

# Response to 1st and 2nd referee reports on “Finite Difference Computing with PDEs”

H. P. Langtangen

S. Linge

Sep 24, 2016

## 1 Comments from referee 1

**Referee.** I have briefly looked at the book and find certain parts very interesting, e.g. the focus on computing, the applications and the use of the python language. On the other hand, I also see items that I would not agree with, e.g. the definition of what is an explicit and an implicit method.

**Response.** Making the distinction according to whether you get a linear system to solve or not is common. Without the referee’s opinion explicitly stated it is not easy to see how he or she wants the difference to be phrased.

We have extended the distinction with a new, second paragraph on the practical implications of increased stability of implicit methods:

“Discretization methods that lead to a coupled system of equations for the unknown function at a new time level are said to be *implicit methods*. The counterpart, *explicit methods*, refers to discretization methods where there is a simple explicit formula for the values of the unknown function at each of the spatial mesh points at the new time level. From an implementational point of view, implicit methods are more comprehensive to code since they require the solution of coupled equations, i.e., a matrix system, at each time level. With explicit methods we have a closed-form formula for the value of the unknown at each mesh point.

Very often explicit schemes have a restriction on the size of the time step that can be relaxed by using implicit schemes. In fact, implicit schemes are frequently unconditionally stable, so the size of the time step is governed by accuracy and not by stability. This is the great advantage of implicit schemes.”

However, bear in mind that when finite elements are applied to the heat equation, we get an implicit scheme that has lower stability than the corresponding explicit scheme produced by the finite difference method (!).

**Referee.** Perhaps the authors should look once at the book by Klaus-Jurgen Bathe “Finite Element Procedures”, 2nd ed., 2014 and read the part on the use of finite difference methods in that book.

**Response.** We have not yet got our hands on the second edition of Bathe's book, but in the first one, there is only one distinction mentioned in the context of IMEX methods. We also cover that in the splitting of nonlinear PDEs. Otherwise, there is very little about finite differences and comparison with finite elements in his excellent book.

## 2 Comments from referee 2

**Referee.** So all I could do is to browse through the book quickly and it gave me a nice impression. In particular I like the very applied and instructive approach with complete code examples in Python and many problems and exercises.

**Response.** We appreciate this comment and that a particular goal of the book came through.

**Referee.** On the other side finite difference methods may be considered a bit outdated in comparison to the ongoing development of Finite-Element and Finite-Volume methods but FD certainly still has a strong user base.

**Response.** It is correct that state-of-the-art methods for PDEs will typically involve finite element or volume methods today. However, we aim at a generation of new students who are to quickly open their laptop and compute solutions whenever they see a basic PDE. The finite element or volume methods will often be an overkill. A lot of problems in science courses are 1D PDEs for which a finite difference method can be coded in minutes, with control of all details. Finite element and volume methods will require much more knowledge and also suitable libraries before they are really useful. We therefore think that the potential of coding 1-2 page long programs with finite differences is great. The book will educate the generation of students who can do this - even without being asked, it is only natural.

Also, a thorough understanding of finite differences greatly helps to understand the behavior of finite element and volume methods. A third argument is that testing of more complicated finite element and volume methods often need some reference code, and that has to be independent. A program based on finite differences for a (very) simplified 1D problem is fast to make and may help debugging and verification. We have put some of these views in the preface now.

**Referee.** Finally, it is rather shocking that the authors seem not to give reference to any other publications. If this is true, I would actually not consider it scientific literature. I think proper scientific reference is expected standard for TCSE.

**Response.** Yes, we understand this reaction and appreciate the comment. We have now included several references.

We have limited the amount of references because the methods we use now are *very basic* and described in a lot of sources. Giving credit to original sources is usually not a point unless there is some special detail about the method in question. We refer to extra and alternative material in several textbooks when we feel that is appropriate. However, our strong focus on ideas and implementation does not have sources that naturally lend themselves for reference, as much has evolved into just folklore in the community.