Journal of Computational Physics 229 (2010) 5742–5754

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Application of Sobolev gradient method to Poisson–Boltzmann system

Abdul Majid *, Sultan Sial

Mathematics Department, Lahore University of Management Sciences, Opposite Sector U, DHA, Lahore Cantt., Pakistan

ARTICLE INFO

Article history:
Received 17 September 2009
Received in revised form 25 March 2010
Accepted 12 April 2010
Available online 20 April 2010

Keywords:
Sobolev gradient
Poisson–Boltzmann
FAS
Finite difference
Steepest descent

ABSTRACT

The idea of a weighted Sobolev gradient, introduced and applied to singular differential equations in [1], is extended to a Poisson–Boltzmann system with discontinuous coefficients. The technique is demonstrated on fully nonlinear and linear forms of the Poisson– Boltzmann equation in one, two, and three dimensions in a finite difference setting. A comparison between the weighted gradient and FAS multigrid is given for large jump size in the coefficient function.

1. Introduction

In this article, a weighted Sobolev gradient is used as a preconditioner to solve the linear and nonlinear Poisson–Boltzmann equation (PBE) with discontinuous coefficient functions. The idea of a weighted gradient, was introduced by Mahavier in [1] who demonstrated its effectiveness in dealing with linear and nonlinear singular differential equations. Sobolev gradients were considered as preconditioners for solution of first order and second order differential equations in [2] in which a one dimensional PBE that arises in semiconductor modeling is considered. An example is also given in [2] of a problem for which the Sobolev gradient method converges but Newton's method does not. In this article, we combine the idea of preconditioning with a weighted Sobolev gradient and present its application to linear and nonlinear PBE. We investigate how well the weighted Sobolev gradient works for large discontinuities in linear and nonlinear PBE and compare to unweighted Sobolev gradient and FAS multigrid.

In [3] Neuberger has introduced and developed the Sobolev gradient technique for solutions of differential equations. This method has proven its usefulness for problems from many fields such as minimization related to Ginzburg–Landau free energy functionals [4,5], the nonlinear Schrodinger equation [6], superconductors [7,8], applications to Differential Algebraic Equations [9], image processing problems [10] and optimal control problems [11]. The underlying idea is to formulate problems in terms of minimizing a functional whose critical points are the desired solutions. The functional that is to be minimized could be a least square functional or energy functional related to the system. Steepest descent is used for the minimization process.

In Section 2, we discuss the PBE in some detail. In Section 3, we build on [1] and give explanations and justifications for a weighted gradient and its possible application to differential equations with discontinuous coefficients. In Section 4, we present problems in the finite difference setting. In Section 5, we give results for numerical test problems for both linear and nonlinear PBE in finite difference settings. In Section 6, we compare the weighted Sobolev gradient with a nonlinear

* Corresponding author.
E-mail addresses: abdulm@lums.edu.pk (A. Majid), sultans@lums.edu.pk (S. Sial).

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.04.017

multigrid method. In Section 7, details of the software used are given. Finally, in Section 8 we conclude our results and offer possible future improvements.

2. Poisson–Boltzmann equation

This section is an introduction to the PBE and its application in estimating the electric potential around biomolecules. The PBE is an elliptic partial differential equation that models many important phenomena such as charge distributions in semiconductor devices [2] and the protein-folding problem. The PBE is being extensively studied to analyze the properties of the biomolecules in physics and chemistry. Here we present an overview and background of the equation sufficient for this article, interested readers are referred to [12,13] for more analysis and the derivation of the equation. For material covered in this section, we heavily relied upon [12].

When a macromolecule such as a protein is immersed in an ionic solution, a thick layer is formed due to the penetration of solvent ions that prevents contact of the molecule with the ionic solvent. This molecule can be identified with a charged cluster of atoms. The extended Deybe–Hückle theory [12] is used to model this phenomenon. Fig. 1 shows a sketch of the Deybe–Hückle model.

The electrostatic potential anywhere in the region Ω, where Ω is a cubical region as shown in Fig. 1, is given by the PBE

$$-\nabla\cdot(a(x)\nabla u)+\bar{\kappa}^2(x)\sinh(u(x))=\frac{4\pi e_c^2}{\kappa_B T}\sum_{i=1}^{N_m} z_i\delta(x-x_i)\quad x\in\Omega\subset\mathbb{R}^3\quad\text{and } u(x)=g(x), x\in\Gamma \tag{1}$$

Our notation follows Holst's work in his thesis [12]. Here Γ denotes the boundary of the domain Ω and $g(x)$ is some boundary function. e_c denotes the charge of electron, T represents the temperature, k_B the Boltzmann constant. If $q_i=z_i e_c$ represents the charge at the location x_i in the molecular region, z_i is the fraction of charge at the location x_i. The dielectric ε and the modified Debye–Hückle parameter $\bar{\kappa}$ are piecewise constant functions. If Ω_1,Ω_2 and Ω_3 denote molecular region, exclusion layer and solvent respectively, the other two coefficients $\varepsilon,\bar{\kappa}$ and the force term f can be defined below.

i. $a:\Omega\mapsto L(\mathbb{R}^3,\mathbb{R}^3),\quad a_{ij}(x)=\delta_{ij}\varepsilon(x)$

$$\varepsilon(x)=\begin{cases}\varepsilon_1 & x\in\Omega_1\\ \varepsilon_2=\varepsilon_3 & x\in\Omega_2\cup\Omega_3\end{cases} \tag{2}$$

ii. $\bar{\kappa}:\Omega\mapsto\mathbb{R}$,

$$\bar{\kappa}(x)=\begin{cases}0 & x\in\Omega_1\cup\Omega_2\\ \sqrt{\varepsilon_3}\kappa & x\in\Omega_3\end{cases} \tag{3}$$

where κ is the Debye–Hückle parameter that depends on the ionic strength I_s of the solvent and is given by the formula $(\frac{8\pi N_A e_c^2}{1000\varepsilon_3 k_B T})^{1/2} I_s^{1/2}$, where N_A is Avogadro's number.

iii. $f:\Omega\to\mathbb{R}$,

$$f(x)=\frac{4\pi e_c^2}{\kappa_B T}\sum_{i=1}^{N_m} z_i\delta(x-x_i) \tag{4}$$

where $x_1,x_2,\ldots,x_{N_m}\in\Omega_1$ denote the charge locations and $z_1,z_2,\ldots,z_{N_m}$ are associated fractional charges respectively.

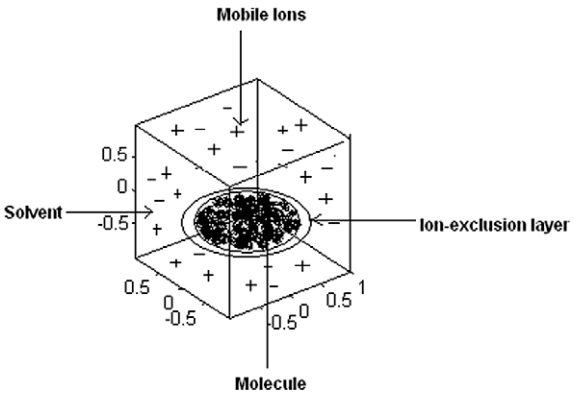

Fig. 1. A sketch of Debye–Hückle model in 3d.

For the linearized PBE we just have to replace $\sinh(u(x)) = u(x) + \frac{u(x)^3}{3!} + \cdots$ by its first approximation $u(x)$.

Since the problem (1) is defined over three separate regions, each region representing a different material, the coefficient functions ε and $\bar{\kappa}$ are discontinuous at the interfaces, giving us an interface problem.

Consider two ways to find the unknown electrostatic potential u in the region Ω. We can solve Eq. (1) directly, or minimize its corresponding energy functional

$$F(u) = \int \frac{1}{2}(\nabla u)^t a(x)(\nabla u) + \bar{\kappa}^2(x)\cosh(u) - fu \tag{5}$$

and in the linear case

$$F(u) = \int \frac{1}{2}(\nabla u)^t a(x)(\nabla u) + \frac{1}{2}\bar{\kappa}^2(x)u^2 - fu \tag{6}$$

Here t denotes the transpose. In fact, Eq. (1) is the Euler–Lagrange equation corresponding to functional (5). This functional is a convex functional that guarantees global minima u for $\nabla F(u) = 0$ in Ω, a solution to the problem (1). It is more efficient to find u by minimizing the energy functionals given by (5) and (6), respectively than minimizing a least square functional related to the problem (1) that involves second order derivative operators. We would like to design a Sobolev gradient method and demonstrate its application to the linear and nonlinear PBE with different jump discontinuities in its coefficient ε. We attempt to find the electrostatic potential u by minimizing functionals (5) and (6) in a finite difference setting. Numerical results and a detailed comparison of different gradients with regards to their efficiency are given in the subsequent sections.

3. General idea of Sobolev gradient

This section covers basic background about Sobolev gradients, weighted Sobolev gradients and steepest descent for energy functional minimization. We develop numerical schemes based on steepest descent for linear and nonlinear problems. Interested readers are referred to [3] for a more detailed discussion of Sobolev gradients. The choice of an inner product space is very important and plays a very significant role in obtaining adequate efficiency and accuracy. We construct a Sobolev inner product depending upon the operators of the underlying differential equation. Actually, in our case we are considering the operators in the functional. The question of the best inner product to use for a given problem is still an open and fundamental question. Having been inspired by Mahavier's idea of a weighted gradient [1], we define a weighted inner product suitable for the minimization of energy functionals (5) and (6).

A Sobolev space $H^{1,2}(\Omega)$ is defined as a space of those square *Lebesgue* integrable functions in L^2 whose first order weak derivatives also belong to $L^2(\Omega)$. $H^{1,2}(\Omega)$ is defined by the inner product

$$\langle u, v\rangle_{H^{1,2}} = \langle u, v\rangle_{L_2} + \langle D(u), D(v)\rangle_{L_2} \tag{7}$$

where D is the derivative operator.

Mahavier's idea of weighted gradients motivates us to define a new inner product on $H^{1,2}$ by taking care of the discontinuous function ε given by (2) that is affecting the derivative terms in (5) and (6).

We define a new inner product on $H^{1,2}$

$$\langle u, v\rangle_{H^{1,2}_w} = \langle u, v\rangle_{L_2} + \langle wD(u), wD(v)\rangle_{L_2} \tag{8}$$

Mahavier in his work [1] shows that if the differentiable weight function w is zero somewhere in the domain of interest then (8) defines an inner product on $H^{1,2}$ and $H^{1,2}$ is a Sobolev space under this inner product. In our case, the weight function $w = \varepsilon$ is a function that is discontinuous at two different points that represent the material discontinuities. Since ε is a piecewise constant function that is discontinuous at finitely many points, it is a function that is differentiable almost everywhere. One can extend the same idea [1] for weight functions such as ε so that $H^{1,2}$ forms a Sobolev space, say, $H^{1,2}_w$ where $w = \varepsilon$, under the inner product defined by (8).

The aim is to find the gradient $\nabla F(u)$ of a convex functional $F(u)$ associated with the problem and to find the zero of the gradient, by using steepest descent minimization process, that is the minima of $F(u)$ and a solution to the original problem.

The gradient $\nabla_{L_2}F(u)$ of the functional $F(u)$: $L_2(\Omega) \to R^n$ where $n = 1,2,3,\ldots$, defined by (5) and (6) is given by the Taylor series

$$F(u+h) = F(u) + \langle \nabla_{L_2}F(u), h\rangle_{L_2} + O(h^2)$$

where h is a test function.

If D denotes the weak derivative of a function u in $L_2(\Omega)$, the Sobolev gradient $\nabla_{H^{1,2}}F(u)$ in $H^{1,2}$ and the weighted Sobolev gradient $\nabla_{H^{1,2}_w}F(u)$ in $H^{1,2}_w$ can be found using Eqs. (7) and (8).

$$(I + D^*D)\nabla_{H^{1,2}}F(u) = \nabla_{L_2}F(u) \tag{9}$$

and

$$(I + D^*w^2D)\nabla_{H^{1,2}_w}F(u) = \nabla_{L_2}F(u) \tag{10}$$

where D^* represents the Hilbert adjoint of derivative operator D. We can solve Eqs. (9) and (10) for $\nabla_{H^{1,2}}F(u)$ and $\nabla_{H^{1,2}_w}F(u)$ using some appropriate iterative scheme respectively.

4. Numerical setting

For numerical experimentation, we used a finite difference operator to discretize the derivative operators. If the domain Ω in one dimension is divided into $1,2,\ldots,n$ grid points with δx the internodal spacing then $D_0{:}R^n \to R^{n-1}$ and $D_1{:}R^n \to R^{n-1}$, denote the averaging operator and first order derivative operator, respectively.

$$(D_0u)_i = \frac{u_{i+1}+u_i}{2} \quad \text{and } (D_1u)_i = \frac{u_{i+1}-u_i}{\delta x} \quad \text{where } i = 1,\ldots,n-1$$

Similarly, one can define averaging operator and finite difference versions of partial derivatives in higher dimensions. We discretize the domain such that each point of discontinuity in the region Ω is one of the grid points, and in fact the set of the points of discontinuities is a subset of the set of grid points. After discretization, the L_2 gradient of the functional described by (5) and (6) are given by, respectively

$$\nabla_{L_2}F(u) = \bar{\kappa}^2(x)\sinh(u) - f + \sum_{i=1}^{d} D_i^*(D_0(\varepsilon(x)) \cdot D_i(u)) \quad \text{for } d = 1,2,3 \tag{11}$$

$$\nabla_{L_2}F(u) = \bar{\kappa}^2(x)u - f + \sum_{i=1}^{d} D_i^*(D_0(\varepsilon(x)) \cdot D_i(u)) \quad \text{for } d = 1,2,3 \tag{12}$$

where D_i and D_i^* for i = 1,2,3, denote the finite difference version of first order partial derivatives and their Hilbert adjoints respectively.

A gradient of a functional gives the direction of the greatest increase per unit change in the argument of the functional. Therefore the most rapid decrease in the functional is in the direction opposite to the gradient. We use the steepest descent method for the minimization process. Steepest descent is the simplest of the gradient methods that moves in the direction opposite to the gradient in order to search for the minima of a functional. One of the advantages of steepest descent is that it converges even for a poor initial approximation. One can use a constant step size in the steepest descent minimization process or a line search technique can also be employed to find the optimum value of step size at each minimization step. The minimization process is terminated when the infinity norm of the gradient vector reaches the desired magnitude. Since the matrices $P = (I + D^*D)$ and $P_w = (I + D^*w^2D)$ are sparse and positive definite, any iterative method such as conjugate gradient, Gauss–Seidel or multigrid could be used to solve (9) and (10).

Suppose ∇ denotes the gradient in $L_2, H^{1,2}$ and $H^{1,2}_w$ spaces. We find the step size λ at each step that minimizes $F(u - \lambda\nabla F(u))$. In the linear case step size λ in the steepest descent is given by

$$\lambda = \frac{\langle \nabla F(u), \nabla F(u)\rangle}{\langle \nabla F(\nabla F(u)), \nabla F(u)\rangle} \tag{13}$$

where the inner product is taken in the corresponding Hilbert space. For the nonlinear problem a univariant line search is used to find the value of step size at each step. Now we can define a general algorithm to solve our problem in all three spaces. If π denotes the projection operator that sets the boundary points of vectors equal to zero, to incorporate the Dirichlet boundary conditions, the steepest descent algorithm looks like

Algorithm 1

1. choose an appropriate initial guess u
2. calculate $\pi\nabla F(u)$
3. find λ that minimizes $F(u - \lambda\nabla F(u))$
4. do $u \to u - \lambda\pi\nabla F(u)$
5. go to step 2 until the infinity norm of $\nabla_{L_2}F(u) <$ tolerance

Since in one dimension the matrices P and P_w are tridiagonal band matrices one can use LU decomposition for the exact solution of (9) and (10). When using inexact line search to find the step size and an iterative solver to solve (9) and (10), proper care should be taken to decide the stopping criterion as it significantly affects the over all convergence of steepest descent.

5. Application to Poisson–Boltzmann equation

We consider some experiments on a test problem similar to those investigated by Holst in [12], using multilevel method in different settings. For the test problem, we specify the coefficients in Eq. (1). This is a hypothetical problem and might not be related to any particular molecule, we just want to evaluate our algorithm in one, two and three dimensions with a

version of the PBE with jump discontinuities in its coefficient functions. We map the molecule into a line segment, a square or a cube, depending upon the dimension we are working in. We take the molecule well within the domain so that it incorporates 30% of the whole domain. Let us assume that for some temperature T and ionic strength I_s of solvent we have the following bounds on the coefficients:

- $f : \Omega \to \mathbb{R}, -1 \leqslant f(x) \leqslant 1 \quad \forall x \in \Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$ and $\Omega_1 \cap \Omega_2 \cap \Omega_3 = \varphi$
- $\varepsilon : \Omega \to \mathbb{R}, \varepsilon(x) = 2$ for $x \in \Omega_1$ and $\varepsilon(x) = 80$ for $x \in \Omega_2 \cup \Omega_3$
- $\bar{\kappa} : \Omega \to \mathbb{R}, \bar{\kappa}(x) = 0$ for $x \in \Omega_1 \cup \Omega_2$ and $\bar{\kappa}(x) = 1.732$ for $x \in \Omega_3$
- $g : \Gamma \to \mathbb{R}, g(x) = 1.0 \quad \forall x \in \Gamma$

Since we have taken $-1 \leqslant f(x) \leqslant 1$, for the sake of simplicity we take z_i, fractional charge at each node x_i, to be equally distributed over [-1,1] in the molecular region. Figs. 3 and 5 show the graphs of f in 1d and 2d respectively. And Figs. 2 and 4 show the function ε in 1d and 2d respectively. We define f and ε in a similar way in 3d.

In the next section we will also demonstrate convergence results for different ratios $D = \frac{\varepsilon_1}{\varepsilon_2}$ of the jump size. We will do the same problem in one, two and three dimensions in both linear and nonlinear cases. This problem can be viewed as our prototypical problem.

5.1. Application to linear PBE

In this section we use energy functional (5) that is in its linearized form in order to assess our algorithm. Before we move ahead some analysis of the spectral radii is in order. The idea of unweighted and weighted gradient can be viewed as a preconditioning strategy to solve the linear system (9) and (10). If we use a constant and optimal value of step size λ then we can give a spectral radii comparison of three different gradients. After discretization the L_2 gradient (12) takes the form $\nabla_{L_2} F(u) = Au - f$. For all three gradients $L_2, H^{1,2}$ and $H^{1,2}_w$ the steepest descent iterative schemes are respectively as follows.

$$u \to (I - \lambda A)u + \lambda f, \quad u \to (I - \lambda P^{-1}A)u + \lambda P^{-1} f, \quad u \to (I - \lambda P_w^{-1}A)u + \lambda P_w^{-1} f$$

From these iteration schemes it is evident that two different Sobolev inner products, weighted and unweighted result in two different preconditioners P_w and P respectively where $T_1 = (I - \lambda A), T_2 = (I - \lambda P^{-1}A)$ and $T_3 = (I - \lambda P_w^{-1}A)$ are the iteration matrices in $L_2, H^{1,2}$ and $H^{1,2}_w$ respectively. A comparison of different spectral radii is given for different grid points M in Table 1. Here ρ denotes spectral radius of the iteration matrix. It may be recalled that a necessary and sufficient condition for an iterative process to converge is that $\rho < 1$ and for small values of ρ the convergence is fast.

Table 1 shows the dominance of weighted gradient over the unweighted gradient and poor performance of the L_2 gradient is also clear. Since spectral radii of both weighted and unweighted Sobolev gradients are independent of the grid points it shows that convergence rate is independent of the problem size. Here convergence rate means the amount of the error that is reduced per iteration. The time required per iteration will be different for different grid sizes and depends on what type of solver is used and what the stopping criterion is set when solving (9) and (10). One important point to note is that although spectral radii in $H^{1,2}$ and $H^{1,2}_w$ came out to be independent of problem size M, the number of iterations for a given M is dependent on the stopping criterion set for the iterative solver to solve systems (9) and (10). The number of iterations m required to reduce the error factor by 10^{-n} can be found by the formula $m \geqslant -\frac{n}{\log_{10}|\rho(T)|}$ [14] where T is the iteration matrix of $u \to Tu - c$. In fact, one of the reasons of the efficiency of the weighted gradient is that it allows much larger step size in the steepest descent compared to unweighted gradient.

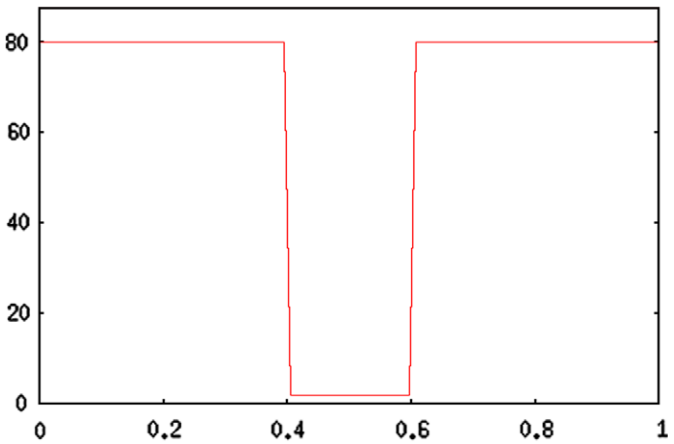

Fig. 2. Graph of ε in 1d.

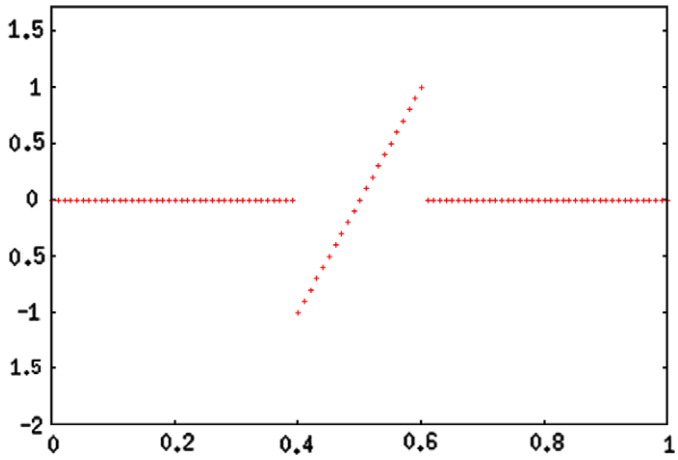

Fig. 3. Graph of force function f in 1d.

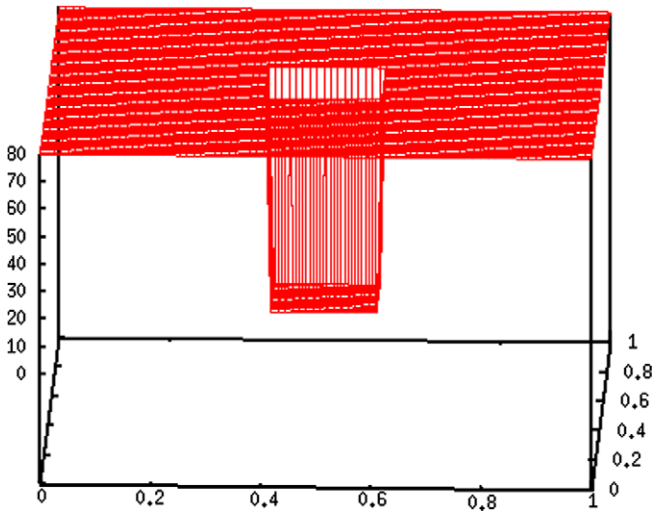

Fig. 4. Graph of ε in 2d.

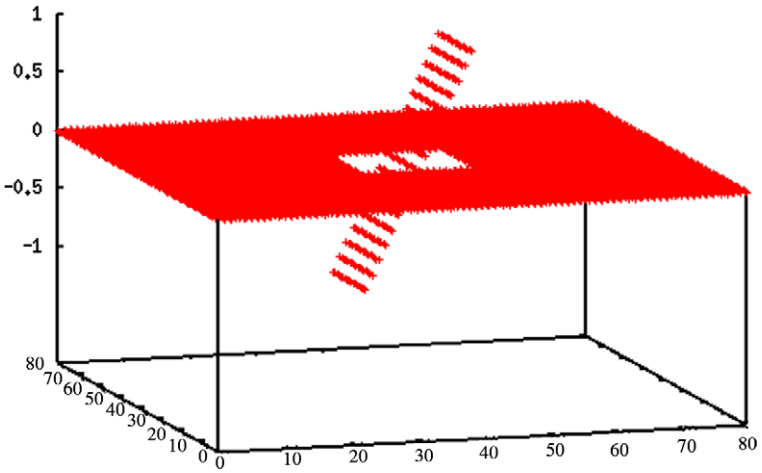

Fig. 5. Graph of force function f in 2d.

Table 1
Spectral radii of the iteration matrices T_1, T_2 and T_3 for different grid size M.

M	$\rho(T_1)$	$\rho(T_2)$	$\rho(T_3)$
51	0.9991	0.9602	0.9513
101	0.9998	0.9602	0.9513
201	0.9999	0.9602	0.9513

Here we give results in 1d, 2d and 3d with different grid size. For all experiments in this section a steepest descent method was run and an initial guess $u = 1.0$ was taken. The minimization process was terminated when the infinity norm of gradient was less than 10^{-7}.

1-Dimension

Let $\Omega = [0,1], \Omega_1 = [0.4,0.6], \Omega_3 = \Omega - (0.2,0.8)$ and $\Omega_2 = \Omega - \Omega_1 \cup \Omega_3$ represent the entire domain, molecular region, ionic solution and ion exclusion layer respectively. In one dimension $a(x) = \varepsilon(x)$ where ε is defined by (2). Table 2 shows the results for different gradients with different number of grid points M.

2-Dimension

In two dimensions, let $\Omega = [0,1] \times [0,1], \Omega_1 = [0.4,0.6] \times [0.4,0.6], \Omega_3 = \Omega - (0.2,0.8) \times (0.2,0.8)$ and $\Omega_2 = \Omega - \Omega_1 \cup \Omega_3$. $a(x)$ is a 2×2 scalar matrix with $\varepsilon(x)$ as diagonal entries. Since the L_2 gradient gives very poor performance in higher dimensions we only compare Sobolev gradient and weighted Sobolev gradient results in higher dimensions. Results were obtained on the two dimensional grid with $M \times M$ grid points.

3-Dimension

In three dimensions, let $\Omega = [0,1] \times [0,1] \times [0,1], \Omega_1 = [0.4,0.6] \times [0.4,0.6] \times [0.4,0.6], \Omega_3 = \Omega - (0.2,0.8) \times (0.2,0.8) \times (0.2,0.8)$ and $\Omega_2 = \Omega - \Omega_1 \cup \Omega_3$. Similarly, here we only compare Sobolev gradient and weighted Sobolev gradient results on three dimensional grid with $M \times M \times M$ grid points. Here $a(x)$ is 3×3 scalar matrix with $\varepsilon(x)$ as the diagonal entries.

Tables 2–4 show that the unweighted gradient is better than L_2 and weighted gradient outperforms the unweighted one and the advantage of weighted gradient over unweighted is unaffected by the dimension we are working in.

Finally, in this section we are interested in how convergence is affected by the large jump size. That is, how these weighted and unweighted gradients behave for different jump size $D = \frac{\varepsilon_1}{\varepsilon_2}$, when the value of dielectric ε in the molecular region is smaller than its value in the ionic solution. Again, spectral radius would be an appropriate measure to check the efficiency of gradients. To this end, we ran the experiment of 1d linear PBE with $M = 101$ grid points and for each values of D. An optimal value of the step size λ was taken to minimize the corresponding spectral radius. Fig. 6 is a graph that compares spectral radii in $H^{1,2}_w$ and $H^{1,2}$ with different jump size in $\varepsilon(x)$. As expected, increase in spectral radii of weighted and unweighted gradients for large jump size is seen. Nonetheless, both gradients continue to converge as spectral radii are always less then 1. But, it can be seen from the graph that spectral radii of the weighted gradient remain less than the spectral radii of the unweighted gradient. This shows that even for large jump size the weighted gradient retains its superiority over the unweighted gradient. One point that is worth mentioning here is that all these observations are limited to this specific problem and the two gradients might result in different ways for different problem.

The inner product (8) provided us with a preconditioner $P_w = (I + D^* w^2 D)$ whose efficiency is independent of problem size and its advantage over $P = (I + D^* D)$ is independent of jump size. Here we don't recommend this method for simple linear problems because for such problems mostly used iterative methods such as linear multigrid would be more efficient. However, we are concerned with its application to complex problems and the real benefit of this method lies in nonlinear problems with discontinuous coefficients with large jump discontinuities.

5.2. Application to nonlinear PBE

We are mainly interested in the convergence behavior of the Sobolev gradient to a nonlinear PBE with jump discontinuous coefficient functions. Here we perform the same experiments that we did for the linear case, but this time we minimize the nonlinear functional (5). Since $\sinh u \simeq u$ for $x \in [0,1]$ and in this interval there is no significant difference between the

Table 2
Comparison between the $L_2, H^{1,2}$ and $H^{1,2}_w$ gradients by solving the linear PBE on different grid size M in 1d.

	L_2			$H^{1,2}$			$H^{1,2}_w$		
M	51	101	201	301	501	701	301	501	701
No. of iterations	22,425	71,967	428,480	652	648	685	349	351	358
CPUs	0.2031	0.9687	15.625	0.1093	0.172	0.281	0.0321	0.0625	0.0937

Table 3
Comparison between the $H^{1,2}$ and $H^{1,2}_w$ gradients by solving linear PBE on different grids size $M \times M$ in 2d.

	$H^{1,2}$			$H^{1,2}_w$		
M	21	31	51	21	31	51
No. of iterations	560	568	587	351	366	392
CPUs	3.391	7.172	33.421	2.140	4.703	25.484

Table 4
Comparison between the $H^{1,2}$ and $H^{1,2}_w$ gradients by solving linear PBE on different grids size $M \times M \times M$ in 3d.

	$H^{1,2}$			$H^{1,2}_w$		
M	11	21	31	11	21	31
No. of iterations	256	400	578	155	259	507
CPUs	6.578	85.656	627.828	5.219	62.469	497.71

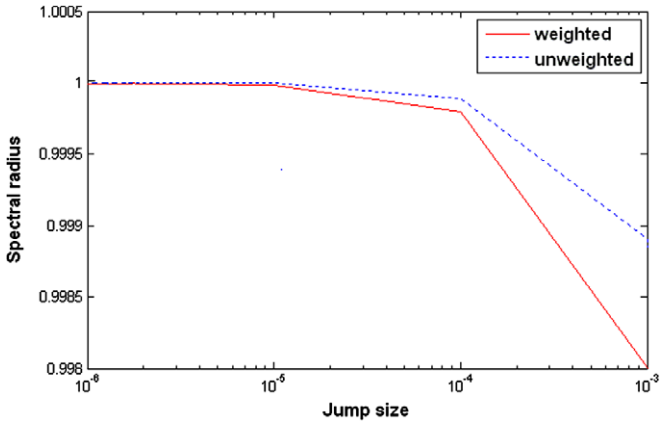

Fig. 6. Comparison of spectral radii in $H^{1,2}_w$ and $H^{1,2}$ with different jump size for 1d linear PBE where $M = 101$.

linear and nonlinear cases, therefore we extend the interval of interest from [0,1] to [0,10] just to emphasize the nonlinearity.

In the nonlinear case, after discretization the Eq. (11) becomes $\nabla_{L_2} F(u) = N(u) - f$. And we use the same algorithm to solve this nonlinear system. One of the major advantages of this method is that we don't require any extra effort for the nonlinear problem, unlike Newton's method where calculation of Hessian matrix is required for each step. Though $\nabla_{L_2} F(u)$ is nonlinear we only need to solve the linear system (9) and (10) to find the $\nabla_{H^{1,2}} F$ and $\nabla_{H^{1,2}_w} F$ respectively.

For all experiments, in this section a steepest descent method was run with an initial guess $u = 1.0$. Here the jump size is $D = \frac{2}{80}$, same as in the linear case. The minimization process was terminated when the infinity norm of gradient was less than 10^{-7}. Coefficient matrix $a(x)$ is the same as in the linear case. Here we give results in 1d, 2d and 3d.

1-Dimension

Let $\Omega = [0,10], \Omega_1 = [4,6], \Omega_3 = \Omega - (2,8)$ and $\Omega_2 = \Omega - \Omega_1 \cup \Omega_3$ represent the entire domain, molecular region, ionic solution and ion exclusion layer respectively.

2-Dimensions

In two dimensions, $\Omega = [0,10] \times [0,10], \Omega_1 = [4,6] \times [4,6], \Omega_3 = \Omega - (2,8) \times (2,8)$ and $\Omega_2 = \Omega - \Omega_1 \cup \Omega_3$. We don't give L_2 gradient results as it gives poor performance in higher dimension. Results were obtained on the two dimensional grid with $M \times M$ grid points.

3-Dimensions

In three dimensions, $\Omega = [0,10] \times [0,10] \times [0,10]$ where $\Omega_1 = [4,6] \times [4,6] \times [4,6], \Omega_3 = \Omega - (2,8) \times (2,8) \times (2,8)$ and $\Omega_2 = \Omega - \Omega_1 \cup \Omega_3$. We only compare Sobolev gradient and weighted Sobolev gradient results on three dimensional grid with $M \times M \times M$ grid points.

From Tables 5–7 it is clear that L_2 gradient is dominated by unweighted Sobolev gradient and the former one is outperformed by the weighted Sobolev gradient. And weighted Sobolev gradient maintains its performance in 1d, 2d and 3d. If we

Table 5
Comparison between the $L_2, H^{1,2}$ and $H_w^{1,2}$ gradients by solving nonlinear PBE on different grid size M in 1d.

	L_2			$H^{1,2}$			$H_w^{1,2}$		
M	51	101	201	301	501	701	301	501	701
No. of iterations	45,929	456,756	455,739	657	676	689	356	366	372
CPUs	0.843	14.678	25.751	0.156	0.297	0.5937	0.062	0.109	0.250

Table 6
Comparison between the $H^{1,2}$ and $H_w^{1,2}$ gradients by solving nonlinear PBE on different grid size $M \times M$ in 2d.

	$H^{1,2}$			$H_w^{1,2}$		
M	21	31	51	21	31	51
No. of iterations	603	619	659	343	359	357
CPUs	6.313	15.765	41.422	2.1093	4.234	23.46

Table 7
Comparison between the $H^{1,2}$ and $H_w^{1,2}$ gradients by solving nonlinear PBE on different grid size $M \times M \times M$ in 3d.

	$H^{1,2}$			$H_w^{1,2}$		
M	11	21	31	11	21	31
No. of iterations	872	954	969	224	244	334
CPUs	13.843	130.156	759.093	7.093	62.656	541.359

compare the results of linear and nonlinear problems one can see that the advantage of weighted gradient over the unweighted is more pronounced in the nonlinear problem as compared to the linear problem. Fig. 7 shows an isosurface of the electrostatic potential u.

After having done an experiment for small jumps in the nonlinear PBE, we would like to apply the same technique for large size of discontinuities $D = \frac{\varepsilon_1}{\varepsilon_2}$ where $D = 10^{-1}, 10^{-2}, 10^{-3}$ and 10^{-4}, when the value of dielectric ε in the molecular region is smaller than its value in the ionic solution. The following graphs show the error vs CPU time in $H^{1,2}$ and $H_w^{1,2}$. All results were obtained in 3d on $11 \times 11 \times 11$ grid points and minimization process was terminated when the infinity norm of $\nabla_{L_2} F$ was less than 10^{-7}. Table 8 summarizes the corresponding results. Figs. 8–11 are the related graphs that are drawn on logscale.

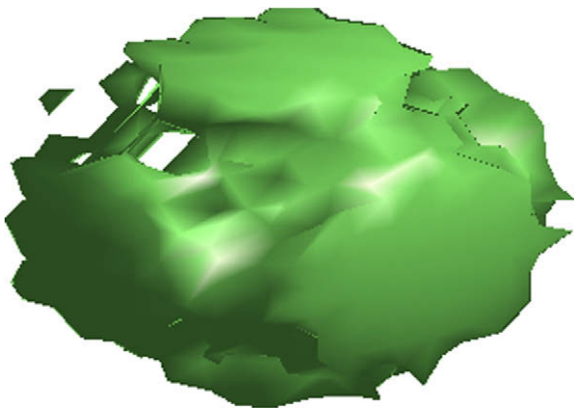

Fig. 7. An isosurface of the electrostatic potential u, obtained by solving nonlinear PBE in 3d using weighted Sobolev gradient.

Table 8
CPU units obtained by solving nonlinear PBE in 3d on grid size $11 \times 11 \times 11$ for different jump size D.

$D = 10^{-1}$		$D = 10^{-2}$		$D = 10^{-3}$		$D = 10^{-4}$	
$H^{1,2}$	$H_w^{1,2}$	$H^{1,2}$	$H_w^{1,2}$	$H^{1,2}$	$H_w^{1,2}$	$H^{1,2}$	$H_w^{1,2}$
8.578	0.5118	15.734	1.938	54.641	7.237	330.516	71.969

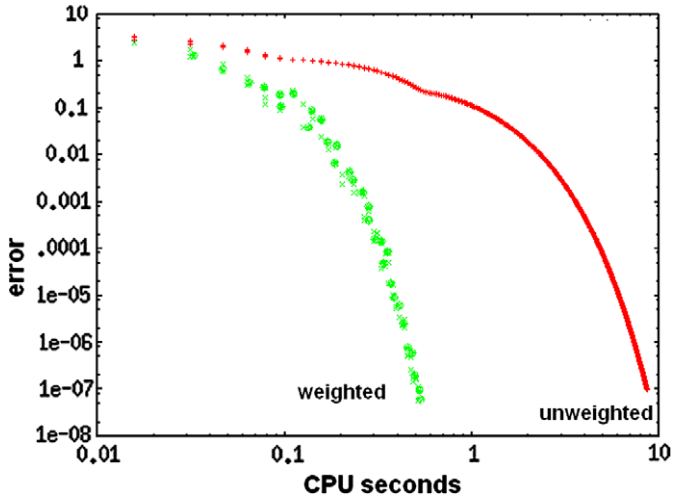

Fig. 8. Error comparison for nonlinear PBE when $D = 10^{-1}$.

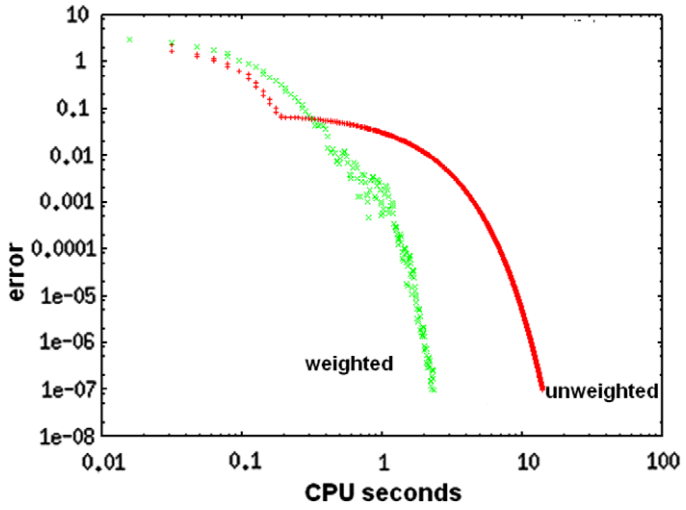

Fig. 9. Error comparison for nonlinear PBE when $D = 10^{-2}$.

From Table 8 and the above graphs one can see that steepest descent in both $H^{1,2}$ and $H^{1,2}_w$ continues to work for large jump size and weighted gradient preserves its effectiveness over the unweighted one. One more point that should be noted from Table 8 is that weighted gradient is more advantageous over the unweighted gradient for small values of D. In the next section we give a comparison between our method and one commonly used for such types of problems.

6. A comparison

This section presents a comparison between weighted gradient and a nonlinear multigrid full approximation scheme (FAS) applied to a nonlinear elliptic PBE. We would like to test these two methods on a problem with exponential nonlinearity and large jump discontinuity in its coefficient function. Rapid nonlinearities and large oscillations in a partial differential equation can significantly affect the convergence of a numerical algorithm. For comparison between the two methods, we consider the same test problem as discussed in [12].

The nonlinear problem has the form

$$-\nabla \cdot (a(x)\nabla u(x)) + b(x, u(x)) = f(x) \quad \text{in } \Omega \subset \mathbb{R}^3, \quad u(x) = g(x) \quad \text{on } \Gamma \tag{14}$$

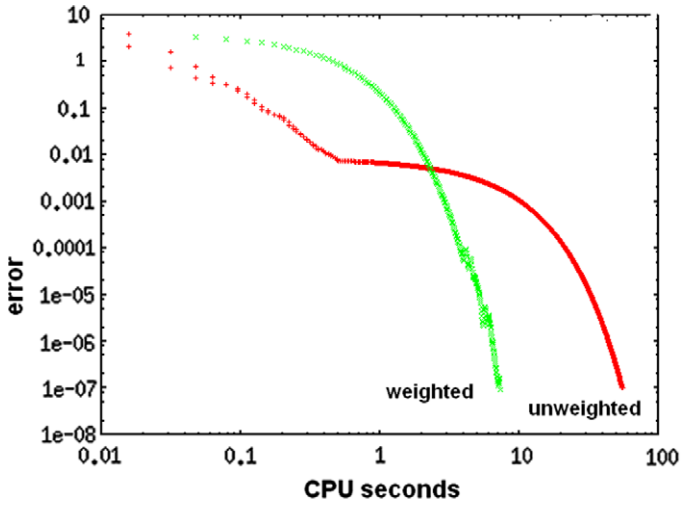

Fig. 10. Error comparison for nonlinear PBE when $D = 10^{-3}$.

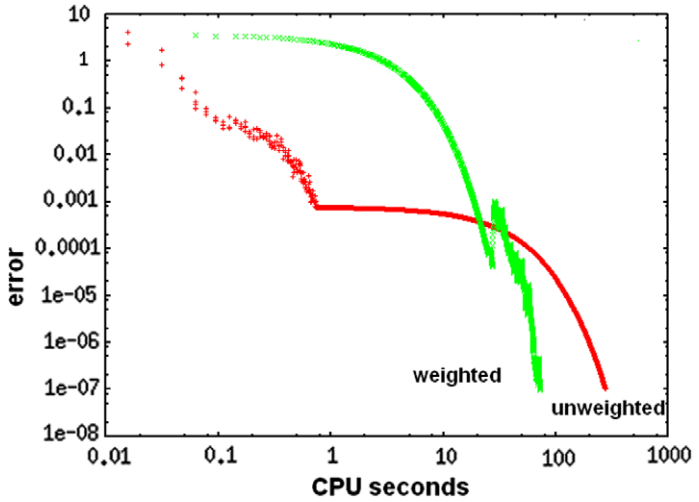

Fig. 11. Error comparison for nonlinear PBE when $D = 10^{-4}$.

where the coefficients are given by

- $a : \Omega \mapsto L(\mathbb{R}^3, \mathbb{R}^3),\ a_{ij}(x) = \delta_{ij}\varepsilon(x),\ 1 \leqslant \varepsilon(x) \leqslant 1.0 \times 10^8\ \forall x \in \Omega$
- $b : \Omega \times \mathbb{R} \mapsto \mathbb{R},\ b(x, u(x)) = \mu e^{u(x)},\ \forall x \in \Omega$
- $f : \Omega \mapsto \mathbb{R},\ -1 \leqslant f(x) \leqslant 1,\ \forall x \in \Omega$
- $g : \Gamma \mapsto \mathbb{R},\ g(x) = 0,\ \forall x \in \Gamma$

where $\varepsilon(x)$ is piecewise constant, defined as

$$\varepsilon(x) = \begin{cases} 1 \leqslant \varepsilon_1 \leqslant 1.0 \times 10^8, & \text{if } x \in \Omega_1 \\ 1 \leqslant \varepsilon_2 \leqslant 1.0 \times 10^8, & \text{if } x \in \Omega \setminus \Omega_1 \end{cases}$$

Here f is the same function as defined in the previous sections and $\mu = 1$. We compare the two methods for a one dimensional problem and we take $\Omega = [0,1]$ and $\Omega_1 = [0.4, 0.6]$. We will consider large jump size in $\varepsilon(x)$ so that the ratio $D = \frac{\varepsilon_1}{\varepsilon_2}$ can be as large as 10^5.

The energy functional corresponding to (14) is

$$F(u) = \int \frac{1}{2}(\nabla u)^t a(x)(\nabla u) + \mu e^u - fu \tag{15}$$

Table 9
Comparison of CPU units between $H_w^{1,2}$ and FAS V-cycle for different jump size D.

$D = 10^1$		$D = 10^2$		$D = 10^3$		$D = 10^5$	
$H_w^{1,2}$	FAS V-cycle	$H_w^{1,2}$	FAS V-cycle	$H_w^{1,2}$	FAS V-cycle	$H_w^{1,2}$	FAS V-cycle
0.0156	0.0938	0.2188	0.4375	2.312	3.344	233.640	356.049

We are looking for a function u that minimizes $F(u)$ where

$$\nabla_{L_2} F(u) = D_1^*(D_0(\varepsilon(x)) \cdot D_1(u)) + \mu e^u - f \tag{16}$$

After discretizing $\nabla_{L_2} F(u)$ using finite difference operator, we have $\nabla_{L_2} F(u) = N(u) - f$. We solve this nonlinear system using weighted gradient and FAS.

For multigrid method we use a FAS V-cycle multigrid method. Gauss–Newton (nonlinear Gauss–Seidel) [15] method was used for the relaxation sweeps on each grid. Linear interpolation operator, full weighting averaging operator and injection operator were used as intergrid transfer operators. Number of relaxation sweeps on each grid were chosen to optimise the efficiency of the algorithm. For weighted gradient, we used the same algorithm that we used in the previous section for nonlinear problem.

One iteration of FAS constitutes one V-cycle with nonlinear relaxation sweeps on each grid all the way from finest grid to coarsest grid and then from coarsest to finest. For $M = 2^l + 1$ grid points one has to visit l grid level in one V-cycle. And to achieve required accuracy many V-cycles may be required. On the other hand, one iteration of steepest descent means solving the linear system (10) and a steepest descent minimization step. Since each iteration of one method is entirely different from the other we do not compare the number of iterations. Instead, we compare CPU units required to reach convergence.

Table 9 gives a comparison of CPU units between the two methods for different jump sizes in the coefficient function $\varepsilon(x)$. The experiment was run on $M = 129$ grid points. The algorithm was terminated when the infinity norm of the gradient vector was less than 10^{-7}. The initial guess was $u = 0$.

From Table 9, one can see an advantage in using the weighted gradient instead of FAS V-cycle. This is more pronounced for small jumps than for large ones. We don't claim a superiority in all cases, FAS performs better than gradient methods for simpler nonlinear problems. The weighted gradient discussed in this work offers an alternative method that proved to be more efficient and robust for nonlinear problems that involve rapid nonlinearities and large discontinuities.

7. Software used

All experiments were performed on the note book PC HP 530 with Intel (R) Core (TM)2CPU T5200 @ 1.60 GHz, 0.99 GB of RAM. We used fortran compiler g95 for all tests except for Section 6 where matlab 7.0 was used for comparison purpose. All computations were performed in double precision. Gnuplot and Matlab both were used to generate all types of graphs.

8. Conclusion and future work

We have shown that one can improve descent direction in the steepest descent by defining various inner products. The weighted Sobolev gradient we constructed has a clear advantage over the unweighted Sobolev gradient. Comparison shows that for exponential nonlinearity and jump discontinuity, the algorithm can be used as an alternative to FAS V-cycle. Previously in [1], the application of weighted gradient was given for singular differential equations. In this article we have used the idea of weighted gradient usefully for a nonlinear and linear problem with jump discontinuities in the coefficient functions and the method remains stable for large jump size. Both weighted and unweighted Sobolev gradients exhibit consistent performance in one, two and three dimensions. In [2] it was shown that for some problems Newton's method fails to converge near the singularity but Sobolev gradient method does converge. Sobolev gradients, unweighted or weighted, offer a competitive method for more complicated and irregular differential equations. The possibility of improvement exists with consideration of inner products different from (7) and (8) that will change the descent direction and may result in better performance.

Given more RAM, one would like to extend the comparison of weighted gradient with FAS to 3d with considerably more nodes. After comparing with FAS, the next candidate for comparison would be Newton's multigrid [14] which is another variation of the nonlinear multigrid method that usually requires a good initial guess and considered more efficient. The next step would be to compare performance with softwares such as DELPHI.

Acknowledgment

Here we take an opportunity to offer special thanks to W. T. Mahavier for his productive and fruitful discussions on weighted gradients.

References

[1] W.T. Mahavier, A numerical method utilizing weighted Sobolev descent to solve singular differential equations, Nonlinear World 4 (1997) 435–455.
[2] W.B. Richardson, Sobolev preconditioning for the Poisson–Boltzmann equation, Compu. Methods Appl. Mech. Eng. 181 (2000) 425–436.
[3] J.W. Neuberger, Sobolev Gradient and Differential Equations, Springer Lecture Notes in Mathematics, second ed., vol. 1670, Springer-Verlag, New York, 1997.
[4] N. Raza, S. Sial, S.S. Siddiqi, Sobolev gradient approach for time the evolution related to the energy minimization of Ginzburg–Landau functional, J. Comp. Phys. 228 (2009) 2566–2571.
[5] S. Sial, J.W. Neuberge, T. Lookman, A. Sexena, Energy minimization using phase Sobolev gradient: application to phase separation and ordering, J. Comp. Phys. 189 (2003) 88–97.
[6] N. Raza, S. Sial, S.S. Siddiqi, T. Lookman, Energy minimization related to the nonlinear Schrodinger equation, J. Comp. Phys. 228 (2009) 2572–2577.
[7] J.W. Neuberger, R.J. Renka, Computational simulation of vortex phenomena in superconductors, Electron. J. Differ. Equ. Conf. 10 (2003) 245–250.
[8] S. Sial, Sobolev gradient algorithm for minimum energy state of s-wave superconductors-finite element setting, Supercond. Sci. Technol. 18 (2005) 675–677.
[9] R. Nittka, M. Sauter, Sobolev gradients for differential algebraic equations, Electron. J. Diff. Eq. 42 (2008) 1–31.
[10] R.J. Renka, Constructing fair curves and surfaces with a Sobolev gradient method, CAGD 21 (2004) 137–149.
[11] I. Knowles, Variational methods for ill-posed problems, Contemp. Math. 357 (2004) 187–199.
[12] M.J. Holst, The Poisson–Boltzmann equation analysis and multilevel numerical solution, Ph.D. Thesis, University Of Illinois, USA, 2000.
[13] I.L. Chern, J. Guoliu, W.C. Wang, Accurate evaluation of electrostatics for macromolecules in solutions, Method Appl. Anal. 10 (2003) 309–328.
[14] W.L. Briggs, V.E. Henson, S.F. McCormick, A Multigrid Tutorial, second ed., SIAM, 2000.
[15] J.M. Ortega, W.C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, San Diego, 1970. Reprinted as vol. 30, Classics in Applied Mathematics, SIAM, Philadelphia, 2000.