

Newton-Krylov-Schwarz Methods: Interfacing Sparse Linear Solvers with Nonlinear Applications

Parallel implicit solution methods are increasingly important in large-scale applications, since reliable low-residual solutions to individual steady-state analyses are often needed repeatedly in multidisciplinary analysis and optimization. We review a class of linear implicit methods called Krylov-Schwarz and a class of nonlinear implicit methods called Newton-Krylov. Newton-Krylov methods are suited for problems in which it is unreasonable to compute or store a true Jacobian, given a strong enough preconditioner for the inner linear system that needs to be solved for each Newton correction. Schwarz-type domain decomposition preconditioning provides good data locality for parallel implementations over a range of granularities. Their composition forms a class of methods called Newton-Krylov-Schwarz with strong potential for parallel implicit solution, as illustrated on an aerodynamics application.

1. Introduction to Newton-Krylov-Schwarz Methods

Several trends contribute to the importance of parallel implicit algorithms in fields modeled by PDEs, whose discretization leads to large nonlinear systems with sparse Jacobians. Multidisciplinary analysis and optimization put a premium on the ability of algorithms to achieve low residual solutions rapidly, since analysis codes for individual components are typically solved iteratively and their results are often differenced for sensitivities. Problems possessing multiple scales provide the classical motivation for implicit algorithms and arise frequently in locally adaptive contexts or in dynamical contexts with multiple time scales, such as aero-elasticity. Meanwhile, the never slackening demand for resolution and prompt turnaround forces consideration of parallelism, and, for cost effectiveness, particularly parallelism of the high-latency, low-bandwidth variety represented by workstation clusters.

A Newton-Krylov-Schwarz (NKS) method combines a Newton-Krylov (NK) method, such as nonlinear GMRES, with a Krylov-Schwarz (KS) method, such as additive Schwarz. The key linkage is provided by the Krylov method, of which restarted GMRES is perhaps the best-known example for nonselfadjoint problems. From a computational point of view, the most important characteristic of a Krylov method for the linear system $Au = f$ is that information about the matrix A needs to be accessed only in the form of matrix-vector products in a small number (relative to the dimension of the matrix) of carefully chosen directions. NK methods are suited for nonlinear problems in which it is unreasonable to compute or store a true, full Jacobian. However, if the Jacobian A is ill-conditioned, the Krylov method will require an unacceptably large number of iterations. The system can be transformed into the equivalent form $B^{-1}Au = B^{-1}f$ through the action of a preconditioner, B , whose inverse action approximates that of A , but at smaller cost. It is in the choice of preconditioning where the battle for low computational cost and scalable parallelism is usually won or lost.

In KS methods, the preconditioning is introduced on a subdomain-by-subdomain basis, which provides good data locality for parallel implementations over a range of granularities, and allows significant architectural adaptivity. What is now known as the Schwarz method was introduced by the Swiss theorist Hermann Schwarz in 1869 as a stationary iteration over a pair of overlapping subdomains. It provided an analytical means of proving convergence for an elliptic problem on a domain not lending itself to any separable coordinate system. The emphasis today is on operation count complexity and parallel efficiency, which means that Schwarz is usually employed with very modest subdomain overlap and in a two-level form, in which a small global problem is solved together with the local subdomain problems at each iteration. Mathematically, if $Au = f$ arises as the linearized correction step of a discretized PDE computation, Schwarz operates by:

1. Decomposing the space of the solution u : $U = \sum_k U_k$;
2. Finding the restriction of A to each U_k : $A_k = R_k A R_k^T$, for some restriction operators $R_k : U \rightarrow U_k$ and extension operators $R_k^T : U_k \rightarrow U$;
3. Forming B^{-1} from the A_k^{-1} , where the inverse of A_k is well defined within the k^{th} subspace.

In Schwarz-style domain decomposition, the subspace U_k corresponding to subdomain k is the set of nodal basis or

other expansion functions with support over the subdomain. A practical Schwarz preconditioner is

$$B^{-1} \equiv \sum_k R_k^T (\tilde{A}_k)^{-1} R_k, \quad (1)$$

where \tilde{A}_k is a convenient approximation to $A_k \equiv R_k A R_k^T$. In this paper, \tilde{A}_k is an ILU factorization of A_k , with zero fill permitted. Of course, we never actually assemble either A or B^{-1} globally. Rather, when their action on a vector is needed, a processor governing each subdomain executes local operations, after receiving a thin buffer of data required from its neighbors to complete stencil operations on the boundary of the subdomain.

The two-level form of additive Schwarz can be proved to possess mesh-independent and granularity-independent condition number in elliptically dominated problems, including nonsymmetric and indefinite problems, when the coarse global and fine local operators are solved with sufficient precision. Ref. [2] contains several examples demonstrating this optimality when exact solves are used, and shows their superiority to global incomplete LU factorizations. Ref. [5] is representative of the state of the art. In this paper, the coarse grid is not used, but see Ref. [8] for numerical results comparing the use of a coarse grid on the same problem, including parallel performance impacts.

The NKS technique is compared in this paper against a defect correction algorithm common to many implicit codes. The objective of either algorithm is to solve the steady-state conservation equations $f(u) = 0$ through the pseudo-transient form $\frac{\partial u}{\partial t} + f(u) = 0$, where the time derivative is approximated by backwards differencing, with a time step that ultimately approaches infinity. A standard defect correction approach employs an accurate right-hand side residual discretization, $f_{high}(u)$, and a convenient left-hand side Jacobian approximation, $J_{low}(u)$, based on a low-accuracy residual $f_{low}(u)$, to compute a sequence of corrections, $\delta u = u^{n+1} - u^n$. Computational short-cuts are employed in the creation of the left-hand side matrix, which may, for instance, be stabilized by a degree of first-order unwinding that would not be acceptable in the discretization of the residual itself.

The so-called ‘‘defect’’ is $f_{high}(u) - f_{low}(u)$, and the nonlinear defect correction scheme to drive $f_{high}(u)$ to zero is:

$$f_{low}(u^{n+1}) = f_{low}(u^n) - f_{high}(u^n), \quad (2)$$

which may be linearized as

$$J_{low}(u^n) \delta u = -f_{high}(u^n). \quad (3)$$

In the case of pseudo-transient computations, the approximate Jacobian J_{low} is based on a low-accuracy residual:

$$J_{low} = \frac{D}{\delta t} + \frac{\partial f_{low}}{\partial u}, \quad (4)$$

where D is a scaling matrix. Inconsistency between the left- and right-hand sides prevents the use of large time steps, δt , and prevents (3) from being a true Newton method.

It is required either to solve with J_{low} , itself, or with some further algebraic or parallel approximation, \tilde{J}_{low} . A Newton-Krylov approach employs a (nearly) consistent left-hand side obtained by directionally differencing the actual residual, f_{high} :

$$J_{high}(u^n) \delta u = -f_{high}(u^n), \quad (5)$$

in which the action of J_{high} on a vector is obtained through directional differencing, namely

$$J_{high}(u^l) v \approx \frac{1}{h} [f_{high}(u^l + hv) - f_{high}(u^l)], \quad (6)$$

where h is a small parameter. The operators on both sides of (5) are based on consistent high-order discretizations; hence time steps can be advanced to arbitrarily large values, recovering a true Newton method in the limit.

In practice, the choice of h in (6) is important and not entirely trivial. In a well-scaled problem, it should ideally sit near the square-root of the machine unit roundoff, or around 10^{-7} in 32-bit double precision. Smaller values improve the Taylor approximation upon which (6) is based. Larger values preserve more significant digits when the perturbed residuals on the right-hand side of (6) are differenced in finite precision. The scaling of u and v must enter into the choice of h , however. In the context of a Newton-Krylov method based on GMRES, v comes with unit 2-norm, and only u need enter directly into the scaling of h . For a more concrete discussion, see [7].

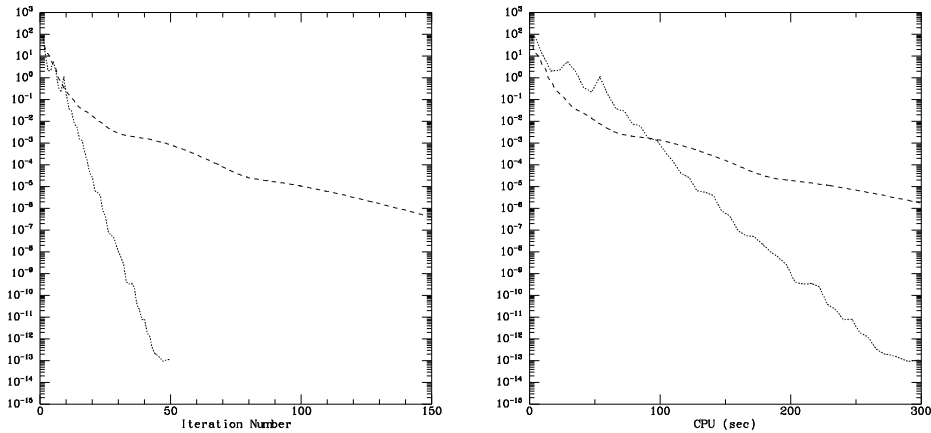


Figure 1: Norm of steady-state residual vs. iterations (left) and vs. CPU time on 32 nodes of the Intel Paragon (right) for the defect correction scheme (dashed), and the NKS method (dotted).

Preconditioning (5) by \tilde{J}_{low} ,

$$(\tilde{J}_{low})^{-1} J_{high}(u^n) \delta u = -(\tilde{J}_{low})^{-1} f_{high}(u^n), \quad (7)$$

shifts the inconsistency from the nonlinear to the linear aspects of the problem. At this level of abstraction it is not clear which is better, many nonlinear steps with cheap subiterations, or a few nonlinear steps with expensive subiterations. CPU comparisons are more practical arbiters than convergence rates for the steady-state residual norm, but depend upon parametric tuning as well as architectural parameters.

2. Aerodynamics Application

The problem of inviscid incompressible flow around a two-dimensional four-element airfoil in landing configuration was studied in terms of convergence rate and parallel performance in [8], and the same code was converted to NKS form for the present study. The details of the discretization are left to the original reference, out of which we choose the vertex-based discretization with first-order Roe on the left, and second-order Roe on the right. The flow is subsonic ($Ma = 0.2$), with an angle of attack of 5° . Adaptively placed unstructured grids of approximately 6,000 and 16,000 vertices decomposed into from 1 to 128 load-balanced subdomains, including all power-of-two granularities in between, are described in [8]. We report below on the problem of 6,019 vertices, with four degrees of freedom per vertex (giving 24,076 as the algebraic dimension of the discrete problem). This is considered small by parallel computational standards, though it is probably reasonably adequate in two dimensions from a physical modeling point of view, since the unstructured grid is not restricted to quasi-uniformity, and mesh cells are concentrated into small regions between the airfoils requiring the greatest refinement.

Figure 1 compares the convergence histories of the defect correction and NKS solvers, over a range of time sufficient to that permit the reduction of the residual of the NKS method to drop to within an order of magnitude of unit roundoff. Both solvers utilize the residual-adaptive setting of the CFL number (or size of the time step in the pseudo-transient code) known as “switched evolution/relaxation” (SER) [6]. Both solvers use the same Schwarz preconditioner, namely one-cell overlap and ILU(0) in each subdomain. NKS is clearly superior to defect correction in convergence rate, though the cost per iteration is sufficiently high that defect correction is faster in CPU time up to a modest residual reduction. (The cross-over point in the right plot is at about a reduction of 10^4 of the initial residual.) Some not-yet-published experiments indicate that a hybrid algorithm, initially defect correction then switched to NKS when defect correction prohibits fast growth in CFL, is ultimately faster than either method exclusively. The asymptotic convergence rate is still linear, since we truncated the Newton iterations well above the tolerances necessary to guarantee superlinear or quadratic convergence.

Table 1 compares the performance of the NKS version of the solver across several doublings of the processor force of the Intel Paragon for this fixed-size problem. The second-order evaluation of fluxes in $f_{high}(u)$ requires that first conserved variables, and later their fluxes, be communicated across subdomain boundaries each time the nonlinear residual is called. This imposes an extra communication burden per iteration on the matrix-free NKS solver, relative to a method that explicitly stores the elements of the low-order Jacobian. Nevertheless, for residual norm reductions of more than a few orders of magnitude, the parallelized NKS solver is faster than the parallelized defect correction solver. The number of subdomains matches the number of processors, so convergence rate of the preconditioned

system degrades slowly with increasing granularity, as coupling is lost in the preconditioner. However, the number of Krylov vectors per Newton iteration is bounded (at 2 restart cycles of 25 each), so the data translates directly to parallelization efficiency of the truncated Newton method.

# proc.	sec./iter.	relative eff.
4	36.09	(1.00)
8	19.21	0.94
16	10.65	0.85
32	6.25	0.72

Table 1: Wall-clock performance and relative parallel efficiency for unstructured Euler code on an Intel Paragon.

Parallel workstation cluster implementations [3] of structured-grid Euler problems reveal the vulnerability of highly synchronous algorithms, such as Krylov methods with their frequent inner product calls, to the non-dedicated environment of clusters. The other three of communication inefficiency in parallel algorithms, namely load imbalance, latency, and finite bandwidth, are believed to impose much less serious limits on the number of workstations that can be clustered together to solve PDEs than frequent synchronization of non-dedicated resources.

3. Conclusions and Future Directions

A variety of CFD applications are (or have inner) nonlinear elliptically-dominated problems amenable to solution by NKS algorithms, which are characterized by low storage requirements (for an implicit method) and locally concentrated data dependencies with small overlaps between the preconditioner blocks. The addition of a global coarse grid in the Schwarz preconditioner is often effective, where architecturally convenient. The main disadvantage of NKS algorithms is the large number of parameters that require tuning. Each component (Newton, Krylov, and Schwarz) has its own set of parameters, the most important of which, in our experience, is the convergence criterion for the inner Krylov subiterations. In large-scale, poorly preconditioned problems, including the test problems of this paper, tunings that guarantee quadratic convergence lead to unacceptable inner iteration counts and/or memory consumption. The *ad hoc* continuation parameter-sensitive tuning of Ajmani et al. [1] seems to offer the best CPU time to attain a given nonlinear residual reduction, when used in combination with an SER [8] continuation scheme. Though parametric tuning is important to performance, conservative robust choices are not difficult.

The NK technique has been compared with V-cycle multigrid on Euler and Navier-Stokes problems without parallelizing the preconditioning in [4]. For a subsonic unstructured grid example, NK trails multigrid in CPU time by a factor of only about 1.5. This penalty can be accepted when it is realized that the NK method has the advantage of doing all of its computation without generation of a family of coarse unstructured grids (which is difficult for three-dimensional unstructured grids). This work has been extended to three-dimensional problems in [7].

4. References

- 1 AJMANI, K., NG, W.-F & LIOU, M.-S.: Preconditioned Conjugate Gradient methods for the Navier-Stokes Equations; J. Comp. Phys. **110** (1994), 68–81.
- 2 CAI, X.-C., GROPP, W. D. & KEYES, D. E.: A Comparison of Some Domain Decomposition and ILU Preconditioned Iterative Methods for Nonsymmetric Elliptic Problems; J. Numer. Lin. Alg. Applics. **1** (1994), 477–504.
- 3 X.-C. CAI, W. D. GROPP, D. E. KEYES & M. D. TIDRIRI: Newton-Krylov-Schwarz Methods in CFD; in “Proc. of the Int. Workshop on Numerical Methods for the Navier-Stokes Equations” (F. Hebeker & R. Rannacher, eds.), Notes in Numerical Fluid Mechanics, Vieweg Verlag, Braunschweig, 1994, 17–30.
- 4 KEYES, D. E.: Aerodynamic Applications of Newton-Krylov-Schwarz Solvers; D. E. Keyes, in “Proc. of the 14th Int. Conf. on Numerical Methods in Fluid Dynamics” (R. Narasimha, et al., eds.), Springer, New York, 1995 (to appear).
- 5 KEYES, D. E. & XU, J., EDS.: Domain Decomposition Methods in Scientific and Engineering Computing; Amer. Math. Soc., Providence, RI, 1995.
- 6 MULDER, W. & VAN LEER, B.: Experiments with Implicit Upwind Methods for the Euler Equations; J. Comp. Phys. **59** (1985), 232–246.
- 7 NIELSEN, E. J., WALTERS, R. W., ANDERSON, W. K. & KEYES, D. E.: AIAA Paper 95-1733, presented at the 12th AIAA Computational Fluid Dynamics Conference, San Diego, June 1995.
- 8 VENKATAKRISHNAN, V.: Parallel Implicit Unstructured Grid Euler Solvers; AIAA J. **32** (1994), 1985–1991.

Addresses: D. E. KEYES, Department of Computer Science, Old Dominion University, Norfolk, VA 23529-0162 and ICASE, MS 132C, NASA LaRC, Hampton, VA 23681-0001, USA.
V. VENKATAKRISHNAN, ICASE, MS 132C, NASA LaRC, Hampton, VA 23681-0001, USA.