

# MSc in CSTE

## Computational Methods & C++

### Assignment

Irene Moulitsas & Peter Sherar  
Cranfield University

October 6, 2016

Hand in date: 05/12/16 (FT), 19/12/16 (PT), 09:30am

## 1 Introduction

In this assignment you are asked to examine the application of numerical schemes discussed in the Computational Methods lectures to a linear advection equation, using C++ Object Oriented and other programming practices discussed in the C++ lectures. In order to do this, we will consider the following problem:

$$\frac{\partial f}{\partial t} + u \frac{\partial f}{\partial x} = 0$$

in a domain

$$x \in [-50, 50]$$

with

$$u = 1.75$$

and two sets of initial/boundary conditions.

First set:

$$\begin{aligned} f(x, 0) &= \frac{1}{2} (\text{sign}(x) + 1.) \\ f(-50, t) &= 0 \\ f(50, t) &= 1 \end{aligned}$$

and second set:

$$\begin{aligned}f(x, 0) &= \frac{1}{2} \exp(-x^2) \\f(-50, t) &= 0 \\f(50, t) &= 0\end{aligned}$$

The analytical solutions for these initial conditions are given respectively.

First set:

$$f(x, t) = \frac{1}{2} (\text{sign}(x - 1.75t) + 1.)$$

and second set:

$$f(x, t) = \frac{1}{2} \exp(-(x - 1.75t)^2)$$

## 2 Computational Methods Tasks

- Write a C++ program which solves the above problem with the prescribed initial and boundary conditions on a uniform grid containing 100 points in  $x$ , using the following schemes:
  - Explicit Upwind
  - Implicit Upwind
  - Lax-Wendroff
  - Richtmyer multi-step
- Solve the problem and output the numerical and exact solutions at time levels:
  - $t = 5$
  - $t = 10$ .
- Compare the analytical and numerical solutions. How do the numerical solutions vary with the Courant number? How does the numerical solution change in time. Comparisons should be both *qualitative* and *quantitative*.
- Increase the grid size to 200 points and then 400 points in  $x$ . Examine the grid convergence of numerical solutions obtained at  $t = 5$ .
- Explain the behaviour of the numerical solutions in terms of the expected properties of the schemes involved. In particular for the implicit upwind scheme, study the accuracy and stability properties in depth and include your detailed mathematical calculations in the Appendix of your report.

### 3 Object Oriented Programming Tasks

Think about the design of your solution before jumping into coding.

- What classes (abstractions) will you use?
- What will be the separation of responsibilities into the chosen classes i.e what is each class going to be responsible for doing?
- What will be the data/methods encapsulation for each class?
- What are the relationships between the chosen classes: aggregation ('has a' or 'contains'), inheritance ('is a kind of'), association ('uses' - one class may use objects of another class in the implementation of its methods)?

Some desirables

- Try to adhere to the SOLID principles for effective design (see BB pages for C++).
- Try to use the Standard Library components where appropriate (containers, iterators, algorithms, numerics). This is well designed, thoroughly tested, robust and efficient code.
- Try to use exceptions to deal with exceptional conditions and catch them at an appropriate higher level.
- Try not to overcomplicate things. Keep things simple but effective.

Documentation is required (see Vector/Matrix classes used in the module for examples of this).

Doxygen documentation for the classes and concise in line comments in the functions/methods to explain what 'chunks of code are doing.

Finally, remember there is no right or wrong answer when it comes to the design but there are better and worse designs. In terms of software, you are aiming for a clean and effective solution to the problem.

### 4 Report

The source program will need to compile on the IT lab machines using Visual Studio 2015 or 2013 without any other external dependencies/libraries/source codes of third parties. Write a report summarising your findings. The report should be no less than 2,500 words and must not exceed 5,000 words. The report can contain any number of figures. All figures and tables in the report should be numbered and discussed.

The report should include a description of the design of your solution explaining your choices and incorporating a UML class diagram. The source code should be included as an Appendix and should be submitted with the doxygen documentation files electronically via the **TurnItInUK** link by 9:30am on the 5<sup>th</sup> December (full-time students) or the 19<sup>th</sup> December (part-time students). The source code and documentation files must also be submitted directly to the instructors via **e-mail** at i.moulitsas@cranfield.ac.uk and p.sherar@cranfield.ac.uk by the prescribed deadline for the assignment submission to be considered complete.

## 5 REFERENCES

S. Scott Collis, ‘An Introduction to Numerical Analysis for Computational Fluid Dynamics’, Technical Report SAND2005-2745, Sandia National Laboratories, 2005.

K.A. Hoffmann and S.T. Chiang, ‘Computational Fluid Dynamics’, Fourth Edition, Vol. I, Engineering Education System Books, pp. 486, 2000.

Links to the following two documents can be found on the C++ BB page under external links:

Pras Pathmanathan, ‘Numerical Methods and Object-oriented Design’.

Dr O Gloth, ‘Object Oriented Techniques and Numerics’.