

Universidad Centroamericana "José Simeón Cañas"

Facultad de Ingeniería y Arquitectura

Departamento de Matemática

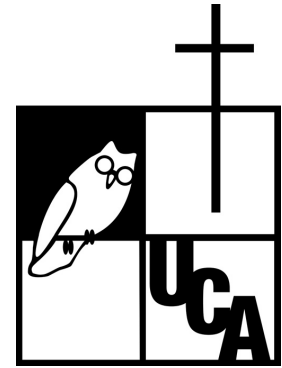
Análisis Numérico

Ciclo 01-2024

Ing. José Daniel Juárez

Secciones 01 y 02

Proyecto 1



1 Información General

- **Descripción:** Este proyecto tiene como objetivo que los estudiantes investiguen, implementen y pongan en práctica diversos experimentos numéricos; comprendiendo la teoría que los sustenta y utilizando herramientas relevantes para producir, comunicar y evaluar resultados computacionales y además, presentar dichos resultados de una manera académica y matemáticamente rigurosa.
- **Modo de trabajo:** Se desarrollará en equipos de 4 personas como máximo. Estos equipos pueden estar conformados por estudiantes de diferentes secciones.
- **Fecha de entrega:**
 - Consultar en el SRN
 - Entregar a las 23:00 horas del día estipulado.
 - No se responderán consultas la semana de entrega.
- **Nota y calificación:** El Proyecto 1 corresponde al %15 de la nota final de la materia. Se realizarán 4 de los 6 problemas presentados, cada uno con una ponderación del %25 de la nota del proyecto.

La única condición respecto a la selección de los problemas, es que tanto el Problema 1 y el Problema 6 deben realizarse obligatoriamente, y los otros quedan a su elección.
- **Productos a entregar:**
 - Notebook de Jupyter en formato `.ipynb`

2 Indicaciones

Se elaborará un Notebook de Jupyter en [Google Colaboratory](#), donde se realicen distintos experimentos numéricos con los problemas seleccionados. Dichas implementaciones deberán ser aplicadas también a casos particulares, ilustrando con gráficos y tablas en caso de ser oportuno. Los elementos ilustrativos deberán estar implementados con el lenguaje de programación (es decir, no se deben utilizar imágenes obtenidas de otras fuentes).

Se permite el uso de las paquetes, librerías o **APIs** necesarias para el desarrollo de dichos elementos como operaciones simbólicas y analíticas, generación de gráficos, tablas, archivos, etc. Sin embargo, no se permite la inclusión de métodos numéricos predefinidos en librerías externas que sean directamente los mismos que hayan sido asignados a cada grupo.

El Notebook debe ir respectivamente documentado haciendo uso de celdas que contengan Markdown, y utilizar \LaTeX , para escribir expresiones matemáticas. El código debe ser claro y ordenado, además de estar comentado. Todo lo implementado debe ser coherente con la base teórica de la primera parte.

Recursos auxiliares:

- [Numerical Tours in Python](#): Colección de notebooks de Jupyter programados en Python, elaborados por el matemático [Gabriel Peyré](#). Cada notebook presenta un tema de matemática aplicada y resultados numéricos correspondientes.
- [Graficación en Python con librería Matplotlib](#)
- [Tabulación en Python con librería Pandas](#)
- [Arreglos numéricos en Python con NumPy](#)
- [Operaciones simbólicas en Python con SymPy](#)
- [Editor de \$\text{\LaTeX}\$ en línea](#)
- [Guías para el uso de \$\text{\LaTeX}\$](#)
- [\$\text{\LaTeX}\$ avanzado](#)
- [Gráficas en \$\text{\LaTeX}\$ usando TikZ](#)
- [Ejemplo de investigación matemática](#)
- [Ejemplo de contenido sobre método numérico](#)

3 Problemas Propuestos

1. **Números de Máquina:** Dados los parámetros de una máquina ficticia $F(\alpha, \beta, t, n, M)$ donde:

- α es la base de la mantisa
- β es la base del exponente
- t es el número de dígitos en la mantisa
- n es el menor exponente de su característica
- M es el mayor exponente de su característica

Escriba un programa que sea capaz de:

- (a) Convertir números en nuestro sistema decimal a los números de la máquina ficticia.
- (b) Convertir números de la máquina ficticia a números en nuestro sistema decimal.

2. **Calculadora de Raíces:** Escriba un programa que eventualmente converja de forma cuadrática y calcule la raíz cuadrada de cualquier número, utilizando únicamente operaciones de suma, multiplicación y división. Pruebe este programa con números del 10 al 10000.

- (a) Explore opciones para elegir un punto de inicio y utilice uno de los métodos vistos en clase para encontrar el resultado.
- (b) Proporcione un informe conciso y detallado de sus exploraciones haciendo uso de recursos visuales como estadísticas o gráficos de la información.
- (c) Generalice el método para calcular cualquier raíz enésima, donde $n \in \mathbb{Z}$ de un número real dado e implementar sus excepciones.

3. **Método de Newton Modificado:** Escriba un programa que modifique el método de Newton cuando esté involucrada una raíz múltiple.

- (a) Utilice los ejemplos de $(x - 4)^2 \sin(x)$ y $(x - 4)^3 \sin(x)$ como casos de prueba.
- (b) El algoritmo debería ser capaz de identificar la multiplicidad de la raíz ($p = 2$ o 3 en los ejemplos anteriores) y modificar su enfoque para obtener una rapidez de convergencia cuadrática.
- (c) Demuestre que converge cuadráticamente en ambos casos.
- (d) Expandir la implementación del método para que funcione en el conjunto de los números complejos.

4. **Método de la Secante:** Implemente el método de la secante para encontrar la raíz de una función no lineal dada.

- (a) Investigue sobre la demostración de su convergencia (como mínimo encuentre su orden y constante asintótica del error).
- (b) Implemente el método de la secante y utilícelo para encontrar la raíz de una función no lineal dada.
- (c) Compare el rendimiento del método de la secante con los métodos de Bisección, Iteración de Punto Fijo y Newton.

5. **Método de Müller:** Implementa el método de Müller para encontrar las raíces de una función polinómica compleja dada.

- (a) Investigue sobre la demostración de su convergencia.
- (b) Implemente el método de Müller para encontrar raíces complejas de una función en el plano complejo.
- (c) Utilice técnicas de visualización para representar gráficamente las raíces encontradas y verificar su precisión.
- (d) Discuta la eficacia y las limitaciones del método de Müller en la búsqueda de raíces de funciones polinómicas complejas en comparación con otros métodos: Bisección, Iteración de Punto Fijo y Newton.

6. **Polinomios Interpolantes:** Elabore un programa que sea capaz de dibujar una curva en el espacio 2D y 3D a partir de una serie de arreglos dados haciendo uso de polinomios interpolantes. El programa debe poder recibir un arreglo que represente n puntos en dos dimensiones de la forma $\{(x_0, y_0), (x_1, y_1), \dots, (x_{n-1}, y_{n-1})\}$ o en tres dimensiones de la forma $\{(x_0, y_0, z_0), (x_1, y_1, z_1), \dots, (x_{n-1}, y_{n-1}, z_{n-1})\}$.

- (a) Utilice un enfoque paramétrico, es decir, genere y muestre los polinomios para cada coordenada de la forma $x(t), y(t)$ para dos dimensiones y $x(t), y(t), z(t)$ para tres dimensiones.
- (b) Grafique las curvas utilizando los polinomios generados.

Ejemplo:

La curva en 2D está generada por los siguientes n puntos en dos dimensiones:

$$\{(1, 6), (-1, -6), (5, 3), (5, -5), (-3, -10)\}$$

La curva en 3D está generada por los siguientes n puntos en tres dimensiones:

$$\{(-4, 0, 1), (2, -2, -2), (-3, 4, 2), (4, 4, 0)\}$$

