Using Mathematica to Illustrate the Race Track Principle in Calculus

by Alkiviadis G. Akritas and Zamir Bavel

An astonishing innovation in the teaching of Calculus is the use of the *race track principle*. This little-known principle is elegantly used in the Calculus and Mathematica (C&M) series of books ([1], [2], [4]) to explain and prove many concepts. Below we present two different versions of this principle and, using Mathematica, we show how it is used to explain the power series expansion of a function and the round-off errors that appear in certain computations.

Although implicitly used in most Calculus books (see for example [5] and [6]) the Race Track Principle is extensively used in the Calculus and Mathematica (C&M) book series by Davis, Porta and Uhl ([1]–[4]). This series of books is a valuable and well thought-out method for teaching Calculus. As indicated by the title of this series of books, Mathematica facilitates the exploration.

Following their lead, we use Mathematica to present two versions of the Race Track Principle and then, in the two sections that follow, we show how this principle is applied.

**First Version of the Race Track Principle**

**Horses:** If two horses start a race at the same point, then the faster horse is always ahead.

**Functions:** If \( f[a] = g[a] \) and \( f'[x] \geq g'[x] \) for \( x \geq a \) then \( f[x] \geq g[x] \) for \( x \geq a \).

This version of the Race Track Principle is good for explaining why one function plots out above another function. For example, consider the functions \( f[x] = \sin[x] + \arcsin[x] \) and \( g[x] = x \) and look at their plot for \( 0 \leq x \leq 1 \) (Figure 1):

```
Clear[f, g, x]
f[x_] = Sin[x] + ArcSin[x];
g[x_] = x;
Plot[{{f[x], g[x]}, (x, 0, 1)},
    PlotStyle -> {{RGBColor[1, 0, 0], Thickness[0.01],
                   Dashing[0.05, 0.05]}, {Thickness[0.01],
                   AspectRatio -> 1/GoldenRatio, AxesLabel ->
                   {"x", " "}}];
```

The reason the plot in Figure 1 turned out this way can be easily explained using (the first version of) the Race Track Principle. Since \( f[0] = g[0] = 0 \),

\[
\{f[0], g[0]\}
\]

the two functions start their race at \( x = 0 \) together. Now, \( f'[x] \) and \( g'[x] \) for \( x \geq 0 \) come into play:

\[
\{f'[x], g'[x]\}
\]

\[
\left\{\frac{1}{\sqrt{1-x^2}} + \cos[x], 1\right\}
\]

For \( 0 \leq x \leq 1 \) we have \( f'[x] \geq 2 > 1 = g'[x] \); that is to say, \( f[x] = \sin[x] + \arcsin[x] \) grows faster than \( g[x] = x \) for \( 0 \leq x \leq 1 \). By the Race Track Principle, \( f[x] \geq g[x] \) for \( 0 \leq x \leq 1 \) and this explains the plot.

**Second Version of the Race Track Principle**

**Horses:** If two horses are tied at one point, and they run at the same speed at this point, then they run close together near this point.

**Functions:** If \( f[a] = g[a] \) and \( f'[a] = g'[a] \), then the two functions plot out nearly the same as \( x \) advances from a little bit to the left of \( x = a \) to a little bit to the right of \( x = a \).

(The usage of the expressions “nearly,” “from a little bit to the left of \( x \)” and “to a little bit to the right of \( x \)” is based on intuition. With the help of Mathematica students can “zoom in” and get a clear estimate of the numerical values involved. We do not present the zooming process in this article.)

This version of the Race Track Principle is good for explaining why at the point \( x = a \) we have a smooth transition from \( f[x] \) to \( g[x] \) (or vice-versa) as \( x \) advances across \( x = a \). For example, consider the functions \( f[x] =

FIGURE 1: The function \( f[x] = \sin[x] + \arcsin[x] \) (dashed line) plots above the function \( g[x] = x \).
Cosb[x] + Cos[x] and g[x] = 2 and look at their values and the values of their first few derivatives at $x = 0$.

Clear[f, g, x]
f[x_] = Cosh[x] + Cos[x];
g[x_] = 2;

{f[0], g[0]}
{2, 2}

{f'[0], g'[0]}
{0, 0}

{f''[0], g''[0]}
{0, 0}

{f'''[0], g'''[0]}
{2, 0}

The first three derivatives are the same at $x = 0$. Both functions start their race at $x = 0$ together and their growth rates are the same at $x = 0$. According to (the second version of) the Race Track Principle the two functions plot out nearly the same as $x$ advances from a little bit to the left of $x = 0$ to a little bit to the right of $x = 0$. Moreover, at the point $x = 0$ we have a smooth transition from $f(x)$ to $g(x)$ (or vice-versa) as $x$ advances across $x = 0$ (Figure 2).

Plot[{f[x], g[x]}, {x, -0.5, 0.5},
PlotStyle -> {{RGBColor[1, 0, 0], Thickness[0.01],
Dashing[0.05, 0.05]}, Thickness[0.01]},
AspectRatio -> 1/GoldenRatio, AxesLabel ->
{"x", " "}];

Clear[fg];
fg[x_] := f[x]/;x > 0;
fg[x_] := g[x]/;x < 0;

first = Plot[fg[x], {x, -0.5, 0.5}, AspectRatio -> 1/
GoldenRatio, PlotStyle -> {Thickness[0.03]},
DisplayFunction -> Identity];

FIGURE 2: At the point $x = 0$ we have a smooth transition from the function $f(x) = \cosh(x) + \cos(x)$ to the function $g(x) = 2$ (or vice-versa) as $x$ advances from the left of $x = 0$ to the right of $x = 0$. (See also the next plot, Figure 3.)

The two functions are nearly identical near $x = 0$. As $x$ advances from the left of $0$ to the right of $0$, we can smoothly transfer from one curve to the other.
This phenomenon has to do with derivatives. Consider the two functions at $x = 0$ and their first four derivatives there:

\[
\{f[0], g[0]\} \\
\{f'[0], g'[0]\} \\
\{f''[0], g''[0]\} \\
\{f'''[0], g'''[0]\}
\]

Both functions go through $[0, 1]$ and they have the same first, second, and third derivatives at $x = 0$. They differ only when we get to the fourth derivatives.

We are now ready for the following definition: We say that $f[x]$ and $g[x]$ have order of contact $m$ at $x = a$ if:

\[
f[a] = g[a], \\
f'[a] = g'[a], \\
f''[a] = g''[a], \\
\ldots, \\
f^{m-1}[a] = g^{m-1}[a], \\
f^m[a] = g^m[a],
\]

so that the functions and their first $m$ derivatives agree at $x = a$.

Now, according to the second version of the Race Track Principle, if $f[a] = g[a]$ and $f'[a] = g'[a]$, then the two functions plot out nearly the same as $x$ advances from a little bit to the left of $x = a$ to a little bit to the right of $x = a$. This explains why, when we have order of contact 1 at $x = a$, then we have a smooth transition as $x$ advances from the left of $x = a$ to the right of $x = a$. (See also the example in the second version of the Race Track Principle in the Introduction.)

To see why it is that, when we have order of contact 2 at $x = a$, then we can expect an even smoother transition as $x$ advances from the left of $x = a$ to the right of $x = a$, recall that order of contact 2 at $x = a$ means

\[
f[a] = g[a], \\
f'[a] = g'[a], \\
f''[a] = g''[a],
\]

The fact that $f''[a] = g''[a]$ and $f''[a] = g''[a]$ ensures that $f''[x]$ is very close to $g''[x]$ as $x$ advances across $x = a$. This, in turn, means that $f'[x]$ is very close to $g'[x]$ as $x$ advances from the left of $x = a$ to the right of $x = a$. (It is even more so for order of contact 3 at $x = a$ as was the case for the above example.)

It is now a simple matter to compute the power series expansion of a function. Given a function $f[x]$, the expansion of $f[x]$ in powers of $x$ is the expression

\[
ad[0] + a[1] x + a[2] x^2 + a[3] x^3 + \ldots + a[k] x^k + a[k+1] x^{k+1} + \ldots
\]

where the numbers $a[0], a[1], a[2], a[3], \ldots, a[k], a[k+1], \ldots$ are chosen so that for every positive integer $m$, the function $f[x]$ and the polynomial

\[
ad[0] + a[1] x + a[2] x^2 + a[3] x^3 + \ldots + a[m-1] x^{m-1} + a[m] x^m
\]

have order of contact $m$ at $x = 0$.

Let us expand the function $f[x] = \cos[x]$ in powers of $x$ up to degree 4. Notice that Mathematica has the special function Series for obtaining such expansions up to any degree. Below we expand Cos[x] up to degree 6:

\[
\text{Series}[\cos[x], \{x, 0, 6\}]//\text{Normal}
\]

and note that $24 = 4!$ and $720 = 6!$.

To get an idea how this algorithm works, we start by entering $f[x]$ and the fourth degree polynomial of the form $a[0] + a[1] x + a[2] x^2 + a[3] x^3 + a[4] x^4$:

\[
f[x_] = \cos[x];
\]

\[
g[x_] = \sum_{k=0}^{4} a[k] x^k
\]

\[
\]

According to the preceding discussion we need the following equations:

\[
eq\text{eq1} = f[0] == g[0]
\]

\[
eq\text{eq2} = f'[0] == g'[0]
\]

\[
eq\text{eq3} = f''[0] == g''[0]
\]

\[
eq\text{eq4} = f'''[0] == g'''[0]
\]

\[
eq\text{eq5} = f''''[0] == g''''[0]
\]

and note that $24 = 4!$ and $720 = 6!$.

Now we have five equations with five unknowns $a[0], a[1], a[2], a[3], a[4]$. We next solve the equations, to obtain:

\[
\text{coefficients} = \text{Solve}\{\text{eq1, eq2, eq3, eq4, eq5}\}
\]

\[
\{\{a[0] \to 1, a[1] \to 0, a[2] \to -\frac{1}{2}, a[3] \to 0, a[4] \to \frac{1}{24}\}\}
\]

The fourth degree polynomial we seek is then

\[
\text{Clear}[\text{polynomial}];
\]

\[
\text{polynomial}[x_] = g[x]/.\text{coefficients}[\{1\}]
\]

\[
1 - \frac{x^2}{2} + \frac{x^4}{24}
\]
Indeed, it checks out that \( f[x] \) and \( \text{polynomial}[x] \) have order of contact 4:

\[
\text{polynomial}[x] = 1 - \frac{x^2}{2} + \frac{x^4}{24};
\]

In Figure 5 below we see the plots of \( f[x] \) and \text{polynomial}[x] \] in the interval \(-2 \leq x \leq 2\):

\[
\begin{align*}
\text{Clear}[f, \text{polynomial}, x] \\
f[x] &= \cos[x]; \\
\text{polynomial}[x] &= 1 - \frac{x^2}{2} + \frac{x^4}{24}; \\
\text{Plot}[\{f[x], \text{polynomial}[x]\}, \{x, -2.5, 2.5\}, \\
\text{PlotStyle} -> \{(\text{RGBColor}[1, 0, 0], \text{Thickness}[0.01], \\
\text{Dashing}[(0.05, 0.05)]), \text{Thickness}[0.01]\}, \\
\text{AspectRatio} -> 1/\text{GoldenRatio}, \text{AxesLabel} -> \\
\{"x," "\"\}
\end{align*}
\]

![Figure 5: The functions \( f[x] = \cos[x] \) (dashed line) and \( \text{polynomial}[x] \) have strikingly similar plots for \(-2 \leq x \leq 2\).](image)

(Compared to Figure 4, the interval is wider now.)

We close this section with the following observations:

The first three terms of the \( \text{polynomial}[x] = 1 - \frac{x^2}{2} + \frac{x^4}{24} \) we computed matches the first three (lower) terms of the expansion of \( \cos[x] = 1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{720} \) we computed earlier. In general, working as above we can compute an arbitrary number of terms.

As we use more and more terms from the expansion in powers of \( x \), we increase the order of contact between \( f[x] \) and the corresponding polynomial at \( x = 0 \).

As we increase the order of contact at \( x = 0 \), we increase the quality of the transition from one curve to the other at \( x = 0 \).

### THE RACE TRACK PRINCIPLE AND ESTIMATION OF ROUNDOFF ERRORS

Roundoff errors and calculators do not mix very well. For example, note what happens when we feed 8 decimals of 10 e into a calculator and then compute \( (10\ e)^2 \) on that basis:

\[
\begin{align*}
\text{N}[10 \text{ E,10}] &
\end{align*}
\]

Compute the square of the above number and compare the result with \( e^2 \) to 7 decimals. (Computing \( e^2 \) and then rounding is a more accurate procedure.)

\[
\begin{align*}
\text{N}[\{27.18281828 \} - 2.10], \text{N}[(10 \text{ E} - 2.10)] \\
\{738.906, 738.906\}
\end{align*}
\]

Only the first 6 decimals match.

This behavior can be explained with (the first version of) the Race Track Principle. More precisely, we can use (the first version of) the Race Track Principle to predict the accuracy of \( a \) needed to maintain a desired accuracy of \( f[a] \), where \( f[x] \) can be any function.

We use the squaring function \( f[x] = x^2 \) and plot its derivative on the interval \( a - 1 \leq x \leq a + 1 \), for \( a = 10\ e \) in Figure 6:

\[
\begin{align*}
\text{Clear}[x, f] \\
f[x_] &= x^2; \\
a &= 10\ e; \\
growthplot = \\
\text{Plot}[[f'[x], \{x, a - 1, a + 1\}, \text{PlotRange} -> \text{All}, \\
\text{PlotStyle} -> \{(\text{RGBColor}[1, 0, 0], \text{Thickness}[0.01]), \\
\text{AspectRatio} -> 1/\text{GoldenRatio}, \text{AxesLabel} -> \\
\{"x," "\"\}
\end{align*}
\]

![Figure 6: A plot of the derivative of \( f[x] = x^2 \) on the interval \( 10\ e - 1 \leq x \leq 10\ e + 1 \).](image)

From the plot it is clear that \( |f'[x]| < 100 = 10^2 \) for all the values of \( x \) with \( 10\ e - 1 \leq x \leq 10\ e + 1 \). In Figure 7 we plot, on the same interval, \( f[x] \) and the lines through the point \( \{10\ e, f[10\ e]\} \) with slopes \( 10^2 \) and \(-10^2\):

\[
\begin{align*}
\text{Clear}[\text{bowtie}, \text{lines}, \text{downline}, \text{pointplot}, \text{upline}]; \\
\text{upline}[x_] &= 10^2 (x - a) + f[a]; \\
\text{downline}[x_] &= -10^2 (x - a) + f[a]; \\
\text{lines} = \\
\text{Plot}[[f[x], \text{upline}[x]], \text{downline}[x]], \\
\{x, a - 1, a + 1\},
\end{align*}
\]

![Figure 7: The functions \( f[x] \) and \( \text{upline}[x] \) and \( \text{downline}[x] \) on the interval \( 10\ e - 1 \leq x \leq 10\ e + 1 \).](image)
PlotStyle ->
{{RGBColor[1, 0, 0], Dashing[{0.05, 0.02}]},
Thickness[0.01], RGBColor[1, 0, 0],
RGBColor[1, 0, 0],
DisplayFunction -> Identity};
pointplot = Graphics[{{RGBColor[1, 0, 1],
PointSize[0.03],
Point[{(a, f[a])}]}];

FIGURE 7: The derivative of \( f(x) \) is the length of the little vertical line segment on the right.

Note that the discrepancy between \( f(x) \) and \( f(10e) \) in Figure 9 is smaller than \( \frac{1}{2} \) the length of the new vertical segment plotted:

\[
\text{bigdiscrepancy} = \frac{|upline[a] - lowline[a]|}{2} = \frac{|(10^2(x - a) + f(a)) - (-10^2(x - a) + f(a))|}{2} = \frac{2 \times 10^4|x - a|}{2} = 10^4|x - a|\]

Thus, for \( a = 10e \), the discrepancy between \( f(x) \) and \( f[a] \) is smaller than \( 10^2 |x - a| \).

Now we use this information to determine the number of decimals of \( 10e \) we need in order to guarantee accuracy to 4 decimals of \( f(10e) \). The discrepancy between \( f(x) \) and \( f[a] \) is no more than \( 10^4|x - a| \). Therefore, if \( x \) is accurate to 6 decimals, that is \( |x - a| \leq 10^{-6} \), then the discrepancy between \( f(x) \) and \( f[a] \) is less than \( 10^2 \times 10^{-6} = 10^{-4} \).

Consequently, if \( x \) approximates \( a \) to 6 decimals, then \( f(x) \) approximates \( f[a] \) to 4 decimals.

In the actual computation:

\[
N[a, 8]
\]
27.1828

27.1828 is \( a = 10e \) to 6 decimals; compare:

\[
\{N[f[a], 8], N[f[27.182818], 8]\}
\{738.906, 738.906\}
\]

If we round these to 4 decimals, the results are the same, just as predicted.

In the same manner, the (first version of) the Race Track Principle may be used to determine the accuracy needed for a desired accuracy in computing any function evaluated at any point. (However, care should be taken not to introduce other rounding errors when calculating by machine.)
CONCLUSION
We have presented only two versions and examples of the use of the Race Track Principle. Additional versions and examples may be found in [1]–[4].

REFERENCES

ABOUT THE AUTHORS
Akritas is on leave at the University of Thessaly, Department of Theoretical and Applied Sciences, GR-38221 Volos, Greece
University of Kansas / EECS-IPS
415 Snow Hall
Lawrence, Kansas 66045, USA
As the title of the present document, ProblemText in Advanced Calculus, is intended to suggest, it is as much an extended problem set as a textbook. The proofs of most of the major results are either exercises or problems. The distinction here is that solutions to exercises are written out in a separate chapter in the ProblemText while solutions to problems are not given. I hope that this arrangement will provide flexibility for instructors who wish to use it as a text. For those who prefer a (modified) Moore-style development, where students work out and present most of the material, there is a question astonishing innovation in the teaching of calculus is the use of the race track principle. This little-known principle is elegantly used in the Calculus and Mathematica (C&M) series of books ([1], [2], [4]) to explain and prove many concepts. Below we present two different versions of this principle and, using Mathematica, we show how it is used to explain the power series expansion of a function and the round-off errors that appear in certain computations. Below we present two different versions of this principle and, using Mathematica, we show how it is used to explain the power series expansion of a function and the round-off errors that appear in certain computations. Complex Analysis with Mathematica offers a new way of learning and teaching a subject that lies at the heart of many areas of pure and applied mathematics, physics, engineering and even art. This book offers teachers and students an opportunity to learn about complex numbers in a state-of-the-art computational environment. The innovative approach also offers insights into many areas too often neglected in a student treatment, including complex chaos and mathematical art. Thus readers can also use the book for self-study and for enrichment. The use of Mathematica enables the author to cover sev