

## CONCEITO E USO DE FERRAMENTAS SIMBÓLICAS E GRÁFICAS VOLTADAS PARA ESTUDO DE ELETROMAGNETISMO E ÁLGEBRA VETORIAL

**Adriano Mendes MAGALHÃES (1); Kléber Zuza NÓBREGA (2)**

(1) Centro Federal de Educação Tecnológica do Maranhão, Av. Getúlio Vargas, N.º 04, Monte Castelo, São Luís –  
MA, 3218-9040, e-mail: [webamm@hotmail.com](mailto:webamm@hotmail.com)

(2) Centro Federal de Educação Tecnológica do Maranhão, e-mail: [bzuza@ieee.org](mailto:bzuza@ieee.org)

### RESUMO

Este projeto de pesquisa, de cunho computacional, estruturado em algoritmos programados em MATLAB, tem como fim o desenvolvimento de uma aplicação orientada às disciplinas de Eletromagnetismo e áreas afins, lecionadas no CEFET-MA, no intuito de proporcionar a resolução de exercícios de cunho vetorial e gráfico, mediante as leis físico-matemáticas requeridas por esta disciplina. Por isso, dividiu-se o trabalho em três etapas: a primeira etapa para a elaboração das funções matemáticas; a segunda etapa, designada para o desenvolvimento do layout gráfico do aplicativo; e a etapa final, para teste das outras etapas, com elaboração de algoritmos de exercícios. Como resultado, obteve-se um aplicativo gráfico que cria, salva e carrega scripts de programas, com as mesmas funções do MATLAB, bem como realiza operações vetoriais e projeta os lugares geométricos das aplicações que contemplam os exercícios a serem resolvidos por um usuário.

**Palavras-chave:** Eletromagnetismo, Álgebra Vetorial, programação, MATLAB.

## 1. INTRODUÇÃO

É um fato que um raciocínio lógico tem uma sequência estruturada de etapas, das quais, ao serem rigorosamente seguidas, chega-se à solução ou conclusão do problema pelo qual tal raciocínio o modela. Em outras palavras, tal encadeamento estruturado idéias, quando definido em regras claras, seguindo uma ordem de desenvolvimento, recebe o nome de algoritmo.

Na disciplina de Eletromagnetismo, a grande maioria dos padrões de exercícios e exemplos segue um modelo típico de algoritmo. Basicamente, este algoritmo é dividido em três blocos de desenvolvimento, devidamente ordenados: a) entradas de dados, que organiza e prepara os dados do problema; b) processamento de dados, que os recebe desta entrada, processando-os em seguida, conforme análise física do problema; c) saída ou retorno de processamento, que capta os valores consequentes deste processamento, adequando-os às condições físicas do problema. Dessa forma é possível obter algoritmos de operações típicas daquela disciplina, como as operações vetoriais de divergente, gradiente e rotacional, bem como as integrais de linha, de superfície e de volume, por exemplo.

Porém, às vezes, para melhor entendimento do estudante, não basta só ter como consequência o resultado final de seus programas. É preciso “ver” algo mais parecido possível com a sua idéia, ainda mais quando se trata de visualizações de lugares geométricos representativos de um dado sistema de coordenadas.

Estas são razões suficientes para ter-se desenvolvido um conjunto de esforços focalizado no cálculo vetorial, tendo como alvo principal as particularidades físicas denotadas pelo Eletromagnetismo, cujo fim é estabelecer uma aplicação estruturada em algoritmos programados em Matlab, objetivando, como propósito fundamental, a ilustração, gráfica e vetorial, de exemplos e exercícios propostos por esta disciplina.

## 2. FUNDAMENTAÇÃO TEÓRICA

Fundamentalmente, as equações básicas relativas ao Cálculo Vetorial voltado para o Eletromagnetismo, como conversões de coordenadas escalares e vetoriais, produto escalar e produto vetorial, componentes escalares e vetoriais, foram baseadas em Hayt Junior (2003), Sadiku (2004) e Winterle (2000).

No entanto merecem destaque as equações dos operadores diferenciais e integrais, conforme também Hayt Junior (2003), Sadiku (2004) e Gonçalves (1999), a serem mostradas na Tabela 1.

**Tabela 1 – Formulário dos operadores diferenciais e operadores integrais nos seus respectivos sistemas de coordenadas**

Coordenadas	Divergente	
Cartesianas	$\nabla \cdot \vec{D} = \frac{\partial D_x}{\partial x} + \frac{\partial D_y}{\partial y} + \frac{\partial D_z}{\partial z}$	[Eq. 1]
Cilíndricas	$\nabla \cdot \vec{D} = \frac{1}{\rho} \frac{\partial}{\partial \rho} (\rho D_\rho) + \frac{1}{\rho} \frac{\partial D_\phi}{\partial \phi} + \frac{\partial D_z}{\partial z}$	[Eq. 2]
Esféricas	$\nabla \cdot \vec{D} = \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 D_r) + \frac{1}{r \sin(\theta)} \frac{\partial}{\partial \theta} (\sin(\theta) D_\theta) + \frac{1}{r \sin(\theta)} \frac{\partial D_\phi}{\partial \phi}$	[Eq. 3]
Coordenadas	Gradiente	
Cartesianas	$\nabla f = \frac{\partial f}{\partial x} \vec{a}_x + \frac{\partial f}{\partial y} \vec{a}_y + \frac{\partial f}{\partial z} \vec{a}_z$	[Eq. 4]
Cilíndricas	$\nabla f = \frac{\partial f}{\partial \rho} \vec{a}_\rho + \frac{1}{\rho} \frac{\partial f}{\partial \phi} \vec{a}_\phi + \frac{\partial f}{\partial z} \vec{a}_z$	[Eq. 5]
Esféricas	$\nabla f = \frac{\partial f}{\partial r} \vec{a}_r + \frac{1}{r} \frac{\partial f}{\partial \theta} \vec{a}_\theta + \frac{1}{r \sin(\theta)} \frac{\partial f}{\partial \phi} \vec{a}_\phi$	[Eq. 6]
Coordenadas	Rotacional	
Cartesianas	$\nabla \times \vec{H} = \left( \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \right) \vec{a}_x + \left( \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \right) \vec{a}_y + \left( \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right) \vec{a}_z$	[Eq. 7]

Cilíndricas	$\nabla \times \vec{H} = \left( \frac{1}{\rho} \frac{\partial H_z}{\partial \varphi} - \frac{\partial H_\varphi}{\partial z} \right) \vec{a}_\rho + \left( \frac{\partial H_\rho}{\partial z} - \frac{\partial H_z}{\partial \rho} \right) \vec{a}_\varphi + \left( \frac{1}{\rho} \frac{\partial(\rho H_\varphi)}{\partial \rho} - \frac{1}{\rho} \frac{\partial H_\rho}{\partial \varphi} \right) \vec{a}_z$		[Eq. 8]
Esféricas	$\nabla \times \vec{H} = \frac{1}{r \sin(\theta)} \left( \frac{\partial(\sin(\theta) H_\varphi)}{\partial \theta} - \frac{\partial H_\theta}{\partial \varphi} \right) \vec{a}_r + \frac{1}{r} \left( \frac{1}{\sin(\theta)} \frac{\partial H_r}{\partial \varphi} - \frac{\partial(r H_\varphi)}{\partial r} \right) \vec{a}_\theta + \frac{1}{r} \left( \frac{\partial(r H_\theta)}{\partial r} - \frac{\partial H_r}{\partial \theta} \right) \vec{a}_\varphi$		[Eq. 9]
<b>Integral de Volume:</b>		<b>Coordenadas</b>	<b>Elemento diferencial escalar <math>dv</math></b>
$\iiint_V f \cdot dv \quad [\text{Eq. 10}]$  Onde $f$ é um campo escalar qualquer e $dv$ é um elemento diferencial escalar de volume.		Cartesianas	$dv = dx dy dz$
		Cilíndricas	$dv = \rho d\rho d\varphi dz$
		Esféricas	$dv = r^2 \sin(\theta) dr d\theta d\varphi$
<b>Integral de Superfície Aberta e Fechada, respectivamente:</b>		<b>Coordenadas</b>	<b>Elemento diferencial vetorial <math>d\vec{S}</math>.</b>
$\iint_S \vec{D} \cdot d\vec{S} \wedge \oint_S \vec{D} \cdot d\vec{S} = \sum_{i=1}^n \iint_{S_i} \vec{D} \cdot d\vec{S} \quad [\text{Eq. 11}]$  Onde $\vec{D}$ é um campo vetorial qualquer e $d\vec{S}$ é o elemento diferencial vetorial de superfície, conforme os sistemas de coordenadas.		Cartesianas	$d\vec{S} = dy dz \vec{a}_x + dx dz \vec{a}_y + dx dy \vec{a}_z$
		Cilíndricas	$d\vec{S} = \rho d\varphi dz \vec{a}_\rho + \rho dz d\varphi \vec{a}_\varphi + \rho d\rho d\varphi \vec{a}_z$
		Esféricas	$d\vec{S} = r^2 \sin(\theta) d\theta d\varphi \vec{a}_r + r \sin(\theta) dr d\varphi \vec{a}_\theta + r dr d\theta \vec{a}_\varphi$
<b>Integral de Linha Aberta e Fechada, respectivamente:</b>		<b>Coordenadas</b>	<b>Elemento diferencial vetorial <math>d\vec{L}</math>.</b>
$\int_L \vec{H} \cdot d\vec{L} \wedge \oint_L \vec{H} \cdot d\vec{L} = \sum_{i=1}^n \int_{L_i} \vec{H} \cdot d\vec{L} \quad [\text{Eq. 12}]$  Onde $\vec{H}$ é um campo vetorial qualquer e $d\vec{L}$ é o elemento diferencial vetorial de comprimento, conforme os sistemas de coordenadas.		Cartesianas	$d\vec{L} = dx \vec{a}_x + dy \vec{a}_y + dz \vec{a}_z$
		Cilíndricas	$d\vec{L} = d\rho \vec{a}_\rho + \rho d\varphi \vec{a}_\varphi + dz \vec{a}_z$
		Esféricas	$d\vec{L} = dr \vec{a}_r + r d\theta \vec{a}_\theta + r \sin(\theta) d\varphi \vec{a}_\varphi$

Ressaltam-se algumas observações a respeito destes operadores:

- As variáveis independentes do sistema cartesiano, cilíndrico e esférico são, respectivamente,  $[x, y, z]$ ,  $[\rho, \varphi, z]$  e  $[r, \theta, \varphi]$ ;
- As variáveis  $\theta$  e  $\varphi$  são angulares e tem unidade em radianos.

### 3. METODOLOGIA

Para desenvolver o trabalho de maneira rápida e eficaz, este projeto foi dividido em três etapas, onde cada etapa prioriza um conjunto de procedimentos que orientam o processo do projeto. A primeira etapa é denominada conjunto vetorial e trabalhou especialmente com o cálculo vetorial. Nesta etapa, as metodologias utilizadas foram os estudos de Cálculo Vetorial, sendo estes relacionados com o Cálculo I, com o Cálculo II e com o Cálculo III, conforme Hayt Junior (2003), Sadiku (2004), Winterle (2000), Leithold (1986), Munem (1982) e Gonçalves (1999), bem como a aplicação de técnicas de programação, além da execução de testes no que tange o desenvolvimento das funções necessárias para esta etapa. A segunda etapa é denominada conjunto gráfico e trabalhou com a construção de gráficos 2D e 3D, a fim de mostrar as formas gráficas geradas por alguns cálculos obtidos na primeira etapa: para tal usaram-se as metodologias de estudo e pesquisa das funções de plotagem 2D e 3D do Matlab, conforme Chapman (2003), bem como se usou também as metodologias de técnicas de programação e de execução de testes necessários para a produção das funções pertinentes a esta etapa. A terceira etapa, ou etapa final, trabalhou a finalização do projeto, desenvolvendo um tipo de interface gráfica para o usuário, tal que ele possa ter acesso aos gráficos

sem necessariamente ter que digitar um programa com as funções desenvolvidas. As metodologias adotadas nesta etapa não foram somente o estudo e a pesquisa das funções do Matlab, correlacionadas com desenvolvimento deste tipo de interface, conforme Chapman (2003), mas também foram aquelas que contemplam o planejamento do design gráfico, a implementação deste com as técnicas de programação necessárias para tal, e o conjunto de testes finais essenciais para a conclusão do projeto.

Quanto às técnicas de programação acima citadas, foram estabelecidas normas de programação que visaram, para certas funções desenvolvidas, reger a padronização dos programas. Estas normas foram utilizadas, em larga escala, principalmente no conjunto vetorial, já que nesta etapa a programação é mais estruturada em cálculos matemáticos.

No decorrer da finalização de cada etapa, criou-se um conjunto de testes dos quais os algoritmos foram implementados, em Matlab, sobre um exemplo ou exercício, ou ainda um conjunto destes, que dependo do momento, eram hipotético(s) ou extraído(s) de uma das referências do tema “Eletromagnetismo”, conforme Hayt (2003) e Sadiku (2004).

## 4. RESULTADOS OBTIDOS

Em seguida, estão comentados algumas das funções desenvolvidas, com suas respectivas sintaxes de uso, relativas às etapas que foram as bases deste trabalho.

### 4.1. Primeira etapa

Para esta etapa, dentre os resultados obtidos, merecem destaque os operadores diferenciais, como divergente, gradiente e rotacional, e os operadores integrais, como as integrais de volume, superfície e linha.

#### 4.1.1. Divergente e Rotacional de um campo vetorial

A função divergente extrai o divergente de um campo vetorial, enquanto que a função rotacional extrai o rotacional deste campo, conforme as Equações 1, 2, 3 e Equações 7, 8, 9 respectivamente.

Sintaxe:

```
saida=divergente(ventrada)
saida=rotacional(ventrada)
```

onde:

ventrada: é um vetor simbólico ou célula do tipo 3x1.

Exemplo 1: Encontrar o divergente e o rotacional dos seguintes campos vetoriais:  $\vec{D} = \rho^2 \cos^2 \phi \vec{a}_\rho + z \sin \phi \vec{a}_\phi$  e  $\vec{H} = 6xy \vec{a}_x - 3y^2 \vec{a}_y$ , conforme Sadiku (2004) e Hayt Junior (2003).

```
>> syms rho phi z real; vG=[rho^2*cos(phi)^2;z*sin(phi);0];
div_vG=divergente(vG)
```

```
div_vG =
cos(phi)*(3*rho^2*cos(phi)+z)/rho
```

```
>> vH=[6*x*y;-3*y^2;0]; rot_vH=rotacional(vH)
```

```
rot_vH =
0
0
-6*x
```

#### 4.1.2. Gradiente de uma função

Extrai o gradiente de uma função, conforme as Equações 4,5 e 6.

Sintaxe:

```
saida=gradiente(funcao)
```

onde:

funcao: é uma expressão simbólica ou célula que contem a função da qual se deseja encontrar o gradiente.

Exemplo 2: Calcular o gradiente do campo escalar  $V = e^{-z} \sin(2x) \cosh(y)$ , segundo Sadiku (2004).

```
>> syms x y z real; V=exp(-z)*sin(2*x)*cosh(y); gradV=gradiente(V)
gradV =
      2*exp(-z)*cos(2*x)*cosh(y)
      exp(-z)*sin(2*x)*sinh(y)
      -exp(-z)*sin(2*x)*cosh(y)
```

#### 4.1.3. Integral de volume de uma função

Calcula a integral de volume de uma função, conforme a Equação 10.

Sintaxe:

```
saida=integral_volume(funcao,p_min,p_max)
```

onde:

- funcao: é a função (simbólica ou célula) que contém uma função da qual se deseja encontrar a integral de volume.
- p\_min: um vetor numérico 3x1 contendo os pontos mínimos das variações das variáveis que determinam o volume analisado, nas respectivas ordens do sistema de coordenadas. Exemplo: Se  $x_o \leq x \leq x_f$ ,  $y_o \leq y \leq y_f$  e então  $p\_min=[x\_o;y\_o;z\_o]$  para o sistema de coordenadas cartesiano.
- p\_max: um vetor numérico 3x1 contendo os pontos máximos das variações das variáveis que determinam o volume analisado, nas respectivas ordens do sistema de coordenadas. Exemplo: Se  $x_o \leq x \leq x_f$ ,  $y_o \leq y \leq y_f$  e  $z_o \leq z \leq z_f$ , então  $p\_max=[x\_f;y\_f;z\_f]$  para o sistema de coordenadas cartesiano.

Exemplo 3: Calcular o segundo membro do teorema de Gauss com por meio da integral de volume do divergente do campo vetorial  $\vec{D} = \rho^2 \cos^2 \varphi \vec{a}_\rho + z \sin \varphi \vec{a}_\varphi$ , numa superfície fechada de um cilindro  $0 \leq z \leq 1, \rho = 4$ , conforme Sadiku (2004).

```
>> syms rho phi z real; vG=[rho^2*cos(phi)^2;z*sin(phi);0];
div_vG=divergente(vG); p_min=[0;0;0]; p_max=[4;2*pi;1];
psiVolume=double(integral_volume(div_vG,p_min,p_max))
psiVolume =
      201.0619
```

#### 4.1.4. Integral de superfície, aberta ou fechada, de um campo vetorial

Calcula a integral de superfície, tanto fechada quanto aberta, de um campo vetorial, conforme a Equação 11. A determinação de que a integral a ser calculada é de superfície aberta ou superfície fechada é feita automaticamente pelo programa, conforme explicação a seguir:

- Integral de Superfície Fechada: sempre uma superfície fechada é o delimitador de um volume. Sendo assim, existe um intervalo de 6 pontos distintos que irão compor as variáveis p\_min e p\_max.
- Integral de Superfície Aberta: para este tipo de integral, o programa efetua apenas cálculo sobre superfícies em que uma das coordenadas sistema permanece constante, em outras palavras, as superfícies serão sempre as formas básicas dos três sistemas de coordenadas utilizados. Desta forma, a entrada de dados nas variáveis p\_min e p\_max será repetida em 1 das 3 variações.

Sintaxe:

```
saida=integral_superficie(campo_vetor,p_min,p_max)
```

onde:

- `campo_vetor`: é um vetor simbólico ou célula do tipo 3x1(padão) que contém um campo vetorial no qual se deseja encontrar a integral de superfície.
- `p_min`: é um vetor numérico 3x1 que, assim como na integral de volume, contém os pontos mínimos das variações variáveis que determinam o volume analisado, nas respectivas ordens do sistema de coordenadas.
- `p_max`: é um vetor numérico 3x1 que, assim como na integral de volume, contém os pontos máximos das variações das variáveis que determinam o volume analisado, nas respectivas ordens do sistema de coordenadas.

Exemplo 4: Calcular o primeiro membro do teorema de Gauss do exemplo anterior.

```
>> syms rho phi z real; vG=[rho^2*cos(phi)^2;z*sin(phi);0]; p_min=[0;0;0];  
p_max=[4;2*pi;1]; psiSuperficieFechada=integral_superficie(vG,p_min,p_max)  
  
psiSuperficieFechada =  
  
201.0619
```

Exemplo 5: Calcular o segundo membro do teorema de Stokes do campo vetorial  $\vec{H} = 6xy\vec{a}_x - 3y^2\vec{a}_y$  para um caminho retangular ao redor da região  $2 \leq x \leq 5$ ,  $-1 \leq y \leq 1$ ,  $z = 0$ , conforme Hayt Junior (2003).

```
>> syms rho the phi x y z r real; vH=[6*x*y;-3*y^2;0]; rot_vH=rotacional(vH);  
intSuperficieAberta=integral_superficie(rot_vH,[2;-1;0],[5;1;0])  
  
intSuperficieAberta =  
  
-126
```

#### 4.1.5. Integral de linha, aberta ou fechada, de um campo vetorial

Calcula a integral de linha aberta ou fechada de um campo vetorial, mediante o fornecimento dos pontos inicial e final, da matriz de pontos intermediários e de um caminho, ao passo que a função o exige, de acordo com a Equação 12.

Sintaxe:

```
saida=integral_linha(campo_vetor,p_ini,p_fim,ptos_intermed)
```

onde:

- `saida=integral_linha(campo_vetor,p_ini,p_fim,ptos_intermed)` calcula a integral de linha aberta ou fechada, com ou sem pontos intermediários, mediante a seguinte manipulação: se `p_ini` é igual a `p_fim` e `ptos_intermed` não foi definida então é uma integral de linha fechada sem pontos intermediários, senão se `p_ini` é igual a `p_fim` e `ptos_intermed` foi definida então é uma integral de linha fechada com pontos intermediários, ou então se `p_ini` é diferente de `p_fim` e `ptos_intermed` não é definida então a integral de linha tem o mesmo efeito da sintaxe anterior, caso contrário se `p_ini` é diferente de `p_fim` e `ptos_intermed` é definida então a integral de linha é aberta com pontos intermediários. Nota-se que ao se ter pontos intermediários, têm-se também caminhos intermediários que constitui um único caminho. No fim, esta sintaxe retorna o valor da integral em `saida`.

Observações:

Ao rodar a função, será solicitado ao usuário a entrada de um caminho definido por uma ou duas funções, que devem ser colocadas separadas por ponto-e-vírgula;

- `campo_vetor`: é um vetor simbólico ou célula do tipo 3x1(padão) que contém um campo vetorial da qual se deseja encontrar a integral de linha. Nota-se que para teste, admite-se um vetor célula 3x1.
- `p_ini`: ponto inicial onde começa o percurso. É um vetor numérico 3x1 com as componentes escalares nas respectivas ordens de um sistema de coordenadas, assim como na integral de volume.
- `p_fim`: ponto final onde termina o percurso. É um vetor numérico 3x1 com as componentes escalares nas respectivas ordens de um sistema de coordenadas, assim como na integral de volume.

- `ptos_intermed`: matriz de pontos intermediários  $3 \times i$ , onde  $i$  é o número de pontos intermediários. Nesta matriz, cada linha representa uma componente em uma direção e cada coluna representa um ponto intermediário.

Exemplo 6: Calcular o primeiro membro do teorema de Stokes do campo vetorial  $\vec{H} = 6xy\vec{a}_x - 3y^2\vec{a}_y$  para um caminho retangular ao redor da região  $2 \leq x \leq 5$ ,  $-1 \leq y \leq 1$ ,  $z = 0$ , conforme Hayt Junior (2003).

```
>> syms rho the phi x y z r real; vH=[6*x*y;-3*y^2;0]; p_ini=[2;1;0];  
p_fim=p_ini; ptosInterm=[2 5 5;-1 -1 1;0 0 0];  
intLinhaFechada=integral_linha(vH,p_ini,p_fim,ptosInterm)  
  
intLinhaFechada =  
  
-126
```

## 4.2. Segunda etapa

Basicamente, esta etapa contempla a criação e teste do conjunto de funções que geram graficamente os lugares geométricos pertinentes aos sistemas de coordenadas, anteriormente apresentados.

Por isso, esse conjunto, de fato, é representado por uma função em particular, pela qual é responsável por carregar e organizar os dados, a fim de passá-los para as outras funções que os processam. Essa função é dita visualizador.

### 4.2.1. Visualizador do Lugar Geométrico (LG)

Capta os parâmetros necessários e lança-os, formatados, na função `visual_base`, que responsável por gerar o gráfico de um vetor ou campo vetorial dado em sistema cartesiano, cilíndrico ou esférico.

Sintaxe:

```
visualizador(ventrada)  
visualizador(ventrada,infinito,n)
```

onde:

`visualizador(ventrada)` lança o vetor ou campo vetorial `ventrada`, célula  $3 \times 1$  ou simbólico/numérico  $3 \times 1$ , na função `visual_base` para construir o gráfico do LG do vetor.

`visualizador(ventrada,infinito,n)` lança o vetor ou campo vetorial `ventrada`, célula  $3 \times 1$  ou simbólico/numérico  $3 \times 1$ , na função `visual_base`, além de determinar o maior número, que assume um aparente "infinito" para referência de cálculo da máquina, e um número  $n$  de elementos partições infinitesimais de mesmo tamanho que compõem o contorno da superfície gerada pela esta função.

Observações:

Quanto maior for  $n$ , menores são os elementos infinitesimais e mais bem definido fica o gráfico do lugar geométrico, levando mais tempo para este gráfico ser plotado.

Exemplo 7:

```
>> syms x y z rho phi r the real; vP1=[1;2;3]; vA=[x;1;2]; vB=[5;phi;z];  
vC=[3;the;phi]; visualizador(vP1,10,15), pause % Figura 1  
>> cla; visualizador(vA,15,30), pause % Figura 2  
>> cla; visualizador(vB,10,35), pause % Figura 3  
>> cla; visualizador(vC,10,35), pause % Figura 4
```

## 4.3. Etapa final

Nesta etapa, foi desenvolvida uma função que gera uma interface gráfica para usuários, reunindo assim os elementos da primeira e da segunda etapa. Tal interface apresenta o layout, conforme a Figura 5.

- Menus:

- **Arquivo:** apresenta mais três submenus que são responsáveis por carregar algoritmos \*.m previamente desenvolvidos (Abrir...), gerar algoritmos \*.m, criados na interface gráfica (Salvar como...) e sair da aplicação (Sair).
- **Vetores:** apresenta os submenus que realizam as conversões de sistemas de coordenadas, extraem as componentes de um vetor e realizam os produtos vetorial e interno, conforme detalhados na primeira etapa.
- **Operadores Diferenciais:** apresentam os submenus das operações diferenciais vigentes sobre um vetor (divergente, gradiente e rotacional), descritas também na primeira etapa.
- **Integrais:** apresentam os submenus das integrais de linha, superfície e volume.
- **Tipos de Exibição:**

A primeira das telas da aplicação, segundo a Figura 6, é um tipo de exibição denominada Desktop: esta exibição mostra dois campos de texto editáveis e dois botões: o primeiro campo editável cria ou edita os algoritmos, e o segundo campo editável checam-se os resultados processados ou avisos de erros de processamento. O primeiro botão (Processar) realiza, ao ser clicado, o processamento das linhas de comando do primeiro quadro e retorna os resultados no segundo quadro. O segundo botão (Limpar) apaga os resultados do segundo quadro.

A segunda tela, segundo a Figura 7, é denominada Gráfico, é responsável por apresentar os gráficos mostrados nos exemplos da segunda etapa. Ela apresenta um gráfico com os eixos cartesianos x, y e z, e mais três botões: Editor..., Rotação e Limpar. O botão Editor... abre uma caixa de diálogos com os campos necessários para carregar os dados para que a função visualizador possa plotar os gráficos. O botão Rotação permite girar o gráfico, possibilitando assim visualizações “em ângulos” dos mesmos. O botão Limpar apaga um ou mais gráficos plotados sobre os eixos cartesianos.

A terceira tela, conforme a Figura 8, é denominada Desktop e Gráfico, é um arranjo das duas telas anteriores e proporciona a programação de algoritmos e o uso explícito da função visualizador nas linhas de programação, ou seja, se o operador desejar resolver um problema de eletromagnetismo e também visualizar um dado ponto, uma linha ou uma superfície, ele pode associar o uso daquela função no seu algoritmo e visualizar ao mesmo tempo aquele lugar geométrico no espaço.

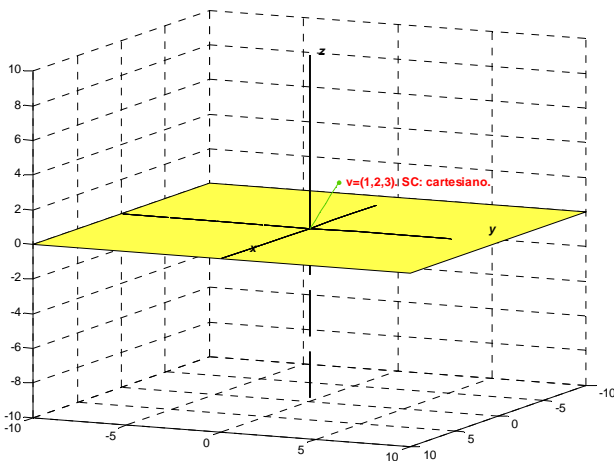


Figura 1 – Exemplo do vetor-posição  $\vec{P}_1$  plotado no espaço

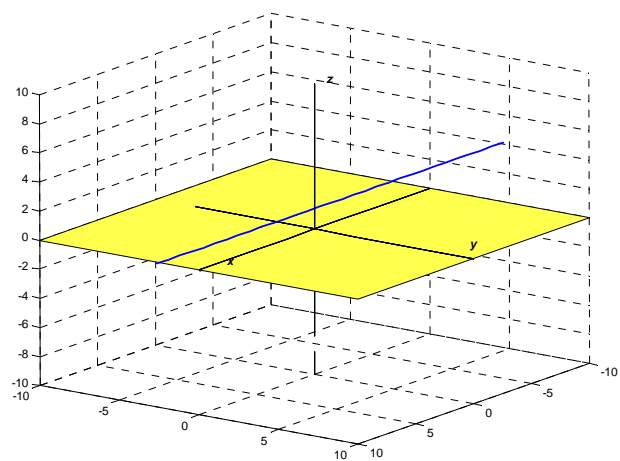
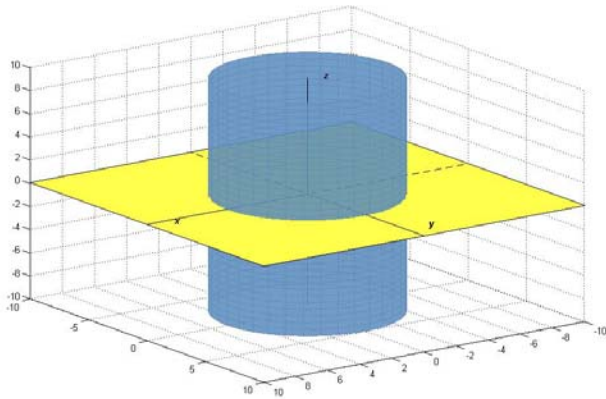


Figura 2 – Exemplo do campo vetorial  $\vec{A}$  plotado no espaço

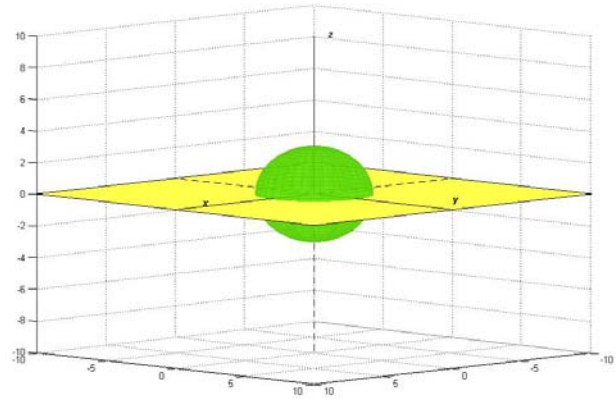
## 5. CONCLUSÕES

Diante do que foi exposto, observa-se que os trabalhos, no decorrer do projeto, estavam acontecendo conforme previsto, até a primeira etapa: nas outras duas etapas houve modificações com relação ao tempo estimado para o decorrer das atividades. Porém, tais modificações não interferiram na fluência e conclusão do projeto. Com isso, vale ressaltar a análise de alguns pontos positivos e negativos.



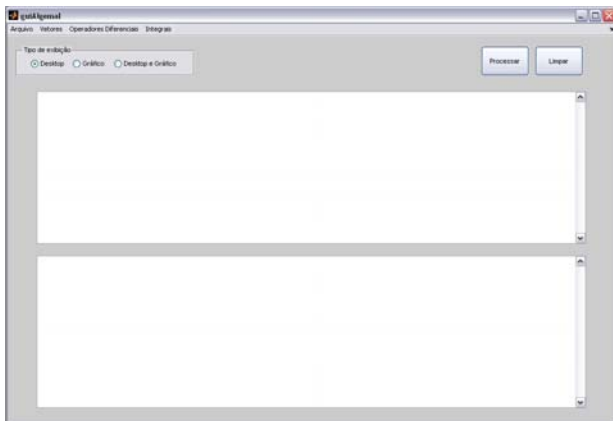


**Figura 3 – Exemplo do campo vetorial  $\vec{B}$  plotado no espaço**

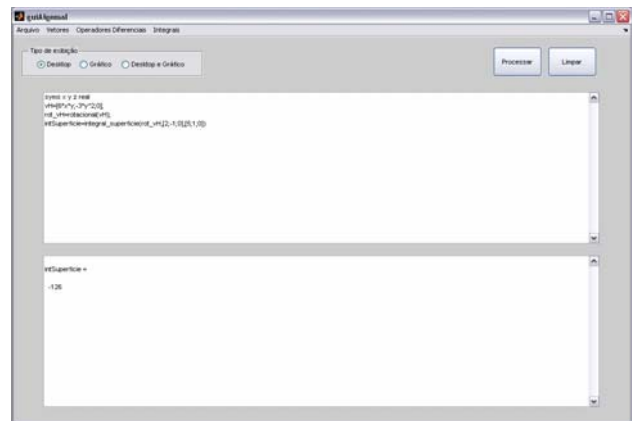


**Figura 4 – Exemplo do campo vetorial  $\vec{F}$  plotado no espaço**

Em relação aos pontos positivos, evidenciam-se o aprendizado de uma nova linguagem, bem como a prática de aplicações físicas diante os assuntos estudados em Física, relacionada com a ampliação de novos horizontes em relação ao Matlab, tendo em vista a flexibilidade do aplicativo quanto aos cálculos utilizados na engenharia. Destacam-se ainda a funcionalidade do conjunto vetorial e a possibilidade da estruturação de alguns exemplos para explicação de assuntos específicos do Cálculo, baseados no conjunto gráfico. Quanto a esta, destaca-se, nos exemplos das Figuras 3, 4 e 7, a noção gráfica de integral definida, por meio de gráficos, cuja resolução é regulada pela quantidade de partições superficiais, que são definidas pelo elemento infinitesimal superficial. Quanto àquela, dentre alguns exercícios resolvidos, conforme Hayt Junior (2003) e Sadiku (2004), destacam-se os exemplos das integrais de superfície e de volume, bem como os exemplos da aplicação dos operadores diferenciais divergente, gradiente e rotacional como sendo os registros da validade do uso das funções na resolução de exercícios de Eletromagnetismo, culminando assim no ápice deste projeto.



**Figura 5 – Exemplo do layout da interface gráfica: primeira tela**



**Figura 6 – Exemplo do uso da primeira tela: aplicação da integral de linha fechada**

Em relação aos pontos negativos, destacam-se algumas dificuldades decorrentes da adaptação com a linguagem, bem como a complexidade da estruturação lógica de algumas ferramentas desenvolvidas, o que neste caso são consideradas dificuldades naturais para quem trabalha com programação em um nível direcionado para cálculos físicos. Para exemplificar, cita-se o desenvolvimento das funções de integrais de superfície e de volume e principalmente a integral de linha. De fato, estas foram as funções que recorreram da racionalização de uma lógica adequada para realizá-las. Isso porque, com relação à função de integral de linha, deve atender algumas peculiaridades de alguns exercícios, como a inserção de caminhos fechados para uma dada situação por exemplos, e uma das questões resolvidas foi exatamente quando inserir estes caminhos e como inseri-los sem afetar a variável de integração no momento da avaliação da estrutura da função `int`, cuja qual o Matlab realiza integrações de funções.

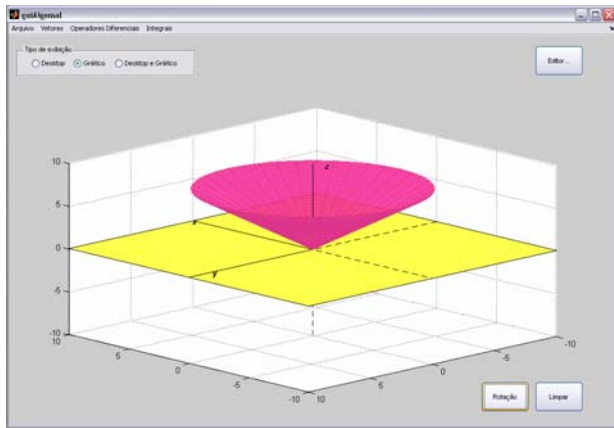


Figura 7 – Exemplo do uso da segunda tela

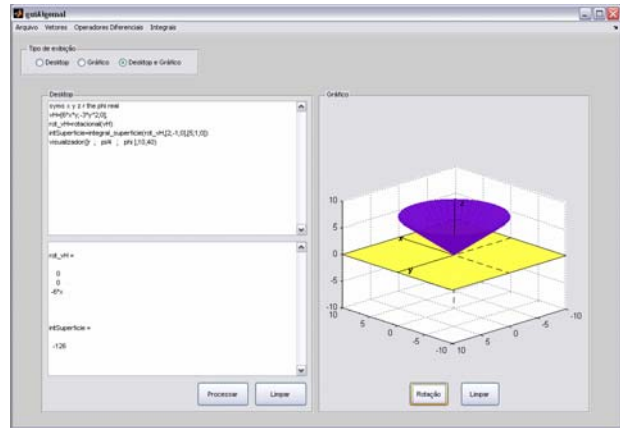


Figura 8 – Exemplo do uso da terceira tela

Com relação a integral de superfície, o problema encontrava-se no fato de encontrar um mecanismo de poder fazer com que a função identificasse se a integral era de superfície aberta ou fechada: isso foi resolvido observando que ao se inserir uma superfície fechada, todas as três coordenadas dos pontos inicial e final eram diferentes, o que não acontecia com a integral aberta, a qual uma das três coordenadas era igual, tanto no ponto inicial quanto no ponto final. Assim construiu-se um algoritmo que fosse sensível a esta informação, e assim pudesse fazer a devida diferença mencionada.

Conforme foi mostrado, chega-se a conclusão deste trabalho, ressaltando principalmente o alcance de todas as metas propostas por este projeto, mesmo com algumas dificuldades tidas no seu decorrer, porém superadas com muito esforço e dedicação, tanto do orientando como do orientador.

## REFERÊNCIAS

CHAPMA, Stephen J. **Programação em Matlab para engenheiros**. São Paulo: Pioneira Thomson Learning, 2003. 482 p.

GONÇALVES, Mirian Buss; FLEMMING, Diva Marília. **CÁLCULO B: funções de várias variáveis, integrais duplas e triplas**. São Paulo: MAKRON Books, 1999. 372 p.

HAYT JUNIOR, Willian H. BUCK, John A. **ELETROMAGNETISMO**. 6. ed. Rio de Janeiro: LTC, c2003. 338 p.

LEITHOLD, Louis. **O CÁLCULO COM GEOMETRIA ANALÍTICA**. 2. ed. São Paulo: HARBRA, c1986. 2 v. (O CÁLCULO COM GEOMETRIA ANALÍTICA, v. 2).

MUNEM, Mustafá A.; FOULIS, David J. **CÁLCULO**. Rio de Janeiro: Ed. Guanabara, c1982. 2 v. (CÁLCULO, v. 1).

SADIKU, Matthew N. O. **elementos de eletromagnetismo**. 3. ed. Porto Alegre: Bookman, 2004. 687 p.

WINTERLE, Paulo. **VETORES E GEOMETRIA ANALÍTICA**. São Paulo: MAKRON Books, 2000. 230 p.