Jurnal Materi dan Pembelajaran Fisika
Volume 14, Number 2, pp. 81-87 2024
Website : https://jurnal.uns.ac.id/jmpf

A Study of the Results of Time-Dependent Ginzburg-Landau Equation Using Scilab

Tanzillal Alfargana, Artono Dwijo Sutomo, Fuad Anwar*
Prodi Fisika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sebelas Maret
Jl Ir. Sutami N0. 36A, Jebres, Surakarta, Jawa Tengah, 57126, Indonesia
**Corresponding author: fanwar@staff.uns.ac.id*

Article's Info
Received: 17th, August, 2024
Accepted: 20th, August, 2024
Published: 26th, November, 2024

DOI:
https://doi.org/10.20961/jmpf.v14i2.92340

How to Cite : Alfargana, T., Sutomo, A.D., Anwar, F. (2024). A Study of the Results of Time-Dependent Ginzburg-Landau Equation Using Scilab. Jurnal Materi dan Pembelajaran Fisika, volume 14 number 2, 81-87

Abstract. This study aims to examine computational (numerical) programs in solving the Ginzburg-Landau Time-Dependent Equation. The Ginzburg-Landau Time-Dependent Equation (TDGL), which is often used to model the dynamics of superconductors in external magnetic fields. The TDGL equation is made using the Scilab platform, then the results are compared with the results of Fortran. This study involves experiments with variations in superconductor size to observe their effect on simulation results. The main focus of this study is to study the relationship between the mean magnetization of superconductors and external magnetic fields (He), as well as to evaluate the accuracy and efficiency of the results obtained from the two computing platforms. These results show the relationship between the mean of the superconductor magnetization and the external magnetic field, as well as a comparison of the Vs He graph of Scilab and Fortran showing the relationship between the superconductor order parameters and the external magnetic field.

Keywords: External magnetic fields; Fortran; Numerical simulation; Scilab; Superconductors; Time-Dependent Ginzburg-Landau equations; U method

INTRODUCTION

In research on superconductor materials, one of the main focus is to study the magnetic properties they produce. It is known that the magnetic properties of superconductors are perfectly diamagnetic, which is very useful to be applied in several technology (Mourachkine, 2004).

Based on the critical field, superconductor are divided into two types: Type I and II. Type I has only one critical field, H_c, while Type II has two critical fields, that is H_{c1} (lower critical field) and H_{c2} (higher critical field). Thin superconductor or known as mesoscopic superconductor has one more level of critical field, H_{c3} (surface critical field). According to Tinkham (1996) and Mourachkine (2004) these critical fields describe the state of superconductors where when the field value is below H_{c1} then the phenomenon of perfect superconductivity occurs, if the field between H_{c1} and H_{c2} then a mixed state is formed, and if the field between H_{c2} and H_{c3} then there is a surface superconductivity, more than this value the material will returns as normal material.

One of the superconductivity theories that explains the magnetic property of superconductor is Ginzburg-Landau Theory. The Theory has two nonlinear and coupled differential equations known as the Ginzburg-Landau equation (Andryushin, 1994), and developed into Ginzburg-Landau Time Dependent Equation (Time Dependent Ginzburg-Landau Equation/TDGL Equation). The analytical

solution of these equation is difficult to be solved, so a numerical method is developed, one of which is using ψU method and produce TDGL computational program. TDGL program is widely used to study the properties of superconductor magnetism, here are some experts who use this program (Buscaglia *et al*., 2000), (Gropp *et al*., 1995), (Grishakov *et al*., 2012). Some latest research related to this program is also still being carried out, for example (Anwar, 2015), (Anwar *et al.*, 2019), (Anwary *et* al., 2019), (Barba-Ortega *et al.*, 2018), (Aguirre *et al*., 2020), (Hafiz *et al.*, 2019).

The TDGL programs studied are more often written using the Fortran programming language (platform) as done by (Buscaglia et al., 2000), (Anwar et al., 2019), (Anwary et al., 2019), and (Hafiz et al., 2019). However, on the other hand, the platform still needs license to be fully utilized and the price offered is not affordable for some people, besides that there are often compatibility problems between the platform and the operating system (OS) of the researchers so that it ends up not being able to be operated on their own devices. Therefore, the author looked for an alternative and found the Scilab platform that can be accessed for free (free of charge). Besides its free to use Scilab also offers other things, such as its mechanism similar to Matlab, and last but not least it is compatible with many OSes ranging from Linux to Windows 11. In addition to these things, the author also still has not found any research related to the TDGL program which uses Scilab in its entirety, so this Scilab platform was chosen. From these explanations, the author aims to study the TDGL program made with Scilab, then the results of this program will be compared with the results obtained if using Fortran.

METHOD

In general, the implementation of this research is illustrated by the following Figure 1.

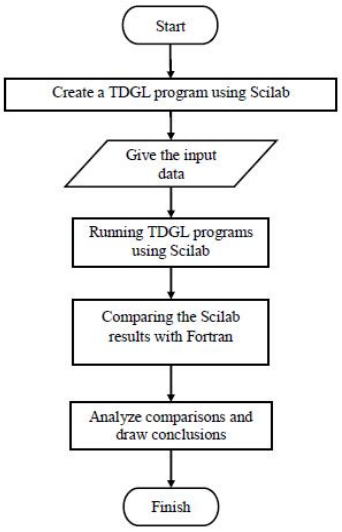

Figure 1. Flowchart of research

The creation of the TDGL program refers to the following journals, those are (Anwar *et al*., 2019), (Anwary *et al*., 2019), and (Hafiz *et al*., 2019) which was then created for the Fortran platform. In general, the program has many important equations for the calculation process, but some will be written here. Here are some equations quoted from (Anwar, 2015).

$$\psi_{i,j}(t+\Delta_t) = \psi_{i,j}(t) + \left(\frac{U^x_{i,j}(t)\psi_{i+1,j}(t) + 2\psi_{i,j}(t) + \underline{U^x_{i-1,j}}(t)\psi_{i-1,j}(t)}{(\Delta_x)^2}\right)\Delta_t + \left(\frac{U^y_{i,j}(t)\psi_{i,j+1}(t) + 2\psi_{i,j}(t) + \underline{U^y_{i,j-1}}(t)\psi_{i,j-1}(t)}{(\Delta_y)^2}\right)\Delta_t + \left(1 - \left|\psi_{i,j}(t)\right|^2\right)\psi_{i,j}(t)\Delta_t \quad [1]$$

$$U^x_{i,j}(t+\Delta_t)\hat{x} = U^x_{i,j}(t)\hat{x} - iU^x_{i,j}(t)Im\left(U^x_{i,j}(t)\underline{\psi_{i,j}}(t)\right)\Delta_t\hat{x} - \frac{\kappa^2}{(\Delta_y)^2}U^x_{i,j}(t)\left(L_{i,j}(t)\underline{L_{i,j-1}}(t) - 1\right)\Delta_t\hat{x} \quad [2]$$

$$U^y_{i,j}(t+\Delta_t)\hat{y} = U^y_{i,j}(t)\hat{y} - iU^y_{i,j}(t)Im\left(U^y_{i,j}(t)\underline{\psi_{i,j}}(t)\psi_{i,j+1}(t)\right)\Delta_t\hat{y} - \frac{\kappa^2}{(\Delta_x)^2}U^y_{i,j}(t)\left(L_{i,j}(t)\underline{L_{i-1,j}}(t) - 1\right)\Delta_t\hat{y} \quad [3]$$

Details:

$\psi_{i,j}(t) =$ Order Parameter
$U^x_{i,j}(t) =$ Link Variable of x-axis
$U^y_{i,j}(t) =$ Link Variable of y-axis
$\Delta_x =$ The length of the cell side relative to the x-axis
$\Delta_y =$ The length of the cell side relative to the y-axis
$\Delta_t =$ Time Step
$\kappa =$ Ginzburg-Landau Parameter
$L_{i,j}(t) = U^y_{i+1,j}(t)\underline{U^y_{i,j}}(t)\underline{U^x_{i,j+1}}(t)U^x_{i,j}(t)$

with $i, j = 1,2,3, \ldots$

The algorithm of the TDGL Program is illustrated in the following Figure 2.

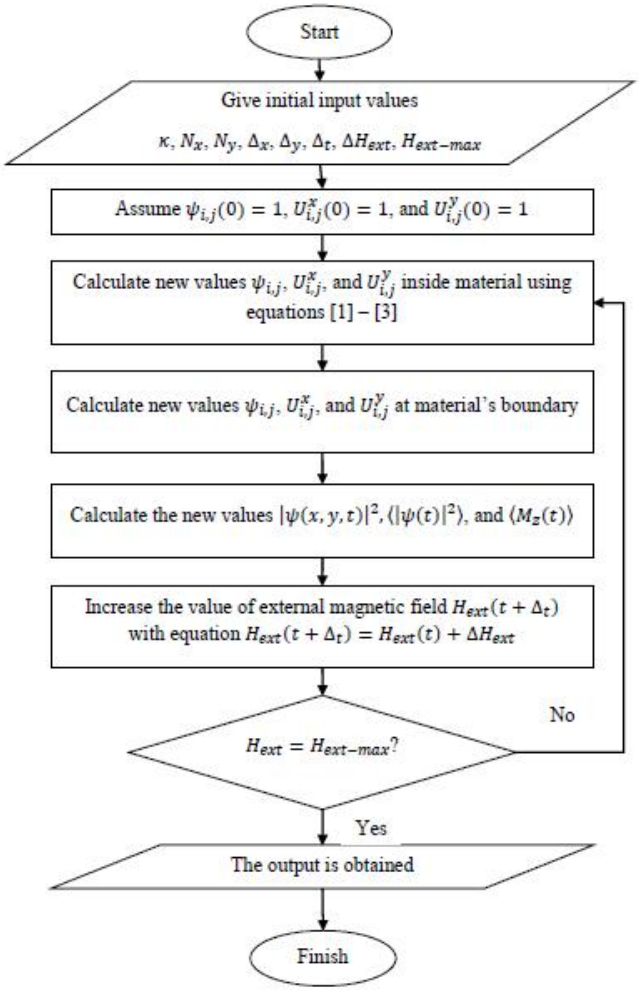

Figure 2. Flowchart of TDGL program.

The input data is divided into two types, namely constant input, and variation input as shown in the following tables.

Table 1. Input Data

Input Data	Value
κ	2
H_{ext} initial	0
Increment of H_{ext}	1×10^{-6}
$H_{ext-max}$	5
Δ_t	0,01

Table 2. Variation of superconductor's size

$N_x \times N_y$	Size
6 x 6	Small
8 x 8	Medium
12 x 12	Large

RESULT AND DISCUSSION

From the research that has been conducted, the TDGL Program has been successfully created using the Scilab platform. Furthermore, this program is executed using inputs as written in Tables 1 and 2. From the results of running the program, an output was obtained in the form of a graph of the

modulus value of the average squared fix parameter $\langle|\psi|^2\rangle$ as a function of the external magnetic field H_{ext}, and the average Magnetization $\langle M\rangle$ as a function of H_{ext}, or written as $\langle|\psi|^2\rangle$ vs He and $\langle M\rangle$ vs He. The output of both types of graphs, both from Scilab and Fortran, is shown in Figures 3 and 4. From the two figures, the values of H_{c1} and H_{c3} are searched based on the first local minimum point and the zero point (Anwar, 2015), the results of which are written in Table 3.

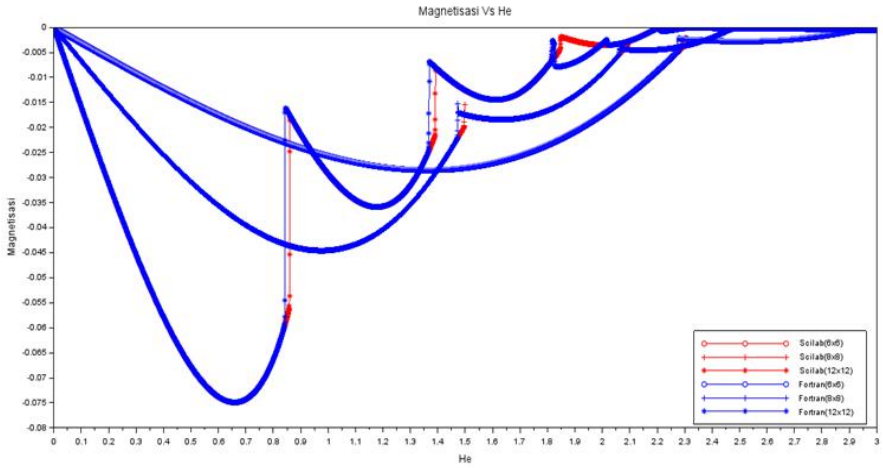

Figure 3. Graph of $\langle M\rangle$ vs He on superconductors 6×6, 8×8, and 12×12.

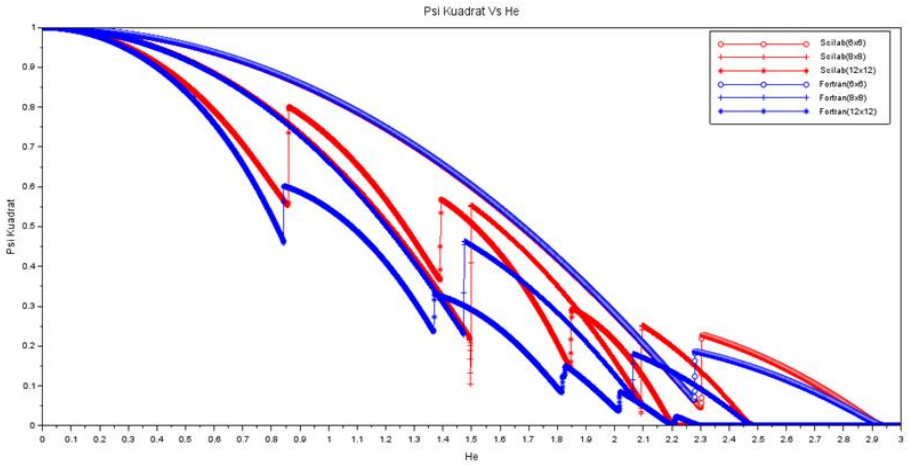

Figure 4. Graph of $\langle|\psi|^2\rangle$ vs He at superconductors 6×6, 8×8, and 12×12

Table 3. List of H_{c1} and H_{c3} values on each platform

Size of Superconductor	Scilab				Fortran			
	$\langle M\rangle$ vs He		$\langle\|\psi'\|^2\rangle$ vs He		$\langle M\rangle$ vs He		$\langle\|\psi'\|^2\rangle$ vs He	
	H_{c1}	H_{c3}	H_{c1}	H_{c3}	H_{c1}	H_{c3}	H_{c1}	H_{c3}
6×6	2,304	2,944	2,306	2,948	2,280	2,937	2,282	2,947
8×8	1,497	2,444	1,498	2,485	1,473	2,444	1,475	2,483
12×12	0,861	2,170	0,861	2,211	0,845	2,170	0,845	2,297

From figures 3 and 4, for the 6×6 superconductor, it can be seen that the curve plots of the Scilab and Fortran versions have the same pattern and mostly overlap each other. This shows that the output results of the Scilab and Fortran versions of the program are the same. For all three types of superconductor sizes, the curve plots of the Scilab and Fortran versions tend to have the same pattern, but the larger the size of the superconductor, the greater the difference between the values of $\langle|\psi|^2\rangle$ and $\langle M\rangle$, especially for large He values. From Table 3, it can be seen that there is a difference in the values of H_{c1} and H_{c3} as a result of the Scilab and Fortran version of the program.

When talking about platform comparisons, the reason why the results obtained can be diverse (such as some are different and some are also the same), the author does not know the exact cause, but the author suspects that this possibility is related to the accuracy of Scilab and Fortran's calculations. This assumption is based on the fact that in size 12×12 in the graph $\langle|\psi|^2\rangle$ vs He when approaching H_{c3} the Fortran curve still does not show a zero value on its y-axis and forms a kind of hill, whereas in Scilab it shows a zero value and looks sloping. Judging from the curve patterns produced by Scilab and Fortran look very similar or even same, though there may be several different points. This can also be a basis that may indeed be a problem in precision of decimal numbers after the comma, because if the results are significantly different of course the resulting curve patterns are very different. In addition, it is also known that the larger the superconductor's size, the more visible the difference in graph between Scilab and Fortran even though the patterns remain the same. This can be supporting evidence that there is indeed a difference in accuracy between two platforms, because with different precision while the number of superconducting cells increase as the size increase, it tends to make rounding of the different results.

For the difference in the values of H_{c1} and H_{c3} between $\langle M\rangle$ vs He and $\langle|\psi|^2\rangle$ vs He, this can be caused by the calculation process that $\langle M\rangle$ vs He goes through is longer so that there may be an error in rounding the number.

Regarding the difference in precision between Scilab and Fortran, the author gets little information. Based on Scilab help version 6.1.1 (2021) Scilab by default uses double precision which means it has calculation precision of up to 16 decimal places after the comma, while Fortran in general has at least 17 decimal digits after the comma (Fortran Community, 2020), but in particular the author still has not found a library that discusses the level of accuracy in more detail so that it cannot be concluded with certainty whether there is really a difference in precision in the calculation between Scilab and Fortran.

CONCLUSION

The conclusion that can be drawn from this study is that there is a difference in curve values between the Scilab and Fortran versions of the TDGL program. The value of the curve is the one that forms the curve as a whole and the value that provides information about the superconductor Hc1 and Hc3 simulated in the TDGL program. The difference in these values can be categorized into two, namely in terms of platform and calculation method. In terms of platforms, this is due to the difference in accuracy or writing decimal numbers after the comma in each platform, while in terms of calculation methods, there are calculation methods whose process is longer than other methods so that they are susceptible to errors in rounding the final results.

REFERENCES

Aguirre, C. A., Sardella, E., & Barba-Ortega, J. (2020). Length dependence of the number of phase slip lines in a superconducting strip. *Solid State Communications*, *306*, 113799.

Andryushin, E. A., & Ginzburg, V. L. (1994). *Superconductivity*. World Scientific Publishing Company.

Anwar, F., Cari, C., & Ilmi, A. M. (2019). Comparison of the superconducting rectangular critical fields with the same area at κ= 1.5. *Journal of Physics: Theories and Applications*, *3*(2), 62-68.

Anwar, F. (2015). Kajian Model: Pengaruh Efek Proksimitas dan Efek Tak Isotrop pada Sifat Magnet Superkonduktor Mesoskopik Tipe II. *Disertasi*, S3 Ilmu Fisika, Universitas Gajah Mada Yogyakarta.

Anwary, R, Ma'arif, M., Anwar, F., & Purwanto, H. (2019). Numerical Study about The Effect of The Size of Rectangular Type-II Superconductor on The H_{c1} and H_{c3} Values in The Case of Proximity Effect. *International Conference on Physics and Its Applications*, 1153, 1-5.

Barba-Ortega, J., Sardella, E., & Zadorosny, R. (2018). Influence of the deGennes extrapolation parameter on the resistive state of a superconducting strip. *Physics Letters A*, *382*(4), 215-219.

Barba-Ortega, J., Achic, H.B., & Joya, M.R. (2019). Resistive state of a superconducting thin film with rough surface. *Physica C: Superconductivity and its applications*, *561*, 9-12.

Fortran Community. (2020). Floating Point Numbers. Retrieved from Floating Point Numbers — Fortran Programming Language (fortran-lang.org) .

Gropp, W. D., Kaper, H. G., Leaf, G. K., Levine, D. M., Palumbo, M., & Vinokur, V. M. (1996). Numerical simulation of vortex dynamics in type-II superconductors. *Journal of Computational Physics*, *123*(2), 254-266.

Hafiz, R., Ma'arif, M., Anwar, F., & Purwanto, H. (2019, February). Numerical study of dimension effect on critical field of rectangular superconductor when The Ginzburg-Landau parameter is 3. In *Journal of Physics: Conference Series* (Vol. 1153, No. 1, p. 012114). IOP Publishing.

Mourachkine, A. (2004). *Room-Temperature Superconductivity*. Cambridge: Cambridge International Science Publishing.

Scilab help. (2021). %eps-Epsilon (floating-point relative accuracy). Retrieved from %eps - Epsilon (floating-point relative accuracy) (scilab.org) .

Tinkham, M. (1996). *Introduction to Superconductivity*. Singapore: McGraw-Hill Inc.