Numerical Experiments on the Ginzburg-Landau Equations

*Man Kam Kwong**

Abstract. We report on some numerical experiments done on the Ginzburg-Landau equations of the theory of superconductivity. The equations under periodic boundary conditions are solved numerically by using a nonstandard discretization and the sweeping algorithm. The programs, written in Matlab and Fortran, work in an interactive mode and include graphics display.

1991 Mathematics Subject Classification: 65F05, 65N22, 82D55.

1. Introduction

In this article we report on experiments we did on solving the Ginzburg-Landau (GL) equations on a Sun Sparc II workstation. Our contributions consist of a careful formulation of the discrete form of the equations, a choice of a suitable gauge transformation to simplify the equations, a new "sweeping" algorithm, and a practical implementation of the solution procedure that includes run-time graphics display. More details will be given in a forthcoming paper [9].

The GL equations arise in the theory of superconductivity. For a recent survey of previous work done on the equations, see the articles [3] by Chapman, Howison, and Ockendon and [5] by Du, Gunzburger, and Peterson.

* Present address: Mathematics and Computer Science Division Argonne National Laboratory Argonne, IL 60439-4844. This work was supported by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.

Theoretical physicists have long been able to extract useful information from the GL equations by constructing their solutions "analytically," based on series approximations, or by ignoring certain unimportant variations in the solutions. One of the triumphs of this approach was the prediction by Abrikosov [1] that certain superconductors, classified as Type II, admit vortices of supercurrent, a fact confirmed by experiments only ten years later. Numerical solution of the full equations, especially for the simulation of vortices, has been successful only in recent years. The different approaches include simulated annealing (Doria et al.[4]), the relaxation method (Adler and Piran [2], Wang and Hu [13]), optimization (Garner et al. [7]), and finite elements (Du, Gunzburger, and Peterson [5,6]).

Although simulated annealing, being a Monte Carlo method, consumes many computing cycles, it carries a better assurance for finding the global minimizer of the energy functional. All other approaches yield local critical points which, although they are solutions to the GL equations, may not be the physically meaningful global minimizer. The relaxation method is simple in theory, but time-consuming because small time steps are required to avoid instability of the numerical scheme. Wang and Hu applied it only to the case of a single vortex. More recent work by Maekawa, Kato, and Enomoto [12] employs the same technique to treat the time-dependent GL equations. The optimization approach makes use of state-of-the-art optimization and linear algebra techniques in advanced scientific computing and is the choice for designing large-scale production codes. The finite elements method is important because it is applicable to arbitrary geometry and admits higher-order refinements, while the other methods are limited to first-order finite difference approximations.

Our work falls between the optimization and the finite element approaches. Our code runs on a Sparc II with reasonable speed; it can be easily parallelized to work faster on larger computers. It gives the user more control over the experiment. For instance, parameters can be altered interactively and the experiment repeated without having to recompile the program; The results are displayed graphically for the user to monitor the progress. These capabilities are made possible by using the Matlab package, which provides a powerful programming environment. The code is able to employ finer grids to give smoother solutions than those achievable in [6]. Unlike the optimization approach, our method can be and has actually been successfully modified to solve the time-dependent GL equations.

2. The Discrete GL Model

We discuss only the two-dimensional case in this paper; the generalization to the full three-dimensional case is straightforward. The superconducting material is in the form of a thin film occupying a domain in R^2. When placed in a magnetic field, with the direction of the field perpendicular to the film, Type-II superconductors form vortices of superconducting current. Close to regular hexagonal arrangements of these vortices have been observed experimentally. Distortions can be attributed to impurities and defects in the material. A typical numerical result is shown in Figure 1. In the periodic GL model, it is assumed that, at least deep into the interior of the domain when boundary effects are negligible, the magnetic state of a pure and homogeneous material varies doubly periodically: one can carve out a core region Ω such that the state of the entire superconductor, except for a thin boundary layer, can be accurately reproduced by extending that of Ω periodically. Traditionally (see, for example, [4]), Ω is chosen to be a rectangle, such as that shown on the left in Figure 1. In order to reproduce a regular hexagonal lattice, the aspect ratio of the rectangle has to be $1 : \sqrt{3}$.

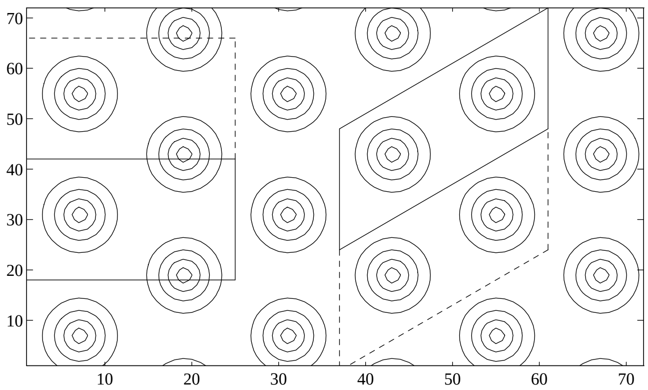


Figure 1. Core regions

Du, Gunzburger, and Peterson [6] suggested the more general parallelogram region, such as that shown on the right of Figure 1. For a hexagonal lattice, the acute angle of the parallelogram is 60°. The two special choices shown here in Figure 1 are actually equivalent. To see this, we can dissect the parallelogram into two triangles as shown in Figure 2, and translate the upper one to form a rectangle that is congruent to the rectangular region.

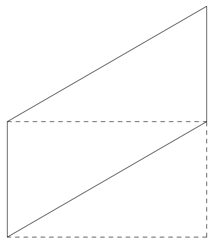


Figure 2. Equivalence of two special core regions

According to Ginzburg and Landau [8], the magnetic state of the material is completely determined by two functions, a complex-valued scalar function ψ and a (two-dimensional in our two-dimensional version) vector potential $\mathbf{A} = (A, B)$. The fundamental postulate is that for equilibrium these functions seek to minimize an energy functional, which, in dimensionless units, has the form

$$\mathcal{G}(\psi, \mathbf{A}) = \int_\Omega \left(\frac{|\psi|^4}{2} - |\psi|^2 + |(\nabla - i\mathbf{A})\psi|^2 + \kappa^2 |\nabla \times \mathbf{A}|^2 \right) d\mathbf{x}, \tag{2.1}$$

where κ is a constant characteristic of the metal. The superconductor is classified as Type II if $\kappa > 1/\sqrt{2}$, and as Type I otherwise.

The functional is invariant under the so-called gauge transformations. Let χ be any real-valued function on Ω. Then the transformed pair $(\overline{\psi}, \overline{\mathbf{A}})$

$$\overline{\psi} = \psi e^{i\chi}, \quad \overline{\mathbf{A}} = \mathbf{A} + \nabla\chi, \tag{2.2}$$

yields the same energy as that of $(\psi, \mathbf{A})$. Indeed, $(\psi, \mathbf{A})$ and $(\overline{\psi}, \overline{\mathbf{A}})$ are considered as equivalent representatives of the same magnetic state of the material.

This gauge invariance has more than one consequence. First, there is not only one single minimizer to the free energy, but an entire class of gauge equivalent solutions. Second, the assumption that the magnetic state varies periodically does not imply that $(\psi, \mathbf{A})$ varies periodically, but only that they are gauge equivalent on opposite edges of Ω. These facts gives rise to complications in the formulation of the boundary value problem, but they can also be used to our advantage. We can confine our search to solutions that have special properties. The technique has traditionally been called "fixing a gauge." For instance, the London gauge requires that $\nabla \cdot \mathbf{A} = 0$, and Wang and Hu note that B can be chosen independent of y. An excellent discussion of gauge fixing can be found in Du, Gunzburger, and Peterson [5]. In [9], we show the existence of a gauge stronger than that of Wang and Hu and compatible with a simplified set of "quasi-periodic" boundary conditions. To summarize:

A is fully periodic and is constant along the top edge,
B and ψ are periodic with respect to the top and bottom edges,
B is independent of y and has a jump of h on the vertical edges, and
ψ has a phase change e^{ihy} on the vertical edges.

Here $h = 2\pi n/L_y$ is a constant determined by the number of vortices n in Ω and the length L_y of the vertical edge of Ω. In turn, h determines the external magnetic field. The fact that B is independent of y significantly reduces the total number of unknowns to be solved. Another advantage of this choice of gauge is that the ensuing differential equations for A and B are "decoupled" in the sense that a cross-term involving B is absent in the equation for A and vice versa. This makes it simple to solve for A and B in each iterative step.

Gauge field theorists do not use standard finite difference methods on the continuous form of the energy functional for numerical purposes. With such methods, the discretized form of the gauge transform no longer preserves the energy, as a result of truncation errors. Field theorists have adopted the following modifications (see, for example, [2]). Suppose that Ω is discretized as usual with a uniform rectangular grid, with grid spacings of h_x and h_y. The order parameter ψ takes values at each of the grid points. The x-component A of $\mathbf{A}$ takes values on the bond (line) joining a grid point with its right-hand neighbor, while B takes values on the bond joining a grid point with its upper neighbor. To simplify the formulas, we define

$$U = \mathrm{e}^{iAh_x} \quad \text{and} \quad V = \mathrm{e}^{iBh_y}. \tag{2.3}$$

The free energy is then approximated by the discrete functional

$$\begin{aligned} \mathcal{G}_d = \sum_{\text{grid}} \Bigg(& \frac{|\psi|^4}{2} - |\psi|^2 + \left|\frac{\psi^{\rightarrow} - U\psi}{h_x}\right|^2 + \left|\frac{\psi^{\uparrow} - V\psi}{h_y}\right|^2 \\ & + \kappa^2 \left|\frac{A^{\uparrow} - A}{h_y}\right|^2 + \kappa^2 \left|\frac{B^{\rightarrow} - B}{h_x}\right|^2 \Bigg) h_x h_y . \end{aligned} \tag{2.4}$$

The arrow direction indicates the appropriate neighbors (boundary grid points have neighbors that are outside of Ω, but these can be interpreted as the appropriate value on grid points lying on the the opposite edge, modified according to the given boundary conditions). The conventional justification for this formula is that after an expansion in Taylor series, (2.1) and (2.4) agree up to first-order terms. Invariance is restored under a discrete gauge transformation:

$$\overline{\psi} = \psi \mathrm{e}^{i\chi}, \quad \overline{A} = A + \frac{\chi^{\rightarrow} - \chi}{h_x}, \quad \overline{B} = B + \frac{\chi^{\uparrow} - \chi}{h_y}. \tag{2.5}$$

In [9] we give a new interpretation to the discrete functional, clarify what "A and B are defined on the bonds" means, and show that the approximation offered by (2.4) is, in fact, of a higher order than is traditionally believed.

There is another disadvantage of working directly with the discretized form of (2.2). The order parameter ψ is a highly oscillatory function, as a result of large variations of its phase. A moderately fine spatial grid, as is allowable in practice, is usually not adequate to resolve ψ in regions of high oscillation. In such cases, the finite difference $(\psi^{\rightarrow} - \psi)/h_x$ can be a very poor approximation for the derivative $\partial\psi/\partial x$. On the other hand, in (2.4) it is the difference between $\psi^{\rightarrow}$ and $U\psi$ that is being computed. As pointed out in [9], we are actually approximating the derivative $\partial(\psi e^{i\int A})/\partial x$ by means of a central difference. The product function $U\psi$ oscillates much less than ψ itself. We have, therefore, chosen (2.4) as the starting point of our numerical method.

The Euler equations corresponding to the minimization of (2.4) are then

$$\frac{U^{\leftarrow}\psi^{\leftarrow} - 2\psi + U^{*}\psi^{\rightarrow}}{h_x^2} + \frac{V^{\downarrow}\psi^{\downarrow} - 2\psi + V^{*}\psi^{\uparrow}}{h_y^2} + (1 - |\psi|^2)\psi = 0, \tag{2.6}$$

$$\frac{A^{\uparrow} - 2A + A^{\downarrow}}{h_y^2} + \frac{\Im\,(\psi^{\rightarrow}\psi^{*}U^{*})}{h_x} = 0, \tag{2.7}$$

and

$$\frac{B^{\leftarrow} - 2B + B^{\rightarrow}}{h_x^2} + \frac{1}{N_y}\sum_{\text{column}} \frac{\Im\left(\psi^{\uparrow}\psi^{*}V^{*}\right)}{h_y} = 0, \tag{2.8}$$

where $\Im(.)$ denotes the imaginary part of a complex number, and N_y is the number of rows in the grid. Although (2.6) is probably known to everyone who uses the discrete functional, it has not been mentioned in the literature. Note the simplicity of the linear operator on ψ represented by the first two terms (as compared to the corresponding Euler equation obtained from (2.1)); it is a generalized discrete Laplace operator, having variable coefficients instead of the familiar $\{1, -2, 1\}$. Each of (2.6) and (2.7) stands for as many equations as there are grid points, while (2.8) stands for only N_x equations. Note also that there would be an extra term involving B in (2.7) and one involving A in (2.8), if we had not used our special gauge.

To conclude the section, we summarize the numerical problem we are tackling:

> Given a rectangular region Ω of suitable size $(L_x \times L_y)$, a grid (with $N_x \times N_y$ points), and the number of vortices in Ω (n which is usually 2, 4, or 8), we wish to find the $2N_x \times N_y + N_x$ unknown values: ψ (at each grid point), A (on each bond), and B (on each column of bonds) that satisfy the GL equations (2.6)–(2.7) and the appropriate boundary conditions.

3. The Solution — the Sweeping Algorithm

We describe in this section how the solution to the problem is computed by using an iterative scheme. The system of equations (2.6)–(2.8) are quasi-linear and can be rewritten in the operator form:

$$\mathcal{L}\psi = (|\psi|^2 - 1)\psi = F(\psi), \tag{3.1}$$

$$\mathcal{M}A = f(\psi, A), \tag{3.2}$$

$$\mathcal{N}B = g(\psi, B). \tag{3.3}$$

After inverting the linear operators, we obtain the following iterative formulas:

$$\psi_{n+1} = \mathcal{L}^{-1}F(\psi_n), \quad A_{n+1} = \mathcal{M}^{-1}f(\psi_n, A_n), \quad B_{n+1} = \mathcal{N}^{-1}g(\psi_n, B_n). \tag{3.4}$$

For our initial triple (ψ_0, A_0, B_0), we take ψ_0 to be random, A_0 to be 0, and B_0 to be linear and increasing from $-h/2$ to $h/2$. The formulas (3.4) are used recursively to form the sequence of approximations (ψ_n, A_n, B_n) until they converge.

In our actual program, we employed some additional techniques to gain faster convergence. Rather than using (3.4), we solve (3.1)–(3.3) by using a quasi-Newton method. A damping factor is introduced to achieve stability. Also instead of taking (ψ_n, A_n, B_n) to be the starting point for the next iteration, a line search in the direction of this triple is performed to obtain a better starting point with lower energy.

Equation (3.3) is a cyclic tridiagonal linear system and can easily be solved by the linear algebra package Matlab. Although (3.2) comprises more equations than (3.3), once the right-hand side is known (computed from the results from the previous iterative step), the equations for each column of grid bonds are decoupled from those of another column. Thus, there are N_x systems of cyclic tridiagonal linear systems, each of which can be solved in exactly the same way as for (3.3).

The more interesting and difficult job is to invert $\mathcal{L}$, a five-point stencil that is a linear combination of the values of ψ at each grid point and its four neighbors: left, right, down, and up. The general form of a five-point stencil is

$$S[z] = Cz + Lz^{\leftarrow} + Rz^{\rightarrow} + Dz^{\downarrow} + Uz^{\uparrow}, \tag{3.5}$$

where C, L, D, and U are coefficients that can vary from one grid point to another. A nonhomogeneous stencil equation is

$$S[z] = b, \tag{3.6}$$

where b is given at each grid point.

The classical method for solving (3.6) is to treat it as a sparse linear system of order equal to the total number of grid points. Efficient use of existing sparse matrix solvers can then be used. The technique, however, is not simple for many who are not familiar with the subject area. In [10], we proposed an intuitively more straightforward method, the sweeping algorithm, for inverting certain regular five-point stencils including, $\mathcal{L}$. The method is related to the shooting method for solving boundary value problems of ordinary differential equations. Regular forms of the method are not hard to implement, and Matlab programs have been successfully written and reported in [11]. These Matlab programs are adequate for the purpose of our experiments on solving the GL equations.

One of the basic ideas of the sweeping algorithm is that once the values of the variable z is known on two columns (or rows) of grid points, its values on an adjacent column (row) can be computed from (3.6), provided that none of the stencil coefficients vanishes. For instance, take the column to the right of the two known columns. Each grid point is a right-hand neighbor of some grid point on the second column. Hence, from (3.6),

$$z^{\rightarrow} = \frac{b - (Cz + Lz^{\leftarrow} + Dz^{\downarrow} + Uz^{\uparrow})}{R}. \tag{3.7}$$

Continuing in this way, we can compute the values of z on every column, and can solved (3.6). The difficulty, of course, is to figure out the values on the two initial columns.

Our idea is to start with a wild guess, find out how much error we have made, and then fine-tune the initial guess to obtain the correct answer. Since one wild guess is just as good (or as bad) as another, we pick the simplest one, namely, that $z = 0$ on the two initial columns. We then compute z on the remaining columns. After we have finished with the computation, we will have used $N_x - 2$ columns of the stencil equations. There are two more columns of equations that have not been used; they are not likely to be satisfied, however, since our initial guess is rather haphazard. The discrepancy (from satisfying these remaining equations) is a vector E of order $2N_y$ and serves as a measure of how much we have missed the target and also as our guide for making corrections in the steps to come.

We next sweep the homogeneous stencil for a unit variation in each of the grid points on the two initial columns. The ensuing discrepancy in the last two columns of equations tells us what difference will be introduced by tuning each of the initial grid points. All this information is thee gathered to form a rectifying matrix $\mathcal{R}$. The knowledge of E and $\mathcal{R}$ will inform us how to choose the values of z on the initial columns correctly.

This simple procedure works remarkably well for grids of moderate sizes. For large grid sizes, some sophisticated modifications are needed to avoid instability (the rapid growth of z after repeated application of (3.7)). Two methods, multistage and partial sweepings, together with their implementations are described in detail in [10] and [11].

Besides its simplicity, the sweeping algorithm has other advantages. The complete information for constructing the inverse of the five-point stencil is stored in the relatively small inverse of the $2N_y \times 2N_y$ matrix $\mathcal{R}$. There is also great potential for exploiting parallelism in the algorithm, but further work needs to be done along this line.

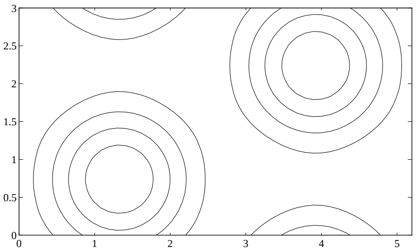
To conclude this section, we make some remarks concerning local minimizers of the GL equations. Although many people are apt to claim that, in their computation, they never see a critical point that is not the global minimizer, our experience seems to indicate the opposite. As the results in the next section show, the vortices appear at different heights in the core region. However, if one starts with a symmetric initial guess, then, at least in theory, the nonannealing methods should lead to a symmetric solution with the two vortices located at the same height. One can argue that such a solution is normally "unstable," so that any small perturbation (such as those arising from rounding errors), is enough to shift the unstable solution to a stable one. Yet some numerical methods for solving nonlinear equations, such as Newton's method, do not distinguish between stable and unstable solutions. Indeed, in our experience, this is what will happen if we use a quasi-Newton's method on (3.1) too early in the process. Our use of a random initial ψ_0 is an attempt to maximize the chance of starting near the correct solution. Also, our experiments indicate that, for regions containing many vortices, there can be numerous locally stable configurations distinct from the global minimizer.

4. Numerical Experiments

The main programs are written in Matlab to take advantage of its interactive features. In order to achieve greater speed, those commands connected with the sweeping algorithm have been recoded in Fortran by A. J. Lindeman and the compiled object file have been dynamically linked to Matlab through the “mex” facility. Lindeman, who also helped with most of the experiments mentioned in this section, is a participant in the summer 1992 Student Research Participation Program at Argonne National Laboratory, and is currently a physics graduate student at Purdue University

A typical run of an experiment is as follows. One sets, either interactively or by using a Matlab script file, the various parameters of the problem: the size of the domain L_x and L_y, the number of vortices n, the superconductor constant κ, the number of grid points N_x and N_y, the tolerance used to terminate the iteration, etc. The user needs to issue only two commands to get the experiment going. The first is a subroutine `setup` to set up the grid, to pick the initial state as a starting point for iteration, and to compute the various quantities needed to impose the quasi-periodic boundary conditions. The command `setup` calls an other subroutine `rr` automatically to compute the coefficients of the five-point stencil equation (3.4) and the rectifying matrices for the sweeping algorithm. This command is needed only at the beginning of the experiment. The second command, `rn`, performs the iteration process. It calls two subroutines, `onep` and `oneAB`, automatically and repeatedly to accomplish each iteration step until the desired accuracy is reached. After each step, various information, such as the maximum value of ψ_n, and the free energy are printed on the screen. A contour plot of $|\psi_n|$ is also displayed, for the user to monitor the progress.

In our experiments, we obtained plots of $|\psi|^2$ for various domain sizes. Two of these are shown below. Both are computed with $\kappa = 5$. The contour heights in the graphs are (0.05,0.1,0.15,0.20) and (0.1,0.3,0.5,0.7,0.9), respectively.



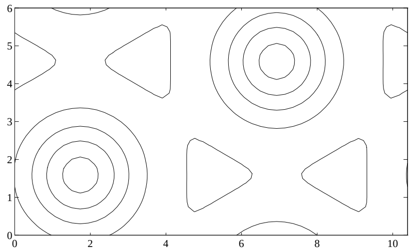


Figure 3. Two sample contour plots of $|\psi|^2$

We have investigated whether a parallelogram core region with angle 60°, corresponding to a strict hexagonal lattice, gives the best vortex configuration, with the lowest energy. To this end, we used core regions of the same area ($= 3 \times 3\sqrt{3}$) but different aspect ratios, and computed the least energy solutions for each. At first we discovered, to our surprise, that 60° is not the optimal angle. A plot of the energy (minus 16.2295 against the aspect ratio of the core region (width/height) is shown below. We used a grid size of 24×24 and $\kappa = 5$.

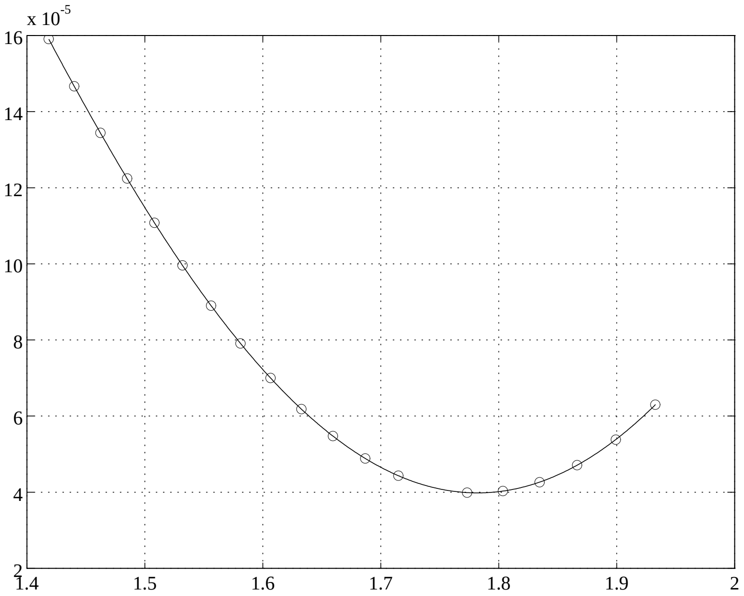


Figure 4. Energy against aspect ratio of core region

More experimentation revealed that this was due to discretization error. Increasing the grid size shifts the optimal angle towards 60°, and indeed an extrapolation with $N_x, N_y \to \infty$ shows the optimal angle to be 60°, within approximation error.

We have experimented with the phenomenon of pinning. A pinning site is caused by some defect in the superconducting material. It has a high resistivity to currents and so tends to pin down the center of a vortex. We modeled a pinning center by adding a penalty term $\gamma|\psi(\mathbf{x})|^2$ to the energy functional, where γ is a large positive constant and $\mathbf{x}$ is the location of the defect. This extra term alters the stencil coefficient C at the grid point corresponding to the pinning site. Our method successfully simulated the phenomena by giving a solution that nearly vanishes at the site.

Finally, we were also able to detect numerically the upper critical field. An increase in the external magnetic field corresponds to a more tightly packed vortex lattice, or a smaller core region. We solved the GL equations over core regions of different sizes and discovered that the maximum of ψ falls as the area decreases. Below the bifurcation point, $L_y = 2.68929$, which corresponds to an upper critical field of $H_{c2} = 1.00316868$, the trivial solution, $\psi \equiv 0$, takes over as the global minimizer. A plot of $\max|\psi|$ against the external field H is given below.

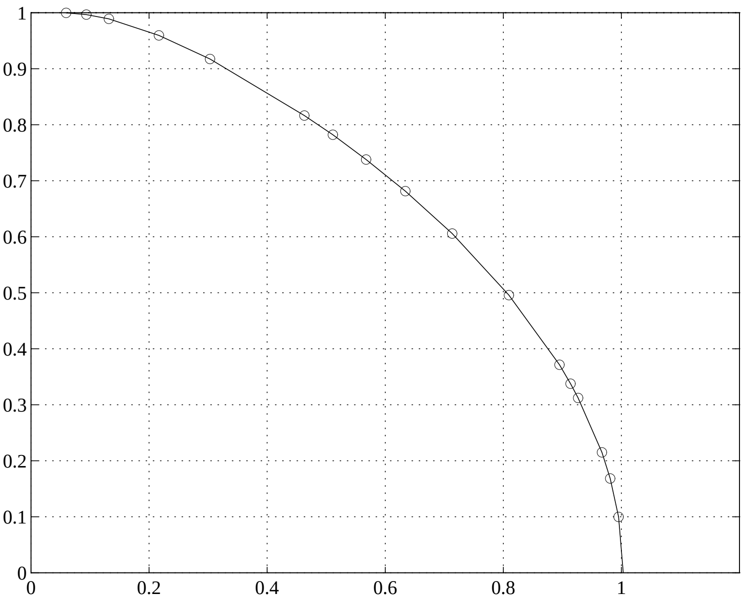


Figure 5. $\max|\psi|$ against external field

References

[1] Abrikosov, A., On the magnetic properties of superconductors of the second type, Zh. Eksperim. i Teor. Fiz., 32 (1957), 1442–1452. [English translation: Soviet Phys. – JETP, 5 (1957), 1174–1182.]

[2] Adler, S. L., and Piran, T., Relaxation methods for gauge field equilibrium equations, Reviews of Modern Phys., 56 (1984), 1–40.

[3] Chapman, S. J., Howison, S. D., and Ockendon, J. R., Macroscopic models for superconductivity, SIAM Review, to appear.

[4] Doria, M., Gubernatis, J., and Ranier, D., Solving the Ginzburg-Landau equations by simulated annealing, Phys. Rev. B, 41 (1990), 6335–6340.

[5] Du, Q., Gunzburger, M., and Peterson, J., Analysis and approximation of a periodic Ginzburg-Landau model for type-II superconductors, SIAM Review, 34 (1992), 54–81.

[6] Du, Q., Gunzburger, M., and Peterson, J., Modeling and analysis of a periodic Ginzburg-Landau model for type-II superconductors, preprint, 1992.

[7] Garner, J., Spanbauer, M., Benedek, R., Strandburg, K., Wright, S., and Plassmann, P., Critical fields of Josephson-coupled superconducting multilayers, Physical Review B 45 (1992), 7973–7983.

[8] Ginzburg, V., and Landau, L., On the theory of superconductivity, Zh. Eksperim. i Teor. Fiz., 20 (1950), 1064–1082. [English translation: Men of Physics: L. D. Landau, ed. D. ter Harr, Pergamon, Oxford (1965), 138–167.]

[9] Kaper, H. G., and Kwong, Man Kam, Vortex configurations in high-T_c superconducting films, preprint.

[10] Kwong, Man Kam, Sweeping algorithms for inverting the discrete Ginzburg-Landau operator, preprint MCS-P307-0492, MCS Division, Argonne National Laboratory (1991), to appear in Applied Mathematics and Computation.

[11] Kwong, Man Kam, Sweeping algorithms for five-point stencils and banded matrices, Technical Memorandum ANL/MCS-TM-165, MCS Division, Argonne National Laboratory (1991).

[12] Kato, R., Enomoto, Y., and Maekawa, S., Computer simulations of dynamics of flux lines in Type-II superconductors, preprint, 1991.

[13] Wang, Z. D., and Hu, C. R., A numerical relaxation approach for solving the general Ginzburg-Landau equations for type-II superconductors, preprint, 1991.