J Sci Comput (2018) 74:396–411
https://doi.org/10.1007/s10915-017-0466-z

Multigrid Methods for a Mixed Finite Element Method of the Darcy–Forchheimer Model

Jian Huang[1] · **Long Chen**[2,3] · **Hongxing Rui**[1]

Received: 16 March 2016 / Revised: 9 March 2017 / Accepted: 25 April 2017 /
Published online: 9 June 2017

Abstract An efficient nonlinear multigrid method for a mixed finite element method of the Darcy–Forchheimer model is constructed in this paper. A Peaceman–Rachford type iteration is used as a smoother to decouple the nonlinearity from the divergence constraint. The nonlinear equation can be solved element-wise with a closed formulae. The linear saddle point system for the constraint is reduced into a symmetric positive definite system of Poisson type. Furthermore an empirical choice of the parameter used in the splitting is proposed and the resulting multigrid method is robust to the so-called Forchheimer number which controls the strength of the nonlinearity. By comparing the number of iterations and CPU time of different solvers in several numerical experiments, our multigrid method is shown to convergent with a rate independent of the mesh size and the Forchheimer number and with a nearly linear computational cost.

Keywords Darcy–Forchheimer model · Multigrid method · Peaceman–Rachford iteration

The work of Jian Huang and Hongxing Rui was supported by the National Natural Science Foundation of China Grant No. 11671233, and in part by the Science Challenge Project No. JCKY2016212A502. Long Chen was supported by NSF Grant DMS-1418934, in part by NIH Grant P50GM76516, and in part by the Sea Poly Project of Beijing Overseas Talents. The work of Jian Huang was supported by 2014 China Scholarship Council (CSC).

✉ Long Chen
chenlong@math.uci.edu

Jian Huang
yghuangjian@sina.com

Hongxing Rui
hxrui@sdu.edu.cn

[1] School of Mathematics, Shandong University, Jinan 250100, Shandong, China

[2] Beijing Institute for Scientific and Engineering Computing, Beijing University of Technology, Beijing 100124, China

[3] Department of Mathematics, University of California at Irvine, Irvine, CA 92697, USA

1 Introduction

Darcy's law

$$\boldsymbol{u} = -\frac{\boldsymbol{K}}{\mu}\nabla p,$$

with the permeability tensor $\boldsymbol{K}$ and the viscosity coefficient μ, describes the linear relationship between the velocity $\boldsymbol{u}$ of the creep flow and the gradient of the pressure p, which is valid when the Darcy velocity $\boldsymbol{u}$ is extremely small [5]. Forchheimer in [14] carried out flow experiments and pointed out that when the velocity is relatively high, Darcy's law should be replaced by the so-called Darcy–Forchheimer (DF) equation by adding a quadratic nonlinear term to the velocity, shown as follows:

$$\frac{\mu}{\rho}\boldsymbol{K}^{-1}\boldsymbol{u} + \frac{\beta}{\rho}\left|\boldsymbol{u}\right|\boldsymbol{u} + \nabla p = \boldsymbol{0}, \tag{1.1}$$

where ρ and β represent the density of the fluid and its dynamic viscosity, respectively. The parameter β is also referred to as the Forchheimer number, which controls the strength of nonlinearity. A theoretical derivation of the Darcy–Forchheimer equation (1.1) can be found in [26]. Equation (1.1) coupled with the conservation law

$$\operatorname{div}\boldsymbol{u} = g \tag{1.2}$$

are usually called Darcy–Forchheimer model.

In recent years, many numerical methods of the Darcy–Forchheimer model have been developed. Girault and Wheeler in [15] proved the existence and uniqueness of the solution of the Darcy–Forchheimer model (1.1)–(1.2) by proving the nonlinear operator $\mathcal{A}(\boldsymbol{v}) = \frac{\mu}{\rho}\boldsymbol{K}^{-1}\boldsymbol{v} + \frac{\beta}{\rho}\left|\boldsymbol{v}\right|\boldsymbol{v}$ is monotone, coercive and hemi-continuous, and establishing an appropriate inf-sup condition. Then they considered mixed finite element methods by approximating the velocity and the pressure by piecewise constant and nonconforming Crouzeix–Raviart (CR) elements, respectively. They proved a discrete inf-sup condition and the convergence of the mixed finite element scheme. They also proposed a Peaceman–Rachford (PR) type iterative method to solve the discretized nonlinear system and proved convergence of this iterative solver. In the PR iteration, the nonlinear equation can be decoupled with the divergence constraint and solved in a closed form; see Sect. 4 for details. López et al. in [17] carried out numerical tests of the methods proposed in [15], and made a comparative study between Newton's method and the PR iterative method. They pointed out that Newton's method is not competitive with the PR iteration. In each iteration, Newton's method needs to evaluate a Jacobian and solves a linear saddle point system, but the PR iteration computes an intermediate solution for a decoupled nonlinear equation and then solves a simplified linear saddle point system. The cost of solving the decoupled nonlinear equation can be negligible in comparison with the Jacobian evaluation. Furthermore the PR iteration required fewer iterations to converge than Newton's method with the same initial guess; see [17] for details.

Park in [21] developed a mixed finite element method with a semi-discrete scheme for the time dependent Darcy–Forchheimer model. Pan and Rui in [20] gave a mixed element method for the Darcy–Forchheimer model based on the Raviart–Thomas (RT) element or the Brezzi–Douglas–Marini (BDM) element approximation of the velocity and piecewise constant (P0) approximation of the pressure. Rui and Pan in [24] proposed a block-centered finite difference method for the Darcy–Forchheimer model, which was thought of as the lowest-order RT-P0 mixed element with proper quadrature formula. Rui et al. in [25] presented a block-centered finite difference method for the Darcy–Forchheimer model with variable Forchheimer number

$\beta(\boldsymbol{x})$. Wang and Rui in [30] constructed a stabilized CR element for the Darcy–Forchheimer model. Rui and Liu in [23] introduced a two-grid block-centered finite difference method for the Darcy–Forchheimer model. Salas et al. in [27] presented a theoretical study of the mixed finite element method proposed in [17], and showed the well-posedness and convergence.

Most of work mentioned above mainly focus on the discretization of the Darcy–Forchheimer model. Except the PR iteration presented in [15], no other work concentrates on fast solvers of the discretized nonlinear saddle point system which will be the topic of this paper. Multigrid method is one of the most efficient methods on solving the linear and nonlinear elliptic systems. It should be clarified that for nonlinear problems we no longer have a simple linear residual equation, which is the most significant difference between linear and nonlinear systems. The multigrid scheme we used here is the most commonly used nonlinear version of multigrid. It is called the *full approximation scheme* (FAS) [9] because the problem in the coarse grid is solved for the full approximation rather than the correction; see Sect. 5 for details.

We shall use piecewise constant (P0) and continuous piecewise linear polynomial (P1) to discretize the velocity and the pressure, respectively. We refer to [27] for the convergence analysis of this scheme and focus on fast solvers in our study. We shall apply FAS to construct an efficient V-cycle multigrid method for the nonlinear Darcy–Forchheimer model and demonstrate the efficiency of our multigrid method. Similar application of FAS to a nonlinear saddle point system (for Cahn–Hillard type equations) can be found in [4,31]. Recall that the success of multigrid method relies on two ingredients: the high frequency can be damped efficiently by the smoother, and the low frequency can be well approximated by the coarse grid correction. Notice that for saddle point systems, both smoothing and coarse grid corrections can easily violate the constraint [11]. The main difficulty of developing robust and effective multigrid methods for the saddle point system is to design an effective smoother with the consideration of the constraint $\operatorname{div} \boldsymbol{u} = g$. We shall use the Peaceman–Rachford iteration developed in [15] as a smoother since the nonlinearity can be handled efficiently and the constraint is always satisfied after solving a linear saddle point system. To enforce the constraint after the coarse grid correction, we also project the correction into the divergence free subspace. This is in the sprit of the B-S smoother developed in [7] for the Stokes equation except here we are dealing with a harder nonlinear equation instead of a linear Stokes equation.

The most relevant work is [17] and our improvement are:

1. We reduce the linear saddle point system into a SPD system and demonstrate the efficiency of our approach.
2. We report a better choice of the splitting parameter α for decoupling the nonlinearity from the constraint rather than the suggested value $\alpha = 1$ in [17] for different values of the Forchheimer number β, and show the advantage of our choice by comparing the number of iterations and CPU time.
3. We carry out some experiments to show the efficiency of our multigrid solver. Our method is convergent with a rate independent of the mesh size and the Forchheimer number and with a nearly linear computational cost. Notice that it is not easy to construct a fast solver robust to a critical parameter, see, for example, a linear Stokes-type equation [18,19].

The remainder of this article is organized as follows: The model problem is demonstrated in Sect. 2. The mixed weak formulation and the discrete weak formulation are presented in Sect. 3. The PR iteration and an efficient solver for the linear saddle point systems are posted in Sect. 4. A V-cycle multigrid scheme by applying FAS for the nonlinear problem is constructed in Sect. 5. Some numerical experiments using our multigrid method are carried out in Sect. 6 to verify that the efficiency of our method in comparison with solving this nonliear problem using the other iterative methods. Finally, conclusions and further ideas are presented in Sect. 7.

2 The Problem and Notation

We consider the steady Darcy–Forchheimer flow of a single phase fluid in a porous medium in a two-dimensional bounded domain Ω, with Lipschitz continuous boundary $\partial\Omega$:

$$\frac{\mu}{\rho}\boldsymbol{K}^{-1}\boldsymbol{u}+\frac{\beta}{\rho}|\boldsymbol{u}|\,\boldsymbol{u}+\nabla p=\boldsymbol{f}\quad \text{in } \Omega, \tag{2.1}$$

with the divergence constraint

$$\operatorname{div}\boldsymbol{u}=g\quad \text{in } \Omega, \tag{2.2}$$

and Neumann boundary condition,

$$\boldsymbol{u}\cdot\boldsymbol{n}=g_N\quad \text{on } \partial\Omega, \tag{2.3}$$

where $\boldsymbol{u}$ and p are the velocity vector and the pressure, respectively; μ, ρ and β are given positive constants that represent the viscosity of the fluid, its density and its dynamic viscosity, respectively; $|\cdot|$ denotes the Euclidean vector norm $|\boldsymbol{u}|^2=\boldsymbol{u}\cdot\boldsymbol{u}$, $\boldsymbol{n}$ is the unit exterior normal vector to the boundary of the given domain Ω; $\boldsymbol{K}$ is the permeability tensor, assumed to be uniformly positive definite and bounded. According to the divergence theorem, g and g_N are given functions satisfying the compatibility condition

$$\int_\Omega g\,(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}=\int_{\partial\Omega} g_N\,(\sigma)\,\mathrm{d}\sigma. \tag{2.4}$$

We use the standard notation of the Sobolev spaces and the associated norms, see e.g.[1].

3 The Weak Formulation

Following [15], we define the function spaces as follows:

$$X=L^3(\Omega)^2,$$
$$M=W^{1,\frac{3}{2}}(\Omega)\cap L_0^2\,(\Omega)\,,$$

where the zero mean value condition

$$L_0^2\,(\Omega)=\left\{v\in L^2\,(\Omega):\int_\Omega v\,(\boldsymbol{x})\;\mathrm{d}\boldsymbol{x}=0\right\},$$

is added because p is only defined by (2.1)–(2.3) up to an additive constant. Given $\boldsymbol{f}\in L^3(\Omega)^2$, $g\in L^{\frac{6}{5}}(\Omega)$, and $g_N\in L^{\frac{3}{2}}\,(\partial\Omega)$, the variational formulation of (2.1)–(2.3) is: find a pair $(\boldsymbol{u},p)$ in $X\times M$ such that

$$\frac{\mu}{\rho}\int_\Omega\left(\boldsymbol{K}^{-1}\boldsymbol{u}\right)\cdot\boldsymbol{\varphi}\;\mathrm{d}\boldsymbol{x}+\frac{\beta}{\rho}\int_\Omega|\boldsymbol{u}|\,(\boldsymbol{u}\cdot\boldsymbol{\varphi})\;\mathrm{d}\boldsymbol{x}$$
$$+\int_\Omega\nabla p\cdot\boldsymbol{\varphi}\;\mathrm{d}\boldsymbol{x}=\int_\Omega\boldsymbol{f}\cdot\boldsymbol{\varphi}\;\mathrm{d}\boldsymbol{x},\quad\forall\boldsymbol{\varphi}\in X, \tag{3.1}$$

$$\int_\Omega \nabla q \cdot \boldsymbol{u} \, \mathrm{d}\boldsymbol{x} = -\int_\Omega g q \, \mathrm{d}\boldsymbol{x} + \int_{\partial\Omega} g_N q \, \mathrm{d}\boldsymbol{x}, \quad \forall q \in M. \tag{3.2}$$

The variational formulation (3.1)–(3.2) and the original problem (2.1)–(2.3) are equivalent by using the Green's formula:

$$\int_\Omega \boldsymbol{v} \cdot \nabla q \, dx = -\int_\Omega q \operatorname{div} \boldsymbol{v} \, dx + \langle q, \boldsymbol{v} \cdot \boldsymbol{n} \rangle_{\partial\Omega}, \quad \forall q \in M, \forall \boldsymbol{v} \in H, \tag{3.3}$$

where

$$H = \left\{ \boldsymbol{v} \in L^3(\Omega)^2 : \operatorname{div} \boldsymbol{v} \in L^{\frac{6}{5}}(\Omega) \right\}.$$

In [15], Girault and Wheeler showed that if the given functions g and g_N satisfy the compatibility condition (2.4), then the problem has a unique solution $(\boldsymbol{u}, p)$ in $X \times M$.

Let Ω be a polygon in two dimensions which can be completely triangulated by triangles. Let $\mathcal{T}_1$ be a triangulation of Ω, and the triangulations $\mathcal{T}_k$ $(k = 2, 3, \ldots)$ be obtained form $\mathcal{T}_1$ via regular subdivision, i.e. edge midpoints in $\mathcal{T}_{k-1}$ are connected by new edges to form $\mathcal{T}_k$. Therefore, $\mathcal{T}_k$ is a family of conforming triangulations of $\overline{\Omega}$,

$$\overline{\Omega} = \bigcup_{T \in \mathcal{T}_k} T \quad \text{for } k = 1, 2, 3, \ldots,$$

The family $\mathcal{T}_k$ is shape regular in the sense of Ciarlet [13].

We discretize $\boldsymbol{u}$ and p in different finite element spaces. The velocity $\boldsymbol{u}$ is approximated in the following space:

$$X_k = \left\{ \boldsymbol{v} \in L^2(\Omega)^2 : \forall T \in \mathcal{T}_k, \boldsymbol{v}|_T \in \mathbb{P}_0^2 \right\}, \tag{3.4}$$

and the pressure p is approximated in the following space:

$$M_k = Q_k \cap L_0^2(\Omega), \tag{3.5}$$

where $\mathbb{P}_m$ denotes the space of polynomials of degree m, and Q_k is the linear finite element space

$$Q_k = \left\{ q \in C^0(\bar{\Omega}) : \forall T \in \mathcal{T}_k, q|_T \in \mathbb{P}_1 \right\}.$$

With these spaces, we can have the k-th level discrete formulation of the problem (3.1)–(3.2):

$$\begin{aligned} &\frac{\mu}{\rho} \int_\Omega \left(\boldsymbol{K}^{-1} \boldsymbol{u}_k \right) \cdot \boldsymbol{\varphi}_k \, \mathrm{d}\boldsymbol{x} + \frac{\beta}{\rho} \int_\Omega |\boldsymbol{u}_k| \left(\boldsymbol{u}_k \cdot \boldsymbol{\varphi}_k \right) \mathrm{d}\boldsymbol{x} \\ &\quad + \sum_{T \in \mathcal{T}_k} \int_T \nabla p_k \cdot \boldsymbol{\varphi}_k \, \mathrm{d}\boldsymbol{x} = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{\varphi}_k \, \mathrm{d}\boldsymbol{x}, \quad \forall \boldsymbol{\varphi}_k \in X_k, \end{aligned} \tag{3.6}$$

$$\sum_{T \in \mathcal{T}_k} \int_T \nabla q_k \cdot \boldsymbol{u}_k \, \mathrm{d}\boldsymbol{x} = -\int_\Omega g q_k \, \mathrm{d}\boldsymbol{x} + \int_{\partial\Omega} g_N q_k \, \mathrm{d}\boldsymbol{x}, \quad \forall q_k \in M_k. \tag{3.7}$$

By our construction,

$$h_{k-1} = 2h_k, \quad \text{for } k = 2, 3, \ldots.$$

Note that $\mathcal{T}_k$ are nested meshes, and thus

$$X_{k-1} \subset X_k, \quad M_{k-1} \subset M_k.$$

In [27], the authors demonstrated that the discrete problem (3.6)–(3.7) has a unique solution. Moreover, if $\mathcal{T}_h$ is shape regular with mesh size h and the solution $\boldsymbol{u}$ belongs to $W^{1,4}(\Omega)$ and p belongs to $W^{2,\frac{3}{2}}(\Omega)$, then the following error estimations are obtained in [27, Theorem 4.10]:

$$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L^2(\Omega)} \le Ch|\boldsymbol{u}|_{W^{1,4}(\Omega)}, \tag{3.8}$$

$$\|\nabla(p-p_h)\|_{L^{\frac{3}{2}}(T)} \le Ch\left(|p|_{W^{2,\frac{3}{2}}(\Omega)} + \|\boldsymbol{u}\|_{W^{1,4}(\Omega)}\right). \tag{3.9}$$

4 A Nonlinear Iteration

In this section, we present the Peaceman–Rachford (PR) iterative method developed in [15] to decouple the nonlinearity and the constraint.

First, choose an initial guess $\left(\boldsymbol{u}_k^0, p_k^0\right)$ by solving a linear Darcy system:

$$\frac{\mu}{\rho}\int_\Omega \left(\boldsymbol{K}^{-1}\boldsymbol{u}_k^0\right)\cdot\boldsymbol{\varphi}_k \,\mathrm{d}\boldsymbol{x} + \sum_{T\in\mathcal{T}_k}\int_T \nabla p_k^0\cdot\boldsymbol{\varphi}_k \,\mathrm{d}\boldsymbol{x} = \int_\Omega \boldsymbol{f}\cdot\boldsymbol{\varphi}_k \,\mathrm{d}\boldsymbol{x},\ \forall\boldsymbol{\varphi}_k\in X_k, \tag{4.1}$$

$$\sum_{T\in\mathcal{T}_k}\int_T \nabla q_k\cdot\boldsymbol{u}_k^0 \,\mathrm{d}\boldsymbol{x} = -\int_\Omega g q_k \,\mathrm{d}\boldsymbol{x} + \int_{\partial\Omega} g_N q_k \,\mathrm{d}\boldsymbol{x},\ \forall q_k\in M_k. \tag{4.2}$$

The linear Darcy system (4.1)–(4.2) can be rewritten in the matrix form as

$$\begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{u} \\ p \end{bmatrix} = \begin{bmatrix} \boldsymbol{f_d} \\ w \end{bmatrix}, \tag{4.3}$$

where A is the symmetric and positive definite matrix associated to the term

$$\frac{\mu}{\rho}\int_\Omega \left(\boldsymbol{K}^{-1}\boldsymbol{u}_k\right)\cdot\boldsymbol{\varphi}_k \,\mathrm{d}\boldsymbol{x},$$

B is the matrix corresponding to

$$\sum_{T\in\mathcal{T}_k}\int_T \nabla p_k\cdot\boldsymbol{\varphi}_k \,\mathrm{d}\boldsymbol{x},$$

and $\boldsymbol{f_d}$ and w represent the right hand side of (4.1) and (4.2), respectively.

Then, knowing $\left(\boldsymbol{u}_k^0, p_k^0\right)$, construct a sequence $\left(\boldsymbol{u}_k^{n+1}, p_k^{n+1}\right)$ for $n \ge 0$ in two steps. Let α be a positive parameter chosen to enhance the convergence.

1. A nonlinear step without constraint: knowing $\left(\boldsymbol{u}_k^n, p_k^n\right)$ compute the intermediate velocity $\boldsymbol{u}_k^{n+\frac{1}{2}}$ by solving the following equation:

$$\begin{aligned}\frac{1}{\alpha}\int_\Omega \left(\boldsymbol{u}_k^{n+\frac{1}{2}} - \boldsymbol{u}_k^n\right)\cdot\boldsymbol{\varphi}_k \,\mathrm{d}\boldsymbol{x} + \frac{\beta}{\rho}\int_\Omega \left|\boldsymbol{u}_k^{n+\frac{1}{2}}\right|\left(\boldsymbol{u}_k^{n+\frac{1}{2}}\cdot\boldsymbol{\varphi}_k\right)\mathrm{d}\boldsymbol{x} = \int_\Omega \boldsymbol{f}\cdot\boldsymbol{\varphi}_k \,\mathrm{d}\boldsymbol{x} \\ -\frac{\mu}{\rho}\int_\Omega \left(\boldsymbol{K}^{-1}\boldsymbol{u}_k^n\right)\cdot\boldsymbol{\varphi}_k \,\mathrm{d}\boldsymbol{x} - \sum_{T\in\mathcal{T}_k}\int_T \nabla p_k^n\cdot\boldsymbol{\varphi}_k \,\mathrm{d}\boldsymbol{x},\quad \forall\boldsymbol{\varphi}_k\in X_k.\end{aligned} \tag{4.4}$$

2. A linear step with constraint: compute $\left(\boldsymbol{u}_k^{n+1}, p_k^{n+1}\right)$ with the known $\boldsymbol{u}_k^{n+\frac{1}{2}}$

$$\frac{1}{\alpha}\int_\Omega\left(\boldsymbol{u}_k^{n+1}-\boldsymbol{u}_k^{n+\frac{1}{2}}\right)\cdot\boldsymbol{\varphi}_k\,\mathrm{d}\boldsymbol{x}+\frac{\mu}{\rho}\int_\Omega\left(\boldsymbol{K}^{-1}\boldsymbol{u}_k^{n+1}\right)\cdot\boldsymbol{\varphi}_k\,\mathrm{d}\boldsymbol{x}+\sum_{T\in\mathcal{T}_k}\int_T\nabla p_k^{n+1}\cdot\boldsymbol{\varphi}_k\,\mathrm{d}\boldsymbol{x}$$
$$=\int_\Omega\boldsymbol{f}\cdot\boldsymbol{\varphi}_k\,\mathrm{d}\boldsymbol{x}-\frac{\beta}{\rho}\int_\Omega\left|\boldsymbol{u}_k^{n+\frac{1}{2}}\right|\left(\boldsymbol{u}_k^{n+\frac{1}{2}}\cdot\boldsymbol{\varphi}_k\right)\mathrm{d}\boldsymbol{x},\quad\forall\boldsymbol{\varphi}_k\in X_k,\tag{4.5}$$
$$\sum_{T\in\mathcal{T}_k}\int_T\nabla q_k\cdot\boldsymbol{u}_k^{n+1}\,\mathrm{d}\boldsymbol{x}=-\int_\Omega gq_k\,\mathrm{d}\boldsymbol{x}+\int_{\partial\Omega}g_Nq_k\,\mathrm{d}\boldsymbol{x},\quad\forall q_k\in M_k.\tag{4.6}$$

A key observation in [15] is that because the test functions $\boldsymbol{\varphi}_k$, the solution $\boldsymbol{u}_k^{n+\frac{1}{2}}$, and ∇p_k^n are constant in each element T, the nonlinear step (4.4) can be solved in a closed-form:

$$\boldsymbol{u}_T^{n+\frac{1}{2}}=\frac{1}{\gamma}\boldsymbol{F}_T^{n+\frac{1}{2}}\tag{4.7}$$

where

$$\boldsymbol{F}_T^{n+\frac{1}{2}}=\frac{1}{\alpha}\boldsymbol{u}_T^n-\frac{\mu}{\rho}\boldsymbol{K}_T^{-1}\boldsymbol{u}_T^n-\nabla_T p_k^n+\boldsymbol{f}_T,$$
$$\boldsymbol{K}_T^{-1}=\frac{1}{|T|}\int_T\boldsymbol{K}^{-1}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x},$$
$$\gamma=\frac{1}{2\alpha}+\frac{1}{2}\sqrt{\frac{1}{\alpha^2}+4\frac{\beta}{\rho}\left|\boldsymbol{F}_T^{n+\frac{1}{2}}\right|}.$$

In the second step, the linear system (4.5)-(4.6) can be rewritten in the following matrix form:

$$\begin{bmatrix}A_\alpha & B\\ B^T & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{u}\\ p\end{bmatrix}=\begin{bmatrix}\boldsymbol{f}_{\boldsymbol{n+\frac{1}{2}}}\\ w\end{bmatrix},\tag{4.8}$$

where A_α is the matrix corresponding to the bilinear form

$$\frac{1}{\alpha}\int_\Omega\left(\boldsymbol{u}_k^{n+1}\right)\cdot\boldsymbol{\varphi}_k\,\mathrm{d}\boldsymbol{x}+\frac{\mu}{\rho}\int_\Omega\left(\boldsymbol{K}^{-1}\boldsymbol{u}_k^{n+1}\right)\cdot\boldsymbol{\varphi}_k\,\mathrm{d}\boldsymbol{x},$$

and $\boldsymbol{f}_{\boldsymbol{n+\frac{1}{2}}}$ is the vector corresponding to

$$\int_\Omega\boldsymbol{f}\cdot\boldsymbol{\varphi}_k\,\mathrm{d}\boldsymbol{x}+\frac{1}{\alpha}\int_\Omega\left(\boldsymbol{u}_k^{n+\frac{1}{2}}\right)\cdot\boldsymbol{\varphi}_k\,\mathrm{d}\boldsymbol{x}-\frac{\beta}{\rho}\int_\Omega\left|\boldsymbol{u}_k^{n+\frac{1}{2}}\right|\left(\boldsymbol{u}_k^{n+\frac{1}{2}}\cdot\boldsymbol{\varphi}_k\right)\mathrm{d}\boldsymbol{x}.$$

In [15], the authors proved that (4.1)–(4.2) and (4.5)–(4.6) have a unique solution. The PR iterative method is convergent for an arbitrary choice of the initial guess $\left(\boldsymbol{u}_k^0, p_k^0\right)$ and an arbitrary positive α. Numerically, different choices of α will affect the convergence rate of the nonlinear iteration. We shall report a choice of α in Sect. 6.

We can reduce the linear saddle point system into a SPD system when we implement the PR iteration. Because of A and A_α are symmetric positive definite operators, without loss of generality, we take (4.8) as an example to expound an idea as follows.

Eliminate $\boldsymbol{u}$ from the first equation of (4.8), i.e.

$$\boldsymbol{u}=A_\alpha^{-1}\left(\boldsymbol{f}_{\boldsymbol{n+\frac{1}{2}}}-Bp\right),\tag{4.9}$$

and then, substituting to the second equation of (4.8), we get

$$Sp = b, \tag{4.10}$$

where $S = B^T A_\alpha^{-1} B$, and $b = B^T A_\alpha^{-1} \boldsymbol{f}_{\boldsymbol{n}+\frac{1}{2}} - w$. After solving (4.10), we can get $\boldsymbol{u}$ by solving (4.9).

Since A_α is block-diagonal, A_α^{-1} can be formed easily. Indeed Eq. (4.10) is the linear finite element discretization of an elliptic equation in the primary formulation. The equivalence between (4.9)–(4.10) and (4.8) is obvious. Solving the SPD system (4.10) is much easier than the saddle point system (4.8) and many fast solvers are available. In our numerical experiments, we use the direct solver built in MATLAB© to solve (4.10). We could also use the multigrid solver, but due to the relative-small size of the linear SPD system we have tested, the direct solver is faster.

In the continuous level, the Darcy–Forchheimer equation can be rewritten into a nonlinear primary formulation. For simplicity, we assume that the permeability is a scalar. Taking the norm of Eq. (2.1), we obtain

$$\frac{\beta}{\rho}|\boldsymbol{u}|^2 + \frac{\mu}{\rho K}|\boldsymbol{u}| - |\nabla p - \boldsymbol{f}| = 0,$$

and can solve for $|\boldsymbol{u}|$

$$|\boldsymbol{u}| = \frac{-\frac{\mu}{\rho K} + \sqrt{\left(\frac{\mu}{\rho K}\right)^2 + 4\frac{\beta}{\rho}|\nabla p - \boldsymbol{f}|}}{2\frac{\beta}{\rho}}.$$

and consequently $\boldsymbol{u}$

$$\boldsymbol{u} = -\frac{\nabla p - \boldsymbol{f}}{\frac{\mu}{\rho K} + \frac{\beta}{\rho}|\boldsymbol{u}|} = -\frac{2(\nabla p - \boldsymbol{f})}{\frac{\mu}{\rho K} + \sqrt{\left(\frac{\mu}{\rho K}\right)^2 + 4\frac{\beta}{\rho}|\nabla p - \boldsymbol{f}|}}.$$

Then substituting back to (2.2), we get the primary formulation of pressure p only

$$-\nabla \cdot \left(\frac{2(\nabla p - \boldsymbol{f})}{\frac{\mu}{\rho K} + \sqrt{\left(\frac{\mu}{\rho K}\right)^2 + 4\frac{\beta}{\rho}|\nabla p - \boldsymbol{f}|}} \right) = g. \tag{4.11}$$

Its well-posedness can be found in [20].

In the discretization level, we could also eliminate the piecewise constant velocity and obtain an equivalent P_1 discretization of (4.11). However, we only eliminate $\boldsymbol{u}$ of the linear system (4.8) in the PR iteration rather than that of the nonlinear equation (3.6) because we still need to solve the resulting nonlinear equation. The PR iteration corresponds to a variant of Picard iteration for solving (4.11). We stick to the mixed formulation as the convergence of the PR iteration has been rigorously proved in [15].

5 A Non-linear Multigrid Algorithm

In this section, we consider a generic system of nonlinear equations,

$$\mathcal{L}(z) = \boldsymbol{s}$$

where $\boldsymbol{z}, \boldsymbol{s} \in \boldsymbol{R}^n$. Suppose that $\boldsymbol{v}$ is an approximation to the exact solution $\boldsymbol{z}$. Define the error $\boldsymbol{e}$ and the residual $\boldsymbol{r}$:

$$\begin{aligned} \boldsymbol{e} &= \boldsymbol{z} - \boldsymbol{v}, \\ \boldsymbol{r} &= \boldsymbol{s} - \mathcal{L}(\boldsymbol{v}). \end{aligned}$$

Quantities in the k-th level will be denoted by a subscript k.

Because of the iterative nature, multigrid ideas should be effective on the nonlinear problem. The multigrid scheme here we used for this nonlinear problem is the most commonly used nonlinear version of multigrid. It is called the *full approximation scheme* (FAS) [9] because the problem in the coarse grid is solved for the full approximation $\boldsymbol{z}_{k-1} = I_k^{k-1}\boldsymbol{v}_k + \boldsymbol{e}_{k-1}$ rather than the error $\boldsymbol{e}_{k-1}$. A two-level FAS is described as follows.

Full Approximation Scheme (FAS)

1. Pre-smoothing: For $1 \le j \le m$, relax m times with an initial guess $\boldsymbol{v}^0$ by $\boldsymbol{v}^j = R_k\boldsymbol{v}^{j-1}$. The current approximation $\boldsymbol{v}_k = \boldsymbol{v}^m$.
2. Restrict the current approximation and its fine grid residual to the coarse grid: $\boldsymbol{r}_{k-1} = I_k^{k-1}(\boldsymbol{s}_k - \mathcal{L}_k(\boldsymbol{v}_k))$ and $\boldsymbol{v}_{k-1} = I_k^{k-1}\boldsymbol{v}_k$.
3. Solve the coarse grid problem: $\mathcal{L}_{k-1}(\boldsymbol{z}_{k-1}) = \mathcal{L}_{k-1}(\boldsymbol{v}_{k-1}) + \boldsymbol{r}_{k-1}$.
4. Compute the coarse grid approximation to the error: $\boldsymbol{e}_{k-1} = \boldsymbol{z}_{k-1} - \boldsymbol{v}_{k-1}$.
5. Interpolate the error approximation up to the fine grid and correct the current fine grid approximation: $\boldsymbol{v}_{m+1} \leftarrow \boldsymbol{v}_k + I_{k-1}^k\boldsymbol{e}_{k-1}$.
6. Post-smoothing: For $m + 2 \le j \le 2m + 1$, relax m times by $\boldsymbol{v}^j = R_k{}'\boldsymbol{v}^{j-1}$.

then we get the approximate solution $\boldsymbol{v}^{2m+1}$. Here m denotes the number of pre-smoothing and post-smoothing steps, R_k denotes the chosen relaxation method, and I_k^{k-1} is an intergrid transfer operator from the fine grid to the coarse grid. As usual, the V-cycle will be obtained by applying the two-level FAS to solve the coarse grid equation in Step 3.

We choose the PR iteration (4.4)–(4.6) as the smoother R_k and the nonlinear solver in the coarsest grid. We switch the ordering of the linear and nonlinear steps of the PR iteration in the post-smoothing step in order to keep the symmetry of the V-cycle. It is worth pointing out that although the chosen finite element spaces are nested, the constrained subspaces are non-nested when we interpolated the correction of the velocity, which was obtained in the coarser space, to the finer space. Namely, if we directly interpolated the correction obtained on the coarser grid to the finer grid, the approximation we got may not satisfy the divergence equation in this Darcy–Forchheimer model. Therefore we construct a weighted L^2 projection to map the correction obtained before into the constrained space in the fine grid which can be realized by solving a saddle point system:

$$\begin{bmatrix} A_{\boldsymbol{\delta}} & B \\ B^T & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{\delta} \\ \theta \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ B^T\boldsymbol{e}_{\boldsymbol{u}} \end{bmatrix}, \tag{5.1}$$

where $A_{\boldsymbol{\delta}}$ is the matrix corresponding to

$$\frac{\mu}{\rho}\int_\Omega \left(\boldsymbol{K}^{-1}\boldsymbol{\delta}\right)\cdot\boldsymbol{\varphi}_k \,\mathrm{d}\boldsymbol{x} + \frac{\beta}{\rho}\int_\Omega |\boldsymbol{\delta}|\left(\boldsymbol{\delta}\cdot\boldsymbol{\varphi}_k\right)\mathrm{d}\boldsymbol{x},$$

$\boldsymbol{\delta}, \theta$ represent the error between the restriction of the approximation of velocity and pressure on the finer grid and their approximation obtained on the coarser grid, respectively, and $\boldsymbol{e}_{\boldsymbol{u}}$ is the prolonged correction to the fine space. For non-nested constrained subspaces, an additional projector is usually needed to preserve the constraint [7].

Again, (5.1) can be reduced to a SPD system. We can get $\boldsymbol{\delta} = A_{\boldsymbol{\delta}}^{-1} B\theta$ through the idea demonstrated in Sect. 4. Then we obtain a corrected approximation of velocity $\boldsymbol{v} = \boldsymbol{v} - \boldsymbol{\delta}$, which satisfies the divergence equation.

Remark 1 When RT or BDM element is used to discretize the velocity and the pressure is piecewise constant, we may use patch-wise smoothers designed for $H(\text{div})$ problems; see [2, 3]. The constraint can be preserved in these smoothers. A rigorous proof for the convergence of a multigrid method using constrained smoothers for linear saddle point systems can be found in [11,12]. Note that in this paper, we consider continuous pressure discretization and nonlinear saddle point systems and thus neither the constrained smoother nor the convergence proof can be applied. □

A convergence proof of a variant of FAS for a class of monotone nonlinear elliptic problems is given by Hackbusch in [16] and Reusken in [22]. They proved convergence by linearising the FAS iteration and used the convergence theory for linear two-grid methods for symmetric elliptic problems as in [6]. Their proof was rigorous but requiring restrictive assumptions (the initial guess is close enough to the solution). Tai and Xu in [28,29] gave some uniform convergence estimates for a class of subspace correction methods applied to some nonlinear unconstrained and constraint convex optimization problems. But their methods is built upon nested finite element spaces and slightly expensive than FAS. Yavneh and Dardyk in [32] employed a simplified scalar analogy to provide an insight to the reason why FAS works but a rigorous proof is lacking. None of these theoretical work can be applied directly to our problem. We are investigating the convergence theory of FAS in different perspectives and will report our finding somewhere else.

6 Numerical Experiments

In this section, some numerical results are presented to illustrate the efficiency of our multigrid method for the Darcy–Forchheimer model (2.1)–(2.3). The following test problems are taken from [17]. All of our experiments are implemented based on the MATLAB© software package *i*FEM [10]. They were run on a laptop with an Inter i7-4720HQ 2.60 GHz CPU and 16.0 GB RAM.

We choose $\mu = 1$, $\rho = 1$, $\boldsymbol{K} = \boldsymbol{I}$, and $\Omega \subset \mathbb{R}^2$ as the square $(-1, 1)^2$. We use the uniform triangulation of Ω.

- Problem 1:

$$
\begin{aligned}
\boldsymbol{u}(x, y) &= [x + y, x - y]^T, \\
p(x, y) &= x^3 + y^3, \\
\boldsymbol{f}(x, y) &= \begin{bmatrix} \left(1 + \beta\sqrt{2x^2 + 2y^2}\right)(x + y) + 3x^2 \\ \left(1 + \beta\sqrt{2x^2 + 2y^2}\right)(x - y) + 3y^2 \end{bmatrix}, \\
g_N(x, y) &= \begin{cases} 1 + y, & x = 1, \\ 1 - y, & x = -1, \\ x - 1, & y = 1, \\ -x - 1, & y = -1. \end{cases}
\end{aligned}
$$

- Problem 2:

$$
\begin{aligned}
\boldsymbol{u}(x,y) &= \left[\frac{(x+1)^2}{4}, -\frac{(x+1)(y+1)}{2}\right]^T,\\
p(x,y) &= x^3+y^3,\\
\boldsymbol{f}(x,y) &= \begin{bmatrix} \frac{(x+1)^2}{4}\left(1+\beta\frac{(x+1)}{4}\sqrt{(x+1)^2+4(y+1)^2}\right)+3x^2\\ -\frac{(x+1)(y+1)}{2}\left(1+\beta\frac{(x+1)}{4}\sqrt{(x+1)^2+4(y+1)^2}\right)+3y^2 \end{bmatrix},\\
g_N(x,y) &= \begin{cases} 1, & x=1,\\ 0, & x=-1,\\ -x-1, & y=1,\\ 0, & y=-1. \end{cases}
\end{aligned}
$$

Numerically Problem 2 is harder to solve. Probably it is due to the fact that the initial guess, which is obtained by solving a linear Darcy system, is further away from the true solution.

For all above test problems, $g=0$. The chosen termination criterion is

$$r = r_u + r_p \le tol,$$

where

$$
\begin{aligned}
r_u &= \begin{cases} \left\| \boldsymbol{f} - \frac{\mu}{\rho}\boldsymbol{K}^{-1}\boldsymbol{u}_h^n + \frac{\beta}{\rho}\left|\boldsymbol{u}_h^n\right|\boldsymbol{u}_h^n + \nabla p_h^n \right\| / \|\boldsymbol{f}\|, & \text{when } \|\boldsymbol{f}\| \ne 0,\\ \left\| \boldsymbol{f} - \frac{\mu}{\rho}\boldsymbol{K}^{-1}\boldsymbol{u}_h^n + \frac{\beta}{\rho}\left|\boldsymbol{u}_h^n\right|\boldsymbol{u}_h^n + \nabla p_h^n \right\|, & \text{when } \|\boldsymbol{f}\| = 0. \end{cases}\\
r_p &= \begin{cases} \left\|g - \mathrm{div}\boldsymbol{u}_h^n\right\| / \|g\|, & \text{when } \|g\| \ne 0,\\ \left\|g - \mathrm{div}\boldsymbol{u}_h^n\right\|, & \text{when } \|g\| = 0. \end{cases}
\end{aligned}
$$

We first use the accuracy test to confirm that our nonlinear multigrid iteration will convergent to an approximation of the problem of consideration. In the following experiments, the letter N stands for 'Number of unknowns of p', which is the same as 'Numbers of vertices', so $h = \frac{2}{\sqrt{N}-1}$, which represents the mesh size in one direction. Numerical results, see Figs. 1a and 2a, confirmed the convergence order for $\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L^2}$ and $\|p-p_h\|_{H^1}$ are $O(h) = O(N^{1/2})$. The accuracy of the pressure approximations, however, is not as good as that of velocity. Meanwhile, in consideration of the computation cost, the sufficiently accurate results were achieved when $tol = 10^{-6}$ for Problems 1 and 2. The stopping tolerance can be varying in different levels to further reduce the cost. A guide line is below the truncation error [8]. The authors in [17], however, use $tol = 1.95h$, which is only enough for the L^2-norm approximation for velocity. We shall use $tol = 10^{-6}$ in the remaining numerical experiments.

For all tests, the iteration steps and CPU time of each solver are listed in tables. We are aware that the CPU time depends on the implementation and testing environment: the programming language, optimization of codes, and the hardware (memory and cache), etc. Our code has been optimized using vectorization technique and all results were measured and compared in the same test environment so that the CPU time could be a good indicator of the efficiency. The CPU time will be also used to find the asymptotic time complexity of each method; see Figs. 1b and 2b.

As it has been proved in [15], the PR nonlinear iteration converges for any $\alpha > 0$. Its rate of convergence, however, is very sensitive to the choice of this parameter. From the

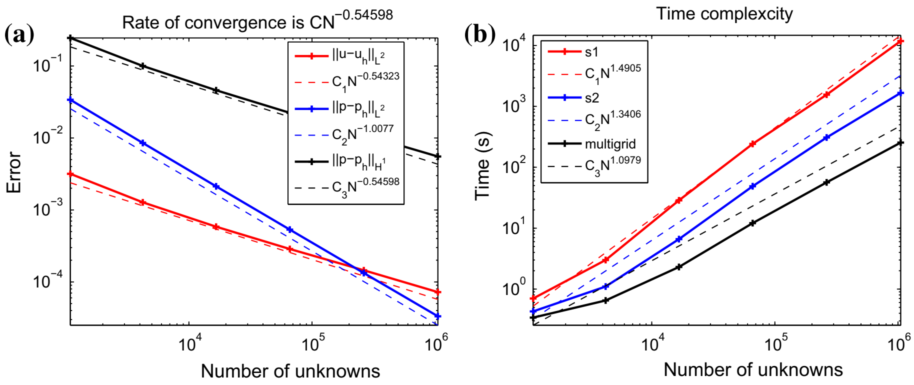

Fig. 1 Convergence rate by using multigrid solver and time complexity by using different solvers for Problem 1 with $\beta = 30$. **a** Convergence rate by using multigrid solver, **b** time complexity

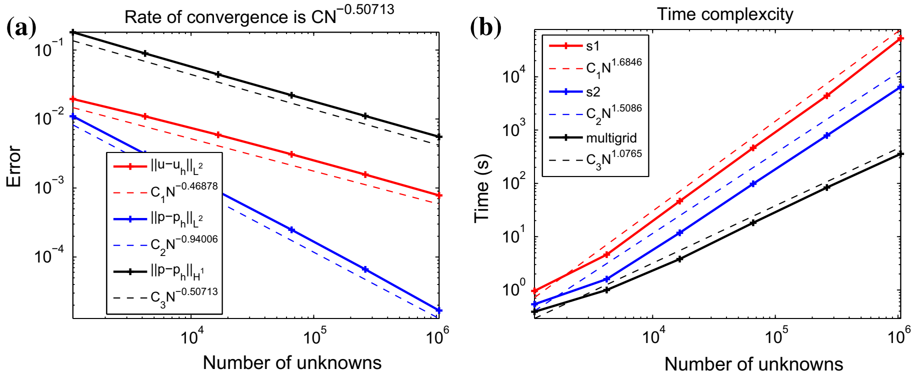

Fig. 2 Convergence rate by using multigrid solver and time complexity by using different solvers for Problem 2 with $\beta = 30$. **a** Convergence rate by using multigrid solver, **b** time complexity

Table 1 Comparison of different values of α in PR iteration with $h = \frac{1}{64}$ for $\beta = 10, 20, 30$

Problem		$\beta = 10$		$\beta = 20$		$\beta = 30$	
		$\alpha = 1$	$\alpha = 1/10$	$\alpha = 1$	$\alpha = 1/20$	$\alpha = 1$	$\alpha = 1/30$
Problem 1	Iter	229	73	457	105	686	120
	CPU time	14 s	4 s	26 s	6 s	38 s	7 s
Problem 2	Iter	230	171	459	183	688	191
	CPU time	13 s	10 s	26 s	11 s	38 s	11 s

convergence proof of the PR iteration in [15], we inferred that the choices of α depends on the Forchheimer number β which controls the magnitude of the nonlinearity as ρ is fixed. We give an empirical choice of parameter $\alpha = 1/\beta$ and compared with the choice $\alpha = 1$ suggested in [17] in Tables 1 and 2. As shown in Tables 1 and 2, the choice of the parameter $\alpha = 1/\beta$ is much better than the fixed selection for different values of β. Therefore, this choice of α will be used in the remaining numerical experiments.

Table 2 Comparison of different values of α in PR iteration with $h = \frac{1}{64}$ for $\beta = 40, 50, 60$

Problem		$\beta = 40$		$\beta = 50$		$\beta = 60$	
		$\alpha = 1$	$\alpha = 1/40$	$\alpha = 1$	$\alpha = 1/50$	$\alpha = 1$	$\alpha = 1/60$
Problem 1	Iter	914	126	1143	129	1371	131
	CPU time	53 s	7 s	66 s	7 s	79 s	8 s
Problem 2	Iter	917	198	1146	205	1376	213
	CPU time	52 s	11 s	65 s	11 s	79 s	12 s

Table 3 Comparison of number of iterations and CPU time of Problem 1 by using different solvers with $\beta = 30$

h	DoFs	I (pr)	I (mg)	CPU (s1)	CPU (s2)	CPU (mg)
$\frac{1}{16}$	5185	50	1	0.70 s	0.43 s	0.34 s
$\frac{1}{32}$	20,609	81	6	3.0 s	1.1 s	0.65 s
$\frac{1}{64}$	83,177	120	6	28.6 s	6.6 s	2.3 s
$\frac{1}{128}$	328,193	154	6	242.3 s	48.8 s	12.1 s
$\frac{1}{256}$	1,311,745	168	6	1554.7 s	308.3 s	56.5 s
$\frac{1}{512}$	5,244,929	185	5	11,857.3 s	1667.7 s	254.6 s

We then compare the FAS multigrid method using PR as smoother with the PR iterative method for solving this nonlinear system. Here we choose $m = 3$ for all the following tests. It means that we apply three PR iterations in the pre-smoothing step and post-smoothing step, respectively. Each V-cycle step is approximately 9 PR iterations (6 for the finest level and 3 for iterations in all coarser levels as the size of the system is reduced by 1/4) in terms of complexity. In order to keep the symmetry of the V-cycle, we switch the ordering of the linear and nonlinear steps of the PR iteration in the post-smoothing step. We set $h = 1/16$ as the coarsest mesh and solve the nonlinear problem in the coarsest mesh using PR iteration.

The PR solver is denoted by pr, whereas the multigrid solver is denoted by mg. I - number of iterations, and CPU - CPU time. ‘s1’ represents that we solve these linear saddle point systems (4.8) directly in each step, ‘s2’ is that we solve the primal SPD system (4.10) mentioned in Sect. 4 rather than solving the saddle point system. ‘mg’ stands for our multigrid solver, in which the PR iteration is constructed based on ‘s2’. In all examples we achieve optimal order convergence of $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L^2}$ and $\|p - p_h\|_{H^1}$. Compared with the PR iteration, we can obtain the same accuracy by using our multigrid method with less iterations. We can get similar results for different values of the Forchheimer number β.

Since our focus is on the efficiency of solvers, we mainly report the comparison of the number of iterations and CPU time by using different solvers. Numerical tests were performed for several cases of different values of the Forchheimer number β for Problems 1 and 2, and the behavior of these experiments is similar for all chosen cases. All problems are becoming harder to solve as the Forchheimer number β increases, mainly because β enhances the nonlinearity. Therefore, without loss of critical substance and clarity, here we only show the results for $\beta = 30$ to demonstrate the merits of our method.

Table 4 Comparison of iteration steps of multigrid solver according to different h and β for Problem 1 with $\alpha = 1/\beta$

h	$\beta = 10$	$\beta = 20$	$\beta = 30$	$\beta = 40$	$\beta = 50$
$\frac{1}{32}$	4	6	6	7	7
$\frac{1}{64}$	4	6	6	7	7
$\frac{1}{128}$	4	5	6	6	7
$\frac{1}{256}$	4	5	6	6	6
$\frac{1}{512}$	3	5	5	6	6

Table 5 Comparison of number of iterations and CPU time of Problem 2 by using different solvers with $\beta = 30$

h	DoFs	I (pr)	I (mg)	CPU (s1)	CPU (s2)	CPU (mg)
$\frac{1}{16}$	5185	92	1	0.96 s	0.54 s	0.39 s
$\frac{1}{32}$	20,609	128	9	4.6 s	1.6 s	1.0 s
$\frac{1}{64}$	83,177	191	9	46.5 s	11.8 s	3.8 s
$\frac{1}{128}$	328,193	296	9	462.9 s	98.3 s	18.2 s
$\frac{1}{256}$	1,311,745	468	8	4412.9 s	792.6 s	83.6 s
$\frac{1}{512}$	5,244,929	746	7	>14 h	6440.3 s	357.2 s

Table 6 Comparison of iteration steps of multigrid solver according to different h and β for Problem 2 with $\alpha = 1/\beta$

h	$\beta = 10$	$\beta = 20$	$\beta = 30$	$\beta = 40$	$\beta = 50$
$\frac{1}{32}$	5	7	9	11	12
$\frac{1}{64}$	5	7	9	11	12
$\frac{1}{128}$	5	7	9	10	11
$\frac{1}{256}$	4	6	8	9	10
$\frac{1}{512}$	4	5	7	8	9

It can be observed that our multigrid solver required significantly fewer iterations and CPU time than the other two solvers in Tables 3 and 5. More importantly, iteration steps are uniformly stable with respect to h and the time complexity of our multigrid solver is nearly linear, i.e., $O(N)$, shown in Figs. 1 and 2. In contrast, for the PR methods, iteration steps increase as h decreases and the time complexity seems to be more than linear. For the largest size we have tested, our multigrid solver is more than 40 times faster than the original PR iteration. In Tables 4 and 6, the number of iterations are compared for different values of β and it is demonstrated that our multigrid method is also robust to both mesh size h and the Forchheimer number β while PR iteration is not, see Tables 1 and 2. It is worth noting that even for a linear Stokes type equation, construct a solver robust to a critical parameter is not easy [18,19].

7 Conclusions

In this paper, we constructed a nonlinear multigrid method for a mixed finite element method of the two-dimensional Darcy–Forchheimer model. We presented a comparative study between the multigrid solver and the PR iterative solver, at the same time compared CPU time of the efficient solver of solving the SPD systems with that obtained by solving the linear saddle point systems directly. We took into account the pressure accuracy when we set the termination criterion, and chose a better value of the stopping criterion tol. In comparison with the authors in [17] always chose $\alpha = 1$ for different values of the Forchheimer number β, we reported a better choice and compared with the previous choice through comparing the number of iterations and CPU time. The results obtained from our tests indicate that the multigrid solver is very efficient for numerically solving this nonlinear elliptic equation. The number of iterations and CPU time for using multigrid solver are shown to be significantly less than that obtained by using the PR iteration alone.

In the future work, we shall extend our results to three directions. One is that we would like to find a better smoother, which is used in the pre-smoothing and post-smoothing step, to further reduce CPU time and make the multigrid solver more efficient. Another is that we intend to carry out some studies on the three-dimensional Darcy–Forchheimer problem and the real application in a porous medium. We shall also investigate the theoretical study of the convergence proof of FAS.

Acknowledgements We would like to thank the anonymous referee for the valuable suggestions and careful reading, which have helped us to improve the presentation.

References

1. Adams, R.A.: Sobolev Spaces. Academic Press, New York (1975)
2. Arnold, D.N., Falk, R.S., Winther, R.: Preconditioning in $H(\mathrm{div})$ and applications. Math. Comp. **66**(219), 957–984 (1997)
3. Arnold, D.N., Falk, R.S., Winther, R.: Multigrid in $H(\mathrm{div})$ and $H(\mathrm{curl})$. Numer. Math. **85**(2), 197–217 (2000)
4. Aristotelous, A., Karakashian, O., Wise, S.: A mixed discontinuous Galerkin, convex splitting scheme for a modified Cahn–Hilliard equation and an efficient nonlinear multigrid solver. Discrete Contin. Dyn. Syst. Ser. B **18**(9), 2211–2238 (2013)
5. Aziz, K., Settari, A.: Petroleum Reservoir Simulation. Applied Science Publishers LTD, London (1979)
6. Bank, R.E., Douglas, C.C.: Sharp estimates for multigrid rates of convergence with general smoothing and acceleration. SIAM J. Numer. Anal. **22**, 617–633 (1985)
7. Braess, D., Sarazin, R.: An efficient smoother for the Stokes equation. Appl. Numer. Math. **23**(1), 3–20 (1997)
8. Brandt, A.: Multi-level adaptive solutions to boundary-value problems. Math. Comp. **31**(138), 333–390 (1977)
9. Briggs, W.L., Henson, V.E., McCormick, S.F.: A Multigrid Tutorial, 2nd edn. SIAM, Philadelphia (2000)
10. Chen, L.: iFEM: an integrated finite element methods package in MATLAB. Technical report, University of California at Irvine (2009)
11. Chen, L.: Multigrid methods for saddle point systems using constrained smoothers. Comput. Math. Appl. **70**(12), 2854–2866 (2015)
12. Chen, L.: Multigrid methods for constrained minimization problems and application to saddle point problems. arXiv:1601.04091, pp. 1–27 (2016)
13. Ciarlet, P.G.: The finite element method for elliptic problems. North-Holland, Amsterdam (1978)
14. Forchheimer, P.: Wasserbewegung durch Boden. Z. Ver. Deutsch. Ing. **45**, 1782–1788 (1901)
15. Girault, V., Wheeler, M.F.: Numerical discretization of a Darcy–Forchheimer model. Numer. Math. **110**, 161–198 (2008)
16. Hackbusch, W.: Multigrid Methods and Applications. Springer, Heidelberg (1985)
17. López, H., Molina, B., Salas, J.J.: Comparison between different numerical discretizations for a Darcy–Forchheimer model. ETNA **34**, 187–203 (2009)
18. Mardal, K., Winther, R.: Uniform preconditioners for the time dependent Stokes problem. Numer. Math. **98**(2), 305–327 (2004)

19. Olshanskii, M., Peters, J., Reusken, A.: Uniform preconditioners for a parameter dependent saddle point problem with application to generalized Stokes interface equations. Numer. Math. **105**(1), 159–191 (2006)
20. Pan, H., Rui, H.: Mixed element method for two-dimensional Darcy–Forchheimer model. J. Sci. Comp. **52**, 563–587 (2012)
21. Park, E.J.: Mixed finite element method for generalized Forchheimer flow in porous media. Numer. Methods Partial Differ. Equ. **21**, 213–228 (2005)
22. Reusken, A.: Convergence of the multigrid full approximation scheme for a class of elliptic mildly nonlinear boundary value problems. Numer. Math. **52**, 251–277 (1988)
23. Rui, H., Liu, W.: A two-grid block-centered finite difference method for Darcy–Forchheimer flow in porous media. SIAM J. Numer. Anal. **53**(4), 1941–1962 (2015)
24. Rui, H., Pan, H.: A block-centered finite difference method for the Darcy–Forchheimer model. SIAM J. Numer. Anal. **50**(5), 2612–2631 (2012)
25. Rui, H., Zhao, D., Pan, H.: A block-centered finite difference method for Darcy–Forchheimer model with variable Forchheimer number. Numer. Methods Partial Differ. Equ. **31**, 1603–1622 (2015)
26. Ruth, D., Ma, H.: On the derivation of the Forchheimer equation by means of the averaging theorem. Transp Porous Media **7**, 255–264 (1992)
27. Salas, J.J., López, H., Molina, B.: An analysis of a mixed finite element method for a Darcy–Forchheimer model. Math. Comput. Model. **57**, 2325–2338 (2013)
28. Tai, X.: Rate of convergence for some constraint decomposition methods for nonlinear variational inequalities. Numer. Math. **93**(4), 755–786 (2003)
29. Tai, X., Xu, J.: Global and uniform convergence of subspace correction methods for some convex optimization problems. Math. Comput. **71**(237), 105–125 (2001)
30. Wang, Y., Rui, H.: Stabilized Crouzeix–Raviart element for Darcy–Forchheimer model. Numer. Methods Partial Differ. Equ. **31**, 1568–1588 (2015)
31. Wise, S.: Unconditionally stable finite difference, nonlinear multigrid simulation of the Cahn–Hilliard–Hele–Shaw system of equations. J. Sci. Comp. **44**(1), 38–68 (2010)
32. Yavneh, I., Dardyk, G.: A multilevel nonlinear method. SIAM J. Sci. Comp. **28**(1), 24–46 (2006)