# GDB and OpenFOAM - a quick summary

Fabian Peng Kärrholm

March 11, 2008

`break [b]`

Inserts a breakpoint. This can be done in several ways, most standard way is: `break UEqn.H:12` breaks at line 12 of UEqn.H. For something more advanced, let's say we have the following piece of code:

```
forAll(rho,celli)
{
     rho[celli] = p[celli]/R/T[celli];
}
```

and it crashes for some reason, we want to find out why. T[celli]=0 would be bad, so let's put a breakpoint if that happens. Note that we can not do this until celli has been defined, i.e. in the loop. So first we have to put a breakpoint inside the loop to get us to that position. Once there, specify the new breakpoint

```
b if T[celli]<1e-10
```

if we could continue the code now, it would stop again in the next iteration, so the breakpoint that occurs on the line inside the loop needs to be removed. This is done by `delete 1`, if the line breakpoint is the first breakpoint. Another way of setting a breakpoint is to break for all functions that matches a specific expression, for instance `rbreak adjustPhi`

`call`

Calls a function in OpenFOAM. For instance `Foam::min(3,4)` returns the OpenFOAM function min of two integers. Note that the specific function (depending on type) has to be chosen, and since many functions in OpenFOAM is overloaded, this can be quite difficult.

`checkpoint`

Inserts a checkpoint in the code which you can use to go back to in case the program crashes to re-inspect certain variables.

`file`

Loads a library before it is loaded in the code, can be handy for using certain functions or specifying breakpoints.

`info`

Displays info about current state of affairs, which functions are defined, what variables there are etc. Tip, do not check for all functions, as this list is incredibly long and takes quite a bit of time to generate. You can use expressions to find the function you want, for instance `info functions adjustPhi`

`list`

This command will show you the source code around the line you've stopped at. You can also specify which file you want to see, as well as how many lines you want to see. For instance: `list 40,60` will display lines 40 through 60 of the current source file.

`next [n]`

Steps to the next line in the code, does not go into subfunctions or included files. `n 5` repeats this 5 times.

`print [p]`

Prints a variable, `p U[0]` prints the value of U in cell 0. You can supply

`run`

Runs the program until breakpoints, if no breakpoints exist, the program will continue running until crash or normal exit.

`start`

Starts the program until you reach the main procedure

`step [s]`

Steps one line in the code, into subfunctions etc. Also possible to repeat by `s 5`.

`where`

Shows you where in the source code you are