

Computación



Funciones

Ya sabemos que Fortran tiene funciones intrínsecas pre-programadas para que el usuario las utilice:
Ejemplo: $\cos(x)$, \sqrt{x} , $\log(x)$

Y ahora veremos que hay formas de agregar funciones a un programa que nosotros construiremos.

¿Cómo las creamos ?

Hay dos maneras:

1) Función sentencia

2) Función externa

Todos los lenguajes de programación tienen funciones, aunque se las encuentra con distintos nombres:

Procedimientos, subprogramas, función lambda, etc.

Ventajas:

Si uso un cálculo muy seguido y este tiene se puede escribir como función, ahorro mucho tiempo y mi programa es mucho más compacto.

$$\sum_{i=1}^n 1/i$$

Puedo utilizar funciones ya programadas y que están en libros o en internet. No tengo que volver a inventar la pólvora.

Estas colecciones de funciones cumplen con dos requisitos:

- 1) Se sabe que no tienen errores porque están muy probadas
- 2) Son muy rápidas, a veces (o casi siempre!!!) no suelen utilizar los algoritmos convencionales.

Función Sentencia

Consiste en una sola sentencia aritmética definido al principio del programa.

Funcion(var1,var2,var3,..)=expr_mat(var1,var2,va3)

Ejemplos:

$\text{DISC}(A,B,C) = B^{**}2 - 4.*A*C$

$\text{Area}(\text{base}, \text{altura}) = \text{base} * \text{altura} / 2.$

Detalles a tener en cuenta:

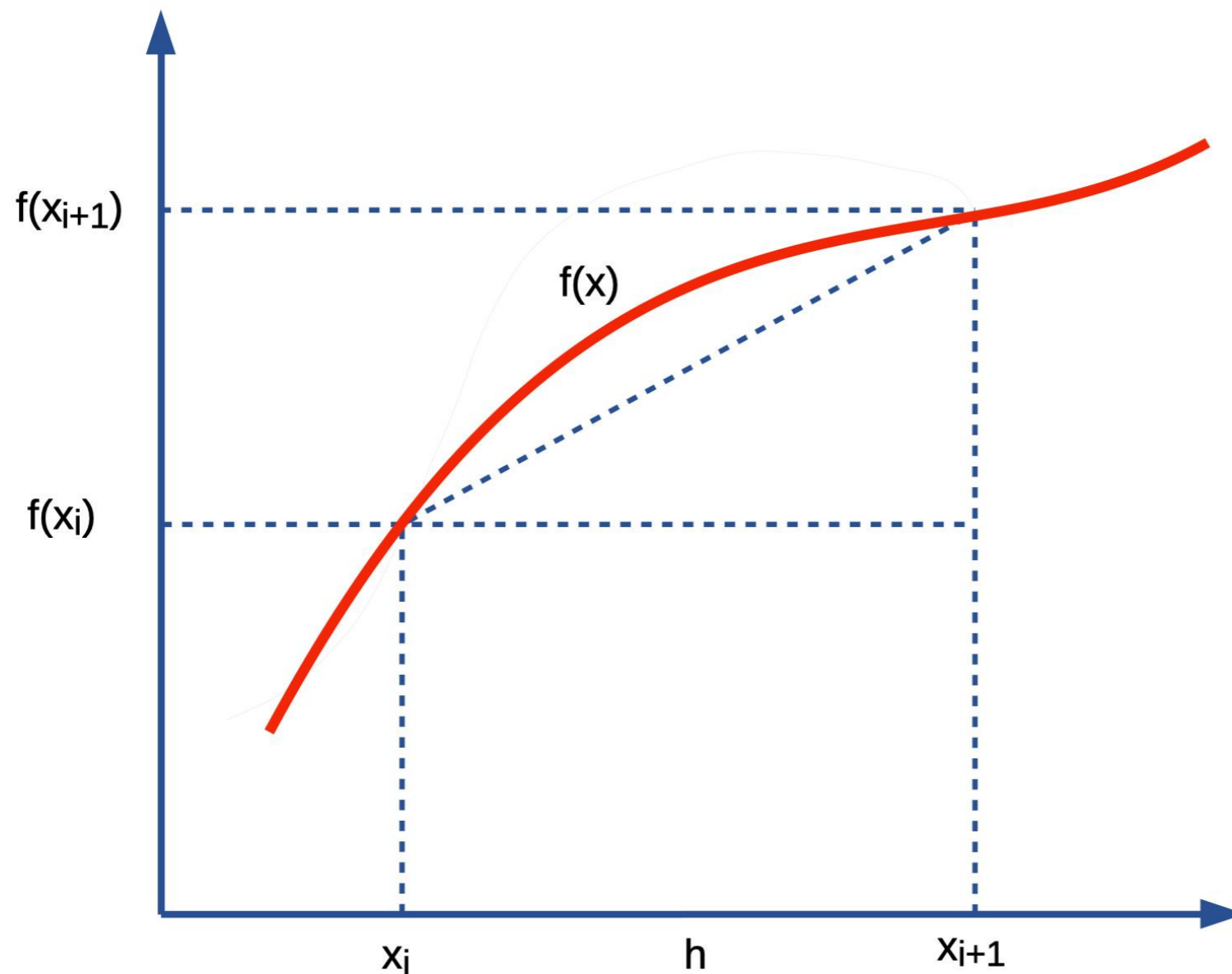
- 1) Las variables (`var1`, `var2`, ...) pueden ser de cualquier tipo, incluyendo lógicas y de carácter.
- 2) Para el programa el resultado del cálculo de una función es otra variable, por esta razón las funciones pueden ser de todos los tipos de variables posibles. Ejemplo: una función puede ser: entera, `real*4`, `real*8`, compleja e incluso de carácter. En estos últimos casos hay que indicar en la definición de la función su tipo:

Complex `fs3(x,y,x)=x*y+z`

Entonces `fs3` es una función cuyo resultado de cálculo es un número complejo y para el programa `fs3` es una variable que guarda números complejos.

Integral por trapecios

$$A_i = hf(x_i) + \frac{h}{2}[f(x_{i+1}) - f(x_i)] = \frac{h}{2}[f(x_{i+1}) + f(x_i)]$$



Integral por trapecios

Y sumando todos los trapecios, obtenemos el valor de la integral

$$Integral = \sum_{i=0}^n A_i$$

Y con un poco de álgebra simple:

$$Integral = \frac{h}{2}[f(a) + f(b)] + h \sum_{i=1}^{n-1} f(x_i)$$

Hagamos un programa que integre $f(x)=x^2$

Recordando que :

$$\int_0^1 x^2 dx = 1/3$$

Si corremos el programa obtenemos:

Cantidad de intervalos	Resultado de la aproximación de la integral
10	0.33500001696869747
100	0.33334997741140865
1,000	0.33333354714617608
10,000	0.33333330969899705
100,000	0.33333330800677946
1,000,000	0.33333333080419814
10,000,000	0.33333334502431866
100,000,000	0.33333332725965703
1,000,000,000	0.33333330505514902

Si bien nunca obtenemos un resultado perfecto, si podemos conseguir una cantidad de decimales muy importante y por lo tanto para la vida real resolver el problema.

Función externa

- La función externa permite escribir más de una renglón en su definición.
- Ya no está en el programa sino fuera de este aunque el programa la puedo llamar para su uso
- Veamos la forma genérica:

```
program Principal
```

```
real*4 F
```

```
:
```

```
A= F(x)+5.0
```

```
:
```

```
END
```

```
Function F(x)
```

```
:
```

```
F = ...
```

```
RETURN
```

```
END
```

Propiedades de la funciones externas

Puede haber muchas funciones que son utilizadas por un programa.

Las funciones tienen variables internas que guardan datos que son de la función.

La orden RETURN devuelve la ejecución al lugar exacto donde la función fue llamada.

Las funciones externas, al igual que la función sentencia son para el programa principal variables y cumplen con las reglas de estas.

La función cuando es llamada depende para realizar su cálculo de las variables de referencia que se encuentran dentro del paréntesis. A estas variables se las denomina argumentos de la función. Los nombres de estos argumentos en el programa que llama a la función, pueden o no coincidir con los que se usan en la función para hacer los cálculos.

programa principal

REAL*4 A(100)

Complex C

⋮

Z = F(A,C)

⋮

END

Function F(X2,B)

REAL*4 X2(100)

COMPLEX B

⋮

RETURN

END

Notar que X2 es un vector de 100 elementos y B una variable compleja

Function Serie(n,k)

Serie=0.

DO i=1, n

x = i

Serie = Serie + 1/ xk**

ENDDO

RETURN

end