



MATHEMATICA IN THE CLASSROOM: NEW TOOLS FOR EXPLORING PRECALCULUS AND DIFFERENTIAL CALCULUS

Ana C. Conceição

*Departamento de Matemática, Faculdade de Ciências e Tecnologia
Universidade do Algarve – Campus de Gambelas
8000-810 Faro, Portugal*

José C. Pereira

*Departamento de Engenharia Electrónica e Informática, Faculdade de Ciências e Tecnologia
Universidade do Algarve – Campus de Gambelas
8000-810 Faro, Portugal*

Cristina R. Simão

*Departamento de Matemática, Faculdade de Ciências e Tecnologia
Universidade do Algarve – Campus de Gambelas
8000-810 Faro, Portugal*

Cátia M. Silva

*Departamento de Matemática, Faculdade de Ciências e Tecnologia
Universidade do Algarve – Campus de Gambelas
8000-810 Faro, Portugal*

Abstract

The main goal of this paper is to present some interactive tools, F-Tools, designed by us and implemented with the computer algebra system *Mathematica*, which we hope will improve the teaching-and-learning process by providing teachers and students alike with new ways to explore some of the main mathematical subjects, at the secondary and university levels, specifically in the areas of precalculus and differential calculus. We believe that these new tools are an important contribution to Mathematical Education, providing new ways for teaching and learning. We intend to make available several F-Tools, such as F-Linear, F-Quadratic, F-Exponential, F-Logarithm, and F-Trigonometric, at the Wolfram Demonstrations Project site.

1. Introduction

It is not possible to achieve the objectives and skills of a modern mathematics course at the secondary school without resorting to graphic concepts. These concepts can be more easily apprehended when the students work with a large number and variety of graphics, in an interactive way, with the support of the appropriate technology. Obviously, calculations with the support of technology are not a replacement for paper and pencil calculations, and they should be properly combined with other methods of calculation, including mental calculation. Students should be prepared for an intelligent dialogue with the tools they have available.

The main goal of this paper is to present some interactive tools, designed by us with the computer algebra system *Mathematica*¹, which will improve the teaching-and-learning process precisely by providing teachers and students alike with new ways to explore some of the main mathematical subjects, at the secondary and university levels, specifically in the areas of precalculus and differential calculus.

The use of the *Mathematica* software system in the classroom (in Precalculus and Calculus) and in research, by some of the authors of this paper (see (Conceição, Kravchenko, Pereira, 2010), (Conceição, Kravchenko, Pereira, 2011), and (Conceição, Kravchenko, Pereira, 2012)) motivated the construction of new tools for exploring the concept of real functions and their principal properties, such as the domain, range, existence of zeros, existence of asymptotes, existence of critical points, existence of maximums and minimums, existence of inflection points, invertibility, symmetries, and the first and second derivatives.

We created the concept of F-Tool, an interactive Mathematica notebook, specifically designed to explore the concept of real functions and their graphics, by analyzing the effect caused by changing parameters present in the corresponding analytical expression. Going forward, we intend to make available online to the general public, via the Wolfram Demonstrations Project² several F-Tools, such as F-Linear, F-Quadratic, F-Exponential, F-Logarithm, and F-Trigonometric (with the functions *sine*, *cosine*, and *tangent*). The corresponding source codes will also be available at the Wolfram Demonstrations website.

We believe that these new tools are an important contribution to Mathematics Education, providing new ways for teaching and learning Mathematics. They also open the possibility for the development of other interactive tools, based on the symbolic computation capabilities of modern computer algebra systems such as *Mathematica*, for use at the elementary, middle, secondary, and university levels.

* This research was partly supported by CEAF at Instituto Superior Técnico (Portugal), under the project PEest-OE/MAT/UI4032/2011.

Email addresses: aicdoisg@gmail.com (Ana C. Conceição), unidadeimaginaria@gmail.com (José C. Pereira), cristinasilva@sapo.pt (Cristina R. Simão), catiamasilva@hotmail.com (Cátia M. Silva).

¹ Available at <http://www.wolfram.com>

² In this project (<http://demonstrations.wolfram.com>) the creators of *Mathematica* promote and divulge globally the innovations designed by its users.

The rest of this paper is organized as follows.

In Section 2 we discuss briefly the modern concept of computer algebra system and explain the reasons behind our choice to use *Mathematica* for the construction of our tools.

In Section 3 we present the concept of the F-Tool. We discuss all the functionalities available within each F-Tool, and how the dynamism and interactivity of these tools play a major role in the illustration of many precalculus and calculus concepts.

In Section 4, we discuss the designing of the F-Tools and present the main characteristics specific to each of the F-Tools. We also explain some of the major symbolic computation aspects of the code itself.

In Section 5, we make some remarks on how we intend, going forward, to extend the concept of F-Tool to other school levels and to other areas of Science. We also discuss how we plan to try and generalize the use of these tools in the classroom.

2. The Computer Algebra System *Mathematica*

In recent years, several software applications with extensive capabilities of symbolic and numeric computation were made available to the general public. These applications receive the generic designation of Computer Algebra Systems (CAS).

Symbolic manipulations supported by CAS typically include, but are not restricted to, the simplification of mathematical expressions in reference to a given canonical format, the analytical resolution of several types of equations and inequalities, symbolic differentiation and integration, or operations with vectors and matrices.

Also, in alignment with other major evolutions of Information Technology, many CAS include additional functionalities such as the use of a numeric system with infinite precision or the use of their own programming language to enhance the design of perfectly integrated applications.

All these capabilities make CAS a tool with increasing usage in many areas of Science and Technology. Today there is a real and significant effort to try and delegate to digital computers many of the decision making processes that mathematicians and other researchers usually perform with the traditional tools pencil and paper.

Wolfram's *Mathematica* is a powerful CAS used in scientific, engineering, and mathematical fields and in other areas of technical computing. In what concerns the work presented in this paper, *Mathematica* graphics are completely integrated into its dynamic interactive language. Any visualization can immediately be animated or made interactive using a single command and developed into sophisticated, dynamic visual applications. Creating interactive visual models with *Mathematica* allows students to explore hard-to-understand concepts, test theories, and quickly gain a deeper understanding of the materials being taught firsthand. The students can explore changes to text, functions, formulas, matrices, graphics, tables, or even data.

In our work, the concept of F-Tool arose naturally from the dynamic, interactive, and visual capabilities of *Mathematica*. The F-Tool provides rigorous analytical information on various classes of functions and concepts associated with their study, allowing simultaneously students to visualize several features, change values and graphics, and see results in real time.

As the Wolfram Education Group wittingly states: "Calculators are for calculating, *Mathematica* is for Calculus".

3. The F-Tool concept

The F-Tool is a new concept that allows to explore in a dynamic, interactive, and visual manner various classes of real functions and features associated with their study.

In Figure 1 we can see a usage example of the F-Exponential, one of the already existing F-Tools, designed to explore a class of functions of the exponential type. Since all the F-Tools share a common design, we will use this example to present the main features of the F-Tool concept.

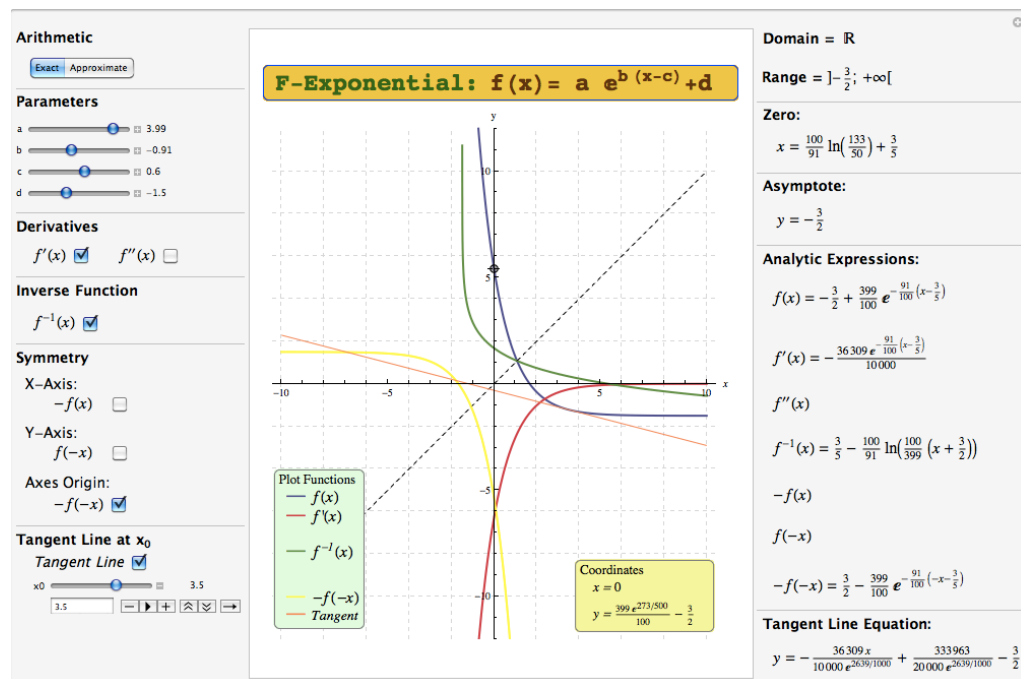



Fig. 1. Usage example of the F-Exponential tool.

A F-Tool is divided in three main panels:

- Left Panel:* In this panel the user can vary the values of one or more parameters, choose which “transformations” of the main function are to be presented, and whether or not to show the tangent line in a chosen tangent point. In addition, the user has the option to see the results in the exact or approximate forms.
- Middle Panel:* In this panel all the functions are plotted, according to the options made by the user in the left panel. For the main function it is also possible to see, in an interactive way, the coordinates of the corresponding graphical points (click-and-drag the \oplus locator placed over the function line).
- Right Panel:* In this panel it is presented all the analytical information concerning the main function and its “transformations”, again in accordance with the options chosen by the user in the left panel.

Precalculus and differential calculus, are concerned with the study of functions and their main properties. The functionalities of each F-Tool, along with the displayed information, are all designed to conform with the way those subjects are presented in the classroom, at the secondary and university levels. One of the main contributions for this design option was the sharing of ideas with two of the co-authors of this paper, who are currently master students in Mathematics Education and are actively enrolled in preservice teaching.

As mentioned in Section 2, all the graphical and analytical information presented in the middle and right panels is displayed in real time, thus rendering the F-Tools an eminently dynamic tool. In particular, the *Mathematica* controls allow the automatic continuous change of the parameter values (the  button depicted, for instance, in the tangent line option in Figure 1). When choosing this option, the user will then see the corresponding graphics move continuously and the analytical information change in accordance.

We believe that this kind of dynamic interaction, by providing teachers and students alike with new ways to explore some of the main subjects in the areas of precalculus and differential calculus, renders the F-Tool concept an important contribution to Mathematics Education, improving the teaching-and-learning process.

4. Designing the F-Tool

In this section we discuss how the symbolic computation capabilities of *Mathematica*, and its own programming language, along with the *pretty-print*³ functionality, allowed us to implement on a computer, and in a rather straightforward manner, all the ideas that go into the F-Tool concept.

Each and every F-Tool has a common general code structure, depicted in Figure 2, which is actually very simple.

```

Initial Definitions →[ ...]

Manipulate[
  DynamicModule[
    [...],
    MiddlePanel → Plot[f[x, a, b, c, d],
      [...],
      Plot Options
    ],
    LeftPanel → Option Controls,
    RightPanel → Display Analytical Information
  ](* End DynamicModule *)
](* End Manipulate *)

```

Fig. 2. General code structure of the F-Tool

³ *Pretty-print* is a functionality that allows to write mathematical expressions on the computer using the traditional notation, as if on paper.

The code consists of some initial definitions that pertain to each particular class of real function, followed by a single command **Manipulate**. This command is responsible for creating the interactive object that contains the three panels described in Section 3. In particular, **Manipulate** generates all the functional controls, such as *sliders* for the parameter values and *checkboxes* for the plot options. Also, integrating this command with a `DynamicModule[...]` is what allow us to create, for instance, the *locator* functionality, available in the middle panel, by enabling the dynamic update of selected variables.

In summary, each F-Tool is essentially created by a single **Manipulate** command, whose output is not just a static result but a running program that we can interact with.

4.1. F-Linear

The class of linear functions is the simplest family of functions to become acquainted with. Accordingly, F-Linear is, code wise, the simplest of the existing F-tools.

Linear functions model constant rates of change of a given quantity and can be written in the general form $f(x) = ax + b$, $a, b \in \mathbb{R}$.

This class of functions includes the constant functions, for when $a = 0$ (see Figure 3). In spite of their simplicity, constant functions have to be dealt separately because they have no inverse function, and their range and set of zeros are completely different from other linear functions.

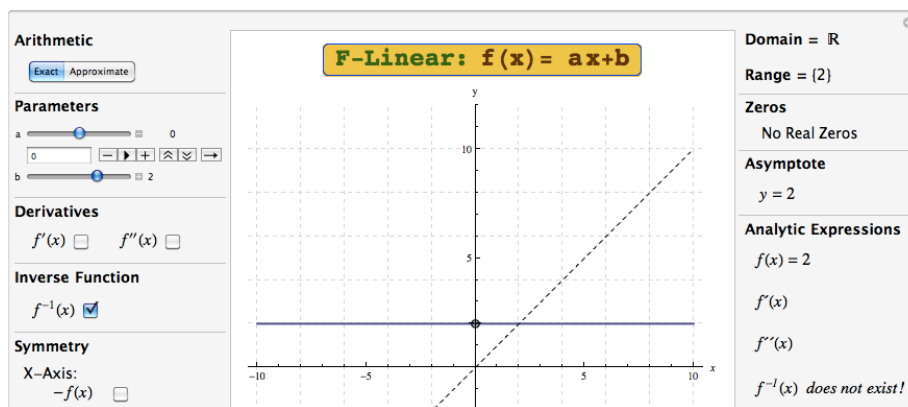


Fig. 3. The case $a = 0$ for the F-Linear tool.

We note that, constant functions are a common element to all the classes of functions discussed in this paper. This means that, in all F-Tools, the constant case had to be coded separately, in order to generate the correct analytical information for those functions.

4.2. F-Quadratic

Quadratic functions can be written in the standard form $f(x) = ax^2 + bx + c$, $a, b, c \in \mathbb{R}$. However, when studying quadratic functions there are many advantages in using the derived vertex form $f(x) = a(x - h)^2 + k$.

F-Quadratic uses the latter form mainly because the parameters a , h , and k have a more direct relation with the graphical transformations that we intend to explore with this tool. Nonetheless, because of its general importance the quadratic standard form is

always displayed in the right panel of F-Quadratic , as depicted in Figure 4. We note that the vertex form does not include the class of polynomials of degree one. In fact, this case can be more properly explored with the F-Linear tool.

Quadratic functions have a multivalent inverse function, which is not a topic discussed in most courses at the secondary and university levels. Therefore, the F-Quadratic tool does not make available the inverse function option.

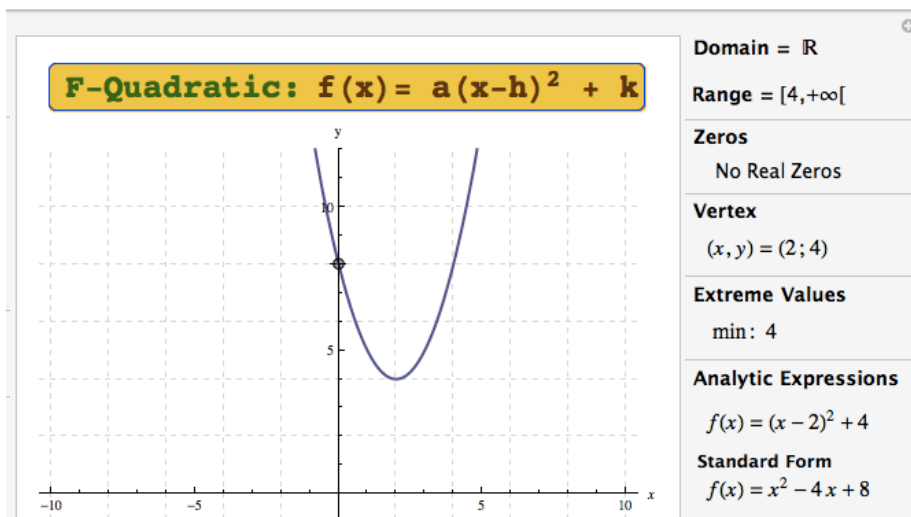


Fig. 4. Usage example of the F-Quadratic tool.

In Figure 5 it is depicted the block of code that generates the analytical information about the vertex and extreme values which are characteristic of any quadratic function. Note the need to deal separately with the constant case ($a = 0$), as already mentioned in Subsection 4.1.

```

[...]
```

```

Style[" Vertex", Bold], "",
  Dynamic[If[a != 0, TradForm[Row[{"", x, "", "y, If[rationalize, "] = ("", ") = ("",
    If[rationalize, Rationalize[h], h], " ; ",
      If[rationalize, Rationalize[k], k], ""]}], 16], (* End TradForm *)
    (* Else *) Style[" No Vertex", 14]
  ]],
[...]
```

```

Style[" Extreme Values", Bold], "",
  Dynamic[TradForm[If[a == 0, Row[{Style[" max, min : ", 16], If[rationalize, Rationalize[k], k]}],
    (* else *) If[a > 0, Row[{Style[" min : ", 16], If[rationalize, Rationalize[k], k]}],
      (* else *) Row[{Style[" max : ", 16], If[rationalize, Rationalize[k], k]}]
  ], 16]]

```

Fig. 5. Code snippet of the F-Quadratic tool. This block of code generates the analytical information about the vertex and extreme values.

4.3. F-Exponential

The F-Exponential tool explores the class of functions of exponential type, that can be written in the general form $f(x) = a e^{b(x-c)} + d$, $a, b, c, d \in \mathbb{R}$.

One good usage example for this tool is when we choose to plot the inverse function, as depicted in Figure 6. In this case, when using exact arithmetic, the exact analytical expressions of the function and its inverse are presented. The user can then easily verify, for instance, the composition law $f^{-1}(f(x)) = f(f^{-1}(x)) = x$. The dashed line seen on the plot of Figure 6 has the equation $y = x$ and corresponds to the symmetry axis of the inverse transformation.

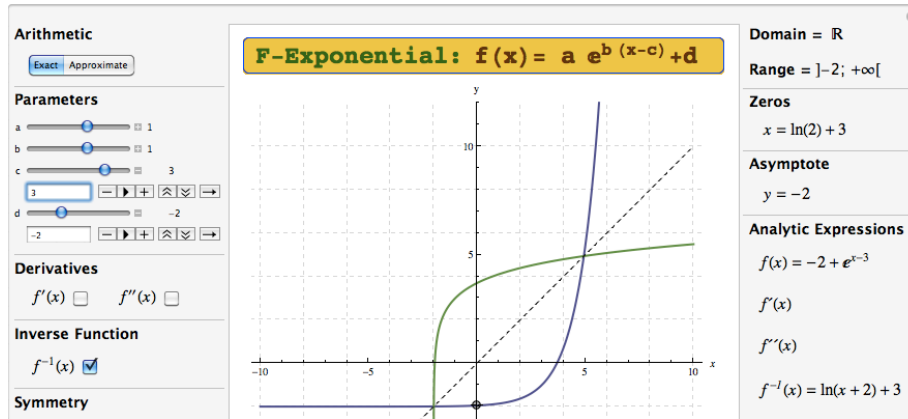


Fig. 6. Usage example of the F-Exponential tool. Note the computation of the inverse function in the exact form.

In Figure 7 it is depicted the block of code that generates the analytical information about the zeros of the exponential type function. Note the several conditionals, necessary due to all the degrees of freedom that the general expression possesses.

```

F-Exponential.nb
[...]
```

$$\text{Style[" Zeros", Bold], "",$$

$$\text{Dynamic[If[b \neq 0 \&\& a \cdot d < 0, TradForm[Row[{" x", If[rationalize, " = ", " = "],$$

$$\text{If[rationalize, FullSimplify[RinvF[0, a, b, c, d]] /. lnFormat,$$

$$\text{(* Else *) FullSimplify[N[invF[0, a, b, c, d]]}], 16],$$

$$\text{(* Else *) If[(b = 0 \&\& a = -d) || (a = 0 \&\& d = 0), TradForm[Row[{" x \in \mathbb{R} "}], 16],$$

$$\text{(* Else *) Style[" No Real Zeros", 14]]]$$

```

[...]
```

Fig. 7. Code snippet of the F-Exponential tool. This block of code generates the analytical information about the zeros of the exponential type function.

4.4. F-Logarithm

The F-Logarithm tool explores the class of functions of logarithmic type, that can be written in the general form $f(x) = a \ln(b(x-c)) + d$, $a, b, c, d \in \mathbb{R}$

In Figure 8 it is depicted the block of code that generates the analytical information about the domain of the logarithmic type function. Contrary to the other classes of functions already discussed, this logarithmic class does not possess the trivial domain \mathbb{R} .

```

[...]
Dynamic[Style[Row[If[b ≠ 0, If[a = 0, {Style[" Domain = R", Bold]},
(* Else *) If[b < 0, {Style[" Domain = ", Bold], Style["-∞", 16],
If[rationalize, Rationalize[c], c], Style["", 16]},
(* Else *) If[b > 0, {Style[" Domain = ", Bold], Style["", 16],
If[rationalize, Rationalize[c], c], Style["; +∞", 16]}]],
[(* Else *) {Style[" Domain", Bold]}]], 14]
[...]
```

Fig. 8. Code snippet of the F-Logarithm tool. This block of code generates the analytical information about the domain of the logarithmic type function.

Note that the domain is only computed when $b \neq 0$. In fact, for $b = 0$ the corresponding function is not well defined and, in this case, the user will receive the error message depicted in Figure 9. Simultaneously the user will hear the sentence “Sorry. I cannot do that!”, which is generated with a simple command `Speak`. As a consequence, all options will be unavailable until any other value is chosen for the parameter b .

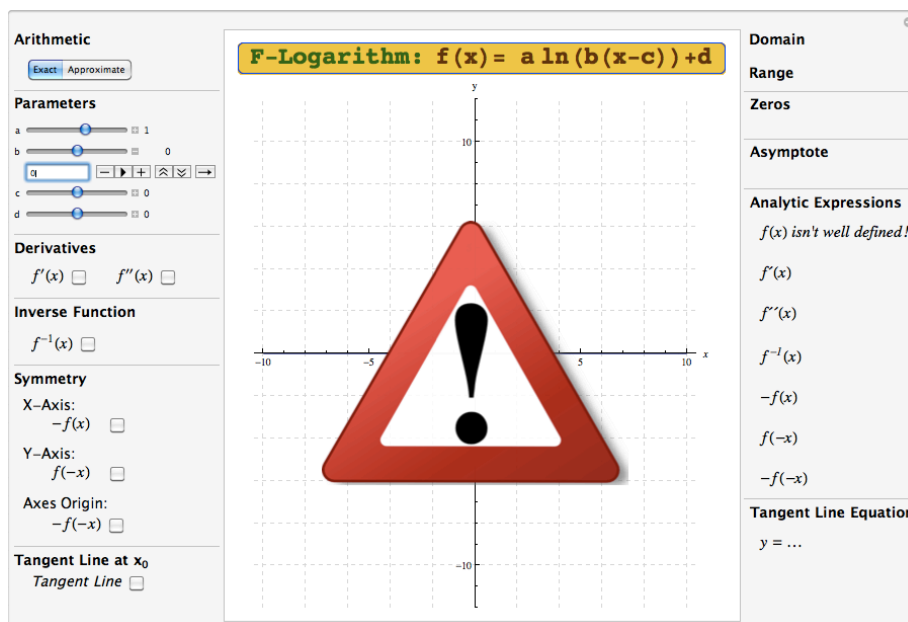


Fig. 9. Error message for the case $b = 0$ in the F-Logarithm.

4.5. F-Trigonometric

The F-Trigonometric tool explores the class of functions of trigonometric type, that includes the functions *Sine*, *Cosine*, and *Tangent*. Because of the similarity between the

F-Tools that explore those three kinds of functions, we will only discuss the F-Sine tool in this section.

The F-Sine tool explores the class of functions that can be written in the general form $f(x) = a \sin(b(x - c)) + d$, $a, b, c, d \in \mathbb{R}$.

In Figure 10 it is depicted a usage example for the F-Sine tool where both the function and its first derivative are plotted. Also worth noting is the analytic information pertaining the inflection points, which is not relevant for any of the previously discussed F-Tools. For the same reasons considered in the F-Quadratic tool, the inverse function option is not present in F-Sine.

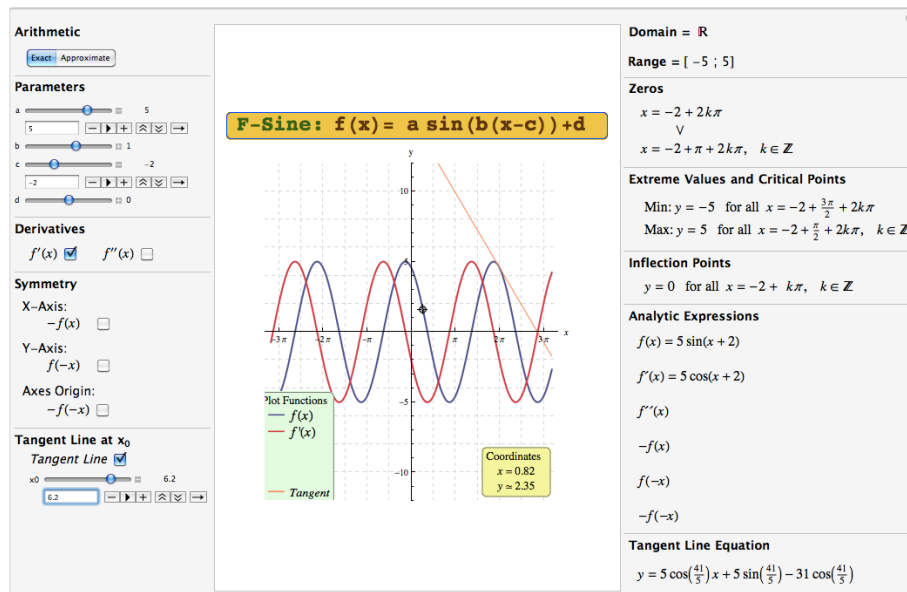


Fig. 10. Usage example for the F-Sine tool.

5. Final remarks

We believe that the F-Tool concept is an important contribution to Mathematics Education, providing new ways for teaching and learning Mathematics. It also opens the possibility for the development of other interactive tools, based on the symbolic computation capabilities of modern computer algebra systems such as *Mathematica*, for use at the elementary, middle, secondary, and university levels.

Going forward, we would like to see the usage of the F-Tools in the classroom become increasingly more common. One important step to achieve this goal would be the realization of a study *in loco*, statistically rigorous, to estimate the real pedagogical value of the F-Tools.

Acknowledgements

The authors thank Ricardo Pereira (Agrupamento de Escolas José Belchior Viegas, Portugal) for his early contributions to the genesis of this project.

References

- Conceição, A. C., Kravchenko, V. G., and Pereira, J. C., 2010. Factorization Algorithm for Some Special Non rational Matrix Functions. *Oper. Theory Adv. Appl.* 202, 87-109. Birkhäuser Verlag
- Conceição, A. C., Kravchenko, V. G., and Pereira, J. C., 2011. Computing some classes of singular integrals with Mathematica software. Submitted in *Adv. Comput. Math.*
- Conceição, A. C., Kravchenko, V. G., and Pereira, J. C., 2012. Rational Functions Factorization Algorithm: a symbolic computation for the scalar and matricial cases. Submitted in *J. Symbolic Comput.*