INSTITUTO TECNOLÓGICO SUPERIOR DE CHICONTEPEC

INGENIERÍA EN SISTEMAS COMPUTACIONALES

NOMBRE DE LA MATERIA:

Métodos Numéricos.

NOMBRE DEL TEMA:

Resumen de las unidades 5 y 6.

Método de trapecio múltiple en Python.

UNIDADES:

5.- Interpolación y ajuste de funciones.

6.- Solución de ecuaciones diferenciales.

NOMBRE DEL ALUMNO:

Miguel Angel Martinez Martinez.

NOMBRE DEL DOCENTE:

Ing. Efrén Flores Cruz.

Índice

Introducción 3

Unidad 5. Interpolación y ajuste de funciones 4

Unidad 6. Solución de ecuaciones diferenciales 10

Ejemplo de método de trapecio múltiple en Python 15

Conclusión 17

Bibliografía 18

Introducción

En el apartado se vera un pequeño ejemplo de cómo se resolvió un ejemplo, donde el alumno se apoyará viendo un video propuesto por el profesor, donde tendrá que analizar, ver el porque de como fue que se programó. De tal forma que el alumno tenga esa lógica para poder programar en Python, ya que de otra forma se puede programar un problema en Python, dado los conocimientos aprendidos sabrá por qué se programó, ya que se utilizaría siclos para que el problema haga su función correspondiente.

Unidad 5. Interpolación y ajuste de funciones

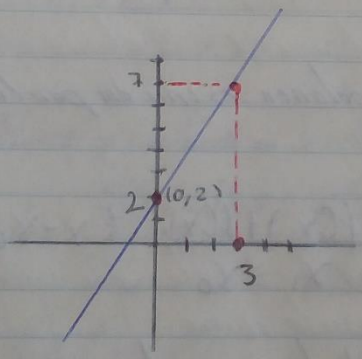
Unidad 5. Interpolación y ajuste de funciones

Interpolación.

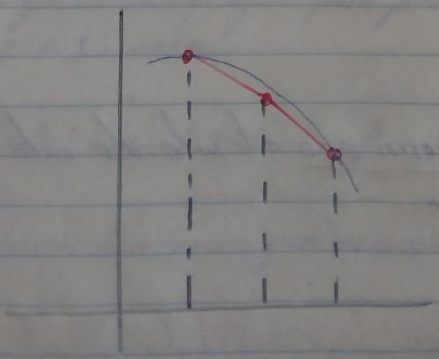
- La interpolación se refiere a estimar valores intermedios entre datos definidos por puntos
- El método más común que se utiliza en la interpolación polinomial
- Dados n+1 puntos, hay uno y solo polinomio de grado n que pasa a través de todos

Ejemplo:

- Solo hay una línea recta que une a dos puntos



- Solo hay una línea parabola que une un conjunto de tres puntos



Interpolacion Polinomial

- Consiste en determinar el polinomio unico de n-ésimo grado que se ajuste a n+1 puntos

Aunque hay uno y solo polinomio de n-ésimo grado que se ajusta n+1 puntos.

Interpolacion de Newton

El polinomio de interpolacion de Newton en diferencias divididas es uno de los metodos mas populares que existen.

Interpolacion lineal

La forma mas sencilla de interpolacion es unir dos puntos con una linea recta

$$f_1(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0)$$

En donde f_1 indica que se trata de un polinomio de interpolacion de primer grado

Interpolacion lineal

En general mientras mas sea el intervalo entre los datos, mejor sera la aproximacion

Interpolación cuadrática

- Una mejor estimación se logra introduciendo una curvatura en las líneas que unen los puntos.
- Si se trata o se tienen tres puntos como datos, estos pueden ajustarse a un polinomio de segundo grado

$$f_2(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1)$$

- Para encontrar b_0 se evalúa la ecuación anterior con $x = x_0$
- Sustituyendo y evaluando $x = x_1$

$$b_0 = f(x_0)$$

$$b_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

Sustituyendo y evaluando en $x = x_2$ se tiene

$$b_2 = \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}$$

- Forma general del polinomio de Newton

Se puede generalizar el ajuste para polinomios de n-ésimo grado a n+1 datos

$$f_n(x) = b_0 + b_1(x - x_0) + \ldots + b_n(x - x_0)(x - x_1)\ldots(x - x_{n-1})$$

Para un polinomio de n-ésimo grado se requieren n+1 puntos.

$$[x_0, f(x_0)], [x_1, f(x_1)], \ldots, [x_0, f(x_n)]$$

Forma general

De ahí se desprende

$$b_0 = f(x_0)$$
$$b_1 = [x_1, f(x_1)]$$
$$b_n = f[x_n, x_{n-1}, \ldots x_1, x_0]$$

Polinomio general

El polinomio general de interpolación de Newton es

$$f_n(x) = f(x_0) + (x - x_0)[x_1, x_0] + (x - x_0)(x - x_1) f[x_2, x_1, x_0]$$
$$+ \ldots + (x - x_0)(x - x_1)(x - x_{n-1}) f[x_n, x_{n-1}, \ldots, x_1, x_0]$$

Que también se conoce como polinomio de interpolación de Newton en diferencias divididas

Interpolación de Lagrange

Es una formulación de polinomio de Newton que evita el cálculo de diferencias divididas y se representa como

$$f_n(x) = \sum_{i=0}^{n} L_i(x) f(x_i)$$

En donde:

$$L_i(x) = \prod_{j=0}^{n} \frac{x - x_j}{x_i - x_j}$$

Interpolación de Spline

Es conveniente dividir el intervalo en pequeños subintervalos más pequeños y de esta manera utilizar polinomios de grado menor.
Una función spline se forma de varios polinomios unidos entre sí por ciertas condiciones de continuidad.
La interpolación a través de Spline Cubico ha demostrado ser la más eficiente.

Definición.

Dado los siguientes datos:

x	x_0	x_1	...	x_n
y	y_0	y_1	...	y_n

En donde $x_0 < x_1 < \ldots < x_n$ y k es un número entero positivo la función de Spline $S(x)$ es:

- $S(x) = y_i$, para todo $i = 0, 1, 2 \ldots, n$
- $S(x)$ es un polinomio de grado menor o igual a k en cada subintervalo $[x_{i-1}, x_i]$
- $S(x)$ tiene derivada continua hasta el orden $k-1$ en $[x_0, x_n]$

Polinomios de Splines

Dado el conjunto de datos, el sistema de Splines se define de la siguiente manera:

$$S(x) = \begin{cases} S_1(x) \text{ Si } x \in [x_0, x_1] \\ S_2(x) \text{ Si } x \in [x_1, x_2] \\ S_3(x) \text{ Si } x \in [x_2, x_3] \\ \vdots \end{cases}$$

En donde $S_n(x)$ es un polinomio de grado n

Spline de Grado 1, 2 y 3.

Spline de grado 1:

$$S(x) = \begin{cases} y_0 + f[x_1, x_0](x - x_0) & \text{si } x \in [x_0, x_1] \\ y_{n+1} + f[x_1, x_0](x - x_{n-1}) & \text{si } x \in [x_{n-1}, x_n] \end{cases}$$

en donde $f(x_1, x_0)$ es una diferencia dividida de Newton

Spline de grado 2:

$$S(x) = \begin{cases} a_1 x^2 + b_1 x + c_1 & \text{si } x \in [x_0, x_1] \\ a_{n-1} x^2 + b_{n-1} x + c_{n-1} & \text{si } x \in [x_{n-1}, x_n] \end{cases}$$

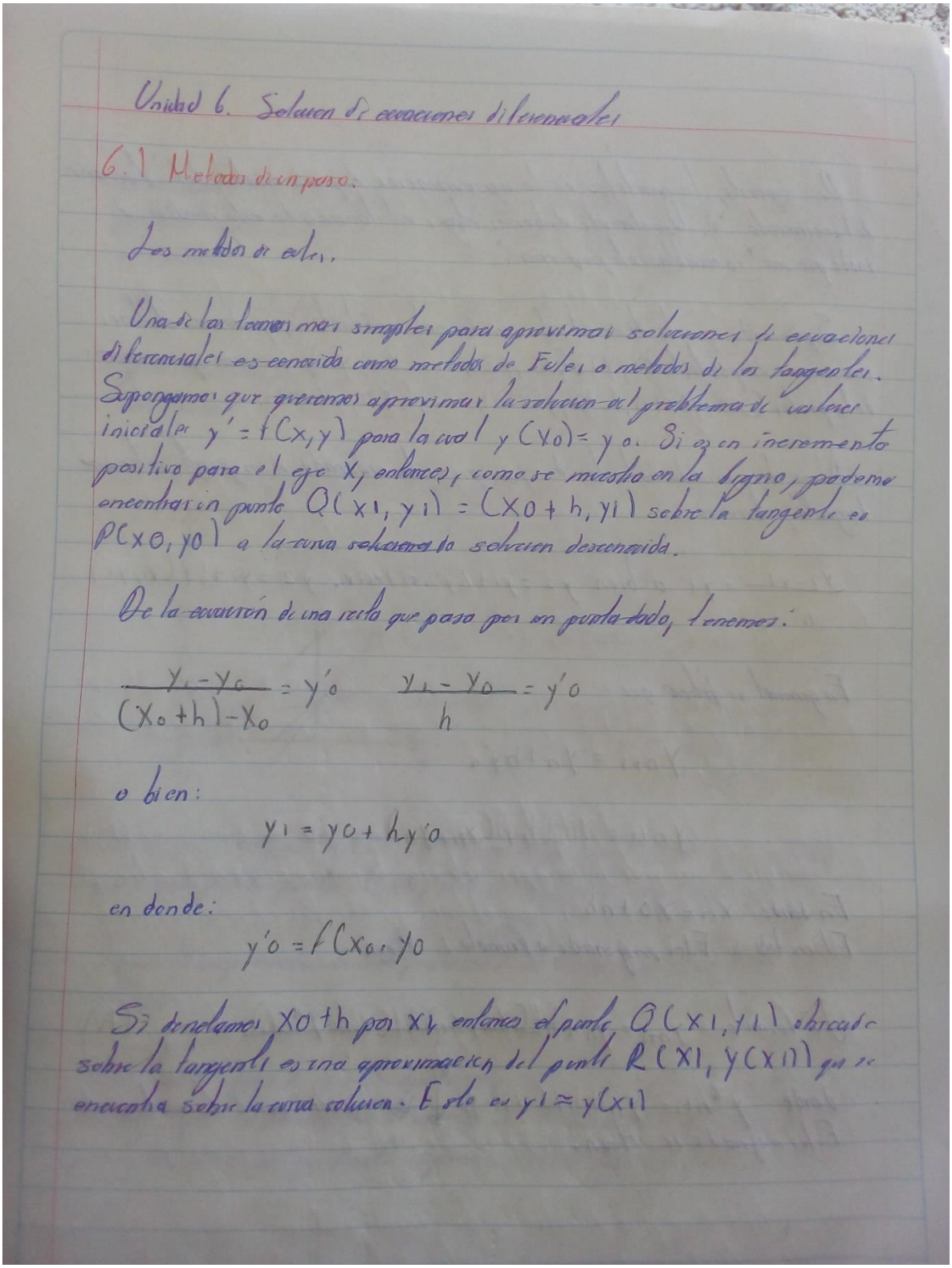
Spline de grado 3.

$$S(x) = \begin{cases} a_1 x^3 + b_1 x^2 + c_1 x + d_1 & \text{si } x \in [x_0, x_1] \\ a_{n-1} x^3 + b_{n-1} x^2 + c_{n-1} + d_{n-1} & \text{si } x \in [x_{n-1}, x_n] \end{cases}$$

Consideraciones:

- Los valores de la función de polinomios adyacentes deben ser iguales en los nodos interiores
- La primera y la última función deben pasar a través de los puntos extremos
- Las primeras derivadas en los nodos interiores deben ser iguales
- Suponer que el primer punto, la segunda derivada es cero, suponer que el primer interior tiene un comportamiento lineal

Unidad 6. Solución de ecuaciones diferenciales



Unidad 6. Solución de ecuaciones diferenciales

6.1 Métodos de un paso.

Los métodos de Euler.

Una de las técnicas más simples para aproximar soluciones de ecuaciones diferenciales es conocida como métodos de Euler o métodos de las tangentes. Supongamos que queremos aproximar la solución del problema de valores iniciales $y' = f(x,y)$ para la cual $y(x_0) = y_0$. Si es un incremento positivo para el eje x, entonces, como se muestra en la figura, podemos encontrar un punto $Q(x_1, y_1) = (x_0 + h, y_1)$ sobre la tangente en $P(x_0, y_0)$ a la curva solución desconocida.

De la ecuación de una recta que pasa por un punto dado, tenemos:

$$\frac{y_1 - y_0}{(x_0 + h) - x_0} = y'_0 \qquad \frac{y_1 - y_0}{h} = y'_0$$

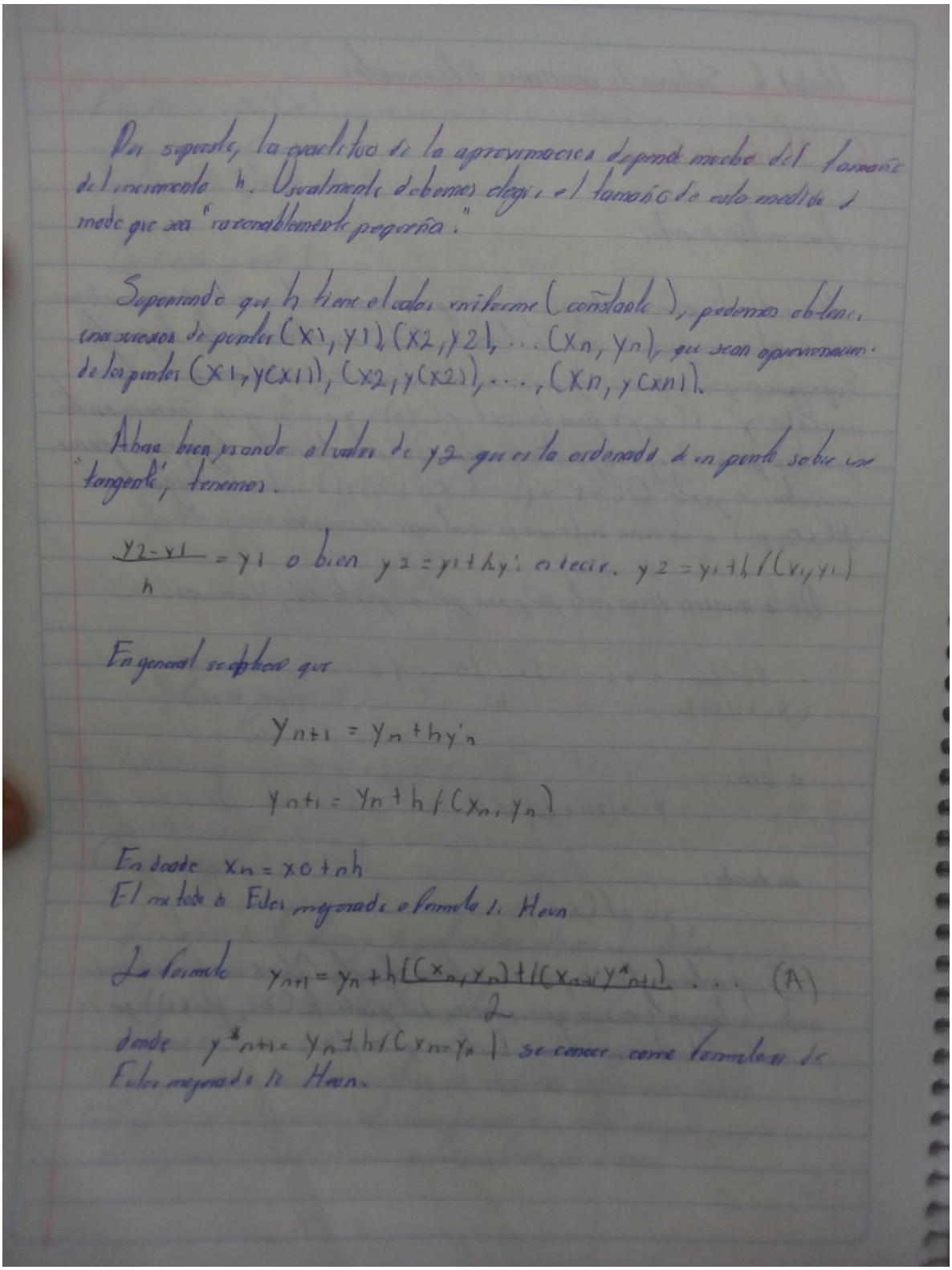
o bien:

$$y_1 = y_0 + h y'_0$$

en donde:

$$y'_0 = f(x_0, y_0)$$

Si denotamos $x_0 + h$ por x_1, entonces el punto $Q(x_1, y_1)$ ubicado sobre la tangente es una aproximación del punto $R(x_1, y(x_1))$ que se encuentra sobre la curva solución. Esto es $y_1 \approx y(x_1)$



Por supuesto, la exactitud de la aproximación depende mucho del tamaño del incremento h. Usualmente debemos elegir el tamaño de esta medida de modo que sea "razonablemente pequeña."

Suponiendo que h tiene el valor unitario (constante), podemos obtener cinco sucesivos de puntos $(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)$, que sean aproximaciones de los puntos $(x_1, y(x_1)), (x_2, y(x_2)), \ldots, (x_n, y(x_n))$.

Ahora bien usando el valor de y_2 que es la ordenada de un punto sobre una "tangente", tenemos.

$\frac{y_2 - y_1}{h} = y_1'$ o bien $y_2 = y_1 + hy_1'$ o lo cual. $y_2 = y_1 + hf(x_1, y_1)$

En general se obtiene que

$$y_{n+1} = y_n + hy'_n$$

$$y_{n+1} = y_n + hf(x_n, y_n)$$

En donde $x_n = x_0 + nh$
El método de Euler mejorado o fórmula de Heun

La fórmula $y_{n+1} = y_n + h\frac{[(x_n, y_n) + f(x_{n+1}, y^*_{n+1})}{2}$ . . . (A)

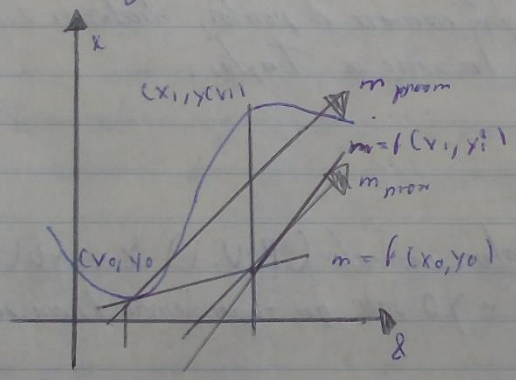
donde $y^*_{n+1} = y_n + hf(x_n, y_n)$ | se conoce como fórmula de Euler mejorado de Heun.

Los valores de $f(x_n, y_n)$ y $f(x_{n+1}, y^*_{n+1})$ son aproximaciones de la pendiente de la curva en $(x_n, y(x_n))$ y $(x_{n+1}, y(x_{n+1}))$ y en consecuencia el cociente

$$\frac{f(x_n, y_n) + f(x_{n+1}, y^*_{n+1})}{2}$$

puede ser interpretado como una pendiente promedio en el intervalo entre x_n, x_{n+1}. Las ecuaciones de (A) se pueden visualizar fácilmente.

En la figura se muestra el caso en que $n = 0$

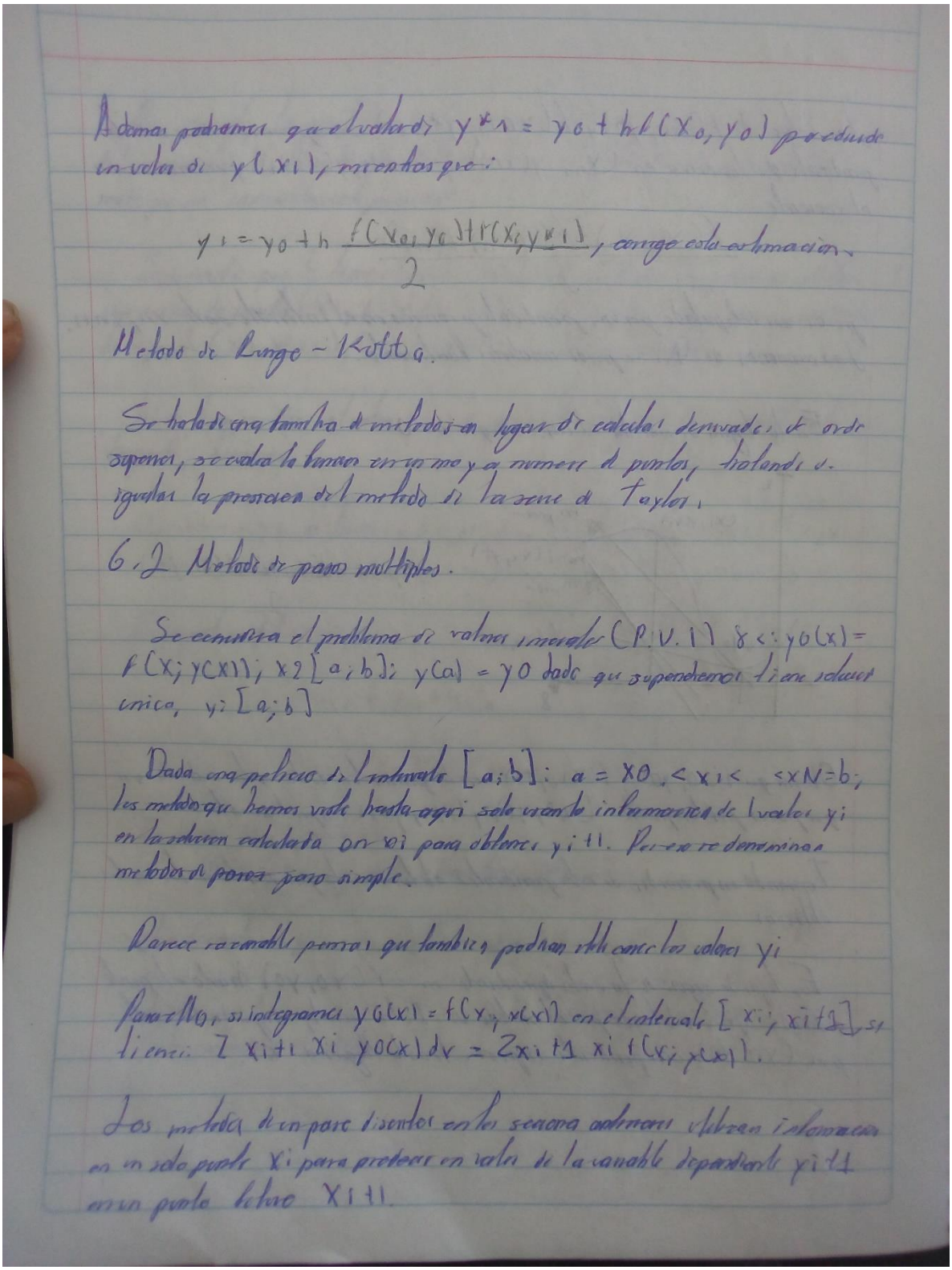


$$m_{prom} = \frac{f(x_0, y_0) + f(x_1, y^*_1)}{2}$$

Obsérvese que $f(x_0, y_0)$ y $f(x_1, y^*_1)$ son las pendientes de la recta indicada que pasan por los puntos (x_0, y_0) y (x_1, y^*_1) respectivamente

Tomando un promedio de estas pendientes obtenemos la pendiente de las rectas obtenidas

En lugar de seguir la recta dependiente $m = f(x_0, y_0)$ hasta el punto ordenada y^*_1 obtenida por el método de Euler usual, seguido la recta por (x_0, y_0) con pendiente m_{prom} hasta llegar a x_1.



Además podremos que el valor de $y^*_1 = y_0 + h f(x_0, y_0)$ produce un valor de $y(x_1)$, mientras que:

$$y_1 = y_0 + h \frac{f(x_0, y_0) + f(x_1, y^*_1)}{2}$$, corrige esta estimación.

Método de Runge - Kutta.

Se trata de una familia de métodos en lugar de calcular derivadas de orden superior, se evalúa la función en varios puntos, tratando de igualar la precisión del método de la serie de Taylor.

6.2 Método de pasos multiples.

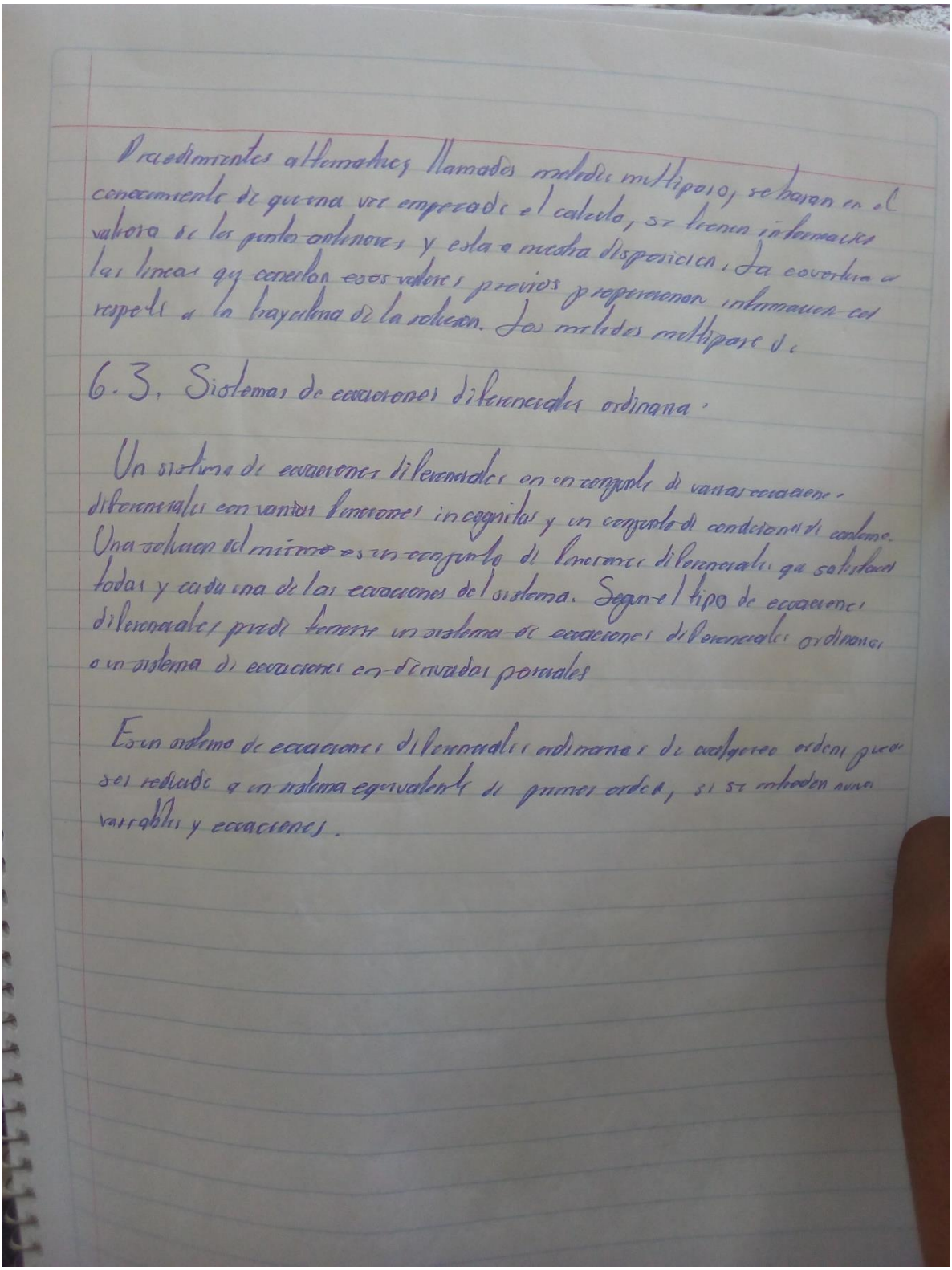
Se considera el problema de valores iniciales (P.V.I) $\forall x: y_0(x) = f(x; y(x1); x2[a;b]; y(a) = y_0$ dado que suponemos tiene solución única, $y: [a;b]$

Dada una partición de el intervalo $[a;b]: a = x_0 < x_1 < \ldots < x_N = b$; los métodos que hemos visto hasta ahora solo usan la información de los valores y_i en la abscisa calculada en x_i para obtener y_{i+1}. Por eso se denominan métodos de pasos simple.

Parece razonable pensar que también podrían utilizarse los valores y_i

Para ello, si integramos $y_0(x) = f(x, y(x))$ en el intervalo $[x_i; x_{i+1}]$, se tiene: $\int_{x_i}^{x_{i+1}} y_0(x)\,dx = \int_{x_i}^{x_{i+1}} x_i f(x, y(x))$.

Los métodos de un paso discutidos en las secciones anteriores utilizan información en un solo punto x_i para proceder en valor de la variable dependiente y_{i+1} en un punto futuro x_{i+1}.



Procedimientos alternativos llamados metodos multipaso, se basan en el conocimiento de que una vez empezado el calculo, se tienen informacion valiosa de los puntos anteriores y esta a nuestra disposicion. La cuestion es las lineas que conectan esos valores previos proporcionan informacion con respecto a la trayectoria de la solucion. Los metodos multipaso se

6.3. Sistemas de ecuaciones diferenciales ordinarias.

Un sistema de ecuaciones diferenciales es un conjunto de varias ecuaciones diferenciales con varias funciones incognitas y un conjunto de condiciones de contorno. Una solucion del mismo es un conjunto de funciones diferenciables que satisfacen todas y cada una de las ecuaciones del sistema. Segun el tipo de ecuaciones diferenciales, puede tenerse un sistema de ecuaciones diferenciales ordinarias o un sistema de ecuaciones en derivadas parciales

Es un sistema de ecuaciones diferenciales ordinarias de cualquier orden puede ser reducido a un sistema equivalente de primer orden, si se entienden nuevas variables y ecuaciones.

Ejemplo de método de trapecio múltiple en Python

El ejemplo es el siguiente, se trata sobre el método múltiple de un trapecio y de tal forma estructurarlo en el lenguaje Python, donde para eso tendremos que ver como se estructura el problema, de tal forma lo podemos ver en el siguiente video que nos enviaron, donde nos dan los siguientes datos del trapecio:

	x	F(x)
1	0	0.9162
2	0.25	0.8109
3	0.5	0.6931
4	0.75	0.5596
5	1	0.4055

Tabla 1. Tabla de valore para el método del trapecio múltiple.

Dado los valores, tendremos que conocer la formula necesario para poder realizar el código, la fórmula es la siguiente:

Donde:

a = Límite inferior.

b = Límite superior.

n = Número de trapecios.

$$f(x_0) + 2\sum_{i=1}^{n-1} f(x_j) + f(x_n)$$

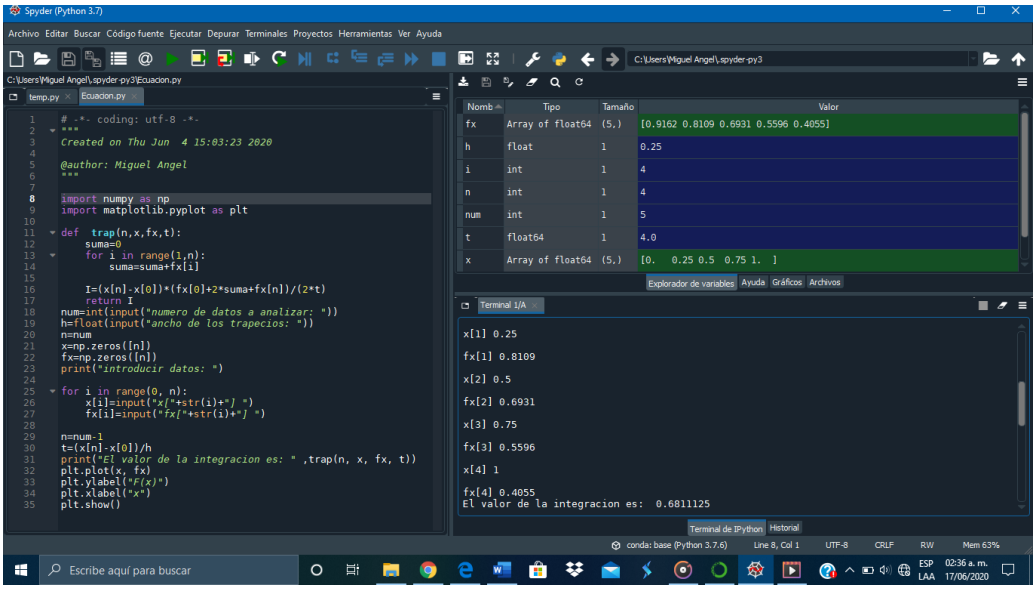


Imagen 1. Ejemplo de método de trapecio múltiple en Python

Luego tendremos que reconstruir el código que aparece en el video, pero antes se tiene que tener en cuenta como es que funciona el método, se que se explica en el video, pero hay saber también como funciona para poder saber cómo funciona el código.

Al final hay que compilarlo para ver el posible resultado. La grafica aparece en el apartado de **Gráficos** y veremos cómo salió la gráfica.

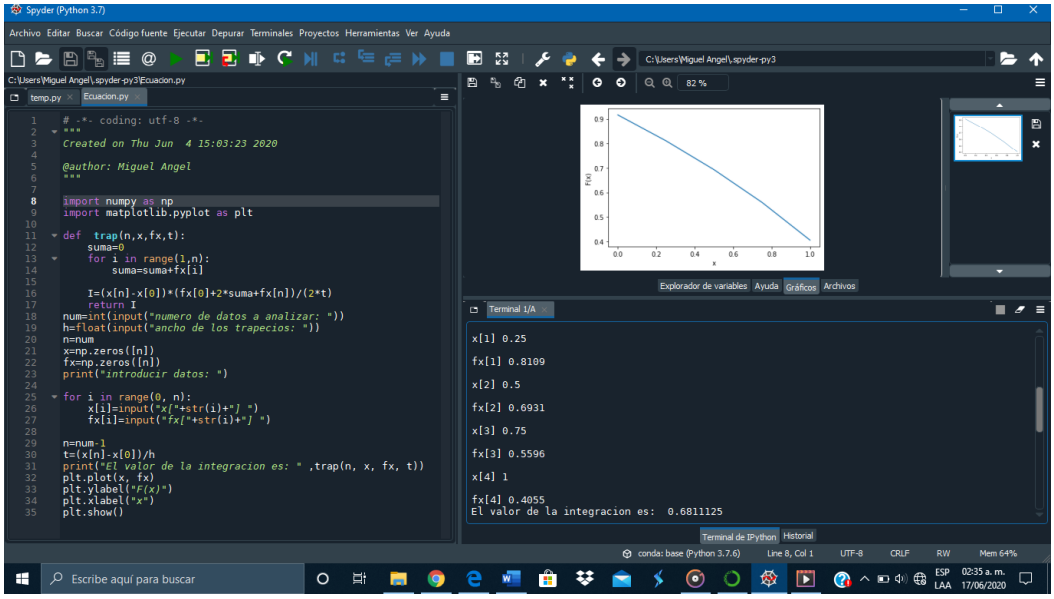


Imagen 2. Interfaz gráfica de primer ejemplo de método de trapecios múltiples en Python.

Conclusión

En el pequeño apartado vimos un pequeño ejemplo sobre el método de trapecio múltiples realizado en Python, dado dicho ejemplo fue parte de la unidad que estábamos cursando, donde era de entender como funcionaba el problema que nos plantío el profe que viéramos, de tal forma de dar a entender que cualquier problema que tengamos que realizar se puede hacer tanto en el lenguaje Java como en la de Python, y de una forma también que nos familiaricemos con otro lenguaje.

En el video que vimos tuvimos que comprender es que funcionaba, porque, antes que nada, saber qué información ocuparemos, de tal forma para poder realizar la programación y darle los valores que se necesita, dicho esto, para ver el posible resultado de problema.

Bibliografía

Metodos Numericos. (s.f.). Recuperado el 24 de Junio de 2020, de http://itpn.mx/recursosisc/4semestre/metodosnumericos/Unidad%20V.pdf