

Math 4530
Computing curvature and torsion
Wednesday, January 26 2005

These procedures are like those in our text, but may not be exactly the same....(Maple has added subroutines for dot and cross product, for example, so that you don't need to write your own like I did in this file from three years ago.)

```
[ > restart:
[ > dp:=proc(X,Y)  #dotproduct
[   X[1]*Y[1]+X[2]*Y[2]+X[3]*Y[3];
[   end:
[ > x:=[a,b,c]:  #example
[   y:=[r,s,t]:
[   dp(x,y);
[
[                                      $a\,r+b\,s+c\,t$ 
[ > nrm:=proc(X)  #magnitude (norm)
[   sqrt(dp(X,X));
[   end:
[ > nrm(y);
[
[                                      $\sqrt{r^2+s^2+t^2}$ 
[ > xp:=proc(X,Y)  #cross product
[   local a,b,c;
[   a:=X[2]*Y[3]-X[3]*Y[2];
[   b:=X[3]*Y[1]-Y[3]*X[1];
[   c:=X[1]*Y[2]-X[2]*Y[1];
[   [a,b,c];
[   end:
[ > xp(x,y);
[
[                                      $[b\,t-c\,s, c\,r-t\,a, a\,s-b\,r]$ 
[ > curv:=proc(alpha)  #curvature
[   local alphap,alphapp,num,denom;
[   alphap:=diff(alpha,t);
[   alphapp:=diff(alphap,t);
[   num:=simplify(nrm(xp(alphap,alphapp)),radical,symbolic);
[   denom:=nrm(alphap)^3;
[   RETURN(kappa=simplify(num/denom));
[   end:
[ > tor:=proc(alpha)
[   local alphap,alphapp,alphappp,num;
[   alphap:=diff(alpha,t);
[   alphapp:=diff(alphap,t);
[   alphappp:=diff(alphapp,t);
[   num:=dp(xp(alphap,alphapp),alphappp);
[   RETURN(tau=simplify(num/nrm(xp(alphap,alphapp))^2));
[   end:
[ Helicoid, we just did:
[ > hel:=[a*cos(t),a*sin(t),b*t];
```

```
curv(hel);tor(hel);
```

$$hel := [a \cos(t), a \sin(t), b \ t]$$

$$\kappa = \frac{a}{b^2 + a^2}$$

$$\tau = \frac{b}{b^2 + a^2}$$

[>