

# ODESOLVE

Malcolm A.H. MacCallum  
Queen Mary and Westfield College, London  
Email: mm@maths.qmw.ac.uk

Other contributors: Francis Wright, Alan Barnes

The ODESOLVE package is a solver for ordinary differential equations. At the present time it has very limited capabilities,

1. it can handle only a single scalar equation presented as an algebraic expression or equation, and
2. it can solve only first-order equations of simple types, linear equations with constant coefficients and Euler equations.

These solvable types are exactly those for which Lie symmetry techniques give no useful information.

## 1 Use

The only top-level function the user should normally invoke is:

```
ODESOLVE(EXPRN:expression, equation,  
          VAR1:variable,  
          VAR2:variable):list-algebraic
```

ODESOLVE returns a list containing an equation (like solve):

**EXPRN** is a single scalar expression such that  $\text{EXPRN} = 0$  is the ordinary differential equation (ODE for short) to be solved, or is an equivalent equation.

**VAR1** is the name of the dependent variable.

**VAR2** is the name of the independent variable

(For simplicity these will be called  $y$  and  $x$  in the sequel) The returned value is a list containing the equation giving the general solution of the ODE (for simultaneous equations this will be a list of equations eventually). It will contain occurrences of the operator **ARBCONST** for the arbitrary constants in the general solution. The arguments of **ARBCONST** should be new, as with **ARBINT** etc. in **SOLVE**. A counter **!!ARBCONST** is used to arrange this (similar to the way **ARBINT** is implemented).

Some other top-level functions may be of use elsewhere, especially:

**SORTOUTODE**(EXPRN:*algebraic*, Y:*var*, X:*var*): *expression*

which finds the order and degree of the EXPRN as a differential equation for Y with respect to Y and sets the linearity and highest derivative occurring in reserved variables ODEORDER, ODEDEGREE, ODELINEARITY and HIGHESTDERIV. An expression equivalent to the ODE is returned, or zero if EXPRN (equated to 0) is not an ODE in the given vars.

## 2 Tracing

Some rudimentary tracing is provided and is activated by the switch **TRODE** (analogous to **TRFAC** and **TRINT**)

## 3 Comments

The intention in the long run is to develop a rather general and powerful ordinary differential equation solver incorporating the methods detailed below. At present the program has not been optimized for efficiency and much work remains to be done to convert algebraic mode procedures to more efficient symbolic mode replacements.

No attempt is made to extend the **REDUCE** integrator, although this is in some sense a problem of ODEs. Thus the equation  $\frac{dy}{dx} = g(x)$  will be solved if and only if  $\int g(x)dx$  succeeds.

The available and planned coverage is as follows:

- First-order equations: (first degree unless otherwise stated)
  - Quadrature of  $\frac{df}{dx} = g(x)$
  - Linear equations
  - Separable equations
  - (Algebraically) homogeneous equations
  - Equations reducible to the previous case by linear transformations
  - Exact equations
  - Bernoulli equations

The above are already implemented. Further 1st order cases are not:

- Riccati equations using Schmidt's methods and other special cases
  - Hypotheses on the integrating factor following Char (SYMSAC 81) or Shtokhamer, Glinos and Caviness.
  - Higher degree cases
- Linear equations of higher order
    - Constant coefficients case for driving terms solvable by variation of parameters using the integrator (Choice of method is discussed in the source of module lccode).

The above is already implemented. Further higher order methods are not:

- More complex driving terms via Laplace transforms (?)
  - Variable coefficients: Watanabe (EUROSAM 84) methods including Kovacic's algorithm as extended by Singer
  - Factorization of operators as in Schwarz's ISSAC-89 paper or Berkovich's 1990 book
  - Other methods based on Galois theory (see Ulmer's preprints from Karlsruhe, 1989, 1990 and Singer's 1989 review) or other ways of hunting Liouvillian solutions (see Singer's review in J. Symb. Comp., 1990).
- Non-linear equations of order 2 and higher
    - Lie algebra of point symmetries e.g. using Wolf's CRACK now available in REDUCE
    - Other special ansatze (see Wolf. op. cit), in particular contact transformations for 2nd order cases

- Possibly (?) exploitation of Cartan's methods for equivalence of differential equations.