}$ as $\mathbf{v}$ gets larger? What happens to the magnitude of $\mathbf{v} \rfloor \mathbf{B}$ as $\mathbf{B}$ gets larger (remember that you can manually enlarge a bivector with right-click-and-drag)?

5 Chapter 5

GAViewer notes

- The meet and join can be called using either `meet(a,b)` and `join(a,b)` or as `a&b` and `a|b`.

5.1 Drills

1. Compute the outer product and the join of the following:
   (a) $\mathbf{e}_1$, $\mathbf{e}_2$
   (b) $\mathbf{e}_1 \wedge \mathbf{e}_2$, $\mathbf{e}_2 \wedge \mathbf{e}_3$
   (c) $\mathbf{e}_1 + \mathbf{e}_2$, $\mathbf{e}_3$
   (d) $\mathbf{e}_1 \wedge \mathbf{e}_2 + \mathbf{e}_2 \wedge \mathbf{e}_3$, $2\mathbf{e}_1 \wedge \mathbf{e}_3 + \mathbf{e}_1 \wedge \mathbf{e}_2$
2. Compute the meet of the following, drawing the individual elements as well as their meet:
   (a) $\mathbf{e}_1 \wedge \mathbf{e}_2$, $\mathbf{e}_2 \wedge \mathbf{e}_3$
   (b) $\mathbf{e}_1 \wedge \mathbf{e}_2 + \mathbf{e}_2 \wedge \mathbf{e}_3$, $2\mathbf{e}_1 \wedge \mathbf{e}_3 + \mathbf{e}_1 \wedge \mathbf{e}_2$
3. Let $\mathbf{a} = \mathbf{e}_1 \wedge \mathbf{e}_2$ and $\mathbf{b} = \mathbf{e}_2 \wedge \mathbf{e}_3 + \mathbf{e}_3 \wedge \mathbf{e}_1$.
   (a) Draw $\mathbf{a}$, $\mathbf{b}$, and $\text{meet}(\mathbf{a}, \mathbf{b})$.
   (b) Draw $\mathbf{a}$, $\mathbf{b}$, and $\text{join}(\mathbf{a}, \mathbf{b})$.

6 Chapter 6

6.1 Drills

1. The Geometric Product For Vectors
   (a) Given $\mathbf{x} \wedge \mathbf{e}_2 = \mathbf{e}_1 \wedge \mathbf{e}_2$ and $\mathbf{x} \cdot \mathbf{e}_2 = 2$, find $\mathbf{x}$.
   (b) Compute the following geometric products.
      i. $(\mathbf{e}_1 + \mathbf{e}_2)\,(\mathbf{e}_2 + \mathbf{e}_3)$
      ii. $(\mathbf{e}_2 + \mathbf{e}_3)\,(\mathbf{e}_1 + \mathbf{e}_2)$
2. The Geometric Product of Multivectors
   (a) Compute the following geometric products.
      i. $(\mathbf{e}_1 \wedge \mathbf{e}_2)\,(\mathbf{e}_2 \wedge \mathbf{e}_3)$
      ii. $(\mathbf{e}_1 + \mathbf{e}_2)/(\mathbf{e}_2 \wedge \mathbf{e}_3)$
   (b) Let $\mathbf{B} = \mathbf{e}_1 \wedge \mathbf{e}_2$. Compute $\mathbf{e}_3 \wedge \mathbf{B}$ directly and through the formula $\frac{1}{2}(\mathbf{a}\,\mathbf{B} + \widehat{\mathbf{B}}\mathbf{a})$. The grade involution in GAViewer is '`gin( )`'.
   (c) Let $\mathbf{a} = \mathbf{e}_1 + \mathbf{e}_2$, $\mathbf{b} = \mathbf{e}_1 + \mathbf{e}_2 + 2\mathbf{e}_3$ and $\mathbf{c} = \mathbf{e}_1 + 2\mathbf{e}_2 - \mathbf{e}_3$.
      i. Compute $\mathbf{D} = \mathbf{a}\,\mathbf{b}\,\mathbf{c}$.
      ii. Compute the grade 0, 1, 2, and 3 parts of $\mathbf{D}$, using the '`grade( , )`' command.
      iii. Draw $\mathbf{a}, \mathbf{b}, \mathbf{c}$, $\text{grade}(\mathbf{D}, 1)$ and $\text{grade}(\mathbf{D}, 3)$.
3. Projection To Subspaces
   (a) Make a dynamic drawing showing the rejection and projection of $2\mathbf{e}_1 + \mathbf{e}_2$ onto $\mathbf{e}_1$, then change either by dragging.
   (b) Make a dynamic drawing showing the rejection and projection of $2\mathbf{e}_1 + \mathbf{e}_2$ onto $\mathbf{e}_1 \wedge \mathbf{e}_2$, then change either by dragging.
4. Reflection
   (a) Draw two vectors $\mathbf{a} = \mathbf{e}_1 + \mathbf{e}_2$ and $\mathbf{b} = \mathbf{e}_1 + 0.5\mathbf{e}_2$. Label these vectors using the `label` command; for example, to label $\mathbf{a}$, you would type `label(a)`. Now draw the reflection of $\mathbf{a}$ in $\mathbf{b}$ using a dynamic statement. Label the reflection with the equation you used to compute it (you can specify a text string for the label by giving a second argument to `label` that is the double quoted string you want as the label); be sure to make your label dynamic, too.
   (b) Redo the previous question, but draw the reflection of $\mathbf{b}$ in $\mathbf{a}$.

7 Chapter 7

7.1 Drills

1. Reflections of Subspaces

   (a) Perform the following reflections:

      i. Reflect $\mathbf{e}_1 + \mathbf{e}_2$ in $\mathbf{e}_3$

      ii. Reflect $\mathbf{e}_1 + \mathbf{e}_2$ in $\mathbf{e}_1 + \mathbf{e}_3$

      iii. Reflect $\mathbf{e}_1 \wedge \mathbf{e}_3$ in $\mathbf{e}_1 + \mathbf{e}_2$

      iv. Reflect $\mathbf{e}_1 + \mathbf{e}_2$ in $\mathbf{e}_1 \wedge \mathbf{e}_3$

   (b) Equation (4.7) shows that a transformation applied to the pseudoscalar scales the pseudoscalar by the determinant of a transformation. This equation can be rewritten to give a formula for the determinant of a transformation.

      Use GAViewer to compute the determinant of each of the reflections given in Drill 7.1.1(a) relative to the pseudoscalar $\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3$.

   (c) Use GAViewer to compute the determinant of the reflection given in part 7.1.1(a)(iii) relative to the pseudoscalar $\mathbf{e}_1 \wedge \mathbf{e}_2$.

2. Rotations of Subspaces

   (a) Let $\mathbf{a} = \mathbf{e}_1$ and $\mathbf{b} = (\mathbf{e}_1 + \mathbf{e}_2)/\sqrt{2}$ and $\mathbf{r} = \mathbf{b}/\mathbf{a}$. Apply the rotor $\mathbf{r}$ to

      i. $\mathbf{e}_1$

      ii. $\mathbf{e}_2 + \mathbf{e}_3$

      iii. $\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3$

      iv. $\mathbf{e}_1 \wedge \mathbf{e}_3$

      v. $\mathbf{e}_1 \wedge \mathbf{e}_2$

   (b) Use GAViewer to compute the determinant of $\mathbf{r}$ of the previous question relative to the pseudoscalar $\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3$.

   (c) Use GAViewer to compute the determinant of $\mathbf{r}$ relative to the pseudoscalar $\mathbf{e}_1 \wedge \mathbf{e}_2$.

   (d) Let $\mathbf{a} = (\mathbf{e}_1 + \mathbf{e}_2)/\sqrt{2}$ and $\mathbf{b} = \mathbf{e}_1$. Compute and draw the rotor $\mathbf{b}/\mathbf{a}$. Now compute and draw the rotor $\mathbf{a}/\mathbf{b}$.

3. Rotors as Exponentials of Bivectors

   (a) Construct the rotor $\mathbf{R}$ that rotates by $\pi/3$ around $\mathbf{e}_3$ by constructing the unit dual to $\mathbf{e}_3$ and using the exponentiation formula for construction a rotor (7.15). Let $\mathbf{a} = \mathbf{e}_1 + \mathbf{e}_3$, and draw $\mathbf{a}_i$ for $i = 0, ..., 11$ where $\mathbf{a}_0 = \mathbf{a}$ and $\mathbf{a}_{i+1}$ is $\mathbf{a}_i$ rotated by rotor $\mathbf{R}$.

10 Chapter 10

GAViewer notes

In GAViewer, you can exponentiate vectors, etc., but there is no built in log functions. However, in `figures.zip` (available on the book's webpage) you will find Log.g, which has some basic routines for take logarithms of elementary GA objects. To perform some of the exercises in this section, you will need to load Log.g into GAViewer; see the documentation on GAViewer for instructions on how to do this.

10.1 Drills

1. Angular Relationships

   Use (10.9) to compute the oriented areas of the following triangles having edges **a**, **b**, and **c** as illustrated in Figure 10.1(a), with $\mathbf{I} = \mathbf{e}_1 \wedge \mathbf{e}_2$:

   (a) $\mathbf{a} = \mathbf{e}_1$, $\mathbf{b} = \mathbf{e}_2$, $\mathbf{c} = -\mathbf{e}_1 - \mathbf{e}_2$.

   (b) $\mathbf{b} = \mathbf{e}_2$, $\mathbf{a} = \mathbf{e}_1$, $\mathbf{c} = -\mathbf{e}_1 - \mathbf{e}_2$.

   (c) $\mathbf{a} = \mathbf{e}_1 + \mathbf{e}_2$, $\mathbf{b} = \mathbf{e}_1 - 2\mathbf{e}_2$, $\mathbf{c} = -2\mathbf{e}_1 + \mathbf{e}_2$.

   (d) $\mathbf{a} = 0.5\mathbf{e}_1 + 1.5\mathbf{e}_2$, $\mathbf{b} = -\mathbf{e}_1 - \mathbf{e}_2$, $\mathbf{c} = 0.5\mathbf{e}_1 - 0.5\mathbf{e}_2$.

2. The Logarithm of a 3-D Rotor

   Compute the plane of rotation of the following rotors using slog.

   (a) $\mathbf{a} = \mathbf{e}_1 + \mathbf{e}_2$, $\mathbf{b} = 2\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3$, $\mathbf{R}_1 = \mathbf{a}/norm(\mathbf{a}) \wedge \mathbf{b}/norm(\mathbf{b})$

   (b) $\mathbf{c} = \mathbf{e}_3 + \mathbf{e}_2$, $\mathbf{d} = \mathbf{e}_1 + 2\mathbf{e}_2 + \mathbf{e}_3$, $\mathbf{R}_2 = \mathbf{c}/norm(\mathbf{c}) \wedge \mathbf{d}/norm(\mathbf{d})$

   (c) $\mathbf{R} = \mathbf{R}_2/\mathbf{R}_1$

3. Rotation Interpolation

   Compute the rotor that is "half-way" between $\mathbf{R}_1$ and $\mathbf{R}_2$.

11 Chapter 11

GAViewer Notes

- In GAViewer, $\mathbf{e}_0$ represents the origin.
- For this section, you should change the default model to p3ga by using the command `default_model(p3ga)`.

11.1 Drills

1. All Points Are Vectors

   (a) Construct the following points:
   - `p1` at location $\mathbf{e}_1$
   - `p2` at location $\mathbf{e}_2$
   - `p3` at location $\mathbf{e}_3$
   - `p3a` at location $\mathbf{e}_3$ with weight 2

   (b) Compute `p4=p1+p2`. What is its weight? What is its location?

   (c) Compute `p4a=(p1+p2)/2`. What is its weight? What is its location?

   (d) Compute `p5=(p2+p3a)/2`. What is its weight? What is its location?

   (e) Compute `p6=(p1+p2+p3)/3`. What is its weight? What is its location?

2. All Lines are 2-Blades

   (a) Construct `p1`, `p1`, `p3`, and `p3a` as in the previous question,and `p4=(p1+p2)/2`.

   (b) Construct the line `l1=p1∧p2`. What is the direction of the line? What is `l1`'s moment?

   (c) Test to see if `p1`, `p3`, and `p4` lie on `l1`

   (d) Construct line `l2` to pass through point `p3` and have direction $\mathbf{e}_2$. Construct `l2a` to pass through the point `p3a` and have direction $\mathbf{e}_2$. What difference is there in the representations of `l2` and `l2a`?

   (e) Compute the support vectors for `l1` and `l2`.

3. All Planes Are 3-Blades
   (a) Construct points `p1`, `p2`, and `p3` as in the previous question.
   (b) Construct the plane `Pi1` passing through points `p1`, `p2`, and `p3`.
   (c) Construct the plane `Pi2` passing through point `p1` and parallel to $\mathbf{e}_1 \wedge \mathbf{e}_2$.
   (d) Compute the support vectors for `Pi1` and `Pi2`.

4. Incidence Relationships
   (a) Construct lines `L=(e0+e1)∧e2` and `M=(e0+e2)^(e1+e2)`. Find their intersection by taking the `meet` of the two lines.
   (b) Construct the line `N=(e0+2*e1)^e2` and compute the `meet` of `N` and `L`.
   (c) Construct the line `O=(e0+e3)^(e1-e2)` and compute the `meet` of `O` and `L`.

13 Chapter 13

GAViewer Notes

- In GAViewer, you can construct a point with the c3ga_point() command, giving a vector as the argument.
- For this section, you should change the default model to c3ga by using the command '`default_model(c3ga)`'.

13.1 Drills

1. Points as Null Vectors.

   Use equation (13.3) to construct two points `p1` and `p2` with Euclidean locations vectors $\mathbf{e}_1$ and $\mathbf{e}_2 + 0.3\mathbf{e}_3$. Use the equation (13.4) to compute the distance from `p1` to itself and from `p1` to `p2`.

2. Dual Planes and Spheres

   Do the following, drawing all the elements you construct. You will likely want to use c3ga_point() to construct your points rather than using equation (13.3).

   (a) Construct points `p1` with Euclidean component $\mathbf{e}_1 + \mathbf{e}_2$, `p2` with Euclidean component $\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3$, and `p3` with Euclidean component $2\mathbf{e}_1 - \mathbf{e}_2 + \mathbf{e}_3$.
   (b) Construct the dual planes `pi1` with normal $\mathbf{e}_1$ that passes through the origin; and `pi2` with normal $\mathbf{e}_1$ that is a distance of 1 from the origin. Note: plane `pi1` won't be drawn as a plane in GAViewer unless you've changed the default model to c3ga.
   (c) Probe plane `pi2` with points `p1`, `p2`, and `p3`.
   (d) Construct the dual spheres `s1` at the origin with radius 2 and `s2` with center `p1` with radius 1.
   (e) Probe both dual spheres with the points `p1`, `p2`, `p3`. Note the resulting values when the points are in, on, or outside the spheres.
   (f) Probe the two spheres with the two planes. When is the result 0?

3. Versors
   (a) Construct `p1`, `p2`, `p3`, `pi1`, and `pi2` as in the previous question.
   (b) Construct the translation `T=pi2*pi1` and apply it to `p1`, `p2`, `p3`, drawing the translated points in blue.
   (c) Construct the plane `pi3` with normal $\mathbf{e}_1 + 0.5\mathbf{e}_2$ that passes through the origin, and construct the rotation `R=pi3*pi1`. Apply `R` to `p1`, `p2`, and `p3`, drawing the rotated points in yellow.

4. Covariant Preservation of Structure

   For this question, you are not interested in the graphics, but only in the text results.

(a) Construct pi1, pi2 and T as in the previous question. Apply the versor T to $\mathbf{e}_1\,\mathbf{e}_2$ and see that the result is equal to applying T to $\mathbf{e}_1$ and $\mathbf{e}_2$ separately and then taking the geometric product of the results.

(b) Now apply T to $\mathbf{e}_1 + 2\mathbf{e}_2$ and see that the result is equal to applying T to $\mathbf{e}_1$ and $\mathbf{e}_2$ separately and then taking the same linear combination of the results.

5. Flats and directions.

   (a) Construct p1, p2, and p3 as in question 2.

   (b) Construct the plane Pi through the points p1, p2, p3.

   (c) Construct the lines L1 through p2 and p3; L2 through p3 and p1; and L3 through p1 and p2. Line L1 should be the default color (green); L2 should be blue; and L3 should be yellow.

   (d) Construct the line through the origin parallel to $\mathbf{e}_1$ and the plane through the origin parallel to $\mathbf{e}_1 \wedge \mathbf{e}_2$.

6. General Planar Reflections

   (a) Construct p1, p2, and p3 as in question 2, and construct the plane Pi as in the previous question.

   (b) Construct the point p4 with Euclidean component $3\mathbf{e}_1$ and the line L4 through p1 and p4, with L4 drawn in red.

   (c) Reflect L4 in Pi.

13.2 Questions

1. If you compute the inner product of the points and the planes in drill 2, you will find that the inner product appears to be 0 when the point is on the plane and is related to the distance to the plane otherwise. Under what conditions is the inner product of a point and a dual plane (in the conformal model) the distance from the point to the plane? What needs to be done in the other cases to result of the inner product to obtain the distance to the plane? Prove your results.

2. From the drills, it appears that the inner product of a dual plane and a dual sphere is zero when the center of the sphere is on the plane. State when this is true and prove your result.

14 Chapter 14

GAViewer Notes

- Some constructs have different geometric interpretations when considered as a primal or a dual. GAViewer tends to draw the primal interpretation. If you have dual representation D, you should draw $dual(D)$ to see the correct geometry.
- Remember to change the default model to c3ga by using the command `default_model(c3ga)`.

14.1 Drills

1. Dual rounds

   (a) Construct the dual sphere S at the origin with radius 2.

   (b) Intersect S with the plane through the origin with normal $\mathbf{e}_1$. The result will be a dual circle; be sure to draw the dual of this object to see the circle.

   (c) Intersect S with two planes through the origin, one with normal $\mathbf{e}_1$, the other with normal $\mathbf{e}_2$. The result will be a point pair; be sure to draw the dual of this object to see the circle.

2. Direct rounds

   (a) Construct points p1 with Euclidean component $\mathbf{e}_1 + \mathbf{e}_2$, p2 with Euclidean component $\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3$, and p3 with Euclidean component $2\mathbf{e}_1 - \mathbf{e}_2 + \mathbf{e}_3$.

(b) Construct the circle containing points `p1`, `p2`, `p3`.

(c) Construct the point `p4` with Euclidean component $3\mathbf{e}_1$ and then construct the sphere passing through points `p1`, `p2`, `p3` and `p4`.

3. Sphere intersections

   (a) Construct two dual spheres, `s1` and `s2`, with `s1` centered at $\mathbf{e}_1$ with radius 2 and `s2` centered at $-\mathbf{e}_1$ with radius 3.

   (b) Computer the intersection of these two dual spheres using the outer product. The result will be a dual circle; be sure to take its dual to display it in GAViewer.

   (c) Reassign `s2` to be the dual sphere centered at $-3\mathbf{e}_1$ with radius 2. Compute the intersection of the dual spheres `s1` and `s2`.

   (d) Move `s2` so that its center is at $-4\mathbf{e}_1$. Compute the intersection between the dual spheres `s1` and `s2`.

4. Parameters

   (a) Construct the dual sphere `s1` of radius 3 whose center is at $\mathbf{e}_1$.

   (b) Compute the squared radius of `s1` by squaring it.

   (c) Construct the points located at $\mathbf{e}_1$, $-\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$. Construct the direct sphere `s2` that passes through these four points.

   (d) Compute `s2*s2`. Did this give the negative of the squared radius? Hint: is the sphere normalized?

15 Chapter 15

GAViewer Notes

- Remember to change the default model to c3ga by using the command `default_model(c3ga)`.

15.1 Drills

1. Incidence revisited and coincidence

   (a) Construct the following three dual spheres:

   - `s1` of radius 1, centered at the origin
   - `s2` of radius 1.5, centered at $2\mathbf{e}_1$
   - `s3` of radius 1.25, centered at $1.5\mathbf{e}_3$

   (b) Use the outer product to intersect the dual spheres pairwise, producing three dual circles. Remember to dualize these circles to see them drawn as circles.

   (c) Intersect all three spheres to produce a point-pair. Remember to dualize this point pair to see it drawn as a point pair.

   (d) Draw the dual of this point pair and convince yourself that it intersects the spheres and the planes of the three circles perpendicularly.

   (e) Move `s3` so its center is at $2.5\mathbf{e}_3$. Recompute the intersection of the three dual spheres, which gives a dual point pair. To see the dual point pair, you would need to take the dual of the result of the intersection; however, we're interested in the dual of this dual point pair, which is a circle. Convince yourself that this circle is perpendicular to all three spheres.

2. Tangents, carriers, tangent flats, and surrounds

   (a) Construct the following points:

   - `p1` at $\mathbf{e}_1$
   - `p2` at $\mathbf{e}_1 + \mathbf{e}_2$

- p3 at $\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3$
- p4 at $\mathbf{e}_3$

(b) Construct the circle c through p1, p2, and p3 and construct the sphere through all four points.

(c) Construct the tangents to the circle at p1, p2, p3 and the tangent plane to the sphere at p4.

(d) Construct the carrier for c.

(e) Construct the tangent flats for each of the tangents to the circle that you constructed for points p1, p2, p3.

(f) Construct the surround for c.

(g) Multiply the surround and the carrier for c to see that the result is indeed c. You may need to multiply by -1 to get the correct sign.

3. Affine Combinations

(a) Construct two points p1 and p2 at $\mathbf{e}_1$ and $\mathbf{e}_2$. Take the following affine combinations of these two points, and extract the centers (see Table 14.1) to see that the center of the spheres travels along the line segment between p1 and p2:

- `0.75*p1 + 0.25*p2`
- `0.5*p1 + 0.5*p2`
- `0.25*p1 + 0.75*p2`

4. Euclidean Projections

(a) Construct the following elements:

- Point p1 at $\mathbf{e}_1$
- Point p2 at $\mathbf{e}_1 + \mathbf{e}_2$
- Point p3 at $\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3$
- Point p4 at $2\mathbf{e}_1 + \mathbf{e}_3$
- The circle c1 through p1, p2, p3
- The circle c2 through p1, p3, p4
- The plane Pi through the origin perpendicular to $\mathbf{e}_1$ (be sure to dualize if you construct the plane using the dual plane construction).
- The line l1 through p2 and p4.

(b) Project c1 onto Pi. Construct the surround of this projection and check that both circles lie on this sphere.

(c) Project c2 onto Pi. Construct the surround of this projection and check that both circles lie on this sphere. Call this surround c2ps.

(d) Project l1 onto Pi.

(e) Project l1 onto c2ps.

5. All Kinds of Vectors

(a) Create the following objects:

- Point p1 at $\mathbf{e}_1$
- Point p2 at $2\mathbf{e}_1$
- Point p3 at $\mathbf{e}_1 + \mathbf{e}_3$
- Circle c passing through p1, p2, p3.
- The tangent Tv to c at p1
- The normal n to the plane of the circle
- The free vector fv parallel to $\mathbf{e}_2$
- The line vector lv parallel to $\mathbf{e}_1$

- The rotation versor `R=e1*(e1+0.5*e1)/norm(e1+0.5*e1)`
- The translation versor `T` that translates by 2 units in the $\mathbf{e}_2$ direction (you will likely want to use the `tv` GAViewer command to create this).

(b) Transform `c`, `Tv`, `fv`, `n` by `R`.

(c) Transform `c`, `Tv`, `fv`, `n` by `T`.

16 Chapter 16

GAViewer Notes

- For this section, you should change the default model to c3ga by using the command '`default_model(c3ga)`'.

16.1 Drills

1. Spherical Inversion

   Construct the following elements:

   (a) Point `p1` at $\mathbf{e}_1$

   (b) Point `p2` at $\mathbf{e}_1 + \mathbf{e}_2$

   (c) Point `p3` at $2\mathbf{e}_1$

   (d) Point `p4` at $2\mathbf{e}_1 + 3\mathbf{e}_2$

   (e) Point `p5` at the centroid of `p1`, `p2`, and `p3`.

   (f) Point `p6` at $3\mathbf{e}_1 + 2\mathbf{e}_2$

   (g) The circle `C` through `p1`, `p2`, `p3`

   (h) Line `l1` through `p1` and `p2`.

   (i) Line `l2` through `p4` and `p6`.

   Now do the following exercises.

   (a) Compute the spherical (circular) inversion of `p4` through `C`.

   (b) Compute the spherical (circular) inversion of `l1` through `C`.

   (c) Compute the spherical (circular) inversion of `l2` through `C`.

   (d) Construct a line `l3` tangent to `c1` at `p2` and compute its spherical inversion in `C`.