spline curves

A planar curve is, by definition, the continuous image of some interval \([a..b]\) in the plane, i.e., a planar set of the form

\[
\{ f(t) : a \leq t \leq b \}
\]

with \(f\) a continuous function from some interval \([a..b]\) into the plane \(\mathbb{R}^2\). As the argument \(t\) moves from \(a\) to \(b\), the point \(f(t)\) in the plane traces out the curve. Many different functions \(f\) give rise to the same curve. E.g., the functions

\[
g : [a..b] \rightarrow \mathbb{R}^2 : t \mapsto f((a + b) - t)
\]

and

\[
h : [0..1] \rightarrow \mathbb{R}^2 : t \mapsto f(a + t(b - a))
\]

trace out the same curve as \(f\). Each such continuous is said to be, or to provide, a parametrization for the curve it traces.

Given some sequence \(o_1, \ldots, o_n\) of points in the plane, we can certainly use polynomial interpolation or our various spline interpolation techniques to construct a curve that passes through these points. For example,

\[
P_1(t) := o_1 + t(o_2 - o_1)
\]

traces out the straight line that passes through the two points \(o_1\) and \(o_2\). In effect, \(P_1\) is a polynomial with vector coefficients, i.e., a vector-valued polynomial.

If we put our plane point sequence \(o_1, \ldots, o_n\) into the \(2 \times n\) matrix \(o\), then the MATLAB command \(\text{curve} = \text{spline}(1:n,o)\); provides a spline curve through these points, and we could plot that curve then as follows

```matlab
points = ppval(curve,linspace(1,n));
plot(points(1,:),points(2,:));
```

If we had some feeling about the tangents at the ends of that curve, we could use complete spline interpolation. For example, to get a pretty good circle, we could use the following:

```matlab
x = linspace(0,2*pi,5);
circle = spline(x,[0 1 0 -1 0 1 0; ...
   1 0 1 0 -1 0 1], linspace(0,2*pi));
plot(circle(1,:),circle(2,:));
axis equal
```

Here, we have specified the vector \([0;1]\) as the tangent vector at the two endpoints which is the correct direction and length, given that we are presumably running once around the unit circle as the parameter moves from 0 to \(2\pi\), hence the speed should be 1. If we had used \(x = 0 : 4\) instead, the speed should have been \((2\pi)/4 = \pi/2\), i.e., the tangent vector should have been \([0; \pi/2]\).