**pdsolve** - Find analytical solutions for partial differential equations (PDEs)

### Calling Sequences:

- `pdsolve(PDE)`
- `pdsolve(PDE,f,HINT=...,INTEGRATE,build)`

### Parameters:

- **PDE** - a partial differential equation
- **f** - the (name of the) indeterminate function, when necessary
- **HINT=...** - (optional) to input user's hints
- **INTEGRATE** - (optional) indicates the automatic integration of the set of ODEs found when a PDE is solved through separation of variables
- **build** - (optional), to try to build an explicit expression for the "indeterminate function", no matter what the generality of the solution found

- Optional parameters can be given alone or in conjunction and in any order.

### Description:

- Given a PDE, `pdsolve`'s main goal is to find an analytical solution for it. There are no restrictions as to the type, differential order or number of independent variables of the PDEs `pdsolve` can try to solve.

- The `pdsolve` command currently solves a single PDE with respect to a single indeterminate function. Future versions should include a first approach to solving systems of PDEs.

- This command recognizes a certain number of PDE families which can be tackled using standard methods. When the given PDE belongs to an unrecognized family, `pdsolve` uses a heuristic algorithm that looks for a separation of variables by taking into account the specific structure of the PDE.

- The strategy `pdsolve` uses is to look for the most general solution to the given PDE, or, in the worst case, for a complete separation of variables. Thus, when successful, the command returns one of the following:
  1. a general solution;
  2. a quasi-general solution (a solution containing arbitrary functions, but not in sufficient number or not having enough variables in their functionality to constitute a general solution);
  3. a set of non-coupled ODEs with all the variables separated, or a complete solution obtained after
integrating this set (when the option INTEGRATE is indicated).

• When an incomplete separation of variables is reached, the program re-calls itself (now with a smaller problem), possibly using different methods of solution on each round. If the smaller problem cannot be solved, the incomplete separation of variables is returned (together with a warning message).

• The results of \texttt{pdsolve} are returned, by default, in one of three forms:
  a) When the general solution to the PDE is obtained, the program returns an explicit result for the indeterminate function. For example:

\begin{verbatim}
> PDE := x*diff(f(x,y),y) - y*diff(f(x,y),x) = 0;

PDE := x \left( \frac{\partial}{\partial y} f(x, y) \right) - y \left( \frac{\partial}{\partial x} f(x, y) \right) = 0

> pdsolve(PDE);    # _F1 is an arbitrary function

f(x, y) = _F1(x^2 + y^2)
\end{verbatim}

b) When a solution is obtained but it is not the most general one, \texttt{pdsolve} expresses the result using the \texttt{PDESolStruc} (new) function, displayed using "&where", with the functional form found for the indeterminate function as first argument. The second argument will contain a list with any ODEs found while separating the variables, as well as any arbitrary functions or changes of variables introduced by \texttt{pdsolve}. The purpose of \texttt{PDESolStruc} is to allow the user to see how particular the solution obtained is. In these cases, an explicit result for the indeterminate function (actually a particular solution) can be obtained from this PDE solution structure by using the \texttt{build} command (see the examples below). 

c) When \texttt{pdsolve} fails, it returns NULL.

• You can change the default ways of expressing results by assigning the values 0, 1, or 2 to the environment variable \texttt{_EnvBuildPdsolve}. By default, \texttt{_EnvBuildPdsolve} is assigned to 1. When \texttt{_EnvBuildPdsolve} is set to 0, \texttt{pdsolve} always returns the structure of the solution using \texttt{PDESolStruc}; this structure can be transformed into an explicit solution using the \texttt{build} command. When \texttt{_EnvBuildPdsolve} is set to 2, \texttt{pdsolve} will always try to build an explicit solution, no matter what the generality of the result obtained.

THE ARGUMENTS

• When the given PDE contains derivatives of more than one function, the function which should be considered the indeterminate function must be given as extra argument; either its name or the function itself will work.

• Three other optional arguments are allowed:
  1. You can give the extra argument "build" to \texttt{pdsolve}, meaning to directly build an explicit result, no
matter what the generality of the solution obtained;

2. You can request the automatic integration of the system of ODEs found by \texttt{pdsolve} when separating the variables (option \texttt{INTEGRATE});

3. You can give a hint (option \texttt{HINT=...}) indicating a method of solution or a form for the indeterminate function. When given, the hint is taken by \texttt{pdsolve} as the departure point in looking for the solution. This option is remarkably useful when a result obtained by \texttt{pdsolve} is not the most general one, and permits a detailed study of the possible solutions for a given PDE.

- The arguments which can be used at present with the \texttt{HINT=...} option are:
  a) \texttt{HINT=`+`}: forces \texttt{pdsolve} to begin by looking for a solution trying to separate the variables by sum;
  b) \texttt{HINT=`*`}: forces \texttt{pdsolve} to begin by looking for a solution trying to separate the variables by product;
  c) \texttt{HINT=...(any algebraic expression)...}: forces \texttt{pdsolve} to begin by looking for a solution trying to simplify the PDE or to separate the variables taking the indeterminate function as equal to the indicated algebraic expression. It is possible to give \texttt{pdsolve} a "functional" hint, say \texttt{HINT=f1(x)/f2(y)\^f2(z)} (where the indeterminate function is \( f(x,y,z) \)); i.e., suggesting only the functional form that the indeterminate function should have. This option also permits the introduction of more than one indeterminate function of more than one variable each, say \texttt{HINT=...f1(x,y)...f2(y,z)...etc.};
  d) \texttt{HINT=strip}: for first order PDEs only, forces \texttt{pdsolve} to look for a solution by trying to solve the associated characteristic strip. In this case, \texttt{pdsolve} tries to find the differential invariants associated with the given PDE and arrive at its general solution. This is usually possible when the characteristic strip does not require the extension of the configuration space by introducing the partial derivatives of the indeterminate function as variables. Otherwise, the integrated characteristic strip is returned in terms of a parameter \texttt{_s}.

CONVENTIONS

- \texttt{pdsolve} usually introduces new functions to express the solution for the indeterminate function, as in, for example:

  \[
  f(x,y,z)= _F1(x) + _F2(y) + _F3(z)
  \]

  All functions introduced by \texttt{pdsolve} beginning with \texttt{_F} and followed by a number are assumed to be arbitrary, sufficiently differentiable functions of their arguments.

- Any arbitrary constants introduced while separating the variables are represented as \texttt{_c[1]}, \texttt{_c[2]},..., and are global.

REFERENCES

- E.S. Cheb-Terrab and K. von B"ulow, "A Computational Approach for the Analytical Solving of
Examples:

1. General solution of a first order PDE

```maple
> with(PDEtools);

[PDEplot, build, casesplit, charstrip, dchange, dcoeffs, declare, difforder, dsolve, mapde,
separability, splitstrip, splitsys, undeclare]

> PDE := x*diff(f(x,y),y)-diff(f(x,y),x)=f(x,y)^2*g(x)/h(y);

    \[
    PDE := x \left( \frac{\partial}{\partial y} f(x,y) \right) - \left( \frac{\partial}{\partial x} f(x,y) \right) = \frac{f(x,y)^2 g(x)}{h(y)}
    \]

> ans := pdsolve(PDE);

    \[
    ans := \frac{1}{\int_{x}^{x} \frac{g(-a)}{h\left(-\frac{1}{2}a^2 + y + \frac{1}{2}x^2\right)} d_{-a} + \text{F1}\left(y + \frac{1}{2}x^2\right)}
    \]

For the integral above, see ?intat. Results can be tested using pdetest:

```maple
> pdetest(ans,PDE);

```

0
```

2. Laplace equation in spherical coordinates

```maple
> PDE := Diff(r^2*diff(F(r,t,p),r),r) + 1/sin(t)*Diff(sin(t)*diff(F(r,t,p),t),t) + 1/sin(t)^2*diff(diff(F(r,t,p),p),p) = 0;

    \[
    PDE := r^2 \left( \frac{\partial}{\partial r} \left( \frac{\partial}{\partial r} F(r,t,p) \right) \right) + \frac{\partial}{\partial t} \sin(t) \left( \frac{\partial}{\partial t} F(r,t,p) \right) + \frac{\partial^2}{\partial p^2} F(r,t,p) = 0
    \]

The structure of the solution expressed using PDESolStruc (displayed using &where)

```maple
> ans := pdsolve(PDE);
```

To build an explicit expression for \( F(r,t,p) \) try build(ans).

3. Hamilton-Jacobi type PDE (see for instance Landau's book on Mechanics)

\[
PDE := -\text{diff}(S(t,xi,eta,phi),t) = 1/2*\text{diff}(S(t,xi,eta,phi),xi)^2*(xi^2-1)/sigma^2/m/(xi^2-eta^2) + 1/2*\text{diff}(S(t,xi,eta,phi),eta)^2*(1-eta^2)/m/sigma^2/(xi^2-eta^2) + 1/2*\text{diff}(S(t,xi,eta,phi),phi)^2/m/sigma^2/(xi^2-1)/(1-eta^2) + (a(xi)+b(eta))/(xi^2-eta^2);
\]

The structure of a complete solution obtained through separation of variables by sum

\[
\text{ans} := \text{pdsolve}(PDE);
\]

\[
\text{ans} := (S(t,xi,eta,phi) = _F1(t) + _F2(xi) + _F3(eta) + _F4(phi)) \text{where}
\]

\[
\frac{\partial^2}{\partial r^2} F_1(r) = \frac{F_1(r)}{r^2} - 2 \frac{\partial}{\partial r} F_1(r), \\
\frac{\partial^2}{\partial t^2} F_2(t) = -F_2(t) c_1 - \left( \frac{\partial}{\partial t} F_2(t) \right) \cos(t) - \frac{c_3 F_2(t)}{\sin(t)}.
\]
\[
\left( \frac{\partial}{\partial \xi} F_2(\xi) \right)^2 = -2 \frac{\xi^2 \sigma m_{-c_1}}{\xi^2 - 1} - \frac{-c_4}{2} \frac{\xi^2}{\xi^2 - 1} - 2 \frac{\sigma^2 m_{-c_3}}{\xi^2 - 1} - 2 \frac{a(\xi) \sigma^2 m_{-c_1}}{\xi^2 - 1},
\]

\[
\frac{\partial}{\partial \phi} F_4(\phi) = -c_4,
\]

\[
\left( \frac{\partial}{\partial \eta} F_3(\eta) \right)^2 = -2 \frac{\eta^2 \sigma m_{-c_1}}{\eta^2 - 1} - \frac{-c_4}{2} \frac{\eta^2}{(\eta + 1)(-1 + \eta)(\eta^2 - 1)} - 2 \frac{\sigma^2 m_{-c_3}}{\eta^2 - 1} + 2 \frac{b(\eta) \sigma^2 m_{-c_1}}{\eta^2 - 1}
\]

Testing this PDE solution structure returned by pdsolve:

\[
> \text{pdetest}(\text{ans}, \text{PDE});
\]

0

To build an explicit expression for \( S(t, x, y, \phi) \) try \text{build}(\text{ans}).

4. A second-order PDE and the HINT=... option

\[
> \text{PDE} := S(x, y) \cdot \text{diff}(S(x, y), y, x) + \text{diff}(S(x, y), x) \cdot \text{diff}(S(x, y), y) = 1;
\]

\[
PDE := S(x, y) \left( \frac{\partial^2}{\partial y \partial x} S(x, y) \right) + \left( \frac{\partial}{\partial x} S(x, y) \right) \left( \frac{\partial}{\partial y} S(x, y) \right) = 1
\]

A particular result obtained separating the variables by product

\[
> \text{struc} := \text{pdsolve}(\text{PDE}, \text{HINT}=f(x) \cdot g(y));
\]

\[
\text{struc} := (S(x, y) = f(x) \cdot g(y)) \& \text{where} \begin{bmatrix} \frac{\partial}{\partial x} f(x) = \frac{-c_1}{f(x)}; \frac{\partial}{\partial y} g(y) = \frac{1}{2 \cdot g(y) \cdot c_1} \end{bmatrix}
\]

An explicit expression for \( S(x, y) \)

\[
> \text{build}(	ext{struc});
\]

\[
S(x, y) = \sqrt{2 \cdot c_1 \cdot x + C1} \sqrt{c_1 \cdot y + C2 \cdot c_1^2}
\]

A general solution using the HINT option, getting some inspiration from the solution above
pdsolve(PDE, HINT=P(x, y)^(1/2));

\[ S(x, y) = \sqrt{-F2(x) + F1(y) + 2xy} \]

General solutions involve N arbitrary functions \_F of K-1 variables, where N is the differential order and K the number of independent variables of the PDE.

5. Non-linear first order PDE solved using the characteristic strip method

\[ PDE := \text{diff}(f(x, y, z), x) + \text{diff}(f(x, y, z), y)^2 = f(x, y, z)+z; \]

\[ PDE := \left( \frac{\partial}{\partial x} f(x, y, z) + \frac{\partial}{\partial y} f(x, y, z) \right)^2 = f(x, y, z) + z \]

\[ \text{pdsolve}(PDE, \text{HINT} = \text{strip}); \]

\[ \left( \frac{\partial}{\partial x} f(x, y, z) + \frac{\partial}{\partial y} f(x, y, z) \right)^2 - f(x, y, z) - z = 0 \]&where \{ \{ _p2(_s) = _C4 e^{-s}, _p1(_s) = _C3 e^{-s}, z(_s) = _C5, y(_s) = 2 _C4 e^{-s} + _C2, x(_s) = _s + _C6, f(_s) = _C3 e^{-s} - _C5 + e^{2 _s} _C1 \} \}, \&and\{ \{ _p2 = \frac{\partial}{\partial y} f(x, y, z), _p1 = \frac{\partial}{\partial x} f(x, y, z) \} \]