情報処理概論

第１０回　関数

情報基盤研究開発センター　谷本 輝夫

今回の内容

- 前回の課題 解答例
- 関数：台形積分

先週の課題プログラム例（1/3）

```
program Score_management
  implicit none
  integer :: i, j, total, number, kamoku
  integer, dimension(:,:), allocatable :: score
  integer, dimension(:), allocatable :: k_total
  intrinsic dble

  open(10, file='score.dat')
  read(10, *) number, kamoku

  allocate(score(number, kamoku))
  allocate(k_total(kamoku))

  ! Input score data
  do  i = 1, number
     read(10, *) score(i, 1:kamoku)
  end do

  close(10)
```

先週の課題プログラム例（2/3）

```
  ! Calculate total and average
    k_total = 0
  do  i = 1, number
     write(*,'(A3,I3,2X)', advance="NO") "No.",i
       total = 0
     do j = 1, kamoku
        total = total + score(i, j)
        write(*,'(I4,3X)', advance="NO") score(i,j)
        k_total(j) = k_total(j) + score(i,j)
     end do
     write(*,'(I4,3X)', advance="NO") total
     write(*,'(F6.1,1X)') total/dble(kamoku)
  end do

  write(*,'(A7,1X)', advance="NO") "Average"
  do j = 1, kamoku
     write(*,'(F6.1,1X)', advance="NO") k_total(j)/dble(number)
  end do
  write(*,*)
  stop
end program Score_management
```

先週の課題プログラム例（3/3）

```
$ cat score.dat
5 3
10 20 30
20 30 40
30 40 50
40 50 60
50 60 70
$ gfortran -o score score.f90
$ ./score
No.  1    10     20     30     60     20.0
No.  2    20     30     40     90     30.0
No.  3    30     40     50    120     40.0
No.  4    40     50     60    150     50.0
No.  5    50     60     70    180     60.0
Average   30.0   40.0   50.0
$
```

今回の内容

- 前回の課題 解答例
- 関数：台形積分

今日の予習プログラム（1/3）

```
program sample3
  implicit none
  integer :: number1, number2
  integer, dimension(:), allocatable :: english, math
  real(8), external :: average
  intrinsic dble

  open(10, file="eng.dat")
  read(10, *) number1
  allocate(english(number1))
  call input_data(10, number1, english)
  close(10)

  write(*, '(A20, F6.2)') "Ave. of English = ", &
                                    average(number1, english)

  open(11, file="math.dat")
  read(11, *) number2
  allocate(math(number2))
  call input_data(11, number2, math)
  close(11)
```

今日の予習プログラム（2/3）

```
  write(*, '(A20, F6.2)') "Ave. of Math = ",  &
                          average(number2, math)
stop
end program

subroutine input_data(file, n, a)
  implicit none
  integer, intent(IN) :: file
  integer, intent(IN) :: n
  integer, dimension(n), intent(OUT) :: a
  integer :: i

  do i = 1, n
    read(file, *) a(i)
  end do
end subroutine
```

今日の予習プログラム（3/3）

```
function average(n, a)
  implicit none
  integer, intent(IN) :: n
  integer, dimension(n), intent(IN) :: a
  real(8) :: average
  integer :: i, total
  intrinsic dble

  total = 0
  do i = 1, n
    total = total + a(i)
  end do

  average = dble(total) / dble(n)

end function
```

関数の定義

```
function 関数名(引数)
  implicit none
  変数,関数の宣言（引数やこの関数自身の宣言も行う）

  ... 計算 ...

  関数名 = 式
end function
```

- 主プログラムの外で function ～ end function により定義
- 関数自身も変数として宣言する
- 関数と同名の変数に格納された値が「返り値」として呼び出し側に返される.

関数定義の例

名前

引数

```
function average(n, a)
  implicit none
  integer, intent(IN) :: n
  integer, dimension(n), intent(IN) :: a
  real(8) :: average
  integer :: i, total
  intrinsic dble

  total = 0
  do i = 1, n
    total = total + a(i)
  end do

  average = dble(total) / dble(n)

end function
```

引数の宣言

関数（の返り値）の宣言

関数内で用いる変数や関数の宣言

返り値の計算，代入

関数の利用法

- 使用する関数の宣言

```
データ型名, external :: 関数名
```

- 関数の呼び出し
  - 通常の関数と同じ
  - 引数の順番に注意
  - 関数からさらに他の関数を呼び出しても良い
    - 呼び出し側の関数の中で，呼び出される側の関数の宣言が必要

関数利用の例

```
program sample3
  implicit none
  integer :: number1, number2
  integer, dimension(:), allocatable :: english, math
  real(8), external :: average
  intrinsic dble

  ...

  write(*, '(A20, F6.2)') "Ave. of Math = ",  &
                          average(number2, math)
stop
end program
```

使用する関数の宣言

関数の呼び出し

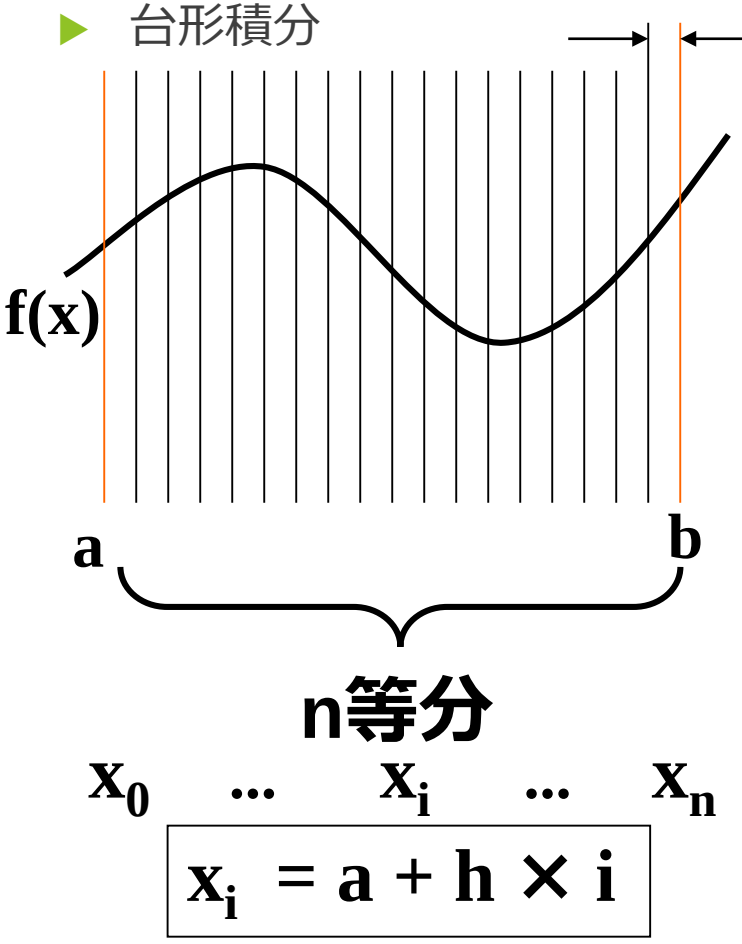
引数

intrinsic 関数と external 関数

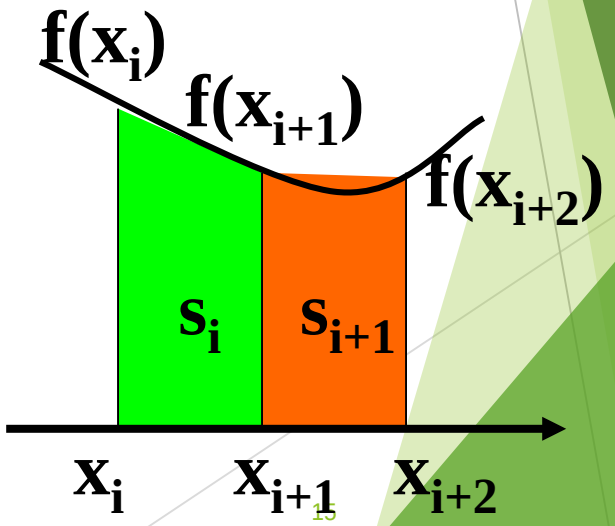
- intrinsic関数
  - Fortranの規格で定義されている組み込み関数
    - sin, cos, sqrt, dble 等
  - https://gcc.gnu.org/onlinedocs/gcc-3.4.6/g77/Table-of-Intrinsic-Functions.html
- external関数
  - プログラマが定義した関数

関数を使ったプログラム例：台形積分のプログラム

- 台形積分



区間 $[x_i, x_{i+1})$ の積分を台形で近似



台形積分

- 1区間の面積

$$s_i = (f(x_i)+f(x_{i+1})) \times h / 2$$

- 全区間の面積

$$\begin{aligned} S &= s_0 + s_1 + \dots + s_{n-1} \\ &= (h / 2) \times (f(x_0)+f(x_1)+f(x_1)+f(x_2)+f(x_2)+f(x_3) \\ &\qquad \dots +f(x_{n-1})+f(x_n)) \\ &= (h / 2) \times (f(x_0) + f(x_n) + 2 \times(f(x_1) + \dots + f(x_{n-1})) \\ &= (h / 2) \times (f(a) + f(b) + 2\times(f(x_1) + \dots + f(x_{n-1})) \end{aligned}$$

台形積分プログラム（1/3）

```
program trapezoid
  implicit none
  integer, parameter :: n = 1000
  integer :: i
  real(8) :: start, end, s1, s2, h
  real(8), external :: f, ff

  write(*, *) "Start point? "
  read(*, *) start
  write(*, *) "End point? "
  read(*, *) end

  h = (end - start) / dble(n)
  s1 = 0.0D0
  do i = 1, n-1
    s1 = s1 +  f(start + h * i)
  end do
  s1 = (s1 * 2.0D0 + f(start) + f(end)) * h / 2.0D0
```

台形積分プログラム（2/3）

```
  write(*, '(A, F15.12)') 'Trapezoid: ', s1

  s2 = ff(end) – ff(start)
  write(*, '(A, F15.12)') 'Integral:  ', s2
stop
end program
```

台形積分プログラム（3/3）

```
function f(x)
  implicit none
  real(8), intent(IN) :: x
  real(8) :: f
  intrinsic sin

  f = sin(x)
end function

function ff(x)
  implicit none
  real(8), intent(IN) :: x
  real(8) :: ff
  intrinsic cos

  ff = -1d0 * cos(x)
end function
```

求根プログラム

方程式を解くプログラム　＝　求根プログラム

- 解析的な方法では解けない方程式がある。例えば

$$f(x) = -e^{-(x-1)^2} + \log x + \sqrt{x}$$

- 区間 (a, b) に $f(x)$=0 の解が1つあると考える。
- すると$f(a)$と$f(b)$の極性（正負）が異なる。
- aとbの中央を考える。$f(x) = 0$ の極性から解がどちらにあるか分かる
- 新しい（より狭い）区間 (a, b) を決める事ができる
- 区間が充分に狭くなるまで、繰り返すと解が求まる

課題

- 方程式を数値計算で解くプログラムを作成する
- 繰り返しは do while が向いている
  - aとbの区間が充分に小さくなったら終了
  - 例えば $|a-b|$ が0.01以下
  - 絶対値の計算は関数 abs で
- 解く方程式を関数として定義して用いる
  - 関数の例　区間 (0, 1) で解を持つ

$$f(x) = x^3 + x - 1$$