

Drawing 3D: Coordinates

**CSCI 4229/5229
Computer Graphics
Summer 2025**

Lorenz Observations

- Compute coordinates or store?
 - Readability over efficiency
- The attractor does not fit in $(-1,+1)$
 - Transform coordinates
- The attractor should remain the same when viewed from different positions
 - Recompute from fixed position or store
- Avoid IO (read or write)
 - Bad for performance in display
 - Bad user interface anywhere else

The Issue

- Lorenz attractor (x,y,z) values are generally in the range (-50,50) in all three dimensions
- How do you get OpenGL to display the whole range of values?
- Assume
 - Lorenz values are in an array $x[i], y[i], z[i]$, $i=0, \dots, n-1$
 - double *dim* is defined somewhere as 50

Option 1

```
void display()  
{...  
    for (i=0;i<n;i++)  
        glVertex3d(x[i]/dim , y[i]/dim , z[i]/dim);  
...}
```

Option 2

```
void display()  
{...  
    for (i=0;i<n;i++)  
        glVertex4d(x[i] , y[i] , z[i] , dim);  
...}
```

Option 3

```
void display()  
{...  
    glScaled(1/dim , 1/dim , 1/dim);  
    for (i=0;i<n;i++)  
        glVertex3d(x[i] , y[i] , z[i]);  
...}
```

Option 4

```
void display()
{...
    for (i=0;i<n;i++)
        glVertex3d(x[i] , y[i] , z[i]);
...}

void reshape()
{...
    glScaled(1/dim , 1/dim , 1/dim);
...}
```

Option 5

```
void display()
{...
    for (i=0;i<n;i++)
        glVertex3d(x[i] , y[i] , z[i]);
...}

void reshape()
{...
    glOrtho(-asp*dim,asp*dim,-dim,dim,-dim,dim);
...}
```