Project 17

Lagrange Interpolation and Goodness of Fit

For this project you will need familiarity with the commands:

\[
\begin{array}{ccc}
\text{fsolve} & \text{plot} & \text{taylor} \\
\text{subs} & \text{interp} & \text{diff} \\
\text{map} & \text{convert} & \text{seq} \\
\text{int} & \text{Int} & \text{evalf} \\
? & \end{array}
\]

In connection with the \texttt{plot} command, you will need to use both the \textit{plot ranges} and \textit{multiple plots} capabilities. The form \texttt{name := [...]} will be used to form a \textit{list}. The \texttt{polynom} option of \texttt{convert} will be used.

\section*{Background}

In this session, we will concern ourselves with the question of finding a polynomial that passes through a prescribed set of points. From analytic geometry we know that

- 2 points determine exactly one line,
- 3 points determine at most one parabola in $x$,
- 4 points determine at most one cubic polynomial in $x$.

In general

\[ n + 1 \text{ points determine exactly one polynomial function of degree at most } n. \]

Since a polynomial of degree $n$ has $n+1$ coefficients, by substituting $n+1$ different values for $x$ into a general polynomial and setting the resulting expressions equal to $n+1$ corresponding values of $y$, we get $n+1$ linear equations. If the resulting system is nondegenerate, then one may solve for the $n+1$ coefficients. In principle, the
question is answered. However, solving large systems of equations is a very expensive (computations, time, and memory) process. As a matter of practical computation, other approaches are used.

We will use a “divide and conquer” scheme. Let the \( n + 1 \) data points be given as \((x_i, y_i)\). For \( x_1 \), build a polynomial \( p \) for which \( p(x_1) = y_1 \) but \( p(x_2) = p(x_3) = \ldots = p(x_{n+1}) = 0 \). Repeat the process for \( x_2 \) to build a polynomial \( q \) for which \( q(x_1) = 0, q(x_2) = y_2, \) and \( q(x_3) = \ldots = q(x_{n+1}) = 0 \). Continue this process for the remaining pairs \((x_i, y_i)\). Summing the resulting polynomials, we obtain the Lagrange\(^1\) interpolating polynomial. The points used are called knots.

Consider the following example.

```plaintext
> xdata := [1, 2, 3]:
> ydata := [0.841, 0.909, 0.141]:

Any polynomial with zeroes at \( x = 2 \) and \( x = 3 \) is a multiple of

\[
\text{first} := (x-2)(x-3)
\]

Similarly:

\[
\text{second} := (x-1)(x-3)
\]

\[
\text{third} := (x-1)(x-2)
\]

We we build a quadratic from \text{first} taking on the value 0.841 when \( x = 1 \).

\[
\text{First} := 0.841 / subs(x=1, first)
\]

\[
First := 0.4205000000 (x - 2) (x - 3)
\]

From \text{second} and \text{third} we likewise obtain:

\[
\text{Second} := -0.909 (x - 1) (x - 3)
\]

\[
\text{Third} := 0.141 / subs(x=3, third) \times \text{third}
\]

\[
Third := 0.07050000000 (x - 1) (x - 2)
\]

\(^1\)Joseph L Lagrange (1736–1813), among the first to prove the Mean Value Theorem, was one of the founders of the Calculus of Variations.
Having divided, we conquer:

\[ \text{Lagrange} := \text{expand(First+Second+Third)} \]

\[ \text{Lagrange} := -0.4180000000 x^2 + 1.322000000 x - 0.0630000000 \]

A quick check to see that Lagrange works:

\[ \text{seq(ydata[j] = subs(x=xdata[j], Lagrange), j=1..3)} \]

\[ 0.841 = 0.8410000000, 0.909 = 0.9090000000, 0.141 = 0.1410000000 \]

In Maple this process is automated by

\[ L := \text{interp(xdata, ydata, x)} \]

\[ L := -0.4180000000 x^2 + 1.322000000 x - 0.0630000000 \]

This is a familiar looking polynomial.

To assess Lagrange polynomials, we compare the function

\[ f := x \rightarrow \frac{x}{x^2 + 1} \]

over the domain \( 0 \leq x \leq 4 \) with different interpolating polynomials. Begin by defining \( f \) as a Maple function and plotting \( f \) with domain \([-1..5] \) and range \([-1..1] \). Now enter

\[ f := x \rightarrow x/(x^2+1) \]

Calculate the list of corresponding \( y_i = f(x_i) \) using the Maple command \text{map}. The \text{map} command applies the function \( f \) to each element in the list of \( x \) values.

\[ ydata := \text{map(f, xdata)} \]

Obtain a Lagrange interpolating polynomial of degree 2 with

\[ L_2 := \text{interp(xdata, ydata, t)} \]

Compare \( L_2 \) to \( f \) by plotting with domain \([-1..5] \) and range \([-1..1] \).

\textbf{Question 1} Since \( f(x) \) has a horizontal asymptote and \( L_2 \) is a parabola, what can you conclude about the goodness of fit for large values, positive or negative, of \( x \)?

By increasing the number of data points, there will be more values where \( f(x) \) and the interpolating polynomial coincide. Let’s try a fourth-degree polynomial; this requires five knots.

\[ xdata := [0, 1, 2, 3, 4] \]

\[ ydata := \text{map(f, xdata)} \]

\[ L_4 := \text{interp(xdata, ydata, t)} \]

Produce a plot of \( L_4 \) with the same scale as those of \( f(x) \) and \( L_2 \).

\textbf{Question 2} Is \( L_4 \) generally closer to or farther from \( f(x) \) than \( L_2 \)

1. over the domain \([0..4]\)?

2. for large values, positive or negative, of \( x \)?
Once more, increase the number of knots, however, not equally spaced over the interval. Place more knots where the function changes more rapidly.

```maple
> xdata := [0, 1/3, 2/3, 1, 3/2, 2, 3, 4]
> ydata := map(f, xdata)
> L7 := interp(xdata, ydata, t)
```

Plot $L_7$ in the same fashion as $L_2$ and $L_4$.

**Question 3** Is $L_7$ an improvement in fit over $L_4$? How well does $L_7$ fit $f(x)$ where the knots are more dense? Where the knots are few?

Finally, compute the twelfth degree approximation with equally spaced knots.

```maple
> xdata := [seq(k/3, k=0..12)]
> ydata := map(f, xdata)
> L12 := interp(xdata, ydata, t)
```

Plot $L_{12}$ and $f(x)$ as before.

**Project Report**

Is $L_{12}$ worth the effort of producing? Is $L_{12}$ significantly better than $L_4$ or $L_7$? There are two ways to answer these questions.

The first is to ask what is the worst possible error associated with using $L_k$ to approximate $f := x \rightarrow x/(x^2 + 1)$. To answer that we need to find the maximum value of $|f(x) - L_k|$.

1. From plots of $\text{abs}(f(x) - L_k)$, estimate the maximum error for $k = 2, 4, 7,$ and 12. Use `fsolve` with interval restriction to determine more precise values of the maximum error. This measure of fit is called the *supremum norm*.

2. The second way in which to answer the question is to determine the average error. Recall the definition of the average value of a function $h$ over the interval $[a, b]$ is

   $$\bar{h} = \frac{1}{b - a} \int_a^b h(x) \, dx.$$ 

   This leads us to compute

   $$\int_0^4 |f(x) - L_k| \, dx$$

   as the total error, four times the average error. Determine the average error for $k = 2, 4,$ and 7. Use the command `evalf(Int(abs(f(x) - L12), x=0..4))` to access purely numerical integration methods. This integral is used as another measure of goodness of fit. It is called the $L_1$ norm.

**Question 4** Is $L_{12}$ worth the effort of producing? Is $L_{12}$ significantly better than $L_4$ or $L_7$?
3. Is Lagrange interpolation worth the effort at all? We have already seen that Taylor polynomials can be approximate functions.

Compute the Taylor polynomials $T_k$ of degree $k$ (order $k+1$) centered at $x = 2$ for $k = 2, 4, 7,$ and $12$. Be sure to convert the Taylor expansion to polynomial form by using `convert(..., polynom)`.

**Question 5** Why is it appropriate to take the Taylor expansion of $f$ about the point $x=2$?

4. Use the supremum norm and the $L_1$ norm to compare $T_k$ and $L_k$ for $k = 2, 4, 7,$ and $12$.

**Question 6** Is Lagrange interpolation worth the effort?

**Extension**

There are two theorems answering the question, “Will Lagrange interpolation always work and work well?”

For every function there is a bad choice of knots.

and

For every family of the knots there is a function that will be badly represented by the Lagrange interpolating polynomials.

However, judicious choice of knots can lead to excellent approximations over a fixed interval.

As a small illustration of the dangers associated with an incautious choice of knots, consider the following Maple session.

Define $g(x)$ with

```maple
> g := x -> cos(2*Pi*x)
```

Set

```maple
> x_data := [0, 1, 2, 3, 4]
> y_data := map(g, x_data)
```

Now construct the interpolating polynomial

```maple
> L := interp(x_data, y_data, t)
```

and plot the results

```maple
> plot([g(x), L], x=0..5, y=-1.5..1.5)
```

Describe the goodness of fit that you observe.
Report Requirements

A minimal project report will include:

- English responses to Questions 1 through 6.
- A narrative of your investigation of Problems 1 through 4 supported with tables and graphs as appropriate.