Journal of Computational Physics 229 (2010) 1621–1625

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Approximating time evolution related to Ginzburg–Landau functionals via Sobolev gradient methods in a finite-element setting

Nauman Raza [a,*], Sultan Sial [b], S. Siddiqi [c]

[a] Department of Mathematics, University of the Punjab, Lahore, Research Associate, Lahore University of Management Sciences, Lahore, Pakistan
[b] Department of Mathematics, Lahore University of Management Sciences, Opposite Sector U, DHA, Lahore Cantt 54792, Pakistan
[c] Department of Mathematics, University of the Punjab, Lahore, Pakistan

ARTICLE INFO

Article history:
Received 18 February 2009
Received in revised form 19 October 2009
Accepted 27 October 2009
Available online 1 November 2009

Keywords:
Sobolev gradient
Model A

ABSTRACT

Sobolev gradients have previously [1] been used to approximate time evolution related to a model A functional in a finite-difference setting in this journal. Here a related approach in a finite-element setting is discussed.

1. Introduction

An important approach to the solution of PDEs is to seek a critical point of a functional, constructed so that the equation can be considered to be solved when the functional is minimized. The recent theory of Sobolev gradients [2] gives a unified approach for such problems. Sobolev gradients have been used for ODEs [2,3] in a finite-difference setting, PDEs in finite-difference [3] and finite-element settings [4], minimizing energy functionals associated with Ginzburg–Landau models in finite-difference [5] and finite-element [6,7] settings, the electrostatic potential equation [8], nonlinear elliptic problems [9], semilinear elliptic systems [10], simulation of Bose–Einstein condensates [11], inverse problems in elasticity [12] and groundwater modelling [13]. [2] has a detailed analysis regarding the construction and the application of Sobolev gradients. For a quick overview of Sobolev gradients, applications and some open problems in the subject we refer to [14].

Sobolev gradients are also useful for preconditioning for linear and nonlinear problems. Sobolev preconditioning [15] has been tested on some first order and second order linear and nonlinear problems and it is found comparable in terms of efficiency and stability with other methods such as Newton's method and Jacobi method. For differential equations with non-uniform behavior on long intervals, Sobolev gradients have proved effective if we divide the interval of interest into pieces and take a recursive approach [16]. Sobolev gradients have interesting applications in the field of geometric modelling [17]. It has been proved [17] that Sobolev gradient is a very useful tool for minimizing functionals that pertain to the length of curves, curvatures, surface area etc. Recently, [18] have shown the possible applications of Sobolev gradient technique for systems of Differential Algebraic Equations.

Sobolev gradients have been discussed in [1] as a method for approximating time evolution related to Ginzburg–Landau functionals. In this article, we would give the equivalent algorithm in a finite-element setting.

* Corresponding author.
E-mail addresses: raza_nauman@yahoo.com (N. Raza), sultans@lums.edu.pk (S. Sial), shahidsiddiqiprof@yahoo.co.uk (S. Siddiqi).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.10.048

It is found that the approach is efficient while, at the same time, the algorithm retains the simplicity of less efficient steepest descent methods. Results are reported for numerical experiments. All numerical experiments are carried out on an Intel Xeon 3.2 GHZ dual processor machine with 2GB RAM.

2. Model A time evolution

The minimization of a Model A Ginzburg–Landau free energy functional

$$F(u) = \int_{\omega} \frac{u^4}{4} - \frac{u^2}{2} + \frac{\kappa}{2}|\nabla u|^2 \tag{1}$$

has been considered in [5]. The static and dynamical properties of this model have been extensively studied, primarily in numerical work related to coarsening and growth of domains [19–21].

In the continuous case, the related Ginzburg–Landau time evolution is

$$u_t = -\nabla F(u), \tag{2}$$

which on the interior of the system is

$$u_t = u - u^3 + \nabla^2 u, \tag{3}$$

and this equation has been known for many years as the time-dependent Ginzburg–Landau (TDGL) equation, after the pair of Russian physicists who first used it in connection with modelling superconductivity. TDGL equation is the kinetic equation for the temporal evolution of a continuum field, which assumes the rate of evolution of the field is linearly proportional to the thermodynamical driving force. The computation model based on this equation is also called phase field model. Phase field simulation can predict quite beautiful patterns of microstructures of material. It has been widely applied to simulating the evolution of microstructure by choosing different field variables. For example, continuum phase field models has been employed to describe the pattern formation in phase separating alloys [22] and the nanoscale pattern formation of an epitaxial monolayer [23].

The method suggested for evolving systems from one time to another in the primary reference for Sobolev gradients [2] is as follows. Form a functional

$$\phi(u) = \left\| u - f + \frac{\delta_t}{2}(u - u^3 + \nabla^2 u) + \frac{\delta_t}{2}(f - f^3 + \nabla^2 f) \right\|^2, \tag{4}$$

where f represents the system at a time t and u represents the system at a time $t + \delta_t$. When this functional is minimized sufficiently, the system is considered to have evolved to the next time step. We will follow this approach with the modification that we consider the minimization of a functional

$$\phi(u) = ||u - f + \delta_t(u - u^3 + \nabla^2 u)||^2 \tag{5}$$

for a fully implicit scheme. So the problem is considered to be solved when $\phi(u)$ is smaller than some set tolerance. We suggest another method.

We form an associated functional

$$G(u) = \int_{\Omega} \delta_t \frac{u^4}{4} + (1 - \delta_t)\frac{u^2}{2} - fu + \delta_t \frac{\kappa}{2}|\nabla u|^2 \tag{6}$$

over some two or three-dimensional region Ω subject to Dirichlet boundary conditions and seek to minimize this. This functional is a convex functional that guarantees global minima at $\nabla G(u)$, a solution to the problem (3). The aim is to find the gradient of a convex functional $G(u)$ associated with the problem and use the gradient in steepest descent minimization process to find the zero of the functional that is minima of $G(u)$ and a solution to the original problem.

3. Gradients and steepest descent

The gradient $\nabla G(u)$ in $L_2(\Omega) = H_0^2(\Omega)$ (the space of square integrable functions on some two- or three-dimensional domain Ω) for some functional $G(u)$ is defined by

$$G(u + h) = G(u) + \langle \nabla G(u), h \rangle + O(h^2), \tag{7}$$

where h is a small test function.

The gradient $\nabla G(u)$ in (7) points to the direction of greatest increase of G in the function space L_2 of square integrable functions. So, one could seek to minimize G by moving one minimizing step in the direction $-\nabla G(u)$, recalculating the gradient, moving one minimizing step, etc.

1. Calculate $\nabla G(u)$.
2. Update u by $u \to u - \lambda \nabla G(u)$ where λ is some fixed positive number.
3. Repeat until $\nabla G(u)$ is less than some set tolerance.

A more sophisticated scheme would be to vary the step size λ so as to optimize each minimization step but such line search techniques are not discussed here. We desire to develop a very simple algorithm and then demonstrate its efficiency.

The CFL condition [24] implies a problem with the steepest descent approach in L_2. When the grid is made finer or if we go to higher dimensional version of the problem the step size λ will have to be reduced.

4. Sobolev gradients

That steepest descent is inefficient is something well-known to numerical analysts. The philosophy of Sobolev gradients is demonstrated in the way the cause of this inefficiency is viewed. Rather than abandoning steepest descent the gradient is reconsidered. The gradient had been calculated in the space of square integrable functions $L_2(\Omega) = H_0^2(\Omega)$ with the standard inner product

$$\langle u, v\rangle = \int_\Omega uv. \tag{8}$$

An alternative inner product to (8) is

$$\langle u, v\rangle_S = \langle u, v\rangle + \langle \nabla u, \nabla v\rangle. \tag{9}$$

The inner product (9) for the Sobolev space $H_1^2(\Omega)$ takes spatial gradients into account, unlike (8). The Sobolev gradient $\nabla_s G(u)$ satisfies

$$G(u+h) = G(u) + \langle \nabla_s G(u), h\rangle + \langle \nabla\nabla_s G(u), \nabla h\rangle + O(h^2) \tag{10}$$

for test function h. For our particular problem, we need to solve

$$\int_\Omega (\delta_t u^3 + (1-\delta_t)u - f)h - \int_\Omega \kappa\delta_t \nabla u\cdot\nabla h = \int_\Omega \nabla_s G(u)h + \int_\Omega \nabla\nabla_s G(u)\cdot\nabla h \tag{11}$$

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, as it is in a finite-difference setting [1].

The steepest descent algorithm in H_1^2 is as follows:

- Calculate $\nabla_s G(u)$ from (10).
- Update u by $u \to u - \lambda\nabla_s G(u)$ where λ is some fixed positive number.
- Repeat until $\nabla_s G(u)$ is less than some set tolerance.

In the case of Dirichlet boundary conditions, the value of u is fixed on the boundaries of the system and so we look for gradients that are zero on the boundary of Ω. So one wishes to use not $\nabla G(u)$ but $\pi\nabla G(u)$ where π is a projection that sets values of test function h zero on the boundary of the system. We use Freefem ++ [25], a free software designed to solve partial differential equations using the finite-element method. This software has the facility to define the gradient zero at the boundary.

Thus we need to solve

$$\pi\int_\Omega (\delta_t u^3 + (1-\delta_t)u - f)h - \int_\Omega \kappa\delta_t\nabla u\cdot\nabla h = \pi\int_\Omega P\nabla G(u)h + \int_\Omega \nabla P\nabla G(u)\cdot\nabla h. \tag{12}$$

It is seen that when using the Sobolev gradient the step size λ does not have to be reduced as the numerical grid becomes finer and the number of minimization steps remains reasonable. At the same time the conceptual simplicity and elegance of the steepest descent algorithm has been retained.

5. Numerical results

For the two-dimensional case, we let Ω 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 $u = 0.0$ and the Dirichlet condition was that $u = -1$ on the outer boundary and $u = 1$ on the inner boundary on the boundaries. We let $\kappa = 1$ and time step $\delta_t = 0.95$.

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. FreeFem++ solves the equations of the same type as (12) that determine the L_2 and Sobolev gradients. We did numerical experiments with $M = 20, 40, 80$ and 160 nodes on each border. The system was evolved over 15 time steps. For each time step δ_t the functional defined by (6) was minimized using steepest descent steps with both L_2 and H_1^2 until the infinity norm of the gradient was less than 10^{-6}. The step size, number of steps and the CPU time are given in Table 1.

For the three-dimensional case, we let Ω be the ball centered at the origin of radius 8. The initial state was $u = 2.0$ with Dirichlet boundary condition. We let $\kappa = 1$ and time step $\delta_t = 0.95$. The system was evolved over 15 time steps. For each

Table 1
Numerical results of steepest descent in L_2, H_1^2 using δ_t=0.95 in the two-dimensional case.

λ		Iterations		CPUs		M	Triangles
L_2	H_1^2	L_2	H_1^2	L_2	H_1^2	–	–
2.0×10^{-2}	0.7	6026	205	83.1	2.9	20	88
4.0×10^{-3}	0.7	30,604	206	1504.0	9.7	40	350
7.0×10^{-4}	0.7	>96,591	207	>50400.0	45.0	80	1350

Table 2
Numerical results of steepest descent in L_2, H_1^2 using $\delta_t = 0.95$ in the three-dimensional case.

λ		Iterations		CPUs		M
L_2	H_1^2	L_2	H_1^2	L_2	H_1^2	-
5.0×10^{-3}	0.9	903	31	26.8	0.98	8
1.0×10^{-3}	0.9	9266	43	3 704.1	14.31	16
2.2×10^{-4}	0.9	>5877	180	>9120.0	707.73	32

time step δ_t the functional defined by (6) was minimized using steepest descent steps with both L_2 and H_1^2 until the infinity norm of the gradient was less than some set tolerance.

We used the free finite-element software FreeFem3*d* [25] 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 $M = 8, 16$ and 32 nodes on each axis. A record was kept of the total number of minimization steps, the largest value of λ that can be used, and the CPU time in Table 2.

We see that as the mesh becomes finer with increasing M, the step size λ needs to decrease drastically whereas this is not the case for the Sobolev gradient and so the number of iterations required does not increase substantially either. In the two-dimensional case, for $M = 80$ the minimization of time evolution Model A was not finished at 96,591 steps using the L_2 gradient but concluded after 207 steps using the Sobolev gradient. At $M = 160$ the L_2 gradient is almost divergent. Similarly, in the three-dimensional case, for $M = 32$ the minimization of time evolution Model A was not finished at 5877 steps using the L_2 gradient but concluded after 180 steps using the Sobolev gradient.

6. Summary and conclusions

A scheme has been developed for time evolution model A Ginzberg–Landau functionals based on the Sobolev gradient technique in a finite-element setting. The Sobolev gradient technique is computationally more efficient than the usual steepest descent method as the mesh is made finer, the dimension of the problem is increased.

The FreeFem++ software was quite easy to use and the code is only a few pages long. These problems have previously been attempted using Sobolev gradients in a finite-difference setting [1] and we have shown here that the Sobolev gradient approach has advantages in a finite-element setting.

Acknowledgments

We acknowledge the enabling role of the Higher Education Commission Islamabad, Pakistan, and appreciate its financial support through the *Indigenous Ph.D. 5000 Fellowship Program Batch-I*. We are also very grateful to the refree for his helpful suggestions.

References

[1] N. Raza, S. Sial, S. Siddiqi, Sobolev gradient approach for the time evolution related to energy minimization of Ginzberg–Landau functionals, J. Comput. Phys. 228 (2009) 2566–2571.
[2] J.W. Neuberger, in: Sobolev Gradients and Differential Equations, Springer Lecture Notes in Mathematics, vol. 1670, Springer-Verlag, New York, 1997.
[3] W.T. Mahavier, A numerical method utilizing weighted Sobolev descent to solve singular differential equations, Nonlinear World 4 (1997) 435–455.
[4] C. Beasley, Finite Element Solution to Nonlinear Partial Differential Equations. Ph.D. Thesis, University of North Texas, Denton, TX, 1981.
[5] S. Sial, J.W. Neuberger, T. Lookman, A. Saxena, Energy minimization using Sobolev gradients: application to phase separation and ordering, J. Comput. Phys. 189 (2003) 88–97.
[6] 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.
[7] S. Sial, Sobolev gradient algorithm for minimum energy states of s-wave superconductors: finite-element setting, Supercond. Sci. Technol. 18 (2005) 675–677.
[8] J. Karatson, L. Loczi, Sobolev gradient preconditioning for the electrostatic potential equation, Comput. Math. Appl. 50 (2005) 1093–1104.
[9] J. Karatson, I. Farago, Preconditioning operators and Sobolev gradients for nonlinear elliptic problems, Comput. Math. Appl. 50 (2005) 1077–1092.

[10] J. Karatson, Constructive Sobolev gradient preconditioning for semilinear elliptic systems, Electron. J. Diff. Eq. 75 (2004) 1–26.
[11] 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.
[12] B. Brown, M. Jais, I. Knowles, A variational approach to an elastic inverse problem, Inverse Probl. 21 (2005) 1953–1973.
[13] I. Knowles, A. Yan, On the recovery of transport parameters in groundwater modelling, J. Comput. Appl. Math. 171 (2004) 277–290.
[14] R.J. Renka, J.W. Neuberger, Sobolev gradients: Introduction, applications, Contemp. Math. 257 (2004) 85–99.
[15] W.B. Richardson, Sobolev preconditioning for the Poisson–Boltzman equation, Comput. Method Appl. Mech. Eng. 181 (2000) 425–436.
[16] D. Mujeeb, J.W. Neuberger, S. Sial, Recursive forms of Sobolev gradient for odes on long intervals, Int. J. Comput. Math. 87 (2008) 1727–1740.
[17] R.J. Renka, Geometric curve modeling with Sobolev gradients, 2006, <ww.cse.unt.edu/renka/papers/curves.pdf>.
[18] R. Nittka, M. Sauter, Sobolev gradients for differential algebraic equations, J. Diff. Eq. 42 (2008) 1–31.
[19] P.C. Hohenberg, B.I. Halperin, Theory of dynamic critical phenomena, Rev. Mod. Phys. 49 (1977) 435–479.
[20] T.M. Rogers, K.E. Elder, R.C. Desai, Numerical study of the late stages of spinodal decomposition, Phys. Rev. B 37 (1988) 9638–9649.
[21] J.W. Cahn, J.E. Hilliard, Free energy of a nonuniform system I. Interfacial free energy, J. Chem. Phys. 28 (1958) 258–267.
[22] H. Nishimori, A. Onuki, Pattern formation in phase-separating alloys with cubic symmetry, Phys. Rev. B 42 (1990) 980–983.
[23] W. Lu, Z. Suo, Symmetry breaking in self-assembled monolayers on solid surfaces: anisotropic surface stress, Phys. Rev. B 65 (2002) 085401.
[24] R. Courant, K.O. Friedrichs, H. Lewy, Uber die Partiellen Differenzengleichungen der Mathematisches Physik, Math. Ann. 100 (1928) 32–74.
[25] F. Hecht, O. Pironneau, K. Ohtsuka, FreeFem++Manual, <www.freefem.org>.