Referee's report for

Towards a Unified Framework for Scientific Computing

by

Peter Bastian, Mark Droske, Christian Engwer, Robert Klöfkorn, Thimo Neubauer, Mario Ohlberger, Martin Rumpf

The article presents a new general framework for numerical analysis software. It is motivated by the fact that nowadays finite element and finite volume software packages come with their own grid management and algebra routines. As each grid management concept has its particular strengths and weaknesses, the applicability of the whole software package is limited by those. The authors propose a new modular concept, where grid management, algebra, etc. are separated from the application code by abstract interfaces. This allows to formulate numerical algorithms independently from particular grid and algebra implementations. The authors show how such a framework can be realized efficiently using static polymorphism.

The referee considers the proposed framework a very promising approach. It offers a large step beyond the limitations of current scientific computing software. Also, it is an economic concept. First, it allows the incorporation of existing grid management routines and other legacy code. Secondly, by giving standardized interfaces, it makes sharing of application code easier.

For these reasons, the referee recommends the article for publication.

Details

Page 4, L. 2: The expression 'grid definition' makes the referee think of 'definition' in the C++ sense (which supposedly is not the intended meaning). Maybe use 'grid interface' instead?

Page 4, L. 10: Why does Ω have to be connected? On the other hand, it should be bounded.

Page 4, Def. 1.2: Is the expression 'contained' to be understood in the set-theoretic sense? If yes, this would exclude parametrized boundaries and grid smoothing procedures across different hierarchy levels. Why is this restriction necessary?

Page 4, Def. 1.2: What does $\exists! j$ mean?

Page 4, Def. 1.2: The formal definition of $\mathcal{C}_L(\Omega^0_{k,i})$ includes not only the direct children but all descendants of an entity. Therefore the set should

possibly be called ‘set of all descendant entities’ instead of ‘set of all child entities’

Page 5, Def. 1.4: Shouldn’t $m^c_{l,i}$ even be a diffeomorphism?

Page 5, Def. 1.4: The dimension of the grid has been defined already on Page 4, L. 13.

Page 5, Def. 1.5: ‘... in the sense that the intersection of the closure of two grid entities...’

Page 5, L. 11 from the bottom: What parameters do the triple dots in the signature of `Grid` stand for?

Page 6: Does the hierarchic iterator traverse all children or all descendants of an entity?

Chapter 4: Is the example problem well-posed with the given boundary values?

Page 7, Table 1: Why is F so much slower than E, even though is uses more or less the same number of elements?

Page 7, Table 1: It would be interesting to see the approximate memory consumptions for the different implementations.

Language and Spelling

Third Address: put a comma after ‘Universität Duisburg’

Chapter 1, L. 9: ‘... sequential or parallel mesh data structures are possible alternatives.’

Chapter 1, L. 10: ‘Using one particular code it may be impossible [...] or a feature may be very inefficient to use (e.g., a structured mesh...’

Page 2, L. 3: ‘This fact is well known in computer science (Brooks [1975]).’

Page 2, 7th line from the bottom: ‘...abstract interfaces and shows how these interfaces...’

Page 2, 5th line from the bottom: no comma after ‘finite element’

Page 3, L. 1: no comma after ‘subroutines’

Page 3, L. 3: put a comma after ‘E. g.’

Page 3, L. 8: ‘extent’

Page 3, L. 9: put a comma after ‘E. g.’

Page 3, L. 18: put a comma before and after ‘e. g.’

Page 3, L. 20: '... programming techniques we use...'

Page 3, last line: '... finite element and finite volume...'

Page 4, L. 3: put a comma after 'particular'

Page 4, L. 7: 'support local, hierarchical...'

Page 4, L. 14: 'Each entity set...'

Page 5, L. 6: '... reference element to the actual element...'

Page 5, L. 14: no comma after 'require'

Page 5, L. 17: 'Classes in the DUNE Grid Interface'

Page 5, List of classes: Use a different font for the classes

Page 5, L. 8 from the bottom: '... access to its entities.'

Page 6, L. 13: 'displays'