ACM40080 Advanced Computational Science – Assignment 2

miguel.bustamante@ucd.ie

(Dated: 25.03.2014)

Instructions

This is Assignment 2 of the Module, consisting of 1 question. This assignment is worth approximately 20% of the module's final mark.

You will need to complete two parts each carrying equal marks:

- A MATLAB numerical code to solve the problem. Minimum requirements: The MATLAB code must be original and it must contain clear comments in every section. Of course, once you have created a program to solve a problem, you can use it as part of another program to solve a different problem. Figures created with MATLAB must be properly labelled (time axis, etc.) and the plot region must coincide with the region of interest.

- A report describing the numerical and analytical methods used, your results, figures and conclusions. Minimum requirements: The report must be clear and explain in detail the numerical methods used. Figures must have proper captions. Every quantity appearing in the report must be properly defined.

Please **write your name and student number** in the first page of your answers.

The answers must be sent by e-mail to the Lecturer before Friday 25th April, 2014 at 23:59 hrs. Please include your working *m*-files (not as text) and zip the whole submission (using gzip or Peazip programs, for example). Late assignments are not accepted.

The Diffusion Equation

We will numerically solve the diffusion equation

$$\frac{\partial u}{\partial t} = \nabla \cdot (D(x) \nabla u) \,. \tag{1}$$

Here $D(x)$ is the diffusion coefficient, and ∇ is the vector derivative in two dimensions. In this assignment, we will take the diffusion coefficient to be constant, $D = 1$.

Question – The 2D Diffusion Equation

2.1. Apply the Alternating Direction Implicit (ADI) method to solve the diffusion equation in 2D on a grid $-1 \leq x \leq 1$, $-1 \leq y \leq 1$, with Dirichlet boundary conditions and an initial condition $U_0(x, y) = (1 - x^2)(1 - y^4)$. Use values of τ and h using the ideas from the *hint* in 1.3., part (ii). Make 3D snapshots at selected times.

2.2. In order to make sure that your integration is reliable, you will produce a resolution study of your ADI method.

(i) Use spatial resolution $h = \frac{1}{200}$ in both directions. This will be your maximum resolution. Use a total time of about 0.1. Assuming you are saving data at all time steps, you will now determine the maximum number N of time steps from memory constraints: by inspection, find the maximum N allowed in your runs. Once you obtain N, estimate quantitatively the amount of memory used to store your data, in GigaBytes.

(ii) Now that you have fixed the maximum number of time steps and the total time, plot the L^2 norm of the solution $||u(\cdot, t)||_2 \equiv \sqrt{\int u(x, y, t)^2 \, dx \, dy}$, as a function of time, for resolutions $h_0 = \frac{1}{200}$, $h_1 = \frac{1}{100}$ and $h_2 = \frac{1}{50}$. Plot them together with different colors and/or dashing. Comment on the differences and the convergence.

(iii) A more quantitative study is produced by plotting the relative difference between the L^2 norms, i.e. the quantities $e_j(t) = 1 - \frac{||u_j(\cdot,t)||_2}{||u_0(\cdot,t)||_2}$, $\quad j = 1, 2$, where u_j is the numerical solution corresponding to the spatial resolution h_j. Is $|e_1(t)| < |e_2(t)|$ for all times? Are these quantities bounded by a small number? Quantify and comment.

(iv) Modify your code so that it does not save the whole field data $u(x,t)$ at all time steps. This will allow you to go to higher resolutions while keeping the number of time steps fixed at the value N obtained in point (i) above. Extend your resolution study from part (iii) to higher resolutions $h_3 = \frac{1}{400}, \quad h_4 = \frac{1}{800}$, etc., until your simulation time at the highest resolution becomes of the order of an hour or until you reach a memory constraint, whatever happens first. Produce also a plot of the simulation time as a function of the resolution.

Hint: It is important for part (iv) to be clever in the coding part, so that, without saving $u(x,t)$ for all time steps, you can still process on the go the snapshot data $u(x,t_n)$ at each step (or, at selected time steps) in order to obtain the L^2 norms that you need to compare.