Instrucciones

El objetivo de este ejercicio es resolver de manera numérica la siguiente integral

$$\vec{F}_{C'\to C} = \frac{\mu_0}{4\pi}\int_0^{2\pi}\int_0^{2\pi} a\left(-\sin\varphi\hat{i} + a\cos\varphi\hat{j}\right)\mathrm{d}\varphi' \times \left[a\left(-\sin\varphi'\hat{i} + a\cos\varphi'\hat{j}\right)\mathrm{d}\varphi' \times \frac{(\vec{r}-\vec{r}')}{|\vec{r}-\vec{r}'|^3}\right]$$

Donde $\vec{r}-\vec{r'}$ esta dado por

$$\vec{r}-\vec{r'} = a(\cos\varphi - \cos\varphi')\hat{i} + a(\sin\varphi - \sin\varphi')\hat{j} + d\hat{k}$$

Y entonces $|\vec{r}-\vec{r}'|$ queda de la siguiente forma

$$|\vec{r}-\vec{r}'| = \sqrt{a^2(\cos\varphi - \cos\varphi')^2 + a^2(\sin\varphi - \sin\varphi')^2 + d^2}$$

Dicha integral se trabajo con el fin de hacerla más fácil expresar en el programa, de modo que se separó en tres integrales, las cuales son:

$$A_1 = \frac{\mu_0 a^2}{4\pi}\int_0^{2\pi}\int_0^{2\pi}\frac{-a\cos\varphi\left[\cos\left(\varphi-\varphi'\right)-1\right]\hat{i}}{\left(a^2(\cos\varphi-\cos\varphi')^2 + a^2(\sin\varphi-\sin\varphi')^2 + d^2\right)^{3/2}}\mathrm{d}\varphi\mathrm{d}\varphi'$$

$$A_2 = \frac{\mu_0 a^2}{4\pi}\int_0^{2\pi}\int_0^{2\pi}\frac{-a\sin\varphi\left[\cos\left(\varphi-\varphi'\right)-1\right]\hat{j}}{\left(a^2(\cos\varphi-\cos\varphi')^2 + a^2(\sin\varphi-\sin\varphi')^2 + d^2\right)^{3/2}}\mathrm{d}\varphi\mathrm{d}\varphi'$$

$$A_3 = \frac{\mu_0 a^2}{4\pi}\int_0^{2\pi}\int_0^{2\pi}\frac{-d\cos\left(\varphi-\varphi'\right)\hat{k}}{\left(a^2(\cos\varphi-\cos\varphi')^2 + a^2(\sin\varphi-\sin\varphi')^2 + d^2\right)^{3/2}}\mathrm{d}\varphi\mathrm{d}\varphi'$$

Cargando librerías

[1]:
```
import sys
sys.path.append("0027numeric_integration"0027)
from numeric_integration.integration import Integrate
from numpy import sin, cos, pi
```

Definiendo la función

[2]:
```
def test(x, y):
    return sin(x)*sin(y)
```

[3]:
```
# definiendo el valor de las constantes de la integral:
mu_0 = 1.25663706212*(10**(-6))
a = 1
d = 0.01
pres = 500 #mientras mas grande, mejora la precisión del resultado (aumenta le␣
↪tiempo de calculo)
```

[4]:
```
def A1(x, y):
    return -(a*cos(x)*(cos(x-y)-1))/(((a**2)*(cos(x)-cos(y))**2 +␣
↪(a**2)*(sin(x)-sin(y))**2 + d**2)**(3/2))
```

[5]:
```
def A2(x, y):
    return -(a * sin(x)*(cos(x-y)-1))/(((a**2)*(cos(x)-cos(y))**2 +␣
↪(a**2)*(sin(x)-sin(y))**2 + d**2)**(3/2))
```

[6]:
```
def A3(x, y):
    return -(d*cos(x-y))/(((a**2)*(cos(x)-cos(y))**2 +␣
↪(a**2)*(sin(x)-sin(y))**2 + d**2)**(3/2))
```

Resultado

A1

[7]:
```
# Build an Integrate object
integral = Integrate(A1)

# Calculate the integral
A1_result = integral.double_integral([[0, 2*pi], [0, 2*pi]], precision=pres)

# Show the result
print("The result is", A1_result)
print("\nThe accuracy of this result is", integral.error)
```

```
100%|␣␣␣␣␣␣␣␣␣␣| 3140/3140 [09:39<00:00,  5.42it/s]

The result is 2.4721723827130296e-15

The accuracy of this result is 5.754304774644933e-15
```

A2

[8]:
```
# Build an Integrate object
integral = Integrate(A2)

# Calculate the integral
A2_result = integral.double_integral([[0, 2*pi], [0, 2*pi]], precision=pres)

# Show the result
print("The result is", A2_result)
print("\nThe accuracy of this result is", integral.error)
```

```
100%|␣␣␣␣␣␣␣␣␣␣␣| 3140/3140 [10:21<00:00,  5.06it/s]

The result is 3.179133935467724e-14

The accuracy of this result is 1.1890124482780083e-13
```

A3

[9]:
```
# Build an Integrate object
integral = Integrate(A3)

# Calculate the integral
A3_result = integral.double_integral([[0, 2*pi], [0, 2*pi]], precision=pres)

# Show the result
print("The result is", A3_result)
print("\nThe accuracy of this result is", integral.error)
```

```
100%|␣␣␣␣␣␣␣␣␣␣␣| 3140/3140 [09:14<00:00,  5.67it/s]

The result is -1256.3613304219446

The accuracy of this result is -2.2737367544323206e-13
```

Suma de las integrales

[10]:
```
#componentes

iv = ((mu_0*a**2)/(4*pi))*A1_result
jv = ((mu_0*a**2)/(4*pi))*A2_result
kv = ((mu_0*a**2)/(4*pi))*A3_result
```

Fuerza total

```
[11]: print("0027F ="0027, iv , "0027i +"0027, jv , "0027j +"0027, kv , "0027k"0027  )
```

```
F = 2.4721723840588204e-22 i + 3.1791339371983674e-21 j +
-0.00012563613311058773 k
```

Modulo de la fuerza

```
[12]: print("0027|F| ="0027, iv**2 + jv**2 + kv**2)
```

```
|F| = 1.5784437942981317e-08
```

```
[ ]:
```