Informe Tarea 1 - Métodos Numéricos

Bruno Quezada

27 de septiembre

Asignatura: Métodos Numéricos - FI 3104-1
Profesor Cátedra: Valentino González
Auxiliares: José Vines, Jou-Hui Ho

1 Pregunta 1

1.1 Introducción

En esta pregunta se solicita programar y comparar 2 algoritmos para el cálculo de derivadas, el algoritmo 1 consiste en determinar la derivada mediante la ecuación $f'(x) = \frac{f(x+h)-f(h)}{h}$ la cual posee un error de orden $O(h)$ y el segundo algoritmo determina la derivada a través de la ecuación $f'(x) = \frac{-f(x+2h)+8f(x+h)-8f(x-h)+f(x-2h)}{12h}$ el cual posee un error de orden $O(h^4)$.
Para derivar, se utiliza la función $f(x) = cos(x)$ con $X = 1.271$ (RUT = 19.133.271-7).
Para poder comparar la precisión del ajuste se utiliza el valor entregado de *math.sin(x)*.

También se realizarán los cálculos para **float32** y **float64** con el objetivo de comparar la diferencia en la derivada para cada uno de ellos.

1.2 Procedimiento

En primer lugar se construyen ambas derivadas como funciones generales en python y se elige un rango de h entre 0.1 y 10^{-21}, el rango contiene 1000 valores de h con un espaciamiento logaritmico, otorgado por la función *np.logspace()*.

Para estimar cuan bueno es el ajuste se define la función $error(h) = |-math.sin(x) - derivada(x,h)|$ La cual mide la distancia entre el valor otorgado por - *math.sin(x)* y el valor calculado para las derivadas
Una vez obtenidas ambas funciones se evalúan para la función $cos(x = 1.271)$ utilizando en primera instacia objetos de tipo **float32** y en segunda instancia de tipo **float64**

1.3 Resultados

En la figura 1.3 se presenta el resultado de la función error utilizando **float32** y en la figura 2 el caso **float64**. Para float32 mínimo error encontrado para ambos algoritmos es del orden de 10^{-8}. Para el caso de float64 el mínimo error encontrado para la derivada con error de orden $O(h)$ es del orden de 10^{-10} y para error de orden $O(h^4)$ es del orden de 10^{-15}

1.4 Análisis y Conclusiones

Para todos los casos, se observa que las derivadas son una función que alcanza una presición máxima en algún rango de h y a medida que ha se aleja de su maxima presición, el error aumenta en forma exponencial. El hecho de que la derivada empeora al utilizar un h grande no es sopresa, debido a que mientras mas grande h, más se aleja de 0 y por lo tanto representa peor el límite de la derivada por definición. En cambio al tener h muy pequeño, se observa que el error es desporporcionado y e incluso sale del recorrido de estas funciones trigonométricas alcanzando incluso valores del orden de 100. El aumento del error a medida que se utiliza un h más pequeño se debe a la presición finita que tiene el computador para representar los números reales, por lo que un h muy pequeño queda muy mal representado y en consecuencia la derivada tiene un error alto.
Otra característica asociada a la presición finita del computador son los saltos que se pueden apreciar en la zona izquierda de la función error, en donde la presición varía ampliamente un una localidad pequeña.

Para el caso de float32, como se mencionó anteriormente la diferencia entre ambos algoritmos tienen errores pequeños y del mismo orden en torno a su presición máxima $\approx 10^{-7}$ pero para el caso del algritmo de error de orden $O(h^4)$ la presición máxima se alcanza en un rango más amplio de valores para h.

Par el caso de float 64, se aprencian mayores diferencias entre los algoritmos, donde notoriamente el de orden $O(h^4)$ alcanza un error mínimo menor, en el rango de 10^{-4}. El algoritmo de error de orden $O(h)$ tiene un comportamiento muy dimilar al caso de float32.

Como concusión final se tiene que un objeto de mayor presición numérica como float64 es más adecuado para calcular operaciones matemáticas que busquen valores muy precisos, a pesar de su mayor uso de memoria. Además, es muy importante considerar el rango de mejor presición de estos objetos para nuestros cálculos, ya que se puede caer en el error de utilizar números demasiado pequeños que no serán bien representados por la máquina y generarán un error al tomar valores demasiado grandes que estén alejados de los valores matemáticos supuestos para los cálculos.

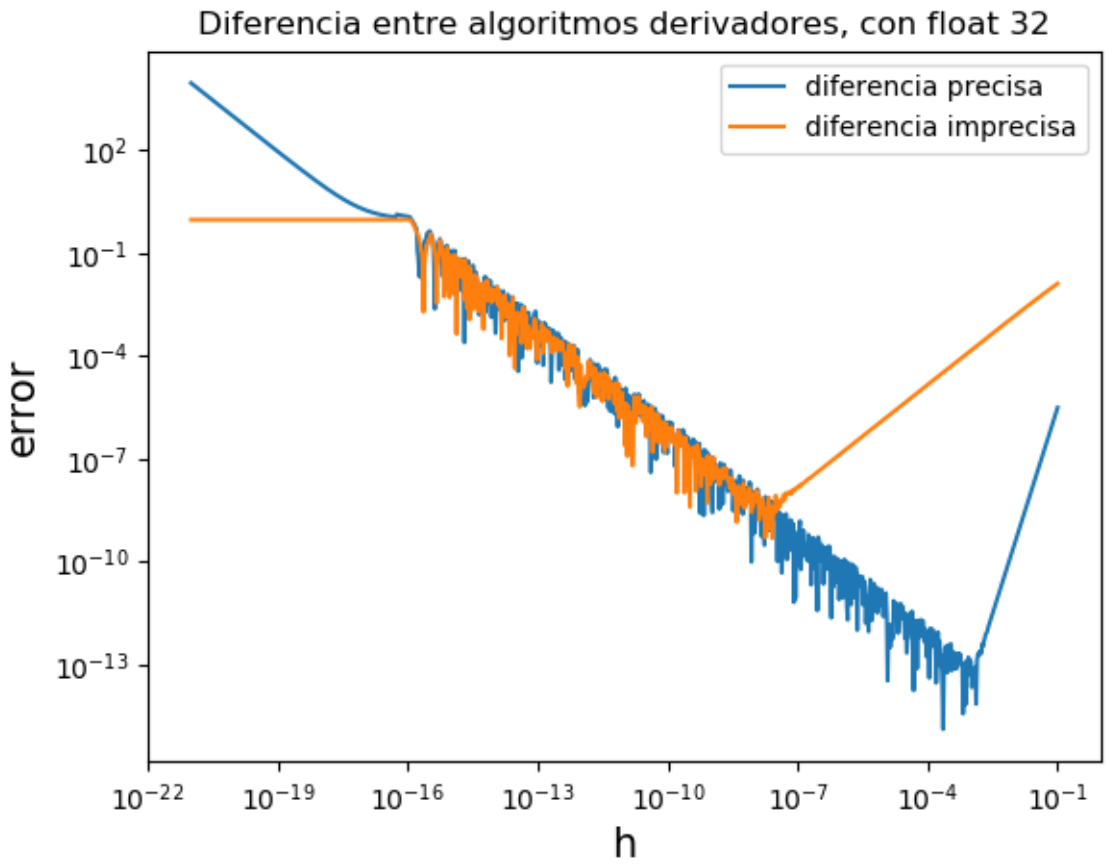

Figure 1: Distancia entre la función $-seno(x)$ y la derivada de $cos(x)$ calculada a través de los métodos numéricos, utilizando float32

En esta tarea se utilizó la resolución de la derivada de la función $cos(x)$ en vez de la $-cos(x)$ por confusión del autor, se conserva el resultado propio considerando que sólo cambia el signo de la derivada y el análisis en esencia es el mismo.

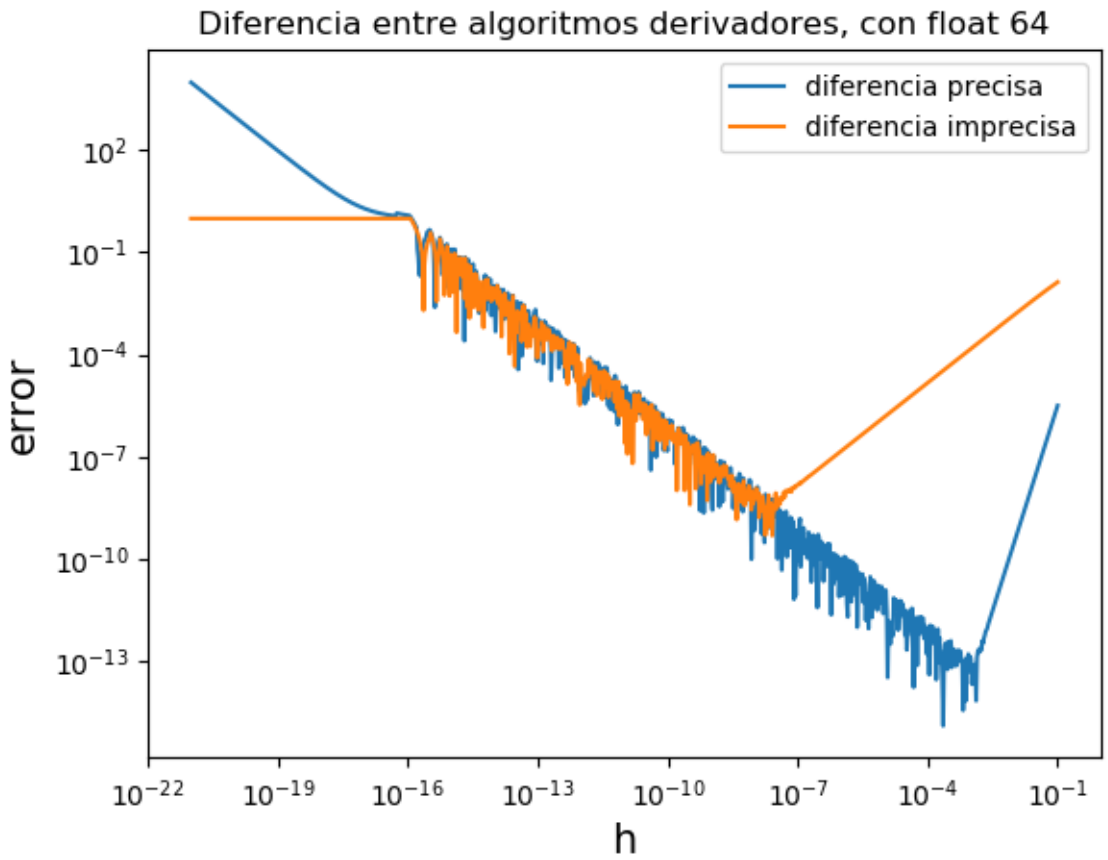

Figure 2: Distancia entre la función $-seno(x)$ y la derivada de $cos(x)$ calculada a través de los métodos numéricos, utilizando float64

2 Pregunta 2

2.1 Introducción

En este problema se busca estimar la temperatura de la radiacion remanente del Big Bang. Según la teoría, la radiación debiese corresponder a la de un cuerpo negro, la cual se modela a través de la función de Planck: $B_\nu(T) = \frac{2h\nu v^3}{c^2(e^{\frac{h\nu}{k_B T}}-1)}$.
Para ello se utilizarán los datos FIRAS del satélite COBE, los cuales contienen el espectro de radiación de fondo de microondas en frecuencia.

2.2 Procedimiento

2.2.1 Gráfico Espectro

Para graficar el es necesario importar los datos en archivo texto de FIRAS, esto se realiza utilizando el comando *np.loadtxt()*.
Una vez cargados los datos se ajustan las unidades a sistema internacional, se amplifica el error y se gafica utilizando *plt.subplots()*.

2.2.2 Integración Función de Planck

Para integrar la función de Planck se genera la función *plankAux()*, utilizando el cambio de variables $y = arctan(x)$, quedando definida como: $plankAux(x) = \frac{tan(x)^3(1+tan(x)^2)}{e^{tan(x)}-1}$.
Posteriormente se define la función *integración(función,a,b,dx)*, la cual genera una suma de Riemann de la

función entregada entre los puntos a y b con un $\Delta x = dx$.
Ésta integal se evalúa refinando Δx hasta que la diferencia entre el valor calculado y el valor esperado se de $\epsilon = 10^{-5}$

2.2.3 Determminación de Temperatura

Para determinar la temperatura se utiliza el área bajo la curva del espectro FIRAS, α, calculada con el método de Simpson y la integral de Planck anteriormente calculada.
La temperatura queda determinada entonces con la expresión $T^4 = \frac{\alpha c^2 h^3 15}{2 k_B^4 \pi^4}$

2.2.4 Comparacion con otros métodos de integración

Se comparan los resultados obtenidos de las integrales con los resultados que ofrecen librerías de python. Para la integral bajo el espectro de radiación, calculado con el método de Simpson se compara con *numpy.trapz()* y para la integral de la función de Planck se utiliza *scipy.integrate.quad()*, en donde hay que tener cuidado en definir bien los bordes de integración para que no se indefina el resultado.

2.3 Resultados

En la figura 3 se presenta el espectro de radiación de microondas en función de la frecuencia, con sus respectivas barras de error amplificadas.
En la figura 4 se presentan las funciones de Planck para la temperatura calculada y para la real sobre la radiación de fondo de FIRAS en función de la frecuencia. En la figura 5 se presenta el mismo gráfico con un acercamiento para poder apreciar la diferencia entre las curvas.
El área bajo el espectro es de $2.46 * 10^{-7}$ y el área bajo la integral de Planck de $8.188 * 10^{-8}$ lo que estima una temperatura de $2.691[K]$ al resolver la ecuación, para el espectro de radiación. Este resultado es cercano a los $2.75[K]$ reportada por COBE.
En la tabla 1 se muestra la comparación de las integrales realizadas con las dadas por los módulos de Python.

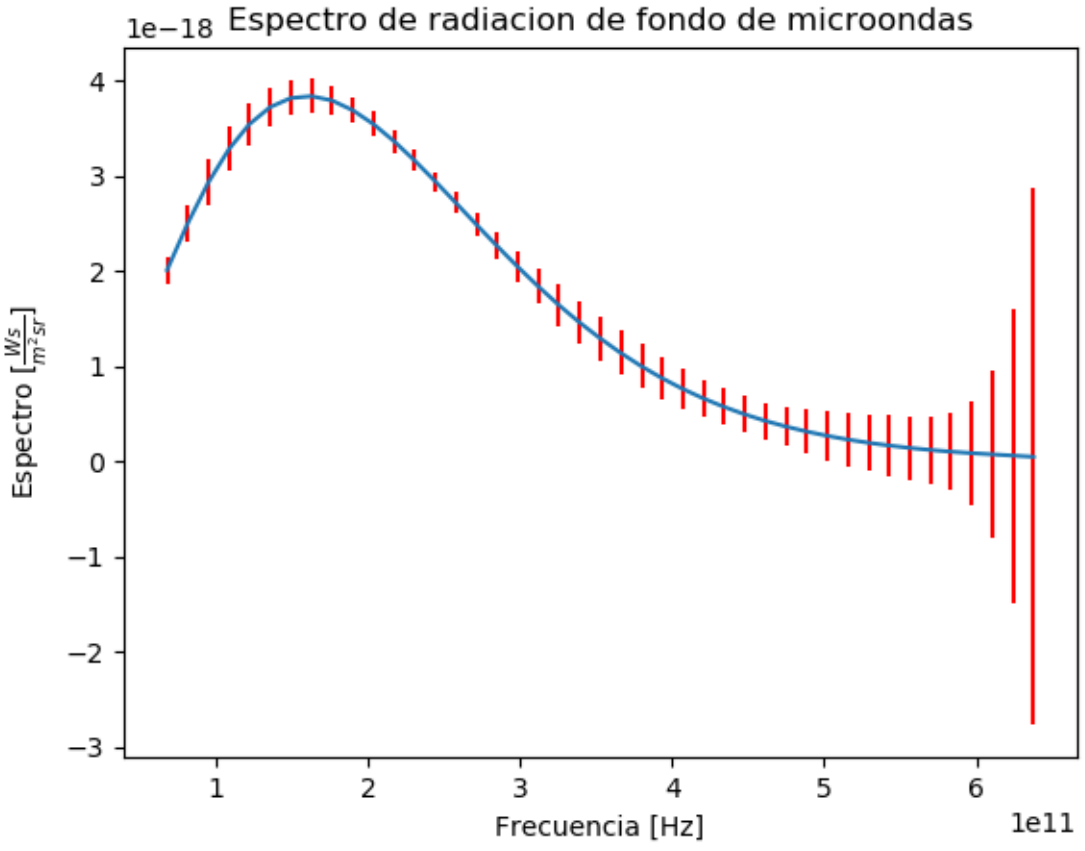

Figure 3: Radiación de fondo de Microondas medida por FIRAS (azul), el error está amplificado (rojo)

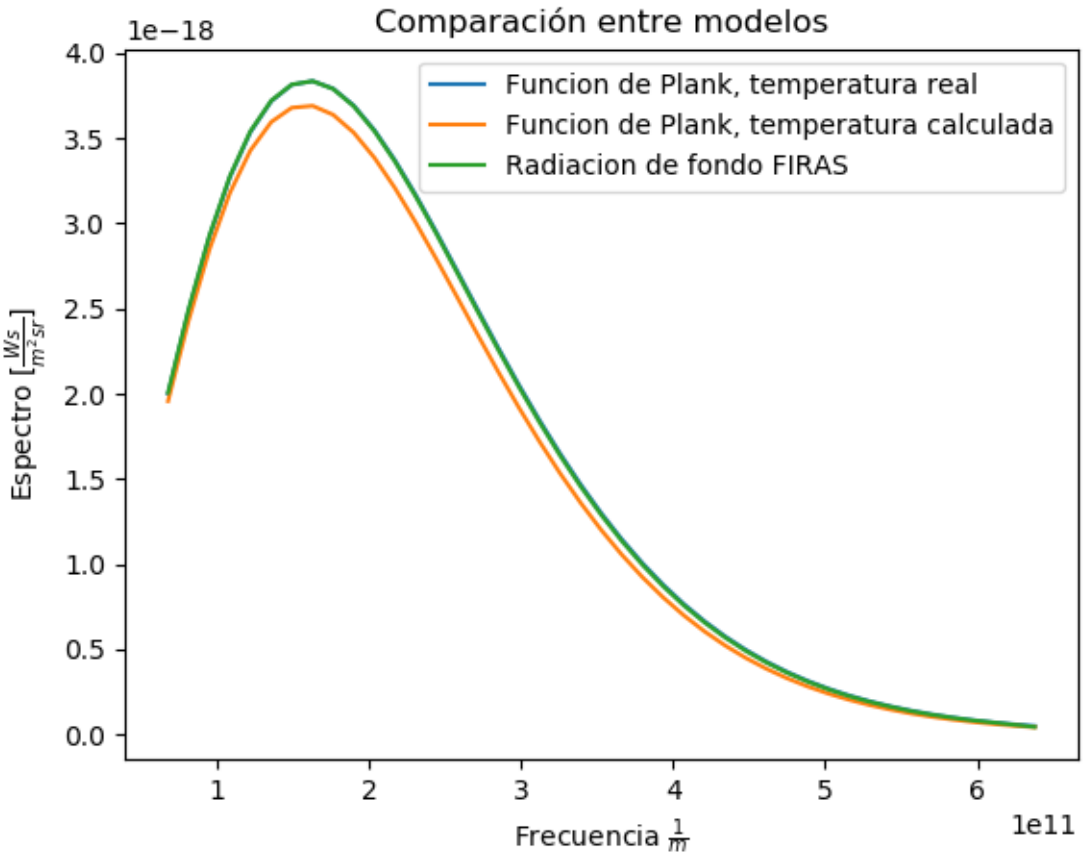

Figure 4: Gráficos del espectro de radiación de fondo (verde) y la función de Planck para la temperatura real (azul) y la calculada(amarillo)

Integrales Calculadas y Estimadas por paquetes de Python	
Error Integral de Planck con Suma de Riemann	$8.188 * 10^{-8}$
Error Integral de Planck con integrate.quad()	$3.321 * 10^{-7}$
Área bajo espectro con método de Simpson	$9.462 * 10^{-7}$
Área bajo espectro con numy.trapz()	$9.338 * 10^{-7}$

2.4 Análisis y Conclusiones

Se puede observara que en el espectro de radiación (figura 3) que la incerteza aumenta con la frecuencia. La función tiene un máximo global en torno a los $2[Hz]$.

En las figuras 4 y 5 se aprecia que la diferencia entre las curvas calculadas con la temperatura de COBE y la obtenida a través de la integración de la función de Planck, si bien ambas figuras son muy similares, las diferencias pueden ser explicadas por (1) la aproximación de cuerpo negro de Stephan-Boltzmann, en donde puede que no se comporte perfectamente como un cuerpo negro y (2) impresiciones de la integración del área bajo el espectro.
Con respecto a la integración a través de la sumatoria de Riemann, si bien este es un método bastante rudimentario, es bastante sencillo de implementar, considerando que se conoce a *priori* el resultado esperado basta con agregar una tolerancia pequeña para conveger cuanto uno quiera al resultado, por lo que se considera una manera apropiada de afrontar el problema. Para otras instancias en que no se conozca el resultado sería mejor estimar la integral a través de otros métodos más precisos.

En la tabla 1 se aprecia que ambas integraciones realizadas tienen resultados satisfactorios. En el caso de la integración de la función de Planck el error es varios ordenes de magnitud mas pequeño que el resultado esperado e incluso es más preciso que el resultado por *integrate.quad()*. En el caso de la integración del área bajo el espectro de radiación, el método de Simpson converge a un valor muy parecido al de *numpy.trapz()*
Se observa que los datos entregados por FIRAS son de una presición impresionante, en este contexto es impor-

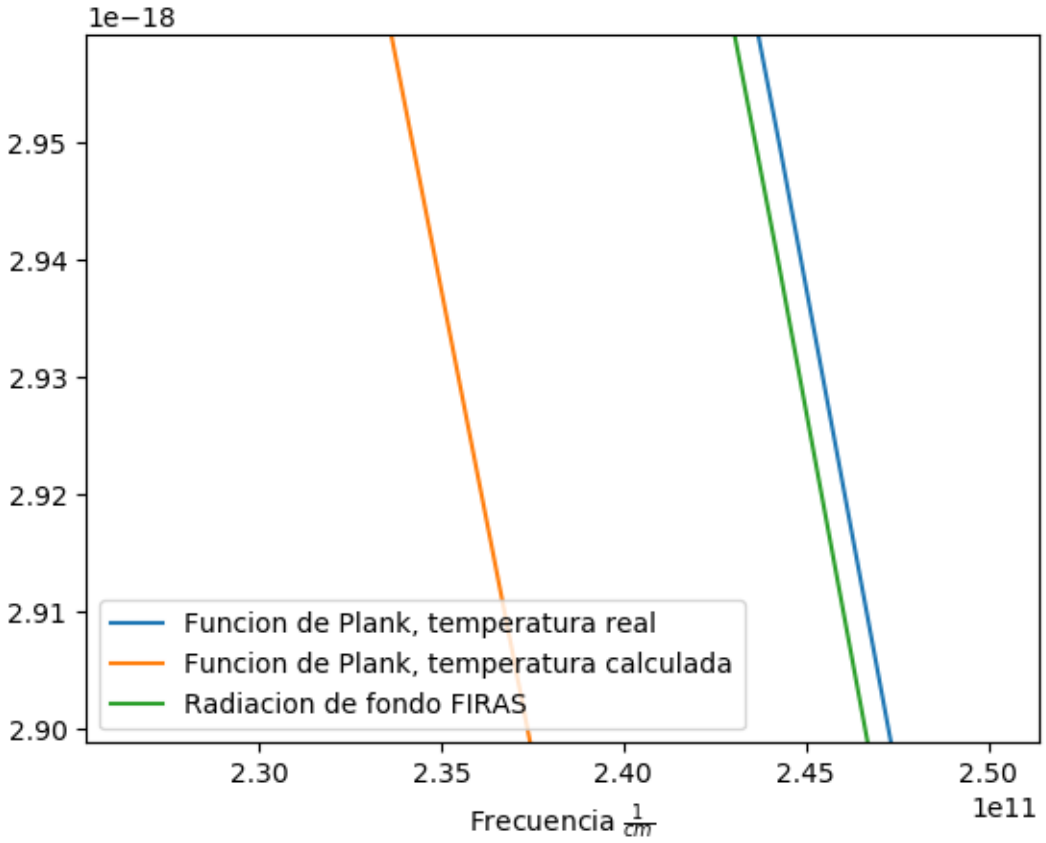

Figure 5: Gráficos del espectro de radiación de fondo (verde) y la función de Planck para la temperatura real (azul) y la calculada(amarillo) con acercamiento

tante utilizar métodos precisos de cálculo, para no perder cifras significativas al operar con estos valores.

Como conclusion final se considera que es importante utilizar criterio para establecer una presición adecuada al problema que se está resolviendo, considerando las limitaciones teóricas matemáticas y las limitaciones computacionales para poder estimar los valores los mejores valores de las variables a utilizar para encontrar una solución óptima.