TEC | Tecnológico de Costa Rica

Instituto Tecnológico de Costa Rica

Análisis numérico para la ingeniería

Trabajo Extra-clase: Atributo de Análisis de Problemas

Profesor:

Juan Pablo Soto Quirós

Estudiantes:

Giancarlo Vega Marín, jungianca6@estudiantec.cr

Isac Marín Sirias, isacms@estudiantec.cr

Noviembre 2025

II Semestre, 2025

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.**

Para cada uno de los ejercicios planteados en el avance 4 del trabajo extra clase, como grupo primeramente se identificó el tipo de problemas qué se estaba planteando, no era complicado saber si cada uno se trataba de interpolar datos, aproximar una función, calcular una integral o bien estimar una cota de error. Se menciona sencillo ya qué los mismos enunciados siempre dan la referencia explícitamente el método a usar lo cual permitió reconocer de manera inmediata el método numérico que correspondía aplicar según el ejercicio.

Adicionalmente, cada una de estas preguntas representa una situación común en el análisis numérico aplicado a la ingeniería como la reconstrucción de funciones a partir de puntos, estimación de cotas de error para evaluar la confiabilidad de cálculos o bien la aproximación de intégrales cuando no se tiene a disposición una solución exacta. Este tipo de problemas por su naturaleza aproximada y dependiente siempre del análisis del error. Todos estos ejercicios entran dentro del proceso de análisis qué determinan al ingeniero al momento de resolver una situación dente no existe una respuesta exacta y esta tiene que aproximarse.

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 realizar una implementación programada, se realizó una revisión y recopilación de elementos principales en cada uno de los ejercicios. En el ejercicio de interpolación, primero se verificó la discretización del intervalo, el grado del polinomio y las características qué la función dada. Esto era necesario para poder evaluar si el método tuviese una buena estabilidad numérica.

En las cotas de error se analizó el comportamiento de la derivada de orden superior, lo que influye consecuentemente en la magnitud del error. Esto hizo que revisara si era necesario usar derivación simbólica o si era más conveniente la aproximación numérica

En el ejercicio de integración numérica, lo primero qué se revisó fue su convergencia y eficiencia para cada uno de los métodos. Simpson por ejemplo nos sugiere que el numero de los subintervalos sea par, la cuadrada Gaussiana, es mucho más precisa, por ejemplo, pero el uso iterativo requiere tener un mayor control con la tolerancia y nos invita a evitar el exceso de subdivisiones que a pesar de ser mas costoso computacionalmente, da una mejor aproximación.

En fin, antes de programar es importante verificar la continuidad, dominio de integración, estabilidad de los algoritmos y costos de ejecución de estos. De esta manera, el análisis previo orientó la estrategia de las implementaciones previniendo mas errores a la hora del proceder con el código.

c) ¿Cuál fue el plan de solución que formuló para resolver cada ejercicio? Detalle los pasos que siguió: selecciones 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 a seguido en todos los ejercicios mantuvo una estructura similar tal qué

- Identificar el método para la resolución del ejercicio según lo indicado en el enunciado
- Justificar el uso el método por eficiencia, exactitud o bien porque era el método estudiado en el curso
- Diseñar un algoritmo a partir a partir de la teoría vista en clase o en base de un pseudocódigo dado.
- Implementar un paso a paso asegurando respetar los detalles indicados en el enunciado.

- Establecer criterios de parada en los métodos iterativos.
- Validación de los resultados
- Interpretación de los resultados para confirmar si la solución numérica era razonable dentro del problema planteado.

Este proceso permitió abordar cada uno de los ejercicios de manera clara y consistente asegurando qué cada solución se respondiera según lo solicitado.

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.

Cada una de las implementaciones fue distinta en Python y Octave, pero el enfoque general fue el mismo, asegurando qué se pudiese transformar la teoría del método numérico en funciones programables, modulares y fáciles de probar.

En base a lo anterior se resolvieron los problemas con los siguientes métodos:

Diferencias dividas de newton: Este método construye un polinomio que interpola un conjunto de puntos (x_i, y_i), empleando coeficientes obtenidos por diferencias divididas. La ventaja es que permite construir el polinomio de manera incremental y evita recalcular toda la matriz como ocurre con otras formulaciones. Su implementación se llevó a cabo de la siguiente manera:

Se construyó una tabla triangular donde cada columna contiene diferencias divididas de orden superior. Luego se generó el polinomio en forma de Newton:

$$P(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots$$

Con Sympy se expande para obtener una forma simbólica clara.
Finalmente se verifica su comportamiento comparándolo con la función original mediante una gráfica.

- **Cota de error del polinomio de interpolación:** Para este problema se utilizó la forma vista en clase:

$$|\, \boldsymbol{R_n(x)} \,| \leq \frac{\mathbf{max}\ |\, \boldsymbol{f^{(n+1)}(\xi)} \,|}{\boldsymbol{(n+1)!}} \prod_{\boldsymbol{j=0}}^{\boldsymbol{n}} |\, \boldsymbol{x - x_j} \,|.$$

Su implementación se llevó a cabo de la siguiente manera:

Se encontró una derivada orden n+1 usando el cálculo simbólico
Se convirtió esa derivada a función numérica y se encontró un máximo absoluto en el intervalo [a, b] usando "fminbnd"
Se evaluó el producto en la sumatoria
Se remplazó todo en la fórmula de la cota de error.

Este método permite estimar si el polinomio es confiable sin la necesidad de conocer el error real.

- **Cota de error de la regla compuesta de Simpson**

La fórmula teórica es:

$$\left| \int_a^b f(x)\, dx - I_{sc}(n) \right| \leq \frac{(b-a)h^4}{2880} \alpha_{\max}, h = \frac{b-a}{n}.$$

Su implementación se llevó a cabo de la siguiente manera:

Se obtiene la cuarta derivada de f.
Se encuentra su máximo en $[a, b]$.
Se construye la ecuación de la cota igualada a la tolerancia deseada y la resuelvo para n.
Se selecciona la solución positiva y real.

El método garantiza la cantidad mínima de subintervalos necesarios para lograr la exactitud deseada.

- **Cuadraturas Gaussianas (simple, compuesta e iterativa)**

I. Cuadratura Gaussiana simple

Es un método que aproxima integrales usando nodos y pesos óptimos en $[-1,1]$. Permite alcanzar un orden de exactitud mayor que otras reglas con el mismo número de puntos.

Su implementación se llevó a cabo de la siguiente manera:

Se uso una tabla fija de nodos y pesos (provista en el enunciado).
Se realizó el cambio de variable estándar para transformar $[a, b]$a $[-1,1]$.
Se Suma $w_i\, f(x_i)$.

II. Cuadratura Gaussiana compuesta

Divide $[a, b]$en subintervalos y aplica la cuadratura gaussiana simple en cada uno. Mejora la exactitud cuando la función cambia mucho dentro del intervalo. Su implementación se llevó a cabo de la siguiente manera:

Se generan n subintervalos uniformes.
Se aplica el cambio de variable en cada subintervalo.
Se suman todas las aproximaciones locales para obtener el resultado final.

III. Cuadratura Gaussiana iterativa

Este método repite la versión compuesta, aumentando el número de subintervalos hasta que la diferencia entre dos aproximaciones sucesivas sea menor que la tolerancia.

Su implementación se llevó a cabo de la siguiente manera:

Se inicia con $n = 1$ subintervalo.

Se incrementa n de uno en uno.

Se calcula el error estimado $| S_n - S_{n-1} |$.

Detenemos el proceso cuando dicho error fue menor a 10^{-8}.

Con cada una de las funciones se utilizaron estructuras de control básicas como los ciclos "for", condicionales y definiciones de funciones internas. Al final se verificó el correcto funcionamiento comparando salidas, con graficas o bien realizando pruebas puntuales con valores conocidos. Esto permitió confirmar qué el código seguía fielmente cada uno de los métodos utilizados en clase. La implementación de estos métodos muestra cómo pasar la teoría numérica a código funcional claro y cada uno de estos algoritmos se adapta a la estructura del problema y todos estos permiten obtener soluciones aproximados con niveles de exactitud controlados y verificables.

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

En la validación final de la solución se consideraron diversos criterios que permitieron evaluar su calidad, exactitud y eficiencia. En primer lugar, se realizó una comparación visual, especialmente en los procesos de interpolación numérica, con el fin de verificar que el polinomio reprodujera adecuadamente el comportamiento de la función original. Asimismo, se llevó a cabo una evaluación del error, tanto a partir de las cotas teóricas como mediante la comparación entre aproximaciones sucesivas en los métodos iterativos. Este análisis se complementó con la verificación de la convergencia, observando si el método tendía a estabilizarse al incrementar el número de nodos o la cantidad de subintervalos.

Otro aspecto relevante fue la revisión de la estabilidad, comprobando que los resultados fueran coherentes dentro del intervalo de estudio y que no se presentaran oscilaciones o comportamientos anómalos. Finalmente, se evaluó la eficiencia de los métodos iterativos, verificando que alcanzaran la tolerancia establecida sin requerir un número excesivo de iteraciones.

Estas verificaciones garantizaron que las soluciones obtenidas fueran confiables, precisas y acordes con el comportamiento esperado en cada método numérico aplicado.