

Fast Multilevel Solvers for a Class of Discrete Fourth Order Parabolic Problems

Bin Zheng¹ · Luoping Chen² · Xiaozhe Hu³ ·
Long Chen⁴ · Ricardo H. Nochetto⁵ · Jinchao Xu⁶

Received: 30 September 2015 / Revised: 21 February 2016 / Accepted: 24 February 2016 /
Published online: 5 March 2016
© Springer Science+Business Media New York 2016

Abstract In this paper, we study fast iterative solvers for the solution of fourth order parabolic equations discretized by mixed finite element methods. We propose to use consistent mass matrix in the discretization and use lumped mass matrix to construct efficient preconditioners. We provide eigenvalue analysis for the preconditioned system and estimate the convergence rate of the preconditioned GMRes method. Furthermore, we show that these preconditioners only need to be solved inexactly by optimal multigrid algorithms. Our numerical examples indicate that the proposed preconditioners are very efficient and robust with respect to both discretization parameters and diffusion coefficients. We also investigate the performance of

✉ Bin Zheng
binzhengmath@gmail.com; bin.zheng@pnnl.gov

Luoping Chen
clpchenluoping@163.com

Xiaozhe Hu
Xiaozhe.Hu@tufts.edu

Long Chen
chenlong@math.uci.edu

Ricardo H. Nochetto
rhn@math.umd.edu

Jinchao Xu
xu@math.psu.edu

¹ Fundamental and Computational Sciences, Pacific Northwest National Laboratory, Richland, WA 99352, USA

² School of Mathematics, Southwest Jiaotong University, Chengdu 611756, China

³ Department of Mathematics, Tufts University, Medford, MA 02155, USA

⁴ Department of Mathematics, University of California, Irvine, CA 92697, USA

⁵ Department of Mathematics and Institute for Physical Science and Technology, University of Maryland, College Park, MD 20742, USA

⁶ Department of Mathematics, Pennsylvania State University, University Park, PA 16802, USA

multigrid algorithms with either collective smoothers or distributive smoothers when solving the preconditioner systems.

Keywords Fourth order problem · Multigrid method · GMRes · Mass lumping · Preconditioner

1 Introduction

Fourth order parabolic partial differential equations (PDEs) appear in many applications including the lubrication approximation for viscous thin films [49], interface morphological changes in alloy systems during stress corrosion cracking by surface diffusion [1], image segmentation, noise reduction, inpainting [13,42], and phase separation in binary alloys [14,22,30], etc. There have been extensive studies on the numerical methods for solving these fourth order parabolic PDEs, including the finite difference methods [39,56], spectral methods [44,66], discontinuous Galerkin methods [27,35,65], C^1 -conforming finite element methods [31,37], nonconforming finite element methods [32,67], and mixed finite element methods [11,15,29,33,34,36]. The resulting large sparse linear systems of equations are typically very ill-conditioned which poses great challenges for designing efficient and robust iterative solvers. In addition, the linear systems become indefinite when mixed methods are employed. In this work, we design fast iterative solvers for the fourth order parabolic equations discretized by mixed finite element methods.

Multigrid (MG) methods are among the most efficient and robust solvers for solving the linear systems arising from the discretizations of PDEs. There have been a number of studies on multigrid methods for fourth order parabolic equations [7,45,47,48,63]. It is known that smoothers play a significant role in multigrid algorithms. In particular, for saddle point systems, there are two different types of smoothers, i.e., decoupled or coupled smoothers. For a decoupled point smoother, each sweep consists of relaxation over variables per grid point as well as relaxation over grid points. On the other hand, for a coupled point smoother, variables on each grid point are relaxed simultaneously which corresponds to solving a sequence of local problems. Distributive Gauss–Seidel (DGS) is the first decoupled smoother proposed by Brandt and Dinar [20] for solving Stokes equations. Later, Wittum [64] introduced transforming smoothers and combined it with incomplete LU factorization to solve saddle-point systems with applications in Stokes and Navier–Stokes equations. Gaspar et al. [40] studied the DGS smoother for solving poroelasticity problems. Recently, Wang and Chen [62] proposed a distributive Gauss–Seidel smoother based on the least squares commutator for solving Stokes equations and showed multigrid uniform convergence numerically. For coupled smoother, in [61], Vanka proposed a symmetrically coupled Gauss–Seidel smoother for the Navier–Stokes equations discretized by finite difference schemes on staggered grids. Later, in [53], Olshanskii and Reusken introduced an MG solver with collective smoothers for the Navier–Stokes equations in rotation form discretized by conforming finite elements. They proved that W-cycle MG method with block Richardson smoother is robust for the discrete problem. Their numerical experiments show that the W-cycle MG method with the damped Jacobi smoother is robust with respect to the mesh size and problem parameters. Collective Jacobi and Gauss–Seidel smoothers have also been studied in multigrid method for elliptic optimal control problems by Lass et al. [50], and Takacs and Zulehner [57]. In our work, we investigate the performance of both coupled and decouple smoothers for the discrete fourth order problem in the mixed form.

In the literature, there are also many studies on the Krylov subspace methods and their preconditioners for solving fourth order parabolic problems. In [8,9], Bansch, Morin and Nochetto solved a discrete fourth order parabolic equation by applying the conjugate gradient method on the two-by-two block system in Schur complement form. In [10], the same authors proposed symmetric and nonsymmetric preconditioners based on operator splitting. Axelsson et al. [2], Axelsson and Neytcheva [3], Boyanova et al. [18] and Boyanova and Neytcheva [19] proposed a block preconditioner by adding an extra small term to the (2, 2) block and then followed by a block LU factorization which results in a preconditioned system when eigenvalues have a uniform bound. Bosch et al. [16] studied block triangular preconditioner for the mixed finite element discretization of a modified Cahn–Hilliard equation for binary image inpainting. In their method, the Schur complement is solved by algebraic multigrid method (AMG) using V-cycle and Chebyshev smoother. Moreover, the convergence rate of their method is independent of the mesh size. Same method has been used by Bosch et al. [17] to solve Cahn–Hilliard variational inequalities. In [41], Gräser and Kornhuber introduced a preconditioned Uzawa-type iterative method with multigrid solvers for subproblems to solve Cahn–Hilliard equation with an obstacle potential. Based on the Hermitian and skew-Hermitian splitting (HSS) iteration introduced in [6], Benzi and Golub [12] proposed a robust preconditioner for the generalized saddle point problems which exhibit similar two-by-two block structure as the discrete fourth order parabolic problems in mixed form. Later, Bai et al. [4] studied preconditioned modified HSS (PMHSS) iteration methods for the block linear systems. In [5], Bai, Chen, and Wang further simplified PMHSS and proposed an additive block diagonal preconditioner for a block linear system of equations arising from finite element discretization of two-phase flow problems and elliptic PDE-constrained optimization problems.

In this work, we construct preconditioners for discrete fourth order parabolic problems based on the mass lumping technique. Mass lumping technique has been widely used in solving time dependent PDEs by the finite element method [46]. It consists of replacing a consistent mass matrix by a diagonal lumped mass matrix so that its inversion at each time step becomes a simple division. The error estimates for lumped mass approximation have been studied in [23,59,60] which show that the order of accuracy for the discretization is preserved. On the other hand, the loss of solution accuracy associated with mass lumping has been studied by Gresho et al. [43] for advection equations and by Niclasen and Blackburn [52] for incompressible Navier–Stokes equations. It is well known that mass lumping may also induce dispersion errors when solving wave equations, see e.g., [28,51]. These studies suggest that it is sometimes advantageous to use consistent mass matrix in the discretization schemes. In the study of fourth order parabolic equations, both consistent mass matrix and lumped mass matrix have been widely used. In this work, we choose consistent mass matrix in the finite element discretization to keep the solution accuracy and utilize lumped mass matrix to design efficient preconditioners so that the cost of inverting consistent mass matrix can be alleviated. We prove that GMRes method with mass lumping preconditioner converges when $\tau \geq Ch^2$ for some constant C depending on the diffusion coefficients (τ , h corresponds to time and spatial discretization parameters, respectively). In a special case when the two diffusion operators A and B only differ by a scaling factor, we are able to prove uniform convergence of GMRes method for the preconditioned system without the constraint $\tau \geq Ch^2$. Furthermore, we show that the preconditioner systems can be solved inexactly by geometric multigrid methods with the two different types of smoother discussed previously. By combining the optimality of multigrid methods with the computational efficiency of the mass lumping technique, we obtain very efficient solvers for the discrete fourth order problems.

The remainder of the paper is organized as follows. In Sect. 2, we describe the model problem and the corresponding mixed finite element discretization. Next, in Sect. 3, we describe the multigrid method and the collective Jacobi/Gauss–Seidel smoothers for our model problem. In Sect. 4, we construct two mass lumping preconditioners. Multigrid method with decoupled smoother are also introduced in this section to solve the preconditioner system approximately. The spectrum bounds of the preconditioned systems and the convergence property of GMRes method are analyzed in Sect. 5. Finally, in Sect. 6, we present numerical experiments to demonstrate the efficiency and robustness of the proposed solvers.

2 Model Problem and Discretization

2.1 Model Problem

We are interested in solving the following fourth order problem:

$$u - \tau \operatorname{div}(a \nabla v) = f \text{ in } \Omega, \tag{2.1}$$

$$\tau \operatorname{div}(b \nabla u) + v = g \text{ in } \Omega, \tag{2.2}$$

with boundary conditions

$$u = v = 0, \text{ on } \Gamma_D, \tag{2.3}$$

$$v \cdot a \nabla v = v \cdot b \nabla u = 0, \text{ on } \Gamma_N. \tag{2.4}$$

where $\tau = \sqrt{\Delta t}$ and Δt is the time-step size, Ω is a bounded polyhedral domain in $\mathbb{R}^d, d \geq 1$, v is the unit outward normal, Γ_D, Γ_N denote the Dirichlet and Neumann boundary part, respectively. We mainly study the Dirichlet boundary condition in this paper, i.e., $\Gamma_N = \emptyset$. The diffusion coefficients $a(x)$ and $b(x)$ are measurable functions satisfying the following continuity and coercivity conditions

$$\lambda_a(x)|\xi|^2 \leq \xi^T a(x)\xi \leq \Lambda_a(x)|\xi|^2, \quad \lambda_b(x)|\xi|^2 \leq \xi^T b(x)\xi \leq \Lambda_b(x)|\xi|^2, \quad \forall \xi \in \mathbb{R}^d.$$

Equations (2.1) and (2.2) may arise from a time semi-discretization of the fourth order parabolic problem

$$u_t = \operatorname{div}(a \nabla (-\operatorname{div}(b \nabla u))).$$

In this case, $f = u^{\text{old}}$ is the solution at the previous time step, and $g = 0$.

As an example, consider the following Cahn–Hilliard equations that model the phase separation and coarsening dynamics in a binary alloy:

$$u_t - \operatorname{div}(M(u) \nabla (-\epsilon \Delta u + F'(u))) = 0, \text{ in } \Omega \times (0, T), \tag{2.5}$$

where $\Omega \subset \mathbb{R}^3$ is a bounded domain, u represents the relative concentration of one component in a binary mixture, $M(u)$ is the degenerate mobility, which restricts diffusion of both components to the interfacial region, e.g., $M(u) = u(1 - u)$ [11], $F'(u)$ is the derivative of a double well potential $F(u)$, a typical choice is

$$F(u) = \frac{1}{4} (u^2 - 1)^2.$$

Introducing v , defined by $v = \tau(-\epsilon \Delta u + F'(u))$ (the chemical potential multiplied by τ), after semi-implicit time discretization, we obtain a splitting of (2.5) into a coupled system of second order equations [33,34,36]

$$\begin{aligned}
 u - \tau \operatorname{div}(M(u)\nabla v) &= f, \quad \text{in } \Omega, \\
 \tau(-\epsilon \Delta u + F'(u)) - v &= 0, \quad \text{in } \Omega.
 \end{aligned}$$

Denote $a = M(u)$, $b = \epsilon$, and assume $F'(u) = 0$, we can get (2.1) and (2.2), which corresponds to the linearization of the nonlinear Cahn–Hilliard equations.

The weak formulation of (2.1)–(2.2) is: find $u, v \in \mathbb{V}$ such that

$$\begin{aligned}
 (u, \phi) + \tau(a\nabla v, \nabla \phi) &= (f, \phi), \quad \phi \in \mathbb{V} \\
 -\tau(b\nabla u, \nabla \psi) + (v, \psi) &= (g, \psi), \quad \psi \in \mathbb{V},
 \end{aligned} \tag{2.6}$$

where \mathbb{V} is the subspace of $H^1(\Omega)$ associated with the boundary condition (2.3). The well-posedness of (2.6) follows from the Lax–Milgram lemma [54].

2.2 Finite Element Discretization

Let \mathcal{T} be a shape-regular triangulation of Ω , $\mathbb{V}_{\mathcal{T}}$ be the piecewise linear finite element space over \mathcal{T} satisfying the homogeneous boundary condition (2.3), and $N_h = \dim(\mathbb{V}_{\mathcal{T}})$. The discrete problem for the PDE system (2.1)–(2.2) is: find $u_h, v_h \in \mathbb{V}_{\mathcal{T}}$, such that $\forall \phi, \psi \in \mathbb{V}_{\mathcal{T}}$

$$\begin{aligned}
 (u_h, \phi) + \tau(a\nabla v_h, \nabla \phi) &= (f, \phi), \\
 -\tau(b\nabla u_h, \nabla \psi) + (v_h, \psi) &= (g, \psi).
 \end{aligned}$$

In matrix form, we have

$$\begin{pmatrix} \tau A & M \\ M & -\tau B \end{pmatrix} \begin{pmatrix} \underline{u} \\ \underline{v} \end{pmatrix} = \begin{pmatrix} \underline{f} \\ \underline{g} \end{pmatrix}, \quad \text{or } \mathcal{A} \begin{pmatrix} \underline{u} \\ \underline{v} \end{pmatrix} = \begin{pmatrix} \underline{f} \\ \underline{g} \end{pmatrix}, \tag{2.7}$$

where \mathcal{A} is a matrix of size $2N_h \times 2N_h$, M is the mass matrix, and A, B are the stiffness matrices defined by

$$M_{i,j} = (\phi_i, \phi_j), \quad A_{i,j} = (a\nabla \phi_i, \nabla \phi_j), \quad B_{i,j} = (b\nabla \phi_i, \nabla \phi_j).$$

Eliminating \underline{v} in (2.7), we get the following Schur complement equation

$$(M + \tau^2 A M^{-1} B) \underline{u} = \underline{f} - \tau A M^{-1} \underline{g}. \tag{2.8}$$

In this paper, we develop efficient preconditioners for the two-by-two block linear systems (2.7); however, we will solve the Schur complement for preconditioners. In order to avoid inverting mass matrix, we employ the mass lumping technique to construct preconditioners.

3 Multigrid with Collective Smoother

A multigrid algorithm typically consists of three major components: the smoother (or relaxation scheme), the coarse grid operator, and the grid transfer operators (interpolation/restriction operators). It is well known that the efficiency of a multigrid method crucially depends on the choice of the smoother. In particular, for the block system (2.7), we observe numerically that multigrid method with point-wise Gauss–Seidel smoother does not converge uniformly. We construct a block Jacobi or Gauss–Seidel smoother that collects the degrees of freedom corresponding to variables u and v for each grid point. In other words, each block corresponds to a 2×2 matrix.

To describe these collective smoothers, we first consider the following matrix splitting, i.e.,

$$A = \mathcal{A}_L + \mathcal{A}_D + \mathcal{A}_L^T, \quad \mathcal{A}_L = \begin{pmatrix} \tau L_A & L_M \\ L_M & -\tau L_B \end{pmatrix}, \quad \mathcal{A}_D = \begin{pmatrix} \tau D_A & D_M \\ D_M & -\tau D_B \end{pmatrix},$$

where L_A, L_B, L_M are strictly lower triangular parts of A, B , and M , and D_A, D_B, D_M are their diagonal parts. The collective damped Jacobi relaxation can be represented as

$$\begin{pmatrix} \mathbf{u}^{k+1} \\ \mathbf{v}^{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{u}^k \\ \mathbf{v}^k \end{pmatrix} + \vartheta \mathcal{A}_D^{-1} \left[\begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix} - \mathcal{A} \begin{pmatrix} \mathbf{u}^k \\ \mathbf{v}^k \end{pmatrix} \right] \tag{3.1}$$

The collective Gauss–Seidel relaxation can be represented as

$$\begin{pmatrix} \mathbf{u}^{k+1} \\ \mathbf{v}^{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{u}^k \\ \mathbf{v}^k \end{pmatrix} + (\mathcal{A}_L + \mathcal{A}_D)^{-1} \left[\begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix} - \mathcal{A} \begin{pmatrix} \mathbf{u}^k \\ \mathbf{v}^k \end{pmatrix} \right] \tag{3.2}$$

More practically, one can rearrange variables \mathbf{u} and \mathbf{v} in \mathbf{w} so that $\mathcal{A}_L + \mathcal{A}_D$ corresponds to a lower block triangular matrix. In fact, let $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{N_h})^T$ with each entry $\mathbf{w}_i = (v_i, u_i)$ corresponds to a pair of variables. Then, each relaxation sweep of (3.1) consists of solving a sequence of small systems

$$\tilde{w}_i^{k+1} = \tilde{w}_i^k + \vartheta \mathcal{A}_{ii}^{-1} \left(F_i - \sum_{j=1}^{N_h} \mathcal{A}_{ij} \tilde{w}_j^k \right),$$

and each sweep of (3.2) consists of solving

$$\tilde{w}_i^{k+1} = \tilde{w}_i^k + \mathcal{A}_{ii}^{-1} \left(F_i - \sum_{j=1}^{i-1} \mathcal{A}_{ij} \tilde{w}_j^{k+1} - \sum_{j=i}^{N_h} \mathcal{A}_{ij} \tilde{w}_j^k \right),$$

where

$$\mathcal{A}_{ij} = \begin{pmatrix} \tau A_{ij} & M_{ij} \\ M_{ij} & -\tau B_{ij} \end{pmatrix}, \quad F_i = \begin{pmatrix} f_i \\ g_i \end{pmatrix}, \quad i, j = 1, 2, \dots, N_h.$$

Note that \mathcal{A}_{ii} is invertible since $\det(\mathcal{A}_{ii}) = -\tau^2 A_{ii} B_{ii} - M_{ii}^2 \neq 0$.

It is clear from the above form that for collective damped Jacobi relaxation, these small system can be solved in parallel, and for collective Gauss–Seidel relaxation, they are solved successively. Numerical experiments in Sect. 6 indicate that by relaxing both variables u_i, v_i corresponding to the same grid point i collectively, geometric multigrid V-cycle converges uniformly with respect to both h and τ .

The collective Gauss–Seidel smoother has been studied by Lass et al. [50] for solving elliptic optimal control problems. It is shown in [50] that the convergence rate of multigrid method with collective Gauss–Seidel smoother is independent of h . The robustness with respect to τ is, however, hard to prove theoretically. Takacs and Zulehner [57] also investigated multigrid algorithm with several collective smoothers for optimal control problems and proved the convergence of the W-cycle multigrid with collective Richardson smoother. For convergence proof of multigrid methods using collective smoothers applied to saddle point systems, we refer to the work by Schöberl [55] and by Chen [25,26].

4 Mass Lumping Preconditioners

In this section, we study preconditioners for the two-by-two block linear systems (2.7) for use with GMRes method. When the meshsize and time-step size are small, these systems are generally very ill-conditioned, especially when the diffusion coefficients a and b degenerate. Hence, efficient preconditioners are necessary in order to speed up the convergence of GMRes method. In [10], Bänsch, Morin, and Nchetto proposed symmetric and non-symmetric preconditioners that work well for the Schur complement system (2.8). However, the convergence rates of these methods are not uniform with respect to h or τ . Besides, the performance of those methods deteriorate for degenerate problems [10].

In the following, we design two preconditioners based on the mass lumping technique and geometric multigrid method for the system (2.7). The main focus is on the efficiency and robustness of the proposed solvers. Numerical experiments in Sect. 6 indicate that GMRes method preconditioned by the mass lumping preconditioners (solved inexactly) converges uniformly with respect to the discretization parameters and is also robust for problems with degenerate diffusion coefficients.

4.1 Preconditioner with Two Lumped Mass Matrices

The mass lumping technique has been widely used in the finite element computations, especially for time-dependent problems as it avoids inverting a full mass matrix M at each time step. The lumped-mass matrix \bar{M} is a diagonal matrix with diagonal elements equal to the row sums of M . By using the diagonal matrix \bar{M} , the computational cost for the preconditioner is reduced significantly.

The mass lumping preconditioner \mathcal{B} for the block system \mathcal{A} is defined by

$$\mathcal{B} = \begin{pmatrix} \tau A & \bar{M} \\ \bar{M} & -\tau B \end{pmatrix}. \tag{4.1}$$

GMRes method is then applied to solve the preconditioned linear system

$$\begin{pmatrix} \tau A & M \\ M & -\tau B \end{pmatrix} \begin{pmatrix} \tau A & \bar{M} \\ \bar{M} & -\tau B \end{pmatrix}^{-1} \begin{pmatrix} p \\ q \end{pmatrix} = \begin{pmatrix} \hat{f} \\ \hat{g} \end{pmatrix}. \tag{4.2}$$

4.2 Preconditioner with One Lumped Mass Matrix

We can also use the following preconditioner $\tilde{\mathcal{B}}$ with one lumped mass matrix for solving the block system (2.7), i.e.

$$\tilde{\mathcal{B}} = \begin{pmatrix} \tau A & M \\ \bar{M} & -\tau B \end{pmatrix}. \tag{4.3}$$

The matrix $\tilde{\mathcal{B}}$ is nonsymmetric and is a better approximation to \mathcal{A} compared with \mathcal{B} . For $\tilde{\mathcal{B}}$, we still have a block factorization which avoids inverting mass matrix M . Numerical experiments of Sect. 6.2 indicate that the performance of the two preconditioners $\tilde{\mathcal{B}}$ and \mathcal{B} used with GMRes method are similar when solved inexactly by geometric multigrid V-cycle when $\tau \geq Ch^2$. However, when τ is very small, $\tilde{\mathcal{B}}$ performs better than \mathcal{B} .

4.3 Multigrid for Preconditioners

The preconditioner systems only need to be solved approximately. We can use geometric multigrid method with the coupled smoothers described in Sect. 3. In the following, we

construct a decoupled smoother following the idea of the distributive Gauss–Seidel relaxation (DGS) which is suitable for the model problem.

DGS is a decoupled smoother introduced by Brandt and Dinar [20] for solving Stokes equations. The main idea of DGS is to apply standard Gauss–Seidel relaxation on decoupled equations using transformed variables. Let us consider the mass-lumping preconditioner system \tilde{B} (a similar scheme can be derived for B),

$$\tilde{B} \begin{pmatrix} v \\ u \end{pmatrix} = \begin{pmatrix} p \\ q \end{pmatrix}. \tag{4.4}$$

We introduce the following change of variables

$$\begin{pmatrix} v \\ u \end{pmatrix} = \mathcal{P} \begin{pmatrix} x \\ y \end{pmatrix}, \quad \text{where } \mathcal{P} = \begin{pmatrix} \tau \bar{M}^{-1} B & 0 \\ I & I \end{pmatrix} \tag{4.5}$$

is called the distribution matrix. Right preconditioning \tilde{B} by \mathcal{P} results in an upper block triangular matrix

$$\tilde{B}\mathcal{P} = \begin{pmatrix} M + \tau^2 A \bar{M}^{-1} B & M \\ 0 & -\tau B \end{pmatrix}.$$

We construct a decoupled smoother for preconditioner (4.4) by solving

$$\begin{pmatrix} v^{k+1} \\ u^{k+1} \end{pmatrix} = \begin{pmatrix} v^k \\ u^k \end{pmatrix} + \mathcal{P}(\tilde{B}\mathcal{P})^{-1} \left[\begin{pmatrix} p \\ q \end{pmatrix} - \tilde{B} \begin{pmatrix} v^k \\ u^k \end{pmatrix} \right] \tag{4.6}$$

More precisely, we have the following algorithm

1. Form the residual:

$$\begin{pmatrix} r_v \\ r_u \end{pmatrix} = \left[\begin{pmatrix} p \\ q \end{pmatrix} - \tilde{B} \begin{pmatrix} v^k \\ u^k \end{pmatrix} \right]$$
2. Apply standard Gauss–Seidel relaxation to solve the error equation

$$-\tau B e_y = r_u$$
 and use the damped Jacobi relaxation for

$$(M + \tau^2 A \bar{M}^{-1} B) e_x = r_v - M e_y. \tag{4.7}$$
 Then, recover e_u, e_v from (4.5), i.e.,

$$e_v = \tau \bar{M}^{-1} B e_x;$$

$$e_u = e_x + e_y$$
3. Update the solution

$$v^{k+1} = v^k + e_v$$

$$u^{k+1} = u^k + e_u$$

We can use multigrid method with the above decoupled smoother to solve the preconditioner system (4.4).

Remark 1 By using the damped Jacobi to solve (4.7), we can avoid matrix multiplication. In fact, we only need to calculate the diagonal part of $M + \tau^2 A \bar{M}^{-1} B$. This requires $\mathcal{O}(N_h)$ operations because \bar{M} is a diagonal matrix.

5 Convergence Analysis of Preconditioned GMRes Method

In this section, we analyze the convergence of the GMRes method preconditioned by the two preconditioners introduced in Sect. 4 for the block system (2.7). We show that the preconditioned GMRes method converges when $\tau \geq Ch^2$ for some constant C depending on the diffusion coefficients. Numerical results in Sect. 6 indicate that the preconditioned GMRes method converges uniformly for any h and τ .

Let $R = \mathcal{A}B^{-1}$ (or $\mathcal{A}\bar{B}^{-1}$) with B (or \bar{B}) being a right preconditioner of \mathcal{A} . To solve the preconditioned system $Rv = b$, the GMRes method starts from an initial iterate v_0 and produces a sequence of iterates v_k and residuals $r_k := b - Rv_k$, such that $r_k = p_k(R)r_0$ for some polynomial $p_k \in \mathcal{P}_k$, and

$$\|r_k\|_2 = \min_{\substack{p \in \mathcal{P}_k \\ p(0)=1}} \|p(R)r_0\|_2, \tag{5.1}$$

where \mathcal{P}_k is the space of polynomials of degree k or less and $\|\cdot\|_2$ is the Euclidean norm. As a consequence, the convergence rate of the GMRes method can be estimated by

$$\frac{\|r_k\|_2}{\|r_0\|_2} \leq \min_{\substack{p \in \mathcal{P}_k \\ p(0)=1}} \|p(R)\|_2. \tag{5.2}$$

Since R may not be a normal matrix, we follow the approach given in [10] using the concept of ϵ -pseudospectrum [58]. Given $\epsilon > 0$, denote by $\sigma_\epsilon(R)$ the set of ϵ -eigenvalues of R , namely those $z \in \mathbb{C}$ that are eigenvalues of some matrix $R + E$ with $\|E\| \leq \epsilon$. We first quote the following two results.

Lemma 1 (Pseudospectrum estimate [58]) *Let Σ_ϵ be a union of closed curves enclosing the ϵ -pseudospectrum $\sigma_\epsilon(R)$ of R . Then for any polynomial p_k of degree k with $p_k(0) = 1$ we have*

$$\max_{z \in \sigma(R)} |p_k(z)| \leq \|p_k(R)\| \leq \frac{L_\epsilon}{2\pi\epsilon} \max_{z \in \Sigma_\epsilon(R)} |p_k(z)|, \tag{5.3}$$

where L_ϵ is the arclength of Σ_ϵ .

Lemma 2 (Bound on the ϵ -pseudospectrum [10]) *If R is a square matrix of order n and $0 < \epsilon \leq 1$, then,*

$$\sigma_\epsilon(R) \subset \cup_{\lambda \in \sigma(R)} B\left(\lambda, C_R \epsilon^{\frac{1}{m}}\right),$$

where $C_R := n(1 + \sqrt{n-p})\kappa(V)$, with $\kappa(V)$ the condition number of V and V is a nonsingular matrix transforming R into its Jordan canonical form J , i.e. $V^{-1}RV = J$, p is the number of Jordan blocks, and m is the size of the largest Jordan block of R .

In order to estimate the error caused by mass lumping, we introduce the following operators [23]. Let $z_j (1 \leq j \leq 3)$ be the vertices of a triangle $K \in \mathcal{T}$, consider the following quadrature formula

$$Q_{K,h}(f) = \sum_{j=1}^3 (1, \phi_j|_K) f(z_j) = \frac{1}{3} |K| \sum_{j=1}^3 f(z_j)$$

$$(\phi_i, \phi_j)_h = \sum_{K \in \mathcal{T}} Q_{K,h}(\phi_i \phi_j) = \int_{\Omega} \pi_h(\phi_i \phi_j) dx$$

where $\pi_h : C^0(\bar{\Omega}) \rightarrow \mathbb{V}_{\mathcal{T}}$ is the standard nodal interpolation operator. Then, the mass lumping procedure can be interpreted as

$$(\phi_j, \phi_j)_h = Q_{K,h}(\phi_j \phi_j) = \sum_{k=1}^{N_h} (\phi_j, \phi_k).$$

The following results are useful for later analysis.

Lemma 3 (Quadrature error [23]) *Let $u, v \in \mathbb{V}_{\mathcal{T}} \subset H_0^1$ and $(u, v)_h$ be the lumped inner product. Then,*

$$|(u, v) - (u, v)_h| \leq C_l h^2 |u|_1 |v|_1, \tag{5.4}$$

where C_l is a constant independent of h .

Lemma 4 (Norm equivalence [33]) *Let M be the mass matrix and \bar{M} be its lumped version. Then, for any $u = \sum_j u_j \phi_j \in \mathbb{V}_{\mathcal{T}}$ we have*

$$C_1 (\bar{M}\underline{u}, \underline{u}) \leq (M\underline{u}, \underline{u}) \leq (\bar{M}\underline{u}, \underline{u}), \tag{5.5}$$

where $\underline{u} = (u_1, u_2, \dots, u_{N_h})^T$ and $C_1 \in (0, 1)$ is a constant independent of h .

Proof We first show that $(M\underline{u}, \underline{u}) \leq (\bar{M}\underline{u}, \underline{u})$. Let $\delta M = M - \bar{M}$, then

$$(\delta M)_{ij} < 0, \quad i = j; \quad (\delta M)_{ij} \geq 0, \quad i \neq j; \quad \text{and} \quad (\delta M)_{ii} = - \sum_{j \neq i} (\delta M)_{ij}.$$

By direct calculations, we have

$$\begin{aligned} \underline{u}^T \delta M \underline{u} &= \sum_{i=1}^{N_h} \sum_{j=1}^{N_h} (\delta M)_{ij} u_i u_j \\ &= \sum_{i=1}^{N_h} \left(u_i \sum_{j < i} (\delta M)_{ij} (u_j - u_i) + u_i \sum_{j > i} (\delta M)_{ij} (u_j - u_i) \right) \\ &= \sum_{i=1}^{N_h} \sum_{j > i} -(\delta M)_{ij} (u_j - u_i)^2 \leq 0. \end{aligned}$$

By Lemma 3 and the inverse inequality, we get

$$(u, u)_h - (u, u) \leq |(u, u) - (u, u)_h| \leq C_l h^2 |u|_1^2 \leq C \|u\|_0^2 = C(u, u).$$

Hence, $(u, u)_h \leq (1 + C)(u, u)$. Equivalently,

$$\frac{1}{1 + C} (\bar{M}\underline{u}, \underline{u}) \leq (M\underline{u}, \underline{u})$$

which implies that the left inequality holds with $C_1 = 1/(1 + C)$ for some positive constant C . □

We also need the following estimates for the eigenvalues of the stiffness matrices and mass matrix [21,38].

Lemma 5 (Eigenvalue estimates) *Let A, B be the stiffness matrices corresponding to diffusion coefficients $a(x)$ and $b(x)$, respectively, and let M be the mass matrix, then,*

$$\lambda_{\max}(A) \leq C_A, \quad \lambda_{\max}(B) \leq C_B, \quad C_M h^2 \leq \lambda_{\min}(M) \tag{5.6}$$

where C_A, C_B depend on the continuity assumption of the bilinear forms and the constant in the inverse inequality.

5.1 Eigenvalue Analysis for $\mathcal{A}\mathcal{B}^{-1}$ with Constraint $\tau \geq Ch^2$

Recall

$$\mathcal{B} = \begin{pmatrix} \tau A & \bar{M} \\ \bar{M} & -\tau B \end{pmatrix}, \quad \mathcal{A} = \begin{pmatrix} \tau A & M \\ M & -\tau B \end{pmatrix}.$$

We have the following spectrum bound for the preconditioned system $\mathcal{B}^{-1}\mathcal{A}$.

Theorem 1 (Spectral bound for $\mathcal{B}^{-1}\mathcal{A}$) *Let h denote the meshsize and τ denote the square root of time-step size. Then, the spectral radius of $\mathcal{B}^{-1}\mathcal{A}$ satisfies*

$$\rho(\mathcal{B}^{-1}\mathcal{A}) < 2,$$

if $\tau \geq Ch^2$ for some positive constant C independent of h and τ .

Proof Let $\lambda \in \mathbb{C}$, $(\mathbf{v}, \mathbf{u})^T \in \mathbb{C}^{2N_h}$ be a pair of eigenvalue and eigenvector of $\mathcal{B}^{-1}\mathcal{A}$, i.e.

$$\begin{pmatrix} \tau A & \bar{M} \\ \bar{M} & -\tau B \end{pmatrix}^{-1} \begin{pmatrix} \tau A & M \\ M & -\tau B \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ \mathbf{u} \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{v} \\ \mathbf{u} \end{pmatrix}.$$

Equivalently,

$$\begin{pmatrix} \tau A & M \\ M & -\tau B \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ \mathbf{u} \end{pmatrix} = \lambda \begin{pmatrix} \tau A & \bar{M} \\ \bar{M} & -\tau B \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ \mathbf{u} \end{pmatrix}. \tag{5.7}$$

Taking the inner product of the Eq. (5.7) with $(\mathbf{v}^T, -\mathbf{u}^T)$, we obtain

$$[\tau(A\mathbf{v}, \mathbf{v}) + \tau(B\mathbf{u}, \mathbf{u}) + 2i\text{Im}(M\mathbf{v}, \mathbf{u})] = \lambda[\tau(A\mathbf{v}, \mathbf{v}) + \tau(B\mathbf{u}, \mathbf{u}) + 2i\text{Im}(\bar{M}\mathbf{v}, \mathbf{u})]. \tag{5.8}$$

Then, we have

$$2i\text{Im}(\delta M\mathbf{v}, \mathbf{u}) = (\lambda - 1) [\tau(A\mathbf{v}, \mathbf{v}) + \tau(B\mathbf{u}, \mathbf{u}) + 2i\text{Im}(\bar{M}\mathbf{v}, \mathbf{u})]. \tag{5.9}$$

Since A and B are symmetric positive definite matrices, $\|\mathbf{v}\|_A^2 = (A\mathbf{v}, \mathbf{v})$ and $\|\mathbf{u}\|_B^2 = (B\mathbf{u}, \mathbf{u})$ are nonnegative real numbers. By taking the modulus of (5.9) we get

$$4|\text{Im}(\delta M\mathbf{v}, \mathbf{u})|^2 = |\lambda - 1|^2 (\alpha^2 + 4|\text{Im}(\bar{M}\mathbf{v}, \mathbf{u})|^2),$$

where $\alpha = \tau(\|\mathbf{v}\|_A^2 + \|\mathbf{u}\|_B^2)$. Hence,

$$|\lambda - 1|^2 = \frac{4|\text{Im}(\delta M\mathbf{v}, \mathbf{u})|^2}{\alpha^2 + 4|\text{Im}(\bar{M}\mathbf{v}, \mathbf{u})|^2}. \tag{5.10}$$

Let $v = \sum_i v_i \phi_i, u = \sum_i u_i \phi_i$, we get

$$\begin{aligned} |(\delta M \underline{v}, \underline{u})| &\leq C_l h^2 |v|_1 |u|_1 \quad (\text{by (5.4)}) \\ &\leq C_l C_a C_b h^2 \|\underline{v}\|_A \|\underline{u}\|_B \quad (C_a = \min_x \lambda_a(x), C_b = \min_x \lambda_b(x)) \\ &\leq C_l C_a C_b h^2 \frac{\|\underline{v}\|_A^2 + \|\underline{u}\|_B^2}{2} \quad (\text{by Young's inequality}) \\ &= \frac{C_l C_a C_b h^2}{2\tau} \tau (\|\underline{v}\|_A^2 + \|\underline{u}\|_B^2), \end{aligned}$$

Hence,

$$4\text{Im}|(\delta M \underline{v}, \underline{u})|^2 \leq \frac{(C_l C_a C_b)^2 h^4}{\tau^2} \tau^2 (\|\underline{v}\|_A^2 + \|\underline{u}\|_B^2)^2 = \frac{(C_l C_a C_b)^2 h^4}{\tau^2} \alpha^2.$$

Let $C = 2C_l C_a C_b$, when $\tau \geq Ch^2$, we get $\rho(B^{-1}A) < 2$. □

Using the same proof of Corollary 5.11 in [10] with $\epsilon_0 = 1 - \rho(B^{-1}A)/2$, we have

Corollary 1 (Convergence rate of GMRes for AB^{-1}) *For the preconditioned system AB^{-1} , GMRes method converges with an asymptotic linear convergence rate bounded by*

$$\theta = \frac{\rho(B^{-1}A)}{2},$$

if $\tau \geq Ch^2$. Moreover,

$$\frac{\|r_k\|_2}{\|r_0\|_2} \leq C_0 \theta^k,$$

with $C_0 = 2^{m-1} C_R^m (1 + \rho(E)) \dim \mathbb{V}_{\mathcal{T}} / (1 - \rho(E))^m$, C_R, m are the parameters defined in the Lemma 2 and $\mathbb{V}_{\mathcal{T}}$ is the finite element space.

Similarly, we have the following result.

Corollary 2 (Convergence rate of GMRes for $A\tilde{B}^{-1}$) *The spectral radius of $\tilde{B}^{-1}A$ satisfies*

$$\rho(\tilde{B}^{-1}A) < 2,$$

if $\tau \geq Ch^2$ for some constant C independent with h and τ . For the preconditioned system $A\tilde{B}^{-1}$, GMRes method converges with an approximate linear convergence rate bounded by

$$\tilde{\theta} = \frac{\rho(\tilde{B}^{-1}A)}{2}.$$

5.2 Analysis Without the Constraint $\tau \geq Ch^2$

As we will see in Sect. 6, numerical results show uniform convergence rate of GMRes method for the preconditioned system irrespective of the relation between h and τ . A proof of this result appears elusive due to the fact that the Schur complement is nonsymmetric. In this subsection, we give a proof in this direction for the special case $B = \alpha A$ with $\alpha > 0$ being some scaling constant in the mass lumping preconditioner \tilde{B} . For simplicity, we choose $\alpha = 1$ for the remaining part of this paper.

Notice that

$$\begin{pmatrix} \tau A & M \\ \bar{M} & -\tau B \end{pmatrix}^{-1} = \begin{pmatrix} I & -(\tau A)^{-1}M \\ 0 & I \end{pmatrix} \begin{pmatrix} (\tau A)^{-1} & 0 \\ 0 & (-\tau B - \bar{M}(\tau A)^{-1}M)^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ -\bar{M}(\tau A)^{-1} & I \end{pmatrix}$$

$$\tilde{B}^{-1}A = \begin{pmatrix} \tau A & M \\ \bar{M} & -\tau B \end{pmatrix}^{-1} \begin{pmatrix} \tau A & M \\ M & -\tau B \end{pmatrix} = \begin{pmatrix} I + X & 0 \\ (-\tau B - \bar{M}(\tau A)^{-1}M)^{-1}(M - \bar{M}) & I \end{pmatrix},$$

where

$$X = (\tau A)^{-1}M (\tau B + \bar{M}(\tau A)^{-1}M)^{-1} (M - \bar{M}).$$

Then, any eigenvalue of $\tilde{B}^{-1}A$ is either 1 or $1 + \lambda$ for some $\lambda \in \sigma(X)$ where $\sigma(X)$ represents the spectrum of X .

To derive the bounds of $\sigma(X)$, we first recall the Sherman-Morrison-Woodbury formula

$$(A + UV^T)^{-1} = A^{-1} - A^{-1}U (I + V^T A^{-1}U)^{-1} V^T A^{-1},$$

and the following identity

$$V^T (A + UV^T)^{-1} U = (I + (V^T A^{-1}U)^{-1})^{-1}. \tag{5.11}$$

Applying (5.11), we get

$$\begin{aligned} X &= (\tau A)^{-1}M (\tau B + \bar{M}(\tau A)^{-1}M)^{-1} \bar{M} [\bar{M}^{-1}(M - \bar{M})] \\ &= [I + ((\tau A)^{-1}M(\tau B)^{-1}\bar{M})^{-1}]^{-1} (\bar{M}^{-1}M - I) \\ &= (I + \tau^2(A^{-1}MB^{-1}\bar{M})^{-1})^{-1} (\bar{M}^{-1}M - I) \\ &= (I + \tau^2\bar{M}^{-1}BM^{-1}A)^{-1} (\bar{M}^{-1}M - I). \end{aligned}$$

Lemma 6 (Spectrum of X) *If $A = B$, and $C_1 < 1$ is the constant in (5.5), then*

$$\sigma(X) \subset (C_1 - 1, 0].$$

Proof Let (λ, w) be an eigenpair of the matrix X , then

$$\begin{aligned} Xw = \lambda w &\Leftrightarrow (\bar{M}^{-1}M - I)w = \lambda (I + \tau^2\bar{M}^{-1}AM^{-1}A)w \\ &\Leftrightarrow (M - \bar{M})w = \lambda (\bar{M} + \tau^2AM^{-1}A)w \\ &\Leftrightarrow \lambda = \frac{((M - \bar{M})w, w)}{((\bar{M} + \tau^2AM^{-1}A)w, w)} \\ &\Leftrightarrow 0 \geq \lambda > (C_1 - 1) \frac{(\bar{M}w, w)}{(\bar{M}w, w) + (\tau^2AM^{-1}Aw, w)} \\ &\Leftrightarrow C_1 - 1 < \lambda \leq 0. \end{aligned}$$

□

By Lemma 6 and the relation between $\tilde{B}^{-1}A$ and X , we obtain the following result.

Theorem 2 (Spectrum of $\tilde{B}^{-1}A$) *Let A and \tilde{B} be given by (2.7) and (4.3), respectively. If $A = B$, we have*

$$\sigma(\tilde{B}^{-1}A) \subset (C_1, 1].$$

The following result can be proved by using the same proof for the Corollary 5.11 in [10] with $\epsilon_0 = C_1/2$.

Corollary 3 (Convergence rate of GMRes for $A\tilde{B}^{-1}$) *GMRes's iteration converges for system $A\tilde{B}^{-1}$ with an asymptotic linear convergence rate bounded by*

$$\theta = \frac{1 - C_1/2}{1 + C_1},$$

Moreover,

$$\frac{\|r_k\|}{\|r_0\|} \leq C_0\theta^k,$$

with $C_0 = 2^{2m-1}C_R^m \dim \nabla_{\mathcal{T}} / C_1^{m-1}$.

Remark 2 (Preconditioning for $\tau < Ch^2$) By inverse inequality, we have $\tau \|u\|_A^2 \leq \tau h^{-2} \|u\|_M^2$. So if $\tau < h^2$, then M will dominate A . For the case when $A \neq \alpha B$ and $\tau < Ch^2$, we rewrite the system in the following form

$$\begin{pmatrix} M & \tau A \\ -\tau B & M \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \tag{5.12}$$

and consider the diagonal preconditioner

$$B_d = \begin{pmatrix} M & 0 \\ 0 & M \end{pmatrix}. \tag{5.13}$$

To give an estimate of the spectrum bound for the preconditioned system, we let

$$\begin{pmatrix} M & 0 \\ 0 & M \end{pmatrix}^{-1} \begin{pmatrix} M & \tau A \\ -\tau B & M \end{pmatrix} = I + E_d$$

where

$$E_d = \begin{pmatrix} M & 0 \\ 0 & M \end{pmatrix}^{-1} \begin{pmatrix} 0 & \tau A \\ -\tau B & 0 \end{pmatrix} \tag{5.14}$$

Let $(\lambda, (\underline{x}^T, \underline{y}^T))$ be an eigenpair of E_d , i.e.,

$$\begin{pmatrix} 0 & \tau A \\ -\tau B & 0 \end{pmatrix} \begin{pmatrix} \underline{x} \\ \underline{y} \end{pmatrix} = \lambda \begin{pmatrix} M & 0 \\ 0 & M \end{pmatrix} \begin{pmatrix} \underline{x} \\ \underline{y} \end{pmatrix}$$

Taking the inner product of the above equation with the vector $(\underline{x}^T, \underline{y}^T)$ we obtain

$$(\tau A \underline{y}, \underline{x}) - (\tau B \underline{x}, \underline{y}) = \lambda \left[(M \underline{x}, \underline{x}) + (M \underline{y}, \underline{y}) \right].$$

Denote $\| \cdot \|_M^2 = (M \cdot, \cdot)$, we have

$$|\lambda| = \frac{\tau \left| (A \underline{y}, \underline{x}) - (B \underline{x}, \underline{y}) \right|}{\|\underline{x}\|_M^2 + \|\underline{y}\|_M^2}$$

Since M is symmetric positive definite,

$$\begin{aligned} |(A\mathbf{y}, \mathbf{x})| &\leq \left| \left(M^{-\frac{1}{2}} A M^{-\frac{1}{2}} M^{\frac{1}{2}} \mathbf{y}, M^{\frac{1}{2}} \mathbf{x} \right) \right| \\ &\leq \left\| M^{-\frac{1}{2}} A M^{-\frac{1}{2}} \right\| \|\mathbf{y}\|_M \|\mathbf{x}\|_M \\ &\leq \left\| M^{-\frac{1}{2}} A M^{-\frac{1}{2}} \right\| \frac{\|\mathbf{x}\|_M^2 + \|\mathbf{y}\|_M^2}{2}. \end{aligned}$$

Similarly,

$$|(B\mathbf{x}, \mathbf{y})| \leq \left\| M^{-\frac{1}{2}} B M^{-\frac{1}{2}} \right\| \frac{\|\mathbf{x}\|_M^2 + \|\mathbf{y}\|_M^2}{2}$$

Then, by Lemma 5,

$$\begin{aligned} |\lambda| &\leq \frac{\tau}{2} \left(\left\| M^{-\frac{1}{2}} A M^{-\frac{1}{2}} \right\| + \left\| M^{-\frac{1}{2}} B M^{-\frac{1}{2}} \right\| \right) \\ &\leq \frac{\tau}{2} \left(\frac{\lambda_{\max}(A)}{\lambda_{\min}(M)} + \frac{\lambda_{\max}(B)}{\lambda_{\min}(M)} \right) \\ &\leq \frac{\tau}{2} \left(\frac{C_A}{C_M h^2} + \frac{C_B}{C_M h^2} \right) = \frac{\tau(C_A + C_B)}{2C_M h^2} < 1, \end{aligned}$$

if $\tau < Ch^2$, where $C = 2C_M/(C_A + C_B)$. Hence, all the eigenvalues of the preconditioned system can be bounded in $B(1, \rho(E_d))$ with $\rho(E_d) < 1$.

6 Numerical Experiments

In this section, we provide numerical experiments to demonstrate the performance of multi-grid method and the preconditioned flexible GMRes method (without restart) using mass lumping preconditioners. The MATLAB adaptive finite element package *iFEM* [24] is used for all experiments. We choose the following two test problems from [10].

Consider the L-shaped domain $\Omega = (-1, 1)^2 \setminus [0, 1)^2$, set $g = 0$, $f = 1$, and choose the Dirichlet boundary condition in both examples. For the diffusion coefficients $a(x_1, x_2)$, $b(x_1, x_2)$, we choose

1. Nice problem:

$$a(x_1, x_2) = 1, \quad b(x_1, x_2) = \begin{cases} 0.6, & \text{if } x_2 < x_1, \\ 1.2, & \text{otherwise.} \end{cases}$$

2. Degenerate problem:

$$a(x_1, x_2) = 0.1|x_1| + |x_2|, \quad b(x_1, x_2) = 10 + 3 \sin(5\pi x_1) \sin(8\pi x_2).$$

The range of parameter $\tau \in [10^{-4}, 1]$, which corresponds to the time-step size $\Delta t \in [10^{-8}, 1]$. The stopping criterion of the multigrid solver and GMRes iteration is chosen to be the relative residual error in L^2 norm less than 10^{-7} . The initial guess for the iterative methods are chosen randomly. For time-dependent problems, we can use the approximation in the previous time step as the initial guess and our solvers can converge in just a few steps. All computations are done using MATLAB R2015b on a desktop computer with a 3.7 GHz Intel processor and 32 GB RAM.

Table 1 Example 1: iteration counts and CPU time (in seconds) of CGS/CJ-MG

| Solver | h | τ | | | | | | | |
|--------|-------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | | 1e-0 | 1e-1 | 1e-2 | 1e-3 | 1e-4 | 1e-5 | 1e-6 | 1e-7 |
| CGS-MG | 1/64 | 8 (0.31) | 8 (0.31) | 8 (0.31) | 8 (0.31) | 13 (0.49) | 8 (0.31) | 7 (0.27) | 7 (0.27) |
| | 1/128 | 8 (1.5) | 8 (1.5) | 8 (1.5) | 8 (1.5) | 11 (2.0) | 12 (2.2) | 7 (1.3) | 7 (1.3) |
| | 1/256 | 8 (6.6) | 8 (6.6) | 8 (6.6) | 8 (6.6) | 10 (8.3) | 11 (9.1) | 9 (7.5) | 7 (5.9) |
| CJ-MG | 1/64 | 15 (0.51) | 15 (0.50) | 15 (0.51) | 15 (0.50) | 14 (0.47) | 12 (0.40) | 13 (0.43) | 13 (0.43) |
| | 1/128 | 15 (2.3) | 15 (2.2) | 15 (2.2) | 15 (2.2) | 14 (2.0) | 12 (1.8) | 13 (1.9) | 13 (1.9) |
| | 1/256 | 15 (10) | 15 (10) | 15 (10) | 15 (10) | 15 (10) | 14 (9.6) | 12 (8.2) | 13 (9.0) |

Table 2 Example 2: iteration counts and CPU time (in seconds) of CGS/CJ-MG

| Solver | h | τ | | | | | | | |
|--------|-------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | | 1e-0 | 1e-1 | 1e-2 | 1e-3 | 1e-4 | 1e-5 | 1e-6 | 1e-7 |
| CGS-MG | 1/64 | 8 (0.31) | 8 (0.31) | 8 (0.31) | 8 (0.31) | 10 (0.38) | 11 (0.42) | 7 (0.27) | 7 (0.27) |
| | 1/128 | 8 (1.5) | 8 (1.5) | 8 (1.5) | 8 (1.5) | 8 (1.5) | 12 (2.2) | 9 (1.7) | 7 (1.3) |
| | 1/256 | 8 (6.8) | 8 (6.7) | 8 (6.7) | 8 (6.7) | 8 (6.8) | 10 (8.4) | 12 (10) | 7 (5.9) |
| CJ-MG | 1/64 | 15 (0.53) | 15 (0.52) | 15 (0.51) | 15 (0.54) | 14 (0.48) | 12 (0.41) | 14 (0.48) | 13 (0.44) |
| | 1/128 | 15 (2.2) | 15 (2.2) | 15 (2.2) | 15 (2.2) | 14 (2.1) | 13 (1.9) | 13 (1.9) | 14 (2.1) |
| | 1/256 | 15 (11) | 15 (10) | 15 (10) | 15 (10) | 15 (10) | 14 (10) | 12 (8.3) | 14 (8.6) |

6.1 Results of Multigrid Solver with Collective Smoothers

We compare the performance of multigrid method with collective Gauss–Seidel and collective Jacobi smoother (with damping parameter $\vartheta = 0.8$) for solving (2.7). We choose this particular Jacobi damping parameter because it gives the best performance compared with other values we have tried. We use multigrid V-cycle with one pre-smoothing and one post-smoothing (i.e., $V(1, 1)$) which achieves the best efficiency in terms of CPU time. The number of iterations and CPU time (in parentheses) are summarized in Tables 1 and 2. We observe that multigrid methods converge uniformly with respect to h . The convergence rate is close to uniform with respect to τ except for some deterioration when τ is around 10^{-4} or 10^{-5} as shown in Table 1. For very small τ , the block system is dominated by the two mass matrix blocks, and multigrid performs better as shown in Tables 1 and 2. Although it is not covered by our analysis in Sect. 5, numerical results in Table 2 show that multigrid methods with collective smoothers are robust for problems with degenerate diffusion coefficient.

6.2 Results of GMRes with Preconditioners Solved Inexactly by CGS-MG

In this subsection, we compare the performance of the preconditioned GMRes method using three different preconditioners $V_B(1, 1)$, $V_{\tilde{B}}(1, 1)$, $V_A(1, 1)$. Here, we use $V_B(1, 1)$, $V_{\tilde{B}}(1, 1)$, $V_A(1, 1)$ to represent the inexact preconditioners corresponding to B , \tilde{B} and A respectively. For example, $V_B(1, 1)$ is one CGS-MG V-cycle with one pre-smoothing and one post-smoothing. As we can see from Tables 3 and 4 that GMRes method converges uni-

Table 3 Example 1: iteration counts and CPU time (in seconds) of GMRes with one $V(1, 1)$ CGS-MG

| Preconditioner | h | τ | | | | |
|---------------------------------|-------|----------|----------|----------|----------|-----------|
| | | 1e-0 | 1e-1 | 1e-2 | 1e-3 | 1e-4 |
| $V_{\mathcal{B}}(1, 1)$ | 1/64 | 8 (0.16) | 8 (0.16) | 8 (0.16) | 7 (0.14) | 12 (0.25) |
| | 1/128 | 8 (0.76) | 8 (0.72) | 8 (0.72) | 7 (0.63) | 8 (0.72) |
| | 1/256 | 8 (3.5) | 8 (3.4) | 8 (3.4) | 7 (3.0) | 7 (3.0) |
| $V_{\tilde{\mathcal{B}}}(1, 1)$ | 1/64 | 8 (0.24) | 8 (0.21) | 8 (0.22) | 7 (0.19) | 9 (0.24) |
| | 1/128 | 8 (1.0) | 8 (1.0) | 8 (0.98) | 7 (0.87) | 7 (0.86) |
| | 1/256 | 8 (4.4) | 8 (4.4) | 8 (4.4) | 7 (3.9) | 7 (3.9) |
| $V_{\mathcal{A}}(1, 1)$ | 1/64 | 8 (0.33) | 8 (0.33) | 8 (0.34) | 8 (0.33) | 7 (0.29) |
| | 1/128 | 8 (1.5) | 8 (1.5) | 8 (1.5) | 8 (1.5) | 8 (1.5) |
| | 1/256 | 8 (6.9) | 8 (6.9) | 8 (6.9) | 7 (6.1) | 8 (6.9) |

Table 4 Example 2: iteration counts and CPU time (in seconds) of GMRes with one $V(1, 1)$ CGS-MG

| Preconditioner | h | τ | | | | |
|---------------------------------|-------|----------|----------|----------|----------|-----------|
| | | 1e-0 | 1e-1 | 1e-2 | 1e-3 | 1e-4 |
| $V_{\mathcal{B}}(1, 1)$ | 1/64 | 8 (0.17) | 8 (0.17) | 8 (0.17) | 8 (0.17) | 14 (0.30) |
| | 1/128 | 8 (0.77) | 8 (0.73) | 8 (0.73) | 8 (0.73) | 9 (0.82) |
| | 1/256 | 8 (3.5) | 8 (3.4) | 8 (3.4) | 8 (3.4) | 7 (3.0) |
| $V_{\tilde{\mathcal{B}}}(1, 1)$ | 1/64 | 8 (0.22) | 8 (0.21) | 8 (0.21) | 8 (0.22) | 8 (0.24) |
| | 1/128 | 8 (1.0) | 8 (0.98) | 8 (0.98) | 8 (0.98) | 7 (0.86) |
| | 1/256 | 8 (4.5) | 8 (4.5) | 8 (4.5) | 8 (4.5) | 7 (3.9) |
| $V_{\mathcal{A}}(1, 1)$ | 1/64 | 8 (0.32) | 8 (0.33) | 8 (0.34) | 8 (0.33) | 9 (0.37) |
| | 1/128 | 8 (1.5) | 8 (1.5) | 8 (1.5) | 8 (1.5) | 8 (1.5) |
| | 1/256 | 8 (6.9) | 8 (6.9) | 8 (6.9) | 8 (6.9) | 8 (6.9) |

Table 5 Example 1: iteration counts and CPU time (in seconds) of GMRes with one $V(1, 1)$ DGS-MG

| Preconditioner | h | τ | | | | |
|---------------------------------|-------|-----------|-----------|-----------|-----------|-----------|
| | | 1e-0 | 1e-1 | 1e-2 | 1e-3 | 1e-4 |
| $V_{\mathcal{B}}(1, 1)$ | 1/64 | 22 (0.28) | 22 (0.28) | 21 (0.26) | 19 (0.25) | 20 (0.25) |
| | 1/128 | 22 (1.1) | 22 (1.1) | 21 (1.1) | 20 (1.0) | 19 (0.98) |
| | 1/256 | 22 (5.4) | 22 (5.3) | 22 (5.3) | 21 (5.1) | 19 (4.6) |
| $V_{\tilde{\mathcal{B}}}(1, 1)$ | 1/64 | 22 (0.29) | 22 (0.30) | 21 (0.28) | 19 (0.25) | 17 (0.40) |
| | 1/128 | 22 (1.2) | 22 (1.2) | 21 (1.1) | 20 (1.1) | 19 (1.0) |
| | 1/256 | 22 (5.5) | 22 (5.5) | 22 (5.5) | 21 (5.2) | 19 (4.8) |

formly with respect to both h and τ . The CPU time for GMRes preconditioned by $V_{\mathcal{B}}(1, 1)$ is approximately half of the time when preconditioned by $V_{\mathcal{A}}(1, 1)$ which clearly demonstrate the efficiency of mass lumping.

Table 6 Example 2: iteration counts and CPU time (in seconds) of GMRes with one $V(1, 1)$ DGS-MG

| Preconditioner | h | τ | | | | |
|---------------------------------|-------|-----------|-----------|-----------|-----------|-----------|
| | | 1e-0 | 1e-1 | 1e-2 | 1e-3 | 1e-4 |
| $V_{\mathcal{B}}(1, 1)$ | 1/64 | 23 (0.29) | 22 (0.28) | 21 (0.27) | 19 (0.24) | 20 (0.25) |
| | 1/128 | 21 (1.1) | 22 (1.1) | 22 (1.1) | 20 (1.0) | 18 (0.92) |
| | 1/256 | 21 (5.0) | 21 (5.0) | 21 (5.0) | 20 (4.8) | 18 (4.3) |
| $V_{\tilde{\mathcal{B}}}(1, 1)$ | 1/64 | 23 (0.30) | 23 (0.30) | 22 (0.29) | 19 (0.26) | 17 (0.23) |
| | 1/128 | 22 (1.2) | 23 (1.2) | 22 (1.2) | 20 (1.1) | 17 (0.94) |
| | 1/256 | 21 (5.3) | 22 (5.5) | 21 (5.3) | 20 (5.0) | 18 (4.5) |

Table 7 Example 1: iteration count, CPU time (in seconds), and convergence factor (italics) of GMRes preconditioned by one $V_{\mathcal{B}}(k, k)$ CGS-MG, $k = 1, 2, 3$

| Precond | h | τ | | | | |
|-------------------------|-------|-------------|-------------|-------------|-------------|-------------|
| | | 1e-0 | 1e-1 | 1e-2 | 1e-3 | 1e-4 |
| $V_{\mathcal{B}}(1, 1)$ | 1/64 | 8 (0.16) | 8 (0.16) | 8 (0.16) | 7 (0.14) | 12 (0.25) |
| | | <i>0.09</i> | <i>0.09</i> | <i>0.09</i> | <i>0.10</i> | <i>0.23</i> |
| | 1/128 | 8 (0.76) | 8 (0.72) | 8 (0.72) | 7 (0.63) | 8 (0.72) |
| $V_{\mathcal{B}}(2, 2)$ | 1/64 | 8 (3.5) | 8 (3.4) | 8 (3.4) | 7 (3.0) | 7 (3.0) |
| | | <i>0.09</i> | <i>0.09</i> | <i>0.09</i> | <i>0.09</i> | <i>0.10</i> |
| | 1/128 | 6 (0.67) | 6 (0.63) | 5 (0.53) | 5 (0.53) | 6 (0.63) |
| $V_{\mathcal{B}}(3, 3)$ | 1/64 | 6 (0.16) | 5 (0.14) | 5 (0.14) | 5 (0.14) | 11 (0.30) |
| | | <i>0.02</i> | <i>0.02</i> | <i>0.04</i> | <i>0.4</i> | <i>0.24</i> |
| | 1/128 | 6 (0.75) | 5 (0.59) | 5 (0.59) | 4 (0.48) | 6 (0.70) |
| $V_{\mathcal{B}}(3, 3)$ | 1/64 | 6 (3.5) | 5 (2.9) | 5 (2.9) | 4 (2.3) | 5 (2.9) |
| | | <i>0.02</i> | <i>0.02</i> | <i>0.02</i> | <i>0.02</i> | <i>0.05</i> |

6.3 Results of GMRes with Preconditioners Solved Inexactly by DGS-MG

In this subsection, we present the numerical results for GMRes method using inexact mass lumping preconditioners $V_{\mathcal{B}}(1, 1)$ and $V_{\tilde{\mathcal{B}}}(1, 1)$. Namely, the preconditioner systems \mathcal{B} and $\tilde{\mathcal{B}}$ are solved approximately by one multigrid V-cycle using the decoupled smoother proposed in Sect. 4.3 with one pre-smoothing and one post-smoothing. Here, we choose the Jacobi damping parameter to be 0.5. Our numerical experiments show that the preconditioners have similar performance when the Jacobi damping parameter varies between 0.4 and 0.6. As we can see from Tables 5 and 6 that GMRes converges uniformly when varying h and τ .

Table 8 Example 1: iteration count, CPU time (in seconds), and convergence factor (italics) of GMRes preconditioned by one $W_{\mathcal{B}}(k, k)$ CGS-MG, $k = 1, 2, 3$

| Precond | h | τ | | | | |
|-------------------------|-------|-------------|-------------|-------------|-------------|-------------|
| | | 1e-0 | 1e-1 | 1e-2 | 1e-3 | 1e-4 |
| $W_{\mathcal{B}}(1, 1)$ | 1/64 | 8 (0.32) | 8 (0.32) | 7 (0.28) | 7 (0.28) | 12 (0.47) |
| | | <i>0.07</i> | <i>0.07</i> | <i>0.07</i> | <i>0.09</i> | <i>0.23</i> |
| | 1/128 | 8 (1.2) | 8 (1.1) | 7 (1.0) | 7 (1.0) | 8 (1.1) |
| $W_{\mathcal{B}}(2, 2)$ | 1/64 | 8 (5.1) | 8 (4.9) | 8 (4.9) | 7 (4.3) | 7 (4.3) |
| | | <i>0.07</i> | <i>0.07</i> | <i>0.07</i> | <i>0.07</i> | <i>0.08</i> |
| | 1/128 | 6 (0.28) | 6 (0.28) | 5 (0.23) | 5 (0.24) | 11 (0.51) |
| $W_{\mathcal{B}}(3, 3)$ | 1/64 | 6 (1.0) | 6 (1.0) | 5 (0.86) | 5 (0.87) | 6 (1.0) |
| | | <i>0.03</i> | <i>0.03</i> | <i>0.03</i> | <i>0.03</i> | <i>0.08</i> |
| | 1/128 | 6 (4.5) | 6 (4.4) | 5 (3.7) | 5 (3.7) | 5 (3.7) |
| $W_{\mathcal{B}}(1, 1)$ | 1/64 | 5 (0.27) | 5 (0.27) | 4 (0.22) | 5 (0.27) | 11 (0.59) |
| | | <i>0.02</i> | <i>0.02</i> | <i>0.02</i> | <i>0.4</i> | <i>0.24</i> |
| | 1/128 | 5 (1.0) | 5 (0.98) | 5 (0.98) | 4 (0.79) | 6 (1.2) |
| $W_{\mathcal{B}}(2, 2)$ | 1/64 | 5 (4.2) | 5 (4.2) | 5 (4.2) | 4 (3.4) | 5 (4.2) |
| | | <i>0.02</i> | <i>0.02</i> | <i>0.02</i> | <i>0.02</i> | <i>0.03</i> |
| | 1/128 | 5 (4.2) | 5 (4.2) | 5 (4.2) | 4 (3.4) | 5 (4.2) |
| $W_{\mathcal{B}}(3, 3)$ | 1/64 | 5 (0.27) | 5 (0.27) | 4 (0.22) | 5 (0.27) | 11 (0.59) |
| | | <i>0.02</i> | <i>0.02</i> | <i>0.02</i> | <i>0.4</i> | <i>0.24</i> |
| | 1/128 | 5 (1.0) | 5 (0.98) | 5 (0.98) | 4 (0.79) | 6 (1.2) |
| $W_{\mathcal{B}}(1, 1)$ | 1/64 | 5 (4.2) | 5 (4.2) | 5 (4.2) | 4 (3.4) | 5 (4.2) |
| | | <i>0.02</i> | <i>0.02</i> | <i>0.02</i> | <i>0.02</i> | <i>0.03</i> |
| | 1/128 | 5 (4.2) | 5 (4.2) | 5 (4.2) | 4 (3.4) | 5 (4.2) |
| $W_{\mathcal{B}}(2, 2)$ | 1/64 | 5 (0.27) | 5 (0.27) | 4 (0.22) | 5 (0.27) | 11 (0.59) |
| | | <i>0.02</i> | <i>0.02</i> | <i>0.02</i> | <i>0.4</i> | <i>0.24</i> |
| | 1/128 | 5 (1.0) | 5 (0.98) | 5 (0.98) | 4 (0.79) | 6 (1.2) |
| $W_{\mathcal{B}}(3, 3)$ | 1/64 | 5 (4.2) | 5 (4.2) | 5 (4.2) | 4 (3.4) | 5 (4.2) |
| | | <i>0.02</i> | <i>0.02</i> | <i>0.02</i> | <i>0.02</i> | <i>0.03</i> |
| | 1/128 | 5 (4.2) | 5 (4.2) | 5 (4.2) | 4 (3.4) | 5 (4.2) |

6.4 Choice of Multigrid Parameters for Solving Preconditioner Systems

In the following, we compare the performance of the preconditioner \mathcal{B} solved by CGS-MG or DGS-MG with different types of cycles and different number of smoothing steps. From Tables 7, 8, 9 and 10, we can see that increasing the number of smoothing steps can reduce the number of GMRes iterations. However, the CPU time is not always decreasing since more smoothing steps may also increase the computational cost. The GMRes iteration numbers are similar when using W cycle or V cycle. Since W cycle is usually more expensive, its efficiency in terms of CPU time is not as good as V cycle in our experiment. For practical purposes, we suggest using V cycle with one pre- and one post-smoothing step for solving the preconditioner systems discussed in this work.

6.5 Results of GMRes with Preconditioners $\tilde{\mathcal{B}}$ and \mathcal{B}_d

The theoretical analysis in Remark 2 shows that when $\tau < Ch^2$, the block diagonal preconditioner \mathcal{B}_d defined by (5.13) is also good for GMRes method. In Tables 11 and 12, we demonstrate the performance of inexact \mathcal{B}_d preconditioner when τ is very small. We use $GS_{\mathcal{B}_d}(3)$ to represent three steps of Gauss–Seidel relaxation applied to \mathcal{B}_d . It can be observed from the numerical results that $GS_{\mathcal{B}_d}(3)$ is very efficient when τ is very small. However, we would like to point out that from the approximation point of view, τ should be of order h^2 . Hence, in practice, a reasonable choice of τ may not be too small.

Table 9 Example 1: iteration count, CPU time (in seconds), and convergence factor (italics) of GMRes preconditioned by one $V_{\mathcal{B}}(k, k)$ DGS-MG, $k = 1, 2, 3$

| Precond | h | τ | | | | |
|-------------------------|-------|-------------|-------------|-------------|-------------|-------------|
| | | 1e-0 | 1e-1 | 1e-2 | 1e-3 | 1e-4 |
| $V_{\mathcal{B}}(1, 1)$ | 1/64 | 22 (0.28) | 22 (0.28) | 21 (0.26) | 19 (0.25) | 20 (0.25) |
| | | <i>0.48</i> | <i>0.45</i> | <i>0.44</i> | <i>0.44</i> | <i>0.42</i> |
| | 1/128 | 22 (1.1) | 22 (1.1) | 21 (1.1) | 20 (1.0) | 19 (0.98) |
| $V_{\mathcal{B}}(2, 2)$ | 1/64 | 22 (5.4) | 22 (5.3) | 22 (5.3) | 21 (5.1) | 19 (4.6) |
| | | <i>0.44</i> | <i>0.44</i> | <i>0.45</i> | <i>0.44</i> | <i>0.43</i> |
| | 1/256 | 16 (0.25) | 15 (0.23) | 15 (0.23) | 14 (0.23) | 16 (0.25) |
| $V_{\mathcal{B}}(3, 3)$ | 1/64 | 15 (0.97) | 15 (0.97) | 15 (0.96) | 14 (0.91) | 13 (0.85) |
| | | <i>0.3</i> | <i>0.31</i> | <i>0.33</i> | <i>0.29</i> | <i>0.28</i> |
| | 1/256 | 15 (4.4) | 15 (4.4) | 15 (4.3) | 14 (4.1) | 13 (3.8) |
| $V_{\mathcal{B}}(1, 1)$ | 1/64 | 13 (0.23) | 13 (0.24) | 13 (0.24) | 12 (0.23) | 14 (0.25) |
| | | <i>0.26</i> | <i>0.23</i> | <i>0.26</i> | <i>0.24</i> | <i>0.29</i> |
| | 1/128 | 13 (0.98) | 13 (0.97) | 13 (0.97) | 12 (0.91) | 11 (0.84) |
| $V_{\mathcal{B}}(2, 2)$ | 1/64 | 13 (4.3) | 13 (4.3) | 13 (4.3) | 12 (4.0) | 11 (3.7) |
| | | <i>0.24</i> | <i>0.24</i> | <i>0.25</i> | <i>0.25</i> | <i>0.24</i> |
| | 1/256 | 13 (0.23) | 13 (0.24) | 13 (0.24) | 12 (0.23) | 14 (0.25) |
| $V_{\mathcal{B}}(3, 3)$ | 1/64 | 13 (0.98) | 13 (0.97) | 13 (0.97) | 12 (0.91) | 11 (0.84) |
| | | <i>0.24</i> | <i>0.26</i> | <i>0.28</i> | <i>0.24</i> | <i>0.21</i> |
| | 1/256 | 13 (4.3) | 13 (4.3) | 13 (4.3) | 12 (4.0) | 11 (3.7) |
| $V_{\mathcal{B}}(1, 1)$ | 1/64 | 13 (4.3) | 13 (4.3) | 13 (4.3) | 12 (4.0) | 11 (3.7) |
| | | <i>0.24</i> | <i>0.24</i> | <i>0.25</i> | <i>0.25</i> | <i>0.24</i> |

6.6 Robustness with Respect to Other Boundary Conditions

In previous numerical experiments, homogeneous Dirichlet boundary conditions are prescribed. In Tables 13 and 14, we show that the proposed preconditioners also work well for problems with other boundary conditions. In particular, we consider example 1 defined at the beginning of Sect. 6 and choose mixed homogeneous Neumann and Dirichlet boundary conditions 2.3 and 2.4 where $\Gamma_N := \{(x, y)|x = 0, y \in (0, 1)\} \cup \{(x, y)|x \in (0, 1), y = 0\}$ and $\Gamma_D := \partial\Omega \setminus \Gamma_N$. There is a slight increase of GMRes iteration numbers when τ is large and the preconditioned systems are solved by CGS-MG.

6.7 Spectrum Distribution of the Preconditioned Systems

In Figs. 1 and 2, we show the spectrum distribution of the preconditioned systems, $\mathcal{B}^{-1}\mathcal{A}$ and $\tilde{\mathcal{B}}^{-1}\mathcal{A}$, respectively, when varying h (left) or τ (right) (with diffusion coefficients $a(x_1, x_2) = b(x_1, x_2) = 1$). We can see that the eigenvalues are more concentrated when h decreases with fixed τ . They become more scattered when τ gets smaller while h is fixed, but away from zero which means the original system is well conditioned. Moreover, the preconditioned system $\tilde{\mathcal{B}}^{-1}\mathcal{A}$ has more favorable spectral property than $\mathcal{B}^{-1}\mathcal{A}$ for GMRes method which is consistent with the numerical results reported in Sects. 6.2 and 6.3.

Table 10 Example 1: iteration count, CPU time (in seconds), and convergence factor (italics) of GMRes preconditioned by one $W_{\mathcal{B}}(k, k)$ DGS-MG, $k = 1, 2, 3$

| Precond | h | τ | | | | |
|-------------------------|-------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| | | 1e-0 | 1e-1 | 1e-2 | 1e-3 | 1e-4 |
| $W_{\mathcal{B}}(1, 1)$ | 1/64 | 21 (0.56) <i>0.41</i> | 21 (0.56) <i>0.40</i> | 20 (0.53) <i>0.42</i> | 19 (0.51) <i>0.41</i> | 19 (0.51) <i>0.42</i> |
| | 1/128 | 21 (1.8) <i>0.40</i> | 21 (1.9) <i>0.40</i> | 21 (1.8) <i>0.41</i> | 19 (1.7) <i>0.41</i> | 19 (1.7) <i>0.41</i> |
| | 1/256 | 21 (7.4) <i>0.40</i> | 21 (7.4) <i>0.41</i> | 21 (7.4) <i>0.41</i> | 20 (7.1) <i>0.42</i> | 19 (6.7) <i>0.42</i> |
| $W_{\mathcal{B}}(2, 2)$ | 1/64 | 15 (0.49) <i>0.26</i> | 15 (0.50) <i>0.26</i> | 14 (0.46) <i>0.27</i> | 13 (0.43) <i>0.27</i> | 15 (0.50) <i>0.33</i> |
| | 1/128 | 15 (1.7) <i>0.26</i> | 15 (1.6) <i>0.26</i> | 14 (1.5) <i>0.27</i> | 13(1.4) <i>0.26</i> | 13 (1.4) <i>0.28</i> |
| | 1/256 | 15 (6.5) <i>0.26</i> | 15 (6.5) <i>0.26</i> | 14 (6.0) <i>0.27</i> | 14 (6.1) <i>0.27</i> | 13 (5.7) <i>0.26</i> |
| $W_{\mathcal{B}}(3, 3)$ | 1/64 | 12 (0.46) <i>0.19</i> | 12 (0.47) <i>0.19</i> | 12 (0.46) <i>0.19</i> | 11 (0.43) <i>0.20</i> | 14 (0.53) <i>0.29</i> |
| | 1/128 | 12 (1.6) <i>0.19</i> | 12 (1.6) <i>0.19</i> | 12 (1.6) <i>0.19</i> | 11 (1.4) <i>0.20</i> | 11 (1.4) <i>0.21</i> |
| | 1/256 | 12 (6.0) <i>0.19</i> | 12 (6.0) <i>0.19</i> | 12 (6.0) <i>0.19</i> | 11 (5.6) <i>0.19</i> | 11 (5.6) <i>0.19</i> |

Table 11 Example 1: iteration counts and CPU time (in seconds) of GMRes with preconditioner $\tilde{\mathcal{B}}$ or \mathcal{B}_d

| Preconditioner | h | τ | | | |
|---------------------------------|-------|------------|-----------|-----------|-----------|
| | | 1e-4 | 1e-5 | 1e-6 | 1e-7 |
| $V_{\tilde{\mathcal{B}}}(1, 1)$ | 1/64 | 9 (0.24) | 12 (0.31) | 15 (0.39) | 15 (0.39) |
| | 1/128 | 7 (0.87) | 9 (1.1) | 14 (1.7) | 15 (1.8) |
| | 1/256 | 7 (3.8) | 8 (4.4) | 11 (6.0) | 15 (8.2) |
| $GS_{\mathcal{B}_d}(3)$ | 1/64 | 164 (0.99) | 19 (0.1) | 6 (0.03) | 4 (0.022) |
| | 1/128 | * | 68 (1.3) | 10 (0.17) | 5 (0.096) |
| | 1/256 | * | * | 29 (2.7) | 7 (0.62) |

* No convergence within 200 iterations

7 Conclusions

In this work, we propose mass lumping preconditioners for use with GMRes method to solve a class of two-by-two block linear systems which correspond to some discrete fourth order parabolic equations. Using consistent matrix in the discretization has the advantage of better solution accuracy as compared with applying mass lumping directly in the discretization. We propose to use lumped-mass matrix \bar{M} and optimal multigrid algorithm with either coupled or decoupled smoother to construct practical preconditioners which are shown to be

Table 12 Example 2: iteration counts and CPU time (in seconds) of GMRes with preconditioner \tilde{B} or B_d

| Preconditioner | h | τ | | | |
|-----------------------|-------|----------|-----------|-----------|-----------|
| | | 1e-4 | 1e-5 | 1e-6 | 1e-7 |
| $V_{\tilde{B}}(1, 1)$ | 1/64 | 8 (0.21) | 13 (0.35) | 15 (0.39) | 15 (0.39) |
| | 1/128 | 7 (0.88) | 12 (1.5) | 15 (1.8) | 15 (1.8) |
| | 1/256 | 7 (3.9) | 8 (4.4) | 14 (7.7) | 15 (8.2) |
| $GS_{B_d}(3)$ | 1/64 | * | 58 (0.35) | 9 (0.043) | 5 (0.026) |
| | 1/128 | * | * | 25 (0.43) | 7 (0.12) |
| | 1/256 | * | * | 96 (10) | 12 (1.0) |

* No convergence within 200 iterations

Table 13 Example 1 with mixed boundary conditions: iteration counts and CPU time (in seconds) of GMRes with one V(1, 1) CGS-MG

| Preconditioner | h | τ | | | | |
|-----------------------|-------|-----------|-----------|----------|----------|-----------|
| | | 1e-0 | 1e-1 | 1e-2 | 1e-3 | 1e-4 |
| $V_B(1, 1)$ | 1/64 | 10 (0.21) | 9 (0.20) | 8 (0.17) | 7 (0.15) | 12 (0.26) |
| | 1/128 | 10 (0.93) | 10 (0.91) | 9 (0.83) | 7 (0.64) | 8 (0.73) |
| | 1/256 | 10 (4.4) | 10 (4.3) | 9 (3.9) | 8 (3.4) | 7 (3.0) |
| $V_{\tilde{B}}(1, 1)$ | 1/64 | 10 (0.26) | 10 (0.26) | 8 (0.21) | 7 (0.20) | 9 (0.24) |
| | 1/128 | 10 (1.3) | 10 (1.3) | 9 (1.4) | 7 (0.87) | 7 (0.88) |
| | 1/256 | 10 (5.7) | 10 (5.7) | 9 (5.1) | 8 (4.6) | 7 (4.0) |
| $V_A(1, 1)$ | 1/64 | 10 (0.40) | 10 (0.40) | 8 (0.33) | 8 (0.32) | 7 (0.28) |
| | 1/128 | 10 (2.0) | 10 (2.0) | 9 (1.7) | 8 (1.6) | 8 (1.6) |
| | 1/256 | 10 (8.8) | 10 (8.8) | 9 (7.9) | 8 (7.1) | 8 (7.0) |

Table 14 Example 1 with mixed boundary conditions: iteration counts and CPU time (in seconds) of GMRes with one W(2, 2) DGS-MG

| Preconditioner | h | τ | | | | |
|-----------------------|-------|-----------|-----------|-----------|-----------|-----------|
| | | 1e-0 | 1e-1 | 1e-2 | 1e-3 | 1e-4 |
| $W_B(2, 2)$ | 1/64 | 15 (0.53) | 15 (0.51) | 15 (0.50) | 13 (0.44) | 15 (0.50) |
| | 1/128 | 15 (1.7) | 15 (1.7) | 15 (1.7) | 14 (1.6) | 13 (1.5) |
| | 1/256 | 15 (6.6) | 15 (6.6) | 15 (6.6) | 14 (6.2) | 13 (5.8) |
| $W_{\tilde{B}}(2, 2)$ | 1/64 | 15 (0.58) | 15 (0.56) | 15 (0.55) | 13 (0.48) | 12 (0.45) |
| | 1/128 | 15 (1.9) | 15 (1.9) | 15 (1.9) | 14 (1.7) | 13 (1.6) |
| | 1/256 | 15 (7.2) | 15 (7.2) | 15 (7.2) | 14 (6.7) | 13 (6.3) |

computationally very efficient. Numerical experiments indicate that preconditioned GMRes method converges uniformly with respect to both h and τ and is robust for problems with degenerate diffusion coefficient. We provide theoretical analysis about the spectrum bound of

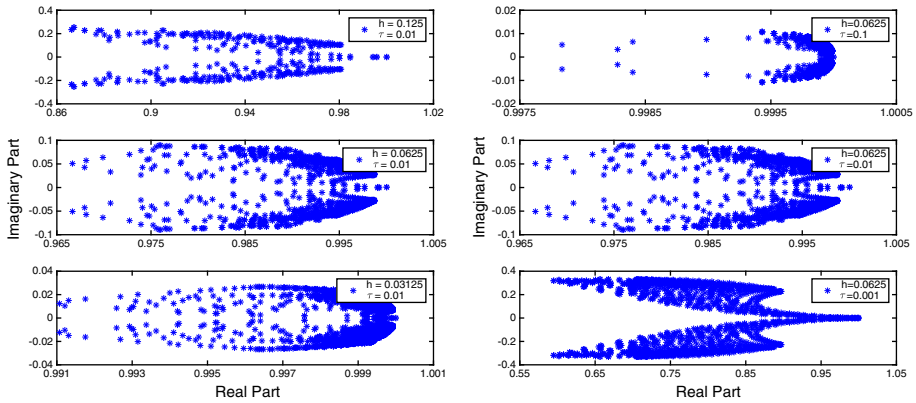


Fig. 1 Spectrum distribution of $B^{-1}A$ when varying h (left) and τ (right)

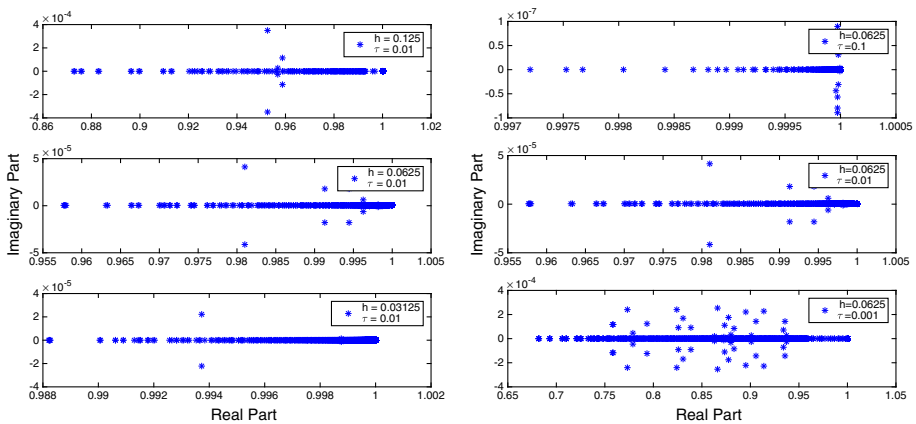


Fig. 2 Spectrum distribution of $\tilde{B}^{-1}A$ when varying h (left) and τ (right)

the preconditioned system and estimate the convergence rate of the preconditioned GMRes method.

Acknowledgments B. Zheng would like to acknowledge the support by NSF Grant DMS-0807811 and a Laboratory Directed Research and Development (LDRD) Program from Pacific Northwest National Laboratory. L.P. Chen was supported by the National Natural Science Foundation of China under Grant No. 11501473. L. Chen was supported by NSF Grant DMS-1418934 and in part by NIH Grant P50GM76516. R.H. Nochetto was supported by NSF under Grants DMS-1109325 and DMS-1411808. J. Xu was supported by NSF Grant DMS-1522615 and in part by US Department of Energy Grant DE-SC0014400. Computations were performed using the computational resources of Pacific Northwest National Laboratory (PNNL) Institutional Computing cluster systems. The PNNL is operated by Battelle for the US Department of Energy under Contract DE-AC05-76RL01830.

References

1. Asaro, R.J., Tiller, W.A.: Interface morphology development during stress corrosion cracking: part I. Via surface diffusion. *Metall. Trans.* **3**(7), 1789–1796 (1972)

2. Axelsson, O., Boyanova, P., Kronbichler, M., Neytcheva, M., Wu, X.: Numerical and computational efficiency of solvers for two-phase problems. *Comput. Math. Appl.* **65**(3), 301–314 (2013)
3. Axelsson, O., Neytcheva, M.: Operator splitting for solving nonlinear, coupled multiphysics problems with application to numerical solution of an interface problem. TR2011-009 Institute for Information Technology, Uppsala University (2011)
4. Bai, Z.-Z., Benzi, M., Chen, F., Wang, Z.-Q.: Preconditioned MHSS iteration methods for a class of block two-by-two linear systems with applications to distributed control problems. *IMA J. Numer. Anal.* **33**(1), 343–369 (2013)
5. Bai, Z.-Z., Chen, F., Wang, Z.-Q.: Additive block diagonal preconditioning for block two-by-two linear systems of skew-Hamiltonian coefficient matrices. *Numer. Algorithms* **62**(4), 655–675 (2013)
6. Bai, Z.-Z., Golub, G.H., Ng, M.K.: Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. *SIAM J. Matrix Anal. Appl.* **24**(3), 603–626 (2003)
7. Bañas, L., Nürnberg, R.: A multigrid method for the Cahn–Hilliard equation with obstacle potential. *Appl. Math. Comput.* **213**(2), 290–303 (2009)
8. Bänsch, E., Morin, P., Nochetto, R.H.: Surface diffusion of graphs: variational formulation, error analysis, and simulation. *SIAM J. Numer. Anal.* **42**(2), 773–799 (2004)
9. Bänsch, E., Morin, P., Nochetto, R.H.: A finite element method for surface diffusion: the parametric case. *J. Comput. Phys.* **203**(1), 321–343 (2005)
10. Bänsch, E., Morin, P., Nochetto, R.H.: Preconditioning a class of fourth order problems by operator splitting. *Numer. Math.* **118**(2), 197–228 (2011)
11. Barrett, J.W., Blowey, J.F., Garcke, H.: Finite element approximation of the Cahn–Hilliard equation with degenerate mobility. *SIAM J. Numer. Anal.* **37**(1), 286–318 (1999)
12. Benzi, M., Golub, G.H.: A preconditioner for generalized saddle point problems. *SIAM J. Matrix Anal. Appl.* **26**(1), 20–41 (2004)
13. Bertozzi, A.L., Esedoglu, S., Gillette, A.: Inpainting of binary images using the Cahn–Hilliard equation. *IEEE Trans. Image Process.* **16**(1), 285–291 (2007)
14. Blowey, J.F., Elliott, C.M.: The Cahn–Hilliard gradient theory for phase separation with non-smooth free energy part I: mathematical analysis. *Eur. J. Appl. Math.* **2**(3), 233–280 (1991)
15. Blowey, J.F., Elliott, C.M.: The Cahn–Hilliard gradient theory for phase separation with non-smooth free energy. Part II: numerical analysis. *Eur. J. Appl. Math.* **3**, 147–179 (1992)
16. Bosch, J., Kay, D., Stoll, M., Wathen, A.J.: Fast solvers for Cahn–Hilliard inpainting. *SIAM J. Imaging Sci.* **7**(1), 67–97 (2014)
17. Bosch, J., Stoll, M., Benner, P.: Fast solution of Cahn–Hilliard variational inequalities using implicit time discretization and finite elements. *J. Comput. Phys.* **262**, 38–57 (2014)
18. Boyanova, P., Do-Quang, M., Neytcheva, M.: Efficient preconditioners for large scale binary Cahn–Hilliard models. *Comput. Methods Appl. Math.* **12**(1), 1–22 (2012)
19. Boyanova, P., Neytcheva, M.: Efficient numerical solution of discrete multi-component Cahn–Hilliard systems. *Comput. Math. Appl.* **67**(1), 106–121 (2014)
20. Brandt, A., Dinar, N.: Multi-grid solutions to elliptic flow problems. *Numerical Methods for Partial Differential Equations*, pp. 53–147 (1979)
21. Brenner, S.C., Scott, R.: *The Mathematical Theory of Finite Element Methods*, vol. 15. Springer, Berlin (2008)
22. Cahn, J.W., Hilliard, J.E.: Free energy of a nonuniform system. I. Interfacial free energy. *J. Chem. Phys.* **28**(2), 258–267 (2004)
23. Chen, C.M., Thomée, V.: The lumped mass finite element method for a parabolic problem. *J. Aust. Math. Soc. Ser. B Appl. Math.* **26**(03), 329–354 (1985)
24. Chen, L.: iFEM: an integrated finite element methods package in MATLAB. Technical Report, University of California at Irvine (2009)
25. Chen, L.: Multigrid methods for constrained minimization problems and application to saddle point problems. Submitted (2014)
26. Chen, L.: Multigrid methods for saddle point systems using constrained smoothers. *Comput. Math. Appl.* **70**(12), 2854–2866 (2015)
27. Choo, S.M., Lee, Y.J.: A discontinuous Galerkin method for the Cahn–Hilliard equation. *J. Appl. Math. Comput.* **18**(1–2), 113–126 (2005)
28. Christon, M.A.: The influence of the mass matrix on the dispersive nature of the semi-discrete, second-order wave equation. *Comput. Methods Appl. Mech. Eng.* **173**(1), 147–166 (1999)
29. Du, Q., Nicolaides, R.A.: Numerical analysis of a continuum model of phase transition. *SIAM J. Numer. Anal.* **28**(5), 1310–1322 (1991)
30. Elliott, C.M.: The Cahn–Hilliard model for the kinetics of phase separation. In: Rodrigues, J.F. (ed.) *Mathematical Models for Phase Change Problems*, pp. 35–73. Springer (1989)

31. Elliott, C.M., French, D.A.: Numerical studies of the Cahn–Hilliard equation for phase separation. *IMA J. Appl. Math.* **38**(2), 97–128 (1987)
32. Elliott, C.M., French, D.A.: A nonconforming finite-element method for the two-dimensional Cahn–Hilliard equation. *SIAM J. Numer. Anal.* **26**(4), 884–903 (1989)
33. Elliott, C.M., French, D.A., Milner, F.A.: A second order splitting method for the Cahn–Hilliard equation. *Numer. Math.* **54**(5), 575–590 (1989)
34. Elliott, C.M., Larsson, S.: Error estimates with smooth and nonsmooth data for a finite element method for the Cahn–Hilliard equation. *Math. Comput.* **58**(198), 603–630 (1992)
35. Feng, X., Karakashian, O.: Fully discrete dynamic mesh discontinuous Galerkin methods for the Cahn–Hilliard equation of phase transition. *Math. Comput.* **76**(259), 1093–1117 (2007)
36. Feng, X., Prohl, A.: Error analysis of a mixed finite element method for the Cahn–Hilliard equation. *Numer. Math.* **99**(1), 47–84 (2004)
37. Feng, X., Prohl, A.: Numerical analysis of the Cahn–Hilliard equation and approximation for the Hele–Shaw problem. *J. Comput. Math.* **26**(6), 767–796 (2008)
38. Fried, I.: Bounds on the spectral and maximum norms of the finite element stiffness, flexibility and mass matrices. *Int. J. Solids Struct.* **9**(9), 1013–1034 (1973)
39. Furihata, D.: A stable and conservative finite difference scheme for the Cahn–Hilliard equation. *Numer. Math.* **87**(4), 675–699 (2001)
40. Gaspar, F.J., Lisbona, F.J., Oosterlee, C.W., Wienands, R.: A systematic comparison of coupled and distributive smoothing in multigrid for the poroelasticity system. *Numer. Linear Algebra Appl.* **11**(2–3), 93–113 (2004)
41. Gräser, C., Kornhuber, R.: On preconditioned Uzawa-type iterations for a saddle point problem with inequality constraints. In: Wundlung, O.B., Keyes, D.E. (eds.) *Domain Decomposition Methods in Science and Engineering XVI*, Volume 55 of Lecture Notes in Computational Science Engineering, pp. 91–102. Springer, Berlin (2007)
42. Greer, J.B., Bertozzi, A.L.: H^1 solutions of a class of fourth order nonlinear equations for image processing. *Discrete Contin. Dyn. Syst.* **10**(1/2), 349–366 (2004)
43. Gresho, P.M., Lee, R.L., Sani, R.L.: Advection-dominated flows, with emphasis on the consequences of mass lumping. *Finite Elem. Fluids* **1**, 335–350 (1978)
44. He, Y., Liu, Y.: Stability and convergence of the spectral Galerkin method for the Cahn–Hilliard equation. *Numer. Methods Partial Differ. Equ.* **24**(6), 1485–1500 (2008)
45. Henn, S.: A multigrid method for a fourth-order diffusion equation with application to image processing. *SIAM J. Sci. Comput.* **27**(3), 831–849 (2005)
46. Hinton, E., Rock, T., Zienkiewicz, O.C.: A note on mass lumping and related processes in the finite element method. *Earthq. Eng. Struct. Dyn.* **4**(3), 245–249 (1976)
47. Kay, D., Welford, R.: A multigrid finite element solver for the Cahn–Hilliard equation. *J. Comput. Phys.* **212**(1), 288–304 (2006)
48. Kim, J., Kang, K., Lowengrub, J.: Conservative multigrid methods for Cahn–Hilliard fluids. *J. Comput. Phys.* **193**(2), 511–543 (2004)
49. King, B.B., Stein, O., Winkler, M.: A fourth-order parabolic equation modeling epitaxial thin film growth. *J. Math. Anal. Appl.* **286**(2), 459–490 (2003)
50. Lass, O., Vallejos, M., Borzi, A., Douglas, C.C.: Implementation and analysis of multigrid schemes with finite elements for elliptic optimal control problems. *Computing* **84**(1–2), 27–48 (2009)
51. Mullen, R., Belytschko, T.: Dispersion analysis of finite element semidiscretizations of the two-dimensional wave equation. *Int. J. Numer. Methods Eng.* **18**(1), 11–29 (1982)
52. Niclason, D.A., Blackburn, H.M.: A comparison of mass lumping techniques for the two-dimensional Navier–Stokes equations.pdf. In: *Twelfth Australasian Fluid Mechanics Conference*, pp. 731–734. The University of Sydney (1995)
53. Olshanskii, M.A., Reusken, A.: Navier–Stokes equations in rotation form: a robust multigrid solver for the velocity problem. *SIAM J. Sci. Comput.* **23**(5), 1683–1706 (2002)
54. Quarteroni, A.: On mixed methods for fourth-order problems. *Comput. Methods Appl. Mech. Eng.* **24**(1), 13–34 (1980)
55. Schöberl, J.: Multigrid methods for a parameter dependent problem in primal variables. *Numer. Math.* **84**, 97–119 (1999)
56. Sun, Z.: A second-order accurate linearized difference scheme for the two-dimensional Cahn–Hilliard equation. *Math. Comput.* **64**(212), 1463–1471 (1995)
57. Takacs, S., Zulehner, W.: Convergence analysis of multigrid methods with collective point smoothers for optimal control problems. *Comput. Vis. Sci.* **14**(3), 131–141 (2011)
58. Trefethen, L.N., Embree, M.: *Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators*. Princeton University Press, Princeton (2005)

59. Ushijima, T.: On the uniform convergence for the lumped mass approximation of the heat equation. *J. Fac. Sci. Univ. Tokyo* **24**, 477–490 (1977)
60. Ushijima, T.: Error estimates for the lumped mass approximation of the heat equation. *Mem. Numer. Math.* **6**, 65–82 (1979)
61. Vanka, S.P.: Block-implicit multigrid solution of Navier–Stokes equations in primitive variables. *J. Comput. Phys.* **65**, 138–158 (1986)
62. Wang, M., Chen, L.: Multigrid methods for the stokes equations using distributive Gauss–Seidel relaxations based on the least squares commutator. *J. Sci. Comput.* **56**(2), 409–431 (2013)
63. Wise, S., Kim, J., Lowengrub, J.: Solving the regularized, strongly anisotropic Cahn–Hilliard equation by an adaptive nonlinear multigrid method. *J. Comput. Phys.* **226**(1), 414–446 (2007)
64. Wittum, G.: Multigrid methods for Stokes and Navier–Stokes equations with transforming smoothers: algorithms and numerical results. *Numer. Math.* **54**(5), 543–563 (1989)
65. Xia, Y., Xu, Y., Shu, C.W.: Local discontinuous Galerkin methods for the Cahn–Hilliard type equations. *J. Comput. Phys.* **227**(1), 472–491 (2007)
66. Ye, X., Cheng, X.: The Fourier spectral method for the Cahn–Hilliard equations. *Numer. Math.* **171**(1), 345–357 (2005)
67. Zhang, S., Wang, M.: A nonconforming finite element method for the Cahn–Hilliard equation. *J. Comput. Phys.* **229**(19), 7361–7372 (2010)