Parabola Volume 47, Issue 1 (2011)

The Meanings of Equality

John W Perram [1]

1 Abstract

Using a computer algebra system (CAS) such as *Mathematica* to relate a mathematical narrative requires a more disciplined use of symbols and terms than is common in mathematical text. We distinguish 3 examples in the usage of equality, the process of naming a mathematical expression, the process of assigning a value to a variable in an expression and the conditional assignment of equality between the two sides of an equation in order to solve it. We show that *Mathematica* has 3 different constructions for describing these relationships and explain why we believe that fully integrating a CAS into mathematics education for scientists and engineers should provide a better understanding of these concepts than the traditional narrative in terms of mathematical text.

2 Preface

The first of five video lectures based on the material in this notebook can be found at the URL `http://www.youtube.com/watch?v=Yw8F0mkSJhE` at my YouTube channel "hypermath".

3 Introduction

This article is based on my 15 years experience in attempting to enhance mathematical learning using computer algebra systems, and, more recently, my experience here as a sessional first year tutor. A significant proportion of the test scripts handed in consist of a collection of mathematical statements of the form

$$\text{left-hand-side expresssion} = \text{right-hand-side expression}$$

without any comments as to what is meant, why it is useful or how it contributes to the solution to the problem. The process is complicated because the failure to carry out simple transformations correctly leads to statements of the form

$$\text{left-hand-side expresssion} \neq \text{right-hand-side expression}$$

[1]Dr John Perram is a Visiting Fellow in the School of Mathematics and Statistics at the University of New South Wales.

usually making further progress impossible.

Part of this basic lack of understanding of the processes involved in solving a mathematical problem is that such statements can occur in three contexts. The first context is illustrated as mathematical text by stating something like: Let the function $f[x]$ be defined by

$$f[x] = x^2 + bx + 1 \tag{3.1}$$

which actually establishes an identity between the two patterns of symbols on either side of the equality sign, so that, for this problem, the symbol $f[x]$ is actually the name of the expression on the right.

The second context occurs when we wish to visualise $f[x]$ for some numerical values of the parameter b, for which the context is something like: For the purpose of plotting $f[x]$, it is necessary to assign a numerical value to the parameter, so we set $b = 1$. In this context, we are certainly not associating the name b with the integer 1, but instead providing an instantiation of the function $f[x]$.

The third context involves the solution of equations, an example of which is

$$x^2 + bx + 1 = 0 \tag{3.2}$$

The meaning of the equality sign in this context is "if the expression on the left is equal to the value on the right, find the value(s) of *x* for which this is so". As we can see, normal mathematical text uses the same symbol "=" for all three contexts. As we shall see, the computer algebra system *Mathematica* uses different syntaxes for the three distinct contexts.

In the remainder of this article, I begin with a discussion of the relation between using a CAS for solving mathematical problems and the process of programming a computer. I then present a discussion of the algebraic geometry of ellipses to illustrate how *Mathematica* deals with the three contexts of equality. Finally, I present a solution to the problem of drawing tangents to an ellipse, showing how mathematical tools in the form of code fragments developed in solving one problem can be reused to solve another.

4 Mathematics and software engineering

Despite all its impressive functionality, a computer algebra system such as *Mathematica* is basically a programming language, so that users should get more out of using one by adherring to some of the basic principles of software design, such as modularity, reuse and interoperability. In my 15 years of experience of investigating how to best use a CAS in mathematics education, I have attempted to derive a syntax for mathematical knowledge which is consistent with these principles.

As I have always found it easiest to extract general principles by studying examples, I will describe this syntax in connection with a simple well-known example, which is the algebraic geometry of ellipses. Finally, I shall show how to reuse some

of the tools and construct some others to solve the problem of drawing tangent lines to an ellipse.

5 The naming of mathematical objects

Most students will be familiar with the parametric representation of the coordinates of a point on an ellipse in terms of a vector-valued function namely

$$r(t) = (a\cos t, b\sin t) \tag{5.1}$$

in terms of the lengths a and b of the semi-axes, which for any value of the parameter t gives values for the coordinates (x, y) of a point on the ellipse. These can be established as executable *Mathematica* objects using

In[1] := $\mathbf{rvec} = \{\mathbf{a}\,\mathbf{Cos}[\mathbf{t}], \mathbf{b}\,\mathbf{Sin}[\mathbf{t}]\}$

Out[1] = {a Cos[t], b Sin[t]}

to define the components of the vector-valued function, and

In[2] := $\mathbf{coords} = \{\mathbf{x}, \mathbf{y}\}$

Out[2] := {x,y}

for the name of the vector of coordinates. Here the equals sign "=" is used to establish an identity between the name on its left and the mathematical object on its right, which can be seen using

In[3] := $\mathbf{rvec}$

Out[3] := a Cos[t], b Sin[t]

Routine mathematical operations, such as differentiation with respect to the parameter t, can be carried out on the object using

In[4] := $\mathbf{drvec} = \mathbf{D}[\mathbf{rvec}, \mathbf{t}]$

Out[4] := {-a Sin[t], b Cos[t]}

which is a vector in the direction of the tangent to the ellipse.

The names given to mathematical objects should be illustrative and easy to spell. Since all built-in *Mathematica* functions begin with upper case letters, this should be avoided in the names of our objects. The single Roman and Greek letters are reserved for the names of parameters and variables and are never used as names.

I have also found it useful to reserve names like "coords" for lists of variables, because the first coordinate can be referenced using

In[5] := $\mathbf{coords}[[\mathbf{1}]]$

Out[5] := x

independently of the actual name of the variable. This means that we can build tools which work independently of the actual names of variables. Names of objects such as vectors and matrices are given the suffices "vec" or "mat". The names of objects which assign values to variables and parameters end in "set" or "sol", the latter being the case when they emerge as solutions of equations.

6 The assignment of values to variables

To evaluate the coordinates of a point on an ellipse, it is necessary to assign numerical values to the semi-axes and the parameter. A mathematician might say "We let $a = 2, b = 1, t = \pi/4$ ",
which is another usage of the symbol "=". Executing statements like this in *Mathematica* would be disastrous, because such an assignment would actually change the definition of the vector-valued function by replacing the symbols *a* and *b* by the numbers 2 and 1 respectively.

Instead, *Mathematica* provides a construction for assigning temporary values of parameters using

In[6] := $\mathbf{parset} = \{\mathbf{a} \to \mathbf{2}, \mathbf{b} \to \mathbf{1}\}$

Out[6] := $\{\text{a} \to 2, \text{b} \to 1\}$

naming a list of assignment rules for temporarily assigning values of the parameters, whose effect can be seen using

In[7] := $\mathbf{points} = \mathbf{rvec}/.\mathbf{parset}$

Out[7] := {2 Cos[t], Sin[t]}

involving "/.", which means to replace the symbols in the formula with the assigned values in the parameter list. Because of the 2π-periodicity of the trigonometric functions, these points move one revolution around the ellipse as the parameter *t* changes from 0 to 2π, which can be used to define the endpoints of the domain using

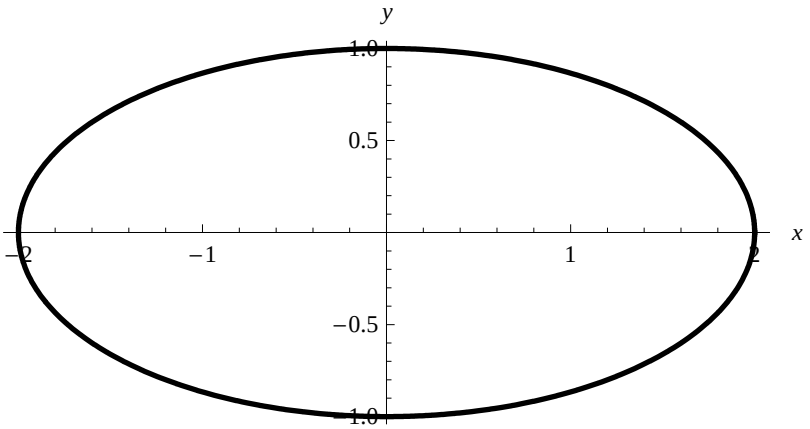
In[8] := $\mathbf{endpts} = \{\mathbf{0}, \mathbf{2}\pi\}$

Out[8] := $\{0, 2\pi\}$

This can be used to visualise the ellipse as a graph using

$$\begin{aligned}&\text{In[9]} := \mathbf{fig1} = \mathbf{ParametricPlot}[\mathbf{points}, \{\mathbf{t}, \mathbf{endpts}[[\mathbf{1}]], \mathbf{endpts}[[\mathbf{2}]]\},\\ &\mathbf{PlotStyle} \to \{\mathbf{Thick}, \mathbf{Black}\}, \mathbf{AxesLabel} \to \mathbf{coords},\\ &\mathbf{AspectRatio} \to \mathbf{Automatic}]\end{aligned}$$

Out[9]=



Notice how the axes are properly labelled with the coordinates
by referring to the name "coords" where they are defined, an example of interoperability. The assignment of the aspect ratio ensures that the units on both axes are the same.

A list of assignment rules for assigning the expressions of the coordinates in terms of the components of the vector-valued function can be generated using

$$\text{In[10]} := \mathbf{xyset} = \mathbf{Table}[\mathbf{coords}[[\mathbf{j}]] \to \mathbf{rvec}[[\mathbf{j}]], \{\mathbf{j}, \mathbf{Length}[\mathbf{coords}]\}]$$

$$\text{Out[10]} := \{\text{x} \to \text{aCos}[t],\ y \to \text{bSin}[t]\}$$

The *Mathematica* function **Length**[list] evaluates to the length of the list. The advantage of using it in this tool is to make it easier to generalise to a space of different dimension.

7 The solution of equations

Another way of representing an ellipse is by defining the equation connecting the Cartesian coordinates, namely

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} - 1 = 0 \tag{7.1}$$

which, stated in words, says that, if the point with coordinates $\{x, y\}$ lies on the ellipse, then the numerical value of the left-hand-side is equal to the number on the right. I have found it useful to always write equations in such a way that the right-hand-side is zero because I can describe an equation by its left-hand-side alone. The left-hand-side of this equation can be established as a *Mathematica* object using

In[11] := $\mathbf{xyeq = \frac{x^2}{a^2} + \frac{y^2}{b^2} - 1}$

Out[11] := $-1 + \frac{x^2}{a^2} + \frac{y^2}{b^2}$

The result of expressing the coordinates in terms of the parametric components can be seen using

In[12] := $\mathbf{Simplify[xyeq /. xyset]}$

Out[12] := 0

thus verifying that the parametric representation actually generates points on the ellipse.

The most common operation carried out on an equation is to solve it for one of the variables in terms of the others. From the point of view of plotting the graph, the most natural one to choose is the vertical coordinate y. The process of solution can be described as "if the left-hand-side of equation (9) is zero, find the value of y", clearly yet another interpretation of the symbol "=".

This solution can be carried out by the *Mathematica* function **Solve** using

$$\mathbf{In[13] := sol = Simplify[Solve[xyeq == 0, y],}$$
$$\mathbf{\{a > 0, b > 0, Abs[x] < a\}]}$$

Out[13] := $\left\{\left\{y \to -b\sqrt{1 - \frac{x^2}{a^2}}\right\}, \left\{y \to b\sqrt{1 - \frac{x^2}{a^2}}\right\}\right\}$

The meaning of this statement is "if the object xyeq is zero, find the values of y". Notice that *Mathematica* reserves the symbol "==" for this use of conditional or logical equality. Needless to say, the
function **Solve** contains a lot of technical knowledge about how to find the symbolic solution of equations. **Simplify** is one of a family of *Mathematica* functions which can be used to try to reduce complex expressions to more compact or simpler form. The list of inequalities express constraints on the parameters and variables to assist in the simplification process.

We see that two solutions are found, one the negative of the other. It is important to notice that the solutions are presented in the form of assignment rules, making it

relatively easy to check that they actually are solutions by replacing the value of y in the left-hand-side of the equation using

In[14] := **Simplify**[**xyeq**/.**sol**[[**1**]]]

Out[14] := 0

for the first solution and

In[15] := **Simplify**[**xyeq**/.**sol**[[**2**]]]

In[15] := 0

for the second. Since the value of x changes from $-a$ to a, for the purpose of plotting, the endpoints of the domain can be established for the current ellipse using

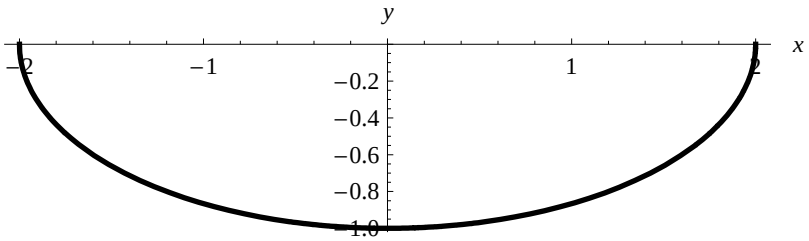
In[16] := **endpts** = {−**a**, **a**}/.**parset**

Out[16] := {−2, 2}

The value of y for the first solution can be plotted as a function of x using

In[17] :=
fig2 = **Plot**[**y**/.**sol**[[**1**]]/.**parset**, {**x**, **endpts**[[**1**]], **endpts**[[**2**]]},
PlotStyle → {**Thick**, **Black**}, **AxesLabel** → **coords**,
AspectRatio → **Automatic**]

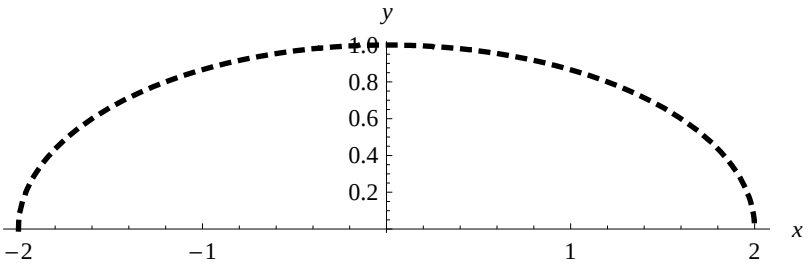
Out[17] :=



and the second using

In[18] := **fig3** = **Plot**[**y**/.**sol**[[**2**]]/.**parset**, {**x**, **endpts**[[**1**]], **endpts**[[**2**]]},
PlotStyle → {**Thick**, **Black**, **Dashing**[**Small**]}, **AxesLabel** → **coords**,
AspectRatio → **Automatic**]

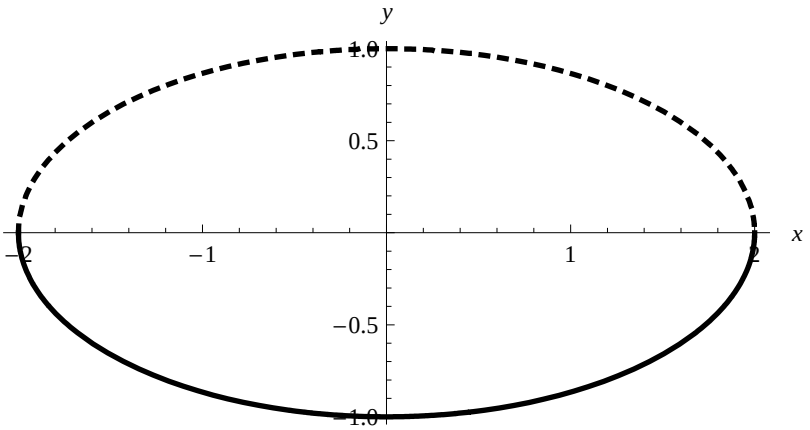
Out[18] :=



The two graphs can be superimposed using

In[19] := **fig4 = Show[fig2, fig3, PlotRange → All]**

Out[19] :=



which is the same as that found in the previous section, except that the top and bottom halves of the ellipse are shaded differently.

8 The tangents to an ellipse

A common tutorial (and examination) problem is to find and visualise the tangent to a curve defined by an equation of the form (4). As the solution of this problem involves all 3 uses of equality, and illustrates the concept of reuse, we carry it out for the case of the ellipse studied above at the points on the ellipse for which $x = 1$.

8.1 The first tangent line

8.1.1 Coordinates of the point on the ellipse

The values of the coordinates of the first point can be found by evaluating $\{x, y\}$ at the first solution of equation (4) with y, a function of x, evaluated at $x = 1$ for the current values of the semi-axes. This can be done using

In[20] := **coords0 = coords/.sol[[1]]/.x → 1/.parset**

Out[20] := $\left\{1, -\frac{\sqrt{3}}{2}\right\}$

which has been named for future reference.

8.1.2 The normal vector

The normal vector to the curve at the first point can be computed by calculating the vector of partial derivatives of the left-hand-side of equation (4) with respect to the coordinates and evaluating the result at the first point on the ellipse using

$$\begin{aligned}&\text{In[21]} := \mathbf{nvec} = \mathbf{Table}[\mathbf{D}[\mathbf{xyeq}, \mathbf{coords}[[\mathbf{j}]]], \{\mathbf{j}, \mathbf{Length}[\mathbf{coords}]\}]\\ &/.\mathbf{sol}[[\mathbf{1}]]/.\\ &\mathbf{x} \to \mathbf{1}/.\mathbf{parset}\end{aligned}$$

$$\text{Out[21]} := \left\{\tfrac{1}{2}, -\sqrt{3}\right\}$$

8.1.3 Equation of the tangent

Since the vector joining a point $\{x, y\}$ on the tangent to the first point on the ellipse must be perpendicular to the normal vector at that point, the left-hand-side of the Cartesian equation of the tangent line can be found by taking the scalar product of the two vectors using

$$\text{In[22]} := \mathbf{tangenteq} = \mathbf{Simplify}[(\mathbf{coords} - \mathbf{coords0}).\mathbf{nvec}]$$

$$\text{Out[22]} := \tfrac{1}{2}\left(-4 + x - 2\sqrt{3}y\right)$$

which can be set equal to zero and solved for y as a function of x using

$$\text{In[23]} := \mathbf{ysol} = \mathbf{Flatten}[\mathbf{Solve}[\mathbf{tangenteq} == \mathbf{0}, \mathbf{y}]]$$

$$\text{Out[23]} := \left\{\mathrm{y} \to \tfrac{-4+\mathrm{x}}{2\sqrt{3}}\right\}$$

Flatten is a *Mathematica* function which suppresses surplus pairs of "{}" brackets.

8.1.4 Visualising the tangent to the ellipse

Convenient endpoints of the tangent graph can be entered using

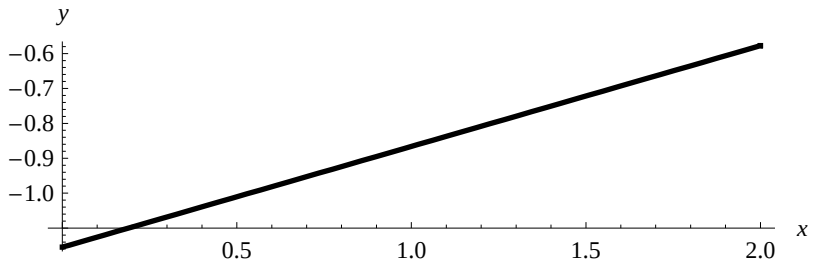
$$\text{In[24]} := \mathbf{endpts} = \{\mathbf{0}, \mathbf{2}\}$$

$$\text{Out[24]} := \{0, 2\}$$

A graph of the first tangent line can be drawn using

In[25] := **fig5** = **Plot**[**y**/.**ysol**/.**parset**, {**x**, **endpts**[[**1**]], **endpts**[[**2**]]},
PlotStyle → {**Thick**, **Black**}, **AxesLabel** → **coords**,
AspectRatio → **Automatic**]

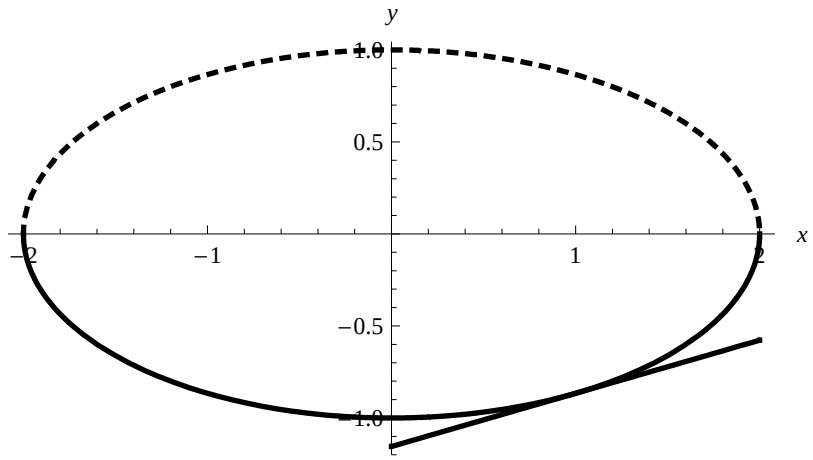
Out[25] :=



which can be superimposed on the graph of the ellipse using

In[26] := **Show**[**fig4**, **fig5**]

Out[26] :=



The graph of the second tangent line is easily generated by customizing a copy of this section.