Computational Geometry Seminar

Wichita State University, Dept. of Mathematics

Vector fields along space curves

Last changed: 3 Oct 2017

Authors: Jen Pinkston, Justin Ryan, Arelis Silva

In [1]:
```
%display latex
```

In [2]:
```
# define the curve
var('t x y z');
a(t) = [cos(t),sin(t),2*t];
V(t) = [t,1-t^2,1+t^2];
C = matrix([[-1,0,0],
          [0,1/sqrt(2),-1/sqrt(2)],
         [0,1/sqrt(2),1/sqrt(2)]]);
```

In [3]:
```
A = -pi;
n = 10;
tp = 2*A/(n-1);

p = {}
for i in range(n):
    p[i] = [a[0].substitute(t = A+tp*i),a[1].substitute(t = A+tp*i),a[2].substitute(t = A+tp*i)];

VV = {}
for i in range(n):
    VV[i] = [V[0].substitute(t=A+tp*i),V[1].substitute(t=A+tp*i),V[2].substitute(t=A+tp*i)];

var = {}
for i in range(n):
    var[i] = plot((1/3)*vector(VV[i]), start=vector(p[i]), color='blue', width='1.5');

vector_plot = sum(var[i] for i in range(n));
```

In [4]:

```
graph = parametric_plot3d(a,(t,-4*pi,4*pi),color='red');
show(graph + vector_plot)
```

In [5]:

```
Cp = {}
for i in range(n):
    Cp[i] = C*vector([a[0].substitute(t = A+tp*i),a[1].substitute(t = A+tp*i),a[2].substitute(t = A+tp*i)]);

CVV = {}
for i in range(n):
    CVV[i] = C*vector([V[0].substitute(t=A+tp*i),V[1].substitute(t=A+tp*i),V[2].substitute(t=A+tp*i)]);

Cvar = {}
for i in range(n):
    Cvar[i] = plot((1/3)*vector(CVV[i]), start=vector(Cp[i]), color='purple', width='1.5');

C_vector_plot = sum(Cvar[i] for i in range(n));
```

In [6]:
```
Ca = C*vector(a(t));
C_graph = parametric_plot3d(Ca,(t,-4*pi,4*pi),color='green');
```

In [7]:
```
show(graph + C_graph,aspect_ratio=1)
```

In [8]:
```
show(graph + C_graph + vector_plot + C_vector_plot, aspect_ratio=1)
```

In [9]:
```
%%octave

B = randi(10,3,3);
B
```

```
B =

 5 7 3
 9 4 2
 10 3 9
```

In [10]:
```
B = matrix([[5,1,2],
          [3,4,3],
          [7,3,10]]);
```

In [11]:
```
B.det()
```
Out[11]: 108

In [12]:
```
B = B/(det(B))^(1/3);
```

In [13]:
```
B
```
Out[13]:
$$\begin{pmatrix} \frac{5}{12} \cdot 4^{\frac{2}{3}} & \frac{1}{12} \cdot 4^{\frac{2}{3}} & \frac{1}{6} \cdot 4^{\frac{2}{3}} \\ \frac{1}{4} \cdot 4^{\frac{2}{3}} & \frac{1}{3} \cdot 4^{\frac{2}{3}} & \frac{1}{4} \cdot 4^{\frac{2}{3}} \\ \frac{7}{12} \cdot 4^{\frac{2}{3}} & \frac{1}{4} \cdot 4^{\frac{2}{3}} & \frac{5}{6} \cdot 4^{\frac{2}{3}} \end{pmatrix}$$

In [14]:
```
det(B)
```
Out[14]: 1

In [15]:
```
B = matrix([[1,2,3],
        [3,2,1],
        [-1,1,-1]]);
det(B)
```
Out[15]: 16

In [16]:
```
Bp = {}
for i in range(n):
   Bp[i] = B*vector([a[0].substitute(t = A+tp*i),a[1].substitute(t = A+tp*i),a[2].substitute(t = A+tp*
i)]);

BVV = {}
for i in range(n):
   BVV[i] = B*vector([V[0].substitute(t=A+tp*i),V[1].substitute(t=A+tp*i),V[2].substitute(t=A+tp*i)
]);

Bvar = {}
for i in range(n):
   Bvar[i] = plot((1/3)*vector(BVV[i]), start=vector(Bp[i]), color='cyan', width='1.5');

B_vector_plot = sum(Bvar[i] for i in range(n));
```

In [17]:
```
Ba = B*vector(a(t));
B_graph = parametric_plot3d(Ba,(t,-4*pi,4*pi),color='brown');
```

In [18]:

```
show(graph + C_graph + B_graph,aspect_ratio=1)
```

In [19]:
```
show(graph + B_graph + vector_plot + B_vector_plot,aspect_ratio = 1)
```

In [20]:
```
B = matrix([[t,0,0],
      [0,t^2,0],
      [0,0,1-t]]);
```

In [21]:
```
B
```

Out[21]:

$$\begin{pmatrix} t & 0 & 0 \\ 0 & t^2 & 0 \\ 0 & 0 & -t+1 \end{pmatrix}$$

In [22]:
```
B_star = matrix([[1,0,0],
            [0,2*t,0],
            [0,0,-1]]);
B_star
```

Out[22]:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2\,t & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

In [23]:
```
Bp = {}
for i in range(n):
   Bp[i] = B(t = A+tp*i)*vector([a[0].substitute(t = A+tp*i),a[1].substitute(t = A+tp*i),a[2].substitut
e(t = A+tp*i)]);

BVV = {}
for i in range(n):
   BVV[i] = B_star(t = A+tp*i)*vector([V[0].substitute(t=A+tp*i),V[1].substitute(t=A+tp*i),V[2].sub
stitute(t=A+tp*i)]);

Bvar = {}
for i in range(n):
   Bvar[i] = plot((1/3)*vector(BVV[i]), start=vector(Bp[i]), color='cyan', width='1.5');

B_vector_plot = sum(Bvar[i] for i in range(n));
```

In [24]:
```
Ba = B*vector(a(t));
B_graph = parametric_plot3d(Ba,(t,-4*pi,4*pi),color='brown');
```

In [25]:
```
show(graph + B_graph + vector_plot + B_vector_plot)
```

In [26]:
```
f1 = x^2;
f2 = y^3;
f3 = sqrt(z^2 + 1);
```

In [27]:
```
F = vector((f1,f2,f3))
```

In [28]:
```
F
```

Out[28]: $\left(x^2,\ y^3,\ \sqrt{z^2 + 1}\right)$

In [29]:
```
Fx = F.diff(x)
```

In [30]:
```
Fy = F.diff(y)
```

In [31]:
```
Fz = F.diff(z)
```

In [32]:
```
JF = matrix([[Fx[0],Fy[0],Fz[0]],
           [Fx[1],Fy[1],Fz[1]],
           [Fx[2],Fy[2],Fz[2]]]);
```

In [33]:
```
JF
```

Out[33]:

$$\begin{pmatrix} 2x & 0 & 0 \\ 0 & 3y^2 & 0 \\ 0 & 0 & \frac{z}{\sqrt{z^2+1}} \end{pmatrix}$$

In [34]:
```
p = (1,2,3);
V = vector([2,-3,1]);
```

In [35]:
```
Fp = F(x=1,y=2,z=3)
```

In [36]:
```
JF_p = JF(x=1,y=2,z=3);
JF_p
```

Out[36]:

$$\begin{pmatrix} 2 & 0 & 0 \\ 0 & 12 & 0 \\ 0 & 0 & \frac{3}{10}\sqrt{10} \end{pmatrix}$$

In [37]:
```
VFp = JF_p*V
```

In [38]:
```
show(plot(point((1,2,3),size=10,color='red'))+ plot(V,start=vector(p),color='blue') +
    plot(point(Fp,size=10,color='purple')) +plot(VFp,start=vector(Fp),color='green') )
```

In [39]:
```
jacobian(F,(x,y,z))
```

Out[39]:

$$\begin{pmatrix} 2\,x & 0 & 0 \\ 0 & 3\,y^2 & 0 \\ 0 & 0 & \frac{z}{\sqrt{z^2+1}} \end{pmatrix}$$

In [ ]: