

Thursday's homework

Mike May, S.J.

Someone asked about the assign command.

Let me add some context. I want to evaluate a function f at a set of points where two curves intersect.

```
> f := (x,y) -> x^2 + y^3 + x*y^2;
curve1 := x^2+y^2-25;
curve2 := 4*x-3*y;
```

$$f := (x, y) \rightarrow x^2 + y^3 + xy^2$$
$$\text{curve1} := x^2 + y^2 - 25$$
$$\text{curve2} := 4x - 3y$$

(1)

To find the points of intersection I use the solve command.

```
> answers := solve({curve1, curve2}, [x, y]);
answers := [[x=3, y=4], [x=-3, y=-4]]
```

(2)

I can use subscripts to refer to the answers.

```
> answer1 := answers[1];
answer2 := answers[2];
```

$$\text{answer1} := \{x=3, y=4\}$$
$$\text{answer2} := \{x=-3, y=-4\}$$

(3)

We can use the eval command to evaluate the function at these points.

```
> v1 := eval(f(x,y), answer1);
v2 := eval(f(x,y), answer2);
```

$$v1 := 121$$
$$v2 := -103$$

(4)

I can now use the assign command to assign the values in the answer1 to x and y and evaluate f at the point.

```
> assign(answer1);
v1a := f(x,y);
```

$$v1a := 121$$

(5)

The problem occurs when I try to repeat the process with answer2.

```
> assign(answer2);
v2a := f(x,y);
Error, (in assign) invalid arguments
v2a := 121
```

(6)

The heart of the problem is apparent when I look back at answers.

```
> answers;
answer1;
answer2;
```

$$\{3=3, 4=4\}, \{3=-3, 4=-4\}$$
$$\{3=3, 4=4\}$$
$$\{3=-3, 4=-4\}$$

(7)

Note that the variables in the equations have been replaced with the assigned values.

The remedy is to use the unassign command.

```

> unassign('x', 'y');
answers;
answer1;
answer2;

```

$\{x=3, y=4\}, \{x=-3, y=-4\}$
 $\{x=3, y=4\}$
 $\{x=-3, y=-4\}$

(8)

Some other technical details:

We defined answer1 as the first element in the set answers. We can refer to the second element of answer1 by several methods.

```

> an12a :=answer1[2];
an12b :=answers[1][2];
an12c :=answers[1,2];

```

$an12a := y = 4$
 $an12b := y = 4$
 $an12c := y = 4$

(9)

We can also use rhs and lhs to get sides of an equation.

```

> rt := rhs(an12a);
lt := lhs(an12a);

```

$rt := 4$
 $lt := y$

(10)

```

> nops(an12a);

```

2

(11)

```

> op(1, an12a);
op(2, an12a);

```

y
 4

(12)

```

>

```