

## 1 Quaternions

Quaternions are four-dimensional numbers. As the complex numbers have a real component and an imaginary component,  $i$ , quaternions have a real component and three imaginary components,  $i, j, k$ . In this program, quaternions are represented with a 4-vector. Depending on the version, the data type may be float, double, long double, or some other data type which represents real values.

An example of how a quaternion is represented:

$$1 + 4i - 2j + 1.125k \mapsto [1, 4, -2, 1.125]$$

## 2 Four-dimensional space

Since this is a four dimensional space, navigating this space is more complicated than simply using arrow keys to move in four cardinal directions. The method I have chosen to navigate one point through this space is to assign four keys to an increase in the coordinate for four dimensions, and to assign the four keys below those to a decrease in the corresponding coordinate.

To choose a seed from the Mandelbrot set, the following four dimensions of control are used

1. (A, Z): Move the seed in the (+, -) real direction.
2. (S, X): Move the seed in the (+, -)  $i$  direction.
3. (D, C): Move the seed in the (+, -)  $j$  direction.
4. (F, V): Move the seed in the (+, -)  $k$  direction.

## 3 Changing our view in Four-dimensional space

Each view has a center  $c$ , a horizontal vector  $h$  from the center to the right edge, and a vertical vector  $v$  from the center to the top edge. I have imposed the restrictions  $|h| = |v|$  and  $h \perp v$ , so that there is no stretching or skewing in the view. With these restrictions in place, representing each view is a 10-dimensional problem.

- Four dimensions to identify the center of the view.
- Four dimensions to identify a horizontal vector.
- Two dimensions to identify a vertical vector, given our restrictions.

Because of our  $h \perp v$  restriction,  $h \in \text{span}(v)^\perp$ , a 3-dimensional space. Because of our  $|h| = |v|$  restriction, this is further reduced to a 2-sphere of radius  $|h|$ . There are many ways that we can identify points on this sphere, with distinct advantages. As of version 3.3, the only implemented option is with a latitude and longitude system arranged such that  $(0^\circ, 0^\circ)$  corresponds to the  $i$ -axis when  $h$  is aligned with the real axis. The current implementation has some known flaws:

- The map is not continuous. When the real component of  $h$  changes from positive to negative,  $v$  becomes  $-v$ .
- There are two poles, at latitude  $\frac{\pi}{2}$  and  $-\frac{\pi}{2}$ , where a change in longitude has no effect.

As of version 3.3, the view is limited to a square, although the view could in principle be any aspect ratio, as long as  $h$  and  $v$  are scaled appropriately to prevent stretching.

Within this 10-dimensional problem, we need 10 dimensions of controls.

1. (1, Q): Move the center in the (+, -) real direction.
2. (2, W): Move the center in the (+, -)  $i$  direction.
3. (3, E): Move the center in the (+, -)  $j$  direction.
4. (4, R): Move the center in the (+, -)  $k$  direction.
5. (5, T): Adjust  $h$  in the (+, -) real direction.
6. (6, Y): Adjust  $h$  in the (+, -)  $i$  direction.
7. (7, U): Adjust  $h$  in the (+, -)  $j$  direction.
8. (8, I): Adjust  $h$  in the (+, -)  $k$  direction.
9. (9, O): Adjust the latitude (+,-).
10. (0, P): Adjust the longitude (+,-).