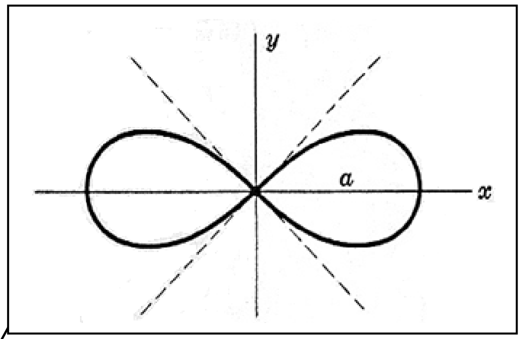
Fundamentos de Programación (13 de septiembre de 2018)
(Grados en Ingeniería Mecánica, Eléctrica, Electrónica Industrial y Química Industrial)

Ejercicio 1 – Programación Estructurada (2 puntos): construir un programa en ***C*** que calcule e imprima en pantalla el autovector (o autovectores) asociado(s) a un autovalor, dados por teclado los coeficientes del sistema lineal homogéneo de ecuaciones a resolver:

$$\boldsymbol{a_{11} * x + a_{12} * y + a_{13} * z = 0}$$
$$\boldsymbol{a_{21} * x + a_{22} * y + a_{23} * z = 0}$$
$$\boldsymbol{a_{31} * x + a_{32} * y + a_{33} * z = 0}$$

El sistema debe de ser compatible indeterminado y los autovectores son soluciones no nulas, linealmente independientes. <u>Nota</u>: para determinación de autovectores analizar rango de la matriz de coeficientes ***M***:

Rango=3	***Determinante(M)≠0*** → Sistema compatible determinado. La única solución es ***(0,0,0)*** que no es un autovector → el sistema lineal de ecuaciones no se corresponde con ningún autovalor.
Rango=2	$\boldsymbol{a_{ij}*a_{kl}-a_{il}*a_{kj}\neq 0}$ →Hay dos ecuaciones linealmente independientes (i y k): ***a*x+b*y+c*z=0*** ***d*x+e*y+f*z=0*** Las variables de las columnas j y l se pueden expresar en función de la tercera variable. Hay un autovector asociado al autovalor que se puede calcular asignando el valor 1 a la tercera variable, calculando los valores de las otras variables, y normalizando el vector resultante. Ej: ***j=1, l=2*** → ***x=(-c*e+b*f)/(a*e-b*d)*** ***y=(-a*f+c*d)/(a*e-b*d)*** ***z=1*** $\boldsymbol{norma=(x*x+y*y+z*z)^{1/2}}$ → ***x=x/norma*** ***y=y/norma*** ***z=z/norma***
Rango=1	$\boldsymbol{a_{ij}\neq 0}$ → Solo hay una ecuación linealmente independiente (i): ***a*x+b*y+c*z=0*** La variable j se puede expresar en función de las otras dos. Hay dos autovectores asociados al autovalor, que se pueden calcular asignando alternativamente los valores 0 y 1 a las dos variables libres, calculando la tercera variable, y normalizando los dos vectores resultantes. Ej: ***j=1*** → ***(x,y,z)=(-b/a,1,0)*** ***(x,y,z)=(-c/a,0,1)*** $\boldsymbol{norma=(x*x+y*y+z*z)^{1/2}}$ → ***x=x/norma*** ***y=y/norma*** ***z=z/norma***
Rango=0	$\boldsymbol{a_{ij}\neq 0}$ → Hay tres autovectores asociados al autovalor: ***(1,0,0)***, ***(0,1,0)*** y ***(0,0,1)***.



Ejercicio 2 – Diseño Modular (2 puntos): 2.1) construir una función en ***C*** que calcule y devuelva el valor absoluto de la función conocida como lemniscata (curva usada en el diseño de peraltes de carreteras) para un valor de ***x***, dados como parámetros el valor del radio ***a*** y el de ***x***, adecuadamente validados (***a>0*** y ***x:[-a,+a]***).
Ecuación paramétrica de la lemniscata:

$$\boldsymbol{r^2(\theta) = a^2 * cos(2 * \theta)}$$

Considerar el siguiente prototipo para dicha función:

```
/* Prototipo de función a construir en ejercicio */
     double lemniscata(double a, double x);
```

2.2) Construir un programa que, haciendo uso de la función anterior, calcule e imprima en pantalla todos los puntos de la trayectoria empezando y terminando en el punto ***(a,0)***, y con un incremento de la variable independiente ***x*** de ***±a/100***, donde el valor de ***a*** es introducido por teclado y adecuadamente validado

Ejercicio 3 – Estructuras de Datos (2 puntos): construir una función que calcule aproximadamente el volumen encerrado por una pieza maciza sin oquedades a partir de tres imágenes digitales de sus contornos obtenidas mediante las tres proyecciones ortogonales de la misma en los planos XY, XZ e YZ, y que determine además las dimensiones de la caja (ortoedro) más pequeña donde se puede guardar dicha pieza. La imagen digital de cada uno de los contornos consta de una retícula de celdillas (píxeles) de datos binarios (1: contorno, 0: vacío). La resolución de cada pixel (largo y ancho) es de 1 cm. El volumen aproximado de la pieza se calculará mediante la intersección de los prismas rectos generados por extrusión (estiramiento) de los contornos en la dirección perpendicular al plano sobre el que están definidos.

```
     /* Estructuras de datos a considerar en el ejercicio */
          #define MAX 100  /* Dimensión máxima: 1 m (100 píxeles de 1cm) */
          typedef double tipo_matriz[MAX][MAX];
          typedef struct{
               tipo_matriz mxy,mxz,myz;
               int nx,ny,nz;    // Longitud en píxeles de cada dirección
          }tipo_proy;
```

```
/* Prototipo de función a construir en el ejercicio */
      double volumen(tipo_proy *p, double *dx, double *dy, double *dz);
```

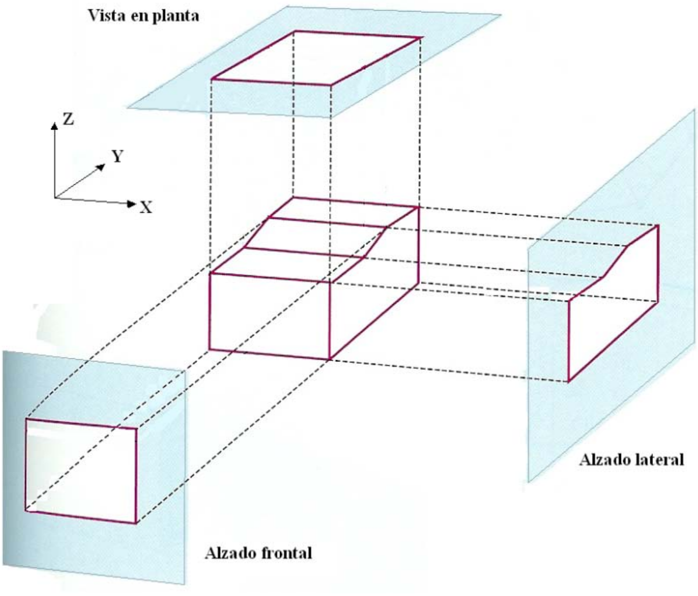
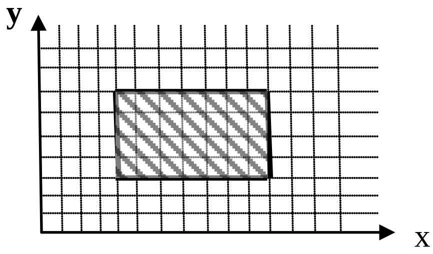


Imagen digital vista en planta



....
00000000000000...
00000000000000...
00001111111000...
00001111111000...
00001111111000...
00001111111000...
00000000000000...
00000000000000...
00000000000000...

Ejercicio 4 – Archivos (2 puntos): construir un programa que calcule un modelo de regresión lineal $\boldsymbol{Y_i=\beta_0+\beta_1 * x_i+u_i}$ entre dos variables $\boldsymbol{Y}$ (var. respuesta) y $\boldsymbol{X}$ (var. explicativa o independiente) a partir de un conjunto no acotado de parejas de datos $\boldsymbol{(x_1,y_1),\ldots,(x_i,y_i),\ldots(x_n,y_n)}$ almacenadas en el archivo de texto ***"datos.txt"***, y que evalúe a continuación las hipótesis del modelo (normalidad, linealidad y homocedasticidad) mediante un análisis de los residuos. El programa leerá inicialmente los datos del archivo de texto y simultáneamente calculará los estadísticos descriptivos básicos de las dos variables involucradas para poder estimar los parámetros del modelo:

Medias muestrales de X e Y: $\boldsymbol{\overline{x} = \frac{\sum_{i=1}^{n} x_i}{n}}$ $\boldsymbol{\overline{y} = \frac{\sum_{i=1}^{n} y_i}{n}}$

Varianzas muestrales de X e Y: $\boldsymbol{v_x = \frac{\sum_{i=1}^{n}(x_i-\overline{x})^2}{n} = \frac{\sum_{i=1}^{n} x_i^2 - n*\overline{x}^2}{n}}$ $\boldsymbol{v_y = \frac{\sum_{i=1}^{n}(y_i-\overline{y})^2}{n} = \frac{\sum_{i=1}^{n} y_i^2 - n*\overline{y}^2}{n}}$

Covarianza muestral entre X e Y: $\boldsymbol{cov_{x,y} = \frac{\sum_{i=1}^{n}(x_i-\overline{x})*(y_i-\overline{y})}{n} = \frac{\sum_{i=1}^{n} x_i*y_i - n*\overline{x}*\overline{y}}{n}}$

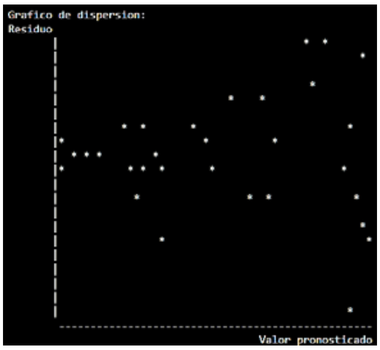
Seguidamente calculará por máxima verosimilitud la estimación puntual de los parámetros del modelo (coeficientes de la recta de regresión) junto con los coeficientes de evaluación del ajuste (Error Cuadrático Medio y Coeficiente de Correlación Lineal), presentando los resultados en pantalla:

$$\boldsymbol{\hat{\beta}_0 = \overline{y} - \frac{cov_{x,y}}{v_x}\overline{x}} \qquad \boldsymbol{\hat{\beta}_1 = \frac{cov_{x,y}}{v_x}} \qquad \boldsymbol{E.C.M. = v_y - \frac{(cov_{x,y})^2}{v_x}} \qquad \boldsymbol{r = \frac{cov_{x,y}}{\sqrt{v_x * v_y}}}$$

Finalmente realizará el diagnóstico de las hipótesis del modelo mediante la presentación en pantalla del histograma de los residuos (para la hipótesis de normalidad de los datos el histograma se parece a una curva normal) y del gráfico de dispersión de los residuos frente a los valores pronosticados (la hipótesis de homocedasticidad o de igualdad de varianzas e cumple si la anchura vertical del gráfico se mantiene razonablemente constante, y la hipótesis de linealidad de los datos cuando la línea central del gráfico sea razonablemente recta). Notas:

Valor pronosticado: $\boldsymbol{\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 * x_i}$

Residuo de cada dato: $\boldsymbol{u_i = y_i - \hat{y}_i}$



Histograma de residuos → considerar 20 intervalos de clase equiespaciados entre los valores mínimo y máximo y presentar el histograma en vertical con asteriscos escalando la frecuencia absoluta más alta a 50.

Gráfico de dispersión → dividirlo en un rectángulo de cuadrículas de 20 de alto y 50 de ancho y presentar un asterisco si hay al menos un dato en dicha cuadrícula.