**Salida de Archivos**

Este programa tendrá un programa principal, una subrutina y una función, que se encargaran de producir un archivo en donde se evalúen los valores de x & y, que pertenecen a una función de dos variables.

Programa Principal

Este solo se encargará de pedirle al usuario los intervalos de cada variable y la cantidad de puntos que se desean evaluar en x & y. Además, le pedirá al usuario el nombre del archivo que se va a producir.

También se calculara “h” para cada variable, usando los intervalos.

Se declaran las siguientes variables:

* REAL :: a, b, c, d, hx, hy

a y b son los intervalos de x, mientras que c y d son los intervalos en y

* INTEGER :: n, m

n: puntos a evaluar en x

m: puntos a evaluar en y

* INTEGER :: num\_error

variable para IOSTAT, nos dirá si hay algun error en el OPEN

* CHARACTER :: nombre\_archivo, mensaje\_error

nos mostrará un mensaje indicandonos en donde fue que se cometió un error(IOMSG)

variable que almacenará el nombre del archivo que el usuario introduzca

Las características de nuestro archivo serán los siguientes (se escribirán en un open())

* UNIT=80

unidad del archivo

* FILE=nombre\_archivo

nombre del archivo que se introducirá

* STATUS=”UNKNOWN”

el estado del archivo es desconocido, puede ser un archivo ya existente o uno que no se ha creado

* ACTION=”WRITE”

la acción que se realizará en el archivo de texto, se escribirá en el nuevo archivo.

* IOSTAT=num\_error

un número que estará asociado a un error relacionado con el OPEN

* IOMSG=mensaje\_error

nos mostrará un mensaje en donde se nos indique en donde se encuentra el error que se cometió

Por si se detecta un error introducimos un IF en donde la expresión lógica será (num\_error == 0)

SI se cumple la expresión lógica se le llamara a la subrutina, que es la que se encarga de evaluar los puntos.

Si no se cumple se mostrará un mensaje de error.

**Subrutina**

Aquí se declaran las siguientes variables dummy, que son aquellas que también se declararon en el principal y se usarán en la subrutina. (a las variables usadas se les agrego una o, ya que no deben de ser exactamente iguales a las que fueron usadas en el principal)

REAL, INTENT(IN) :: ao, co, hxo, hyo

INTEGER, INTENT(IN) :: no, mo

Ahora se declaran las variables locales (usadas solo en la subrutina)

REAL :: x, y, z

INTEGER :: i, j

REAL, EXTERNAL :: f

esta variable se usará en la función

Ahora se usará un DO iterativo, en donde i=0 y el valor inicial será el valor de no. Esto nos servirá a evaluar los valores de x no veces

Se escribirá lo siguiente.

x = ao + i\*hxo , no veces

Habrá otro DO iterativo dentro de este, será igual, pero ahora sera para evaluar los valores de y mo veces.

y = co + j\*hyo

z = f(x, y)

se evalúan los dos valores usando la función usada, viene desde la función

WRITE(80,\*) x, y, z

Esto hará que todos los valores obtenidos se escriban en el archivo. Tendremos tres columnas; una para los valores de x, una para los valores de y, y una para los dos valores evaluados en la función

Función

Aquí solo se escribirá la función que queremos evaluar, que será usada para evaluar los valores obtenidos.

Las variables a usar serán

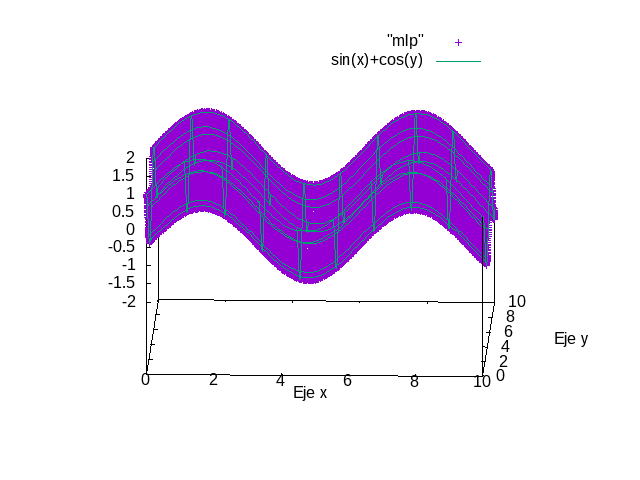
REAL, INTENT(IN) :: xo, yo

variables dummy que vienen desde el programa principal

REAL :: f

f= SIN(xo) + COS(yo)

función utilizada



**Entrada de Archivos**

Este programa se encargará de utilizar un archivo con datos ya existentes para encontrar la ecuacion de una recta usando el metodo de minimos cuadrados.

Aquí solo se usará un programa principal y una subrutina.

Principal

Como en el programa principal del trabajo anterior, aquí se le pedirá al usuario que solo introduzca el nombre del archivo con los datos.

Variables declaradas:

INTEGER :: num\_error

CHARACTER :: nombre\_archivo, mensaje\_error

Las características del archivo serán los siguientes

* UNIT=90
* FILE= nombre\_archivo
* STATUS=”OLD”

El archivo que tiene los datos ya existe

* ACTION=”READ”

El programa solo leerá el archivo de datos

* IOSTAT=num\_error
* IOMSG=mensaje\_error

COmo en el anterior, se utilizará un IF en donde se le llama a la subrutina si se cumple la expresión lógica (num\_error==0), y mostrará un mensaje de error si no se cumple.

Subrutina

No hay variables dummy así que solo se pone ( ) después del nombre de la subrutina

Las variables declaradas son las siguientes

REAL :: x, y, promy, promx

REAL :: sumxy, sumxx, sumx, sumy

REAL :: m, b

CHARACTER :: mensaje\_error

INTEGER :: num\_error, n

Se usarán las siguientes ecuaciones





El programa leerá las variables x & y del archivo y checara si hay errores.

Se agrega un IF, en donde el programa encuentre num\_error /= 0 dejará de repetirse el ciclo, mientras num\_error=0 se llevarán a cabo estas sumatorias.

sumx = sumx+x

sumxx = sumxx+x\*x

sumxy = sumxy+x\*y

sumy = sumy + y

n=n+1

Ahora sacaremos los siguientes promedios

promx = sumx/n

promy = sumy/n

y ahora se usarán las ecuaciones ya mencionadas para obtener el valor de la pendiente y el valor de la ordenada en el origen.

m = (sumxy-sumx\*promy)/(sumxx-sumx\*promx)



b = promy - m\*promx



