Programação estruturada no Fortran 90 - 6

Alexandre Diehl

Departamento de Física – UFPel

Sub-Programas como argumentos de rotinas

- Usado nos casos em que os nomes de rotinas são usados como argumentos mudos de outras rotinas.
- No caso de **rotinas EXTERNAS** (sem CONTAINS), definidas em arquivos .f90 em separado:
  → No padrão do Fortran 90 devemos usar **blocos de** INTERFACE.

```
INTERFACE
  <corpo interface>
END INTERFACE
```

- O <corpo interface> consiste das declarações FUNCTION ou SUBROUTINE, declarações dos tipos e espécies dos argumentos dos Subprogramas, **sem seus comandos executáveis**.
- Em qualquer unidade de programa, os **blocos de** INTERFACE deves ser inseridos após as declarações de variáveis mudas ou de variáveis locais.
- Os nomes das variáveis usados nos **blocos de** INTERFACE podem ser distintos dos nomes mudos definidos na rotina ou usados na chamda da mesma, mas os tipos e espécies devem ser os mesmos.
- Os **blocos de** INTERFACE fornecem os mecanismos de controle para o compilador garantir a coerência entre as diferentes unidades de programa.

Sub-Programas como argumentos de rotinas

Exemplo: Cálculo do máximo da função $y = e^x \sin(x)$, entre 0 e π.

```
subroutine calc_maximo(x,f,FUNC)
  implicit none
  real, intent(in) :: x
  real, intent(out) :: f
  real :: FUNC
  f = FUNC(x)
end subroutine calc_maximo
```

```
real function FUNC(x)
  real, intent(in) :: x

  FUNC = exp(x)*sin(x)

end function FUNC
```

```
program maximo
  [...]
  INTERFACE
     real function FUNC(x)
       real, intent(in) :: x
     end function FUNC
  END INTERFACE
  [...]
end program maximo
```

- O **bloco de** INTERFACE, contendo a Função FUNC(x) não pode ser omitido no Programa Principal.

Sub-Programas como argumentos de rotinas

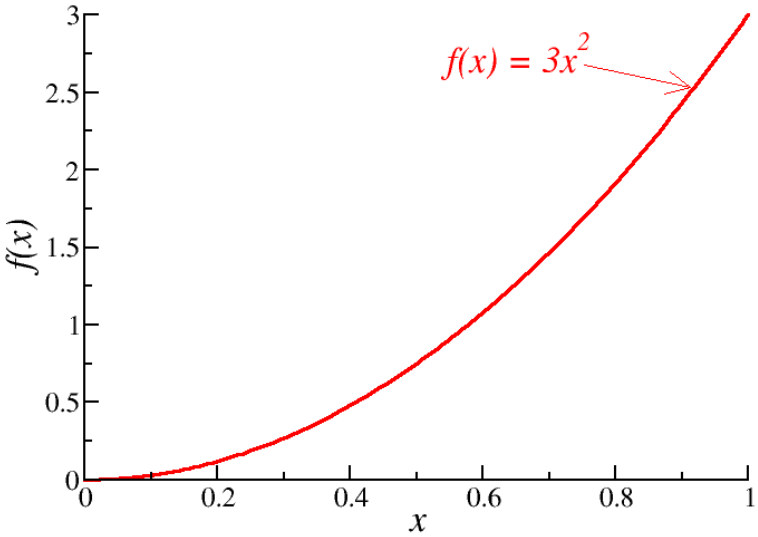
Exemplo: Cálculo do máximo da função $y = e^x \sin(x)$, entre 0 e π.

```
program maximo
  real :: a, b, x, f ;    real :: x_maximo, f_maximo
  INTERFACE
     real function FUNC(x)
       real, intent(in) :: x
     end function FUNC
  END INTERFACE
  print*,"␣limites␣para␣a␣funcao?" ;      read*,a,b
  x = a ;     f_maximo = TINY(f_maximo)
  do
     if (x > b) exit
     call calc_maximo(x,f,FUNC)
     if (f >= f_maximo) then
        f_maximo = f ;          x_maximo = x
     end if
     x = x + 0.01
  end do
  print*,"␣x␣onde␣y␣e␣maximo␣=␣",x_maximo
  print*,"␣maximo␣de␣y␣entre␣a␣e␣b␣=␣",f_maximo
end program maximo
```

Sub-Programas como argumentos de rotinas

Cálculo de integrais

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

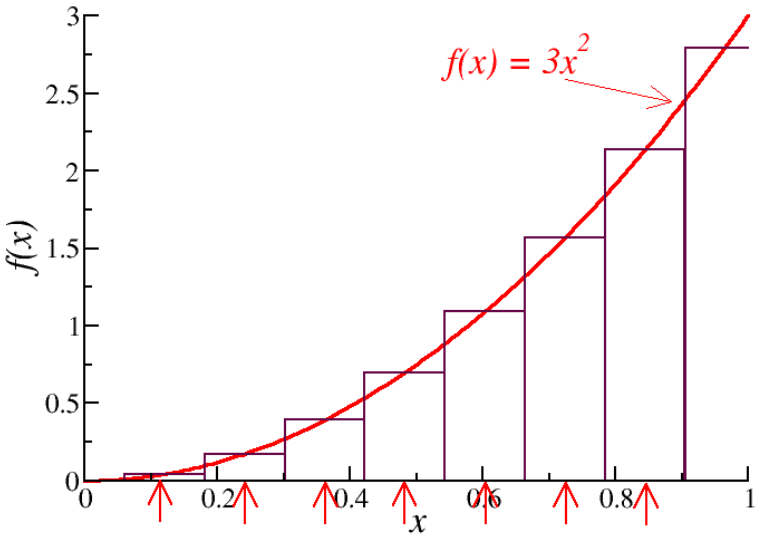


Como calcular numericamente esta integral I?

Sub-Programas como argumentos de rotinas

Cálculo de integrais - método de quadratura

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



- Métodos tradicionais :

... trapezoidal, Simpson, etc.

devemos somar áreas com formas que se aproximam da forma da curva $f(x)$

$$I \approx \Delta x \sum_{i=1}^{n} f(x_i) = \frac{b-a}{n} \sum_{i=1}^{n} f(x_i)$$

n é o número de intervalos entre os limites de integração.

Sub-Programas como argumentos de rotinas

Cálculo de integrais - regra trapezoidal

```
program integra
  real :: a, b, s
  integer :: n

  interface
     real function FUNC(x)
       real, intent(in) :: x
     end function FUNC
  end interface

  [...]

  call trapezio(FUNC,a,b,n,s)

  [...]
end program integra
```

```
subroutine trapezio(FUNC,a,b,n,s)

  [...]

end subroutine trapezio
```

```
real function FUNC(x)
  real, intent(in) :: x

  FUNC = 3.0*x**2

end function FUNC
```

Sub-Programas como argumentos de rotinas

Cálculo de integrais - regra trapezoidal

```
subroutine trapezio(FUNC,a,b,n,s)
  integer, intent(in) :: n
  real, intent(in) :: a, b
  real, intent(out) :: s
  real :: delta, x
  real :: FUNC
  delta = (b-a)/real(n)
  x = a
  s = 0.0
  do
     if (x > b) exit
     s = s + FUNC(x)
     x = x + delta
  end do
  s = s*delta
end subroutine trapezio
```

- A rotina tem limitações de precisão, especialmente para funções com variações acentuadas entre os limites de integração.
- Para rotinas mais sofisticadas, visite o site do Numerical Recipes (http://numerical.recipes/).

Sub-Programas como argumentos de rotinas

Use a regra do trapézio para obter o resultado da integral abaixo:

$$I = \int_0^{2\pi} \sin^2(x)\, dx \; .$$

As seguintes condições devem ser verificadas no programa: a) use precisão dupla do tipo real(8); b) use rotinas externas; c) as estimativas numéricas da integração devem ser obtidas de forma iterativa, tomando valores crescentes do número de intervalos n, desde $n = 1$ até um dado valor de n, tal que se obtenha uma precisão de 1×10^{-5} entre duas estimativas numéricas sucessivas; d) compare o resultado aproximado obtido com o valor exato da integral I, que é π; e) produza um arquivo de saída com os valores de n, das sucessivas estimativas numéricas para a integral I e o erro obtido, a fim de usar no xmgrace.