Journal of Computational Physics 228 (2009) 2572–2577

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Energy minimization related to the nonlinear Schrödinger equation

Nauman Raza [a,b,*], Sultan Sial [c], Shahid S. Siddiqi [a], Turab Lookman [d]

[a] Department of Mathematics, University of the Punjab, House No. 15, Street No. 39, Lahore, Pakistan
[b] Lahore University of Management Sciences, Lahore, Pakistan
[c] Department of Mathematics, Lahore University of Management Sciences, Opposite Sector U, DHA, Lahore Cantt. 54792, Pakistan
[d] Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

ARTICLE INFO

Article history:
Received 11 January 2008
Received in revised form 24 October 2008
Accepted 12 December 2008
Available online 25 December 2008

Keywords:
Sobolev gradients
Nonlinear Schrödinger equation

ABSTRACT

In this the window of the Sobolev gradient technique to the problem of minimizing a Schrödinger functional associated with a nonlinear Schrödinger equation. We show that gradients act in a suitably chosen Sobolev space (Sobolev gradients) can be used in finite-difference and finite-element settings in a computationally efficient way to find minimum energy states of Schrödinger functionals.

1. Introduction

Many problems in physics can be formulated in terms of finding critical points of energy functionals. The recent theory of Sobolev gradients [1] provides a unified point of view on such problems, both in function spaces and in finite dimensional approximations to such problems. Sobolev gradients have been used for ODE problems [1,2] in a finite-difference setting, PDEs in finite-difference [2] and finite-element settings [3], minimizing energy functionals associated with Landau–Ginzburg models in finite-difference [4] and finite-element [5,6] settings, the electrostatic potential equation [7], nonlinear elliptic problems [8], semilinear elliptic systems [9], simulation of Bose–Einstein condensates [10], and inverse problems in elasticity [11] and groundwater modeling [12]. The Sobolev gradient technique has been discussed previously for minimizing Schrödinger functionals in Fourier space setting [13]. In this article a Sobolev gradient method in finite-difference setting and finite-element setting is discussed for minimizing Schrödinger functionals.

A gradient of a functional gives the direction of change per unit change in the argument of the functional. The direction of a gradient strongly depends on how the size of arguments of a functional are measured. Schrödinger functionals include derivatives of the arguments. Such arguments have to be considered large if some of its derivatives are large. Theoretical considerations of such functionals must take this into account and is often overlooked in numerical approximations. The theory of Sobolev gradient [1] is an organized account of how to choose a metric for a finite dimensional problem related to an infinite dimensional theoretical problem. It is found that a good choice leads to gradients (Sobolev gradients) which are considerably smoother than those normally used [14].

For the model we studied, the Sobolev gradient technique becomes increasingly attractive as grid spacing is refined, and accordingly, dimension is increased.

In this paper, we briefly consider the theory of Sobolev gradients as applied to minimizing a free energy functional in finite-difference and finite-element setting. This theory leads to a steepest descent method in an appropriate Sobolev space

[*] Corresponding author. Address: Department of Mathematics, University of the Punjab, House No. 15, Street No. 39, Lahore, Pakistan.
E-mail addresses: raza_nauman@yahoo.com (N. Raza), sultans@lums.edu.pk (S. Sial), shahidsiddiqiprof@yahoo.co.uk (S.S. Siddiqi).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.12.016

[4]. We then consider numerical implementation of steepest descent in this Sobolev space for minimizing the Schrödinger functional associated with a cubic nonlinear Schrödinger equation subject to some constraint in a finite-difference and finite-element settings. Lastly, soliton solutions are mentioned. For the computational comparison an Intel Pentium 2.8 GHZ dual core machine with 512 MB RAM was used.

The nonlinear Schrödinger equation [15] is an example of a universal nonlinear model that describes many physical systems. The nonlinear Schrödinger equation appears in quantum mechanics [16] and is related to the Hamiltonian functional

$$H(\psi) = \int_V \alpha|\psi|^4 + \beta|\psi|^2 + |\nabla\psi|^2, \tag{1}$$

where V is a general bounded region. The coefficients α and β might vary across V and ψ is a complex field defined on V. This functional gives rise to the dynamics

$$i\psi_t = \nabla H(\psi). \tag{2}$$

The equation has applications in hydrodynamics, nonlinear optics, nonlinear acoustics, quantum condensates, heat pulses in solids and various other nonlinear instability phenomena. We are interested in finding soliton solutions of the nonlinear Schrödinger equation for which $\psi_t = 0$. The solitary wave (or soliton) is a wave that consists of a single symmetrical hump that propagates at uniform velocity without changing its shape.

2. Energy minimization associated with Schrödinger functionals

We will study numerical approximations for the system defined by the functional associated with a cubic nonlinear Schrödinger equation:

$$\phi(\psi) = \int_V \frac{\beta}{4}|\psi|^4 + \frac{1}{2}|\nabla\psi|^2 \tag{3}$$

subject to the condition

$$\int_V |\psi|^2 = N, \tag{4}$$

where V is a bounded region and N is a constant. $\psi = u + iv$ is a complex field in V. Without the above restriction (4) the absolute minimum of the energy is always reached at the trivial solution $\psi = 0$. We will see that (4) defines a subspace T that we will make use of.

It will be convenient to work with real quantities, so we consider numerical estimation of the minimum of

$$F(u, v) = \int_V \frac{\beta}{4}|u^2 + v^2|^2 + \frac{1}{2}(\nabla u^2 + \nabla v^2). \tag{5}$$

subject to the condition

$$\int_V u^2 + v^2 = N, \tag{6}$$

where N is a positive integer and V is a bounded region in two or three dimensions.

2.1. Finite-difference setting

We now consider numerical approximations to the functional (5) in one, two, and three spatial dimensions.

We will use a uniform grid for all numerical experiments in this paper. We think of the grid as being composed of nodes and edges which form cells. In one dimension, the cells are adjacent nodes connected by a line segment of length δ. In two dimensions, the cells consist of squares with nodes on the corners joined by line segments of length δ. In three dimensions, the cells are square cubes. We will define operators that estimate the value of a function u and its derivatives on the cell, given the value of u on the nodes.

We define an operator $D_0 : R^M \to R^{M-1}$ in one spatial dimension. This acts on a vector u of M real numbers considered to be the values of a function on the nodes of the grid and returns a vector $D_0(u) \in R^{M-1}$ of $M - 1$ real numbers considered to be values defined on the cells of the grid. This is done by averaging the values of u on the nodes of the cell.

We use the same name D_0 for the operators in two and three spatial dimensions that assign the average of u on the nodes of the cell to the cell.

In one spatial dimension, define an operator $D_1 : R^M \to R^{M-1}$ that takes the difference in u at the two nodes of a cell and divides by the internodal spacing δ to assign an estimated derivative in the x direction to the cell. In two dimensions, D_1 takes the differences between values of u joined by line segments lying in the x direction and averages them on a cell to assign an estimate of the x derivative to the cell. In three dimensions, D_1 averages four differences of u in the x direction.

Operators D_2 and D_3 are defined analogously for derivatives in the y and z directions.

A numerical analogue of (5) is

$$F(u,v) = \frac{\beta}{4}\langle D_0(u^2+v^2), D_0(u^2+v^2)\rangle + \sum_{i=1}^{d}\frac{1}{2}\langle D_i(u), D_i(u)\rangle + \sum_{i=1}^{d}\frac{1}{2}\langle D_i(v), D_i(v)\rangle, \tag{7}$$

where u and v are vectors in $R^M, R^M \times R^M$ or $R^M \times R^M \times R^M$ and d is the spatial dimension of the problem. For the rest of the paper we will only deal with real vectors.

F is strictly convex so the minimum is unique. We search for the minimum of F using descent techniques modified to account for the restriction on the norm (6).

The gradients $\nabla_u F$ and $\nabla_v F$ in $L_2 = H_0^2$ (the space of vectors in equipped with the inner product $\langle p,q\rangle = \sum p(i)q(i)$) for some functional F can be calculated by

$$F(u+h, v+k) = F(u,v) + \langle \nabla_u F, h\rangle + \langle \nabla_v F, k\rangle + O(h^2, k^2) \tag{8}$$

for test functions h and k. The gradients in this space are given by

$$\nabla_u F = \beta u D_0^t D_0(u^2+v^2) + \sum_{i=1}^{d} D_i^t D_i(u) \tag{9}$$

and

$$\nabla_v F = \beta v D_0^t D_0(u^2+v^2) + \sum_{i=1}^{d} D_i^t D_i(v). \tag{10}$$

The gradient points to the direction of greatest increase of F in the function space L_2. So, one could seek to minimize F by moving u and v one minimization step in the directions $-\nabla_u F$ and $-\nabla_v F$, recalculating the gradient, moving one minimization step, etc.:

- Calculate $\nabla_u F$ and $\nabla_v F$.
- Update u and v by $u \to u - \lambda\nabla_u F$, $v \to v - \lambda\nabla_v F$ where λ is some fixed positive number.
- Renormalize u and v.
- Repeat.

A line-search technique in which the step-size λ varies could be used but here we used a constant step-size in all that follows.

Consider how changing the vectors u and v in this way changes the value of $\sum u(i)^2 + v(i)^2$ before the renormalization step. This differs from the original value N (6) by $-\lambda u(i)\nabla_u F(i) - \lambda v(i)\nabla_v F(i) + O(\lambda^2)$. We can then renormalize in order to respect the constraint.

Another approach is to try to make the $O(\lambda)$ terms zero. That is, we want to project $\nabla_u F$ and $\nabla_v F$ onto vectors ψ and χ so that

$$\langle u, \psi\rangle + \langle v, \chi\rangle = 0. \tag{11}$$

This way the deviation from the constraint (6) is of order λ^2. We will call the subspace defined by (11) T. Projecting onto T does not remove the necessity for renormalization, however it makes the steepest descent process smoother.

The CFL condition [17] says that for stability in such an explicit scheme the step-size λ will be less than some number which depends on the nodal spacing. When the grid is made refined or if we go from one to two or three spatial dimensions the step-size will have to be reduced. Steepest descent in L_2 is easy to understand and to implement, but is increasingly inefficient as grids become finer.

Rather than abandoning the steepest descent, the gradient is reconsidered. The gradient was calculated above in the space L_2 which is R^M with the standard inner product $\langle,\rangle$. Instead of using the L_2 inner product we could define an inner product space H_1^2 which is R^M with inner product‘

$$(u,v) = \langle D_0 u, D_0 v\rangle + \sum_{i=1}^{d}\langle D_i u, D_i v\rangle. \tag{12}$$

The gradient $P\nabla_u F$ in H_1^2 can be found from the gradient $\nabla_u F$ in L_2 by solving

$$\left(D_0^t D_0 + \sum_{i=1}^{d} D_i^t D_i\right) P\nabla_u F = \nabla_u F, \tag{13}$$

where D_0^t and D_1^t are the adjoints of D_0 and D_1. Similarly, solving

$$\left(D_0^t D_0 + \sum_{i=1}^{d} D_i^t D_i\right) P\nabla_v F = \nabla_v F \tag{14}$$

gives $P\nabla_v F$.

The steepest descent algorithm in this new space is as follows:

- Calculate ∇_u and $\nabla_v F$.
- Project $\nabla_u F$ and $\nabla_v F$ on to H_1^2 by solving (13) and (14).
- Project $P\nabla_u F$ and $P\nabla_v F$ onto the subspace T.
- Update u and v by $u \to u - \lambda P\nabla_u F$, $v \to v - \lambda P\nabla_v F$ where λ is some fixed positive number.
- Renormalize u and v.
- Repeat.

For a one-dimensional problem, let the domain V of (5) to be the interval $[0,10]$. For the two-dimensional problem $V = [0,10] \times [0,10]$ and in the three-dimensional problem $V = [0,10] \times [0,10] \times [0,10]$.

Numerical experiments for the minimization of the Schrödinger functional as given by (5) with the constraint (6) were conducted as follows: Systems of M, M^2 and M^3 nodes were set up with $u = \sin x$ and v with random initial values for the one and two-dimensional cases and $u = \sin x$ and $v = 0$ for the three-dimensional case. The internodal spacing δ was the same in each direction in the two- and three-dimensional cases. The values of β and N were set to 1 and 10 for all the experiments. Minimization was done in L_2 and in H_1^2. The programs were terminated when the difference between new and old u and v values was less than 10^{-6}. For the gradients in H_1^2 we used the same step-size regardless of the internodal spacing δ. When the grid was refined in L_2 the step-size had to be decreased, otherwise the minimization became unstable as predicted by the CFL condition. The step-sizes reported for minimization in L_2 are the largest that could be used for a stable run. The functional was minimized. The total number of minimization steps, the largest value of λ that can be used, the minimum value of F and CPU time which are given in Tables 1–3.

We note that the finer the spacing, the less CPU time the Sobolev gradient technique uses in comparison to the usual steepest descent method. The step-size for minimization in L_2 has to decrease as the spacing is refined. From the tables we see that the results in H_1^2 are far better than in L_2.

The minimum energy states that we have found by the steepest descent technique are solitons since at the minimum energy the gradient of the functional is zero.

2.2. Finite-element setting

Consider again the functional (5) subject to the constraint (6). It is convenient in the finite-element setting to think in terms of integrals. We define the L_2 inner product as

Table 1
Numerical results of the Sobolev gradients algorithms compared with L_2 gradients in the one-dimensional case.

λ		Iterations		CPUs		$F(u,v)$		M
L_2	H_1^2	L_2	H_1^2	L_2	H_1^2	L_2	H_1^2	
0.0012	0.9	2606	14	0.4527	0.0156	281.84	281.84	201
0.00031	0.9	10032	25	3.0445	0.0312	281.89	281.89	401
0.000078	0.9	40041	58	29.67	0.2652	281.9	281.9	801
0.000018	0.9	173096	134	268.3	1.311	281.91	281.91	1601

Table 2
Numerical results of the Sobolev gradients algorithms compared with L_2 gradients in the two-dimensional case.

λ		Iterations		CPUs		$F(u,v)$		M
L_2	H_1^2	L_2	H_1^2	L_2	H_1^2	L_2	H_1^2	
0.18	0.9	2895	7	13.25	0.1092	43.05	43.03	51
0.092	0.9	9235	7	168.37	0.5775	48.93	48.93	101
0.048	0.9	25995	8	1741.2	2.513	57.43	57.23	201
0.018	0.9	>17444	10	>5394	15.234	69.6	68.56	401

Table 3
Numerical results of the Sobolev gradients algorithms compared with L_2 gradients in the three-dimensional case.

λ		Iterations		CPUs		$F(u,v)$		M
L_2	H_1^2	L_2	H_1^2	L_2	H_1^2	L_2	H_1^2	
0.12	0.9	763	8	21.53	1.048	8.514	8.516	21
0.031	0.9	2080	10	422.75	10.694	8.542	8.544	41
0.0075	0.9	>2635	27	>5394	331.68	8.549	8.55	81

$$\langle f, g\rangle = \int_V fg \tag{15}$$

and the H_1^2 inner product as

$$\langle f, g\rangle_S = \langle f, g\rangle + \langle f_x, g_x\rangle + \langle f_y, g_y\rangle. \tag{16}$$

This inner product takes spatial gradients into account, unlike inner product in L_2. The L_2 gradients $\nabla_u F$ and $\nabla_v F$ can be found by

$$F(u+h, v+k) = F(u, v) + \langle \nabla_u F, h\rangle + \langle \nabla_v F, k\rangle + O(h^2, k^2). \tag{17}$$

The Sobolev gradients $P\nabla_u F$ and $P\nabla_v F$ can be found by

$$F(u+h, v+k) = F(u, v) + \langle P\nabla_u F, h\rangle + \langle P\nabla_v F, k\rangle + \langle P\nabla_u F_x, h_x\rangle + \langle P\nabla_u F_y, h_y\rangle + \langle P\nabla_v F_x, k_x\rangle + \langle P\nabla_v F_y, k_y\rangle + O(h^2, k^2). \tag{18}$$

For our particular problem, we need to solve

$$\int_V \beta u(u^2+v^2)h + \int_V \nabla u \cdot \nabla h = \int_V P\nabla_u F h + \int_V \nabla P\nabla_u F \cdot \nabla h \tag{19}$$

and

$$\int_V \beta v(u^2+v^2)k + \int_V \nabla v \cdot \nabla k = \int_V P\nabla_v F k + \int_V \nabla P\nabla_v F \cdot \nabla k \tag{20}$$

in order to find the Sobolev gradients. Note that in the finite-element setting it is not necessary to find the L_2 gradient first and then find the Sobolev gradient next. The software we used can find the different gradients by solving (17) and (18) directly.

The steepest descent algorithm is as follows:

- Calculate $P\nabla_u F$ and $P\nabla_v F$ from (18).
- Project $P\nabla_u F$ and $P\nabla_v F$ onto the subspace T.
- Update u and v by $u, v \to u, v - \lambda P\nabla_u F, P\nabla_v F$ where λ is some fixed positive number.
- Renormalize u and v.
- Repeat.

In the two-dimensional case we let V be the circular disk centered at the origin of radius 10 with an oval region removed that has border $x(t) = 8\cos(t), y(t) = 2\sin(t)$ with $t \in [0, 2\pi]$. The initial state was $u = \sin x$ and $v = \cos y$. The program was terminated when the difference between new and old u and v values was less than 10^{-6}.

We used the free finite-element software FreeFem++ [18] for this problem. FreeFem++ requires one to specify the borders of the region and the number of nodes required on each border. The software then creates a mesh. We did numerical experiments with $M = 10$, 20 and 30 nodes on each border. The energy was minimized using steepest descent steps with both L_2 and H_1^2. Comparisons of the step-size, number of steps and CPU time are given in Table 4.

Similarly, for the three-dimensional case we let V be the ball centered at the origin of radius 8. The initial state was set up $u = \sin x$ and $v = \cos y$. The program was terminated when the difference between new and old u and v values was less than 10^{-6}.

Table 4
Numerical results of the Sobolev gradients algorithms compared with L_2 gradients in the two-dimensional case.

λ		Iterations		CPUs		$F(\psi)$		M	Triangles
L_2	H_1^2	L_2	H_1^2	L_2	H_1^2	L_2	H_1^2	–	–
0.1	0.9	243	38	66.15	10.41	25.81	25.794	10	20
0.01	0.9	1508	77	5453.7	282.9	27.03	26.551	20	88
0.003	0.9	>314	137	>5963	2511.1	57.9374	25.72	30	198

Table 5
Numerical results of the Sobolev gradient algorithms compared with L_2 gradients in the three-dimensional case.

λ		Iterations		CPUs		$F(\psi)$		M
L_2	H_1^2	L_2	H_1^2	L_2	H_1^2	L_2	H_1^2	
0.01	0.6	1169	39	26.14	1.4	3.125	3.1255	5
0.002	0.6	3681	37	1178.2	15.9	18.366	18.366	10
0.00005	0.6	>6101	92	>1984	51.2	51.2	51.2	15

We used the free finite-element software FreeFem3*d* [18] for this problem. FreeFem3*d* requires one to specify the number of nodes on each axis. The software then creates a mesh. We did numerical experiments with M = 5, 10 and 15 nodes on each axis. We minimized the energy using steepest descent steps with both L_2 and H_1^2 and compared the step-size, number of steps and CPU time in Table 5.

We see that as the mesh becomes finer, i.e. M increases, the step-size λ needs to decrease drastically whereas this is not the case for the Sobolev gradient, and accordingly, the number of required iterations does not substantially increase either. In the two-dimensional case for $M = 30$ the minimization was not finished at 324 steps using the L_2 gradient but concluded after 137 steps using the Sobolev gradient. In the three-dimensional case for $M = 15$ the minimization was not finished at 6101 steps using the L_2 gradient but concluded after 92 steps using the Sobolev gradient.

3. Summary and conclusions

A Sobolev gradient scheme has been developed for both finite-difference and finite-element settings for the energy minimization of Nonlinear Schrödinger functionals. The Sobolev gradient technique is computationally more efficient than the usual steepest descent method as the spacing of the numerical grid is made finer and the dimension of the problem is increased.

Acknowledgments

We acknowledge the enabling role of the Higher Education Commission Islamabad, Pakistan, and appreciate its financial support through the *Indigenous PhD 5000 Fellowship Program Batch-I*.

References

[1] J.W. Neuberger, Sobolev Gradients and Differential Equations, Springer Lecture Notes in Mathematics, 1670, Springer-Verlag, New York, 1997.

[2] W.T. Mahavier, A numerical method utilizing weighted Sobolev descent to solve singular differential equations, Nonlinear World 4 (1997) 4.

[3] C. Beasley, Finite Element Solution to Nonlinear Partial Differential Equations. Ph.D. Thesis, University of North Texas, Denton, TX, 1981.

[4] S. Sial, J. Neuberger, T. Lookman, A. Saxena, Energy minimization using Sobolev gradients: application to phase separation and ordering, J. Comput. Phys. 189 (2003) 88–97.

[5] S. Sial, J. Neuberger, T. Lookman, A. Saxena, Energy minimization using Sobolev gradients: finite element setting, in; Chaudhary, Bhatti (Eds.), Proceedings of the World Conference on 21st Century Mathematics, Lahore, Pakistan, March 2005.

[6] S. Sial, Sobolev gradient algorithm for minimum energy states of s-wave superconductors: finite element setting, Supercond. Sci. Technol. 18 (2005) 675–677.

[7] J. Karatson, L. Loczi, Sobolev gradient preconditioning for the electrostatic potential equation, Comput. Math. Appl. 50 (2005) 1093–1104.

[8] J. Karatson, I. Farago, Preconditioning operators and Sobolev gradients for nonlinear elliptic problems, Comput. Math. Appl. 50 (2005) 1077–1092.

[9] J. Karatson, Constructive Sobolev gradient preconditioning for semilinear elliptic systems, Electron. J. Differ. Equations 75 (2004) 1–26.

[10] J. Garcia-Ripoll, V. Konotop, B. Malomed, V. Perez-Garcia, A quasi-local Gross–Pitaevskii equation for attractive Bose–Einstein condensates, Math. Comput. Simulat. 62 (2003) 21–30.

[11] B. Brown, M. Jais, I. Knowles, A variational approach to an elastic inverse problem, Inverse Probl. 21 (2005) 1953–1973.

[12] I. Knowles, A. Yan, On the recovery of transport parameters in groundwater modelling, J. Comput. Appl. Math. 171 (2004) 277–290.

[13] J. Garcia-Ripoll, V. Perez-Garcia, Optimizing Schrodinger functionals using Sobolev gradients: application to quantum mechanics and nonlinear optics, SIAM J. Sci. Comput. 23 (2001) 1316–1334.

[14] W.B. Richardson, Steepest descent using smoothed gradients, Appl. Math. Comput. 112 (2000) 241–254.

[15] G. Agrawal, Nonlinear Fiber Optics (Optics and Photonics), Academic Press, London, 2001.

[16] NLSE in quantum mechanics. <ttp://tosio.math.toronto.edu/wiki/index.php/schrodinger_equations>.

[17] R. Courant, K.O. Friedrichs, H. Lewy, Uber die Partiellen Differenzengleichungen der Mathematisches Physik, Math. Ann. 100 (1928) 32–74.

[18] F. Hecht, O. Pironneau, K. Ohtsuka, FreeFem++ Manual. <www.freefem.org>.