Tarea 8
(para el 5 abril)

Nota: esta tarea es más larga de la usual. Les recomiendo empezar hacerla temprano y mandarme cualquier duda, no esperar a la última hora.

1. Leer las notas de la clase de 16 marzo y completar los problemas de las tarea 6 y 7 que no has hecho. Sobre todo prob. 17 y 19 de la pág. 255 de Palm.

   Sugerencias para el problema 19.
   - Primero, haces (b) y (c), graficando la $y(t)$ sin ningun "criterio", para ver de qué están hablando. Para esto, defines la función $y(t)$ del texto (por archivo separado, o en una sola linea en el script, como vimos en la última clase) y luego la t de alguna manera arbitraria. Digamos así
     ```
     t_step=.1;
     t_max=10;
     t=0:t_step:t_max;
     ```
     Usa esto para graficar la $y(t)$ en (a) y (b), jugando con los valores de las variables `t_step, t_max` y observando qué pasa.
   - Ahora lees con cuidado la parte (a). Básicamente, la idea es que para τ grande, le toma a las oscilaciones más tiempo desaparecer, así que la `t_max` tiene que ser más grande. Luego, cuando la ω es más grande, las oscilaciones son más "rápidas"(periodo pequeño), así que necesitas resolución más fina, tomando el `t_step` más pequeño.

2. Esto es un problema similar al problema 4 de la guía del primer parcial, para la función $y = \sin(x)$.

   El polinomio de Taylor de grado n de $y = \sin(x)$ en $x = 0$ es

   $$p_n(x) = \sum_{k=0}^{(n+1)/2} (-1)^k \frac{x^{2k+1}}{(2k+1)!} = x - \frac{x^3}{3!} + \ldots + (-1)^{n+1}\frac{x^n}{n!}, \quad n = 1, 3, 5, 7, \ldots.$$

   Nota que $p_n(x)$ está definido solo para n impar. Así que, por ejemplo,

   $$p_1(x) = x,$$
   $$p_3(x) = x - \frac{x^3}{6}$$
   $$p_5(x) = x - \frac{x^3}{6} + \frac{x^5}{120}$$

   (En teoría, $p_n(x)$ está definido también para n par, pero es el mismo polinomio que $p_{n-1}(x)$, así que no nos interesa. Esto sucede para cualquier función *impar*, $f(-x) = -f(x)$).

   a) Define una función `sin_taylor(n,x)` cuyo valor en `n,x` es $p_n(x)$. Asegura que tu función acepta un vector x, no solo escalar. En las notas de la clase de 16 de marzo explico como hacerlo para el problema 4 de la guia, que es muy similar.

b) Usa el inciso anterior para graficar la función $y = \sin(x)$ junto con $p_n(x)$, para $n = 1, 3, 5, 7, 9, 11, 13$. Exprimenta con distintos rangos de x para obtener un rango que ilustra bien la situación. Agrega en tu gráfica los ejes de x y y (Matlab por alguna razon no los pone), y etiqueta las gráficas, para poder ver claramente qué gráfica corresponde a qué función.

Nota: compara tu gráfica con la primera gráfica del artículo de Wikipedia sobre la Serie de Taylor.

c) Vamos a investigar qué tan buenos son los $p_n(x)$ para calcular valores de $\sin(x)$. Tomamos digamos $x = 0.1$

Acuerdate que estamos trabajando en radianes. ¿Cuántos grados (aprox.) son 0.1 radianes?

Respuesta: $0.1(180/\pi) \approx 18/3 = 6$ grados.

Usamos los polinomios `sin_taylor(n,x)` para calcular $\sin(0.1)$. ¿Qué tan grande tiene que ser n para que nos de una precisión de 10^{-5}? Es decir, que el valor aproximado sea correcto hasta el 5 dígito decimal. Vamos hacerlo de dos maneras.

- Con un ciclo `while`, aumentando la n hasta que $|p_n(.1) - \sin(.1)| < 10^{-5}$. (Usa la función `abs`).

  Esto es bastante simple, pero un poco "tramposo", ya que estamos confiando en la función predefinida de Matlab $\sin(x)$. La pregunta es ¿qué hacemos si no tenemos la función "verdadera" $\sin(x)$ a la mano y queremos saber qué tan buena es una aproximación $p_n(.1)$? La respuesta es que tenemos que observar aproximaciones *sucesivas*. Esta sería la segunda manera.

- En un ciclo `while`, vamos aumentar la n y preguntar si la aproximación ofrecida por $p_n(.1)$ da una *correción* de la aproximación anterior, dada por $p_{n-2}(.1)$ (acuerdate que solo consideramos n impar), por más de 10^{-5}. Cuando ya estamos corregiendo por menos que 10^{-5}, sabemos que no tiene sentido seguir y detenemos el ciclo.

  (En prinicipio, podemos pensar que este argumento está mal, ya que aunque cada corrección es pequeña, es posible que muchas de ellas (inifinidad!) acumalan a una corrección grande. Es un tema delicado y no vamos entrar en ello ahora, pero básicamente se demuestra en cálculo avanzado que esto no sucede, ya que las correcciones se disminuyen muy rápido).

Resumen: usar ambos métodos para encontrar la n necesaria para aproximar $\sin(.1)$ con un error $< 10^{-5}$.

d) Usa el segundo metodo del inciso anterior para definir una función `grado_aprox_sin(x,E)` que nos de el grado n del polinimio de Taylor $p_n(x)$ necesario para aproximar $\sin(x)$ con un error $< E$. Verifique que `grado_aprox_sin(.1,10^(-5))` coincide con el inciso anterior.

e) Usa el inciso anterior para hacer gráficas de `n=grado_aprox_sin(x,E)` (n como función de x) para varios valores de E (por ejemplo, $E = 10^{-2}, 10^{-3}, 10^{-4}, \ldots$ etc).

3. Ejercicio 9 de esta página.

   Nota: el diagonal de una matriz cuadrada es la lista de los elementos desde la esquina superior-izquierda hasta la inferior-derecha. El anti-diagonal es la lista de los elementos desde la esquina superior-derecha hasta el inferior-izquierda.

4. Programar en Matlab el algoritmo de *Ordenamiento por Inserción* descrito en Wikepdia. Probarlo con un conjunto de 100 números, los cuales los puedes generar con la función `rand` de Matlab.