



Estructuras de control

Sentencia DO WHILE()

La sentencia DO WHILE(*expresión lógica*) tiene forma de un DO que se realiza mientras la *expresión lógica* sea verdadera

```
DO WHILE (expresión lógica)  
  sentencia 1  
  sentencia 2  
  sentencia 3  
  sentencia 4  
ENDDO
```

Ventajas

Es muy útil cuando la pregunta que debe hacerse es clara en el problema. Por ejemplo, en una serie de Taylor.

Cuando los incrementos (o decrementos) de un bucle o loop no son aditivos. Es decir tengo un paso variable.

Cuidados

El bucle tiene que terminar, por lo cual debo cuidar de poner el paso adecuado para que esto suceda, de lo contrario quedaría corriendo hasta el infinito.

Ejemplo: $i=i+1$

Esto no sucede en el DO común donde queda muy claro el paso.

Ejemplos

Ecuación de Kepler:

$$M = E - e \sin(E)$$

Conociendo M y e tengo que encontrar el valor de E que satisfaga la ecuación.

Se resuelve por ITERACIÓN de la siguiente manera:

- Se asigna un valor de E , al cual lo llamo E_0 .
- Con este valor calculo un nuevo valor de E , que lo llamo E_1 .
Haciendo $E_1 = M + e \sin(E_0)$
- Utilizo como nuevo E_0 al valor E_1
- Para tener una cota del error utilizo $|E_1 - E_0|$ y esto lo hago porque supongo que la repetición de una y otra vez del algoritmo convergerá a valor que resuelve la ecuación. Por lo cual este valor se va reduciendo en cada paso.

Ejemplos

Ecuación de Kepler:

$$M = E - e \sin(E)$$

```
Program kepler
```

```
real*8 M,E,exc
```

```
PI=3.1415926
```

```
write(*,*) 'Ingrese el valor de la excentricidad'
```

```
read(*,*) exc
```

```
write(*,*) 'Ingrese el valor de la Anomalía Media'
```

```
read(*,*) M
```

```
M = M/180*PI
```

```
E0=0
```

```
E1=1
```

```
DO WHILE(abs(E1-E0).gt.1e-8)
```

```
    E0 = E1
```

```
    E1 = M + exc * sin(E0)
```

```
    write(*,*) E0,E1,E0-E1
```

```
ENDDO
```

```
write(*,*) 'La Anomalía excéntrica es:',E1/pi*180,' Grados'
```

```
end
```

Otro ejemplo que ya vimos: la Serie de Taylor

do while (error.gt.valor)

 Calculo un término más de la serie y lo sumo

 Calculo el error (o una cota o una aproximación)

enddo

Donde valor puede ser un número muy pequeño,
Como $10e-8$ que me asegura 7 (o más!) decimales
correctos.

Pero ojo -> puede tardar mucho si necesito calcular
muchos términos.

Vemos como calcular la serie de Taylor del coseno utilizando esta sentencia.

$$\text{Cos}(x) = \sum_{i=1}^{\infty} \frac{(-1)^i x^{2i}}{(2i)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots + |\text{error}(\xi)|$$

Con la idea de indicarle al programa la precisión del resultado, ya que el error se lo podríamos dar como entrada o fijarlo en un valor que nos interese.

Fijamos el error en un ejemplo como $\text{error}=1\text{E}-08$

