

# NEW ADVANCES IN THE MODELING OF HIGH-TEMPERATURE SUPERCONDUCTORS

Lori Freitag  
MCS Division  
Argonne National Lab  
Argonne, IL 60439

Mark Jones  
Computer Science Department  
University of Tennessee  
Knoxville, TN 37996

Paul Plassmann  
MCS Division  
Argonne National Lab  
Argonne, IL 60439

## ABSTRACT

In this paper, we present a new discrete formulation that maintains discrete invariance and is suitable for use on nonorthogonal meshes. This formulation, unlike its predecessors, allows us to easily use adaptive mesh refinement to concentrate grid points where error contributions are large (near vortex centers). In this way we reduce the total number of grid points required to accurately capture vortex configurations and allow the possibility of solving problems previously considered intractable. To solve these large problems, we require the memory capabilities and computational power of state-of-the-art parallel computers. To use these machines, we have developed scalable libraries for adaptive mesh refinement and partitioning on two-dimensional triangular grids. This general-purpose software uses bisection of the longest side to refine triangles in which error contributions are large. These adaptive meshes are both unstructured and dynamic, and we present a new geometric partitioning algorithm that strives to minimize communication cost by ensuring good partition aspect ratios. We present computational results showing the efficiency of these adaptive techniques for a superconductivity application.

## INTRODUCTION

The recent discovery of high-temperature superconductors has led to extensive research designed to obtain an understanding of the magnetic properties of these materials. Practical applications of superconductors are limited primarily by the relatively small value of the critical current density which causes the material to revert back to normal conductivity. This critical current density is in turn linked to the mixed, or vortex, state in which magnetic flux quanta (vortices) penetrate the sample so that normal and superconducting regions coexist. It is hoped that an understanding of the vortex state will lead to techniques designed to increase the critical current density and hence allow these materials to be widely used in scientific and industrial applications.

To study the internal structures and lattice configurations of vortices in isotropic materials, we use the nondimensionalized Ginzburg-Landau model. The total free en-

ergy over the volume,  $\Omega$ , is given by

$$F = F_{\text{cond}}(\psi) + F_{\text{kin}}(\psi, \mathbf{A}) + F_{\text{fld}}(\mathbf{A}). \quad (1)$$

These three terms are generally known as the condensation, kinetic and field energy terms and are given by the formulae

$$F_{\text{cond}} = \int_{\Omega} -|\psi|^2 + \frac{1}{2}|\psi|^4 d\Omega, \quad (2)$$

$$F_{\text{kin}} = \int_{\Omega} |(\nabla + i\mathbf{A})\psi|^2 d\Omega, \quad (3)$$

$$F_{\text{fld}} = \int_{\Omega} \kappa^2 |\nabla \times \mathbf{A}|^2 d\Omega. \quad (4)$$

The variable  $\psi$  is the complex-valued order parameter and  $\mathbf{A}$  is the vector potential. The physical quantities of interest are  $|\psi|^2$ , the local density of superconducting electron pairs, and  $\mathbf{B} = \nabla \times \mathbf{A}$ , the magnetic field induced by the motion of the electron pairs through the sample. The parameter  $\kappa$  gives the ratio of the characteristic length over which  $|\psi|^2$  varies (the correlation length  $\xi$ ), to the characteristic length over which  $\mathbf{B}$  varies (the penetration depth  $\lambda$ ). An important property of the free energy functional,  $F$ , is that its value is unchanged by a gauge transformation. That is, given  $(\psi, \mathbf{A})$  and any scalar function,  $\chi$ , we find that the pair  $(\psi', \mathbf{A}')$  given by

$$\psi' = \psi e^{i\chi}, \quad \mathbf{A}' = \mathbf{A} - \nabla\chi \quad (5)$$

leaves the free energy invariant.

In previous work, a finite-difference discretization of the free energy functional on a regular, orthogonal mesh was used (Garner *et al.* 1992). With that approach, a high density of grid points is required to accurately model the vortex core. Such an approach is inefficient because the constant density of grid points means that many superfluous grid points are used away from the vortex centers. Instead, we would like to concentrate grid points near the vortex core (where the solution changes rapidly) and use relatively few grid points far from the vortex cores. In this way, we reduce the total number of grid points required to accurately model the vortex state in high-temperature superconductors.

Toward this purpose, we have developed the algorithms and software necessary to adaptively refine triangular meshes on parallel computers (Freitag, Jones, and Plassmann 1994). Triangular elements to be refined are determined by a user-defined indicator function (in this case proximity to the vortex core). Currently, the software uses triangle bisection to refine elements and employs a modification of Rivara's technique (Rivara 1984) to remove non-conforming points in the refined mesh. Though this approach is efficient in its use of mesh points, the resulting discretized problem is both unstructured and dynamic. In order to achieve good load balancing and performance on parallel computers, this dynamic mesh must be partitioned (an assignment of unknowns and elements to processors) after each modification. We have developed a new geometric partitioning algorithm that strives to minimize both latency and transmission communication costs on distributed memory architectures.

A more subtle problem that arises from a nonorthogonal discretization using triangular elements is that the standard finite-difference and finite-element discretizations fail to maintain a discrete version of gauge invariance. Therefore, we have developed a new discretization scheme for the Ginzburg-Landau free-energy functional that maintains gauge invariance and is suitable for use on nonorthogonal meshes.

The remainder of the paper is organized as follows. First, we describe the new discrete formulation and present an asymptotic analysis showing the validity of this new formulation. Then, we discuss the parallel algorithms used for adaptive mesh refinement and partitioning and appropriate refinement and interpolation rules for the superconductivity problem. Next, we present computational results obtained on the Intel DELTA located at Caltech. In the final section, we summarize the results of this paper and offer some remarks about future research.

## DISCRETE FORMULATION FOR NONORTHOGONAL MESHES

To accurately model the vortex core singularities in high temperature superconductors, we would like to adaptively refine the mesh to concentrate grid points where error contributions are large. Adaptive refinement on orthogonal meshes requires constraints be placed at nonconforming nodes. These constraints unnecessarily increase the work required to solve the system and complicate implementation. Thus, we choose to use triangular meshes. The question now arises as to whether standard discretization techniques maintain the important property of discrete gauge invariance on nonorthogonal meshes. To answer this question, let us first define the requirements for a gauge invariant discretization.

**Definition 1** *Let  $\bar{\psi}$  and  $\bar{\mathbf{A}}$  be the discrete representations of  $\psi$  and  $\mathbf{A}$ . In addition, let  $\bar{F}$  be a discrete formulation of the free-energy functional given in Equation (1), and let  $\bar{\chi}$*

*be any scalars defined on the same grid points as  $\bar{\psi}$ . Consider the transformation  $\bar{\psi}' = \bar{\psi}e^{i\bar{\chi}}$ . We define discrete gauge invariance to be the property of the discretization scheme that allows for a corresponding transformation of  $\bar{\mathbf{A}}$  to some  $\bar{\mathbf{A}}'$  such that  $\bar{F}(\bar{\psi}', \bar{\mathbf{A}}') = \bar{F}(\bar{\psi}, \bar{\mathbf{A}})$ .*

Note that we do not specify the transformation of  $\bar{\mathbf{A}}$  in this definition since that depends on a discrete representation of the gradient of  $\bar{\chi}$ . We require only that such a transformation exist.

It is easily shown that standard finite-difference and finite-element discretization techniques do not maintain discrete gauge invariance as defined above. Consider the problem in two dimensions: let  $\psi = a + ib$ ,  $\mathbf{A} = [A_x, A_y]$ , and the area of the domain be  $A_\Omega$ . Standard finite-difference discretization minimizes  $F$  over the field variables  $a, b, A_x$ , and  $A_y$  where all of the unknowns are associated with discrete grid points. This approach can satisfy the discrete gauge invariance definition only when the mesh is orthogonal. When the mesh is not orthogonal (here we consider triangular meshes), finite-difference discretizations are inappropriate. More appropriate in this case are finite-element techniques, and we consider polynomial interpolation functions on triangular meshes. Substituting the discrete variables into the kinetic term shows immediately that this discretization is not gauge invariant for functions of  $\chi$  that are not polynomial.

Rather than assign all of the field variables to grid points, a more natural scheme for discretizing this functional is to assign the values for  $\bar{\psi}$  at mesh vertices and the values for  $\bar{\mathbf{A}}$  on edges between vertices. In this way, we can adjust the discrete gauge transformation at the vertices in  $\bar{\psi}$  along each edge or link in  $\bar{\mathbf{A}}$ . Specifically, the relative phase shift  $\Delta\bar{\chi}$  between two neighboring vertices can be subtracted from the vector potential value on the edge (a natural discretization of subtracting the gradient of  $\chi$  from  $\mathbf{A}$  in the continuous case). This correspondence of discrete variables works however, only if  $\bar{F}$  is invariant under this transformation. We now present a new discrete formulation that ensures this property.

The unknowns of the new formulation are  $a, b$ , and the phase  $\theta$ , where

$$\theta = \int \mathbf{A} \cdot \mathbf{T} ds$$

is associated with the links between the grid points. Let the vertices of a typical triangle in the mesh be  $v_1, v_2$ , and  $v_3$  and the links opposite each vertex be  $\mathbf{l}_1, \mathbf{l}_2$ , and  $\mathbf{l}_3$ , as illustrated in Figure 1. Let the area of the triangle be  $A_\Delta$  and the area of the domain be  $A_\Omega$ . We give the discretization of each of the three terms in Equation (1) that ensures that a discrete form of gauge invariance is preserved.

The condensation energy density in Equation (2) is evaluated at the vertices of the triangle and averaged to obtain

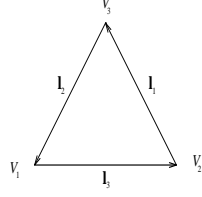


Figure 1: A typical triangular element in the mesh

a value for the entire element:

$$\bar{F}_{\text{cond}} = \frac{A_{\Delta}}{3A_{\Omega}} \sum_{i=1}^3 \left[ -(a(v_i)^2 + b(v_i)^2) + \frac{1}{2}(a(v_i)^2 + b(v_i)^2)^2 \right]. \quad (6)$$

For the field energy density, we use Green's theorem to obtain

$$F_{\text{fld}} = \int_{\Delta} \kappa^2 |\nabla \times \mathbf{A}|^2 d\Delta = \oint_{\partial\Delta} \kappa^2 |\mathbf{A} \cdot \mathbf{T}|^2 ds,$$

where  $\mathbf{T}$  is the unit tangent vector such that  $\mathbf{n} \times \mathbf{T}$  always points into the domain. Then the discrete representation using the field variable  $\theta$  is

$$\bar{F}_{\text{fld}} = \frac{1}{A_{\Delta}A_{\Omega}} \left( \kappa \sum_{i=1}^3 \theta(l_i) \right)^2. \quad (7)$$

The kinetic energy density 3 is calculated at each vertex in the triangle, and the average of these values is used to represent the energy of the element. To maintain gauge invariance, we use a link variable formulation similar to that used in the standard finite-difference discretization. First, we rotate the element so that all values of the order parameter are in the same gauge basis. That is,

$$\hat{\psi}(v_2) = \psi(v_2)e^{-i\theta(l_3)}, \quad \hat{\psi}(v_3) = \psi(v_3)e^{-i(\theta(l_3)+\theta(l_1))},$$

$$\hat{\psi}(v_1) = \psi(v_1)e^{-i(\theta(l_3)+\theta(l_1)+\theta(l_2))}.$$

In this case the gauge invariant differences along each link are given by

$$K_1 = \hat{\psi}(v_3) - \hat{\psi}(v_2), \quad K_2 = \hat{\psi}(v_1) - \hat{\psi}(v_3),$$

$$K_3 = \hat{\psi}(v_2) - \hat{\psi}(v_1).$$

The contribution to the kinetic term at the vertex  $v_1$  is then

$$K(v_1) = K_2^2 l_3^2 + K_3^2 l_2^2 + 2l_2 \cdot l_3 K_2 K_3^*.$$

The cross term  $2l_2 \cdot l_3 K_2 K_3^*$  is not required for orthogonal meshes but is incorporated here to adjust for the nonorthogonality of the triangle sides. Similar terms for the other two

vertices are summed to give the kinetic energy for the element,

$$\bar{F}_{\text{kin}} = \frac{1}{12 A_{\Delta}A_{\Omega}} \sum_{i=1}^3 K(v_i). \quad (8)$$

We assume  $\Omega$  is far from the boundary of the sample, and we use pseudo-periodic boundary conditions to maintain a fixed magnetic flux of  $2\pi\nu$ , where  $\nu$  is the number of vortices. For a two-dimensional domain of size  $L_x \times L_y$  these boundary conditions are given by

$$\psi(L_x, y) = \psi(0, y)e^{\frac{2\pi\nu y}{L_y}}, \quad \psi(x, L_y) = \psi(x, 0)$$

$$\theta(L_x, y) = \theta(x, 0) - \frac{2\pi\nu}{L_y}h, \quad \theta(x, L_y) = \theta(0, y).$$

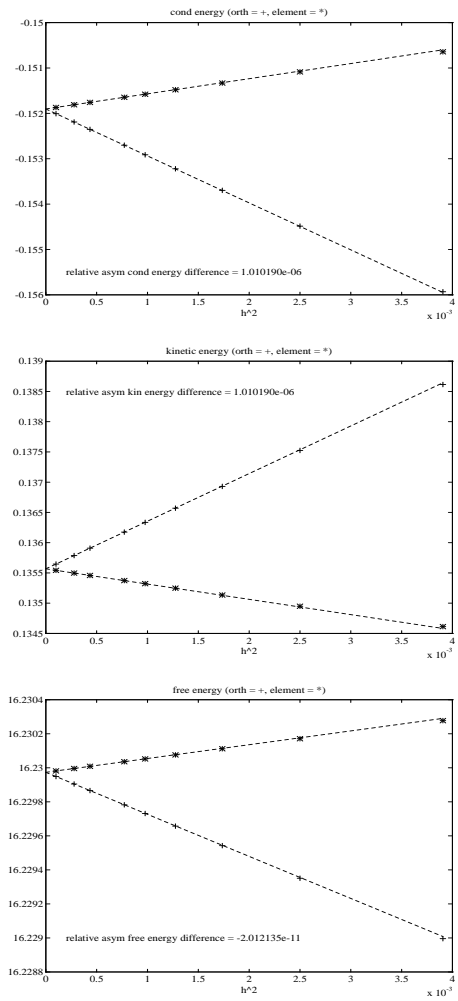


Figure 2: Asymptotic results for the condensation, kinetic terms and final free-energy value

We use asymptotic analysis on a sample containing two vortices with  $\kappa = 5$  to show the validity of the new discrete

model. The results obtained on uniform, orthogonal meshes using finite differences are compared with those obtained on uniform triangular meshes using the new discrete formulation. As the element area  $\mathcal{O}(h^2)$  approaches zero, we see in Figure 2 that condensation, kinetic, and total free-energy terms converge linearly to the same result. The methods approach the solution from different directions: finite differences from below for the condensation and total free-energy terms, the new formulation from above and vice versa for the kinetic term. The kinetic energy is highest around the vortex core, which, for the new element technique, may be located anywhere in the element, not just at grid points, as is the case in finite differences. In this case the gradient term in  $F_{kin}$  is not well approximated, and the kinetic energy term is always underestimated.

### PARALLEL MESH REFINEMENT AND PARTITIONING

The discretization presented in the preceding section is suitable for use on nonorthogonal meshes. In particular, we use adaptive refinement to concentrate grid points where errors are large (in this case, close to the vortex core). As an adaptive mesh tracks vortex development and movement, maintaining mesh quality is a primary concern. As the minimum angle of the mesh decreases, the condition number of the linear system grows, making solution more difficult. As the maximum final angle of the mesh approaches  $\pi$ , the interpolation error in the approximate solution increases. Two basic approaches to refinement guarantee mesh quality: regular refinement and bisection (see Mitchell 1989 for a comprehensive review). Regular refinement divides a triangle marked for refinement into four similar triangles by connecting the midpoints of the three sides. Bisection of the longest side, however, divides this triangle by connecting the midside of the longest side to the opposite vertex. The two techniques are illustrated in Figure 3, where the shaded triangles are those currently marked for refinement. Mitchell compared these techniques and found that the number of grid points needed to obtain a given error bound on the final solution is essentially the same for several test problems. Since this is the case, we choose to implement the simpler method, bisection, in our parallel adaptive mesh refinement software.

Adaptive mesh refinement on distributed memory architectures requires careful implementation to maintain data coherence across the processors. For instance, as vertices are dynamically added and deleted, all processors must agree to a unique global list of vertices, to a list of three unique vertices associated with any triangle, and to triangle ownership if vertices of a triangle are distributed across processors. That is, two processors cannot simultaneously refine the same triangle, nor may neighboring triangles be refined simultaneously, since extraneous copies of vertices may be created across processors. We avoid these problems by refining independent sets of triangles in parallel. These independent sets are chosen in the dual graph of the triangular mesh (the dual graph is the graph whose ver-

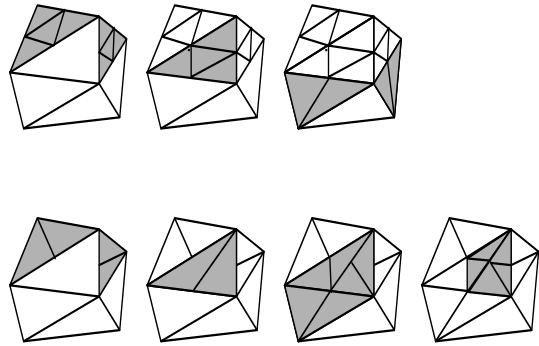


Figure 3: Regular refinement (top row) and bisection of the longest side (bottom row)

tices correspond to the triangles and for which two vertices are adjacent if their triangles share a common edge in the mesh). Triangles that are not adjacent in the dual graph are independent of each other and may be updated in parallel (see Freitag, Jones, and Plassman 1994) for a complete description of the parallel algorithm). Our algorithm terminates in a finite number of steps and has a parallel execution time of  $\mathcal{O}(\log M_k) \times L_P$  where  $M_k$  is the number of triangles marked for refinement and  $L_P$  is the number of levels of propagation.

As grid points are adaptively added and deleted in the mesh, we must determine good partitionings for distributed memory architectures. We define a good partition to be one in which the grid points are evenly distributed to the processors in such a way that interprocessor communication costs are minimized. We may minimize the latency and transmission communication costs by minimizing the number of partition neighbors and number links crossing the partition boundary, respectively. For uniform meshes a good partitioning of grid points may be determined *a priori* by the geometric domain. For unstructured adaptive meshes, however, the partitioning cannot be predetermined because it changes with each new refinement of the mesh.

Several interesting techniques are used to determine the partitioning of an unstructured mesh. Spectral methods (see Pothen, Simon, and Liou 1990, for example) have the advantage of global access to information about the graph to find good separators at the cost of eigenvalue/eigenvector computation. Although the eigenvectors generally do not need to be found to much accuracy, spectral methods fail to utilize the geometric information known about the vertices of the mesh, which may significantly decrease execution time. Geometric information is used in bisection partitioning algorithms such as the orthogonal recursive bisection (ORB) algorithm of (Berger and Bokhari 1987). This algorithm makes an initial cut to divide the grid points in half. Orthogonal cuts are then made recursively in the new subdomains until the grid points are evenly distributed among the processors. Although this algorithm obtains

good load balancing, it ignores the communication minimization problem. Long, thin partitions may be created that have a high ratio of links crossing the partition boundaries to total number of links in the partition. This leads to high ratios of communication to computation.

To address this problem, we have developed a modification of ORB which we call the unbalanced recursive bisection (URB) algorithm. Instead of dividing the unknowns in half, we choose the cut that minimizes partition aspect ratio and divides the unknowns into  $\frac{nk}{P}$  and  $\frac{n(P-k)}{P}$  size groups where  $n$  is the total number of unknowns,  $P$  is the number of processors, and  $k = 1, 2, \dots, P - 1$ . This algorithm leads to an even distribution of grid points with more balanced partition aspect ratios. This minimizes the communication costs in two ways. First, partitions with good aspect ratios (close to one) tend to have fewer partition neighbors and hence fewer messages to send. Second, the percentage of mesh links crossing the partition boundary to the total number of links in the nearly square partitions is small compared with the long, thin partitions generated by the ORB algorithm. Thus, the ratios of computation to communication are increased compared with the ORB algorithm, while execution time is significantly less than for spectral techniques.

## COMPUTATIONAL RESULTS

The invariance of the free-energy functional means that a minimizer of  $F$  is unique only up to a gauge transformation. This degeneracy significantly complicates the computation of such a minimizer. We have found that an effective approach to computing a minimizer  $\mathbf{u}^* = (\psi^*, \mathbf{A}^*)$  of the discretized free-energy functional is a damped Newton's method. Each step of Newton's method requires computation of the gradient vector,  $\nabla F$ , and Hessian matrix,  $\nabla^2 F$  for which we use the automated differentiation package ADIFOR (Bischof *et al.* 1991). Because of the gauge symmetries of the problem, the Hessian is highly singular at the solution, and we include a damping term to improve the convergence of the method as described by (Garner *et al.* 1992). The computational kernel of this technique is the solution of the damped Newton system—a large, sparse linear system of equations. We do not explicitly invert this system, but use an iterative solver to obtain an approximate (inexact) solution. The BlockSolve package developed at Argonne National Laboratory by Mark Jones and Paul Plassmann (Jones and Plassmann 1992) is used for this purpose.

To adaptively refine the mesh around vortex core singularities, we use the following refinement rule: A triangle,  $\Delta_k$ , is refined if

$$\min (|\psi(v_i)|^2 \cdot A_\Delta) < \epsilon_T,$$

where the  $\epsilon_T$  is a user-defined tolerance. Suppose that the triangle we are refining,  $\Delta_k$ , has its longest edge opposite  $v_1$  and has link directions as indicated in Figure 1. The new values of  $a$  and  $b$  are obtained by linear interpolation

at the new grid point which is halfway between  $v_2$  and  $v_3$ . The new phase  $\theta_{new}$  is chosen such that the magnetic flux density is equal in the two new triangles to the original flux density.

To demonstrate the efficiency and scalability of the refinement and partitioning algorithms for the superconductivity problem, we allowed problem sizes to increase proportional to the number of processors used. Because our current implementation of the adaptive refinement algorithms allows data to be associated with vertices only, we have used a preliminary formulation of the finite element given previously. However, we feel that the computational results presented here would be quantitatively the same for the revised element. The results of four typical runs are shown in Table 1 where  $P$  gives the number of processors and  $E$  indicates the number of triangular elements in the final solution mesh. The number of vortices in each sample are 32, 48, 64, and 72 for 16, 32, 64, and 128 processors, respectively. We indicate the amount of time required for refinement and partitioning as a percentage of total solution time. We see that these operations require less than one percent of the execution time in all cases.

$P$	$E$	Percent Refine Time	Percent Partition Time	Percent Setup Time	Percent Solution Time
16	30484	.229	.193	30.0	69.5
32	48416	.091	.117	13.5	86.3
64	111660	.087	.167	10.8	88.8
128	196494	.181	.452	13.3	86.0

Table 1: Timing results for the superconductivity problem on 16–128 processors of the Intel DELTA

Statistics on the partitions generated by the new geometric partitioning algorithm, URB, are given in Table 2. The average aspect ratio for the partitions is less than two in all cases, and the maximum aspect ratio is less than 3.6. These result in a partition quotient graph whose average degree is between five and six, which corresponds to an average of five to six messages sent per processor to transfer nearest neighbor information. Finally, to estimate the amount of data that must be transferred between processors, we consider the percentage of edges that cross partition boundaries to the total number of edges in the partition. This number is less than 15 percent in all cases.

The final triangular mesh of a 32 vortex problem is shown in Figure 4. Vortex cores are indicated by the location of the heavily refined areas of the mesh. This problem was run on 64 processors of the Intel DELTA and partitions are indicated by the numbered boxes. We see that the partitions tend to split the vortex cores to evenly distribute grid points and are nearly square in most cases. We find that, as we refine the mesh around vortex cores, the fact that the kinetic term is approaching the asymptotic result from below causes the vortex to drift toward regions containing larger mesh elements. We are currently working to

$P$	Avg. Graph Degree	Max. Graph Degree	Avg. Aspect Ratio	Max. Aspect Ratio	Percent Cross Edges
16	5.31	7.00	1.47	2.88	6.72
32	5.40	8.00	1.89	3.55	8.32
64	5.64	8.00	1.34	2.49	10.0
128	5.71	9.00	1.81	3.55	13.7

Table 2: Partition statistics for the superconductivity problem on 16–128 processors of the Intel DELTA

eliminate this problem and are temporarily using Gaussian well pinning sites to fix vortex position.

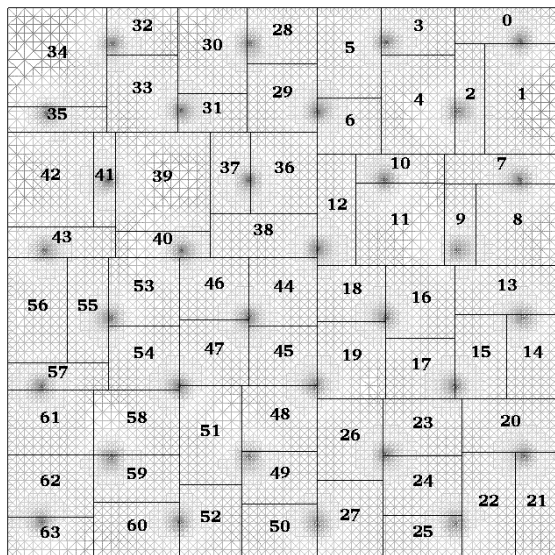


Figure 4: Results of a sample containing 32 vortices pinned to a square lattice configuration on 64 processors of the Intel DELTA

## CONCLUDING REMARKS

In this paper, we have presented an important advance in the numerical modeling of high-temperature superconductors: a new formulation that allows the use of adaptively refined meshes. Previous numerical models have been limited by the use of orthogonal meshes needed to maintain gauge invariance in the discrete formulation of the Ginzburg-Landau free-energy functional. We have developed and presented a new discrete formulation that allows the use of nonorthogonal meshes. This formulation was validated by using asymptotic analysis to compare to standard finite difference techniques on uniform triangular meshes for two-dimensional problems. More important, this new formulation is suitable for use on adaptively refined meshes. By concentrating grid points near the vortex core singularities, we obtain a more detailed look at vortex core struc-

ture, and we significantly reduce the total number of grid points required to obtain an accurate representation of the lattice configuration. This formulation will in turn allow us to study samples with very large numbers of vortices, as well as superconducting materials in the high- $\kappa$  regime. Some work remains to be done to eliminate the problem of vortex drift, and we are currently refining the discrete formulation to handle this situation.

## Acknowledgments

This work was supported by the Office of Scientific Computing, U.S. Department of Energy, under Contract W-31-109-Eng-38.

## References

- Berger, M. and S. Bokhari, 1987, A partitioning strategy for nonuniform problems on multiprocessors, *IEEE Transactions on Computers*, C-36(5).
- Bischof, C., A. Carle, G. Corliss, A. Griewank, and P. Hovland, 1992, ADIFOR: Generating derivative codes from Fortran programs, *Scientific Programming*, 1(1):11–29.
- Freitag, L., M. Jones, and P. Plassmann., 1994, Parallel algorithms for unstructured mesh computation, In *Proceedings of the Applied Linear Algebra Conference, "Snowbird, Utah"*. SIAM.
- Garner, J., M. Spanbauer, R. Benedek, K. Strandburg, S. Wright, and P. Plassmann, 1992, Critical fields of Josephson-coupled superconducting multilayers *Physical Review B*, 45:7973–7983.
- Jones, M. and P. Plassmann, 1992, BlockSolve v1.0: Scalable library software for the parallel solution of sparse linear systems, ANL Report ANL-92/46, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Ill..
- Mitchell, W., 1989, A comparison of adaptive refinement techniques for elliptic problems. *ACM Transactions of Mathematic Software*, 15(4):326–347.
- Pothen, A., H. Simon, and K-P. Liou, 1990, Partitioning sparse matrices with eigenvectors of graphs. *SIAM Journal on Matrix Analysis*, 11:430–452.
- Rivara., M., 1984, Mesh refinement processes based on the generalized bisection of simplices. *SIAM Journal on Numerical Analysis*, 21:604–613.