Max Kramer

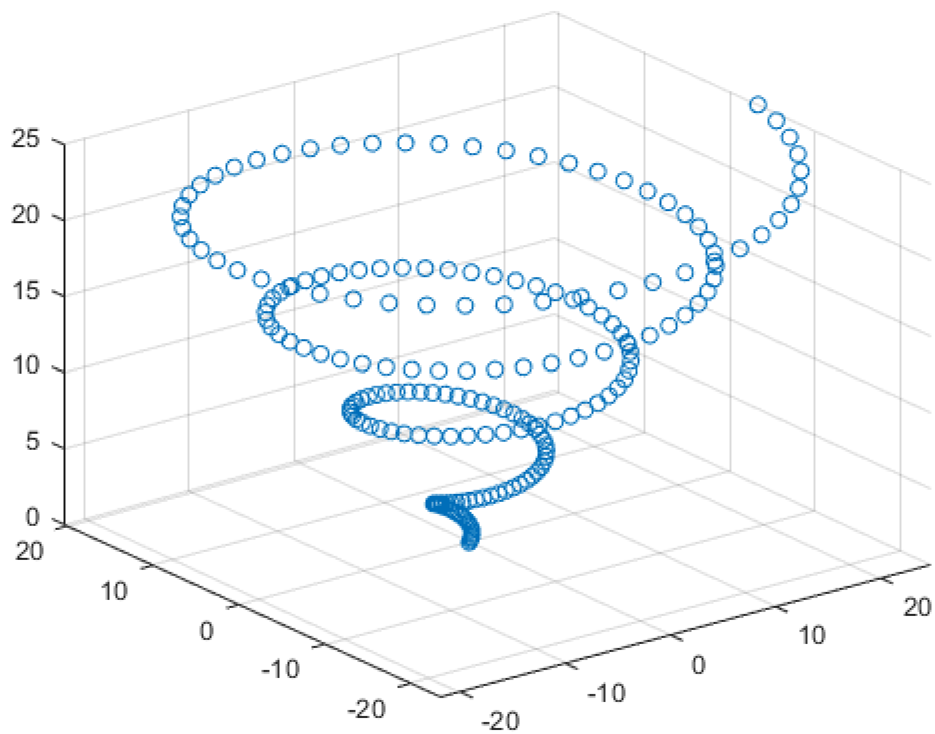
I affirm that I have adhered to the honor code on this assignment

Hello again, scientist! I'll do all my writing in italics, and problems for you will be in ***bold.*** *Comment your code, and* explain your ideas in plaintext. *As a general rule, I expect you to do at least as much writing as I do. Code should be part of your solution, but I expect variables to be clear and explanation to involve complete sentences. Cite your sources; if you work with someone in the class on a problem, that's an extremely important source. Don't work alone.*

Problem 12.3.

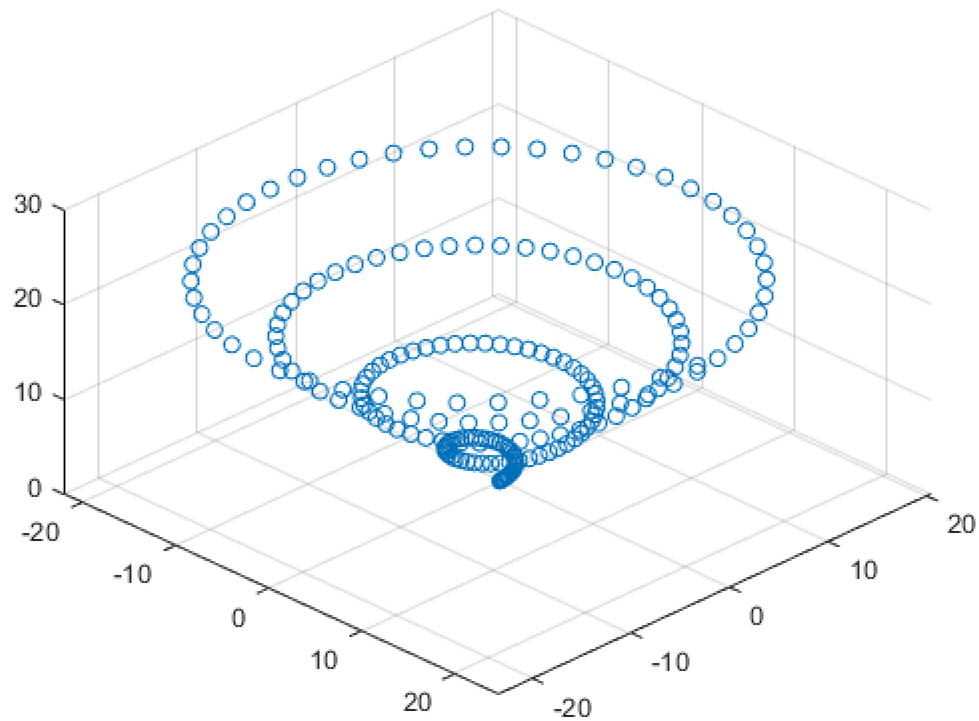
Reread Lay p.342, "An Orthogonal Projection."

```
C = csvread("corkscrew.csv");
scatter3(C(1,:),C(2,:),C(3,:));
```

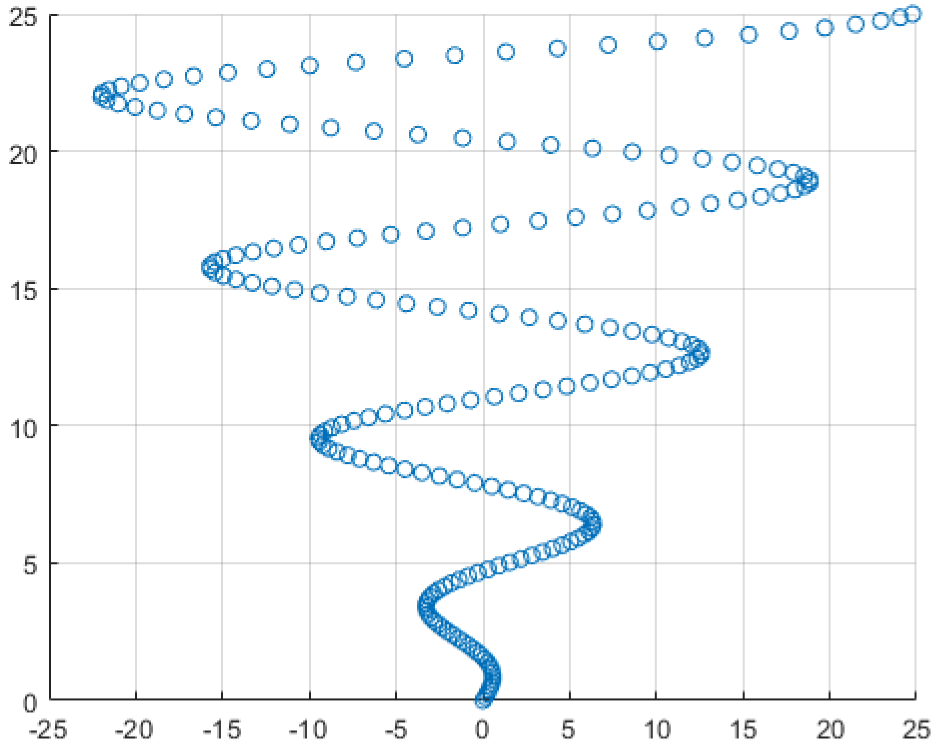


We can look at this picture from different angles too!

```
view([45 45])
```




```
view([0 0])
hold off;
```




```
v1 = [0;0;1]; v2 = [0;1;1]; v3=[1;1;-1];
```

Projection matrices are really cool! Let's see what they look like. ***Let Projv1 be the matrix that projects a point onto Span(v1), Projv2 be the matrix that projects onto Span(v2), and Projv3 be the matrix that projects onto Span(v3). Show me C, Projv1*C, Projv2*C, and Projv3*C on the same set of axes.*** *If you're stuck: a linear transformation is determined by its action on the standard basis. Compute Projv1(e1), Projv1(e2), Projv1(e3). I strongly recommend Running this code (hit F5) rather than just Publishing it so you can rotate the figure by hand to see it from different angles.*

```
Projv1 = [proj(v1,[1;0;0])';proj(v1,[0;1;0])';proj(v1,[0;0;1])']
Projv2 = [proj(v2,[1;0;0])';proj(v2,[0;1;0])';proj(v2,[0;0;1])']
Projv3 = [proj(v3,[1;0;0])';proj(v3,[0;1;0])';proj(v3,[0;0;1])']

C1 = Projv1 * C;
C2 = Projv2 * C;
C3 = Projv3 * C;

scatter3(C(1,:),C(2,:),C(3,:));
hold on
scatter3(C1(1,:),C1(2,:),C1(3,:));
scatter3(C2(1,:),C2(2,:),C2(3,:));
scatter3(C3(1,:),C3(2,:),C3(3,:));
```

```
Projv1 =

     0     0     0
```

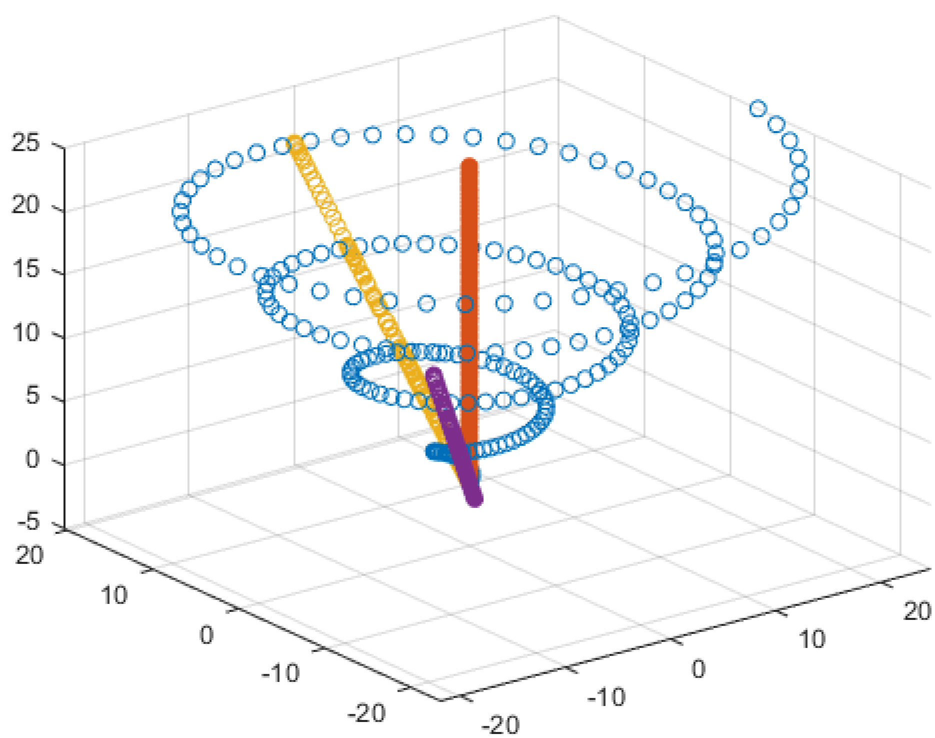
```
     0     0     0
     0     0     1


Projv2 =

         0         0         0
         0    0.5000    0.5000
         0    0.5000    0.5000


Projv3 =

    0.3333    0.3333   -0.3333
    0.3333    0.3333   -0.3333
   -0.3333   -0.3333    0.3333
```



I used the provided proj.m function to compute the action of each projection matrix on the standard basis for R^3. The resultant matrices, Projv1, Projv2, and Projv3 were then multiplied with C to get C1, C2, and C3. I then used the scatter3() function to show each matrix on the same axes.

Prove that v2 and v3 are orthogonal.

```
dot(v2,v3)
```

```
ans =

     0
```

The dot product of v2 and v3 (both in R^3) is 0. Therefore, they are orthogonal to each other.

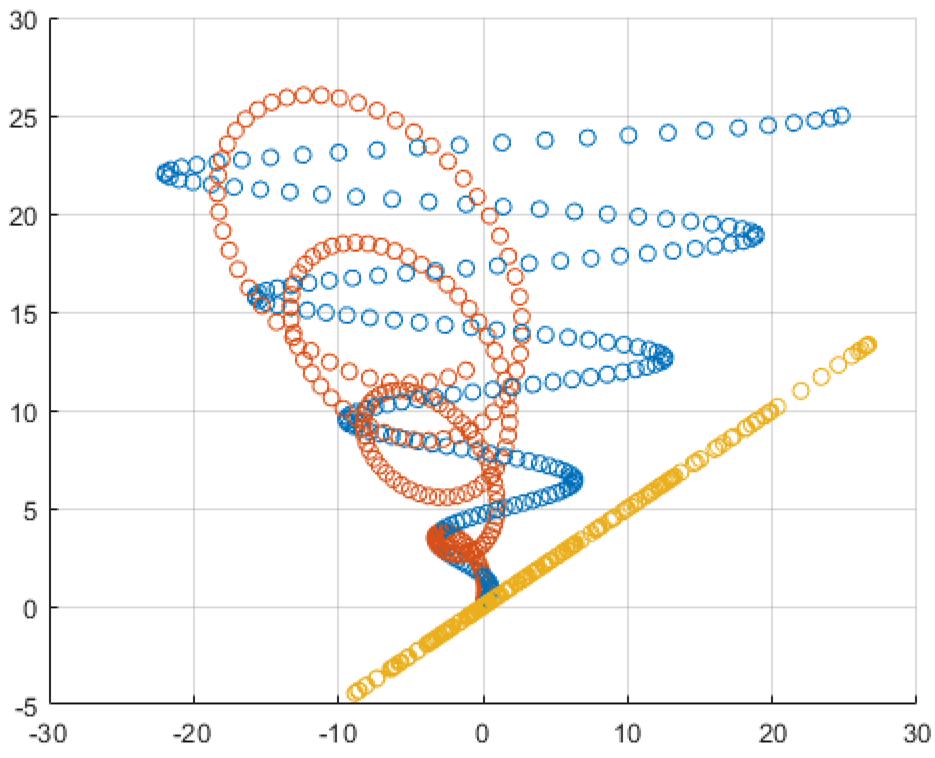
Reread Lay p.350. Notice that the formula in Theorem 8 is literally just the sum of the projection maps. Let W = Span(v2,v3), so ProjW = Projv2+Projv3. Plot C, ProjW*C, and (eye(3)-ProjW)*C on the same set of axes. *(I did this one in MATLAB, not quite knowing what it would look like, and said "that is the coolest shit!" aloud to nobody when I saw the answer. view([0 0]) produces a particularly striking figure, but you should look at it from other angles too!)*

```
figure;
scatter3(C(1,:),C(2,:),C(3,:));
hold on

ProjW = Projv2 + Projv3;
CW = ProjW * C;
scatter3(CW(1,:),CW(2,:),CW(3,:));

CEW = (eye(3)-ProjW) * C;
scatter3(CEW(1,:),CEW(2,:),CEW(3,:));

view([0 0])
```



The projection matrix ProjW is created by adding together Projv2 and Projv3. The projection matrices are then multiplied against C and plotted on the same axes. View([0 0]) was used because Steve said it would look cool, and it does.

Published with MATLAB® R2019b