

# 1- Python: Métodos de Newton Raphson y Punto fijo para solución de ecuaciones, Euler y Taylor para ecuaciones diferenciales.

## 2- Autores

GUILLERMO ANDRES GOMEZ JIMENEZ	<a href="mailto:guille_1031@hotmail.com">guille_1031@hotmail.com</a>
JESUS DAVID MACIAS ALVAREZ	<a href="mailto:jixus0123@hotmail.com">jixus0123@hotmail.com</a>
RAYMUNDO DIAZ VEGA	<a href="mailto:raydiazvega@hotmail.com">raydiazvega@hotmail.com</a>
YARED HENRIQUEZ BLANQUICETT	<a href="mailto:yared2017@gmail.com">yared2017@gmail.com</a>

## 3- Ingeniería en Sistemas

## 4- Introducción al problema

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma. Python trabajará con un módulo llamado matplotlib el cual se encargará de las partes graficas de nuestro programa.

Matplotlib es una biblioteca para la generación de gráficos a partir de datos contenidos en listas o arrays en el lenguaje de programación Python y su extensión matemática NumPy. Proporciona una API, pylab, diseñada para recordar a la de MATLAB. Estas dos en la parte web se unirán con Flask.

Flask es un framework minimalista escrito en Python que permite crear aplicaciones web rápidamente y con un mínimo número de líneas de código. Está basado en la especificación WSGI de Werkzeug y el motor de templates Jinja2 y tiene una licencia BSD.

## 5- Problema o situación

Lograr que el usuario tenga la opción de resolver ecuaciones con el uso de distintos métodos numéricos.

## 6- Proceso de solución

Paso1: En el navegador de su preferencia descargar Python en el sitio oficial y realizar la configuración recomendada. Para su facilidad le brindaremos un proceso de guía para instalación en Windows.

Enlace1: <https://www.python.org/downloads/windows/>

Paso2: En el proceso de instalación es importante agregar la ruta en el PATH de ruta en Windows.

Paso3: Luego de la correcta instalación de Python, acceder al cmd en modo administrador para instalar pip.

Enlace2: <https://recursospython.com/guias-y-manuales/instalacion-y-utilizacion-de-pip-en-windows-linux-y-os-x/>

Paso4: Al instalar el pip podrás instalar los módulos de Python que necesites, entre esos módulos se encuentra matplotlib.

Enlace3: <https://programminghistorian.org/es/lecciones/instalar-modulos-python-pip>

Paso5: Se realiza primero la programación de los métodos en Python con matplotlib.

Paso6: Se revisará la documentación de flask para que los programas hechos puedan visualizarse en la web. Aquí es importante el llamado a los métodos.

## 7- Codificación

Index.html página principal donde estarán contenidos los métodos y los enlaces a dichos métodos con un diseño responsive, es decir un diseño adaptable.

```
</div>
</div>
</nav>

<div id="fh5co-work">
  <div class="container">
    <div class="row top-line animate-box">
      <div class="col-md-7 col-md-push-5 text-left intro">
        <h1>Grupo de Python con flask para metodos numericos - UTB </h1>
        <h2><span class="fh5co-highlight">Made with <i class="icon-heart2"></i> by <a href="#">Grupo Python</a> </span></h2>
        <!-- <h2>Shift is a Collection of a Beautiful &amp; Premium Themes.</h2> -->
      </div>
    </div>
    <div class="row">
      <div class="col-md-4 text-center animate-box">
        <a class="work" href="Newton.html">
          <div class="work-grid" style="background-image:url(images/project-1.jpg);">
            <div class="inner">
              <div class="desc">
                <h3>Newton</h3>
                <span class="cat">1</span>
              </div>
            </div>
          </div>
        </a>
      </div>
      <div class="col-md-4 text-center animate-box">
        <a class="work" href="Puntofijo.html">
          <div class="work-grid" style="background-image:url(images/project-2.jpg);">
            <div class="inner">
              <div class="desc">
                <h3>Punto Fijo</h3>
                <span class="cat">2</span>
              </div>
            </div>
          </div>
        </a>
      </div>
    </div>
  </div>
</div>
```

Se puede evidenciar el diseño en el siguiente enlace: <https://youtu.be/VGvOtm-HGnU>

-Forms

```
File Edit Selection Find View Goto Tools Project Preferences Help

main.py forms.py Newton.html puntofijo.html

1 from wtforms import Form, TextField
2
3 class CommentForm(Form):
4     ecuacion = TextField('Digite la ecuacion:')
5     #variable de newton y punto fijo
6     inicio = TextField('Digite el valor inicial:')
7     grado = TextField('Digite el grado de tolerancia:')
8     #variable de euler y taylos
9     A = TextField('Digite el grado de tolerancia:')
10    B = TextField('Digite el grado de tolerancia:')
11    H = TextField('Digite el grado de tolerancia:')
12
```

Usamos esa clase para guardar todas las variables posibles usadas en el programa

## -Macro

```
File Edit Selection Find View Goto Tools Project Preferences Help
main.py forms.py base.html x _macro.html x Newton.html
1 {% macro render_field(field) %}
2     <td> {{ field.label }}
3     <dd> {{ field(**kwargs) |safe }}
4 {% endmacro %}
```

Acá heredaremos todas las variables llamadas desde flask (comment\_form.NOMBRE DE LA VARIABLE.data) será la sintaxis que usaremos desde flask para declarar las variables.

```
13 @app.route('/Newton.html', methods = ['GET', 'POST'])
14 def new(t = "", m = "", df = ""):
15     comment_form = forms.CommentForm(request.form)
16     if request.method == 'POST':
17         t, m, df= Newton(comment_form.ecuacion.data,comment_form.inicio.data,comment_form.grado.data).resultado()
18     return render_template('Newton.html', form=comment_form, decuacion=str(df), tabla = t, mensaje = m)
```

## -HTML

```
File Edit Selection Find View Goto Tools Project Preferences Help
main.py forms.py Newton.html puntofijo.html Euler.html
1 <html>
2     <head>
3         <title> Newton </title>
4     </head>
5     <body>
6         {% block content %}
7         {% from "_macro.html" import render_field %}
8         <form name="sentmessage" id="contactForm" method="POST">
9
10             <div class="row control-group">
11                 <div class="form-group col-xs-12 floating-label-form-group controls">
12                     {{ render_field(form.ecuacion) }}
13                     <p class="help-block text-danger"></p>
14                 </div>
15             </div>
16
17             <div class="row control-group">
18                 <div class="form-group col-xs-12 floating-label-form-group controls">
19                     {{ render_field(form.inicio) }}
20                     <p class="help-block text-danger"></p>
21                 </div>
22             </div>
23             <div class="row control-group">
24                 <div class="form-group col-xs-12 floating-label-form-group controls">
25                     {{ render_field(form.grado) }}
26                     <p class="help-block text-danger"></p>
27                 </div>
28             </div>
29             <div id="success"></div>
30             <div class="row">
31                 <div class="form-group col-xs-12">
32                     <button type="submit" class="btn btn-default">Generar</button>
33                 </div>
34             </div>
35         </form>
36     {% endblock %}
37     {% if decuacion != "" %}
```

Esta será la forma la cual heredaremos las variables de la clase de forms para crear su cuadro de texto para que el usuario pueda digitar las variables, solo es copiar y pegar la cantidad de variables que necesite el método a usar.

Ejemplo con Newton Rapson:



Link de la grafica: <https://drive.google.com/file/d/1xD1CnYYam4xFazTyAachU8XVCR-OhUrP/view?usp=sharing>

## 8- Conclusiones

Los métodos numéricos son técnicas mediante las cuales es posible formular problemas matemáticos de tal forma que puedan resolverse usando operaciones aritméticas.

Los métodos numéricos nos vuelven aptos para entender esquemas numéricos a fin de resolver problemas matemáticos, de ingeniería y científicos en una computadora, reducir esquemas numéricos básicos, escribir programas y resolverlos en una computadora y usar correctamente el software existente para dichos métodos y no solo aumenta nuestra habilidad para el uso de computadoras, sino que también amplía la pericia matemática y la comprensión de los principios científicos básicos.

El análisis numérico trata de diseñar métodos para “aproximar” de una manera eficiente las soluciones de problemas expresados matemáticamente.

El objetivo principal del análisis numérico es encontrar soluciones “aproximadas” a problemas complejos utilizando sólo las operaciones más simples de la aritmética. Se requiere de una secuencia de operaciones algebraicas y lógicas que producen la aproximación al problema matemático.

En el ámbito de la programación nos facilita mucho más la eficiencia y la facilidad en el manejo de las ecuaciones haciendo muchos cálculos por segundo.

## **9- Bibliografía**

- Métodos numéricos para ingenieros, Libro de Raymond P. Canale
- Métodos numéricos con MATLAB, John H. Mathews, Kurtis D. Fink