SCALABILITY OF NON-GALERKIN PARALLEL ALGEBRAIC MULTIGRID

AMANDA BIENZ
WITH ROB FALGOUT, LUKE OLSON, JACOB SCHRODER

Abstract. Algebraic multigrid (AMG) is an effective iterative solver for systems of linear equations. The Galerkin product, an essential component to the setup phase of AMG, causes fill-in on the coarse grids. In parallel implementations of AMG, this added density to coarse levels yields an increase in communications costs, reducing the scalability of the method. As problem size grows, the number of levels in the hierarchy will increase, yielding increasingly large, dense levels. The communication costs can be reduced through the use of non-Galerkin coarse grids, in which unnecessary entries are removed reintroducing sparsity to the coarser levels. A study of the parallel performance of GMRES preconditioned by non-Galerkin shows a consistent decrease in solve times for a three dimensional Laplace problem. The setup times show that any added cost to this phase is masked by the reduction in communication, and the overall time spent in the setup phase is often also reduced. Further tests show increasing the tolerance for collapsing entries on coarser grids has added benefits of removing the larger percentage of unnecessary entries occurring further in the hierarchy. Retaining symmetry of coarse grids allows use of more competitive methods such as CG or its three-term analogue for indefinite matrices, minimal residual method.

1. Introduction.

Sparse matrix problems are abundant in parallel. Iterative methods solve sparse problems with a complexity bound dependent on that of a sparse matrix-vector multiply. Multigrid methods iterate through hierarchies containing varying levels of sparsity and patterns, each impacting the overall performance. The Galerkin coarse product, $A_c = P^T AP$ creates fill-in on coarse grids, decreasing the efficiency of matrix-vector products throughout the hierarchy.

Multigrid is composed of two phases: setup and solve. The setup phase creates a hierarchy of levels, of which each level l contains a coarse matrix A_l, an interpolation operator P_l, and a restriction operator $R_l = P_l^T$. This process is highly dependent on the Galerkin product $A_{l+1} = P_l^T A_l P_l$. The solve phase uses this hierarchy to iteratively approach a solution through smoothing and coarse grid correction [1, 2, 4, 7].

There are currently multiple approaches that improve the efficiency of this method in parallel. The combination of aggressive coarsening strategies, such as HMIS and PMIS, and distance-two interpolation significantly reduces the complexity of coarse grids. Investigating the Galerkin product RAP leads to a further reduction in coarse grid cost. The method of non-Galerkin coarse grids requires forming the coarse grid RAP before manually removing unnecessary entries [3, 11, 8, 4].

The focus of this paper is on the parallel aspects of the non-Galerkin coarse grid approach. The contributions include a performance study of the parallel implementation, a study of setup costs, and an analysis of the effects of varying drop tolerances as well as symmetric collapsing. Non-Galerkin coarse grids are highly dependent on the selection of drop tolerance. If too many entries from a coarse grid are dropped, non-Galerkin AMG will diverge. However, if too few entries are collapsed, there is little to no reduction in sparsity yielding a minimal benefit. Initial tests in the performance study require all coarse levels to drop values with one given tolerance. However, studies have shown that the finer levels in a hierarchy require an accurate approximation to the Galerkin matrix, indicating only a small number of nonzeros can safely be removed. Furthermore, the density increases with coarseness, indicating the drop tolerance should increase with the coarseness.

This paper is organized as follows. The remainder of this section describes the method of non-Galerkin multigrid as well as predicted benefits. Section 2 displays the performance benefits of parallel non-Galerkin multigrid with various options such as varying drop tolerances and symmetric dropping. This section also investigates the reason behind any reduction in solve times. Section 3 describes future work to improve the efficiency as well as the applicability of this method.

1.1. Parallel Setup and Coarse Level Fill-in. Parallel implementations of AMG on distributed-memory systems require the distribution of the matrix A across processors. Each processor stores a section of contiguous rows of both the matrix A and all vectors x, r, and e. The local entries in A are split into a diagonal block, which contains the columns corresponding to local vector entries, and two off-diagonal blocks, each holding values corresponding to the vector entries stored on other processors. Figure 1 displays the partitioning of the rows stored on some processor k. Every nonzero entry stored in an off-diagonal block must obtain the corresponding vector entry from another processor, implying the number of elements in off-diagonal blocks is directly correlated to the amount of communication required. Throughout AMG,

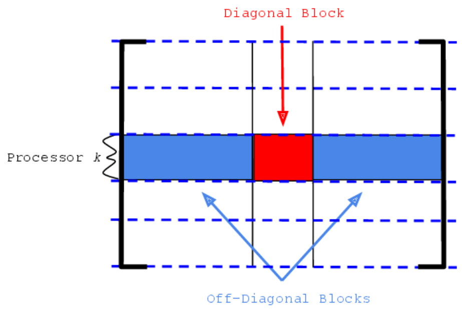


Fig. 1: Partitioning of matrices in distributed-memory systems

the smoothing and coarse grid correction are performed on problems of varying levels of coarseness. The matrix on some coarse level $l+1$, is obtained through the Galerkin product, an operation that preserves both the symmetry and positive-definiteness of a problem, but also creates fill-in on the coarser matrix. Figure 2 displays the decrease of sparsity on coarse grids for a three-dimensional Laplace problem with one-million degrees-of-freedom. While the problem size on coarser levels reduces rapidly, the fill-in resulting from the Galerkin product obstructs the reduction in communication costs. Figure 3 shows the amount of time spent on each level during a single v-cycle when using Galerkin coarse grids on both classical and best practices parallel AMG. Classical AMG uses Falgout coarsening and classical interpolation while best practices AMG aggressively coarsens with HMIS combined with extended classical interpolation. In classical AMG, more time is spent on some coarse grids than on the finest grid due to an increase in communication resulting from the coarse grid fill-in. Best practices yields a less severe, but still noticeable, increase in communication time [4].

1.2. Non-Galerkin Method. A method of further reducing coarse grid fill-in requires initially building the Galerkin coarse grids and then manually removing unnecessary nonzeros to reintroduce sparsity. This process of creating a non-Galerkin coarse grid is broken into two parts: creating a sparsity pattern to determine essential nonzeros and removing all entries not in this sparsity pattern. The minimal sparsity

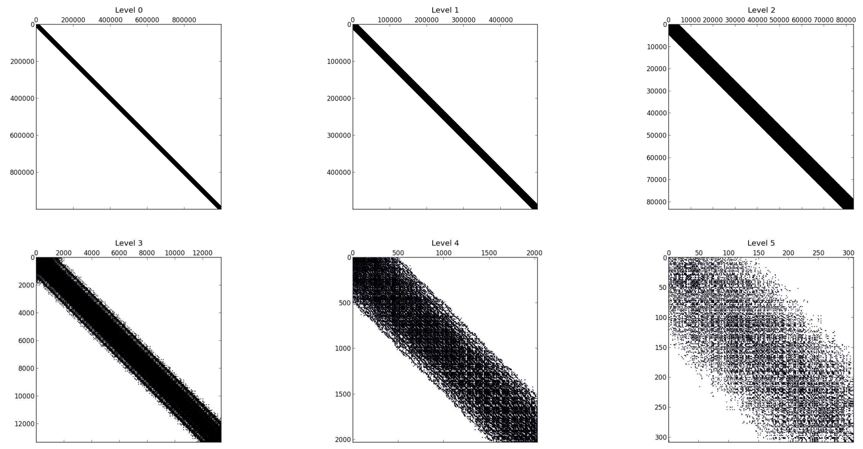


Fig. 2: The sparsity of coarse levels in a 3 dimensional Laplace hierarchy

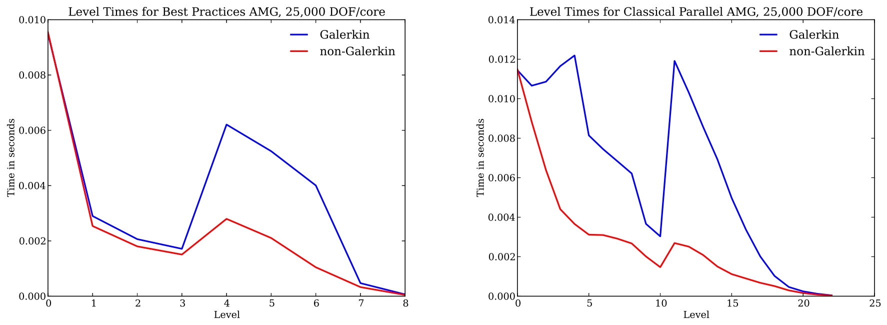


Fig. 3: Time spent on each level of hierarchy during a single v-cycle

pattern is equal to $P_I^T AP + P^T AP_I$, where P_I is the injection operator. The initial sparsity pattern of each row i, N_{ci}, is equal to the sparsity pattern of row i of A. Given some tolerance, entries are removed from N_{ci} while

$$2 \sum_{j \notin N_{ci}} \left| a_{ij}^g \right| \leq \gamma \sum_j \left| a_{ij}^g \right| \tag{1.1}$$

holds. After entries are removed, the remaining entries in the minimal sparsity pattern and reintegrated in N_{ci} [3].

Once the sparsity pattern has been created, all entries appearing in A but not N_c are removed. The values of dropped entries are lumped to strong neighbors based on the strength of connection matrix S. If some dropped value a_{ij} has no strong connections, the entire value is added to a_{ii}. Otherwise, a percent of a_{ij} is added to each strong connection. So, if σ is defined to be the absolute sum of all values in row j of S, for each column k in S_j, the percent $\frac{|s_{jk}|}{\sigma}$ of a_{ij} is added to a_{ik} [3].

This method was initially tested on multiple sequential problems, including two-dimensional rotated anisotropic diffusion, three-dimensional Laplace, and jumping coefficient problems. All tested problems showed a reduction in stencil size with little, if any, decrease in convergence rate for some drop tolerance γ, indicating this method should work well for a variety of problems in parallel.

2. Parallel Performance.

Non-Galerkin parallel AMG was tested using hypre [5] on the intel cluster Sierra at Lawrence Livermore National Lab using drop tolerances of 0.01 and 0.03. A weak scaling performance study was performed on various problem sizes with both classical and best practices parallel AMG. Classical AMG uses Falgout coarsening and classical interpolation while best practices AMG aggressively coarsens via HMIS combined with extended classical interpolation. Figures 5a and 5b display the solve times for both Galerkin and non-Galerkin coarse grids for GMRES preconditioned with classical parallel AMG on problems of size 10,000 and 25,000 degrees-of-freedom per core, respectively. Figures 5c and 5d show solve times for equivalent problems solved by GMRES preconditioned with best practices AMG.

For every test case the non-Galerkin coarse grids showed a reduction in the time spent in the solve phase. This is explained by the work per digit of accuracy (WPD) and stencil sizes resulting from dropping insignificant entries. WPD is defined as the operator complexity divided by the negative log of the convergence rate, and shows the amount of work required throughout all levels to gain one digit of accuracy. The stencil size is the number of nonzeros on any given row, implying a correlation between large stencil size and the amount of communication. For both drop tolerances, the WPD and stencil sizes are lower for the non-Galerkin coarse grids than the original. This reduction in both the communication and computation costs can explain the speedup of the non-Galerkin coarse grids.

While there is an added cost in the setup phase associated with collapsing entries, enough sparsity is added to coarse grids to mask this cost. Figure 4 displays the setup phase times for each test case. The currently unoptimized non-Galerkin code reduces the overall setup times for classical parallel AMG and adds little time to best practices AMG.

Coarse Level	1	2	3	All Remaining
Series 1	0.0	0.01	0.03	0.1
Series 2	0.01	0.03	0.1	0.1

Table 1: Series of Drop Tolerances

2.1. Varying Drop Tolerances.

The percent of fill-in on any grid increases with the level of coarseness, as displayed in Figure 2. While very few entries should be dropped from the fine grids, a larger percent can be removed as the levels become coarser. The previous non-Galerkin results can be improved upon by increasing the drop tolerance on coarser levels. Two series of drop tolerances, described in Table 1, consistently outperformed the original drop tolerances for the given problem, as shown in Figure 5. This further reduction in solve time is again explained by a greater reduction in both WPD and stencil sizes for both series of drop tolerances.

2.2. Symmetric Collapsing.

While non-Galerkin coarse grids yield a reduction in solve time for GMRES preconditioned with AMG, CG preconditioned with Galerkin AMG still outperforms all instances. The current non-Galerkin coarse grids do not meet the symmetric positive-definite criteria required to precondition CG. To retain

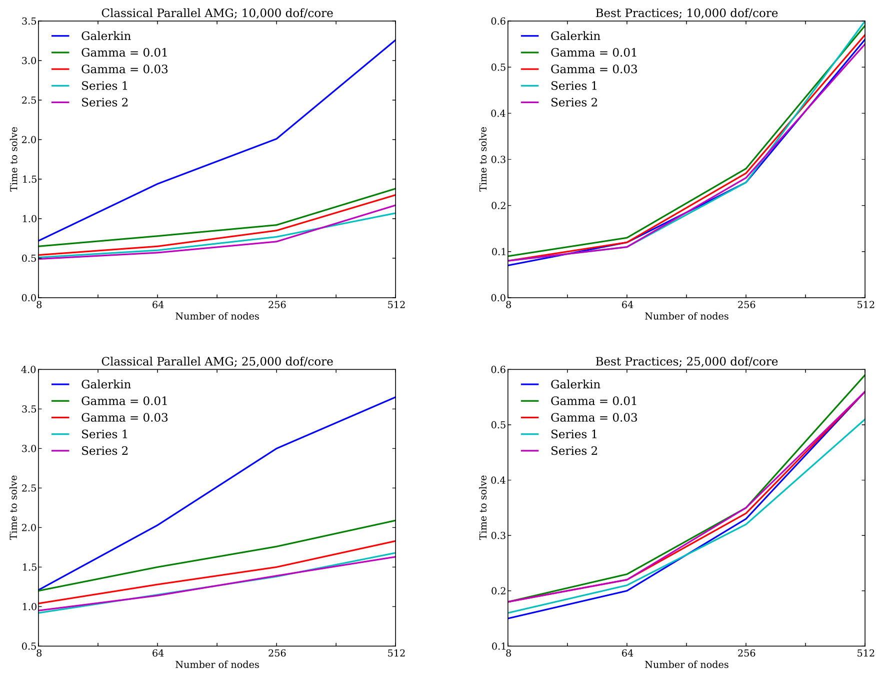


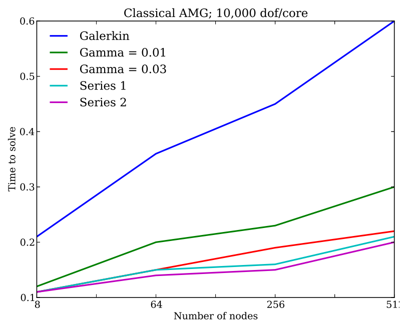
Fig. 4: Times required by setup phase

the symmetry of Galerkin coarse grids, the nonzero entries falling below tolerance must be collapsed symmetrically.

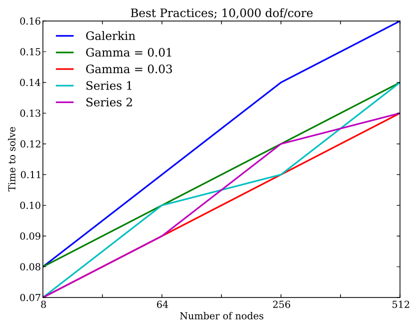
For guaranteed symmetry of non-Galerkin coarse grids, the sparsity pattern N_c must be symmetric. For any dropped entry a_{ij}, if some percent of the value is added to a strong connection a_{ik}, this must also be added to a_{ki}. However, to retain constant row sums, this portion of the value must also be subtracted from the diagonal, a_{kk}. While this will always result in a symmetric sparsity pattern, subtracting values from a_{kk} has the potential to flip an eigenvalue, yielding a symmetric indefinite matrix.

For the given problem, this method retains both symmetry and positive-definiteness of the coarse grids, yielding solve times for CG preconditioned with AMG as shown in Figures 6. The non-Galerkin method yields an improvement in the solve time for CG when the entries are collapsed symmetrically. However, because the symmetric dropping process does not guarantee that positive-definiteness will be retained, CG may not always be a viable option. The minimum residual method, a three-term analogue for symmetric indefinite matrices, can be used as an alternative [6].

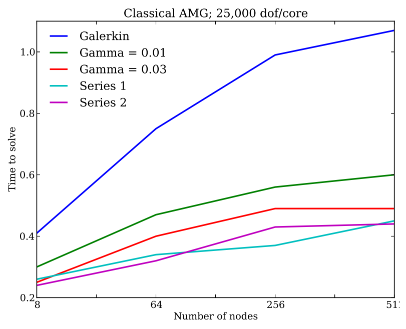
2.3. Per-Level Improvement. The non-Galerkin coarse grid method improves the solve phase time by reducing the fill-in on coarse grids. This diminishes the extra communication cost originally required by coarse levels. Figure 7 displays the time spent on each level of the hierarchy during a single v-cycle. The time spent on each non-Galerkin level near the middle of the hierarchy is significantly less than the equivalent Galerkin levels. There is still a small increase in communication on coarse grids, as all of the fill-in is not removed. However, the non-Galerkin coarse grids show a large improvement in the amount of increased communication.



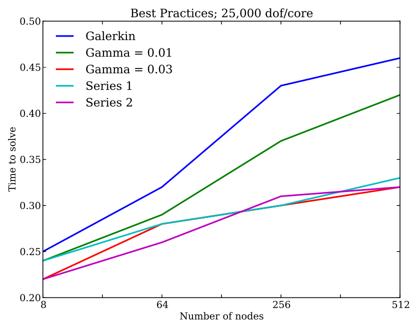
(a)



(b)



(c)



(d)

Fig. 5: Solve Times for GMRES Preconditioned with AMG

2.4. Communication Reduction. The reduction in communication costs comes from two different sources: the number of messages sent and the size of each message. The amount of communication is tracked on each level through the maximum amount of communication performed among all of the processors. The maximum number of sends and receives on any processor approximates the cost of communication due to the number of messages. The maximum size of the data sent by any processor estimates the other component of communication cost: the size of the messages being sent. Figures 8, 9, 10, and 11 display the reduction in communication costs from non-Galerkin coarse grids. These figures show that for both classical and best practices parallel AMG, the reduction in communication cost is due to a significant reduction in the number of messages sent on the coarse grids.

3. Future Work. Non-Galerkin coarse grids reduce the costs of both the setup and solve phases when optimal drop tolerances are chosen for the three dimensional Laplace problem. However, there remains room for improvement in both the reduction in cost as well as the applicability of the strategy. As shown in Figure 7, there remains an increase in communication costs on coarse grids for all drop tolerances. This could be further reduced by adjusting the drop rate of entries based on the physical properties of the cluster. Dropping entries in the diagonal block has no impact on

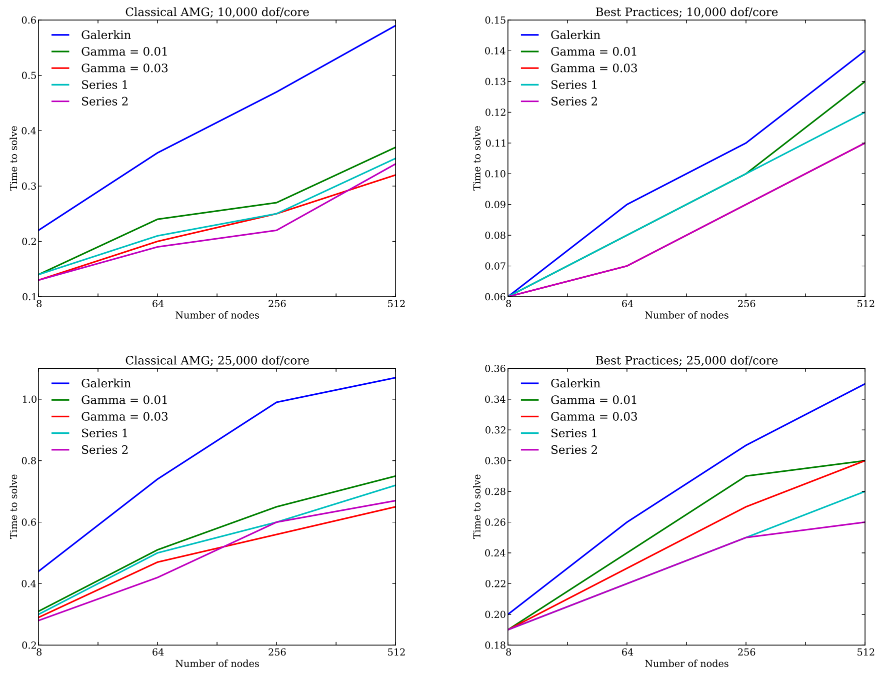


Fig. 6: Solve Time for CG preconditioned with AMG

communication, so the off-diagonal blocks will be targeted more heavily. Similarly, entries within the off-diagonal blocks will be weighted based on their physical distances from a given processor. Assigning a higher drop rate to entries lying on distant processors than those on neighboring nodes has potential to reduce the cost of the messages being sent.

This method will also be tested with more complex problems. Serially, promising results were shown for a variety of test problems. However, both the optimal series of drop tolerances as well as the overall benefit of non-Galerkin levels varied for the different test cases. Testing these in parallel will display the benefits of this method for a wider variety of problems.

The optimal series of drop tolerances varies with problem type, yielding the need for a method of determining these values. For non-Galerkin to be widely applicable, the drop tolerances should be chosen automatically by the program. This can be done based on the increase on stencil size from one level to the next. If the stencil size increases by a large amount, the drop tolerance can be increased. Removing the requirement of selecting a set of drop tolerances would improve the usability of this method.

Lastly, theoretical convergence bounds for M-matrices need to be determined. The convergence of the non-Galerkin method should have a bound based on the convergence of the Galerkin AMG. This theory would guarantee convergence of the method, and could possibly be extended to general matrices as well.

While there is room for improvement, the use of non-Galerkin coarse grids is currently an effective method for improving the scalability of parallel AMG. The drop

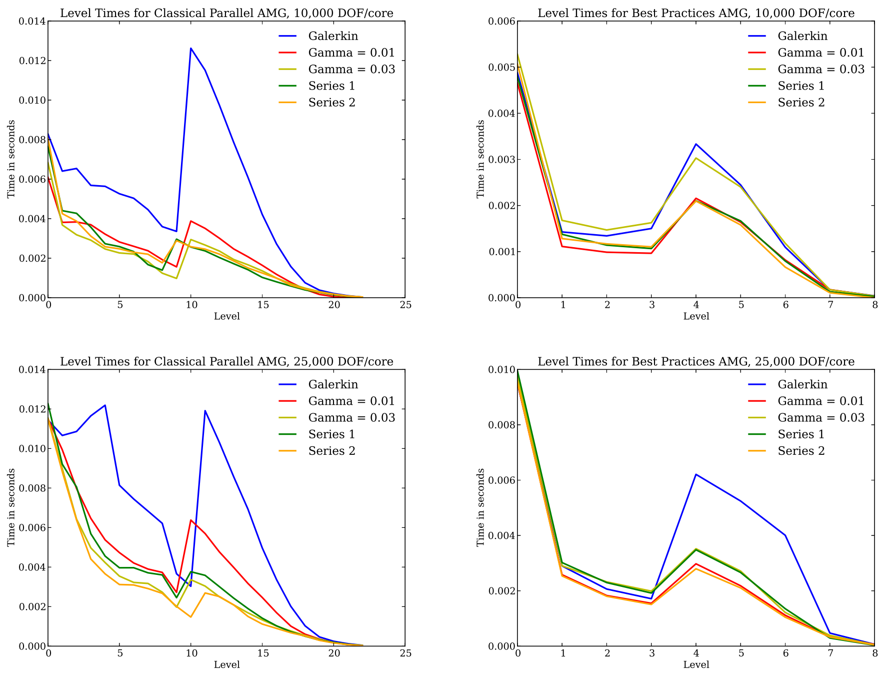


Fig. 7: Time spent on each level of the hierarchy during one v-cycle

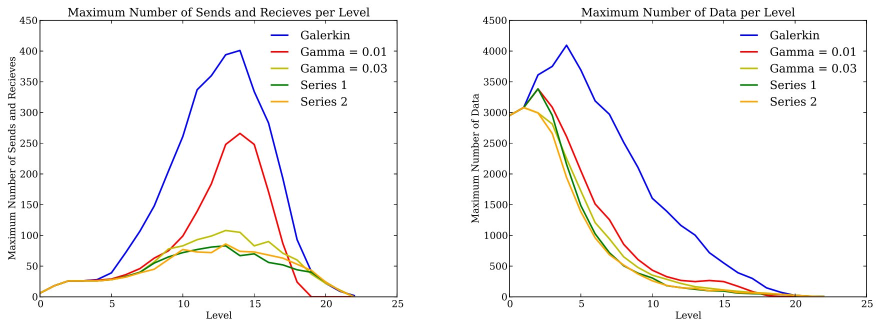


Fig. 8: Communication Costs; 10,000 dof/core; Classical Parallel AMG

tolerances for this method can be adjusted to obtain similar convergence to Galerkin coarse grids while also significantly inducing sparsity.

REFERENCES

[1] W.L. Briggs, V.E. Henson, and S.F McCormick. *A multigrid tutorial*, SIAM, Philadelphia, PA, USA, 2nd edition, 2000.

[2] R.D. Falgout. An Introduction to Algebraic Multigrid. *Computing in Science and Engineering*, Special Issue on Multigrid Computing, 8 (2006), pp. 24-33. UCRL-JRNL-220851.

[3] R. D. Falgout and Jacob Schroder. Non-Galerkin Coarse Grids for Algebraic Multigrid ,

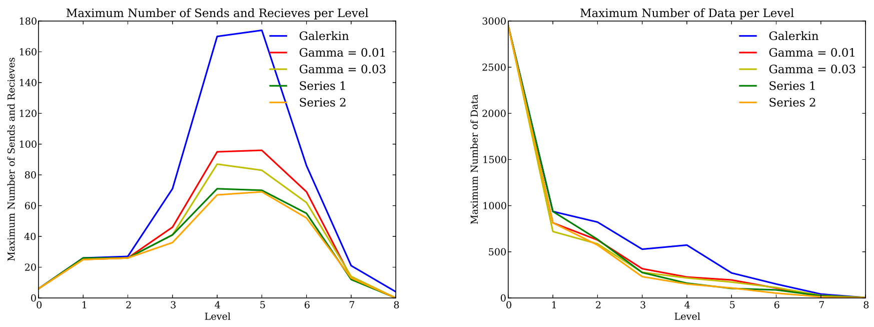


Fig. 9: Communication Costs; 10,000 dof/core; Best Practices AMG

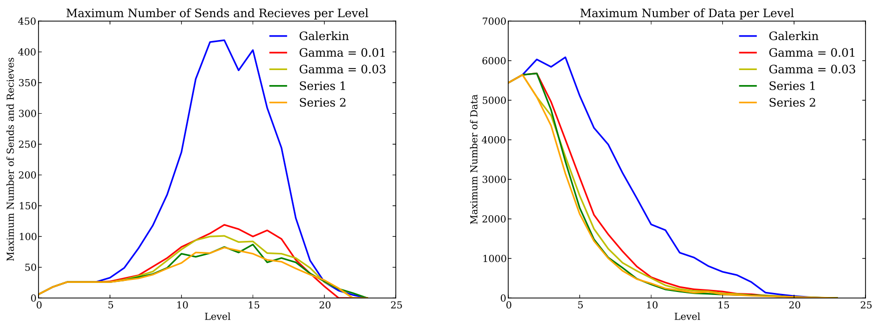


Fig. 10: Communication Costs; 25,000 dof/core; Classical Parallel AMG

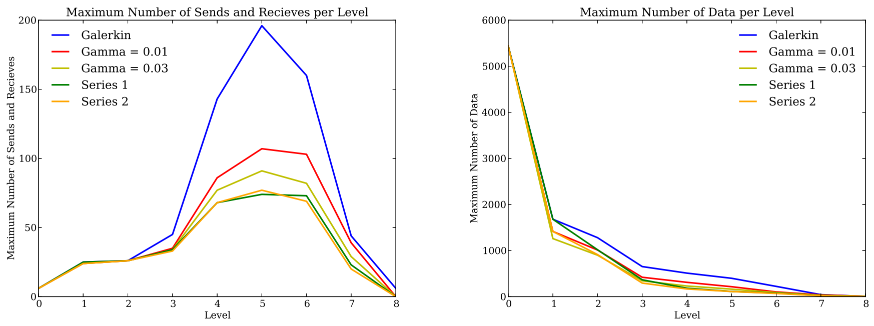


Fig. 11: Communication Costs; 25,000 dof/core; Best Practices AMG

SIAM J. Sci. Comput., (submitted). LLNL-JRNL-641635.

[4] V. Henson and U. Yang. BoomerAMG: a parallel algebraic multigrid solver and preconditioner. *Applied Numerical Mathematics*, 41:155-177, 2002.

[5] *hypre:* High performance preconditioners. http://www.llnl.gov/CASC/hypre.

[6] C.C. Paige, M.A. Saunders Solution of Sparse Indefinite Systems of Linear Equations, *SIAM J. Numer. Anal.*, 12 (1975), pp. 617-629.

[7] J.W. Ruge and K. Stüben. Algebraic multigrid (AMG). In S.F. McCormick, editor, *Multigrid Methods*, Frontiers Appl. Math., pages 73-130. SIAM, Philadelphia, 1987.

[8] H. De Sterck, R.D. Falgout, J.W. Nolting, and U.M. Yang Distance-Two Interpolation for Parallel Algebraic Mutigrid, *Numer. Linear Algebra Appl.*, Special Issue on Multigrid Methods, 15(2008), pp. 115-138. UCRL-JRNL-230844.

[9] E. Treister. Sparsified coarsening multigrid for convection diffusion, 2013. Student paper at the Copper Mountain Conference on Multigrid Methods, Copper Mountain, CO, March 17-22.

[10] U. Trottenberg, C. Oosterlee, and A. Schüller *Multigrid*. Academic Press, London, UK, 2001.

[11] U. M. Yang, On Long Range Interpolation Operators for Aggressive Coarsening *Numer. Linear Algebra Appl.,* Special Issue on Multigrid Methods, 17 (2010), pp. 453-472. LLNL-JRNL-417371.