Evaluation of overlapping restricted additive Schwarz preconditioning for parallel solution of very large power flow problems

Shrirang Abhyankar
Mathematics and Computer Science Division
Argonne National Laboratory
9700 South Cass Avenue
Argonne, Illinois
abhyshr@mcs.anl.gov

Barry Smith
Mathematics and Computer Science Division
Argonne National Laboratory
9700 South Cass Avenue
Argonne, Illinois
bsmith@mcs.anl.gov

Emil Constantinescu
Mathematics and Computer Science Division
Argonne National Laboratory
9700 South Cass Avenue
Argonne, Illinois
emconsta@mcs.anl.gov

ABSTRACT

The computational bottleneck for large nonlinear AC power flow problems using Newton's method is the solution of the linear system at each iteration. We present a parallel linear solution scheme using the Krylov subspace-based iterative solver GMRES preconditioned with overlapping restricted additive Schwarz method (RASM) that shows promising speedup for this linear system solution. This paper evaluates the performance of RASM with different amounts of overlap and presents its scalability and convergence behavior for three large power flow problems consisting of 22,996, 51,741, and 91,984 buses respectively.

Categories and Subject Descriptors

I.6 [**Computing Methodologies**]: Simulation and Modeling; J.2 [**Computer Applications in Physical Sciences and Engineering**]: Engineering

General Terms

Algorithms, Design, Performance

Keywords

Power flow, Parallel processing, Krylov subspace, Preconditioner

1. INTRODUCTION

Power flow is the backbone of steady-state analysis of power systems. Various steady-state applications, for example contingency analysis, transfer limit calculations, and transient stability analysis, are based on the solution of power flow equations. Hence, any performance improvement in power flow solution will directly benefit the performance of various steady-state applications. Steady state security analysis, involving repeated power flow solutions, is a computationally challenging problem for large regional and national interconnections due to the large number of equations to be solved. Table 1 gives an idea of the size of several power system interconnections in the United States.

HiPCNA-PG'13, November 17 - 21 2013, Denver, CO, USA

http://dx.doi.org/10.1145/2536780.2536784.

Table 1: Sizes of several power system networks

Interconnection	Number of Buses
PJM	13000+ [23]
ERCOT	18000+ [16]
U.S. Eastern Interconnection	50000+
U.S. Western Interconnection	15000+

With the addition of new equipment and increase in load growth, the size of power system networks is expected to increase. Moreover, various ISOs have indicated the need to model and gather real-time information from the subtransmission and distribution systems in order to provide finer-granularity load modeling [21]. The sheer volume of the components needed for such a combined transmission-distribution power flow analysis presents an onerous computational task. As processor speeds were increasing, steady-state security analysis for very large interconnections on a single processor appeared possible in the not-too-distant future. Unfortunately, processor clock speeds saturated about a decade ago.

A natural way to speed this computation is to use parallel computing techniques, namely, share the computational load among multiple processors. The need for parallelizing existing power system applications is even greater as the computer hardware industry moves toward multicore and many core architectures. All major computer vendors are aggressively introducing a new generation of hardware that incorporates multiple cores on a chip, sometimes with additional simultaneous multithreading capabilities. Products incorporating 6 and 8 cores are already on the market, and number of cores per chip is expected to grow rapidly, so that even in the relatively short term, a single chip is expected to support the execution of a few hundred threads. These multicore architectures can be utilized efficiently with parallel algorithms that distribute the computational load over multiple cores.

The aim of this work is to obtain a scalable solution of very large power flow problems. Our focus is specifically on

efficient parallel solution of the linear system of equations during each power flow Newton iteration. We use a preconditioned generalized minimal residual (GMRES) algorithm as the linear solver. As a preconditioner, we use an overlapping restricted additive Schwarz method (RASM) that has proved efficient for applications involving partial differential equations [5]. The performance of RASM is evaluated in terms of scalability and convergence behavior in this work. The results presented for three very large networks, comprising of 22996, 51741, and 91984 buses respectively, show promising scalability.

2. POWER FLOW BACKGROUND

Power flow analysis, sometimes referred to as load flow analysis, is the linchpin of steady-state power systems analysis. Several power system applications, ranging from planning to operation, and from economic scheduling to exchange of power between utilities, hinge on the solution of power flow equations. Contingency analysis and available transfer capacity (ATC) calculations require repeated power flow solutions, while the minimization problem of economically dispatching generating resources includes the power flow equations as a constraint. Transient stability analysis also includes power flow equations in its differential-algebraic model. Moreover, power flow analysis serves as a starting step for transient stability analysis by providing an initial operating point.

Power flow formulation first appeared in the late 1960s [35]. In the early 1970s, a fast decoupled technique was introduced [33] based on the physical insight of weak coupling between real power-voltage magnitude (PV) and reactive power-voltage angles (Qθ). Since then several variations and improvements of power flow formulations and techniques have been introduced [3], [31], [28], [12], [17], [34], [11].

The nodal power balance formulation is the most widely used formulation for power flow analysis. In the power balance form, the set of nonlinear equations to be solved is described by complex power balances at each bus. The summation of the power injected at each bus and absorbed by the network must equate to zero. The resultant complex power balance equation for each bus, or network node, is given by the following.

$$\bar{S}_i^{inj} = \bar{V}_i(\sum_{k=1}^{n}(G_{ik} + jB_{ik})\bar{V}_k)^* \quad (1)$$

In Equation 1, $S_i^{\bar{inj}}$ denotes the complex power injection, that is, $\bar{S}_i^{inj} = \bar{S}_{Gi} - \bar{S}_{Di}$, where $\bar{S}_{Gi}$ is the complex power injected by generators and $\bar{S}_{Di}$ is the complex power absorbed at bus i. Decomposing Equation 1 into real and imaginary parts, we obtain the real and reactive power balance equations as follows:

$$P_i^{inj} - \sum_{k=1}^{n}|V_i||V_k|(G_{ik}cos(\theta_{ik}) + B_{ik}sin(\theta_{ik})) = \Delta P = 0 \quad (2)$$

$$Q_i^{inj} - \sum_{k=1}^{n}|V_i||V_k|(G_{ik}sin(\theta_{ik}) - B_{ik}cos(\theta_{ik})) = \Delta Q = 0, \quad (3)$$

where $\theta_{ik} = \theta_i - \theta_k$. In the power balance form, the variables are expressed in polar coordinates; that is, the variables are the magnitudes and angles of the complex voltage $\bar{V}$ at the buses. This coordinate system is convenient for representing the power balance equations as compared with expressing the equations in rectangular coordinates. In the power flow formulation, PQ buses (constant load, uncontrolled voltage magnitude) are given by Equations 2 and 3 while PV buses (controlled voltage magnitude) are described only by Equation 2. For PV buses (voltage-controlled buses), it is assumed that generators incident on these buses can produce adequate reactive power to regulate the terminal voltage and hence Equation 3 can be omitted. If a generator incident at PV bus reaches its reactive power limit, then it can no longer control the terminal bus voltage and in such a case the bus is switched from PV to PQ. Since our primary goal is to evaluate the preconditioner scalability, we ignore this reactive power constraint. As a result of the assumption of controlled-voltage magnitude for the PV buses, the total number of equations to be solved in the power balance form is $2npq + npv$, where npq is the number of PQ buses and npv is the number of PV buses.

The resultant set of nonlinear equations are solved iteratively by a Newton-Raphson method. The Newton-Rhapson method entails linearization of the power flow equations around the current iterate and then solving a linear system of equations (4) to get an update for the next iterate. Solution of the linear system can be done by either direct or iterative methods. Direct solution via LU factorization is the widely used scheme. Flueck and Chiang [18] presented an iterative solution using GMRES with a physics-based preconditioner derived from the decoupled power flow matrix.

$$\begin{bmatrix} \frac{\partial \Delta P}{\partial \theta} & \frac{\partial \Delta P}{\partial V} \\ \frac{\partial \Delta Q}{\partial \theta} & \frac{\partial \Delta Q}{\partial V} \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix} = - \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} \quad (4)$$

In the context of parallel solution of (4), Wu and Bose [38] presented a parallel LU factorization scheme by rearranging the forward-backward substitution process and grouping together independent tasks. A speedup of 13 on a 20-processor shared-memory computer was obtained in this work. Chen and Chen [10] proposed a partitioning scheme to reorder the matrix in a block bordered diagonal form and presented speedup of 1 to 4 for a 288-bus system. Tu and Flueck [37] presented a parallel direct solution on 15,359 and 30,910-bus systems. A speedup of about 6 was obtained for the two systems. Tu and Flueck [36] also proposed a GMRES-based solution using the matrix obtained from fast-decoupled power flow as the preconditioner. They obtained an increased speedup of about 12 on 16 processors for the same two cases. Massively parallel graphical processing units have also been explored by researchers for the solution of power flow equations [20], [25]. A parallel power flow implementation for shared memory multicore machines using OpenMP was presented in [13].

3. LINEAR SOLVER DETAILS

In this section, we describe the details of our choice of linear solution scheme beginning with the domain decomposition approach. We then briefly describe the Krylov-subspace-based iterative linear solver GMRES followed by overlapping additive Schwarz preconditioning and RASM

3.1 Domain Decomposition

Domain decomposition algorithms operate by dividing the entire domain W into smaller non-overlapping subdomains W_i where

$$W = \bigcup_{i=1}^{N} W_i^0 \tag{5}$$

and each subdomain W_i^0 is the computational assignment of each processor. In the context of power systems, this would mean a division of the power system network into several subnetworks where each subnetwork is assigned to a processor. Domain decomposition algorithms are specifically posed as graph partitioning problems where the dual objective is to obtain balanced partitions (balance workload) along with minimizing the edge-cuts (minimize communication). Several graph partitioning methods, namely combinatorial [27], spectral [29], geometric [9], and multilevel [22], [26], can be found in the literature. Although obtaining good partition is a critical aspect for parallel iterative linear solution, it is beyond the scope of this paper; we refer interested readers to [24] for a good comparison of different graph partitioning algorithms and available packages. We use a multilevel graph partitioning algorithm available through the Chaco [22] package. The multilevel approach recursively creates smaller approximated graphs of the original graph and then does a projection from the smallest level back through intermediate levels. Every few levels of projection, the combinatorial algorithm by Kerninghan and Lin [27] is used for further refinement.

3.2 Generalized Minimum Residual Method (GMRES)

Krylov subspace iterative methods are the most popular among the iterative methods for solving large linear systems. These methods are based on projection onto subspaces called Krylov subspaces of the form $b, Ab, A^2b, A^3b, \ldots$. A general projection method for solving the linear system

$$Ax = b \tag{6}$$

is a method that seeks an approximate solution x_m from an affine subspace $x_0 + K_m$ of dimension m by imposing

$$b - Ax_m \perp L_m,$$

where L_m is another subspace of dimension m and x_0 is an arbitrary initial guess to the soution. A Krylov subspace method is a method for which the subspace K_m is the Krylov subspace

$$K_m(A, r_0) = \text{span}\{r_0, Ar_0, A^2r_0, A^3r_0, \ldots, A^{m-1}r_0\},$$

where $r_0 = b - Ax_0$. The different versions of Krylov subspace methods arise from different choices of the subspace L_m and from the ways in which the system is preconditioned.

The generalized minimum residual method (GMRES)[30] is a projection method based on taking $L_m = AK_m(A, r_0)$ in which K_m is the mth Krylov subspace. This technique minimizes the residual norm over all vectors $x \in x_0 + K_m$. In particular, GMRES creates a sequence x_m that minimizes the norm of the residual at step m over the mth Krylov subspace

$$||b - Ax_m||_2 = min||b - Ax||_2. \tag{7}$$

At step m, an Arnoldi process is applied for the mth Krylov subspace in order to generate the next basis vector. When the norm of the new basis vector is sufficiently small, GMRES solves the minimization problem

$$y_m = argmin||\beta e_1 - \bar{H_m} y||_2,$$

where $\bar{H_m}$ is the $(m+1)xm$ upper Hessenberg matrix.

3.3 Preconditioning

The convergence of the Krylov subspace linear solvers depends on the eigenvalues of the operating matrix A and can be slow if the matrix has widely dispersed eigenvalues, such as ill-conditioned power system matrices. Hence, in order to speed the convergence, a preconditioner matrix M, where M^{-1} approximates A^{-1}, is generally used. A *preconditioner* is a matrix that transforms the linear system

$$Ax = b$$

into another system with a better spectral properties for the iterative solver. If M is the preconditioner matrix, then the transformed linear system is

$$M^{-1}Ax = M^{-1}b. \tag{8}$$

Equation 8 is referred to as being preconditioned from the left, but one can also precondition from the right

$$AM^{-1}y = b, \qquad x = M^{-1}y, \tag{9}$$

or split preconditioning

$$M_1^{-1}AM_2^{-1}y = M_1^{-1}b, \qquad x = M^{-1}y, \tag{10}$$

where the preconditioner is $M = M_1M_2$.

When Krylov subspace methods are used, it is not necessary to form the preconditioned matrices $M^{-1}A$ or AM^{-1} explicitly. Instead, matrix-vector products with A and solutions of linear systems of the form $Mz = r$ are performed.

Designing a good preconditioner depends on the choice of iterative method, problem characteristics, and so forth. In general a good preconditioner should be inexpensive to construct and apply, and the preconditioned system should be easy to solve.

3.4 Overlapping Additive Schwarz Method and RASM

In the area of domain decomposition algorithms, considerable research has been done on overlapping Schwarz algorithms [6], [8], [15], [32], [5], [7]. Our interest for this paper is on overlapping additive Schwarz preconditioner (ASM), and in particular its variant *restricted additive Schwarz preconditioner* introduced by Cai and Sarkis[5]. We first describe the general mechanism of constructing an ASM preconditioner as described in [6] and then present the variant proposed by Cai and Sarkis [5].

3.4.1 Additive Schwarz Method (ASM)

Consider a linear system of the form

$$Ax = b, \tag{11}$$

where $A = (a_{ij})$ is an $n \times n$ nonsingular sparse matrix, having a symmetric nonzero pattern. Defining the graph $G = (W, E)$, where the set of vertices $W = \{1, ..., n\}$, represent the n unknown and the edge set $E = \{(i, j) \,|a_{ij} \neq 0\}$ represent the pairs of vertices that are coupled by a nonzero element in A. Here n is the size of the matrix. Since the nonzero pattern is symmetric the adjacency graph G is indirected. Assume that the graph partition has been applied

resulting in N non-overlapping subsets W_i^0 whose union is W. Define $\{W_i^1\}$ bs the one-overlap partition of W, where $W_i^1 \supset W_i^0$ is obtained by including all the immediate neighboring vertices of the vertices in W_i^0. An example for a domain with two subdomains with one-overlap partition is shown in Figure 1.

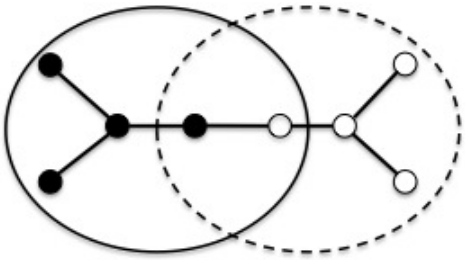

Figure 1: Illustrative example for two overlapping subdomains. Nodes marked • are in subdomain 1 (W_1^0) while ∘ represents nodes in subdomain 2 (W_2^0). The solid circle is the one-overlap for subdomain 1 (W_1^1) and dashed circle for subdomain 2 (W_2^1)

Extending this to δ-overlaps, we can define the δ-overlap partition of W can be defined as,

$$W \bigcup_{i=1}^{N} W_i^{\delta},$$

where $W_i^{\delta} \supset W_i^0$ with δ levels of overlaps with its neighboring subdomains. With each subdomain W_i^0, we define a restriction operator R_i^0. In matrix terms, R_i^0 is a $n \times n$ sub-identity matrix whose diagonal elements are set to one if the corresponding node belongs to W_i^0 and to zero otherwise. Similarly for δ-overlapping restriction operator R_i^{δ} can be defined for each W_i^{δ}. With this, the subdomain operator can be defined as

$$A_i = R_i^{\delta} A R_i^{\delta}. \tag{12}$$

Note that although A_i is not invertible, its restriction to the subspace

$$A_i^{-1} \equiv \left((A_i)_{|L_i} \right)^{-1}$$

can be inverted. Here L_i is the vector space spanned by the set W_i^{δ} in $\mathbb{R}^n$. The additive Schwarz preconditioner can be defined as

$$M_{AS}^{-1} = \sum R_i^{\delta} A_i^{-1} R_i^{\delta}. \tag{13}$$

3.4.2 Restricted Additive Schwarz Method (RASM)

Cai and Sarkis [5] proposed a variant of the additive Schwarz preconditioning called *restricted additive Schwarz preconditioner*. They introduced a simple and efficient change by removing the overlap in the interpolation operator as given in the following.

$$M_{RAS}^{-1} = \sum R_i^0 A_i^{-1} R_i^{\delta} \tag{14}$$

Their main motivation was to save half the communication cost because $R_i^0 x$ does not involve any data exchange with neighboring processors. They also observed that in comparison to the original additive Schwarz algorithm, there was a reduction both in iteration count and CPU time.

3.5 Subdomain Linear Solves

The overlapping Schwarz preconditioners need the solution of subdomain linear system

$$A_i^{-1} r_i \tag{15}$$

during the GMRES iterations, where $r_i = R_i^{\delta}(b - Ax)$ is the restricted error. The solution of Equation 15 can be done either by direct or iterative linear solution schemes. As matrices arising from power system applications are slightly ill-conditioned, our experiments corroborate that the stationary iterative schemes such as as Gauss-Jacobi, Gauss-Siedel, and successive over-relaxation (SOR) converge very slowly. In this work, we solve the linear system in Equation 15 by a direct solver (LU factorization + triangular solves). For a small number of subdomains, direct solve using LU is expensive as the subdomains are large, but the subdomain solve is robust. With more processors, the subdomain sizes become smaller, reducing the cost of direct solves.

To minimize the fill-ins created by LU factorization, we experimented with various reordering strategies, available on the test systems to determine the optimal reordering strategy, namely, the ordering scheme resulting in the least number of nonzeros in the factored matrix. Our prior work [1] showed that a quotient minimum degree [19] was the most efficient among the reordering schemes tested and we continue to use it in this work.

4. PERFORMANCE RESULTS

In this section we present the results obtained for the parallel solution of power flow equations with RASM-preconditioned GMRES scheme as the linear solver.

4.1 Hardware, Software, and Testcase details

The parallel performance runs were done on a shared-memory machine with four 2.2 GHz AMD Opteron 6274 processors. Each processor has 16 cores, giving a total of 64 cores. The code for the developed simulator is written in C using the PETSc [4] library framework, described in Appendix A, and compiled with GNU's gcc compiler with -O3 optimization. We created larger representative test cases synthetically by combining two MatPower [39] package distribution (version 4.1). MatPower includes a variety of power flow test cases, with the smallest being a 4-bus network and the largest consisting of over 3000 buses. We first created a 5,749-bus test case by combining two MatPower test cases, case3012wp (3012 buses) and case2737sop (2737). The 22,996-bus test case was then created by combining four 5,749 cases in a 2 by 2 grid structure as shown in Figure 2. Each 5,749 area is connected to its neighboring area through 6 randomly chosen lines.

The 51,741-bus test case was created similarly by using a 3-by-3 grid and the largest case case91984, was created using four 22,996 bus test cases placed in a 2-by-2 grid. For the largest test case, we used twelve randomly chosen tie lines between the areas instead of six. The inventory for the test cases used is given in Table 2.

4.2 Scalability Results

We compared the performance of GMRES with overlapping restricted additive Schwarz preconditioning scheme for different amounts of overlap, 1, 2, and 3 respectively, with neighboring subdomains. The performance was also com-

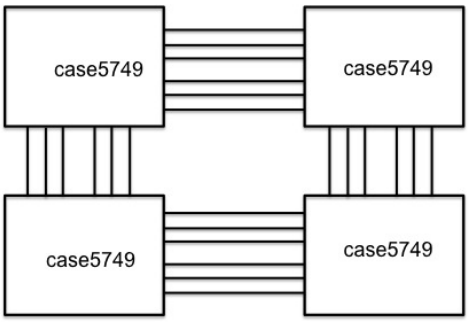

Figure 2: Creation of 22,996 bus test case from four 5,749 bus cases.

Table 2: Inventory of test cases

casename	Buses	Gens	Lines
case22996	22,996	2,416	27,408
case51741	51,741	5,436	61,686
case91984	91,984	9,664	109,680

pared with block-Jacobi preconditioner, essentially a non-overlapping additive Schwarz scheme, and parallel direct solver SuperLU_Dist [14]. All the results were also compared with a Newton-LU on a single core. Figures (3)-(8) present the scalability results on the three test systems ranging from 1 core to 16 cores. For all the test cases, the parallel direct solver SuperLU_Dist and GMRES with block-Jacobi preconditioner were not found to be scalable. In fact, the results show an increase in the execution time as compared with on a single processor. On the other hand, GMRES preconditioned with overlapping restricted additive Schwarz preconditioning produced substantial speedup. For case22996, a maximum speedup of 6.7 was seen with overlap-3 on 16 cores. This yielded in a reduction of the execution time from 0.6 seconds on a single core to 0.089 seconds on 16 cores. Both case51741 and case91984 showed similar speedup characteristics with maximum speedup of 9.6 and 9.1 respectively, achieved with an overlap of 3 on 16 cores. Although the results on higher overlap (overlap > 3) are not presented, we note that a speedup saturation and slowdown for higher overlap were observed for all three test cases. These results can be attributed to the increased communication for higher overlaps.

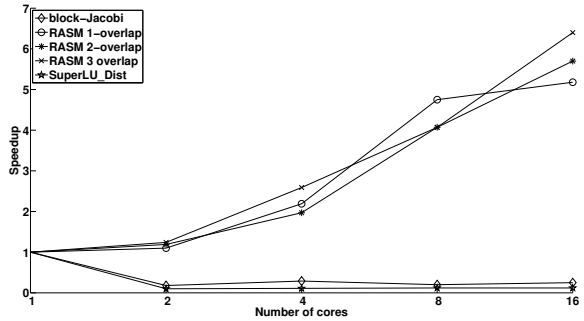

Figure 3: Speedup for case22996 with different preconditioning schemes

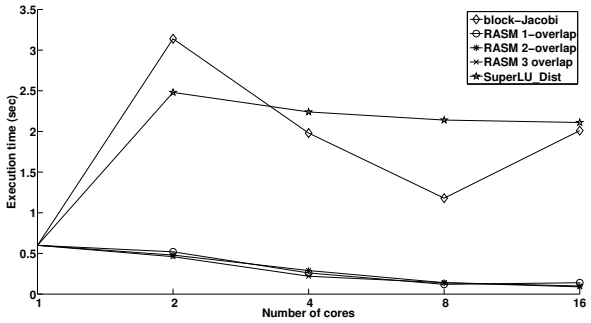

Figure 4: Execution time in seconds for case22996 with different preconditioning schemes

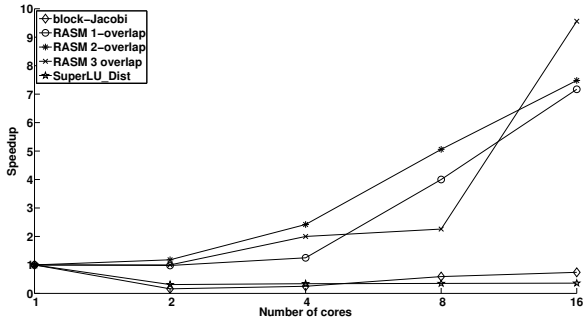

Figure 5: Speedup for case51741 with different preconditioning schemes

4.3 Convergence and Linear Solver Execution Time

Figures 9-11 present the convergence characteristics and the linear solver execution time for case22996. Similar results were observed for case51741 and case91984, hence we haven't included them. These results give an insight on the slowdown observed with the block-Jacobi preconditioner and SuperLU_Dist. For all the different linear solution schemes, the Newton method takes 3-4 iterations to converge. With the block-Jacobi preconditioner, the weakest of all the preconditioners tested, an exponential increase in the GMRES iterations was observed. Although the subdomain LU solver becomes cheaper as the number of cores increase, it is not able to compensate the total cost of the linear solver due to increased GMRES iterations. The parallel direct solution using SuperLU_Dist has the least GMRES iterations but the cost of each iteration is large, negating any speedup.

5. CONCLUSIONS

Scalable solution of very large power flow problems using iterative linear solver GMRES with an overlapping restricted additive Schwarz preconditioner (RASM) was presented in this work. An evaluation of RASM preconditioned GMRES, in terms of scalability and convergence, was presented with different amount of overlap between neighboring subdomains and presented the convergence behavior of

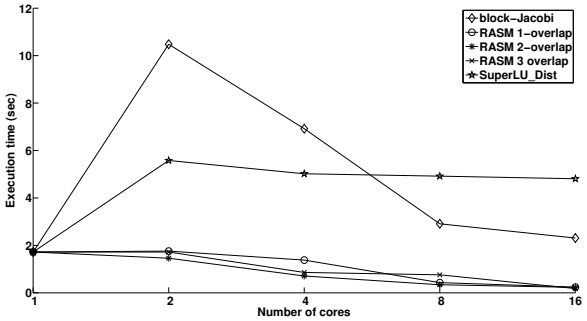

Figure 6: Execution time in seconds for case51741 with different preconditioning schemes

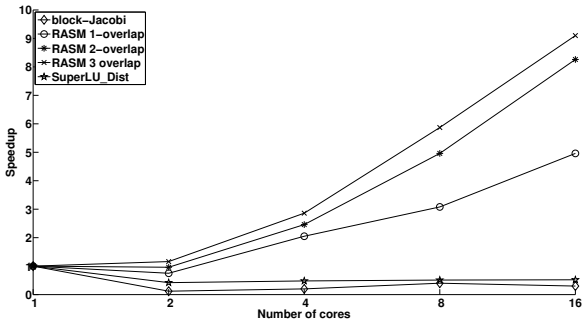

Figure 7: Speedup for case91984 with different preconditioning schemes

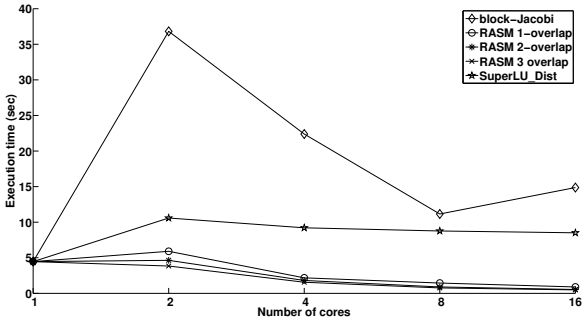

Figure 8: Execution time in seconds for case91984 with different preconditioning schemes

GMRES. Results obtained for three large test cases 22,996, 51574, and 91984 bus show good scalability with speedup ranging from 6 to 9 on 16 cores.

ACKNOWLEDGEMENTS

This work was supported by U.S. Dept. of Energy, Office of Science, under Contract DE-AC02-06CH11357.

APPENDIX

A. PETSC: PORTABLE EXTENSIBLE TOOLKIT FOR SCIENTIFIC COMPUTATION

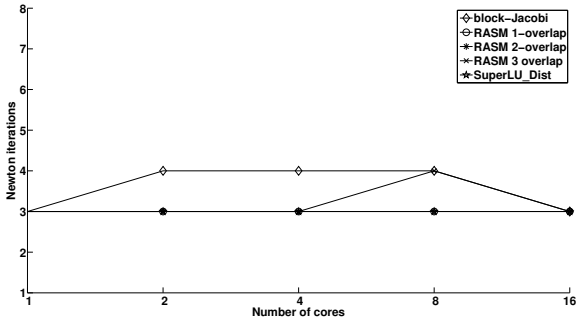

Figure 9: Total Newton iterations for case22996 with different preconditioning schemes

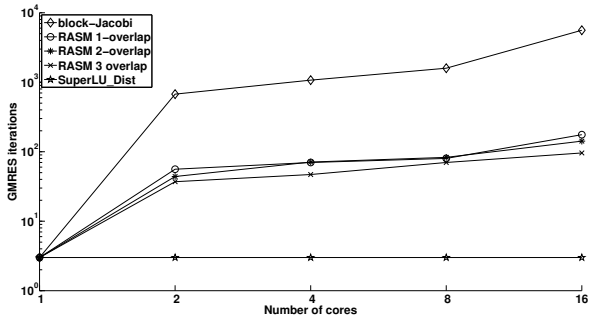

Figure 10: Total GMRES iterations for case22996 with different preconditioning schemes

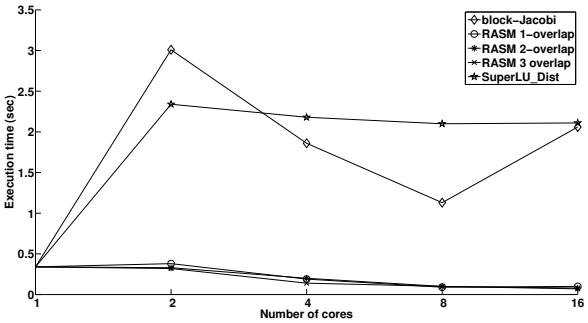

Figure 11: Linear solve time for case22996 with different preconditioning schemes

The PETSc package consists of a set of libraries for creating parallel vectors, matrices, and distributed arrays, scalable linear, nonlinear, and time-stepping solvers. A review of PETSc and its use for developing scalable power system simulations can be found in [2]. In this work, we used the linear solver GMRES and the overlapping additive Schwarz preconditioner class (ASM) available in PETSc through the *ASM* preconditioner class. The ASM preconditioner class includes several variants with the default being the *restricted additive Schwarz preconditioner*. PETSc has a plug-in architecture for third party solvers and has interfaces available for several parallel direct solvers, one of which is the package SuperLU_Dist [14] used in this work.

B. REFERENCES

[1] S. Abhyankar and A. Flueck, "Real-time power system dynamics simulation using a parallel block-jacobi preconditioned Newton-GMRES scheme," in *Proceedings of the 2nd International Workshop on High Performance Computing, Networking and Analytics for the Power Grid*. IEEE, 2012.

[2] S. Abhyankar, B. Smith, H. Zhang, and A. Flueck, "Using PETSc to develop scalable applications for next-generation power grid," in *Proceedings of the 1st International Workshop on High Performance Computing, Networking and Analytics for the Power Grid*. ACM, 2011. [Online]. Available: http://www.mcs.anl.gov/uploads/cels/papers/P1957-0911.pdf

[3] R. Bacher and W. F. Tinney, "Faster local power flow solutions: the zero mismatch approach," *IEEE Transactions on Power Systems*, vol. 4, no. 4, pp. 1345–1354, 1989.

[4] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, "PETSc users manual," Argonne National Laboratory, Tech. Rep. ANL-95/11 - Revision 3.3, 2013. [Online]. Available: http://www.mcs.anl.gov/petsc

[5] X. C-Cai and M. Sarkis, "A restricted additive Schwarz preconditioner for general sparse linear systems," *SIAM Journal of Scientific Computing*, vol. 21, pp. 792–797, 1999.

[6] X.-C. Cai and Y. Saad, "Overlapping domain decomposition algorithms for general sparse matrices," *Numerical Linear Algebra Applications*, vol. 3, pp. 221–237, 1996.

[7] X. C. Cai and O. B. Widlund, "Multiplicative Schwarz algorithms for non symmetric and indefinite elliptic problems," *SIAM Journal of Numerical Analysis*, vol. 30, pp. 936–952, 1993.

[8] T. Chan and T. Mathew, "Domain decomposition algorithms," *Acta Numerica*, pp. 61–143, 1994.

[9] T. Chan, J. R. Gilbert, and S.-H. Teng, "Geometric spectral partitioning," ftp.parc.xerox.com, 1994.

[10] S.-D. Chen and J.-F. Chen, "Fast load flow using multiprocessors," *International Journal of Electrical Power and Energy Systems*, vol. 22, pp. 231–236, 2000.

[11] V. M. da Costa and A. L. S. Rosa, "A comparative analysis of different power flow methodologies," *IEEE Transactions on Power Systems*, no. 978-1-4244-2218-0, 2008.

[12] V. M. da Costa, N. Martins, and J. L. R. Pereira, "Developments in the Newton-Raphson power flow formulation based on current injections," *IEEE Transactions on Power Systems*, vol. 14, no. 4, pp. 1320–1326, 1999.

[13] H. Dag and G. Soykan, "Power flow using thread programming," in *Proceedings of IEEE PowerTech conference*, 2011, pp. 1–5.

[14] J. Demmel, J. Gilbert, and X. Li, "SuperLU Web page," http://crd.lbl.gov/x̃iaoye/SuperLU.

[15] M. Dryja and O. B. Widlund, "Domain decomposition algorithms with small overlap," *SIAM Journal of Scientific Computating*, vol. 15, pp. 604–620, 1994.

[16] ERCOT, "Electrical buses and outage scheduling," August 2006, http://nodal.ercot.com.

[17] A. G. Exposito and E. R. Ramos, "Augmented rectangular load flow model," *IEEE Transactions on Power Systems*, vol. 17, no. 2, pp. 271–276, 2002.

[18] A. Flueck and H. Chiang, "Solving the nonlinear power flow equations with an inexact newton method using GMRES," *IEEE Transactions on Power Systems*, vol. 13, pp. 267–273, 1998.

[19] A. George and J. W. H. Liu, "A fast implementation of the minimum degree algorithm using quotient graphs," *ACM Transactions on Mathematical Software*, vol. 6, pp. 337–358, 1980.

[20] A. Gopal, D. Niebur, and S. Venkatasubramaniam, "DC power flow based contingency analysis using graphics processing units," in *Proceedings of IEEE PowerTech conference*, 2007, pp. 731–736.

[21] S. Grijalva, "Research needs in multi-dimensional, multi-scale modeling and algorithms for next generation electricity grids," in *DOE workshop on computing challenges for the next-generation power grid*, 2011.

[22] B. Hendrickson and R. Leland, "The chaco user's guide version 2.0," Sandia National Laboratories, Tech. Rep., 1995.

[23] P. Interconnection, "Pjm statistics," February 2011, http://www.pjm.com.

[24] J. Izaguirre, "An empirical evaluation of graph partitioning algorithms," Master's thesis, University of Illinois at Urbana Champaign, 1996.

[25] A. A. Kamiabad, "Implementing a preconditioned iterative linear solver using a massively parallel graphics processing units," Master's thesis, University of Toronto, 2011.

[26] G. Karypis and V. Kumar, "ParMETIS: Parallel graph partitioning and sparse matrix ordering library," Department of Computer Science, University of Minnesota, Tech. Rep. 97-060, 1997, http://www.cs.umn.edu/ metis.

[27] B. W. Kerninghan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell System Technical Journal*, vol. 49, pp. 291–307, 1970.

[28] G. Luo and A. Semlyen, "Efficient load flow for large weakly meshed networks," *IEEE Transactions on Power Systems*, vol. 5, no. 4, pp. 1309–1316, 1990.

[29] A. Pothen, H. Simon, and K. Liou, "Partitioning sparse matrices with eigenvectors of graphs," *SIAM Journal of Matrix Analysis Applications*, vol. 11, pp. 430–452, 1990.

[30] Y. Saad, *Iterative Methods for Sparse Linear Systems*. SIAM, 2000.

[31] A. Semlyen, "Fundamental concepts of a Krylov subspace power flow methodology," *IEEE Transactions on Power Systems*, vol. 11, no. 3, pp. 1528–1537, 1996.

[32] B. Smith, P. Bjorstad, and W. Gropp, *Domain decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University, 1996.

[33] B. Stott and O. Alsac, "Fast decoupled load flow," *IEEE Transactions on Power Systems*, vol. 17, no. no. 3: 859-869, 1974.

[34] J. Tate and T. Overbye, "A comparison of the optimal multiplier in polar and rectangular coordinates," *IEEE*

Transactions on Power Systems, vol. 20, pp. 1667–1674, 2005.

[35] W. F. Tinney and C. E. Hart, "Power flow solution by Newton's method," *IEEE Transactions on Power Systems*, vol. PAS-86: 1449-1456, 1967.

[36] F. Tu and A. Flueck, "A message-passing distributed-memory Newton-GMRES parallel power flow algorithm," in *Proceedings of IEEE Power Engineering Society Summer Meeting*, vol. 3, 2002, pp. 1477–1482.

[37] ——, "A message-passing distributed-memory parallel power flow algorithm," in *Proceedings of IEEE Power Engineering Society Winter Meeting*, vol. 1, 2002, pp. 211–216.

[38] J. Q. Wu and A. Bose, "Parallel solution of large sparse matrix equations and parallel power flow," *IEEE Transactions on Power Systems*, vol. 10, pp. 1343–1349, 1995.

[39] R. Zimmerman and C. Murillo-Sanchez, "Matpower 4.0 users manual," Power Systems Engineering Research Center, Tech. Rep., 2011.