# Nonlinear Solver Algorithms at the Exascale: Rethinking the Full Linearization Bottlenecks

Peter Brune[1], Barry Smith[1], and Matthew Knepley[2]

**Exascale computation** as applied to grand challenge problems [18, 19] in areas like climate, materials science, and biology will invariably require extensive use of parallel nonlinear solver technology. However, it is unlikely that present methodologies will be up to the task. Solvers exhibiting high arithmetic intensity and lightweight global synchronization [15] will be absolutely key in exploiting million-way parallelism on heterogeneous compute nodes. In this position paper, we briefly describe the wider world of nonlinear solution techniques and advocate nonlinear preconditioning and solver composition [4]. We emphasize that there must be significant research and software effort invested in nonlinear solver design before exascale machines are practical simulation platforms.

**Standard nonlinear solvers** cannot be expected to make the jump to exascale. The current best practice of full Newton linearization and multigrid-preconditioned linear Krylov solver (Newton-MG) is too dependent on high relative memory bandwidth utilization and blocking synchronous communication to be viable using any foreseeable technology at the exascale [16].

**Previous work** on the challenge of solvers at the extreme scale has largely focused on the design of the linear solvers and preconditioners. There exist pipelined [12] or s-step [10] variants of Krylov solvers that amortize global communication. Preconditioners designed for the extreme scale include hierarchical Krylov methods [17] that utilize inner Krylov methods on repeatedly nested subproblems and stabilize global convergence with a flexible Krylov method [3]. Domain decomposition [13] and multigrid methods [9] adapted to the extreme scale are also popular choices.

**Nonlinear solver design** is an attractive next place to look, and the design space is vast and underexplored. Solving the nonlinear problem at all levels of a nested solver has long been realized to have many potential advantages [8]. Local nonlinear methods are arithmetically dense and better suited to the memory hierarchies of modern node-level architectures than equivalent linear methods. With predicted future architectures incorporating heterogeneous nodes with extensive vectorization, exploiting locality at the extreme scale will become paramount. However, we are presently unprepared for what will be required to scale past the current generation of large machines and therefore must speculate.

**Deeply hierarchical solvers** are anticipated to be a large part of the transition. Nonlinear solver hierarchies can be built similarly to nested linear solvers, with block nonlinear preconditioners for nonlinear Krylov methods. Level-by-level adaptation may be used to maximize performance.

**Identifying algorithmic building blocks** is an important step towards being able to develop extreme-scale solvers. Some examples of potential methods one may include:

- **Nonlinear Krylov methods** generally combine several previous steps into an optimal next step. This family includes Nonlinear GMRES (NGMRES) [20], Quasi-Newton [14] and others and have communication patterns similar to linear Krylov methods.
- **Nonlinear additive Schwarz methods** (NASM) [7] and nonlinear block-Jacobi methods are a very simple decomposition of a problem into subproblems defined on subdomains.
- **Nonlinear fieldsplit** may be considered in the same breath as NASM methods, but with the splitting being into fields rather than subdomains.

[1]{brune,bsmith}@mcs.anl.gov, Mathematics and Computer Science Division, Argonne National Laboratory
[2]knepley@ci.uchicago.edu, Computations Institute, University of Chicago

- **Full approximation scheme** (FAS) is the extension of multigrid to nonlinear problems, with solution as well as residual represented at each level. Segmental Refinement [1] methods imbue FAS with extremely lean and localized communication compared to standard multigrid.
- **Newton-Krylov** may be used as an incredibly effective local solver.

**Nonlinear preconditioning** is one strategy for formulating globally convergent solvers that treat the full nonlinear problem at the local level. Analogous to linear preconditioning, nonlinear preconditioning constructs a new solution or a new step direction by considering the action of an inner nonlinear solver. As with linear preconditioning, there are two choices:

- **Right** nonlinear preconditioning has the outer method consider only solutions that have been treated by a nonlinear preconditioner. For instance, NGMRES may be used to optimally combine successive iterations from Newton-Krylov, NASM, or FAS. Right preconditioning is related to nonlinear elimination [5].
- **Left** nonlinear preconditioning modifies the outer method's residual to be the change in solution generated by the preconditioner. The left-preconditioned residual is a good indicator of the error. ASPIN [6] is a popular left-preconditioned method.

**General nonlinear solver composition** provides another set of opportunities for the construction of scalable solvers. Simple additive composition allows for different solvers to be converged asynchronously in parallel, with the solutions combined to accelerate global convergence. For example, a lot of the parallel shortcomings of Newton-MG may be lessened by truncation or lagging. One may average between the results of FAS and weakened Newton-MG solutions; outside the basin of Newton convergence FAS will dominate, with Newton-MG taking over when q-quadratic. Alternatively, multiplicative composition of FAS and Newton-MG uses one after the other; FAS "nudges" Newton-MG towards q-quadratic convergence.

**Automated solver composition** by runtime analysis of a simulation would enable rapid adoption of efficient hierarchical nonlinear solvers. Parallel nonlinear solvers have all the communication and load-balancing issues of linear solvers, plus the potential for imbalance arising from varying nonlinearity. Infrastructure beyond standard load balancing is absolutely key at the exascale [11]. Efficient use of the nonlinear solver building blocks will demand on-the-fly adaptivity. Developing detailed performance models for nonlinear solvers is also of utmost importance [2].

**The challenge** of nonlinear solver design at the extreme scale will require new ways of thinking. Preconditioned nonlinear methods and compositions of nonlinear methods may be flexibly adapted to suit a particular problem in conjunction with a particular architecture. We should take particular interest in compositions that include fast multilevel nonlinear algorithms, like FAS, that are arithmetically intense and potentially asynchronous.

**To conclude**, we must consider not only new nonlinear solvers, but new patterns of nonlinear solver design. Issues such as arithmetic intensity, communication patterns, and load balancing will only be magnified by going to exascale. There are compelling reasons to drop the standard Newton-Krylov model and look elsewhere in the design space. New parallel nonlinear solver methodologies, possibly using nonlinear preconditioning and composition, are a research topic that will come to the fore as the current standard methods fail to scale. Computational science can utilize the extreme scale only if we push forward the development of nonlinear solvers that conform to both the limitations and strengths of future high performance computing platforms.

# References


[1] M. F. Adams, *A low memory, highly concurrent multigrid algorithm*, ArXiv e-prints, (2012).

[2] J. Ang, K. Evans, A. Geist, M. Heroux, P. Hovland, O. Marques, L. McInnes, E. Ng, and S. Wild, *Report on the workshop on extreme-scale solvers: Transitions to future architectures*. Office of Advanced Scientific Computing Research, U.S. Department of Energy, 2012. Washington, DC, March 8-9, 2012.

[3] P. N. Brown and Y. Saad, *Hybrid Krylov methods for nonlinear systems of equations*, SIAM J. Sci. Stat. Comput., 11 (1990), pp. 450–481.

[4] P. Brune, M. Knepley, B. Smith, and X. Tu, *Composing scalable nonlinear solvers*, Preprint ANL/MCS-P2010-0112, Argonne National Laboratory, 2013.

[5] X. Cai and X. Li, *Inexact Newton methods with restricted additive Schwarz based nonlinear elimination for problems with high local nonlinearity*, SIAM Journal on Scientific Computing, 33 (2011), pp. 746–762.

[6] X.-C. Cai and D. E. Keyes, *Nonlinearly preconditioned inexact Newton algorithms*, SIAM J. Sci. Comput., 24 (2002), pp. 183–200.

[7] X.-C. Cai, D. E. Keyes, and L. Marcinkowski, *Nonlinear additive Schwarz preconditioners and applications in computational fluid dynamics*, International Journal of Numerical Methods in Fluid Mechanics, 40 (2002), pp. 1463–1470.

[8] T. Chan and K. Jackson, *Nonlinearly preconditioned Krylov subspace methods for discrete Newton algorithms*, SIAM Journal on Scientific and Statistical Computing, 5 (1984), pp. 533–542.

[9] E. Chow, R. Falgout, J. Hu, R. Tuminaro, and U. Yang, *A survey of parallelization techniques for multigrid solvers*, Parallel Processing for Scientific Computing, MA Heroux, P. Raghavan, and HD Simon, eds, 20 (2006), pp. 179–201.

[10] A. T. Chronopoulos and C. W. Gear, *S-step iterative methods for symmetric linear systems*, Journal of Computational and Applied Mathematics, 25 (1989), pp. 153–168.

[11] J. Dongarra, P. Beckman, and et al., *The International Exascale Software Project Roadmap*, Int. J. High Perf. Comput. Applics., 25 (2011), pp. 3–60.

[12] P. Ghysels, T. Ashby, K. Meerbergen, and W. Vanroose, *Hiding global communication latency in the GMRES algorithm on massively parallel machines*, Tech. report 04.2012.1, Intel Exascience Lab, Leuven, Belgium, 2012.

[13] L. Giraud, A. Haidar, and S. Pralet, *Using multiple levels of parallelism to enhance the performance of domain decomposition solvers*, Parallel Computing, 36 (2010), pp. 285–296.

[14] J. J. E. Dennis and J. J. More, *Quasi-Newton methods, motivation and theory*, SIAM Review, 19 (1977), pp. 46–89.

[15] D. Keyes, M. Knepley, and K. Yelick, *Synchronization-reducing and communication-reducing algorithms and programming models for large-scale simulations*, 2012. Workshop sponsored by the Institute for Computational and Experimental Research in Mathematics (ICERM), Jan. 9-13, 2012, Providence, RI, `http://icerm.brown.edu/tw12-1-exascale`.

[16] P. Kogge, K. Bergman, S. Borkar, D. Campbell, W. Carlson, W. Dally, M. Denneau, P. Franzon, W. Harrod, K. Hill, J. Hiller, S. Karp, S. Keckler, D. Klein, R. Lucas, M. Richards, A. Scarpelly, S. Scott, A. Snavely, T. Sterling, R. S. Williams, and K. Yelick, *Exascale computing study: Technology challenges in achieving exascale systems.* DARPA, 2008.

[17] L. C. McInnes, B. Smith, H. Zhang, and R. T. Mills, *Hierarchical and nested Krylov methods for extreme-scale computing*, Preprint ANL/MCS-P2097-0512, Argonne National Laboratory, 2012. submitted to Parallel Computing.

[18] R. Rosner (Chair), *The Opportunities and Challenges of Exascale Computing.* Office of Science, U.S. Department of Energy, 2010.

[19] H. Simon, T. Zacharia, and R. Stevens, *Modeling and Simulation at the Exascale for Energy and the Environment*, 2007. See `http://www.sc.doe.gov/ascr/ProgramDocuments/Docs/TownHall.pdf`.

[20] T. Washio and C. W. Oosterlee, *Krylov subspace acceleration for nonlinear multigrid schemes with application to recirculating flow*, SIAM Journal on Scientific Computing, 21 (2000), pp. 1670–1690.