

OBJETIVOS DO CAPÍTULO

- Calcular o resultado de séries infinitas
- Aplicar os métodos da iteração linear, bisseção e Newton para determinar raízes de equações
- Revisar o uso de arquivos de dados e de resultados, Notepad e Wgnuplot
- Alterar dinamicamente arquivo de comandos do programa de gráficos Wgnuplot e usar novos comandos

18.1 projeto programa18a

- 1) Objetivos do programa:
 - (a) exemplificar o cálculo do resultado de séries infinitas;
 - (b) revisar o uso de arquivos de dados e de resultados, Notepad e Wgnuplot; e
 - (c) alterar dinamicamente arquivo de comandos do programa de gráficos Wgnuplot e usar novos comandos.
- 2) No Fortran, **criar um projeto** com o nome **programa18a**
- 3) No Fortran, **criar e inserir** no projeto o programa-fonte **programa18a.f90**
- 4) No Fortran, **copiar** exatamente o texto em vermelho mostrado na **Tabela 18.1**.
- 5) **Criar e inserir** na pasta do projeto o arquivo **dados18a.txt** mostrado na Figura 18.1.
- 6) **Criar e inserir** na pasta do projeto o arquivo **comandos18a.gnu** mostrado na Figura 18.2.
- 7) **Inserir** na pasta do projeto o arquivo **Wgnuplot.exe**
- 8) Comentários sobre o programa:
 - (a) O programa18a.f90 é composto pelo programa-principal e seis sub-rotinas.
 - (b) A sub-rotina DADOS é usada para ler os dois dados do programa do arquivo dados18a.txt.
 - (c) A sub-rotina SERIE_1 é usada para calcular a soma dos N termos da seguinte série geométrica infinita:

$$S = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots = \sum_{I=1}^{\infty} \left(\frac{1}{2^I} \right) = 1 \quad (18.1)$$

- (d) A sub-rotina SERIE_2 é usada para calcular o valor de π através da soma dos N termos da seguinte série infinita:

$$S = 4 \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots \right) = 4 \sum_{I=0}^{\infty} \left[\frac{(-1)^I}{(2I+1)} \right] = \pi \quad (18.2)$$

- (e) A sub-rotina SERIE_3 é usada para calcular o valor do número e através da soma dos N termos da seguinte série infinita, onde cada termo envolve a sub-rotina FATORIAL:

$$S = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots = \sum_{I=0}^{\infty} \left(\frac{1}{I!} \right) = e \quad (18.3)$$

- (f) As sub-rotinas das três séries também calculam o erro para a solução exata e escrevem no arquivo saída18a.txt o número e valor de cada termo da série, a soma dos termos e o erro.

Tabela 18.1 Programa18a.f90

```

program series

use portlib

implicit none
integer :: i, n, ver, tipo_serie
real*8 :: soma, exato, erro, termo

call dados

open(6,file="saída18a.txt")
write(6,10)
10 format("#      i", t17,"termo", t47,"soma", t77,"erro")

select case ( tipo_serie )
  case ( 1 )
    call serie_1
  case ( 2 )
    call serie_2
  case ( 3 )
    call serie_3
end select

close(6)

ver = system ("Notepad saída18a.txt")

call grafico

!-----  

contains

```

```

!-----

subroutine dados

ver = system ("Notepad dados18a.txt")

open(5,file="dados18a.txt")
read(5,*) tipo_serie
read(5,*) n
close(5)

end subroutine dados

!-----

subroutine serie_1

! calcula a soma de uma série geométrica

exato = 1

soma = 0

do i = 1, n
  termo = 1.0d0 / ( 2.0d0 ** i )
  soma = soma + termo
  erro = dabs(exato - soma)
  write(6,10) i, termo, soma, erro
  10 format( i8, 3(1pe30.15e3) )
end do

end subroutine serie_1

!-----

subroutine serie_2

! calcula a soma da série de pi

exato = dacos(-1.0d0)

soma = 0

do i = 0, n

```

```

termo = 4.0d0 * ((-1)**i) / ( 2.0d0 * i + 1 )
soma  = soma + termo
erro  = dabs(exato - soma)
write(6,10) i, termo, soma, erro
10 format( i8, 3(1pe30.15e3) )
end do

```

```
end subroutine serie_2
```

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

```
subroutine serie_3
```

```
! calcula a soma da série de e
```

```
real*8 fatorial_i
```

```
exato = dexp(1.0d0)
```

```
soma = 0
```

```
do i = 0, n
    call fatorial ( i, fatorial_i)
    termo = 1 / fatorial_i
    soma  = soma + termo
    erro  = dabs(exato - soma)
    write(6,10) i, termo, soma, erro
    10 format( i8, 3(1pe30.15e3) )
end do
```

```
end subroutine serie_3
```

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

```
subroutine fatorial(j,fat)
```

```
! calcula o fatorial de j
```

```
integer j
real*8  fat, k
```

```
fat = 1
```

```
do k = 2, j
    fat = fat * k
```

```

end do

end subroutine fatorial

!-----

subroutine grafico

integer k

open(9,file="comandos18a.gnu")
do k = 1, 7
  read(9,*)
end do

select case ( tipo_serie )
  case ( 1 )
    write(9,*) "set title 'série geométrica'"
  case ( 2 )
    write(9,*) "set title 'série de pi'"
  case ( 3 )
    write(9,*) "set title 'série de e'"
end select

write(9,*) "replot"

close(9)

ver = system ("Wgnuplot comandos18a.gnu")

end subroutine grafico

!-----

end program series

```

- (g) A sub-rotina GRAFICO: abre o arquivo comandos18a.gnu; desce as sete primeiras linhas deste arquivo com o comando READ vazio; na oitava linha, escreve um comentário para representar o título do gráfico de acordo com a série escolhida pelo usuário, no arquivo de dados, para cálculo; na nona linha, escreve outro comentário, que é o comando replot do Wgnuplot; finalmente, esta sub-rotina executa o programa Wgnuplot para fazer o gráfico semilog do erro do cálculo da série escolhida, em função do número de termos N usado.

- (h) O programa-principal: chama a sub-rotina DADOS; cria o arquivo saída18a.txt; escreve nele um comentário na primeira linha para indicar o significado de cada coluna de resultados; o símbolo # é usado para informar ao Wgnuplot que a linha é apenas um comentário, e não dados para o gráfico; dependendo do tipo de série escolhida pelo usuário, é chamada a sub-rotina adequada; o Notepad abre o arquivo de saída; e, é chamada a sub-rotina para fazer o gráfico do erro.
- (i) O arquivo comandos18a.gnu usa comandos do Wgnuplot vistos no Capítulo 12, exceto o “notitle” que é usado para não colocar título em uma curva.

```

dados18a - Bloco de Notas
Arquivo Editar Formatar Exibir Ajuda
 1 = tipo_serie
 100 = n = número de termos da série

tipo_serie:
 1 = série geométrica
 2 = série de pi
 3 = série de e

Arquivo de dados do programa18a.f90

```

Figura 18.1 Arquivo dados18a.txt do projeto programa18a.

```

comandos18a - Bloco de Notas
Arquivo Editar Formatar Exibir Ajuda
set grid
set time
set data style lines
set logscale y
set xlabel 'número de termos'
set ylabel 'módulo do erro = analítico - numérico'
plot 'saída18a.txt' using 1:4 notitle
set title 'série geométrica'
replot

```

Figura 18.2 Arquivo comandos18a.gnu do projeto programa18a.

- 9) **Estudar o programa-fonte** considerando os comentários do item 8, acima.
- 10) Executar **Build, Compile** para compilar o programa.
- 11) Gerar o programa-executável fazendo **Build, Build**.
- 12) Executar o programa através de **Build, Execute. Usar, os dados mostrados na Figura 18.1.**
- 13) **Analizar os resultados** mostrados nas Figuras 18.3 e 18.4. Observações:

- (a) As linhas de $i = 6$ a 94 foram cortadas para reduzir o tamanho da Figura 18.3.
- (b) O símbolo `#` na primeira linha da Figura 18.3 é usado para que o aplicativo Gnuplot ignore o conteúdo desta linha, já que ela só contém comentários e não valores.
- 14) Executar novamente o programa usando `tipo_serie = 2` com **1000 termos** e **analisar** os novos resultados.
- 15) Executar novamente o programa usando `tipo_serie = 3` com **100 termos** e **analisar** os novos resultados.
- 16) Executar novamente o programa usando outros dados e **analisar** os novos resultados.

*saida18a - Bloco de Notas

Arquivo Editar Formatar Exibir Ajuda

#	i	termo	soma	erro
1		5.000000000000000E-001	5.000000000000000E-001	5.000000000000000E-001
2		2.500000000000000E-001	7.500000000000000E-001	2.500000000000000E-001
3		1.250000000000000E-001	8.750000000000000E-001	1.250000000000000E-001
4		6.250000000000000E-002	9.375000000000000E-001	6.250000000000000E-002
5		3.125000000000000E-002	9.687500000000000E-001	3.125000000000000E-002
95		2.524354896707238E-029	1.000000000000000E+000	0.000000000000000E+000
96		1.262177448353619E-029	1.000000000000000E+000	0.000000000000000E+000
97		6.310887241768094E-030	1.000000000000000E+000	0.000000000000000E+000
98		3.155443620884047E-030	1.000000000000000E+000	0.000000000000000E+000
99		1.577721810442024E-030	1.000000000000000E+000	0.000000000000000E+000
100		7.888609052210118E-031	1.000000000000000E+000	0.000000000000000E+000

Ln 7, Col 1 100% Windows (CRLF) UTF-8

Figura 18.3 Parte do arquivo de resultados do projeto programa18a para os dados da Figura 18.1.

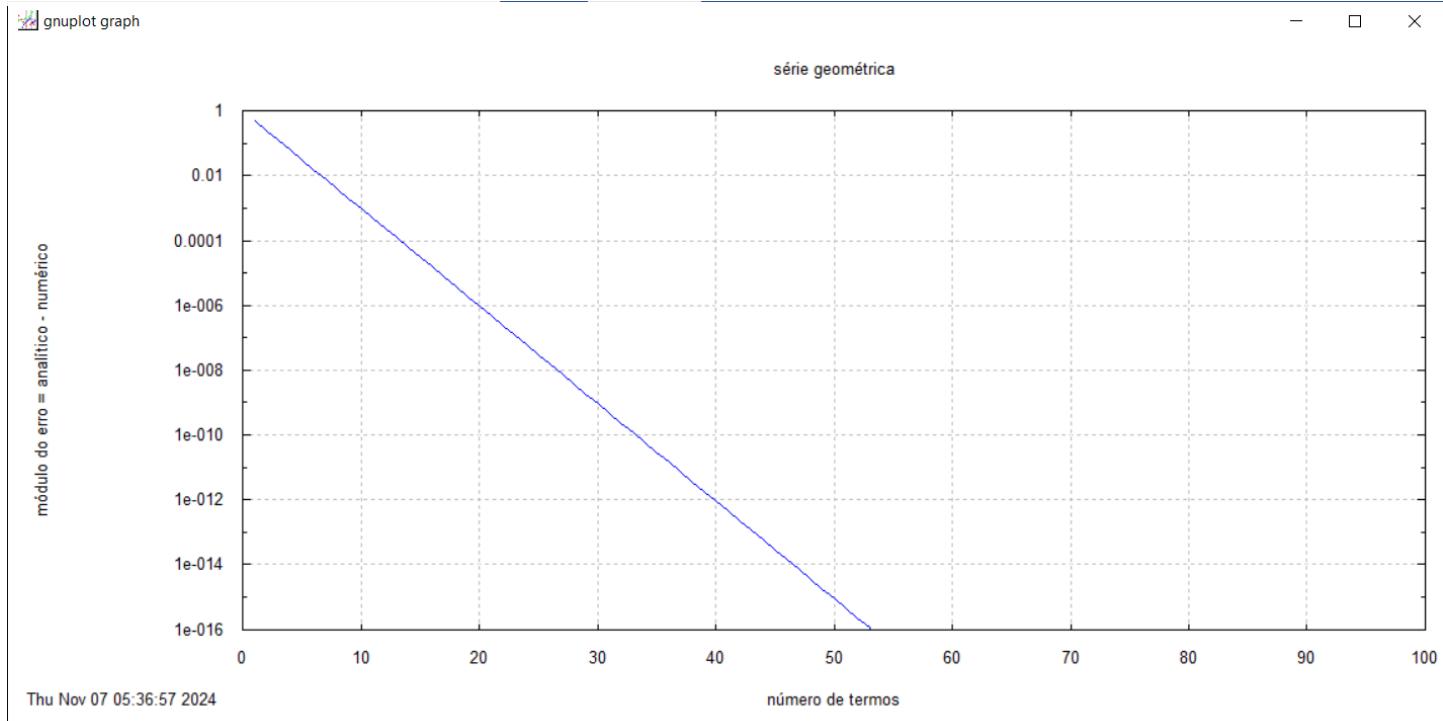


Figura 18.4. Gráfico com resultados do projeto programa18a para os comandos da Figura 18.2.

18.2 projeto programa18b

- 1) Objetivo do programa: para uma equação quadrática, específica, aplicar os métodos da iteração linear, bisseção e Newton para determinar suas raízes.
- 2) No Fortran, **criar um projeto** com o nome **programa18b**
- 3) No Fortran, **criar e inserir** no projeto o programa-fonte **programa18b.f90**
- 4) No Fortran, **copiar** exatamente o texto em vermelho mostrado na **Tabela 18.2**.
- 5) **Criar e inserir** na pasta do projeto o arquivo **dados18b.txt** mostrado na Figura 18.5.
- 6) **Criar e inserir** na pasta do projeto o arquivo **comandos18b.gnu** mostrado na Figura 18.6.
- 7) **Inserir** na pasta do projeto o arquivo **Wgnuplot.exe**
- 8) Comentários sobre o programa:
 - (a) O programa18b.f90 é composto pelo programa-principal e cinco sub-rotinas.
 - (b) Para as sub-rotinas DADOS e GRAFICO, e o programa-principal, valem os mesmos comentários da seção anterior, exceto que os arquivos envolvidos são agora dados18b.txt, comandos18b.gnu e saida18b.txt.
 - (c) As sub-rotinas LINEAR, BISSECAO e NEWTON resolvem a equação

$$x^2 - 5x + 6 = 0 \quad (18.4)$$

através dos métodos da iteração linear, bisseção e Newton para determinar suas raízes. Detalhes sobre estes métodos podem ser encontrados nos livros de cálculo numérico básico.

- 9) **Estudar o programa-principal** considerando os comentários do item 8, acima.
- 10) Executar **Build, Compile** para compilar o programa.
- 11) Gerar o programa-executável fazendo **Build, Build**.
- 12) Executar o programa através de **Build, Execute. Usar, os dados mostrados na Figura 18.5**.
- 13) **Analizar os resultados** mostrados nas Figuras 18.7 e 18.8. Observações:
 - (a) Diversas linhas foram cortadas para reduzir o tamanho da Figura 18.7.
 - (b) O símbolo # na primeira linha da Figura 18.7 é usado para que o aplicativo Wgnuplot ignore o conteúdo desta linha, já que ela só contém comentários e não valores.
- 14) **Executar** novamente o programa usando **tipo_metodo = 2** e **analisar** os novos resultados.
- 15) **Executar** novamente o programa usando **tipo_metodo = 3** e **analisar** os novos resultados.
- 16) **Executar** novamente o programa usando **tipo_metodo = 3, n = 20, exato = 3 e xo = 100** e **analisar** os novos resultados.
- 17) **Executar** novamente o programa usando outros dados e **analisar** os novos resultados.

Tabela 18.2 Programa18b.f90

```

program raizes

use portlib

implicit none
integer :: i, n, ver, tipo_metodo
real*8 :: exato, erro, xo, x, r

call dados

x = xo

open(6,file="saida18b.txt")
write(6,10)
10 format("#      i", t17,"x", t47,"erro")

erro = dabs(exato - x)
write(6,11) 0, x, erro
11 format( i8, 2(1pe30.15e3) )

select case ( tipo_metodo )
  case ( 1 )
    call linear
  case ( 2 )
    call bissecao
  case ( 3 )
    call newton
end select

close(6)

ver = system ("Notepad saida18b.txt")

call grafico

!-----
contains
!-----
subroutine dados

```

```

ver = system ("Notepad dados18b.txt")

open(5,file="dados18b.txt")
read(5,*) tipo_metodo
read(5,*) n
read(5,*) exato
read(5,*) xo
read(5,*) r
close(5)

end subroutine dados

!-----

subroutine linear

! calcula a raiz de uma equação quadrática com o método da iteração linear

do i = 1, n
  x = ( ( x ** 2 ) + 6 ) / 5
  erro = dabs(exato - x)
  write(6,10) i, x, erro
  10 format( i8, 2(1pe30.15e3) )
end do

end subroutine linear

!-----

subroutine bissecao

! calcula a raiz de uma equação quadrática com o método da bisseção

real*8 a, b, fa, fx

a = xo - r
b = xo + r

do i = 1, n
  fx = x**2 - 5*x + 6
  fa = a**2 - 5*a + 6
  if ( fx*fa < 0 ) then
    b = x
  else

```

```

    a = x
  end if
  x = (a + b) / 2
  erro = dabs(exato - x)
  write(6,10) i, x, erro
  10 format( i8, 2(1pe30.15e3) )
end do

end subroutine bissecao

!-----

subroutine newton

! calcula a raiz de uma equação quadrática com o método de Newton

real*8 f, fl

do i = 1, n
  f = x**2 - 5*x + 6
  fl = 2*x - 5
  x = x - f / fl
  erro = dabs(exato - x)
  write(6,10) i, x, erro
  10 format( i8, 2(1pe30.15e3) )
end do

end subroutine newton

!-----

subroutine grafico

integer k

open(9,file="comandos18b.gnu")
do k = 1, 7
  read(9,*)
end do

select case ( tipo_metodo )
  case ( 1 )
    write(9,*) "set title 'método da iteração linear'"
  case ( 2 )
    write(9,*) "set title 'método da bisseção'"

```

```

    case ( 3 )
        write(9,*) "set title 'método de Newton'"
    end select

    write(9,*) "replot"

    close(9)

    ver = system ("Wgnuplot comandos18b.gnu")

end subroutine grafico

!-----
end program raizes

```

```

dados18b - Bloco de Notas
Arquivo Editar Formatar Exibir Ajuda
1 = (tipo_metodo)
100 = (n)      número de iterações
2 = (exato)  solução exata
0 = (xo)      estimativa inicial da solução
3 = (r)       raio de convergência; só para método da bisseção

tipo_metodo:
1 = linear
2 = bisseção
3 = Newton

Arquivo de dados do programa18b.f90

```

Figura 18.5 Arquivo dados18b.txt do projeto programa18b.

```

comandos18b - Bloco de Notas
Arquivo Editar Formatar Exibir Ajuda
set grid
set time
set data style lines
set logscale y
set xlabel 'iteração'
set ylabel 'módulo do erro = analítico - numérico'
plot 'saída18b.txt' using 1:3 notitle
set title 'método da iteração linear'
replot

```

Figura 18.6 Arquivo comandos18b.gnu do projeto programa18b.

#	i	x	erro
0		0.000000000000000E+000	2.000000000000000E+000
1		1.200000000000000E+000	8.000000000000000E-001
2		1.488000000000000E+000	5.120000000000000E-001
3		1.642828800000000E+000	3.571712000000002E-001
4		1.739777293221888E+000	2.602227067781122E-001
50		1.999993216329564E+000	6.783670436494305E-006
51		1.999994573072855E+000	5.426927145446570E-006
52		1.999995658464174E+000	4.341535825957976E-006
53		1.999996526775109E+000	3.473224891026305E-006
54		1.999997221422500E+000	2.778577500084367E-006
99		1.999999999878977E+000	1.210231914683391E-010
100		1.999999999903181E+000	9.681855317467125E-011

Figura 18.7 Parte do arquivo de resultados do projeto programa18b para os dados da Figura 18.5.

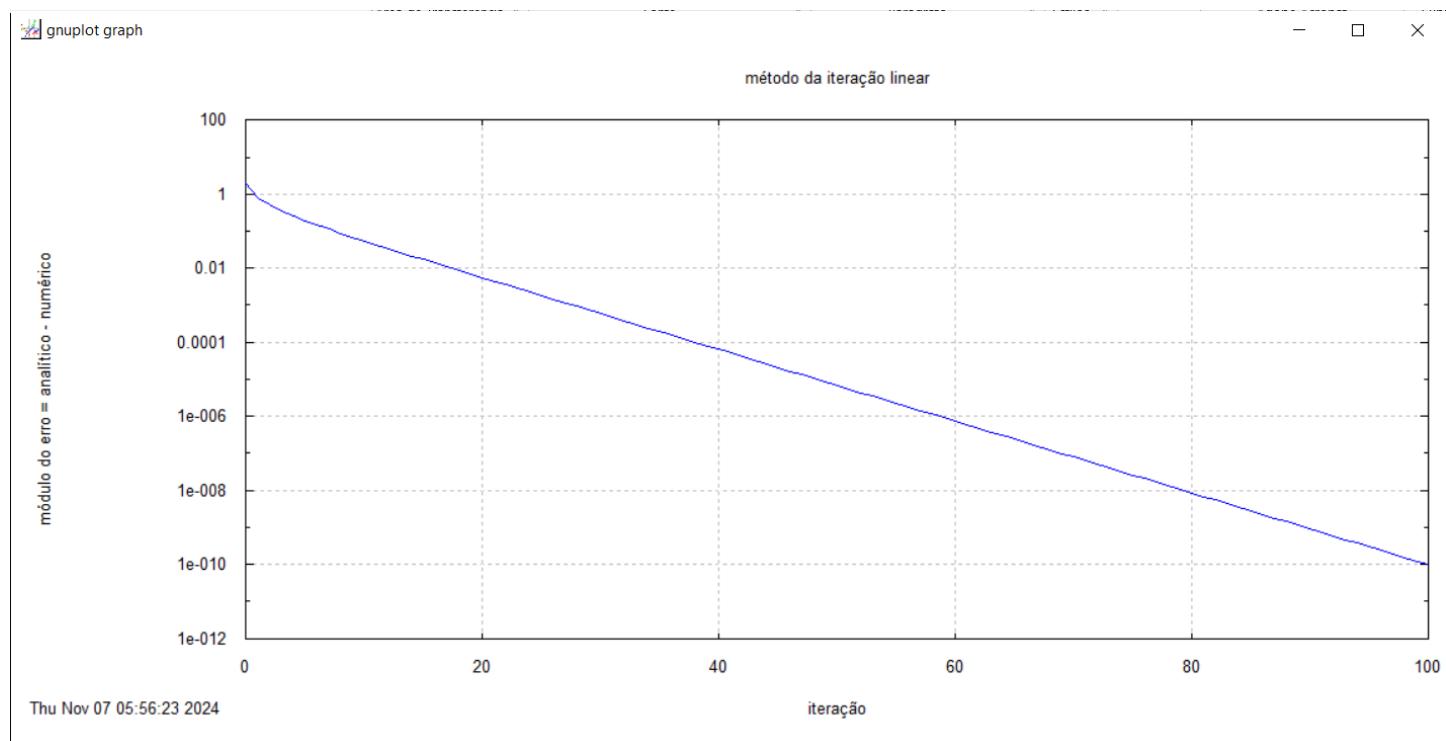


Figura 18.8. Gráfico com resultados do projeto programa18b para os comandos da Figura 18.6.

18.3 EXERCÍCIOS

Exercício 18.1

Adaptar o programa18a.f90 para incluir as seguintes opções de séries:

$$S = \frac{1}{1^2} + \frac{1}{2^3} + \frac{1}{3^4} + \dots = \sum_{I=1}^{\infty} \left[\frac{1}{I(I+1)} \right] = 1 \quad (18.5)$$

$$S = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots = \sum_{I=0}^{\infty} \left(\frac{x^I}{I!} \right) = e^x \quad (18.6)$$

$$S = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots = \sum_{I=0}^{\infty} \left[\frac{(-1)^I x^{2I+1}}{(2I+1)!} \right] = \operatorname{sen}(x) \quad (18.7)$$

$$S = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots = \sum_{I=0}^{\infty} \left[\frac{(-1)^I x^{2I}}{(2I)!} \right] = \operatorname{cos}(x) \quad (18.8)$$

Exercício 18.2

Adaptar o programa18a.f90 para usar precisão simples em todos os cálculos com números reais. Comparar os resultados com a versão original do programa18a.f90 na qual usa-se precisão dupla.

Exercício 18.3

Adaptar o programa18b.f90 para usar precisão simples em todos os cálculos com números reais. Comparar os resultados com a versão original do programa18b.f90 na qual usa-se precisão dupla.

Exercício 18.4

Adaptar o programa18b.f90 para resolver a seguinte equação:

$$x^2 - x - 6 = 0 \quad (18.9)$$