

```

>> a = [1,2,3,4,5];
>> b = [1,1,2,6,24];
>> c = 1:0.1:5;
>> cs = cubicspline(a,b,c);
>> ndd = NDD(a,b,c);
>> g = gamma(c);
>> plot(c,cs,'red',c,ndd,'black',c,g,'blue')
>> plot(c,abs(cs-g),'red',c,abs(ndd-g),'black')
>> %the ndd had a more consistent error for the first 3 sub intervals, only really
blowing up after the 4
>> %this is makes sense because this area should be close to linear which doesn't
mesh well with how ndd forms its polynomial
>> %for the cubic spline, the last interval caught me off guard intially but when i
thought back to what not a knot is doing
>> %it makes more sense because it would limit the third derivative to 0 which isn't
going to be the case for the underlying function
>> a = [0.1,0.15,0.2,0.3,0.35,0.5,0.75];
>> b = [3,2,1.2,2.1,2.0,2.5,2.5];
>> c = 0.05:0.01:0.8;
>> cs = cubicspline(a,b,c);
>> ndd = NDD(a,b,c);
>> plot(c,cs,'red',c,ndd,'black')
>> plot(c,ndd,'black')
>> plot(c,cs,'black')
>> plot(c,ndd,'black')
>> plot(c,ndd,'black',a,b,'o')
>> ndd = NDD(a,b,c);
File: NDD.m Line: 17 Column: 186
Invalid expression. When calling a function or indexing a variable, use parentheses.
Otherwise, check for mismatched delimiters.

>> ndd = NDD(a,b,c);
File: NDD.m Line: 17 Column: 186
Invalid expression. When calling a function or indexing a variable, use parentheses.
Otherwise, check for mismatched delimiters.

>> ndd = NDD(a,b,c);
>> plot(c,ndd,'black',a,b,'o')
>> plot(c,cs,'red',c,ndd,'black')
>> % the ndd varies wildly between our data points compared to cubic splines, showing
the benefits of cubic splines, and why generally they are the better approach for
looking at more realistic data
>> %the near linear nature between some of the points highlighted the struggle of
polynomial interpolants have to fit these types of data
>> %I am making assumptions about the data here, but it is not unreasonable given the
lack of derivative information
>>

```