



Instituto Tecnológico  
de Buenos Aires

# TRABAJO PRÁCTICO N°3

## ECUACIONES NO-LINEALES

93.54 Métodos Numéricos

Grupo N°4

Legajo N°	Nombre
61428	Kevin Amiel Wahle
61430	Francisco Basili
61431	Nicolás Bustelo

05/05/2022

## Índice

1. Introducción	2
2. Función <i>solver</i> ( $L, l, n$ )	2
3. Función <i>graphRvL</i> ()	2
4. Anexo	3
4.1. Código completo en Python . . . . .	3

## 1. Introducción

El objetivo de este informe es presentar y explicar el funcionamiento de un programa realizado en *Python*, el cual consiste en encontrar la solución a una ecuación no-lineal.

## 2. Función *solver(L,l,n)*

## 3. Función *graphRvL()*

## 4. Anexo

### 4.1. Código completo en Python

```

1 # -----
2 # @file      +mri.py+
3 # @brief     +Ecuaciones no-lineales+
4 # @author    +Grupo 4+
5 # -----
6
7 # -----
8 # LIBRARIES
9 # -----
10 from termios import NL1
11 import numpy as np
12 import pandas as pd
13 import matplotlib.pyplot as plt
14 import math as mt
15
16 # -----
17 # FUNCTION DEF
18 # -----
19 # Resolución de ecuaciones no-lineales
20 def solver(L,l,n):
21     mu = 4 * np.pi * 10**(-7) # Permeabilidad del vacío
22     r = 1 # Radio del solenoide en metros
23     L = ((mu * n**2 * np.pi * r**2) / (l**2)) * (np.sqrt(r**2 + l**2) - r)
24     #Hacer la función que resuelva la ecuación!
25
26     return r
27
28 def graphRL():
29     l = 0.2 # Longitud del solenoide en metros
30     L = np.linspace(1e-9,100e-9,1000) # Inductancia del solenoide H
31     N1 = 10
32     N2 = 100
33     N3 = 1000
34
35     r1 = solver(L,l,N1)
36     r2 = solver(L,l,N2)
37     r3 = solver(L,l,N3)
38
39     plt.plot(L,r1, label = 'N = 10')
40     plt.plot(L,r2, label = 'N = 100')
41     plt.plot(L,r3, label = 'N = 1000')
42     plt.xlabel('Inductancia [H]')
43     plt.ylabel('Radio [m]')
44     plt.title('Radio vs Inductancia')
45     plt.legend()
46     plt.show()

```