Minimization techniques for p(x)-Laplacian problem using WEB-Spline based mesh-free method

Article in International Journal of Computer Mathematics · February 2019

DOI: 10.1080/00207160.2019.1585824

3 authors:

Naraveni Rajashekar
Indian Institute of Technology Delhi

Sudhakar Chaudhary
Institute of Infrastructure Technology Research And Management

V V K Srinivas Kumar
Indian Institute of Technology Delhi

International Journal of Computer Mathematics

ISSN: 0020-7160 (Print) 1029-0265 (Online) Journal homepage: https://www.tandfonline.com/loi/gcom20

Minimization techniques for $p(x)$-Laplacian problem using WEB-Spline based mesh-free method

Rajashekar Naraveni, Sudhakar Chaudhary & V.V.K. Srinivas Kumar

To cite this article: Rajashekar Naraveni, Sudhakar Chaudhary & V.V.K. Srinivas Kumar (2019): Minimization techniques for $p(x)$-Laplacian problem using WEB-Spline based mesh-free method, International Journal of Computer Mathematics, DOI: 10.1080/00207160.2019.1585824

To link to this article: https://doi.org/10.1080/00207160.2019.1585824

Accepted author version posted online: 22 Feb 2019.
Published online: 12 Mar 2019.

INTERNATIONAL JOURNAL OF COMPUTER MATHEMATICS
https://doi.org/10.1080/00207160.2019.1585824

Taylor & Francis
Taylor & Francis Group

ARTICLE

Minimization techniques for $p(x)$-Laplacian problem using WEB-Spline based mesh-free method

Rajashekar Naraveni[a], Sudhakar Chaudhary[b,*] and V.V.K. Srinivas Kumar[a]

[a]Department of Mathematics, Indian Institute of Technology, Delhi, India; [b]Department of Mathematics, SGT University, Gurgaon, India

ABSTRACT

In this paper, we propose minimization techniques with WEB-Spline based mesh-free finite element method for the solution of $p(x)$-Laplacian problem. The WEB-Spline method uses weighted extended B-splines on a regular grid as basis functions and does not require any mesh generation which eliminates a difficult, time-consuming preprocessing step and accurate approximations are possible with relatively low-dimensional subspaces. We perform some numerical experiments to demonstrate the efficiency of the WEB-Spline method.

ARTICLE HISTORY
Received 22 August 2017
Revised 19 December 2018
Accepted 12 February 2019

KEYWORDS
Meshless method; WEB-Splines; steepest descent method; Quasi-Newton method; $p(x)$-Laplacian

2010 AMS SUBJECT CLASSIFICATIONS
65N12; 65N22; 65N30; 41A15; 35Q93

1. Introduction

In this paper, we study the following $p(x)$-Laplacian problem with homogeneous Dirichlet boundary condition:

$$\begin{aligned} -\nabla\cdot(|\nabla u(x)|^{p(x)-2}\nabla u(x)) = f(x) \quad &\text{in} \quad x\in\Omega, \\ u(x) = 0 \quad &\text{on} \quad x\in\partial\Omega, \end{aligned} \tag{1}$$

where Ω is a bounded open subset of $\mathbb{R}^d$ ($d=2$ or 3) with Lipschitz continuous boundary. Here $p:\Omega\to(1,\infty)$ is a measurable function and $1/p(x)$ is globally log-Hölder continuous [13] with $1<p_1\le p(x)\le p_2<\infty$.

Note that the $p(x)$-Laplacian is the extension of classical Laplacian for $p(x)\equiv 2$ and p-Laplacian for $p(x)\equiv p$ with $1<p<\infty$. It has applications in image processing and in the modelling of electrorheological fluids [8,16,24].

In the literature, there are many results related to existence and uniqueness of solution for $p(x)$-Laplacian problem (cf. [1,14,15] and references therein) but there are only few results towards the numerical solution (cf. [11,12] for variable exponent case and [4,20,26] for $p(x)=p$ constant case) and these are based on usual finite element methods (mesh-based methods). Recently Chaudhary et al. in [7] have proposed the WEB-Spline-based mesh-free finite element approximation of the p-Laplacian problem with homogeneous Dirichlet boundary condition. In this work, authors provide numerical errors in $L^2(\Omega)$ norm only by using WEB-Spline of degree 2.

CONTACT Rajashekar Naraveni snaraveni@gmail.com
*Sudhakar Chaudhary is presently working in The Department of Mathematics, Jaypee Institute of Information Technology, Noida, Uttar Pradesh, India

In engineering applications, the description of geometry and mesh generation process are often bottlenecks. Hence meshless methods as a tool attract a great attention for solving complex boundary value problems in science and engineering. Over the past decade, essentially two B-spline-based techniques have been developed, namely finite element methods with WEB-Splines (Weighted Extended B-splines) and isogeometric analysis [19,21]. Authors in [10,17] provide a comprehensive treatment of the theory for isogeometric methods and WEB-Spline methods. Which technique is best suited for a particular problem depends on the topological form of the simulation domain and its representation. Weighted methods are a good choice for problems with natural (Neumann) boundary conditions or if the part of the boundary, where essential (Dirichlet) boundary conditions are prescribed, has a convenient implicit description. Isogeometric methods can handle domains well which are parametrized over rectangles or cuboids or which can be expressed as union of few such parametrizations, e.g. NURBS representation for CAD/CAM applications. There are also problems where a combination of both techniques might be useful [18].

Our aim in this paper is to use some minimization techniques with the WEB-Spline method in order to get numerical solution of $p(x)$-Laplacian problem. The reason we choose the WEB-Spline method is that in this method no mesh generation is required and accurate approximations are possible with relatively low-dimensional subspaces. In this method, hierarchical bases permit adaptive refinement and smoothness can be chosen arbitrarily (cf. [17,19]). To the best of our knowledge, this is the first attempt when a mesh free method is used for solving $p(x)$-Laplacian problem.

The rest of the paper is organized as follows. In Section 2, we describe the WEB-Spline method. In Section 3, we provide the results for the well-posedness of the $p(x)$-Laplacian problem. In Section 4, we discuss variational formulation and minimization techniques for solving $p(x)$-Laplacian problem. We devote Section 5 to the numerical results of the problem and in Section 6, we give conclusions of the whole paper.

2. WEB-Spline procedure

The standard uniform B-spline of degree n is defined by the recursion

$$b^n(x) = \frac{x}{n} b^{n-1}(x) + \frac{n+1-x}{n} b^{n-1}(x-1), \tag{2}$$

starting from b^0, the characteristic function of the unit interval between 0 and 1 [17]. Figure 1 shows the uniform B-splines of degrees 1, 2 and 3. These are also known as linear, quadratic and cubic B-splines respectively. In this paper, we use the following notational conventions [17]. For functions f and g, we write $f \preceq g$ if $f \le cg$ with a constant c which does not depend on the grid width h, indices or arguments of functions. The symbols $\succeq$ and $\asymp$ are defined analogously. We describe the procedure for constructing the WEB-Spline basis which are used in finite element analysis.

A d-variate uniform B-spline of degree $n \ge 1$ and grid width h is a product of scaled translates of univariate cardinal B-splines:

$$b_k(x) = \prod_{m=1}^{d} b\Big((x_m/h) - k_m\Big),$$

with $k = (k_1, \dots, k_d) \in \mathbb{Z}^d$, $\quad x = (x_1, \dots, x_d) \in \mathbb{R}^d$ and b univariate B-spline with knots $0, \dots, n+1$. b_k is nonnegative, $(n-1)$-times continuously differentiable function with support as

$$\operatorname{supp} \quad b_k = kh + [0, n+1]^d h,$$

on each grid cell $Q_l = lh + [0,1]^d h$, where $l = (l_1, \dots, l_d) \in \mathbb{Z}^d$. On each grid cell Q_l, b_k is a polynomial of degree n in each variable.

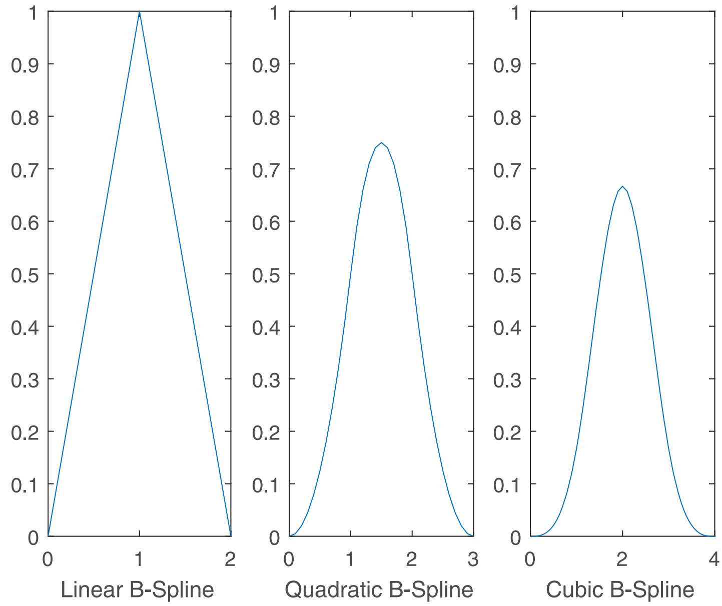

Figure 1. *B*-splines of different orders.

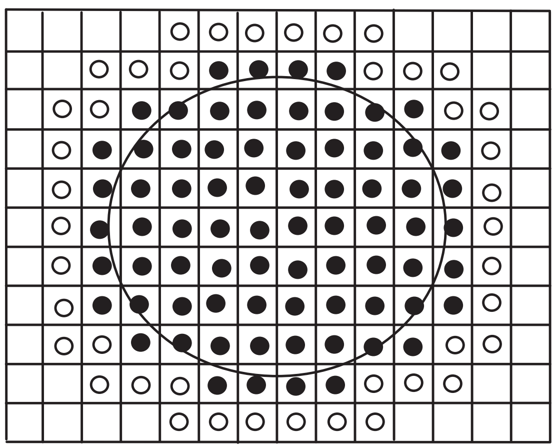

Figure 2. Relevant *B*-splines for the circular domain, marked at the centre of their supports, inner *B*-splines with dots and outer ones with circles for the degree $n = 2$.

In two dimension for $k \in K := \{\mathbf{l} \in \mathbb{Z}^2 : \mathrm{supp}(b_\mathbf{l}) \cap \Omega \neq \phi\}$ (the relevant index set for Ω), if $\mathrm{supp}(b_\mathbf{k})$ has at least one grid cell completely inside Ω then $b_\mathbf{k}$ is named as inner B-spline, otherwise it is outer. The corresponding subsets of K are I and J (cf. Figure 2): $K = I \cup J$. Splines provide suitable finite element subspace for problems with natural boundary conditions. Essential boundary conditions, however, must be incorporated into the spline space. This can be done via suitable weight functions. The idea of using weight functions to represent essential boundary conditions is due to Kantorowitsch and Krylow [22] and has been intensively investigated by Rvachev et al. in [25].

Definition 2.1 ([17]): A weight function w of order $\gamma \in \mathbb{N}_0$ is continuous on $\overline{\Omega}$ and satisfies

$$w(x) \asymp \mathrm{dist}(x, \Gamma)^\gamma, \quad x \in \Omega,$$

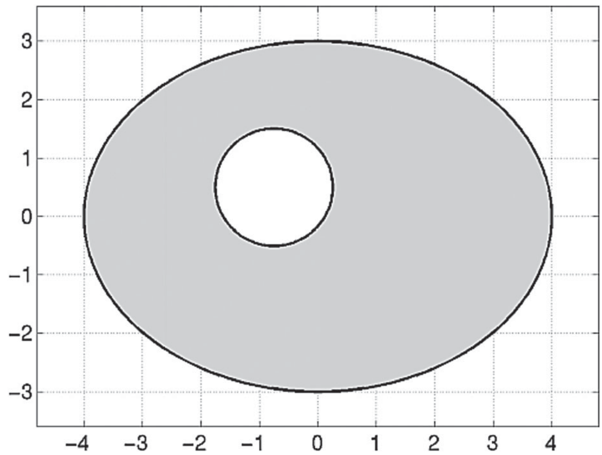

Figure 3. Planar domain bounded by an ellipse and a circle.

for a subset Γ of $\partial\Omega$. We assume that Γ has positive $(m-1)$-dimensional measure and sufficiently regular, so that the distance function has a bounded gradient. If w is smooth and vanishes linearly on the entire boundary $(\gamma = 1)$, it is called a standard weight function. The order $\gamma = 0$ corresponds to the trivial weight function $w = 1$.

In the following we describe, by way of example, two different techniques for constructing weight functions. Here we consider a planar domain bounded by an ellipse and a circle and it is shown in Figure 3.

The boundary of the domain is represented in implicit form, i.e. with the following two functions.

$$\text{Ellipse} : e(x,y) = 1 - \frac{x^2}{16} - \frac{y^2}{9},$$
$$\text{Circle} : k(x,y) = \left(x + \frac{3}{4}\right)^2 + \left(y - \frac{1}{2}\right)^2 - 1.$$

The function $e(x,y)$ vanishes on the ellipse and $k(x,y)$ is zero only on the circle.

Analytic weight function: Many elementary domains permit ad hoc definitions of weight functions. For the example, the product of e and k is an obvious choice,

$$w(x,y) = \left(1 - \frac{x^2}{16} - \frac{y^2}{9}\right)\left(\left(x + \frac{3}{4}\right)^2 + \left(y - \frac{1}{2}\right)^2 - 1\right).$$

Figure 4 shows weight function $w(x,y)$ on the domain, where it is strictly positive. If, as in this example, the weight function is globally defined and negative outside the domain, it can be conveniently used for in/out tests.

The construction of analytic weight functions is possible only for rather special domains, which include, however, many cases of practical interest. Below we briefly discuss, how to construct weight function for an arbitrary domain (Ω) with very complex boundaries $(\partial\Omega)$.

R-Function method: R-function proposed by V. L. Rvachev [25] is defined as a signed weight function that is a globally defined continuous function which is positive on domain Ω and negative on the compliment of $\overline{\Omega}$. The purpose of the R-function method is to construct weight functions for a domain with complex boundary. Through this method, we approximate the domain by

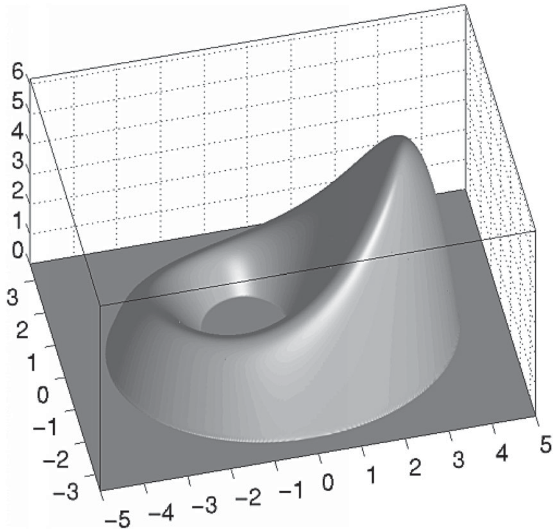

Figure 4. Analytic weight function.

many other known primitives (those domains whose weight function is not very difficult to construct) through the boolean operations. Examples of such primitives are disks, polygons, ellipses, etc. Once we have weight function for the primitives, we can find the weight function for arbitrary domain by combining the weight functions of the primitives through set operations. If r corresponds to boolean set operations, then set operations and corresponding R-functions are given as follows.

Set operations	Corresponding R-functions
Compliment: Ω^c	$r_c(w) = -w$
Intersection: $\Omega_1 \cap \Omega_2$	$r_\cap(w_1, w_2) = w_1 + w_2 - \sqrt{w_1^2 + w_2^2}$
Union: $\Omega_1 \cup \Omega_2$	$r_\cup(w_1, w_2) = w_1 + w_2 + \sqrt{w_1^2 + w_2^2}$

For example, the weight function

$$w(x, y) = e(x, y) + k(x, y) - (e^2(x, y) + k^2(x, y))^{1/2}$$

is shown in Figure 5 for the domain in the Figure 3. More details about R-function method can be found in [17,25].

Now, the wb-space $B_w^h := \text{span}\{wb_k : k \in K\}$ spanned by weighted B-splines, is a possible finite element subspace for Dirichlet boundary value problems yielding optimal order approximations. But, the condition number of the Galerkin matrix can become extremely large. This is due to the outer B-splines which have only very small support inside Ω. One might think that these basis functions do not contribute much to the approximation power and can simply be omitted. Unfortunately, this is not the case. A suitable solution to the problem of controlling the unstable outer B-splines is provided by adjoining them appropriately with the inner B-splines. This is done in such a way that the approximation power of the finite element subspace is retained. This procedure of adjoining outer B-splines with inner B-splines is known as extension procedure [17].

Combining the ideas of weight function and extension procedure with B-splines gives rise to the following definition of weighted extended B-splines (WEB-Splines).

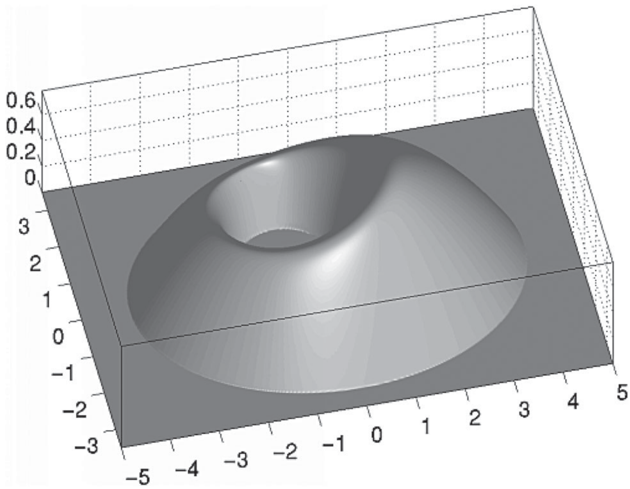

Figure 5. Rvachev's R-function type weight function.

Definition 2.2 ([17,19]): For each $\mathbf{i} \in I$, weighted extended B-spline (WEB-Spline) $B_\mathbf{i}$ is defined as

$$B_\mathbf{i} := \frac{w}{w(x_\mathbf{i})}\Big[b_\mathbf{i} + \sum_{\mathbf{j}\in J} e_{\mathbf{i},\mathbf{j}} b_\mathbf{j}\Big], \tag{3}$$

where $x_\mathbf{i}$ denotes the centre of a grid cell $Q_{\mathbf{i}+\mathbf{l}}$, $l \in \mathbb{Z}^d$ which is completely inside the domain Ω. The coefficients $e_{\mathbf{i},\mathbf{j}}$ satisfy

$$|e_{\mathbf{i},\mathbf{j}}| \preceq 1, \quad e_{\mathbf{i},\mathbf{j}} = 0 \quad \text{for } ||\mathbf{i} - \mathbf{j}|| \succeq 1$$

and are chosen so that all weighted polynomials (wp) of degree n are contained in the WEB-space $B_h := \text{span}\{B_\mathbf{i} : \mathbf{i} \in I\}$.

An explicit expression for the coefficients $e_{i,j}$ is given in the following theorem [19].

Theorem 2.1 ([17,19]): *For an outer index* $\mathbf{j} \in J$*, let* $I(\mathbf{j}) = \mathbf{l} + \{0, \ldots, n\}^m \subset I$ *be an m- dimensional array of inner indices closest to* $\mathbf{j}$ *assuming that h is small enough so that such an array exists. Then the coefficients*

$$e_{\mathbf{i},\mathbf{j}} = \prod_{\nu=1}^{m} \prod_{\mu=0;\quad l_\nu+\mu\neq \mathbf{i}_\nu}^{n} \frac{\mathbf{j}_\nu - l_\nu - \mu}{\mathbf{i}_\nu - l_\nu - \mu}$$

are admissible for constructing WEB-Splines according to Definition 2.2.

More details about the construction of WEB-Spline basis and its properties are given in [17,19].

3. Variational formulation of $p(x)$-Laplacian problem

First we define some generalized Lebesgue–Sobolev spaces which are required for the analysis of $p(x)$-Laplacian problem. Let $L^{p(x)}(\Omega)$ be the variable exponent Lebesgue space and it is defined as

$$L^{p(x)}(\Omega) = \{u | u \quad \textit{is a measurable real valued function}, \int_\Omega |u(x)|^{p(x)} \mathrm{d}x < \infty\}$$

with the norm

$$||u||_{L^{p(x)}(\Omega)} = \inf_{\eta>0}\left\{\eta : \int_\Omega \left|\frac{u(x)}{\eta}\right|^{p(x)} \mathrm{d}x \leq 1\right\}. \tag{4}$$

It can be shown that $L^{p(x)}(\Omega)$ is separable reflexive Banach space [13]. The norm $||.||_{p(x)}$ defined in Equation (4) is known as Luxemburg norm.

Further we denote $W^{1,p(x)}(\Omega)$ the variable exponent Sobolev space, which is defined by

$$W^{1,p(x)}(\Omega) = \{u \in L^{p(x)}(\Omega)|\nabla u \in L^{p(x)}(\Omega)\}$$

with the norm

$$||u||_{W^{1,p(x)}(\Omega)} = ||u||_{L^{p(x)}(\Omega)} + ||\nabla u||_{L^{p(x)}(\Omega)}.$$

Now we give the definition of $W_0^{k,p(x)}(\Omega)$.

Definition 3.1 ([13]): Let p be a measurable exponent and $k \in \mathbb{N}$. The Sobolev space $W_0^{k,p(x)}(\Omega)$ with zero boundary values is the closure of the set of $W^{k,p(x)}(\Omega)$ functions with compact support, i.e.

$$\{u \in W^{k,p(x)}(\Omega) : u = u\chi_K \quad \text{for a compact} \quad K \subset \Omega\}$$

in $W^{k,p(x)}(\Omega)$, where χ_K represents characteristic function on K.

Therefore, the variational formulation of problem (1) is given as follows. Given $f \in L^{p'(x)}(\Omega)$ (where $p'(x)$ denotes the dual variable exponent of $p(x)$), find $u \in V_1 = W_0^{1,p(x)}(\Omega)$
such that

$$a(u, v) = (f, v), \tag{5}$$

where

$$a(u, v) = \int_\Omega (|\nabla u|^{p(x)-2}\nabla u) \cdot \nabla v, \quad \forall v \in V_1 \tag{6}$$

and (f, v) denotes the duality pairing between $W_0^{1,p(x)}(\Omega)$ and its dual space.

Define an operator K from V_1 to V_1' (which is dual of V_1) such that

$$(Ku, v) = \int_\Omega (|\nabla u|^{p(x)-2}\nabla u) \cdot \nabla v, \quad \forall v \in V_1. \tag{7}$$

Also problem (5) is equivalent to the following minimization problem [4,9], namely

$$u = \operatorname*{arg\,min}_{v\in W_0^{1,p(x)}(\Omega)} J(v), \tag{8}$$

where

$$J(v) := \int_\Omega \frac{|\nabla v|^{p(x)}}{p(x)} - \int_\Omega f v.$$

For all $v, w \in W_0^{1,p(x)}(\Omega)$, first and second derivatives of the functional J are given by

$$J'(u)(v) = \int_\Omega |\nabla u|^{p(x)-2}\nabla u \cdot \nabla v - \int_\Omega f v \tag{9}$$

and

$$J''(u)(v, w) = \int_\Omega (p(x) - 2)|\nabla u|^{p(x)-4}(\nabla u \cdot \nabla w)(\nabla u \cdot \nabla v) + \int_\Omega |\nabla u|^{p(x)-2}\nabla w \cdot \nabla v, \tag{10}$$

respectively.

Now we discuss the well-posedness of the weak formulation (5). For this, first we recall some basic definitions.

Let X be a normed linear space and X' be the dual of X. An operator $L : X \to X'$ satisfies

$$\langle L(u) - L(v), u - v \rangle \geq 0, \tag{11}$$

for any $u, v \in X$ is called a monotone operator and the operator is said to be strictly monotone if for $u \neq v$ the strict inequality holds in (11). An operator L is called strongly monotone if there exists a constant $C > 0$ such that

$$\langle L(u) - L(v), u - v \rangle \geq C\|u - v\|_X^2, \quad \forall u, v \in X.$$

Moreover, the operator L is coercive if $\lim_{||u||_X \to \infty}((Lu, u)/||u||_X) = \infty$ and also is hemicontinuous if the map $t \mapsto (L(u+t v), w)$ is continuous on [0,1]. For the existence and uniqueness of (5), we use following theorem due to Browder.

Theorem 3.1: *Let $L : X \to X'$ be an operator from a reflexive Banach space X to its dual space X' which is bounded, monotone, hemicontinuous and coercive on the space X. Then the equation $L(u) = f$ has at-least one solution $u \in X$ for each $f \in X'$. Moreover, if L is strongly monotone on X, then solution is unique.*

It is obvious that K is continuous and bounded. Also for any $\xi, \eta \in \mathbb{R}^d$, we have the following inequalities (cf. [15] and references therein):

for $1 < p(x) < 2$,

$$\left((|\xi|^{p(x)-2}\xi - |\eta|^{p(x)-2}\eta)(\xi - \eta)\right) \cdot (|\xi|^{p(x)} + |\eta|^{p(x)})^{(2-p(x))/p(x)} \geq (p(x) - 1)|\xi - \eta|^{p(x)} \tag{12}$$

for $p(x) \geq 2$,

$$\left((|\xi|^{p(x)-2}\xi - |\eta|^{p(x)-2}\eta)(\xi - \eta)\right) \geq \left(\frac{1}{2}\right)^{p(x)} |\xi - \eta|^{p(x)}. \tag{13}$$

From the above inequalities, we get strict monotonicity of K.

Since

$$\lim_{\|u\| \to \infty} \frac{(Ku, u)}{\|u\|_{V_1}} = \lim_{\|u\| \to \infty} \frac{\int_\Omega |\nabla u|^{p(x)} dx}{\|u\|_{V_1}} = \infty, \tag{14}$$

therefore K is coercive.

By the above properties, operator K defined in (7) is strictly monotone, hemicontinuous and coercive on $W_0^{1,p(x)}(\Omega)$. Therefore existence and uniqueness of the solution of the problem (5) follows from Theorem 3.1.

4. Discretization of $p(x)$-Laplacian problem using WEB-Spline basis

Let $V_{1h} = span\{B_i : i \in I\}$ be the finite dimensional subspace of $W_0^{1,p(x)}(\Omega)$ where I is the set of all indices of inner B-splines and B_i 's are web-splines of degree one.

The WEB-Spline based finite element approximation is then to seek $u_h \in V_{1h}$ such that

$$a(u_h, v_h) = (f, v_h), \quad \forall v_h \in V_{1h}. \tag{15}$$

Let N be the number of inner indices for approximating u_h in V_{1h}. Since V_{1h} is a finite dimensional space, any $u_h \in V_{1h}$ can be written as

$$u_h = \sum_{\mathbf{i}=1}^{N} \psi_{\mathbf{i}} \Phi_{\mathbf{i}}, \quad \psi_{\mathbf{i}} \in \mathbb{R}, \tag{16}$$

where $\Phi_1, \Phi_2, \ldots \Phi_N$ are weighted extended B-spline functions with the coefficients $\psi_{\mathbf{i}}$.

Well-posedness of (15) follows in a similar way as the well-posedness of continuous problem. Also it can be shown in the same way as in the continuous case that problem (15) is equivalent to the following minimization problem:

$$u_h = \arg\min_{v_h \in V_{1h}} J(v_h). \tag{17}$$

Several minimization techniques have been used to solve Equation (17) but in our work we mainly solve this minimization problem by steepest descent and quasi-Newton methods.

4.1. Steepest descent method

First we discuss the steepest descent method for the $p(x)$-Laplacian problem [20] using the WEB-Spline method. For this, let $v_h, w_h \in V_{1h}$ and $\|\cdot\|_{V_{1h}}$ be a norm on V_{1h}. Now we define steepest descent direction (normalized) w_h such that

$$J'(v_h)w_h = -\|J'(v_h)\|_*, \quad \|w_h\|_{V_{1h}} = 1, \tag{18}$$

where $\|\cdot\|_*$ is the dual norm of V_{1h} defined by

$$\|J'(v_h)\|_* = \sup_{u_h \in V_{1h}} \frac{|J(v_h)(u_h)|}{\|u_h\|_{V_{1h}}}. \tag{19}$$

Let u be the exact solution of (8) and u_h^m be the approximate solution of our minimization problem at mth iteration. Then the approximate solution in the next iteration, i.e. u_h^{m+1}, is given by

$$u_h^{m+1} = u_h^m + \alpha_m w_h^m, \tag{20}$$

where α_m can be determined by either backtracking line search method [6] or by the exact line search method, i.e.

$$J(u_h^m + \alpha_m w_h^m) = \min_{\alpha \geq 0} J(u_h^m + \alpha w_h^m). \tag{21}$$

Further, the steepest descent direction w_h^m is defined by

$$\int_\Omega \nabla w_h^m \nabla v_h = -J'(u_h^m)(v_h) = -\int_\Omega |\nabla u_h^m|^{p(x)-2} \nabla u_h^m \nabla v_h + \int_\Omega f v_h, \quad \forall v_h \in V_{1h}. \tag{22}$$

Now we show that w_h^m in fact is the steepest descent direction at $u = u_h^m$. By (9), we have

$$J'(u_h^m)(v_h) = \int_\Omega |\nabla u_h^m|^{p(x)-2} \nabla u_h^m \nabla v_h - \int_\Omega f v_h \tag{23}$$

$$= \int_{\Omega} (|\nabla u_h^m|^{p(x)-2} \nabla u_h^m - |\nabla u|^{p(x)-2} \nabla u) \nabla v_h. \tag{24}$$

Since w_h^m is the Riesz representation of the functional $-J'(u_h^m)$ in the space V_{1h}, so

$$\|w_h^m\|_{V_{1h}} = \|J'(u_h^m)\|_*, \tag{25}$$

and

$$J'(u_h^m)(w_h^m) = -\|w_h^m\|_{V_{1h}}^2 = -\|J'(u_h^m)\|_* \|w_h^m\|_{V_{1h}}. \tag{26}$$

As suggested in [20], one may equip V_{1h} with a weighted norm $|\cdot|_{u_h^m}^2 = \int_\Omega (\epsilon + |\nabla u_h^m|^{p(x)-2}) |\nabla \cdot|^2$ for $p(x) > 2$ case and $|\cdot|_{u_h^m}^2 = \int_\Omega (\epsilon + |\nabla u_h^m|)^{p(x)-2} |\nabla \cdot|^2$ for the case $p(x) < 2$. The introduction of the small parameter ϵ is to handle the possible degeneracy when $\nabla u_h^m = 0$. With this information, we define the descent direction w_h^m as follows.

For $p(x) > 2$, find w_h^m such that

$$\int_\Omega (\epsilon + |\nabla u_h^m|^{p(x)-2}) \nabla w_h^m \nabla v_h = -\int_\Omega |\nabla u_h^m|^{p(x)-2} \nabla u_h^m \nabla v_h + \int_\Omega f v_h, \quad \forall v_h \in V_{1h} \tag{27}$$

and for $p(x) < 2$, find w_h^m such that

$$\int_\Omega (\epsilon + |\nabla u_h^m|)^{p(x)-2} \nabla w_h^m \nabla v_h = -\int_\Omega |\nabla u_h^m|^{p(x)-2} \nabla u_h^m \nabla v_h + \int_\Omega f v_h, \quad \forall v_h \in V_{1h}. \tag{28}$$

The generalized steepest descent method [6,20] leads to the algorithm described in Algorithm 1.

Algorithm 1: Algorithm for solving $p(x)$-Laplacian problem in the steepest descent method.

Initialize tolerance(μ), maximum iterations (l), smoothing parameter (ϵ), step length (α), continuation steps (β), function $p(x)$ and $s(x)$ where $s(x) = (p(x) - 2)/2$
Compute the initial solution u_h^0 (which is obtained by solving Poisson problem)
for $t = 1 : \beta$ *(loop over continuation steps)* **do**
    Find $st(x) = (t \quad s(x))/\beta$
    for $m = 1 : l$ *(loop over steepest descent iterations)* **do**
        Find steepest descent direction w_h^m by solving
        $\int_\Omega |\nabla u_h^m|_\epsilon^{(2st(x))} \nabla w_h^m \nabla v_h = -\int_\Omega |\nabla u_h^m|^{(2st(x))} \nabla u_h^m \nabla v_h + \int_\Omega f v_h, \quad \forall v_h \in V_{1h}$, where $|\nabla u_h^m|_\epsilon^{(2st(x))} = \epsilon + (\epsilon + |\nabla u|)^{(2st(x))}$ or $|\nabla u_h^m|_\epsilon^{(2st(x))} = \epsilon + (\epsilon^2 + |\nabla u|^2)^{st(x)}$.
        Find step length α_m based on Armijo condition using α
        Update u_h^{m+1} by $u_h^{m+1} = u_h^m + \alpha_m w_h^m$
        if $||w_h^m||/||u_h^{m+1}|| < \mu$ **then**
            break
    Repeat the iterations with initial step length (α)

4.2. *Quasi-Newton method*

Here we discuss the quasi-Newton method [6] for the $p(x)$-Laplacian problem using the WEB-Spline method.

In this method also if u_h^m be the approximate solution of our minimization problem at mth iteration, then the approximate solution in the next iteration, i.e. u_h^{m+1}, is given by

$$u_h^{m+1} = u_h^m + \alpha_m w_h^m, \tag{29}$$

where α_m can be determined by backtracking line search (Armijo rule) method as we did in the case of steepest descent method. For finding direction w_h^m in quasi-Newton method, we start with the second differential of J defined in (10). Thus for $u_h, v_h, w_h \in V_{1h}$

and for $p(x) \geq 2$, we have

$$\begin{aligned} J''(u_h)(v_h, w_h) &= \int_\Omega (p(x) - 2)|\nabla u_h|^{p(x)-4}(\nabla u_h \cdot \nabla w_h)(\nabla u_h \cdot \nabla v_h) \\ &\quad + \int_\Omega |\nabla u_h|^{p(x)-2} \nabla w_h \cdot \nabla v_h \\ &= \int_\Omega |\nabla u_h|^{p(x)-2} \left((p(x) - 2)\frac{\nabla u_h}{|\nabla u_h|} \cdot \nabla w_h \frac{\nabla u_h}{|\nabla u_h|} \cdot \nabla v_h + \nabla w_h \cdot \nabla v_h \right). \end{aligned} \tag{30}$$

This gives us

$$\begin{aligned} J''(u_h)(v_h, w_h) &= \int_\Omega |\nabla u_h|^{p(x)-2} \big((p(x) - 2)(Sgn(\nabla u_h) \cdot \nabla w_h)(Sgn(\nabla u_h) \cdot \nabla v_h) \big) \\ &\quad + \int_\Omega |\nabla u_h|^{p(x)-2} (\nabla w_h \cdot \nabla v_h), \end{aligned} \tag{31}$$

where

$$Sgn(\nabla u_h(x)) = \frac{\nabla u_h(x)}{\sqrt{(|\nabla u_h(x)|^2 + (1 - sgn(|\nabla u_h(x)|^2)))}}.$$

The above Equation (31) is well-defined and is numerically computable for the case $p(x) \geq 2$ even though $\nabla u_h(x) = 0$ somewhere.

Further J'' defined in (31) is in general not positive definite and is not defined if $p(x) < 2$ and if $\nabla u_h(x) = 0$ somewhere. In order to overcome these difficulties, we replace the term $|\nabla u_h|^{p(x)-2}$ in Equation (31) by

$$|\nabla u_h|_\epsilon^{p(x)-2} = \epsilon + (\epsilon^2 \cdot (1 - sgn(|\nabla u_h|^2)) + |\nabla u_h|^2)^{\frac{p(x)-2}{2}}, \tag{32}$$

where ϵ is a small positive parameter. Then the regularized second differential of J after modification is

$$J_\epsilon''(u_h)(v_h, w_h) = \int_\Omega |\nabla u_h|_\epsilon^{p(x)-2} ((p(x) - 2)(Sgn(\nabla u_h) \cdot \nabla w_h)(Sgn(\nabla u_h) \cdot \nabla v_h) + \nabla w_h \cdot \nabla v_h). \tag{33}$$

Now for quasi-Newton, the direction w_h^m is given by

$$B_m(w_h^m, v_h) = -J'(u_h^m)v_h, \quad \forall\, v_h \in V_{1h},$$

where $B_m(w_h^m, v_h) = J_\epsilon''(u_h^m)(w_h^m, v_h)$.

We provide the algorithm for quasi-Newton method to solve the $p(x)$-Laplacian problem in Algorithm 2.

Algorithm 2: Algorithm for solving $p(x)$-Laplacian problem in quasi-Newton method.

Initialize tolerance (μ), maximum iterations (l), smoothing parameter (ϵ), step length (α), continuation steps (β), function $p(x)$ and $s(x)$ where $s(x) = (p(x) - 2)/2$
Compute the initial solution u_h^0(which is obtained by solving Poisson problem)
for $t = 1 : \beta$ *(loop over continuation steps)* **do**
  Find $st(x) = (t \quad s(x))/\beta$
  for $m = 1 : l$ *(loop over quasi-Newton iterations)* **do**
    Find quasi-Newton direction w_h^m by solving
    $\int_\Omega |\nabla u_h^m|_\epsilon^{(2st(x))}((2st(x))(Sgn(\nabla u_h^m) \cdot \nabla v_h)(Sgn(\nabla u_h^m) \cdot \nabla w_h^m) + \nabla v_h \cdot \nabla w_h^m) = -\int_\Omega |\nabla u_h^m|^{(2st(x))} \nabla u_h^m \nabla v_h + \int_\Omega f v_h$
    Find step length α_m based on Armijo condition using α
    Update u_h^{m+1} by $u_h^{m+1} = u_h^m + \alpha_m w_h^m$
    if $||w_h^m||/||u_h^{m+1}|| < \mu$ **then**
      break
  Repeat the iterations with initial step length (α)

4.3. Approximation of the Luxemburg Norm

In this paper, we calculate errors for $p(x)$-Laplacian problem in Luxemburg norm. In order to approximate Luxemburg norm numerically, we use the idea given in [5] and it is discussed below.

If $u(x)$ is not zero almost everywhere, then the Luxemburg norm η is implicitly defined by

$$G(u,\eta) = \int_\Omega \left|\frac{u(x)}{\eta}\right|^{p(x)} \frac{\mathrm{d}x}{p(x)} - 1 = 0 \tag{34}$$

and the derivative of $G(u,\eta)$ with respect to η is given by

$$\partial_\eta G(u,\eta) = -\frac{1}{\eta}\int_\Omega \left|\frac{u(x)}{\eta}\right|^{p(x)} \mathrm{d}x. \tag{35}$$

Thus one can apply the Newton's method for finding η. For a given $u(x)$, $G(u,\eta)$ is a monotonically decreasing convex function (see [5]) in η. This guarantees the convergence of the Newton's method whenever the initial guess $\eta_0 > 0$ is chosen such that $G(u,\eta_0) \geq 0$.

In this work, we choose initial guess η_0 as follows:

$$\eta_0 = \begin{cases} \left(\displaystyle\int_\Omega \frac{|u(x)|^{p(x)}}{p(x)}\mathrm{d}x\right)^{1/p_{\max}}, & \text{if } \displaystyle\int_\Omega \frac{|u(x)|^{p(x)}}{p(x)}\mathrm{d}x \geq 1, \\ \left(\displaystyle\int_\Omega \frac{|u(x)|^{p(x)}}{p(x)}\mathrm{d}x\right)^{1/p_{\min}}, & \text{if } \displaystyle\int_\Omega \frac{|u(x)|^{p(x)}}{p(x)}\mathrm{d}x < 1, \end{cases}$$

where $p_{\max} = \max p(x)$ and $p_{\min} = \min p(x)$.

5. Numerical experiments

In this section, we performed some numerical experiments with different values of the exponent $p(x)$ in different domains for solving $p(x)$-Laplacian problem using the WEB-Spline method together with steepest descent and quasi-Newton algorithms. For all experiments, we choose degree of WEB-Spline

$n = 1$, initial guess u_h^0 as the solution to the corresponding linear problem (i.e. $p(x) = 2$ case), tolerance $\mu = 10^{-4}$ and smoothing parameter $\epsilon = 10^{-2}$ for both steepest descent and quasi-Newton algorithms. For the first example, exact solution is known while for other examples exact solution is not known. For the case when exact solution is known we calculate the error as $||u - u_h||_Y$ whereas for the case when exact solution is not known we calculate the error as $||u_h - u_{h/2}||_Y$ with $Y = L^{p(x)}(\Omega), W^{1,p(x)}(\Omega)$. The order of convergence has been calculated w.r.t. the degree of freedom. All experiments have been done in Matlab and computation time is measured in an intel i7 CORE processor.

Example 5.1: In this example, we consider $p(x)$ to be constant, i.e. $p(x) = p$. For $f = 1$ and the domain $\Omega = \{(x, y) \in \mathbb{R}^2 : ((x - \frac{1}{2})^2 + (y - \frac{1}{2})^2) < \frac{1}{4}\}$, the exact solution of the given problem for

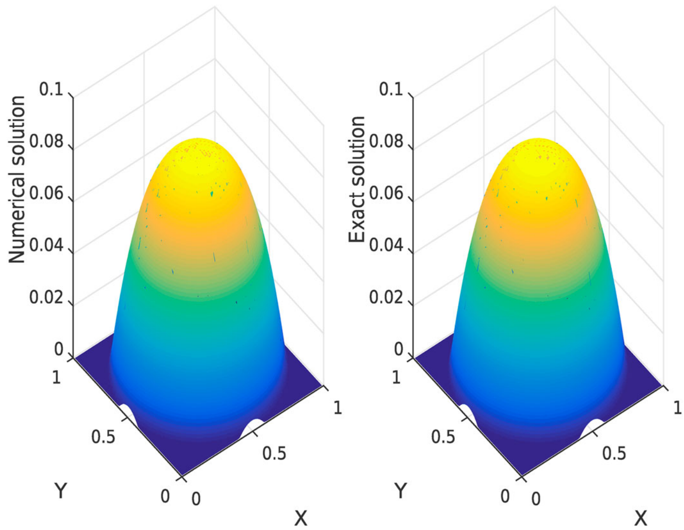

Figure 6. Numerical and exact solutions for Example 5.1, $p = 1.5$ case.

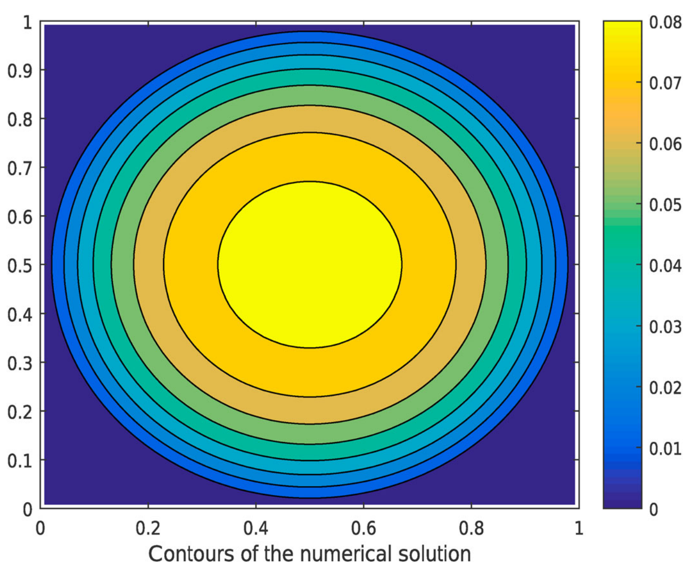

Figure 7. Contours of the numerical solution for Example 5.1, $p = 1.5$ case.

$p(x) = p$ case is given by

$$u(x,y) = \left(\frac{p-1}{p}\right)\left(\frac{1}{2}\right)^{1/(p-1)}\left(1 - 4\left((x-\frac{1}{2})^2 + (y-\frac{1}{2})^2\right)^{p/2(p-1)}\right).$$

In this first example, we provide numerical results for two different values of p, namely $p = 1.5, \quad 4$. For $p = 1.5$, the numerical and exact solutions are shown in Figure 6 and contours of the numerical solution are shown in Figure 7.

Errors, rate of convergence in $W^{1,p}(\Omega)$ norm for both steepest descent and quasi-Newton algorithms are given in Tables 1 and 2 respectively. For $p = 4$, the numerical and exact solutions are shown in Figure 8 and contours of the numerical solution are shown in Figure 9. Errors, rate of convergence in $W^{1,1}(\Omega)$ norm for both steepest descent and quasi-Newton algorithms are given in Tables 3 and 4 respectively. In [2], authors have shown theoretically the linear rate of convergence in $W^{1,p}(\Omega)$ norm for the case $p < 2$ and in $W^{1,1}(\Omega)$ norm for the case $p > 2$. In this example, our numerical results confirm with the theoretical estimates of [2].

Table 1. Errors using the steepest descent method for Example 5.1, $p = 1.5$ case.

Grid width h	DOF	No. of iterations	CPU time	$W^{1,p}(\Omega)$ error	$W^{1,p}(\Omega)$ rate
$\frac{1}{2^2}$	25	11	1.32 s	3.4724×10^{-1}	
$\frac{1}{2^3}$	81	11	3.02 s	1.1779×10^{-1}	0.7798
$\frac{1}{2^4}$	289	11	7.91 s	3.3711×10^{-2}	0.9024
$\frac{1}{2^5}$	1089	11	23.92 s	8.2152×10^{-3}	1.0184

Table 2. Errors using the quasi-Newton method for Example 5.1, $p = 1.5$ case.

Grid width h	DOF	No. of iterations	CPU time	$W^{1,p}(\Omega)$ error	$W^{1,p}(\Omega)$ rate
$\frac{1}{2^2}$	25	4	0.47 s	3.4741×10^{-1}	
$\frac{1}{2^3}$	81	3	0.83 s	1.1811×10^{-1}	0.7781
$\frac{1}{2^4}$	289	3	2.19 s	3.4055×10^{-2}	0.8971
$\frac{1}{2^5}$	1089	3	6.55 s	8.5043×10^{-3}	1.0008

Table 3. Errors using the steepest descent method for Example 5.1, $p = 4$ case.

Grid width h	DOF	No. of iterations	CPU time	$W^{1,1}(\Omega)$ error	$W^{1,1}(\Omega)$ rate
$\frac{1}{2^2}$	25	19	2.30 s	2.6722×10^{-1}	
$\frac{1}{2^3}$	81	19	5.36 s	1.6225×10^{-1}	0.3599
$\frac{1}{2^4}$	289	19	14.12 s	6.0196×10^{-2}	0.7152
$\frac{1}{2^5}$	1089	19	42.21 s	1.2650×10^{-2}	1.1252

Table 4. Errors using the quasi-Newton method for Example 5.1, $p = 4$ case.

Grid width h	DOF	No. of iterations	CPU time	$W^{1,1}(\Omega)$ error	$W^{1,1}(\Omega)$ rate
$\frac{1}{2^2}$	25	3	0.35 s	2.6699×10^{-1}	
$\frac{1}{2^3}$	81	3	0.83 s	1.6268×10^{-1}	0.3573
$\frac{1}{2^4}$	289	3	2.18 s	6.0652×10^{-2}	0.7117
$\frac{1}{2^5}$	1089	3	6.56 s	1.3135×10^{-2}	1.1035

Example 5.2: In this second example, we take $f = 1$ and the domain $\Omega = \{(x, y) \in \mathbb{R}^2 : ((x - \frac{1}{2})^2 + (y - \frac{1}{2})^2) < \frac{1}{4}\}$, with discontinuous piecewise linear exponent $p(x, y)$, namely

$$p(x,y) = \begin{cases} p^+ \quad \text{if} \quad 2x-1 < -0.01, \\ p^- + (p^- - p^+)\left(\dfrac{x-0.01}{0.02}\right) \quad \text{if } |2x-1| \le 0.01, \\ p^- \quad \text{if } 2x-1 > 0.01, \end{cases}$$

where $p^+ = 4$ and $p^- = 1.3$. For this example, numerical solution and contours of the numerical solution are shown in Figure 10. Errors in $L^{p(x)}(\Omega)$, $W^{1,p(x)}(\Omega)$ norms for steepest descent and quasi-Newton algorithms are given in Tables 5 and 6 respectively.

Example 5.3: In this third example, we take the source function f as

$$f = \begin{cases} 2 \quad \text{if} \quad 2x-1 > 0, \\ 1 \quad \text{if} \quad 2x-1 \le 0, \end{cases}$$

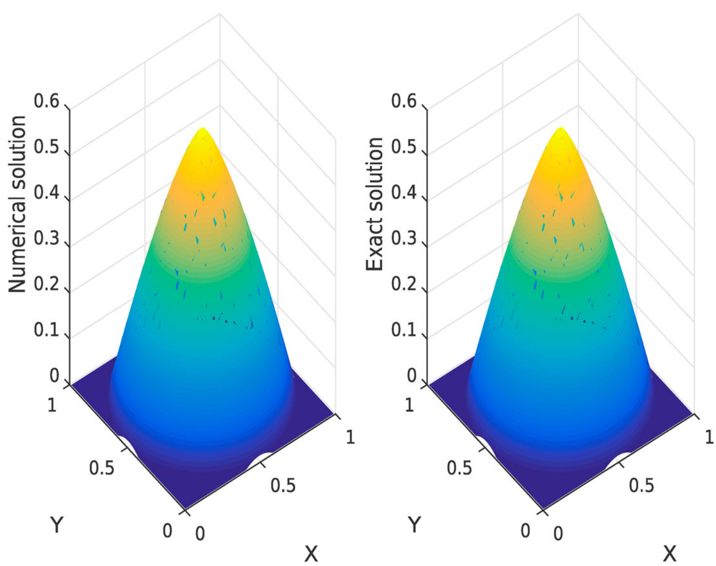

Figure 8. Numerical and exact solutions for Example 5.1, $p = 4$ case.

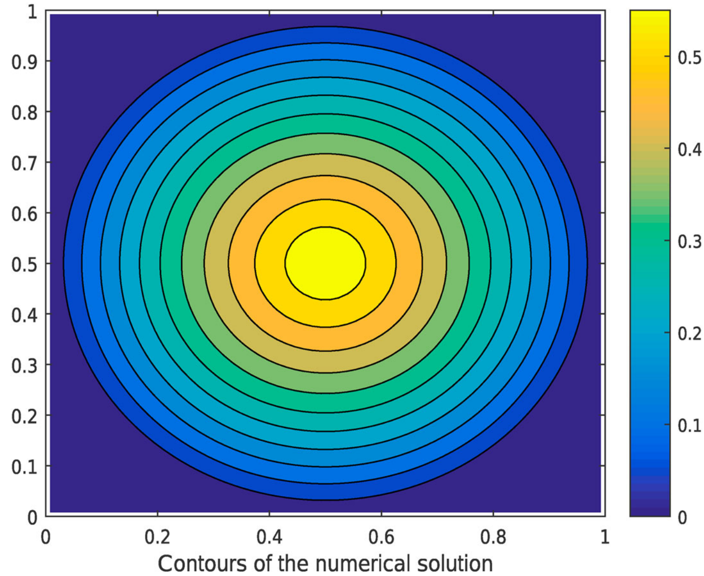

Figure 9. Contours of the numerical solution for Example 5.1, $p = 4$ case.

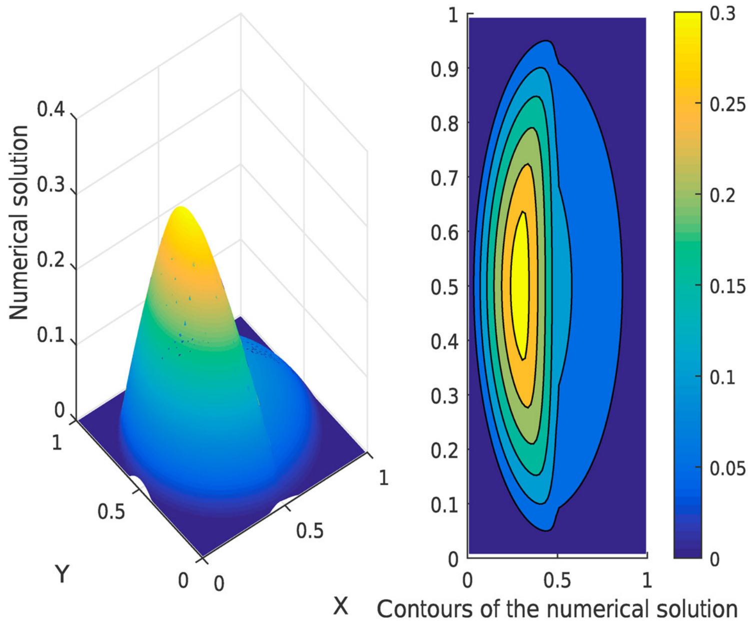

Figure 10. Numerical solution and contours of the numerical solution for Example 5.2.

and the domain $\Omega = \{(x, y) \in \mathbb{R}^2 : ((x - \frac{1}{2})^2 + (y - \frac{1}{2})^2) < \frac{1}{4}\}$ with exponent $p(x, y)$, namely

$$p(x, y) = 1 + \frac{1}{x + y + 1}.$$

For this example, numerical solution and contours of the numerical solution are shown in Figure 11. Errors in $L^{p(x)}(\Omega)$, $W^{1,p(x)}(\Omega)$ norms, rate of convergence in $W^{1,p(x)}$ norm for steepest descent and

Table 5. Errors using the steepest descent method for Example 5.2.

Grid width h	DOF	No. of iterations	CPU time	$L^{p(x)}(\Omega)$ error	$W^{1,p(x)}(\Omega)$ error
$\frac{1}{2^2}$	25	25	3.04 s	1.8077×10^{-2}	2.2109×10^{-2}
$\frac{1}{2^3}$	81	27	7.78 s	1.6070×10^{-2}	1.9041×10^{-2}
$\frac{1}{2^4}$	289	25	21.62 s	1.0811×10^{-2}	1.2492×10^{-2}
$\frac{1}{2^5}$	1089	25	57.63 s	3.3893×10^{-3}	3.5973×10^{-3}

Table 6. Errors using the quasi-Newton method for Example 5.2.

Grid width h	DOF	No. of iterations	CPU time	$L^{p(x)}(\Omega)$ error	$W^{1,p(x)}(\Omega)$ error
$\frac{1}{2^2}$	25	4	0.52 s	1.8031×10^{-2}	2.2061×10^{-2}
$\frac{1}{2^3}$	81	4	1.12 s	1.6095×10^{-2}	1.9065×10^{-2}
$\frac{1}{2^4}$	289	4	3.01 s	1.0808×10^{-2}	1.2489×10^{-2}
$\frac{1}{2^5}$	1089	5	11.34 s	3.3987×10^{-3}	3.6076×10^{-3}

quasi-Newton algorithms are given in Tables 7 and 8 respectively. As shown in [12], the linear rate of convergence in $W^{1,p(x)}(\Omega)$ norm is achieved.

Example 5.4: In this fourth example, we have taken the source function which is same as given in Example 5.3, i.e.

$$f = \begin{cases} 2 & \text{if} \quad 2x - 1 > 0, \\ 1 & \text{if} \quad 2x - 1 \leq 0, \end{cases}$$

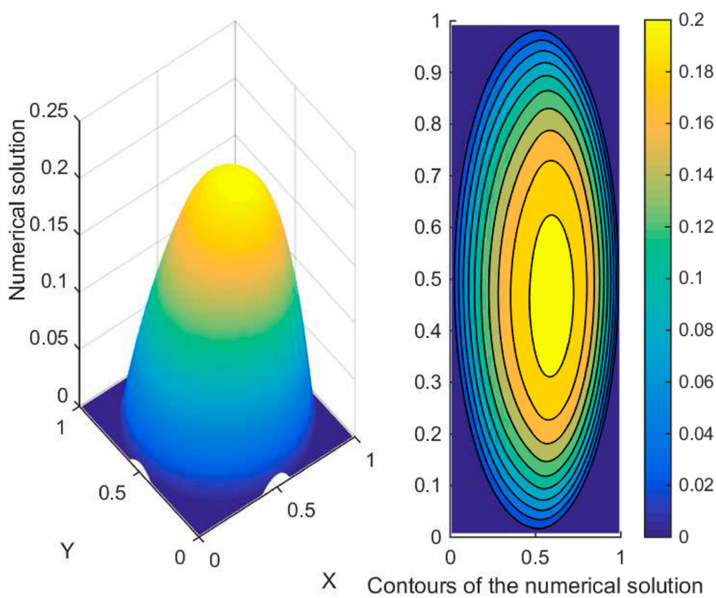

Figure 11. Numerical solution and contours of the numerical solution for Example 5.3.

and the domain $\Omega = [0,1]^2$ with exponent $p(x,y)$, namely

$$p(x,y) = 1 + \frac{1}{x+y+1}.$$

For this example, numerical solution and contours of the numerical solution are shown in Figure 12. Errors in $L^{p(x)}(\Omega)$, $W^{1,p(x)}(\Omega)$ norms for steepest descent and quasi-Newton algorithms are given in Tables 9 and 10 respectively.

Table 7. Errors using the steepest descent method for Example 5.3.

Grid width h	DOF	No. of iterations	CPU time	$L^{p(x)}(\Omega)$ error	$W^{1,p(x)}(\Omega)$ error	$W^{1,p(x)}(\Omega)$ rate
$\frac{1}{2^2}$	25	11	1.30 s	8.5591×10^{-3}	1.0431×10^{-2}	
$\frac{1}{2^3}$	81	11	3.10 s	7.0777×10^{-3}	8.7598×10^{-3}	0.1259
$\frac{1}{2^4}$	289	11	8.24 s	2.2406×10^{-3}	3.1510×10^{-3}	0.7375
$\frac{1}{2^5}$	1089	11	29.76 s	7.1588×10^{-4}	9.1728×10^{-4}	0.8901

Table 8. Errors using the quasi-Newton method for Example 5.3.

Grid width h	DOF	No. of iterations	CPU time	$L^{p(x)}(\Omega)$ error	$W^{1,p(x)}(\Omega)$ error	$W^{1,p(x)}(\Omega)$ rate
$\frac{1}{2^2}$	25	3	0.36 s	8.5576×10^{-3}	1.0429×10^{-2}	
$\frac{1}{2^3}$	81	3	0.85 s	7.0782×10^{-3}	8.7603×10^{-3}	0.1258
$\frac{1}{2^4}$	289	3	2.25s	2.2413×10^{-3}	3.1516×10^{-3}	0.7374
$\frac{1}{2^5}$	1089	3	6.85 s	7.2021×10^{-4}	9.2164×10^{-4}	0.8869

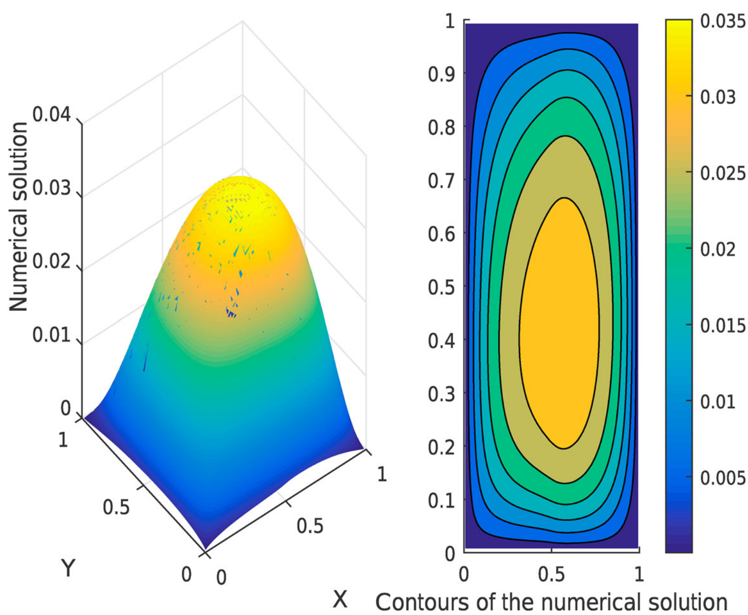

Figure 12. Numerical solution and contours of the numerical solution for Example 5.4.

Table 9. Errors using the steepest descent method for Example 5.4.

Grid width h	DOF	No. of iterations	CPU time	$L^{p(x)}(\Omega)$ error	$W^{1,p(x)}(\Omega)$ error
$\frac{1}{2^2}$	25	12	1.03 s	2.9936×10^{-2}	3.6302×10^{-2}
$\frac{1}{2^3}$	81	12	2.71 s	5.1552×10^{-3}	6.5933×10^{-3}
$\frac{1}{2^4}$	289	12	7.75 s	7.2002×10^{-4}	8.9746×10^{-4}
$\frac{1}{2^5}$	1089	12	34.02 s	5.0006×10^{-5}	7.1399×10^{-5}

Table 10. Errors using the quasi-Newton method for Example 5.4.

Grid width h	DOF	No. of iterations	CPU time	$L^{p(x)}(\Omega)$ error	$W^{1,p(x)}(\Omega)$ error
$\frac{1}{2^2}$	25	3	0.27 s	2.9927×10^{-2}	3.6291×10^{-2}
$\frac{1}{2^3}$	81	4	0.93 s	5.1536×10^{-3}	6.5913×10^{-3}
$\frac{1}{2^4}$	289	4	2.60s	7.1971×10^{-4}	8.9705×10^{-4}
$\frac{1}{2^5}$	1089	4	7.98 s	4.8628×10^{-5}	6.9963×10^{-5}

6. Conclusion

In this paper, we propose the WEB-Spline based mesh-free finite element method together with steepest descent and quasi-Newton algorithms for solving $p(x)$-Laplacian problem. There are many papers in the literature which discuss about the numerical solution of $p(x)$-Laplacian problem using usual finite element (mesh-based) method but to the best of our knowledge, this is the first attempt to use mesh-free method (based on WEB-Spline) for solving $p(x)$-Laplacian problem. We have calculated errors in $W^{1,p}(\Omega)$, $L^{p(x)}(\Omega)$ and $W^{1,p(x)}(\Omega)$ norms for different values of p and $p(x)$. For handling the degeneracy and singularity of the $p(x)$-Laplacian problem, we have borrowed the idea given in [6]. In all experiments, we have observed that quasi-Newton method is taking less CPU time and less number of iterations as compared to the steepest descent method.

Theoretical estimates available in the literature [2,12] for the order of convergence are confirmed via numerical experiments. This work forms a basis for time-dependent p-Laplacian problems and p-biharmonic problems [3,23].

Acknowledgments

The authors would like to thank the referees for thoroughly reading the manuscript and their valuable suggestions.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

First author is thankful to University Grants Commission, India, for their financial support through Senior Research Fellowship (ID No. 421249).

References

[1] C.O. Alves, *Existence of solution for a degenerate $p(x)$-Laplacian equation in $\mathbb{R}^N$*, J. Math. Anal. Appl. 345 (2008), pp. 731–742.

[2] J.W. Barrett and W.B. Liu, *Finite element approximation of the p-Laplacian*, Math. Comp. 61(204) (1993), pp. 523–537.

[3] J.W. Barrett and W.B. Liu, *Finite element approximation of the parabolic p-Laplacian*, SIAM J. Numer. Anal., 31 (1994), pp. 413–428.
[4] R. Bermejo and J.-A. Infante, *A multigrid algorithm for the p-Laplacian*, SIAM J. Sci. Comput. 21(5) (2000), pp. 1774–1789.
[5] M. Caliari and S. Zuccher, *The inverse power method for the $p(x)$-Laplacian problem*, J. Sci. Comput. 65 (2015), pp. 698–714.
[6] M. Caliari and S. Zuccher, *Quasi-Newton minimization for the $p(x)$-Laplacian problem*, J. Comput. Appl. Math 309 (2017), pp. 122–131.
[7] S. Chaudhary, V. Srivastava, V.V.K. Srinivas Kumar and B. Srinivasan, *WEB-Spline based mesh-free finite element approximation for p-Laplacian*, Int. J. Comput. Math. 93(6) (2016), pp. 1022–1043.
[8] Y. Chen, S. Levine and M. Rao, *Variable exponent, linear growth functionals in image restoration*, SIAM J. Appl. Math. 66(4) (2006), pp. 1383–1406.
[9] P.G. Ciarlet, *The Finite Element Method for Elliptic Problems*, Amsterdam, North-Holland, 1978.
[10] J.A. Cottrell, T.J.R. Hughes and Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA*, Wiley, Chichester, UK, 2009.
[11] L.M. Del Pezzo, A.L. Lombardi and S. Martnez, *Interior penalty discontinuous Galerkin FEM for the $p(x)$-Laplacian*, SIAM J. Numer. Anal. 50(5) (2012), pp. 2497–2521.
[12] L.M. Del Pezzo and S. Martnez, *Order of convergence of the finite element method for the $p(x)$-Laplacian*, IMA J. Numer. Anal. 35(4) (2015), pp. 1864–1887.
[13] L. Diening, P. Harjulehto, P. Hasto and M. Ruzicka, *Lebesgue and Sobolev Spaces with Variable Exponents*, Lect. Notes Math., vol. 2017, Springer,, 2011.
[14] X. Fan, *Existence and uniqueness for the $p(x)$-Laplacian Dirichlet problems*, Math. Nachr. 284(11–12) (2011), pp. 1435–1445.
[15] X. Fan and Q.H. Zhang, *Existence of solutions for $p(x)$-Laplacian Dirichlet problems*, Nonlinear Anal. 52 (2003), pp. 1843–1852.
[16] P. Harjulehto, P. Hästö, U. Le and M. Nuortio, *Overview of differential equations with non-standard growth*, Nonlinear Anal. 72 (2010), pp. 4551–4574.
[17] K. Höllig, *Finite Element Methods with B-Splines*, SIAM, Philadelphia, 2003.
[18] K. Höllig, J. Hörner and A. Hoffacker, Finite element analysis with B-splines: Weighted and isogeometric methods, in: J.-D. Boissonnat, P. Chenin, A. Cohen, C. Gout, T. Lyche, M. -L. Mazure, L. Schumaker (Eds.), Curves and Surfaces, Lect. Notes Computer Science, vol. 6920, Springer, Berlin, Heidelberg, 2012, pp. 330–350..
[19] K. Höllig, U. Reif and J. Wipper, *Weighted extended B-spline approximation of Dirichlet problems*, SIAM J. Numer. Anal. 39 (2001), pp. 442–462.
[20] Y.Q. Huang, R. Li and W. Liu, *Preconditioned descent algorithms for p-Laplacian*, J. Sci. Comput. 32(2) (2007), pp. 343–371.
[21] T.J.R. Hughes, J.A. Cottrell and Y. Bazilevs, *Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement*, Comput. Methods Appl. Mech. Engrg. 194 (2005), pp. 4135–4195.
[22] L.W. Kantorowitsch and W.I. Krylow, *Näherungsmethoden der Höheren Analysis*, VEB Deutscher Verlag der Wissenschaften, Berlin, 1956.
[23] T. Pryer, *Discontinuous Galerkin methods for the p-Biharmonic equation from a discrete variational perspective*, Electron. Trans. Numer. Anal., 41 (2014), pp. 328–349.
[24] K. Rajagopal and M. Rika, *On the modeling of electrorheological materials*, Mech. Res. Commun. 23(4) (1996), pp. 401–407.
[25] V.L. Rvachev and T.I. Sheiko, *R-functions in boundary value problems in mechanics*, Appl. Mech. Rev. 48 (1995), pp. 151–188.
[26] G. Zhou, Y. Huang and C. Feng, *Preconditioned hybrid conjugate gradient algorithm for p-Laplacian*, Int. J. Numer. Anal. Model. 2 (2005), pp. 123–130.