Departamento de Física
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE FI3104
Métodos Numéricos para la Ciencia
Álvaro Núñez, Otoño 2022

Tarea #2

Benjamín Mancilla

Introducción

Para esta tarea se introdujo el concepto de Integral numérica y sus métodos, además de explayando la función y precisión de cada una de estos. A continuación, el alumno presentara su metodología y lógica utilizadas para la realización de los problemas otorgados, además de como se usaron los conceptos vistos en clases para la resolución de estos.

Pregunta 1

Se nos pide calcular el periodo de un péndulo en función a su velocidad angular, para esto debemos simplemente integrar dϕ y dt por separado, lo que nos entrega una igualdad entre el periodo y una integral. Para no tener que calcular todo el movimiento, integramos desde 0 hasta ϕ0, lo cual seria un cuarto del periodo total, lo cual nos deja que el periodo es 4 veces la integral que va desde 0 hasta ϕ0. Una vez hecho esto, se puede notar que la integral queda de la siguiente manera, como se puede apreciar esta se indetermina para ϕ0.

$$2\sqrt{2}\int_0^{\varphi_0} {\sqrt{cos\varphi \; - cos\varphi_0}}^{-1} d\varphi$$

Para arreglar esto, usaremos la identidad trigonométrica $2sen^2\frac{\alpha}{2} = 1 - cos\alpha$, luego despejamos y factorizamos el seno del ángulo inicial.

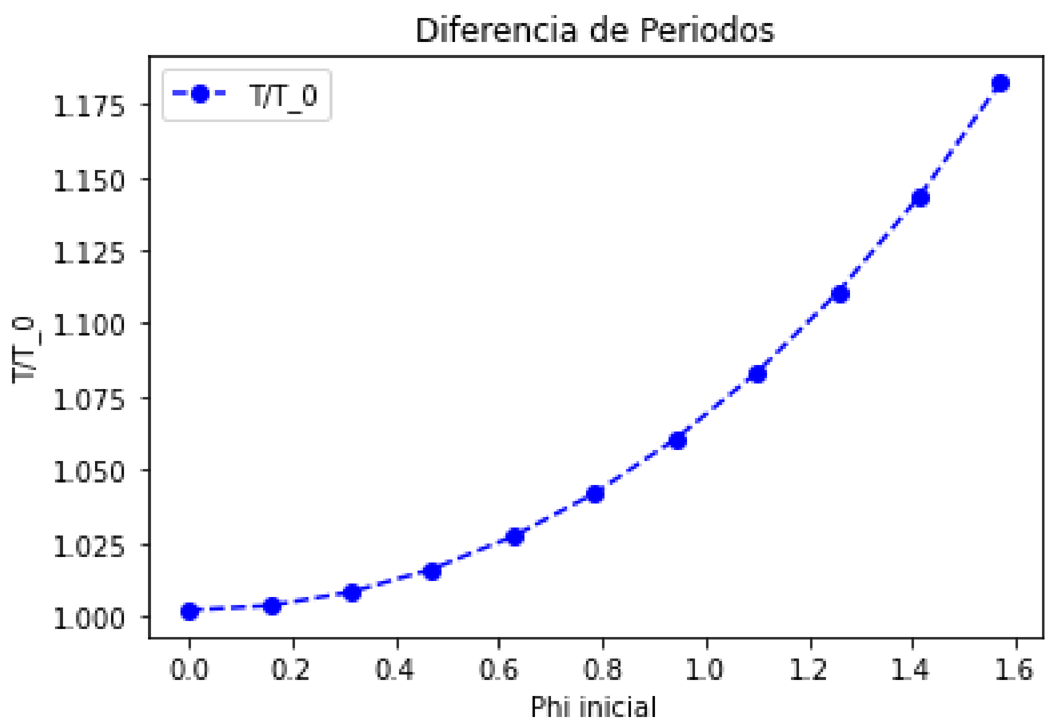
$$\sqrt{2}\int_0^{\varphi_0} (sen\frac{\varphi_0}{2}{\sqrt{1 \; - \frac{sen^2\frac{\varphi}{2}}{sen^2\frac{\varphi_0}{2}})}}^{-1} \; d\varphi$$

Luego haciendo el cambio de variable $sen\theta = \frac{sen\frac{\varphi}{2}}{sen\frac{\varphi_0}{2}}$, calculando su derivada para obtener que $d\varphi = \frac{2cos\theta sen\frac{\varphi_0}{2}d\theta}{cos\frac{\varphi}{2}}$, remplazando y despejando, obtenemos la siguiente integral.

$$4\int_0^{\frac{\pi}{2}} {\sqrt{1 - sen^2\theta sen^2\frac{\varphi_0}{2}}}^{-1} \; d\theta$$

Esta integral no se nos indefine para ningún punto, ya que la función seno al cuadrado tiene un condominio de [0,1], y nuestra constante, ósea el seno al cuadrado del ángulo inicial, vale menos de 1, ya que para que entregue este valor el ángulo tiene que ser π, el cual es un valor absurdo ya que el sistema no evoluciona y además de sale de los valores que serán calculados. Todo este desarrollo se baso en el trabajo sobre integrales elípticas y como resolverlas. Nótese que en ningún momento se menciono L y g en la integral, esto fue porque son irrelevantes para el trabajo de la integral.

Una vez que tenemos nuestra integral bien definida, solo debemos plasmarla en el código a través de una función. Ya con esto podemos trabajar en el ámbito numérico. Para esto ultimo se define una función llamada Periodo, que se encarga de integrar la función antes dicha utilizando el método de Simpson, el cual será explicado (junto a los otros dos) en la pregunta 2, y toma 4 valores, uno es el ángulo inicial, el largo, la constante de gravedad, y n que simboliza con que precisión se quiere el resultado (número de particiones). Con esto podemos repetir esta función con los 10 ángulos requeridos y luego simplemente dividir estos resultados por T_0. Finalmente, solo queda graficar y comentar.



Como se puede ver, mientras más crece el ángulo inicial mayor es la diferencia, esto tiene sentido ya que el periodo T_0 se basa en ángulos pequeños, por lo que, al crecer el ángulo inicial, las oscilaciones se alejan cada vez mas de esta aproximación.

Pregunta 2

Se nos pide estudiar 3 métodos de integración numérica, siendo estos Simpson, Trapecio y Punto medio en este caso.

Primero que todo hay que definir y plasmar en nuestro código estos 3 métodos distintos. En una primera instancia el cálculo de cada termino de estos métodos se hizo de un paso, es decir, uno por uno. Esto hacia que se demorara mas el programa y además producía errores de truncamiento y redondeo de Python, generando para ciertos n picks de error. Esto se soluciono implementado una forma distinta de pasar por lo términos, que es básicamente "ir de dos en dos", esto quiere decir que se hace el calculo inmediato de 2 términos, el par y el impar para el caso de Simpson. Además, en vez de juntar todos los resultados en una lista para luego usar la no confiable función *sum* de Python para sumarlos todos, se fue sumando inmediatamente los resultados guardando en memoria la suma total y actualizando por cada resultado calculado. Esto ultimo arreglo unos detalles de la función trapecio, además de la velocidad en general del cálculo y la precisión de los métodos.

Lo segundo es que hay que entender cada uno de los métodos empleados teóricamente.

La regla de Simpson lo que hace teóricamente es crear parábolas a través de la curva, estas se construyen con 3 puntos sucesivos, los cuales tiene una distancia delta entre ellos. Ya que no se considera el tamaño de estas parabolitas dentro de la sumatoria, ni tampoco de que cada una se construye con 3 puntos, luego de sumar cada resultado para cada partición del intervalo, se divide por 3 y se multiplica por el delta.

Por otro lado, el método del Trapecio lo que hace es unir los máximos y los mínimos de cada punto de la función a integrar, generando así estructuras trapezoides entre los intervalos, es decir, rectángulos con punta. Esto es mucho mas eficiente que considerar los típicos rectángulos que se forman en la característica sumatoria de Riemann. Esto en el código se ve como la suma de todos los resultados multiplicados por 2, excepto para el inicial y el final, esto ocurre porque estos lados no se repiten dentro de los trapecios. Como se considera dos veces cada lado se tiene que dividir por dos y por la misma razón anterior se multiplica por delta la sumatoria.

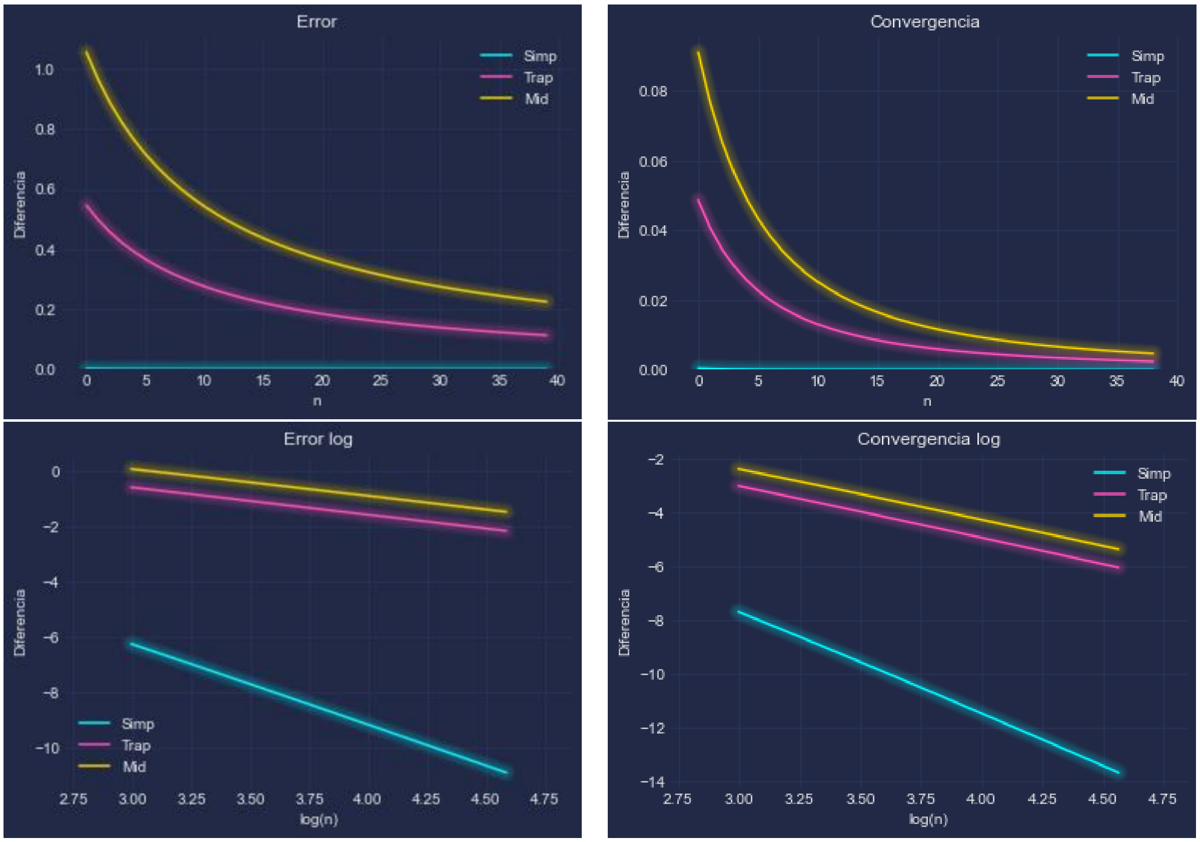
Por último, el método del punto medio lo que hace es unir la mitad de cada rectángulo, y calcular el área que queda debajo de esto. Luego como no se considera dos veces ningún dato, solo falta agregar el ancho de las particiones, es decir, delta.

Lo tercero es analizar y comentar el Error y la Convergencia de estos métodos, para ello se mostrarán 4 gráficos, 2 con los resultados y 2 con los mismos, pero en forma logarítmica.

Como se puede ver en los gráficos del error, para todos los métodos, mientras más espaciados se hagan, menor será el error. También se puede decir que el método menos preciso es Midpoint, mientras que el mas preciso es Simpson, este último es tan preciso que a una escala normal no se puede apreciar el error de este, se necesita el grafico a escala logarítmica para poder notarlo.

En la convergencia podemos observar el mismo comportamiento, Simpson comienza preciso y rápidamente refina sus resultados a medida que aumenta n, esto se puede ver en el grafico logarítmico de la convergencia, ya que la mayor pendiente la tiene Simpson. Los otros dos métodos convergen a la misma velocidad, pero el trapecio da un resultado mas preciso desde el principio, lo que lo hace superior a Midpoint.

En conclusión, el mejor método es Simpson, ya que no requiere muchas particiones para ser altamente preciso y si se quiere un mayor nivel exactitud, no hay que aumentar en una gran cantidad el numero de particiones.



Error
Convergencia
Error log
Convergencia log
Simp
Trap
Mid
Diferencia
n
log(n)

.

Pregunta 3

El primer paso para resolver esta pregunta es trabajar con una integral compleja. Mas que el tipo de número, es la cantidad de dimensiones con que se están trabajando, que son mas de 1. Para resolver este problema podemos elegir la forma polar del numero complejo y dejar R fijo, ya que en teoría este parámetro no afecta en el resultado siempre y cuando no agregue o quite el punto estudiado en cuestión, es decir 2.

Por comodidad, se elige un radio fijo igual a 1, y se hace un recorrido circular entorno a 0 y a 2, el primero seria la situación del punto z0, donde el resultado debería ser 0, y el segundo es el caso donde se incluye z0 y el resultado debería ser $2\pi i f(z0)$, es decir 343.05i aproximadamente.

Con esto definido, nos quedan dos integrales que dependen solo de theta y que van de 0 a 2π:

$$i \oint_0^{2\pi} \frac{exp\left(e^{i\theta} + i\theta\right)}{e^{i\theta} - 2} d\theta$$

$$i \oint_0^{2\pi} exp\left(e^{2i\theta} + 4e^{i\theta} + 4\right) d\theta$$

Estas integrables son fácilmente procesables con el método de Simpson (se escoge este por su precisión).

Al integrar numéricamente se obtienen los siguientes resultados.

```
In [74]: runfile('D:/Users/benja/Desktop/T2P3XD.py', wdir='D:/Users/benja/Desktop')
Reloaded modules: T2P2
(-5.3657187510571556e-11-8.539734193818645e-06j) es el valor que entrega fuera del circulo.
(2.399230381924503e-07+343.075750676387j) es el valor que entrega dentro del circulo.
```

Como se puede ver, fuera del circulo se obtiene un numero bastante pequeño, si se aumenta aun mas el n estos numeros se van haciendo cada vez mas despreciables. Para un n igual a 10 millones entrega resultados con un exponente -16, lo cual ya se acerca al error de la maquina. Este calculo no se muestra en la imagen ya que le toma bastante tiempo al computador.

Lo mismo ocurre para dentro del circulo, la parte real se acerca bastante a 0 y la parte imaginaria da 343.07, cuando el resultado real es 343.05 aproximadamente. El resultado es bastante preciso, sin embargo se uso un n igual a 100 mil, lo cual es una gran cantidad para un resultado no del todo exacto.

Conclusión, aunque el método de Simpson sea bastante bueno, para algunas integrales en especifico, en este caso para la funcion exponencial de un numero complejo, decae en su efectividad, por lo que siempre hay que comprobar el resultado analitico con el numérico.