Uppsala University
Institute for Information Technology

Parallel algorithms for Scientific Computing
VT 2013

Reviewing an article on 'Parallel algorithms for Scientific Computing'

1 Introduction and task description

The aim of this assignment is to act as a reviewer of a scientific report and evaluate its suitability for publication in a high quality scientific journal.
The scientific journal has a broad audience, consisting of experts in Scientific Computing and parallel computations, as well as of applied scientists that follow the development of the field and intend to use resent achievements for their particular numerical simulations.
You should prepare a written review report that addresses the following issues:

(a) what is the general idea of the article

(b) what is the novelty in the described research

(c) what is your own evaluation of the claimed achievements in the article, based on the issues discussed during the course

(d) is the article well written and can be understood by the journal audience

(e) are there fair references in the bibliography list to related work or the authors referee mostly to their own results

(f) is there a good motivation why the research is performed -

(a) minor comments - organization of the paper, clear structure, presentations of the results, readability, language, etc.

(a) recommendations to the Editor (accept as is, major revision, minor revision, reject)

Reviewing of scientific articles is difficult. Sometimes we have to do it even if we are not familiar enough with the topic of the research. When commenting on the articles, try to show as much as possible your understanding and opinion on the article in view of the topics discussed in the course.
As examples, I attach two reviewers reports on papers, submitted to an existing scientific journal.

2 Participants and assigned articles

1. Malin Kälen
   Minimizing communication in sparse matrix solvers, http://dl.acm.org/citation.cfm?id=1654096
2. Ali Dorostkar
   Ultrascalable implicit finite element analyses in solid mechanics with over a half a billion degrees of freedom, www.columbia.edu/~ma2325/adams.sc04.pdf\u200e
3. Afshin Zafari
   Parallel and Fully Recursive Multifrontal Supernodal Sparse Cholesky, http://link.springer.com/chapter/10.1007%2F3-540-46080-2_35
4. Jan Klosa
   On parallel stochastic simulation of diffusive systems, link.springer.com/chapter/10.1007%2F978-3-540-88562-7_16\u200e
5. Martin Tillenius
   Parallel two-stage reduction to Hessenberg form using dynamic scheduling on shared-memory architecture, http://dx.doi.org/10.1016/j.parco.2011.05.001
6. Pavol Baur
   Reduced-Bandwidth Multithreaded Algorithms for Sparse Matrix-Vector Multiplication, gauss.cs.ucsb.edu/~aydin/ipdps2011.pdf
7. Behrang Mahiani
   SuiteSparseQR: Algorithm 9xx: SuiteSparseQR, a multifrontal multithreaded sparse QR factorization package, paper, describing the features of the sparse QR algorithm from the package 'SuiteSparseQR', http://www.cise.ufl.edu/research/sparse/SPQR/
8. Magnus Gustafsson
   A parallel eigensolver using contour integration for generalized eigenvalue problems in molecular simulation, www.cs.tsukuba.ac.jp/techreport/data/CS-TR-08-14.pdf
9. Siyang Wang
   Numerical Evaluation of the Communication-Avoiding Lanczos Algorithm http://www.it.uu.se/research/publications/reports/2012-001/
10. Kristoffer Virta
    Parallel 3-D viscoelastic finite difference seismic modelling, http://dx.doi.org/10.1016/S0098-3004(02)00006-7
11. Marcus Holm
    Parallel graph partitioning on multicore architectures, http://dl.acm.org/citation.cfm?id=1964553

12. Ababacar Diagne
Parallel Particle Advection and FTLE Computation for Time-Varying Flow Fields, `conferences.computer.org/sc/2012/papers/1000a102.pdf`

3 Examples of reviews from a scientific journal

You find two reviews on one and the same paper. One of them suggests to reject the paper and the other suggests only minor revision.

3.1 Review no.1

Recommendation: **reject**
The paper deals with the solution of block tridiagonal block Toeplitz systems where the matrix has the form trid(T;D; T) and the blocks T;D commute. For such systems there exists a wide literature on algorithms based on the cyclic reduction technique.
The purpose of this paper, as explained by the authors in the introduction, is to present an alternative and more intuitive way of deriving the radix-4 cyclic reduction method. In fact, the authors survey on radix-2 and radix-4 cyclic reduction algorithms, describe some partial fraction formulae for the radix-4 method, and perform the error analysis. The results of some experiments are also shown.
The paper does not contain relevant advances in this research eld. No new idea or methodology is introduced. The algorithmic improvement of the radix-4 algorithm is minor, moreover the practical computational advances are negligible. The acceleration w.r.t. the radix-2 algorithm of the CPU time is by a factor of at most 1.3 for two-dimensional problems. Moreover, the acceleration deteriorates when the size increases. Also the advantage of having a multi-core machine is negligible, if any. In fact, the acceleration obtained with a 4-core machine is lower than the acceleration obtained with a single core. English requires revision. Grammar errors have been encountered. Here is a sample
p2 l4: consist – consists
p2 l20: suggest – suggests
p2 l31: analyze – analysis
p10 l11: that – than
p11 l15: analyze – analysis
p12 l33: it can concluded – we can conclude
p13 l17: can described – can be described
p17 l24: There exist – There exists

3.2 Review no.2

Recommendation: **minor revision**

The authors consider well-known block cyclic reduction algorithm and propose a trick which usually is employed in FFT algorithms|combining reduction formulas for neighbouring levels, thus moving from, say, radix-2 to radix-4 reduction tree. Further, matrices of linear systems of each level being rational functions with known roots and poles are represented in the form suitable for parallel computation (matrix inverse is represented as a sum of simple poles) – this turns out to work for radix-4 case, too. The authors consider the operation count and numerical stability of the new method. Finally, the connection with fast direct (PSCR) method is noticed for the case of a simple Poisson problem and some numerical experiments are presented. The paper is interesting and worth publishing, however I have several remarks.

- A natural question arises, whether radix-4 is optimal in (sequential) operation count. For example, FFT algorithms with split-radix reduction are known to be faster than, say, radix-4 algorithms. Not a word on this can be found in the article.

- There are several misprints, e.g. the word "analyze" is sometimes used as a noun, superscript (r) is missing over $w_i i$ in formula (31), p. 13, excessive "r" in formula (32).

- p. 2, at the top: "therefore requires an initialization stage". For the problems considered (Toeplitz case), the PSCR method also does not require initialization stage, since tridiagonal Toeplitz matrices have analytical expressions for eigen- values and eigenvectors [1]. In fact, CR initialization is "hidden" in recurrent formulas for matrices of linear systems on each level, even after partial fraction transformation.

- p. 4, formula (6) – probably it is worth noting that these are the roots of the polynomials given in (7). Item p. 6, formulas (12) . – it seems that the combining of two neighbouring reduction steps is not complete, since odd levels, like D^{2r-1}, show up in these formulas: this is strange. If one just writes two steps in a sequence, without any transformation, this would also look like (12) but without any dierence in Figure 1. Another variant when odd-order operators appear in nal formulas is split-radix reduction, but this is not the case.

- p. 7, partial fraction formulas: it is well known that without this technique, the order of "inverting" the terms $D - c_jT)$ showing up in (6) is very important [2] and may cause numerical instability in computations on a single level. It would be nice if authors could provide a reference (or a proof) that in the evaluation of corresponding sum, say (18), the order of the terms is not important.

- numerical stability argument (section 4) seems to be rather simple and probably the section can be shortened, since radix-4 relevant part is (24) and what follows should be similar to corresponding radix-2 analysis (existing somewhere, [12]?)

- p. 13, partial solution discussion: seems incomplete, probably (35) should also be included or (better) removed completely – is it relevant here?

- p. 16, line 2: "normal" could be understood in the sense of matrix analysis (normal matrices") which probably was not desired.

- p. 18: please add some comments to numerical results. Why radix-4 efficiency is lower than expected? The "expected" curve does not seem to approach the observed results even in the limit of large n. Why 4 cores scale worse than 1? It seems that partial fraction technique has much larger potential than just 4 cores.

DEADLINE: Please deliver a printed copy of the article and a printed copy of your report to Maya Neytcheva not later than by **June 10, 2013**.