Solving the Frenet System,  
for prescribed curvature and torsion functions  
Math 4530  Spring 02

[ > restart:

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,ta,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
        p1;  #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=numerical);
    p1:=odeplot(p,[alph1(s),alph2(s),alph3(s)],a..b,
        numpoints=200,thickness=1,axes=boxed,color=black):
    display(p1,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*s;
    tor1:=s->.5;
    kap1 := s → .2 s
\[ tor1 := 0.5 \]

\[ > \text{recreate3dview}({\text{kap1}}, \{\text{tor1}, 0, 20, -2, 8, -5, 5, -5, 5\}); \]