AN OVERVIEW OF THE CARP-CG ALGORITHM

Dan Gordon
Dept. of Computer Science, University of Haifa,
Haifa 31905, Israel. gordon@cs.haifa.ac.il

Rachel Gordon
Dept. of Aerospace Engineering, the Technion,
Haifa 32000, Israel. rgordon@tx.technion.ac.il

Abstract

Recent work shows that the block-parallel CARP-CG algorithm [Parallel Computing 36, 2010] is extremely effective on sparse nonsymmetric linear systems with very small diagonal elements, including cases with discontinuous coefficients. In contrast to most known solvers, the effectiveness of CARP-CG often improves as the diagonal elements become smaller. This property is shown to follow from the foundations on which CARP-CG is based. The unique behavior of CARP-CG is demonstrated with some old and new results. Previously studied problems consist of convection-dominated elliptic PDEs, with and without discontinuous coefficients. New results using high-frequency Helmholtz equations in the heterogeneous Marmousi model, discretized with 2nd, 4th and 6th order finite difference schemes, indicate the insufficiency of low-order schemes and the good parallel scalability of CARP-CG at high frequencies.

1 Introduction

Some of the most difficult problems in numerical linear algebra consist of solving sparse nonsymmetric linear systems in which the off-diagonal elements are very large compared to the diagonal elements. Such problems arise in the discretizations of several types of partial differential equations (PDEs), such as convection-diffusion equations in which the convection terms are relatively very large compared to the diffusion term. Another type of problems which gives rise to large off-diagonal elements is the Helmholtz equation at high frequencies.

Compounding the situation is the fact that the underlying physical domain is often highly heterogeneous, giving rise to large differences between the coefficients—so-called "discontinuous coefficients". Such problems are often handled using domain decomposition (DD), but this approach may be difficult to implement on unstructured grids or if subdomain boundaries are irregular. Note, however, that recent advances in FETI DD methods can also handle highly heterogeneous problems in some cases—see [5, 16, 17].

Another difficulty, specific to the Helmholtz equation, is the so-called "pollution effect" [1]. The numerical significance of this effect is that when the frequency increases, the grid must be refined even beyond the simple requirement of keeping a fixed number of grid points per wavelength. It is generally considered that the number of grid points per wavelength, denoted N_g, should be at least 8–12. However, the pollution effect causes the relation between the wave number k and the number of intervals per domain side, N, to be non-linear: N is proportional to $k^{(p+1)/p}$, where p is the order of the accuracy of the scheme; see [6]. Therefore, higher order schemes are clearly advantageous at high frequencies.

The usual approach to solving problems with large off-diagonal elements often involve reorderings, with the aim of placing the large elements of the system matrix on the diagonal [2]. This may not always be helpful when there are four or more large off-diagonal elements in each equation.

In recent work, we obtained very good results on such problems with our block-parallel CARP-CG algorithm [10]. This algorithm seems to "thrive" on small diagonal elements; in fact, the smaller the diagonal, the better it usually performs. In addition, discontinuous coefficients can be handled without any problems.

CARP-CG is described in §2, and §3 explains its behavior on problems with discontinuous coefficients and small diagonal elements. §4 presents sample results from previous work for convection-dominated PDEs, and new results for high-frequency Helmholtz equations in the heterogeneous Marmousi model [15], using 2nd, 4th and 6th order finite difference schemes.

2 Description of CARP-CG

We present a brief and informal explanation of CARP-CG; for full details, see [10]. Consider a system of m linear equations in n variables, $Ax = b$. The Kaczmarz algorithm (KACZ) [14] is fundamental to CARP-CG. Starting from some arbitrary initial guess, KACZ successively projects the current iterate onto a hyperplane defined by one of the equations in cyclic order. Each cycle of projections is called a KACZ sweep. The projections can be modified with a relaxation parameter $0 < \omega < 2$. If equation i has a fixed relaxation parameter ω_i, then the projections are said to be done with cyclic relaxation.

In a landmark paper, Björck and Elfving [3] showed that if a forward sweep of KACZ is followed by a backward sweep (i.e., the projections are done in reverse order), then the resulting iteration matrix is symmetric and positive semi-definite. Therefore, the double KACZ sweep can be accelerated using the conjugate-gradient (CG) algorithm. The iteration matrix is not actually calculated; instead, whenever it has to be multiplied by a vector, the calculation is done by performing a suitable KACZ double sweep. The resulting algorithm, called CGMN, is one cornerstone of CARP-CG.

A second cornerstone for CARP-CG is CARP [8], which is a block-parallel version of KACZ. Note that this is different from the standard block-sequential version of KACZ, which requires the equations in a block to be independent (i.e., there are no shared variables in a block). CARP divides the equations into blocks, which may overlap, and equations in a block need not be independent. In a parallel setting, every processor is in charge of a block of equations. Every processor has a copy, or "clone", of every variable that it shares with another block. The following two steps are now repeated until convergence:

1. Every processor performs a KACZ sweep on the equations of its assigned block, updating the block's variables. For shared variables, each processor updates its copy of the variable.
2. The processors exchange information about the new values of the clones. Every shared variable is now updated to be the average of all its clones in the different blocks, and the new value of every shared variable is distributed among the processors which share it.

If the blocks are chosen according to spatial domains, then the exchange of data occurs only at the boundaries between domains. CARP can be viewed as a type of DD technique which

differs from other DD methods because values from neighboring domains (the clones) are also modified by the internal processing. Furthermore, partitioning a domain into subdomains can be based entirely on efficiency considerations and not on differences between coefficients.

For a detailed parallel implementation of CARP, see [8]. An important point about CARP is that the averaging operations between shared variables are equivalent to certain KACZ row projections (with relaxation $\omega = 1$) in some superspace. Hence, CARP is equivalent to KACZ with cyclic relaxation in the superspace. This property provides a convenient convergence proof for CARP, and it enables the CG acceleration of CARP.

The CG acceleration is obtained in [10] as follows. Firstly, the construction of CGMN is extended to KACZ with cyclic relaxation parameters. It then follows that the superspace formulation of CARP can be accelerated by CG. This means that in the regular space, CARP can be accelerated by CG in the same manner as CGMN, i.e., by running CARP in a double sweep. On one processor, CARP-CG and CGMN are identical.

3 On the Effectiveness of CARP-CG

The effectiveness of CARP-CG follows from two inherent features of KACZ, which will be explained below. Let $Ax = b$ be the given linear system. We denote the L_2-norm of a vector x by $\|x\|$. Consider the projection of the current iterate x onto the hyperplane defined by the ith equation, yielding a new iterate x' defined by

$$x' \;=\; x \;+\; \omega_i \,\frac{b_i - \langle a_{i*},\, x\rangle}{\|a_{i*}\|^2}\, a_{i*}, \tag{1}$$

where $0 < \omega_i < 2$ is the relaxation parameter associated with the ith equation and a_{i*} is the ith row of A. Eq. (1) shows that KACZ inherently normalizes the equations, i.e., the ith equation is divided by $\|a_{i*}\|$. Therefore, we can assume that this normalization has already been done as a preliminary step; it is also more efficient to do so in practice. Note that KACZ is a geometric algorithm in the following sense: the iterates depend only on the hyperplanes defined by the equations and not on any particular algebraic representation of these hyperplanes. For this reason, we call the normalization GRS (geometric row scaling).

This inherent use of GRS by KACZ is the reason that CARP-CG can handle discontinuous coefficients effectively. GRS is a diagonal scaling of A, and the positive effect of such scalings, when dealing with discontinuous coefficients has been known for a long time; see, for example, [22]. More recently, an extensive study in [11] has shown that normalization significantly improves the convergence properties of Bi-CGSTAB [21] and restarted GMRES [19] (both of them with and without the ILU(0) preconditioner) on nonsymmetric linear systems with discontinuous coefficients. However, for these algorithm/preconditioner combinations, these results held only for small- to moderate-sized convection terms; For large convection terms, [9] shows that CARP-CG is preferable.

An examination of the normalization in [11, 9] shows that it "pushes" heavy concentrations of eigenvalues around the origin away from the origin. It is known that such concentrations are detrimental to convergence. These results explain why CARP-CG can handle the problem of discontinuous coefficients. Note that the normalization has a certain "equalization" effect

on the coefficients of different equations (but not on different-sized coefficients in the same equation).

A second point concerning KACZ is the fact that it is actually SOR (successive over-relaxation) on the normal equations system $AA^Ty = b$ ($x = A^Ty$). Using AA^T is generally considered detrimental because its condition number is the square of the condition number of A. However, a fact that seems to have been overlooked is that if A is normalized first (as is inherent in KACZ), then the diagonal elements of AA^T are equal to 1, and all the off-diagonal elements are < 1, provided no two rows of A are colinear [9, Thm. 3.1]. In other words, the two simple operations of normalizing the equations and then using AA^T, form a very simple means of dealing with systems with large off-diagonal elements, including cases of discontinuous coefficients.

Additionally, as the diagonal element of A decreases, the sum of the off-diagonal elements of AA^T also decreases. To see this, consider, a simplified case of a 5-point stencil matrix A obtained with 2nd order finite difference scheme on a regular grid. Assume that A has been normalized, and has a small d on the diagonal and a off the diagonal. In AA^T, the diagonal element is $4a^2 + d^2 = 1$. By considering all possible placements of a cross of 5 grid points w.r.t. another cross, it is easy to see that in a general setting, the sum of the absolute values of the off-diagonal elements of AA^T is $S = 12a^2 + 8|ad|$. Substituting $a = \sqrt{1 - d^2}/2$ into S, we get $\lim_{d\to 0} S'(d) = 4 > 0$, so $S(d)$ decreases as $d \searrow 0$.

4 Experimental Results and Discussion

All the problems were tested on a Linux cluster of 16 PCs, using MPICH for message passing. The initial estimate was taken as $x_0 = 0$, and convergence goals were based on setting a tolerance for the relative residual: $\|Ax - b\|/\|b\| < \epsilon$, with the equations always taken as normalized. In the tables, timings are in seconds and dashes indicate no convergence or impractically slow convergence.

4.1 Convection-Dominated PDEs

We present illustrative results from [10, 11, 9] for three convection-dominated elliptic PDEs. On these problems, we also tested Bi-CGSTAB, restarted GMRES (restart=10), and CGNR. These algorithms were tested with and without GRS, and Bi-CGSTAB and GMRES were also tested with and without the ILU(0) preconditioner. Note that as shown in [9], CGNR+GRS is actually identical to the CG-acceleration of the Cimmino algorithm [4].

4.1.1 Problem 1

The PDE for this problem is $\Delta u - \partial(De^{xy}u)/\partial x - \partial(De^{-xy}u)/\partial y = F$, where $\Delta u = u_{xx} + u_{yy}$. Two values of D were considered: $D = 10$ and $D = 1000$. The purpose of this example is to demonstrate how CARP-CG's convergence behavior improves as the convection term is increased. This problem originally appeared in [18, §3.7, Problem F3D] with $D = 10$, and also in [10] as Problems 8 and 9. The right-hand-side was created artificially by first computing A, and then computing $b = Av$, where $v = (1, 1, \ldots, 1)^T$. Problem 1 was solved on the unit

square $[0,1]^2$, using a 2nd order finite difference scheme on a grid of 80^2, resulting in 64,000 equations.

Table 1 shows the runtimes to achieve a convergence goal of 10^{-7} for the two values of the convection term D, for the tested algorithms on the normalized equations. The un-normalized equations gave very similar results, because there were no jumps in the coefficients. We can see that ILU(0) was useful for Bi-CGSTAB and GMRES with $D = 10$, but not with $D = 1000$. Also, CARP-CG is similar to GMRES with $D = 10$, but its convergence improves significantly with $D = 1000$, both absolutely and relatively to the others. Note that CGNR also performed reasonably well with the higher convection.

Convection:	D=10	D=1000
Bi-CGSTAB+GRS	6.7	—
Bi-CGSTAB+GRS+ILU(0)	3.3	—
GMRES+GRS	14.4	6.6
GMRES+GRS+ILU(0)	3.5	—
CGNR+GRS	—	5.0
CARP-CG	15.0	1.9

Table 1: Problem 1: time (sec.) to achieve convergence to 10^{-7} on 16 processors.

4.1.2 Problem 2

This problem originally appeared in [7, §5.1] as one of three related problems, which were solved there using DD. They were also examined in [9, §5]. The problem is a 3D convection-diffusion-reaction equation with discontinuous coefficients. The domain consists of the unit cube divided into two subdomains by the plane $x = \frac{1}{2}$. The PDE is $-\text{div}(\nu(x)\nabla u) + u_x + 3u_y + 5u_z + u = 0$, where $\nu(x) = 10^{-1}$ for $x < \frac{1}{2}$ and $\nu(x) = 10^{-5}$ otherwise.

Dirichlet boundary conditions are taken as $u = 1$ on the $z = 0$ plane and $u = 0$ on the other boundaries of the unit cube. The small value of ν on half the cube results in an equation that is strongly convection dominated.

Fig. 1 shows the effect of GRS on the eigenvalue distribution, on a grid of $12 \times 12 \times 12$. As can be seen, GRS "pushed" the eigenvalues away from the origin.

Table 2 shows the time (on one processor) to converge to 10^{-4}, 10^{-7} and 10^{-10}, for all cases that achieved at least one goal. These results were obtained on a grid of 80^3. We can see that Bi-CGSTAB has no entries at all in the table, and ILU(0) was no help. Note that GRS improved the convergence properties of GMRES and CGNR. On this problem, CGMN (CARP-CG on one processor) has a very strong lead over GMRES+GRS and CGNR+GRS.

4.1.3 Problem 3

This problem, which appeared in [9, §7] and [11, §7], is based on a 3D example from [13], with additional convection terms. The domain is the unit cube $[0,1]^3$, and the differential equation is $-\text{div}(a\nabla u) + D\nabla u = 0$, where $a(x,y,z) = 10^4$ for $\frac{1}{3} < x,y,z < \frac{2}{3}$, and $a(x,y,z) = 1$

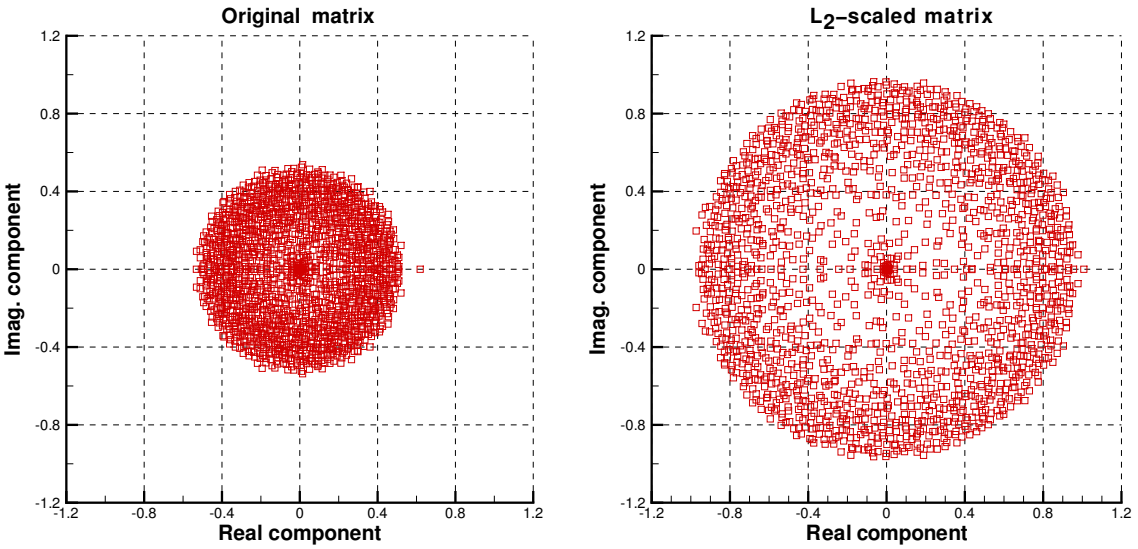

Figure 1: Problem 2: eigenvalue distribution for the original and the scaled matrices.

Convergence goal:	10^{-4}	10^{-7}	10^{-10}
GMRES	1388	—	—
GMRES+GRS	609	1126	—
CGNR	1179	—	—
CGNR+GRS	246	444	608
CGMN (CARP-CG)	67	120	188

Table 2: Problem 2: time (sec.) to achieve three convergence goals on one processor.

otherwise. D determines the size of the convection terms. Dirichlet boundary conditions were used with $u = 1$ on the $z = 0$ plane and $u = 0$ on the other boundaries. Discretization was done on a grid of 40^3. The resulting linear system is indefinite, with eigenvalues in the four quadrants of the imaginary plane. It was shown in [11] that the usefulness of GRS for Bi-CGSTAB and GMRES degrades as the convection terms are increased.

In order to compare the performance of the various methods as the convection increases, we ran tests with convection terms of $D = 100, 200, 500, 1000$. The results with one processor are summarized in Table 3 (only for methods which achieved at least one convergence goal).

These results show that GRS is useful for Bi-CGSTAB and GMRES, with and without ILU(0), but only for the lower values of D. For the higher values of D, CGMN (CARP-CG on one processor) provides the best approach. CGNR+GRS also provides good results for large D.

The optimal relaxation parameter ω of CGMN varied with the convection in the range 1.35 to 1.65. However, the runtimes of CGMN varied very little when ω was changed between these values, so a fixed value of $\omega = 1.50$ is sufficient to obtain reasonable results on this problem.

Convection:	D=100		D=200		D=500		D=1000	
Convergence goal:	10^{-4}	10^{-10}	10^{-4}	10^{-10}	10^{-4}	10^{-10}	10^{-4}	10^{-10}
Bi-CGSTAB+GRS	1.00	3.62	2.77	9.85	9.74	38.24	—	—
Bi-CGSTAB+ILU(0)	1.35	1.74	2.30	—	—	—	—	—
with GRS	**0.76**	**1.67**	**1.90**	—	—	—	—	—
GMRES+GRS	2.05	—	2.23	—	4.50	—	—	—
with ILU(0)	**0.76**	—	—	—	—	—	—	—
CGNR+GRS	5.10	11.28	4.78	10.23	5.96	13.42	7.29	16.15
CGMN (CARP-CG)	2.01	4.60	1.93	**4.95**	**1.96**	**6.30**	**2.45**	**7.59**

Table 3: Problem 3: time (sec.) for the different methods to achieve convergence to 10^{-4} and 10^{-10}, with increasing convection terms. Minimal times are in boldface.

4.2 High-Frequency Helmholtz Equations

The fourth problem is the well-known Marmousi model [15]. The domain Ω corresponds to a 6000m×1600m vertical slice of the Earth's subsurface, with the x-axis as horizontal and the y-axis pointing downwards. A point disturbance source is located at the center of the upper boundary. The wave equation is $\Delta u + k^2 u = g$, where $g(x, y) = \delta(x - (x_{\min} + x_{\max})/2)\delta(y)$. The domain is highly heterogeneous, with velocity c ranging from 1500 m/s to 4450 m/s. Three grid sizes are available, and here we used the mid-sized grid of 1501×401.

This problem was discretized using 2nd, 4th and 6th order finite difference schemes, with the Sommerfeld radiation condition at the boundary. The two higher order schemes were done along the lines of [20, 6]; both schemes produce a 9-point stencil matrix.

The frequency f is determined by the user, and our test cases consisted of $f = 20$ and $f = 40$. For $f = 20$, the number of grid points per wavelength, $(N_g(20))$, was in the range $18.75 \leq N_g(20) \leq 55.625$. For $f = 40$, we get $9.375 \leq N_g(40) \leq 27.8125$.

Fig. 2 shows the relative residual results with CARP-CG for one to 16 processors, using the 6th order scheme, with frequencies of 20 and 40. We can see that the high frequency case requires fewer iterations than the lower frequency. We can also see that when the number of processors increases, more iterations are required for convergence to some specific relative residual. However, CARP-CG scales better with the high frequency case in the following sense: when the number of processors increases, the percentage increase in the number of iterations is much smaller than in the case of the low frequency. Similar results were obtained with the 2nd and 4th order schemes.

Unfortunately, there is no known analytic solution for the Marmousi problem. Our previous work with high-order schemes on problems with known analytic solutions [12] showed that the 2nd-order scheme achieved very poor error results, even with 18 grid points per wavelength, compared to the 6th order scheme. In contrast, the 6th order scheme achieved very good error results. We can conclude from this that the 6th order scheme is much closer to the true solution.

In order to evaluate the various schemes on this problem, we run the 6th order scheme to a relative residual of 10^{-13}, saved the result as the "true" solution, and compared the results of

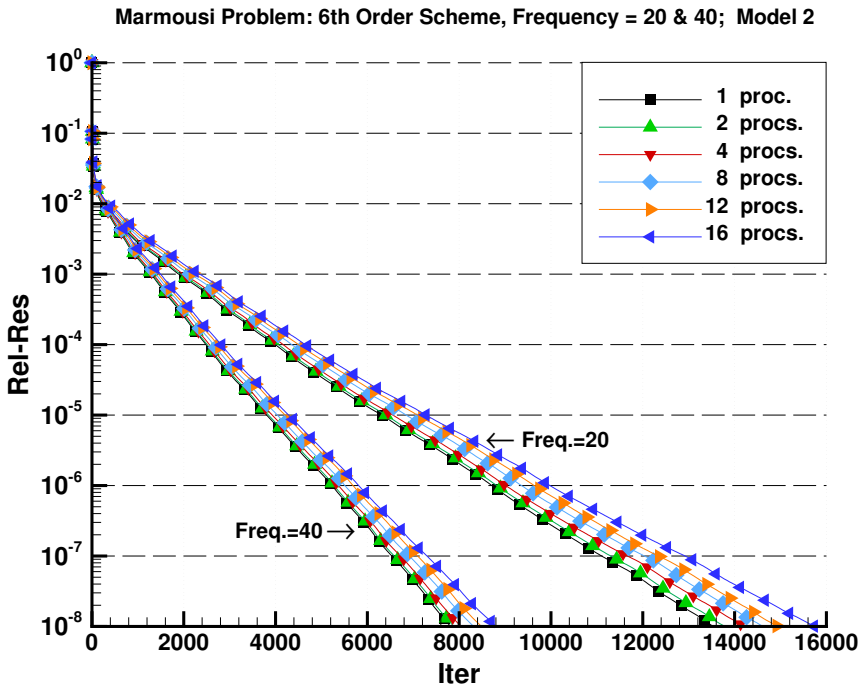

Figure 2: The Marmousi problem: convergence of the relative residual for 1 to 16 processors.

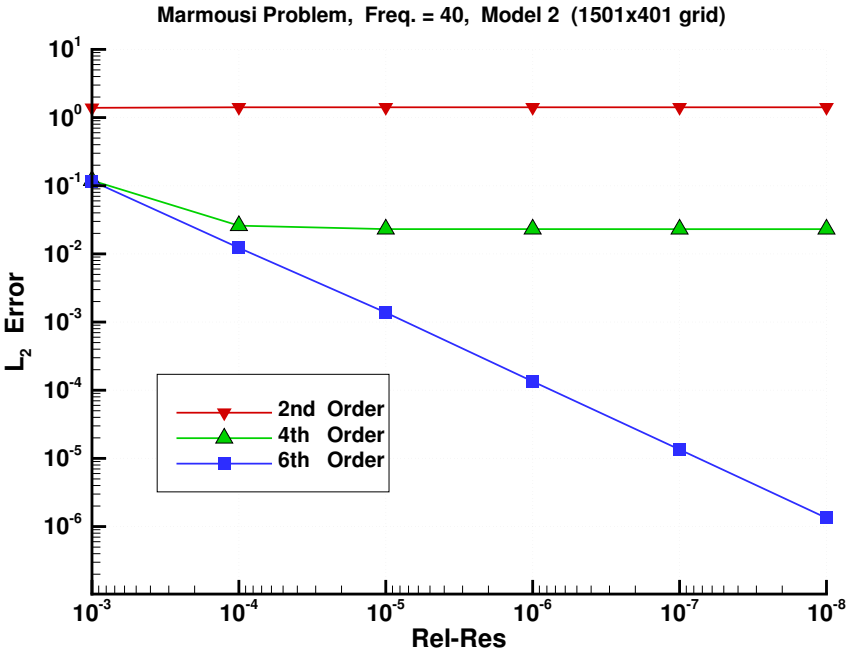

Figure 3: The Marmousi problem: relative error compared to the solution obtained with the 6th-order scheme, for frequency = 40.

the three schemes with this solution. Fig. 3 shows the relative L_2-error of the three schemes w.r.t. to the "true" solution, as a function of the relative residual, for frequency $f = 40$. We can see that the 2nd order scheme is virtually useless, and even the 4th order scheme is insufficient for most purposes. In any case, since the 4th and 6th order scheme take the same time per iteration, the 6th order scheme is preferable.

5 Conclusions and further research

This paper reviewed the block-parallel CARP-CG algorithm for solving nonsymmetric linear systems in which the off-diagonal elements are very large, including cases of discontinuous coefficients. Examples included convection-dominated elliptic PDEs in homogeneous and heterogeneous media. Additionally, a numerical study of a high frequency Helmholtz equation in heterogeneous media, using 2nd, 4th and 6th order finite difference schemes, demonstrated the importance of high order schemes and the good scalability of CARP-CG on such problems.

Future research directions will include the application of CARP-CG to high-frequency Helmholtz equations in 3D heterogeneous media, using high-order finite difference schemes. Another potential application is the solution of high-frequency Maxwell equations.

References

[1] I. M. Babuška and S. A. Sauter, Is the Pollution Effect of the FEM Avoidable for the Helmholtz Equation Considering High Wave Numbers? *SIAM Review 42* (2000), 451–484.

[2] M. Benzi, Preconditioning Techniques for Large Linear Systems: a Survey. *Journal of Computational Physics 182* (2002), 418–477.

[3] Å Björck and T. Elfving, Accelerated Projection Methods for Computing Pseudoinverse Solutions of Systems of Linear Equations. *BIT 19* (1979), 145–163.

[4] G. Cimmino, Calcolo Approssimato per le Soluzioni dei Sistemi di Equazioni Lineari. *La Ricerca Scientifica XVI, Series II, Anno IX 1* (1938), 326–333.

[5] C. Dohrmann, A. Klawonn, and O. Widlund, Domain decomposition for less regular subdomain: overlapping Schwarz in two dimensions. *SIAM J. on Numerical Analysis 46*, (2008), 2153–2168.

[6] Y. A. Erlangga and E. Turkel, Iterative Schemes for High Order Compact Discretizations to the Exterior Helmholtz Equation. *ESAIM: Mathematical Modelling and Numerical Analysis to appear* (2011).

[7] L. Gerardo-Giorda, P. L. Tallec and F. Nataf, A Robin-Robin Preconditioner for Advection-Diffusion Equations with Discontinuous Coefficients. *Computer Methods in Applied Mechanics & Engineering 193* (2004), 745–764.

[8] D. Gordon and R. Gordon, Component-Averaged Row Projections: A Robust, Block-Parallel Scheme for Sparse Linear Systems. *SIAM J. on Scientific Computing 27* (2005), 1092–1117.

[9] D. Gordon and R. Gordon, Solution Methods for Linear Systems with Large off-Diagonal Elements and Discontinuous Coefficients. *Computer Modeling in Engineering & Sciences 53*, 1 (Nov. 2009), 23–45.

[10] D. Gordon and R. Gordon, CARP-CG: A Robust and Efficient Parallel Solver for Linear Systems, Applied to Strongly Convection-Dominated PDEs. *Parallel Computing 36*, 9 (Sept. 2010), 495–515.

[11] D. Gordon and R. Gordon, Row Scaling as a Preconditioner for Some Nonsymmetric Linear Systems with Discontinuous Coefficients. *J. of Computational & Applied Mathematics 234*, 12 (Oct. 2010), 3480–3495.

[12] D. Gordon and R. Gordon, Parallel Solution of High-Frequency Helmholtz Equations Using High-Order Finite Difference Schemes. In *Proc. 10th International Conf. on Mathematical & Numerical Aspects of Waves* (Vancouver, Canada, 2011).

[13] I. G. Graham and M. J. Hagger, Unstructured Additive Schwarz-Conjugate Gradient Method for Elliptic Problems with Highly Discontinuous Coefficients. *SIAM J. on Scientific Computing 20*, 6 (1999), 2041–2066.

[14] S. Kaczmarz, Angenäherte Auflösung von Systemen Linearer Gleichungen. *Bulletin de l'Académie Polonaise des Sciences et Lettres A35* (1937), 355–357.

[15] The Marmousi Model. *Workshop on Practical Aspects of Seismic Data Inversion.* In Proc. 52nd Annual Meeting of the European Association of Geophysicists & Engineers, 1990.

[16] C. Pechstein and R. Scheichl, Analysis of FETI methods for multiscale PDEs. *Numerische Mathematik 111* (2008), 293–333.

[17] C. Pechstein and R. Scheichl, Analysis of FETI methods for multiscale PDEs – part II: interface variation. *Numerische Mathematik 118* (2011), 485–529.

[18] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd.ẽd. SIAM, Philadelphia, PA, 2003.

[19] Y. Saad and M. H. Schultz, GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM J. on Scientific & Statistical Computing 7* (1986), 856–869.

[20] I. Singer and E. Turkel, Sixth Order Accurate Finite Difference Schemes for the Helmholtz Equation. *J. of Computational Acoustics 14* (2006), 339–351.

[21] H. A. van der Vorst, Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems. *SIAM J. on Scientific & Statistical Computing 13* (1992), 631–644.

[22] O. B. Widlund, On the Effects of Scaling of the Peaceman-Rachford Method. *Mathematics of Computation 25*, 113 (Jan. 1971), 33–41.