Available online at www.sciencedirect.com
ScienceDirect
Computers and Mathematics with Applications 53 (2007) 1–11
An International Journal
computers & mathematics with applications
www.elsevier.com/locate/camwa

Large sparse linear systems arising from mimetic discretization

F.F. Hernández[a], J.E. Castillo[b], G.A. Larrazábal[c,*]

[a] *Science Department, University of Oriente, Anzoátegui, Venezuela*
[b] *Computational Sciences Research Center (CSRC), 5500 Campanile Drive, San Diego State University, San Diego, CA 92182-1245, USA*
[c] *Multidisciplinary Center of Scientific Visualization and Computing (CeMViCC), Faculty of Sciences and Technology (FACYT), Av. Montes de Oca, No. 120-267, Edf. FACYT, C.P. 2001, Carabobo University, Valencia, Venezuela*

Received 30 November 2005; received in revised form 27 August 2006; accepted 30 August 2006

Abstract

In this work we perform an experimental study of iterative methods for solving large sparse linear systems arising from a second-order 2D mimetic discretization. The model problem is the 2D Poisson equation with different boundary conditions. We use GMRES with the restarted parameter and BiCGstab as iterative methods. We also use various preconditioning techniques including the robust preconditioner ILUt. The numerical experiments consist of large sparse linear systems with up to 643 200 degrees of freedom.

Keywords: Mimetic methods; Iterative methods; Preconditioning; Boundary value problem; Staggered grid

1. Introduction

Mathematical models of continuum mechanics problems are often described by boundary value problems, expressed as systems of partial differential equations or integral equations. To facilitate their numerical solution, these equations can be discretized by any one of a large number of techniques. Standard methods include various finite difference and finite element approaches. Often, these traditional methods are applied by discretizing the defining system of equations directly. One disadvantage of such an approach is that the discretization scheme that is selected may have little connection with the underlying physical problem.

The mimetic methods produce discrete differential operators that mimic some fundamental properties of divergence and gradient operators, among others. A wide summary of methods related with the construction of mimetic discrete operators can be found in [1–4], and some of their applications in [5,6]. Recent advances in mimetic or compatible discretizations have been discussed in three recent workshops (see http://www.sci.sdsu.edu/compscims/MIMETIC and http://www.math.unm.edu/stanly) with one of the main focus of discussion being the importance of differential geometry in producing mimetic or compatible discretizations and some practical use in 3D computational electromagnetism.

Recently, in [7], a methodology to build discrete operators that satisfy a discrete analog of the divergence theorem was presented. In this technique, higher-order mimetic differential operators can be built in the interior as well as on

* Corresponding author. Tel.: +58 241 8678243; fax: +58 241 8678243.
E-mail address: glarraza@uc.edu.ve (G.A. Larrazábal).

0898-1221/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2006.08.034

the boundary for uniform grids in 1D. This technique has been extended to non-uniform grids [8]. A second-order mimetic discretization is constructed and is compared to another second-order mimetic discretization applied to an elliptic boundary value problem in one dimension in [9]. Such a technique, which employs a matrix formulation to incorporate mimetic constraints, produces approximations whose order at a grid boundary is equal to that in the grid interior. Mimetic schemes for the steady state diffusion equation based on Castillo–Grone approach have been reported in [9–11]. Each one of these articles provides evidence of quadratic convergence rates for such schemes.

In addition, the computational solution of linear equation systems is one of the most important research areas nowadays. This is true specially for those systems that come from modelling physical problems with certain complexity, which are generally associated with industrial applications, such as fluid dynamics and structural mechanics. These applications are governed by partial differential equations which are discretized using a variety of numerical methods. A linear system of equations arise from the discretization process.

Iterative methods are among the most appropriate computational tools used to solve these problems, because they allow fulfilling a specific precision and do not introduce fill-in (thus having minimum memory requirements). Furthermore, they are highly parallelizable, which differs from the more traditional direct methods. However, it is well known that iterative methods may have very slow convergence, which sometimes is not even assured. In order to overcome this disadvantage, it is required to apply some preconditioning techniques that accelerate the convergence and improve the performance, while having a low impact on the overall computational cost.

In this work, we use the second-order operators obtained by Castillo [11,12]. To obtain these operators we use the Castillo–Grone [7] approach and we use these results to build a mimetic discretization applied to an elliptic boundary value problem in two dimensions. For details about the construction of the 2D operators see [13]. We make an experimental study about the sparse linear systems of equations arising from Poisson equation in 2D. We would like to mention that we are already using 2D operators to solve elastic wave propagation problems (see [14]) with very promising results which are in preparation to be submitted for publication very soon. We are also extending the operators to 3D for elastic wave propagation, oil reservoir and porous media flow problems. Also an approach to construct higher dimension operators using graph theory and Kronecker products is also in preparation to be submitted for publication [15].

2. Mimetic methods

Mimetic methods begin by first discretizing the continuum theory underlying the problem. By "discretizing the continuum theory", we mean that mimetic methods initially construct a discrete mathematical analog of a relevant description of continuum mechanics. Typically, this description takes the form of a physical conservation or constitutive law. The discrete form of the conservation or constitutive law constrains the structure that discrete operators can take. After building discrete operators that obey the discretized physical law, these mimetic operators can then be substituted into the system of partial differential equations or integral equations. This yields a mimetic discretization for the partial differential equations which automatically satisfies the discrete version of the physical law.

As a result, discretizations obtained using mimetic methods tend to replicate much of the behavior found in the actual continuum problem. Because the physical laws are, in effect, built into the discretization, mimetic methods turn out to be good candidates for modelling even challenging problems such as those involving anisotropic or strongly inhomogeneous material properties [16]. Furthermore, the physical basis for mimetic discretizations tends to reduce the occurrence of various non-physical numerical artifacts that can occur when using a traditional discretization technique.

Given a discrete conservation law and a uniform grid, one can readily create mimetic approximations of high order on the interior of a region. Building mimetic operators with comparably high-order approximation at or near the boundary of a region, however, has been considered challenging even in the simplest case, namely, that of a uniform bi-dimensional grid. Recently, Castillo and Grone [7] have developed an approach to create mimetic operators that manages to overcome this difficulty. In addition, a wide summary of methods related with the construction of mimetic discrete operators are given in [1–4], as well as some of their applications [5,6].

A numerical mimetic method produces the following discretizations:

- Gradient, denoted by **G**
- Divergence, denoted by **D**
- Boundary operator, denoted by **B**.

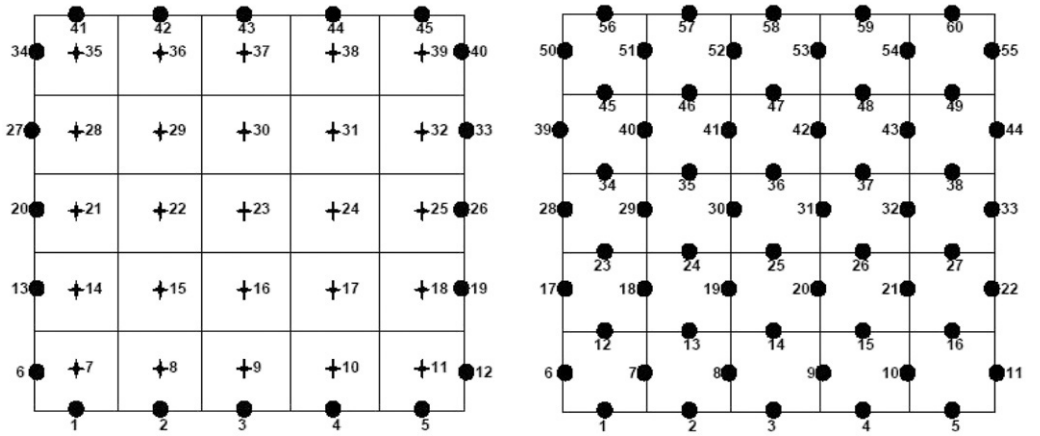

Fig. 1. Staggered grid for the divergence and gradient.

The mimetic divergence **D** maps function values evaluated at the nodes in 1D, edges in 2D and faces in 3D to values located at the cell centers of the grid. In contrast, the mimetic gradient **G** maps functions evaluated at the cell centers and boundary points to values located at the nodes in 1D, edges in 2D and faces in 3D. So, the operators **D** and **G** can be composed by matrix multiplication, resulting in a discrete second-order differential operator that maps function values evaluated at the cell centers and boundary points to values located at the cell centers. In the Fig. 1, we can observe an example of a staggered grid with 5×5 cells. We compute the divergence at the center of the cells in the left figure using the bullet points from the right figure and we compute the gradient at the bullet points in the right figure with the marked points in the left figure. In this paper we use the natural ordering.

Castillo and Grone [7] constructed a discrete analog of the conservation law as

$$\langle \mathbf{D}u, f\rangle_Q + \langle u, \mathbf{G}f\rangle_P = \langle \mathbf{B}u, f\rangle_I \tag{1}$$

for a generalized inner product $\langle x, y\rangle_A = y^{\mathrm{T}}Ax$. The positive definite matrices Q and P, produce various quadrature rules (see [7] and [17]) driving the form of **D** and **G**, respectively. In this context, the matrix **B** helps to maintain the global conservation requirement for the discrete conservation law. The local and global conservation requirement also constrains the structure of **D** and **G**.

3. Problem formulation and generalization

We want to solve the linear system $Ax = b$, where the matrix A is large and sparse. The linear system is obtained by discretizing the Poisson equation using the Castillo–Grone approach to Dirichlet and Robin problems on the rectangular domain $[0, L] \times [0, L]$.

For the 2D case, the Poisson equation has the following structure:

$$-\nabla^2 f = \mathbf{F}, \quad \text{on } \Omega = [0, L] \times [0, L], \tag{2}$$

with Robin boundary conditions:

$$\begin{cases} \left(\alpha f - \beta \dfrac{\partial f}{\partial y}\right)(x, 0) = \gamma(x, 0), \\ \left(\alpha f + \beta \dfrac{\partial f}{\partial x}\right)(L, y) = \gamma(L, y), \\ \left(\alpha f + \beta \dfrac{\partial f}{\partial y}\right)(x, L) = \gamma(x, L), \\ \left(\alpha f - \beta \dfrac{\partial f}{\partial x}\right)(0, y) = \gamma(0, y), \end{cases} \tag{3}$$

where $f = f(x, y)$ is the unknown function, $F = F(x, y)$ is the forcing or source function and $\partial\Omega$ is the boundary. In this work, the boundary is a unitary square and $\{\alpha, \beta, \gamma\}$ are parameters.

The Poisson equation is discretized using approximations for the continuum divergence $(\nabla\cdot)$ through the extended mimetic divergence $(\hat{\mathbf{D}})$, the continuum gradient (∇) through the order two mimetic gradient on the boundary $(\mathbf{G_2})$ developed by Castillo [9,10]. For the boundary conditions, the directional derivative on the border $(\frac{\partial}{\partial\vec{n}})$ is expressed in terms of the nabla operator and discretized using the mimetic operator $\mathbf{B_1G_2}$. This is shown as follows:

$$-\nabla^2 f = -\nabla\cdot(\nabla f) \approx -\hat{\mathbf{D}}\cdot(\mathbf{G_2})f = \mathbf{F} \quad \text{on } \Omega \tag{4}$$

$$\alpha f + \beta\vec{n}\cdot\nabla f \approx \alpha f + \beta\mathbf{B_1G_2}f = \gamma \quad \text{on } \partial\Omega. \tag{5}$$

For details about the construction of the 2D mimetic operators see [13].

3.1. Dirichlet problem

The linear system associated with discretization, with Dirichlet boundary condition has the following form:

$$(\mathbf{B_D} + \hat{\mathbf{D}}\mathbf{G_2})\mathbf{f} = \mathbf{b}, \tag{6}$$

where $\mathbf{B_D}$ is a matrix associated with the boundary conditions given in (3) and $\hat{\mathbf{L}}_\mathbf{2} = \hat{\mathbf{D}}\mathbf{G_2}$ is a matrix associated with the main equation, given in (2). The operator $\mathbf{B_D}$ is a coefficient matrix for the Dirichlet boundary condition, $\mathbf{G_2}$ is the mimetic gradient, $\hat{\mathbf{L}}_\mathbf{2} = \hat{\mathbf{D}}\mathbf{G_2}$ is the extended mimetic Laplacian, $\mathbf{f}$ is a vector column of the unknown functions and $\mathbf{b}$ is the right-hand side vector column of the system.

3.1.1. Coefficient matrix for the Dirichlet $(\mathbf{B_D})$ boundary condition

The entries of this matrix are the α parameters of the Dirichlet boundary conditions, which are set on the main diagonal. For a 5×5 grid, the order of this matrix is 45×45. On an $N\times N$ grid the matrix size is $(4N+N^2)\times(4N+N^2)$.

3.1.2. Column vector $\mathbf{f}$

This vector is obtained using the elements of the unknown function $\mathbf{f}$, which are set on an extended grid. For a 5×5 grid, the vector size is 45×1. This vector can be written as $f = (f_1, \ldots, f_{45})^t$. On an $N\times N$ grid, the vector size is $(4N + N^2)\times1$ and it can be written as $f = (f_i)^t$, $i = 1, \ldots, 4N + N^2$.

3.1.3. Column vector $\mathbf{b}$

This vector is generated using the function F, the right-hand side of the Poisson equation and the γ parameters. These parameters are set on the right member of the boundary conditions. For a 5×5 grid, the vector size is 45×1. On an $N\times N$ grid, the vector size is $(4N + N^2)\times1$.

3.1.4. Extended mimetic divergence $(\hat{\mathbf{D}})$

This operator is obtained from the mimetic divergence. On a 5×5 grid, the size is 45×60. On a $N\times N$ grid, this operator has six rows with zero at the beginning and at the end of the matrix. Also, this matrix has two rows with zero between each block generating eight groups, each one with two rows with zero, for a total of twenty rows with zero. On an $N\times N$ grid, there is $(N + 1)$ rows with zero at the beginning and at the end. Between each block, there are two rows with zero given a total of $2(N - 1)$. Thus, the total number of rows with zero is $4N$. For details about the construction of this 2D operator see [13].

3.1.5. Mimetic gradient $(\mathbf{G_2})$

This mimetic gradient is obtained using the methodology proposed by Castillo in [7,9]. On a 5×5 grid, the matrix size is 60×45. On an $N\times N$ grid, the matrix size is $(2N+2N^2)\times(4N+N^2)$. There are rows with only three non-zero entries $(-8/3,\ 3,\ -1/3)$. These entries are put on the bottom and left part of the grid. Also, other entries $(1/3,\ -3,\ 8/3)$ are put on the top and right part of the grid. The rest of the entries in this gradient coincide with the centered finite differences, i.e., there are only two values different from zero: -1 and 1. For details about the construction of this 2D operator see [13].

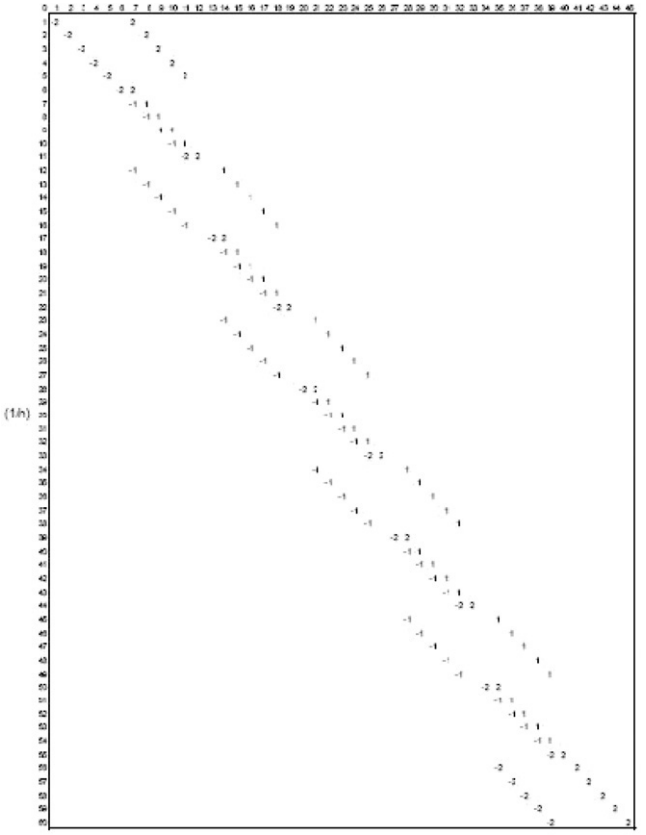

Fig. 2. Matrix **G**.

3.1.6. Extended mimetic Laplacian ($\hat{\mathbf{L}}_\mathbf{2}$)

This operator is obtained in the same way as that of the 1D case, i.e. we compute the matricial product ($\hat{\mathbf{D}}\mathbf{G_2}$). On a 5×5 grid, the matrix size is 45×45. On an $N \times N$ grid, the matrix size is $(4N + N^2) \times (4N + N^2)$.

3.2. Robin problem

Using the Robin boundary conditions, the linear system associated with the differential operator discretization is:

$$(\mathbf{B_D} + \mathbf{B_N}\mathbf{B_1}\mathbf{G_2} + \hat{\mathbf{D}}\mathbf{G_2})\mathbf{f} = \mathbf{b}. \tag{7}$$

3.2.1. Coefficient matrix for the Neumann ($\mathbf{B_N}$) *boundary conditions*

This matrix is used to preserve the boundary conditions matrix ($\mathbf{B_1}\mathbf{G_2}$) in the system (7). The matrix entries are β parameters, which are on the main diagonal. It is important to mention that, these entries can be overlapped at some place on the diagonal. On a 5×5 grid, the matrix size is 45×45. On an $N \times N$ grid, the matrix size is $(4N + N^2) \times (4N + N^2)$.

3.2.2. Matrix for the boundary operator ($\mathbf{B_1}$)

From (1) above we get that:

$$\mathbf{B} = \mathbf{Q}\hat{\mathbf{D}} + \mathbf{G^t}\mathbf{P}, \tag{8}$$

where **Q** is the weight matrix of the extended mimetic divergence operator, $\hat{\mathbf{D}}$ is the mimetic divergence extended, $\mathbf{G^t}$ is the gradient transposed and **P** is the weight matrix of the gradient operator **G**. For example, Fig. 2 shows the

Fig. 3. Matrix $\mathbf{B_1}$.

G matrix for a 5×5 grid, h is the discretization step. The matrix size is 60×45. This gradient is order one on the boundary and order two into the grid.

The weight matrix **Q** is the extended identity matrix for the second-order divergence. On a 5×5 grid, the matrix size is 45×60. On an $N \times N$ grid, the matrix size is $(4N + N^2) \times (2N + 2N^2)$.

The matrix **P** is the weight matrix of the gradient operator proposed by Castillo [9]. This matrix is square. On a 5×5 grid, the matrix size is 60×60. The only matrix entries different from zero are: $1/2$ and 1. These elements are on the main diagonal. On an $N \times N$ grid, the size of the weight matrix **P** is $(2N + 2N^2) \times (2N + 2N^2)$.

In [9,10], it is demonstrated that the matrix **B** can be written as $\mathbf{B} = \mathbf{B_1} + \mathbf{O(h)}$ for the second-order mimetic operators. The $\mathbf{B_1}$ matrix has only -1 and 1; and the rest of the entries are zeros. In addition, the solution of the Poisson problem using **B** or $\mathbf{B_1}$ is the same. In this paper we will use $\mathbf{B_1}$.

On a 5×5 grid, the size of matrix $\mathbf{B_1}$ is 45×60 (see Fig. 3). On an $N \times N$ grid, its size is $(4N + N^2) \times (2N + 2N^2)$.

Thus, for the Robin boundary condition the structure of the final coefficient matrix is different for both $\mathbf{B's}$. If we use the $\mathbf{B_1}$ then the resultant system is non-symmetric. If we use the **B** then the structure of the resultant system is symmetric.

4. Numerical experiments

Sparse linear systems were generated for the test cases with the dimensions depending on the number of intervals, N, in [0, L]. In all cases we used $L = 1$. For this study, we use two iterative methods, GMRES [18] and BiCGstab [19], and two different preconditioner, incomplete factorization proposed by Saad [20] and SParse Approximation of the Inverse (SPAI) proposed by Grote [21]. The restarted parameter of GMRES was 32 and the incomplete factorization were ILU(0) and ILUt(10). All these methods are used from UCSparseLib library [22]. As the stopping criterion we used the relative residual test: $\|r\|_2/\|b\|_2 < \text{tol}$. In all cases, we used $\text{tol} = 10^{-12}$. For ILUt(10) the numerical threshold was 10^{-3}. For the SPAI preconditioner, we used two approximations to control the density (0.1 and 0.4). The computed solution was compared with the exact solution.

4.1. Dirichlet problem

We use the Castillo–Grone approach to build a mimetic discretization on a uniform grid to solve

$$-\nabla^2 f = \left(\frac{\lambda^2}{e^{\lambda} - 1}\right) e^{\lambda x} \tag{9}$$

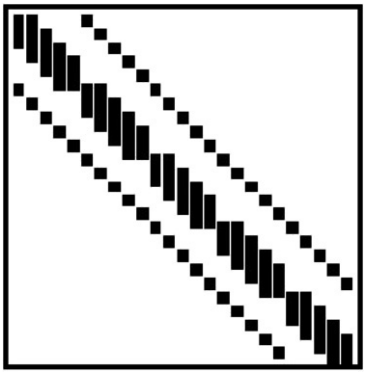

Fig. 4. Matrix structure for a 5×5 grid.

Table 1
Dirichlet, Description of the preconditioners (case $N = 100$)

Precond	Den (%)	Required space	*Precond* × *vector*
ILU(0)	0.05	1.292E+00	9.920E−02
ILUt(10)	0.24	3.476E+00	4.808E−01
SPAI(0.1)	0.18	2.323E+00	3.527E−01
SPAI(0.4)	0.04	8.592E−01	9.682E−02

Table 2
Dirichlet, Iterations of the iterative methods (case $N = 100$)

Name	Iterations
BiCGstab-ILU(0)	92
BiCGstab-ILUt(10)	17
BiCGstab-SPAI(0.1)	97
BiCGstab-SPAI(0.4)	139
GMRES-ILU(0)	202
GMRES-ILUt(10)	26
GMRES-SPAI(0.1)	252
GMRES-SPAI(0.4)	>500

subject to boundary conditions: $f(x, y) = \gamma(x, y)$, on the domain $[0, L] \times [0, L]$ where λ is a scalar. The exact solution is

$$f(x, y) = \frac{e^{\lambda x} - 1}{e^{\lambda} - 1}. \tag{10}$$

The function $\gamma(x, y)$ is simply the evaluation of $f(x, y)$ on the boundary of the domain.

In this case, the dimension of the system is N^2 where N is again the number of intervals in [0, 1]. Here, $\lambda = -1$. Fig. 4 show the symmetric structure matrix associated to linear system.

Table 1 shows the properties of the preconditioners for $N = 100$ (i.e., the dimension of the system is 10 000). The first column gives the preconditioner. The second column shows the density of the preconditioner. The third column shows the storage requirement (in Mbytes) for the preconditioner. The last column shows the operations (in Mflops) needed to apply the preconditioner to a vector in the iterative method. In Table 1, we see that ILU(0) and SPAI(0.4) are the preconditioners with the least density.

Table 2 shows the number of iterations to achieve convergence of the iterative methods for $N = 100$. The first column gives the preconditioned iterative method. The second column shows the number of iterations. We observe that BiCGstab-ILUt(10) is the most effective method. Also, GMRES-ILUT(10) performs well.

Table 3
Dirichlet, Description of the preconditioners (case $N = 200$)

Precond	Den (%)	Required space	*Precond* × *vector*
ILU(0)	0.01	5.179E+00	3.984E−01
ILUt(10)	0.06	1.412E+01	1.962E+00
SPAI(0.1)	0.04	9.377E+00	1.425E+00
SPAI(0.4)	0.01	3.473E+00	3.936E−01

Table 4
Dirichlet, Iterations of the iterative methods (case $N = 200$)

Name	Iterations
BiCGstab-ILU(0)	155
BiCGstab-ILUt(10)	37
BiCGstab-SPAI(0.1)	230
BiCGstab-SPAI(0.4)	287
GMRES-ILU(0)	603
GMRES-ILUt(10)	53
GMRES-SPAI(0.1)	863
GMRES-SPAI(0.4)	>1000

Table 3 shows the properties of the preconditioners for $N = 200$ (i.e., the dimension of the system is 40 000). The first column gives the preconditioner. The second column shows the density of the preconditioner. The third column shows the storage requirement (in Mbytes) for the preconditioner. The last column shows the operations (in Mflops) needed to apply the preconditioner to a vector in the iterative method.

In Table 3, we see the low density of the preconditioners. The preconditioner ILUt(10) has more density; it needed about 14 MB of storage.

Table 4 shows the number of iterations required to achieve convergence of the iterative methods for $N = 200$. The first column gives the preconditioned iterative method. The second column shows the number of iterations. In this case, BiCGstab-ILUt(10) and GMRES-ILUt(10) have the best performance.

4.2. Robin problem

We use the Castillo–Grone approach to build a mimetic discretization on a uniform grid to solve

$$-\nabla^2 f = \left(\frac{\lambda^2}{2(e^{\lambda}-1)}\right) e^{0.5\lambda(x+y)} \tag{11}$$

subject to boundary conditions: $\alpha f + \beta(\nabla f \cdot \vec{n}) = \gamma(x, y)$, on the domain $[0, L] \times [0, L]$ where α, β, λ are scalars. The exact solution is:

$$f(x, y) = \frac{e^{0.5\lambda(x+y)} - 1}{e^{\lambda} - 1}. \tag{12}$$

The function $\gamma(x, y)$ is defined on any side of the square domain, such that, the boundary condition is satisfied for $f(x, y)$ where n is the exterior normal vector. For example,

$$\gamma(0 \le x \le L, y = 0) = \left(\frac{1}{e^{\lambda} - 1}\right)[(\alpha - 0.5\beta\lambda)e^{0.5\lambda x} - \alpha].$$

In this case, the dimension of the system is $4N + N^2$, since we have N^2 inner unknowns and N unknowns on each boundary side. Here, N is again the number of intervals in [0, 1], and $\lambda = -1$, $\alpha = -e^{\lambda}$ and $\beta = \frac{e^{\lambda}-1}{\lambda}$. In the Fig. 5, we can see the non-symmetric structured matrix for 5×5 grid.

Table 5 shows the properties of the preconditioners for $N = 100$ (i.e., the dimension of the system is 10 400). The first column gives the preconditioner. The second column shows the density of the preconditioner. The third column

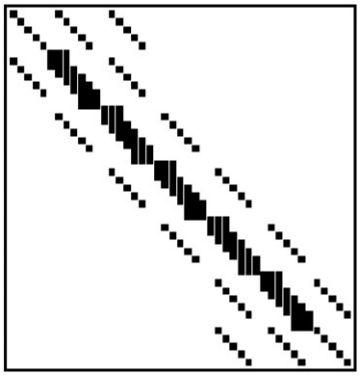

Fig. 5. Matrix structure for a 5×5 grid.

Table 5
Robin, Description of the preconditioners (case $N = 100$)

Precond	Den (%)	Required space	*Precond* × *vector*
ILU(0)	0.05	1.340E+00	1.024E−01
ILUt(10)	0.23	3.604E+00	4.981E−01
SPAI(0.1)	0.16	2.369E+00	3.586E−01
SPAI(0.4)	0.04	9.267E−01	1.065E−01

Table 6
Robin, Iterations of the iterative methods (case $N = 100$)

Name	Iterations
BiCGstab-ILU(0)	210
BiCGstab-ILUt(10)	33
BiCGstab-SPAI(0.1)	>1000
BiCGstab-SPAI(0.4)	>1000
GMRES-ILU(0)	>1000
GMRES-ILUt(10)	86
GMRES-SPAI(0.1)	>1000
GMRES-SPAI(0.4)	>1000

Table 7
Robin, Description of the preconditioners (case $N = 200$)

Precond	Den (%)	Required space	*Precond* × *vector*
ILU(0)	0.01	5.273E+00	4.048E−01
ILUt(10)	0.06	1.438E+01	1.996E+00
SPAI(0.1)	0.04	9.467E+00	1.437E+00
SPAI(0.4)	0.01	3.608E+00	4.129E−01

shows the storage requirement (in Mbytes) for the preconditioner. The last column shows the operations (in Mflops) needed to apply the preconditioner to a vector in the iterative method. In Table 5, we see that ILU(0) and SPAI(0.4) are the preconditioners with the least density.

Table 6 shows the number of iterations required to achieve convergence of the iterative methods for $N = 100$. The first column shows the preconditioned iterative method. The second column shows the number of iterations.

We observe that BiCGstab-ILUt(10) is the most effective method. Also, GMRES-ILUT(10) has good performance.

Table 7 shows the properties of the preconditioners for $N = 200$ (i.e., the dimension of the system is 40 800). The first column gives the preconditioner. The second column shows the density of the preconditioner. The third column

Table 8
Robin, Iterations of the iterative methods (case $N = 200$)

Name	Iterations
BiCGstab-ILU(0)	434
BiCGstab-ILUt(10)	65
BiCGstab-SPAI(0.1)	>1000
BiCGstab-SPAI(0.4)	>1000
GMRES-ILU(0)	>1000
GMRES-ILUt(10)	541
GMRES-SPAI(0.1)	>1000
GMRES-SPAI(0.4)	>1000

Table 9
Matrices description for the tests

Name	N	Dim	NZ	Den (%)
Test 1	400	161 600	804 800	0.0031
Test 2	800	643 200	3 209 600	0.0008

Table 10
Solver performance

Name	Den (%)	Required space	Operations	*Precond* × *vector*	Iter
Test 1	0.0153	5.612E+01	4.777E+01	7.976E+00	210
Test 2	0.0039	2.245E+02	1.919E+02	3.195E+01	409

shows the storage requirement (in Mbytes) for the preconditioner. The last column shows the operations (in Mflops) needed to apply the preconditioner to a vector in the iterative method.

In Table 7, we see the low density of the preconditioners. The preconditioner ILUt(10) has more density; it needed about 14 MB of storage.

Table 8 shows the number of iterations needed to achieve convergence of the iterative methods for $N = 200$. The first column shows the preconditioned iterative method. The second column shows the number of iterations. In this case, BiCGstab-ILUt(10) and GMRES-ILUt(10) have the best performance.

In general, the numerical experiments have shown that the BiCGstab-ILUt has the best performance for our examples of medium size. Moreover, we have built two tests with large sparse linear systems from Poisson equation with Robin boundary condition. In Table 9, we show the matrices description associated with these large sparse linear systems: **N** represents the number of cells, **dim** represents the matrix dimension, **NZ** represents the no nulls elements and **Den** represents the matrix density.

To solve the large sparse linear systems shown in the Table 9, we use the BICGstab + ILU(10) method due to the previous numerical results. In Table 10, we present the solver performance: **Den** is the preconditioner density, **Required Space** is the Mbytes used for preconditioner storage, **Operations** is the amount of operations (in Mflops) to build the preconditioner, *Precond* × *vector* is matrix–vector operations (in Mflops) to solve the triangular systems in each iterations of the iterative method and **Iter** is the iterations of the BiCGstab to reach convergence.

The results obtained shows the good performance of the BiCGstab-ILU(10). The preconditioner density is low with respect to dimension problems and the iterations of the iterative method are acceptable for our large sparse linear systems.

5. Conclusions

In this work, we have done an experimental study about solving the large sparse linear systems arising from a second-order 2D mimetic discretization method. This study was done using different iterative methods and preconditioners. Our experiments showed that the BiCGstab preconditioned with ILUt had the best performance for

our examples. Moreover, the preconditioner density is low with respect to dimension problems and the iterations of the iterative method are acceptable.

References

[1] M. Shashkov, Conservative Finite-Difference Methods on General Grids, CRC Press, Florida, USA, 1996.
[2] J. Castillo, J. Hyman, M. Shashkov, S. Steinberg, Fourth and six-order conservative finite difference approximation of the divergence and gradient, Applied Numerical Mathematics 37 (2001) 171–187.
[3] J. Hyman, M. Shashkov, Adjoint operators for the natural discretizations of the divergence, gradient and curl on logically rectangular grids, Applied Numerical Mathematics 25 (1997) 413–442.
[4] J. Hyman, M. Shashkov, Approximation of boundary conditions for mimetic finite-difference methods, Computers and Mathematics with Applications 36 (5) (1998) 79–99.
[5] J. Hyman, M. Shashkov, S. Steinberg, The numerical solution of diffusion problems in strongly heterogeneous non-isotropic materials, Journal of Computational Physics 132 (1997) 130–148.
[6] M. Shashkov, S. Steinberg, Solving diffusion equations with rough coefficients in rough grids, Journal of Computational Physics 129 (1996) 383–405.
[7] J. Castillo, R. Grone, A matrix analysis approach to higher-order approximations for divergence and gradients satisfying a global conservation law, SIAM Journal on Matrix Analysis and Applications 25 (1) (2003) 128–142.
[8] O. Montilla, C. Cadenas, J. Castillo, Matrix approach to mimetic discretizations for differential operators on non-uniform grids, Journal of Applied Numerical Mathematics (2005) (in press).
[9] J. Castillo, M. Yasuda, Linear system arising for second-order mimetic divergence and gradient discretizations, Journal of Mathematical Modelling and Algorithms 4 (2005) 67–82.
[10] J. Castillo, M. Yasuda, A comparison of the two matrix operator formulation for mimetic divergence and gradient discretizations, in: Proceedings of the Int. Conf. on Parallel and Distributed Processing Technique and Applications, vol. III, Las Vegas NV, 2003, pp. 1281–1285.
[11] M. Freites-Villegas, Un estudio comparativo de los metodos mimeticos para la ecuacion estacionaria de difusion, Thesis, Department of Mathematics, Universidad Central de Venezuela, 2004.
[12] J. Castillo, Private communication, 2005.
[13] H. Vu, A mimetic workbench for reservoir simulation, M.S. Thesis, San Diego State University, CA, 2006.
[14] O. Rojas, J. Castillo, R. Mellors, S. Day, Accurate and explicit modeling of a planar free surface boundary condition using mimetic finite differences. Poster presentation at Southern California Earthquake Center meeting, Palm Spring, CA, September, 2005.
[15] J.E. Castillo, R.D. Grone, High order mimetic gradient and divergence operators in higher dimensions, 2006 (in preparation).
[16] J. Hyman, M. Shashkov, Mimetic finite difference methods for Maxwell's equations and the equations of magnetic diffusion, Progress in Electromagnetics Research 32 (2001) 89–121.
[17] J. Hyman, S. Steinberg, The convergence of mimetic discretization for rough grids, Computers and Mathematics with Applications 47 (10–11) (2004) 1565–1610.
[18] Y. Saad, M.H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on Scientific and Statistical Computing 7 (3) (1986) 856–869.
[19] H.A. van der Vorst, Bi-CGstab: A fast and smoothly converging variant of the BiCG for the solution of nonsymmetric linear systems, SIAM Journal on Scientific and Statistical Computing 13 (2) (1992) 631–644.
[20] Y. Saad, ILUT: A dual threshold incomplete LU factorization, Numerical Linear Algebra with Applications 1 (4) (1994) 387–402.
[21] M. Grote, T. Huckle, Parallel preconditioning with sparse approximate inverse, SIAM Journal on Scientific Computing 18 (3) (1997) 838–853.
[22] G. Larrazábal, UCSparseLib: Una biblioteca numérica para resolver sistemas lineales dispersos, in: Simulación Numérica y Modelado Computacional, SVMNI, TC19–TC25, ISBN: 980-6745-00-0, 2004.