NONLINEAR ELLIPTIC PROBLEMS WITH THE METHOD OF FINITE VOLUMES

SANJAY KUMAR KHATTRI

Received 21 February 2006; Revised 30 May 2006; Accepted 13 June 2006

We present a finite volume discretization of the nonlinear elliptic problems. The discretization results in a nonlinear algebraic system of equations. A Newton-Krylov algorithm is also presented for solving the system of nonlinear algebraic equations. Numerically solving nonlinear partial differential equations consists of discretizing the nonlinear partial differential equation and then solving the formed nonlinear system of equations. We demonstrate the convergence of the discretization scheme and also the convergence of the Newton solver through a variety of practical numerical examples.

1. Introduction

Nonlinear elliptic equations arise in many applications in many fields, so solving such systems is important. Solving nonlinear partial differential equations consists of discretizing the partial differential equations and solving the formed nonlinear algebraic system of equations. Work has been done on numerically solving nonlinear elliptic partial differential equations (PDEs). For example, Schwarz alternating methods (see [14] and references therein), multigrid methods [5], and preconditioned FFT [12]. In this work, we explore the convergence of the discretization method, and also the convergence behaviour of the Newton-Krylov method for solving the nonlinear algebraic equations [7, 8]. Let us consider the following nonlinear elliptic problem:

$$-\operatorname{div}(\mathbf{K}\,\operatorname{grad} p) + f(p) = s(x,y) \quad \text{in } \Omega, \tag{1.1}$$

$$\begin{aligned} p(x,y) &= p^D \quad \text{on } \partial\Omega_D, \\ \mathbf{g}(x,y) &= -\mathbf{K}\nabla p \quad \text{on } \partial\Omega_N. \end{aligned} \tag{1.2}$$

The above problem captures the fundamental features of the Poisson-Boltzmann equation arising in molecular biophysics (see [2–6]). Here, Ω is a domain in $\mathbb{R}^2$, the source

Hindawi Publishing Corporation
Differential Equations and Nonlinear Mechanics
Volume 2006, Article ID 31797, Pages 1–16
DOI 10.1155/DENM/2006/31797

function s is assumed to be in $L^2(\Omega)$, and the diagonal tensor coefficient $\mathbf{K}(x, y)$ is positive definite and piecewise constant. $\mathbf{K}$ is allowed to be discontinuous in space. In biophysics literature, the medium properties $\mathbf{K}$ is referred to as the permittivity [2–5]. It takes the values of the appropriate dielectric constants in the different regions of the domain Ω. In (1.1) and (1.2), $\partial\Omega_D$ and $\partial\Omega_N$ represent the Dirichlet and Neumann parts of the boundary, respectively. $f(p)$ represents the nonlinear part of the problem. Equations (1.1) and (1.2) model a wide variety of processes with practical applications, some examples are pattern formation in biology, viscous fluid flow phenomena, chemical reactions, biomolecule electrostatics, and crystal growth.

This paper presents the two-point finite volume discretization (2P-FVM) [9, 10] of the nonlinear problem (1.1) and (1.2) on the rectangular meshes. An implementation of boundary conditions is also presented. Several numerical examples are reported for showing the convergence of the finite volume discretization scheme. A Newton-Krylov algorithm is also mentioned for solving the system of nonlinear equations formed by the discretization scheme. Convergence of the Newton method is demonstrated through numerical work. For higher-order finite volume discretization of linear problems, the interested reader are referred to [1, 11]. Handling complex geometries is a difficult task; radial basis functions (RBFs) are a new numerical method that can offer very high accuracy even on complicated domains [13]. RBFs can be promising in solving nonlinear partial differential equations.

An outline of the paper is as follows. In Section 2, finite volume discretization of the nonlinear elliptic equations is presented. An implementation of Neumann and Dirichlet boundary conditions is also reported. Section 3 presents a Newton-Krylov algorithm for solving the system of nonlinear equations. Numerical work is reported in Section 4. Finally Section 5 concludes the paper.

2. Two-point finite volume discretization of the nonlinear elliptic problem

For solving partial differential equations (PDEs) on a domain by numerical methods such as the 2P-FVM, the domain is divided into smaller good quality elements (meshing of the domain). These elements are called finite volumes or cells. The degrees of freedom (DOFs) for the 2P-FVM lie at the cell centers. Thus, each finite volume in the mesh gives rise to an algebraic nonlinear equation corresponding to (1.1). A residual form of (1.1) is $-\operatorname{div}(\mathbf{K}\operatorname{grad} p) + f(p) - s(x, y) = 0$. Integrating it over one of the finite volumes V in the mesh and using the Gauss divergence theorem lead to

$$-\int_{\partial V} \mathbf{K}\nabla p \cdot \hat{\mathbf{n}} + \int_V f(p) - \int_V s = 0, \tag{2.1}$$

where $\hat{\mathbf{n}}$ is the outward unit normal on the boundary (∂V) of the cell V. Let us assume that finite volumes V are rectangular or quadrilateral in shape. Thus, the boundary of these finite volumes consists of four segments and the above equation can be written as

$$-\sum_{i=1}^{4} \int_{\partial V_i} \mathbf{K}\nabla p \cdot \hat{\mathbf{n}} + \int_V f(p) - \int_V s = 0. \tag{2.2}$$

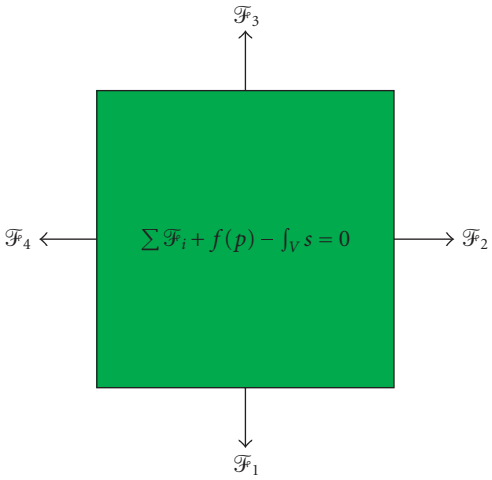

FIGURE 2.1. A nonlinear discrete equation for a finite volume. Here, $\mathscr{F}_i$ with $i = 1,\ldots,4$ are the fluxes through the cell boundaries.

The term $\int_{\partial V_i} -\mathbf{K}\nabla p \cdot \hat{\mathbf{n}}$ is referred to as the flux through the edge ∂V_i. Let us denote it by $\mathscr{F}_i$. Thus, (2.2) can be written as

$$\sum_{i=1}^{4} \mathscr{F}_i + \int_V f(p) - \int_V s = 0. \tag{2.3}$$

The degrees of freedom for the 2P-FVM lie at the cell centers, so the scalar variable p is assumed constant in each cell. Thus, the integral $\int_V f(p)$ is approximated as $\int_V f(p) \approx f(p)V$. Here, V is the area of the cell. Thus, the above equation can be written as

$$\sum_{i=1}^{4} \mathscr{F}_i + f(p)V - \int_V s = 0. \tag{2.4}$$

For evaluating the integral $\int_V s$, we use the Newton-Cotes formulas. Each finite volume in the mesh results in the nonlinear equation (2.4). Figure 2.1 shows the equation for a cell. Collecting all such nonlinear equations will result in a discrete system of nonlinear equations $\mathbf{A}(\mathbf{p_h}) = 0$. Now let us consider computing $\mathscr{F}_i$ in (2.4). Figure 2.2 shows two cells P and E. Let us compute the flux across the common edge AB of the cells. Let the $\mathbf{K}$ of the cells P and E be $\mathrm{K}_P\mathbf{I}$ and $\mathrm{K}_E\mathbf{I}$. Here, $\mathbf{I}$ is the identity matrix. Flux across the edge AB by the 2P-FVM is given as [10]

$$\mathscr{F}_{AB} = \Phi_{12}(p_E - p_P), \tag{2.5}$$

where Φ_{12} is a scalar and is given as

$$\Phi_{12} = \mathrm{K}_P\mathrm{K}_E\left(\frac{l}{h_1 h_2}\right)\frac{1}{(\mathrm{K}_P/h_1 + \mathrm{K}_E/h_2)}. \tag{2.6}$$

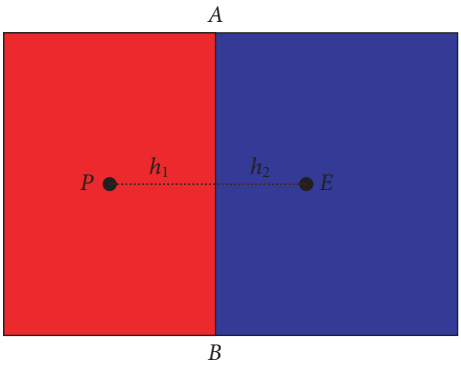

FIGURE 2.2. Flux $\mathcal{F}_{AB}$ through the interface AB shared by the cells P and E; see (2.5).

Here, l is the length of the common edge AB. h_1 is the perpendicular distance from the center of the cell P on the edge AB. Similarly h_2 is defined. Let there be total n cells in the mesh (DOF $= n$). Each cell in the mesh provides a nonlinear equation (2.4). Collecting all such equations results in a system of nonlinear equations given as

$$\mathbf{A}(\mathbf{p}) = \begin{pmatrix} A_1(\mathbf{p}) \\ A_2(\mathbf{p}) \\ \vdots \\ A_n(\mathbf{p}) \end{pmatrix}. \tag{2.7}$$

The above nonlinear system of equations can be solved by a Newton-Krylov method [8]. In Section 3, a Newton-Krylov algorithm is presented for solving the nonlinear system $\mathbf{A}(\mathbf{p}) = 0$.

2.1. Implementation of boundary conditions. In the case of finite volume discretization, every finite volume in the mesh results in a nonlinear discrete equation (2.3). It requires computing flux through boundary edges. Thus, for handling boundary cells, boundary conditions are converted into an equivalent flux expression. Flux or Neumann boundary condition can be easily implemented and is more accurate than Dirichlet boundary condition. Since we convert Dirichlet boundary condition into flux, in this conversion we loose accuracy.

Since flux through an edge is given as $\int -\mathbf{K}\nabla p \cdot \hat{\mathbf{n}}$, thus computation of flux across an edge requires computation of scalar potential gradient (∇p). So, let us write an expression for the gradient of the scalar potential p. Let the potential at the three vertices of Figure 2.3(b) be p_1, p_2, and p_3. Assuming that the potential is varying linearly inside the triangle, the constant gradient of the potential (∇p) in Figure 2.3(b) is

$$\nabla p = \frac{-1}{2\Omega} \sum_{i=1}^{3} p_i \mathbf{n_i}. \tag{2.8}$$

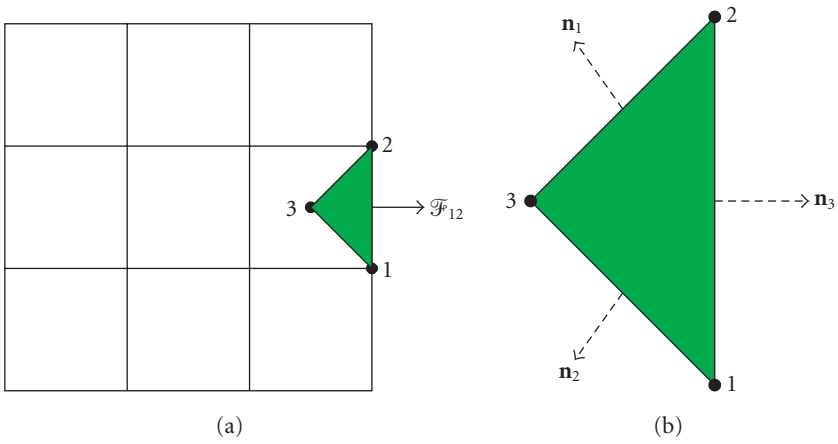

FIGURE 2.3. Implementation of the Dirichlet boundary condition. (a) A 3×3 mesh. Pressure is specified at the boundary points 1 and 2. Flux ($\mathscr{F}_{12}$) through the edge 12 is expressed as a linear combination of the potentials at the locations 1, 2, and 3; see (2.10). (b) Boundary triangle. Here, $\mathbf{n}_i$ with $i = 1,\ldots,3$ are the normal vectors on the edges.

Here, Ω is the area of the triangle, and $\mathbf{n}_i$ is the normal vector on the edge opposite to the vertex i. The magnitude of the vector $\mathbf{n}_i$ is equal to the length of the edge.

2.1.1. Dirichlet boundary condition. Figure 2.3(a) shows a 3×3 mesh. Let the pressure be specified at the boundary points 1 and 2. For applying the finite volume formulation (2.4) to the boundary cell 3, we have to compute the flux ($\mathscr{F}_{12}$) through the boundary edge 12. For computing the flux, let us form a boundary triangle 123 as shown in Figure 2.3(b). Let the unknown potential at the center of the boundary cell 3 be p_3. The potential gradient inside the boundary triangle can be approximated by the expression (2.8). Thus, the flux through the boundary edge 12 is $F_{12} = -(\mathbf{K}\nabla p) \cdot \mathbf{n}_3$. Let the outward normal vector on the edge (see Figure 2.3(b)) opposite to the vertex i be $\mathbf{n}_i = (nx_i, ny_i)^t$. The vector $\mathbf{n}_i$ is pointing away from the node i and the magnitude of the vector is equal to the length of the edge. Let the property $\mathbf{K}$ of the boundary cell 3 be

$$\mathbf{K} = \begin{pmatrix} kx & 0 \\ 0 & ky \end{pmatrix}. \tag{2.9}$$

Substituting the values of $\mathbf{K}$ and ∇p (given by (2.8)) in the equation $\mathscr{F}_{12} = -(\mathbf{K}\nabla p) \cdot \mathbf{n}_3$ results in

$$\mathscr{F}_{12} = -\frac{1}{2\Omega}\left[kx\left(\sum_{i=1}^{3} p_i\, nx_i\right) nx_3 + ky\left(\sum_{i=1}^{3} p_i\, ny_i\right) ny_3 \right]. \tag{2.10}$$

Here, Ω is the absolute value of the area of the boundary triangle 123.

2.1.2. Flux boundary condition. Implementation of Neumann or flux boundary condition is even simpler. Flux across a boundary edge will be added with the source term $\int_V s$ in the discrete nonlinear equation (2.4).

3. Newton-Krylov algorithm

The nonlinear algebraic system of equations $\mathbf{A}(\mathbf{p})$ can be expanded by the Taylor's series around an initial guess $\mathbf{p}_0$ as

$$\mathbf{A}(\mathbf{p}) = \mathbf{A}(\mathbf{p_0}) + \boldsymbol{J}(\mathbf{p}_0)\Delta\mathbf{p} + \text{HOT}, \tag{3.1}$$

where $\mathbf{J}$ is the Jacobian matrix, HOT exists for higher-order terms, $\mathbf{J}(\mathbf{p}_0)$ is the value of the Jacobian matrix at the initial guess $\mathbf{p}_0$, and the difference vector $\Delta\mathbf{p} = \mathbf{p} - \mathbf{p}_0$. The Jacobian $\mathbf{J}$ is an $n \times n$ (n is the DOF) linear system. The Jacobian $\mathbf{J}$ is given as

$$\mathbf{J} = \begin{pmatrix} \dfrac{\partial A_1}{\partial p_1} & \dfrac{\partial A_1}{\partial p_2} & \cdots & \dfrac{\partial A_1}{\partial p_n} \\ \dfrac{\partial A_2}{\partial p_1} & \dfrac{\partial A_2}{\partial p_2} & \cdots & \dfrac{\partial A_2}{\partial p_n} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial A_n}{\partial p_1} & \dfrac{\partial A_n}{\partial p_2} & \cdots & \dfrac{\partial A_n}{\partial p_n} \end{pmatrix}. \tag{3.2}$$

Here, $p_1, p_2, \ldots, p_n$ are the potential associated with the n cells in the mesh. The Jacobian is symmetric, and it will be positive definite and diagonal dominant for positive nonlinearities, that is, for $f > 0$. Setting (3.1) equal to zero and neglecting the higher-order terms results in a basis for the Newton algorithm

$$\mathbf{J}(\mathbf{p}_0)\Delta\mathbf{p} = -\mathbf{A}(\mathbf{p}_0). \tag{3.3}$$

The above linear system is a basis for the Newton algorithm for finding the zeros of the nonlinear function $\mathbf{A}(\mathbf{p})$. The linear system (3.3) is solved by the conjugate gradient solver [15, chapter 5]. The Newton iteration for solving $\mathbf{A}(\mathbf{p}) = 0$ is

$$\begin{aligned} \mathbf{J}(\mathbf{p}_k)\Delta\mathbf{p}_k &= -\mathbf{A}(\mathbf{p}_k), \\ \mathbf{p}_{k+1} &= \mathbf{p}_k + \Delta\mathbf{p}_k \quad k = 0, 1, 2, \ldots, m. \end{aligned} \tag{3.4}$$

A Newton-Krylov iteration for solving $\mathbf{A}(\mathbf{p}) = 0$ is given by Algorithm 3.1.

In Algorithm 3.1, $\|\cdot\|_{L_2}$ denotes the discrete L_2 norm and $\max_{\text{iter}}$ is the maximum allowed Newton's iterations. It is interesting to note the stopping criteria in Algorithm 3.1, we are using three stopping criteria in the algorithm. Apart from the maximum allowed iterations, we are using L_2 norm of residual vector ($\|\mathbf{F}(\mathbf{p})\|_{L_2}$) and also L_2 norm of difference in scalar potential vector ($\|\Delta\mathbf{p}\|_{L_2}$) as stopping criteria for the algorithm. Generally in the literature maximum allowed iterations and the residual vector are used as stopping

Mesh the domain;
Form the nonlinear system: $\mathbf{F}(\mathbf{p})$;
Set the Newton iteration counter: $k = 0$;
While $k \leq \max_{\text{iter}}$ or $\|\Delta\mathbf{p}\|_{L_2} \geq \text{tol}$ or $\|\mathbf{A}(\mathbf{p})\|_{L_2} \geq \text{tol}$ *do*
  Solve the discrete system: $\mathbf{J}(\mathbf{p}_k)\Delta\mathbf{p} = -\mathbf{A}(\mathbf{p}_k)$ with a tolerance 1.0×10^{-10};
  $\mathbf{p_{k+1}} = \mathbf{p_k} + \Delta\mathbf{p}$;
  k^{++};
end

Algorithm 3.1. Quasi-Newton-Krylov algorithm.

criteria. If the Jacobian is singular, then the residual vector alone cannot provide a robust stopping criterion. We have implemented the algorithm in the C^{++} language.

4. Numerical examples

Let $\mathbf{p}$ be the exact solution vector and let $\mathbf{p}_h$ be the finite volume solution vector on a mesh. Let us further assume that p^k denotes the exact solution at the center of the cell k and p_h^k denotes the discrete solution by the finite volume approximation for the same location. The error in the L_∞ norm is defined as

$$\left|\left|\mathbf{p} - \mathbf{p}_h\right|\right|_{L_\infty} := \max_{k \in \text{cells}}\left[\, \left| p^k(x) - p_h^k(x) \right| \,\right], \tag{4.1}$$

and error in the L_2 norm is defined as

$$\left|\left|\mathbf{p} - \mathbf{p}_h\right|\right|_{L_2} := \left(\sum_{\text{cells}} \left[p^k(x) - p_h^k(x) \right]^2 \Omega_k \right)^{1/2}. \tag{4.2}$$

Here, Ω_k is the area of the finite volume k in the mesh.

For solving the Jacobian system, we use the conjugate gradient (CG) solver with the ILU preconditioner. For all the numerical examples, our stopping criteria in Algorithm 3.1 are $\text{tol} = 10 \times 10^{-20}$ and $\max_{\text{iter}} = 20$. The tolerance for the CG solver is 1×10^{-10}. During numerical experiments, we observed that the algorithm does not converge for all initial guess of p as it is expected with Newton's iteration.

Example 4.1. Let us solve the following problem:

$$-\triangle p + \gamma p e^p = f \quad \text{in } \Omega, \tag{4.3}$$

$$p(x, y) = p^D \quad \text{on } \partial\Omega. \tag{4.4}$$

Assume $\mathbf{K} = \mathbf{I}$ and the nonlinear function $f = \gamma p e^p$ in (1.1). Let the exact solution be $p = x(x-1)y(y-1)$ and $\gamma = 1.0$. Let the domain of definition be $\Omega = (0,1) \times (0,1)$. Solution inside the domain is enforced by the source term and the Dirichlet boundary condition. The initial guess for starting the Newton-Krylov algorithm is the zero vector. For understanding the convergence behaviour of the finite volume method, we performed

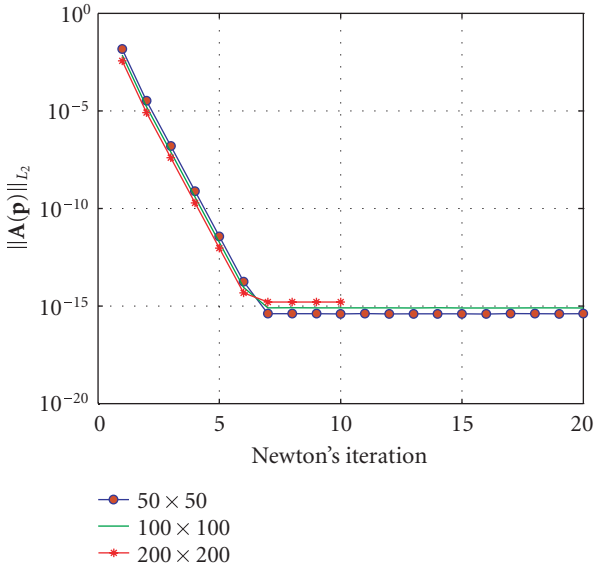

FIGURE 4.1. Example 4.1: Newton's iteration versus the residual norm $\|\mathbf{A}(\mathbf{p})\|_{L_2}$ on various meshes.

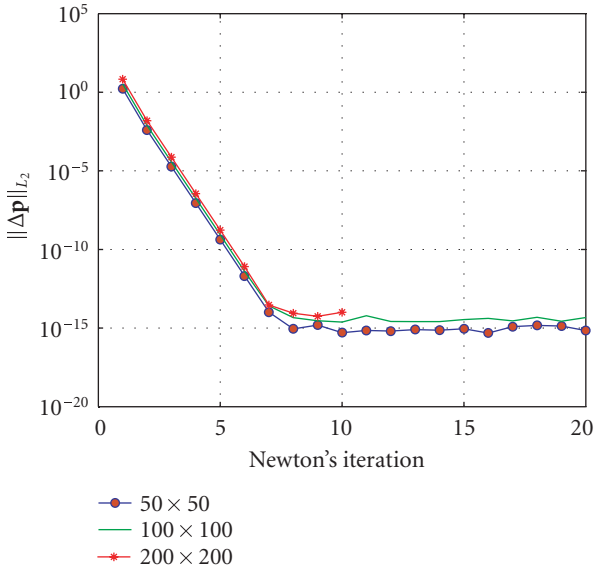

FIGURE 4.2. Example 4.1: Newton's iteration versus norm of the difference vector $\|\Delta\mathbf{p}\|_{L_2}$ on various meshes.

numerical work on the three meshes: 50×50, 100×100, and 200×200. Figures 4.1, 4.2, 4.3, and 4.4 report the outcome of our numerical work. Figures 4.1 and 4.2 present

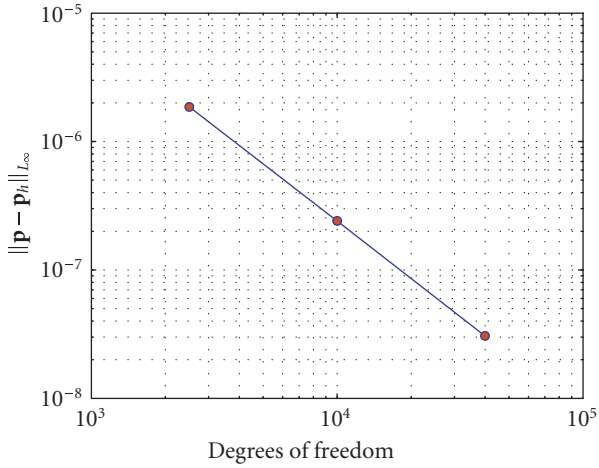

FIGURE 4.3. Example 4.1: convergence of the discretization scheme. L_∞ error versus the degrees of freedom in the mesh. We are observing $\|\mathbf{p}-\mathbf{p}_h\|_{L_\infty} \approx Ch^{2.90}$. Here h denotes the size of the smallest edge in the mesh.

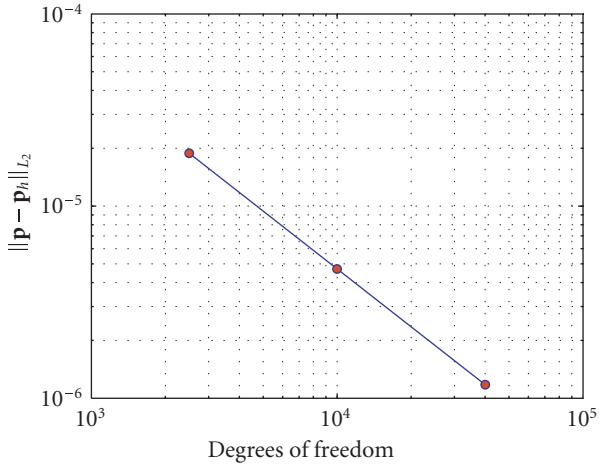

FIGURE 4.4. Example 4.1: convergence of the discretization scheme. L_2 error versus the degrees of freedom in the mesh. We are observing $\|\mathbf{p}-\mathbf{p}_h\|_{L_2} \approx Ch^{1.99}$. Here h denotes the size of the smallest edge in the mesh.

convergence of the Newton method while Figures 4.3 and 4.4 report the convergence of the finite volume method.

It is interesting to notice in Figures 4.1 and 4.2 that for the three meshes, 5-6 Newton iterations are sufficient for reducing the residuals $\|\mathbf{A}(\mathbf{p})\|_{L_2}$ and $\|\Delta\mathbf{p}\|_{L_2}$ to about 1×10^{-15}. The convergence of $\|\mathbf{A}(\mathbf{p})\|_{L_2}$ and $\|\Delta\mathbf{p}\|_{L_2}$ is asymptotically quadratic. Figures 4.3 and 4.4

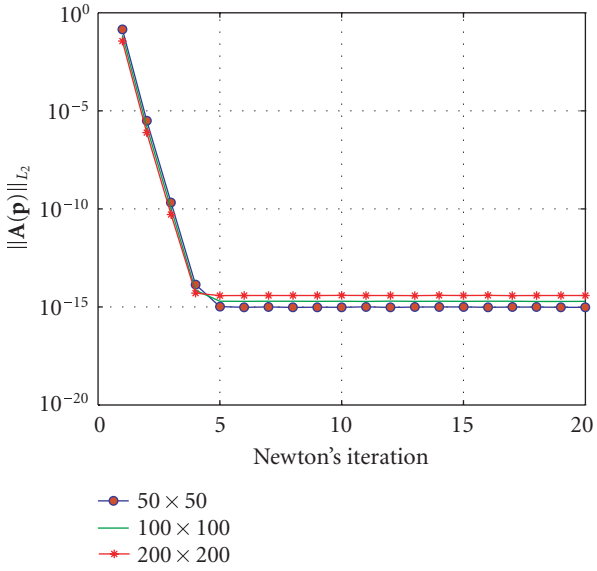

FIGURE 4.5. Example 4.2: Newton's iteration versus the residual norm $\|\mathbf{A}(\mathbf{p})\|_{L_2}$ on various meshes.

report the convergence behaviour of the 2P-FVM method for the nonlinear elliptic equation. It can be seen in these figures that the method is converging as the mesh is refined. In the L_2 and L_∞ norms, the method is super convergent.

Example 4.2. In this experiment, we solve (4.5) in $\Omega = (0,1) \times (0,1)$. Let the exact solution be $p = (x^2 - x^2)\sin(3\pi y)$. Solution inside the domain is enforced by the Dirichlet boundary condition and the source term. We assume zero initial guess for the Newton-Krylov algorithm:

$$-\nabla^2 p + k\sinh(p) = f \quad \text{in } \Omega, \tag{4.5}$$

$$p(x,y) = p^D \quad \text{on } \partial\Omega. \tag{4.6}$$

Again, we performed experiments on the three meshes. Figures 4.5, 4.6, 4.7, and 4.8 report outcome of our experiments. Figures 4.5 and 4.6 demonstrate the convergence of the Newton-Krylov Algorithm 3.1. It is interesting to see that for all three meshes, 5-6 iterations of Algorithm 3.1 are sufficient. Convergence of the 2P-FVM method is presented in Figures 4.7 and 4.8. It can be seen in these figures that the method is superconvergent (convergence rate ≥ 2.0).

Now let us consider the examples where the medium properties are discontinuous. For applications, see [2–6] and references therein.

Example 4.3. In this experiment, the exact solution is not known. We are solving simplified Poisson-Boltzmann equation (4.7) on $\Omega = (-1,1) \times (-1,1)$ with $k = 1.0$ and source

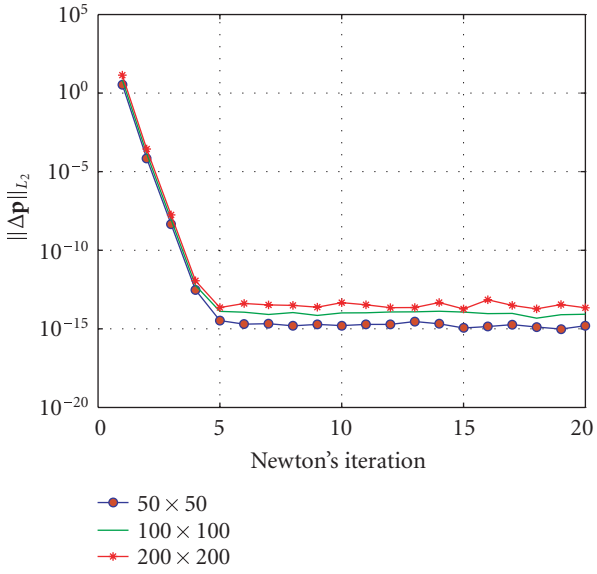

FIGURE 4.6. **Example 4.2:** Newton's iteration versus norm of the difference vector $\|\Delta \mathbf{p}\|_{L_2}$ on various meshes.

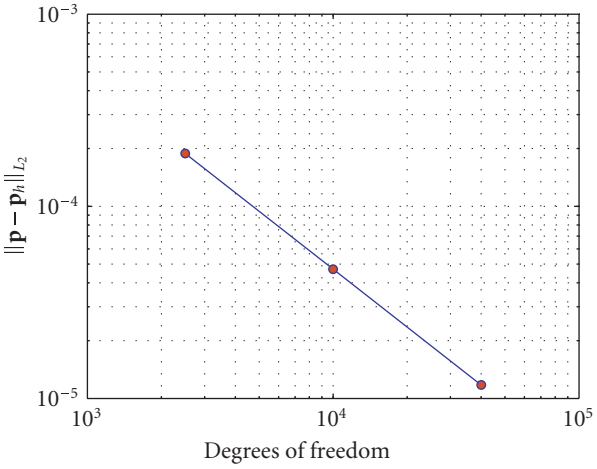

FIGURE 4.7. **Example 4.2:** convergence of the discretization scheme. L_2 error versus the degrees of freedom in the mesh. We are observing $\|\mathbf{p} - \mathbf{p}_h\|_{L_2} \approx Ch^{2.00}$.

$f = 0$. The domain Ω is divided into four equal subdomains (Ω_i, $i = 1, \ldots, 4$) as shown in Figure 4.9 based on the medium properties $\boldsymbol{\epsilon}$; see Figure 4.10 for the distribution of properties in the domain. Let $\boldsymbol{\epsilon}_1 = \boldsymbol{\epsilon}_3 = 100.0\mathbf{I}$ and $\boldsymbol{\epsilon}_2 = \boldsymbol{\epsilon}_4 = 1.0\mathbf{I}$. $\boldsymbol{\epsilon}_1$, $\boldsymbol{\epsilon}_2$, $\boldsymbol{\epsilon}_3$, and $\boldsymbol{\epsilon}_4$ refer to the medium properties in the subdomains Ω_1, Ω_2, Ω_3, and Ω_4, respectively.

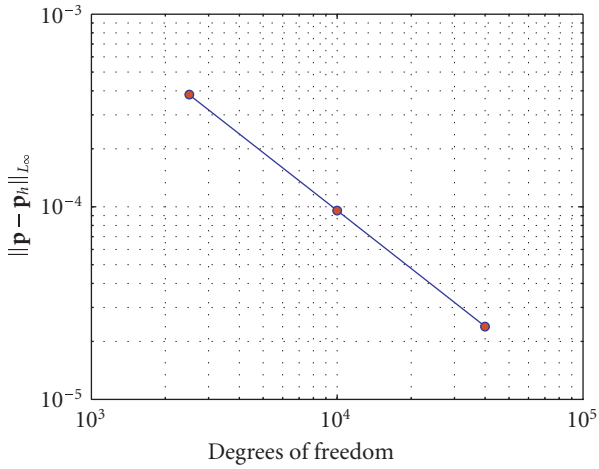

FIGURE 4.8. Example 4.2: convergence of the discretization scheme. L_∞ error versus the degrees of freedom in the mesh. We are observing $\|\mathbf{p}-\mathbf{p}_h\|_{L_\infty} \approx Ch^{1.990}$.

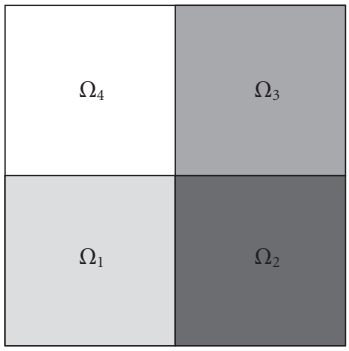

FIGURE 4.9. Example 4.3: domain is divided into four subdomains Ω_i, $i = 1,\ldots,4$. The property $\boldsymbol{\epsilon}$ in the subdomain Ω_i is $\boldsymbol{\epsilon}_i$.

Figure 4.11 is showing the surface plot of the discrete solution on a 64×64 mesh. There is a singularity in the solution at origin as it can be expected for the interface problems [10]. Figure 4.12 shows the convergence of the Newton-Krylov Algorithm 3.1. Again we see that 5-6 Newton's iterations are sufficient. It should be noted here that problems with discontinuous $\boldsymbol{\epsilon}$ can produce badly conditioned Jacobian systems:

$$-\operatorname{div}(\boldsymbol{\epsilon}\operatorname{grad} p) + k\sinh(p) = f \quad \text{in } \Omega, \tag{4.7}$$

$$p(x,y) = x(x-1)y(y-1) \quad \text{on } \partial\Omega. \tag{4.8}$$

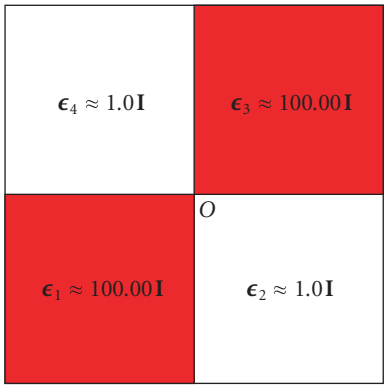

FIGURE 4.10. Example 4.3: property distribution in $\Omega = (-1,1) \times (-1,1)$.

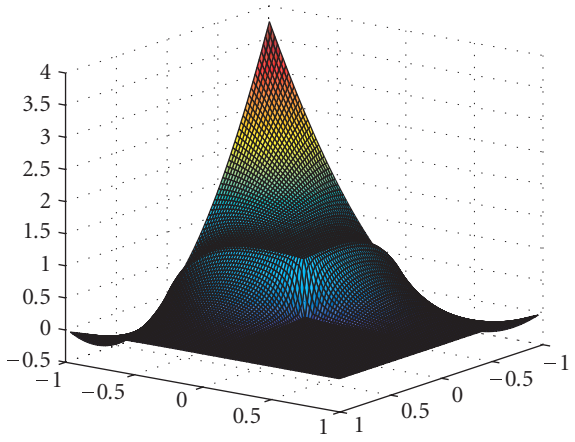

FIGURE 4.11. Example 4.3: surface plot of discrete solution on a 64×64 mesh. Solution is exhibiting a singularity at (0,0).

Example 4.4. We are solving simplified Poisson-Boltzmann equation (4.9) on $\Omega = (-1,1) \times (-1,1)$ with $k = 1.0$ (see Figure 4.15). In this example, the source function exhibits a huge variation inside the domain unlike the previous example (source is zero). The source is $f = 2.0y(y-1) + 2.0x(x-1) - (x(x-1)y(y-1))\exp(x(x-1)y(y-1))$. In this experiment, the exact solution is not known. The domain Ω is divided into four equal subdomains (see Figure 4.9) based on the medium properties $\boldsymbol{\epsilon}$. Further assume $\boldsymbol{\epsilon}_1 = 1.0\mathbf{I}$, $\boldsymbol{\epsilon}_2 = 4.0\mathbf{I}$, $\boldsymbol{\epsilon}_3 = 300.0\mathbf{I}$, and $\boldsymbol{\epsilon}_4 = 2.0\mathbf{I}$. $\boldsymbol{\epsilon}_1$, $\boldsymbol{\epsilon}_2$, $\boldsymbol{\epsilon}_3$, and $\boldsymbol{\epsilon}_4$ refer to the medium properties in the subdomains Ω_1, Ω_2, Ω_3, and Ω_4, respectively. Figure 4.13 is a surface plot of the discrete solution on a 50×50 mesh. Figure 4.14 reports convergence behaviour of the

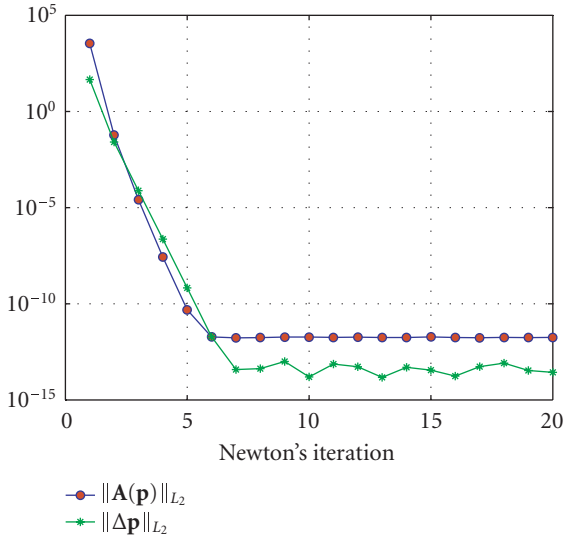

FIGURE 4.12. Example 4.3: residual and difference vector versus Newton's iteration.

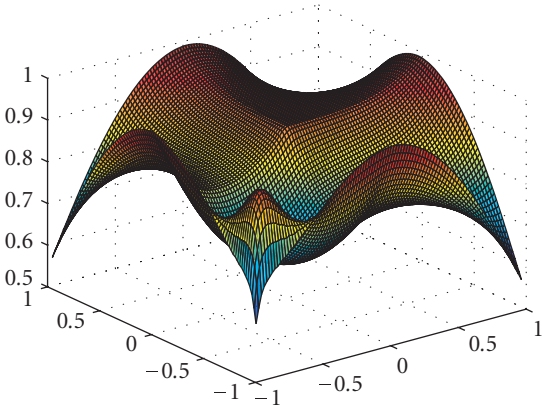

FIGURE 4.13. Example 4.4: surface plot of discrete solution on a 50×50 mesh.

Newton-Krylov Algorithm 3.1. It is observed that Newton's algorithm converged to a tolerance of 10×10^{-10} in 5-6 iterations:

$$-\operatorname{div}(\boldsymbol{\epsilon} \operatorname{grad} p) + k \sinh(p) = f \quad \text{in } \Omega, \tag{4.9}$$

$$p(x, y) = \cosh(xy) \quad \text{on } \partial\Omega. \tag{4.10}$$

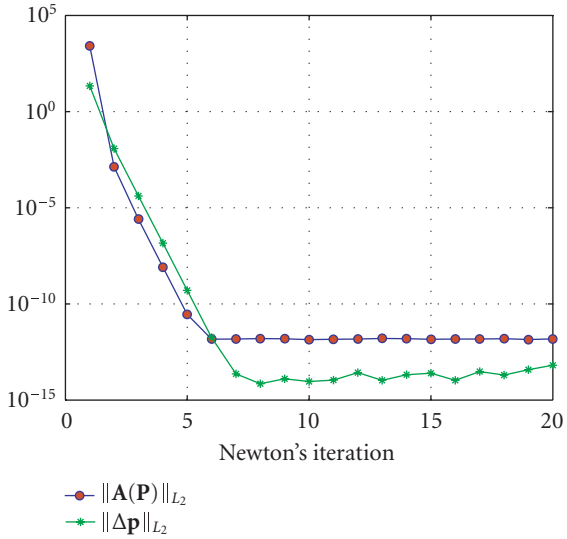

FIGURE 4.14. Example 4.4: residual and difference vector versus Newton's iteration.

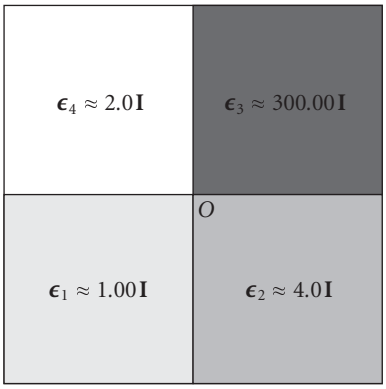

FIGURE 4.15. Example 4.4: property distribution in $\Omega = (-1,1)\times(-1,1)$.

5. Conclusions

We have presented the two-point finite volume discretization of the nonlinear elliptic problems. An implementation of Dirichlet and Neumann boundary conditions is also mentioned. A Newton-Krylov algorithm for solving the system of nonlinear equations is given. Reported numerical work validates the convergence of the discretization scheme for nonlinear elliptic problems. Convergence of the Newton-Krylov method is also reported.

References

[1] I. Aavatsmark, *An introduction to multipoint flux approximations for quadrilateral grids*, Computational Geosciences **6** (2002), no. 3-4, 405–432.

[2] M. Holst, *MCLite: an adaptive multilevel finite element MATLAB package for scalar nonlinear elliptic equations in the plane*, Tech. Rep., UCSD, California, 2004, guide to the MCLite software package, http://cam.ucsd.edu/~mholst/pubs/dist/mclite.

[3] M. Holst, R. E. Kozack, F. Saied, and S. Subramaniam, *Multigrid-based Newton iterative method for solving the full nonlinear Poisson-Boltzmann equation*, Biophysical Journal **66** (1994), A130.

[4] ———, *Protein electrostatics: rapid multigrid-based Newton algorithm for solution of the full nonlinear Poisson-Boltzmann equation*, Journal of Biomolecular Structure and Dynamics **11** (1994), 1437–1445.

[5] ———, *Treatment of electrostatic effects in proteins: multigrid-based Newton iterative method for solution of the full nonlinear Poisson-Boltzmann equation*, Proteins: Structure, Function, and Genetics **18** (1994), no. 3, 231–245.

[6] M. Holst and F. Saied, *Numerical solution of the nonlinear Poisson-Boltzmann equation: developing more robust and efficient methods*, Journal of Computational Chemistry **16** (1995), no. 3, 337–364.

[7] S. K. Khattri, *Computationally efficient technique for nonlinear Poisson-Boltzmann equation*, Proceedings of the 6th International Conference on Computational Science, Lecture Notes in Computer Science, vol. 3991, Reading, May 2006, pp. 860–863.

[8] ———, *Newton-Krylov algorithm with adaptive error correction for the Poisson-Boltzmann equation*, MATCH Communications in Mathematical and in Computer Chemistry **56** (2006), no. 1, 197–208.

[9] ———, *Analyzing and adaptive finite volume for flow through highly heterogenous porous medium*, http://www.mi.uib.no/~sanjay/.

[10] ———, *Analyzing finite volume for single phase flow in porous media*, to appear in Journal of Porous Media.

[11] S. K. Khattri and I. Aavatsmark, *Numerical convergence on adaptive grids for control volume methods*, http://www.mi.uib.no/~sanjay/.

[12] S. S. Kuo, M. D. Altman, J. P. Bardhan, B. Tidor, and J. K. White, *Fast methods for simulation of biomolecule electrostatics*, International Conference on Computer Aided Design, Valenciennes, May 2002.

[13] E. Larsson and B. Fornberg, *A numerical study of some radial basis function based solution methods for elliptic PDEs*, Computers & Mathematics with Applications **46** (2003), no. 5-6, 891–902.

[14] S. H. Lui, *On Schwarz alternating methods for nonlinear elliptic PDEs*, SIAM Journal on Scientific Computing **21** (2000), no. 4, 1506–1523.

[15] H. A. van der Vorst, *Iterative Krylov Methods for Large Linear Systems*, Cambridge Monographs on Applied and Computational Mathematics, vol. 13, Cambridge University Press, Cambridge, 2003.

Sanjay Kumar Khattri: Department of Mathematics, University of Bergen, 5008 Bergen, Norway
E-mail addresses: sanjay@mi.uib.no; sanjaykhatri1976@yahoo.com