>

Partial Differential Equations

Partial Derivatives with `diff`

Partial derivatives of an expression can be calculated using **`diff`** command. The structure of **`diff`** is

diff(**a**, **x1**, **x2**, ..., **xn**)

diff computes the partial derivative of the expression **a** with respect to **x1, x2, ..., xn**, respectively. The most frequent use is **diff(f(x),x)**, which computes the derivative of the function **f(x)** with respect to **x**. Note that where **n** is greater than **1**, the call to **diff** is the same as diff called recursively. Thus **diff(f(x), x, y);** is equivalent to the call **diff(diff (f(x), x), y).**
The sequence operator $ is useful for forming higher-order derivatives. **diff(f(x),x$4)**, for example, is equivalent to **diff(f(x),x,x,x,x)** and **diff(g(x,y),x$2,y$3)** is equivalent to **diff(g(x,y),x,x,y,y,y)**

Let's consider the function $f(x,y) = e^{(x+y^2)}$

```
> f:=(x,y)->exp(x+y^2);
```

$$f := (x,y) \to e^{(x+y^2)}$$

The first order partial derivatives $\frac{\partial}{\partial x} f(x,y)$ and $\frac{\partial}{\partial y} f(x,y)$ can be obtained as follows:

```
> diff(f(x,y),x);
```

$$e^{(x+y^2)}$$

```
> diff(f(x,y),y);
```

$$2\, y\, e^{(x+y^2)}$$

The second order partial derivatives $\frac{\partial^2}{\partial x^2} f(x,y)$, $\frac{\partial^2}{\partial y\, \partial x} f(x,y)$, $\frac{\partial^2}{\partial y^2} f(x,y)$ have the expressions:

```
> diff(f(x,y),x,x);
```

$$e^{(x+y^2)}$$

or

```
> diff(f(x,y),x$2);
```

$$e^{(x+y^2)}$$

```
> diff(f(x,y),x,y);
```

$$2\, y\, e^{(x+y^2)}$$

```
> diff(f(x,y),y$2);
```

$$2\, e^{(x+y^2)} + 4\, y^2\, e^{(x+y^2)}$$

We can calculate also higher order partial derivatives using the sequnce operator $. For example $\frac{\partial^5}{\partial y^3 \partial x^2} f(x, y)$ is

```
> diff(f(x,y),x$2,y$3);
```

$$12\, y\, e^{(x+y^2)} + 8\, y^3\, e^{(x+y^2)}$$

\>

Partial derivatives with D

When we need to evaluate some partial derivative in some given point we have to use the operator D. The structure is

D[**i**](**f**)

`f` - expression which can be applied as a function
`i` - positive integer or expression or sequence of such integers

Given **f**, a function of one argument, the call **D(f)** computes the derivative of the function **f**. For example, **D(sin)** returns **cos**. The derivative is a function of one argument such that **D(f)(x) = diff(f(x), x)**. That is, **D(f) = unapply(diff(f(x), x), x)**. Thus **D** is a mapping from unary functions to unary functions.
Given **f**, a function of **n** arguments, the call **D[i](f)** computes the partial derivative of **f** with respect to its **i**th argument. More generally, **D[i, j](f)** is equivalent to **D[i](D[j](f))**, and **D[ ](f) = f**. Thus **D[i]** is a mapping from n-ary functions to n-ary functions.
It is an error to specify a call to the derivative of a function with respect to the **k**-th variable and apply it to fewer than **k** variables. For instance, **D[2](f)(x)** yields an error.

Let's consider, again, the function $f(x, y) = e^{(x+y^2)}$
The first order partial derivatives $\frac{\partial}{\partial x} f(x, y)$ and $\frac{\partial}{\partial y} f(x, y)$ can be obtained as follows:

```
> D[1](f)(x,y);
```

$$e^{(x+y^2)}$$

```
> D[2](f)(x,y);
```

$$2\, y\, e^{(x+y^2)}$$

Using **D** operator we can obtain directly the value of $\frac{\partial}{\partial x} f$ (0,0) and $\frac{\partial}{\partial y} f$ $(0, \ln(2))$:

```
> D[1](f)(0,0);
```

$$1$$

```
> D[2](f)(0,ln(2));
```

$$2 \ln(2)\, e^{(\ln(2)^2)}$$

```
> evalf(%);
```

$$2.241369972$$

The second order partial derivatives $\frac{\partial^2}{\partial x^2} f(x,y)$, $\frac{\partial^2}{\partial y\,\partial x} f(x,y)$, $\frac{\partial^2}{\partial y^2} f(x,y)$ have the expressions:

```
> D[1,1](f)(x,y);
```

$$e^{(x+y^2)}$$

```
> D[1,2](f)(x,y);
```

$$2\,y\,e^{(x+y^2)}$$

```
> D[2,2](f)(x,y);
```

$$2\,e^{(x+y^2)} + 4\,y^2\,e^{(x+y^2)}$$

For higher order partial derivatives we have to specify the list with the derivation order, for example $\frac{\partial^5}{\partial y^3\,\partial x^2} f(x,y)$ is

```
> D[1,1,2,2,2](f)(x,y);
```

$$12\,y\,e^{(x+y^2)} + 8\,y^3\,e^{(x+y^2)}$$

or simply

```
> D[1$2,2$3](f)(x,y);
```

$$12\,y\,e^{(x+y^2)} + 8\,y^3\,e^{(x+y^2)}$$

```
> restart;
```

Partial differential equations

Partial differential equations can be solved using **pdsolve** command.

First we load the partial differential equation package

```
> with(PDEtools):
```

The results of **pdsolve** are returned, by default, in one of three forms.

a) When the general solution to the PDE is obtained, the function returns an explicit result for the indeterminate function.

For example in the case of the equation: $x\left(\frac{\partial}{\partial y} u(x,y)\right) - y\left(\frac{\partial}{\partial x} u(x,y)\right) = 0$

```
> eq:=x*diff(u(x,y),y)-y*diff(u(x,y),x) = 0;
```

$$eq := x\left(\frac{\partial}{\partial y} u(x,y)\right) - y\left(\frac{\partial}{\partial x} u(x,y)\right) = 0$$

```
> pdsolve(eq,u(x,y));
```

$$u(x,y) = \_F1(x^2+y^2)$$

_F1 is an arbitrary function. In order to get some particular solution we have to consider particular cases for the function **_F1** and to compute $\_F1(x^2+y^2)$. For example if _F1(t)=t^2 then $u(x,y) = (x^2+y^2)^2$ is a particular solution of this equation. We can check if some given function is a solution of some given partial differential equation using **pdetest** command.

```
> u1:=(x,y)->(x^2+y^2)^2;
```

$$u1 := (x, y) \to (x^2 + y^2)^2$$

```
> pdetest(u(x,y)=u1(x,y),eq);
```

$$0$$

If **pdetest** returns 0 then the specified function is a solution, if **pdetest** returns something different from 0 then it is not. For example $\mathrm{u2}(x, y) = (x^2 + y^2)^2 + x$ is not a solution of this equation

```
> u2:=(x,y)->(x^2+y^2)^2+x;
```

$$u2 := (x, y) \to (x^2 + y^2)^2 + x$$

```
> pdetest(u(x,y)=u2(x,y),eq);
```

$$-y$$

```
>
```

b) When a solution, but not the most general one, is obtained **pdsolve** expresses the result using the internal **PDESolStruc** function, displayed using **&where**, with the functional form found for the indeterminate function as the first argument. The second argument contains a list with any ODEs found while separating the variables, as well as any arbitrary functions or changes of variables introduced by **pdsolve**. **PDESolStruc** enables the user to see how particular the solution obtained is. In these cases, an explicit result for the indeterminate function, a particular solution, can be obtained from this PDE solution structure by using the **build** command. See the examples below.

```
> eq2:=u(x,y)*diff(u(x,y),y,x) + diff(u(x,y),x)*diff(u(x,y),y) = 1;
```

$$eq2 := \mathrm{u}(x, y) \left(\frac{\partial^2}{\partial y\, \partial x} \mathrm{u}(x, y) \right) + \left(\frac{\partial}{\partial x} \mathrm{u}(x, y) \right) \left(\frac{\partial}{\partial y} \mathrm{u}(x, y) \right) = 1$$

```
> sol:=pdsolve(eq2,u(x,y));
```

$$sol := (\mathrm{u}(x, y) = \_F1(x)\, \_F2(y)) \ \&where$$

$$\left[\left\{ \frac{d}{dx} \_F1(x) = \frac{\_c_1}{\_F1(x)}, \frac{d}{dy} \_F2(y) = \frac{1}{2} \frac{1}{\_F2(y)\, \_c_1} \right\} \right]$$

```
> u1:=build(sol);
```

$$u1 := \mathrm{u}(x, y) = \frac{\sqrt{2\, \_c_1\, x + \_C1}\, \sqrt{\_c_1\, y + \_C2\, \_c_1^{\,2}}}{\_c_1}$$

The obtained answer is not a function, so we have to use **unapply** command in order to transform the expression in function

```
> u_s:=unapply(rhs(u1),x,y,_c[1],_C1,_C2);
```

$$u\_s := (x, y, \_c\_1, \_C1, \_C2) \to \frac{\sqrt{2\, \_c\_1\, x + \_C1}\, \sqrt{\_c\_1\, y + \_C2\, \_c\_1^{2}}}{\_c\_1}$$

Now, we can obtain some particular solution given some values to the parameters _c[1], _C1, _C2

```
> u_s(x,y,1,1,0);
```

$$\sqrt{2\, x + 1}\, \sqrt{y}$$

A particular result can be obtained by separating the variables by product.

```
> sol2:=pdsolve(eq2,HINT=f(x)*g(y));
```

$$sol2 := (u(x,y) = f(x)\, g(y)) \;\&\text{where}\left[\left\{\frac{d}{dx} f(x) = \frac{\_c_1}{f(x)}, \frac{d}{dy} g(y) = \frac{1}{2}\frac{1}{g(y)\,\_c_1}\right\}\right]$$

```
> u2:=build(sol2);
```

$$u2 := u(x,y) = \frac{\sqrt{2\,\_c_1\,x + \_C1}\,\sqrt{\_c_1\,y + \_C2\,\_c_1^{\,2}}}{\_c_1}$$

Use the **HINT** option to obtain a general solution, getting some inspiration from the solution above.

```
> sol3:=pdsolve(eq2,HINT=P(x,y)^(1/2));
```

$$sol3 := u(x,y) = \sqrt{\_F2(x) + \_F1(y) + 2\,x\,y}$$

\>

\>

c) When **pdsolve** fails, it returns **NULL**.