

Enseñanza y aplicación de métodos numéricos en problemas de ingeniería, utilizando la potencialidad de la inteligencia artificial (ChatGPT) en el aula combinada con programación en Python y entorno de trabajo colaborativo en Google Colab

Diego Federico Amiconi

Universidad Tecnológica Nacional, Facultad Regional La Plata, Av. del Petróleo Argentino 417, (1923) Berisso, Pcia. de Buenos Aires, Argentina

Diegoamiconi@frlp.utn.edu.ar

Recibido el 5 de octubre de 2023, aprobado el 13 de noviembre de 2023

Resumen

La inteligencia artificial (IA) se encuentra presente ya de una manera indiscutible en la sociedad y en muchos ámbitos académicos y laborales se han comenzado a utilizar las diferentes herramientas de IA disponibles en línea que hay en la actualidad como ser por ejemplo ChatGPT (Openai, 2023), Midjourney (Midjourney, 2023) y Dall-e 2 (Openai, 2023). En este trabajo se explicita lo que se considera una experiencia innovadora en la enseñanza y aplicación de diferentes métodos numéricos utilizados en la cátedra de Análisis Numérico y Cálculo Avanzado para resolver problemas de ingeniería, aprovechando como herramienta novedosa la potencialidad de la IA de ChatGPT, combinado con la generación y programación de los diferentes métodos numéricos usando código en lenguaje Python y la plataforma de trabajo Google Colab, lo que nos permite analizar el funcionamiento de los diferentes algoritmos generados y que las y los estudiantes puedan resolver luego ejemplos aplicados. Google Colab permite a su vez la posibilidad de editar estos algoritmos y trabajarlos en forma colaborativa a través de la formación de diferentes grupos de trabajo formados dentro del curso por las y los estudiantes, para mejorar estos algoritmos y hacer diversas pruebas con ejemplos relacionados a la carrera de Ingeniería Industrial.

PALABRAS CLAVE: MÉTODOS NUMÉRICOS - INGENIERÍA INDUSTRIAL - INTELIGENCIA ARTIFICIAL - PROGRAMACIÓN PYTHON - GOOGLE COLAB

Abstract

Artificial intelligence (AI) is already present in an indisputable way in society and in many academic and work fields they have begun to use the different AI tools available online today, such as ChatGPT (Openai, 2023), Midjourney (Midjourney, 2023) and Dall-e 2 (Openai, 2023). From the National Technological University (UTN), La Plata regional faculty, we want to show in this work what we consider an innovative experience in the teaching and application of different numerical methods used in the course of Numerical Analysis and Advanced Calculation to solve engineering problems, taking advantage of the potential of the AI of ChatGPT as a novel tool, combined with the generation and programming of the different numerical methods using code in the Python language and the Google Colab work platform, which allows us to analyze the operation of the different algorithms generated and that the students can then solve applied examples. Google Colab in turn allows the possibility of editing these algorithms and working on them collaboratively through the formation of different work groups within the course formed by the students, to improve these algorithms and carry out various tests with examples related to the Industrial Engineering degree.

KEYWORDS: NUMERICAL METHODS - INDUSTRIAL ENGINEERING - ARTIFICIAL INTELLIGENCE - PYTHON PROGRAMMING - GOOGLE COLAB

Introducción

El desarrollo de esta experiencia educativa consistió inicialmente en explicarles a las y los estudiantes de Análisis Numérico y Cálculo Avanzado de tercer año de ingeniería Industrial de la Facultad Regional La Plata de la Universidad Tecnológica Nacional (UTN-FRLP) diferentes métodos numéricos que se desarrollan a lo largo del curso con sus formulaciones teóricas, condiciones y formas de aplicación (Burden, 2011).

Se trata de un trabajo que se desarrolla durante el ciclo 2023, la experiencia muestra muy buenos resultados iniciales desde el punto de vista de cómo las y los estudiantes se sorprenden inicialmente al ver la potencialidad de la IA como herramienta de acompañamiento en sus estudios, su interés en aprender a utilizar programación para el desarrollo de sus trabajos de Análisis Numérico y Cálculo Avanzado y a su vez hacer extensiva esta habilidad a otras materias, como así también su buena predisposición a trabajar en grupos.

Luego en una segunda instancia se pasó a trabajar con ChatGPT y la potencia de su inteligencia artificial en la formulación de problemas de ingeniería para ser resueltos utilizando los diferentes métodos numéricos vistos en las clases. Una vez analizados y verificados los problemas generados por ChatGPT, se le pidió a la IA la generación de los códigos de programación en Python, lo que este tipo de IA trabaja de una manera muy natural y completa y con los códigos generados se trabajó por último en la plataforma de Google Colab con la resolución de los problemas de ejemplo, una explicación de los algoritmos detallando y editando las partes que lo componen como ser ajustes de variables, declaración de funciones, ciclos de búsqueda y el posterior análisis final de los resultados obtenidos.

Objetivos de la experiencia

El desarrollo de esta experiencia tuvo como objetivos principales de la cátedra los siguientes puntos:

- a) Generar nuevas experiencias de enseñanza-aprendizaje en los cursos.
- b) Mostrar la potencialidad de la IA en la actualidad al utilizarla como una herramienta complementaria para la resolución de problemas de ingeniería de diferentes áreas.
- c) Explorar el conocimiento que tienen las y los estudiantes sobre nociones de programación e introducir a quienes no habían tenido experiencia en programación anteriormente, en los conceptos básicos y fundamentales de un algoritmo y la elaboración e interpretación de sus códigos.
- d) Destacar como se viene realizando en trabajos de años anteriores la importancia de la aplicación de los diferentes métodos numéricos para resolver problemas de Ingeniería, no sólo como una alternativa de cálculo con respecto a otros métodos sino como una herramienta necesaria para aquellos casos en que la resolución exacta resulte imposible o de difícil obtención.
- e) Fomentar el trabajo y aprendizaje colaborativo en grupos, permitiendo que las y los estudiantes interactúen entre sí buscando metas de aprendizaje en común, resuelvan problemas y a su vez pueden compartir conocimientos y experiencias individuales enriqueciéndose unos de otros.

Descripción de las herramientas utilizadas para el trabajo

1- ChatGPT: Es un modelo de lenguaje basado en inteligencia artificial desarrollado

por OpenAI. Utiliza la arquitectura de transformadores, específicamente el modelo GPT (*Generative Pre-trained Transformer- Transformador Generativo Preentrenado*), para generar respuestas coherentes y contextualmente relevantes en conversaciones de texto. El modelo ChatGPT se entrena con grandes cantidades de texto de conversaciones para aprender las estructuras y patrones del lenguaje natural en diálogos. Utiliza técnicas de aprendizaje automático para capturar la información contextual y generar respuestas que se asemejan a las que un humano podría dar en una conversación similar. ChatGPT es capaz de entender preguntas, solicitudes y otros tipos de entradas en lenguaje natural, y puede generar respuestas apropiadas y relevantes. Aunque no tiene conocimiento específico sobre un dominio o tema en particular, puede utilizar información general del lenguaje aprendida durante el entrenamiento para proporcionar respuestas útiles y coherentes. (Nota: Definición generada por el mismo ChatGPT).

2- Python: Es un lenguaje de programación de alto nivel ampliamente utilizado en la informática y la ciencia de datos. Se caracteriza por su sintaxis simple y legible, lo que lo hace accesible para principiantes, así como por su flexibilidad y amplia comunidad de desarrolladores. Python se utiliza en una variedad de aplicaciones, desde desarrollo web y automatización hasta análisis de datos y aprendizaje automático, lo que lo convierte en una herramienta esencial en la investigación y la programación en general

3- Google Colab: Es una plataforma que fue creada por la empresa Google, que permite correr códigos de programación en Python y generar informes (del tipo Notebook, similar a la plataforma Jupiter Notebook) sin la necesidad de instalar nada en los equipos más que tener un navegador web. Los recursos de espacio en disco y memoria ram necesarias para trabajar son asignadas en línea por los servidores de Google. En nuestro caso, lo consideramos muy útil para aquellos estudiantes que no tienen experiencia en configurar entornos de trabajo que a veces pueden ser complejos y que no disponen de los recursos de hardware necesarios y a su vez nos permite trabajar en las aulas teniendo que disponer solamente de equipos con conexión a internet. Colab tiene incorporadas a su vez muchas bibliotecas de programación ya preinstaladas como ser entre otras Numpy (Librería de funciones numéricas relacionadas con Algebra Lineal), SciPy (Librería de funciones para matemáticas avanzadas), Sympy (Librería de manejo matemático simbólico) o Matplotlib (Librería para graficar funciones) que resultan de mucha utilidad para implementar proyectos de programación.

Métodos numéricos desarrollados

Un método numérico es una técnica utilizada en matemáticas y ciencias de la computación que consiste en un procedimiento mediante el cual se obtiene, casi siempre de manera aproximada, la solución de ciertos problemas realizando cálculos puramente aritméticos y lógicos (operaciones aritméticas elementales, cálculo de funciones, consulta de una tabla de valores, cálculo proposicional, etc.). Tal procedimiento consiste de una lista finita de instrucciones precisas que especifican una secuencia de operaciones algebraicas y lógicas (algoritmo), que producen o bien una aproximación de la solución del problema (solución numérica) o bien un mensaje. Estos métodos involucran también la discretización de los problemas continuos y su resolución a través de iteraciones y cálculos aproximados. La eficiencia en el cálculo de dicha aproximación depende, en parte, de la facilidad de implementación del algoritmo y de las características especiales y limitaciones de los instrumentos de cálculo (las computadoras). (Rodríguez, 2016)

En el trabajo realizado se desarrollaron y resolvieron problemas de diferentes métodos que se mencionan a continuación:

1- Cálculo de raíces para funciones continuas de una variable

Podemos definir a una raíz de una función continua como aquel valor de "x" que satisface

la ecuación " $f(x) = 0$ ", donde " $f(x)$ " es la función continua con la que estamos trabajando. Encontrar estos valores que hacen que $f(x)=0$ es de múltiple utilidad para muchos problemas de Ingeniería.


a- Método de Bisección

El método de bisección es un método numérico utilizado para encontrar raíces de una función continua en un intervalo dado. Este método se basa en el teorema del valor intermedio, que establece que, si una función es continua en un intervalo cerrado y tiene signos opuestos al evaluarla en los extremos del intervalo, entonces existe al menos un punto en el intervalo donde la función se anula (es decir, tiene una raíz).

Para cada uno de los métodos se consultó a la IA de ChatGPT sobre el método en cuestión, se analizaron y verificaron las respuestas y luego se le solicitó programarlo con código Python.

DI Podrías explicarme la técnica de bisección para calcular la raíz de la siguiente ecuación? $f(x) = x^3 - 2x^2 + 3x - 2$, el intervalo inicial puedes generarlo

Fig. 1. Ejemplo de consulta Método de Bisección para ChatGPT

 ¡Por supuesto! La técnica de bisección es un método numérico utilizado para encontrar la raíz de una función en un intervalo dado. A continuación, te explicaré cómo puedes aplicar la técnica de bisección para calcular la raíz de la ecuación $f(x) = x^3 - 2x^2 + 3x - 2$.

La idea básica de la técnica de bisección es dividir repetidamente el intervalo inicial en dos partes iguales y determinar en qué subintervalo se encuentra la raíz de la función. Aquí están los pasos:

Fig. 2: Respuesta generada por ChatGPT

DI Podrías pasarme ese ejemplo con código Python para poder practicar la programación?

Fig. 3. Consulta de programación Python

```

def f(x):
    return x**3 - 2*x**2 + 3*x - 2

def bisection_method(f, a, b, tol):
    print("La función en a es:", f(a))
    print("La función en b es:", f(b))

    if f(a)*f(b) >= 0:
        print("El método de bisección no es aplicable en este intervalo.")
        return None
    else:
        print("Cumple con Bolzano")

    c = (a + b)/2
    n = 0

    while abs(b-a) >= tol:
        c = (a + b)/2
        if f(c) == 0:
            break
        elif f(c)*f(a) < 0:
            b = c
        else:
            a = c
        n += 1

    return c, n

a = 0
b = 1.2
tol = 0.00000001

if f(a)*f(b) < 0:
    root, iterations = bisection_method(f, a, b, tol)
    print("La raíz es: {:.4f}.".format(root))
    print("El número de iteraciones necesarias para obtener la solución fue de {}.".format(iterations))
else:
    print("El intervalo proporcionado no contiene una raíz de la función.")

```

Fig. 4. Código Generado por ChatGPT y corregido luego en Google Colab (Google, s.f.)

```

La función en a es: -2
La función en b es: 0.4479999999999995
Cumple con Bolzano
La raíz es: 1.0000.
El número de iteraciones necesarias para obtener la solución fue de 27.

```

Fig. 5. Resultados obtenidos para el problema propuesto

b- Método de Newton (Raphson)

El método de Newton-Raphson, más conocido como el método de Newton, es un método numérico utilizado para encontrar raíces de una función. A diferencia del método de bisección, el método de Newton-Raphson utiliza información adicional sobre la derivada de la función para iterativamente aproximar la raíz y solo un punto inicial en vez de un intervalo.

DI Podrías explicarme la técnica de Newton para calcular la raíz de la siguiente ecuación?
 $f(x) = x^3 - 5x^2 + 10x - 5$, el punto inicial puedes generarlo

Fig. 6. Ejemplo de consulta Método de Newton para ChatGPT

¡Claro! La técnica de Newton, también conocida como método de Newton-Raphson, es un método numérico utilizado para encontrar raíces de funciones. Permite encontrar una raíz aproximada de la ecuación $f(x) = 0$ utilizando un punto inicial cercano a la raíz buscada. A continuación, te explicaré cómo aplicar el método de Newton para calcular la raíz de la ecuación $f(x) = x^3 - 5x^2 + 10x - 5$.

Fig. 7. Respuesta generada por ChatGPT

DI Podrías pasarme ese ejemplo con código Python para poder practicar la programación?

Fig. 8. Consulta de programación Python

¡Por supuesto! Aquí tienes un ejemplo de código en Python que implementa el método de Newton para calcular la raíz de la función $f(x) = x^3 - 5x^2 + 10x - 5$:

```
def f(x):
    return x**3 - 5*x**2 + 10*x - 5

def df(x):
    return 3*x**2 - 10*x + 10

def newton_method(f, df, x0, tol):
    n = 0

    while abs(f(x0)) >= tol:
        if df(x0) == 0:
            print("Error: la derivada de la función evaluada en el punto actual es igual a cero.")
            return None, None
        x0 = x0 - f(x0)/df(x0)
        n += 1

    return x0, n

x0 = 1.5
tol = 0.0001

root, iterations = newton_method(f, df, x0, tol)
print("La raíz de la función es: {:.4f}.".format(root))
print("El número de iteraciones necesarias para obtener la solución fue de {:.format(iterations))
```

Fig. 9. Código Generado por ChatGPT y corregido luego en Google Colab (Google, s.f.)

La raíz de la función es: 0.7243.
 El número de iteraciones necesarias para obtener la solución fue de 4.

Fig. 10. Resultados obtenidos para el problema propuesto

2- Resolución de sistemas de ecuaciones lineales

Un sistema de ecuaciones lineales es un conjunto de ecuaciones lineales que se deben resolver simultáneamente. La solución de un sistema de ecuaciones lineales implica encontrar los valores de las incógnitas que satisfacen todas las ecuaciones del sistema. Resolver un sistema de ecuaciones lineales de forma iterativa implica utilizar un método que realiza una serie de pasos repetitivos para aproximar gradualmente la solución.

a - Método de Jacobi

El método de Jacobi es un método iterativo en el cual si se trabaja con una matriz que es diagonalmente dominante, se parte de un vector inicial para calcular las incógnitas de una iteración basándose en las soluciones obtenidas anteriormente. El método se aplica hasta encontrar una solución dentro de una tolerancia de error definida.

```
import numpy as np

# Definimos la matriz de coeficientes A y el vector de términos
independientes b
A = np.array([[3, 1, 1], [2,5, 2], [2, 1, 7]])
b = np.array([6, 4, 27])

# Definimos el vector inicial de la solución x
x0 = np.zeros_like(b)

# Definimos la tolerancia y el número máximo de iteraciones
tol = 1e-6
max_iter = 1000

# Definimos la función que implementa el método de Jacobi
def jacobi(A, b, x0, tol, max_iter):
    n = len(A)
    x = x0.copy()
    iter_num = 0
    error = np.inf

    # Comprobamos si la matriz es diagonalmente dominante
    if not is_diagonally_dominant(A):
        print("La matriz no es diagonalmente dominante.")
        return None

    while error > tol and iter_num < max_iter:
        x_prev = x.copy()
        for i in range(n):
            sigma = 0
            for j in range(n):
                if j != i:
                    sigma += A[i][j] * x_prev[j]
            x[i] = (b[i] - sigma) / A[i][i]
            iter_num += 1
        error = np.linalg.norm(x - x_prev) / np.linalg.norm(x)
    print("La cantidad de iteraciones necesarias fue de:", iter_num)
```



```

if iter_num == max_iter:
    print("El método de Jacobi no converge.")
    return None

return x

# Definimos la función que comprueba si una matriz es
# diagonalmente dominante
def is_diagonally_dominant(A):
    n = len(A)
    for i in range(n):
        diagonal_sum = np.sum(np.abs(A[i])) - np.abs(A[i][i])
        if np.abs(A[i][i]) <= diagonal_sum:
            return False
    return True

# Llamamos a la función jacobi para obtener la solución del sistema
x = jacobi(A, b, x0, tol, max_iter)

# Imprimimos la solución si el método de Jacobi converge
if x is not None:
    print("Solución:", x)

```

Fig. 11. Código Generado por ChatGPT y corregido luego en Google Colab⁽⁵⁾

```

La cantidad de iteraciones necesarias fue de: 4
Solución: [1 0 3]

```

Fig. 12. Resultados obtenidos para el problema propuesto

b - Método de Gauss – Seidel

El método de Gauss-Seidel es muy semejante al método de Jacobi. Mientras que en el método de Jacobi se utiliza el valor de las incógnitas de la iteración anterior (o del vector arrancador en la primera iteración) para determinar una nueva aproximación, en cambio en el método de Gauss-Seidel se van utilizando los valores de las incógnitas recién calculadas en la misma iteración combinados con valores de la iteración anterior en caso de necesitar el valor de alguna variable que todavía no se ha calculado en la iteración actual. Es decir, combina valores de la iteración anterior con los de la iteración actual. La ventaja que presenta esta combinación de valores de distintas iteraciones es que, si el sistema es diagonalmente dominante, se garantiza la convergencia y este método es más rápido en convergencia a la solución que el método de Jacobi, es decir, que el método se acelera.

```

# Definimos la matriz A y el vector b
A = [[5, -1, 2, 0], [-1, 4, -1, 1], [2, -1, 5, -1], [0, 1, -1, 3]]
b = [2, 1, -5, 3]

# Verificamos si la matriz es diagonalmente dominante
dominante = True
for i in range(len(A)):
    suma = 0
    for j in range(len(A)):
        if i != j:
            suma += abs(A[i][j])
    if abs(A[i][i]) <= suma:
        dominante = False
        break

# Si la matriz es diagonalmente dominante, aplicamos el método de
Gauss-Seidel
if dominante:
    # Definimos la precisión deseada y un valor inicial para cada
variable
    precisión = 0.0001
    x0 = [0, 0, 0, 0]

    # Iteramos hasta que se alcance la precisión deseada
iteraciones = 0 #Inicializamos el contador de iteraciones
    while True:
        x = []
        for i in range(len(A)):
            suma = 0
            for j in range(len(A)):
                if i != j:
                    suma += A[i][j] * x0[j]
            xi = (b[i] - suma) / A[i][i]
            x.append(xi)

        # Comprobamos si se ha alcanzado la precisión deseada
iteraciones += 1 #Contamos las iteraciones
        diferencia = [abs(x[i] - x0[i]) for i in range(len(A))]
        if max(diferencia) < precisión:
            break
        x0 = x

    # Imprimimos la solución
    print("La solución es:", x)
    print("La cantidad de iteraciones necesarias fue de:", iteraciones)
else:
    print("La matriz no es diagonalmente dominante, no se puede
aplicar el método de Gauss-Seidel")

```

Fig. 13. Código Generado por ChatGPT y corregido luego en Google Colab⁽⁵⁾

```

La solución es: [0.9000336013286194, 0.019960984528177877, -1.239959696704862, 0.5799628644447379]
La cantidad de iteraciones necesarias fue de: 26

```

Fig. 14. Resultados obtenidos para el problema propuesto

3- Otros Métodos

De la misma manera se está trabajando actualmente con la resolución de sistemas de Ecuaciones Lineales por el método de Relajación y de Crout, como así también en la resolución de problemas utilizando las técnicas de Regresión o Ajuste por Mínimos Cuadrados (Casos Lineal, potencial, exponencial, polinómicos y de crecimiento Saturado)

Aspectos positivos y dificultades encontrados

Como aspectos positivos podemos destacar por un lado la posibilidad de disponer en línea de una IA tan potente como lo es ChatGPT como herramienta de soporte para realizar prácticas de todo lo visto en el curso (y también para otros cursos/temas), por otro lado, el interés de las y los estudiantes en aprender a utilizar programación para resolver problemas (en este caso bajo código Python) como una habilidad y competencia adicional a lo que se ve habitualmente en el curso. Cabe destacar en este aspecto que al ser estudiantes de la carrera de Ingeniería Industrial no poseen por lo general conocimientos de programación como puede pasar en otras carreras de la regional (por ejemplo, ingeniería en sistemas). El hecho de ver funcionar los códigos que ellos mismos programan les genera siempre mucha satisfacción. También como aspecto positivo podemos mencionar la visualización de los métodos numéricos como una forma potente de solucionar problemas que no tienen solución directamente en forma exacta o que son muy difíciles de implementarlos de manera exacta frente a métodos que por ahí con una metodología bastante simple suelen resolver problemas muy complejos.

En cuanto a las dificultades encontradas se puede mencionar algunas respuestas “erróneas” dadas por la IA, algunos errores en fórmulas, o ejemplos pedidos que al querer correrlos en la plataforma de Google Colab no terminan siendo adecuados o tienen errores, por eso se recuerda siempre con énfasis a las y los estudiantes el hecho de tener que saber primero sobre los contenidos para poder utilizar adecuadamente la IA y poder verificar que lo que nos está contestando sea siempre correcto. Se les hace hincapié en realizar las preguntas de una manera adecuada y poseer los conocimientos sobre el tema como para interpretar correctamente la respuesta dada por la IA. Lo que internamente llamamos el “Aprender a preguntar y comprender la respuesta”.

Conclusiones

Esta experiencia educativa nos muestra que estamos frente a una revolución con las herramientas que están apareciendo de IA y eso no se puede negar, lo cual nos lleva a plantearnos como docentes cual será nuestro rol, considerando en lo personal que debe ser el de guías fundamentalmente en el uso de estas herramientas y en que debemos tener capacitación constante en estos temas y generar de esta manera experiencias innovadoras para poder trabajar con las y los estudiantes en nuestros cursos.

El uso de la IA como copiloto de nuestras tareas como docentes, nos puede potenciar de una manera en la que los aprendizajes sean cada vez más significativos y enriquecedores por parte de nuestros grupos de estudiantes y para nosotros mismos.

En cuanto a los objetivos esperados, los resultados han sido satisfactorios, la experiencia fue muy bien recibida por las y los estudiantes, el uso de la IA les pareció de mucha utilidad como herramienta de soporte para la resolución de problemas de este curso y de otros, se pudo explorar las nociones que tenían sobre programación y de un curso por ejemplo de 20 estudiantes, solo uno tenía experiencia en haber programado algo anteriormente, lo cual permitió con esta experiencia introducir y fortalecer el desarrollo de habilidades y competencias asociadas a la programación para todo el grupo.

A su vez el trabajar de manera colaborativa ayuda a la comprensión y desarrollo de los temas propios de la materia, donde se pudo apreciar la potencia de los métodos numéricos dados.

Agradecimientos

Deseo agradecer a mi hijo Gaetano y a mi esposa Karina por ser el motor de todo lo que hago, y a la Universidad Tecnológica Nacional (UTN-FRLP) por haberme dado la posibilidad de estudiar, graduarme, ser docente e investigador de esta casa de altos estudios.

Referencias

- BURDEN, R. L. (2011). *Análisis Numérico*. 9th Revised edition. Cengage Learning Editores S.A.
- Google. (s.f.). <https://colab.google/>.
- Midjourney. (2023). <https://www.midjourney.com/>.
- Openai. (2023). <https://chat.openai.com/>.
- Openai. (2023). <https://openai.com/blog/dall-e-2-extending-creativity>.
- RODRÍGUEZ, L., (2016). "ANÁLISIS NUMÉRICO BÁSICO. Un enfoque algorítmico con el soporte de Python", Libro digital, Versión 4.4 . Guayaquil Ecuador: Departamento de Matemáticas, Facultad de Ciencias Naturales y Matemáticas (FCNM), ESPOL.