

Doing Moore with Less – Leapfrogging Moore's Law with Inexactness for Supercomputing

Sven Leyffer, Stefan M. Wild, Mike Fagan, Marc Snir, Krishna Palem, Kazutomo Yoshii, Hal Finkel, *Member, IEEE*

Abstract—We investigate the use of inexactness, to trade solution quality with energy savings to reduce power consumption in HPC systems, using commercial off-the-shelf processors for lower-precision arithmetic. We implement an inexact Newton algorithm for solving nonlinear equations to show how to reinvest energy savings to improve solution quality. We provide an analytical model to explain our findings, and provide a justification for using nonstandard half-precision arithmetic for ill-conditioned problems.



1 INTRODUCTION

The performance of leading supercomputers has increased by an order of magnitude every 3–4 years in the past two decades. The main factors contributing to this growth have been (i) the sustained exponential increase in the performance of processors, as exemplified by Moore's Law; (ii) a growth in the size of the largest supercomputers and of their energy consumption; and (iii) the use of increasingly specialized and more energy-efficient components, such as GPUs, long vector units, and multicores.

Two of these three factors are running out of steam. Moore's law is coming to an end as it has become increasingly difficult, and hence expensive, to reduce the energy consumption of transistors. There are few alternative technologies that hold a promise of a better energy×delay product and none are close to commercial deployment. Current top supercomputers require power in the range of 10–20MW; it is hard to significantly increase these numbers, because of the cost of energy and the limitation of existing installations. Continued increase in effective supercomputer performance will increasingly depend on a smarter use of existing systems.

The study of the precision level of numerical algorithms has been a fundamental subject in numerical analysis since the inception of the field [1], [2]. Much attention has been devoted to the numerical errors induced in simulations by the discretization of continuous space and time; and to the tradeoff between precision level and number of iterations in iterative methods. Less attention has been paid in the HPC world to the discretization of real numbers into 16-, 32-, or 64-bit floating-point values, since the use of lower precision did not result in significant performance gains on commodity processors.

2 SOLVING NONLINEAR EQUATIONS

We investigate the effect of low-precision arithmetic on an inexact Newton method. The inexact Newton method [3] considered here, and Jacobian-free Newton-Krylov methods [4] in

general, are workhorses in modern nonlinear solvers and provide insight into how low-precision arithmetic can be exploited in scientific computing. Our goal is to solve the nonlinear system of equations

$$F(x) = 0, \quad (1)$$

where $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is twice continuously differentiable.

The basic inexact Newton method starts from an initial iterate x^0 , and consists of outer and inner iterations. The outer iterations correspond to approximate Newton steps that produce a sequence $\{x^k\}_k$ of iterates. Given x^k , the inner iteration solves the Newton system

$$\nabla F(x^k)s = -F(x^k) \quad (2)$$

approximately for $s \in \mathbb{R}^n$; in our case, this is done by using BICGSTAB [5], but the specific form of inner solver is not a critical part of the present analysis. Our implementation is matrix-free, in that the Jacobian matrix $\nabla F(x)$ is never explicitly evaluated; the user needs only to implement vector products with the Jacobian matrix.

We consider two sets of test problems of variable dimension. The first problem, Laplace, is a well-conditioned linear system of equations, derived from a central-difference discretization of the Poisson equation. We note, however, that because we use an inexact Newton solver, the linear system is not solved in a single outer iteration. The second problem, Rosenbrock, is nonlinear and notoriously ill-conditioned, and provides a more strenuous test for our low-precision implementation.

3 EXPERIMENTAL RESULTS

Our hardware testbed for this study was a Dell precision T1700 workstation operated by CentOS 7 and equipped with an Intel core i7 4770 (3.40 GHz) and 16 GB of DDR3 RAM. The processor had 4 cores, each with 2 hyperthreads, giving 8 logical CPUs. The Intel core i7 4770 processor has 2 important architectural features:

- 1) It implements the AVX2 instructions set
- 2) It supports the Running Average Power Limit (abbreviated RAPL) hardware counter [6]

Item 2 gives us a way to measure the energy consumption for a given program. The specific RAPL tool that we used, `etrace2`, was written by one of the authors of this paper.

The energy expended by each of the four variants of the Laplace experiment are shown in Fig. 1. The energy consumption for single- and double-precision arithmetic operations are

- S. Leyffer, S. Wild, K. Yoshii, and H. Finkel are at Argonne National Laboratory, Lemont, IL, USA.
E-mail: leyffer@anl.gov
- M. Fagan and K. Palem are at the Department of Computer Science, Rice University, Houston, TX, USA.
- M. Snir is at the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA.
- This material is based upon work supported by the US Dept. of Energy, Office of Science, ASCR, under contract DE-AC02-06CH11357 and by DARPA Grant FA8750-16-2-0004.

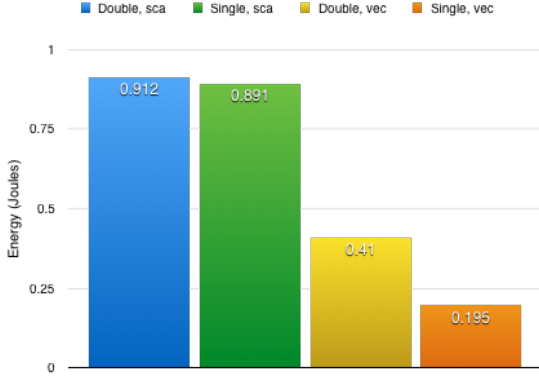


Fig. 1. Energy expended by Inexact Newton variants on Laplace; convergence to accuracy $\epsilon = 10^{-6}$.

the same. On the other hand, vector arithmetic instructions will perform twice as many operations for the same time and energy budget in single precision as in double precision. In particular, vector loads will load twice as many data values for the same energy budget. Vector operations are also much more energy efficient per operand than scalar operations – by a factor of 2 to 4. Thus, we expect modest gains when moving from scalar double precision to scalar single precision.

Our initial gain experiments show that the SIMD vector codes have better energy savings. Consequently, we now focus on the vectorized variants for our reinvestment strategy:

- We run the algorithm in double precision to a given error bound and measure energy consumption. This is our energy budget.
- We next run the algorithm in single precision for a number of iterations, followed by double precision for a number of iterations, consuming the same energy as before, and measure the error bound. The ratio between the first error bound and the second is the achieved *improvement factor*.

We have also experimented with alternative half-precision formats after observing that IEEE's standard half-precision format did not provide a sufficient dynamic range for ill-conditioned problems. By changing the number of exponent/mantissa bits from IEEE's 5/10 to 7/8, we were able to also solve ill-conditioned problems in half-precision arithmetic.

4 THE MATHEMATICS OF ENERGY REINVESTMENT

We develop a model of the reinvestment of the energy saved using low-precision arithmetic, and show that under fairly mild assumptions, we can expect to obtain a more accurate solution at a reduced cost, and we quantify the potential savings in the example of our inexact Newton applications.

Given its iterative nature and adaptive accuracy requirements, the inexact Newton method lends itself almost ideally to an approach that seeks to take advantage of inexact and adaptive-precision arithmetic. We are motivated by the observation that most of the work of Newton's method typically occurs before the transition to fast quadratic convergence.

We propose a simple meta-strategy that starts by running the inexact Newton with low-precision arithmetic, and switches to increasingly higher accuracy levels as we approach the solution. It consists of running the inexact Newton method at prescribed precision levels, with an accuracy tolerance, discussed further below, based on the current precision level. Many

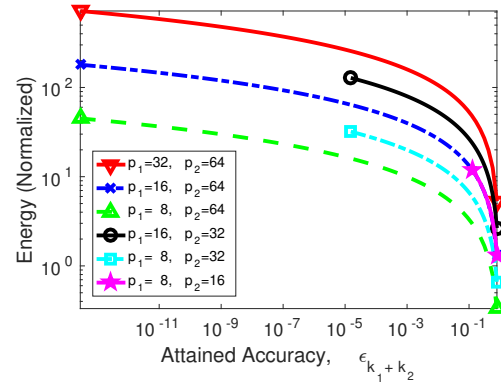


Fig. 2. Energy (less a normalizing constant) expended for linear convergence rate as a function of ϵ for different hybrid precision levels (assumes $s_i = 3 \lg p_i + 1$ and $E(p) \propto p$).

other approaches are possible, but this simple meta-strategy facilitates analysis and is representative of other inexactness-switching-based strategies for linear and nonlinear solvers [7], [8].

We generalize the energy model in [9] and derive an energy model for each individual iteration of inexact Newton that also takes cache misses into account. We note that for both equations (Laplace and Rosenbrock) solved by our inexact Newton code have tridiagonal Jacobians. Hence, the MatVec operation for our particular test cases is $O(n)$, rather than $O(n^2)$ in the dense case. All other operations in any given iteration are also $O(n)$.

We let kn denote the number of floating-point operations at an outer iteration, and ln denote the number of floating-point storage locations. In our examples, both k and l are small integers when compared with the problem dimension n . We let $E_c(p)$ and $E_t(p)$ denote the energy required for a single floating-point compute and transfer, respectively, at precision level p . We neglect the cost of L1 cache transfers, and instead only consider L2 and L3 cache transfers.

Here, we develop a mathematical model for the reinvestment strategy. Formally, we compare the accuracy of a computation done using double precision to the accuracy of a computation using the same amount of energy, but using single precision, where feasible. The analysis can be extended to the case of more than two levels of precision. Our analysis is illustrated in Fig. 2 for a variety of precision levels, under the assumptions $s_i = p_i/2$, the length of the exponent, and $E(p_i) \propto p_i$, the precision level. We show that the improvement factor satisfies

$$\frac{\epsilon}{\epsilon_{k_1+k_2}} \geq 2^{\min\{\lg \epsilon + p_2 - s_2, (p_1 - s_1) \left(1 - \frac{E(p_1)}{E(p_2)}\right)\}}. \quad (3)$$

This bound relates the improvement factor to the original accuracy (ϵ), the precision levels ($p_i - s_i$), and the energy ratio ($\frac{E(p_1)}{E(p_2)}$), where λ is the convergence rate constant.

5 CONCLUSION

We have developed a somewhat paradoxical procedure for reducing the errors in numerical computations by reducing the precision of the floating point operations used. Our focus has been on using in the best possible manner a given energy budget. The paper illustrated one possible tradeoff between computation budget and quality, namely the one achieved by changing numerical precision of floating point numbers.

REFERENCES

- [1] R. W. Hamming, *Numerical Methods for Scientists and Engineers* (2nd Ed.). New York: Dover Publications, 1986.
- [2] J. H. Wilkinson, *Rounding Errors in Algebraic Processes*. New York: Dover Publications, 1994.
- [3] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, "Inexact Newton methods," *SIAM Journal on Numerical Analysis*, vol. 19, no. 2, pp. 400–408, 1982.
- [4] D. A. Knoll and D. E. Keyes, "Jacobian-free Newton-Krylov methods: A survey of approaches and applications," *Journal of Computational Physics*, vol. 193, no. 2, pp. 357–397, 2004.
- [5] H. A. van der Vorst, "Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 13, no. 2, pp. 631–644, 1992.
- [6] H. David, E. Gorbato, U. R. Hanebutte, R. Khanna, and C. Le, "RAPL: memory power estimation and capping," in *16th ACM/IEEE International Symposium on Low Power Electronics and Design*, ser. ISLPED '10, 2010, pp. 189–194.
- [7] P. D. Hovland and L. C. McInnes, "Parallel simulation of compressible flow using automatic differentiation and PETSc," *Parallel Computing*, vol. 27, no. 4, pp. 503–519, 2001.
- [8] G. H. Golub and Q. Ye, "Inexact preconditioned conjugate gradient method with inner-outer iteration," *SIAM Journal on Scientific Computing*, vol. 21, no. 4, pp. 1305–1320, 1999.
- [9] P. Dübén, J. Schlachter, Parishkrati, S. Yenugula, J. Augustine, C.ENZ, K. Palem, and T. N. Palmer, "Opportunities for energy efficient computing: A study of inexact general purpose processors for high-performance and big-data applications," in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE '15)*, 2015, pp. 764–769.