

Classroom Tips and Techniques: Partial Derivatives by Subscripting

Robert J. Lopez

Emeritus Professor of Mathematics and Maple Fellow
Maplesoft

Introduction

As output, Maple can display the partial derivative $\frac{\partial}{\partial x} f(x, y)$ as f_x ; that is, subscript notation can be used to display partial derivatives, and it can be done with two completely different mechanisms. This article describes these two techniques, and then investigates the extent to which partial derivatives can be calculated by subscript notation.

Discussion

The prime as the symbol of ordinary differentiation is very compact (and generally convenient), especially when the derivative is to be evaluated at some specific value of the independent variable. Thus, for example, we have $g'(h(u))$ to indicate that the derivative of $g(x)$ is to be evaluated at the argument $h(u)$. Of course, the notation

$$\left. \frac{dg}{dx} \right|_{x=h(u)}$$

is perhaps a bit less ambiguous, but if g is declared to be a function of x , and prime to mean differentiation with respect to x , then the prime notation is pretty clear. Moreover, for typesetting in a manuscript, the prime notation is a definite space-saver.

The parallel for partial derivatives is the literal *subscript*. There are texts such as [1] that indicate partial derivatives with *numeric* subscripts so that for the function $f(x, y)$, partial differentiation with respect to x would be indicated by $f_1(x, y)$. If it is clear that f is a function of x and y , then, like the case for primes with ordinary derivatives, the notation $f_1(u, v)$ is unambiguous.

Given the function $f(x, y)$, for evaluating its partial with respect to x at $(x, y) = (u, v)$, the notation $f_x(u, v)$ is more compact and just as clear as the operator form

$$\left. \frac{\partial f}{\partial x} \right|_{(x, y) = (u, v)}$$

Many texts use the literal subscript for partial differentiation; this notation appears, for example, in [2, 3], two of the classics I still have on my shelf.

The *declare* Command in the PDEtools Package

The first mechanism in Maple that allowed partial derivatives to be displayed with subscripts is the **declare** command in the *PDEtools* package. This command predates the introduction of the *Typesetting* package that allows commands to be entered in math mode, or in what Maplesoft calls "2D math". Thus, declare would function in math mode, but it is best understood as an artifact of the "1D math" environment.

Within the confines of *PDEtools* package, executing the **declare** command changes the display of the partial derivative of $f(x, y)$ with respect to x to f_x . So, the derivative is expressed with a literal subscript, and the display of the independent variables is suppressed. Table 1 illustrates the basic functionality of this command.

<ul style="list-style-type: none"> Partial derivative by the explicit diff command. 	$diff(f(x, y), x) = \frac{\partial}{\partial x} f(x, y)$
<ul style="list-style-type: none"> Load the <i>PDEtools</i> package. 	$with(PDEtools) :$
<ul style="list-style-type: none"> The package itself does not change the behavior of the display of a derivative. 	$diff(f(x, y), x) = \frac{\partial}{\partial x} f(x, y)$
<ul style="list-style-type: none"> One form of the display command. 	$declare() :$ <i>Nothing declared</i>
<ul style="list-style-type: none"> Partial derivative now displayed with literal subscript. 	$diff(f(x, y), x) = f_x$

Table 1 Basic functionality of the **declare** command in the *PDEtools* package

The "Nothing declared" echo in the fourth line of the table indicates that no specific function has been "declared;" hence, all unspecified functions will display with independent variables suppressed, and all partial derivatives will display as literal subscripts. There is a "quiet" option for the declare command that suppresses any output it would otherwise provide. There are **undeclare**, **ON**, and **OFF** commands to remove declarations, and to turn on and off the functionality of the display without having to invoke **undeclare**. These details are available in the [help page](#) for **declare**, where one can find the important excerpt we provide below.

- Different from macro and alias, this scheme handles only '**print/foo**' subroutines. That is, it does not change input or output, only the typesetting on the screen.

This paragraph clarifies that the changes induced by **declare** affect only the *display* (i.e., printing) of derivatives and functions. Neither input nor output is changed in any essential way. Just the display changes.

The *Typesetting* Package

The *Typesetting* package contains the code by which Maple translates "2D math" notation to the original linear text form of its commands. Within the package, the **Settings**, **Suppress**, and **Unsuppress** commands are used to effect changes in the display of partial derivatives. In addition, the typesetting "level" must be changed from the default *standard* to *extended*. The use of these commands and options is summarized in Table 2.

<ul style="list-style-type: none"> Remove the <i>PDEtools</i> package and its declare command. 	<i>restart</i>
<ul style="list-style-type: none"> Install the <i>Typesetting</i> package. 	<i>with(Typesetting) :</i>
<ul style="list-style-type: none"> Set the <i>typesetting</i> level to <i>extended</i> (from <i>standard</i>). 	<i>interface(typesetting = extended) :</i>
<ul style="list-style-type: none"> The <i>userep</i> option in the Settings command tells Maple that partial derivatives are to be indicated by repeated subscripts. 	<i>Settings(userep = true) :</i>
<ul style="list-style-type: none"> The <i>userep</i> option only works if the independent variables have been suppressed by the Suppress command. 	<i>Suppress(f(x,y)) :</i>
<ul style="list-style-type: none"> The partial differential operator from the Expression palette is applied to f. The Context Menu option Evaluate and Display Inline is used to evaluate the partial derivative. Note that the Suppress command, unlike the declare command, allows suppression of independent variables upon <i>input</i>. 	$\frac{\partial}{\partial x} f = f_x$
<ul style="list-style-type: none"> The Unsuppress command removes the effects of the Suppress command. 	<i>Unsuppress(f) :</i>
<ul style="list-style-type: none"> That independent variables must now be explicitly included is evident from the vanishing 	$\frac{\partial}{\partial x} f = 0$

of the derivative.	
Table 2 Partial derivatives displayed with literal subscripts through the <i>Typesetting</i> package	

Subscript as a Partial Differentiation Operator - 1

Maple does not make directly available the subscript as a partial differentiation operator. The experiment that led to this article consists of the display in Table 3.

Remove the <i>Typesetting</i> package and its commands.	<code>restart</code>
Restore the <i>standard</i> typesetting level. Note that a restart does not restore the typesetting level to its default value.	<code>interface(typesetting = standard) :</code>
Type the equation $f(x) = \dots$ Context Menu: Assign Function	$f(x, y) = x^2 + y^2 \xrightarrow{\text{assign as function}} f$
Attempt to compute $\frac{\partial f}{\partial x}$ at (a, b) by subscripting f . The returned value is not $f_x(a, b)$, it's $f(a, b)$.	$f_x(a, b) = a^2 + b^2$
Table 3 The failure of subscripts as partial differentiation operators	

The desire to use subscripts as partial differentiation operators is at the heart of this article. That Maple fails as per Table 3 was a surprise, even after 23 years of using Maple as the mathematical tool of daily usage. Fortunately, the Maple programmers explained that this is a feature of the software, namely, that both f and f_x evaluate to the same function. In fact, this is the feature that allows $\log(x)$ to point to the natural log function, and $\log_a(x)$ to point to the base a logarithm. The secret is in the coding, an example of which is given in Table 4.

```

f := proc(u, v)
  local var, g, F;
  F := (U, V) → U2 + V2;
  if type(procname, indexed) then
    var := op(procname);
    if var = x then g := D1(F)(u, v);
    elif var = y then g := D2(F)(u, v);
    else g := 0;
    end if;
  else
    g := F(u, v);
  end if;
  g;
end proc:

```

Table 4 Coding $f(x, y) = x^2 + y^2$ so that f_x and f_y compute partial derivatives

Proof that the code in Table 4 defines the function $f(x, y) = x^2 + y^2$ in such a way as to permit literal subscripts to act as partial differentiation operators is in the calculations in Table 5.

$$\begin{aligned}
 f_x(a, b) &= 2a \\
 f_y(a, b) &= 2b \\
 f_w(a, b) &= 0 \\
 f(a, b) &= a^2 + b^2
 \end{aligned}$$

Table 5 Partial differentiation by subscripting

Of course, the coding of the function in Table 4 is beyond tedious. Although the code illustrates what is possible, it is not a practical solution to the problem of using subscripts operationally for partial differentiation.

Subscript as a Partial Differentiation Operator - 2

Table 6 summarizes a practical and workable method for defining subscripts as partial differentiation operators. The method hinges on the *Atomic Identifier* construct available in math mode. Loosely speaking, an Atomic Identifier is a collection of symbols that have been "locked" together to form a single *name*. In Maple, an unassigned letter or string of letters is the name of a variable. Some strings of symbols - for example, $x y$ - do not form a Maple name because of the space between the two letters. Assignments cannot be made to such collections of characters. However, when frozen into the construction called an Atomic Identifier, they form a valid name.

There are several ways to form an Atomic Identifier. For complex collections of symbols, it is

best to assemble the object, select it, and use Context Menu: 2D Math>Convert To>Atomic Identifier. The alternative to the Context Menu is the Format Menu's "Convert To" option, ending at "Atomic Identifier." The keyboard equivalent in the ® operating systems is Control+Shift+A. For the

For a literal subscript, it's easier. Instead of setting the subscript with the underscore key, add the Control key to the combination. Thus, it's Control+Shift+minus in the first two systems, and Command+Shift+minus on the last. Since this adds only one key to the normal combination for setting a subscript, setting the literal subscript and thereby forming an Atomic Identifier is relatively painless. Finally, there is the a_* template in the Expression palette. This also creates a subscripted name as an Atomic Identifier.

Table 6 summarizes how to use subscripts, formed as Atomic Identifiers, as partial differentiation operators.

<ul style="list-style-type: none"> Execute the restart command to clear all previous assignments. 	<i>restart</i>
<ul style="list-style-type: none"> Type the equation $f(x,y) = \dots$ Context Menu: Assign Function 	$f(x,y) = x^2 + y^2 \xrightarrow{\text{assign as function}} f$
Type the equation shown to the right. After typing the first f , set the subscript as an Atomic Identifier. <ul style="list-style-type: none"> Context Menu: Assign Function 	$f_x(x,y) = \frac{\partial}{\partial x} f(x,y) \xrightarrow{\text{assign as function}} f_x$
Type the left side of the equation shown to the right. The symbol f_x must be set as an Atomic Identifier. <ul style="list-style-type: none"> Context Menu: Evaluate and Display Inline 	$f_x(a,b) = 2a$
Table 6 Use of the Atomic Identifier construct to set a subscript as a partial differentiation operator	

The astute reader will realize that the partial derivative has been defined as a function, just as $f(x,y)$ has been. The Atomic Identifier device has allowed the symbols f and f_x to be seen as unrelated by Maple, thus bypassing the built-in rule that Maple functions do not distinguish between a function's name and that name subscripted. This may not be completely satisfactory, since each partial derivative has to be defined as a separate function. But the notational clarity that this device can bring to a manuscript makes it worth considering.

References

- [1] David V. Widder, *Advanced Calculus*, Second Edition, Dover Publications, 1989.
- [2] Watson Fulks, *Advanced Calculus: An Introduction to Analysis*, John Wiley & Sons, Publishers, 1961.
- [3] John M. H. Olmsted, *Advanced Calculus*, Prentice-Hall, 1961.