MCS 471 Project Four: Fitting Data with Maple

The goal is to study several methods to fit data, and in particular the use of splines to approximate functions, smoothen data and geometrical design. See http://www.math.uic.edu/~jan/mcs471/index.html for the hypertext version of this project and to download the Maple worksheet with the example commands.

0. The commands in the Maple package CurveFitting

Maple provides a convenient interface to four different types of fitting data:

1. interpolation with a polynomial;
2. least squares approximation with a polynomial of a given degree;
3. rational interpolation;
4. smoothening with cubic splines.

Below we give a very simple illustration of these four ways of fitting, mainly to illustrate how to use the commands. As example, we sample `n+1` equidistant points in `[a,b]` from the exponential function, using a floating-point evaluation of three decimal places:

```
[> n := 9: a := -1: b := +1: dx := (b-a)/n:
```

The `(epx,efx)` forms the exact, symbolic data set:

```
[> epx := Vector([seq(a+(k-1)*dx,k=1..n+1)]):
[> efx := map(exp,epx):
```

The numerical data set is accurate up within three decimal places:

```
[> npx := evalf(epx,3): nfx := evalf(efx,3):
```

First we construct an interpolating polynomial through all the sampled points:

```
[> ip := CurveFitting[PolynomialInterpolation](npx,nfx,x);
```

The advantage of least squares is that we can fit with a lower degree polynomial and still obtain accurate (if not more accurate) results:

```
[> degree_of_fit := 5:
[> form := sum(c[i]*x^i,i=0..degree_of_fit);
[> lp := CurveFitting[LeastSquares](npx,nfx,x,curve=form);
```

One quirk about rational approximation in Maple is that it only accepts exact rational numbers. While what we are about to do is not really good practice, we do it anyway, converting floating-point numbers to exact rational numbers...

```
[> rnpx := convert(npx,rational): rnfx := convert(nfx,rational):
[> rp := CurveFitting[RationalInterpolation](rnpx,rnfx,x);
```

Our fourth approximation is a with cubic spline, observe the output: we obtain a piecewise defined polynomial of degree three:

```
[> sp := CurveFitting[Spline](npx,nfx,x);
```

To compare the quality of the approximations, we plot the error function as the difference between the exact function we approximate and the approximating function. We observe not only the magnitude of the errors, but also the shape of the error function, bearing in mind that our data was taken with limited precision.

```
[> plot(exp(x)-ip,x=a..b); # error of interpolating polynomial
[> plot(exp(x)-lp,x=a..b); # error of least-squares polynomial
[> plot(exp(x)-rp,x=a..b); # error of rational interpolating polynomial
[> plot(exp(x)-sp,x=a..b); # error of cubic spline approximation
```

1. Assignment One: Approximation of Functions

One application of fitting is to replace a complicated function by a simpler one. Consider for example $f(t) = \int_0^t e^{(-\sin(x)^2)} dx$, for which there is no symbolic antiderivative. In Maple, we type:

```
[> f := int(exp(-(sin(x))^2),x=0..t); plot(f,t=0..10);
```

The plot over the interval $[0, 10]$ looks fairly simple. The goal of this assignment is to compare the four different methods in the project to fit $f(t)$ over the interval $[0, 10]$. There are two parts in this assignment:

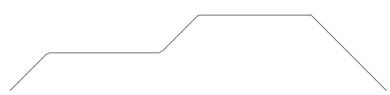
Assignment One Part I: constructing the approximations. Just like in the example in the previous section, construct four approximations to $f(t)$, over the interval $[a, b] = [0, 10]$, using 10 function evaluations. To see if the approximations get any better when increasing the working precision of evaluation, evaluate the integral for $f(t)$ with various precisions: with 3, 5, 10, and 20 decimal places. Look what happens to the magnitude of the error function and compare shapes of the errors.

Assignment One Part II: discussing the approximations. Compare the four methods with each other. The two main criteria to be used in the comparison are accuracy and efficiency. List for every method at least one advantage and at least one disadvantage. Illustrate referring to the results obtained for the four approximations of this function $f(t)$.

2. Assignment Two: Smoothening Data and Geometric Design

A second application of fitting (for which splines are very suitable) is to make given data smoother. Consider the following Maple statements:

```
[> lx := [0,1,4,5,8,9,10]: ly := [0,1,1,2,2,1,0]:
[> nlx := evalf(lx,3): nly := evalf(ly,3):
[> sp1 := CurveFitting[Spline](nlx,nly,x,degree=1):
[> plot(sp1,x=0..10,scaling=constrained,axes=none);
```

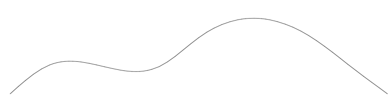


With some imagination, we can recognize a profile of a car. The integral of the profile provides a measure for the amount of metal we would need to produce the car:

```
> int(sp1,x=0..10);
```

But the shape is so unattractive that no one would want to drive a car with such a profile. Also the aerodynamic properties are very bad. Here is a naive first application of a cubic spline:

```
[> sp3 := CurveFitting[Spline](nlx,nly,x,degree=3):
[> plot(sp3,x=0..10,scaling=constrained,axes=none);
[> int(sp3,x=0..10);
```



This naive smoothening has given another extreme profile, and the value for the integral has also increased significantly. The goal of this assignment is to take more points to arrive at a smoother profile, keeping approximately the same value of the integral. In your answer, give all points used in the construction of a smoother profile, the plot, and the value of the integral.

3. Deadline is Friday 14 November, at 1PM

Bring your project solution to class. The plots are very important in the answers of the assignments. The solution of the project could be organized as the print out of a Maple worksheet, provided the worksheet has a very clean and clear structure. In general, the output of the calculations should be listed as an appendix to the actual solution of the project.

If you have questions, comments, or difficulties, feel free to come to my office for help.