

Análisis Numérico - TP 2

Métodos numéricos aplicados a la Ingeniería de procesos

Rocío Gallo, padrón 97490 (rochimgg@gmail.com)

Facundo Monpelat, padrón 92716 (facundo.monpelat@gmail.com)

4 de julio de 2019

1. Introducción

Uno de los procesos más comunes de la industria metalúrgica es el tratamiento térmico de aceros. Este tratamiento térmico le dará al acero diversas propiedades mecánicas en el cual el trabajo se enfocará a calcular solamente el tratamiento de Revenido. Este trabajo consiste en emplear métodos numéricos para encontrar resoluciones a una ecuación diferencial del calor que da la pauta, por ejemplo, de cuanto tiempo deberá estar el acero (en este caso un tubo) en el horno bajo ciertas temperaturas.

2. Objetivos

- Experimentar el uso de métodos numéricos para la resolución de ecuaciones diferenciales no lineales.
- Aplicar métodos numéricos de diferentes temas de la materia y ensamblarlos para resolver un problema de mayor complejidad.
- Adquirir conocimientos básicos de la ingeniería de procesos industriales.

3. Desarrollo

El intercambio de calor entre un sólido y su entorno está dado por la ley de conservación de energía bajo la siguiente expresión:

$$-mc \frac{dT}{dt} = h_c S(T - T_\infty) + \gamma S(T^4 - T_\infty^4)$$

donde el primer término representa el intercambio de calor debido al proceso de convección, mientras que el segundo término representa el intercambio de calor debido a la radiación.

1. Métodos de resolución

Para plantear los métodos, se acomodó la ecuación diferencial de la siguiente forma

$$\begin{cases} T' = \frac{-h_c S(T - T_\infty)}{mc} \\ T_0 = 293,5K \end{cases} \quad (1)$$

a. Método de Euler

Para este método la iteración viene dada por la expresión explícita:

$$T_{i+1} = T_i + hT'(T_i, t(i))$$

El valor de h es el tiempo de permanencia de un tubo en cada uno de los espacios que hay dentro del horno de revenido. El h puede ser calculado mediante la siguiente expresión:

$$h = \frac{t_{Per}}{N_{Espacios}}$$

donde t_{Per} es el tiempo de permanencia de cada tubo en el horno de revenido y $N_{Espacios}$ es la cantidad de espacios para tubos dentro del mismo horno.

La función escrita en python que implementa este método es:

```
#-----
# FUNCION euler(f,g,h,x0,xf,y0,data):
#
# PARAMETROS
# f:          Funcion de EDO
# g:          Funcion solucion analitica (Calculo de error)
# h:          Incremento o cadencia
# x0 y y0:    Valores iniciales
# xf :        Valor final
# data :      Array de diccionarios que contiene los datos X e Y de cada iteracion
# USO        Calcula los valores con el metodo de Euler
#-----
def euler(f,g,h,x0,xf,y0,data):
    t = x0
    y = y0
    i=0
    while t <= xf:
        delta = abs( y-g(t) )/g(t)
        aux_dict={
            'i':i,
            'X':t,
            'Y':y,
            'E':delta,
        }
        data.append(aux_dict)
        i=i+1
        y += h * f(t,y)
        t += h
    return data[-1]['X'],data[-1]['Y']
```

b. Método de Runge-Kutta Orden 4

Para el calculo de este método se utilizaron las expresiones:

$$\begin{cases} k_1 = f(t_0, T_0) \\ k_2 = f(t_0 + \frac{h}{2}, T_0 + h(\frac{k_1}{2})) \\ k_3 = f(t_0 + (\frac{h}{2}), T_0 + h(\frac{k_2}{2})) \\ k_4 = f(t_0 + h, T_0 + h k_3) \end{cases}$$

donde t_0 es las semilla de la variable tiempo (en segundos) y T_0 es la semilla de la variable temperatura (en Kelvin).

En los programas creados para este trabajo se las renombro de la forma: $t_0 = x_0$ y $T_0 = y_0$.

La iteración de Runge - Kutta con estos coeficientes viene dada por la expresión:

$$T_{n+1} = T_n + \frac{h}{6} (k_1 + 2 k_2 + 2 k_3 + k_4)$$

La función escrita en python que implementa este método es:

```
#-----
# FUNCION rungeKutta(f,g,h,x0,xf,y0,data,i)
#
# PARAMETROS
# f:          Funcion de EDO
# g:          Funcion solucion analitica (Calculo de error)
# h:          Incremento o cadencia
```

```

# x0 y y0: Valores iniciales
# xf :      Valor final
# data :    Array de diccionarios que contiene los datos X e Y de cada iteracion
# i :      Contador de iteraciones
# USO      Calcula los valores con el metodo de Runge-Kutta
#-----

def rungeKutta(f,g,h,x0,xf,y0,data,i):
    debug = 0
    # K1 = F(Xn,Yn)
    # K2 = F( Xn + h/2 , Yn + h*K1 /2 )
    # K3 = F( Xn + h/2 , Yn + h*K2 /2 )
    # K4 = F( Xn + h/2 , Yn + h*K3 )
    k1 = f(x0, y0)
    k2 = f(x0 + (h/2), y0 + h*(k1/2))
    k3 = f(x0 + (h/2), y0 + h*(k2/2))
    k4 = f(x0 + h, y0 + h*k3)
    # Yn+1 = Yn + h/6( K1 + 2K2 + 2K3 + K4 )

    y = y0 + (h/6)*(k1 + 2*k2 + 2*k3 + k4)
    delta = abs( y0-g(x0) )/g(x0)

    aux_dict={
        'i':i,
        'X':x0,
        'Y':y0,
        'E':delta
    }

    if debug: pprint(aux_dict)

    data.append(aux_dict)

    if( round(x0) == xf ):
        if debug: print('X final: ' + str(data[-1]['X']) + ' / Y final: ' + str(data[-1]['Y']))
        return data[-1]['X'],data[-1]['Y']

    else:
        return rungeKutta( f, g, h, x0+h, xf, y, data, i+1)

```

II. Ejercicio 1

Resolver el modelo planteado considerando solamente el intercambio de calor por convección. Aplicar los métodos de Euler y Runge Kutta de orden 4 para una condición inicial $T_0 = 20^\circ\text{C}$ y paso de tiempo $h = \text{cadencia}$. Para los cálculos se utilizó el número de padrón de Roció Gallo (97490).

Propiedades del material	ρ	7850 kg/m ²
	C	480 J/kgK
Geometría del material	OD	0.244 48 m
	WT	0.013 84 m
	Lt	12 m
Geometría del horno	L	50 m
	nbol	50
Parámetros de proceso	cad	$ROUND(-10 \text{ sec}/10000 * (97490 - 90000) + 35 \text{ sec}) = 28 \text{ sec}$
	T ₁	$ROUND(200^\circ\text{C}/10000 * (97490 - 90000) + 500^\circ\text{C}) = 650^\circ\text{C} = 923.15 \text{ K}$
	T ₂	$ROUND(200^\circ\text{C}/10000 * (97490 - 90000) + 500^\circ\text{C}) = 650^\circ\text{C} = 923.15 \text{ K}$
Parámetros de la transferencia de calor	hc	20 W/m ² K
	σ	5,6703 10 ⁻⁸ W/m ² K ⁴
	ϵ	0.85

Notar que para este caso particular donde $T_\infty = T_1 = T_2$, la solución analítica (exacta) tiene la siguiente expresión:

$$T(t) = T_{\infty} + (T_0 - T_{\infty})e^{\frac{h_c S}{mC}t}$$

- Graficar la temperatura en función del tiempo superponiendo los resultados de cada método junto con la solución exacta.
- Graficar el error relativo cometido con cada método en función del tiempo. Utilizar una escala en el eje vertical que permita visualizar con claridad ambas curvas.
- Obtener conclusiones sobre la precisión de ambos métodos aplicados.

IMPORTANTE: Los cálculos de temperatura y tiempo asociados a la ecuación diferencial deberán realizarse en grados Kelvin [K] y segundos [sec], respectivamente. Por otro lado, los resultados (ya sean numéricos o plasmados en gráficos) deberán expresarse en grados Celsius [°C] y minutos [mín].

Sabiendo que la superficie es:

$$S = \pi OD L_t = 9.2167 \text{ m}^2$$

y la masa:

$$m = \rho \pi OD WT \left(1 - \frac{WT}{OD}\right) L_t = 944.6512 \text{ kg}$$

La ecuación a resolver con $T_0 = 20^\circ\text{C} = 293.15 \text{ K}$ es:

$$\begin{aligned} T' &= -4,0631 \cdot 10^{-4} (T - T_{\infty}) \\ T' &= -4,0631 \cdot 10^{-4} (T - 923,15) \end{aligned} \quad (2)$$

Se resuelven ambos métodos creando un script en python. Se ejecuta el código y este abre 2 ventanas en un navegador predeterminado con los gráficos.

Ejecucion:

```
Skywalker:EJ1 fmonpelat$ python3.7 TP2-ej1.py
Datos calculados:
m [kg] = 945.9618816430382
S [m2] = 9.228742579185376
Cadencia [seg] : 28.0
v_0 [m/s] = 0.03571428571428571
```

La gráfica comparativa de los métodos con respecto a la solución analítica es la siguiente:

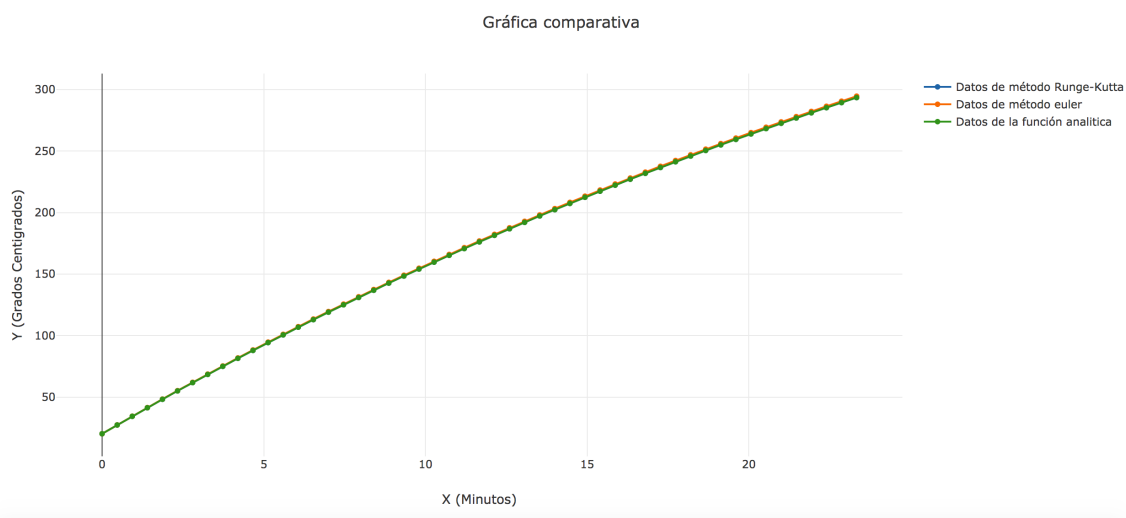


Figura 1: Comparativa entre métodos.

y la imagen que muestra los errores de cada método con escala vertical logarítmica es:

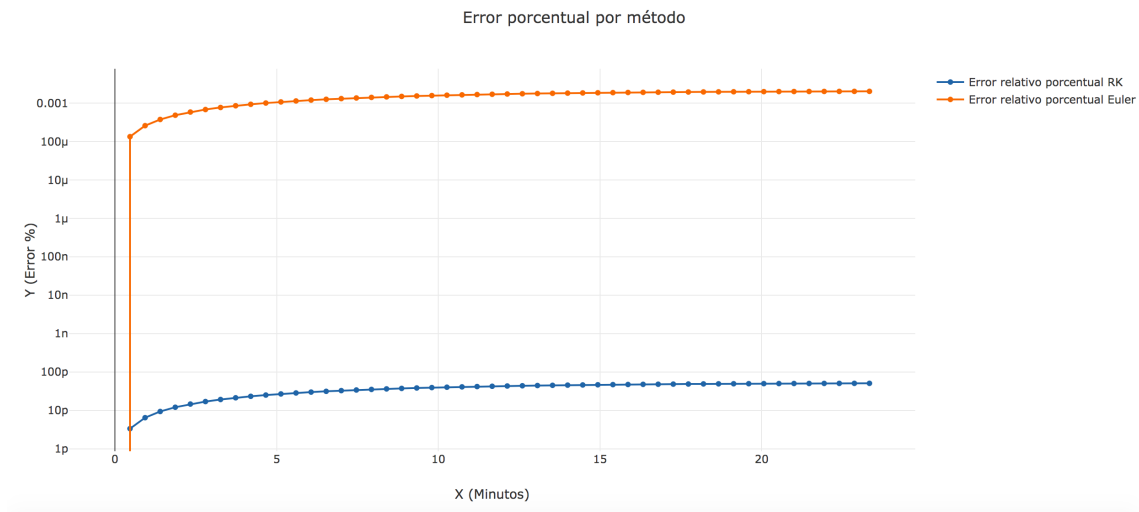


Figura 2: Comparativa entre métodos.

Se nota que los errores con respecto a la solución analítica son mas apreciables con Euler que con el método de Runge-Kutta. Uno esta en el rango de los micro y el otro en el rango de los pico y por este motivo se puede afirmar que el método de Runge-Kutta es mas preciso.

III. Ejercicio 2

a. Elección

Para hacer el análisis de la ecuación de conservación de energía con ambos términos se optará por utilizar el método numérico de Runge-Kutta de orden 4. Cómo se ha visto en el primer punto, la distancia entre la gráfica analítica y la aproximada mediante Runge-Kutta es significativamente menor que la obtenida mediante el método de Euler y por lo tanto mas precisa.

Para resolver este punto se creó otro script en python donde su ejecución fue la siguiente:

```
Skywalker:EJ2 fmonpelat$ python3.7 TP2-ej2.py
Datos calculados:
m [kg] = 945.9618816430382
S [m2] = 9.228742579185376
Cadencia [seg] : 28.0
v_0 [m/s] = 0.03571428571428571
Tsk(K): 874.613710100969 K
Tsk(C): 601.613710100969 C
Sk(seg): 224.0 seg
Sk(min): 3.7333333333333334 min
```

y la gráfica fue la siguiente:

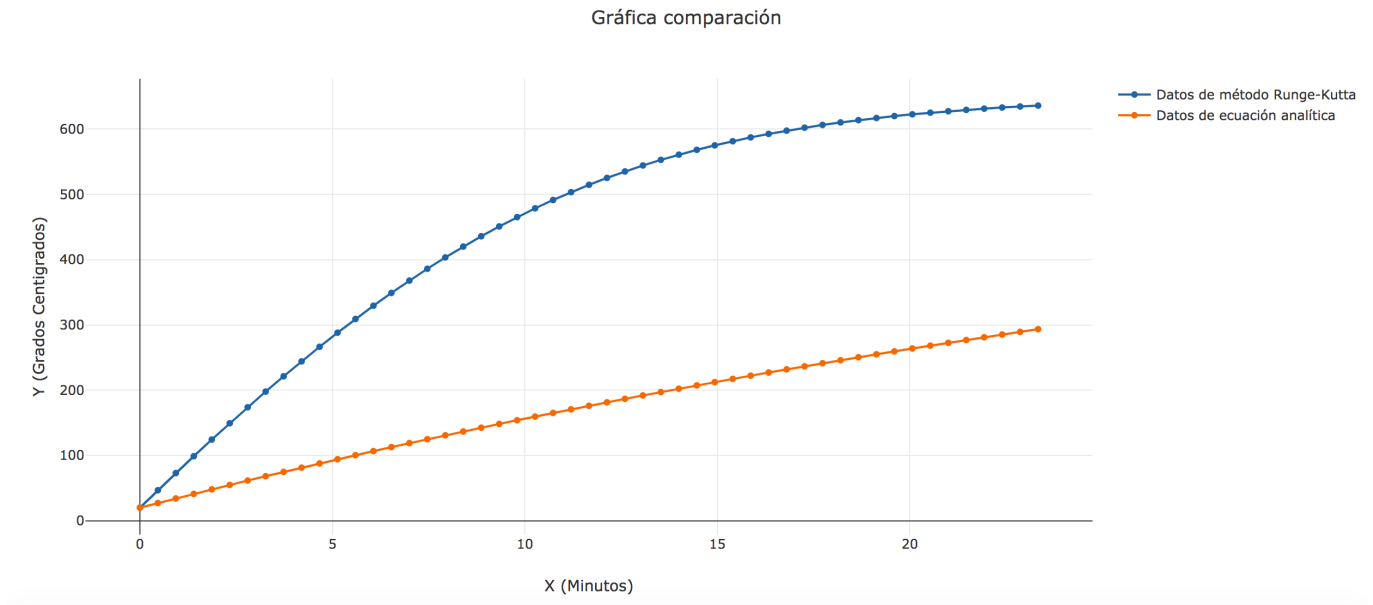


Figura 3: Solución RK con intercambio radiativo y solución analítica sin intercambio radiativo.

Como se puede ver en la gráfica el intercambio por radiación no es despreciable. El código retornó la temperatura media en el tiempo de soaking y su tiempo correspondiente: Tsk(K): 874.613 Kelvin Tsk(C): 601.613 Centígrados
Sk(seg): 224.0 segundos Sk(min): 3.733 minutos

IV. Ejercicio 3

Este ejercicio pide encontrar manualmente una combinación de T_1 y T_2 para que el S_k sea de 10 minutos manteniendo la T_{sk} obtenida en el ítem 2. La tolerancia para S_k es de $\pm 0,5min$.

Para resolverlo fue fijada la temperatura de soaking del ejercicio anterior $T_{sk}(K)$: 874.613 K. Cambiando T_1 y T_2 manualmente e imprimiendo el tiempo en minutos para una temperatura próxima a T_{sk} obtenemos con los valores: $T_1 = 1002$ Kelvin y $T_2 = 922.5$ Kelvin.

```
Skywalker:EJ3 fmonpelat$ python3.7 TP2-ej3.py
```

Datos calculados:

m [kg] = 945.9618816430382

S [m2] = 9.228742579185376

Cadencia [seg] : 28.0

v_0 [m/s] = 0.03571428571428571

Sk Objetivo [segundos] : 600

Tsk Objetivo [C] = 602

tiempo Sk: 10.267 [min] tiempo obj: 10.000 [min] temp mean: 606.066 [C] temp obj: 602.00 [C]

como la tolerancia pedida es 0.5 minutos la medición que llegamos con esos valores cumple ($S_k = 10,267 - 10 = 0,267 minutos$).

La nueva gráfica que cumple lo pedido será:

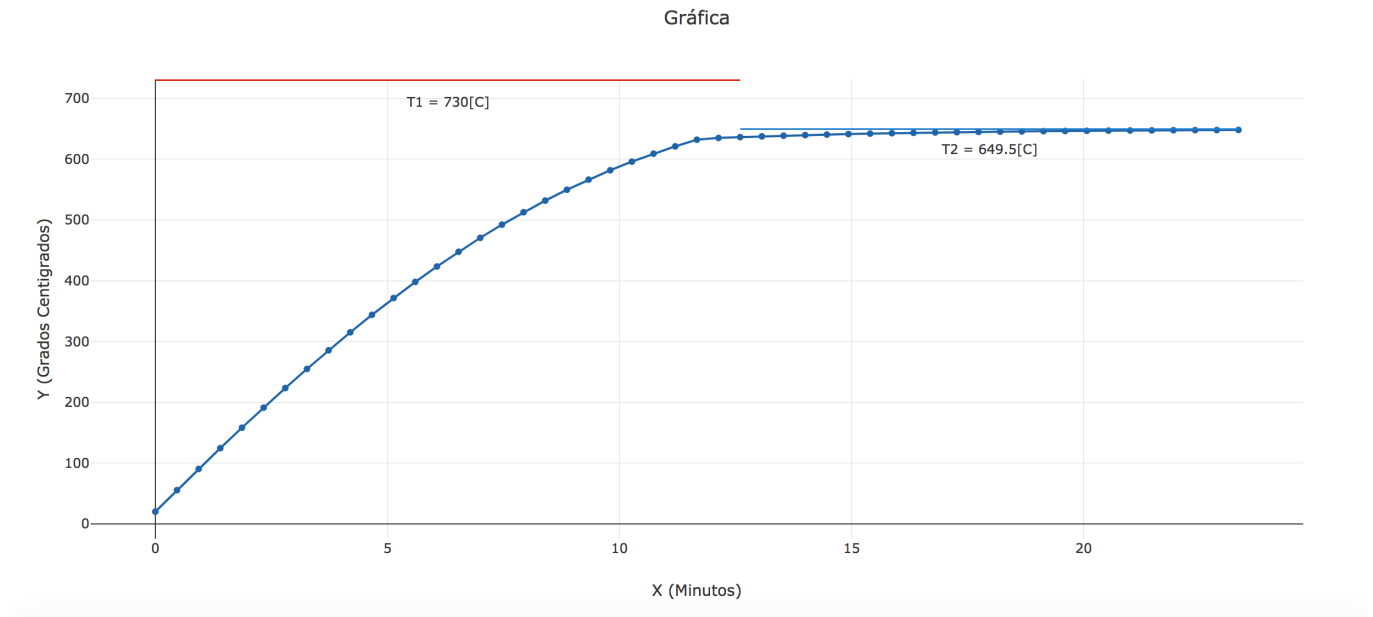


Figura 4: Solución numérica con los valores $T_1 = 1002$ Kelvin y $T_2 = 922.5$ Kelvin

V. Ejercicio 4

Realizar una mejora del ciclo térmico obtenido en el ítem 3 en términos de productividad. Es decir, aumentar la productividad un 5 % sin alterar los parámetros metalúrgicos de interés (iguales valores de S_k y T_{sk}).

a. Algoritmo

Para lograr un rendimiento un 5 % mayor será necesario llegar a los mismos T_{Sk} y S_k pero en un tiempo menor.

Para conseguir esto, se optó por multiplicar la cadencia de la consigna por 0,95 y así obtener unas temperaturas T_1 y T_2 que corresponden a un rendimiento un 5 % mayor. Los valores obtenidos para T_1 y T_2 fueron

$$T_1 = 846,93C^\circ$$

$$T_2 = 722,733C^\circ$$

Este punto fue realizado luego de plantear el inciso 5, porque era más directo el cálculo habiendo resuelto el problema del hallazgo automatizado de T_1 y T_2 .

VI. Ejercicio 5

a. Algoritmo

Para la resolución de este ejercicio, el enunciado propone utilizar un esquema de punto fijo con la siguiente matriz jacobiana

$$J = \begin{bmatrix} 0,25 & 0,75 \\ 0,75 & 0,25 \end{bmatrix}$$

cuya inversa está dada por

$$J^{-1} = \begin{bmatrix} -0,5 & 1,5 \\ 1,5 & -0,5 \end{bmatrix}$$

Dado que T_{Sk-obj} y S_{kobj} son las raíces del modelo planteado anteriormente se plantea la siguiente iteración de punto fijo.

$$\begin{bmatrix} T_1^{k+1} \\ T_2^{k+1} \end{bmatrix} = \begin{bmatrix} T_1^k \\ T_2^k \end{bmatrix} - \begin{bmatrix} -0,5 & 1,5 \\ 1,5 & -0,5 \end{bmatrix} \begin{bmatrix} F_1(T_1^k, T_2^k) \\ F_2(T_1^k, T_2^k) \end{bmatrix}$$

Donde F_1 y F_2 vienen dados por la siguiente expresión

$$F_1(T_1, T_2) = f_1(T_1, T_2) - T_{Sk-obj} = 0$$

$$F_2(T_1, T_2) = f_2(T_1, T_2) - S_{k_{obj}} = 0$$

Luego de realizados los cálculos, fueron obtenidas las siguientes temperaturas T_1 y T_2 para cada caso solicitado.

Caso	Sk	Tsk	T1	T2	Iteración
A	10	602	860.382	708.72	62
B	10	624.9	772.171	733.519	52
C	10	674.9	666.048	776.556	36

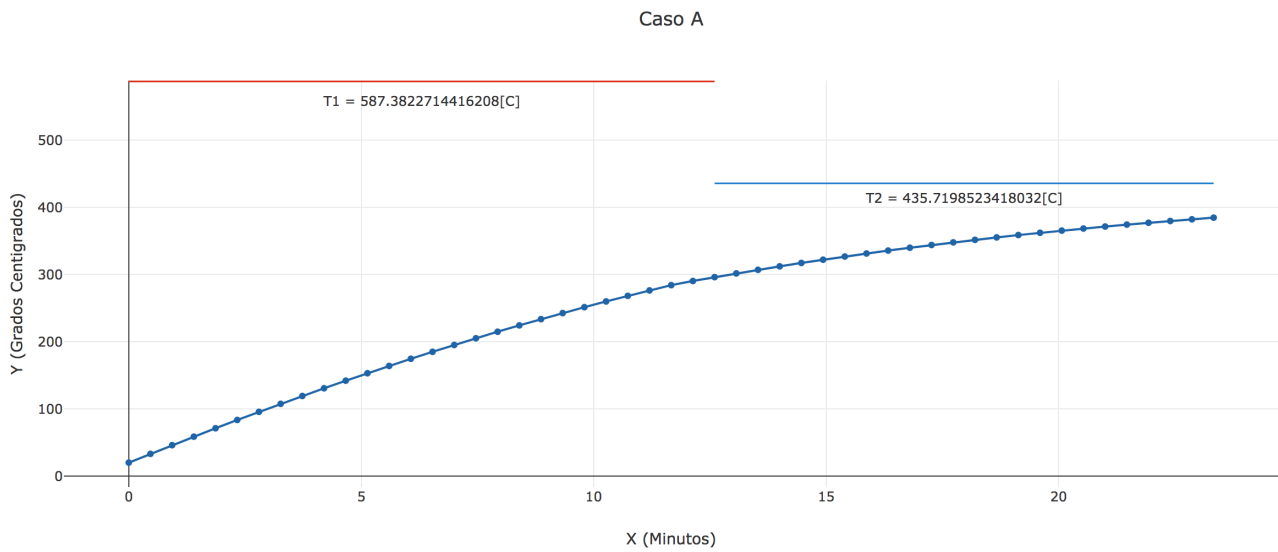


Figura 5: Gráfico para el caso A

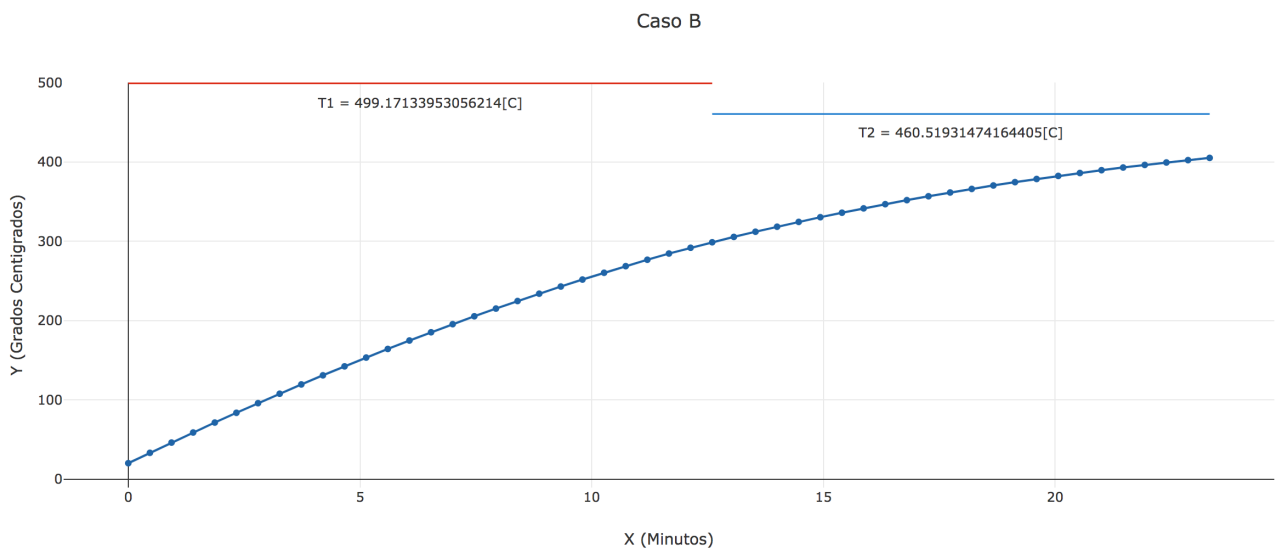


Figura 6: Gráfico para el caso B

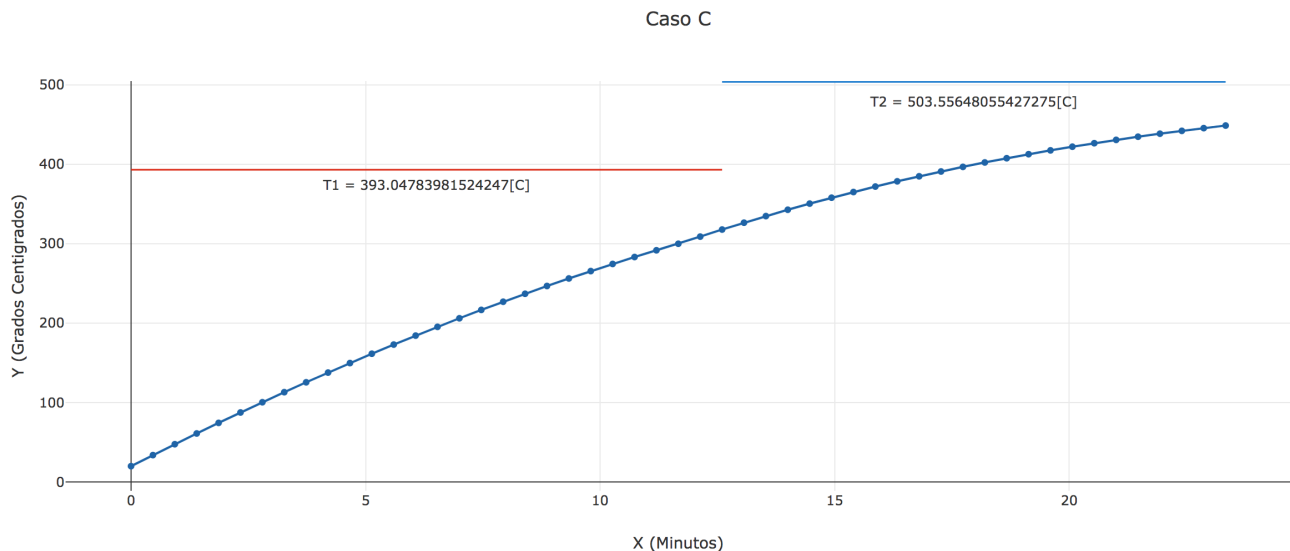


Figura 7: Gráfico para el caso C

4. Conclusiones

En este trabajo práctico han sido utilizados dos métodos de resolución de ecuaciones diferenciales para analizar el problema del revenido en la fabricación de tubos de acero con distintas necesidades mecánicas.

En un comienzo se usó únicamente el término referente al pasaje de calor mediante convección, el cual es una ecuación lineal. Para terminar de analizar el problema se le agregó a esta ecuación, el término correspondiente al pasaje de calor debido a la radiación, este término es exponencial. De ambos modos, fueron resueltas las ecuaciones diferenciales pero la principal diferencia entre una y otra es la dificultad del despeje de la función solución.

En las ecuaciones diferenciales no lineales suele ser muy dificultoso o casi imposible el despeje a mano de la función a encontrar, debido a esto, los métodos numéricos sirven como una poderosa herramienta para hallar soluciones con una precisión aceptable a ecuaciones diferenciales no lineales.

En este caso, fueron utilizados los métodos numéricos para encontrar la solución a un problema inverso. Conociendo los valores de la imagen de la función, usarlos para así encontrar valores específicos de la preimagen de la misma. Esto es posible porque al ir aplicando cada uno de los métodos, quedan grabadas relaciones entre preimagen e imagen en una tabla, que pueden ser utilizadas a posteriori en el análisis.

Para acercarse mejor a la realidad, podría plantearse en la ecuación de conservación de energía, el calor debido al proceso de conducción. Ya que los tubos están apoyados sobre cada uno de los bolsillos, podría analizarse la pérdida o ganancia de calor referente al contacto entre el tubo y la superficie.

Referencias

- [1] Algoritmo de Euler y Runge-Kutta de orden 2 para resolver una EDO
<https://www.youtube.com/watch?v=hgKzFdAZHwY>
- [2] Wikipedia - Método de Runge - Kutta
https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods
- [3] Wikipedia - Método de Euler
https://es.wikipedia.org/wiki/M%C3%A9todo_de_Euler