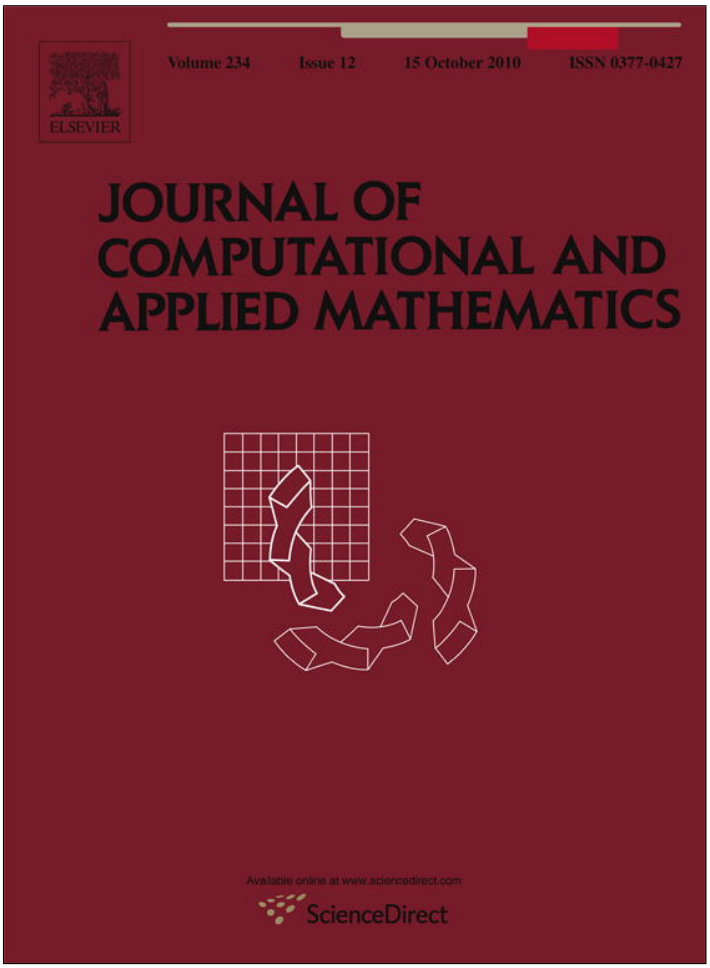

Journal of Computational and Applied Mathematics 234 (2010) 3480–3495

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Row scaling as a preconditioner for some nonsymmetric linear systems with discontinuous coefficients

Dan Gordon [a,*], Rachel Gordon [b]

[a] Department of Computer Science, University of Haifa, Haifa 31905, Israel

[b] Department of Aerospace Engineering, The Technion–Israel Inst. of Technology, Haifa 32000, Israel

ARTICLE INFO

Article history:
Received 18 October 2009
Received in revised form 13 April 2010

Keywords:
Bi-CGSTAB
Diagonal scaling
Discontinuous coefficients
Domain decomposition
Geometric scaling
GMRES
GRS
L_p-norm
Linear equations
Nonsymmetric equations
Partial differential equations

ABSTRACT

Linear systems with large differences between the coefficients, called "discontinuous coefficients", often arise when physical phenomena in heterogeneous media are modeled by partial differential equations (PDEs). Such problems are usually solved by domain decomposition techniques, but these can be difficult to implement when subdomain boundaries are complicated or the grid is unstructured. It is known that for such systems, diagonal scaling can sometimes improve the eigenvalue distribution and the convergence properties of some algorithm/preconditioner combinations. However, there seems to be no study outlining both the usefulness and limitations of this approach. It is shown that L_2-scaling of the equations is a generally useful preconditioner for such problems when the system matrices are nonsymmetric, but only when the off-diagonal elements are small to moderate. Tests were carried out on several nonsymmetric linear systems with discontinuous coefficients derived from convection–diffusion elliptic PDEs with small to moderate convection terms. It is shown that L_2-scaling improved the eigenvalue distribution of the system matrix by reducing their concentration around the origin very significantly. Furthermore, such scaling improved the convergence properties of restarted GMRES and Bi-CGSTAB, with and without the ILU(0) preconditioner. Since ILU(0) is theoretically oblivious to diagonal scaling, these results indicate that L_2-scaling also improves the runtime numerical stability.

1. Introduction

Many physical phenomena are modeled by PDEs which describe the relations between one or more scalar or vector fields and the surrounding media. In the case of heterogeneous media, the discretization of the PDEs can yield so-called systems with "discontinuous coefficients", meaning that there are very large differences between the coefficients. Examples include modeling flow through heterogeneous media with varying permeability, oil reservoir simulation, electromagnetics and semiconductor modeling.

One of the most common methods for tackling such problems is the domain–decomposition (DD) approach, in which the domain is partitioned into subdomains, with the subdomain boundaries conforming to the boundaries between the different media. DD techniques typically operate as follows: some boundary conditions are assumed to exist on the interfaces between subdomains, and a solution of the equations in each subdomain is obtained, often with only low accuracy. The boundary conditions at the interfaces are then updated according to these solutions, and the process is repeated until convergence. There exists a vast amount of literature on this subject; see, for example, [1–4]. DD techniques may be difficult to implement on unstructured grids or when the boundaries between domains have a complicated geometry.

* Corresponding author. Tel.: +972 4 8240159; fax: +972 4 8249331.
E-mail addresses: gordon@cs.haifa.ac.il, dann.gordon@gmail.com (D. Gordon), rgordon@tx.technion.ac.il (R. Gordon).

0377-0427/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2010.05.021

To a much lesser extent than the DD approach, diagonal scaling is known to improve the eigenvalue distribution and the convergence properties of some algorithms on certain problems with discontinuous coefficients. Some examples of this are given in the next subsection. However, the numerical results are rather sporadic and there seems to be no comprehensive study of what is useful and in which cases. Also, the limitations of this approach have not been studied.

In this paper we consider nonsymmetric linear systems with discontinuous coefficients derived from convection–diffusion elliptic PDEs with small to moderate convection terms. It is shown that L_2-scaling of the equations improves the convergence properties of some algorithms applied to such systems. For convenience, we call this scaling *geometric row scaling* (GRS). GRS is a geometric operation in the following sense: any results produced by an algorithm applied to the system after GRS depend only on the hyperplanes defined by the equations and not on any particular algebraic representation of those hyperplanes. GRS is inherent in some algorithms such as Kaczmarz (KACZ) [5] and Cimmino [6]. L_2-scaling was also found to be useful for certain problems in image reconstruction from projections [7].

In order to examine the general usefulness of GRS in conjunction with various solution methods, we tested it with the two leading Krylov subspace solvers for nonsymmetric systems: GMRES [8] and Bi-CGSTAB [9]. Both algorithms were tested with and without the ILU(0) preconditioner on several nonsymmetric problems taken from (or based on) [10,9,11]. Four basic problems were considered, as well as several variations of the problems, such as modifications of the differences between the coefficients and various grid sizes. Our results can be characterized as follows:

- In most cases, when the tested method (algorithm/preconditioner combination) converges to the specified accuracy criterion, GRS speeds up the convergence time.
- In many cases, when the tested method stagnates or diverges on the original system, it converges on the scaled system.
- GRS was not particularly useful on problems without discontinuous coefficients.
- GRS was not helpful on problems derived from PDEs with large convection terms. Such PDEs produce linear systems with very large off-diagonal elements, and these require other solution methods. For a recent treatment of such problems, see [12].

We also provide data on the effect of GRS on the distribution of the eigenvalues. It is generally considered to be detrimental to the convergence of Krylov subspace methods to have a large accumulation of eigenvalues near the origin. Such an accumulation appears in all the cases examined here, and the effect of GRS is to "push" many eigenvalues away from the origin.

Note that the above choice of algorithm/preconditioner combinations does not imply that these methods are optimal for the above problems. They were chosen for their very wide usage, with the purpose of showing the general usefulness of GRS.

DD methods are often mentioned in connection with parallel processing. The equations of different domains can, in principle, be solved in parallel by different processors. However, the different domains that arise in many practical situations do not necessarily lead to an optimal load-balancing assignment of equations to processors. With the GRS approach, inherently parallel algorithms such as Bi-CGSTAB and GMRES can be implemented efficiently in parallel without regard to subdomain boundaries. Needless to say, GRS is inherently parallel.

The rest of this paper is organized as follows. Section 2 lists some of the known results concerning diagonal scaling. Section 3 presents the background and setup of the experiments. Sections 4–7 deal with the four different problems, while Section 8 shows the limitations of GRS as the convection terms are increased. Section 9 concludes with a summary and some future research directions.

2. Related work

It is known that diagonal scaling can sometimes improve the eigenvalue distribution and the convergence properties of some algorithms on problems with discontinuous coefficients. Businger [13] provides sufficient conditions for a matrix to be optimally scalable. Van der Sluis [14] deals with the effect of scaling on the condition number of matrices. Fenner and Loizou [15] improved on previously achieved bounds on the condition number of optimally scaled matrices.

Scalings have also been studied in the context of certain problems in operations research. Marshall and Olkin [16], and O'Leary [17] considered two-sided scalings aimed at achieving specified row and column sums, while Rothblum et al. [18] considered the two-sided scalings required in order to achieve prescribed row and column maxima. These three papers deal with symmetric matrices, so their techniques are not applicable to the nonsymmetric matrices obtained in our problems.

It should be mentioned that the purpose of scaling in this work is to improve the convergence properties of certain algorithm/preconditioner combinations on nonsymmetric problems with discontinuous coefficients. To this end, the condition number of the scaled matrix plays a rather minor role; in two of the four problems studied here, the scaling does not improve the condition number very much. What is relevant to the performance of the algorithms on the studied problems is the *distribution* of the eigenvalues; this is actually a well known fact.

Widlund [19, p. 34–35] writes that well-scaled ADI methods give good rates of convergence when the coefficients of elliptic problems vary very much in magnitude (ADI stands for the alternating direction implicit method [20]). Graham and Hagger [11, p. 2042–2043] write that diagonal scaling has been observed in practical computations to be very effective as a preconditioner for problems with discontinuous coefficients. Duff and van der Vorst [21] write that on vector machines, diagonal scaling is often competitive with other approaches.

When the system matrix is symmetric, diagonal scaling is usually performed on the left and on the right in order to preserve symmetry. Golub and Varah [22] characterized the best two-sided L_2-scaling of a matrix in terms of its singular

value decomposition. Vuik et al. [23] show that for a class of layered problems, two-sided scaling can reduce the number of extreme eigenvalues, making it equal to the number of layers of high permeability. Gambolati et al. [24] used the least square logarithm (LSL) scaling on the rows and the columns of the system matrix for a certain problem in geomechanics with discontinuous coefficients. On this topic, see also Meurant [25, Th. 8.1, 8.2], who also has some reservations about the usefulness of diagonal scaling for parallel processing [25, p. 401].

Gambolati et al. [24] remark that with ILU(0), the iteration matrix of Bi-CGSTAB on a diagonally scaled system (on the left and/or the right) is theoretically the same as the one with ILU(0) on the original matrix; yet in [24], the scaling produced better numerical results. Our results confirm these findings, which are probably due to the fact that the scaled system does not have very large differences between coefficients, so it is much less prone to numerical errors.

3. Background and setup of the experiments

3.1. General background

Throughout the rest of the paper, we assume that all vectors are column vectors, and we use the following notation: $\langle p,\ q\rangle$ denotes the dot product of two vectors p and q, which is also $p^T q$. Given a vector $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$, we denote its L_p-norm by $\|x\|_p = (x_1^p + \cdots + x_n^p)^{1/p}$. For $p = 2$, we will omit the index and just write $\|x\| = \|x\|_2 = \sqrt{\langle x, x\rangle}$. If A is an $n \times n$ matrix, we denote by a_i the ith row-vector of A; i.e., $a_i = (a_{i1}, \ldots, a_{in})^T$.

Consider a system of n linear equations in n variables:

$$\sum_{j=1}^{n} a_{ij}x_j = b_i \quad \text{for } 1 \le i \le n, \text{ or, in matrix form: } Ax = b. \tag{3.1}$$

We shall assume throughout that (3.1) is consistent and that A does not contain a row of zeros. For $p \ge 1$, we define a diagonal matrix $D = \mathrm{diag}(1/\|a_1\|_p, \ldots, 1/\|a_n\|_p)$. The geometrically-scaled system is defined as

$$DAx = Db. \tag{3.2}$$

In some algorithms, GRS(p) is an inherent step in the following sense: either the scaling is executed at the beginning as an intrinsic part of the algorithm, or, executing the algorithm produces identical results to those obtained when GRS(p) is done at the beginning. As an example, it is easy to see that GRS(2) is inherent in the Kaczmarz algorithm (KACZ) [5]. KACZ can be described geometrically as follows: starting from an arbitrary point $x^0 \in \mathbb{R}^n$ as the initial iterate, KACZ projects the current iterate orthogonally towards a hyperplane defined by one of the equations. The hyperplanes are chosen in cyclic order.

The tests were run using the AZTEC software library [26], which includes a wide range of algorithms and preconditioners, suitable for sequential and parallel implementations. Geometric scaling with the L_1-norm is a built-in option in AZTEC (where it is called "row-sum scaling"). As mentioned we used Bi-CGSTAB and GMRES, with and without ILU(0). Restarted GMRES was used with Krylov subspace size of 10. In AZTEC, GMRES is implemented with a double classical Gram–Schmidt orthogonalization step. No doubt, better convergence properties and runtimes would be obtained if the restart value for GMRES was fine-tuned to each problem, or if ILU(0) was replaced with ILUT [27] with suitable parameters. However, the purpose of the experiment was not to find the optimal solution method for each problem, but to demonstrate the general usefulness of GRS on common solution methods. The eigenvalue computations were done with the LAPACK linear algebra package [28].

We also experimented with the option of multiplying the system matrix A by $D^{\frac{1}{2}}$ on the left and on the right, resulting in the system $D^{\frac{1}{2}}AD^{\frac{1}{2}}y = D^{\frac{1}{2}}b$, with $x = D^{\frac{1}{2}}y$. In one problem, the results were somewhat poorer than those obtained with GRS, but otherwise, the results were similar. Note that this option is computationally more expensive.

3.2. Setup of the numerical experiments

In two dimensions, the general form of the second-order differential equations in this study was

$$\frac{\partial}{\partial x}(a(x,y)u_x) + \frac{\partial}{\partial y}(b(x,y)u_y) + \cdots = F,$$

where a and b are given functions of x and y, "$\cdots$" stands for lower-order derivatives, and F is a prescribed RHS. In three dimensions, there are three given functions a, b, c of x, y, z, and a second order partial derivative w.r.t. z was also included. Boundary conditions were either Dirichlet or mixed Dirichlet and Neumann. The regions were taken as the unit square or the unit cube. The discretization of the second-order derivatives at a given grid point (i, j) was done using central differences; e.g., $\frac{\partial}{\partial x}(au_x)$ was approximated as

$$\begin{aligned}
\frac{\partial}{\partial x}(au_x)_{i,j} &= \left((au_x)_{i+\frac{1}{2},j} - (au_x)_{i-\frac{1}{2},j}\right)/\Delta x \\
&= \left(a_{i+\frac{1}{2},j}(u_{i+1,j} - u_{i,j})/\Delta x - a_{i-\frac{1}{2},j}(u_{i,j} - u_{i-1,j})/\Delta x\right)/\Delta x \\
&= \left(-(a_{i+\frac{1}{2},j} + a_{i-\frac{1}{2},j})u_{i,j} + a_{i+\frac{1}{2},j}u_{i+1,j} + a_{i-\frac{1}{2},j}u_{i-1,j}\right)/\Delta x^2.
\end{aligned}$$

Table 4.1
No. of iterations and runtimes for Problem 1. Gridsize = 128×128.

Method	No. of iterations and time (in s)					
	Rel-res = 10^{-4}		Rel-res = 10^{-7}		Rel-res = 10^{-10}	
Bi-CGSTAB	No conv.		No conv.		No conv.	
With GRS	91	(0.30)	299	(0.99)	361	(1.19)
Bi-CGSTAB+ILU(0)	31	(0.23)	107	(0.67)	142	(0.88)
With GRS	30	(0.23)	90	(0.59)	130	(0.81)
GMRES	Converged to 3.8×10^{-2}					
With GRS	265	(0.85)	Converged to 1.1×10^{-5}			
GMRES+ILU(0)	Converged to 3.9×10^{-3}					
With GRS	39	(0.23)	Converged to 1.1×10^{-5}			

All problems were discretized with equally-spaced grids, and the initial value was taken as $u^0 = 0$. The tests were run on a Pentium IV 2.8 GHz processor with 3 GB memory, running Linux.

3.3. Stopping tests

There are several stopping criteria which one may apply to iterative systems. Our stopping criterion was to use the relative residual: $\|b - Ax\|/\|b - Ax^0\| < \varepsilon$, where ε was taken as 10^{-4}, 10^{-7} and 10^{-10}. In some of the cases, this was not attainable. Since this stopping criterion depends on the scaling of the equations, we always made this test on the geometrically-scaled system using the L_2-norm. In order to limit the time taken by the methods implemented in AZTEC, the maximum number of iterations was set to 10,000. The AZTEC library has several other built-in stopping criteria: numerical breakdown, numerical loss of precision and numerical ill-conditioning. In the results, we denote the relative residual by rel-res, and non-convergence by "no conv."

One should note that the test for numerical breakdown in AZTEC uses the machine precision `DBL_EPSILON`, and this may result in a premature notice of numerical breakdown in some cases. To get around this problem, we multiplied the variable `brkdown_tol` in the Bi-CGSTAB algorithm by some small factor, such as 10^{-16}. (`brkdown_tol` is normally set equal to `DBL_EPSILON`, which is approximately 2.22×10^{-16} on our machine.)

4. Problem 1

Problem 1 is taken from [10, Section 3.7, problem F2DB]. It is a two-dimensional PDE

$$-\frac{\partial}{\partial x}(au_x) - \frac{\partial}{\partial y}(bu_y) + \frac{\partial(du)}{\partial x} + \frac{\partial(eu)}{\partial y} = h,$$

where

$$a(x, y) = b(x, y) = \begin{cases} 10^3 & \text{if } \dfrac{1}{4} < x,\ y < \dfrac{3}{4}, \\ 1 & \text{otherwise} \end{cases}$$

and $d(x, y) = 10(x+y)$ and $e(x, y) = 10(x-y)$. The Dirichlet boundary condition $u = 0$ is used on the boundary, and the RHS h is immaterial since the vector b of the linear system is chosen as $b = Ae$, where A is the system matrix and $e = (1, \ldots, 1)^T$.

Table 4.1 shows the number of iterations and runtimes of the various algorithm/preconditioner methods, with and without GRS, on a grid of 128×128. In cases of stagnation, the table shows the relative residual that was achieved before stagnation. Note that GRS enables the convergence of Bi-CGSTAB without ILU(0); this is useful in a parallel setting, because Bi-CGSTAB is inherently parallel but ILU(0) is not an ideal parallel preconditioner. We can also see that GRS is slightly helpful to Bi-CGSTAB with ILU(0), and, for low-accuracy, also to GMRES with and without ILU(0). Another observation is that when GMRES (with and without ILU(0)) stagnates before reaching the prescribed convergence goal, GRS postpones the stage at which stagnation sets in, and enables convergence to a level that is acceptable for most practical applications.

Three additional experiments, based on Problem 1, were also done:

1. The values of $a(x, y)$ and $b(x, y)$ were increased to 10^4 in the inner square. The results were very similar to those shown in Table 4.1, but with slightly increased runtimes in the higher-accuracy cases.
2. A "continuous" case: the values of $a(x, y)$ and $b(x, y)$ were taken as 1 throughout the unit square. Here, all the methods converged, and GRS made very little difference.
3. A second continuous case, with $a(x, y) = b(x, y) = 1000$. Table 4.2 shows the results of this experiment. As can be seen, GRS made very little difference.

We turn now to studying the effect of GRS on the distribution of the eigenvalues. For Problem 1, this was done with a grid size of 40×40 for the purpose of a clear presentation. Table 4.3 shows the values of the minimum and maximum eigenvalues and the condition number, for the original, the scaled and the continuous ($a = b = 1000$) cases of Problem 1. Also shown for each case is the number of eigenvalues in the first interval (out of 100) in a histogram of the eigenvalue distribution. We can

Table 4.2
No. of iterations and runtimes for a continuous version of Problem 1, with $a = b = 1000$ everywhere.

Method	No. of iterations and time (in s)					
	rel-res $= 10^{-4}$		rel-res $= 10^{-7}$		rel-res $= 10^{-10}$	
Bi-CGSTAB	121	(0.41)	224	(0.76)	286	(0.96)
With GRS	123	(0.42)	217	(0.74)	261	(0.88)
Bi-CGSTAB+ILU(0)	39	(0.28)	61	(0.41)	85	(0.56)
With GRS	40	(0.29)	60	(0.40)	85	(0.56)
GMRES	1365	(4.31)	3704	(11.69)	6045	(19.19)
With GRS	1356	(4.28)	3582	(11.45)	5722	(18.16)
GMRES+ILU(0)	140	(0.69)	359	(1.69)	579	(2.67)
With GRS	140	(0.69)	359	(1.69)	579	(2.67)

Table 4.3
Basic eigenvalue information for Problem 1.

Matrix	λ_{min}	λ_{max}	$\lambda_{max}/\lambda_{min}$	No. of eigenvalues in first interval
Original	1.87E−2	7.96E+3	4.25E+5	1126
With GRS	7.07E−6	1.78E+0	2.52E+5	4
Cont. coef.	1.17E+1	8.00E+3	6.81E+2	8

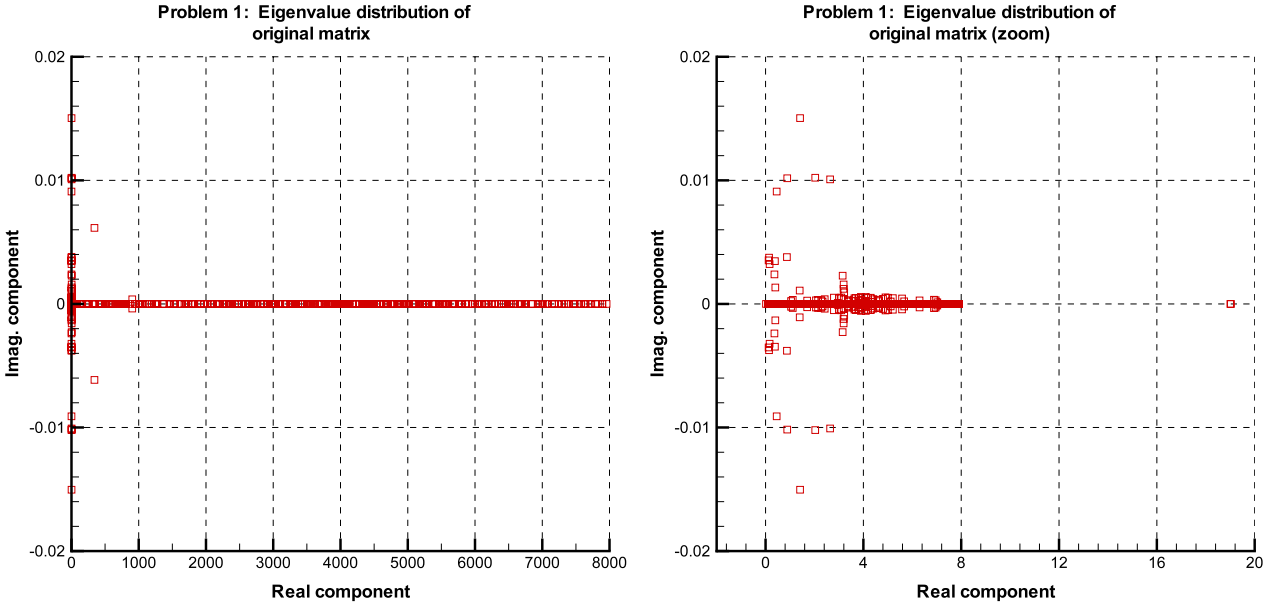

Fig. 4.1. Eigenvalue distribution of the original matrix of Problem 1, with a zoom in the range 0–20.

see that while the condition number is not changed much by the scaling, the number of eigenvalues in the first interval is reduced very significantly.

Fig. 4.1 shows the distribution of the eigenvalues of Problem 1 at the full scale (left) and with a zoom (right). We can see that there is a congestion of eigenvalues close to zero.

Fig. 4.2 shows the distribution of the eigenvalues of Problem 1 after geometric scaling with the L_1-norm (left) and the L_2 (right). Both scalings produce fairly similar distributions; however we will present results throughout the paper for the L_2 scaling. As to the continuous version of the problem, its eigenvalues were all real and distributed evenly in the range $(0, 8000)$.

Fig. 4.3 shows a histogram of the eigenvalue distributions for the original Problem 1, for the geometrically scaled problem, and for the continuous case with $a = b = 1000$.

5. Problem 2

Problem 2 is based on Example 2 from [9], to which we added convection terms. This example demonstrates that GRS also works with mixed Dirichlet and Neumann boundary conditions. The governing PDE is

$$-\frac{\partial(D(x,y)u_x)}{\partial x} - \frac{\partial(D(x,y)u_y)}{\partial y} + au_x + bu_y = 1,$$

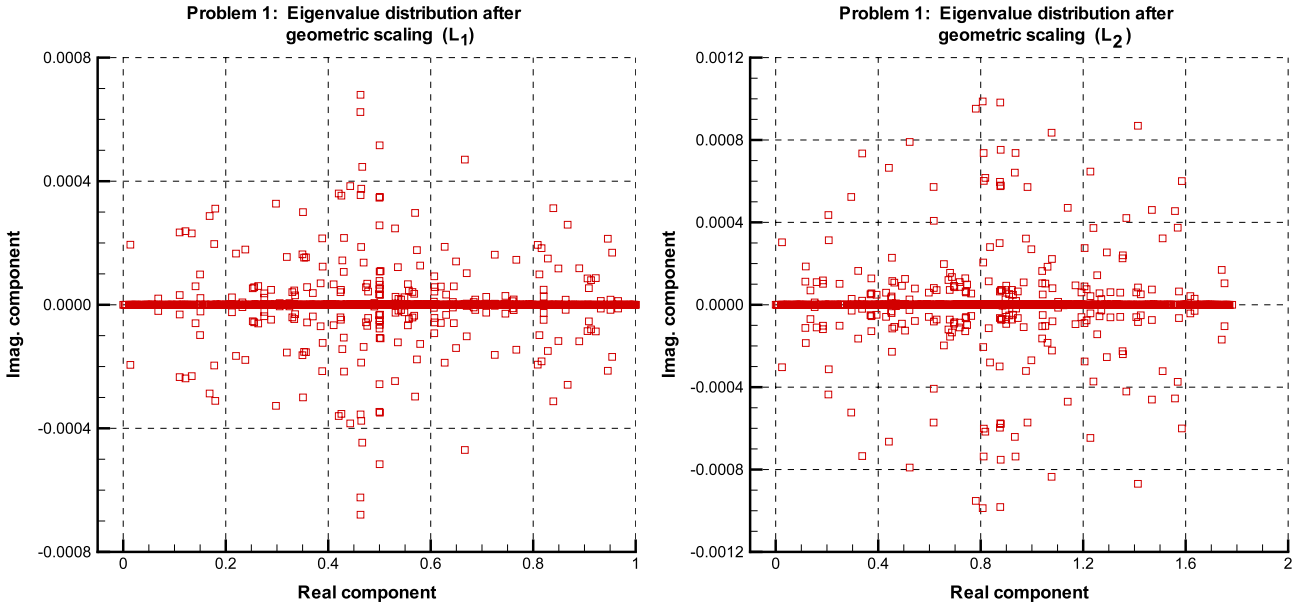

Fig. 4.2. Eigenvalue distribution for Problem 1 after geometric scaling with the L_1- and the L_2-norms.

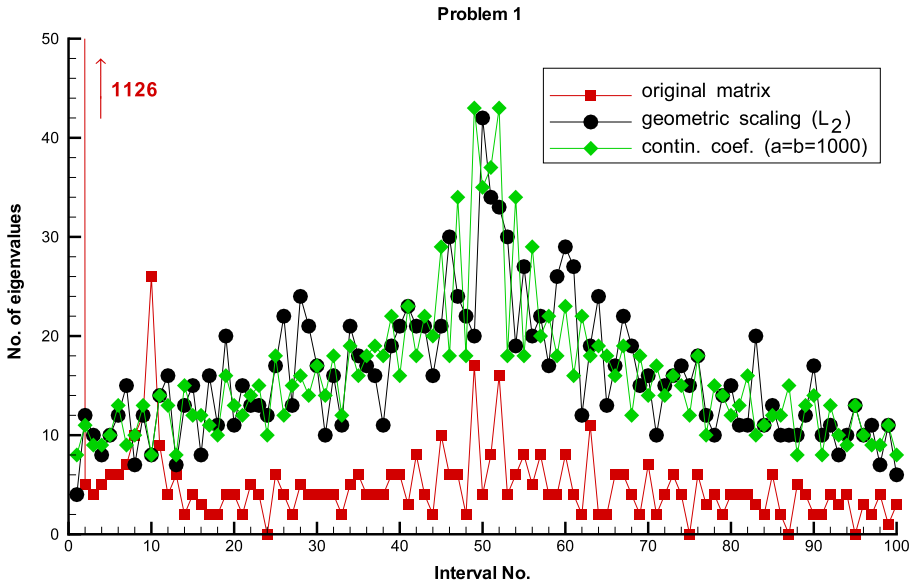

Fig. 4.3. Histogram of eigenvalues for Problem 1, for the original matrix, the geometrically scaled matrix, and for a variation of Problem 1 with continuous coefficients ($a = b = 1000$).

with $D(x, y)$ and boundary conditions as shown in Fig. 5.4. The convection terms were $a = b = 200$. In the original example, the internal value of D is 10^3, but we also tested internal values of 10^4 and 10^5. Also, the internal square in our case is somewhat smaller. The unit square was discretized with a grid size of 150×150. Together with the boundary equations, the system consisted of 22,952 equations. The resulting system is indefinite, with eigenvalues in the four quadrants of the imaginary plane.

Tables 5.4–5.6 show the number of iterations and runtimes of the various algorithm and preconditioner combinations, with and without GRS, for $D = 10^3$, $D = 10^4$, and $D = 10^5$, respectively. When there was no convergence to the prescribed goal, the table shows the relative residual that was achieved.

The tables show that GRS was helpful in all cases, with the exception of Bi-CGSTAB with ILU(0) in Table 5.4, where it increased the number of iterations required in the highest accuracy case. A close examination of this case revealed that starting from iteration 220, BiCGSTAB with ILU(0) (after GRS) was extremely oscillatory, with fluctuations of up to three orders of magnitude in the relative residual; the required convergence goal of 10^{-10} was almost attained at 225 iterations.

The three tables show that the behavior of GRS is not necessarily "smooth": consider the results of Bi-CGSTAB with ILU(0) for rel-res $= 10^{-10}$: we can see that the goal was not reached for $D = 10^3$ and $D = 10^5$, but it was reached for $D = 10^4$. The explanation for this is the large number of iterations required by Bi-CGSTAB in this case, as seen in Table 5.5 and the oscillatory nature of Bi-CGSTAB. The accumulated roundoff error is sometimes too large to reach the convergence goal,

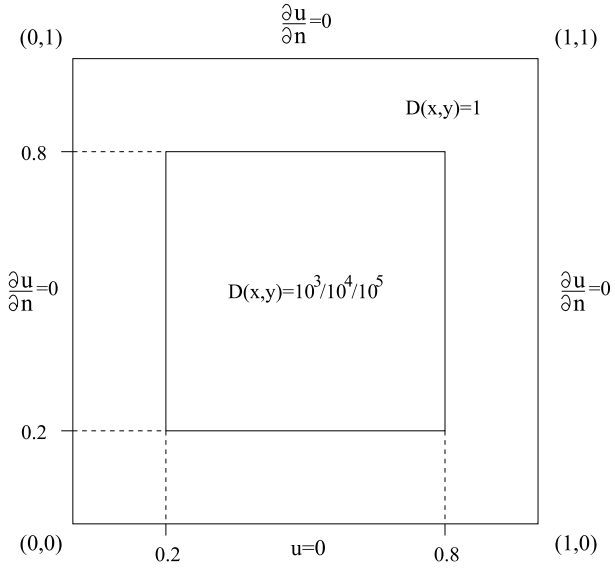

Fig. 5.4. Description of Problem 2.

Table 5.4
No. of iterations and runtimes for Problem 2 (internal $D = 10^3$).

Method	No. of iterations and time (in s)					
	Rel-res $= 10^{-4}$		Rel-res $= 10^{-7}$		Rel-res $= 10^{-10}$	
Bi-CGSTAB	No conv.		No conv.		No conv.	
With GRS	591	(2.13)	1025	(3.69)	Conv. to 7.23×10^{-10}	
Bi-CGSTAB+ILU(0)	92	(0.75)	139	(1.09)	215	(1.68)
With GRS	88	(0.72)	125	(1.00)	324	(2.45)
GMRES	Converged to 6.15					
With GRS	Converged to 2.145×10^{-3}					
GMRES+ILU(0)	Converged to 0.927					
With GRS	3361	(19.86)	Converged to 3.67×10^{-5}			

Table 5.5
No. of iterations and runtimes for Problem 2 (internal $D = 10^4$).

Method	No. of iterations and time (in s)					
	Rel-res $= 10^{-4}$		Rel-res $= 10^{-7}$		Rel-res $= 10^{-10}$	
Bi-CGSTAB	No conv.		No conv.		No conv.	
With GRS	589	(2.01)	707	(2.41)	1493	(5.37)
Bi-CGSTAB+ILU(0)	96	(0.76)	192	(1.44)	255	(1.86)
With GRS	79	(0.64)	121	(0.92)	163	(1.19)
GMRES	Converged to 1.84					
With GRS	Converged to 2.19×10^{-4}					
GMRES+ILU(0)	Converged to 0.924					
With GRS	Converged to 1.90×10^{-4}					

and while the updated error measured by the algorithm showed convergence, the true relative residual did not always decrease sufficiently.

Another interesting point is the behavior of GRS with GMRES and ILU(0) in the low-accuracy case: for $D = 10^3$, there was convergence after many iterations, for $D = 10^4$ there was no convergence, and then for $D = 10^5$ there was a very fast convergence. In order to examine this behavior, we also tested this particular case with $D = 10^6$, and we noticed that starting $D = 10^4$, GRS improved the convergence of GMRES by one order of magnitude for every increase in D by one order of magnitude. Not doubt, this phenomenon should be studied further. In any case, for the larger values of D, GRS enabled the convergence of GMRES, with and without ILU(0), to practical levels of accuracy.

For the eigenvalue data, we discretized the domain by a grid of $40 \times 40 \times 40$. Table 5.7 provides the basic eigenvalue information for Problem 2, for the original ($D = 10^3$) and the scaled matrices. Also shown are the eigenvalues for a variation of Problem 2 with "continuous" coefficients, obtained by taking $D = 10^3$ throughout the unit square. The last column of the

Table 5.6
No. of iterations and runtimes for Problem 2 (internal $D = 10^5$).

Method	No. of iterations and time (in s)					
	Rel-res $= 10^{-4}$		Rel-res $= 10^{-7}$		Rel-res $= 10^{-10}$	
Bi-CGSTAB	No conv.		No conv.		No conv.	
With GRS	224	(0.82)	730	(2.65)	Conv. to 1.69×10^{-9}	
Bi-CGSTAB+ILU(0)	96	(1.15)	192	(1.56)	Conv. to 5.16×10^{-10}	
With GRS	21	(0.23)	125	(0.98)	217	(1.68)
GMRES	Converged to 1.37					
With GRS	357	(1.38)	Converged to 2.19×10^{-5}			
GMRES+ILU(0)	Converged to 0.90					
With GRS	40	(0.32)	Converged to 1.89×10^{-5}			

Table 5.7
Basic eigenvalue information for Problem 2.

Matrix	λ_{min}	λ_{max}	$\lambda_{max}/\lambda_{min}$	Eigenvalues around $x = 0$
Original ($D = 10^3$)	3.33E−1	1.34E+7	4.02E+7	1058
With GRS	9.15E−5	1.78E0	1.95E+4	8
Continuous coef.	1.48E−2	1.34E+7	9.09E+8	130

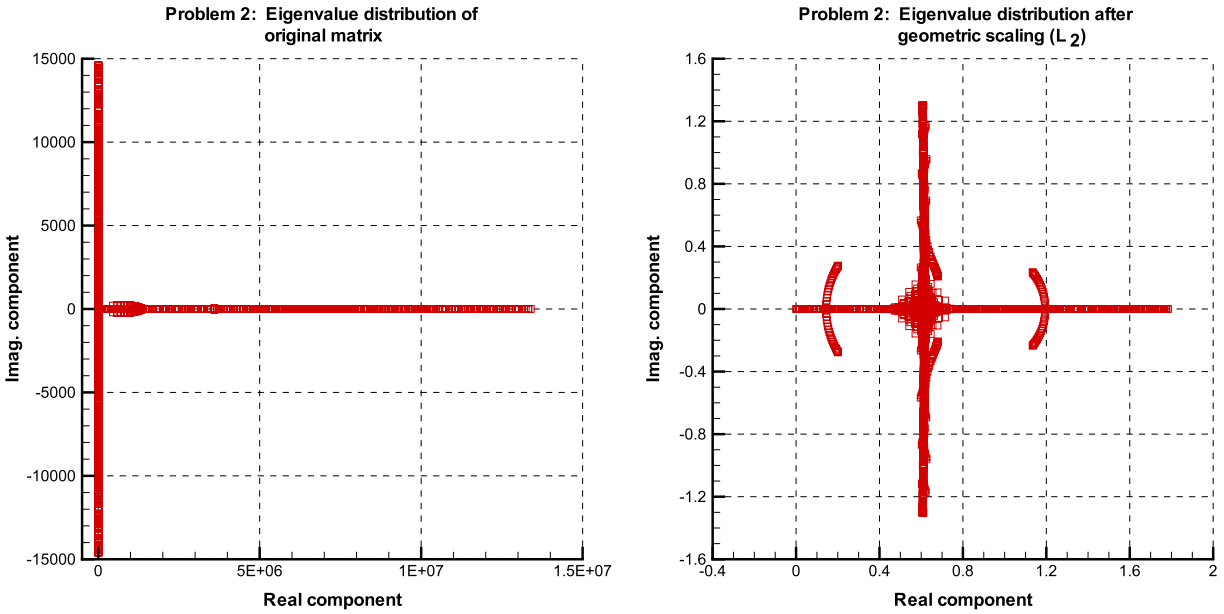

Fig. 5.5. Eigenvalue distribution for Problem 2, for the original and the scaled cases.

table shows the number of eigenvalues whose real part lies in the same interval as the origin, when the range of (the real parts) of the eigenvalues is divided into 100 intervals. We can see that GRS reduces the condition number by three orders of magnitude, and the reduction is even greater w.r.t. the continuous case. Furthermore, the number of eigenvalues around the origin is reduced very significantly, even below that of the continuous case.

Fig. 5.5 shows the distribution of the eigenvalues for Problem 2 with $D = 10^3$, and for the geometrically scaled problem. It can be seen that the eigenvalues of the original matrix are very concentrated around the origin. In the scaled matrix, there are very few eigenvalues around the origin, but many eigenvalues have their real part around 0.6.

Fig. 5.6 provides another view of the eigenvalue distribution with a histogram of the real part of the eigenvalues for the original and the scaled matrices.

6. Problem 3

Problem 3 is also taken from [9, Example 4]. It describes a certain groundwater flow problem which leads to a nonsymmetric system, with a complex geometry and several jumps in the discontinuities of the equations. This problem is

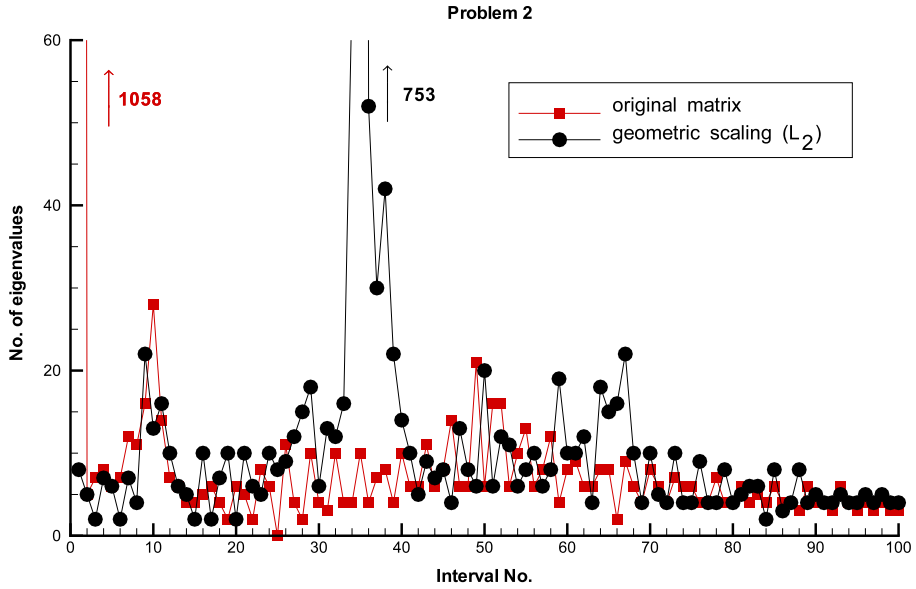

Fig. 5.6. Histogram of the real part of the eigenvalues for Problem 2, for the original and the scaled cases.

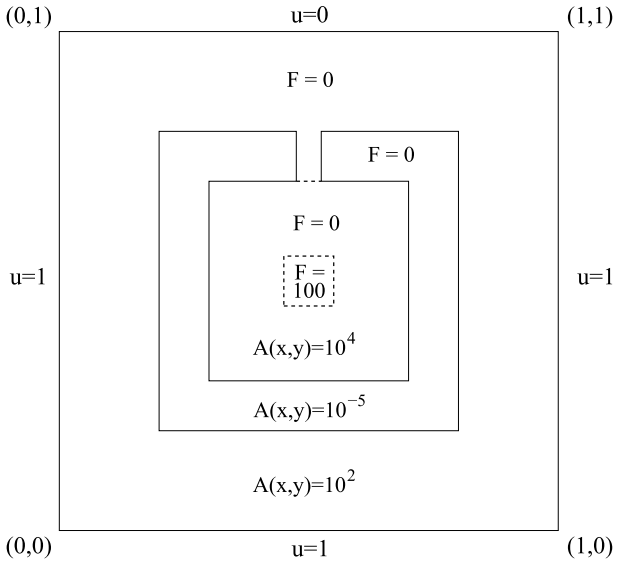

Fig. 6.7. Description of Problem 3.

well-known for its difficulty. The basic equation is the following:

$$-\frac{\partial}{\partial x}\left(A(x,y)u_x\right)-\frac{\partial}{\partial y}\left(A(x,y)u_y\right)+B(x,y)u_x=F,$$

where $A(x, y)$ and F are taken as shown in Fig. 6.7, and $B(x, y) = 2\exp(2(x^2 + y^2))$. The Dirichlet boundary conditions are taken as shown in Fig. 6.7. The unit square was discretized with a grid size of 128×128, resulting in 16,129 equations.

Nothing converged without ILU(0), so we present in Table 6.8 only the results with ILU(0), with and without GRS. On this difficult problem, GRS reduced the runtime of Bi-CGSTAB with ILU(0) by about 25%. GRS also enabled the convergence of GMRES with ILU(0), though the achieved runtimes were an order of magnitude larger than those of the scaled Bi-CGSTAB with ILU(0) in the higher-accuracy cases.

For the eigenvalue data, we discretized Problem 3 with a grid of 40×40. The data for the original and the scaled matrices is summarized in Table 6.9, which also shows the number of eigenvalues in the first interval (out of 100) of the eigenvalue histogram. We can see that the condition number is decreased very significantly, and so is the number of eigenvalues in the first interval.

Fig. 6.8 shows the distribution of the eigenvalues for the original (with a zoom) and the scaled matrices of Problem 3.

Fig. 6.9 is a histogram of the eigenvalues of the original and the scaled matrices. We can see from Figs. 6.8 and 6.9 that in the original matrix, the eigenvalues are very concentrated around the origin, while in the scaled matrix, many are "pushed" away from the origin.

Table 6.8
No. of iterations and runtimes for Problem 3.

Method	No. of iterations and time (in s)					
	Rel-res $= 10^{-4}$		Rel-res $= 10^{-7}$		Rel-res $= 10^{-10}$	
Bi-CGSTAB+ILU(0)	93	(0.60)	124	(0.79)	152	(0.95)
With GRS	67	(0.44)	90	(0.59)	112	(0.72)
GMRES+ILU(0)	No conv.		No conv.		No conv.	
With GRS	338	(1.58)	1008	(4.61)	1683	(7.71)

Table 6.9
Basic eigenvalue information for Problem 3.

Matrix	λ_{min}	λ_{max}	$\lambda_{max}/\lambda_{min}$	No. of eigenvalues in first interval
Original	6.97E−5	7.93E+4	1.14E+9	1284
With GRS	1.21E−4	1.78E+0	1.47E+4	167

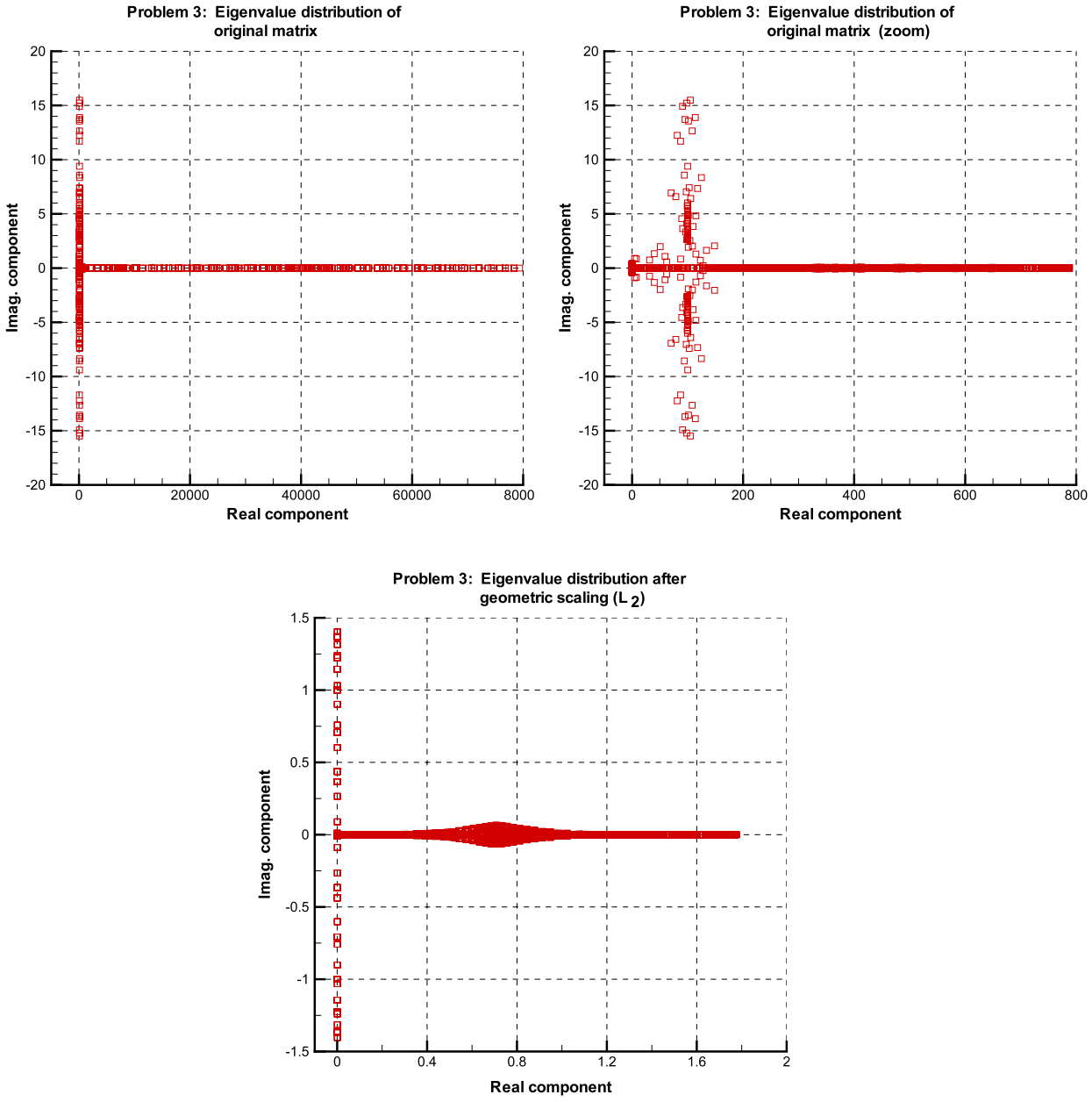

Fig. 6.8. Eigenvalue distribution for Problem 3, with a zoom to the region 0–800, and the distribution for the scaled matrix.

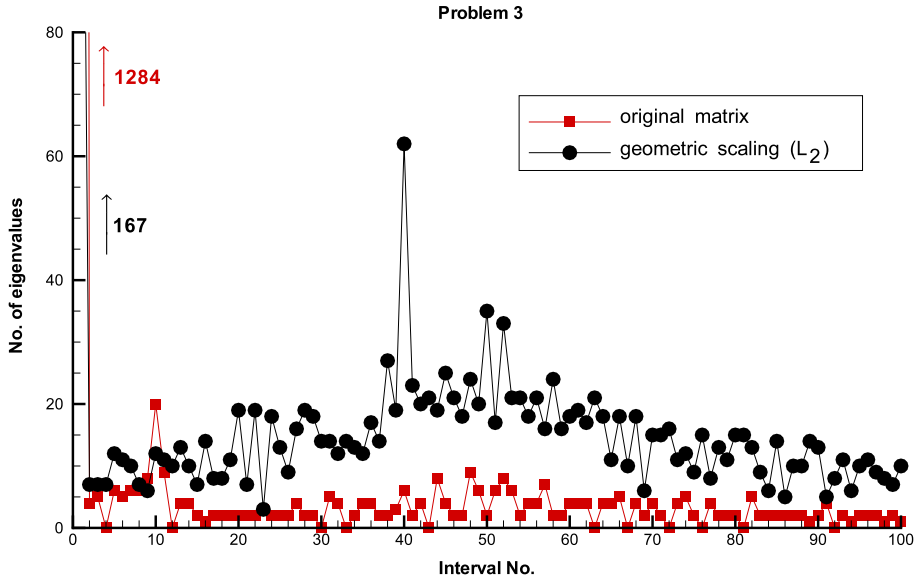

Fig. 6.9. Histogram of the eigenvalues for Problem 3, for the original and the scaled cases.

Table 7.10
No. of iterations for Problem 4, with $D = 10^4$, $D = 10^6$, and grid $80 \times 80 \times 80$.

Method	Rel-res $= 10^{-4}$		Rel-res $= 10^{-7}$		Rel-res $= 10^{-10}$	
	$D = 10^4$	$D = 10^6$	$D = 10^4$	$D = 10^6$	$D = 10^4$	$D = 10^6$
Bi-CGSTAB	–	–	–	–	–	–
With GRS	112	111	262	160	406	374
Bi-CGSTAB+ILU(0)	77	52	112	139	123	169
With GRS	13	13	52	19	84	86
GMRES	–	–	–	–	–	–
With GRS	208	207	–	286	–	–
GMRES+ILU(0)	–	–	–	–	–	–
With GRS	28	28	–	46	–	–

Note: "–" denotes no convergence to the prescribed accuracy.

7. Problem 4

This problem is based on a three-dimensional symmetric problem from [11], where it is solved using domain decomposition techniques. We added convection terms to this problem in order to make it nonsymmetric. The differential equation is the following:

$$-\frac{\partial}{\partial x}(au_x) - \frac{\partial}{\partial y}(au_y) - \frac{\partial}{\partial z}(au_z) + du_x + eu_y + fu_z = 0,$$

where the domain is the unit cube, and $a(x, y, z)$ is defined as

$$a(x, y, z) = \begin{cases} D & \text{if } \frac{1}{3} < x, y, z < \frac{2}{3}, \\ 1 & \text{otherwise.} \end{cases}$$

The Dirichlet boundary conditions are prescribed with $u = 1$ on the $z = 0$ plane and $u = 0$ on the other boundaries. The convection terms were taken as $d = e = f = 100$, and two values of D were tested: $D = 10^4$ (as in the original problem) and $D = 10^6$. Two grids were tested: $40 \times 40 \times 40$ and $80 \times 80 \times 80$. This problem will also be used to test the limit of usefulness of GRS as the convection increases.

Due to the many different cases, the data for this problem is presented differently. Table 7.10 shows the number of iterations required for convergence, for a grid of $80 \times 80 \times 80$, with $D = 10^4$ and $D = 10^6$. We can see that GRS enabled the convergence of Bi-CGSTAB without ILU(0), and it speeded up Bi-CGSTAB with ILU(0) quite significantly. GRS also enabled the convergence of GMRES, with and withou ILU(0), for rel-res $= 10^{-4}$, and also for rel-res $= 10^{-7}$ with $D = 10^6$.

Figs. 7.10 and 7.11 show bar-plots of the runtimes for the two grid sizes, for $D = 10^4$ and $D = 10^6$, for the three levels of accuracy. In cases of stagnation, the figures show (in parentheses) the relative residual achieved before stagnation.

The results can be summarized as follows.

Results for $D = 10^4$ Results for $D = 10^6$

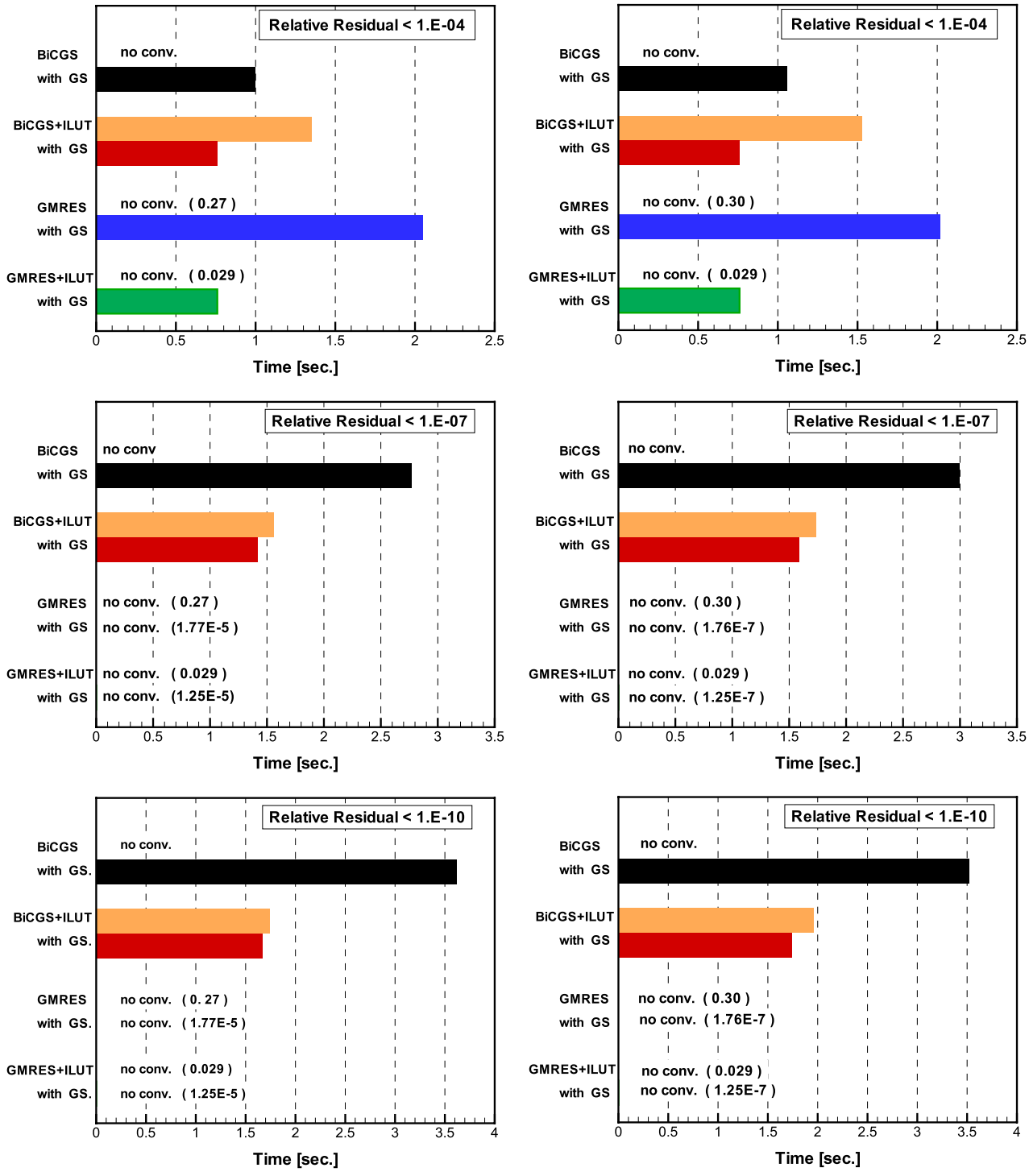

Fig. 7.10. Runtimes and convergence status for Problem 4, grid $= 40 \times 40 \times 40$.

- Bi-CGSTAB (without ILU(0)) and GMRES (with and without ILU(0)) did not converge in any of the cases.
- GRS enabled the convergence of Bi-CGSTAB in all cases, and the convergence of GMRES (with and without ILU(0)) in the low-accuracy cases and in one mid-accuracy case.
- GRS significantly improved the accuracy of GMRES (with and without ILU(0)) by postponing the stage at which stagnation set in, enabling convergence to much higher levels of accuracy.
- For the $80 \times 80 \times 80$-grid, GRS speeded up the convergence of Bi-CGSTAB with ILU(0) quite significantly. This was also true for the coarser grid, but only in the low-accuracy case.

Results for $D = 10^4$ Results for $D = 10^6$

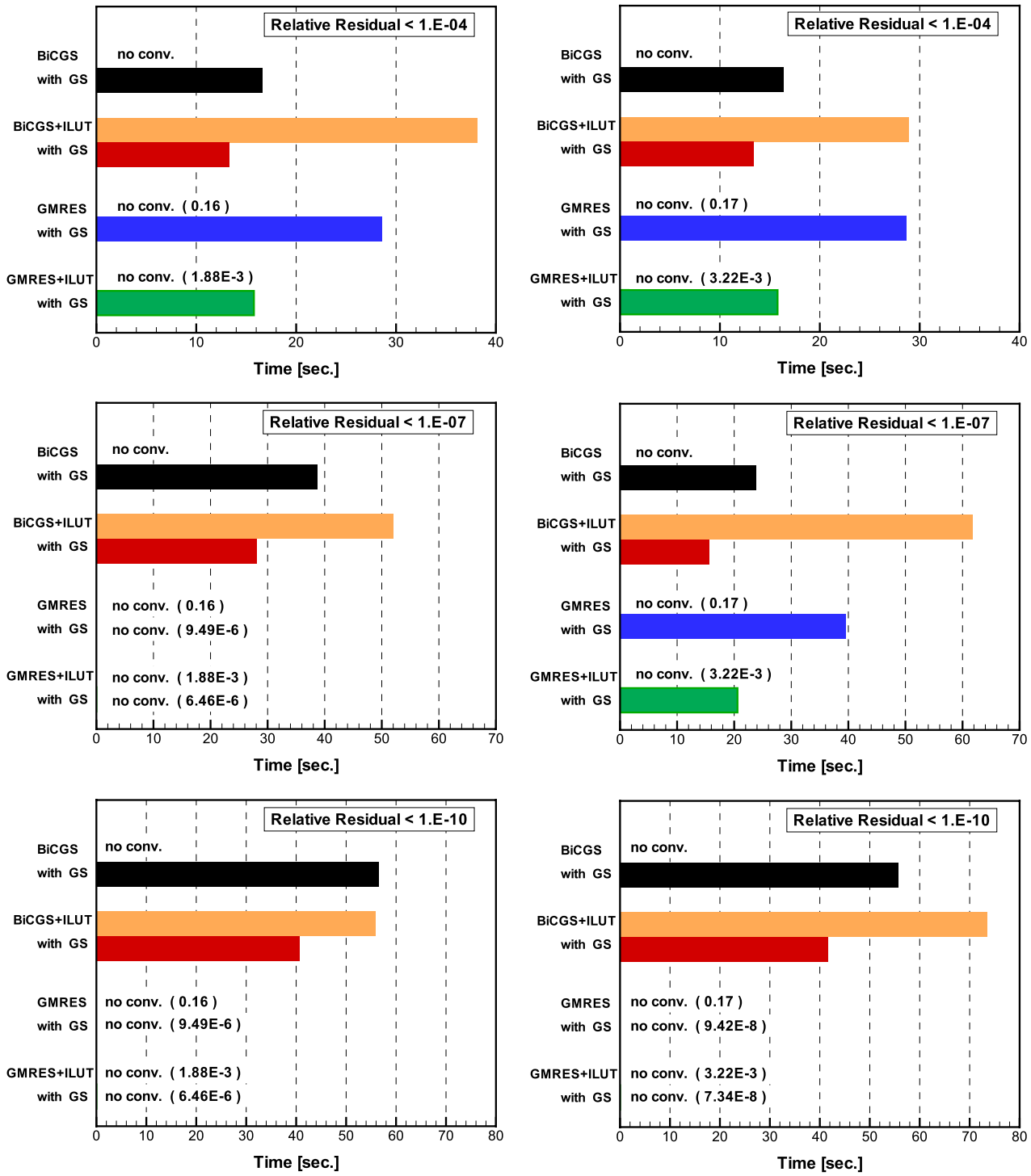

Fig. 7.11. Runtimes and convergence status for Problem 4, grid = 80 × 80 × 80.

- In the above cases, GRS (by itself) was generally a better preconditioner than ILU(0) for Bi-CGSTAB.
- In all cases with GRS, the results for $D = 10^6$ were either similar to or better than those for $D = 10^4$.
- In summary, GRS was beneficial in all cases to all the algorithm/ILU(0) combinations, and in many cases, GRS provided a very significant advantage.

Table 7.11 provides the basic eigenvalue information for Problem 4 with $D = 10^4$, for the original and the scaled matrices, and also for a continuous version with $D = 10^4$ everywhere. The grid size for this data was 12 × 12 × 12. The last column shows the number of eigenvalues whose real part lies in the first interval out of 100 intervals.

Table 7.11
Basic eigenvalue information for Problem 4.

Matrix	λ_{min}	λ_{max}	$\lambda_{max}/\lambda_{min}$	No. of eigenvalues in 1st interval
Original	3.62E+0	1.09E+5	3.02E+4	1131
With GRS	8.83E−5	1.92E+0	2.17E+4	1
Cont. coef. ($D = 10^4$)	1.74E+3	1.18E+5	6.78E+1	1

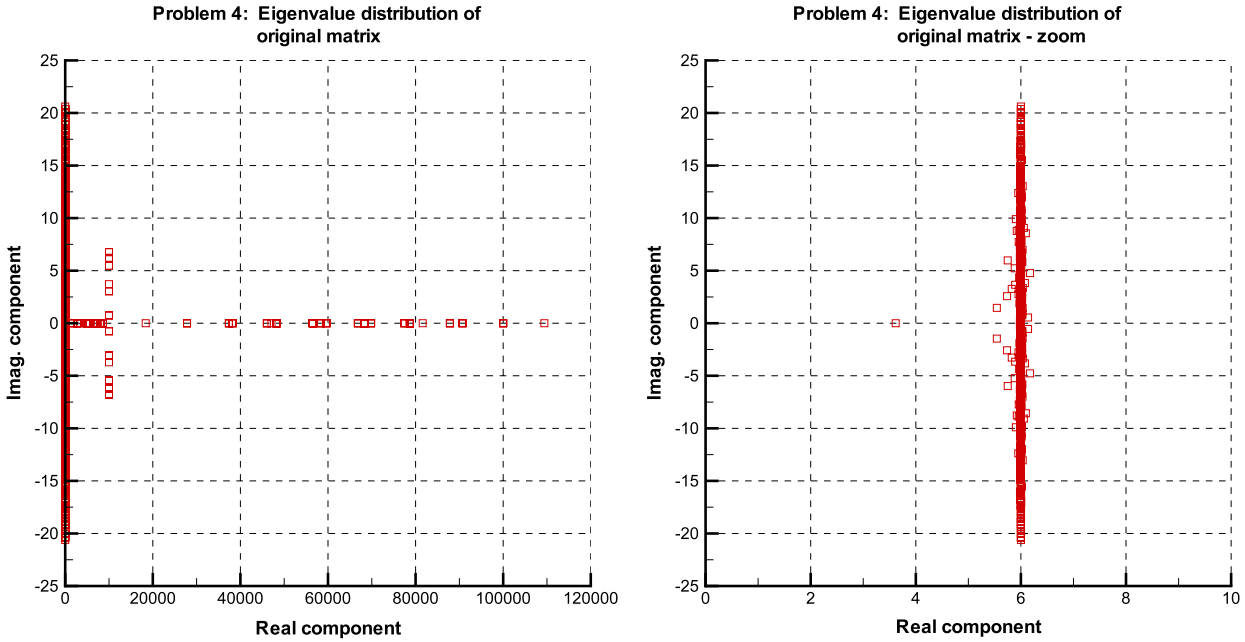

Fig. 7.12. Eigenvalue distribution for Problem 4, with a zoom to the region $\Re \leq 10$.

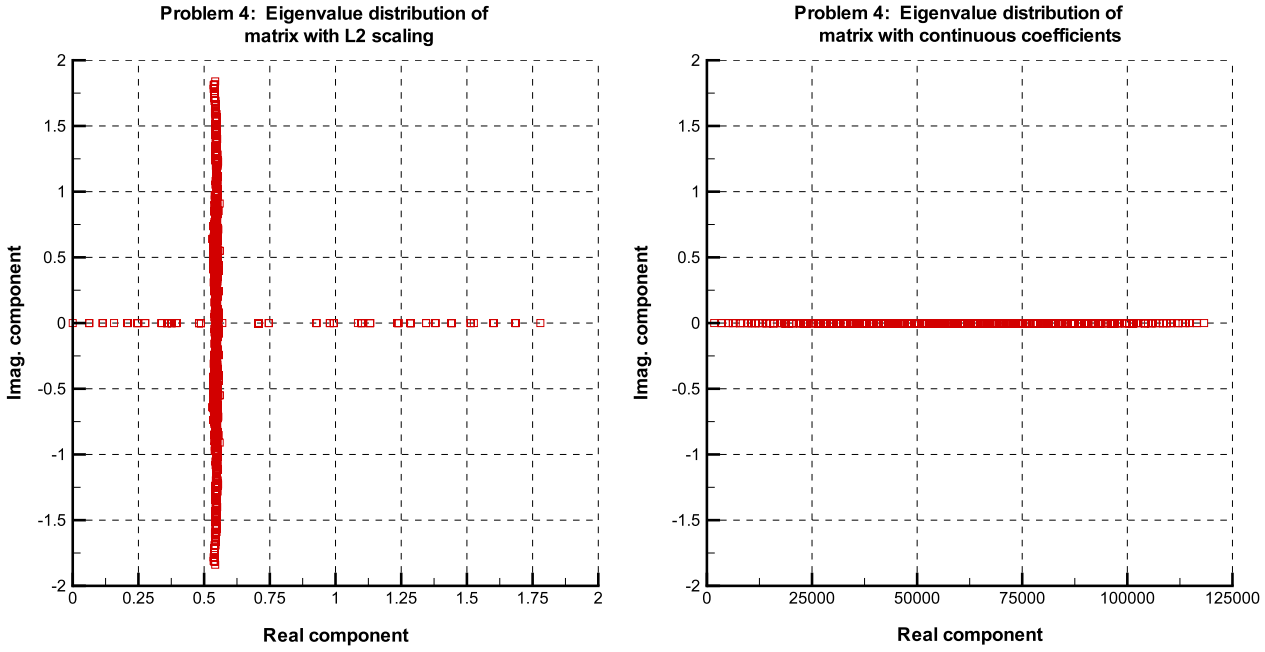

Fig. 7.13. Eigenvalue distribution for Problem 4, for the scaled and the continuous cases.

Fig. 7.12 shows the distribution of the eigenvalues for the original Problem 4, and a zoom to the region where the real part of the eigenvalues are $\leq$10. We can see that the eigenvalues are very concentrated around the origin (relative to the eigenspectrum). Fig. 7.13 shows the distribution of the eigenvalues for the scaled and the continuous ($D = 10^4$) cases. We can see that in the scaled matrix, there is also a large concentration of eigenvalues at a small region, but this is relatively far from the origin. In the continuous version of the problem, the eigenvalues are distributed quite evenly.

Table 8.12
Degradation of the various methods as the convection terms are increased.

Method/convection:	100	200	500	1000
Bi-CGSTAB	–	–	–	–
With GRS	1.2E−12	1.7E−11	3.4E−13	–
Bi-CGSTAB+ILU(0)	2.5E−14	3.2E−5	–	–
With GRS	2.5E−14	4.1E−5	–	–
GMRES	–	–	–	–
With GRS	1.8E−5	2.7E−5	5.6E−5	–
GMRES+ILU(0)	–	–	–	–
With GRS	1.2E−5	–	–	–

Note: '–' means divergence or early stagnation. The numbers indicate the relative error obtained.

8. Limitations of GRS

GRS is proposed as being useful for discontinuous coefficients when the convection terms are small to moderate. In order to demonstrate the limitations of GRS as the convection terms are increased, we considered Problem 4 with $D = 10^4$, a grid size of $40 \times 40 \times 40$, and varying convection terms. The previous results were based on convection terms of 100, so we also tested this case with convection terms of 200, 500, and 1000.

The results, which are summarized in Table 8.12 below, show how the usefulness of GRS degrades as the convection terms are increased. Note that Bi-CGSTAB with GRS does not degrade monotonically due to its somewhat oscillatory nature. Also, with the higher convection, GRS is more useful to Bi-CGSTAB without ILU(0). Not shown in the table is the fact that as the convection increases, many more iterations are required by Bi-CGSTAB with GRS to achieve a given convergence goal. For example, for convection = 100, 200, and 500, Bi-CGSTAB with GRS required approximately 4, 10, and 38 s, respectively, to converge to 10^{-10}.

In the previous section, Fig. 7.13 showed the distribution of the eigenvalues of the scaled matrix for convection = 100. The distributions for the larger convection terms (which are not shown here) appear quite similar to that of convection = 100, but it is interesting to note that the large concentration of eigenvalues, as seen in Fig. 7.13, gets closer to the origin as the convection increases. Specifically, with convection = 100, 200, 500, and 1000, we get about 1500 eigenvalues in the 31st, 18th, 7th, and 3rd percentiles. This is an almost linear connection which requires additional theoretical study, especially in view of the fact that the total number of eigenvalues is only 1728.

9. Conclusions and further research

This paper examined the usefulness and limitations of L_2-scaling of the equations for improving the convergence properties of algorithms on nonsymmetric linear systems with discontinuous coefficients. This preconditioning, called GRS for convenience, was tested on four nonsymmetric problems derived from convection–diffusion elliptic PDEs, with small to moderate convection terms. Bi-CGSTAB and restarted GMRES, with and without ILU(0), were tested on four problems with and without GRS. Normally, such problems are solved by domain decomposition (DD) techniques, but these can be difficult to implement if the boundaries between the subdomains have a complicated geometry or the grid is unstructured. The problems were taken from (or based on) well-known examples from the literature, and included two- and three-dimensional cases, with Dirichlet and mixed Dirichlet/Neumann boundary conditions. One problem had a complicated geometry. Three different convergence goals were prescribed: relative residual $\leq 10^{-4}$, 10^{-7}, 10^{-10}.

Table 9.13 summarizes the convergence behavior of the four different algorithm/preconditioner combinations, with and without scaling, on the four problems, for the three convergence goals. The results indicate that GRS was very useful in improving the convergence status of the different solution methods on the tested problems. With one exception, when the solution method converged on a problem with discontinuous coefficients, GRS speeded up the convergence.

Eigenvalue distribution maps show that in all four problems, there is a strong concentration of values around the origin, GRS reduces this concentration very significantly and "pushes" the values away from the origin.

On the negative side, our results show that the usefulness of GRS diminishes when the convection terms are increased. Such problems exhibit not only discontinuous coefficients, but also very large off-diagonal elements, and other approaches are needed. Recently, some successful results on such problems were obtained in [12] with algorithms that were already shown to be useful on problems with large off-diagonal elements: the (sequential) CGMN algorithm [29,30] and its block-parallel version CARP-CG [31], and the CG-accelerated Cimmino algorithm (which is actually CGNR applied to the system preconditioned by GRS); though the latter is slower than the former ones.

Additional studies on row scaling are necessary. On the theoretical side, it is not clear why GRS improves the eigenvalue distribution. On the practical side, GRS needs to be tested in conjunction with other algorithms and preconditioners, on various other problems with discontinuous coefficients. Also, some finite-precision analysis is needed in order to explain the better numerical results when GRS is used together with ILU(0).

A natural question that arises is how does GRS compare with DD techniques? Over the years, these methods have achieved a high level of sophistication and efficiency, and a head-on runtime comparison is very much problem-dependent and a topic

Table 9.13
Summary of convergence of the different methods on the four problems, for the three convergence goals (10^{-4}, 10^{-7}, and 10^{-10}).

Method	Problem 1	Problem 2	Problem 3	Problem 4
Bi-CGSTAB	− − −	− − −	− − −	− − −
With GRS	+ + +	7.2×10^{-10}	− − −	+ + +
Bi-CGSTAB+ILU(0)	+ + +	+ + ∗	+ + +	+ + +
With GRS	+ ∗ ∗	∗ ∗ +	∗ ∗ ∗	∗ ∗ ∗
GMRES	3.8×10^{-2}	1.37	− − −	0.27
With GRS	1.1×10^{-5}	2.2×10^{-5}	− − −	1.8×10^{-5}
GMRES+ILU(0)	3.9×10^{-3}	0.90	− − −	0.29
With GRS	1.1×10^{-5}	1.9×10^{-5}	+ + +	1.25×10^{-5}

Note: '−' means no convergence, '+' means convergence, '∗' means better convergence. The numbers indicate the best relative error obtained.

for further research. An interesting topic for further study is the combination of the two methods: apply GRS to the equations, and then apply some standard DD method. It is conceivable that on some problems, GRS could also improve the convergence properties of DD.

GRS can also help the parallelization of solution methods in several ways. Firstly, GRS itself is a parallel computational step. Secondly, it enables the partitioning of a domain into subdomains along boundaries that do not necessarily follow the physical boundaries of the problem; this way, better load balancing can be achieved. Thirdly, as seen in some of the cases, GRS enabled the convergence of algorithms that are inherently parallel without the need for a preconditioner (such as ILU(0)) that may not be ideal for parallelism.

References

[1] R. Glowinski, M.F. Wheeler, Domain decomposition and mixed finite element methods for elliptic problems, in: R. Glowinski, G.H. Golub, G.A. Meurant, J. Périaux (Eds.), First International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, PA, USA, 1988, pp. 144–172.
[2] B. Smith, P. Bjørstad, W. Gropp, Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations, Cambridge University Press, Cambridge, UK, 1996.
[3] A. Quarteroni, A. Valli, Domain Decomposition Methods for Partial Differential Equations, Oxford University Press, Oxford, 1999.
[4] J.R. Rice, P. Tsompanopoulou, E. Vavalis, Interface relaxation methods for elliptic differential equations, Applied Numerical Mathematics 32 (2000) 219–245.
[5] S. Kaczmarz, Angenäherte Auflösung von Systemen linearer Gleichungen, Bulletin de l'Académie Polonaise des Sciences et Lettres A35 (1937) 355–357.
[6] G. Cimmino, Calcolo approssimato per le soluzioni dei sistemi di equazioni lineari, La Ricerca Scientifica XVI, Series II, Anno IX 1 (1938) 326–333.
[7] D. Gordon, R. Mansour, A geometric approach to quadratic optimization: an improved method for solving strongly underdetermined systems in CT, Inverse problems in Science & Engineering 15 (8) (2007) 811–826.
[8] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on Scientific & Statistical Computing 7 (1986) 856–869.
[9] H.A. van der Vorst, Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM Journal on Scientific & Statistical Computing 13 (1992) 631–644.
[10] Y. Saad, Iterative Methods for Sparse Linear Systems, 2nd ed., SIAM, Philadelphia, PA, 2003.
[11] I.G. Graham, M.J. Hagger, Unstructured additive Schwarz-conjugate gradient method for elliptic problems with highly discontinuous coefficients, SIAM Journal on Scientific Computing 20 (6) (1999) 2041–2066.
[12] D. Gordon, R. Gordon, Solution methods for linear systems with large off-diagonal elements and discontinuous coefficients, Computer Modeling in Engineering & Sciences 53 (1) (2009) 23–45.
[13] P.A. Businger, Matrices which can be optimally scaled, Numerische Mathematik 12 (1968) 346–348.
[14] A. van der Sluis, Condition numbers and equilibration of matrices, Numerische Mathematik 14 (1969) 14–23.
[15] T.I. Fenner, G. Loizou, Some new bounds on the condition numbers of optimally scaled matrices, Journal of the ACM 21 (3) (1974) 514–524.
[16] A.W. Marshall, I. Olkin, Scaling of matrices to achieve specified row and column sums, Numerische Mathematik 12 (1968) 83–90.
[17] D.P. O'Leary, Scaling symmetric positive definite matrices to prescribed row sums, Linear Algebra & Its Applications 370 (2003) 185–191.
[18] U.G. Rothblum, H. Schneider, M.H. Schneider, Scaling matrices to prescribed row and column maxima, SIAM Journal on Matrix Analysis & Applications 15 (1) (1994) 1–14.
[19] O.B. Widlund, On the effects of scaling of the Peaceman–Rachford method, Mathematics of Computation 25 (113) (1971) 33–41.
[20] D.W. Peaceman, H.H. Rachford Jr., The numerical solution of parabolic and elliptic differential equations, Journal of the Society of Industrial and Applied Mathematics 3 (1955) 28–41.
[21] I.S. Duff, H.A. van der Vorst, Preconditioning and parallel preconditioning, Technical Report TR/PA/98/23, CERFACS, Toulouse, France, 1998.
[22] G.H. Golub, J.M. Varah, On a characterization of the best ℓ_2-scaling of a matrix, SIAM Journal on Numerical Analysis 11 (3) (1974) 472–479.
[23] C. Vuik, A. Segal, J.A. Meijerink, An efficient preconditioned CG method for the solution of a class of layered problems with extreme contrasts in the coefficients, Journal of Computational Physics 152 (1999) 385–403.
[24] G. Gambolati, G. Pini, M. Ferronato, Scaling improves stability of preconditioned CG-like solvers for FE consolidation equations, International Journal for Numerical & Analytical Methods in Geomechanics 27 (2003) 1043–1056.
[25] G. Meurant, Computer Solution of Large Linear Systems, Elsevier, Amsterdam, 1999.
[26] R.S. Tuminaro, M.A. Heroux, S.A. Hutchinson, J.N. Shadid, AZTEC user's guide, Technical Report SAND99-8801J, Sandia National Laboratories, Albuquerque, New Mexico, 1999.
[27] Y. Saad, ILUT: a dual threshold incomplete ILU factorization, Numerical Linear Algebra with Applications 1 (1994) 387–402.
[28] LAPACK—Linear Algebra Package. http://www.netlib.org/lapack.
[29] Å. Björck, T. Elfving, Accelerated projection methods for computing pseudoinverse solutions of systems of linear equations, BIT 19 (1979) 145–163.
[30] D. Gordon, R. Gordon, CGMN revisited: robust and efficient solution of stiff linear systems derived from elliptic partial differential equations, ACM Transactions on Mathematical Software 35 (3) (2008) 18:1–18:27.
[31] D. Gordon, R. Gordon, CARP-CG: a robust and efficient parallel solver for linear systems, applied to strongly convection-dominated PDEs, Parallel Computing (May 2010), doi:10.1016/j.parco.2010.05.004.