



Práctica - Ejercicio 1: El Sistema de Lorenz es un sistema dinámico caótico cuya derivada viene definida por las expresiones:

Siendo $a=10$, $b=28$ y $c=8/3$. Dada la expresión de las derivadas y un paso de tiempo $dt=0.001$, se puede obtener el estado del sistema en un tiempo T a partir del instante anterior $T-dt$ sin más que usar el método de Euler:

$$[x(T) \ y(T) \ z(T)] = [x(T-dt) \ y(T-dt) \ z(T-dt)] + dt * [dx/dt \ dy/dt \ dz/dt];$$

Considerando una condición inicial, $X0=rand(1,3)$,; haced un programa (lorenzSystem.m) que reproduzca la figura mostrada en la presentación.

Práctica - Ejercicio 2: Escribir un programa (modificadores.m) que reproduzca, con las modificaciones necesarias, los ejemplos de los modificadores de las cadenas de control pero con el nombre y edad del alumno, y transformando la edad a su valor en minutos.

Práctica - Ejercicio 3: Escribir los programas definidos en la presentación asignando los nombres: potencias2.m, potencias.m, menuPotencias.m, menuBuclePotencias.m y menuPotenciasOO.m.

Práctica - Ejercicio 4: Escribir un programa (mediaAritmetica.m) que muestre un menú para pedir un número o salir, y que al salir calcule la media de todos los números que el usuario ha ido introduciendo. Modificar el programa anterior para que muestre al media geométrica (mediaGeometrica.m).

Práctica - Ejercicio 5: A partir de los programas anteriores y de alguna de las definiciones de desviación estándar crear un programa (standardDeviation.m) que modifique los anteriores para calcular este estadístico de dispersión (variabilidad).

Práctica - Ejercicio 6: Escribir un programa (calculaEstadistico.m) que muestre un menú con los tres estadísticos anteriores y la opción de salir, y que, a continuación, solicite números hasta que el usuario decida, devuelva el estadístico y vuelva al menú inicial.

Práctica - Ejercicio 7: Realizar un programa (copiarLineasParesImpares.m) que leerá líneas hasta el final de fichero, y escribirá en un fichero las líneas pares y en otro las impares. El programa preguntará por los nombres de los ficheros, y se asegurará que los puede abrir sin ningún problema, si no fuese así volverá a preguntar. Si hubiera alguna línea de cabecera, ésta debe ser incluida en ambos ficheros. Usar como ejemplo el fichero stations.txt utilizado en la sesión de ayer e incluido en este tema.

Práctica - Ejercicio 8: Crear un fichero de texto con las 2 siguientes líneas de texto

2 3 4 5 6 7 8 12

8 7 6 5 4 3 2 1 14

Práctica - Ejercicio 9: Escribir un programa en Octave (operarVectores.m) que lea el fichero anterior de datos, sume todos los números en la primera fila y luego haga lo mismo para la segunda fila. Ha de pedir el nombre del fichero al usuario y asegurarse de que ha podido abrir el fichero sin problema, si no fuese así volverá a preguntar.

Práctica - Ejercicio 10: Multiplicar la suma de la primera fila por la suma de los segundos. Mostrar por pantalla el resultado de la siguiente forma (en una sola línea), para cada uno de los vectores (pista: sprintf):

La suma de los elementos del vector [1 2 3 4 5 6 7 8 12] vale: 00324

Práctica - Ejercicio 11: Hacer lo mismo para el producto escalar, mostrando en una línea el contenido de los vectores y el producto escalar.

Práctica - Ejercicio 12: Escribir todo lo anterior en un fichero de texto. Preguntar al usuario por el nombre del fichero y asegurarse de que ha podido abrir el fichero sin problema, si no fuese así volverá a preguntar

Práctica - Ejercicio 13: Realizar un programa (NombresFechas.m) que lea los apellidos, nombres y fechas de nacimiento (dd/mm/aaaa) que hay escritos en un fichero. Cada línea del fichero corresponderá con una persona, y cada campo (nombre, apellidos y fecha de nacimiento) estarán separados por un espacio en blanco (sscanf):

Adrian Lopez Garcia 12/09/1991

Victor Sanchez Lopez 03/2/1993

Fundamentos de Computación

Enunciados de Prácticas – Bloque 03

En otro fichero escribirá la lista, pero ajustando los campos de apellido y nombre a un ancho de 10 caracteres y en lugar de escribir la fecha de nacimiento, escribirá la edad de la persona. El programa preguntará por ambos ficheros, y se asegurará de que los puede abrir sin ningún problema, si no fuese así volverá a preguntar.

Práctica - Ejercicio 14: Descargar y descomprimir el fichero Precip.zip. Crear un programa precipAnual.m que calcule la precipitación anual en Cantabria a partir de los ficheros de precipitación mensual contenidos en dicho archivo. Comparar el resultado con el contenido en el fichero Precip1km.txt de ese mismo archive. Representar los resultados (help pcolor).

Práctica - Ejercicio 15: Realizar un programa (sacarDatos.m) que leerá el fichero stations.txt, utilizado en la práctica, y escribirá un fichero por cada campo con los datos asociados a ese campo. Es decir, el fichero de latitudes contendrá todas las latitudes que aparecen en el fichero stations.txt y sólo ese dato. Cada fichero se llamará como el nombre del campo que aparece en la cabecera del fichero stations.txt. Por ejemplo, el fichero con los nombres de las estaciones se llamará name.txt.

El programa preguntará por los nombres de los ficheros, y se asegurará que los puede abrir sin ningún problema. Si no fuese así volverá a preguntar.

Se entregarán los ficheros generados y el script de forma independiente. No se permitirán ficheros comprimidos, por más que contengan los ficheros generados.

Tarea 1: Realizar un programa (menuFicheros.m) que solicite con menú el nombre de la Comunidad Autónoma (CCAA) y el tipo de energía y extraiga del fichero T_125CA_0114.txt el dato correspondiente. En ambos casos, la CCAA y el tipo de energía, existirá una opción que permita extraer todos los datos, es decir, todos los tipos de energía para una CCAA dada o viceversa, los datos de todas las CCAA para un tipo de energía definido.

Los resultados se mostrarán por pantalla debidamente formateados de modo que puedan ser interpretados por el usuario (no un conjunto de números aislados).

Práctica - Ejercicio 16: Los 4 ficheros de ejemplo recogen datos reales de producción de electricidad obtenidos de la web del Ministerio de Energía, Turismo y Agenda Digital. El fichero original se ha modificado para implementar 4 formatos que recogen gran parte de la casuística asociada a la lectura de ficheros.

Práctica - Ejercicio 17: Realizar un programa para leer cada tipo de fichero y comparar los datos obtenidos. El nombre de cada programa seguirá la siguiente estructura: T_125CA_0114_ Format .m, siendo Format la extensión del fichero (csv, txt, mixed y blanks).

Tarea 2: En un aerogenerador de 600 kW, el rango óptimo de producción se encuentra a velocidades de viento entre 15 y 20 m/s, si bien se mantiene encendido con rangos de velocidad entre 5 y 25 m/s. Para velocidades por debajo de 5 m/s o por encima de 25 m/s, el aerogenerador no se conecta por defecto o exceso de viento, respectivamente. Por otro lado, para velocidades en el rango de funcionamiento, la potencia obtenida se puede estimar con el cubo del valor de velocidad (v^3).

La empresa de electricidad toma medidas día a día en todos los aerogeneradores de un parque, guardando los registros de cada año en un fichero. En el fichero wss_WFDEI_1981.txt se han incluido los datos correspondientes al año 1981.

Realizar un programa (windSpeed.m) que lea los datos de velocidad del viento de dicho fichero, estime la producción total diaria del parque, dada por la suma de las producciones de cada aerogenerador, y la producción total anual de cada aerogenerador y del parque eólico al completo. Finalmente, el programa debe escribir un nuevo fichero (production_1981.txt) que contenga la información del fichero original más una columna nueva con los totales diarios y una nueva fila con los totales anuales de cada aerogenerador. El nuevo fichero respetará el formato del fichero original e incluirá en el cabecero del fichero el nombre de los nuevos campos introducidos, el cual será definido por el alumno.

Práctica - Ejercicio 18: Considerar el fichero stations.txt adjunto y las coordenadas de Torrelavega (Longitud: -4.0511019, Latitud: 43.3445828, Altura: 35 m). Leer los datos del fichero, estimar la distancia Euclídea de cada estación a Torrelavega, y escribir un nuevo fichero stationsComplete.txt que contenga los datos originales y que añada un nuevo campo, distTorrelavega, con la distancia del punto a Torrelavega. El programa deberá comprobar que ambos ficheros se abren correctamente.

Fundamentos de Computación

Enunciados de Prácticas – Bloque 03

Se entregará tanto el script como el fichero generado por el programa. El programa debe hacer uso del encabezado definido en la asignatura y respetar las normas de estilo establecidas.

Práctica - Ejercicio 19: Realizar un programa (fgECAD.m) que considere el fichero FG_STAID000036.txt, que contiene las observaciones de velocidad de viento en una estación europea, y escriba un nuevo fichero (FG_STAID000036.csv) el cual contenga una única línea de cabecera describiendo el formato y los datos incluidos, y los valores incluidos en el fichero original para todos los campos contenidos en el fichero (STAID, SQUID, DATE, FG, Q_FG) para todos los tiempos, separando éstos por el símbolo '|'. El programa incluirá dos nuevas filas con los valores promedios de los días identificados como válidos (validMean, validMean, 2008-2017, Valor Medio obtenido, 0) y otro con los no válidos (novalidMean, novalidMean, 2008-2017, Valor Medio obtenido, 9). Además, el programa dibujará la serie temporal de la velocidad de viento y mostrará por pantalla la velocidad media en todo el periodo, así como la velocidad máxima registrada en el periodo observado.

Práctica - Ejercicio 20: Realizar las siguientes funciones:

- notaMedia.m: que reciba un vector de notas y devuelva las notas en formato texto (suspense: $x < 5$; aprobado: $5 \leq x < 7$; notable: $7 \leq x < 8.5$ y sobresaliente: $x \geq 8.5$) y numérico.
- ecCuadratica.m: que reciba un vector con los coeficientes de una ecuación cuadrática ($a \cdot x^2 + b \cdot x + c = 0$), la clasifique en función del discriminante y devuelva las soluciones y la clasificación. Las soluciones complejas se darán en función de su parte real e imaginaria. La función debe comprobar que el vector tiene 3 elementos, uno para cada coeficiente de la ecuación, y salir con un error en caso contrario.
- horaDia.m: que reciba un número entre 0 y 24 con la hora y devuelva si es por la mañana ($6 < \text{hora} \leq 12$), por la tarde ($12 < \text{hora} \leq 20$) o de noche ($20 < \text{hora} \leq 6$). La función debe comprobar que el valor introducido está en el rango adecuado y salir con un error en caso contrario.
- miFactorial.m: que reciba un número natural positivo y devuelva su factorial. La función debe comprobar que se cumplen ambas condiciones y salir con un error en caso contrario.
- esPrimo.m: que reciba un número natural positivo y devuelva un 0/1 en el caso de que no sea/sea primo. La función debe comprobar que se cumplen ambas condiciones y salir con un error en caso contrario.
- bisectriz.m: que reciba un vector definiendo un intervalo y un número real positivo definiendo una tolerancia y devuelva la raíz del polinomio $x^3 - 3x^2 - 2x + 6$ en ese intervalo con dicha tolerancia. En el caso de no existir raíz en el intervalo que devuelva un NaN
- Escribir una función, llamada miTriangulo a la cual le pasemos la longitud de los 3 lados y devuelva el área y perímetro del triangulo usando la formula de Heron
- Un vector puede venir dado por sus coordenadas (x,y) en el plano. Escribe un programa que pida al usuario las coordenadas de un vector y que muestre su modulo y dirección.
- Crea la función binomial(i,j) definida por:
- Crear una función que calcule la exponencial de un número utilizando la suma de los m primeros términos de la serie de Maclaurin. Los argumentos de entrada de esta función son el número x y el número de términos de la serie. Realizar un script que pida ambas variables, calcule la aproximación y muestre el error cometido comparándolo con la función exp()

Práctica - Ejercicio 21: Considerar los archivos de precipitación incluidos en la carpeta de Moodle, realizar una función que lea ese formato, un programa que obtenga el total y una función que reproduzca el fichero de totales.

Práctica - Ejercicio 22: Los 2 ficheros de ejemplo recogen datos reales de producción de electricidad obtenidos de la web del Ministerio de Energía, Turismo y Agenda Digital. El fichero original se ha modificado para implementar 4 formatos que recogen gran parte de la casuística asociada a la lectura de ficheros.

Realizar una función para leer cada tipo de fichero (p.e. T_125CA_0114_csv.m en el caso de la función que lee el formato CSV) y un programa (fileComparison.m) que compare los datos leídos de cada fichero aplicando la función correspondiente, viendo, por ejemplo, las diferencias entre los datos leídos de uno y

Fundamentos de Computación

Enunciados de Prácticas – Bloque 03

otro fichero. El nombre de cada función seguirá la siguiente estructura: T_125CA_0114_ Format.m, siendo Format la extensión del fichero (csv y txt).