

シミュレーション物理5

運動方程式の方法:
サブルーチンの使い方

サブルーチン(subroutine)

- 多くのプログラムは, 小さい部分に分割出来る
- 例えば運動方程式の方法の場合;
 - 初期値を決める
 - 時間発展を行う
 - 結果を書く
- このとき, 時間発展の部分はいろいろなプログラムで同じ構造をしている; 時間 t の状態を運動方程式に従って $t+\delta t$ にするだけ。

サブルーチンを使う理由

- スペースの節約
- デバッグがしやすい→

一度書いて間違いがないことを確かめたらもうそのまま直さないでよい

関数とサブルーチン

- 関数: $\exp(x)$, $\sin(x)$ のように変数を与えてある手続きで別の数を出すもの。Fortran90を始め, ほとんどのプログラミング言語では, 我々が自分で関数を作ることが出来る。
- サブルーチン: 変数に対してある手続きを行い, 別の変数を作るプロセス。変数は複数でもよい。

関数プログラムの例

ここでは華氏から摂氏の変換プログラムを扱う。

$$\text{摂氏} = (\text{華氏} - 32) * 5/9$$

(私は華氏68度＝摂氏20度として、それからのずれを $5/9 \doteq 0.5$ して、計算している。例えばニュースで華氏80度と言われたら $(80-32)*5/9=26.67$ をやらず、 $(80-68)/2+20=26$ と概算してる。)

```

program fahrenheit2celsius
!-----
! This is a program to convert F to C
! 2005/5/16 Written by T. Ohtsuki
!-----
implicit none ! Always begin with this statement
integer,parameter::double=selected_real_kind(14)
real(kind=double):: fahrenheit,celsius

!-----main program-----
print *, "Input temperature in Fahrenheit."
read *, fahrenheit

celsius=f2c(fahrenheit)

print *, "It is ",celsius,"degree in celsius"

```

これをfahrenheit2celsius.f90としてsave

contains

```

!----- Subroutines -----

function f2c(temperature)
!-----
! function to convert fahrenheit to celsius
!-----
IMPLICIT NONE
integer,parameter::double=selected_real_kind(14)
REAL(kind=double), INTENT(IN) :: temperature
REAL(kind=double):: f2c

f2c=(temperature-32._double)/1.8_double

END function f2c

end program fahrenheit2celsius

```

別のやり方: ファイルを分ける

これはfahrenheit2celsius2.f90としてsave

```
program fahrenheit2celsius
!-----
! This is a program to convert F to C
! 2005/5/16 Written by T. Ohtsuki
!-----
implicit none ! Always begin with this statement
integer,parameter::double=selected_real_kind(14)
real(kind=double):: fahrenheit,celsius
```

```
interface
  function f2c(temperature)
    integer,parameter::double=selected_real_kind(14)
    real(kind=double)::f2c
    real(kind=double),intent(IN)::temperature
  end function f2c
end interface
```

```
!-----main program-----
print *, "Input temperature in Fahrenheit."
read *, fahrenheit
celsius=f2c(fahrenheit)

print *, "It is ",celsius,"degree in celsius"

end program fahrenheit2celsius
```

このプログラムではf2cの
変数の型がわからないので
このinterfaceで判断させる

関数を独立させる

これはf2c.f90としてsave

```
function f2c(temperature)
!-----
! function to convert fahrenheit to celsius
!-----
IMPLICIT NONE
integer,parameter::double=selected_real_kind(14)
REAL(kind=double), INTENT(IN) :: temperature
REAL(kind=double):: f2c !interfaceを使わない例

f2c=(temperature-32._double)/1.8_double

END function f2c
```

前のをfahrenheit2celsius2.f90, これをf2c.f90と名付ける。
f90 fahrenheit2celsius2.f90 f2c.f90で実行ファイルを作る。

関数のモジュール化

- 似たような関数をまとめたい
 - モジュール化
 - 例えば華氏→摂氏の逆で摂氏→華氏を行う関数も一つにしておきたい場合

```
program fahrenheit2celsius
!-----
! This is a program to convert F to C
! 2005/5/16 Written by T. Ohtsuki
!-----
use temperaturelibs
implicit none ! Always begin with this statement
integer,parameter::double=selected_real_kind(14)
real(kind=double):: fahrenheit,celsius

!-----main program-----
print *, "Input temperature in Fahrenheit."
read *, fahrenheit

celsius=f2c(fahrenheit)

print *, "It is ",celsius,"degree in celsius"

print *, "Input temperature in Celsius."
read *, celsius
fahrenheit=c2f(celsius)

print *, "It is ",fahrenheit,"degree in fahrenheit."

end program fahrenheit2celsius
```

モジュールとはこんなもの

```
module temperaturelibs  
implicit none  
contains
```

```
function f2c(temperature)  
!-----  
! function to convert fahrenheit to celsius  
!-----  
IMPLICIT NONE  
integer,parameter::double=selected_real_kind(14)  
REAL(kind=double), INTENT(IN) :: temperature  
REAL(kind=double):: f2c  
  
f2c=(temperature-32._double)/1.8_double  
  
END function f2c
```

```
function c2f(temperature)  
!-----  
! function to convert fahrenheit to celsius  
!-----  
IMPLICIT NONE  
integer,parameter::double=selected_real_kind(14)  
REAL(kind=double), INTENT(IN) :: temperature  
REAL(kind=double):: c2f  
  
c2f=temperature*1.8_double+32  
  
END function c2f
```

```
end module temperaturelibs
```

モジュールの形のプログラムのコンパイルの仕方

- まず使うモジュールプログラム(ここでは `temperaturelibs.f90`とする)を、あらかじめコンパイルしておく。
- `f90 -c temperaturelibs.f90` (`-c` はコンパイルだけやり、実行ファイルは作るなというオプション) すると、`temperaturelibs.o`, `temperaturelibs.mod`というファイルができる。

- 次に

`f90 fahrenheit2celsius3.f90 temperaturelibs.o`
というコマンドで実行ファイルを作る。

サブルーチンとは？

- 関数に似ている
- 関数よりも広く、**手続き**を行う。一群の変数をいれると一群の変数を返してくれる
- 一つの関数の値だけでなく、例えば行列を変数にすると、その固有値、固有ベクトルを返してくれるようなものがサブルーチン
- 具体的に、f2cをサブルーチンの形で書き直そう。

```
program fahrenheit2celsius
```

```
!-----
```

```
! This is a program to convert F to C
```

```
! 2005/5/16 Written by T. Ohtsuki
```

```
! 2005/5/17 subroutine version
```

```
!-----
```

```
implicit none ! Always begin with this statement
```

```
integer,parameter::double=selected_real_kind(14)
```

```
real(kind=double):: fahrenheit,celsius
```

```
!-----main program-----
```

```
print *, "Input temperature in Fahrenheit."
```

```
read *, fahrenheit
```

```
call f2c(fahrenheit,celsius)
```

```
print *, "It is ",celsius,"degree in celsius"
```

```
end program fahrenheit2celsius
```

これはfahrenheit2celsius4.f90としてsave

こちらはf2csubroutine.f90としてsave

```
subroutine f2c(fah,cel)
!-----
! function to convert fahrenheit to celsius
!-----
IMPLICIT NONE
integer,parameter::double=selected_real_kind(14)
REAL(kind=double), INTENT(IN) :: fah
REAL(kind=double),INTENT(OUT):: cel

cel=(fah-32._double)/1.8_double

END subroutine f2c
```

課題

- 中点法のプログラムをサブルーチン化する。
- 来週の授業までに出す。出来た人は、落下運動のプログラムにこれを適用し、**動作を確認して**、来週の出席メールに添付すること。