Imperial College London

Fluidity Training

What to do if your simulation breaks?

Dave Robinson

7th November 2014

Imperial College London

Introduction

This presentation will cover:

- Some reasons your simulation can fail
- Sources of useful data
- What can cause numerical instability?
- Simplifying your problem
- (Occasionally) helpful third party software
- HELP!

Some reasons that your simulation can fail

Roughly speaking, the reason that your simulation can failed will be placed in one of the following categories:

- Bad environment variables, missing/out-of-date dependent software or modules
- Option errors, faulty Python scripts
- Under/over/badly constrained initial and/or boundary conditions
- Numerical instability
- Partitioning, adaptivity, field projection

Sources of useful information

Information that can help you diagnose the failure can be gathered from:

- log file ("`-l -v3`")
- error file (including backtrace)
- stat file
- pbs output (if running on a HPC)
- "matrixdump" and "core" files
- vtus

It is also possible to re-run your simulation with extra diagnostic fields, increase your dump-rate, and re-run the simulation with a Fluidity binary compiled with debugging enabled.

"My simulation broke in 10 seconds flat!"

If your simulation broke very very quickly then it was more than likely due to one of the following reasons:

- Bad environment variables, missing/out-of-date dependent software or modules
- Option errors, faulty Python scripts

A common mistake, for instance, is having a mis-directed PYTHONPATH environment variable. You can check this by typing "`echo $PYTHONPATH`" into your terminal window (the list of paths should include `[Fluidity directory]/python`)

If you're suspicious that your Fluidity build isn't working correctly, navigate to the Fluidity directory and type "`make unittest`".

For an option error, then hopefully the error message should give useful advice on what to correct, but if all else fails, backing-up your `.flml` file and progressively removing fields can help. Also, using the `diff` and `meld` commands to compare against working `.flml` files.

Imperial College London

What can cause numerical instability?

Thankfully, for most causes of failure, the error messages given by Fluidity are fairly self-explanatory. But, unfortunately, the cause of numerical instability is often a little more tricky to pin-point. The symptom described within an error message may not directly relate the underlying cause of failure.

For this reason, the best advice is to sanity check the various factors that can cause numerical instability:

- Timestep, Courant Number, and temporal discretisation
- Spatial discretisation and element pair
- Mesh quality, Condition Number
- Mesh resolution, interpolation error, Grid Reynolds Number
- Field stabilisation, upwinding/slope-limiting
- Strong/weak boundary conditions
- Preconditioner, iterative solver, convergence criteria

Simplifying your simulation

Once you have extracted any useful data from a broken simulation it is important that you make any further progress from a simpler simulation that *did* solve successfully. If no such simulation exists, then you need to create one!

There are several ways in which you can simplify your simulation:

- Reduce the number of spatial dimensions
- Simplify the geometry
- Smooth BCs with discontinuities in space or time
- If you have a viscosity field, then set that to a value that gives you a Reynolds Number $\simeq 1000$
- Remove any turbulence models and prescribe viscosity (as above)
- Progressively remove prognostic fields
- Remove adaptivity, lower the number of partitions

Then, add complexity back in ONE STEP AT A TIME

Monitoring the progress of a simulation

If you are suspicious that your simulation may crash and you want to monitor its progress, then the following commands can be useful:

`statplot *.stat` (press "`r`" to refresh)

```
tail -f fluidity.log-0
```

```
grep ``n/o iterations'' fluidity.log-0
```

```
grep reason fluidity.log-0
```

Note that for the last three to work you need to be logging with a verbosity of two or more, i.e. `./fluidity -v2 -l my.flml`

(Occasionally) helpful third-party software

If you believe there might be a bug in the code then your first step should be to contact the development team and submit a bug report.

The following third-party software can then be useful for slowly stepping through the simulation solve to determine the root cause:

- Valgrind (checking for memory issues)
- GDB (a free Open Source de-bugger)
- DDT (a commercial alternative)

Similarly "vtune" can be useful in determining which lines of code are taking up the most time in the solve.

HELP!!!

If you've tried and failed to fix your problem, then there is a troubleshooting section to the Manual that's worth consulting.

But finally, Fluidity has a very eager and responsive development community. We can be contacted either through the mailing list (`mailman.ic.ac.uk/mailman/listinfo/fluidity`) or through IRC (`# amcg`).

It would be helpful for you to include information about:

- What simulation are you trying to run?
- What momentum equation are you solving?
- What discretisation are you using?
- What similar simulations do you have that *did* work?

Questions?