

Pía Contreras Guerrero

19.840.187-0

Número lista=23

PREGUNTA 1

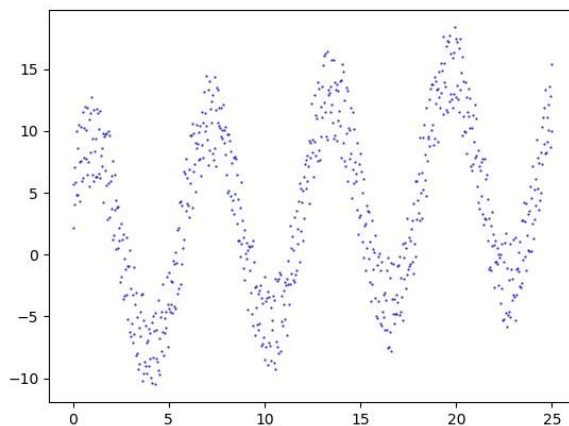
Los datos que se obtuvieron al digitar “19840187” en la sección random.seed(), y “23” en nLista fueron los siguientes:

[8.97568899 0.94854357 0.60212528 0.29828904]

donde el vector es [a omega phi b]

Es decir, es ese vector, al cual se quiere llegar. Ese vector fue llamado xobjetivo.

Además, el gráfico que se obtuvo con los datos fue el siguiente:



PARTE A

i) Explicación del código

Como se puede observar el gráfico de los datos obtenidos es una senoide que proviene del sen().

Lo primero que se hace en el código es cambiar la semilla, es decir, poner mi correspondiente rut y además cambiar mi numero de lista.

Se tiene que el método de Newton funciona como sigue:

$$x_{k+1} = x_k - (J(x_k)^T J(x_k) + S(x_k))^{-1} J(x_k)^T r(x_k)$$

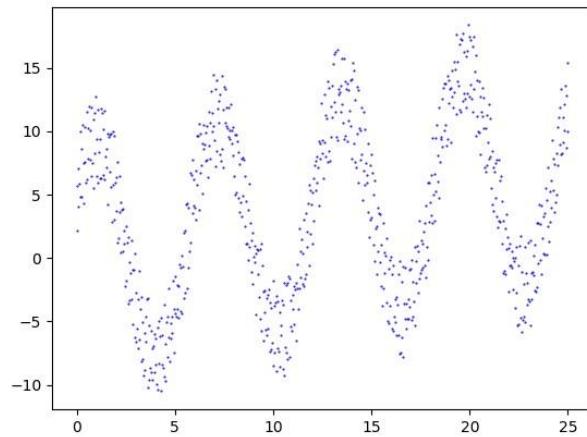
Para llegar a un ajuste mediante el método de newton se procedió de la manera que sigue:

- Se creó la función M_i , la cual es el modelo que permite ajustar los datos.
- Se creó el jacobiano, matriz de 750×4 , la cual se fue llenando columna por columna con la respectiva derivada de r respecto de su variable y recorre las 750 filas con un ciclo for.
- Se creó la función R , que es básicamente " $ydata - M_i$ ", en donde si se encuentra un M_i tal que minimice el error cuadrático de norma de $r(x)$ al cuadrado, entonces se cumple el objetivo.
- Se creó la Matriz S , la cual tiene dimensiones de 4×4 , al igual que el jacobiano, se fue llenando de manera manual fila por fila y columna por columna con un ciclo for.
- Se definió el vector x con las variables $a_i, \omega_i, \phi_i, b_i$. Para poder definirlo de manera correcta se les dio valores iniciales.
- Se definió el vector x_0 , vector de valores iniciales. Se probó con distintos valores iniciales, pero se vio que no todos convergían, los que dieron la convergencia más cercana fueron los valores $x_0 = [8.98 \ 0.95 \ 0.60 \ 0.30]$
- Se definió un error pequeño de 0.01, el número de iteración inicial y el número de máxima interacciones.
- Para implementar el método de newton se entró a un ciclo while, el cual tiene por condiciones que $iteración < maxiteración$. Dentro del ciclo se evaluó la función r , el jacobiano y S , además del error. Se implementó el método de newton, según la fórmula que está más arriba y se le aplicó además otra condición, si el valor max del valor absoluto de la resta entre el x iterado y su anterior es menor que el error, entonces se sigue iterando.
- Se graficaron los datos.
- Luego se implementó a la función r la función predeterminada de Python leastsq de la biblioteca Scipy, la cual minimiza la suma de cuadrados de un conjunto de ecuaciones, para ello sólo se utilizó la función R , se eligió el mismo punto mínimo anterior, se le aplicó la función leastsq a (R, x_0) , con los datos entregados por este vector, se creó otro vector, con el cual se trabajó con su primera componente para crear la nueva variable a graficar.

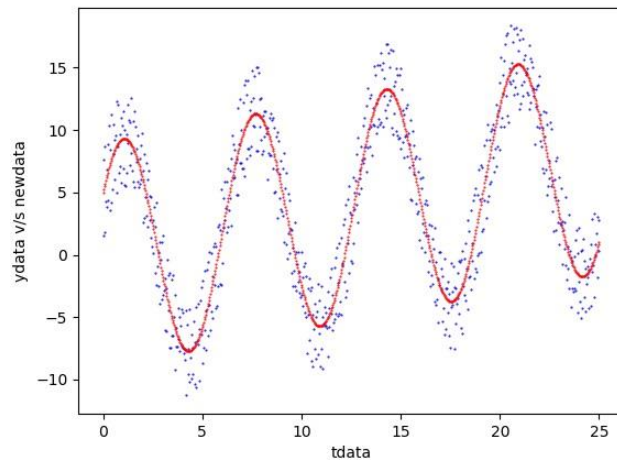
ii) Gráficos obtenidos dependiendo de cada método:

Los gráficos obtenidos fueron:

a) Gráfico inicial de datos:



b) Gráfico método de Newton:



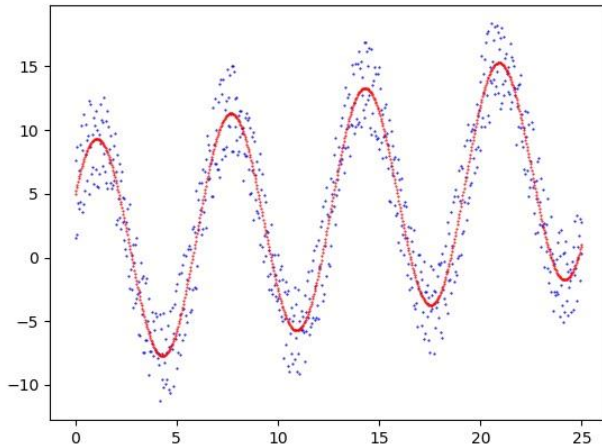
El valor de x que se obtuvo versus el valor del x objetivo difiere por centésimas, esto se puede deber a que el valor inicial de programa no fue elegido de manera detallada, sino más bien se fueron probando uno a uno.

Los valores obtenidos de x fueron los siguientes:

```
[8.97568899 0.94854357 0.60212528 0.29828904]
[9.00945962 0.94965884 0.58996931 0.30023928]
```

Donde el primero es el x objetivo, y el segundo es el x al que se llegó luego de la aplicación del método de Newton.

c) Gráfico con función leastsq:



Con esta función predeterminada se obtuvo los valores para x objetivo (primer vector) versus x_{optimo} (segundo vector) como sigue:

```
[8.97568899 0.94854357 0.60212528 0.29828904]
[9.00945961 0.94965883 0.58996934 0.30023928]
```

Los valores del x al que se debe llegar con respecto al x que se llegó difieren en muy pocas centésimas.

iii) Comentarios, comparación y análisis

➤ Método de Newton versus leastsq

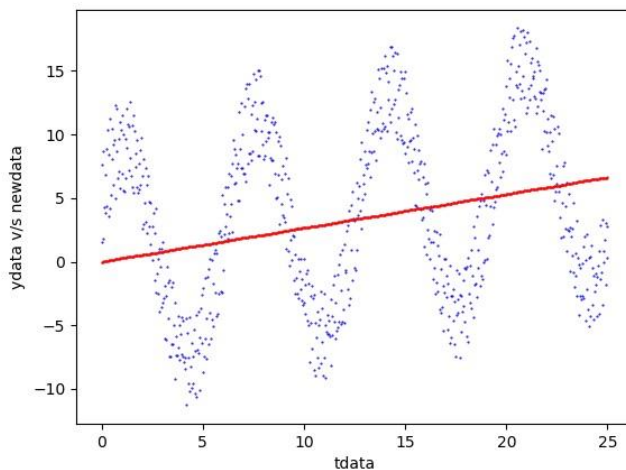
Primero, en el método de Newton, se modificó el número de iteraciones, y sucede que desde la iteración número 3 en adelante se mantiene el valor de x “óptimo” encontrado, en cambio, en la iteración número 1, 2 y 3, los valores “óptimo” cambian, y a medida que se aumenta la iteración (solo de 1 a 3), se acerca el valor del x óptimo al valor de x objetivo. Es decir, el método converge y entre las tres primeras iteraciones se va acercando más al valor óptimo, luego desde la iteración 3 en adelante no cambia el valor de óptimo, lo mantiene cercano, pero no se aleja.

La diferencia entre las componentes objetivas y óptimas de cada método son las siguientes:

	Método de Newton	Función leastsq
Diferencia a objetivo versus a óptimo	0.03385	0.033771
Diferencia omega objetivo versus omega óptimo	0.001115	0.001115
Diferencia phi objetivo versus phi óptimo	0.01222	0.012156
Diferencia b objetivo versus b óptimo	0.00192	0.001950

En este caso, el error entre la diferencia de la componente objetivo versus la componente óptima es muy poca, pero si es por comparar, se puede observar que la función `leastsq` tiene un error milimétricamente mayor en la componente a y b , mientras que el método de Newton tiene un error milimétricamente mayor sólo en la componente ϕ .

Si se cambian los valores iniciales de x en el método de Newton, el método se demora más en cuanto a tiempo en procesar la información y además no converge el método, por ejemplo, si se aplica el valor $x_0 = [7 \ 2 \ 3 \ 0.5]$ se obtiene el siguiente gráfico:

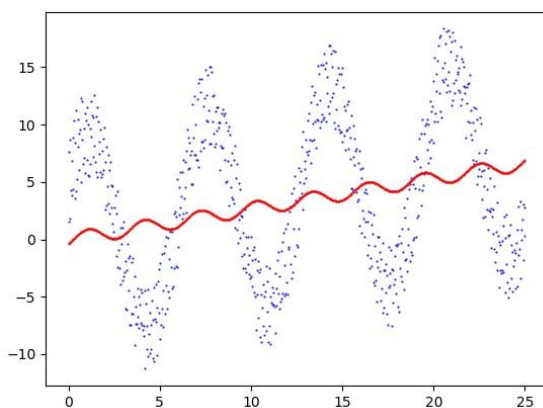


Y los siguientes valores para el x óptimo (segundo vector):

```
[8.97568899 0.94854357 0.60212528 0.29828904]
[-0.0243902 2.21801947 2.0820951 0.26554688]
```

Se puede ver que la diferencia entre el x óptimo y el x objetivo es muy grande.

Con la función `leastsq` si se se da valor inicial $x_0 = [7 \ 2 \ 3 \ 0.5]$, se obtiene el siguiente gráfico:



Y los valores de x objetivo (primer vector) versus x óptimo (segundo vector) están dados por:

```
[8.97568899 0.94854357 0.60212528 0.29828904]
[-0.62002476 2.0518043 2.50660334 0.26695301]
```

En este caso, se ve que la diferencia entre el xoptimo y el xobjetivo también existe.

Comparando la diferencia de cada uno de los métodos, con $x_0=[7 \ 2 \ 3 \ 0.5]$, es decir un valor inicial que no permitiría que ni el método ni la función convergiera, se obtiene:

	Método Newton	Funcion leastsq
Diferencia a objetivo y a optimo	9.000079	9.595714
Diferencia omega objetivo y omega optimo	1.269476	1.103261
Diferencia phi objetivo y phi optimo	1.479970	1.904478
Diferencia b optimo y b optimo	0.032742	0.031336

En este caso, en las componentes a y phi el error es mayor para la función leastsq , por otra parte, en omega y b el error es mayor para el método de newton, sin embargo en cuanto al porcentaje de error, es mayor el de la función leastsq.

PARTE B

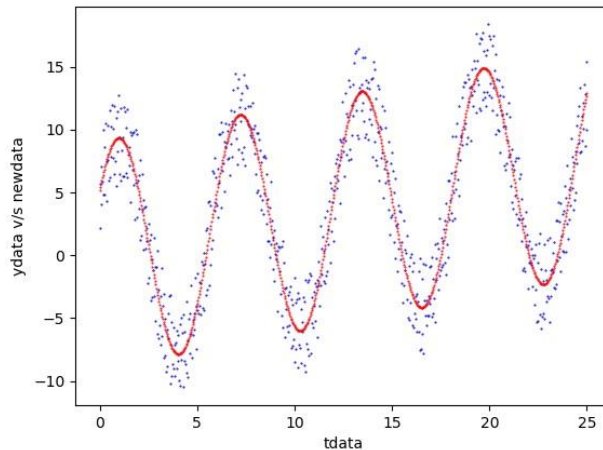
i) Explicación del código

- Se definió la función M_i, el jacobiano y la función R de igual manera que en la parte a), se definió el punto inicial como $[8.98 \ 0.95 \ 0.60 \ 0.30]$, ya que fue el punto el que al probarlo más convergió a lo que se quería llegar.
- En un ciclo while se implementó el método de newton que sigue:

$$x_{k+1} = x_k - (J(x_k)^T J(x_k))^{-1} J(x_k)^T r(x_k)$$

Para ello, a diferencia de la parte anterior se fue aplicando la multiplicación de matrices por separado y también la transpuesta de matrices.

El gráfico obtenido es el siguiente:



Y los valores obtenidos son:

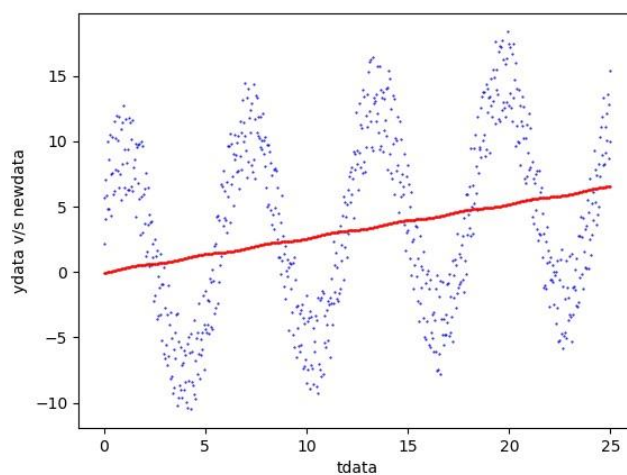
```
[9.01457236 1.00871151 0.58464381 0.30021296]
[9.06294608 1.00701791 0.60939548 0.29662256]
```

Donde el primer vector es el valor objetivo de x , y el segundo vector es el valor que se obtuvo de x luego de la aplicación del método.

i) Comentario, comparación y análisis

Si se modifica el número de iteraciones se pudo observar que desde la iteración uno a la iteración 4, cambia el valor del x óptimo, acercándose al x objetivo, luego de la iteración número 4 el método sigue convergiendo, pero no cambia el valor del x óptimo de la iteración número 4.

Al cambiar el punto inicial por $x_0=[7 \ 2 \ 3 \ 0.5]$, el método no converge en nada, de hecho la gráfica queda como sigue:



Y los valores de x , tanto el x objetivo como el x “óptimo” luego de aplicarle el método con $x_0=[7 \ 2 \ 3 \ 0.5]$ es:

```
[9.01457236 1.00871151 0.58464381 0.30021296]
[-0.06716   -1.92554937 51.70618477 0.26078102]
```

La diferencia entre ellos es bastante, y el gráfico a su vez, no se acerca en nada a la curva inicial.

➤ Método parte b) versus Método de Newton

La comparación respecto de la parte a) de la implementación del método de Newton y la parte b) se puede ver que el método de Newton de la parte a) es más exacto respecto de las mediciones del valor de x objetivo y del valor óptimo, esto se puede ver por dos motivos, el número de iteraciones y la diferencia de cada componente de su valor objetivo versus su valor óptimo, ambos se detallarán en la siguiente tabla:

	Método Newton (parte a)	Método parte b)
Número de iteraciones para alcanzar el óptimo	3	4
Diferencia entre a objetivo y a óptimo	0.03385	0.04837
Diferencia entre ω objetivo y ω óptimo	0.001115	0.001701
Diferencia entre ϕ objetivo y ϕ óptimo	0.01222	0.024752
Diferencia entre b objetivo y b óptimo	0.00192	0.0036

Además, con respecto cuando se cambia el punto inicial por $x_0=[7 \ 2 \ 3 \ 0.5]$, se puede ver que la diferencia entre el método a) y b) es:

	Método Newton (parte a)	Método parte b)
Diferencia entre a objetivo y a óptimo	8.999990	9.081730
Diferencia entre ω objetivo y ω óptimo	1.26946	2.93426
Diferencia entre ϕ objetivo y ϕ óptimo	1.47988	51.121457
Diferencia entre b objetivo y b óptimo	0.03328	0.039510

Se puede ver, en toda coordenada, el método de Newton es más efectivo.

➤ Método parte b) versus función leastsq

En primer lugar, se verá la diferencia que existe entre el x objetivo y el x “óptimo” de cada método utilizado, para así poder visualizar qué método tendría más exactitud.

	Método parte b)	Función leastsq
Diferencia a objetivo versus a optimo	0.04837	0.033771
Diferencia omega objetivo versus omega optimo	0.001701	0.001115
Diferencia phi objetivo versus phi optimo	0.024752	0.012156
Diferencia b objetivo versus b optimo	0.0036	0.001950

En este caso, se puede apreciar que en toda componente el error es mayor para el método de la parte b), ya que existe mayor diferencia entre lo esperado y lo que se llegó.

Luego, en ambos métodos se cambió el punto inicial por $x_0=[7 \ 2 \ 3 \ 0.5]$, así se vio que la diferencia entre cada componente es:

	Método parte b)	Función leastsq
Diferencia a objetivo versus a optimo	9.081730	9.595714
Diferencia omega objetivo versus omega optimo	2.93426	1.103261
Diferencia phi objetivo versus phi optimo	51.121457	1.904478
Diferencia b optimo versus b objetivo	0.039510	0.031336

En este caso, se puede observar que en la componente para “a” el error es mayor para la función leastsq, sin embargo, en todas las demás componentes, el error es mayor para el método de la parte b), incluso, teniendo un error muy grande en la componente phi.

A modo de conclusión se puede observar, que la función leastsq es más eficiente que el método de la parte b).

CONCLUSIÓN FINAL

A modo de conclusión, y con respecto a todos los datos observados, analizados y comparados, se concluye que los tres métodos son eficientes si se trata de ajustar datos para una convergencia, es decir todos lo hacen, con ciertas sensibilidades, como lo puede ser el punto inicial escogido, pero al fin y al cabo se logra el objetivo, sin embargo hay algunos métodos más eficientes que otros, en este caso, con los datos obtenidos para mi modelo, se puede concluir que el método más eficiente es el Método de Newton de la parte a), luego la función predeterminada de la parte a) llamada leastsq, y por último el Método utilizado en la parte b).