School of Mathematics and Statistics
MAST30028 Numerical Methods & Scientific Computing
Week 12

**Drag and drop the folder** `Week12` **from** `L: \MAST30028` **to** `C:\...\MATLAB` **and include it in the path.** Now MATLAB knows how to find the files in `Week11`.

In these exercises are you will see how to produce efficient code. You will also learn how to use Matlab to create movies to present your results.

You should start Exercise 2 after an hour if you have not already.

## Exercise Set 1: Making Movies

The script `MovieExample.m` (in the Week 12 folder) produces the movie that you were shown in lectures.

Download and run the script. Experiment with the various settings to do with the command `VideoWriter`

Delete the file after watching the movie as the file is rather large, this is because it is an uncompressed movie. If you want to make the movies smaller you can use a codec such as DivX to compress the video file once it has been created or experiment with the `VideoWriter` options.

## Exercise Set 2: Speeding up your code

Consider the PDE

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad 0 < x < 1, \quad 0 < t < 0.2.$$

with boundary conditions $u(0,t) = u(1,t) = 0$ and initial condition $u(x,0) = \sin(\pi x)$. This is known as the unsteady heat equation. The exact solution to this is given by $u(x,t) = \sin(\pi x)\exp(-\pi^2 t)$

The function `HeatEquation()` in the Week 12 folder, solves this system using a forward Euler scheme for the temporal derivative and a second order finite difference scheme for the spatial derivative. For more details of the solution method see "Numerical Solution of Partial Differential Equations" by K.W. Morton & D.F. Mayers or the subject MAST90026.

This code is currently very inefficient and can be improved. Using the techniques from lectures and the "Writing Fast Matlab Code" guide (located on LMS) speed up the code.

The speed of the code can be assessed by running

```
tic
Error = HeatEquation(100)
toc
```

which has output

```
Error =
         6.6559e-05
Elapsed time is 3.550563 seconds.
```

where `Error` is the maximum error of the numerical solution compared to the actual solution (in both time and space). Currently running with $N = 1000$ intervals yields an error of $6.6568e-07$ but takes about 10 hours, this should be possible in under 30 seconds by improving the code. There is a driver script `TimedSolveHeatEquation.m` which will help with this.

Remember that in Matlab you can use the command "control-c" to stop a process running.