



**Atributo: Análisis de problemas**

**Grupo #5**

Estudiantes:

Ceciliano Ortega Ángelo Fabian

Chinchilla Rodríguez Kevin Esteban

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computadores

CE-1111

Análisis Numérico para Ingeniería

Profesor:

Juan Pablo Soto Quirós

*a) ¿Cómo identificó el problema matemático o computacional principal en cada ejercicio de la asignación? Explique qué elementos del enunciado le permitieron reconocer qué método numérico debía aplicarse y por qué ese problema constituye un caso de análisis propio de la ingeniería.*

La identificación del problema matemático se realizó analizando palabras clave en el enunciado (como "interpolación", "aproximar raíces" o "resolver sistema") y vinculándolas con los temas del curso de Análisis Numérico. Por ejemplo, la necesidad de un polinomio que pase por puntos dados indica el uso de interpolación, mientras que aproximar una función sugiere métodos como Newton-Raphson. Estos problemas constituyen casos típicos de la ingeniería porque modelan situaciones reales, como el análisis de esfuerzos o la optimización de procesos, donde las soluciones analíticas exactas son inalcanzables, requiriendo en su lugar métodos numéricos robustos para obtener resultados confiables y cuantificables

*b) ¿Qué análisis realizó sobre las condiciones, variables y características del problema antes de implementar la solución? Describa cómo evaluó aspectos como el tipo de función o sistema, convergencia del método, eficiencia computacional, tolerancias o posibles limitaciones, y cómo ese análisis influyó en su estrategia de programación.*

Antes de implementar, se analizaron las condiciones específicas del problema: el intervalo de trabajo, el comportamiento de la función (continua, derivable) y los requisitos de precisión (tolerancia). Se evaluó la convergencia del método seleccionado para el caso particular y se consideró la eficiencia computacional, optando por algoritmos con menor complejidad para evitar carga innecesaria en cálculos iterativos. Este análisis previo influyó directamente en la estrategia de programación, decidiendo estructuras de datos, criterios de parada estrictos y la posible vectorización del código para mejorar el rendimiento y la precisión de los resultados obtenidos

*c) ¿Cuál fue el plan de solución que formuló para resolver cada ejercicio? Detalle los pasos que siguió: selección del método, justificación, diseño del algoritmo, implementación, criterios de parada, validación de resultados, uso de gráficas u otras herramientas de verificación, y cómo esto reflejó un enfoque sistemático para resolver el problema numérico.*

El plan de solución fue sistemático. Tras dividir los ejercicios en pareja para abarcar todos los temas cada uno, se procedió a:

1. Una lectura comprensiva y consulta de material para refrescar los métodos.
2. El diseño de un pseudocódigo basado en el algoritmo teórico.
3. La implementación en un lenguaje como Python o GNU Octave, estableciendo criterios de parada híbridos (por tolerancia y número máximo de iteraciones) para equilibrar precisión y recursos.

4. La validación de resultados mediante gráficas (usando Matplotlib o herramientas nativas) y la comparación con soluciones analíticas cuando fue posible. Este proceso metódico aseguró una resolución eficiente, organizada y verificable para cada problema numérico.

*d) ¿Cómo implementó la solución computacional del problema? Explique los métodos numéricos utilizados, las funciones que programó, las estructuras de control que empleó, y cómo verificó que el código ejecutara correctamente la solución esperada.*

La solución de cada ejercicio dependía del sistema a resolver. Por ejemplo, cuando era necesario interpolar, se empleaban métodos como las diferencias divididas de Newton y se determinaba la cota de error. En otros casos, se debía aproximar la integral de una función en un rango específico utilizando la regla compuesta de Simpson o cuadraturas Gaussianas.

Para la programación, se siguieron los pasos sugeridos de forma indirecta en el enunciado de cada problema. En general, el esquema aplicado fue el siguiente:

- Implementar las funciones importantes.
- Ejecutar ejemplos basados en la teoría para comprobar su correcto funcionamiento.
- Unir todas las funciones necesarias a utilizar.
- Establecer los atributos necesarios basándose en el problema a resolver, como la función, intervalos, errores, tolerancias, iteraciones máximas, entre otros.
- Llamar a la función correspondiente.
- Verificar que el resultado es el esperado.

El uso de estructuras de control también dependió del tipo de problema. Por ejemplo, los ciclos *for* resultaron útiles para definir iteraciones máximas o para llenar vectores y matrices, mientras que las estructuras *if/else* permitieron manejar casos excepcionales, como verificar si el error se encontraba dentro de la tolerancia especificada.

Finalmente, para confirmar que los resultados fueran correctos, se utilizaron herramientas externas ya establecidas, como calculadoras científicas, que permitieron comprobar valores exactos, especialmente en el caso de integrales.

*e) ¿Cómo evaluó la calidad, exactitud y eficiencia de la solución obtenida? Describa cómo analizó los errores, comparó resultados, verificó la convergencia, interpreto gráficos o tablas, o justificó la validez de la aproximación numérica.*

Para evaluar la calidad, exactitud y eficiencia de las soluciones obtenidas, se aplicaron distintos métodos de verificación y análisis basados en los conceptos estudiados en el curso. En primer lugar, se analizaron los errores teóricos asociados a cada método numérico. En los métodos de integración numérica, como Simpson compuesto o las

cuadraturas Gaussianas, se verificó el comportamiento, contrastando el valor aproximado con una solución de referencia.

Para validar los resultados, se compararon las aproximaciones obtenidas con valores calculados mediante herramientas externas confiables, como calculadoras científicas o software matemático. En algunos casos se recurrió a plataformas como ChatGPT para contrastar procedimientos o verificar que los resultados coincidieran con lo esperado teóricamente.

Finalmente, se evaluó la eficiencia computacional considerando la cantidad de operaciones realizadas, el tamaño de las particiones necesarias y el tiempo de ejecución estimado.

Métodos más precisos, pero computacionalmente costosos se compararon con alternativas más simples para justificar su uso según el problema planteado.