Journal of Scientific Computing, Vol. 32, No. 2, August 2007 (© 2007)
DOI: 10.1007/s10915-007-9134-z

Preconditioned Descent Algorithms for p-Laplacian

Y. Q. Huang,[1,3] Ruo Li,[2] and Wenbin Liu[2]

Received July 29, 2006; accepted (in revised form) March 8, 2007; Published online May 21, 2007

In this paper, we examine some computational issues on finite element discretization of the p-Laplacian. We introduced a class of descent methods with multi-grid finite element preconditioners, and carried out convergence analysis. We showed that their convergence rate is mesh-independent. We studied the behavior of the algorithms with large p. Our numerical tests show that these algorithms are able to solve large scale p-Laplacian with very large p. The algorithms are then used to solve a variational inequality.

KEY WORDS: Highly degenerate p-Laplacian; finite element approximation; preconditioned steepest descent algorithms; variational inequalities.

SUBJECT CLASSIFICATION: 49J20; 65N30.

1. INTRODUCTION

In this paper, we investigate some computational issues on the finite element approximation of the p-Laplacian with Dirichlet data:

$$\begin{aligned} -\mathrm{div}(|\nabla u|^{p-2}\nabla u) = f, &\quad \text{in}\,\Omega, \\ u = 0, &\quad \text{on}\,\partial\Omega, \end{aligned} \tag{1.1}$$

where $1 < p < \infty$ and Ω is a bounded open subset of R^2 with a Lipschitz boundary $\partial\Omega$. This equation is viewed as one of the typical examples of a large class of non-linear problems – degenerate non-linear systems, see [10, 19, 20] for some examples. Indeed it is believed that this equation contains most of the essential difficulties in studies of finite element approximations

[1] Laboratory for Computation and Simulation in Science and Engineering, Xiangtan University, Xiangtan, Hunan, China. E-mail: huangyq@xtu.edu.cn
[2] KBS, and Institute of Mathematics and Statistics, University of Kent, Canterbury CT2 7NF, England
[3] To whom correspondence should be addressed. E-mail: huangyq@xtu.edu.cn

0885-7474/07/0800-0343/0 © 2007 Springer Science+Business Media, LLC

for this class of degenerate non-linear systems, where many existing techniques (such as the linearization or deformation procedure) in the finite element method do not seem to work well.

Finite element approximations of the p-Laplacian have been extensively studied in the literature, and one can find some previous work, for example, in [2,10,11,13], and some recent work in [1, 4–6, 12, 19–24, 25]. In particular, the quasi-norm approach has been developed in our work. This approach has proved quite successful in deriving sharp a priori and a posteriori error bounds for the finite element approximation of the degenerate systems. Some accounts of very recent work on the p-Laplacian can be found in the papers [18,19] as well.

Finite element discretization of the p-Laplacian (or related systems) results in highly nonlinear and degenerate algebraic systems (in the sense that either the Jacobi of A_h may not exist everywhere (for $p < 2$), or its inverse may not be differentiable (for $p > 2$)):

$$A_h(u_h) = f_h, \tag{1.2}$$

where u_h is the finite element approximation of the p-Laplacian. Or equivalently one can solve the following minimization problem:

$$\min_{v_h \in V^h} J_h(v_h), \tag{1.3}$$

where V^h is the correspondent finite element space and

$$J_h(v_h) = \int_{\Omega_h} |\nabla v_h|^p / p - \int_{\Omega_h} f v_h.$$

The Euler equation of (1.3) is just (1.2). In [12,13], Glowinski, etc. used the augmented Lagrangian method to solve (1.3). In [4], we used the nonlinear PR-Conjugate algorithm (PRNCG) to solve it, which proves to be quite efficient. The PRNCG is still widely used in solving (1.3), and has recently been found to work well as a smoother in multi-grid algorithms, see [7] for the details. The above numerical methods work well, provided $p + 1/(p-1)$ is not very large. In studying some physical and engineering relevant problems, it is needed to solve highly degenerate cases where $p + 1/(p-1)$ is large, see for example, [3,8], and Sect. 5 for the cases of large p.

However more efficient computational means are yet to be developed for large scale and highly degenerate cases. This is the main issue we wish to address in this work.

Multi-grid finite element method is one of the most powerful computational means for large linear systems arising from finite element discretization of elliptic equations. It has also been found to be very powerful in solving large scale non-linear algebraic systems arising from some non-linear elliptic equations. Numerical and theoretical studies of multi-grid finite element method for the p-Laplacian started only recently, see [7, 26, 27] for some initial work. Subspace correction algorithms have been proposed, in an abstract form, and studied in [26, 27] for the p-Laplacian. A full multi-grid method has been proposed in [7], and tested there for the case $p=4,6$. It used the algorithm (PRNCG) as the smoother. However from our experience, these multi-grid methods only work well when the non-linear systems are not very degenerate.

When $p+1/(p-1)$ is large, the optimization problem (1.3) is essentially difficult to solve. One possible remedy is to use a suitable preconditioner. In this work we introduce the following preconditioned descent algorithm:

$$u_{n+1}=u_n+\rho_n w_n,$$

where w_n is determined by solving

$$B_n(w_n,v)=-\left(\int_\Omega |\nabla u_n|^{p-2}\nabla u_n \nabla v-\int_\Omega f v\right),\quad \forall v\in V^h, \tag{1.4}$$

and ρ_n can be determined by line search:

$$\min_{\rho\geq 0} J_h(u_n+\rho w_n). \tag{1.5}$$

It is well known that descent iterative algorithms of this type could be very inefficient. It can be very slow, especially when the contour maps of the functional are very prolonged near the minimizers. Also its speed could be mesh-dependent, if not formulated in suitable norms. Thus it is necessary to formulate the algorithm in an energy type norm and to precondition the contours into a "good" shape locally in order to obtain higher computational efficiency. The idea is to choose $B_n(u_h,\cdot)$ to be "similar" to the p-Laplacian. Also one has to be able to compute w_n economically. Thus we choose B_n to be a simple linearization of the p-Laplacian to utilize the existing fast MG solvers. Our numerical experiments show that with a suitably chosen B_n, the above iterative method is efficient for a wide range of p. It can be used to solve large scale discretized p-Laplacain with $p=1000$. When $p\geq 2$, we established some mesh-independence convergence results for our algorithms. For the case where p is close to 1, there is an extra difficulty that the problem (1.3) is numerically unstable with respect to f, see Remark 3.1 for the details.

2. PRELIMINARY

Let $V = W_0^{1,p}(\Omega)$ with $p > 1$, and let

$$J(u) = \frac{1}{p}\int_\Omega |\nabla u|^p - \int_\Omega fu, \quad u \in W_0^{1,p}(\Omega). \tag{2.1}$$

Clearly the functional J is strictly convex for $1 < p < \infty$. Furthermore the following minimization problem has a unique solution, see [11]:

$$\min_{v \in V} J(v) \tag{2.2}$$

It is well-known that the above problem (2.2) is equivalent to the following non-linear PDE—the p-Laplacian: (WP) find $u \in V$ such that

$$a(u, v) = \int_\Omega |\nabla u|^{p-2}\nabla u \nabla v = \int_\Omega fv \quad \forall v \in V. \tag{2.3}$$

As mentioned in Introduction, the p-Laplacian has been extensively studied in the literature. Particularly, it is a matter of direct calculations to show that

$$J'(u)(v) = \int_\Omega |\nabla u|^{p-2}\nabla u \nabla v - \int_\Omega fv \tag{2.4}$$

$$J''(u)(v, w) = \int_\Omega |\nabla u|^{p-2}\nabla v \nabla w + (p-2)\int_\Omega |\nabla u|^{p-4}(\nabla u, \nabla v)(\nabla u, \nabla w) \tag{2.5}$$

For $p \geq 2$,

$$(J'(u) - J'(v))(u - v) \geq \|u - v\|_{1,p}^p \tag{2.6}$$

$$|J'(u)(v)| \leq C(\|u\|_{1,p}^{p-1} + \|f\|_{-1,\frac{p}{p-1}})\|v\|_{1,p} \tag{2.7}$$

$$|J''(u)(v, w)| \leqslant C\|u\|_{1,p}^{p-2}\|v\|_{1,p}\|w\|_{1,p} \tag{2.8}$$

The details can be found in [7]. We now introduce the finite element spaces, as in [11]. For sake of simplicity, we assume that Ω is a convex polygonal domain. Let T^h be a regular triangulation of Ω into disjoint open regular triangles K, so that $\bar{\Omega} = \bigcup_{K \in T^h} \bar{K}$. Each element has at most one edge on $\partial\Omega$, and $\bar{K}$ and $\bar{K}'$ have either only one common vertex, or a whole edge if K and $K' \in T^h$. Let h_K denote the diameter of the element K in T^h and let ρ_K denote the diameter of the largest ball contained in K. We assume that there is a regularity constant R of T^h, independent of h, such that $1 \le \max_{K \in T^h}(h_K/\rho_K) \le R$. Let $h = \max_{K \in T^h} h_K$. Furthermore, we assume that there is a $C^1(\overline{\Omega})$-function $h(x)$ such that $c' h_K \le h(x) \le h_K \text{in} K$ for all simplices $K \in T^h$ and some constant $c' > 0$ independent of K.

We shall only discuss the continuous piecewise linear element in this paper due to the limited higher order regularity for the solution of the p-Laplacian, see, for instance, [16–18], for the details. Associated with T^h is a finite dimensional subspace V^h of $C^0(\bar{\Omega})$, such that $\chi|_K \in \mathcal{P}_1$ for all $\chi \in V^h$ and $K \in T^h$, where $\mathcal{P}_1$ is the linear functions space. Let

$$V_0^h = \{\chi \in V^h : \chi(x^k) = 0, \quad \text{for all vertices } x^k \in \partial\Omega\}.$$

Then the finite element approximation of (WP) is as follows $(WP)^h$: Find $u_h \in V_0^h$ such that

$$a(u_h, v_h) = (f, v_h), \quad \forall v_h \in V_0^h, \tag{2.9}$$

where

$$a(u_h, v_h) = \int_\Omega |\nabla u_h|^{p-2} \nabla u_h \cdot \nabla v_h,$$

$$(f, v_h) = \int_\Omega f v_h.$$

It is a simple matter to show that $(WP)^h$ has a unique solution u_h. Also $(WP)^h$ is equivalent to the following minimization problem:

$$\min_{v \in V_0^h} J(v) \tag{2.10}$$

3. PRECONDITIONED DESCENT ALGORITHMS

3.1. Preconditioned Steepest Descent Method

In this subsection we formulate some preconditioned steepest descent algorithms for the p-Laplacian. Let $v_h, w_h \in V_0^h$. In the rest of the paper, we shall drop the subscription h if no confusion is caused. Let $\|\cdot\|$ be a norm on V_0^h. The (normalized) steepest descent direction $w \in V_0^h$ is defined such that

$$J'(v)w = -\|J'(v)\|_*, \; \|w\| = 1, \tag{3.1}$$

where$\|\cdot\|_*$ is the standard dual norm defined by

$$\|J'(v)\|_* = \sup_{u\in V_0^h} |J(v)(u)|/\|u\|.$$

Hence w actually depends on the norm on V_0^h. Unless using a L^2 type norm, it is needed to solve a linear PDE to compute w. However if we formulate a steepest descent method not using the energy norms (e.g., if using the L^2 norm), then convergence rate of the algorithm will probably be h dependent. To derive a steepest descent algorithm, whose performance is mesh independent and whose descent directions can be conveniently computed, we shall formulate our algorithms using norms of $H_0^1(\Omega)$ type.

Let u be the exact solution of (2.2) and $u_n \in V_0^h$ be the current approximation. We wish to find the next approximation u_{n+1}. Define u_{n+1} by

$$u_{n+1} = u_n + \alpha_n w_n, \tag{3.2}$$

where α_n is determined by line search

$$J(u_n + \alpha_n w_n) = \min_{\alpha\geq 0} J(u_n + \alpha w_n), \tag{3.3}$$

and $w_n \in V_0^h$ is defined by

$$\int_\Omega \nabla w_n \nabla v = -J'(u_n)(v) = -\int_\Omega |\nabla u_n|^{p-2}\nabla u_n \nabla v + \int_\Omega f v \quad \forall v \in V_0^h. \tag{3.4}$$

Let us show that w_n is in fact the steepest descent direction at $u = u_n$. Let $V = V_0^h$. By (2.4) one has

$$\begin{aligned} J'(u_n)(v) &= \int_\Omega |\nabla u_n|^{p-2}\nabla u_n \nabla v - \int_\Omega f v \\ &= \int_\Omega (|\nabla u_n|^{p-2}\nabla u_n - |\nabla u|^{p-2}\nabla u)\nabla v. \end{aligned}$$

Since w_n is the Riesz representation of the functional $-J'(u_n)$ in the space V, so

$$\|w_n\|_V = \|J'(u_n)\|_* \tag{3.5}$$

and

$$J'(u_n)(w_n) = -\|w_n\|^2_{V_0^h} = -\|J'(u_n)\|_* \|w_n\|_{V_0^h}. \tag{3.6}$$

Thus we formulate our first steepest descent algorithm as follows:

Algorithm 1. For a given initial value u_0 do the following iterations

- solving $\int_\Omega \nabla w_n \nabla v = -\int_\Omega |\nabla u_n|^{p-2} \nabla u_n \nabla v + \int_\Omega f v \quad \forall v \in V$
- linesearch for α_n, such that $J(u_n + \alpha_n w_n) = \min_{\alpha \geq 0} J(u_n + \alpha w_n)$
- update $u_{n+1} = u_n + \alpha_n w_n$

The direction w_n can be computed by a linear multi-grid solver.

We now consider a different preconditioner. One may equip V with a weighted norm $|\cdot|_{u_n}^2 = \int_\Omega (\epsilon + |\nabla u_n|^{p-2}) |\nabla \cdot|^2$ for $p > 2$. This leads to a bilinear form $((\epsilon + |\nabla u_n|^{p-2}) \nabla \cdot, \nabla \cdot)$, which is a simple linearization of the weak form (2.3) of the partial differential operator. The introduction of the small parameter ϵ is to handle the possible degeneracy when $\nabla u_n = 0$. Then we define the descent direction by: Find $w_n \in V$ such that

$$\int_\Omega (\epsilon + |\nabla u_n|^{p-2}) \nabla w_n \nabla v = -\int_\Omega |\nabla u_n|^{p-2} \nabla u_n \nabla v + \int_\Omega f v \quad \forall v \in V, \tag{3.7}$$

where $\epsilon > 0$ is a parameter one can choose. It can be shown that the direction w_n determined by (3.7) is the steepest descent direction with $V \hookrightarrow H_0^1(\Omega)$ equipped a weighted norm $\|\cdot\|_{\epsilon, u_n}^2 = \int_\Omega (\epsilon + |\nabla u_n|^{p-2}) |\nabla \cdot|^2$. Thus we propose the following steepest descent method.

Algorithm 2. For a given initial value u_0 do the following iterations

- solving $\int_\Omega (\epsilon + |\nabla u_n|^{p-2}) \nabla w_n \nabla v = -J'(u_n)(v) = -\int_\Omega |\nabla u_n|^{p-2} \nabla u_n \nabla v + \int_\Omega f v \quad \forall v \in V$
- linesearch for α_n, such that $J(u_n + \alpha_n w_n) = \min_{\alpha \geqslant 0} J(u_n + \alpha w_n)$
- update $u_{n+1} = u_n + \alpha_n w_n$

Again the direction w_n can be solved by fast MG solvers.

For the case $p < 2$, one can use the weighted norm $|\cdot|_{u_n}^2 = \int_\Omega (\epsilon + |\nabla u_n|)^{p-2} |\nabla \cdot|^2$ and form the following algorithm:

Algorithm 3. For a given initial value u_0 do the following iterations

- solving $\int_\Omega (\epsilon + |\nabla u_n|)^{p-2} \nabla w_n \nabla v = -J'(u_n)(v) = -\int_\Omega |\nabla u_n|^{p-2} \nabla u_n \nabla v + \int_\Omega f v \quad \forall v \in V$
- linesearch for α_n, such that $J(u_n + \alpha_n w_n) = \min_{\alpha \geqslant 0} J(u_n + \alpha w_n)$
- update $u_{n+1} = u_n + \alpha_n w_n$

As in Algorithm 2, the direction w_n is solved by fast MG solvers.

3.2. Convergence Analysis

In this section we carry out some convergence analysis for the algorithms proposed in Section 3.1. It is usually difficult to obtain optimal theoretical results for the p-Laplacian. For example, the speed of subspace correction method has only been shown of a convergence rate $O(n^{-\alpha(p)})$ with $\alpha(p)\to 0$ as $p\to 1$ or ∞, see [27]. We begin with a lemma.

Lemma 1. Suppose there exist constants $\mu>0, \gamma>1$ such that the number series $\lambda_n>0\,(n=1,2,\cdots)$ satisfies the inequality

$$\lambda_n-\lambda_{n+1}\geqslant \mu\lambda_n^{\gamma}$$

Then there holds, with $\beta=\frac{1}{\gamma-1}$

$$\lambda_n\leqslant \frac{1}{n^{\beta}}\max\left\{\lambda_1, \left(\frac{2^{\beta}-1}{\mu}\right)^{\beta}\right\}.$$

Proof. From the assumpsions of the lemma we know that the number series λ_n decreases monotonically. Hence we have

$$\lambda_n-\lambda_{n+1}=\lambda_{n+1}\left(\frac{\lambda_n}{\lambda_{n+1}}-1\right)\geqslant \mu\lambda_n^{\gamma}\geqslant \mu\lambda_{n+1}^{\gamma} \tag{3.8}$$

so

$$\frac{\lambda_n}{\lambda_{n+1}}-1\geqslant \mu\lambda_{n+1}^{\gamma-1} \tag{3.9}$$

set

$$\rho_n=\lambda_n n^{\beta}. \tag{3.10}$$

Then (3.9) is equivalent to

$$\frac{\rho_n}{\rho_{n+1}}\geqslant \left(1+\mu\rho_{n+1}^{\gamma-1}n^{-\beta(\gamma-1)}\right)\left(\frac{n}{n+1}\right)^{\beta}. \tag{3.11}$$

It is easy to see that the right hand side of (3.11) increases when ρ_{n+1} increases. Now we examine when it is not smaller than 1.

$$\left(1+\mu\rho_{n+1}^{\gamma-1}n^{-\beta(\gamma-1)}\right)\left(\frac{n}{n+1}\right)^{\beta}\geqslant 1 \tag{3.12}$$

that means

$$\rho_{n+1} \geqslant \left[\frac{1}{\mu}\left((1+\frac{1}{n})^{\beta}-1\right)n\right]^{\beta}. \tag{3.13}$$

Note the right hand side of (3.13) decreases. Hence if $\rho_{n+1} \geqslant \left(\frac{2^{\beta}-1}{\mu}\right)^{\beta}$, it follows from (3.11) to (3.13) that we have $\rho_{n+1} \leqslant \rho_n$. Therefore we have

$$\rho_n \leqslant \max\left\{\rho_1, \left(\frac{2^{\beta}-1}{\mu}\right)^{\beta}\right\}. \tag{3.14}$$

□

Now we prove the first convergence result. Since Algorithm 2 worked very well in our numerical experiments, we shall only examine convergence of Algorithm 2.

Theorem 1. Let $\{u_n\}$ be generated by Algorithm 2 stated in Sect. 3.1, Then there exists a constant $C > 0$ such that

$$J(u_n) - J(u) \leqslant \frac{1}{n^{\beta}} \max\{C, J(u_0) - J(u)\}, \quad \beta = \frac{p}{p-2}.$$

Proof. By the convexity we have

$$J(u_n) - J(u) \leqslant (J'(u_n) - J'(u))(u_n - u) \leqslant \|J'(u_n)\|_* \|u_n - u\|, \tag{3.15}$$

where $\|\cdot\|$ is the weighted norm used in Algorithm 2. Denote

$$e_n = J(u_n) - J(u).$$

For any $\alpha > 0$ from the definition of the steepest descent direction we have

$$\begin{aligned} J(u_n + \alpha w_n) - J(u_n) &= \alpha J'(u_n)(w_n) + \tfrac{\alpha^2}{2} J''(\theta_n)(w_n, w_n) \\ &\leqslant \alpha J'(u_n)(w_n) + \tfrac{\alpha^2}{2} M \|w_n\|^2 \\ &= (-\alpha + \tfrac{\alpha^2}{2} M)\|J'(u_n)\|_*^2. \end{aligned} \tag{3.16}$$

The minimum of the right hand side is achieved at $\widetilde{\alpha} = \frac{1}{M}$. So

$$\begin{aligned} e_{n+1} - e_n &= J(u_{n+1}) - J(u_n) = J(u_n + \alpha_n w_n) - J(u_n) \\ &\leqslant J(u_n + \tilde{\alpha} w_n) - J(u_n) \\ &= -\tfrac{1}{2M}\|J'(u_n)\|_*^2. \end{aligned} \tag{3.17}$$

It is clear that $\{\|u_n\|_{W^{1,p}(\Omega)}\}$ is bounded, and then it follows from the Holder inequality that $\{\|u_n\|\}$ is also bounded. Thus from the Holder inequality there exists a $C_1 > 0$ such that

$$\|u - u_n\| \leq C_1 \|u - u_n\|_{W^{1,p}(\Omega)}.$$

Since $p > 2$, using (2.6) and $\frac{1}{q} + \frac{1}{p} = 1$ we have

$$\begin{aligned}(J'(u_n) - J'(u))(u_n - u) &\leqslant \|J'(u_n)\|_* \|u_n - u\| \leq C_1 \|J'(u_n)\|_* \|u_n - u\|_{1,p} \\ &\leqslant C_1^q \|J'(u_n)\|_*^q / q + \|u_n - u\|_{1,p}^p / p \\ &\leqslant C_1^q \|J'(u_n)\|_*^q / q + (J'(u_n) - J'(u))(u_n - u)/p\end{aligned}$$

Noting that $p > 2$, thus there is a constant $C_2 > 0$ such that

$$(J'(u_n) - J'(u))(u_n - u) \leqslant C_2 \|J'(u_n)\|_*^q. \tag{3.18}$$

It follows from (3.15) that

$$e_n \leq C_2 \|J'(u_n)\|_*^q.$$

Then it follows from (3.17) and (3.18) that there exists a constant C_3 such that

$$e_n - e_{n+1} \geqslant C_3 e_n^{\frac{2}{q}}. \tag{3.19}$$

Then the desired result follows from Lemma 1. □

Corollary 1. Under the conditions of Theorem 1, there exists a constant $C > 0$ such that

$$|u - u_n|_{(u,p)}^2 \leq C/n^\beta, \quad \beta = \frac{p}{p-2}, \tag{3.20}$$

where $|w - v|_{(v,p)}^2$ is the so called the quasi-norm defined by (see [20, 21])

$$|w - v|_{(v,p)}^2 = \int_\Omega (|\nabla v| + |\nabla(w - v)|)^{p-2} |\nabla(w - v)|^2. \tag{3.21}$$

Proof. From the basic properties of the p-Laplacian we have

$$J(w) - J(v) \geqslant J'(v)(w - v) + c|w - v|_{(v,p)}^2. \tag{3.22}$$

Hence

$$\begin{aligned} J(u_n) - J(u) &\geqslant J'(u)(u_n - u) + c|u_n - u|^2_{(u,p)} \\ &= c|u_n - u|^2_{(u,p)}. \end{aligned} \tag{3.23}$$

From Theorem 1, we have

$$|u_n - u|^2_{(u,p)} \leq C/n^{\beta}.$$

□

In the above algorithms line search is applied at every step which is usually very time consuming. A way to reduce the work on line search is to perform an inexact line search or even to use a fixed step length. It follows from the proof of Theorem 1 that if we choose a fixed step length α_n such that $0 < \delta \leqslant \alpha_n \leqslant \frac{2}{M} - \delta$, then we still have the convergence.

Theorem 2. Suppose that the conditions of Theorem 1 hold and that the step length α_n satisfy $0 < \delta \leqslant \alpha_n \leqslant \frac{2}{M} - \delta$. Then the descent algorithm Agorithm 2 (without line search) is uniformly convergent, and there exists a constant $C_1 > 0$ such that

$$J(u_n) - J(u) \leqslant \frac{1}{n^{\beta}} \max\{C_1, J(u_0) - J(u)\}, \quad \beta = \frac{p}{p-2}.$$

Proof. From (3.16) we see that

$$e_{n+1} - e_n = J(u_n + \alpha_n w_n) - J(u_n) \leqslant \Big(-\delta + \frac{\delta^2}{2} M \Big) \|J'(u_n)\|^2_*. \tag{3.24}$$

Combining (3.15) with lemma 1 yields the desired result. □

4. NUMERICAL EXPERIMENTS

4.1. Implementation Issues

Our computations are carried out in an IBM Thinkpad T22 laptop. The program language is Fortran 90. We used an AMG solver for computing the descent direction w_n. The stopping role for the AMG iterations is to reduce the relative defect to 10^{-8} and the maximum V-Cycles is 50.

For large scale problems functional value evaluation is very time consuming, so is the line search procedure. We used a half-section algorithm with eight iterations for a rough line search. The current step length is used as an initial value for the initialization of the search interval at the next step.

The stopping criterion is $|J'(u_n)|/|J'(u_0)| < 10^{-6}$, where u_0 is an initial solution. Our numerical results show that this criterion can guarantee the accuracy for both the object functional and the solution.

We used piecewise linear triangle finite element approximation in all our computations. We always used zero as the initial solution in all the iterations if not stated otherwise.

4.2. Numerical Examples

In all the following tables, the errors are always measured in L^2 norm if not stated otherwise.

Example 1. $\Omega = \{(x, y), r^2 = x^2 + y^2 < 1\}$, $f = 1$. The exact solution is

$$u = u(r) = \frac{p-1}{p}\left(\frac{1}{2}\right)^{\frac{1}{p-1}}\left(1 - r^{\frac{p}{p-1}}\right). \tag{4.1}$$

In the tables below, $C1, C2, C3, C4$ represent the meshes with unknowns $1601, 6221, 24444, 97118$ respectively. Mesh $C1$ is shown in Fig. 1.

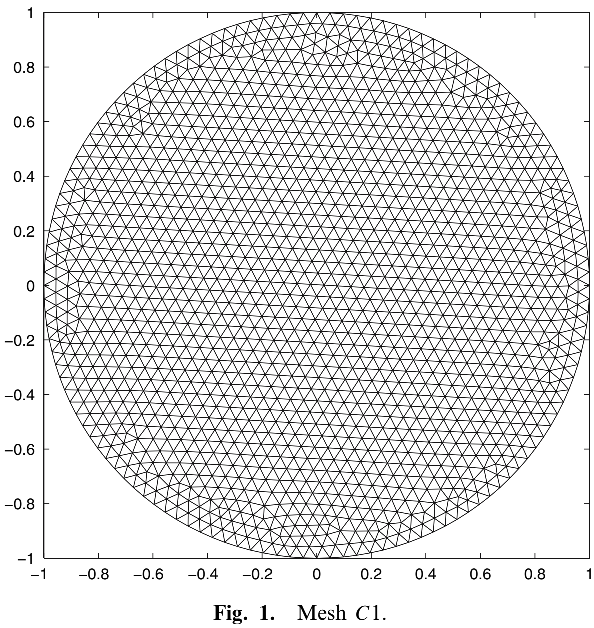

Fig. 1. Mesh $C1$.

Tables I to V show the computational results using the preconditioned descent algorithm with the weighted norm preconditioner and $\varepsilon = 10^{-4}$. The recorded iteration numbers indicate that the convergence for our algorithm is mesh independent for a fixed p. The numerical results also show $O(h^2)$ convergence rate for both the functional and the solution when p is not too large, although the rate seems to decrease as p becomes large. It seems that the lowest rate as p becomes larger tends to $O(h)$.

The convergence behaviors are shown in the Figs. 2–6.

The number of iterations does increase when p becomes very large as the meshes are refined. This seems to indicate that the preconditioner used is not perfect for large p. However the iteration numbers tend to be stable for reasonably large p. It can also be seen that the CPU-time increases linearly with DOFs. We have also tried other solvers such as

Table I. ($\epsilon = 10^{-4}$)

$p=4$	C1	C2	C3	C4
Iter num	9	9	9	9
CPU time	0 m16.52 s	1 m05.75 s	4 m42.46 s	18 m35.92 s
$J(u_h) - J(u)$	5.3997e−04	1.3386e−04	3.2068e−05	6.6333e−06
$\|u - u_h\|$	5.0978e−04	1.2773e−04	3.1754e−05	7.9779e−06
$\|u_I - u_h\|$	6.7468e−05	1.6570e−05	4.0327e−06	1.0704e−06

Table II. ($\epsilon = 10^{-4}$)

$p=10$	C1	C2	C3	C4
Iter num	19	15	14	14
CPU time	0 m33.03 s	1 m55.19 s	7 m31.89 s	30 m43.24 s
$J(u_h) - J(u)$	9.3238e−04	2.3233e−04	5.5800e−05	1.1996e−05
$\|u - u_h\|$	9.2638e−04	2.3968e−04	6.0411e−05	1.6045e−05
$\|u_I - u_h\|$	1.9301e−04	5.4713e−05	1.6180e−05	4.4031e−06

Table III. ($\epsilon = 10^{-4}$)

$p=20$	C1	C2	C3	C4
Iter num	31	28	23	24
CPU time	0 m54.05 s	3 m24.32 s	12 m08.42 s	52 m05.58 s
$J(u_h) - J(u)$	1.3667e−03	3.4489e−04	8.3504e−05	1.8747e−05
$\|u - u_h\|$	1.3890e−03	3.7677e−04	9.6496e−05	2.5680e−05
$\|u_I - u_h\|$	5.6286e−04	1.6770e−04	4.7522e−05	1.2561e−05

Table IV. ($\epsilon = 10^{-4}$)

$p = 100$	C1	C2	C3	C4
Iter num	79	86	71	64
CPU time	2 m24.28 s	11 m04.27 s	37 m26.33 s	152 m15.21 s
$J(u_h) - J(u)$	3.7250e−03	1.0488e−03	2.6971e−04	6.6751e−05
$\|u - u_h\|$	3.4210e−03	1.0793e−03	3.1551e−04	9.1268e−05
$\|u_I - u_h\|$	2.6080e−03	8.7380e−04	2.6672e−04	7.8120e−0

Table V. ($\epsilon = 10^{-4}$)

$p = 1000$	C1	C2	C3	C4
Iter num	161	340	461	411
CPU time	5 m01.97 s	44 m37.02 s	245 m07.93 s	919 m02.79 s
$J(u_h) - J(u)$	7.7599e−03	3.3253e−03	1.2665e−03	4.2736e−04
$\|u - u_h\|$	6.2626e−03	2.8116e−03	1.1740e−03	4.4579e−04
$\|u_I - u_h\|$	5.4966e−03	2.6248e−03	1.1299e−03	4.3451e−04

BICGSTAB with ILU preconditioning. All of them show slow convergence at the beginning. After several steps of iterations the solvers all work well.

Table VI shows some results using the Poisson preconditioner for $p = 4$. The mesh independent convergence is observed from the numerical experiments. However as p becomes large, the Poission preconditioner becomes much less efficient than the weighted norm preconditioner. Thus we shall not show more numerical results using this preconditioner.

The convergence behaviors are shown in Fig. 7.

Different choices of parameter ε may yield different iteration numbers. Table VII shows the results for $p = 10$ using different parameters on the mesh $C3$. More iterations are needed to meet the error tolerance as ε becomes large. However too small ε may cause large round off error and make the program overflow or lowerflow especially when p is large. From our experiments it was found that $\varepsilon = 10^{-4}$ gives a good balance.

It was observed that the accuracy of line search is not very important to the convergence behavior of our algorithms. For example when we increase the accuracy of line search, the iteration number for $p = 10$ on the mesh $C3$ is still the same, although the CPU time increases from 7 m31.89 s to 11 m32.230 s. For $p = 100$ on the mesh $C1$ the iteration number even raises from 79 to 93 and the CPU time increases from 2 m24.28 s to 4 m41.730 s. It was observed that the step lengths are relatively stable

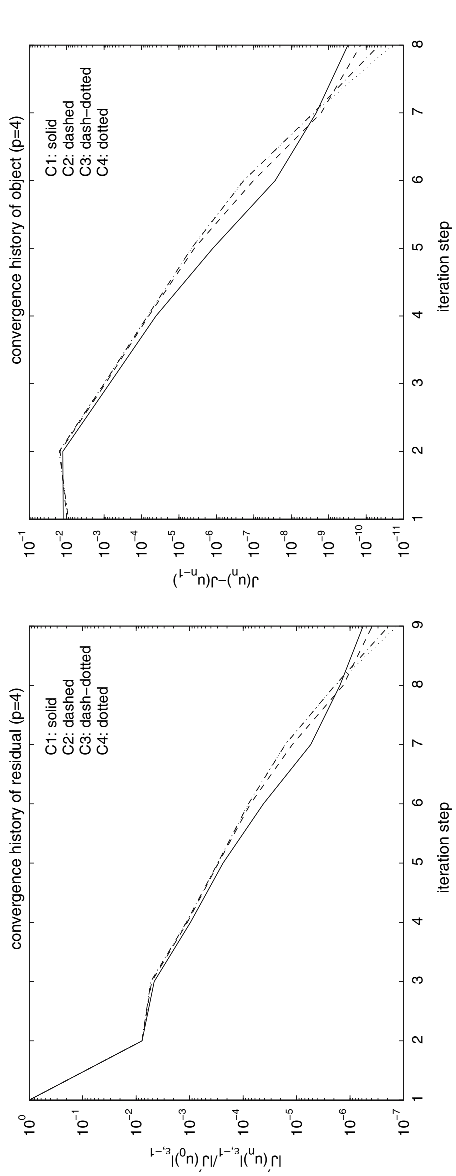

Fig. 2. Weighted preconditioner.

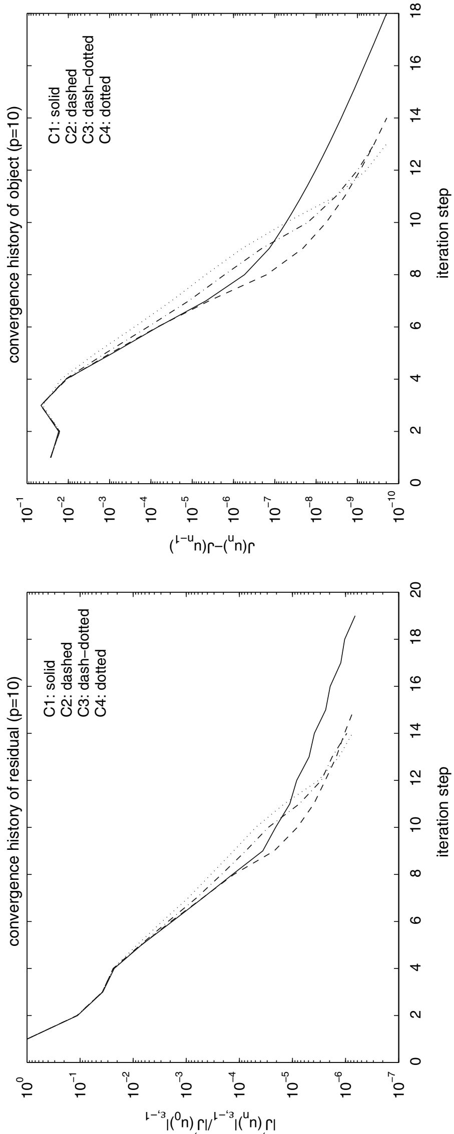

Fig. 3. Weighted preconditioner.

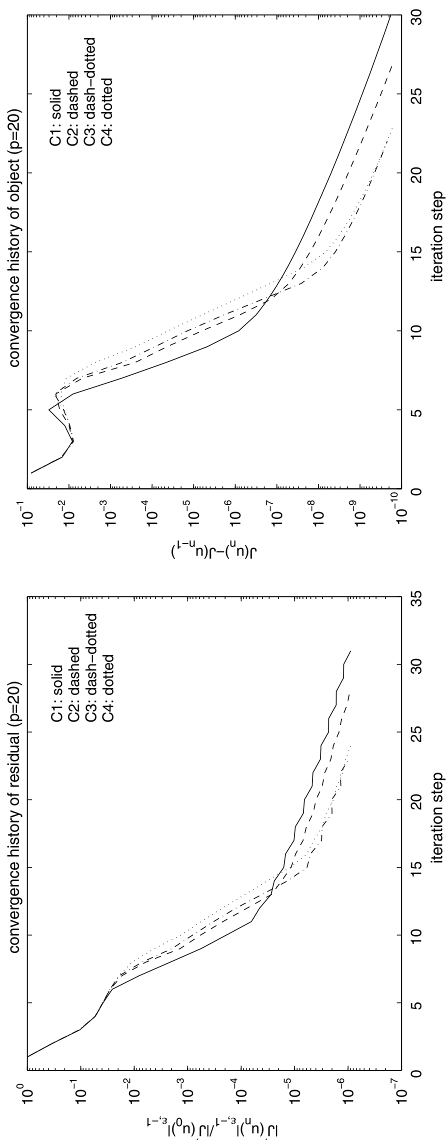

Fig. 4. Weighted preconditioner.

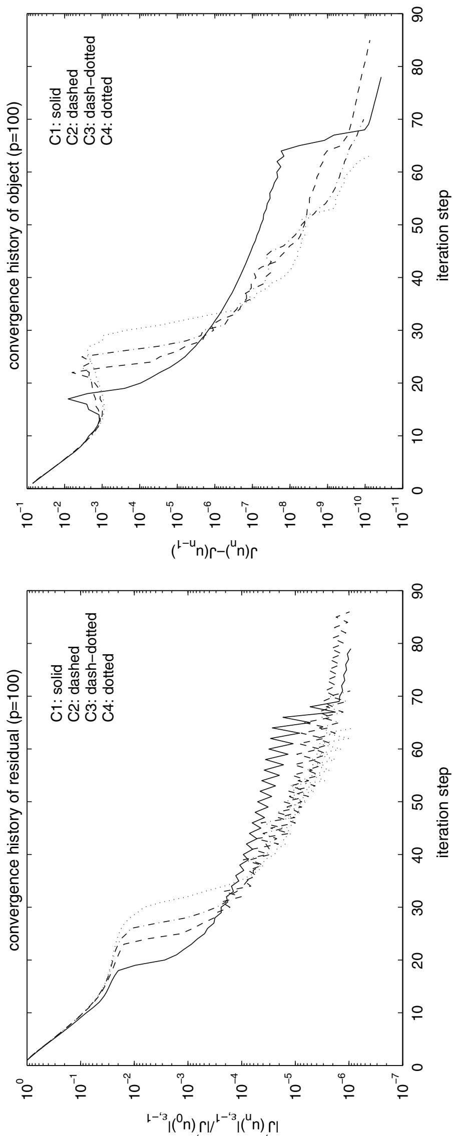

Fig. 5. Weighted preconditioner.

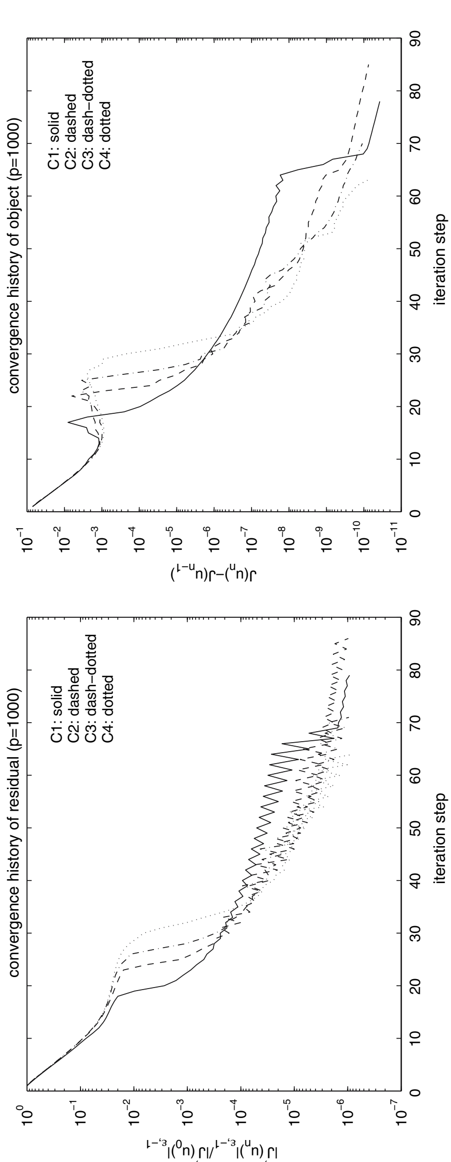

Fig. 6. Weighted preconditioner.

Table VI. (Poisson)

$p=4$	C1	C2	C3	C4
Iter num	71	87	102	130
CPU time	1 m54.15 s	9 m22.50 s	44 m45.57 s	236 m32.67 s
$J(u_h)-J(u)$	5.3997e−04	1.3386e−04	3.2067e−05	6.6333e−6
$\|u-u_h\|$	5.1057e−04	1.2804e−04	3.1856e−05	8.0089e−06
$\|u_I-u_h\|$	6.6419e−05	1.6156e−05	3.9012e−06	9.9516e−07

during the iteration procedure. It seems unnecessary to perform accurate linesearch. A rough one is enough since we have good initial intervals from the previous iterations. See Fig. 8 for example.

Example 2. We consider $\Omega=(0,1)\times(0,1)$ and $f=0$ with a non-homogeneous boundary condition such that the exact solution is

$$u=r^{\frac{p-2}{p-1}}, \quad r=\sqrt{x^2+y^2}. \tag{4.2}$$

We use $R1$, $R2$, $R3$, $R4$ to represent the meshes with 782, 3005, 11800, 46636 nodes respectively. Mesh $R2$ is shown in Fig. 9. The numerical results are presented in Table VIII.

From Table VIII we see that the convergence rate is only of first order. Since the solution is singular some local refinement is needed. Table IX shows the results with local refinements. We use a local refinement such that $h_{\min}=\frac{1}{4}h_{\max}$, and the corresponding meshes $R1LR$, $R2LR$, $R3LR$, $R4LR$, have 1186, 4616, 18075, 72456 nodes respectively. Mesh $R2LR$ is shown in Fig. 9. With the adaptive meshes both the accuracy and convergence rate are improved significantly. It was found that adaptive meshes can improve accuracy, but cannot change the essential behaviors of the algorithms.

Remark 1. As $p\to 1$, the p-Laplacian is numerically unstable. To see this, let $g=(1+\varepsilon)f$ be a small perturbation of f, and u_g, u_f be the solutions of the p-Laplacain for g, f respectively. A direct calculation yields that

$$u_g=(1+\varepsilon)^{\frac{1}{p-1}}u_f. \tag{4.3}$$

Thus if p is very close to 1, u_g may be dramatically different from u_f even though ε may be very small.

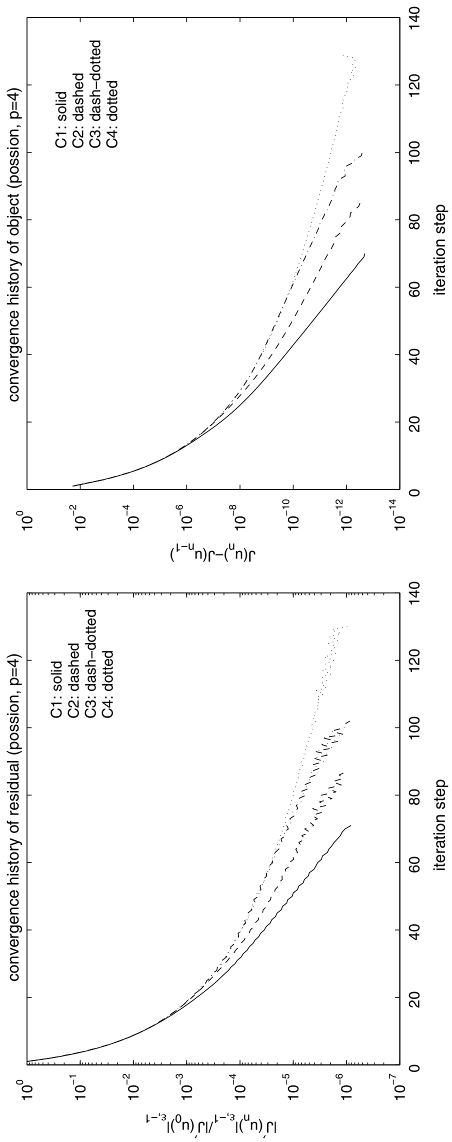

Fig. 7. Poission preconditioner.

Table VII. (C3 mesh)

$p=10$	$\varepsilon=10^{-1}$	$\varepsilon=10^{-2}$	$\varepsilon=10^{-3}$	$\varepsilon=10^{-4}$
Iter num	105	25	18	14
CPU time	49 m19.41 s	12 m3.140 s	8 m31.700 s	7 m31.89 s
$J(u_h)-J(u)$	5.5800e−05	5.5800e−05	5.5800e−05	5.5800e−05
$\|u-u_h\|$	5.9594e−05	5.9662e−05	5.9618e−05	6.0411e−05

To deal with the case where $\frac{1}{p-1}$ is very large, one has to handle the computational procedures of evaluating function power very carefully in order to minimize the accumulative run-off errors. Without special considerations in this aspect, our methods can only handle the case where $\frac{1}{p-1}$ is not too large. Table X shows the results for example 1 with $p=1.1$. We applied Algorithm 3 with a AMG solver for w_n (see [15]). Since the solution may be very small we have shown the relative error.

According to our computational experience, Algorithm 3 works reasonably well as long as $1.1 \leq p \leq 2$. We also tried the case $p=1.06$. The speed is much slower although it eventually converged. In such a case, much attention has to be paid to reduce round-off errors. In some cases, one can use transformations to convert the case $p<2$ into the case $p>2$, although it may not be easy to decide the new boundary conditions for the transformed problems.

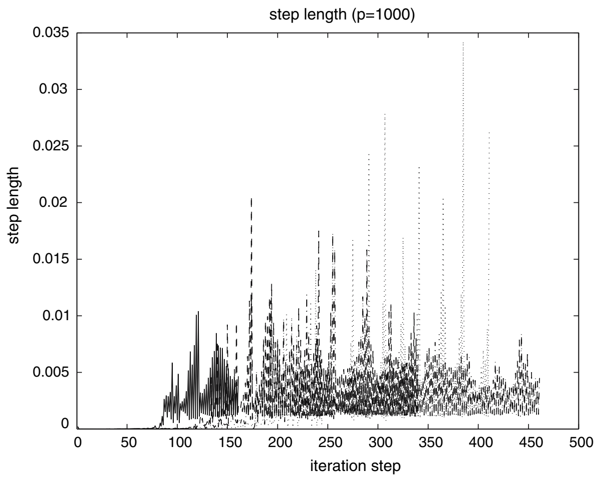

Fig. 8. Steplength history, $p=1000$, $C4$ mesh.

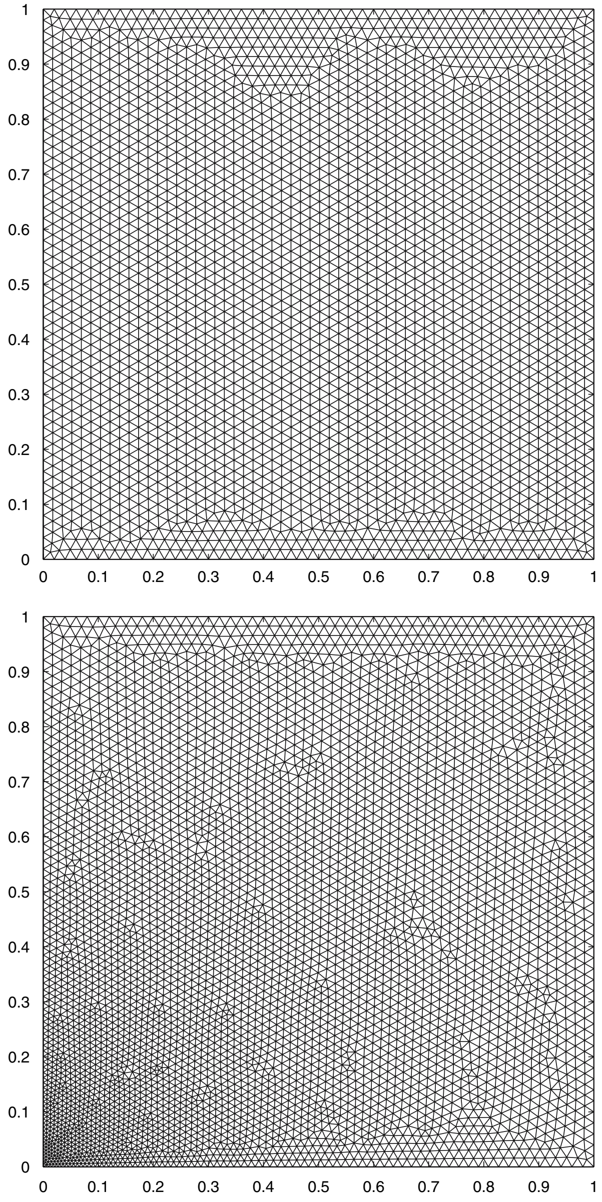

Fig. 9. Mesh R2 and mesh R2LR.

Table VIII. ($\epsilon = 10^{-4}$)

$p = 20$	R1	R2	R3	R4
Iter num	72	69	53	65
CPU time	1 m06.86 s	4 m25.58 s	13 m54.09 s	68 m14.78 s
$J(u_h) - J(u)$	5.8212e−03	3.0068e−03	1.5579e−03	8.0008e−04
$\|u - u_h\|$	1.9953e−03	9.8822e−04	4.7180e−04	2.1732e−04
$\|u_I - u_h\|$	2.1592e−03	1.0322e−03	4.8311e−04	2.2025e−04

Table IX. ($\epsilon = 10^{-4}$)

$p = 20$	R1LR	R2LR	R3LR	R4LR
Iter num	72	72	65	65
CPU time	1 m45.030 s	7 m6.25 s	25 m48.95 s	118 m25.42 s
$J(u_h) - J(u)$	2.9041e−03	1.3939e−03	4.6046e−04	2.2906e−04
$\|u - u_h\|$	7.4261e−04	3.2387e−04	1.1158e−04	4.9229e−05
$\|u_I - u_h\|$	8.1822e−04	3.4302e−04	1.1648e−04	5.0446e−05

Table X. ($\epsilon = 10^{-4}$)

$p = 1.1$	C1	C2	C3	C4
Iter num	7	7	7	7
CPU time	0 m13.96 s	0 m57.70 s	3 m53.53 s	17 m01.33 s
$(J(u_h) - J(u))/J(u)$	1.8317e−03	4.5488e−04	1.1056e−04	2.3950e−05
$\|u - u_h\|/\|u\|$	1.8040e−03	4.4368e−04	9.9125e−05	2.0049e−05

5. EXTENSIONS AND AN APPLICATION TO SOLVING VARIATIONAL INEQUALITIES

The algorithms and the theoretical analysis presented in the previous sections can be readily extended to more general degenerate equations like those studied in [20].

In this section we shall apply our algorithms to the following problem:

$$\min_{u \in W_0^{1,p}(\Omega)} J_p(u), \quad (**)$$

where

$$J_p(u) = \frac{1}{2}\int_\Omega |\nabla u|^2 + \frac{1}{p}\int_\Omega |\nabla u|^p - \int_\Omega fu \quad \text{on } W_0^{1,p}(\Omega),\ f \in L^\infty(\Omega), \quad (5.1)$$

whose solution approximates that of an important variational inequality, to be described below. It is known that as $p \to \infty$, the solution of a parabolic p-Laplacian converges to that of a super-conductivity model, see [3] for the details, and thus open a possible route to computing the model. We shall show that as $p \to \infty$, the solution u_p of (**) converges to that of the following important variational inequality (EPVI) from elastic-plastic mechanics, see [13]: $u \in H_0^1(\Omega)$, $|\nabla u| \le 1$:

$$\int_\Omega (\nabla u, \nabla(v-u)) \ge \int_\Omega f(v-u), \forall v : v \in H_0^1(\Omega), \quad |\nabla v| \le 1. \tag{5.2}$$

Its computational methods have been extensively studied in the literature, see [13] again.

Now let us first show that the problem (**) has a unique solution. Let

$$J_p(u) = \frac{1}{2}\int_\Omega |\nabla u|^2 + \frac{1}{p}\int_\Omega |\nabla u|^p - \int_\Omega fu \quad \text{on } W_0^{1,p}(\Omega),\ f \in L^\infty(\Omega). \tag{5.3}$$

It is easy to check that:

- $J_p(u)$ is convex and continues on $W_0^{1,p}(\Omega) = V$
- $\frac{J_p(u)}{\|u\|_{W^{1,p}}} \to \infty$ as $\|u\|_{W_0^{1,p}} \to \infty$

Therefore $J_p(u)$ has a unique minimizer in V, and its Euler equation reads:

$$(\nabla u_p, \nabla v) + (|\nabla u_p|^{p-2}\nabla u_p, \nabla v) = (f, v) \quad \forall v \in W_0^{1,p}(\Omega) \tag{5.4}$$

Theorem 3. As $p \to \infty$, u_p converges to u, the solution of the variational inequality (EPVI).

Proof. Let $v = u_p$ we have for any fixed $p > 1$,

$$\int_\Omega |\nabla u_p|^2 + \int_\Omega |\nabla u_p|^p = \int_\Omega f u_p \tag{5.5}$$

Thus,

$$\|u_p\|_{H^1} \le \|f\|_{L^q} \tag{5.6}$$

and

$$\|\nabla u_p\|_{L^p} \le (M\|f\|_{L^q})^{\frac{1}{p}}, \tag{5.7}$$

where $1/p+1/q=1$. Thus without losing generality, let $u_p \to u$ in $H_0^1(\Omega)$ weakly so that it converges to u weakly in $W^{1,r}(\Omega)$ for any fixed $r>1$. Then we have

$$|\nabla u| \leq \lim_{p\to\infty} \|\nabla u_p\|_{L^p} \leq 1 \tag{5.8}$$

and

$$\|u\|_{H^1} \leq \|f\|_{L^q}. \tag{5.9}$$

Furthermore, for any $v \in V = \{v \in H_0^1(\Omega), |\nabla v| \leq 1\}$,

$$\begin{aligned}(\nabla u_p, \nabla(u_p - v)) &= \int_\Omega |\nabla u_p|^{p-2} \nabla u \nabla(v-u_p) + \int_\Omega f(u_p - v)\\ &\leqslant \frac{1}{p}\int_\Omega |\nabla u_p|^p + \int_\Omega |\nabla u_p|^{p-2}\nabla u \nabla(v-u_p) + \int_\Omega f(u_p-v).\end{aligned}$$

Using the convexity

$$g(u) - g(v) \leqslant g'(u)(u-v) \tag{5.10}$$

for $g(v) = \frac{1}{p}\int_\Omega |\nabla v|^p$, we have

$$\frac{1}{p}\int_\Omega |\nabla u_p|^p + \int_\Omega |\nabla u_p|^{p-2}\nabla u \nabla(v-u_p) \leqslant \frac{1}{p}\int_\Omega |\nabla v|^p. \tag{5.11}$$

Hence,

$$(\nabla u_p, \nabla(u_p - v)) \leqslant \frac{1}{p}\int_\Omega |\nabla v|^p + \int_\Omega f(u_p - v). \tag{5.12}$$

Let $p \to \infty$. As $u_p \to u$ in H^1 weakly, we have

$$(\nabla u, \nabla(u-v)) \leqslant \lim_{p\to\infty} (\nabla u_p, \nabla(u_p - v)) \leqslant \int_\Omega f(u-v). \tag{5.13}$$

Therefore,

$$(\nabla u, \nabla(v-u)) \geqslant \int_\Omega f(v-u) \tag{5.14}$$

and

$$|\nabla u| \leqslant 1. \tag{5.15}$$

We applied our (slightly modified) algorithms to (**) with $p=100$ to approximate the solution of (EPVI), and satisfactory numerical results have been observed with this new method. Two computational results are presented in the follows.

The first example is on the circle domain

$$\Omega=\{x|x=(x_1,x_2), r^2=x_1^2+x_2^2\leq R^2\}.$$

For $f=C\geq 2/R$, the exact solution is given by

$$u(x)=\begin{cases} R-r, & R'\leq r\leq R \\ -Cr^2/4+(R-1/C), & 0\leq r\leq R' \end{cases}$$

with $R'=2/C$, see page 122 [11]. In our numerical example, $R=1$ and $C=4$. The following figures show the contours of the numerical solution (Figs. 10,11).

The second example is taken from [11], see page 132 for the details. The computational domain is the unit square, while f is the same as the above. The exact solution is unknown. The contours of the numerical solution is shown in the follows, which can be compared with those in [11].

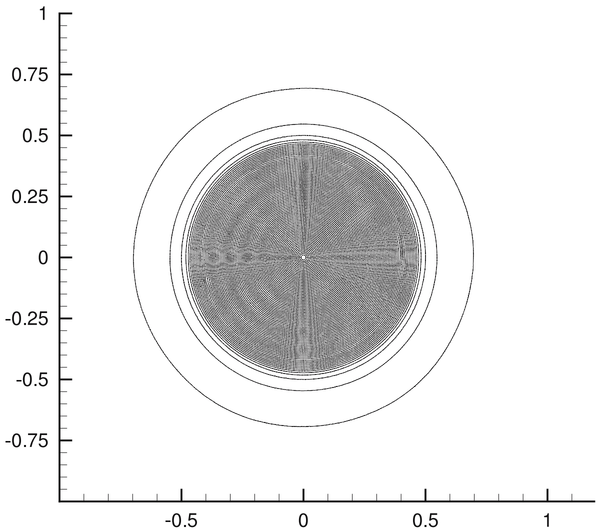

Fig. 10. Contours of the solution.

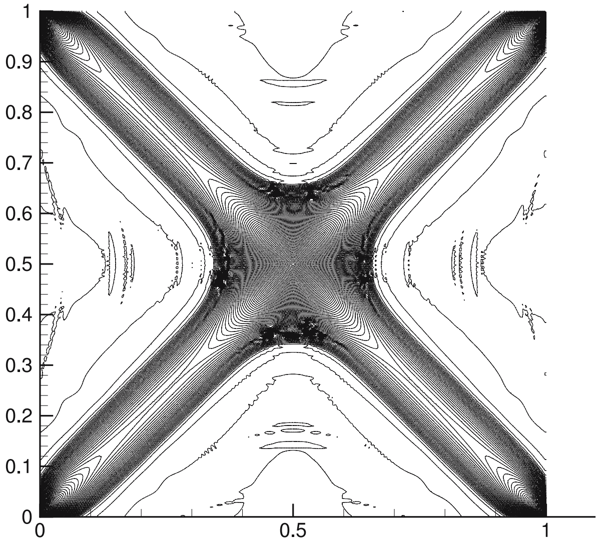

Fig. 11. Contours of the solution.

ACKNOWLEDGMENTS

Supported by EPSRC research grant GR/R31980. Huang's work is supported in part by the NSFC for Distinguished Young Scholars(10625106) and National Basic Research Program of China under the grant 2005CB321701.

REFERENCES

1. Ainsworth, M., and Kay, D. (1999). The approximation theory for the p-version finite element method and application to non-linear elliptic PDEs. *Numer. Math.* **83**, 351–388.
2. Baranger, J., and Amri, H. El. (1991). Estimateurs a posteriori d'erreur pour le calcul adaptatif d'ecoulements quasi-Newtoniens. *R.A.I.R.O. M^2AN* **25**, 31–48.
3. Barrett, J. W., and Prigozhin, L. (2000). Bean's critical-state model as $p \to \infty$ limit of an evolutionary p-Laplacian equation. *J. Nonlinear Anal.* **42**, 977–993.
4. Barrett, J. W., and Liu, W. B. (1993). Finite element approximation of the p-Laplacian. *Math. Comp.* **61**, 523–537.
5. Barrett, J. W., and Liu, W. B. (1994). Finite element approximation of some degenerate quasi-linear problems. *Lecture Notes Math.* **303**, 1–16
6. Barrett, J. W., and Liu, W. B. (1994). Quasi-norm error bounds for finite element approximation of quasi-Newtonian flows. *Numer. Math.* **68**, 437–456.
7. Bermejo, R., and Infante, J. (2000). A multigrid algorithm for the p-Laplaican. *SIAM, J. Sci. Comput*. **21**, 1774–1789.
8. Bouchitte, G., Buttazzo, G., and De Pascale, L. (2003). A p-Laplacian approximation for some mass optimization problems. *JOTA* **118**, 1–25.

9. Carstensen C., and Klose, R. (submitted). Guaranteed a posteriori finite element error control for p-Laplacian.
10. Chow, S. S. (1988). Finite element error estimates for non-linear elliptic equations of monitone type. *Numer. Math*. **54**, 373–393.
11. Ciarlet, P. G. (1978). *The Finite Element Method for Elliptic Problems*. North Holland, Amsterdam.
12. Farhloul, M. (1998). A mixed finite element method for a nonlinear Direchlet problem. *IMA J. Numer. Anal*. **18**, 121–132.
13. Glowinski, R., and Lions, J. L., and Tremolieres, R. (1976). *Numerical Analysis of Variational Inequalities*, North-Holland, Netherlands.
14. Glowinski, R., and Marrocco, A. (1975). Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité, d'une classe de problémes de Dirichlet non linéaires (French). *RAIRO Anal. Numer*. **9**(R-2), 41–76.
15. Huang, Y. Q., Shu, S., and Yu, X.-J. (2006). Preconditioning higher order finite element systems by algebraic multigrid method of linear element. *J. Comput. Math*. **24**(5), 657–664.
16. Liu, W. B., and Barrett, J. W. (1993). Higher order regularity for the solutions of some nonlinear degenerate elliptic equations. *SIAM J. Math. Anal*. **24**(6), 1522–1536.
17. Liu, W. B., and Barrett, J. W. (1993). A remark on the regularity of the solutions of p-Laplacian and its applications to their finite element approximation. *J. Math. Anal. Appl*. **178**, 470–488.
18. Liu, W. B., and Barrett, J. W. (1993). A further remark on the regularity of the solutions of the p-Laplacian and its applications to their finite element approximation. *J. Nonlinear. Anal*. **21**, 379–387.
19. Liu, W. B. (2000). Degenerate quasilinear elliptic equations arising from bimaterial problems in elastic-plastic mechanics II. *Numer. Math*. **86**, 491–506.
20. Liu, W. B., and Barrett, J. W. (1996). Finite element approximation of some degenerate monotone quasi-linear elliptic systems. *SIAM J. Numer. Anal*. **33**, 88–106.
21. Liu, W. B., and Yan, N. (2001). Quasi-norm local error estimates for the finite element approximation of p-Laplacian. *SIAM J. Numer. Anal*. **39**, 100–127.
22. Liu, W. B., and Yan, N. (2003). On quasi-norm interpolation error estimation and a posteriori error estimates for p-Laplacian. *SIAM J. Numer. Anal*.
23. Manouzi, H., and Farhloul, M. (2001). Mixed finite element analysis of a non-linear three-fields Stokes model. *IMA J. Numer. Anal*. **21**, 143–164.
24. Padra, C. (1997). A posteriori error estimators for nonconforming approximation of some quasi-Newtonian flows. *SIAM J. Numer. Anal*. **34**, 1600–1615.
25. Simms, G. (1995). Finite element approximation of some nonlinear elliptic and parabolic problems, thesis. Imperial College, University of London.
26. Verfürth, R. (1994). A posteriori error estimates for non-linear problems. *Math. Comp*. **62**, 445–475.
27. Tai, X. C., and Espedal, M. (1998). Rate of convergence of some space decomposition method for linear and nonlinear elliptic problems. *SIAM J. Numer. Anal*. **38**, 1558–1570.
28. Tai, X. C., and Xu, J. (2002). Global convergence of subspace correction methods for convex optimization problems. *Math. Comp*. **74**, 105–124.