PERGAMON Computers and Mathematics with Applications 45 (2003) 263–272

An International Journal
computers & mathematics with applications

www.elsevier.com/locate/camwa

Explicit Preconditioned Domain Decomposition Schemes for Solving Nonlinear Boundary Value Problems

G. A. GRAVVANIS*
Department of Informatics
Athens University of Economics and Business
GR 104 34 Athens, Greece
and
Department of Mathematics, University of the Aegean
GR 832 00 Karlovasi, Samos, Greece
gag@aueb.gr, gag@aegean.gr

Abstract—A new class of inner-outer iterative procedures in conjunction with Picard-Newton methods based on explicit preconditioning iterative methods for solving nonlinear systems is presented. Explicit preconditioned iterative schemes, based on the explicit computation of a class of domain decomposition generalized approximate inverse matrix techniques are presented for the efficient solution of nonlinear boundary value problems on multiprocessor systems. Applications of the new composite scheme on characteristic nonlinear boundary value problems are discussed and numerical results are given. © 2003 Elsevier Science Ltd. All rights reserved.

Keywords—Arrow-type matrix, Domain decomposition, Approximate factorization procedures, Approximate inverse matrix techniques, Preconditioning, Parallel iterative methods.

1. INTRODUCTION

Many engineering and scientific problems are described by sparse systems of algebraic equations, which arise when solving partial differential equations (PDEs). This category of problems represents a large class of commonly occurring problems in mathematical physics and engineering, i.e., heat conduction, and chemical reaction, laminar flow on non-Newtonian fluids, reactor physics, moving boundary problems (melting and freezing), percolation problems, diffusion theory, and plasma physics problems, etc. Hence, sparse matrix computations, which have inherent parallelism, are therefore of central importance in scientific and engineering computing and the need for high performance computing has had some effect on the design of modern computer systems.

An important achievement over the last decades is the appearance and use of preconditioning methods for the numerical solution of sparse systems. The well-known preconditioning methods based on incomplete factorization or successive over-relaxation (SOR) or approximate inverses by minimizing the Frobenious norm of the error or the residual for fixed sparsity pattern, cf. [1–4], are very difficult to implement them on parallel systems, cf. [2–9]. In the case of polyno-

*Current address: Department of Information and Communication Systems Engineering, University of the Agean, GR-83200 Karlovassi, Samos, Greece.

PII: S0898-1221(02)00338-3

mial preconditioners, although they have inherent parallelism, they do not improve considerably the rate of convergence.

In recent years, research efforts have been directed on the production of numerical software, for solving sparse systems of algebraic equations on parallel machines, i.e., vector or array processors and systolic arrays. Recently, explicit approximate inverse preconditioning methods have been extensively used for solving efficiently sparse systems, resulting from the finite difference of finite element discretization of PDEs in two and three space variables, on multiprocessor systems, cf. [10–15]. The effectiveness of the explicit preconditioned schemes is related to the fact that the approximate inverse exhibits a similar "fuzzy" structure and are close approximates to the coefficients matrix.

Domain decomposition techniques have also been used for solving boundary value problems on regular or irregular domains. A domain is decomposed into smaller regular domains and the resulting system of algebraic equations is of so-called arrow-type systems, which occur in practice, cf. [4–6,11,16–19], and interesting discussions have been given in [4,6,11,19–25].

The purpose of this work is the derivation of a new class of composite iterative schemes based on inner-outer iterative procedures in conjunction with the known Picard-Newton methods, leading to improved composite iterative schemes for solving efficiently nonlinear boundary value problems. The Picard-Newton method can be coupled with the explicit preconditioned schemata. The effectiveness of the preconditioned methods relies on the construction and use of efficient preconditioner factors in the sense that the preconditioners are close approximates to the inverse of the coefficient matrix.

The derivation of suitable parallel methods was the main objective for which several forms of an approximate inverse of a given matrix, based on approximate LU-type factorization procedures have been proposed, cf. [10,11,14,15,26]. The main motive for the derivation of the approximate inverse arrow-type matrix techniques lies in the fact that they can be used in conjunction with explicit preconditioned iterative schemes and are suitable for solving linear systems on parallel and vector processors.

The cost effectiveness of explicit preconditioned iterative schemes over parallel direct solution methods is now commonly accepted. It is known that approximate factorization procedures and inverse matrix algorithms are in general complicated. However, as the demand for solving linear or nonlinear initial/boundary value problems grows, the need to use efficient sparse equations solvers becomes one of great importance, cf. [12,15].

In Section 2, we introduce domain decomposition approximate inverse matrix techniques based on approximate LU-type factorization procedures without inverting the related decomposition factors. In Section 3, composite iterative schemes in conjunction with the known Picard-Newton methods for solving nonlinear problems are presented. In Section 4, explicit preconditioned conjugate gradient-type methods based on approximate inverse matrix techniques are given. Finally, the performance and applicability of the new proposed explicit preconditioned domain decomposition schemes is discussed by solving a characteristic two-dimensional nonlinear boundary value problem and numerical results are presented.

2. DOMAIN DECOMPOSITION APPROXIMATE INVERSE MATRIX TECHNIQUES

In this section, we present algorithmic procedures for computing the elements of the approximate inverse, based on approximate LU-type factorization procedures, cf. [10,11,14,15,26].

Let us consider the linear system, i.e.,

$$Au = s, \tag{2.1}$$

where A is a sparse arrow-type $(n \times n)$ matrix of the following form:

$$A=\begin{bmatrix} b_1 & c_{1,1} & \cdots & c_{1,l_1} & & & & u_{1,1} & \cdots & u_{1,l_2} \\ a_{1,1} & & & & \ddots & & & \vdots & & \vdots \\ \vdots & & & & & \ddots & & u_{n-l_1-l_2,1} & & \\ a_{1,l_1} & & & & & & \ddots & & & u_{n-l_1-1,l_2} \\ & & & & & & & & & c_{n-l_1,l_1} \\ v_{1,1} & \cdots & v_{n-l_1-l_2,1} & & & & & & & \\ \vdots & & & \ddots & & & & & & c_{n-1,1} \\ v_{1,l_2} & & \cdots & & v_{n-l_1-1,l_2} & a_{n-l_1,l_1} & \cdots & a_{n-1,1} & & b_n \end{bmatrix} \tag{2.2}$$

According to the structure of the coefficient matrix A, "fill-in" terms are required during the decomposition process.

Let us now assume the approximate factorization of the coefficient matrix A such that, viz.,

$$A \approx LU, \tag{2.3}$$

$$L=\begin{bmatrix} g_1 & & & & & & & & \\ h_{1,1} & \ddots & & & & & & & \\ \vdots & & \ddots & & & & & & \\ h_{1,l_1} & & & \ddots & & & & & \\ & & & & \ddots & & & & \\ & 0 & & & & \ddots & & & \\ t_{1,1} & \cdots & t_{n-l_1-l_2,1} & & & & \ddots & & \\ \vdots & & & \ddots & & & & \ddots & \\ t_{1,l_2} & & \cdots & & t_{n-l_1-1,l_2} & h_{n-l_1,l_1} & \cdots & h_{n-1,1} & g_n \end{bmatrix} \tag{2.4}$$

retaining exactly the same number of nonzero entries, by applying the so-called "position-principle" in the factorization process, where L and U, cf. (2.4),(2.5), are sparse strictly lower and upper (with main diagonal unity elements) triangular matrices of the same profile as the coefficient matrix A, cf. (2.2). Then, the elements of the L and U decomposition factors can be computed by the domain decomposition approximate LU-type factorization procedure (henceforth called the DODALUFA algorithm).

$$U=\begin{bmatrix} 1 & e_{1,1} & \cdots & e_{1,l_1} & & & & f_{1,1} & \cdots & f_{1,l_2} \\ & \ddots & & & \ddots & & & \vdots & & \\ & & \ddots & & & \ddots & & f_{n-l_1-l_2,1} & & \vdots \\ & & & \ddots & & & \ddots & & \ddots & \\ & & & & \ddots & & & & & f_{n-l_1-1,l_2} \\ & & & & & \ddots & & & & e_{n-l_1,l_1} \\ & & & & & & \ddots & & & \vdots \\ & & & & & & & \ddots & & e_{n-1,1} \\ & & & & & & & & \ddots & 1 \end{bmatrix} \tag{2.5}$$

The memory requirements of the DODALUFA algorithm is $\approx O(2l_1 + 2l_2 + 1)n$ words and the computational work required by the factorization process is $\approx O(3l_1 + 3l_2 + 2)n$ multiplicative operations. The DODALUFA algorithm can be implemented on multiprocessor systems by following certain parallel decomposition techniques, cf. [6,7,9].

Let $M^{\delta l} \equiv (\mu_{i,j})$, $i \in [1,n]$, $j \in [\max(1, i-\delta l+1), \min(n, i+\delta l-1)]$, an $[n \times (2\delta l - 1)]$ matrix, be the approximate inverse of the coefficient matrix A, i.e.,

$$M^{\delta l} \approx (LU)^{-1}. \tag{2.6}$$

The elements of the approximate inverse can be determined by retaining a certain number of elements of the inverse, i.e., only δl elements in the lower part and $\delta l - 1$ elements in the upper part of the inverse (by applying the so-called "position-principle"), next to the main diagonal, the remaining elements not being computed at all. Then, the elements of the approximate inverse can be computed by solving recursively the following systems:

$$M^{\delta l} L = U^{-1} \quad \text{and} \quad U M^{\delta l} = L^{-1}, \qquad \delta l \in [1, \ldots, n], \tag{2.7}$$

without inverting the decomposition factors L and U, cf. [10,11,14,15,26].

It should be noted that the computation of the elements $\mu_{i,j}$ of the approximate inverse, using a "fish-bone" computational procedure, can be successively determined as follows. From the equations of (2.7) for $i = n, \ldots, 1$ and $j = \max(1, i-\delta l+1), \ldots, \min(n, i+\delta l-1)$, respectively, we can obtain the elements of the approximate inverse, cf. [8,9,14].

Then, the elements of the approximate inverse can be computed by the so-obtained domain decomposition generalized approximate inverse matrix technique (henceforth called the DODGAIM algorithm).

In order to solve efficiently linear systems, the DODGAIM algorithm has to be redesigned, by using a moving window shifted from bottom to top, such that only $[n \times (2\delta l - 1)]$-vectors are retained in storage, cf. [13,27]. This optimized form of the domain decomposition generalized approximate inverse matrix (henceforth called the ODODGAIM algorithm) is particularly effective for solving "narrow-banded" sparse systems of very large order, i.e., $\delta l \ll n/2$.

The memory requirements of the ODODGAIM algorithm are $\approx [n \times (2\delta l - 1)]$ words and the computational work involved is $\approx O[(l_1 + l_2 + 1)\delta l]n$ multiplicative operations.

It should be noted that according to the proposed computational strategy, this class of approximate inverses can be considered that includes various families of approximate inverses having in mind the desired requirements of accuracy, storage, and computational work as can be seen by the following diagrammatic relationship, i.e.,

$$\begin{array}{ccccccc} & & \underline{\text{class I}} & & \underline{\text{class II}} & & \underline{\text{class III}}, \\ A^{-1} \equiv M & \leftarrow & M_e^{\delta l} & \leftarrow & M^{\delta l} & \leftarrow & M_i, \end{array} \tag{2.8}$$

where the entries of $M_e^{\delta l}$ have been retained after the computation of the exact inverse, while the entries of $M^{\delta l}$ have been computed and retained during the computational procedure of the approximate inverse. The diagonal inverse, M_i, was computed based on the inversion of the diagonal entries only of the L decomposition factor, i.e., $\delta l = 1$, resulting in a fast inverse algorithm.

It should be mentioned that if $v_{i,j} = 0$ and $u_{i,j} = 0$, then the DODALUFA and ODODGAIM algorithms are reduced to BLUFA and OAIBM algorithms, respectively, cf. [10], for solving banded systems. It should be also noted that, if $l_1 = 0$ and $l_2 = 1$, then the DODALUFA and ODODGAIM algorithms are reduced to ALUFA and OAIAM algorithms, respectively, cf. [11].

3. COMPOSITE ITERATIVE SCHEMES FOR NONLINEAR PROBLEMS

Let us consider a class of nonlinear boundary value problems defined by the nonlinear elliptic PDE in two space-variables, i.e.,

$$Lu = f(u), \qquad (x, y) \in R, \tag{3.1}$$

subject to the boundary conditions

$$\alpha u + \beta \frac{\partial u}{\partial \zeta} = \gamma, \qquad (x, y) \in \partial R, \tag{3.2}$$

where L is a linear partial differential operator.

We may linearize the problem by the Picard method, i.e.,

$$Lu^{(k+1)} = f\left[u^{(k)}\right] \tag{3.3}$$

or the Newton method, i.e.,

$$Lu^{(k+1)} - f'\left[u^{(k)}\right]u^{(k+1)} = f\left[u^{(k)}\right] - u^{(k)}f'\left[u^{(k)}\right]. \tag{3.4}$$

Assuming that a network of mesh spacing h_x, h_y in the X, Y directions, respectively, is superimposed over the region R and, using central finite difference scheme, then the above iterative schemes lead to sparse systems which can be written equivalently as

$$A_k u^{(k+1)} = s\left(u^{(k)}\right), \qquad k > 0, \tag{3.5}$$

where A_k is of form (2.2), with $A_k = A$ for the Picard iteration. A system of form (3.5) can be explicitly solved by means of composite "inner-outer" iterative schemes, i.e., Picard-Newton and exact inversion procedures resulting in one-level iteration or Picard-Newton and explicit preconditioned iterative schemata based on explicit approximate inverse procedures yielding the usual two-level iteration scheme.

Let us consider the nonlinear iterative scheme

$$A_k\left(u^{(k+1)} - u^{(k)}\right) = -G\left(u^{(k)}\right), \qquad k > 0, \tag{3.6}$$

where the matrix A_k can be split as $A_k = B_k - C_k$. Provided that the matrix B_k is nonsingular, we have, cf. [28],

$$A_k^{-1} = \left(I - B_k^{-1}C_k\right)^{-1} B_k^{-1} \approx \left[I + H_k + H_k^2 + \cdots + H_k^{m_k-1}\right] B_k^{-1}, \tag{3.7}$$

where $H_k = B_k^{-1}C_k$, $k > 0$, I is the identity matrix and only m_k first terms have been retained in the expansion of $(I - B_k^{-1}C_k)^{-1}$. Therefore, an explicit iterative scheme is derived, i.e.,

$$u^{(k+1)} - u^{(k)} = -\left(I + H_k + H_k^2 + \cdots + H_k^{m_k-1}\right] B_k^{-1} G\left(u^{(k)}\right), \qquad k > 0, \tag{3.8}$$

which represents the composite iteration in which at the kth stage starting from $u^{(k)}, m_k$ steps of the linear inner iterations are computed in order to approximate a solution of the outer iteration. Choosing $B_k^{-1} = (M^{\delta l})_k$ depending upon k and retaining only the first term in the expansion of (3.8), we obtain the first-order Newton-ODODGAIM iterative scheme, viz.,

$$u^{(k+1)} - u^{(k)} = -M^{\delta l} G\left(u^{(k)}\right), \qquad k > 0. \tag{3.9}$$

The Newton-DODGEIM scheme can be easily derived from (3.6) assuming that $M \equiv A_k^{-1} = (LU)^{-1}$, cf. (2.8), and is given by

$$u^{(k+1)} - u^{(k)} = -MG\left(u^{(k)}\right), \qquad k > 0. \tag{3.10}$$

It can be easily seen that the proposed composite "inner-outer" iterative scheme in the case of the exact inversion reduces to an equivalent one-level iteration. While for the case of approximate inversion, the "inner-outer" iterative scheme reduces to the usual two-level iteration and the explicit preconditioned generalized conjugate gradient-type iterative schemes can be used.

4. EXPLICIT PRECONDITIONED ITERATIVE METHODS

In this section, we present a class of explicit preconditioned iterative schemes based on the ODODGAIM techniques of Section 2 for solving the nonlinear systems (3.5).

The explicit preconditioned generalized conjugate gradient square (EPGCGS) algorithm can be expressed by the following compact scheme.

Let u_0 be an arbitrary initial approximation to the solution vector u. Then,

$$\text{set } u_0 = 0 \text{ and } e_0 = 0, \text{ compute } r_0 = M^{\delta l}(s - Au_0), \tag{4.1}$$

$$\text{set } \sigma_0 = r_0 \text{ and } p_0 = (\sigma_0, r_0)\,. \tag{4.2}$$

Then, for $i = 0, 1, \ldots,$ (until convergence) compute the vectors $u_{i+1}, r_{i+1}, \sigma_{i+1}$ and the scalar quantities α_i, β_{i+1} as follows:

$$\text{form } q_i = A\sigma_i, \text{ calculate } \alpha_i = \frac{p_i}{(\sigma_0, M^{\delta l} q_i)}, \tag{4.3}$$

$$\text{compute } e_{i+1} = r_i + \beta_i e_i - \alpha_i M^{\delta l} q_i, \tag{4.4}$$

$$d_i = r_i + \beta_i e_i + e_{i+1}, \text{ and } u_{i+1} = u_i + \alpha_i d_i, \tag{4.5}$$

$$\text{form } q_i = Ad_i, \text{ compute } r_{i+1} = r_i - \alpha_i M^{\delta l} q_i, \tag{4.6}$$

$$\text{set } p_{i+1} = (\sigma_0, r_{i+1}), \text{ evaluate } \beta_{i+1} = \frac{p_{i+1}}{p_i}, \tag{4.7}$$

$$\text{compute } \sigma_{i+1} = r_{i+1} + 2\beta_{i+1} e_{i+1} + \beta_{i+1}^2 \sigma_i. \tag{4.8}$$

The computational complexity of the EPGCGS method, assuming that $M^{\delta l}$ can be compactly stored in $n \times (2\delta l - 1)$ diagonal vectors is $\approx O(4\delta l + 4l_1 + 4l_2 + 11)n$ mults $+8n$ adds$]\nu$ operations, where ν denotes the number of iterations required for convergence to a predetermined tolerance level.

In the following, we present a modified form of the van der Vorst BICGSTAB method, cf. [29], using the explicit preconditioner $M^{\delta l}$. This modified method, henceforth called the explicit preconditioned biconjugate conjugate gradient-STAB (EPBI-CGSTAB) method, can be expressed by the following compact scheme.

Let u_0 be an arbitrary initial approximation to the solution vector u. Then,

$$\text{set } u_0 = 0, \text{ compute } r_s = s - Au_0, \tag{4.9}$$

$$\text{set } r_0' = r_0,\ \rho_0 = \alpha = \omega_0 = 1, \text{ and } v_0 = p_0 = 0. \tag{4.10}$$

Then, for $i = 0, 1, \ldots,$ (until convergence) compute the vectors u_i, r_i and the scalar quantities α, β, ω_i as follows:

$$\text{calculate } \rho_i = (r_0', r_{i-1})\,, \text{ and } \beta = \frac{\rho_i/\rho_{i-1}}{\alpha/\omega_{i-1}}, \tag{4.11}$$

$$\text{compute } p_i = r_{i-1} + \beta\,(p_{i-1} - \omega_{i-1} v_{i-1})\,, \tag{4.12}$$

$$\text{form } y_i = M^{\delta l} p_i, \text{ and } v_i = Ay_i,\ \alpha = \frac{\rho_i}{(r_0', v_i)}, \tag{4.13}$$

$$\text{compute } x_i = r_{i-1} - \alpha v_i, \text{ form } z_i = M^{\delta l} x_i \text{ and } t_i = Az_i, \tag{4.14}$$

$$\text{set } \omega_i = \frac{(M^{\delta l} t_i, M^{\delta l} x_i)}{(M^{\delta l} t_i, M^{\delta l} t_i)}, \tag{4.15}$$

$$\text{compute } u_i = u_{i-1} + \alpha y_i + \omega_i z_i \text{ and } r_i = x_i - \omega_i t_i. \tag{4.16}$$

The computational complexity of the EPBI-CGSTAB method, assuming that $M^{\delta l}$ can be compactly stored in $n \times (2\delta l - 1)$ diagonal vectors is $\approx [(6\delta l + 4l_1 + 4l_2 + 12)n$ mults $+6n$ adds$]\nu$

operations, where ν denotes the number of iterations required for convergence to a predetermined tolerance level. The effectiveness of the explicit preconditioned iterative methods using the ODODGAIM algorithm is related to the fact that the approximate inverse of the original sparse coefficient matrix A exhibits a similar "fuzzy" structure as the coefficient matrix A.

The convergence analysis of similar explicit approximate inverse preconditioning has been presented in [8,27].

Let us now consider that the M_i class of approximate inverse, i.e., $\delta l = 1$, is used as a preconditioner for the explicit preconditioned generalized conjugate gradient-type method. Assuming a PRAM linear array model with n processors is used, then the computation of the elements of this class of inverse can be done in $O(1)$, i.e., constant time. Additionally, in the implementation of the EPGCG-type iterative scheme, the inner product can be performed in $O(\log n)$, i.e., in the case of a linear array with n processors, using the prefix computation model.

5. NUMERICAL RESULTS

In this section, we examine the applicability and effectiveness of the new proposed composite explicit preconditioned domain decomposition approximate inverse preconditioning schemes by solving the following characteristic problems in two dimensions.

MODEL PROBLEM 1. Let us consider a 2D nonlinear elliptic PDE

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = e^u, \qquad (x, y) \in \Omega, \tag{5.1}$$

subject to boundary conditions

$$u(x, y) = 0, \qquad (x, y) \in \partial\Omega, \tag{5.1a}$$

where Ω is the unit square and $\partial\Omega$ denotes the boundary of Ω.

Equation (5.1) arises in magnetohydrodynamics and is of physical interest in diffusion-reaction, vortex problems and electric charge consideration, cf. [1].

The linearized Picard and quasi-linearized Newton iterations are outer iterative schemes of the form

$$L_h u^{(k+1)} = e^{u^{(k)}} \qquad \text{and} \qquad L_h u^{(k+1)} - e^{u^{(k)}} u^{(k+1)} = \left(1 - u^{(k)}\right) e^{u^{(k)}}, \tag{5.2}$$

respectively, with L_h denoting the finite difference operator.

Table 1. The performance of the composite "inner-outer" iterative scheme using the EPGCGS and EPBICG-STAB method for Model Problem 1.

Method	n	m	δl	Picard Method		Newton Method	
				Outer Iterative	Inner Iterative	Outer Iterative	Inner Iterative
EPGCGS	961	32	1	2	9	2	10
			3	2	9	2	8
			6	2	5	2	5
	3969	64	1	2	10	2	10
			3	2	8	2	8
			6	2	5	2	5
EPBICG-STAB	961	32	1	3	10	3	9
			3	2	12	2	8
			6	2	8	2	7
	3969	64	1	3	8	3	8
			3	2	8	2	8
			6	2	5	2	5

The domain $\Omega \cup \partial\Omega$ was decomposed into a number of subdomains and was covered by an nonoverlapping regular triangular network. The five-point finite difference discretization scheme with a row-wise ordering was used such that the length l_1 of the band was kept to low values, i.e., $l_1 = 3$. The resulting sparse system is of form (3.6).

The initial guess used was $u^{(0)} = 0$. The termination criterion for the inner iteration of the EPGCGS and EPBI-CGSTAB method was $\|r_i\|_\infty < 10^{-5}$, where r_i is the recursive residual. The criterion for the termination of the outer iteration was $\max_j |(u_j^{(k+1)} - u_j^{(k)})/(u_j^{(k+1)})| < 10^{-5}$, $j \in [1, n]$.

Numerical results for the model problem (5.1) are presented in Table 1 for the EPGCGS and the EPBI-CGSTAB methods for several values of order n, m and the "retention" parameter δl of the approximate inverse.

It should be mentioned that the convergence behavior of the EPGCGS and EPBI-CGSTAB methods in conjunction with the DODALUFA and ODODGAIM algorithms, is much better when the domain is subdivided into many subdomains.

MODEL PROBLEM 2. Let us also consider the 2D nonlinear elliptic PDE

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = u\left(1 - |u|^2\right), \qquad (x, y) \in \Omega, \tag{5.3}$$

subject to boundary conditions

$$u(x, 0) = u(0, y) = 0.0, \qquad u(x_{\max}, y) = u(x, y_{\max}) = 5.0. \tag{5.3a}$$

The linearized Picard and quasi-linearized Newton iterations are outer-iterative schemes, respectively, of the form

$$L_h u^{(k+1)} = u^{(k)}\left(1 - \left|u^{(k)}\right|^2\right), \qquad (x, y) \in \Omega, \tag{5.4}$$

and

$$L_h u^{(k+1)} + \left[\left|u^{(k)}\right|^2 + 2u^{(k)}\left|u^{(k)}\right| - 1\right] u^{(k+1)} = 2u^{(k)}\left|u^{(k)}\right|^2, \qquad (x, y) \in \Omega. \tag{5.5}$$

The initial guess used was $u^{(0)} = 0.1$, and the termination criteria of the inner and outer iterative schemes are the same with Model Problem 1. Numerical results for the model problem (5.3) are presented in Table 2 for the EPGCGS and the EPBI-CGSTAB methods for several values of order n, m and the "retention" parameter δl of the approximate inverse.

Table 2. The performance of the composite "inner-outer" iterative scheme using the EPGCGS and EPBI-CGSTAB method for Model Problem 2.

Method	n	m	δl	Picard Method		Newton Method	
				Outer Iterative	Inner Iterative	Outer Iterative	Inner Iterative
EPGCGS	961	32	1	3	9	2	2
			3	3	8	2	3
			6	3	8	2	2
	3969	64	1	4	6	2	2
			3	3	8	2	2
			6	3	5	2	2
EPBICG-STAB	961	32	1	3	10	2	2
			3	3	9	2	2
			6	3	8	2	2
	3969	64	1	3	7	2	2
			3	3	6	2	2
			6	3	5	2	2

It should be mentioned that the iterative GMRES scheme, cf. [30,31], although it has good stability, requires storage of all the basis vectors of the Krylov space and its performance is depending on the restart vectors used, thus making this method problem dependent, cf. [32].

Finally, we state that the explicit preconditioned domain decomposition scheme, using the DODALUFA and ODODGAIM algorithms, can be efficiently used for solving highly nonlinear initial/boundary value problems.

REFERENCES

1. O. Axelsson, *Iterative Solution Methods*, Cambridge University, (1994).
2. G.H. Golub and C. van Loan, *Matrix Computations*, Second Edition, The Johns Hopkins University, (1989).
3. M.J. Grote and T. Huckle, Parallel preconditioning with sparse approximate inverses, *SIAM J. Sci. Comput.* **18**, 838–853 (1997).
4. Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS, (1996).
5. J.J. Dongarra, I.S. Duff, D.C. Sorensen and H.A. van der Vorst, Solving linear systems on vector and shared memory computers, *SIAM*, Philadelphia, PA (1991).
6. J.J. Dongarra, P. Messina, D.C. Sorensen and R.G. Voight, *Proceedings of the Fourth SIAM Conference on Parallel Processing for Scientific Computing*, SIAM, Philadelphia, PA, (1989).
7. D.J. Evans and C. Sutti, Parallel computing: Methods algorithms and applications, In *Proceedings of the International Meeting on Parallel Computing*, Adam Hilger, (1988).
8. G.A. Gravvanis, Parallel matrix techniques, In *Computational Fluid Dynamics 98, Proc. of the Fourth ECCOMAS Computational Fluid Dynamics Conference, 7–11 September 1998, Athens, Greece, Volume 1*, (Edited by K.D. Papailiou, D. Tsahalis, J. Periaux, C. Hirsch and M. Pandoli), Part 1, pp. 472–477, (1998).
9. J.M. Ortega, *Introduction to Parallel and Vector Solution of Linear Systems*, Frontiers of Computer Science Series, (Edited by A.L. Rosenberg), Plenum, (1988).
10. G.A. Gravvanis, Banded approximate inverse matrix techniques, *Engineering Computations* **16** (3), 337–346 (1999).
11. G.A. Gravvanis, An approximate inverse matrix technique for arrowhead matrices, *I. J. Comp. Math.* **70**, 35–45 (1998).
12. G.A. Gravvanis, On the numerical modelling and solution of non-linear boundary value problems, In *Proceedings of the 10th Inter. Confer. on Numerical Methods in Thermal Problems*, U.K.,, *Volume X*, (Edited by R.W. Lewis and J.T. Cross), pp. 898–909, Pineridge, (1997).
13. G.A. Gravvanis, Explicit preconditioned methods for solving 3D boundary value problems by approximate inverse finite element matrix techniques, *I. J. Comp. Math.* **56**, 77–93 (1995).
14. G.A. Gravvanis and E.A. Lipitakis, A three-dimensional explicit preconditioned solver, *Computers Math. Applic.* **32** (2), 111–131 (1996).
15. E.A. Lipitakis and G.A. Gravvanis, A class of explicit preconditioned conjugate gradient methods for solving large finite element system, *I. J. Comp. Math.* **44**, 189–206 (1992).
16. I.S. Duff, A.M. Erisman and J.K. Reid, *Direct Methods for Sparse Matrices*, Oxford University Press, London, (1986).
17. D.J. Evans, Iterative methods for sparse systems, In *Sparsity and Its Applications*, (Edited by D.J. Evans), Cambridge University Press, (1985).
18. D. O'Leary and G. Stewart, Computing the eigenproblem and eigenvectors of arrowhead matrices, *J. Comp. Physics* **90**, 497–505 (1990).
19. H.R. Schwarz, *Finite Element Methods*, Academic, (1989).
20. P.E. Bjorstad and O. Widlund, Solving elliptic problems on regions partitioned into substructures, In *Elliptic Problem Solver II*, (Edited by G. Birkhoff and A. Schoenstadt), pp. 245–256, Academic, New York, (1984).
21. J.H. Bramble, J.G. Pasciak and A.H. Schatz, The construction of preconditioners for elliptic problems by substructuring I, *Math. Comp.* **47**, 103–134 (1986).
22. J.H. Bramble, J.G. Pasciak and A.H. Schatz, An iterative method for elliptic problems on regions partitioned in substructures, *Math. Comp.* **46**, 361–369 (1986).
23. T.F. Chan, Analysis of preconditioners for domain decomposition, *SIAM J. Num. Anal.* **24**, 382–390 (1987).
24. M. Dryja, A finite element capacitance method for elliptic problems on regions partitioned into subregions, *Num. Math.* **44**, 153–168 (1984).
25. M. Dryja, A capacitance matrix method for Dirichlet problem on polygonal region, *Num. Meth.* **39**, 51–64 (1982).
26. E.A. Lipitakis and G.A. Gravvanis, Explicit preconditioned iterative methods for solving large unsymmetric finite element systems, *Computing* **54**, 167–183 (1995).
27. G.A. Gravvanis, The rate of convergence of explicit approximate inverse preconditioning, *I. J. Comp. Math.* **60**, 77–89 (1996).
28. J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic, (1970).

29. H.A. van der Vorst, BI-CGSTAB: A fast and smoothly converging variant BI-CG for the solution of non-symmetric linear systems, *SIAM J. Sci. Stat. Comp.* **13**, 631–644 (1992).
30. H.A. van der Vorst and C. Vuik, GMRESR: A family of nested GMRES methods, *Numer. Linear Alg. Appl.* **1** (4), 369–386 (1994).
31. C. Vuik and H.A. van der Vorst, A comparison of some GMRES-like methods, *Linear Alg. Appl.* **160**, 131–162 (1992).
32. D.R.J. Owen, *Iterative and Parallel Equation Solutions Strategies*, R News, Issue No. 4, Rockfield Software Ltd, Swansea, (1998).
33. N.I. Buleev, A numerical method for solving 2D and 3D diffusion equations, *Mathem. Sbornik* **51**, 227–238 (1960).
34. E.G. D'Jakanov, On an iterative method for the solution of finite difference equations, *Dokl. Akad. Nauk. SSR* **138**, 522–525 (1961).
35. G.A. Gravvanis, Solving non-linear boundary value problems in three dimensions by explicit preconditioning, In *Proc. of the Intern. Confer. on Advanced Computational Methods in Engineering*, (Edited by R. Van Keer, B. Verhegghe, M. Hogge and E. Noldus), Part 2, pp. 755–762, Shaker B.V., (1998).