Downloading and installing Reduce and Dimsym

Reduce

Reduce is a system for doing scalar, vector and matrix algebra. It can be downloaded for free for both Windows and Mac systems from the *Obtaining REDUCE* tab at `http://reduce-algebra.com/`
This website also provides documentation for using Reduce (on the *Documentation* tab) – perhaps the most useful is the *Reduce User's Manual*.

Dimsym

Dimsym is a program purpose written by James Sherring, Geoff Prince and Michael Jerie to determine Lie point symmetries of differential equations. A user can specify an ordinary or partial differential equation (or system of equations) and Dimsym will calculate the determining equations and give the generators of the symmetry group. It will also highlight any conditions on the parameters or arbitrary functions in the differential equation that may allow additional symmetries to be found.

Dimsym can be downloaded for free from
`http://www.latrobe.edu.au/mathematics-and-statistics/research/geometric-and-algebraic-techniques-for-differential-equations/dimsym`
Examples and a User's Manual are also available.

In the `dimsym_working` folder there is a `readme` file that says:

```
Installation
To install dimsym you must
  1.  Extract the source files dimsym23.red and dim2ode.red to an appropriate
directory.  Extract the make file mkds also
  2.  Start Reduce
  3.  Enter at prompt> load mkds;
  4.  This will compile the dimsym package
  5.  To run dimsym enter the following at the reduce prompt>load dimsym23;
```

If the above instructions don't work, try following these (similar) steps:

1. Put the files `dimsym23.red` and `dim2ode.red` (both from the `dimsym23` folder) and `mkds` (from test folder) and the `NonlinearHeat` file into `Reduce\lib\csl`. Ensure that the executable file for Reduce is also in this folder.
2. Start Reduce.
3. At the prompt>, enter `in mkds;`
4. This compiles the dimsym package. It doesn't need to be done every time, just once.

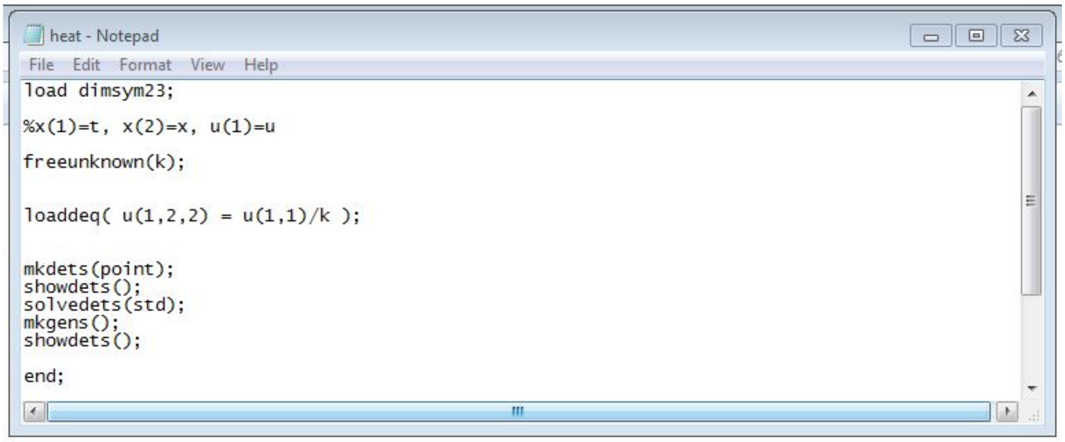
5. To run dimsym, at the reduce prompt> enter `in dimsym23;`. Alternatively, use `in heat;` which already contains this command.

An example file – the linear heat equation

In order to use Dimsym efficiently, it's easiest to enter the differential equation and all the required commands into a text file that can be read by Reduce (note that this input file could have an extension, eg `heat.txt`, but this is not necessary, eg `heat` is fine).

When the following file is read by Reduce (with the command `in heat;`), Dimsym calculates the Lie point symmetry generators for the linear heat equation,

$$u_t = ku_{xx}$$



```
load dimsym23;

%x(1)=t, x(2)=x, u(1)=u

freeunknown(k);

loaddeq( u(1,2,2) = u(1,1)/k );

mkdets(point);
showdets();
solvedets(std);
mkgens();
showdets();

end;
```

The commands are as follows:

`load dimsym23;` – loads the Dimsym package

`%x(1)=t, x(2)=x, u(1)=u` – this line is commented out, but serves to advise the user that the in the vector containing the independent variables, the first one is t, the second is x; in the vector of dependent variables, the first is $u(x,t)$ (in this case, we only have a single pde rather than a system, and so there is only one dependent variable).

`freeunknown(k);` – advises Dimsym that there is an unknown constant, k.

`loaddeq( u(1,2,2) = u(1,1)/k );` – enters the differential equation. The equation must always be rearranged so that the highest order derivative is the subject. The derivatives are written as `u(i,j)`, where the `i` denotes the relevant entry of the u vector (in this case `i`$=1$ represents `u(1)`$= u$, and `j` denotes what we are differentiating with respect to. For example, `u(1,1)`$= u_t$ and `u(1,2,2)`$= u_{xx}$.

`mkdets(point);`

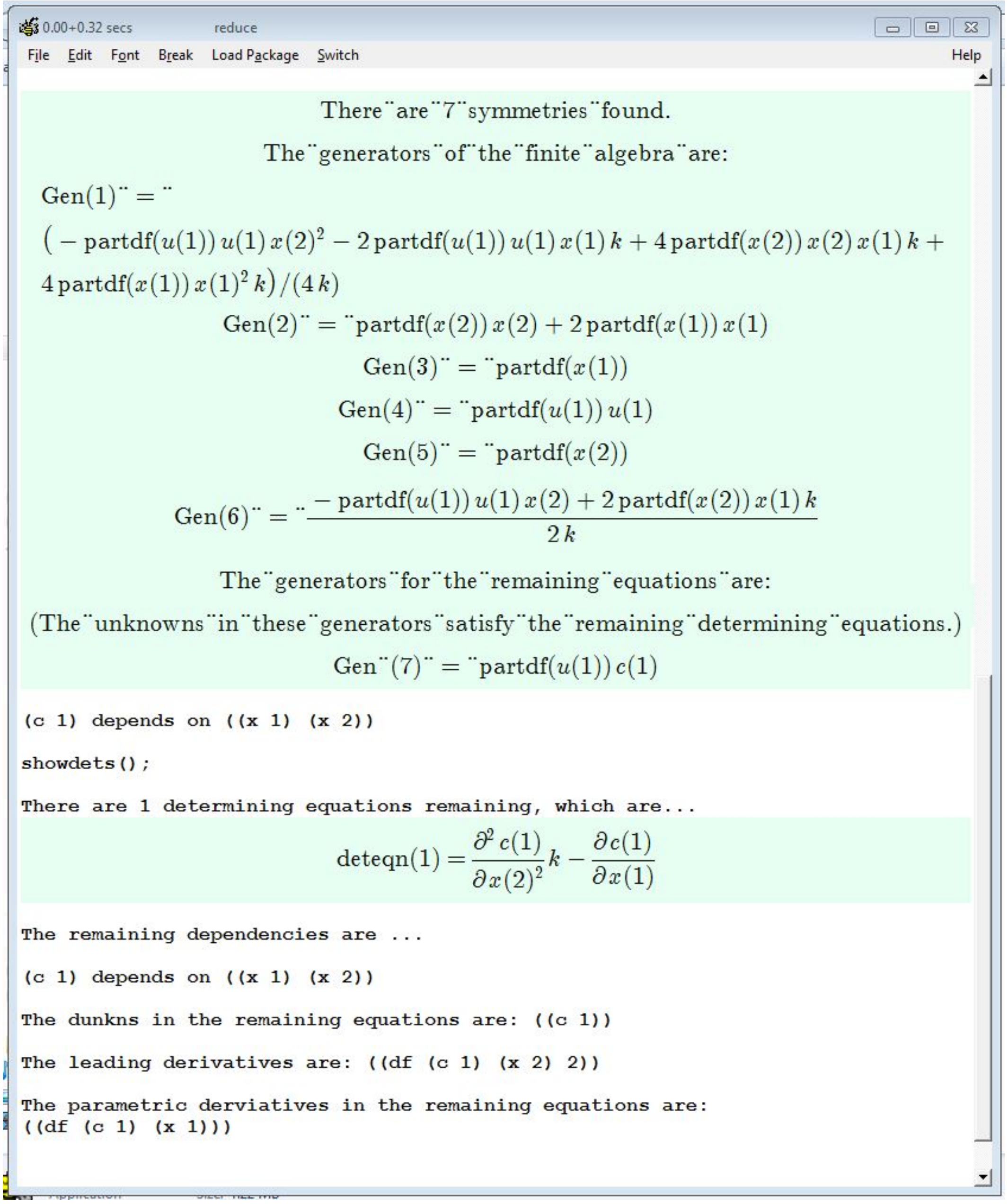
`showdets();`

`solvedets(std);`

`mkgens():`

`showdets();` – ask Dymsym to make and solve the determining equations, then make and show the symmetry generators. More detail can be found in the User's Manual.

The output produced by Dimsym is



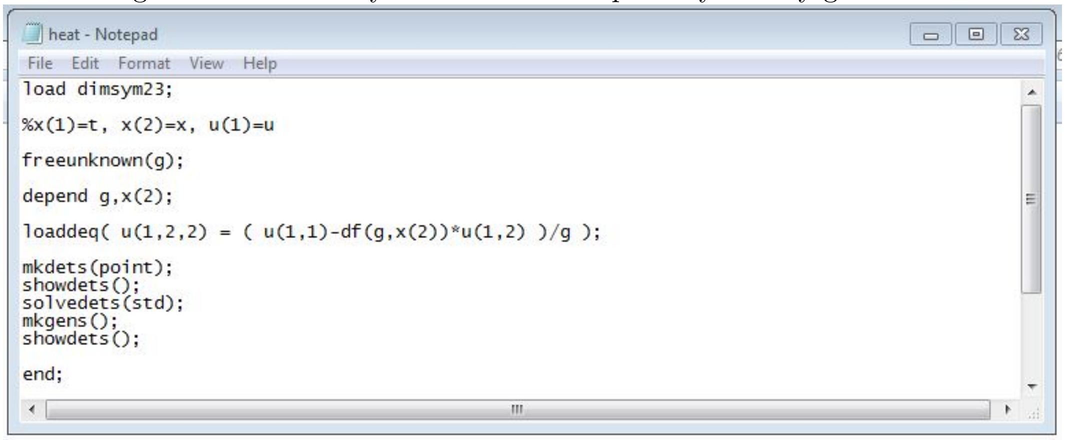
That is, the usual 6 symmetry generators are produced, plus an infinite symmetry.

Before searching for the symmetries of a new equation, Reduce should be exited and reopened – type `bye;` and then open again.

For the nonlinear diffusion equation,

$$u_t = (g(x)u_x)_x$$

the following file will ask dimsym to find the Lie point symmetry generators.



The lines

`freeunknown(g);`

`depend g,x(2);`

advise Dimsym that there is an arbitrary function, g, that depends on x (i.e. `x(2)`). Derivatives of arbitrary functions are written using `df`, eg `df(g,x(2))`$= g_x$. Note that the equation has been rewritten to make the highest order derivative the subject of the equation,

$$u_{xx} = \frac{u_t - g_x u_x}{g}$$