MSc in CSTE ESTIA & MSc in ACE
Computational Methods & C++ Assignment

Irene Moulitsas & Peter Sherar
Cranfield University

September 30, 2024

Hand in date: 25/11/2024 (FT), 09/12/2024 (PT), 09:30am

1 Introduction

In this assignment you are asked to examine the application of numerical schemes for the solution of partial differential equations as discussed in the Computational Methods lectures, using C++ Object Oriented and functional programming practices discussed in the C++ lectures. In order to do this, we will study the following problem.

Consider the first order wave equation:

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

where u, the speed of sound, is $1.75m/s$. Assume that a disturbance is introduced at $t = 0$ in a one-dimensional long tube of length $L = 100m$ with $x \in [-50, 50]$ with both ends closed. We will consider two sets of imposed initial/boundary conditions:
SET1

$$\begin{aligned} t = 0 \qquad & f(x,0) = \frac{1}{2}\left(sign(x) + 1.\right) \\ x = 0 \qquad & f(-L/2,t) = 0 \\ x = L \qquad & f(L/2,t) = 1 \end{aligned}$$

and
SET2

$$\begin{aligned} t = 0 \qquad & f(x,0) = \frac{1}{2}exp\left(-x^2\right) \\ x = 0 \qquad & f(-L/2,t) = 0 \\ x = L \qquad & f(L/2,t) = 0 \end{aligned}$$

2 Computational Methods Tasks

1. Write a C++ program which solves the above problem on a uniform grid with the prescribed initial and boundary conditions using the following methods:

   - Explicit Upwind FTBS (Forward time, Backward space)
   - Implicit Upwind FTBS (Forward time, Backward space)
   - Lax-Wendroff
   - Richtmyer multi-step

2. The analytical solution of this problem, subject to the imposed initial and boundary conditions, is respectively:
   SET1
   $$f\left(x,t\right)=\frac{1}{2}\left(sign\left(x-1.75t\right)+1.\right)$$
   and
   SET2
   $$f\left(x,t\right)=\frac{1}{2}exp\left(-\left(x-1.75t\right)^{2}\right)$$
   Use the analytical solution to compare/validate the results of the above numerical methods. Comparisons should be both *qualitative* and *quantitative.*

   In all cases the solution is to be printed and plotted for all x locations at time levels $t=5$ and $t=10$.

3. Investigate the effect of various number of space grid points N on the accuracy of the solution and required computation time, of one explicit and one implicit method of your choice, for $t=10$ and appropriate Δt of your choice:

   - $N=100$
   - $N=200$
   - $N=400$

   Do not forget to justify your choice of Δt.

4. Explain the behaviour of the solutions of the above numerical methods in terms of the expected properties of the numerical methods involved.

5. In particular for the implicit upwind method, study the accuracy and stability properties in depth and include your detailed mathematical calculations in the Appendix of your report.

3 C++ Programming Tasks

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

1. What classes (abstractions) and functions will you use?
2. What will be the separation of responsibilities into the chosen classes/functions i.e what is each class/function going to be responsible for doing?
3. What will be the data/methods encapsulation for each class?
4. 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

5. Try to adhere to the SOLID principles for effective design (see Canvas pages for C++ under Other Resources: UML Object Oriented Design (OOD) Design Patterns).
6. Try to use the Standard Library components where appropriate (containers, iterators, algorithms, numerics). This is well designed, thoroughly tested, robust and efficient code.
7. Try to use exceptions to deal with exceptional conditions and catch them at an appropriate higher level.
8. 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 Reflective Task

1. The report should finally contain an Appendix, called Individual Contributions, where each group member will present their opinion of the contribution of every group member towards this assessment and reflect on good and bad practices of the team work.

5 Source Code and Report Requirements

The source code must compile on the IT lab PCs using Visual Studio or Intel/GNU compilers on Linux, without any other external dependencies, libraries, source codes of third parties.

Write a report to present and discuss your findings. The report should be no less than 3,000 words and must not exceed 6,000 words. The report can contain any number of figures/tables, however all figures/tables 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 or other high level diagram showing the structure of your solution. The source code with the doxygen documentation files should be included as an Appendix to the report.

6 Assignment Submission

The source code and documentation should be submitted electronically via the **Technical Work submission point** by 9:30am on the 25$^{\text{th}}$ November (full-time students) or the 9$^{\text{th}}$ December (part-time students).

The report should be submitted electronically via the **TurnItInUK submission point** by the prescribed deadline, for the assignment submission to be considered complete.

This is a group assessment, only one submission per group is required.

7 Marking

The assignment will be assessed based on the following marking scheme:

- 20% Introduction, methodology, conclusions
- 40% Source code, documentation
- 30% Analysis of the results and derivation of theoretical properties
- 10% Report structure, presentation, references

8 References

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

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

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

3. Pras Pathmanathan, ‘Numerical Methods and Object-oriented Design’.
4. Dr O Gloth, ‘Object Oriented Techniques and Numerics’.