Recall Frenet system for a *p. b. c.*
\[
\begin{align*}
\alpha' &= T \\
T' &= \kappa N \\
N' &= \kappa T + \tau B \\
B' &= -\tau N
\end{align*}
\]

- As the attached handout shows, computers can solve this first order system to recreate curves having prescribed curvature and torsion!

For curves in the *x-y* plane, we can derive a simplified Frenet system (since \(B=\hat{b}\) is not an issue.) Even better, we can define \(\vec{N}\) globally \(= \pm\) space curve \(N\), as:
\[
\vec{N} = R_{\frac{\tau}{2}}(\vec{T})
\]

rotate \(\vec{T}\) by \(\frac{\tau}{2}\) counterclockwise.

Since \(\vec{T}\) \(=\) \(T\) still holds, \(\vec{N} = R_{\frac{\tau}{2}}(\vec{T})\)

\[
\text{deduce } T' = \kappa N \\
\kappa = \text{can be +, - , 0!}
\]

If we write \(T = \langle \cos \Theta(s), \sin \Theta(s) \rangle\)
then \(N = \langle -\sin \Theta, \cos \Theta \rangle\)
so \(T' = \langle -\sin \Theta, \cos \Theta \rangle \Theta'\)

\[
\text{Deduce } K = \Theta'(s)
\]

Plane Frenet system: (for curve with plane curvature \(K\))
\[
\begin{align*}
\Theta'(s) &= K(s) \\
\Rightarrow \Theta(s) &= \Theta_0 + \int_0^s K(r) \, dr \\
\alpha' &= T = \langle \cos \Theta(s), \sin \Theta(s) \rangle \text{ now known.}
\end{align*}
\]

\[
\text{so } \alpha(s) = \alpha_0 + \int_0^s T(r) \, dr
\]

- See second Maple Handout!
Computing curvature and torsion for given curves (not parallel).
Use the roller coaster eqn:

\[ \alpha''(t) = \frac{v}{v'} \vec{T} + \kappa v^2 \vec{N} \]

\( v = |\alpha'(t)| = \text{speed} \)
(can you rederive this?)

• Pick off formula for \( \kappa \) by crossing with \( \alpha''(t) \)

• Notice roller coaster also holds for plane curves. What is the curvature of a plane curve?

Use:

\[ \langle x'', y'' \rangle = \frac{v}{v'} \langle x'y' \rangle + \kappa v^2 \langle -y', x' \rangle \]

Differentiate roller coaster:

• \( \alpha'''(t) = \frac{v''}{v'} \vec{T} + \frac{v}{v'} \frac{dv}{ds} \frac{ds}{dt} \vec{T} + (\kappa v^2) \frac{N}{v} + (\kappa' v^2) \frac{N}{ds} \frac{ds}{dt} \vec{T} \)

\[ \alpha'''(t) = \frac{v''}{v'} \vec{T} + \frac{v}{v'} \frac{dv}{ds} (\kappa \vec{N}) + (\kappa' v^2) \frac{N}{N} + \kappa v^3 \left(-X \vec{T} + \kappa \vec{B} \right) \]

pick off \( \vec{T} \) from this eqn:
Solving the Frenet System,
for prescribed curvature and torsion functions
Math 4530 Spring 05

> restart:
> with(DEtools):with(plots):
  #need DEtools to solve differential equations
  Warning, the name changecoords has been redefined

Here's a pretty self-explanatory procedure which solves the Frenet system, taken more or less from the text.

> recreate3dview:=proc(kap,t,a,b,c,d,e,f,g,h)
  #kap=curvature, ta=torsion
  #arclength parameter from a to b
  #c..d, e..f, g..h are x-y-z ranges for plot
  local
    sys, #the Frenet system
    p, #dummy for ODE solution to Frenet system
    ics, #initial conditions
    pl; #name for ODEplot of p
  sys:=
    diff(alph1(s),s)=T1(s),
    diff(alph2(s),s)=T2(s),
    diff(alph3(s),s)=T3(s),
    diff(T1(s),s)=kap(s)*N1(s),
    diff(T2(s),s)=kap(s)*N2(s),
    diff(T3(s),s)=kap(s)*N3(s),
    diff(N1(s),s)=-kap(s)*T1(s)+ta(s)*B1(s),
    diff(N2(s),s)=-kap(s)*T2(s)+ta(s)*B2(s),
    diff(N3(s),s)=-kap(s)*T3(s)+ta(s)*B3(s),
    diff(B1(s),s)=-ta(s)*N1(s),
    diff(B2(s),s)=-ta(s)*N2(s),
    diff(B3(s),s)=-ta(s)*N3(s);
  ics:=
    alph1(0)=0, alph2(0)=0, alph3(0)=0,
    T1(0)=1, T2(0)=0, T3(0)=0,
    N1(0)=0, N2(0)=1, N3(0)=0,
    B1(0)=0, B2(0)=0, B3(0)=1;
  p:=dsolve({sys,ics},{alph1(s),alph2(s),alph3(s),
    T1(s),T2(s),T3(s),N1(s),N2(s),N3(s),
    B1(s),B2(s),B3(s)},type=numeric);
  pl:=odeplot(p,[alph1(s),alph2(s),alph3(s)],a..b,
    numpoints=200,thickness=1,axes=boxed,color=black):
  display(pl,scaling=constrained,view=[c..d,e..f,g..h]);
end:

Here are some examples:
Example 1: A helix, with constant curvature and torsion
> kap1:=s->2;
  tor1:=s->.5;

  kap1 := 2
\[ torl := 0.5 \]
\[ > \text{rec} \text{re} \text{ate3dview}(\text{kapl, torl, 0, 20, -1, 3, -1, 3, 0, 4}); \]

Example 2:
\[ > \text{kapl} := s \rightarrow 0.2 \times s; \]
\[ \text{torl} := s \rightarrow 0.5; \]

\[ k\text{apl} := s \rightarrow 0.2 \times s \]
\[ t\text{orl} := 0.5 \]
\[ > \text{re} \text{c} \text{re} \text{ate3dview}(\text{kapl, torl, 0, 20, 0, 10, -5, 5, -5, 5}); \]
Plane Curves, with prescribed planar curvature
Math 4530-1
Wednesday January 26

For plane curves we can define the normal vector globally as the rotation by $+\pi/2$ of the unit tangent vector, and now there is no restriction that the curvature never be zero. This leads to a simpler Frenet system than for space curves, and you can recreate interesting curves:

```maple
> restart;
> with(DEtools):with(plots):
Warning. the name changecoords has been redefined

> kap:=t->t;  # planar curvature function
a:=-5:b:=5:  # s-range
c:=-2:d:=2:  # x-range
f:=-2:g:=2:  # y-range
> sys:=
    diff(theta(s),s)=kap(s),
    diff(b1(s),s)=cos(theta(s)),
    diff(b2(s),s)=sin(theta(s)):
# this is the plane-curve Frenet system
ics:=
    theta(0)=0,  # start flat
    b1(0)=0,     # start at origin
    b2(0)=0:
p:=dsolve({sys,ics},{theta(s),b1(s),b2(s)},
type=numeric):
p1:=odeplot(p,[b1(s),b2(s)],a..b,numpoints=400,
    thickness=1,axes=framed,color=black):

> display(p1,view=[c..d,f..g],title=`kappa(s)=s`);
```
> kap:=t->t*sin(t):  # planar curvature function
> a:=-8:b:=8:  # s-range
> c:=-2:d:=2:  # x-range
> f:=0:g:=4:  # y-range
> sys:=
>     diff(theta(s),s)=kap(s),
>     diff(b1(s),s)=cos(theta(s)),
>     diff(b2(s),s)=sin(theta(s)):
> ics:=
>     theta(0)=0,  # start flat
>     b1(0)=0,  # start at origin
>     b2(0)=0:
> p:=dsolve({sys,ics},{theta(s),b1(s),b2(s)},
>              type=numeric):
> pl:=odeplot(p,[b1(s),b2(s)],a..b,numpoints=400,
>              thickness=1,axes=framed,color=black):
> display(pl,view=[c..d,f..g],title='kappa(s)=s*sin(s)');

![plot of kappa(s)=s*sin(s)](image-url)