

# Visualising Vector Field Model Simulations

D.P. Playne and K.A. Hawick

Institute of Information and Mathematical Sciences

Massey University, North Shore 102-904, Auckland, New Zealand

Email: d.p.playne@massey.ac.nz and k.a.hawick@massey.ac.nz

Tel: +64 9 414 0800 Fax: +64 9 441 8181

February 2009

## Abstract

Many scientific simulations and models are based upon one or more coupled field equations. Fields are often modelled as a regular mesh or grid of individual field variables where each degrees of freedom or site variable is a scalar or vector quantity. Visualising such quantities interactively can be a great aid to debugging as well as understanding and interpreting the results of numerical simulation. We review a number of different approaches and software technologies for visualising scalar and vector fields as they time-evolve in 1, 2 and 3 dimensional simulations. We present some performance data that can be used to plan size and time scoping for simulations on present interactive visualisation computer technology. We discuss the particular merits of different approaches for dense 3-dimensional vector-field models such as the Ginzburg-Landau equation. We offer some speculations on how these techniques could be applied to simulations of other field equations.

**Keywords:** vector field; scalar field; visualisation; scientific simulation; Ginzburg-Landau field equation

## 1 Introduction

Visualising the results from the simulation or modelling of partial differential equations is a long standing problem and area of interest in several areas of computational science. Many interesting physical, chemical and biological systems can be modelled as field equations. A field equation model is simply an equation or set of coupled equations based on field variables that vary spatially and usually also in time.

For example, climate and weather systems are modelled by field variables representing properties of the atmosphere

such as air temperature, pressure, moisture content and wind direction. Ocean or fluid flow models have similar field variables describing water properties. The field variables are sometimes known as the degrees of freedom of the model and are usually arranged on some sort of spatial mesh or grid. Sometimes meshes are regularly spaced, but they need not necessarily be as irregular meshes are sometimes useful to place extra degrees of freedom close to where the field is changing rapidly. It is not always possible to generate simple regular geometric meshes. Weather and climate systems suffer from the need to wrap a mesh around the near spherical surface of the earth for example.

In general then we are interested in being able to examine one or more fields  $\Psi(x, y, z, t)$  as they change in time over some spatial set of coordinates  $\mathbf{r} = (x, y, z)$  in a three-dimensional coordinate system.

There are a number of successful approaches to this problem - often based upon some sort of divide and conquer solution. Visualising a single  $(x, y)$ ;  $(x, z)$ ; or  $(y, z)$  surface at a time is an obvious approach that lends itself well to a pixel based display technology. Animating time sequences into movies that can be started, stopped, speeded up, played forwards or backwards is another obvious approach to visualising in time.

A number of techniques can be applied to visualizing field equations when the field variable is a single scalar variable and we have reported on some work concerning scalar equations such as the Cahn-Hilliard system [1]. In this present paper we consider some particular issues that arise when the field variables are not just separate quantities but are components of a vector quantity or the real and imaginary parts of a complex field variable.

The Time-dependent Ginzburg-Landau (TDGL) equation [2] is an archetypal system that is based around a complex field variable. The TDGL is used to model a number of

physical systems and gaining an understanding of its complex field variable is crucial to gaining an intuitive interpretation of the dynamical and static properties of the model.

In this paper we describe the Time-Dependent Ginzburg-Landau equation in section 2. In section 3.2 we briefly review some of the available software technologies for enabling a suitable visualization and rendering. We present some visualisation results in sections 3.3 and 3.4 and offer some discussion for future work and some concluding remarks in section 4.

## 2 The Ginzburg-Landau Equation

Ginzburg-Landau theory provides a framework for modelling the thermodynamic or macroscopic properties of a superconductor without recourse to simulation of microscopic details such as individual atoms or spins. The formulation is based on the concept of a free energy functional for the superconductor that can be written in terms of a complex order parameter, given here as  $\psi$ . This order parameter give a measure of how deeply the system is into its superconducting phase. The free energy function has the typical form:

$$F = F_n + \alpha|\psi|^2 + \frac{\beta}{2}|\psi|^4 + \frac{1}{2m} |(-i\hbar\nabla - 2e\mathbf{A})\psi|^2 + \frac{|\mathbf{B}|^2}{2\mu_0} \quad (1)$$

where  $F_n$  is the free energy in the normal phase,  $\alpha$  and  $\beta$  are phenomenological parameters,  $m$  is an effective mass,  $\mathbf{A}$  is the electromagnetic vector potential, and  $\mathbf{B}$  ( $=\text{rot } \mathbf{A}$ ) is the magnetic induction. Minimizing the free energy with respect to fluctuations in the order parameter and the vector potential give rise to the Ginzburg-Landau equations:

$$\alpha\psi + \beta|\psi|^2\psi + \frac{1}{2m} (-i\hbar\nabla - 2e\mathbf{A})^2 \psi = 0 \quad (2)$$

$$\mathbf{j} = \frac{2e}{m} \text{Re} \{ \psi^* (-i\hbar\nabla - 2e\mathbf{A}) \psi \} \quad (3)$$

where  $j$  is the electrical current density and  $\text{Re}$  the real part. Equation 2 specifies the order parameter  $\psi$  in terms of the applied magnetic field. This equation has a similar structure to the time-independent Schroeder equation. Equation 3 then determines the superconducting current.

These equations suggest that there are two characteristic (physical) lengths that occur in a superconductor. These are usually known as the coherence length  $\xi \sqrt{\frac{\hbar^2}{2m|\alpha|}}$ .

which characterises the size of thermodynamic fluctuations in the superconducting phase.

The penetration depth  $\lambda$  is given by:  $\lambda = \sqrt{\frac{m}{4\mu_0 e^2 \psi_0^2}}$

where  $\psi_0$  is the equilibrium value of the order parameter in the absence of an electromagnetic field. The penetration depth then characterises the depth to which an external magnetic field can penetrate the superconductor.

The ratio  $\kappa = \lambda/\xi$  is usually known as the Ginzburg-Landau Parameter. Type I superconductors are those for which  $\kappa < 1/\sqrt{2}$ , and Type II superconductors are those for which  $\kappa > 1/\sqrt{2}$ .

In Type I superconductors there is a first order phase transition from the normal state to the superconducting state, and for Type II the transition is second order. This is consistent with Ginzburg-Landau theory.

In 1957, Abrikosov [3] showed that for the case of a Type II superconductor, a (high) magnetic field will penetrate the system in quantized tubes of flux, often in a hexagonal geometric pattern.

Ginzburg-Landau theory also arises as the scaling limit for the XY model and also displays important similarities with the Higgs mechanisms in particle systems. [4].

If we factor the constants and the physical units into scaled variables, we can therefore obtain a simple form of the Ginzburg Landau partial differential equation in term of a dimensionless field  $u$ :

$$i \frac{\partial u}{\partial t} + p \frac{\partial^2 u}{\partial x^2} + q|u|^2 u = i\gamma u \quad (4)$$

where the constants  $p, q \in \mathbb{C}$  and  $\gamma \in \mathbb{R}$ .

## 3 Modelling and Simulation

In this section we describe firstly the numerical methods we employed, and secondly the visualisation techniques we explored for analysing a complex field. We elaborate on the additional techniques used for visualising both 2-d and 3-d fields and describe techniques particularly suited to complex fields including either phase, magnitude or both.

### 3.1 Numerical Methods

We use the usual Laplacian centred spatial stencil (but with complex numbers) for  $\frac{\partial^2}{\partial x^2}$ , so that we have a  $[-+2+]$  stencil and hence:

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{1}{\delta^2} (u_{x-\delta} - 2u + u_{x+\delta}) \quad (5)$$

where we can use  $\delta \equiv 1$  subject to numerically scale the factors of  $p, q, \gamma$ . The memory access data stencil for this is shown in figure 1 for 1-d, 2-d and 3-d field cases.

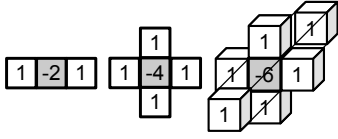


Figure 1: Memory access stencil for 1-2, 2-2 and 3-d cases of equation 5.

The numerical simulation of the Ginzburg-Landau equation 5 is somewhat more complicated than for a simple scalar field. As each cell is a complex number, the calculations for the numerical simulation of the field must be computed using complex mathematics. The simulations used by this research are implemented in C++ which allows a class with overloaded operators to be defined. This allows the numerical integration methods to be written in the same notation as for simple scalar fields.

A number of integration methods were considered but it was determined that the Runge-Kutta  $2^{nd}$  (RK2) order integration method or Midpoint method was the most efficient. The RK2 method provides the required level of stability and accuracy without the time and memory overhead of higher-order methods. This integration method was implemented to operate with complex numbers and has been used for all simulations in this research.

### 3.2 Visualising Complex Fields

Several methods are used for visualising complex or vector fields such as: vector plots [5], image distortion [6], surface plots [7], contour plots [2, 8], interface rendering [2] or coloured field representations [2]. Each of these are valid methods of visualisation, but often either lose data or are not feasible in three-dimensions or for large field lengths. We present here a method capable of rendering large and three-dimensional fields without loss of information.

A number of software problem-solving environments and mathematical modelling tools such as Matlab [9] and Mathematica [10] do support a static visualisation of scalar and vector fields. We are particularly interested in large model systems and also in the time dependent behaviour of such models. Consequently it becomes necessary to develop our own custom *fast* simulation codes and associated interactive visualization tools.

To achieve this fast visualisation, a hardware-accelerated graphics library is required. In our previous research [1] our simulations were written in Java<sup>TM</sup> [11] and visualised

using JOGL [12]. As the simulations in this research are written in C++, OpenGL has been chosen to provide the hardware-accelerated graphics capability.

### 3.3 Visualisation in Two-Dimensions

In previous work we have discussed the visualisation of a scalar field equation [1], however vector fields such as the Ginzburg-Landau equation present even more of a challenge. Because the field  $u$  is a complex number there is no longer a single value that can be converted into a colour to represent the state of the cell. Each of the cells within a vector field have a real and imaginary part (which can be transformed into phase and magnitude values). The challenge is to develop a method that can convey all the necessary data to the observer in an easy to interpret way.

#### Vector Visualisation

The first method developed to visualise a Ginzburg-Landau field displays each cell as a vector (also known as arrow plots [13]). Instead of a single square of colour often used to visualise scalar fields, each cell is represented by an arrow. Each cell is allocated a square of space and an arrow is drawn from the centre of the square to an offset point defined by the real and imaginary parts of the cell's value. This can be seen in Figure: 2.

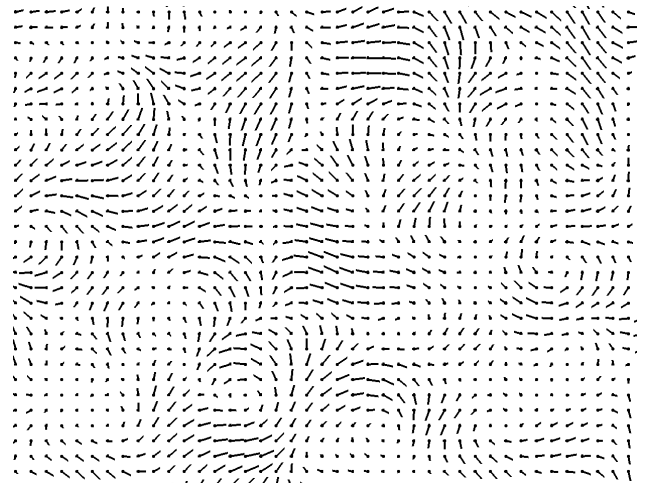


Figure 2: A Ginzburg-Landau field visualised using the Vector Visualisation method. The phase information or “directions” of the complex field variables are rendered as arrows, with length scaled by field magnitude.

This method displays all the raw data of the field, but it is not easy to see all the domains in the field and is only feasible for displaying small fields (the arrows cannot be distinguished

in a large field). However, this visualisation provides vital insight necessary for developing an improved method. It can be seen in the Figure: 2 that the cells have aligned themselves in terms of phase and magnitude, display either of these values will provide a useful visualisation.

Colouring the vector field can act as a guide to the eye, but this approach is still limited in making it clear where the model field is forming structures and interfaces are being formed.

### Phase Visualisation

The phase visualisation method includes only the phase of the cells in the visualisation, the phase of a cell can be easily calculated from its real and imaginary parts. The phase of the cells provides a single value that allows the complex Ginzburg-Landau field to be visualised in a similar way to a simple scalar field. A *rgb* colour  $c_{(x,y)}$  is calculated from the absolute value of each cell's phase  $|\phi_{(x,y)}|$  using equation: 6.

$$c_{(x,y)} = (1.0 - |\phi_{(x,y)}|, |\phi_{(x,y)}|, 0.0) \quad (6)$$

Once these colours have been calculated, the method visualises the field as many squares where each square's corners are located at the centre of four neighbouring cells. The four colours at the corners of these squares are automatically blended by OpenGL to produce a smooth visualisation even for small field lengths. A screen-shot of a field visualised using this method can be seen in Figure: 3.

This method of visualisation is easier for the human eye to interpret than the vector method and is still useful for large field sizes. However, all information about the magnitude of the cells is lost. As the magnitude of the field is not uniform across the field it represents important information and must be included in the visualisation in some form.

### Magnitude Visualisation

Like the phase of the cells, the magnitude is a single value calculated from the real and imaginary parts of the cell and can be visualised in a similar manner to the phase. Visualising the magnitude suffers the same issue as the phase visualisation in that it loses some information about the cell values. However, this method shows the patterns and information that the magnitude values form and is a necessary step in producing an improved visualisation.

Figure: 4 is an example image created by this method of visualisation. The patterns the magnitudes form are unlike those seen in the Phase Visualisation and thus represent important information that must be included in the visualisation.

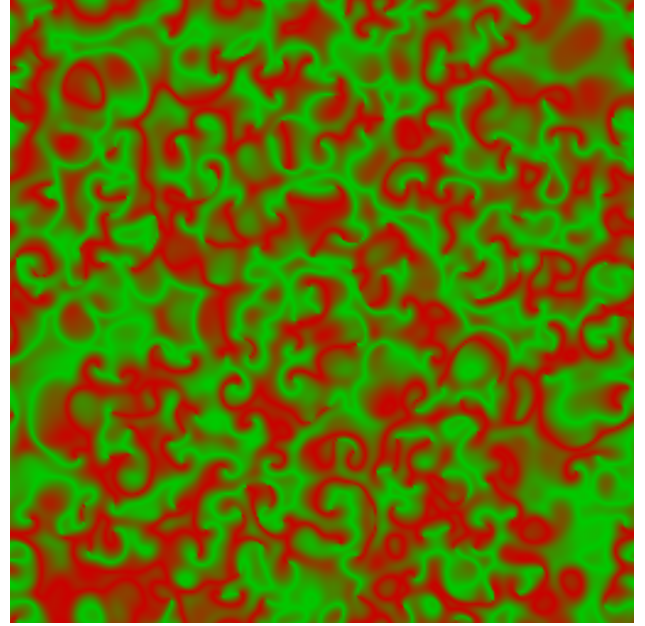


Figure 3: A Ginzburg-Landau field visualised using the Phase Visualisation method. This works well for this particular model configuration and highlights the formation of vortex structures.

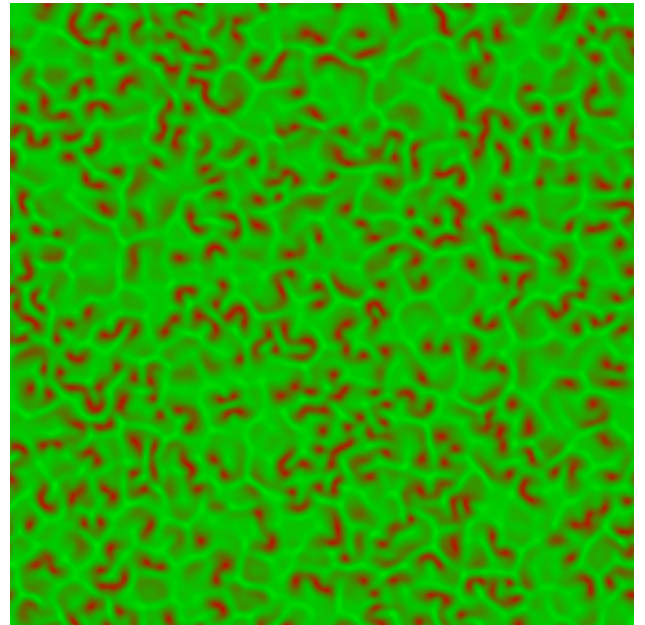


Figure 4: A Ginzburg-Landau field visualised using the Magnitude Visualisation method. This method is less insightful than visualising the phase information – for this particular data set.

There is a need for a method of visualisation that displays all of this information in an easy to interpret way.

### Phase-Magnitude Visualisation

The phase-magnitude visualisation method is designed to visualise both the phase and the magnitude of the field in a single display. It combines the previous two methods by defining the ratio of red and green according to the phase and then scaling by the magnitude. This is effectively controlling the hue according to phase and the brightness according to magnitude. The *rgb* colour  $c_{(x,y)}$  is defined by the absolute value of the cell's phase  $|\phi_{(x,y)}|$  and magnitude  $|u|_{(x,y)}$  according to equation: 7.

$$c_{(x,y)} = ((1.0 - |\phi_{(x,y)}|) \times |u|_{(x,y)}, |\phi_{(x,y)}| \times |u|_{(x,y)}, 0.0) \quad (7)$$

Using this calculation for the colour, both the phase and magnitude is visualised in a single image. Figure: 5 is an example of this method. In this image the phase is visualised by the red-green or hue of the image and the magnitude can be distinguished by the dark and bright areas. In this way a vector field can be visualised without loss of information.

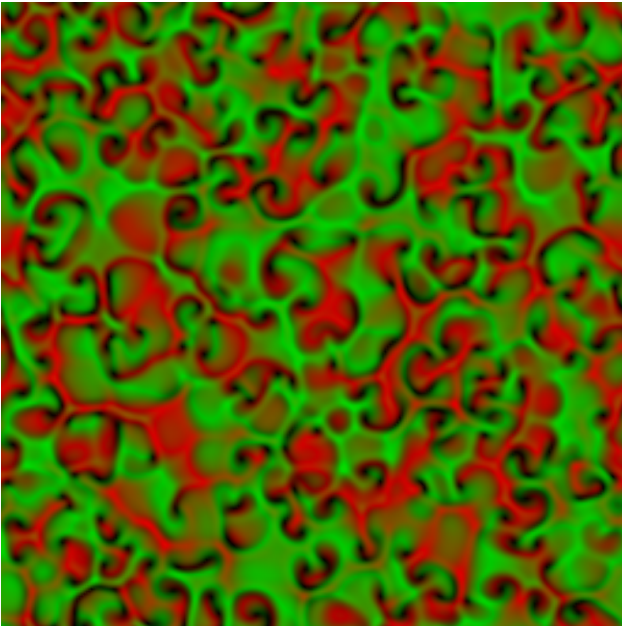


Figure 5: A Ginzburg-Landau field visualised using the Combined Visualisation method. The red and green areas represent cells with phase 0 and  $\pi$  respectively. The dark areas represent cells with a magnitude close to 0.0 and the bright areas magnitudes tending to 1.0.

### Surface Distortion Visualisation

The surface distortion visualisation method is based to the phase-magnitude visualisation method in that the cells are represented by a number of squares coloured according to equation: 7. However, the squares are no longer limited to a single plane. The surface is distorted based on the magnitude of the cells. The height of each square's corners is distorted by the magnitude of the four cells it covers. This method is designed to increase the visibility of the magnitude values to allow easier interpretation of the image. Such a visualisation can be seen in Figure: 6.

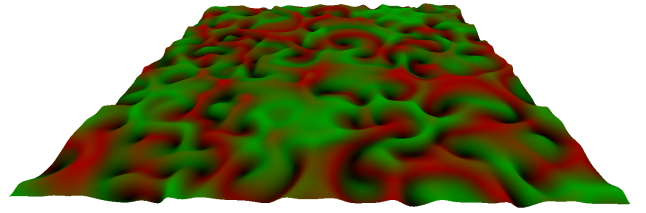


Figure 6: A Ginzburg-Landau field visualised using the Surface Visualisation method. The changing boundaries in the model field show up as crenulations or vein-like structures in the rendered surface.

This Surface Distortion Visualisation method provides an effective way of conveying information about the phase and magnitude of the two-dimensional Ginzburg-Landau equation. As both the phase and magnitude are visualised, this method should be able to visualise any complex field effectively, however the exact parameters for distortion and colouration are based on the tendency of the Ginzburg-Landau cell magnitudes in the range  $[0, 1]$ . This may not be the case of other complex fields and certain adjustments to the visualisation may be necessary on a case-by-case basis.

## 3.4 Visualisation the Three-Dimensions

Visualising a complex field in three-dimensions is even more challenging than in two. In three-dimensions, visualising the cells as vectors is no longer feasible as the arrows become impossible to see. Instead the field can be rendered with colour values determined from the phase and magnitude values (as with the two-dimensional case). We present here three methods of rendering a three-dimensional Ginzburg-Landau field.

### Combined 3D Rendering Method

The simplest method developed for visualising the three-dimensional field is by rendering six faces for each side of



the field. This is performed simply by rendering six rectangular faces coloured exactly the same manner as the combined visualisation method. A snapshot of a field visualised in this way can be seen in Figure: 7.

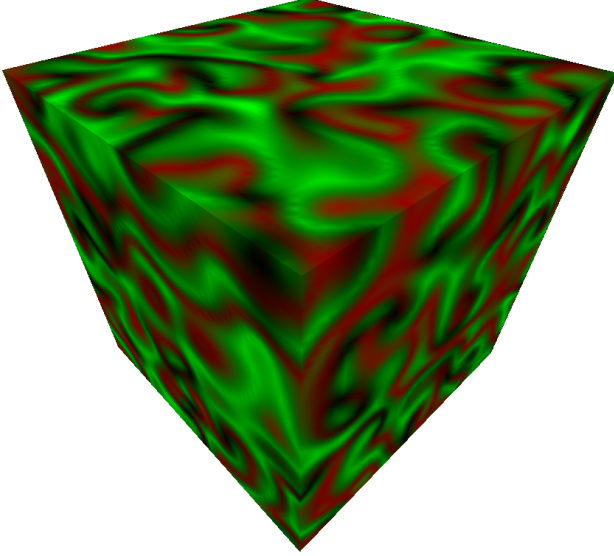


Figure 7: A snapshot of the Phase-Based Surface Distortion Method. The 3-d cube is rendered with flat surfaces that are only coloured by the field data.

This method is effectively the three-dimensional application of a two-dimensional rendering method. The method does display the field but loses all information about the centre of the field. Only the outside faces of the cubic field can be seen.

#### Combined 3D Surface Distortion Method

The second method of rendering a three-dimensional field uses a similar method as the two-dimensional surface-distortion visualisation method. Like the combined 3D rendering method, the field is displayed as six faces displaying the outside of the field and coloured according to phase. However, the positions of the rectangles representing each cell is distorted according to the phase of the cell.

Figure: 8 shows a snapshot of a Ginzburg-Landau field visualised using this method. This distortion and colouring coupled with suitable surface normals provide a three-dimensional visualisation that is easily interpreted by the human eye. However, like the previous method the visualisation is limited by the fact that it can only display information on the edges of the field, the internal state of the field is unseen.

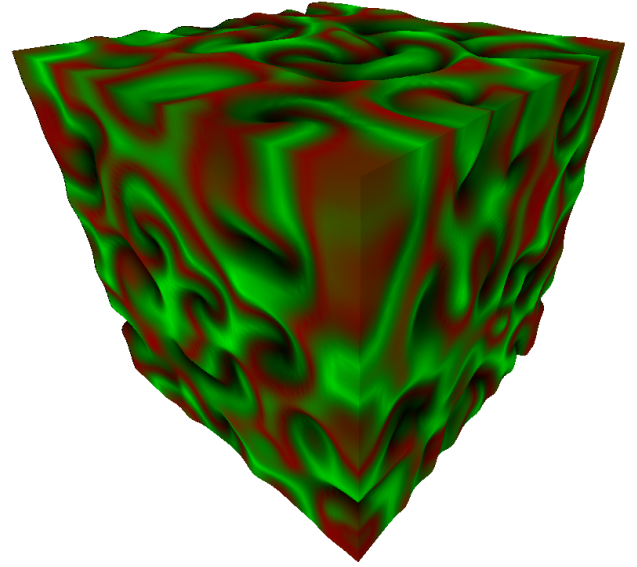


Figure 8: A snapshot of the Phase-Based Surface Distortion Method. The field data are used to distort the surfaces of the cube as well as to colour them. This emphasises the forming boundaries.

#### Interface Rendering Method

The interface rendering method was developed to address the issue of visualising data from within the centre of the three-dimensional field. To overcome the problems faced by the previous methods, this method does not attempt to display the information of all the cells of the field. Instead, only the interfaces between regions are rendered. Rendering only these interfaces allows the user to rotate and move the model to see details from all sections of the field. Results from this method can be seen in Figure: 9.

There has been previous reporting in the literature of rendering of interfaces boundaries for both scalar and vector fields; however, the interface visualisation has been a single colour and thus information has been lost. This method colours the rendered interfaces and thus conserves data while also allowing the internals of the field to be visualised.

## 4 Discussion and Conclusions

We have presented some approaches to visualising the complex field variable of the Time-Dependent Ginzburg-Landau equation as it might be simulated on a regular mesh in multiple dimensions.

We have explored a number of ways of representing the real and imaginary parts of the field, in terms of phase  $\phi$  and

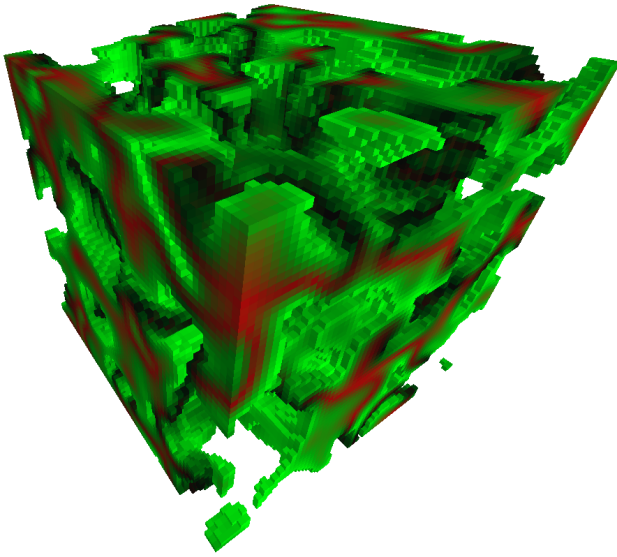


Figure 9: A snapshot of the Interface Rendering Method. This shows how the data field is used to guide what parts of the solid model should be rendered. Only the interface regions are shown, allowing a cut away inspection of the model interior.

magnitude  $|\psi|$  in terms of colours for scalar potential representations and as explicit arrows to render the vector nature of the field. We have described and presented renderings based on geometrical surface distortions and on interface boundary detection and subsequent colouring. Of particular note was the use of colour and model interface information to allow a cut away rendering of the evolving system, giving important insights into its interior structure and evolution.

We are developing our system to allow a more detailed and exhaustive exploration of the properties and emergent behaviours of the TDGL. Of particular interest are the vortex and interface boundary structures we found in the rendered images. Encountering these patterns visually gives insight into the mixing and phase separation behaviour but also suggests various pattern classification and detection algorithms [14] we might apply to count and categorise these structures semi-automatically.

We have framed our experiments and discussion in terms of the TDGL, but we believe these ideas and approaches would apply equally well to other field equations and model systems that are based on complex number fields.

## References

- [1] Hawick, K.A., Playne, D.P.: Modelling and visualizing the Cahn-Hilliard-Cook equation. In: Proceedings of 2008 International Conference on Modeling, Simulation and Visualization Methods (MSV'08), Las Vegas, Nevada (2008)
- [2] Aranson, I., Kramer, L.: The world of the complex Ginzburg-Landau equation. *Rev. Mod. Phys.* **74** (2002) 99–143
- [3] Abrikosov, A.A.: On the magnetic properties of superconductors of the second group. *Sov. Phys. JETP* **5** (1957) 1174
- [4] Kleinert, H.: Vortex origin of tricritical point in Ginzburg-Landau Theory. *Europhys. Lett.* **74** (2006) 889
- [5] Wittenbrink, C., Pang, A., Lodha, S.: Glyphs for visualizing uncertainty in vector fields. *Visualization and Computer Graphics, IEEE Transactions on* **2** (1996) 266–279
- [6] van Wijk, J.J.: Image based flow visualization. In: SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques, New York, NY, USA, ACM (2002) 745–754
- [7] Neuberger, J.W., Renka, R.J.: Critical Points of the Ginzburg-Landau Functional on Multiply-connected Domains. *Experimental Math.* **9** (2000) 523–533
- [8] Mel'nikov, A.S., Nefedov, I.M., Ryzhov, D.A., Shereshevskii, I.A., Vinokur, V.M., Vyshevlavtsev, P.P.: Vortex states and magnetization curve of square mesoscopic superconductors. *Phys. Rev. B* **65** (2002) 140503
- [9] The MathWorks: Matlab. available at <http://www.mathworks.com> (2007)
- [10] Wolfram Research: Mathematica. available at <http://www.wolfram.com> (2007)
- [11] Gosling, J., Joy, B., Steele, G.: The Java Language Specification. JavaSoft Series. Addison Wesley Longman (1996) ISBN 0-201-63451-1.
- [12] Bryson, T., Russell, K.: Java Binding for OpenGL (JOGL) (2007)
- [13] Post, F.H., van Wijk, J.J.: Visual representation of Vector Fields: Recent developments and research directions. In: Scientific Visualization: Advances and Challenges. Academic Press (1994) 367–360
- [14] Hawick, K.A., James, H.A., Scogings, C.J.: Manual and semi-automated classification in a microscopic artificial life model. In: Proc. Int. Conf. on Computational Intelligence (CI'05), Calgary, Canada. (2005) 135–140