

# Matlab Notes, Math 250

## Splines and Handwriting

November 5, 2003

In this lab, we take a look at how splines can be used to construct a set of parameterized curves to represent a handwriting sample.

Ideally, one will take readings directly from a digital tablet so that time can be explicitly read (velocity plays a role in handwriting analysis). In this case, it would be easy to obtain a set of data points for the handwriting sample.

We can digitize handwriting directly from an image, although it is a lot more work, and we do not get time values for velocity. As an example, the image of John Hancock's signature is widely available, and we'll see how to digitize it into Matlab.

A grayscale image is downloadable from the class website as `hancock2.jpg`. Download it into a local directory. We now display the image in Matlab using:

```
X=imread('hancock2.jpg');
X=double(X); %The original image is saved as integer values
image(X);
colormap(gray);
```

Matlab has a built-in feature to read coordinates off the mouse. We will go to the image and click on the desired data points in the image, and they will be read into Matlab.

```
[A,B]=ginput;
```

When you are finished inputting the data, press "Enter" on the keyboard. You will now have the data points stored in  $A, B$ . You can check to see what the data points look like on the image:

```
hold on;
plot(A,B,'r*');
```

In getting the data, we should get only enough data so that the splines can do their work—too much data, and you'll basically just get something similar to a linear piecewise curve.

Now that the data has been input, use `whos` to determine how many data points you have gotten, and remember it (here, we'll say that we have  $n$  data points).

Now, plot  $A$  as a series of points, and use the Basic Fitting tools to get a spline representation. Open the third panel of the GUI, and we'll evaluate the spline at a lot of intermediate points. In the  $X$ -coordinate window, type something like:

```
1:0.1:n
```

where  $n$  is the number of data points total (this is taking the place of time). Now you can save this data directly (although you can also save the splines). Suppose we save it using the default values,  $x1$  and  $fx1$ . Do the same for  $B$ , saving the data as  $x2$  and  $fx2$ .

We could simply plot our curves, but its nice to be able to lay our spline generated signature over the image. To do this, close the basic fitting tools and the figure.

```
image(X);  
colormap(gray);  
hold on  
plot(fx1,fx2,'r-');
```

And you'll be able to compare the two signatures, as shown below.

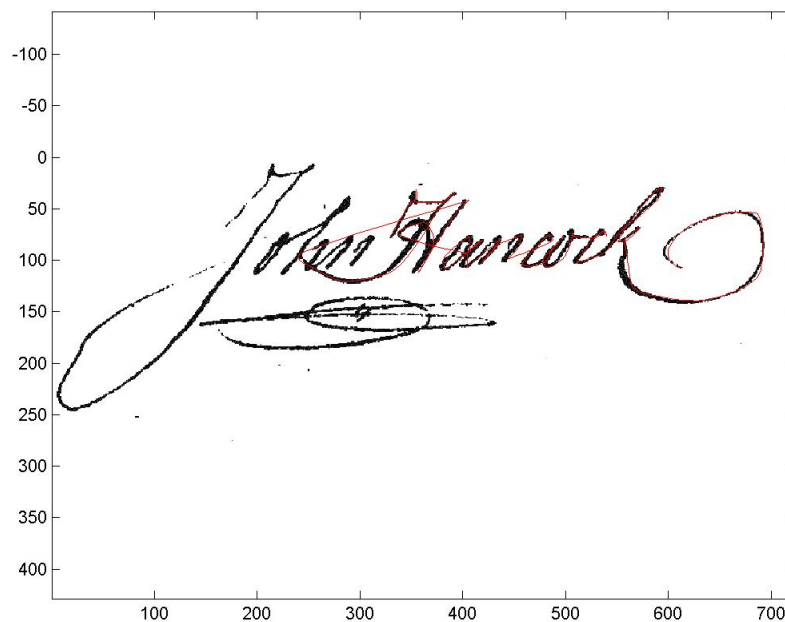


Figure 1: A comparison of an actual signature together with the spline approximation (note that we had to “connect the dots”, otherwise we would have needed separate splines for disconnected characters).