Due date: Monday, February 03, at the beginning of the lab

1 What this Lab is About

The main objective of this lab is for you to get used (again?) to Maple or muPad, their awkward syntax and interface, and their less-than-perfect help system. I also want you to get used to importing output data from a C++/Java application back into Maple or muPad for plotting, verification, and comparison.

2 Reminder: Derivatives

In this lab, we will only consider functions $f : I \rightarrow \mathbb{R}$, with $I \subset \mathbb{R}$, that is, functions of a single real variable that take their values in the set of real numbers. I will assume that the function $f$ that I talk about is $C_\infty$ around $x_0$, that is, for any order $n$, the $n$th derivative of $f$, $f^{(n)}$ is defined and continuous at $x_0$.

2.1 Derivative of a function

First definition
We say that $f$ is derivable at $x_0$ iff the function

$$I \setminus \{x_0\} \rightarrow \mathbb{R}$$

$$x \mapsto \frac{f(x) - f(x_0)}{x - x_0}$$

has a limit as $x$ tends toward $x_0$. This limit is called the derivative of $f$ at $x_0$ and is noted $f'(x_0)$. We say that $f$ is derivable on $I$ iff $f$ is derivable at all points of $I$.

The interpretation of this definition is that the derivative of $f$ at $x_0$ is the local change rate of $f$ at $x_0$, or the slope of its graph at $(x_0, f(x_0))$.

Alternative definition
If there exists a neighborhood $V$ of $x_0$ (an open interval of $\mathbb{R}$ that contains $x_0$) and a number $l \in \mathbb{R}$ such that

$$\forall x \in V, f(x) = f(x_0) + l \cdot (x - x_0) + o(x - x_0),$$
then we will say that \( f \) is derivable\(^1\) at \( x_0 \). We call \( l \) the derivative of \( f \) at \( x_0 \) and note it \( f'(x_0) \).

In case you forgot what it represents, \( o(z) \) essentially means "something that is negligible compared to \( z \) around 0. When we write \( g(z) = o(z) \), we are saying that on some (small) open interval that contains 0 then we can write \( g(z) = z \cdot \varepsilon(z) \), were \( \varepsilon \) is a function such that \( \lim_{z \to 0} \varepsilon = 0 \).

This definition is very interesting, because it says that if \( f \) is derivable at \( x_0 \), then it can be approximated by a linear function on a small neighborhood around \( x_0 \). In other words, if we know that \( f \) is derivable at \( x_0 \), then all we need to build a reasonable approximation of \( f \) around \( x_0 \) are the values of \( f(x_0) \) and \( f'(x_0) \).

### 2.2 What to do, Part I: Check the validity of this approximation

Create a Maple worksheet and define a few functions. The syntax for defining a function in Maple is

\[
> f := x \rightarrow x \cdot \cos(x);
\]

You can plot this function on an interval (here \([-3, 3]\), by executing

\[
> \text{plot}(f(x), x=-3..3);
\]

Then you can define and plot the derivative of your function, alone or together with \( f \).

\[
> \text{df} := x \rightarrow \text{diff}(f(x), x);
\]

\[
> \text{plot}(\text{df}(x), x=-3..3);
\]

\[
> \text{plot}([f(x), \text{df}(x)], x=-3..3);
\]

Finally, you can define the linear approximation of your function around a point \( x_0 \).

\[
> \text{lf} := x \rightarrow f(x_0) + \text{eval}(\text{df}(y), y=x_0)*(x-x_0);
\]

Now you should plot \( f \) and this linear approximation around the \( x_0 \) that you chose and see how good the approximation is.

In fact, Maple has a package named \texttt{student} that does some of the work for you (in the linear case). Just try

\[
> \text{with(student)};
\]

\[
> \text{showtangent}(f(x), x_0);
\]

This package does some neat things, but not Taylor expansions, which we will do next.

---

\(^1\)In fact, this is the 1-dimensional version of the definition of the differentiability of \( f \) at \( x_0 \), which we will use when we deal with optimization. When dealing with functions \( \mathbb{R} \rightarrow \mathbb{R} \), derivability and differentiability are, for all purpose, equivalent notions.
2.3 What to do, Part II: Try this with a few functions

Check the validity of your linear approximation for the following functions and at a few well-chosen points. Zoom in (by changing the range of your plot) to get a better feel for what is going on.

\[
\begin{align*}
    f_1(x) &= \frac{x^5 - 1}{10x^2 + 1}, \\
    f_2(x) &= \cos x \frac{1 - \cos 2x}{(2 + \sin x)^2}, \\
    f_3(x) &= x \sin \left( \frac{1}{x} \right), \\
    f_4(x) &= x^3 \sin \left( \frac{1}{x} \right),
\end{align*}
\]

3 Experimenting with the Taylor Expansion

3.1 The Taylor-Young formula

Let \( f \) be \( n \) times derivable at \( a \). Then

\[
\lim_{t \to a, t \neq a} \frac{1}{(t - a)^n} \left[ f(t) - f(a) - \sum_{k=1}^{n} \frac{(t - a)^k}{k!} f^{(k)}(a) \right] = 0.
\]

Another way to write this would be that, on a small neighborhood \( V \) of \( a \), we would have

\[
\forall t \in V, f(t) = f(a) + \sum_{k=1}^{n} \frac{(t - a)^k}{k!} f^{(k)}(a) + o((t - a)^n).
\]

3.2 The Taylor-Lagrange formula

This formula is interesting because it is not a “local” formula, unlike Taylor-Young since it applies over any interval.

Let \( f \) be a function defined over \([a, b]\) and \( n \) times derivable over \((a, b)\). Then there exists \( c \in (a, b) \) such that

\[
f(b) = f(a) + \sum_{k=1}^{n-1} \frac{(t - a)^k}{k!} f^{(k)}(a) + \frac{(b - a)^n}{n!} f^{(n)}(c).
\]
3.3 The $n$-jet of a function $f$ around a point $x_0$

The $n$-jet of function $f$ at $x_0$ is defined by taking the terms of the Taylor expansion of $f$ at $x_0$ up to degree $n$ only, that is, by ignoring all terms of degree higher than $n$:

$$j^nf : \mathbb{R} \rightarrow \mathbb{R}$$

$$x \mapsto j^nf(x) = f(x_0) + \sum_{k=1}^{n} \frac{1}{k!} f^{(k)}(x_0) \cdot (x - x_0)^k.$$

So, for example, the 0-jet of $f$ at $x_0$ is simply the constant function that always returns $f(x_0)$. The 1-jet of $f$ at $x_0$ is the linear approximation of $f$ around $x_0$ that we studied in Section 1.

Intuitively, we would expect that, as we increase the order of the jet, the approximation gets better. This is correct, but again, only on a small neighborhood around $x_0$.

3.4 Work to do, Part III: Plot the jets

You should write a Maple procedure that does the following, assuming that a function $f : \mathbb{R} \rightarrow \mathbb{R}$ has been properly defined:

- Take as parameters:
  - a value for $x_0$,
  - a (small) number $h$ indicating the size of the interval $(x_0 - h, x_0 + h)$ on which the study should take place,
  - an integer $n > 0$ indicating the maximum jet order that should be considered.

- Plot $f$ over the interval $[x_0 - h, x_0 + h]$.

- for each $k, 0 \leq k \leq n$, plot together $f(x)$ and $j^nf(x)$ over the interval $[x_0 - h, x_0 + h]$. Use a different color for $f$ and the $k$-jet.

Run your procedure for each of the following functions and chosen values of $x_0$:

$$f_1(x) = \frac{x^5 - 1}{10x^2 + 1},$$

$$f_2(x) = \cos x,$$

$$f_4(x) = x^3 \sin \left( \frac{1}{x} \right),$$

Experiment with different values of $h$ and $n$ to get a better idea of what is going on.
3.5 Work to do, Part IV: What about the residual term?

If we look back at two forms of the Taylor formula, we see that the second formula (Taylor-Lagrange) tells us something more interesting about the difference between \( f \) and its \( n \)-jet. The Taylor-Young expression tells us that this difference, at \( x_0 + \varepsilon \) is \( o(\varepsilon^n) \), that is, “negligible compared to \( \varepsilon^n \).” In effect, the Taylor-Young formula tells us more or less that near \( x_0 \), \( f \) is equal to its \( n \)-jet, plus something really small. The Taylor-Lagrange is much more informative: It tells us in particular about the minimum and maximum possible value of this small term.

Now, you should be convinced by now that Taylor-Young is indeed correct, but what about Taylor-Lagrange? Can you think of a way to verify graphically that this formula is correct for the different functions \( f \) that we have experimented with in the previous section?

4 Work to do, Part V: File input in Maple or muPad

This part is simple and should not take you much time at all. All you have to do is find out how to read in Maple or muPad data files that store lists of numerical values, and then use the software list plot functions to plot curves for these lists of values.

The format of the data files is up to you to define. Don’t forget to explain your format in the report.

5 Work to do, Part VI: C++/Java Implementation

5.1 The jet class

In this part of the assignment, you will have to implement an abstract/virtual \( k \)-jet class in C++ or Java. Then, for each of the functions for which you need to implement a \( k \)-jet, you will do so by deriving this class from the abstract \( k \)-jet class. In this and future assignment, we will only consider jets up to order 5. Don’t forget that you can compute the exact expression of the derivatives of your function \( f \) with Maple so that you don’t implement incorrect expressions.

You are going to use these jet classes in future lab assignments. For example, later this semester, you will write a “nonlinear solver” class that finds the zeros of a function whose jet has been passed as an argument (pointer to a \( k \)-jet object in C++, simply a reference to that object in Java).

5.2 Generate data

Now that you have a working jet class, you should use it to generate files of data of the form \( x, j^k f(x) \). So, write a main program that creates some jet objects and uses them to evaluate the corresponding functions around points of your choosing, then outputs these results to text files. Of course, you should then read these data files in Maple or muPad and plot the results.
5.3 Analysis

I should not have to tell you that you are expected to compare the graphs obtained by plotting directly the functions and by plotting the output of your C++/Java programs.

6 What to Hand in

Commented Maple worksheet, C++ or Java CodeWarrior project, and report.