

C 言語 と Fortran の構文比較

関数、サブルーチン

C 言語

関数の呼び出し

```
m = myfunc(n);  
i = mysub(&a, b, c);
```

&a は **a** のアドレス
b, c は値を表わす

関数の定義

```
型名 関数名 (型名 引数){  
処理  
}
```

```
int myfunc(int i){  
    return i+1;  
}
```

```
int mysub(double *x, double y, double z){  
    *x = y + z;  
    return 0;  
}
```

***x** はアドレス **x** の値を表す
y, z は局所変数で return 後に消滅

a のアドレスを渡すと **a** と **x** は同じの実体をもつ。
b, c のように値を渡すと実体は異なったものになる。
上の例では **b** と **y** は呼び出し時に値だけがコピーされ、
その後 **y** が変化したとしても **b** には反映されない。

Fortran

関数呼び出し

```
integer m, n  
m = myfunc(n)
```

引数、返却値の型に注意

関数の定義

```
型名 function 関数名 (引数)  
引数の定義  
処理  
return  
end
```

```
integer function myfunc( i )  
integer i  
myfunc = i+1  
return  
end
```

サブルーチンの呼び出し

```
real*8 a, b, c  
call mysub(a,b,c)
```

引数の型に注意

サブルーチンの定義

```
subroutine サブルーチン名 (引数)  
引数の定義  
処理  
return  
end
```

```
subroutine mysub(x, y, z)  
real*8 x, y, z  
x = y + z  
return  
end
```

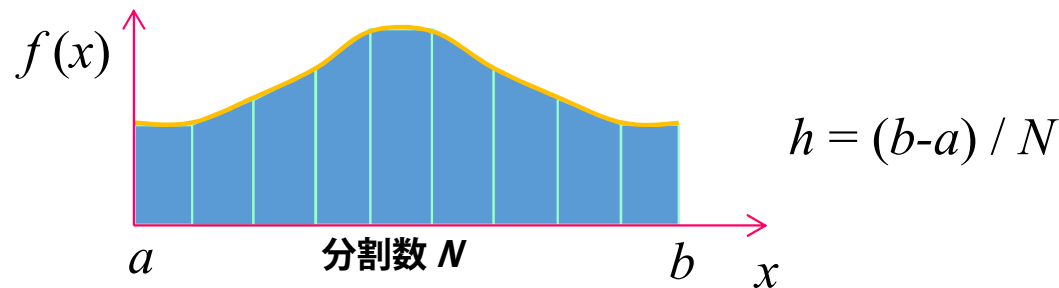
Fortran における引数の実体は、変数のアドレスで、
例えば引数が配列の場合、先頭の要素へのポインタになっている

積分計算 (関数を利用するプログラムの例)

$$I = \int_a^b f(x) dx$$

台形公式

$$I = h \left[\frac{1}{2} f(a) + \sum_{n=1}^{N-1} f(a + nh) + \frac{1}{2} f(b) \right]$$



分割数 N を多くしたときの結果の変化から収束判定を行うことで、最適な N を自動的に決めることができる。

```
real*8 :: s, h, f, PI
integer :: i, N=100
```

```
PI=4.d0*atan(1.d0)
```

```
h=PI/2/N
```

```
s=(f(0.d0)+f(PI/2))/2
```

```
do i=1, N-1
```

```
    s=s+f(i*h)
```

```
end do
```

```
write(*,*) s*h
```

```
end
```

$$I = \int_0^{\pi/2} \cos(x) dx$$

```
real*8 function f(x)
real*8 :: x
```

```
f =cos(x)
```

```
return
```

```
end
```

ただし、分割数 N を多くしたときの（情報落ち）誤差に注意

積分区間が無限の場合には、積分範囲を広げながら収束を判定するか、変数変換によって有限区間の積分に直して実行する。