

Troubleshooting Maple Worksheets: Common Problems

So you've seen plenty of worksheets that work just fine, but that doesn't always help you much when your worksheet isn't doing what you want it to. This worksheet is full of commands that don't work as you'd expect them to; common errors that come up frequently with a mistyped expression and strange output types that require a little more work. This should give you some clues to look for when your own worksheet gives you strange output or no output at all.

[> *restart*;

My command does nothing!

This is one of the most common problems: you enter in a command, everything looks fine, but your output looks almost exactly like your command.

I'd like to generate a random matrix, so I'll use the following command:

```
[> Randommatrix (5, 5);  
Randommatrix (5, 5) (1.1)
```

You've entered in what looks like a perfectly good command, but when you try to execute it, your output is the same as what you entered in the first place. This could be caused by a few things.

Did you use the correct case?

Remember that Maple is case sensitive, so type your command into Maple Help and make sure that you used the correct case and spelling! I should have used a capital *M*, so the correct command is actually:

```
[> RandomMatrix (5, 5);  
RandomMatrix (5, 5) (1.2)
```

but that didn't work either, so we need to try something else.

Did you load all of the packages you need?

Many commands are not available by default, so you may need to load them. Go back to Maple Help and see if this command is part of a package. This one is in *LinearAlgebra*, so load that using the *with* command:

```
[> with (LinearAlgebra) :  
RandomMatrix (5, 5);  
LinearAlgebra:-RandomMatrix (5, 5) (1.3)
```

Now we have something different, but there's still something weird going on.

Did you check your spacing?

Maple will automatically format as you type, so it can be difficult to pick up subtle differences like spacing. Here, I have an extra space after *RandomMatrix*, so if I adjust the command, I get:

```
[> RandomMatrix(5, 5);  
-81 -98 -76 -4 29  
-38 -77 -72 27 44  
-18 57 -2 8 92  
87 27 -32 69 -31  
33 -93 -74 99 67 (1.4)
```

Did you execute your previous commands?

Remember that order matters in Maple; if you saved your worksheet previously, you'll need to execute all of your previous commands. You may know what y is, but Maple has nothing stored until you execute the command:

$$\begin{aligned} > \frac{d}{dt} y; \\ & 0 \end{aligned} \tag{1.5}$$

$$\begin{aligned} > y := t^2 + 4t + 1; \\ & y := t^2 + 4t + 1 \end{aligned} \tag{1.6}$$

You will get different output if you execute these two commands in opposite order.

Why doesn't it simplify?

I have an equation I'd like to simplify:

$$\begin{aligned} > \frac{zx \left(e^{\ln \left(\sin \left(\frac{\pi}{2} \right) \right)} \right)}{z}; \\ & \frac{zx \left(e^{\ln \left(\sin \left(\frac{1}{2} \pi \right) \right)} \right)}{z} \end{aligned} \tag{1.7}$$

This looks as though it should simplify, but nothing happened. A few things are wrong here.

Maple variables can be more than one letter long, the zx here is interpreted as a new variable instead of z times x . When in doubt, use a multiplication symbol:

$$\begin{aligned} > \frac{z \cdot x \left(e^{\ln \left(\sin \left(\frac{\pi}{2} \right) \right)} \right)}{z}; \\ & x \left(e^{\ln \left(\sin \left(\frac{1}{2} \pi \right) \right)} \right) \end{aligned} \tag{1.8}$$

Maple also does not recognize this e ; it is assumed to be a variable and not the constant. You can use autocomplete or the palette to select the constant e , or use the \exp function:

$$\begin{aligned} > \frac{z \cdot x \left(e^{\ln \left(\sin \left(\frac{\pi}{2} \right) \right)} \right)}{z}; \\ & x \left(\sin \left(\frac{1}{2} \pi \right) \right) \end{aligned} \tag{1.9}$$

$$\begin{aligned} > \frac{z \cdot x \left(\exp \left(\ln \left(\sin \left(\frac{\pi}{2} \right) \right) \right) \right)}{z}; \\ & x \left(\sin \left(\frac{1}{2} \pi \right) \right) \end{aligned} \tag{1.10}$$

We have a similar problem with π . Maple identifies 'pi' as a variable and 'Pi' as a constant; case matters! You can also use autocomplete or the palette:

$$\begin{aligned} > \frac{z \cdot x \left(e^{\ln \left(\sin \left(\frac{\text{Pi}}{2} \right) \right)} \right)}{z}; \\ & x(1) \end{aligned} \tag{1.11}$$

$$\begin{aligned} > \frac{z \cdot x \left(e^{\ln \left(\sin \left(\frac{\pi}{2} \right) \right)} \right)}{z}; \\ & x(1) \end{aligned} \tag{1.12}$$

Now we expect to see x , but here Maple is interpreting $x(1)$ as a function so we have a multiplication problem again:

$$\begin{aligned} > \frac{z \cdot x \cdot \left(e^{\ln \left(\sin \left(\frac{\pi}{2} \right) \right)} \right)}{z}; \\ & x \end{aligned} \tag{1.13}$$

Weird output

I got a bunch of extra stuff!

I want to set up an equation:

```
> z := y^2 + 2 x·y + x^2;
z := (t^2 + 4 t + 1)^2 + 2 x (t^2 + 4 t + 1) + x^2
```

 (2.1)

but I got a bunch of extra *t*'s and my *y*'s disappeared. Something is stored to one of our variables, so you have a couple of options. A *restart* command will reset everything in your entire worksheet.

```
> restart;
z := y^2 + 2 x·y + x^2;
z := y^2 + 2 x y + x^2
```

 (2.2)

This is why it's a good idea to start every worksheet with a *restart*, but this might not be a good idea if you need to use any of your previous work. To clear out just one variable, you can use the following:

```
> y := 'y';
z := y^2 + 2 x·y + x^2;
y := y;
z := y^2 + 2 x y + x^2
```

 (2.3)

My output has RootOf instead of the solution

Let's say you're trying to solve an equation or a system of equations, here I'm going to solve a system of equations for *x* and *y*

```
> solve( {x + y = 0, x·y = b}, {x, y});
{x = -RootOf(_Z^2 + b), y = RootOf(_Z^2 + b)}
```

 (2.4)

but instead of a set of solutions, you get this *RootOf* output instead. You could solve this yourself by hand and get a solution with no problem, what happened?

Maple uses *RootOf* to represent the roots of an equation when the solutions are for a polynomial equation with degree greater than one. It is also used when Maple is unable to find an exact solution. You can find the values of all of the roots using the *allvalues* command:

```
> allvalues( solve( {x + y = 0, x·y = b}, {x, y} ) );
{x = -sqrt(-b), y = sqrt(-b)}, {x = sqrt(-b), y = -sqrt(-b)}
```

 (2.5)

You might have trouble finding an exact solution to something like this:

```
> z := solve( cos(x^2) = 2·cos(x) + x, x);
z := RootOf(2 cos(_Z) - cos(_Z^2) + _Z)
```

 (2.6)

```
> zsol := allvalues(z);
zsol := RootOf(2 cos(_Z) - cos(_Z^2) + _Z, -0.6680246832), RootOf(2 cos(_Z) - cos(_Z^2) + _Z,
-1.617201118 + 0.4595592187 I), RootOf(2 cos(_Z) - cos(_Z^2) + _Z, 0.9829208368
+ 0.8182877547 I), RootOf(2 cos(_Z) - cos(_Z^2) + _Z, 0.9829208368 - 0.8182877547 I),
RootOf(2 cos(_Z) - cos(_Z^2) + _Z, 0.6269987145 + 2.502736498 I), RootOf(2 cos(_Z) - cos(_Z^2)
+ _Z, 2.423583949), RootOf(2 cos(_Z) - cos(_Z^2) + _Z, 2.600945398), RootOf(2 cos(_Z)
- cos(_Z^2) + _Z, -0.6636868040 + 2.424595711 I), RootOf(2 cos(_Z) - cos(_Z^2) + _Z,
-1.617201118 - 0.4595592187 I)
```

 (2.7)

That looks even worse, but if you look carefully, you'll see that this time, it included decimal approximations with each *RootOf*. You can use *evalf* to display only decimal approximations:

```
> evalf(zsol);
-0.6680246832, -1.617201118 + 0.4595592187 I, 0.9829208368 + 0.8182877547 I, 0.9829208368
- 0.8182877547 I, 0.6269987145 + 2.502736498 I, 2.423583949, 2.600945398, -0.6636868040
+ 2.424595711 I, -1.617201118 - 0.4595592187 I
```

 (2.8)

Be careful! If you use *evalf* too early, it will only give you the principal root:

```
> evalf(z);
-0.6680246832
```

 (2.9)

The dreaded pink error text

Error text often means that you have something in the wrong place, or forgot to type something. Usually you can click on an error and it will take you to a webpage explaining what's wrong.

In some cases, it will highlight the problem area, as with an invalid sequence or an unable to parse error:

```
> (2.)+4;  
Error, invalid sequence          (2.) + 4;  
  
> 2.  
Error, unable to parse          2.
```

You might encounter an unable to match delimiters error, this means that your brackets don't match up; check to see if you have too many or too few:

```
> 2.5)+4;  
Error, unable to match delimiters  
2.5)+4;
```

If you enter the wrong kind of information into a function, you'll probably get something like the following. It will usually tell you what sort of input the function expects, and at the end it will list the problem input:

```
> plot(2x = 4y);  
Error, invalid input: plot expects its 1st argument, p, to be  
of type {set, array, list, rtable, algebraic, procedure, And  
(`module`, applicable)}, but received 2*x = 4*y
```

In this case, $2x = 4y$ is causing the problem.

Nothing is working!

Sometimes you just get weird output even when you can't find anything wrong. This usually means that there's something subtle happening in one of your functions or it's a typing error that's impossible to see with the automatic formatting.

Try typing it out again

Don't copy and paste, but type out your command again on a separate line.

Make sure that you're in math mode!

When you select the command, make sure that Math is highlighted in the toolbar. I separate my text from commands using Maple input prompts to avoid this problem. If you're using the standard input boxes for commands, you may run into more problems.

Check your packages

If you're using commands from many different packages, sometimes they interact with each other in unexpected ways. Load only the packages you need! If it's a package causing you trouble, you can use the `unwith` command to unload a package that you were using previously. *VecCalc* has a lot of strange interactions with other commands, so if you're using this package, try to use only commands in this package. If I want to use *VectorCalculus* for the first half of the assignment and *VecCalc* for the second half, I could do the following to switch to *VecCalc*:

```
> with( VectorCalculus );
[ &x, `*`, `+`, `^`, `.` , <, >, <|>, About, AddCoordinates, ArcLength, BasisFormat, Binormal, Compatibility, (4.1)
  ConvertVector, CrossProd, CrossProduct, Curl, Curvature, D, Del, DirectionalDiff, Divergence, DotProd,
  DotProduct, Flux, GetCoordinateParameters, GetCoordinates, GetNames, GetPVDDescription,
  GetRootPoint, GetSpace, Gradient, Hessian, IsPositionVector, IsRootedVector, IsVectorField, Jacobian,
  Laplacian, LineInt, MapToBasis, Nabla, Norm, Normalize, PathInt, PlotPositionVector, PlotVector,
  PositionVector, PrincipalNormal, RadiusOfCurvature, RootedVector, ScalarPotential,
  SetCoordinateParameters, SetCoordinates, SpaceCurve, SurfaceInt, TNBFrame, Tangent, TangentLine,
  TangentPlane, TangentVector, Torsion, Vector, VectorField, VectorPotential, VectorSpace, Wronskian,
  diff, eval, evalVF, int, limit, series ]
> # Here are all my commands using the VectorCalculus Package, then I change my mind and switch to VecCalc
  instead
> unwith( VectorCalculus );
  with( VecCalc ) :
Initializing Package:  VecCalc  Version 8.0
All HELP is linked from ?VecCalc
To use aliases, execute:  VCalias;
Loading packages:  student  plots
```

Check the input and output types

Maple help can give you information about what parameters various functions expect, and what they give as output. If you're not sure what sort of output you're getting from a function, you can use *whattype* to find out:

```
> whattype(x);
symbol (4.2)
```