Programação estruturada no Fortran 90 - 2

Alexandre Diehl

Departamento de Física – UFPel

Subprograma do tipo FUNÇÃO

Faça um programa em Fortran 90 que calcule o valor do cos-seno de x através da série abaixo,

$$\text{co-seno}(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \ldots$$

Calcule a diferença entre o valor calculado pela série acima e o valor fornecido pela função intrínseca $\cos(x)$ do Fortran 90, para um ângulo x fornecido pelo usuário, para 4, 8 e 16 termos da série.

Subprograma do tipo FUNÇÃO

```
program serie_cosseno
  implicit none
  real(8) :: x, pi, cosseno
  integer(8):: i, nTermos

  read*, x
  read*, nTermos
  pi = 4.d0 * datan (1.d0)
  x = x * pi / 180.d0
  cosseno = 1.d0
  do i = 1, nTermos - 1
     cosseno = cosseno + (-1)**(i) * x**(2*i) / fatorial(2*i)
  end do
  print*,cosseno,cos(x)

CONTAINS

  integer(8) function fatorial (N)
     [...]
  end function fatorial

end program serie_cosseno
```

Subprograma do tipo FUNÇÃO

```
program serie_cosseno

   [...]

CONTAINS

   integer(8) function fatorial (N)
     implicit none
     integer(8) :: j, N

     fatorial = 1
     do j = 1,N
        fatorial = fatorial * j
     end do

   end function fatorial

end program serie_cosseno
```

Comandos de especificação em Funções

→ usados para designar variáveis com características comuns, particularmente com relação à maneira como as mesmas serão alocadas na memória do computador.

Comando COMMON:

→ especifica que variáveis declaradas em **diferentes blocos**, possivelmente com nomes diferentes, ocupam os **mesmos endereços de memória do computador**.

→ permite que dados declarados no Programa Principal sejam acessados e/ou modificados por uma Função.

```
COMMON [/ <nome-do-bloco> /] <nome-1> ... <nome-n>
```

<nome-do-bloco>: é um nome opcional, associado a uma **lista de variáveis** do bloco COMMON.

<nome-1> ... <nome-n>: são os nomes das variáveis, as quais serão armazenadas em uma mesma área de memória do que outras variáveis, declaradas em outro bloco e que também, por sua vez, deverão ser listadas em comando COMMON no bloco em que foram declaradas.

Comando COMMON: um exemplo simples

```
program testa_common
  implicit none
  real :: a = 1.0, B = 2.0, C
  real :: FUNC
  COMMON /BLOCO1/ A, B

  print*,FUNC()

end program testa_common

real FUNCTION func()
  implicit none
  real :: u,v
  COMMON /BLOCO1/ u, v

  print*,'␣u=',u,'␣v=',v
  FUNC = u + v
end FUNCTION func
```

- As variáveis A e u são nomes distintos para o mesmo endereço de memória.
- As variáveis B e v são nomes distintos para o mesmo endereço de memória.
- Como não usamos CONTAINS, as variáveis declaradas no Programa Principal são válidas apenas neste escopo.

```
diehl@lua:~/Documents/ensino/UFPEL/prog_compu
 u=   1.00000000      v=   2.00000000
  3.00000000
diehl@lua:~/Documents/ensino/UFPEL/prog_compu
```

Recursividade em Funções

- Uma Função A é dita recursiva quando invoca (ou chama) a si mesma, ou seja, A invoca A diretamente. Esta recursividade é dita direta.
- Devemos usar o atributo RECURSIVE e a palavra-chave RESULT na definição da Função:

```
RECURSIVE [tipo] FUNCTION nome-da-função (...) RESULT (var)
```

- O atributo RECURSIVE indica que a Função é recursiva.
- A palavra-chave RESULT especifica o nome de uma variável var à qual o valor do resultado do desenvolvimento da Função deve ser atribuído, em lugar do nome da Função propriamente dito. Se omitirmos a palavra-chave, quando do uso do atributo RECURSIVE, um erro de compilação será gerado.
- A palavra-chave RESULT pode ser usada também em Funções não-recursivas.

Subprograma do tipo FUNÇÃO

Recursividade em Funções: fatorial de n (n!)

```
integer :: f, n
print*,'n ?'
read*,n
if (n >= 0) then
    f = fatorial(n)
    write(*,'("FATORIAL(",I12,")=",I12)')n,f
else
    print*,'Valor invalido para n.'
end if
CONTAINS
RECURSIVE integer FUNCTION FATORIAL(n) RESULT(f)
  implicit none
  integer, intent(in) :: n
  if (n > 0) then
      f = FATORIAL(n-1)*n
      print*,'n = ',n,' F = ',f
  else if (n == 0) then
      f = 1
      print*,'n = ',n,' f = ',f
  end if
end FUNCTION FATORIAL
end
```

Subprograma do tipo FUNÇÃO

Recursividade em Funções: fatorial de n (n!)

```
diehl@lua:~/Documents/ensino/UFPEL/prog_comput_fisica/f90$ gfortran teste.f90
diehl@lua:~/Documents/ensino/UFPEL/prog_comput_fisica/f90$ ./a.out
 n ?
14
 n =            0  f =            1
 n =            1  F =            1
 n =            2  F =            2
 n =            3  F =            6
 n =            4  F =           24
 n =            5  F =          120
 n =            6  F =          720
 n =            7  F =         5040
 n =            8  F =        40320
 n =            9  F =       362880
 n =           10  F =      3628800
 n =           11  F =     39916800
 n =           12  F =    479001600
 n =           13  F =   1932053504
 n =           14  F =   1278945280
FATORIAL(          14)=   1278945280
diehl@lua:~/Documents/ensino/UFPEL/prog_comput_fisica/f90$
```

Teste os valores até n = 12. O que acontece quando n > 12? Como resolver o problema numérico?

Recursividade em Funções: fatorial de n (n!)

- Erro de compilação: declaração da Função sem o atributo RECURSIVE

integer FUNCTION FATORIAL(n) RESULT(f)

```
diehl@lua:~/Documents/ensino/UFPEL/prog_comput_fisica/f90$ gfortran teste.f90
teste.f90:17.19:

       f = FATORIAL(n-1)*n
                   1
Error: Function 'fatorial' at (1) cannot be called recursively, as it is not RECURSIVE
diehl@lua:~/Documents/ensino/UFPEL/prog_comput_fisica/f90$
```

Subprograma do tipo FUNÇÃO

Recursividade em Funções: fatorial de n (n!)

- Erro de compilação: declaração da Função sem a palavra-chave RESULT

RECURSIVE integer FUNCTION FATORIAL(n)

```
diehl@lua:~/Documents/ensino/UFPEL/prog_comput_fisica/f90$ gfortran teste.f90
teste.f90:17.19:

       f = FATORIAL(n-1)*n
                   1
Error: 'fatorial' at (1) is the name of a recursive function and so refers to the result variable. Use an explicit RESULT variable for
recursion (12.5.2.1)
diehl@lua:~/Documents/ensino/UFPEL/prog_comput_fisica/f90$
```

Subprograma do tipo FUNÇÃO

Considere a função $x(t)$, definida como

$$x(t) = t^2 \ ,$$

que descreve o movimento de uma partícula em movimento uniformemente variado. Faça um programa em Fortran 90 que calcule as velocidades médias entre os instantes $t = 5$ segundos e $t = 5 + \Delta t$ segundos e $t = 5$ segundos e $t = 5 - \Delta t$ segundos. Compare estas duas velocidades para valores sucessivos de Δt,

$$\Delta t = 1, 0.1, 0.01, 0.001, 0.0001, \ldots$$

Use um subprograma do tipo FUNCTION para calcular a velocidade média entre os dois instantes de tempo.