A note on the generalized method of lines and its explicit approach

Fabio Silva Botelho
Departamento de Matemática
Universidade Federal de Santa Catarina, UFSC
Florianópolis, SC - Brazil

April 30, 2019

Abstract

This article develops an explicit approach for the generalized method of lines. In such a method, the domain of the PDE in question is discretized in lines and the equation solution is written on these lines as functions of the boundary conditions and domain shape.

In the last section we present the concerning software and perform a numerical example.

1 Introduction

This article develops an improvement relating the generalized method of lines. In our previous publications [2, 4], we highlight the method there addressed may present a relevant error as a parameter \( \varepsilon > 0 \) is too small, that is, as \( \varepsilon \) is about 0.01, 0.001 or even smaller.

In the present paper we develop a solution for such a problem for a large class of non-linear elliptic PDEs.

At this point we reintroduce the generalized method of lines, originally presented in F.Botelho [2]. In the present context we add new theoretical and applied results to the original presentation. Specially the computations are all completely new. Consider first the equation

\[
\varepsilon \nabla^2 u + g(u) + f = 0, \quad \text{in } \Omega \subset \mathbb{R}^2,
\]  

with the boundary conditions

\[
u = 0 \quad \text{on } \Gamma_0 \quad \text{and} \quad u = u_f, \quad \text{on } \Gamma_1.
\]

From now on we assume that \( u_f, g \) and \( f \) are smooth functions (here we mean \( C^\infty \) functions), unless otherwise specified. Here \( \Gamma_0 \) denotes the internal boundary of \( \Omega \) and \( \Gamma_1 \) the external one. Consider the simpler case where

\[
\Gamma_1 = 2\Gamma_0,
\]

1
and suppose there exists \( r(\theta) \), a smooth function such that

\[
\Gamma_0 = \{(\theta, r(\theta)) \mid 0 \leq \theta \leq 2\pi\},
\]

being \( r(0) = r(2\pi) \).

In polar coordinates the above equation may be written as

\[
\frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} + g(u) + f = 0, \quad \text{in } \Omega,
\]

(2)

and

\[
u = 0 \text{ on } \Gamma_0 \text{ and } u = u_f \text{, on } \Gamma_1.
\]

Define the variable \( t \) by

\[
t = \frac{r}{r(\theta)}.
\]

Also defining \( \bar{u} \) by

\[
\bar{u}(r, \theta) = \bar{u}(t, \theta),
\]

dropping the bar in \( \bar{u} \), equation (1) is equivalent to

\[
\frac{\partial^2 u}{\partial t^2} + \frac{1}{t} f_2(\theta) \frac{\partial u}{\partial t} + \frac{1}{t} f_3(\theta) \frac{\partial^2 u}{\partial \theta \partial t} + \frac{f_4(\theta)}{t^2} \frac{\partial^2 u}{\partial \theta^2} + f_5(\theta)(g(u) + f) = 0,
\]

(3)

in \( \Omega \). Here \( f_2(\theta) \), \( f_3(\theta) \), \( f_4(\theta) \) and \( f_5(\theta) \) are known functions.

More specifically, denoting

\[
f_1(\theta) = \frac{-r'(\theta)}{r(\theta)},
\]

we have:

\[
f_2(\theta) = 1 + \frac{f_1'(\theta)}{1 + f_1(\theta)^2},
\]

\[
f_3(\theta) = \frac{2f_1(\theta)}{1 + f_1(\theta)^2},
\]

and

\[
f_4(\theta) = \frac{1}{1 + f_1(\theta)^2}.
\]

Observe that \( t \in [1, 2] \) in \( \Omega \). Discretizing in \( t \) (N equal pieces which will generate N lines ) we obtain the equation

\[
\frac{u_{n+1} - 2u_n + u_{n-1}}{d^2} + \frac{(u_n - u_{n-1})}{d} \frac{1}{t_n} f_2(\theta) \\
+ \frac{\partial(u_n - u_{n-1})}{\partial \theta} \frac{1}{t_n} f_3(\theta) + \frac{\partial^2 u_n}{\partial \theta^2} \frac{f_4(\theta)}{t_n^2} \\
+ f_5(\theta) \left( g(u_n) \frac{d^2}{\varepsilon} + f_n \frac{d^2}{\varepsilon} \right) = 0,
\]

(4)
\[ \forall n \in \{1, \ldots, N - 1\}. \text{ Here, } u_n(\theta) \text{ corresponds to the solution on the line } n. \text{ Thus we may write} \]
\[ u_n = T_n(u_{n-1}, u_n, u_{n+1}), \]
where
\[
T_n(u_{n-1}, u_n, u_{n+1}) = \left( u_{n+1} + u_n + u_{n-1} + \frac{(u_n - u_{n-1})}{d} t_n f_2(\theta) d^2 + \frac{\partial(u_n - u_{n-1})}{\partial \theta} \frac{1}{t_n} f_3(\theta) d^2 + \frac{\partial^2 u_n}{\partial \theta^2} \frac{f_4(\theta)}{t_n^2} d^2 + f_5(\theta) \left( g(u_n) \frac{d^2}{\varepsilon} + f_n \frac{d^2}{\varepsilon} \right) \right) / 3.0. \tag{5} \]

2 Some preliminaries results and the main algorithm

Now we recall a classical definition.

**Definition 2.1.** Let \( C \) be a subset of a Banach space \( U \) and let \( T : C \to C \) be an operator. Thus \( T \) is said to be a contraction mapping if there exists \( 0 \leq \alpha < 1 \) such that
\[ \| T(x_1) - T(x_2) \|_U \leq \alpha \| x_1 - x_2 \|_U, \forall x_1, x_2 \in C. \]

**Remark 2.2.** Observe that if \( \| T'(x) \|_U \leq \alpha < 1 \), on a convex set \( C \) then \( T \) is a contraction mapping, since by the mean value inequality,
\[ \| T(x_1) - T(x_2) \|_U \leq \sup_{x \in C} \{ \| T'(x) \| \} \| x_1 - x_2 \|_U, \forall x_1, x_2 \in C. \]

The next result is the base of our generalized method of lines. For a proof see [3].

**Theorem 2.3** (Contraction Mapping Theorem). Let \( C \) be a closed subset of a Banach space \( U \). Assume \( T \) is contraction mapping on \( C \), then there exists a unique \( \tilde{x} \in C \) such that \( \tilde{x} = T(\tilde{x}) \). Moreover, for an arbitrary \( x_0 \in C \) defining the sequence
\[ x_1 = T(x_0) \text{ and } x_{k+1} = T(x_k), \forall k \in \mathbb{N} \]
we have
\[ x_k \to \tilde{x}, \text{ in norm, as } k \to +\infty. \]

To obtain a fixed point for each \( T_n \) indicated (5) is perfectly possible if \( \varepsilon \approx O(1) \). However if \( \varepsilon > 0 \) is small, the error in process may be relevant.

To solve this problem we propose the following algorithm,

1. Choose \( K \approx 30 - 80 \) and set \( u_0 = 0 \).
2. Calculate \( u = \{u_n\} \) by solving the equation

\[
\begin{align*}
    u_{n+1} - 2u_n + u_{n-1} + \frac{(u_n - u_{n-1})}{d} \int_{t_n}^{t_{n+1}} f_2(\theta) d\theta \\
    + \frac{\partial(u_n - u_{n-1})}{\partial \theta} \int_{t_n}^{t_{n+1}} f_3(\theta) d\theta &+ \frac{\partial^2 u_n}{\partial \theta^2} \int_{t_n}^{t_{n+1}} f_4(\theta) d\theta \\
    + f_5(\theta) \left( g(u_n) \frac{d^2}{\varepsilon} + f_n \frac{d^2}{\varepsilon} \right) \\
    - K(u_n - (u_0)_n) \frac{d^2}{\varepsilon}
\end{align*}
\]

\( = 0. \quad (6) \)

Such an equation is solved through the Banach fixed point theorem, that is, defining

\[
T_n(u_n, u_{n+1}, u_{n-1}) = \left( u_{n+1} + u_n + u_{n-1} + \frac{(u_n - u_{n-1})}{d} \int_{t_n}^{t_{n+1}} f_2(\theta) d\theta \\
+ \frac{\partial(u_n - u_{n-1})}{\partial \theta} \int_{t_n}^{t_{n+1}} f_3(\theta) d\theta &+ \frac{\partial^2 u_n}{\partial \theta^2} \int_{t_n}^{t_{n+1}} f_4(\theta) d\theta \\
+ f_5(\theta) \left( g(u_n) \frac{d^2}{\varepsilon} + f_n \frac{d^2}{\varepsilon} \right) \\
+ K(u_n - (u_0)_n) \frac{d^2}{\varepsilon} \right) / \left( 3 + K \frac{d^2}{\varepsilon} \right)
\]

\( (7) \)

\[(6)\) stands for

\[u_n = T_n(u_{n-1}, u_n, u_{n+1}),\]

so that for \( n = 1 \) we have

\[u_1 = T_1(0, u_1, u_2).\]

We may use the Contraction Mapping Theorem to calculate \( u_1 \) as a function of \( u_2 \). The procedure would be,

(a) set \( x_1 = u_2 \),

(b) obtain recursively

\[x_{k+1} = T_1(0, x_k, u_2),\]

(c) and finally get

\[u_1 = \lim_{k \to \infty} x_k = g_1(u_2).\]

Thus, we have obtained

\[u_1 = g_1(u_2).\]

We can repeat the process for \( n = 2 \), that is, we can solve the equation

\[u_2 = T_2(u_1, u_2, u_3),\]

which from above stands for

\[u_2 = T_2(g_1(u_2), u_2, u_3).\]

The procedure would be:
(a) Set \( x_1 = u_3 \),
(b) calculate
\[ x_{k+1} = T_2(g_1(x_k), x_k, u_3), \]
(c) obtain
\[ u_2 = \lim_{k \to \infty} x_k = g_2(u_3). \]
We proceed in this fashion until obtaining
\[ u_{N-1} = g_{N-1}(u_N) = g_{N-1}(u_f). \]
Being \( u_f \) known we have obtained \( u_{N-1} \). We may then calculate
\[ u_{N-2} = g_{N-2}(u_{N-1}), \]
\[ u_{N-3} = g_{N-3}(u_{N-2}), \]
and so on, up to finding
\[ u_1 = g_1(u_2). \]
Thus this part of the problem is solved.

3. Set \( u_0 = u \) and go to item[2] up to the satisfaction of an appropriate convergence criterion.

**Remark 2.4.** Here we consider some points concerning the convergence of the method.

In the next lines the norm indicated refers to the infinity one for \( C([0, 2\pi]; \mathbb{R}^{N-1}) \). In particular for \( n = 1 \) from above we have:
\[ u_1 = T_1(0, u_1, u_2), \]
that is
\[ u_2 - 2u_1 - \frac{K d^2}{\varepsilon} u_1 + O \left( \frac{K d^2}{\varepsilon} \right) = 0. \]

Hence, denoting
\[ a[1] = \frac{1}{\left( 2 + K \frac{d^2}{\varepsilon} \right)} \]
and
\[ a[n] = \frac{1}{\left( 2 + K \frac{d^2}{\varepsilon} - a[n-1] \right)}, \quad \forall n \in \{2, \ldots, N-1\}, \]
for \( N \) sufficiently big we may obtain
\[ \|u_1 - a[1] u_2\| = O \left( \frac{K d^2}{\varepsilon} \right), \]
and by induction
\[ \|u_n - a[n] u_{n+1}\| = n O \left( \frac{K d^2}{\varepsilon} \right), \]
so that we would have
\[ \|u_n - a[n] u_{n+1}\| \leq O \left( \frac{K d}{\varepsilon} \right), \quad \forall n \in \{1, \ldots, N-1\} \]
This last calculation is just to clarify that the procedure of obtaining the relation between consecutive lines through the contraction mapping theorem is well defined.
3 A numerical example, the explicit approach

In this section we present a numerical example. Consider the equation

\[ \varepsilon \nabla^2 u + g(u) + 1 = 0, \quad \text{in } \Omega \subset \mathbb{R}^2, \]  

(8)

where, for a Ginburg-Landau type equation (see [1, 5] for the corresponding models in physics),

\[ g(u) = -u^3 + u, \]

with the boundary conditions

\[ u = u_1 \text{ on } \partial \Omega = \Gamma_0 \cup \Gamma_1, \]

where \( \Omega = \{(r, \theta) : 1 \leq r \leq 2, \ 0 \leq \theta \leq 2\pi \}, \)

\[ u_1 = 0, \ \text{on } \Gamma_0 = \{(1, \theta) : 0 \leq \theta \leq 2\pi \}, \]

\[ u_1 = u_f(\theta), \ \text{on } \Gamma_1 = \{(2, \theta) : 0 \leq \theta \leq 2\pi \}. \]

Through the generalized method of lines, for \( N = 10 \) (10 lines), \( d = 1/N \) in polar coordinates and finite differences (please see [6] for general schemes in finite differences), equation (8) stands for

\[ (u_{n+1} - 2u_n + u_{n-1}) + \frac{1}{r_n}(u_n - u_{n-1})d + \frac{1}{r_n^2} \frac{\partial^2 u_n}{\partial \theta^2} d^2 - (u_n^3 + u_n) \frac{d^2}{\varepsilon} + \frac{d^2}{\varepsilon} = 0, \]

\( \forall n \in \{1, \ldots, N - 1\}. \)

At this point we present, through the generalized method of lines, the concerning algorithm which may be for the softwares maple or mathematica.
In this software, $x$ stands for $\theta$.

$m_8 = 10$; (number of lines)

$d = 1.0/m_8$; (thickness of the grid)

$e_1 = 0.01$; ($\varepsilon = e_1$)

$K = 70.0$;

Clear[$d_1, u, a, b, h$];

For[$i = 1, i < m_8, i++$,

$z_1[i] = 0.0$; (vector which stores $Ku_0(i)$)

For[$k_1 = 1, k_1 < 180, k_1++$,

Print[$k_1$];

$a = 0.0$;

For[$i = 1, i < m_8, i++$,

Print[$i$];

$t = 1.0 + i * d$;

$b[x_] = u[i + 1][x]$;

$b_{12} = 2.0$;

$A_{18} = 5.0$;

$k = 1$;

While[$b_{12} > 10^{-4}$,

$k = k + 1$;

$z = (u[i + 1][x] + b[x] + a + 1/t * (b[x] - a) * d * d_1^2 + 1/t^2 * D[b[x], \{x, 2\}] * d^2 * d_1^2 - b[x]^3 * d^2 * d_1^2/e_1 + b[x] * d^2 * d_1^2/e_1 + 1.0 * d^2/e_1 + z_1[i] * d^2/e_1)/(3.0 + K * d^2/e_1)$;

$z = Series[z, \{d_1, 0, 2\}, \{u[x], 0, 3\}, \{u''[x], 0, 1\}, \{u''[x], 0, 1\}, \{u''[x], 0, 1\}, \{u''[x], 0, 1\}, \{u''[x], 0, 1\}, \{u''[x], 0, 1\}]$;

$z = Normal[z]$;

$z = Expand[z]$;

$b[x_] = z$;

$u[i + 1][x_] = 0.0$;

$u[x_] = 0.0$;

$d_1 = 1.0$;

$A_{19} = z$;

$b_{12} = Abs[A_{19} - A_{18}]$;

$A_{18} = A_{19}$;

Clear[$u, u_f, d_1$];

$a_1 = b[x]$;

Clear[$b$];

$u[i + 1][x_] = b[x]$;

$h[i] = a_1$;

$a = a_1$;

$b[x_] = u_f[x]$;

$d_1 = 1.0$;
For $i = 1, i < m_8, i + +,$
\[
W_1[m_8 - i] = \text{Series}[h[m_8 - i], \{u_f[x], 0, 3\}, \{u_f'[x], 0, 1\}, \{u_f''[x], 0, 1\}, \{u_f'''[x], 0, 0\}, \{u_f''''[x], 0, 0\}];
\]
\[
W = \text{Normal}[W_1[m_8 - i]]; \quad b[x_] = \text{Expand}[W]; \quad v[m_8 - i] = \text{Expand}[W];
\]
For $i = 1, i < m_8, i + +,$
\[
z_1[i] = K * v[i];
\]
\[
d_1 = 1.0;
\]
\[
\text{Clear}[d_1, u, b]
\]

(10)

At this point we present the expressions for 10 lines, firstly for $\varepsilon = 1$ and $K = 0$. In the next lines $x$ stands for $\theta$.

For each line $u[n]$ we have obtained,

\[
u[1] = 0.0588308 + 0.167434u_f[x] - 0.00488338u_f[x]^2 - 0.00968371u_f[x]^3 + 0.0126122u_f''[x]
\]
\[-0.000641358u_f[x] u_f'[x] - 0.00190065u_f[x]^2 u_f''[x] + 0.000660286u_f[x]^3 u_f'''[x]
\]

\[
u[2] = 0.101495 + 0.316995u_f[x] - 0.00919963u_f[x]^2 - 0.0182924u_f[x]^3 + 0.0225921u_f''[x]
\]
\[-0.00113308u_f[x] u_f''[x] - 0.00336691u_f[x]^2 u_f''[x] + 0.00122652u_f[x]^3 u_f''''[x]
\]

\[
u[3] = 0.1295 + 0.450424u_f[x] - 0.0127925u_f[x]^2 - 0.0257175u_f[x]^3 + 0.0294791u_f''[x]
\]
\[-0.00142448u_f[x] u_f''[x] - 0.00428071u_f[x]^2 u_f''[x] + 0.000160933u_f[x]^3 u_f''''[x]
\]

\[
u[4] = 0.143991 + 0.568538u_f[x] - 0.0153256u_f[x]^2 - 0.0315703u_f[x]^3 + 0.0331249u_f''[x]
\]
\[-0.0014821u_f[x] u_f''[x] - 0.00456619u_f[x]^2 u_f''[x] + 0.000168613u_f[x]^3 u_f''''[x]
\]

\[
u[5] = 0.146024 + 0.672307u_f[x] - 0.0164357u_f[x]^2 - 0.0352883u_f[x]^3 + 0.0336371u_f''[x]
\]
\[-0.00131323u_f[x] u_f''[x] - 0.00421976u_f[x]^2 u_f''[x] + 0.000141442u_f[x]^3 u_f''''[x]
\]

\[
u[6] = 0.136541 + 0.762571u_f[x] - 0.0158578u_f[x]^2 - 0.0361974u_f[x]^3 + 0.031264u_f''[x]
\]
\[-0.00074635u_f[x] u_f''[x] - 0.0033176u_f[x]^2 u_f''[x] + 0.0000901069u_f[x]^3 u_f''''[x]
\]
For each line we present the results relating the software indicated, with \( K = 70 \).

For each line \( u[n] \) we have obtained,

\[
u[7] = 0.116389 + 0.840008u_f[x] - 0.0135378u_f[x]^2 - 0.0336098u_f[x]^3 + 0.0263271u''_f[x] \\
-0.000565842u_f[x] u''_f[x] - 0.00210507u_f[x]^2 u''_f[x] + 0.0000375075u_f[x]^3 u''_f[x]
\]

\[
u[8] = 0.0864095 + 0.905032u_f[x] - 0.00970893u_f[x]^2 - 0.0269167u_f[x]^3 + 0.0192033u''_f[x] \\
-0.000206117u_f[x] u''_f[x] - 0.000856701u_f[x]^2 u''_f[x] + 5.78758 \times 10^{-6}u_f[x]^3 u''_f[x]
\]

\[
u[9] = 0.0473499 + 0.958203u_f[x] - 0.00491745u_f[x]^2 - 0.0157466u_f[x]^3 + 0.0102907u''_f[x].
\]

In the next lines we present the results relating the software indicated, with \( \varepsilon = 0.01 \) and \( K = 70 \).

For each line \( u[n] \) we have obtained,

\[
u[1] = 1.08673 + 4.6508 \times 10^{-7}u_f[x] - 1.11484 \times 10^{-7}u_f[x]^2 + 3.25552 \times 10^{-8}u_f[x]^3 \\
+5.13195 \times 10^{-9}u''_f[x] - 2.24741 \times 10^{-9}u_f[x] u''_f[x] \\
+8.97477 \times 10^{-10}u_f[x]^2 u''_f[x] - 8.86957 \times 10^{-11}u_f[x]^3 u''_f[x]
\]

\[
u[2] = 1.27736 + 1.51118 \times 10^{-6}u_f[x] - 4.39811 \times 10^{-7}u_f[x]^2 + 1.55883 \times 10^{-7}u_f[x]^3 \\
+1.33683 \times 10^{-8}u''_f[x] - 7.50593 \times 10^{-9}u_f[x] u''_f[x] \\
+3.77548 \times 10^{-9}u_f[x]^2 u''_f[x] - 4.9729 \times 10^{-10}u_f[x]^3 u''_f[x]
\]

\[
u[3] = 1.30559 + 6.91602 \times 10^{-6}u_f[x] - 2.21813 \times 10^{-6}u_f[x]^2 + 8.89891 \times 10^{-7}u_f[x]^3 \\
+4.85851 \times 10^{-8}u''_f[x] - 3.22542 \times 10^{-8}u_f[x] u''_f[x] \\
+1.90602 \times 10^{-9}u_f[x]^2 u''_f[x] - 3.20439 \times 10^{-9}u_f[x]^3 u''_f[x]
\]

\[
u[4] = 1.30968 + 0.0000354152u_f[x] - 0.0000116497u_f[x]^2 + 5.06682 \times 10^{-6}u_f[x]^3 \\
+1.93523 \times 10^{-7}u''_f[x] - 1.42948 \times 10^{-7}u_f[x] u''_f[x] \\
+9.56419 \times 10^{-8}u_f[x]^2 u''_f[x] - 2.02133 \times 10^{-8}u_f[x]^3 u''_f[x]
\]

\[
u[5] = 1.31014 + 0.000193272u_f[x] - 0.0000616231u_f[x]^2 + 0.0000278964u_f[x]^3 \\
+7.96293 \times 10^{-7}u''_f[x] - 6.22727 \times 10^{-7}u_f[x] u''_f[x] \\
+4.58264 \times 10^{-7}u_f[x]^2 u''_f[x] - 1.21033 \times 10^{-7}u_f[x]^3 u''_f[x]
\]
\[ u[6] = 1.30935 + 0.0011121u_f[x] - 0.000331366u_f[x]^2 + 0.000149489u_f[x]^3 \\
+ 3.34907 \times 10^{-6}u_f''[x] - 2.66647 \times 10^{-6}u_f[x]\]
\[+2.09295 \times 10^{-6}u_f[x]^2u_f''[x] - 6.87318 \times 10^{-7}u_f[x]^3u_f''[x] \]

\[ u[7] = 1.30383 + 0.00665186u_f[x] - 0.00182893u_f[x]^2 + 0.000789763u_f[x]^3 \\
+ 0.0000140899u_f''[x] - 0.000011257u_f[x]u_f''[x] \\
+ 9.14567 \times 10^{-6}u_f[x]^2u_f''[x] - 3.70794 \times 10^{-6}u_f[x]^3u_f''[x] \]

\[ u[8] = 1.26934 + 0.040481u_f[x] - 0.0101978u_f[x]^2 + 0.00408206u_f[x]^3 \\
+ 0.00005612u_f''[x] - 0.0000457826u_f[x] \\
+ u_f''[x] + 0.000037641u_f[x]^2u_f''[x] - 0.000018566u_f[x]^3u_f''[x] \]

\[ u[9] = 1.05971 + 0.23709u_f[x] - 0.0493175u_f[x]^2 + 0.0173591u_f[x]^3 \\
+ 0.00017652u_f''[x] - 0.000150827u_f[x]u_f''[x] \\
+ 0.000123238u_f[x]^2u_f''[x] - 0.0000709253u_f[x]^3u_f''[x] \]

**Remark 3.1.** Observe that since \( \varepsilon = 0.01 \) the solution is close to the constant value 1.3247 along the domain, which is an approximate solution of equation \( -u^3 + u + 1.0 = 0 \).

## 4 Conclusion

In this article we have developed an improvement concerning an explicit approach for the generalized method of lines. For a large class of models, we have solved the problem of minimizing the error as the parameter \( \varepsilon > 0 \) is small. The main result corresponds to a kind of proximal formulation combined with the standard generalized method of lines approach.

We highlight the method here developed may be applied to a large class of problems, including the Ginzburg-Landau system in superconductivity in the presence of a magnetic field and respective magnetic potential.

We intend to address this kind of model and others such as the Navier-Stokes system in a future research.

## References

[1] J.F. Annet, Superconductivity, Superfluids and Condensates, 2nd edn. (Oxford Master Series in Condensed Matter Physics, Oxford University Press, Reprint, 2010)

[2] F. Botelho, Topics on Functional Analysis, Calculus of Variations and Duality, Academic Publications (IJPAM), Sofia, 2011.
[3] F. Botelho, Functional Analysis and Applied Optimization in Banach Spaces, (Springer Switzerland, 2014).

[4] F. Botelho, *Existence of solution for the Ginzburg-Landau system, a related optimal control problem and its computation by the generalized method of lines*, Applied Mathematics and Computation, 218, 11976-11989, (2012).

[5] L.D. Landau and E.M. Lifschits, *Course of Theoretical Physics, Vol. 5- Statistical Physics*, part 1. (Butterworth-Heinemann, Elsevier, reprint 2008).

[6] J.C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, SIAM, second edition (2004).