Bubble Dynamics Code

Requirements:

- A working Fortran compiler. The `makefile` assumes you will be using `gfortran`. If this is not the case, you will need to edit it to suit your needs.
- LAPACK and BLAS libraries that can be linked to the compiler. You will probably be okay with the following

```
sudo apt install liblapack3 liblapack-dev liblapacke-dev
sudo apt install libopenblas-base libopenblas-dev
```

at least if you are on recent versions of Ubuntu. Earlier versions may need `apt-get` and other Linux flavours will require their own package-manager installs.

First run:

If this is the first time you are running the code, you will need to make the main file executable.

- Run
  `chmod +x runcode.sh`

Future runs:

Once the `runcode.sh` file is exectuable, the `pressure_GAUSSIAN.f90` code can be made, compiled and run with the command:

`./runcode.sh`

The output of this run will create a folder called `data`, inside which a date-stamped folder will be created containing the data files.

Parameters:

Most of the code's parameters are set between lines 15-20 and lines 247-310. These may be cross-referenced with the parameters in the main paper `paper.pdf` in the directory. Some of the more *numerical* parameters are set between lines 136-143.

Plotting graphs from the data:

The current code setup populates the following data files:

```
bub_surf_before.dat
bub_surf_before2.dat
bub_surf_before2.dat
bub_surf_before3.dat
centroid_eqrad.dat
ENERGY.dat
ENERGYTERMS.dat
field_variables.dat
jet_vel.dat
pressurepulses.dat
rad_vs_time.dat
volume.dat
```

of which, most should be fairly self-explanatory in terms of their contents from the filenames, the code comments and the figures in the paper.

The files tend to follow a convention of

```
xdata ydata1 ... ydataN
```

with each of the ydata values explained in the code.

This allows for plotting of the data to be reasonably straightforward and consistent.

Plotting in `gnuplot`

In `gnuplot`, you can use the command

`plot 'filename.dat' using 1:? w l`

where the column number required replaces the `?` in the command. For example, to plot column 2 against the `xdata`, use the command

`plot 'filename.dat' using 1:2 w l`

Further details about plotting from data files in `gnuplot` are avaiable here.

Plotting in `MATLAB`

To import the file into `MATLAB`, you can use something analagous to

```
A = dlmread('filename.dat');
```

which will store the data in `MATLAB` as an array. This can then be plotted in the usual way as

```
plot(A(:,1),A(:,?));
```

where the column number required replaces the `?` in the command. For example, to plot column 2 against the `xdata`, use the command

```
plot(A(:,1),A(:,2));
```

Contact

Questions about running the code and analysing the output and can be directed to Scott Morgan - smorgan@bridgend.ac.uk - at any time.