Tarea 2: Sistemas No Lineales, Interpolación e Integración

Instrucciones

- **Fecha de entrega: 22 de Agosto**
- Se debe entregar en un archivo zip (Ej: 'MiguelSalvatierraTareaX.zip'):
  - informe de la tarea en formato .pdf, hecho en LaTeX.
  - todos los códigos utilizados (funciones y scripts).
- Los códigos deben ir comentados.
- El informe debe contener los resultados principales.
- Se puede trabajar en grupo, pero se deben entregar informes y códigos individuales. Tareas y códigos idénticos tendrán nota 0.
- Todas las operaciones deben estar contenidas dentro del script. Se revisarán sólo los resultados obtenidos al presionar *run*. **No ejecutar por línea de comando**.
- Para cada uno de los siguientes problemas se debe considerar el desarrollo de los programas sin utilizar las funciones establecidas del matlab
- **Todos los códigos deben compilar**.

Pregunta 1

Para la próxima FIDAE se pide a usted que programe la trayectoria y velocidad de un avión no tripulado. Se le entregan a usted los datos de la velocidad parametrizada $v_x(t)$ y $v_y(t)$ (en kilómetros por segundo) que se necesitan para el evento. (Estos datos vienen en un archivo .mat en aula).
Para programar la velocidad se le pide que entregue un polinomio(s) al programa del avión no tripulado.

Figura 1: Avión No Tripulado

Para esto vea las opciones:

- Interpole con Método de Lagrange, haga una función para su método.
- Interpole con el Método de Newton, haga una función para su método.
- Interpole utilizando Splines, haga una función para su método.

Grafique los resultados de la interpolación para cada componente de la velocidad con respecto al tiempo y grafique la velocidad total en función del tiempo.
Nota: La interpolación debe dar como resultado un polinomio que usted pueda evaluar en Matlab.

Distancia que recorre el Avión

Se le pide además que le entregue el recorrido total del avión después de 20 segundos, para esto integre numéricamente desde $t = 0[s]$ hasta $t = 20[s]$ la integral de línea tal que el recorrido esta definido por:

$$\int_a^b \sqrt{\left(\frac{\partial x(t)}{\partial t}\right)^2 + \left(\frac{\partial y(t)}{\partial t}\right)^2} dt$$

Donde $x(t)$ e $y(t)$ son las funciones de posición parametrizadas. **Integre con los métodos de**:

- **Trapecio Compuesto**
- **Simpson 1/3 Compuesto**
- **Simpson 3/8 Compuesto**

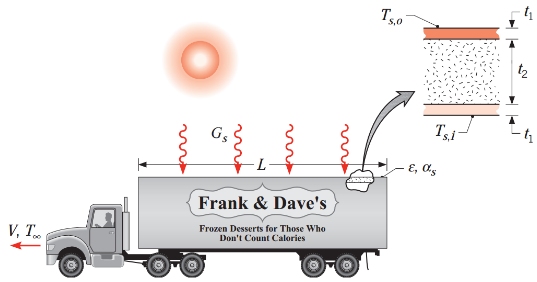
Para los métodos utilice $N = [10, 25, 50, 100]$ particiones. Haga **una** función para cada método.
Estudie como varía el error de cada método con respecto a la solución analítica, que es: 292,438$[m]$. Para esto **grafique en escala logarítmica el error respecto al número de particiones utilizadas para cada caso.** (*Hint: Utilice el comando loglog(X, Y) para su gráfico*).

Además responda a la pregunta ¿Depende del método de interpolación la distancia recorrida? y si es así ¿Cuál es el método que recorre menos distancia?. Esto es importante ya que el combustible depende directamente de esto último.

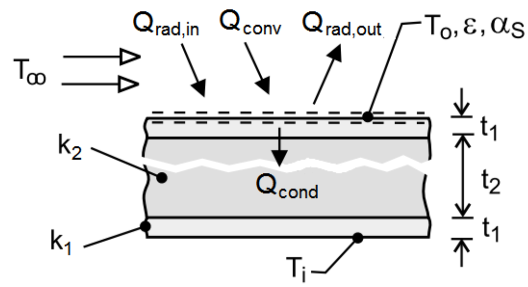
Hint: Para calcular la integral, usted tiene los polinomios resultantes de los métodos de Newton y Lagrange, que son fáciles de evaluar e integrar numéricamente. Para el caso de los splines, usted tiene polinomios por tramos, por lo que la integral debe ser realizada entre cada par de puntos que forman la cada spline.

Pregunta 2

Un camión refrigerado se desplaza por la carretera. Las paredes de la zona refrigerada están compuestas de dos láminas de aluminio separadas entre sí por espuma aislante. Sobre el techo incide radiación solar y, además, está sometido a convección forzada con el aire ambiental.



Esquema del camión.



Balance de Flujos.

Al realizar un balance de energía sobre el techo del camión se obtiene:

$$E_{in} - E_{out} = 0$$
$$Q_{rad,in} + Q_{conv} - Q_{rad,out} - Q_{cond} = 0$$

Los flujos de calor unidimensionales se expresan como:

$$Q_{rad,in} = \alpha G_s A_s \qquad Q_{conv} = hA_s(T_\infty - T_s)$$
$$Q_{rad,out} = \varepsilon\sigma A_s T_s^4 \qquad Q_{cond} = \frac{A_s k_1 k_2}{2t_1 k_2 + t_2 k_1}(T_s - T_i)$$

Donde α corresponde a la absortividad del techo, ε la emisividad del techo, σ la constante de Stefan-Boltzman, G_s la radiación solar incidente, A_s el área superficial del techo del camión, h el coeficiente de convección en el techo del camión, T_s la temperatura superficial exterior del techo, T_i la temperatura interior del techo, T_∞ la temperatura del aire exterior, y, por último, k_i y t_i las conductividades térmicas y espesores del aluminio (1) y la espuma aislante (2).

En un caso particular se tienen las siguientes condiciones: $A_s = 35$ $[m^2]$, $\alpha = \varepsilon = 0{,}5$ $[-]$, $h = 56$ $[W/m^2K]$, $t_1 = 5$ $[mm]$, $t_2 = 50$ $[mm]$, $k_1 = 180$ $[W/mK]$, $k_2 = 0{,}026$ $[W/mK]$, $G_s = 750$ $[W/m^2]$, $T_\infty = 305$ $[K]$ y $T_i = 263$ $[K]$.

1. Reemplazando los flujos de calor Q_i en el balance de energía determine funciones para calcula T_s con los métodos de: Bisección, Punto Fijo y Newton-Raphson.
2. Mostrar en el informe si las funciones encontradas cumplen los requisitos para utilizar cada método (Teorema de Bolzano, Rolle y Contractividad).
3. Elija un método que sea aplicable y calcule con una tolerancia de 10^{-6} la temperatura superficial del camión (T_s) y el calor que ingresa a la zona refrigerada (Q_{cond}) en los siguientes casos:
   a) Caso base (valores entregados).
   b) Si se elimina la espuma aislante ($t_2 = 0$ $[mm]$).
   c) Si se mantiene el aislante ($t_2 = 50$ $[mm]$), pero se utiliza un recubrimiento con propiedades $\alpha = 0{,}15$ y $\varepsilon = 0{,}8$.
4. Indique que opción (a, b o c) minimiza el flujo de calor hacia la zona de refrigeración.

Pregunta 3

Utilizando el método de Newton-Raphson para sistemas de ecuaciones no lineales, encuentre las intersecciones de las curvas:

$$x^2 + 2y^2 + 2xy = 1$$
$$x^5 + y^3 - 2xy = 1{,}5$$

- Grafique ambas curvas con el comando `ezplot()` para estimar la cantidad de soluciones y su posición.
- Encuentre las intersecciones empleando una tolerancia de 10^{-5}.

El método de Newton-Rhapson de sistema de ecuaciones no lineales consiste en despejar $\overrightarrow{x_{n+1}}$ en el sistema:

$$F(\overrightarrow{x_n}) + J(\overrightarrow{x_n})(\overrightarrow{x_{n+1}} - \overrightarrow{x_n}) = 0.$$

donde $\overrightarrow{x_n}$ corresponde a los puntos donde está siendo evaluado (depende del tamaño del sistema de ecuaciones), $J(\overrightarrow{x_n})$ corresponde al Jacobiano evaluado en un punto $\overrightarrow{x_n}$, y $F(\overrightarrow{x_n})$ corresponde a la función evaluada en el punto $\overrightarrow{x_n}$.

El sistema puede resolverse como:

$$\begin{aligned} J(x_n) \cdot \Delta x_{n+1} &= -F(x_n) \\ x_{n+1} &= x_n + \Delta x_{n+1} \end{aligned}$$

O como:

$$\overrightarrow{x_{n+1}} = \overrightarrow{x_n} - J^{-1}(\overrightarrow{x_n})F(\overrightarrow{x_n})$$

Recordatorio:

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad F(\vec{x}) = \begin{bmatrix} f_1(\vec{x}) \\ f_2(\vec{x}) \\ \vdots \\ f_n(\vec{x}) \end{bmatrix} \qquad J(\vec{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\vec{x}) & \frac{\partial f_1}{\partial x_2}(\vec{x}) & \cdots & \frac{\partial f_1}{\partial x_n}(\vec{x}) \\ \frac{\partial f_2}{\partial x_1}(\vec{x}) & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1}(\vec{x}) & \cdots & \cdots & \frac{\partial f_n}{\partial x_n}(\vec{x}) \end{bmatrix}$$

Anexo - Splines

Para formar un spline cúbico $S(x)$ de la forma:

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3 \tag{1}$$

Donde $j = 0, 1, ..., n - 1$ es el número de intervalos (o nodos) de los puntos a interpolar.
Luego aplicando la condición:

$$S_j(x_{j+1}) = S_{j+1}(x_{j+1}), j = 0, 1, ..., n - 2 \tag{2}$$

Tenemos que:

$$a_{j+1} = S_j(x_{j+1}) = S_{j+1}(x_{j+1}) \tag{3}$$

Luego usaremos la siguiente notación $h_j = x_{j+1} - x_j$, para $j = 0, 1, ..., n - 2$. Por lo que a_j es igual a :

$$a_j = f(x_j) = y_j \tag{4}$$

Se define además:

$$b_n = S'(x_n) \tag{5}$$

$$\rightarrow S'(x) = b_j + 2c_j(x - x_j) + 3d_j(x - x_j)^2 \tag{6}$$

Si aplicamos la condición de que la derivada debe ser igual en los extremos de cada spline.

$$S'_j(x_{j+1}) = S'_{j+1}(x_{j+1}), j = 0, 1, ..., n - 2 \tag{7}$$

Por lo que:

$$b_{j+1} = S'_j(x_{j+1}) = S'_{j+1}(x_{j+1}) = b_j + 2c_j(h_j) + 3d_j(h_j)^2 \tag{8}$$

Además digamos que:

$$c_n = \frac{S''(x_n)}{2} \tag{9}$$

Y con la condición de que la segunda derivada debe ser igual en los extremos de cada spline:

$$S''_j(x_{j+1}) = S''_{j+1}(x_{j+1}), j = 0, 1, ..., n - 2 \tag{10}$$

Por lo que c_{j+1} es igual a:

$$c_{j+1} = c_j + 3d_jh_j \tag{11}$$

Por lo que podemos dejar a d_j en función de:

$$d_j = \frac{c_{j+1} - c_j}{3h_j} \tag{12}$$

Si si utilizamos la ecuación 3 tenemos que:

$$a_{j+1} = a_j + b_jh_j + c_jh_j^2 + d_jh_j^3 \tag{13}$$

Reemplazando 12 en 8 y en 3 quedan las ecuaciones:

$$a_{j+1} = a_j + b_jh_j + \frac{h_j^2}{3}(3c_j + c_{j+1}) \tag{14}$$

$$b_{j+1} = b_j + h_j(c_j + c_{j+1}) \tag{15}$$

Por lo que podemos dejar a b_j en función de a_{j+1}, a_j y c_j, c_{j+1}.

$$b_j = \frac{1}{h_j}(a_{j+1} - a_j) - \frac{h_j}{3}(2c_j + c_{j+1}) \tag{16}$$

Finalmente si tomamos la ecuación 15 y la evaluamos en j (en vez de $j+1$) podemos reemplazar la ecuación 16 evaluada en j y $j-1$. Con lo que queda la siguientes ecuaciones lineales:

$$h_{j-1}c_{j-1} + 2(h_{j-1} - h_j)c_j + h_j c_{j+1} = \frac{3}{h_j}(a_{j+1} - a_j) - \frac{h_{j-1}}{3}(2c_{j-1} + c_j) \tag{17}$$

Notar que a_j y h_j son valores conocidos (ecuación 4). Por lo que el sistema lineal Ax=b queda:

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & ... & 0 & 0 & 0 \\ h_0 & 2(h_0+h_1) & h_1 & 0 & ... & 0 & 0 & 0 \\ 0 & h_1 & 2(h_1+h_2) & h_2 & ... & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & ... & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ 0 & 0 & 0 & 0 & ... & 0 & 0 & 1 \end{pmatrix} \tag{18}$$

$$b = \begin{pmatrix} 0 \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \vdots \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 0 \end{pmatrix} ; x = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ \vdots \\ c_n \end{pmatrix} \tag{19}$$

Conociendo los coeficientes c_j despejamos los b_j y d_j tal que:

$$b_j = \frac{1}{h_j}(a_{j+1} - a_j) - \frac{h_j}{3}(2c_j + c_{j+1}) \tag{20}$$

$$d_j = \frac{c_{j+1} - c_j}{3h_j}. \tag{21}$$

Como A es diagonal dominante existe una única solución al sistema, por lo que podemos resolverlo por el método PA=LU.

Anexo - Funciones Anónimas

Las funciones anónimas prestan varios beneficios a la hora de trabajar en matlab. En este anexo se presentan algunas ideas para facilitar el desarrollo de esta tarea.

Variables

Es posible definir las variables antes de declarar una función, de esta forma resulta más fácil modificar parámetros de estudio:

```
% Variables del problema
a = 5;
b = -2

% funcion anonima.
f = @(x) a*exp(b);
```

Se debe tener precaución, ya que al modificar una variable se debe volver a declarar (ejecutar la línea) la función para que esta 'guarde' el nuevo valor.

De la misma forma se pueden calcular variables más complejas para reducir la longitud de una ecuación:

```
% Variables basicas
a = 5;
b = -2

f = @(x) a^2/b*x + b/5;

% Constantes de la funcion
C1 = a^2/b;
C2 = b/5;

f = @(x) C1*x + C2;
```

Vectores

Las funciones anónimas pueden recibir vectores como argumentos y entregar vectores como salida:

```
% Funcion anonima
f = @(v) [ v(1)+v(2) ; v(1)-v(2)]; %Opera con los indices 1 y 2 del argumento.

%Por ejemplo, para el vector
r = [1;3]
A = f(r)
%A seria el vector [4;-2]
```