

第1回レポート

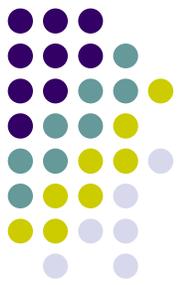
- 以下のプログラムを作成しメールで提出

「級数展開により $\sin(x\pi)$ を求め、表示する」

- $\sin(x\pi)$ の級数展開: 以下で近似する.

$$\sin(x\pi) \cong x\pi - \frac{(x\pi)^3}{3!} + \frac{(x\pi)^5}{5!} - \frac{(x\pi)^7}{7!} + \frac{(x\pi)^9}{9!} - \dots - \frac{(x\pi)^{2n-1}}{2n-1}$$

- x と n は実行時にキーボードから入力.
- 実数の計算は全て倍精度で行う.
- 提出先: `z5nt02in@s.kyushu-u.ac.jp`
- 締め切り: 6月1日(水) 必着



第1回レポート

- 実行例:

- n が 1 以上なら計算結果を表示

```
% ./repl  
x : 0.33333  
n: 10  
sin(x * pi) is about 0.8660202090600015
```

- n が 0 以下の場合, エラーを表示して終了

```
% ./repl  
x : 0.25  
n: 0  
Error! n must be > 0
```

エラーメッセージ

レポート提出について



- 字下げ，改行を使い，見やすいプログラムを書く.
- 必ず自分で実行し，正しい結果が得られることを確認
 - x や n に様々な値を入力して確認する.
 - \sin 関数を使った結果と比較
 - わざと n に負の値などを入力してエラーの表示も確認する.
- ソースファイルをメールに添付して提出
 - プログラム名を間違えないよう注意
 - 末尾に `~` が付いたファイルは最新版ではない
 - 末尾が `.f90` でないファイルはソースファイルではない
 - 本文には苦労した点などを書く.



レポート提出後の注意

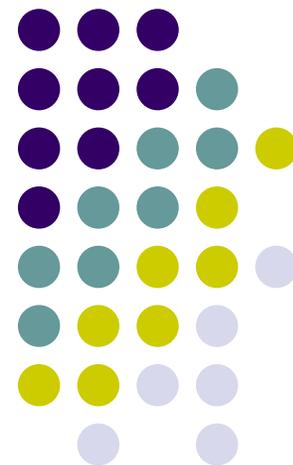
- 必ず返信を確認する.
 - 翌日の午前中までに返信予定
 - 間違い個所の指摘, ヒントなど
 - 締め切り前であれば何度でも再提出可
- Webページも確認.
 - 提出状況.
 - 翌日の午前中までに反映予定
 - レポートに関する追加情報.



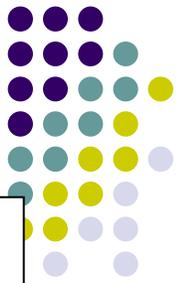
相談日

- 日時 5月31日(火) 17:00~19:00
- 場所 第1講義室
- レポートや講義内容に関する質問受け付け.
 - それ以外の時間の質問は, とりあえずメールで.

第一回レポート 解答



第一回レポートの解答例 (1/2)



```
program report1
  implicit none
  integer :: n, i, j
  real(8) :: x, total, pi, rad, f, g
  intrinsic dble, atan

  pi = 4.0D0 * atan(1.0D0)

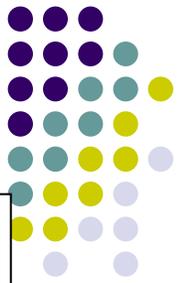
  write(*, *) 'Enter x (times pi): '
  read(*, *) x

  rad = pi * x

  write(*, *) 'Enter n: '
  read(*, *) n

  if (n <= 0) then
    write(*, *) 'Error: n<=0'
    stop
  endif
```

何度も利用する $\pi * x$ を予め計算しておく



第一回レポートの解答例 (2/2)

```
total = 0.0D0
f = 1.0D0
do i = 1, n
  g = 1.0D0
  ! calculate (2i-1)!
  do j = 1, 2 * i - 1
    g = g * dble(j)
  end do
  total = total + f * rad**(2*i-1) / g
  f = -1.0D0 * f                ! flip flag
end do
```

$(-1.0D0)^{(i+1)}$ だと $i+1$ 回の掛け算が必要

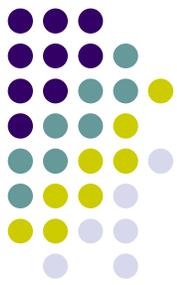
```
write(*, '(a, f7.3, a, f15.10)') &
  'sin (' , x, '*pi) is about ', total
```

```
stop
end program
```

細かい注意点

- 整数と実数
- 字下げ





整数と実数

```
do i=1,n
  a = 1.0D0
  do j=1,2*i-1
    a = a * (x*pi/j)
  end do
  total = total + (-1)**(i+1) * a
end do
```

```
do i=1,n
  a = 1.0D0
  do j=1,2*i-1
    a = a * (x*pi/dbl(j))
  end do
  total = total + (-1.0D0)**(i+1) * a
end do
```

- 実数式の中で整数を使う場合は実数に変換する.
- ただし、べき乗する数は整数でも可

字下げ



```
program report1
implicit none
integer::i,j
real(8)::a,b,total,x,n,pi
intrinsic atan

pi=4.0D0*atan(1.0D0)
write(*,*) 'Enter n: '
read(*,*) n
write(*,*) 'Enter x(*pi):'
read(*,*) x
if (n<=0.0D0) then
write(*,*) 'Error:n must be >0'
end if
total=0.0D0
do j=1,n
a=1.0D0
```

```
program report1
  implicit none
  integer::i,j
  real(8)::a,b,total,x,n,pi
  intrinsic atan

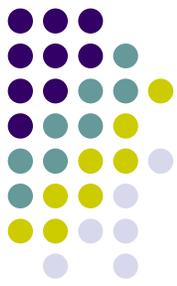
  pi=4.0D0*atan(1.0D0)

  write(*, *) 'Enter n: '
  read(*,*) n
  write(*, *) 'Enter x(*pi):'
  read(*,*) x

  if (n<=0.0D0) then
    write(*,*) 'Error:n must be >0'
  end if

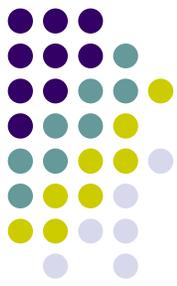
  total=0.0D0
  do j=1,n
    a=1.0D0
```

- 字下げの幅は一定にしたほうが見やすい。
- 文の意味が変化する箇所に改行を入れると見やすい。



プログラムの工夫

- 実数の表示形式
- 桁あふれ対策1: 整数より倍精度実数
- 桁あふれ対策2: 計算の順序



実数の表示形式

- 書式を指定せずに実数を出力した場合

- プログラム

```
write(*, *) 'sin (' , x, '*pi) is about ', total
```

- 出力結果

```
sin ( 1.0000000000000000 *pi) is about 3.458669144327514E-16
```

- $3.458669144327514E-16 = 3.458669144327514 \times 10^{-16}$

- 実数の表示形式を F にする

- プログラム

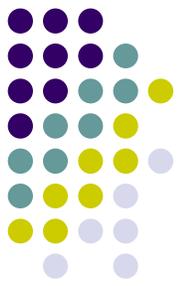
```
write(*, '(a, f7.3, a, f15.10)') &  
      'sin (' , x, '*pi) is about ', total
```

全体で 15桁, 小数点以下 10桁

- 出力結果

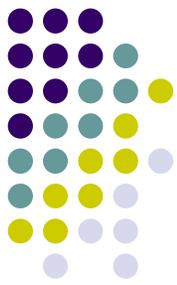
```
sin ( 1.000*pi) is about 0.0000000000
```

小数点以下10桁では表示できないくらい小さい



桁あふれ (overflow)

- n や x が大きい場合 $(2^i-1)!$ が非常に大きくなり、
計算機で普通に扱える範囲を超える。
→ 桁あふれ
- データの型(整数, 単精度実数, 倍精度実数)によって
扱える範囲が違う



桁あふれ対策 1

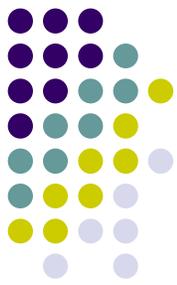
- 整数より実数

- 計算機で扱える値の範囲

- 整数 $-2^{31} \sim 2^{31} - 1$
(= -2,147,483,648 ~ 2,147,483,647)

- 倍精度実数 約 $\pm 1.8 \times 10^{308}$ まで

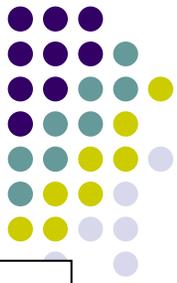
- ちなみに絶対値の小さい値は約 $\pm 4.9 \times 10^{-324}$ まで
 - それより小さくなると 0 となる. (underflow)



桁あふれ対策 2

- 計算の順番を変える
 - 足し算と引き算，掛け算と割り算を交互に行う.
- レポートの例

$$\boxed{\frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots} \quad \longrightarrow \quad \boxed{\frac{x}{1} - \frac{x}{1} \cdot \frac{x}{2} \cdot \frac{x}{3} + \frac{x}{1} \cdot \frac{x}{2} \cdot \frac{x}{3} \cdot \frac{x}{4} \cdot \frac{x}{5} - \dots}$$



改善案 (1) 1/2

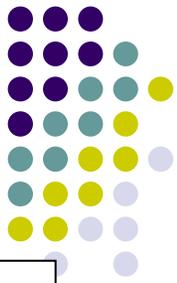
```
program report1
  implicit none
  integer :: n, i, j
  real(8) :: x, total, pi, rad, f, g
  intrinsic sin, atan

  pi = 4.0D0 * atan(1.0D0)

  write(*, *) 'Enter x (times pi): '
  read(*, *) x
  rad = x * pi

  write(*, *) 'Enter n: '
  read(*, *) n

  if (n <= 0) then
    write(*, *) 'Error: n<=0'
    stop
  endif
```



改善案 (1) 2/2

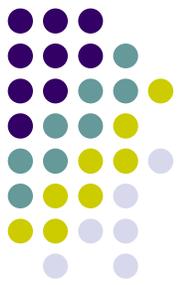
```
total = 0.0D0
f = 1.0D0
do i = 1, n
  g = 1.0D0
  do j = 1, 2 * i - 1
    g = g * rad / dble(j)
  end do
  total = total + f * g
  f = -1.0D0 * f          ! flip flag
end do

write(*, '(a, f7.3, a, f15.10)') &
  'sin (' , x, '*pi) is about ', total
stop
end program
```



面白い工夫

- 計算の節約
- n が整数か否か判別する
- 正しい値が入力されるまで繰り返す
- ガンマ関数を使ったべき乗計算



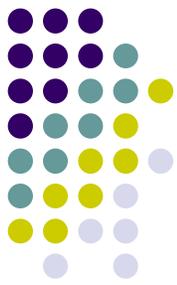
計算の節約

- 同じ計算を何度も行っている

$$\frac{x}{1} - \frac{x}{1} \cdot \frac{x}{2} \cdot \frac{x}{3} + \frac{x}{1} \cdot \frac{x}{2} \cdot \frac{x}{3} \cdot \frac{x}{4} \cdot \frac{x}{5} - \dots$$

- 計算回数と所要時間
 - このプログラムでは掛け算と割り算をそれぞれ n^2 回 ($\sum_{i=1}^n 2i-1$ 回)
 - サーバ ah.s.kyushu-u.ac.jp の計算スピードは、
どんなに頑張っても約 2.2×10^9 回/秒
 - 実際は割り算に時間がかかるので約 3.0×10^7 回/秒程度.
 - 所要時間:
 - n = 10000 の時: 約 3秒
 - n = 100000 の時: 約 300秒

効率の良い計算



- 直前の計算結果を利用

$$total_1 = x$$

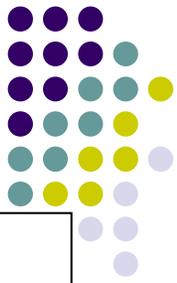
$$y_1 = x$$

$$y_i = (-1) \cdot y_{i-1} \cdot \frac{x}{(2i-2)} \cdot \frac{x}{(2i-1)}$$

$$total_i = total_{i-1} + y_i$$

(**i = 2 ... n**)

改善案 (2) 1/2



```
program report1
  implicit none
  integer :: n, i, h
  real(8) :: x, total, pi, rad, y
  intrinsic sin, atan

  pi = 4.0D0 * atan(1.0D0)

  write(*, *) 'Enter x (times pi): '
  read(*, *) x
  rad = x * pi

  write(*, *) 'Enter n: '
  read(*, *) n

  if (n <= 0) then
    write(*, *) 'Error: n<=0'
    stop
  endif
```



改善案 (2) 2/2

```
total = rad
y = rad
do i = 2, n
  ! calculate  $Y_i = -1 * Y_{(i-1)} * x * x / ((2i-1)*(2i-2))$ 
  h = (2 * i - 2) * (2 * i - 1)
  y = -1.0D0 * y * rad * rad / dble(h)
  ! calculate  $TOTAL_i = TOTAL_{(i-1)} + Y_i$ 
  total = total + y
end do

write(*, '(a, f7.3, a, f15.5)') &
  'sin (' , x, '*pi) is about ', total
stop
end program
```

nが整数か否か



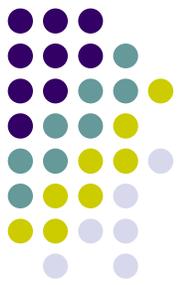
```
real(8) :: n

...

write(*,*) '級数のnは?'
read(*,*) n

c = ceiling(n)

if (n <= 0 .or. n /=c) then
  write(*,*) "nは自然数にしてね♪"
  stop
end if
```

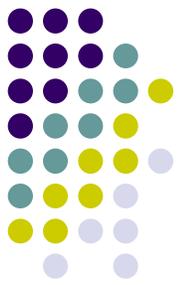


正しい値が入力されるまで繰り返し

```
write(*,*) "n=?"  
read(*,*) n  
do while (n <= 0.0D0)  
    write(*,*)"Error! n must be > 0. Please try again!"  
    write(*,*) "n=?"  
    read(*,*) n  
end do
```

補足: do while(条件) ~ end do

条件が満たされている間, end do までの処理を繰り返す.



x が大きすぎる場合に -1.0 ~ 1.0 になるよう調整

- -1.0~1.0になるまで 1.0 を加減

```
y = x
do while (y >= 1.0D0)
  y = y -1.0D0
end do

do while (y <= -1.0D0)
  y = y +1.0D0
end do
```

- 他の方法:
 - 切り捨てた値を利用

```
y = x - int(x)
```

入力値 x の単位や計算方法を選択



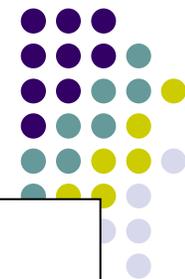
```
degtorad = pi / 180.0D0

write(*, *) '入力値xの種別'
write(*, *) '1:x radian/2: $\pi$ x radian/other:degree ? '
read(*, *) type

select case(type)
  case('1')
    xrad = x
    xpi = x / pi
    xdeg = x / degtorad
  case('2')
    xrad = x * pi
    xpi = x
    xdeg = x * 180.0D0
  case default
    xrad = x * degtorad
    xpi = x / 180.0D0
    xdeg = x
end select
```

計算方法の選択も同様

ガンマ関数を用いた階乗計算



```
total = 0.0D0
f = 1.0D0
do i = 1, n, 1
  total = total + f * rad**(2*i-1)/gamma(2.0D0*dble(i))
  f = -1.0D0 * f
end do
```

- ガンマ関数:

$$\Gamma(x) = \int_0^{\infty} e^{-t} t^{x-1} dt$$

- ガンマ関数の性質

$$\Gamma(x+1) = x\Gamma(x)$$

より

$$\Gamma(n+1) = n!$$