Convection in Nektar++ for HPC

Ed Threlfall, 1 March 2022
ed.threlfall@ukaea.uk

These Nektar++ session files represent a model consisting of a fluid-filled tank with the rough proportions of a cereal packet. Vertical natural convection is triggered by maintaining a constant temperature difference across the space between the two largest faces. This geometry represents experimental apparatus being used at the University of Leeds by a current student of Wayne (Arter) and designed to reproduce and go beyond classic experiments by Elder [1].

Two examples are provided - a 3D one, with a large mesh containing $189,000$ cuboidal finite elements (this has $189,000 \times (p+1)^3$ degrees of freedom, where p is the order of the finite elements and takes the values $2, 3, 4, ...$ in Nektar++; this means a minimum of 1.5M dofs in the simulation), and a 2D 'toy' mesh (3375 quadrilateral finite elements - the relevant cross-section of the 3D case) which can be used for familiarization (suitable for running on single PC). In both cases, there is a file with the solver parameters (e.g. `convection_2d.xml`) and another file containing the relevant mesh (e.g. `convection_2d_mesh.xml`). The meshes can be uncompressed using the command

```
nekmesh convection_2d_mesh.xml convection_2d_mesh.xml:xml:uncompress
```

and can be converted into ParaView format using

```
fieldconvert convection_2d.xml convection_2d_mesh.xml convection_2d_mesh.vtu.
```

Note there is a small thing to do before running the files - uncompress the two 'mesh' files and delete the following lines at the end of the files

```
<EXPANSIONS>
<E COMPOSITE="C[0]" NUMMODES="2" FIELDS="u" TYPE="GLL_LAGRANGE_SEM" />
</EXPANSIONS>
```

(This is because I was not able to compress the mesh files without including these lines, which are actually redundant.)

What to do in order to run the session files - obtain and build the latest version of *Nektar++*. The session files are used with the incompressible Navier-Stokes solver and are executed with e.g. `mpiexec -np 8 incnavierstokessolver convection_2d.xml convection_2d_mesh.xml`. Once run, the output `.fld` files can be processed into ParaView format with the command

```
fieldconvert convection_2d.xml convection_2d_mesh.xml convection_2d.fld convection_2d.vtu.
```

There is one adjustable physical parameter - the dimensionless Rayleigh number (`Ra` in the session file) which is proportional to the applied temperature difference. The dimensionless Prandtl number (`Pr` in the session file) is fixed by the choice of fluid in the tank and I have given this the value of 7.0 which is appropriate for water. For smaller `Ra` (e.g. c.10^2) the output is dominated by conductive heat transfer. For larger values c.10^4, a laminar convective cell forms. The amount of heat transfer is quantified by the mean of the integrals of the heat flux ($= -\nabla T$) on the heated / cooled boundaries, which is called the Nusselt number, and is normalized, to unit area, to give a value of unity in the conducting (small `Ra`) limit. The most interesting behaviour occurs for `Ra` larger than about 10^6 where there is not a stable steady-state solution but rather a transition to a turbulent state triggered by wall-wave instabilities which ascend the hot wall and descend the cold. These features are common to 2D and 3D, but in the 3D case there is additional instability to fluctuations that disrupt the symmetry along the axis newly-present in the 3D case (i.e. the symmetry direction for the 2D case). Plots of the behaviour of the 2D case are included (Fig.).

One possible pitfall is that the explicit part of the time-stepping means that there is a maximum time
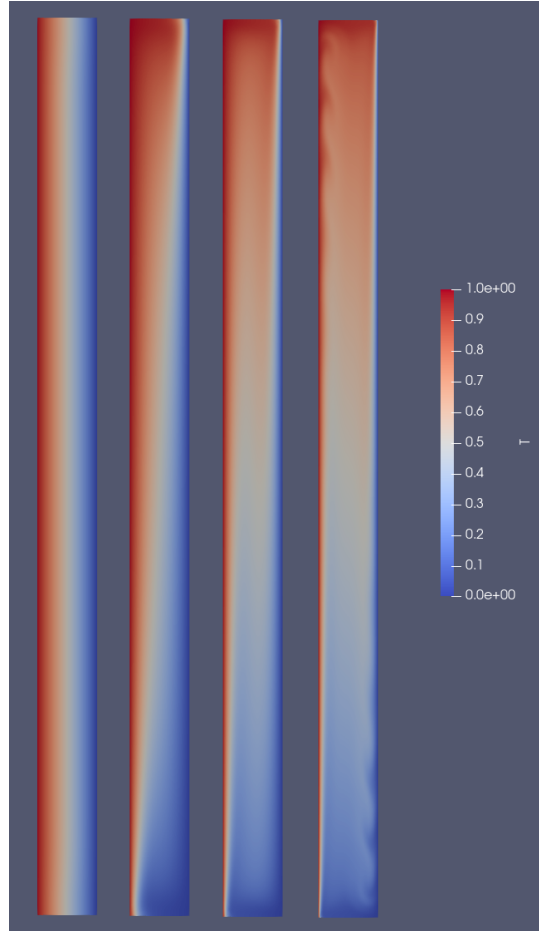
Figure 1: Steady-state temperature profiles for Rayleigh numbers (L-R) 10^2, 10^4, 10^5, 10^6 for the 2D case. Stable time steps used were 10^{-2}, 2×10^{-4}, 10^{-5}, 2×10^{-6} respectively. Simulations were run for durations $1, 1, 1, 0.1$ respectively starting from initial data from the finish of the next-lowest Rayleigh number case. Note wall-wave instability visible in the 10^6 Rayleigh number case.

step size above which the simulation is unstable. If this happens, the solver will crash, giving an error message that NaN has been encountered or max number of iterations is exceeded. Experience in how large a timestep is allowed can be gained by running the 2D simulation. Note the instability is related to the maximum local fluid speed and hence is worse as `Ra` is increased, demanding a smaller time step size. Note also that it is wise to move up in `Ra` e.g. in decades, and using the flow field from the previous `Ra` value simulation as initial data - this avoids a violent perturbation to the system at startup, which is prone to having a very small time step size.

Increasing `Ra` means that the simulation has finer features (e.g. thinner boundary layers) and needs more accuracy. Increasing the precision is done by increasing the integer spectral order p which is represented by the parameter `NUMMODES` in the session file. Note `NUMMODES` is equivalent to $p+1$. Note increasing p increases the number of degrees of freedom and hence the computational demand imposed by the simulation. The 2D examples illustrated in the figures used $p = 3$.

Outputs - Nektar++ *filters* can be used to produce simulation output as time series (uncomment lines relating to the desired filter in `convection_2d.xml` to use). In particular, the Nusselt number can be calculated using the built-in `AeroForces` filter if a small modification is made to one file (`FilterAeroForces.cpp`) in the code before compilation. A copy of the modified file with modifications for 2D and 3D (either-or, must alter file before compilation to choose which) meshes is included (see code marked TRIALCODE in the file, plus attendant comments); if using this, a good test is to ensure a Nusselt number of approx. unity can be obtained from the simulation in the small Rayleigh number limit (you will need to normalize to the

appropriate area). Note the relevant number is the `F1-press` column of the generated output file. Other filters can calculate fields at a point if desired, se e.g. `HistoryPoints`.

References

[1] J.W. Elder, *Turbulent free convection in a vertical slot*, J. Fluid Mech. (1965) vol.23 part 1 pp.99-111.