UNIVERSITY OF BIRMINGHAM

Implementation of machine learning in vortex systems

M.Sci. Theoretical Physics Project Report

Raluca Pantilie
Student ID: 1913486
Supervisor: Dr. Jonathan Watkins

September 3, 2022
Word Count: 4595

Abstract

This project explores the potential of machine learning in two-dimensional vortex systems. The system chosen for this project is a rectangular, type II superconductor. Its topological transition, called melting, is the main subject of this project, as its identification through machine learning could prove the viability of similar approaches. Previous work in the area is reviewed, and relevant background revised. Ginzburg–Landau theory is particularly important, as it is the theoretical basis for the simulation used in this project. Details are given on the characteristics of the simulated material and the motivation behind the choice. The simulated data is pre-processed through Principal Component Analysis, and used for training a logistic regression algorithm, which is a classifier. The resulting model successfully identified the melting transition. Both the Principal Component Analysis and logistic regression are simplistic in comparison to more involved machine learning algorithms. Hence, the success of this approach is very promising.

Contents

1 Introduction

1.1 Motivation

Machine learning started getting more attention recently, as it became easier to employ. It has been used in meteorology for classifying clouds [1], medical physics for classifying X-rays [2], material science, in the prediction of new stable materials, and the calculation of numerous material properties [3], and in superconductivity, it was used to model critical temperature [4] and new superconducting materials [5].

However, this project is interested in the topological transitions. There are numerous papers successfully identifying different phases of matter [6, 7, 8, 9, 10]. An advantage for the validation of these algorithms in statistical physics, is that some of these transitions can be identified by a feature of the system, such as an order parameter. This lends interpretability to the outputs of machine learning algorithms, as it can be concluded that the algorithms can learn to determine such quantities. The features also serve as great ways of determining labels for training supervised algorithms.

Many of the studies on this topic use deep learning or convolutional neural networks, which are computationally expensive. In particular, Beach et al. found that one layer neural networks were unable to accurately determine the transition temperature in the two-dimensional, classical XY model. They aimed to find the critical temperature of the transition by identifying the point where the network had 50% confidence in classifying the state in either of the two categories, low temperature and high temperature. They tested two algorithms, a one layer network and a deep learning one, and found that the simpler one failed to accurately estimate the temperature for moderate sized lattices. Conversely, the deep learning network succeeded in this task, with its accuracy scaling positively with the lattice size. They also discovered that significant pre-processing drastically improves the performance of both networks [10].

It is important to note that the current project uses a simpler algorithm, smaller lattice size, and lighter pre-processing than the ones used by Beach et al. [10]. However, our aim is to observe the behaviour through the algorithm, not to estimate the critical temperature. Providing research in this simpler techniques would help in identifying regions with a cross-over behaviour that warrant further study, without spending resources in complex structures as deep learning networks. The simplicity would also help interpretability, as the process would be easier to understand.

1.2 Aims

The current project tests a simple machine learning algorithm, logistic regression. It is a one node network that classifies data into two categories by finding a hyperplane that separates the two groups of data in an arbitrary number of dimensions. The data used consisted of positions of vortices, pre-processed through PCA. This simplistic approach successfully identified the transition. Considering the high computational cost of previous successful methods, this result is promising for the future of machine learning in condensed matter and statistical physics.

The system chosen for this project is a rectangular, superconductive sheet of type II, simulated in line with Ginzburg-Landau (GL) theory. The edge effects were neglected, resulting in a simulation of the bulk of the material. This set up ensures that vortices form at the surface of the sheet, which arrange themselves in a hexagonal lattice at low temperatures. Increasing the temperature, introduces energy into the system, resulting in unpredictable movement, which can be strong enough to break the lattice. This process involves a topological phase transition called melting, which was theorized in the 1970s [11], and observed in 2009 [12]. At a low temperature, the vortices start in the triangular lattice, also named the solid phase due to its crystalline structure. Then they enter the hexatic phase with the appearance of vortices with 5 or 7 neighbours, called dislocations. Finally, the vortex lattice forms an isotropic liquid. The process is well understood and was clearly observed, making it a great candidate for the validation of this technique.

2 Background

2.1 Superconductivity

2.1.1 Experimental

The phenomenon of superconductivity was discovered in 1911 by H. Kamerlingh Onnes. He measured the resistivity of mercury, lead, and tin at very low temperatures. He found that it dropped to zero when decreasing the temperature under a certain critical temperature (under 10 K) [13, 14].

Superconductors are not strictly perfect conductors. In 1933 Meissner and Ochsenfeld discovered that a superconductor in a magnetic field will expel the field from itself, up to a thin surface, called the penetration depth, λ. Due to perfect conductivity, the behaviour was expected when the field was introduced after cooling the sample to the superconductive state. However, when the field was present before the transition, it was expected for the sample to trap the field in itself. This did not happen, instead, it repelled the field up to a penetration depth λ. The phenomenon is called perfect diamagnetism or the Meissner effect [13]. Its existence implies that the application of a field strong enough would destroy the superconducting state, even below its original critical temperature. That is exactly what was found, with pure metals losing their superconductive state at a lower field than alloys [15]. Which motivated more research into those alloys.

What was found is that many alloys behave differently when the magnetic field is increased, as to break the superconductive state. These materials would exhibit the full Meissner effect below a critical field H_{C1}. However, above this threshold, the material allows the field to flow in isolated pockets, called vortices. Which means that there are areas of normal material in a superconducting sample. For this reason, this stage is called a mixed state. As the field increases, they become larger, until H_{C2}, where superconductivity is broken. This type of material is now called a type II superconductor. To this day, alloys are generally type II superconductors, while pure metals are usually type I.

2.1.2 Mathematical modelling

It took many years for researchers to develop a mathematical understanding for superconductivity. The first attempt was phenomenological, the London equations described the phenomenon, without attempting to find its cause. Then the Ginzburg–Landau theory (GL) was developed, a macroscopic description of superconductivity. In its first iteration, it only described type I superconductors. However, later work by Abrikosov proved its usefulness in describing those of type II. To understand how he did so, an overview of GL follows.

2.1.3 Ginzburg–Landau theory

The theory is based on the assumption that the free energy can be expressed in terms of a complex parameter $\psi(\mathbf{r})$. This parameter is connected to the superconducting behaviour by being linked to the density of superconducting electrons (n_s), as described in the London equations. These electrons determine where the superconducting state is present, therefore $|\psi(\mathbf{r})|^2 = n_s$ ensures that the order parameter is linked to the superconductive behaviour. As such, $\psi(\mathbf{r})$ is zero at any point r that doesn't exhibit superconductivity.

The free energy f is expanded very close to the transition temperature, so $\psi(\mathbf{r})$ is very small; f is expanded with respect to $|\psi(\mathbf{r})|^2$. Then it is noted that the free energy of the material is equal to that of a normal state where $\psi(\mathbf{r}) = 0$. This expansion is used to generate the Ginzburg-Landau equations by minimizing the free-energy density with respect to ψ and the external field. Which reveals the characteristic length, ξ, as follows

$$\xi^2 = \frac{\hbar^2}{2m^*|\alpha|}.$$

Solving the equation shows that the characteristic length controls the decay of a small disturbance in $\psi(\mathbf{r})$ [13]. This parameter can be combined with the penetration depth λ to form the dimensionless Ginzburg-Landau parameter:

$$\kappa = \frac{\lambda}{\xi}.$$

The initial calculations were done assuming a small κ. They showed a sharp behaviour change at H_C, denoting a rapid loss of superconductive properties. This was in line with the behaviour of type I superconductors. However, in 1957, Abrikosov explored the case of κ being very large. This case was very different to the first one, where the Meissner effect would disappear abruptly at a field H_C. Starting at a critical field H_{C1}, the material allowed more of the field to pass through vortices until it would return to a normal (non-superconductive) state at a field H_{C2}. He also found this behaviour to be associated with $\kappa > \frac{1}{\sqrt{2}}$. Due to the drastic difference in behaviour between κ being above or below $\frac{1}{\sqrt{2}}$, Abrikosov and Zavaritskii proposed the classification of superconductors in groups, which changed their name to the now recognizable type I ($\kappa < \frac{1}{\sqrt{2}}$) and type II ($\kappa > \frac{1}{\sqrt{2}}$) [16]. In his work on the mixed state, he calculated the stable configuration of vortices to be in a square lattice, later work by Kleiner, Roth, and Autler proved that the hexagonal lattice is more stable [13, 17]. Three years later, Essmann and Träuble directly observed the hexagonal lattice [18]. This configuration of the mixed state is also called the Abrikosov lattice, despite the original error.

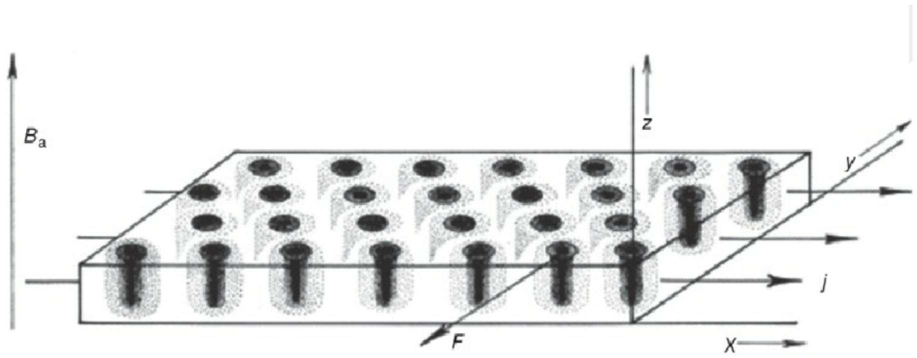


Figure 1: Abrikosov lattice in the mixed state of the bulk of a rectangular superconductor under a transport current of density j. Image taken directly from Fig 5.7 in [19]

2.1.4 Vortices

As the project involves simulating the vortices, they need to be examined in more detail. This section will only focus on vortices in the system relevant for the project. Namely, the vortices are considered to be small, meaning their radii are small compared to λ. This allows for the vortices to be approximated as point-like in simulations. The system used in this project is the bulk of a rectangular superconductor, the Abrikosov lattice in this material is schematically shown in figure 1. The thickness of the sheet is small enough for the vortices to penetrate the sample completely, this makes it possible to study the behaviour of vortices by approximating the sheet to a 2D plane. However, a thin superconductive film does not experience the melting phase[20] for this reason, the superconductor simulated is this project has a thickness $d \gg \lambda$. All results quoted below depend on the characteristics stated above.

Vortices are created by magnetic flux penetrating the material in rods of current, around these rods current moves in a circular motion, as the surrounding material still obeys the Meissner effect. This gives rise to a repulsive force between the vortices. The interaction generated by it can be calculated by considering the free energy contribution from two vortices separated by r_{12} [13]. The force is

$$F_{12} = \frac{\Phi_0^2}{2\pi^2\mu_0\lambda^3} K_1(\frac{r_{12}}{\lambda}) \tag{1}$$

where Φ_0 is the total flux carried by a unit cell ($\Phi_0 = hc/2e$), and K_1 is the modified Bessel function of the second kind. The force acts in the radial direction r_{12} [21].

This repulsive force is responsible for the hexagonal lattice being stable at a low temperature. Raising the temperature would result in the vortices gaining energy, which transfers to unpredictable movement. Repeatedly increasing the temperature results in a system where the ordered lattice in no longer present, a process called melting [21, 22, 12], which is of interest for this project.

The force due to temperature is controlled by a thermostat $\chi(T)$. In the simulation, it was handled by having a one percent chance of a vortex to have $\chi(T) \neq 0$. As such,

$$\chi(T) = \begin{cases} \sqrt{T}\gamma \text{ for } 1\% \text{ of vortices chosen at random every time step} \\ 0 \text{ otherwise} \end{cases} \tag{2}$$

where γ is a sample of a 2D Gaussian distribution, ensuring the random direction of the force.

With both of these forces, the equation of motion for the vortices in this project can be calculated. The final force defining the movement of a vortex, in the conditions imposed by this project, is known as

$$F = F_{12} + \chi(T). \tag{3}$$

2.1.5 Melting the lattice

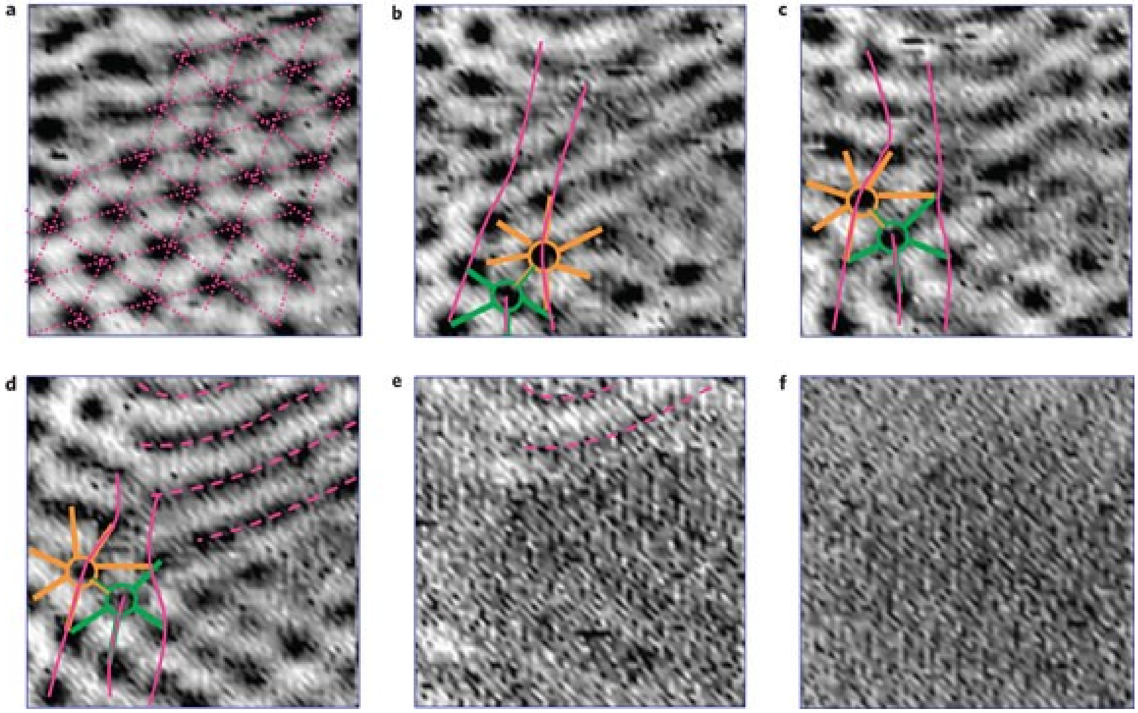


Figure 2: Melting transition directly observed by scanning tunnelling spectroscopy under a magnetic field of 2 T a. The hexagonal lattice, solid state, at 1.2 K; b. Beginning of the hexatic phase at 1.9 K, a dislocation can be seen in the bottom left. The orange and green lines correspond to dislocations, and the magenta lines highlight the effect they have on the lattice; c. Further development of the hexatic phase; d. At 2.1 K the isotropic liquid phase starts to appear on the right of the image, coexisting with the hexatic phase; e. At 2.8 K, no structure can be identified in the middle of the image; f. At 3 K, there is no structure visible in any part of the image, meaning that the vortices are randomly distributed, forming an isotropic liquid. Images taken directly from Fig 3 in [12].

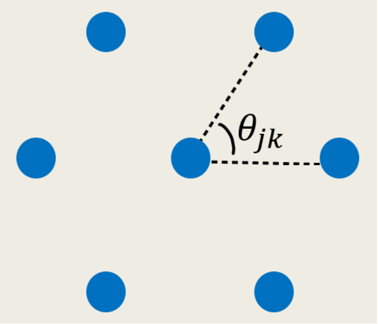


Figure 3: Schematic depicting the angle used in the hexatic order calculation

Now that the force governing the movement of vortices was reviewed, the melting process itself needs to be developed. Up to this point, it was described as the hexatic lattice breaking under increasing temperature. This process was theorized [11] in the 1970s, and observed [12] in 2009. Some of the theory suggests that the transition is caused by dislocations (formations of vortices with 5 or 7 neighbours rather than 6). However, the cause of the transition is not the object of study for this project.

As outlined in section 1.2, the vortices start in the Abrikosov lattice, named the solid state. After a temperature increase, they enter the hexatic phase with the appearance of dislocations. Finally, the vortex lattice forms an isotropic liquid, as vortices are spread randomly in the available space. This process was captured by Guillamón et al. in figure 2 by using use scanning tunnelling spectroscopy to image the lattice as the temperature was increased [12].

It is important to note that the two-dimensional (2D) melting transition does not involve long range order, as it is impossible to achieve in a 2D environment by the Mermin–Wagner theorem [23, 24]. The ordering referred to in this project is an orientational order, controlled by the hexatic order. This means that the orientation of the neighbours determines the value of the hexatic order parameter Ψ_H, as:

$$\Psi_H = \left| \frac{1}{N_v} \sum_{j=1}^{N_v} \frac{1}{z_j} \sum_{k=1}^{z_j} e^{i6\theta_{jk}} \right|^2 \tag{4}$$

, where N_v is the number of vortices, while z_j is the number of neighbours of each vortex. The angle θ_{jk} is one between vortex j and its neighbour k, taken as pictured in Figure 3 [22]. If all angles are equal to each other, this results in $\Psi_H = 1$, while random angles result in $\Psi_H = 0$.

In order to measure these angles, Delaunay triangulation was used. To understand it, one can imagine a set of random points. First, a Voronoi diagram is made, which divides the space in polygons, with each point inside a such polygon being closest to one data point. Any space that is equally distanced between two or more data points represents a boundary between two polygons. Then, any two points whose polygons share a boundary are linked. Any non-triagular polygons are then partitioned into triangles. This process results in the Delaunay triangulation [25]. The relationship between a Voronoi diagram and Delaunay triangulation is showcased in figure 4, where both were performed on four random points. The figure was done in python through the scipy.spatial environment, which includes programs that produce both Voronoi diagrams and Delaunay triangulations.

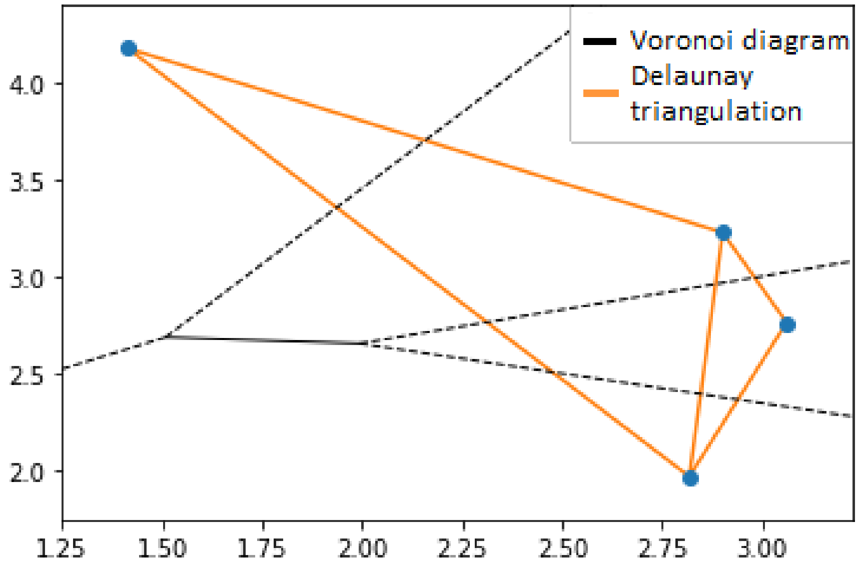


Figure 4: Voronoi diagram and Delaunay triangulation of four random points. The black lines are the bounderies of the Voronoi polygons, while the orange ones are the lines of the Delaunay triangulation, and the blue points are the original data points.

2.2 Dimensionality reduction

The background above concentrated on the characteristics of superconductivity that are important for properly simulating the system and procuring relevant data. However, this data needs to be processed before being introduced into a machine leaning model due to the high number of dimensions of the original data. A machine learning algorithm can not fit a model to a data set with more dimensions than data points. However, this situation occurred due to the number of coordinates for all vortices (dimensions) being higher than the number of snapshots of the system (observations).

To rectify this, dimensionality reduction was used, via Principal Component Analysis (PCA). This is a feature extraction technique that is used to reduce the dimensions needed to encode data, while minimizing loss of information. The way this is done can be explained by considering an example data set **X**, with a number of parameters p for n objects, resulting into an n × p matrix. This means that its columns are n-dimentional vectors: x_1, x_2, ..., x_p. To conserve as much information as possible, the program finds a linear combination of the columns of matrix **X** with the maximum variance. This is done by solving an eigenvector problem with the linear combination being $\sum_{j=1}^{p} a_j x_j$=$\mathbf{Xa}$, where **a** is a vector of constants, and the variance is given by $\boldsymbol{a^tTa}$, where **T** is the sample covariance matrix of **X**. The unit-norm vector restriction can be applied, meaning $\boldsymbol{a^t a = 1}$, where $\boldsymbol{a^t}$ denotes the transpose of $\boldsymbol{a}$. This results in the variance maximization problem becoming $max(a^tTa - \lambda(a^ta - 1))$. After differentiation with respect to **a**, the problem becomes a standard eigenvalue problem:

$$\boldsymbol{Ta = \lambda a}$$

. Revisiting the variance by applying the equation above results in $var(\boldsymbol{Xa}) = \boldsymbol{a^tTa} = \lambda \boldsymbol{a^t a} = \lambda$. Therefore, the greatest variance can be found by simply finding the highest eigenvalue, λ_i, while its corresponding eigenvector, a_i, is the basis that expresses this variance [26]. So, the principal component $\boldsymbol{X}a_i$ carries the highest variance. The computational application automatically orders the eigenvalues from highest to lowest.

The result is that of a set of principal components, the first component always encodes the most variance, and the others codify less and less of it until there are as many principal components as original dimensions. For many data sets, the first couple of components will encode a large amount of variance. Checking for this behaviour is typically done with a scree plot. This graph shows the amount of variance encoded in each component. Ideally, for dimensionality reduction, the scree plot would show a sharp drop in variance at a certain component number, as this would indicate a natural cut-off point. If that is not the case, one needs to make a decision of where the cut-off point should be. Once a cut-off is decided, all data can be projected on the principal components before the cut-off, discarding the ones after it. Due to this procedure the amount of variance, and thus information, preserved can be controlled. In this application, the cut-off point conserved was at least 95% of the variance.

2.3 Machine learning

After the pre-processing stage outlined above, the machine learning model was trained. The type of algorithm chosen for this project was a logistic regression classifier. This is a supervised algorithm, meaning that it takes pre-labeled data and finds a hyperplane that can separate the two classes. This type of classifier inputs the data characteristics in a linear function to produce the predicted label. The sum of the differences between the predicted and real labels is minimized via successive changes to the initial function [27]. Therefore, the basic challenge of a classifier is to minimize

$$\sum \sigma(aX + b) - l$$

, where X is the input data, l is the input label, σ is the sigmoid function, while a and b are constants. The constants are modified by the program to minimize equation 2.3. The sigmoid function is used to place the value between 0 and 1 in order to use it as a probability of being in one of the two classes. It has the form $\frac{1}{1+e^{-x}}$, forming the curve pictured in figure 5.

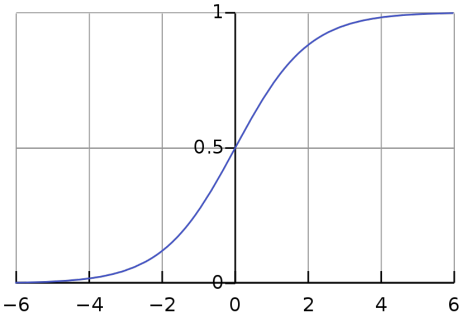


Figure 5: Graph of the sigmoid function $\sigma = \frac{1}{1+e^{-x}}$

To visualize the product of the process mentioned above, let us consider a test data set forming two clusters. The classification can be done using the logistic regression model from the Python module Scikit-learn [28], the same that was used to classify the real data. The resulting classification can be found in figure 6. It can be observed that the green region, the one where the program is uncertain of the label, is very narrow. This is due to the sigmoid function and its sharp increase in the $y = 0.5$ region. However, this algorithm can only separate clusters with a hyperplane. Due to the high number of dimensions of the real data, it is unknown whether it is linearly separable, meaning that it could be separated by a hyperplane. For example, the real data could be comprised by a cluster nested in another one, which would mean that logistic regression would be unable to classify it. In that case, a more complex algorithm, as deep learning, would be needed.

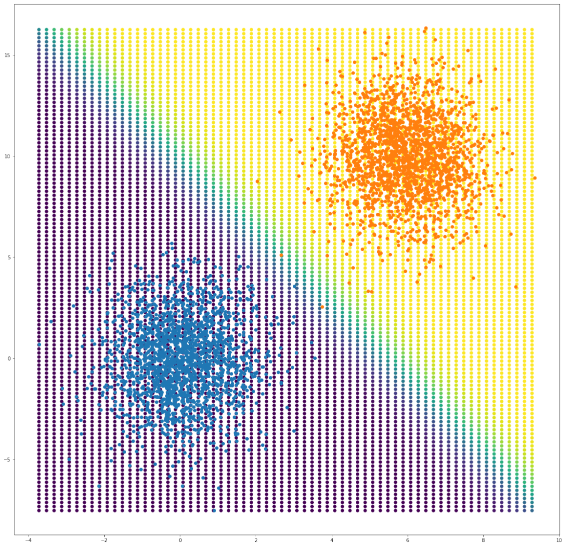


Figure 6: Logistic regression classifying a test data set formed by two clusters. The light blue dots represent a cluster labelled as 0, while the dark yellow ones were labelled as 1. Then, the program, fitted on the two clusters, was asked to output the probability for any point to be labelled as 1, with a darker colour denoting a very low probability, the lighter colour indicates a high probability, and the green one signifies uncertainty in labelling.

3 Method

As stated, the aim of the project is to train a classification algorithm to distinguish between two configurations: the "solid" and melted states. In order to do this, a simulation was developed in line with the Ginzburg-Landau theory.

3.1 Simulation

The project simulated the bulk of a type two superconductor, in its mixed state. The vortexes were approximated to points, which followed the equation of motion outlined in section 2.1. They moved in a 2D space of size 20 by 17, with double periodic boundary conditions, to avoid surface effects. The simulation was initialized in the stable triangular lattice with 320 vortices. It was left to stabilize for 1500 time steps of 0.1 before the temperature was increased. The simulation ran for 15000 time steps. It started from a temperature of 0, increasing by 0.025 until reaching 0.25.

The movement of vortices was determined by two forces: the vortex-vortex interaction and thermostat. The interaction between vortices was calculated according to equation 1, while the thermal effects were based on equation 2. Their contributions are summed by equation 3, leading to the equation of motion

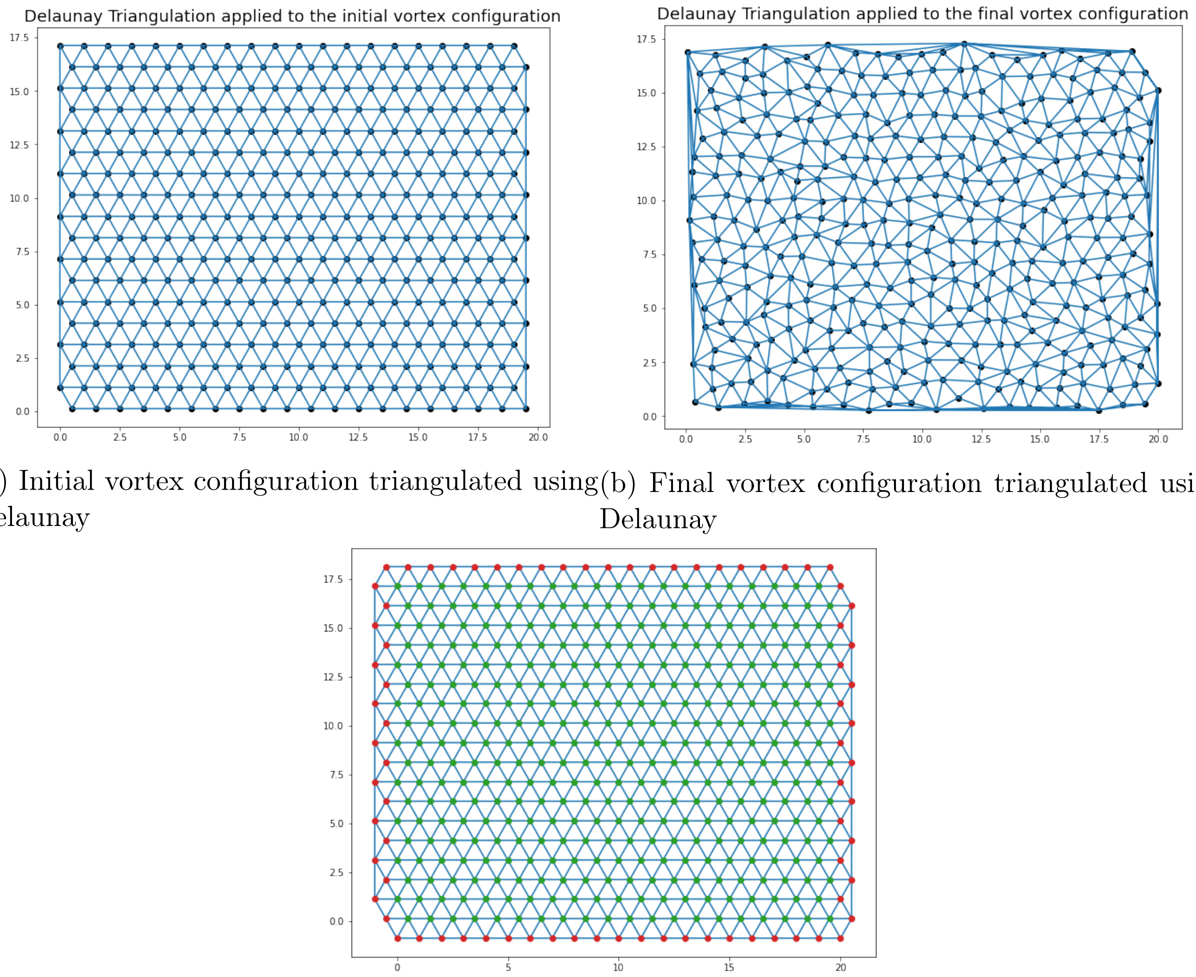
$$m^*\ddot{r} = F - \eta\dot{r}$$

, where η is the viscous drag coefficient. Therefore, each vortex moved by δx in time δt, where $\delta x = \int_t^{t+\delta t} F/\eta \; dt$. By applying the Euler technique

$$\delta x = \delta t \; \frac{F}{\eta} \tag{5}$$

, due to the error of the Euler method being directly proportional to δt, the time increments must be small. In all simulations $\eta = \lambda = 1$ and $\delta t < 10^{-2}$.

For every frame of the simulation, the hexatic order was calculated according to equation 4. Its neighbours were identified via the Delaunay Triangulation, using the Delaunay class of the Python package scipy.spatial, which is plotted in figure 7. This was done by using the list of neighbours provided by the triangulation method and the triangles they share with the main vortex to identify the right angle. Figure 7a shows how the edge vortexes do not form a hexagon, which would affect the hexatic order calculation. Since this project is not interested in studying edge effects, a set of pseudo-vortices were added to the edges as pictured in figure 7c. The pseudo-vortices were only included in the triangulation, they are not included in any of the calculations concerning the dynamics of the system.



(a) Initial vortex configuration triangulated using Delaunay

(b) Final vortex configuration triangulated using Delaunay

(c) Initial vortex configuration triangulated using Delaunay, with the addition of border points (in red), meant to ensure the accurate measurement of the hexatic order at the edges of the system.

Figure 7: Delaunay Triangulation applied to the initial and final vortex configurations.

This allowed for the cross-over behaviour to be identified via figure 8 and for the appropriate data to be extracted. The parameter doesn't reach one due to computational limitations, however it can be seen that the difference between the highest values are around 0.8, while the lowest around 0.2. This difference still enables making the distinction between two states. Figure 9 shows the order parameter with respect to time elapsed for each temperature. It can be seen that the stabilization period of 1500 time steps allows for the following temperatures to settle: 0, 0.2, and 0.23. However, only extracting data from those temperatures would result in too few data points for the solid state. Due to time constrains, more settling time was not an option, therefore temperature 0.03 was also included. While it is not ideal, the states still have a high hexatic order parameter, so the positions of the vortices still form a solid lattice. The other temperatures were discarded from the training data.

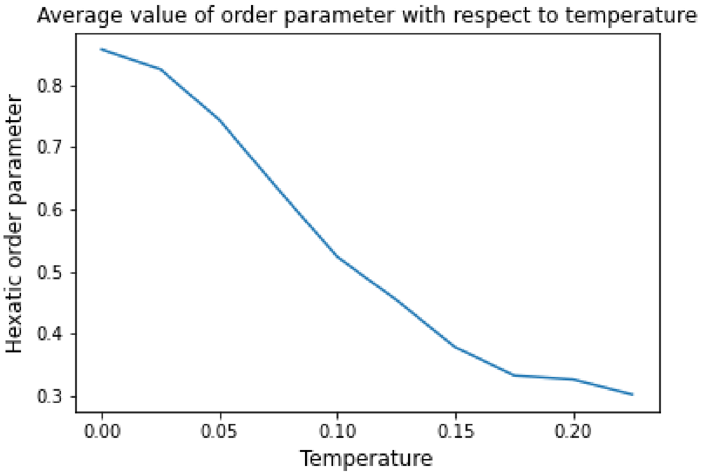


Figure 8: Hexatic order averaged over the last 750 time steps for each temperature

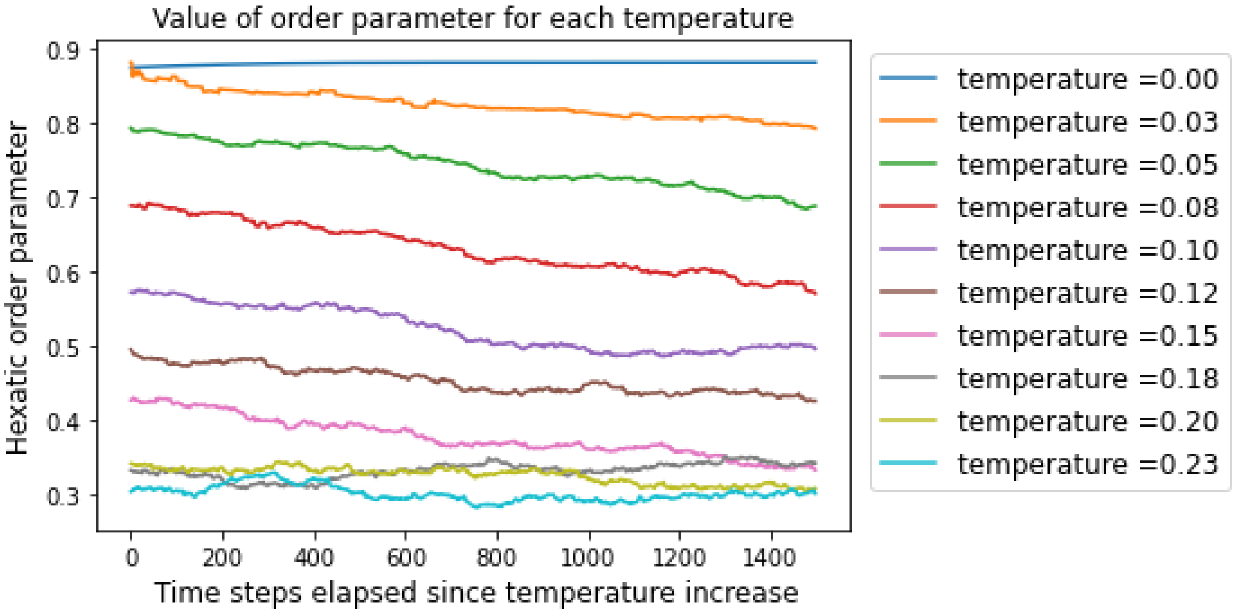


Figure 9: Hexatic order at each time step since the temperature was increased. The system was not reset or altered in any was before the temperature increase.

3.2 Data extraction and preprocessing

Training and testing data was extracted by taking the coordinates of the vortices every 50 time steps. The "solid" state data was extracted from the first 2550 time steps, spanning the first temperatures 0 and 0.03. The melted state was characterized by the last 2550 time steps, corresponding to temperatures 0.2 and 0.23. This resulted in 50 states per category. The coordinate data for each time step was flattened into one dimension, resulting in two data sets of 50 components in 720 dimensions. In this project, the model was trained to distinguish between the two cases. Therefore, while the data points suitable for training are important, they are used later. At this point, it is important to notice that extracting the training and testing data would result in a set of 100 points in 720 dimensions. However, in order to train a machine learning algorithm, the number of dimensions(720) needs to be reduced under the number of observations (100).

This was done via Principal component analysis (PCA). As outlined in section 2.2, this technique reduces the number of dimensions used to describe a data set, while preserving the information contained as much as possible. This is done by identifying the dimension that describes the most variability and doing the same for the next dimension, provided it is linearly independent of the previous ones. This procedure is repeated until the entire data set can be fully described by the dimensions calculated. The result is a set of dimensions that can provide a new basis, one where the data can be projected to a smaller number of dimensions while retaining most of the data variability.

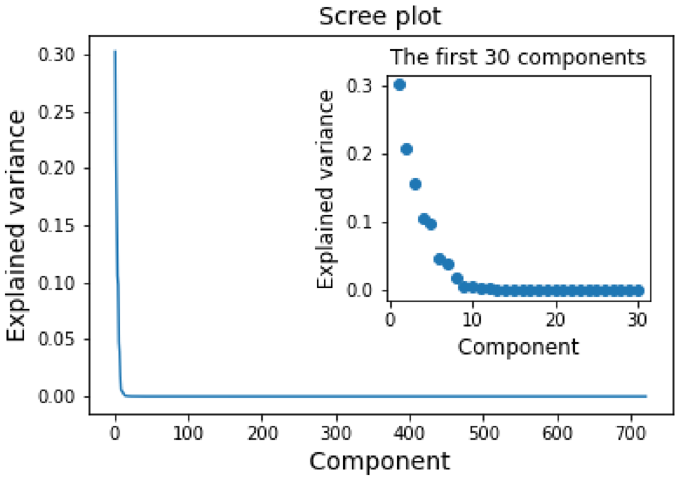


Figure 10: Most of the variability in the data set can be explained by a very small number of components. Due to this, a smaller plot was made, showing only the first 30. The smaller plot was also made into a scatter plot, so that individual values can be seen more clearly. This plot makes it clear that PCA is feasible, as it shows that very little information would be lost through dimensionality reduction.

The resulting data set must retain a significant part of the information from the original data. To ensure this, a scree plot was made. As figure 10 shows, 99% of the variability is contained in the first 20 dimensions. As such, the number of dimensions comprising the final data set needs to be greater than 20 and less than 100. This project used 70 dimensions, as it provided the best results.

Therefore, the final data set can be formed by extracting every 50th column of the matrix for the first and last 2500 columns, representing the solid and melted states, respectively. The explanation for this extraction can be found above. The data set consisted of 100 points in 70 dimensions, which is fit as a basis for training a model. At this stage, the labels were made by defining an array of zeros and ones that correspond to the solid and melted states. Namely, the first 50 values being 1 and the last ones being 0. The data and label sets were split between training and testing data sets.

4 Results

Finally, the model was trained and evaluated, using the Python environment sklearn. The random_state variable was set to 87, but similar results were achieved with it set to 5, 36, 52, 66, and 71. It's score on the training data was 1 and the confusion matrix was: $\begin{bmatrix} 15 & 0 \\ 0 & 15 \end{bmatrix}$ A test data set was composed of random values in a 70 by 100 matrix. These values were expected to resemble a melted state, due to the randomization. The model successfully categorized all values as melted. Likewise, for the solid state, synthetic data was tested. In its case, a perfect triangular lattice was constructed and preprocessed through PCA, just as the training data. The model successfully labelled this state as "solid".

As stated before, a perfect hexagonal lattice is associated with a high hexatic order parameter value. As the model was trained to assign one to a solid state, plotting the hexatic order and predicted labels for all states could be useful. The model was asked to predict the labels of all states simulated. In figure 11, the probability of the predicted label being 1 (solid state) was plotted in blue, while green represents the hexatic order of the same states. The extremes of the graph are well labelled, with the exception of a downturn around the 1700 time step mark. This makes it clear that the model can differentiate between solid and melted states reasonably well.

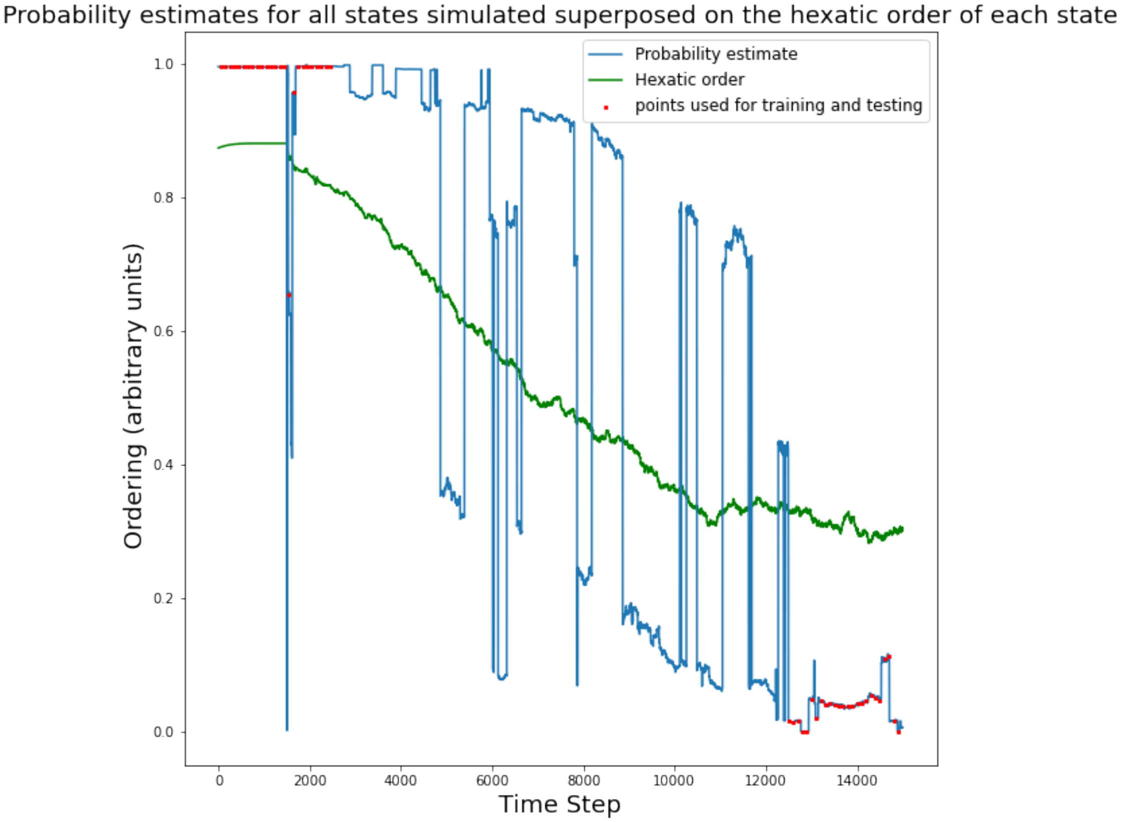


Figure 11: In blue: the probability of a state being labelled as solid by the model. In green: the hexatic order parameter for each state.

However, it is important to note that 100 of the states used in figure 11 were used to train and test the classification model. These points are marked in red, they are spaces at intervals of 50 time steps, and only 70 were used in training. However, since the model achieved a score of 1, it is already known that the test points were identified correctly.

Another important observation is the importance of the dimensionally reduced basis. Selecting the training data first and performing PCA, rather than performing PCA before data selection, results in different data sets. A model trained on one set of data can not distinguish the states in the other. Likewise, a model trained on the data of one simulation fails on the data of another one, when the new data is preprocessed by PCA independently. These results imply that the model can only work on a data set that has the same basis as the one it was trained on.

5 Conclusion

In this project, a logistic regression algorithm was successfully trained to identify the solid state of a type II superconductor. The data used was created via a simulation, using GL theory to model the vortex interactions. PCA was then used to reduce the dimensions of the data, in order to make the training possible.

The resulting model successfully identified the solid states in the simulated data. Its accuracy was 1, perfectly predicting the labels of the training data set. Due to the labels assigned and the properties of the hexatic order, a different way of visualizing the model was employed. Namely, a plot was made of both the hexatic order and the model's prediction of each state simulated. This approach revealed that the model successfully identified the solid and melted states. However, it was noted that a subset of that data was used for training.

In order to explore this further, more data sets were produced with different procedures. The model failed to identify solid states in data from the same simulation as the training one when it was split into categories (solid and melted) before going through feature extraction, rather than after. The same happened when a new simulation was run, and its data pre-processed in the same manner as the training data. An explanation of this could be the PCA choosing a different basis for those data sets than for the training data.

Due to this, it would be useful to train a model on the data of one simulation, then run the simulation again. The data of the second run, projected on the same axis as the training data, could result in a data set the model would succeed in.

References

[1] Kurihana T, Foster I, Willett R, Jenkins S, Koenig K, Werman R, et al. CLOUD CLASSIFICATION WITH UNSUPERVISED DEEP LEARNING:5.

[2] Stryker S, Kapadia AJ, Greenberg JA. Application of machine learning classifiers to X-ray diffraction imaging with medically relevant phantoms. Medical Physics. 2022 Jan;49(1):532-46.

[3] Schmidt J, Marques MRG, Botti S, Marques MAL. Recent advances and applications of machine learning in solid-state materials science. npj Computational Materials. 2019 Aug;5(1):1-36. Number: 1 Publisher: Nature Publishing Group. Available from: https://www.nature.com/articles/s41524-019-0221-0.

[4] Stanev V, Oses C, Kusne AG, Rodriguez E, Paglione J, Curtarolo S, et al. Machine learning modeling of superconducting critical temperature. npj Computational Materials. 2018 Jun;4(1):1-14. Number: 1 Publisher: Nature Publishing Group. Available from: https://www.nature.com/articles/s41524-018-0085-8.

[5] Konno T, Kurokawa H, Nabeshima F, Sakishita Y, Ogawa R, Hosako I, et al. Deep learning model for finding new superconductors. Physical Review B. 2021 Jan;103(1):014509. Publisher: American Physical Society. Available from: https://link.aps.org/doi/10.1103/PhysRevB.103.014509.

[6] Carrasquilla J, Melko RG. Machine learning phases of matter. Nature Physics. 2017 May;13(5):431-4. Number: 5 Publisher: Nature Publishing Group. Available from: https://www.nature.com/articles/nphys4035.

[7] Wetzel SJ, Scherzer M. Machine Learning of Explicit Order Parameters: From the Ising Model to SU(2) Lattice Gauge Theory. Physical Review B. 2017 Nov;96(18):184410. ArXiv: 1705.05582. Available from: http://arxiv.org/abs/1705.05582.

[8] Ponte P, Melko RG. Kernel methods for interpretable machine learning of order parameters. Physical Review B. 2017 Nov;96(20):205146. ArXiv: 1704.05848. Available from: http://arxiv.org/abs/1704.05848.

[9] van Nieuwenburg EPL, Liu YH, Huber SD. Learning phase transitions by confusion. Nature Physics. 2017 May;13(5):435-9. Number: 5 Publisher: Nature Publishing Group. Available from: https://www.nature.com/articles/nphys4037.

[10] Beach MJS, Golubeva A, Melko RG. Machine learning vortices at the Kosterlitz-Thouless transition. Physical Review B. 2018 Jan;97(4):045207. ArXiv: 1710.09842. Available from: http://arxiv.org/abs/1710.09842.

[11] Nelson DR, Halperin BI. Dislocation-mediated melting in two dimensions. Physical Review B. 1979 Mar;19(5):2457-84. Publisher: American Physical Society. Available from: https://link.aps.org/doi/10.1103/PhysRevB.19.2457.

[12] Guillamón I, Suderow H, Fernández-Pacheco A, Sesé J, Córdoba R, De Teresa JM, et al. Direct observation of melting in a two-dimensional superconducting vortex lattice. Nature Physics. 2009 Sep;5(9):651-5. Number: 9 Publisher: Nature Publishing Group. Available from: https://www.nature.com/articles/nphys1368.

[13] Tinkham M. Introduction to superconductivity / Michael Tinkham. 2nd ed. Mineola, N.Y.: Dover Publications; 2004. Book Title: Introduction to superconductivity / Michael Tinkham.

[14] Poole CP. Superconductivity / Charles P. Poole, Jr., Ruslan Prozorov, Horacio A. Farach, Richard J. Creswick. 3rd ed. Elsevier insights. Amsterdam: Elsevier Science; 2014. Book Title: Superconductivity / Charles P. Poole, Jr., Ruslan Prozorov, Horacio A. Farach, Richard J. Creswick. Available from: https://www-sciencedirect-com.ezproxye.bham.ac.uk/book/9780124095090/superconductivity.

[15] SHUBNIKOV LV, KHOTKEVICH VI, SHEPELEV YD, RYABININ YN. MAGNETIC PROPERTIES OF SUPERCONDUCTING METALS AND ALLOYS. Zh Eksper Teor Fiz. 1937;7:221-37.

[16] Abrikosov A. On the Magnetic Properties of Superconductors of the Second Group. Soviet Physics JETP. 1957 Dec;5.

[17] Kleiner WH, Roth LM, Autler SH. Bulk Solution of Ginzburg-Landau Equations for Type II Superconductors: Upper Critical Field Region. Physical Review. 1964 Mar;133(5A):A1226-7. Available from: https://link.aps.org/doi/10.1103/PhysRev.133.A1226.

[18] Essmann U, Träuble H. The direct observation of individual flux lines in type II superconductors. Physics Letters A. 1967 May;24(10):526-7. Available from: https://www.sciencedirect.com/science/article/pii/0375960167908195.

[19] Kleiner R, Buckel W, Huebener R. Superconductivity: An Introduction. Weinheim, GERMANY: John Wiley & Sons, Incorporated; 2016. Available from: http://ebookcentral.proquest.com/lib/bham/detail.action?docID=4044603.

[20] Kosterlit JM. Ordering, metastability and phase transitions in two-dimensional systems:24.

[21] Gartlan JA. Novel excitations in driven vortex channels in a superconductor, and solitary waves of light and atoms in photonic crystal fibres [PhD]. University of Birmingham; 2020. Available from: https://etheses.bham.ac.uk/id/eprint/9918/.

[22] Spencer S, Jensen HJ. Absence of translational ordering in driven vortex lattices. Physical Review B. 1997 Apr;55(13):8473-81. Publisher: American Physical Society. Available from: https://link.aps.org/doi/10.1103/PhysRevB.55.8473.

[23] Halperin BI. On the Hohenberg-Mermin-Wagner theorem and its limitations. arXiv:181200220 [cond-mat]. 2018 Dec. ArXiv: 1812.00220. Available from: http://arxiv.org/abs/1812.00220.

[24] Rice TM. Superconductivity in One and Two Dimensions. Physical Review. 1965 Dec;140(6A):A1889-91. Publisher: American Physical Society. Available from: https://link.aps.org/doi/10.1103/PhysRev.140.A1889.

[25] Boots B, Sugihara K, Chiu SN, Okabe A. Spatial Tessellations: Concepts and Applications of Voronoi Diagrams. Hoboken, UNITED KINGDOM: John Wiley & Sons, Incorporated; 2000. Available from: http://ebookcentral.proquest.com/lib/bham/detail.action?docID=470072.

[26] Jolliffe IT, Cadima J. Principal component analysis: a review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. 2016 Apr;374(2065):20150202. Publisher: Royal Society. Available from: https://royalsocietypublishing.org/doi/10.1098/rsta.2015.0202.

[27] Knox SW. Machine Learning: A Concise Introduction. Newark, UNITED STATES: John Wiley & Sons, Incorporated; 2018. Available from: http://ebookcentral.proquest.com/lib/bham/detail.action?docID=5323676.

[28] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 2011;12(85):2825-30. Available from: http://jmlr.org/papers/v12/pedregosa11a.html.