mathematics

Article

An Approximate Proximal Numerical Procedure Concerning the Generalized Method of Lines

Fabio Silva Botelho

Department of Mathematics, Federal University of Santa Catarina, UFSC, Florianópolis 88040-900, SC, Brazil; fabio.botelho@ufsc.br

Abstract: This article develops an approximate proximal approach for the generalized method of lines. We recall that for the generalized method of lines, the domain of the partial differential equation in question is discretized in lines (or in curves) and the concerning solution is developed on these lines, as functions of the boundary conditions and the domain boundary shape. Considering such a context, in the text we develop an approximate numerical procedure of proximal nature applicable to a large class of models in physics and engineering. Finally, in the last sections, we present numerical examples and results related to a Ginzburg–Landau-type equation.

Keywords: generalized method of lines; approximate proximal approach; Ginzburg-Landau type equations

MSC: 65N40; 65N06

check for updates

Citation: Botelho, F.S. An Approximate Proximal Numerical Procedure Concerning the Generalized Method of Lines. *Mathematics* **2022**, *10*, 2950. https://doi.org/10.3390/math10162950

Academic Editors: Theodore E. Simos and Charampos Tsitouras

Received: 7 July 2022
Accepted: 14 August 2022
Published: 16 August 2022

1. Introduction

We start by recalling that the generalized method of lines was originally introduced in the book entitled "Topics on Functional Analysis, Calculus of Variations and Duality" [1], published in 2011.

Indeed, the present results are extensions and applications of previous ones, which have been published since 2011, in books and articles such as [1–4]. About the Sobolev spaces involved, we would mention [5,6]. Concerning the applications, related models in physics are addressed in [7,8].

We also emphasize that, in such a method, the domain of the partial differential equation in question is discretized in lines (or more generally, in curves), and the concerning solution is written on these lines as functions of boundary conditions and the domain boundary shape.

In fact, in its previous format, this method consists of an application of a kind of a partial finite differences procedure combined with the Banach fixed point theorem to obtain the relation between two adjacent lines (or curves).

In the present article, we propose an approximate approach and a related iterative procedure of proximal nature. We highlight that this is a proximal method inspired by some models concerning duality principles in D.C optimization in the calculus of variations, such as those found in Toland [9].

In the next lines and sections we develop in detail such a numerical procedure.

With such statements in mind, let $\Omega \subset \mathbb{R}^2$ be an open, bounded and connected set where:

$$\Omega = \{(x,y) \in \mathbb{R}^2 \ : \ y_1(x) \leq y \leq y_2(x), \ a \leq x \leq b\}.$$

Here, we assume, $y_1, y_2 : [a,b] \to \mathbb{R}$ are continuous functions.

Consider the Ginzburg–Landau-type equation, defined by:

$$\begin{cases} -\varepsilon\nabla^2 u + \alpha u^3 - \beta u = f, & \text{in } \Omega, \\ u = 0, & \text{on } \partial\Omega, \end{cases} \tag{1}$$

where $\varepsilon > 0, \alpha > 0, \; \beta > 0$ and $f \in L^2(\Omega)$.

Furthermore, $u \in W_0^{1,2}(\Omega)$, and the equation in question must be considered in a distributional sense.

In the next section, we address the problem of approximately solving this concerning equation. We highlight that the methods and ideas exposed are applicable to a large class of similar models in physics and engineering.

2. The Numerical Method

We discretize the interval $[a,b]$ into N same measure sub-intervals, through a partition

$$P = \{x_0 = a, x_1, \cdots, x_N = b\},$$

where $x_n = a + nd, \; \forall n \in \{1, \cdots, N-1\}$. Here,

$$d = \frac{(b-a)}{N}.$$

Through such a procedure, we generate N vertical lines parallel to the Cartesian axis $0y$, so that for each line n, based on the point x_n, we are going to compute an approximate solution $u_n(y)$ corresponding to values of u on such a line.

Considering this procedure, the equation system obtained in partial finite differences (please see [10], for concerning models in finite differences) is given by:

$$-\varepsilon\left(\frac{u_{n+1} - 2u_n + u_{n-1}}{d^2} + \frac{\partial^2 u_n}{\partial y^2}\right) + \alpha u_n^3 - \beta u_n = f_n,$$

$\forall n \in \{1, \cdots, N-1\}$, with the boundary conditions:

$$u_0 = 0,$$

and,

$$u_N = 0.$$

Let $K > 0$ be an appropriate constant to be specified.

In a proximal approach, considering an initial solution:

$$\{(u_0)_n\},$$

we redefine the system of equations in question as below indicated.

$$-\varepsilon\left(\frac{u_{n+1} - 2u_n + u_{n-1}}{d^2} + \frac{\partial^2 u_n}{\partial y^2}\right) + \alpha u_n^3 - \beta u_n + K u_n - K(u_0)_n = f_n,$$

$\forall n \in \{1, \cdots, N-1\}$, with the boundary conditions:

$$u_0 = 0,$$

and,

$$u_N = 0.$$

Hence, we may denote:

$$u_{n+1} - \left(2 + K\frac{d^2}{\varepsilon}\right)u_n + u_{n-1} + T(u_n) + \tilde{f}_n \frac{d^2}{\varepsilon} = 0,$$

where,

$$T(u_n) = \left(-\alpha u_n^3 + \beta u_n\right)\frac{d^2}{\varepsilon} + \frac{\partial^2 u_n}{\partial y^2} d^2,$$

and $\tilde{f}_n = K(u_0)_n + f_n, \forall n \in \{1, \cdots, N-1\}$.

In particular, for $n = 1$, we obtain:

$$u_2 - \left(2 + K\frac{d^2}{\varepsilon}\right)u_1 + T(u_1) + \tilde{f}_1\frac{d^2}{\varepsilon} = 0,$$

so that,

$$u_1 = a_1 u_2 + b_1 T(u_2) + c_1 + E_1,$$

where,

$$a_1 = \frac{1}{2 + K\frac{d^2}{\varepsilon}},$$

$$b_1 = a_1,$$

and,

$$c_1 = a_1 \tilde{f}_1 \frac{d^2}{\varepsilon}$$

and the error E_1, proportional to $1/K$, is given by:

$$E_1 = b_1(T(u_1) - T(u_2)).$$

Now, reasoning inductively, having,

$$u_{n-1} = a_{n-1}u_n + b_{n-1}T(u_n) + c_{n-1} + E_{n-1}$$

for the line n, we have,

$$\begin{aligned} & u_{n+1} - \left(2 + K\frac{d^2}{\varepsilon}\right)u_n + a_{n-1}u_n + b_{n-1}T(u_n) + c_{n-1} + E_{n-1} \\ & + T(u_n) + \tilde{f}_n\frac{d^2}{\varepsilon} = 0, \end{aligned} \tag{2}$$

so that,

$$u_n = a_n u_{n+1} + b_n T(u_{n+1}) + c_n + E_n,$$

where,

$$a_n = \frac{1}{2 + K\frac{d^2}{\varepsilon} - a_{n-1}},$$

$$b_n = a_n(b_{n-1} + 1),$$

and,

$$c_n = a_n\left(c_{n-1} + \tilde{f}_n\frac{d^2}{\varepsilon}\right)$$

and the error E_n, is given by:

$$E_n = a_n E_{n-1} + b_n(T(u_n) - T(u_{n+1})),$$

$\forall n \in \{1, \cdots, N-1\}$.

In particular, for $n = N-1$, we have $u_N = 0$ so that,

$$\begin{aligned} u_{N-1} &\approx a_{N-1}u_N + b_{N-1}T(u_N) + c_{N-1} \\ &\approx a_{N-1}u_N + b_{N-1}\frac{\partial^2 u_{N-1}}{\partial y^2}d^2 + b_{N-1}(-\alpha u_N^3 + \beta u_N)\frac{d^2}{\varepsilon} + c_{N-1} \\ &= b_{N-1}\frac{\partial^2 u_{N-1}}{\partial y^2}d^2 + c_{N-1}. \end{aligned} \tag{3}$$

This last equation is an ODE from which we may easily obtain u_{N-1} with the boundary conditions

$$u_{N-1}(y_1(x_{N-1})) = u_{N-1}(y_2(x_{N-1})) = 0.$$

Having u_{N-1}, we may obtain u_{N-2} though the equation:

$$\begin{aligned} u_{N-2} &\approx a_{N-2}u_{N-1} + b_{N-2}T(u_{N-1}) + c_{N-2} \\ &\approx a_{N-2}u_{N-1} + b_{N-2}\frac{\partial^2 u_{N-2}}{\partial y^2}d^2 + b_{N-2}(-\alpha u_{N-1}^3 + \beta u_{N-1})\frac{d^2}{\varepsilon} + c_{N-2}, \end{aligned} \tag{4}$$

with the boundary conditions:

$$u_{N-2}(y_1(x_{N-2})) = u_{N-2}(y_2(x_{N-2})) = 0.$$

And so on, up to finding u_1.

The next step is to replace $\{(u_0)_n\}$ by $\{u_n\}$ and then to repeat the process until an appropriate convergence criterion is satisfied.

The problem is then approximately solved.

3. A Numerical Example

We present numerical results for $\Omega = [0,1] \times [0,1]$, $\alpha = \beta = 1$, $f \equiv 1$ in Ω, $N = 100$, $K = 50$ and for:

$$\varepsilon = 0.1,\ 0.01 \text{ and } 0.001.$$

For such values of ε, please see Figures 1–3, respectively.

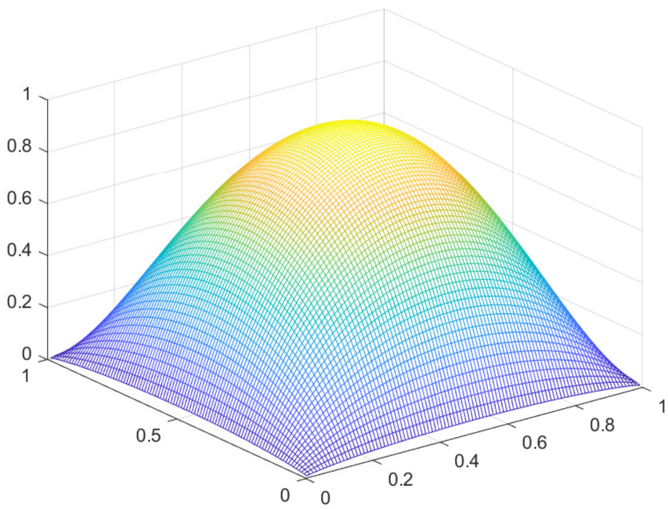

Figure 1. Solution u for $\varepsilon = 0.1$.

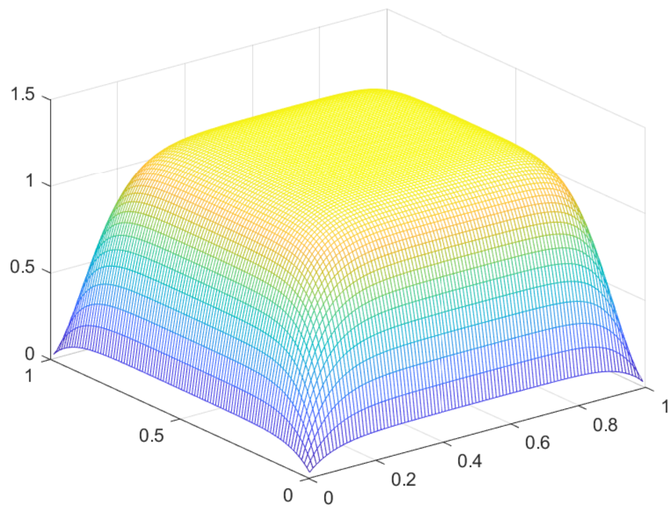

Figure 2. Solution u for $\varepsilon = 0.01$

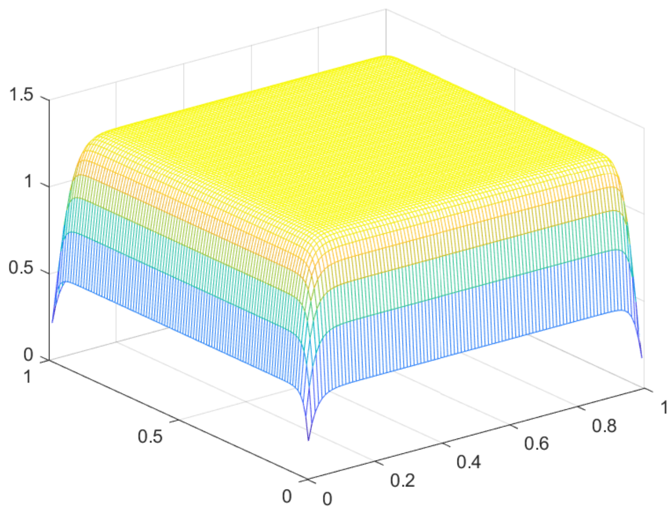

Figure 3. Solution u for $\varepsilon = 0.001$

Remark 1. *We observe that as* $\varepsilon > 0$ *decreases to the value* 0.001*, the solution approaches the constant value* 1.3247 *along the domain, up to the satisfaction of boundary conditions. This is expected, since this value is an approximate solution of equation* $u^3 - u - 1 = 0$.

4. A General Proximal Explicit Approach

Based on the algorithm presented in the last section, we develop a software in MATHEMATICA in order to approximately solve the following equation.

$$\begin{cases} -\varepsilon\left(\frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2}\right) + \alpha u^3 - \beta u = f, & \text{in } \Omega, \\ u = 0, & \text{on } \partial\Omega_1, \\ u = u_f(\theta), & \text{on } \partial\Gamma_2. \end{cases} \tag{5}$$

Here:

$$\Omega = \{(r,\theta) \in \mathbb{R}^2 \; : \; 1 \leq r \leq 2, \; 0 \leq \theta \leq 2\pi\},$$

$$\partial\Omega_1 = \{(1,\theta) \in \mathbb{R}^2 \; : \; 0 \leq \theta \leq 2\pi\},$$

$$\partial\Omega_2 = \{(2, \theta) \in \mathbb{R}^2 \; : \; 0 \leq \theta \leq 2\pi\},$$

$\alpha = \beta = 1$, $K = 10$, $N = 100$ and $f \equiv 1$, on Ω.

At this point we present such a software in MATHEMATICA.

```
****************************************
1.  m8 = 100;
2.  d = 1.0/m8;
3.  K = 10.0;
4.  e1 = 0.01;
5.  A = 1.0;
6.  B = 1.0;
7.  For[i = 1, i < m8 + 1, i++,
    uo[i] = 0.0];
8.  For[k = 1, k < 150, k++,
    Print[k];
    a[1] = 1/(2.0 + K * d^2/e1);
    b[1] = a[1];
    c[1] = a[1]*(K*uo[1] + 1.0)*d^2/e1;
    For[i = 2, i < m8, i++,
    a[i] = 1/(2.0 + K * d^2/e1 - a[i - 1]);
    b[i] = a[i]*(b[i - 1] + 1);
    c[i] = a[i]*(c[i - 1] + (K * uo[i] + 1.0) * d^2/e1)];
9.  u[m8] = uf[x]; d1 = 1.0;
10. For[i = 1, i < m8, i++,
    t[m8 - i] = 1 + (m8 - i)*d;
    A1 = (a[m8 - i]*u[m8 - i + 1] +
    b[m8 - i]*(-A*u[m8 - i + 1]^3 + B*u[m8 - i + 1])*d^2/e1 * d1^2 +
    c[m8 - i] +
    d^2 * d1^2*b[m8 - i]*(D[u[m8 - i + 1], x, 2]/t[m8 - i]^2) +
    d1^2 * 1/t[m8 - i]*b[m8 - i]* d^2 (uo[m8 - i + 1] - uo[m8 - i])/d)/(1.0);
    A1 = Expand[A1];
    A1 = Series[ A1, {uf[x], 0, 3}, {uf'[x], 0, 1}, {uf''[x], 0, 1}, {uf'''[x], 0, 0}, {uf''''[x], 0, 0}];
    A1 = Normal[A1];
    u[m8 - i] = Expand[A1]];
    For[i = 1, i < m8 + 1, i++,
    uo[i] = u[i]]; d1 = 1.0;
    Print[Expand[u[m8/2]]]]
**********************************
```

For such a general approach, for $\varepsilon = 0.1$, we have obtained the following lines (here x stands for θ).

$$\begin{aligned} u[10](x) &= 0.4780 + 0.0122\, u_f[x] - 0.0115\, u_f[x]^2 + 0.0083\, u_f[x]^3 + 0.00069 u_f''[x] \\ &\quad -0.0014\, u_f[x](u_f'')[x] + 0.0016\, uf[x]^2 u_f''[x] - 0.00092\, u_f[x]^3 u_f''[x] \end{aligned} \tag{6}$$

$$\begin{aligned} u[20](x) &= 0.7919 + 0.0241\, u_f[x] - 0.0225\, u_f[x]^2 + 0.0163\, u_f[x]^3 + 0.0012\, u_f''[x] \\ &\quad -0.0025\, u_f[x](u_f'')[x] + 0.0030\, u_f[x]^2(u_f'')[x] - 0.0018\, u_f[x]^3(u_f'')[x]. \end{aligned} \tag{7}$$

$$\begin{aligned} u[30](x) &= 0.9823 + 0.0404\, u_f[x] - 0.0375\, u_f[x]^2 + 0.0266\, u_f[x]^3 + 0.00180\, (u_f'')[x] \\ &\quad -0.00362\, u_f[x](u_f'')[x] + 0.0043\, u_f[x]^2(uf'')[x] - 0.0028\, u_f[x]^3(uf'')[x] \end{aligned} \tag{8}$$

$$\begin{aligned} u[40](x) &= 1.0888 + 0.0698\, u_f[x] - 0.0632\, u_f[x]^2 + 0.0433\, u_f[x]^3 + 0.0026(u_f'')[x] \\ &\quad -0.0051\, u_f[x](u_f'')[x] + 0.0061 u_f[x]^2(u_f'')[x] - 0.0043 u_f[x]^3(u_f'')[x] \end{aligned} \tag{9}$$

$$\begin{aligned} u[50](x) &= 1.1316 + 0.1277\, u_f[x] - 0.1101\, u_f[x]^2 + 0.0695\, u_f[x]^3 + 0.0037\, (u_f'')[x] \\ &\quad -0.0073\, u_f[x](u_f'')[x] + 0.0084\, u_f[x]^2(u_f'')[x] - 0.0062\, u_f[x]^3(u_f'')[x] \end{aligned} \tag{10}$$

$$\begin{aligned} u[60](x) &= 1.1104 + 0.2389\, u_f[x] - 0.1866\, u_f[x]^2 + 0.0988\, u_f[x]^3 + 0.0053(u_f'')[x] \\ &\quad -0.0099\, u_f[x](u_f'')[x] + 0.0105\, u_f[x]^2(u_f'')[x] - 0.0075\, u_f[x]^3(u_f''])[x] \end{aligned} \tag{11}$$

$$\begin{aligned} u[70](x) &= 1.0050 + 0.4298\, u_f[x] - 0.273813\, u_f[x]^2 + 0.0949\, u_f[x]^3 + 0.0070\, (u_f'')[x] \\ &\quad -0.0116\, u_f[x](u_f'')[x] + 0.0102\, u_f[x]^2(u_f'')[x] - 0.0061\, u_f[x]^3(u_f'')[x] \end{aligned} \tag{12}$$

$$\begin{aligned} u[80](x) &= 0.7838 + 0.6855\, u_f[x] - 0.2892\, u_f[x]^2 + 0.0161\, u_f[x]^3 + 0.0075\, (u_f'')[x] \\ &\quad -0.0098 u_f[x](u_f'')[x] + 0.0063084 uf[x]^2(uf'')[x] - 0.0027 u_f[x]^3(u_f'')[x] \end{aligned} \tag{13}$$

$$\begin{aligned} u[90](x) &= 0.4359 + 0.9077\, u_f[x] - 0.1621 u_f[x]^2 - 0.0563\, u_f[x]^3 + 0.0051\, (u_f'')[x] \\ &\quad -0.0047\, u_f[x](u_f'')[x] + 0.0023\, u_f[x]^2(u_f'')[x] - 0.00098\, u_f[x]^3(u_f'')[x] \end{aligned} \tag{14}$$

For $\varepsilon = 0.01$, we have obtained the following line expressions.

$$\begin{aligned} u[10](x) &= 1.0057 + 2.07 * 10^{-11}\, u_f[x] - 1.85 * 10^{-11}\, u_f[x]^2 + 1.13 * 10^{-11}\, u_f[x]^3 \\ &\quad +4.70 * 10^{-13}\, (u_f'')[x] - 8.44 * 10^{-13} u_f[x](u_f'')[x] \\ &\quad +7.85 * 10^{-13}\, u_f[x]^2(u_f'')[x] - 6.96 * 10^{-14}\, u_f[x]^3(u_f'')[x] \end{aligned} \tag{15}$$

$$\begin{aligned} u[20](x) &= 1.2512 + 2.13 * 10^{-10}\, u_f[x] - 1.90 * 10^{-10}\, u_f[x]^2 + 1.16 * 10^{-10}\, u_f[x]^3 \\ &\quad +3.94 * 10^{-12}\, (u_f'')[x] - 7.09 * 10^{-12} u_f[x](uf'')[x] + 6.61 * 10^{-12} \\ &\quad u_f[x]^2(u_f'')[x] - 7.17 * 10^{-13}\, u_f[x]^3(u_f'')[x] \end{aligned} \tag{16}$$

$$\begin{aligned} u[30](x) &= 1.3078 + 3.80 * 10^{-9} uf[x] - 3.39 * 10^{-9}\, u_f[x]^2 + 2.07 * 10^{-9}\, u_f[x]^3 \\ &\quad +5.65 * 10^{-11}(u_f'')[x] - 1.018 * 10^{-10}\, u_f[x](u_f'')[x] \\ &\quad +9.52 * 10^{-11}\, u_f[x]^2(u_f'')[x] - 1.27 * 10^{-11}\, uf[x]^3(u_f'')[x] \end{aligned} \tag{17}$$

$$\begin{aligned} u[40](x) &= 1.3208 + 7.82 * 10^{-8} u_f[x] - 6.98 * 10^{-8}\, u_f[x]^2 + 4.27 * 10^{-8}\, u_f[x]^3 \\ &\quad +9.27 * 10^{-10}\, (u_f'')[x] - 1.67 * 10^{-9} u_f[x](u_f'')[x] \\ &\quad +1.57 * 10^{-}9\, u_f[x]^2(u_f'')[x] - 2.62 * 10^{-10}\, u_f[x]^3(u_f'')[x] \end{aligned} \tag{18}$$

$$
\begin{aligned}
u[50](x) \quad = \quad & 1.3238 + 1.67 * 10^{-6}\ u_f[x] - 1.49 * 10^{-6}\ u_f[x]^2 + 9.15 * 10^{-7}\ uf[x]^3 \\
& + 1.54 * 10^{-8}\ (u_f'')[x] - 2.79 * 10^{-8}\ u_f[x](u_f'')[x] \\
& + 2.64 * 10^{-8}\ u_f[x]^2(u_f'')[x] - 5.62 * 10^{-9}\ u_f[x]^3(u_f'')[x]
\end{aligned} \tag{19}
$$

$$
\begin{aligned}
u[60](x) \quad = \quad & 1.32449 + 0.000036\ u_f[x] - 0.000032\ u_f[x]^2 + 0.000019\ u_f[x]^3 \\
& + 2.51 * 10^{-7}\ (u_f'')[x] - 4.57 * 10^{-7}\ u_f[x](u_f'')[x] \\
& + 4.36 * 10^{-7}\ u_f[x]^2(u_f'')[x] - 1.21 * 10^{-7}\ u_f[x]^3(u_f'')[x]
\end{aligned} \tag{20}
$$

$$
\begin{aligned}
u[70](x) \quad = \quad & 1.32425 + 0.00079\ u_f[x] - 0.00070\ u_f[x]^2 + 0.00043\ u_f[x]^3 \\
& + 3.89 * 10^{-6}\ (u_f'')[x] - 7.12 * 10^{-6}\ u_f[x](u_f'')[x] \\
& + 6.89 * 10^{-6}\ u_f[x]^2(u_f'')[x] - 2.64 * 10^{-6}\ u_f[x]^3(u_f'')[x]
\end{aligned} \tag{21}
$$

$$
\begin{aligned}
u[80](x) \quad = \quad & 1.31561 + 0.017\ u_f[x] - 0.015\ u_f[x]^2 + 0.009\ u_f[x]^3 \\
& + 0.000053\ (u_f'')[x] - 0.000098\ u_f[x](u_f'')[x] \\
& + 0.000095\ u_f[x]^2(u_f'')[x] - 0.000051\ u_f[x]^3(u_f'')[x]
\end{aligned} \tag{22}
$$

$$
\begin{aligned}
u[90](x) \quad = \quad & 1.14766 + 0.296 u_f[x] - 0.1991\ u_f[x]^2 + 0.0638\ u_f[x]^3 \\
& + 0.00044\ (u_f'')[x] - 0.00067\ u_f[x](u_f'')[x] \\
& + 0.00046\ u_f[x]^2(u_f'')[x] - 0.00018\ u_f[x]^3(u_f'')[x]
\end{aligned} \tag{23}
$$

Remark 2. *We observe that as* $\varepsilon > 0$ *decreases to the value* 0.01*, the solution approaches the constant value* 1.3247 *along the domain, up to the satisfaction of boundary conditions. This is expected since this value is an approximate solution of equation* $u^3 - u - 1 = 0$.

5. Conclusions

In this article, we develop an approximate numerical procedure related to the generalized method of lines. Such a procedure is of proximal nature and involves a parameter $K > 0$, which minimizes the concerning numerical error of the initial approximation.

We have presented numerical results concerning a Ginzburg–Landau-type equation. The results obtained are consisting with those expected for such a mathematical model.

In the last section, we present a software in MATHEMATICA for approximately solving a large class of similar systems of partial differential equations.

Finally, for a future research, we intend to extend the results for the Navier–Stokes system in fluid mechanics and related time-dependent models in physics and engineering.

Funding: This research received no external funding

Conflicts of Interest: The author declares no conflict of interest

References

1. Botelho, F. *Topics on Functional Analysis, Calculus of Variations and Duality*; Academic Publications: Sofia, Bulgaria, 2011.
2. Botelho, F. Existence of Solution for the Ginzburg-Landau System, a Related Optimal Control Problem and Its Computation by the Generalized Method of Lines. *Appl. Math. Comput.* **2012**, *218*, 11976–11989. [CrossRef]
3. Botelho, F. *Functional Analysis and Applied Optimization in Banach Spaces*; Springer: Cham, Switzerland, 2014.
4. Botelho, F.S. *Functional Analysis, Calculus of Variations and Numerical Methods in Physics and Engineering*; CRC Taylor and Francis: Boca Raton, FL, USA, 2020.
5. Adams, R.A. *Sobolev Spaces*; Academic Press: New York, NY, USA, 1975.
6. Adams, R.A.; Fournier, J.F. *Sobolev Spaces*, 2nd ed.; Elsevier: New York, NY, USA, 2003.

7. Annet, J.F. *Superconductivity, Superfluids and Condensates*, 2nd ed.; Oxford Master Series in Condensed Matter Physics; Oxford University Press: Oxford, UK, 2010; Reprint.
8. Landau, L.D.; Lifschits, E.M. *Course of Theoretical Physics, Volume 5—Statistical Physics, Part 1*; Elsevier, Butterworth-Heinemann: Oxford, UK, 2008; Reprint.
9. Toland, J.F. A duality principle for non-convex optimisation and the calculus of variations. *Arch. Rath. Mech. Anal.* **1979**, *71*, 41–61. [CrossRef]
10. Strikwerda, J.C. *Finite Difference Schemes and Partial Differential Equations*, 2nd ed.; SIAM: Philadelphia, PA, USA, 2004.