

Faculdade de Ciências e Tecnologia

Departamento de Engenharia Eletrotécnica e de Computadores

Rui Diogo Quesado Alves

EasyMag – Toolbox para cálculo de campos magnéticos de baixa frequência

Dissertação de Mestrado em Engenharia Eletrotécnica e de Computadores, orientada por Tony Richard de Oliveira de Almeida, António Paulo Mendes Breda Dias Coimbra e apresentada à Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Setembro de 2018



UNIVERSIDADE DE COIMBRA

Abstract

The technological evolution and the high industrial development that characterizes the contemporary civilization, makes the society strongly dependent on energy. Due to this evolution, there has been an increase of the electrical devices usage and consequently an increase of the electrical networks, arising an increase in the intensity of magnetic fields originated by the electric current that runs through the networks and electrical devices. This increased intensity of magnetic fields has given rise to the concern in the society about the effects of magnetic fields on human health. This was the motivation for this work: to develop the software '*EASYMAG*' to calculate the magnetic flux density originated by electric currents in conductors, using the Biot-Savart law.

The developed software incorporates the following functionalities: verification of the correct data entry, alerting the user if a circuit consisting of several lines is open; graphical visualization of the electrical circuit defined by the user with different colors depending on whether the circuit is unifilar or bifilar; calculation of the magnetic flux density at the points indicated by the user, showing the location where this value is maximum and minimum; graphical display of magnetic flux density at user-defined points; save simulation done by the user.

The software '*EASYMAG*' is innovative compared to other existing software, because it uses Excel as a data entry method, representing the desired electrical circuit without having to program the circuit in Matlab, making the software '*EASYMAG*' accessible to all people with basic knowledge of Matlab.

Three theoretical examples are presented to validate the software. The software, in these three examples, obtained results with a very small error, depending on the number of points used for the representation of the circuit elements, curve and solenoid. The developed software is a reliable tool for the calculation of theoretical and practical cases of the magnetic flux density originated by electric currents in conductors, the associated error being close to zero.

Keywords: Matlab, magnetic fields, magnetic field simulation, Biot-Savart Law.

Resumo

A evolução tecnológica e o elevado desenvolvimento industrial que caracteriza a civilização contemporânea, torna a sociedade fortemente dependente de energia. Devido a esta evolução, houve um aumento do uso de aparelhos elétricos e conseqüentemente um aumento das redes elétricas, surgindo assim um aumento na intensidade de campos magnéticos originados pela corrente elétrica que percorre as redes e os aparelhos. Este aumento da intensidade de campos magnéticos originou uma preocupação na sociedade sobre os efeitos do campo magnético na saúde dos seres humanos. Foi esta a motivação para este trabalho: desenvolver o software ‘*EASYMAG*’ para calcular a densidade do fluxo magnético originado por correntes elétricas em condutores, utilizando a lei de Biot-Savart.

O software desenvolvido tem incorporadas as seguintes funcionalidades: verificação da introdução correta dos dados, alertando o utilizador se um circuito constituído por várias linhas estiver aberto; visualização gráfica do circuito elétrico definido pelo utilizador com cores diferentes consoante o circuito seja unifilar ou bifilar; cálculo da densidade do fluxo magnético nos pontos indicados pelo utilizador, mostrando a localização onde esse valor é máximo e mínimo; visualização gráfica da densidade do fluxo magnético nos pontos definidos pelo utilizador; guardar a simulação feita pelo utilizador.

O software ‘*EASYMAG*’ é inovador comparado com os outros softwares existentes, porque utiliza o Excel como método de introdução de dados, conseguindo assim representar o circuito elétrico pretendido pelo utilizador, sem que este tenha que programar em Matlab o circuito, tornando ainda o software ‘*EASYMAG*’ acessível para todas as pessoas com conhecimento básico de Matlab.

Para validar o software foram usados três exemplos teóricos. O software, nestes três exemplos, obteve resultados com um erro muito pequeno, dependendo do número de pontos usados para a representação dos elementos circuitais, curva e solenoide. O software desenvolvido é assim uma ferramenta fiável para o cálculo de casos teóricos e práticos da densidade do fluxo magnético originado por correntes elétricas em condutores, sendo o erro associado próximo de zero.

Palavras Chave: Matlab, campo magnético, simulação do campo magnético, Lei Biot-Savart.

Agradecimentos

Esta dissertação não teria sido possível sem o suporte e carinho de um conjunto de pessoas que gostaria de mencionar e agradecer.

Aos meus amigos, a todos eles, pela amizade e apoio incondicional.

Aos meus orientadores, o Professor Tony Almeida e o Professor Paulo Coimbra, pela sua perseverança e pela confiança depositada em mim em todas as fases deste trabalho.

Por fim, à minha família, em especial aos meus pais e irmãos, pela dedicação constante, carinho e confiança demonstrado durante todo o meu percurso de formação. A eles expresso o meu eterno agradecimento.

Obrigado a todos!

Lista de símbolos

a_j	Distância entre x_{E_j} e x_{B_j} [m]
ac	Ângulo central da divisão feita pelo o número de pontos dados
B	Densidade do fluxo magnético [T]
\vec{B}_{j_x}	Componente x da densidade do fluxo magnético [T]
\vec{B}_{j_y}	Componente y da densidade do fluxo magnético [T]
\vec{B}_{j_z}	Componente z da densidade do fluxo magnético [T]
B_j	Ponto inicial do segmento
b_j	Distância entre y_{E_j} e y_{B_j} [m]
c_j	Distância entre z_{E_j} e z_{B_j} [m]
$d\vec{B}$	Contribuição elementar para o campo magnético[T]
dGamma	Número de pontos que representa a curva ou solenoide
$d\vec{l}$	Comprimento do fio[m]
E_j	Ponto final do segmento
I	Corrente instantânea [A]
l_j	Tamanho do segmento
Numx	Número de pontos no eixo x
Numy	Número de pontos no eixo y
Numz	Número de pontos no eixo z
r	Distância entre o ponto A e $d\vec{l}$ [m]
\vec{r}	Vetor unitário incidido no segmento com orientação para o ponto A
R_i	Distância entre o ponto B_j e o ponto T.

R_f	Distância entre o ponto E_j e o ponto T.
S_j	Ponto do segmento mais próximo do ponto T
t_j	Variável para cálculo auxiliar
x_{B_j}	Coordenada x do ponto inicial B_j do segmento
x_{E_j}	Coordenada x do ponto final E_j do segmento
x_{S_j}	Coordenada x do ponto S
y_{B_j}	Coordenada y do ponto inicial B_j do segmento
y_{E_j}	Coordenada y do ponto final E_j do segmento
y_{S_j}	Coordenada y do ponto S
z_{B_j}	Coordenada z do ponto inicial B_j do segmento
z_{E_j}	Coordenada z do ponto final E_j do segmento
z_{S_j}	Coordenada z do ponto S
β	Ângulo de rotação
μ_0	Permeabilidade magnética no vázio [$H \cdot m^{-1}$]
δ_j	Distância entre os pontos T e S

Índice

Abstract.....	ii
Resumo	iii
Agradecimentos	iv
Lista de figuras	ix
Capítulo 1-Introdução.....	1
1.1 Objetivos.....	1
1.2 Estrutura do documento	2
Capítulo 2 - Revisão bibliográfica.....	3
2.1 Campo magnético	3
2.2 Biot-Savart magnetic Toolbox.....	4
2.3 3D Computation of the Power Lines Magnetic Field.....	5
Capítulo 3 - Software desenvolvido	7
3.1 Easymag_Data_in	7
3.2 Representação dos circuitos.....	8
3.2.1 Linhas	8
3.2.2 Solenoides.....	10
3.2.3 Curvas.....	12
3.3 Estrutura de dados.....	13
3.3.1 Easymag_init	13
3.3.2 Easymag_add_segment	13
3.4 Lista dos resultados pretendidos.....	14
3.4.1 Pontos	14
3.4.2 Volume	15
3.4.3 Plano	16
3.4.4 Reta.....	19
3.5 Easymag_get_B	20
3.6 Verificação dos dados	20
3.7 Outputs.....	21
3.7.1 Pontos	21
3.7.2 Volume	22
3.7.3 Plano.....	25

3.7.4 Reta.....	27
3.8 Easymag_save.....	28
3.9 Main.....	28
Capítulo 4 - Validação do código.....	29
4.1 Linhas.....	29
4.2 Curva.....	31
4.3 Solenoide.....	37
Capítulo 5 - Conclusões e trabalho futuro.....	44
5.1 Conclusões.....	44
5.2 Trabalho futuro.....	45
Referências.....	46

Lista de figuras

Figura 2.1 - Representação dos vetores \mathbf{dB} e \mathbf{dl} de Biot-Savart, aplicados num fio condutor com uma corrente elétrica I .	4
Figura 2.2 – Representação dos dados para o cálculo da densidade do fluxo magnético no ponto T .	5
Figura 3.1 - Folhas presentes no ficheiro excel denominado ‘Easymag_Data_in’.	7
Figura 3.2 - Folha ‘lines’ presente do ficheiro excel ‘Easymag_Data_in’.	8
Figura 3.3 - Exemplo de um circuito fechado com seis linhas (troços) bifilares.	9
Figura 3.4 - Representação gráfica da linha bifilar definida na Fig. 3.3.	9
Figura 3.5 - Folha ‘solenoids’ presente do ficheiro excel ‘Easymag_Data_in’.	11
Figura 3.6 - Folha ‘curvs’ presente do ficheiro excel ‘Easymag_Data_in’.	12
Figura 3.7 - Folha ‘points’ presente do ficheiro excel ‘Easymag_Data_in’.	14
Figura 3.8 - Output pontos.	15
Figura 3.9 - Folha ‘volume’ presente do ficheiro excel ‘Easymag_Data_in’.	15
Figura 3.10 - Folha ‘plane’ presente do ficheiro excel ‘Easymag_Data_in’.	16
Figura 3.11 - Output plano.	17
Figura 3.12 - Plano antes da rotação ser realizada.	18
Figura 3.13 - Plano depois realização da rotação.	18
Figura 3.14 - Representação da rotação dos três eixos [8].	19
Figura 3.15 - Folha ‘reta’ presente do ficheiro excel ‘Easymag_Data_in’.	19
Figura 3.16 - Output reta.	20
Figura 3.17 - Representação da densidade do fluxo magnético.	22
Figura 3.18 - Representação da densidade do fluxo magnético resultante.	23
Figura 3.19 - Representação densidade do fluxo magnético segundo a componente do eixo x .	23
Figura 3.20 - Representação densidade do fluxo magnético segundo a componente do eixo y .	24
Figura 3.21 - Representação densidade do fluxo magnético segundo a componente do eixo z .	24
Figura 3.22 - Representação da densidade do fluxo magnético resultante.	25
Figura 3.23 - Representação densidade do fluxo magnético segundo a componente do eixo x .	26
Figura 3.24 - Representação densidade do fluxo magnético segundo a componente .	26
Figura 3.25 - Representação densidade do fluxo magnético segundo a componente do eixo z .	27
Figura 3.26 - Representação da densidade do fluxo magnético.	28
Figura 3.27 - Folha ‘save’ presente do ficheiro excel ‘Easymag_Data_in’.	28
Figura 4.1 - Exemplo teórico.	29
Figura 4.2 - Exemplo teórico usando o software desenvolvido.	30
Figura 4.3 - Resultado do exemplo teórico usando o software.	31
Figura 4.4 - Exemplo teórico.	31
Figura 4.5 - Exemplo teórico representado usando software desenvolvido.	32
Figura 4.6 - Solução sem algoritmo.	33

Figura 4.7	- Solução usando algoritmo.....	34
Figura 4.8	- Exemplo teórico representado usando software desenvolvido.....	35
Figura 4.9	- Solução sem algoritmo.	35
Figura 4.10	- Solução usando algoritmo.....	36
Figura 4.11	- Solenoide (0,01m).	38
Figura 4.12	- Solenoide (0,02m).	38
Figura 4.13	- Solenoide (0,05m).	39
Figura 4.14	- Solenoide (0,1m).	39
Figura 4.15	- Solenoide (0,2m).	40
Figura 4.16	- Solenoide (0,5m).	40
Figura 4.17	- Solenoide (1m).	41
Figura 4.18	- Solenoide (2m).	41
Figura 4.19	- Solenoide (5m).	42
Figura 4.20	- Solenoide (10m).	42
Figura 4.21	- Resultados obtidos usando o software, comprimento (0,01m - 1m). ...	43
Figura 4.22	- Resultados obtidos usando o software, comprimento (0,01m – 10m)..	43

Capítulo 1-Introdução

Desde finais dos anos oitenta existe uma maior preocupação com os efeitos do campo magnético na saúde dos seres humanos, tornando-se assim matéria de vários estudos científicos.

Desta preocupação, foi identificada a hipótese de desenvolver um software capaz de calcular a densidade do fluxo magnético. Após análise da situação, tomou-se a decisão de desenvolver um software em Matlab por forma adquirir informação para uma melhor compreensão deste problema.

O software desenvolvido tem as seguintes características: a capacidade de visualizar o circuito elétrico definido pelo utilizador; escolher a localização onde se pretende saber o valor da densidade do fluxo magnético; verificação da introdução correta dos dados, alertando o utilizador se um circuito constituído por várias linhas estiver aberto; guardar simulação feita pelo utilizador. No final, os resultados são mostrados de uma forma numérica e gráfica a duas e três dimensões. A interface gráfica do software desenvolvido foi feita em inglês.

Relativamente à tipologia do circuito elétrico foram escolhidos três elementos circuitais: linhas (segmento de reta) podendo ser unifilar ou bifilar, curvas (arco de circunferência) e solenoides. Em termos de onde se pretende saber o valor da densidade do fluxo magnético foram escolhidos quatro tipos de localização: pontos, volume, plano e reta.

Finalmente, como forma de demonstrar a aplicação do trabalho desenvolvido, é feita a comparação dos resultados obtidos pelo software com três exemplos teóricos.

1.1 Objetivos

O principal objetivo desta dissertação é o desenvolvimento do ‘*EASYMAG*’ que é um software em Matlab de fácil compreensão e utilização para o cálculo da densidade do fluxo magnético criado por condutores percorridos por correntes elétricas.

A ferramenta desenvolvida inclui cinco grandes funcionalidades:

- Verificação da introdução correta dos dados, alertando o utilizador se um circuito constituído por várias linhas estiver aberto.
- Visualização gráfica do circuito elétrico definido pelo utilizador com cores diferentes consoante o circuito seja unifilar ou bifilar.

- Cálculo da densidade do fluxo magnético nos pontos indicados pelo utilizador, mostrando a localização onde esse valor é máximo e mínimo.
- Visualização gráfica da densidade do fluxo magnético nos pontos definidos pelo utilizador.
- Guardar simulação feita pelo utilizador.

1.2 Estrutura do documento

Este documento encontra-se estruturado em cinco capítulos. No primeiro, apresenta-se uma breve introdução ao tema em questão e no segundo é feito um enquadramento sobre os conceitos usados no desenvolvimento do software. Em seguida, no terceiro capítulo é exposta a informação relativa aos processos utilizados no desenvolvimento, e no quarto capítulo é feita uma validação do código. Finalmente, no quinto capítulo, serão retiradas algumas conclusões sobre o trabalho realizado.

Capítulo 2 - Revisão bibliográfica

Neste capítulo, é descrito com algum pormenor a lei de Biot-Savart, como interveniente para o cálculo da densidade do fluxo magnético, servindo de análise para melhor compreensão do campo magnético gerado por um fio condutor percorrido por uma corrente. Apresenta-se também o software ‘*Biot-Savart magnetic Toolbox*’ e o motivo pelo qual não foi utilizado no software desenvolvido, ‘*EASYMAG*’. Por último descreve-se o algoritmo usado pelo software ‘*EASYMAG*’ para o cálculo da densidade do fluxo magnético.

2.1 Campo magnético

O campo magnético é criado por magnetes permanentes ou através de uma carga elétrica em movimento [1]. Nesta dissertação é analisado o campo magnético criado por uma carga elétrica em movimento utilizando a lei de Biot-Savart para o cálculo da densidade do fluxo magnético.

A lei de Biot-Savart é uma relação quantitativa fundamental entre uma corrente elétrica e o campo magnético que produz. Com base nas experiências efetuadas, em 1820, os dois cientistas franceses Jean-Baptiste Biot e Félix Savart formularam a seguinte lei: uma corrente elétrica que flui em um condutor, ou uma carga elétrica em movimento produz um campo magnético. Ou seja, o valor da densidade do fluxo magnético num ponto no vazio é calculado somando a contribuição de cada segmento de um fio condutor percorrido por corrente [2]. Em primeiro lugar, há que considerar que o valor da densidade de fluxo magnético em um ponto é diretamente proporcional ao valor da corrente no condutor e ao comprimento do segmento percorrido pela corrente em consideração. Se a linha do ponto em análise para o segmento fizer um ângulo de noventa graus com o segmento atual, o valor é máximo. À medida que esse ângulo se torna menor, o valor diminui, tornando-se zero quando o ponto está em linha com o segmento ou no próprio segmento. Além disso, a densidade do fluxo magnético em um ponto depende da distância entre o ponto e o segmento [2] [3].

Esta lei pode ser descrita pela seguinte equação:

$$d\vec{B} = \frac{\mu_0 I}{4\pi} \frac{d\vec{l} \times \vec{r}}{r^3} \quad (\text{Eq.2.1})$$

sendo $d\vec{B}$ a contribuição elementar para o campo magnético num ponto A; I a corrente que percorre o fio condutor; μ_0 corresponde à permeabilidade magnética no vácuo [$4 \times \pi \times 10^{-7} \text{ H} \cdot \text{m}^{-1}$]; r a distância entre o ponto A e $d\vec{l}$ [m] e \vec{r} é o vetor unitário incidido no segmento com orientação para o ponto A (figura 2.1).

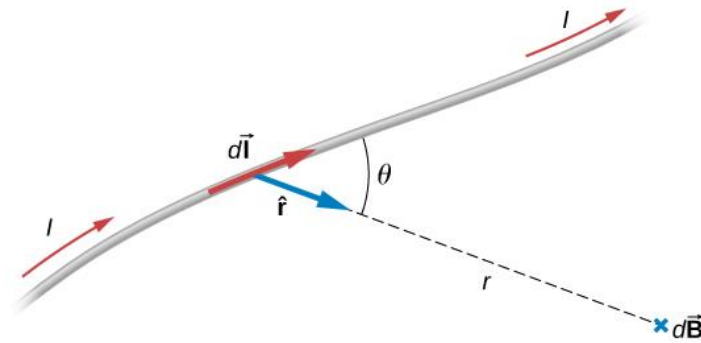


Figura 2.1 - Representação dos vetores $d\vec{B}$ e $d\vec{l}$ de Biot-Savart, aplicados num fio condutor com uma corrente elétrica I .

2.2 Biot-Savart magnetic Toolbox

A ‘*Biot-Savart magnetic Toolbox*’ é um conjunto de funções em Matlab criado por Loic Quéval e utiliza a lei de Biot-Savart para o cálculo da densidade do fluxo magnético, não tendo a opção de introdução de dados, obrigando o utilizador a programar em Matlab o circuito que pretende. O software foi realizado a pensar num só elemento de circuito. Ao usar mais de um elemento de circuito os resultados não correspondem ao esperado. Um exemplo: quando a localização é a mesma ou está em linha com o elemento de circuito o resultado da densidade do fluxo magnético deve ser zero, o que não se verifica. Devido a este software só realizar o cálculo, obriga o utilizador a programar a introdução de dados. Por consequência o erro poderia ter sido aquando da introdução de dados feita pelo utilizador e não pelo software. Foi então decidido não usar este software e desenvolver uma função para o cálculo da densidade do fluxo magnético.

2.3 3D Computation of the Power Lines Magnetic Field

O paper ‘3D Computation of the Power Lines Magnetic Field’, escrito por *Tonci Modric, Slavko Vujevic e Dino Lovri et al.* [7] apresenta um algoritmo utilizando a lei de Biot-Savart para o cálculo da densidade do fluxo magnético para o estudo do campo magnético produzido por redes de alta tensão, descrito nas seguintes equações:

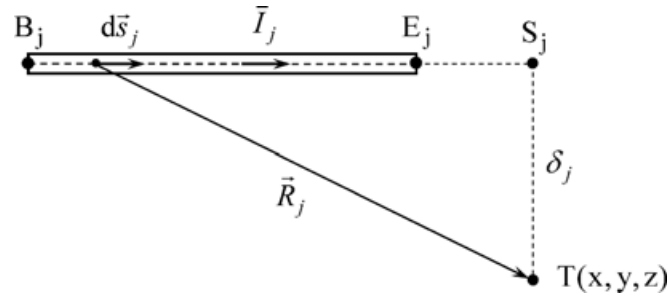


Figura 2.2 – Representação dos dados para o cálculo da densidade do fluxo magnético no ponto T.

$$a_j = x_{E_j} - x_{B_j}; \quad b_j = y_{E_j} - y_{B_j}; \quad c_j = z_{E_j} - z_{B_j}; \quad (\text{Eq.2.3.1})$$

$$l_j = \sqrt{a_j^2 + b_j^2 + c_j^2} \quad (\text{Eq.2.3.2})$$

$$Ri = \sqrt{(x - x_{B_j})^2 + (y - y_{B_j})^2 + (z - z_{B_j})^2} \quad (\text{Eq.2.3.3})$$

$$Rf = \sqrt{(x - x_{E_j})^2 + (y - y_{E_j})^2 + (z - z_{E_j})^2} \quad (\text{Eq.2.3.4})$$

$$t_j = \frac{a_j(x - x_{B_j}) + b_j(y - y_{B_j}) + c_j(z - z_{B_j})}{l_j^2} \quad (\text{Eq.2.3.5})$$

Sendo, x_{E_j} , y_{E_j} e z_{E_j} as coordenadas do ponto E_j ; x_{B_j} , y_{B_j} e z_{B_j} as coordenadas do ponto B_j ; a_j a distância entre x_{E_j} e x_{B_j} ; b_j a distância entre y_{E_j} e y_{B_j} ; c_j a distância entre z_{E_j} e z_{B_j} ; Ri distância do ponto B_j ao ponto T; Rf distância entre o ponto E_j e o ponto T; l_j o tamanho do segmento; t_j é uma variável para um cálculo auxiliar; x , y e z as

coordenadas do ponto onde vai ser calculado a densidade do fluxo do campo magnético, ponto T, ver figura 2.2.

Depois será necessário saber o ponto do segmento mais próximo do ponto T e saber a distância entre eles.

$$S_j(x_{S_j}, y_{S_j}, z_{S_j}) = S_j(x_{B_j} + a_j \cdot t_j, y_{B_j} + b_j \cdot t_j, z_{B_j} + c_j \cdot t_j) \quad (\text{Eq.2.3.6})$$

$$\delta_j = \sqrt{(x - x_{S_j})^2 + (y - y_{S_j})^2 + (z - z_{S_j})^2} \quad (\text{Eq.2.3.7})$$

S_j o ponto do segmento mais próximo do ponto T; x_{S_j} , y_{S_j} e z_{S_j} são as coordenadas x, y e z do ponto S_j ; δ_j é a distância entre os pontos S e T, ver figura 2.2.

Com as variáveis todas calculadas, já se poderá calcular as três componentes da densidade do fluxo magnético.

$$\vec{B}_{j_x} = \mu_0 \cdot \frac{\vec{I}}{4 \cdot \pi \cdot \delta_j^2} \cdot \left[\frac{t_j}{Ri} + \frac{1 - t_j}{Rf} \right] \cdot [(z - z_{S_j}) \cdot b_j - (y - y_{S_j}) \cdot c_j] \quad (\text{Eq.2.3.8})$$

$$\vec{B}_{j_y} = \mu_0 \cdot \frac{\vec{I}}{4 \cdot \pi \cdot \delta_j^2} \cdot \left[\frac{t_j}{Ri} + \frac{1 - t_j}{Rf} \right] \cdot [(x - x_{S_j}) \cdot c_j - (z - z_{S_j}) \cdot a_j] \quad (\text{Eq.2.3.9})$$

$$\vec{B}_{j_z} = \mu_0 \cdot \frac{\vec{I}}{4 \cdot \pi \cdot \delta_j^2} \cdot \left[\frac{t_j}{Ri} + \frac{1 - t_j}{Rf} \right] \cdot [(y - y_{S_j}) \cdot a_j - (x - x_{S_j}) \cdot b_j] \quad (\text{Eq.2.3.10})$$

A densidade do fluxo magnético resultante é dada pela seguinte equação.

$$B = \sqrt{|B_{j_x}|^2 + |B_{j_y}|^2 + |B_{j_z}|^2} \quad (\text{Eq.2.3.11})$$

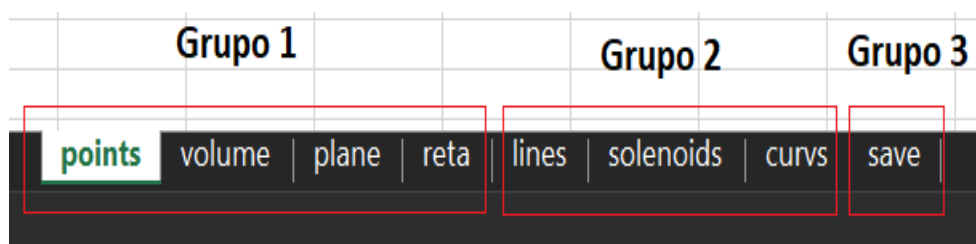
O algoritmo descrito neste paper foi o usado no software ‘EASYMAG’, para realizar o cálculo da densidade do fluxo magnético.

Capítulo 3 - Software desenvolvido

Neste capítulo é descrito o software ‘EASYMAG’ desenvolvido e o papel de cada função implementada. Cada função é apresentada de forma simplificada, para fácil compreensão e percepção do papel desempenhado por ela.

3.1 Easymag_Data_in

Para a introdução de dados é usado um ficheiro ‘Excel’ denominado ‘Easymag_Data_in’. Optou-se pelo ‘Excel’ porque permite guardar os dados para uso no futuro e facilita a sua introdução e alteração. Deve ser tida a atenção para os valores utilizados que deverão ser expressos em unidades SI, i.e., comprimentos em metros, os ângulos em graus e que a unidade de medida da densidade do fluxo magnético é o tesla.



Grupo 1				Grupo 2			Grupo 3
points	volume	plane	reta	lines	solenoids	curvs	save

Figura 3.1 - Folhas presentes no ficheiro excel denominado ‘Easymag_Data_in’.

As oito folhas presentes no ficheiro ‘Easymag_Data_in’ podem ser divididas em três grupos:

O grupo 1 representa os pontos onde a densidade do fluxo magnético é calculada de quatro formas diferentes (figura 3.1). É preciso ter em conta que no grupo 1, só é possível o uso de uma das formas, por consequência, ao correr o software tem que escolher uma das quatro, mesmo se mais de uma das quatro folhas se encontram preenchidas.

O grupo 2 corresponde à introdução de dados para a representação do circuito elétrico. O circuito elétrico pode ser descrito através de linhas, solenoides e curvas. Neste grupo existem três parâmetros iguais nos três elementos circuitais: ‘current’, ‘phase’ e ‘m’.

O parâmetro '*current*' corresponde ao valor da corrente que percorre o elemento de circuito, enquanto que o parâmetro '*phase*' como o próprio nome indica é a fase dessa mesma corrente, sendo que o parâmetro '*m*' é a direção desta podendo assumir os valores (1), assume a direção de '*P1*' para '*P2*' e (-1) de '*P2*' para '*P1*', sabendo que '*P1*' e '*P2*' são extremidades do elemento de circuito.

O grupo 3 serve para o utilizador guardar a simulação realizada.

3.2 Representação dos circuitos

3.2.1 Linhas

Este tipo de elemento de circuito permite considerar dois tipos de condutores elétricos: unifilares ou bifilares.

	A	B	C	D	E
1	line parameters #	I1	I2	I3	I4
2	P1x				
3	P1y				
4	P1z				
5	P2x				
6	P2y				
7	P2z				
8	current(A)				
9	m(1,-1)				
10	phase				
11	Bifilar(0,1)				
12	DVx				
13	DVy				
14	DVz				
15	Distance				

Figura 3.2 - Folha '*lines*' presente do ficheiro excel '*Easymag_Data_in*'.

No caso de um condutor unifilar, devem-se preencher os campos correspondentes aos pontos '*P1*' e '*P2*' para definição dos extremos da linha (segmento de condutor elétrico) e colocando o parâmetro '*Bifilar*' a (0) (figura 3.2).

No caso da representação bifilar, deve-se preencher os mesmos parâmetros que na unifilar, mas neste caso é preciso mais informação e o parâmetro '*Bifilar*' deve ser colocado a (1).

A linha bifilar é definida pelos pontos 'P1' e 'P2', por uma distância definida através do parâmetro 'Distance', sendo ainda necessário associar um vetor através do parâmetro '(DVx, DVy, DVz)' (figura 3.2).

	A	B	C	D	E	F	G
1	line parameters #	I1	I2	I3	I4	I5	I6
2	P1x	0	4	4	2	2	0
3	P1y	0	0	2	2	4	4
4	P1z	0	0	0	0	0	0
5	P2x	4	4	2	2	0	0
6	P2y	0	2	2	4	4	0
7	P2z	0	0	0	0	0	0
8	current(A)	14,1421	14,1421	14,1421	14,1421	14,1421	14,1421
9	m(1,-1)	1	1	1	1	1	1
10	phase	0	0	0	0	0	0
11	Bifilar(0,1)	1					
12	DVx	0	1	0	1	0	-1
13	DVy	-1	0	1	0	1	0
14	DVz	0	0	0	0	0	0
15	Distance	0,5	0,5	0,5	0,5	0,5	0,5

Figura 3.3 - Exemplo de um circuito fechado com seis linhas (troços) bifilares.

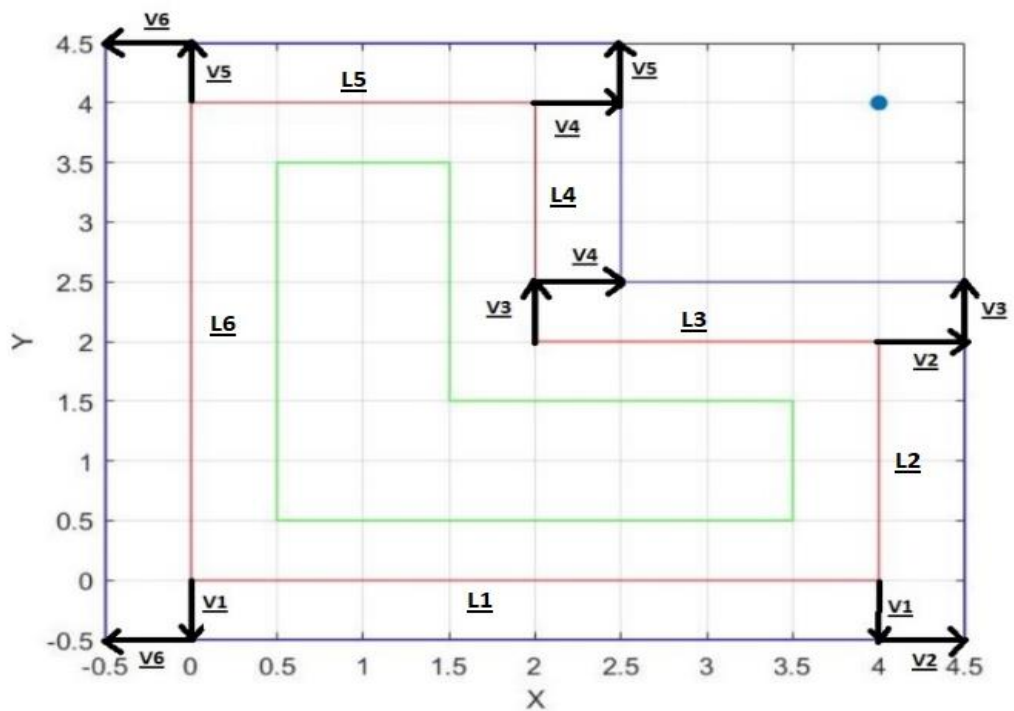


Figura 3.4 - Representação gráfica da linha bifilar definida na Fig. 3.3.

Nas figuras 3.3 e 3.4 mostra-se um exemplo de um circuito bifilar. A linha encarnada é a linha de referência (fictícia), definida pelos pontos 'P1' e 'P2'. A linha azul e a linha verde são as linhas que representam os condutores bifilares reais, e que distam um do outro o dobro do parâmetro '*Distance*'.

A definição da linha encarnada deve seguir uma sequência para que a representação da linha bifilar seja a correta. Para percepção de como funciona o algoritmo usado, é preciso pensar nessa sequência como uma malha.

O ponto inicial da linha azul é definido utilizando a soma de dois vetores e 'P1' como origem. Os dois vetores somados são o vetor da linha a ser representada com o da linha anterior, no caso da linha representada ser a inicial, o vetor usado é o vetor associado à última linha definida. O ponto final da linha azul é definido somando dois vetores e 'P2' como origem. Neste caso, os dois vetores usados são os da linha a ser representada com o da linha seguinte. Para a linha verde aplica-se o mesmo raciocínio, usando o vetor simétrico resultante da soma dos vetores, para a representação da linha azul.

Para cada linha é necessário que o utilizador escolha a amplitude, fase e direção da corrente, sabendo que no caso da linha bifilar, a fase e a direção são simétricas entre si.

Ter atenção que no caso da linha bifilar é obrigatório que o número de linhas seja superior a um, devido a serem necessários no mínimo dois vetores para a sua representação.

Relativamente às linhas unifilar e bifilar, estas apresentam cores diferentes, sendo que a unifilar é encarnada enquanto que a linha bifilar apresenta uma linha a verde e a outra a azul, facilitando assim a visualização e distinção (figura 3.4).

3.2.2 Solenoides

Ao desenvolver esta função é preciso ter em conta vários fatores. Identicamente ao que foi desenvolvido para as linhas, também aqui é pedido ao utilizador um ponto inicial e um ponto onde deve terminar.

Ao desenvolver-se o software, foi também tida em conta a rotação do solenoide. Para que ele não seja sempre representado ao utilizador segundo um eixo, utilizou-se a matriz de rotação. Para fazer a rotação, o software calcula, com a ajuda da função do Matlab '*vrotvec*', um vetor unitário que será usado como eixo de rotação e também um ângulo.

	A	B	C	D
1	Solenoids parameters #	S1	S2	S3
2	Initial point_x			
3	Initial point_y			
4	Initial point_z			
5	End point_x			
6	End point_y			
7	End point_z			
8	Turns			
9	Radius			
10	m(1,-1)			
11	current(A)			
12	wound direction(1,-1)			
13	Number of points			
14	current phase			

Figura 3.5 - Folha 'solenoids' presente do ficheiro excel 'Easymag_Data_in'.

Esta função utilizada tem como parâmetros de entrada dois vetores. Um deles é sempre o vetor do eixo z (0,0,1), e o outro é o vetor formado pelos dois pontos fornecidos pelo utilizador, eixo do solenoide. Com estes dois parâmetros a função 'vrrotvec' vai calcular um vetor unitário que servirá como o eixo de rotação, e um ângulo necessário para rodar o eixo z até ao eixo do solenoide.

Mesmo utilizando a matriz e a função atrás referida, o resultado final não seria o desejado se o ponto inicial fosse diferente do ponto (0,0,0). Para resolver esta situação, antes da rotação subtraiu-se o ponto inicial ao conjunto de pontos do solenoide, colocando-o na posição (0,0,0). A rotação só é possível se o segmento estiver localizado na origem dos eixos, tendo por isso subtraído o ponto inicial antes da rotação, sendo depois somado quando a rotação tiver sido concluída para que a rotação seja bem sucedida.

O parâmetro 'Turns' corresponde ao número de espiras do solenoide, o parâmetro 'Radius' é o raio do solenoide para achar o centro do solenoide basta subtrair na coordenada x do ponto inicial o raio (x_i-R , y_i , z_i), sendo estas as coordenadas no centro do solenoide. O parâmetro 'wound direction' corresponde à direção do enrolamento do solenoide, (-1) sentido horário e (1) anti-horário.

3.2.3 Curvas

	A	B	C	D
1	curvs parameters #	c1	c2	c3
2	P1x			
3	P1y			
4	P1z			
5	Pcx			
6	Pcy			
7	Pcz			
8	vx			
9	vy			
10	vz			
11	current(A)			
12	Number of points			
13	clock(1,-1)			
14	theta			
15	m(1,-1)			
16	phase			

Figura 3.6 - Folha 'curvs' presente do ficheiro excel 'Easymag_Data_in'.

A forma da implementação das curvas no software foi muito semelhante ao trabalho desenvolvido para o solenoide. O procedimento adotado para fazer a rotação foi semelhante, porém com um ponto de divergência relativamente à curva, pois o segundo vetor para o cálculo da rotação é o vetor fornecido pelo utilizador, que serve para definir o plano onde a curva estará localizada.

No entanto, é preciso ter em conta que a maneira de definir a curva é diferente. Para desenhar tem de ser fornecida a seguinte informação: dois pontos (o ponto inicial da curva e o centro da mesma), um vetor (v_x, v_y, v_z), um ângulo '*theta*' e a direção da curvatura '*clock*'.

Tendo em conta que o plano será formado pelo ponto que representa o centro da curva juntamente com o vetor dado pelo utilizador, o ponto inicial tem que ser um ponto que pertença ao plano criado para que seja possível a sua representação.

No caso de o ponto inicial não pertencer ao plano definido pelo utilizador será apresentada a seguinte mensagem de erro: '*The first point is outside the plane*'. Seguidamente a execução do programa será interrompido.

Relativamente ao ângulo, este representa o local onde a curva irá terminar. Por conseguinte, a curva vai começar no ponto inicial e parar após completar o ângulo fornecido pelo utilizador. É também necessário fornecer ao software a introdução do sentido da curvatura que será dado pelo parâmetro '*clock*'. Podem ser assumidos apenas

os valores um (1) ou menos um (-1). Sugere-se que seja assumido o valor (1) para o sentido anti-horário e (-1) para o sentido horário.

Com a representação dos três tipos de elementos circuitais concluída, a próxima tarefa foi criar uma estrutura de dados capaz de armazenar os dados para o cálculo da densidade do fluxo magnético.

3.3 Estrutura de dados

A estrutura de dados utilizada é semelhante à do software colocado na *'Mathworks' 'biot savart toolbox'*, que foi desenvolvido por Loic Quéval [4]. Desta forma, foram criadas duas funções para carregar os dados necessários para o cálculo da densidade do fluxo magnético: *'Easymag_init'* e *'Easymag_add_segment'*, que serão detalhadas em seguida.

3.3.1 Easymag_init

Esta função inicia o campo 'número de elementos circuitais' da estrutura de dados (chamada Easymag) onde o número de elementos circuitais utilizados é carregado.

3.3.2 Easymag_add_segment

Nesta função são carregados na estrutura de dados *'EASYMAG'* todos os dados necessários para o cálculo da densidade de fluxo magnético correspondentes a cada elemento de circuito. São guardadas as coordenadas pertencentes ao elemento de circuito, a corrente que o percorre e a direção dessa corrente.

Depois da estrutura de dados estar definida, a tarefa seguinte passará a ser a representação dos pontos onde se quer calcular a densidade de fluxo magnético. Este cálculo pode ser feito por quatro saídas possíveis: por pontos, volume, plano e reta. Tendo em consideração que só é possível escolher uma saída.

3.4 Lista dos resultados pretendidos

A localização dos pontos onde se calcula a densidade do fluxo magnético pode ser definida de quatro formas alternativas:

3.4.1 Pontos

	A	B	C	D	E
1	output Points #	P1	P2	P3	P4
2	Px	5	-45	-45	-30
3	Py	5	0	0	0
4	Pz	10	1	19	10
5	mark scale (100~1000)	500	500	500	500

Figura 3.7 - Folha 'points' presente do ficheiro excel 'Easymag_Data_in'.

Nesta funcionalidade disponibilizada, o utilizador insere três coordenadas (x , y e z) que correspondem a cada ponto que queira analisar, podendo escolher o tamanho dos pontos através de o parâmetro 'mark scale' (figura 3.7). Contrariamente às outras opções disponibilizadas (volume, reta e plano), esta opção permite ao utilizador escolher mais que um caso, não havendo limite de pontos introduzidos.

Na figura 3.8 é demonstrado os pontos da figura 3.7 com a cor azul.

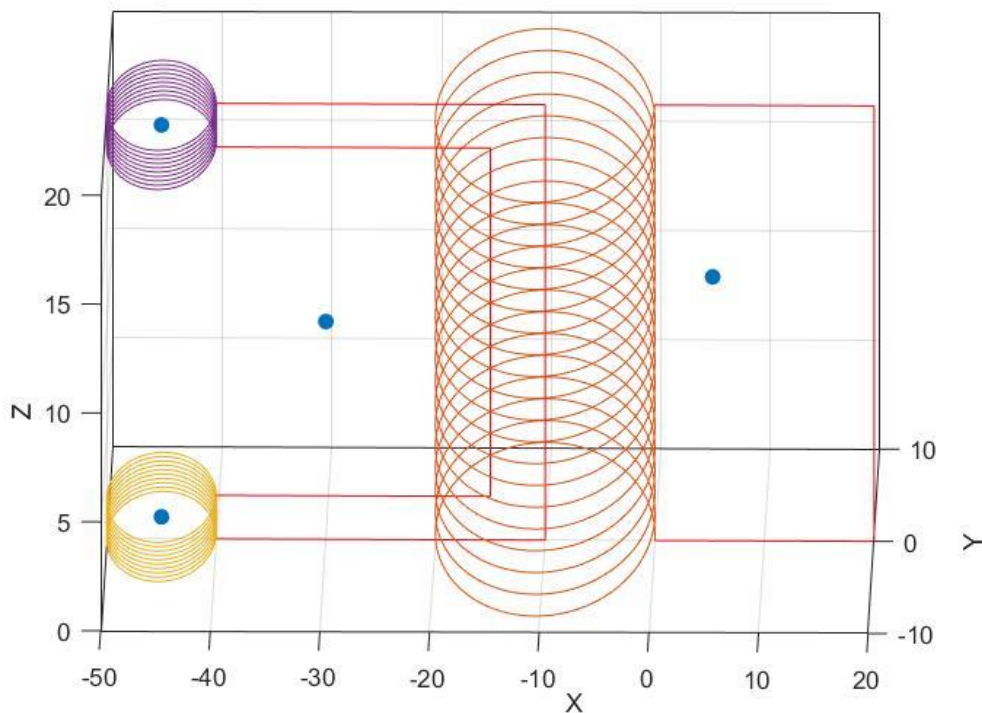


Figura 3.8 - Output pontos.

3.4.2 Volume

	A	B
1	output view volume parameters	values
2	X_MIN	-25
3	X_MAX	-35
4	Y_MIN	-5
5	Y_MAX	5
6	Z_MIN	2
7	Z_MAX	18
8	Num_x (number of points X axis)	10
9	Num_y (number of points Y axis)	10
10	Num_z (number of points Z axis)	20
11	mark scale (100~1000)	500

Figura 3.9 - Folha 'volume' presente do ficheiro excel 'Easymag_Data_in'.

Esta funcionalidade foi desenvolvida usando a função de Matlab 'meshgrid', por conseguinte, o utilizador tem que definir limites máximos e mínimos nos três eixos.

O utilizador escolhe o número de pontos de cada eixo, formando assim uma matriz tridimensional $'Numx' \times 'Numy' \times 'Numz'$.

Deve assumir-se que $'Numx'$ é o número de pontos no eixo x ; $'Numy'$ número de pontos do eixo y e $'Numz'$ o número de pontos do eixo z .

3.4.3 Plano

	A	B
1	output view plane parameters	values
2	Px	-35
3	Py	0
4	Pz	5
5	Vector_x	0
6	Vector_y	0
7	Vector_z	1
8	X_MIN	-25
9	X_MAX	-35
10	Y_MIN	-5
11	Y_MAX	5,00E+00
12	Num_x (number of points X axis)	10
13	Num_y(number of points Y axis)	10

Figura 3.10 - Folha *'plane'* presente do ficheiro excel *'Easymag_Data_in'*.

Para representação e definição do plano o utilizador tem de fornecer um ponto e um vetor. Após a sua definição devem ser definidos os limites e o número de pontos do eixo x ($'Numx'$) e do eixo y ($'Numy'$) que o vai representar (figura 3.10).

Depois de o utilizador ter introduzido os dados, o plano vai ser desenhado usando dois ciclos *'FOR'* que têm em conta os limites e o número de pontos fornecidos em cada eixo, e caso necessário efetua a rotação do plano.

Na Figura 3.11 é demonstrado um exemplo da funcionalidade sendo representado o plano descrito na Figura 3.10.

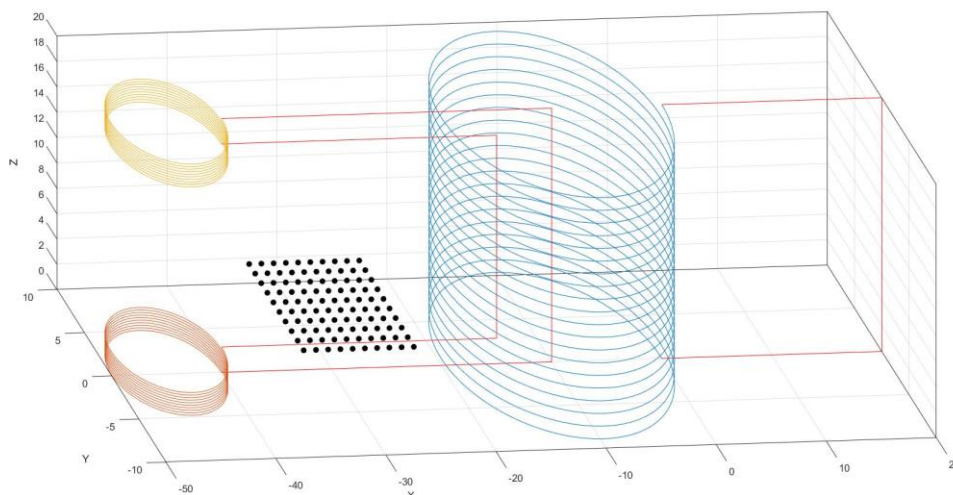


Figura 3.11 - Output plano.

3.4.3.1 Rotação do plano

Inicialmente, o plano será sempre representado pelo plano xoy , e, por esse motivo, apenas são solicitados os limites segundo o eixo x e o eixo y .

A rotação será calculada utilizando a função do Matlab `'vrrotvec'`, esta função vai fornecer o vetor unitário que define o eixo de rotação e vai fornecer o ângulo necessário para a rotação.

O vetor unitário e o ângulo terão sempre em conta a rotação do eixo z , até onde utilizador desejar, sabendo que a rotação será realizada no sentido anti-horário. Com os dados fornecidos pela função `'vrrotvec'`, o vetor e o ângulo é usada a matriz de rotação para que se possa representar o plano pretendido pelo utilizador.

Em seguida é apresentado um exemplo da rotação a ser feita do eixo z para o eixo y e, por consequência, do vetor $[0,0,1]$ para o vetor $[0,1,0]$.

Na Figura 3.12 e 3.13 é demonstrado a rotação do eixo z para o eixo y .

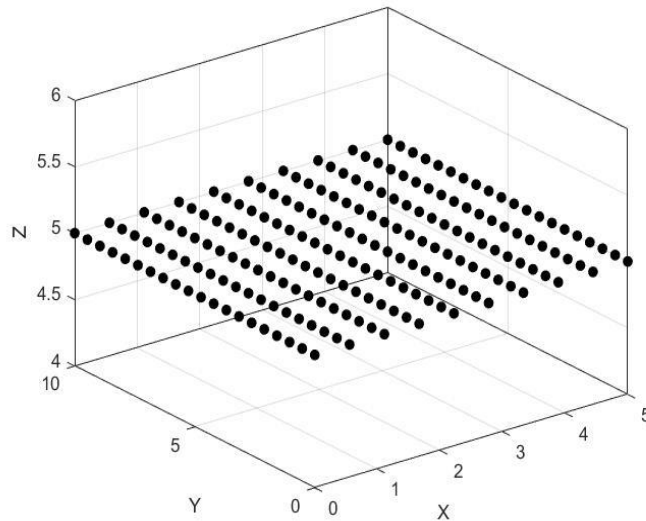


Figura 3.12 - Plano antes da rotação ser realizada.

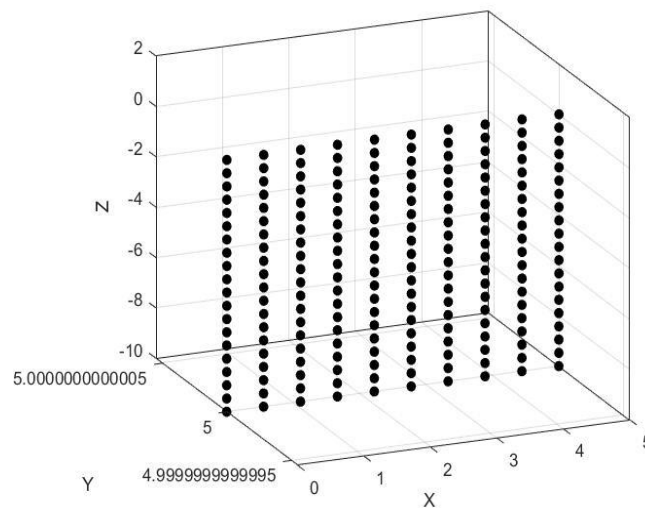


Figura 3.13 - Plano depois realização da rotação.

Para se efetuar a rotação é necessário, conforme já foi referido, um vetor unitário.

No exemplo apresentado o vetor do eixo x $(1,0,0)$ serve como eixo de rotação, tendo o vetor do eixo y efetuado uma rotação a 90° até ao eixo z .

Desta feita, o vetor do eixo y passa a estar localizado na posição do eixo z $(0,0,1)$ e o vetor do eixo z passa a ser o simétrico do antigo vetor do eixo y $(0,-1,0)$.

A Figura 3.14 ilustra esta situação, para uma melhor compreensão do que foi acima descrito.

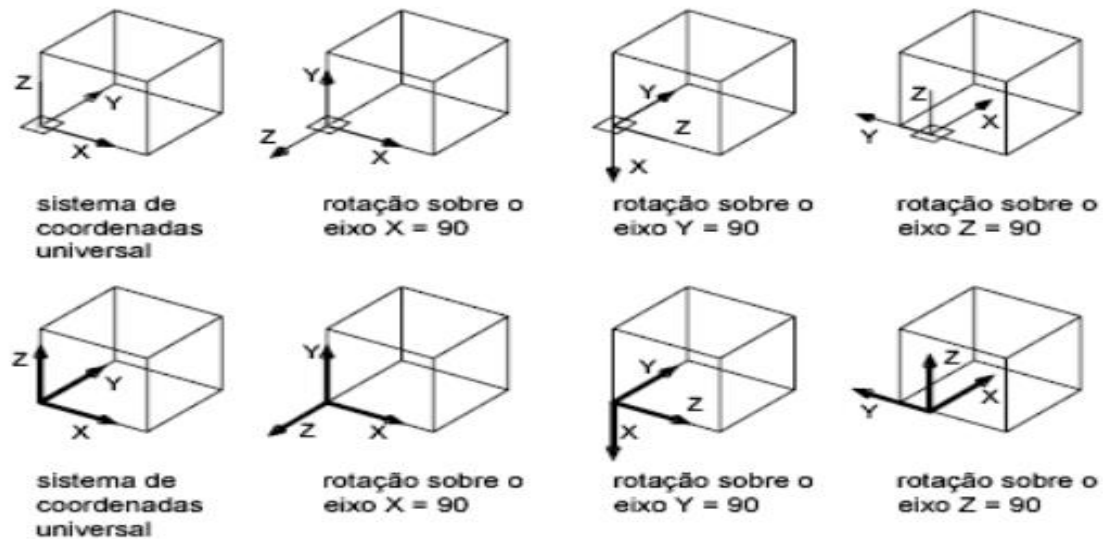


Figura 3.14 - Representação da rotação dos três eixos [5].

3.4.4 Reta

	A	B
1	output segment parameters	values
2	P1x (blue mark)	-35
3	P1y (blue mark)	0
4	P1z (blue mark)	5
5	P2x (red mark)	-35
6	P2y (red mark)	0
7	P2z (red mark)	17
8	Number of points	10
9	B(T) scale limit (MIN)	0,00E+00
10	B(T) scale limit (MAX)	0,00E+00

Figura 3.15 - Folha 'reta' presente do ficheiro excel 'Easymag_Data_in'.

Esta funcionalidade é definida por dois pontos, um ponto inicial de cor azul e um ponto final de cor vermelha. O utilizador tem que escolher o número de pontos entre os dois pontos referidos. Nesta representação o utilizador pode escolher os limites da escala da densidade do fluxo magnético para melhor observação, se estes parâmetros forem colocados a zero a escala é definida pelo Matlab, ver figura 3.15.

Na Figura 3.16 pode ser observado um circuito com três enrolamentos, sendo a reta ao longo da qual é obtida a medição do campo B pelos pontos a azul. Neste caso esta reta corresponde a reta definida na Figura 3.15.

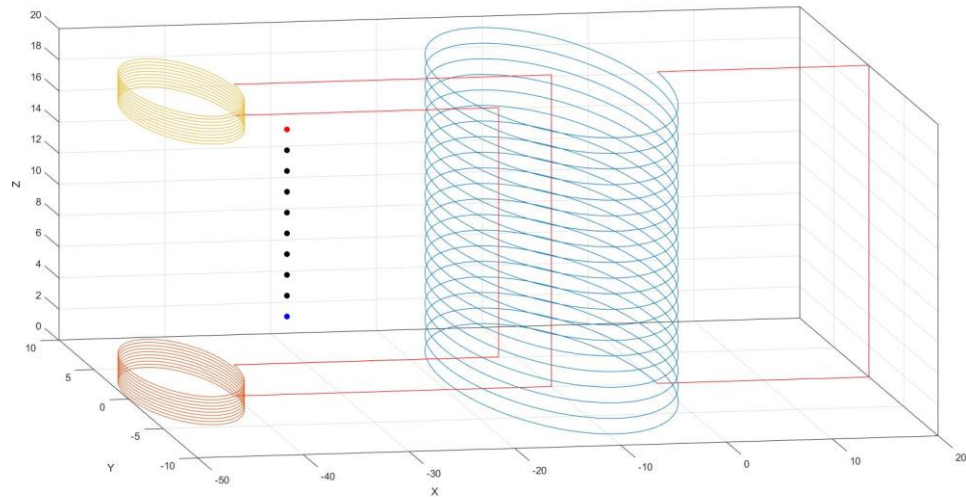


Figura 3.16 - Output reta.

3.5 Easymag_get_B

Para aplicar a lei de Biot-Savart, em Matlab, foi criada a função '*Easymag_get_B*'.

Esta função calcula a densidade do fluxo magnético, dividindo os vários tipos de elementos circuitais que o utilizador escolheu em pequenos segmentos de reta, exceto as linhas que são consideradas segmentos de reta.

Depois de divididos os elementos circuitais, será calculada a contribuição de cada segmento de reta para a densidade do fluxo magnético, segundo cada eixo, utilizando as equações (Eq.2.3.8), (Eq.2.3.9) e (Eq.2.3.10).

3.6 Verificação dos dados

Esta função foi criada para descobrir se o circuito está aberto ou fechado, e descobrir se os dados foram corretamente colocados. Foi-lhe atribuída a denominação '*Easymag_circuitos*'.

O processo vai dividir os pontos iniciais e os pontos finais dos elementos circuitais em duas tabelas diferentes. Depois irá comparar o ponto inicial com todos os pontos finais, se existir algum igual, esses dois pontos serão eliminados das duas tabelas.

Se as tabelas estiverem vazias no final do processamento os circuitos estão fechados. Caso contrário, o número de pontos que sobraram na tabela de pontos iniciais, corresponde ao número de circuitos que ficaram em aberto.

3.7 Outputs

Esta função foi criada para a representação gráfica do circuito elétrico e da densidade do fluxo magnético. Foi-lhe atribuída a denominação '*Easymag_plot_Field*'.

Dependendo do '*output*' escolhido pelo utilizador, a representação gráfica da densidade de fluxo magnético é feita por diversas formas: por pontos, volume, plano e reta.

3.7.1 Pontos

A função '*plot3*' cria um gráfico a três dimensões, permitindo uma visualização dos elementos circuitais num ambiente a três dimensões.

Inicialmente, para esta representação, houve uma tentativa de utilizar a função '*plot3*' do Matlab, porém verificou-se que a visualização não era ideal por isso optou-se por utilizar a função *bar*.

A função *bar* representa os dados usando um gráfico de barras.

Esta função permitiu uma melhor análise dos dados, porém em contrapartida foram detetadas duas situações que tiveram de ser corrigidas. As mesmas consistiam na não visualização de todos os dados e a falta da legenda. Para ultrapassar estas situações as alterações efetuadas foram: a colocação de um ponto extra com valor '*NaN*' (*Not a Number*), e, em caso de alguma componente da densidade de fluxo magnético ser zero, substitui-la por '*NaN*'.

Na Figura 3.17 é apresentado um exemplo da representação que pode ser obtida.

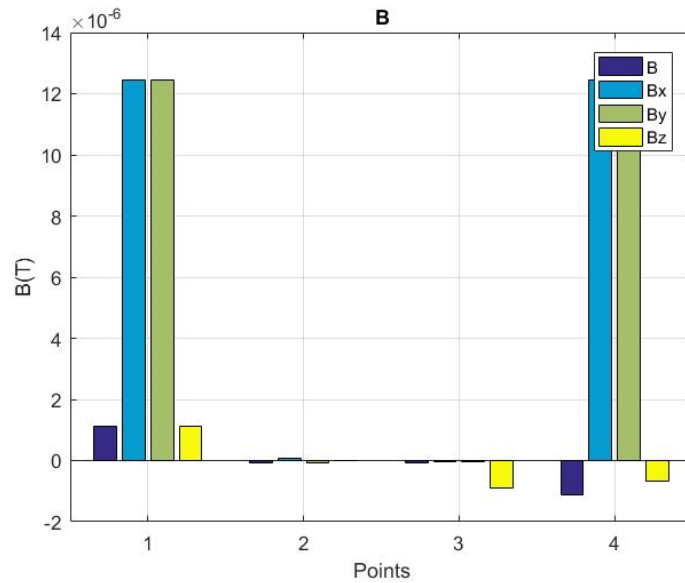


Figura 3.17 - Representação da densidade do fluxo magnético.

3.7.2 Volume

Neste caso de estudo a primeira abordagem foi dividir o volume em vários planos, utilizando a função *'surf'*. No resultado obtido ocorreu uma sobreposição gráfica que impedia assim uma boa análise dos resultados.

A função *'surf'* cria um gráfico a três dimensões usando uma coloração diferente numa superfície consoante a densidade do fluxo magnético na localização pretendida.

Como a função *'surf'* deixou de ser uma opção válida escolheu-se a função *'scatter3'*. A função *'scatter3'* muda a coloração dos pontos em análise consoante a densidade de fluxo magnético a três dimensões, facilitando a análise e visualização gráfica e permitindo a junção também do circuito elétrico na mesma figura. Tendo como resultado final quatro figuras, três correspondentes à densidade do fluxo magnético segundo cada componente e uma figura da densidade do fluxo magnético resultante, ver figuras.

Nas Figuras 3.18, 3.19, 3.20 e 3.21 são apresentados exemplos das representações que podem ser obtidas.

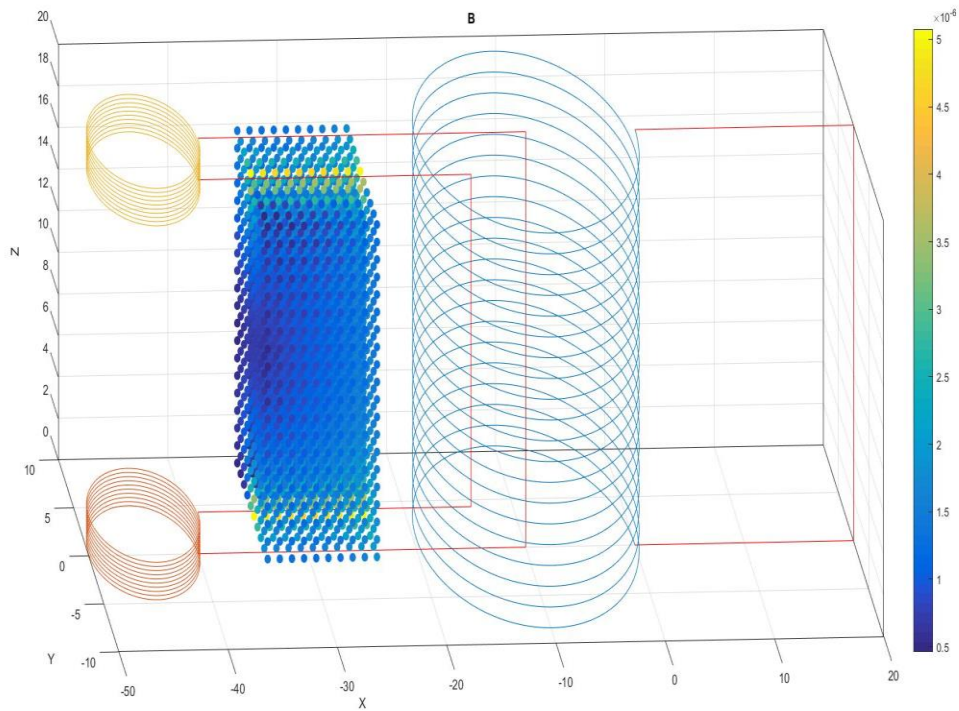


Figura 3.18 - Representação da densidade do fluxo magnético resultante.

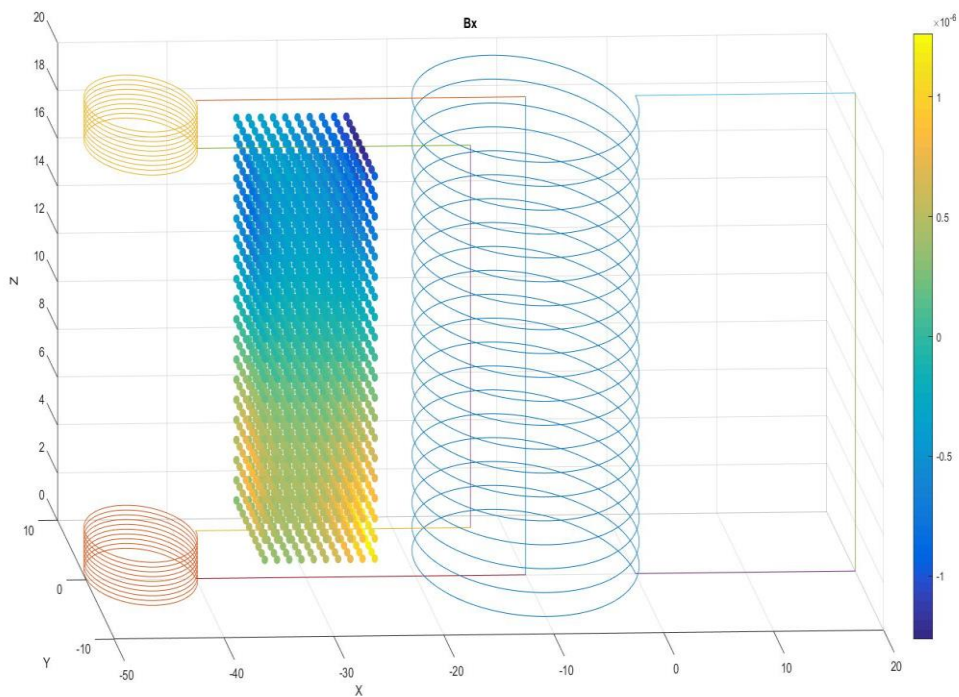


Figura 3.19 - Representação da densidade do fluxo magnético segundo a componente do eixo x.

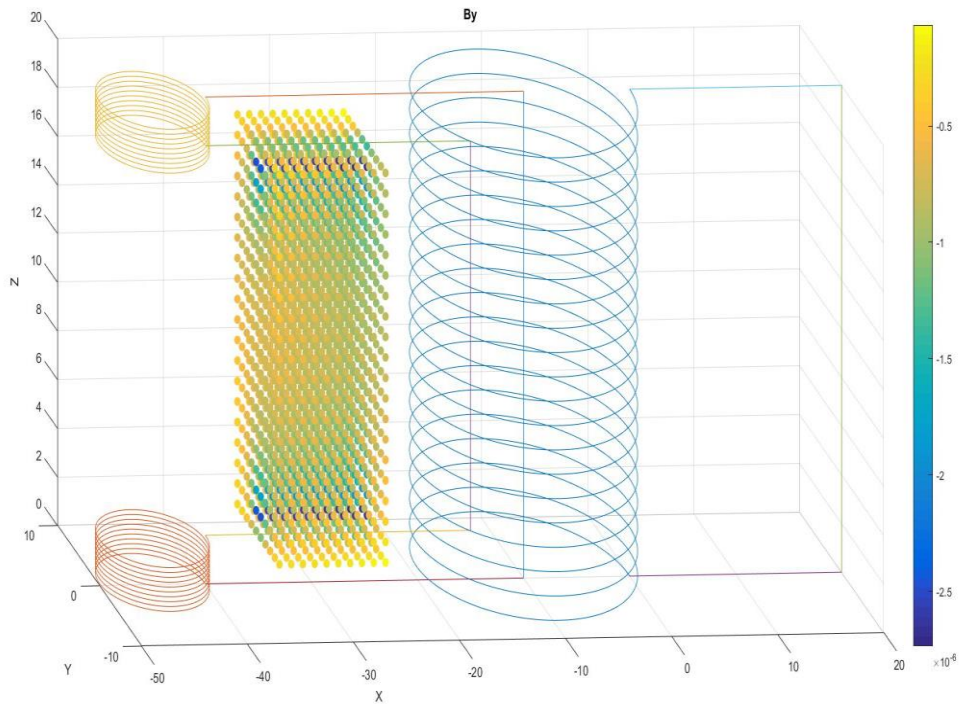


Figura 3.20 – Representação da densidade do fluxo magnético segundo a componente do eixo y.

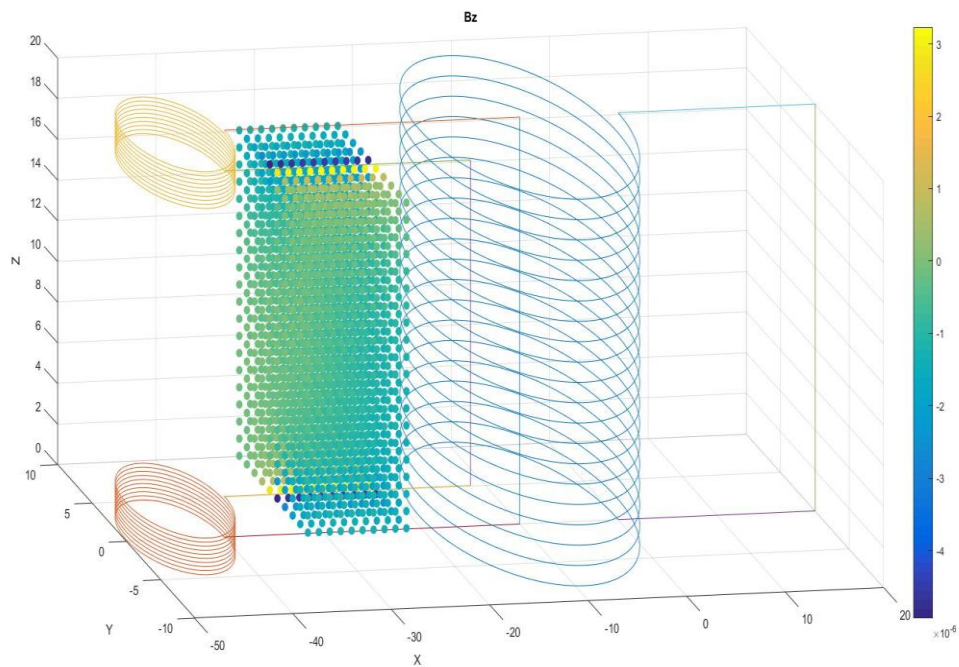


Figura 3.21 – Representação da densidade do fluxo magnético segundo a componente do eixo z.

3.7.3 Plano

No plano optou-se por utilizar a função *'surf'*. Como neste caso só há um plano e não vários, não existirá o problema de sobreposição que existia no volume, logo permitindo assim uma boa análise gráfica. Utilizando assim quatro figuras como no volume para a representação da densidade do fluxo magnético.

Nas Figuras 3.22, 3.23, 3.24 e 3.25 são apresentados exemplos das representações que podem ser obtidas, através do plano definido na Figura 3.11.

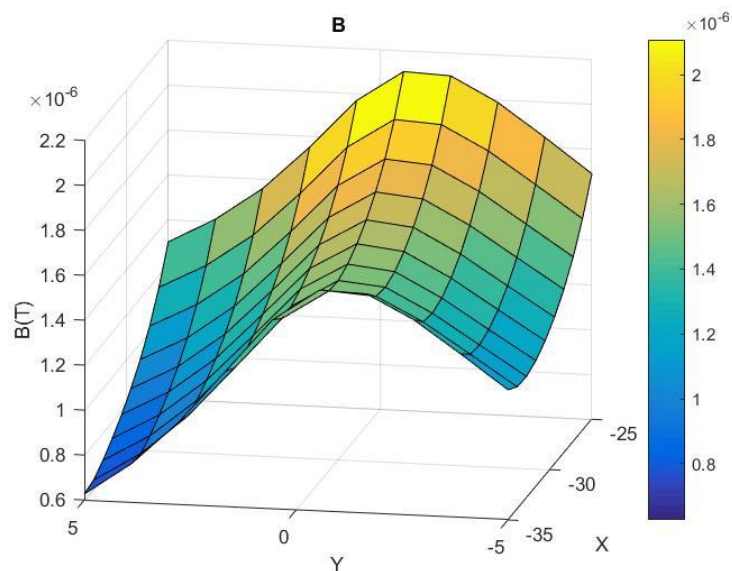


Figura 3.22 - Representação da densidade do fluxo magnético resultante.

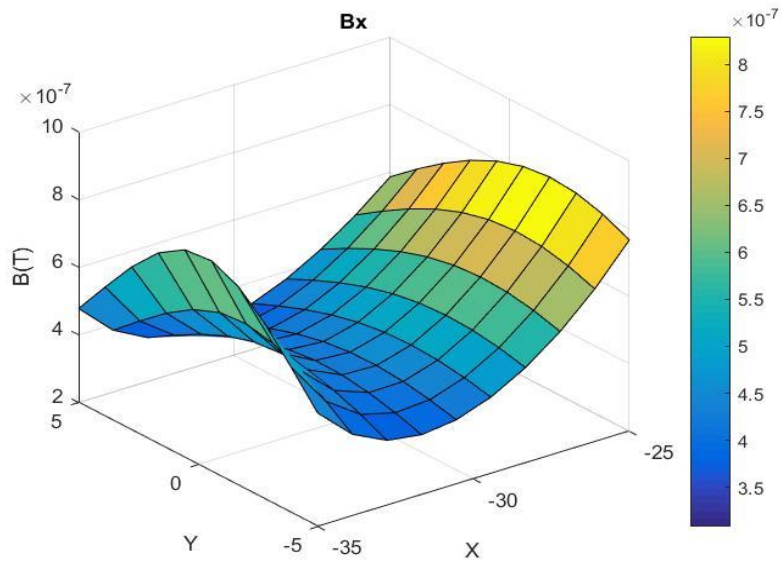


Figura 3.23 - Representação da densidade do fluxo magnético segundo a componente do eixo x.

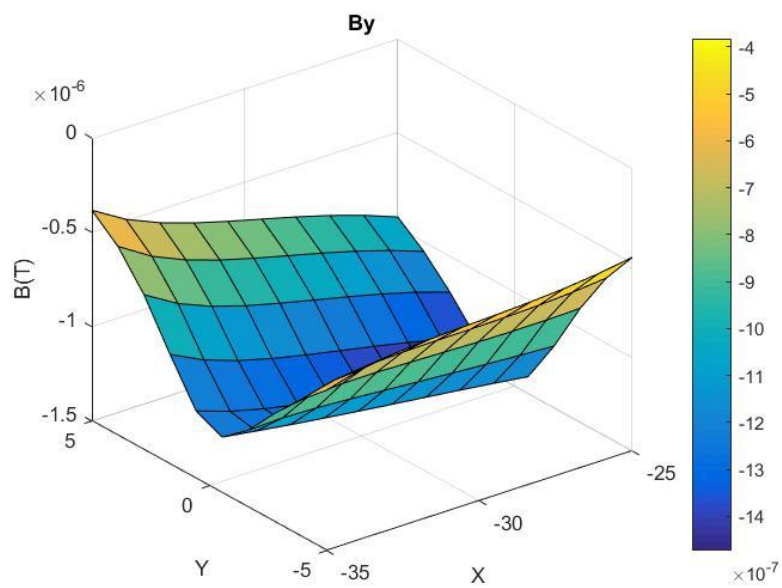


Figura 3.24 - Representação da densidade do fluxo magnético segundo a componente do eixo y.

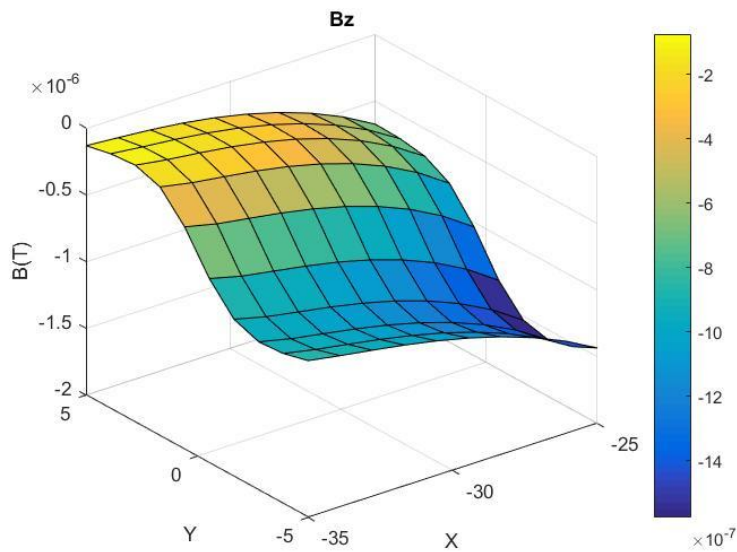


Figura 3.25 - Representação da densidade do fluxo magnético segundo a componente do eixo z.

3.7.4 Reta

Na reta optou-se por usar o *'plot'* em vez do *'plot3'* para que a análise seja facilitada, tendo em conta que cada componente da densidade do fluxo magnético irá ter um traçado e cor diferente para facilitar a visualização.

A função *'plot'* é igual à função *'plot3'*, mas para gráficos a duas dimensões.

Na Figura 3.26 é apresentado um exemplo da representação que pode ser obtida.

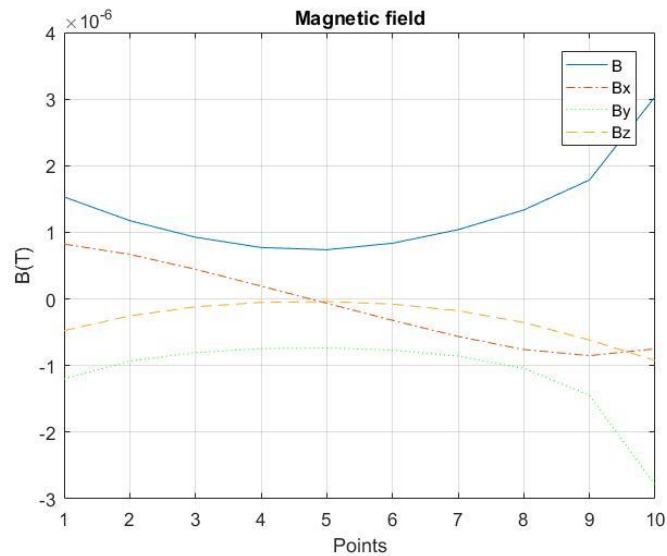


Figura 3.26 - Representação da densidade do fluxo magnético.

3.8 Easymag_save

	A	B
1	file#	
2	save output(0,1)	0
3	Filename	curv

Figura 3.27 - Folha 'save' presente do ficheiro excel 'Easymag_Data_in'.

Esta função foi criada para o caso de o utilizador querer guardar a simulação. Foi-lhe atribuída a denominação 'Easymag_save'.

O desenho da mesma foi feito com base na seguinte premissa: na folha Excel 'save' caso o utilizador queira guardar deve colocar à frente da opção 'save' o valor (1), caso contrário (0). Caso opte por guardar deve ainda indicar o nome desejado na opção 'File Name'. Caso exista um ficheiro com o mesmo nome, será substituído o ficheiro existente pelo novo ficheiro.

3.9 Main

Esta função serve para chamar todas as funções do software e também para o cálculo do máximo e do mínimo da densidade de fluxo magnético resultante descobrindo ainda a posição onde estes se verificam, mostrando-os na consola do Matlab.

Capítulo 4 - Validação do código

Neste capítulo vamos comprovar a precisão do algoritmo usado através dos seguintes exemplos: Linhas, Curvas e Solenoides.

4.1 Linhas

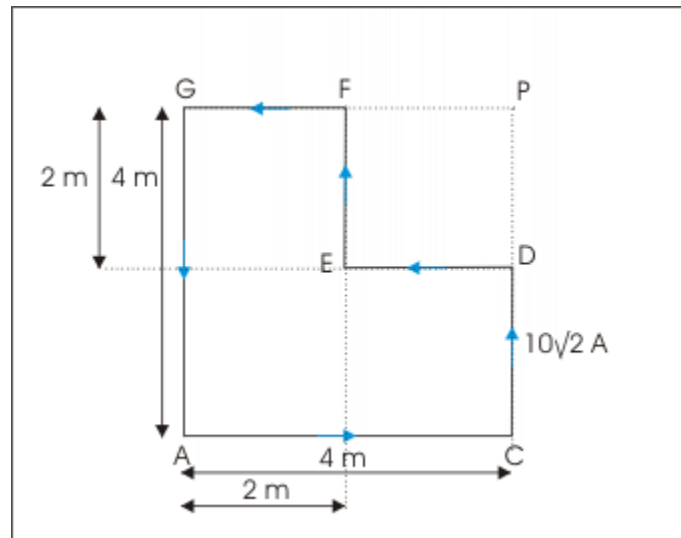


Figura 4.1 - Exemplo teórico.

Este exemplo consistirá no cálculo da densidade do fluxo magnético no ponto P considerando a contribuição das seis linhas representadas na figura 4.1 tendo em conta que o valor da corrente é de $10\sqrt{2}$ A. O ponto P encontra-se no mesmo plano que as seis linhas a ser consideradas, simplificando assim o cálculo da densidade do fluxo magnético. A densidade do fluxo magnético pode então ser descrita pelas seguintes equações [6].

$$B = \frac{\mu_0 \times I}{4 \times \pi \times L} \times (\sin(\theta_1) + \sin(\theta_2)) \quad (\text{Eq.4.1.1})$$

$$B_{AC} = \frac{\sqrt{2} \times \mu_0 \times I}{8 \times \pi \times L} = \frac{\sqrt{2} \times 4 \times \pi \times 10^{-7} \times \sqrt{2} \times 10}{8 \times \pi \times 4} \approx 2,5 \times 10^{-7} \text{T} \quad (\text{Eq.4.1.2})$$

$$B_{CD} = 0 \text{ T} \quad (\text{Eq.4.1.3})$$

$$B_{DE} = -\frac{\sqrt{2} \times \mu_0 \times I}{8 \times \pi \times L} = -\frac{\sqrt{2} \times 4 \times \pi \times 10^{-7} \times \sqrt{2} \times 10}{8 \times \pi \times 2} \approx -5 \times 10^{-7} \text{T} \quad (\text{Eq.4.1.4})$$

$$B_{EF} = -\frac{\sqrt{2} \times \mu_0 \times I}{8 \times \pi \times L} = -\frac{\sqrt{2} \times 4 \times \pi \times 10^{-7} \times \sqrt{2} \times 10}{8 \times \pi \times 2} \approx -5 \times 10^{-7} \text{T} \quad (\text{Eq.4.1.5})$$

$$B_{FG} = 0 \text{T} \quad (\text{Eq.4.1.6})$$

$$B_{CA} = \frac{\sqrt{2} \times \mu_0 \times I}{8 \times \pi \times L} = \frac{\sqrt{2} \times 4 \times \pi \times 10^{-7} \times \sqrt{2} \times 10}{8 \times \pi \times 4} \approx 2,5 \times 10^{-7} \text{T} \quad (\text{Eq.4.1.7})$$

Resultado final:

$$B = B_{AC} + B_{CD} + B_{DE} + B_{EF} + B_{FG} + B_{GA} \approx -5 \times 10^{-7} \text{T} \quad (\text{Eq.4.1.8})$$

I corresponde à corrente que percorre a linha; μ_0 é a permeabilidade magnética no vazio; L distância entre o ponto P e o ponto mais próximo da linha do ponto P .

Fazendo a representação usando o software criado sendo a sua representação segundo o eixo xoy , estando o ponto P (4,4,0).

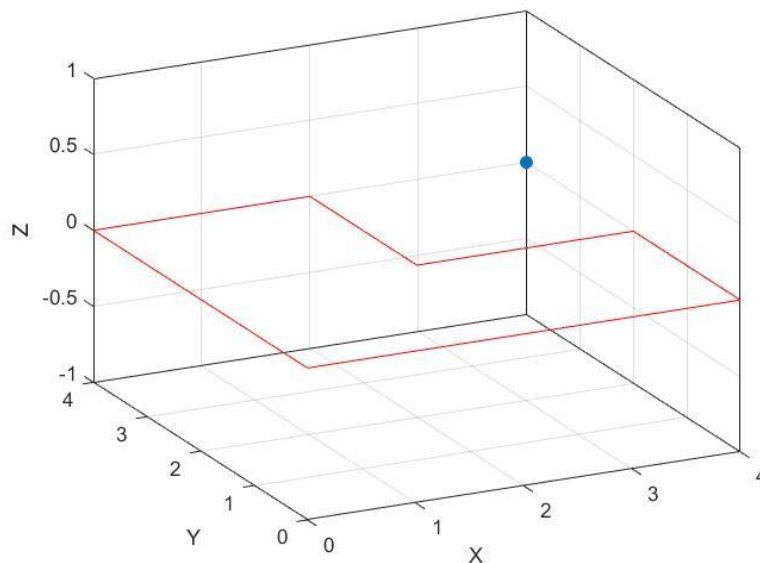


Figura 4.2 - Exemplo teórico usando o software desenvolvido.

Conforme pretendemos demonstrar, a figura 4.2 é igual à figura 4.1. Há que ter em conta que o ponto azul será neste caso o ponto P .

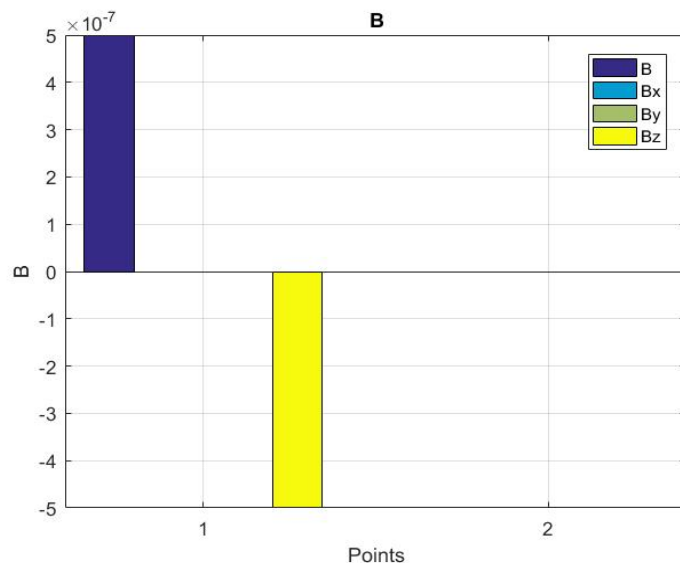


Figura 4.3 - Resultado do exemplo teórico usando o software.

Podemos através da figura 4.3 confirmar os resultados, sabendo que o resultado final se pode ver na equação 4.1.8.

Na figura 4.3 podemos ver que a componente 'Bz' da densidade do fluxo magnético corresponde ao resultado final, e 'B' corresponde à soma do módulo de cada componente por isso o 'B' neste caso, será o inverso de 'Bz'.

4.2 Curva

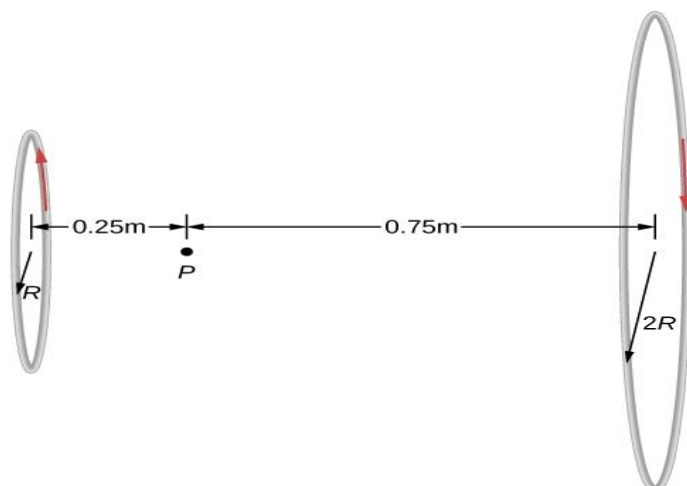


Figura 4.4 - Exemplo teórico.

No exemplo apresentado na figura acima, observam-se duas circunferências representadas segundo o plano xoz , com $R=0,5m$: as circunferências são percorridas por correntes com sentidos opostos com amplitude de 10 mA ambas [7].

No caso de ser uma circunferência é dada a opção ao utilizador para escolher o número de pontos que constituem a circunferência.

Em Matlab não é possível representar uma curva perfeita, logo, a curva será dependente do número de pontos dados. O programa vai ligá-los com uma linha, por conseguinte existirá um erro associado dependendo do número de pontos dados. Deve ter-se em conta que quanto maior o número de pontos menor o erro.

Para diminuir este erro foram usadas duas aproximações: pelo perímetro e pela área. Estas duas abordagens consistiam em calcular um novo raio que fizesse a compensação, dependendo da diferença de áreas ou perímetro da figura real, com a que é representada no Matlab.

Nestas duas abordagens verificou-se que a mais eficaz seria a da área.

O novo raio passou a ser definido por:

$$ac = \frac{2 \times \pi}{dGamma - 1} \quad (\text{Eq.4.2.1})$$

$$l = 2 \times R \times \sin \frac{ac}{2} \quad (\text{Eq.4.2.2})$$

$$A = \pi \times R^2 \quad (\text{Eq.4.2.3})$$

$$R = \frac{A \times 2}{l \times \cos \frac{ac}{2} \times (dGamma - 1)} \quad (\text{Eq.4.2.4})$$

ac corresponde ao ângulo central do triângulo da divisão feita pelo o número de pontos dados; l a base do triângulo; A área da circunferência; R é novo raio da circunferência; $dGamma$ número de pontos que representa a circunferência.

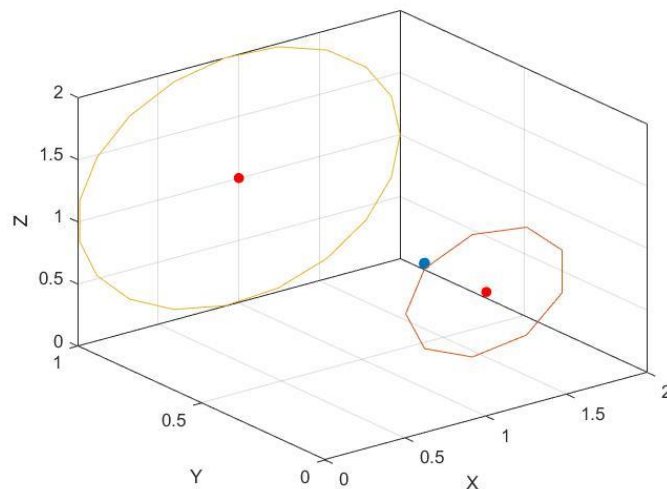


Figura 4.5 - Exemplo teórico representado usando software desenvolvido.

Representada a circunferência de raio 0,5m com dez pontos e a outra com vinte pontos os resultados foram, (figura 4.5):

Resultado teórico:

$$B = \frac{\mu_0 \cdot I \cdot R_1^2}{2(y_1^2 + R_2^2)^{\frac{3}{2}}} - \frac{\mu_0 \cdot I \cdot R_2^2}{2(y_2^2 + R_2^2)^{\frac{3}{2}}} \quad (\text{Eq.4.2.5})$$

$$B = \frac{4 \times \pi \times 10^{-7} \times (0,01) \cdot 0,5^2}{2(0,25^2 + 0,5^2)^{\frac{3}{2}}} - \frac{4 \times \pi \times 10^{-7} \times (0,01) \cdot 1^2}{2(0,75^2 + 1^2)^{\frac{3}{2}}} \quad (\text{Eq.4.2.6})$$

$$\Leftrightarrow B = 5.77477 \times 10^{-9}\text{T}$$

Na figura 4.6 pode ser observado o resultado obtido sem aplicação do algoritmo para correção do erro de aproximação da curva.

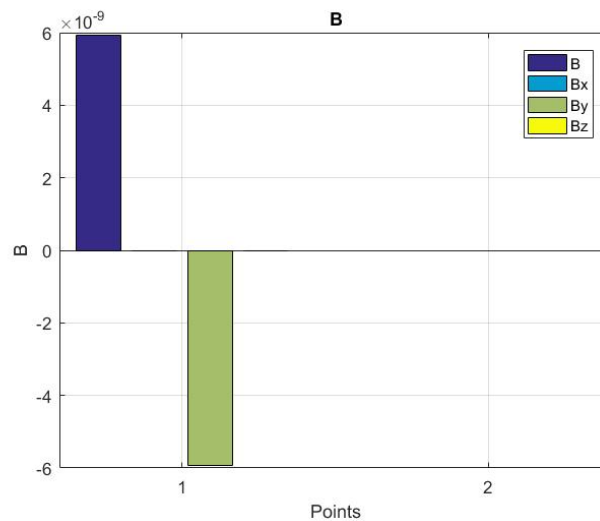


Figura 4.6 - Solução sem algoritmo.

$$B = 5.91929 \times 10^{-9}\text{T} \quad (\text{Eq.4.2.7})$$

$$erro_{abs} = 5.91929 \times 10^{-9} - 5.77477 \times 10^{-9} \approx 0.144518 \times 10^{-9}\text{T} \quad (\text{Eq.4.2.8})$$

$$erro_{\%} = \frac{0.144518 \times 10^{-9}}{5.77477 \times 10^{-9}} \times 100 \approx 2.50258\% \quad (\text{Eq.4.2.9})$$

Na figura 4.7 pode ser observado o resultado obtido com a aplicação do algoritmo para correção do erro de aproximação da curva.

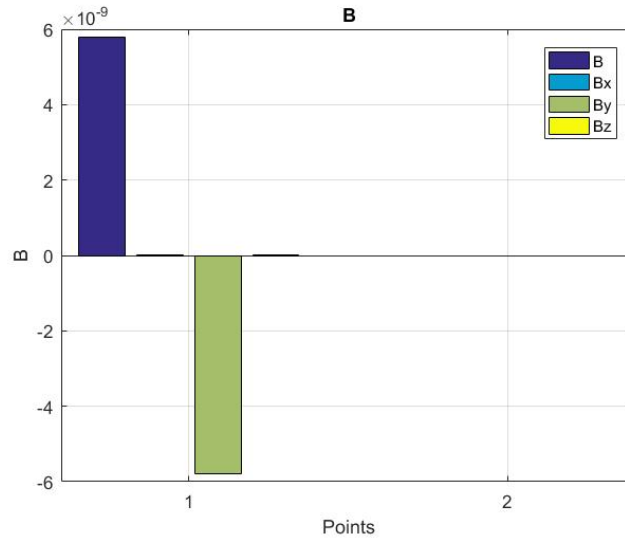


Figura 4.7 - Solução usando algoritmo.

$$B = 5.78295 \times 10^{-9}T \quad (\text{Eq.4.2.10})$$

$$erro_{abs} = 5.78295 \times 10^{-9} - 5.77477 \times 10^{-9} \approx 0.00818 \times 10^{-9}T \quad (\text{Eq.4.2.11})$$

$$erro_{\%} = \frac{0.00818 \times 10^{-9}}{5.77477 \times 10^{-9}} \times 100 \approx 0.14165\% \quad (\text{Eq.4.2.12})$$

Como se pode ver, no caso de o utilizador escolher poucos pontos para a representação da curva, o algoritmo proposto permite uma redução considerável do erro.

Agora a circunferência de raio 0.5m será representada com mil pontos e a outra com dois mil pontos, (figura 4.8).

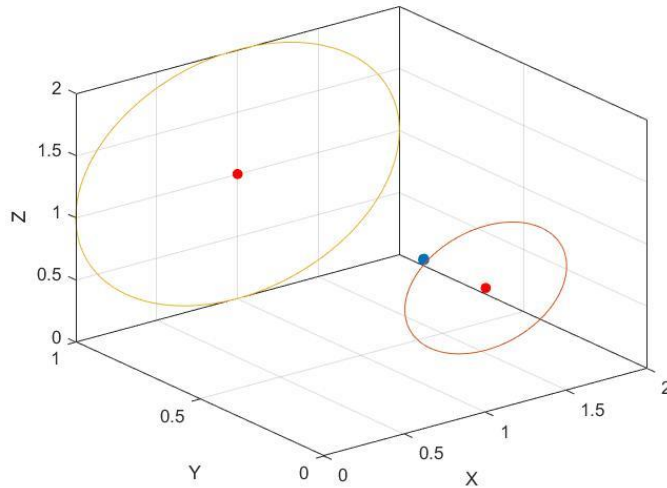


Figura 4.8 - Exemplo teórico representado usando software desenvolvido.

Devido ao elevado número de pontos introduzidos pelo utilizador, ambas circunferências possuem uma curvatura quase perfeita. Vamos comparar a diferença de erro com e sem algoritmo neste caso.

Na figura 4.9 pode ser observado o resultado obtido sem aplicação do algoritmo para correção do erro de aproximação da curva.

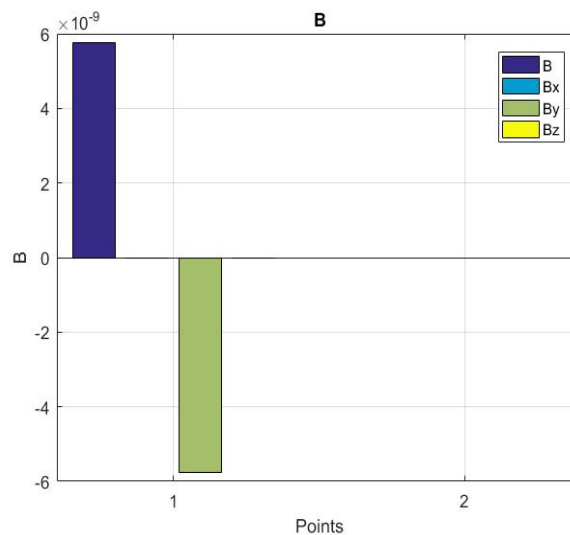


Figura 4.9 - Solução sem algoritmo.

$$B = 5.77478 \times 10^{-9}T \quad (\text{Eq.4.2.13})$$

$$erro_{abs} = 5.77478 \times 10^{-9} - 5.77477 \times 10^{-9} \approx 0.00001 \times 10^{-9}T \quad (\text{Eq.4.2.14})$$

$$erro_{\%} = \frac{0.00001 \times 10^{-9}}{5.77477 \times 10^{-9}} \times 100 \approx 0.0001732\% \quad (\text{Eq.4.2.15})$$

Na figura 4.10 pode ser observado o resultado obtido com a aplicação do algoritmo para correção do erro de aproximação da curva.

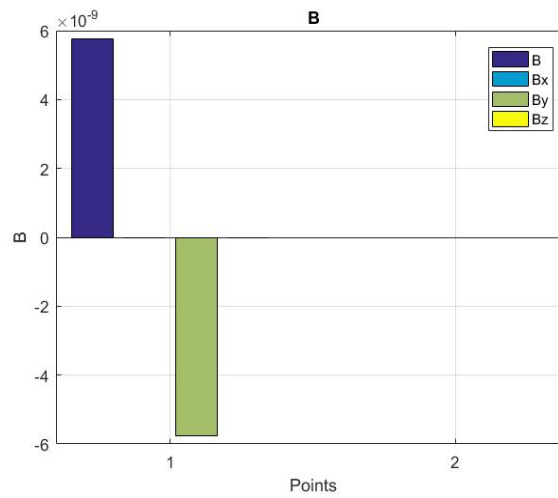


Figura 4.10 - Solução usando algoritmo.

$$B = 5.774773 \times 10^{-9}T \quad (\text{Eq.4.2.16})$$

$$erro_{abs} = 5.774773 \times 10^{-9} - 5.77477 \times 10^{-9} \approx 0.000003 \times 10^{-9}T \quad (\text{Eq.4.2.17})$$

$$erro_{\%} = \frac{0.000003 \times 10^{-9}}{5.77477 \times 10^{-9}} \times 100 \approx 0.0000005\% \quad (\text{Eq.4.2.18})$$

Conclui-se quando se é dado um elevado número de pontos para a representação o erro é quase desprezável, mesmo sem uso do algoritmo.

4.3 Solenoide

O cálculo da densidade do fluxo magnético no centro de um solenoide é dado pela equação:

$$B = \mu_0 \times \frac{N}{L} \times I \quad (\text{Eq.4.3.1})$$

Esta equação 4.3.1 apresentada, representa uma aproximação. Esta aproximação é válida se a distância for superior ao dobro do raio em relação às extremidades do solenoide e o dobro da distância entre espiras em relação à superfície do solenoide [2].

Em seguida apresenta-se um exemplo onde é feita uma variação do tamanho do solenoide ' L ' e do número de espiras ' N ' por forma a demonstrar a validade da equação (Eq.4.3.1).

Considere-se os seguintes parâmetros: um solenoide com um raio de '5cm'; uma corrente de '10A'; uma espira por '1mm' e os tamanhos para o solenoide a considerar devem ser de '1cm', '2cm', '5cm', '10cm', '20cm', '50cm', '1m', '2m', '5m' e '10m'.

Segundo a equação 4.3.1 o resultado devia ser sempre igual independentemente do comprimento do solenoide, devido à densidade de espiras ser constante ($\frac{N}{L}$) mas como já referido anteriormente, esta equação é uma aproximação.

O resultado teórico considerando o tamanho inicial seria de:

$$B = 4 \times \pi \times 10^{-7} \times \frac{10}{0,1} \times 10 = 0,0125664 \text{ T} \quad (\text{Eq.4.3.2})$$

Da figura 4.11 à figura 4.20, podem ser observadas várias representações para os solenoides com vários comprimentos.

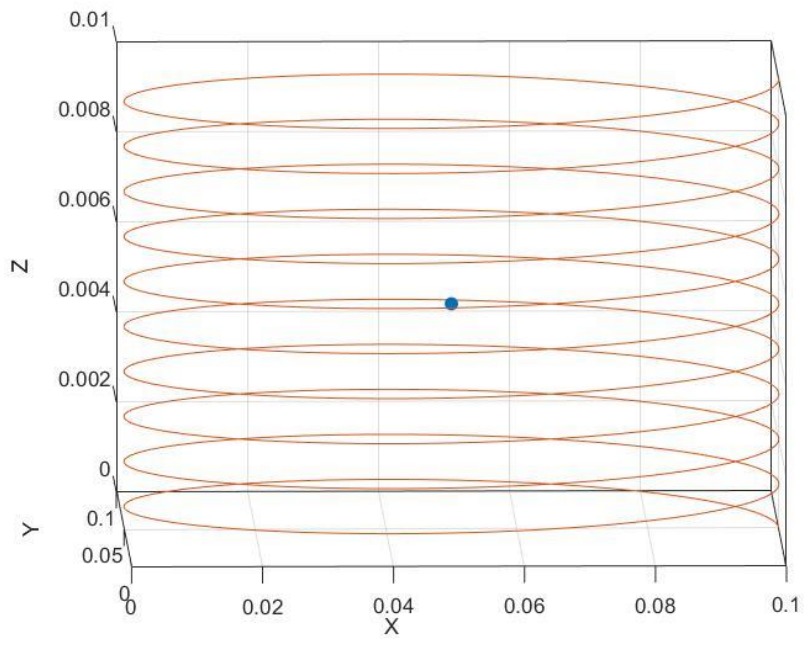


Figura 4.11 - Solenoide (0,01m).

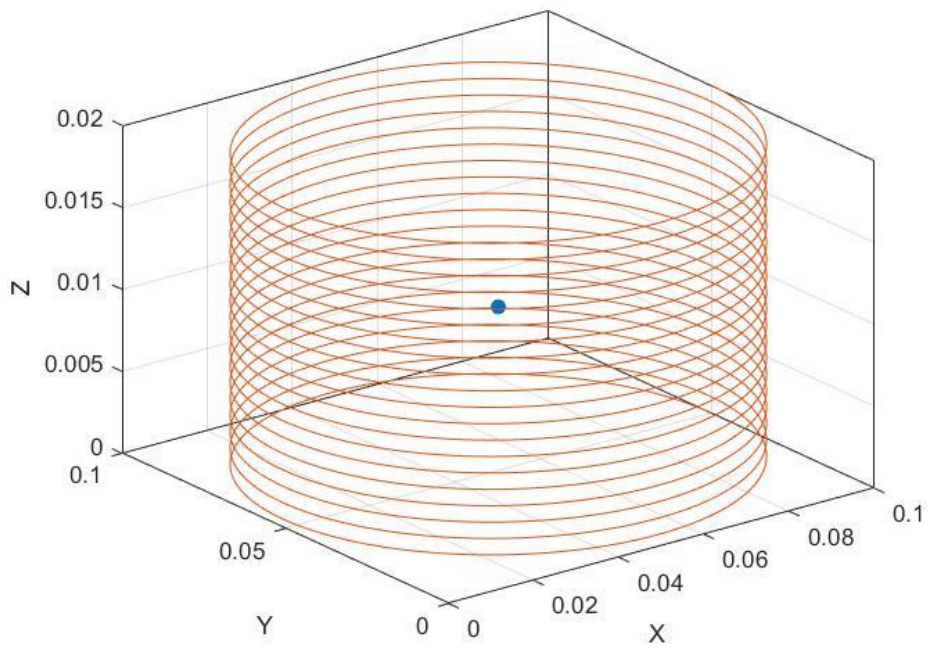


Figura 4.12 - Solenoide (0,02m).

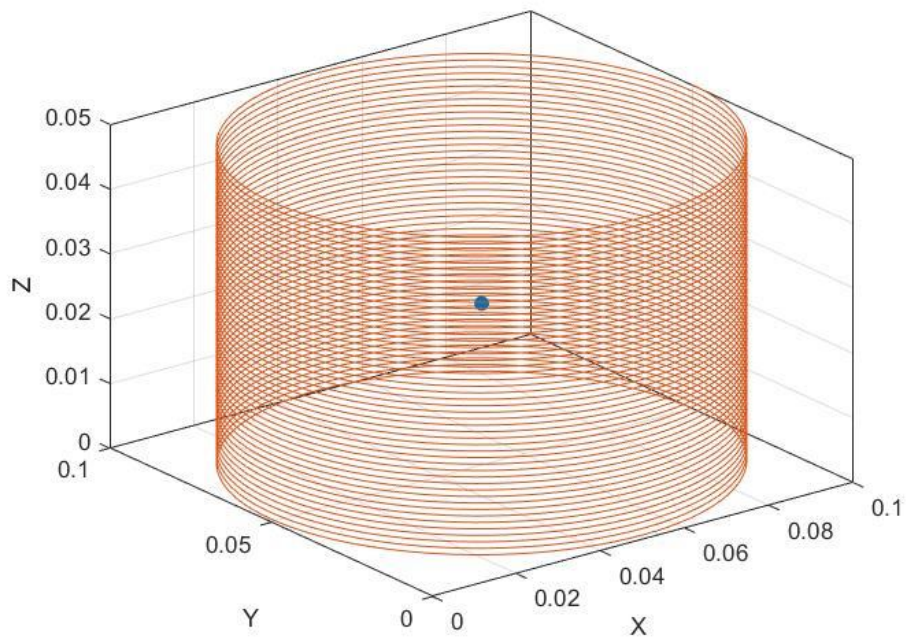


Figura 4.13 - Solenoide (0,05m).

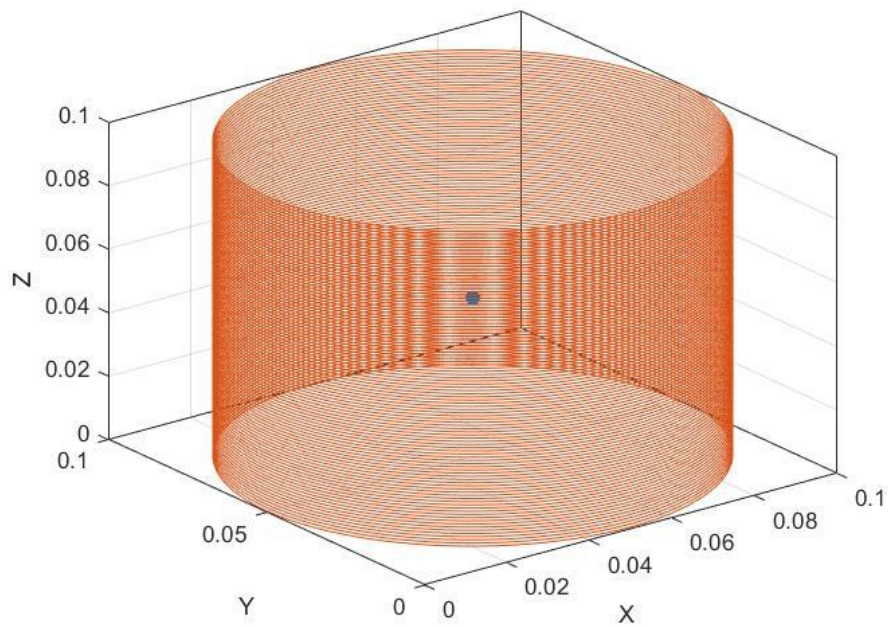


Figura 4.14 - Solenoide (0,1m).

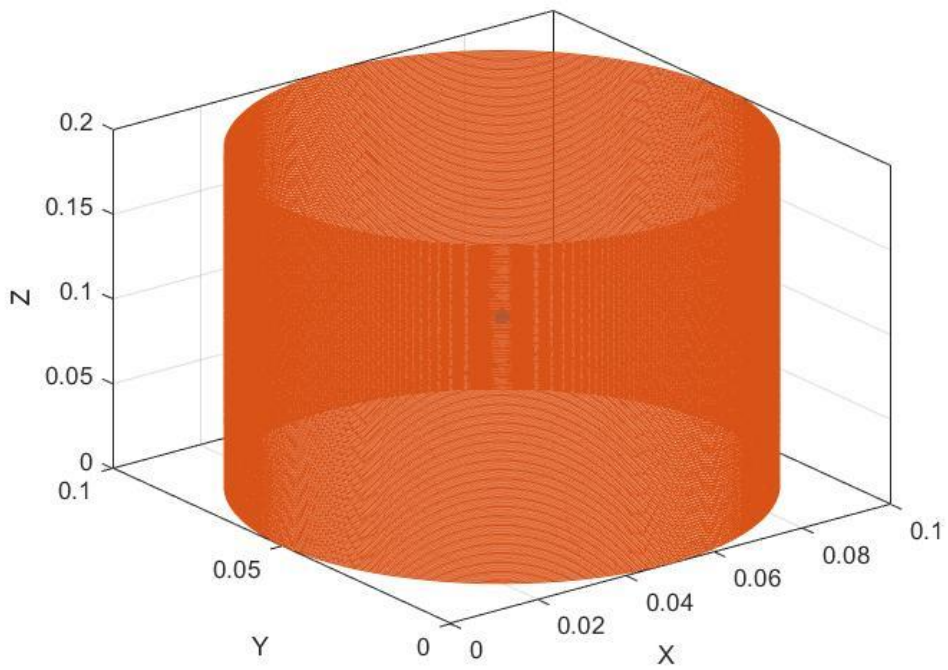


Figura 4.15 - Solenoide (0,2m).

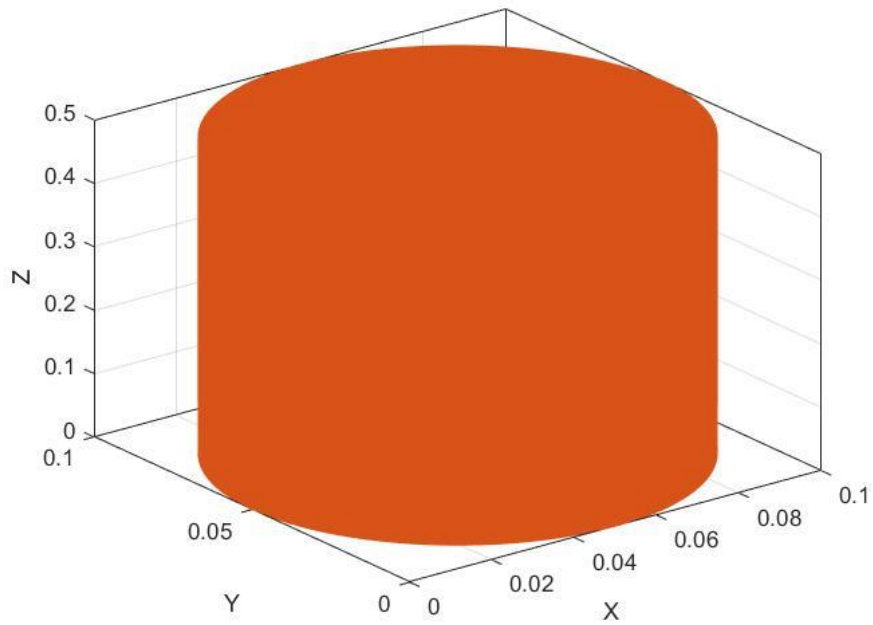


Figura 4.16 - Solenoide (0,5m).

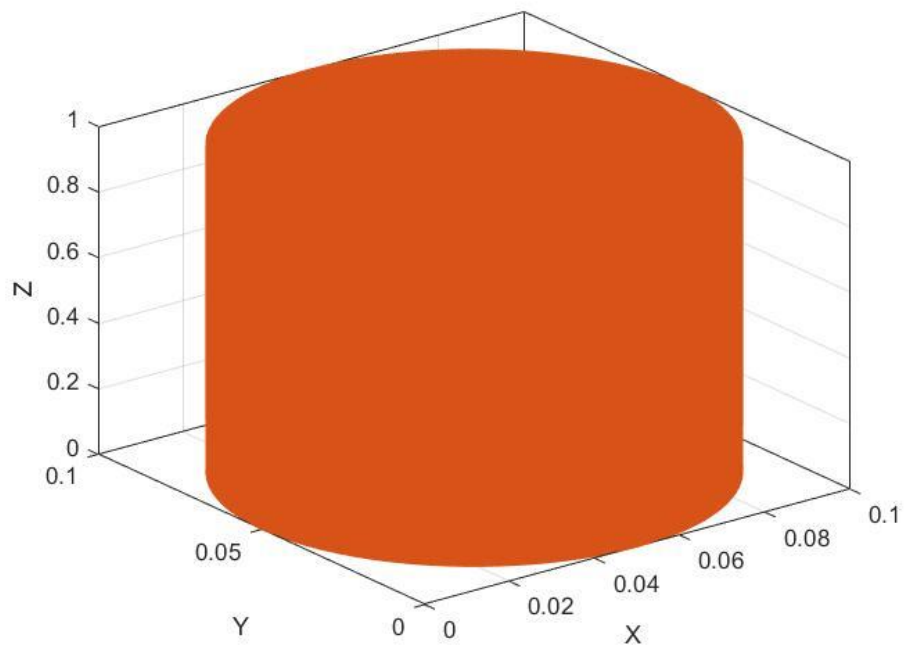


Figura 4.17 - Solenoide (1m).

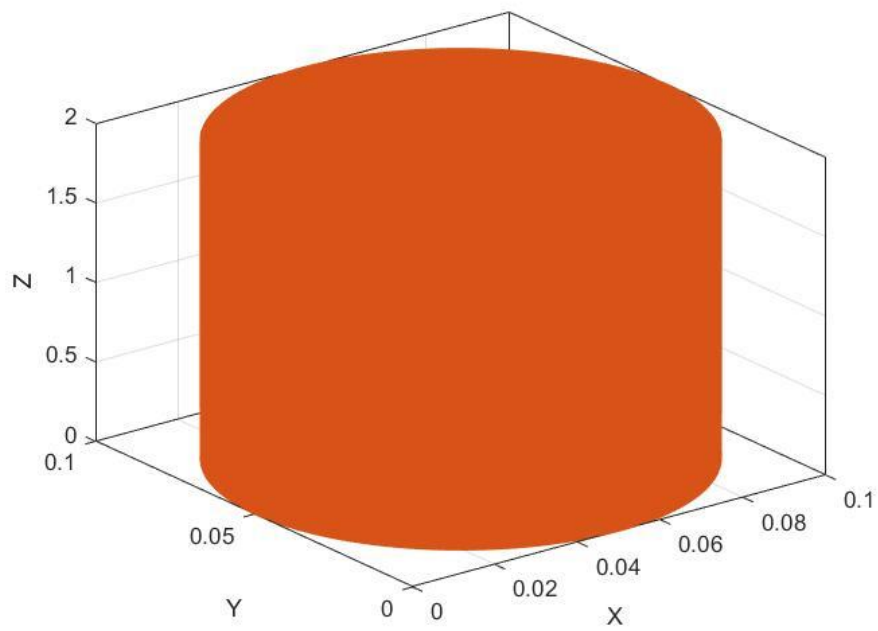


Figura 4.18 - Solenoide (2m).

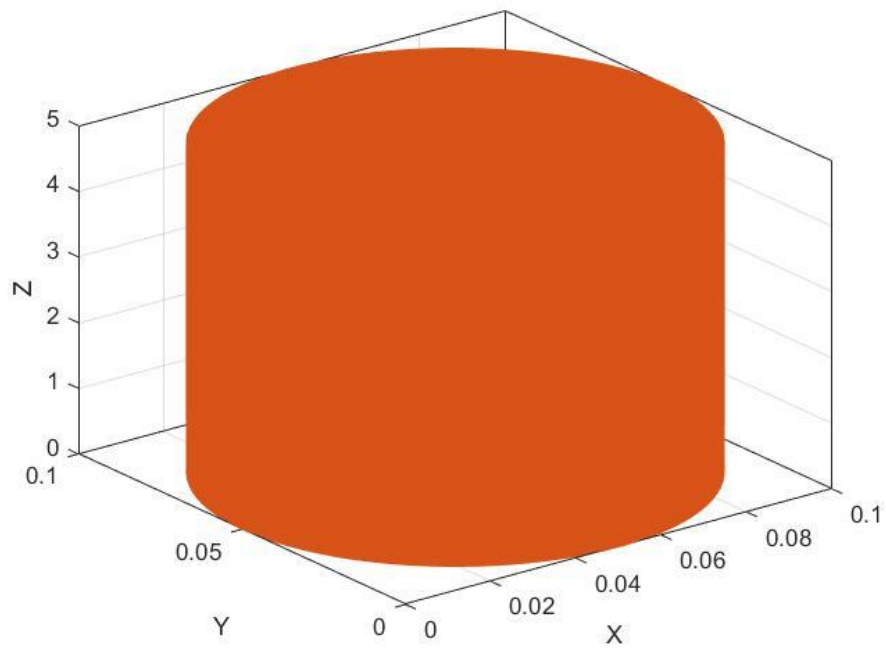


Figura 4.19 - Solenoide (5m).

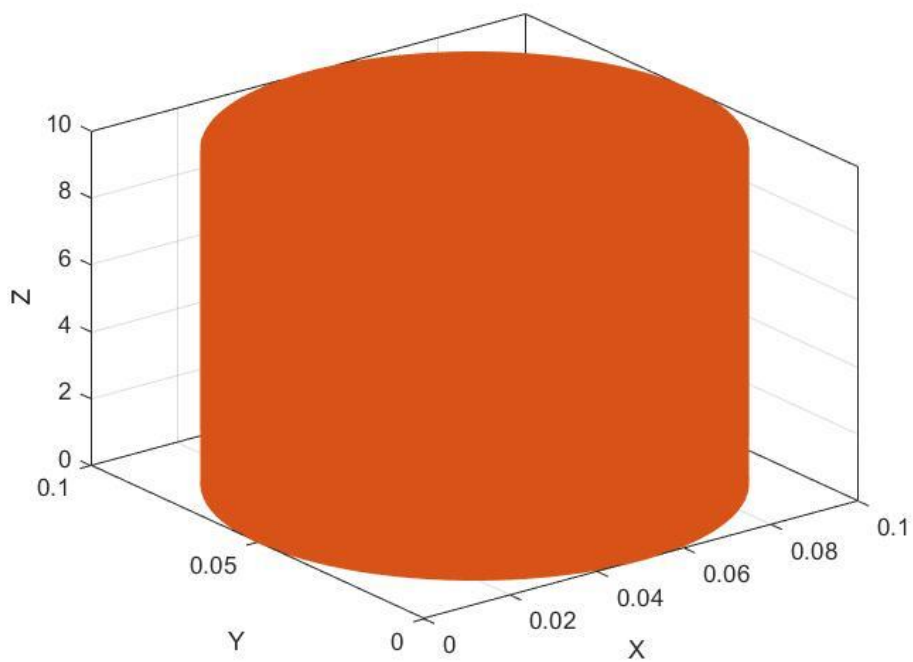


Figura 4.20 - Solenoide (10m).

Nas figuras 4.21 e 4.22 são demonstrados os resultados obtidos usando o software.

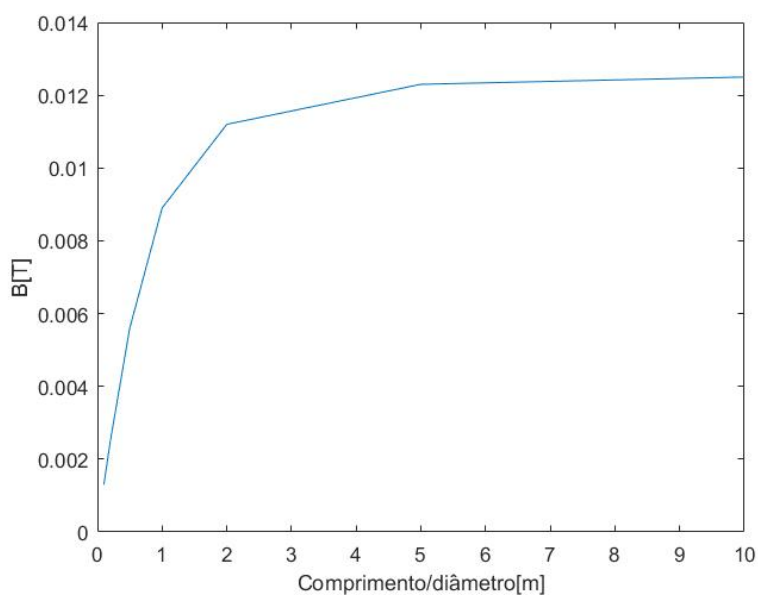


Figura 4.21 - Resultados obtidos usando o software, comprimento (0,01m - 1m).

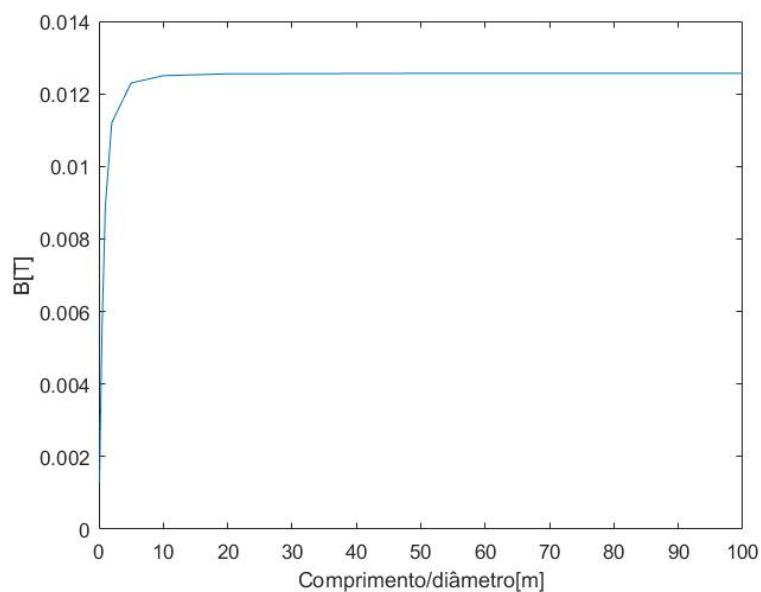


Figura 4.22 - Resultados obtidos usando o software, comprimento (0,01m – 10m).

Conforme se pode concluir a partir das figuras 4.21 e 4.22, para um solenoide com um raio de 5 cm, a aproximação só é verificada a partir de 1 m.

Capítulo 5 - Conclusões e trabalho futuro

5.1 Conclusões

O desenvolvimento deste trabalho tem como objetivo criar um software que possibilite a representação gráfica de três elementos circuitais: linhas (segmento de reta) podendo ser unifilar ou bifilar, curvas (arco de circunferência) e solenoides; verificação da introdução correta dos dados, alertando o utilizador se um circuito constituído por várias linhas estiver aberto; cálculo da densidade do fluxo magnético nos pontos indicados pelo utilizador, mostrando a localização onde esse valor é máximo e mínimo; visualização gráfica da densidade do fluxo magnético nos pontos definidos pelo utilizador; guardar simulação feita pelo utilizador.

A representação disponibilizada pelo software pode ser em duas ou três dimensões, tornando possível desenhar qualquer circuito elétrico desejado pelo utilizador usando os três elementos circuitais de forma facilitada.

Durante o desenvolvimento gráfico dos três elementos circuitais, foi tomada em consideração a limitação do Matlab por não representar uma curva perfeita. Para tal foram adotados novos mecanismos, e desenvolvidos novos algoritmos para compensar o raio, dependendo da diferença de áreas da figura real com a desenhada pelo Matlab. Provou-se assim que se conseguiu diminuir o erro no cálculo da densidade do fluxo magnético.

O software em causa tem também incluída uma função que, em caso de erro de introdução dos dados, avisa quantos circuitos se encontram em circuito aberto, o que possibilita uma validação dos dados introduzidos.

Relativamente à localização para o cálculo da densidade do fluxo magnético, a mesma só pode ser feita no vazio, possibilitando ao utilizador quatro formas de o fazer: por pontos, volume, plano e reta. No final destas simulações será mostrado, recorrendo a gráficos, o circuito elétrico, densidade do fluxo magnético e suas componentes. O tipo de gráficos varia com a forma de localização escolhida, facilitando a visualização e análise ao utilizador.

Com o desenvolvimento do software concluído, procedeu-se à validação do mesmo através de três exemplos teóricos usando os três elementos circuitais. No exemplo das linhas verificou-se que o resultado do software é igual ao exemplo teórico. No segundo exemplo (curvas), devido à limitação já referida do Matlab, o resultado do software envolve um erro que depende do número de pontos dado pelo utilizador, sendo este erro próximo de zero, se o número de pontos para a representação for elevado. Quanto ao terceiro exemplo para comprovar quando a aproximação da equação 4.3.1 se verifica,

concluiu-se que num solenoide com um raio de ‘5cm’ é necessário que o tamanho do solenoide seja de, pelo menos, um metro.

Em suma, após todo o trabalho apresentado, podemos concluir que o software é uma ferramenta fiável para o cálculo de exemplos teóricos e práticos da densidade do fluxo magnético, sendo o erro associado próximo de zero.

5.2 Trabalho futuro

Apesar das funcionalidades já embutidas no software desenvolvido, verifica-se que há ainda margem para implementar diversas melhorias. Podemos referir, entre elas, alguns exemplos:

- Incluir no software o cálculo do campo elétrico para melhor compreensão dos campos criados por circuitos elétricos e sua influência, pois além do campo magnético os circuitos elétricos também criam um campo elétrico.
- Ter em consideração a existência de blindagens metálicas no cálculo da densidade do fluxo magnético.
- Representação gráfica do fluxo magnético usando linhas de campo ou vetores.

Referências

- [1] T. E. o. E. Britannica, “www.britannica.com,” Encyclopædia Britannica, Inc., 2018. [Online]. Available: <https://www.britannica.com/science/magnetic-field>. [Acedido em janeiro 2018].
- [2] J. J. A. B. William H. Hayt, “Engineering Electromagnetics - 8th Edition,” em *Engineering Electromagnetics - 8th Edition*, 1221 Avenue of the Americas, New York, NY 10020, McGraw-Hill, 2012, p. 194.
- [3] R. Feynman, “The Magnetic Field in Various Situations,” [Online]. Available: http://www.feynmanlectures.caltech.edu/II_14.html. [Acedido em janeiro 2018].
- [4] L. Quéval, “mathworks,” [Online]. Available: <https://www.mathworks.com/matlabcentral/fileexchange/50434-biot-savart-magnetic-toolbox?requestedDomain=true>. [Acedido em setembro 2017].
- [5] “UCS (Command),” Autodesk Inc., 2018. [Online]. Available: <https://knowledge.autodesk.com/support/autocad/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/AutoCAD-Core/files/GUID-0BE49DA1-B323-4758-B49B-4C497D194C7A-htm.html>. [Acedido em setembro 2017].
- [6] S. K. Singh, “Magnetic field due to current in straight wire,” [Online]. Available: <http://cnx.org/content/m31103/1.10/>. [Acedido em janeiro 2018].
- [7] R. H. K. D. M. S. Paul Peter Urone, “Magnetic Field of a Current Loop,” Libretexts, [Online]. Available: [https://phys.libretexts.org/TextMaps/General_Physics_TextMaps/Map%3A_University_Physics_\(OpenStax\)/Map%3A_University_Physics_II_\(OpenStax\)/12%3A_A_Sources_of_Magnetic_Fields/12.4%3A_Magnetic_Field_of_a_Current_Loop](https://phys.libretexts.org/TextMaps/General_Physics_TextMaps/Map%3A_University_Physics_(OpenStax)/Map%3A_University_Physics_II_(OpenStax)/12%3A_A_Sources_of_Magnetic_Fields/12.4%3A_Magnetic_Field_of_a_Current_Loop). [Acedido em janeiro 2018].
- [8] T. E. o. E. Britannica, “www.britannica.com,” Encyclopædia Britannica, Inc., 2018. [Online]. Available: <https://www.britannica.com/science/Biot-Savart-law>. [Acedido em janeiro 2018].
- [9] P. G. P. Mendes, “As Linhas de Alta Tensão e a Saúde Pública,” Faculdade de Engenharia da Universidade do Porto, Porto, 2010.
- [10] S. V. e. D. L. Tonci Modric, “3D Computation of the Power Lines Magnetic Field,” *Progress In Electromagnetic Research M*, vol. 41, pp. 1-9, 2015.

