

## **OBJETIVOS DO CAPÍTULO**

- Calcular derivadas numéricas com quatro tipos de aproximações numéricas do método de diferenças finitas: UDS, DDS, CDS-2 e CDS-4.
- Calcular integrais numéricas com três tipos de aproximações numéricas: regras do retângulo, trapézio e Simpson.

Para inicializar as atividades deste capítulo, deve-se acessar o programa Fortran, no Windows, através de: **Start, Programs, Fortran PowerStation 4.0, Microsoft Developer Studio**

### **19.1 programa19a.f90**

- 1) **Objetivo do programa:** calcular derivadas numéricas com quatro tipos de aproximações numéricas do método de diferenças finitas: UDS, DDS, CDS-2 e CDS-4.
- 2) No Fortran, seguindo o **procedimento-padrão**, **criar um projeto** com o nome **projeto\_19**
- 3) **Acessar** o site [ftp://ftp.demec.ufpr.br/Disciplinas/Tm784/projeto\\_19](ftp://ftp.demec.ufpr.br/Disciplinas/Tm784/projeto_19)
- 4) **Clicar** com o botão do lado direito do mouse sobre o arquivo programa19a.f90
- 5) **Escolher** a opção Copiar para pasta... (Save Target As)
- 6) **Localizar** a pasta do projeto
- 7) **Clicar** no botão OK
- 8) **Repetir** os itens 3 a 7, acima, para os arquivos dados19a.txt, comandos19a.gnu, Wgnuplot.exe
- 9) No Fortran, seguindo o **procedimento-padrão**, **inserir** no projeto o programa-fonte **programa19a.f90**, mostrado na **Tabela 19.1**.
- 10) **Estudar o programa-principal** considerando os comentários do item 11, abaixo.
- 11) Comentários sobre o programa:
  - (a) O programa19a.f90 é composto pelo programa-principal e quatro sub-rotinas.
  - (b) A sub-rotina DADOS é usada para ler os dados do programa do arquivo dados19a.txt.
  - (c) A sub-rotina EXPONENCIAL é usada para calcular a derivada numérica da função  $f = e^x$  com quatro tipos de aproximações numéricas do método de diferenças finitas:

$$\left(\frac{df}{dx}\right)_x \approx \frac{f(x) - f(x-h)}{h} \quad (\text{UDS: um ponto a montante}) \quad (19.1)$$

$$\left(\frac{df}{dx}\right)_x \approx \frac{f(x+h) - f(x)}{h} \quad (\text{DDS: um ponto a jusante}) \quad (19.2)$$

$$\left(\frac{df}{dx}\right)_x \approx \frac{f(x+h) - f(x-h)}{2h} \quad (\text{CDS-2: diferença central de 2 pontos}) \quad (19.3)$$

(CDS-4: diferença central de 4 pontos)

$$\left(\frac{df}{dx}\right)_x \approx \frac{8f(x+h) - 8f(x-h) + f(x-2h) - f(x+2h)}{12h} \quad (19.4)$$

onde  $h = \Delta x$ , intervalo em  $x$  usado para aproximar cada derivada. Esta sub-rotina também calcula o erro de cada uma das quatro aproximações numéricas em relação à solução analítica da derivada que é  $e^x$ .

- (d) A sub-rotina POTENCIA é usada para calcular a derivada numérica da função  $f = x^5$  com os quatro tipos de aproximações numéricas das Eqs. (19.1) a (19.4). Esta sub-rotina também calcula o erro de cada uma das quatro aproximações numéricas em relação à solução analítica da derivada que é  $5x^4$ .
- (e) A sub-rotina GRAFICO abre o arquivo comandos19a.gnu e desce as primeiras doze linhas deste arquivo. Na linha seguinte, escreve um comentário para representar o título do gráfico de acordo com a função escolhida para calcular a derivada. E na linha seguinte, escreve outro comentário, que é o comando replot do Wgnuplot. Finalmente, esta sub-rotina executa o programa Wgnuplot para fazer o gráfico log-log do erro do cálculo da derivada da função escolhida, para cada uma das quatro aproximações numéricas, em função do número de refinamentos de  $h$  definido nos dados do programa.
- (f) O programa-principal: (1) define suas variáveis globais; (2) chama a sub-rotina DADOS; (3) cria o arquivo saida19a.txt; (4) 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); (5) dependendo do tipo de função escolhida pelo usuário, é chamada a sub-rotina adequada; (6) o Notepad abre o arquivo de saída; e, (7) é chamada a sub-rotina para fazer o gráfico dos erros com os comandos mostrados na Figura 19.1.

12) Executar **Build, Compile** para compilar o programa.

13) Gerar o programa-executável fazendo **Build, Build**.

14) Executar o programa através de **Build, Execute. Usar, os dados mostrados na Figura 19.2**.

Tabela 19.1 Programa19a.f90

```

program derivadas

use portlib

implicit none

integer :: i    ! número da aproximação de cada delta
integer :: L    ! número de refinamentos de hmax
integer :: ver  ! auxílio de System
integer :: tipo_funcao ! tipo de função a derivar

real*8  :: hmax ! maior delta
real*8  :: r    ! razão de refino dos deltas
real*8  :: X    ! coordenada para calcular a derivada
real*8  :: exato ! solução analítica da derivada

call dados ! lê os dados do programa

open(6,file="saida19a.txt") !

write(6,10)
10 format("#", t5,"h", t20,"E(UDS)", t35,"E(DDS)", t50,"E(CDS-2)", t65,"E(CDS-4)")

select case ( tipo_funcao )
  case ( 1 )
    call exponencial
  case ( 2 )
    call potencia
end select

close(6)

ver = system ("Notepad saida19a.txt")

call grafico

!-----

contains

!-----

```

```

subroutine dados

ver = system ("Notepad dados19a.txt")

open(5,file="dados19a.txt")
read(5,*) tipo_funcao
read(5,*) hmax
read(5,*) r
read(5,*) L
read(5,*) X
close(5)

end subroutine dados

!-----

subroutine exponencial

real*8 :: xw, h, fp, fw, fl_UDS, fl_DDS, xe, fe, fl_CDS2, &
        xww, xee, fww, fee, fl_CDS4

! calcula a derivada da exponencial de X

exato = dexp(X)

do i = 0, L

    h = hmax / ( r ** i )

    xw = X - h
    xe = X + h
    xww = X - 2*h
    xee = X + 2*h

    fp = dexp(x)
    fw = dexp(xw)
    fe = dexp(xe)
    fww = dexp(xww)
    fee = dexp(xee)

    fl_UDS = ( fp - fw ) / h
    fl_DDS = ( fe - fp ) / h
    fl_CDS2 = ( fe - fw ) / ( 2 * h)

```

```

fl_CDS4 = ( 8*fe - 8*fw + fww - fee ) / (12 * h)

write(6,10) h, dabs(exato-fl_UDS), dabs(exato-fl_DDS), &
           dabs(exato-fl_CDS2), dabs(exato-fl_CDS4)
10 format( 5(1pe15.5) )

end do

end subroutine exponencial

!-----

subroutine potencia

real*8 :: xw, h, fp, fw, fl_UDS, fl_DDS, xe, fe, fl_CDS2, &
         xww, xee, fww, fee, fl_CDS4

! calcula a derivada de X**5

exato = 5*X**4

do i = 0, L

    h = hmax / ( r ** i )

    xw = X - h
    xe = X + h
    xww = X - 2*h
    xee = X + 2*h

    fp = x**5
    fw = xw**5
    fe = xe**5
    fww = xww**5
    fee = xee**5

    fl_UDS = ( fp - fw ) / h
    fl_DDS = ( fe - fp ) / h
    fl_CDS2 = ( fe - fw ) / (2 * h)
    fl_CDS4 = ( 8*fe - 8*fw + fww - fee ) / (12 * h)

    write(6,10) h, dabs(exato-fl_UDS), dabs(exato-fl_DDS), &
               dabs(exato-fl_CDS2), dabs(exato-fl_CDS4)

```

```

10 format( 5(1pe15.5) )

end do

end subroutine potencia

!-----

subroutine grafico

integer k

open(9,file="comandos19a.gnu")
do k = 1, 12
  read(9,*)
end do

select case ( tipo_funcao )
  case ( 1 )
    write(9,*) "set title 'derivada de exponencial(x)'"
  case ( 2 )
    write(9,*) "set title 'derivada de X**5'"
end select

write(9,*) "replot"

close(9)

ver = system ("Wgnuplot comandos19a.gnu")

end subroutine grafico

!-----

end program derivadas

```

- 15) **Analisar os resultados** mostrados nas Figuras 19.3 e 19.4. Exceto nos maiores valores de  $h$ , as aproximações UDS e DDS apresentam quase os mesmos valores de erro. Conforme previsto pela teoria, as aproximações CDS-2 e CDS-4 apresentam os menores valores de erro para um mesmo  $h$ . Teoricamente, o erro de cada aproximação deveria sempre se reduzir com a redução de  $h$ , e deveria

atingir o valor nulo para  $h = 0$ . Porém, na prática, os erros de arredondamento impõem um limite mínimo ao erro de cada aproximação, como mostrado na Figura 19.4 para o CDS-2 e CDS-4.

- 16) **Executar** novamente o programa usando  $X = 10$  e **analisar** os novos resultados.
- 17) **Executar** novamente o programa usando  $X = 0.1$  e **analisar** os novos resultados.
- 18) **Executar** novamente o programa usando **tipo de função = 2** e **analisar** os novos resultados.
- 19) **Executar** novamente o programa usando outros dados e **analisar** os novos resultados.

```

comandos19a - Notepad
File Edit Format View Help
set grid
set time
set data style linespoints
set logscale x
set logscale y
set key right bottom
set xlabel 'h'
set ylabel 'módulo do erro = analítico - numérico'
plot 'saida19a.txt' using 1:2 title "ups"
replot 'saida19a.txt' using 1:3 title "DDS"
replot 'saida19a.txt' using 1:4 title "CDS2"
replot 'saida19a.txt' using 1:5 title "CDS4"
set title 'derivada de exponencial(x)'
replot

```

Figura 19.1 Arquivo de comandos para o aplicativo Wgnuplot do programa19a.f90.

```

dados19a - Notepad
File Edit Format View Help
1 = tipo de função (ver relação abaixo)
1.0d-0 = maior delta
2.d0 = razão de refino dos deltas
20 = número de refinamentos de hmax
1.0d0 = coordenada X para calcular a derivada

O programa calcula a derivada numérica das seguintes
funções:
1) exponencial de x
2) x**5

Arquivo de dados do programa19a.f90

```

Figura 19.2 Arquivo de dados do programa19a.f90.

```

saida19a - Notepad
File Edit Format View Help
# h E(UDS) E(DDS) E(CDS-2) E(CDS-4)
1. 0.00000E+00 1.00000E+00 1.95249E+00 4.76246E-01 1.02049E-01
5. 0.00000E-01 5.79161E-01 8.08533E-01 1.14686E-01 5.83411E-03
2. 5.00000E-01 3.13155E-01 3.69963E-01 2.84041E-02 3.56586E-04
1. 2.50000E-01 1.63030E-01 1.77198E-01 7.08439E-03 2.21626E-05
6. 2.50000E-02 8.32039E-02 8.67440E-02 1.77006E-03 1.38323E-06
3. 1.25000E-02 4.20342E-02 4.29191E-02 4.42450E-04 8.64219E-08
1. 5.62500E-02 2.11264E-02 2.13476E-02 1.10609E-04 5.40091E-09
7. 8.12500E-03 1.05907E-02 1.06460E-02 2.76519E-05 3.37531E-10
3. 9.06250E-03 5.30224E-03 5.31606E-03 6.91295E-06 2.10978E-11
1. 9.5313E-03 2.65284E-03 2.65630E-03 1.72824E-06 1.33493E-12
9. 7.6563E-04 1.32685E-03 1.32772E-03 4.32059E-07 2.36255E-13
4. 8.8281E-04 6.63535E-04 6.63751E-04 1.08014E-07 5.39124E-13
2. 4.4141E-04 3.31795E-04 3.31849E-04 2.70039E-08 3.70370E-13
1. 2.2070E-04 1.65904E-04 1.65918E-04 6.75125E-09 3.70370E-13
6. 1.10352E-05 8.29537E-05 8.29571E-05 1.68900E-09 1.58273E-12
3. 0.5176E-05 4.14773E-05 4.14781E-05 4.19344E-10 3.87423E-12
1. 5.2588E-05 2.07388E-05 2.07390E-05 1.06478E-10 2.18936E-12
7. 6.2939E-06 1.03694E-05 1.03695E-05 1.91664E-11 9.93738E-12
3. 8.1470E-06 5.18480E-06 5.18478E-06 9.93738E-12 1.96385E-11
1. 9.0735E-06 2.59246E-06 2.59244E-06 9.93738E-12 9.93738E-12
9. 5.3674E-07 1.29653E-06 1.29627E-06 1.26353E-10 1.65158E-10

```

Figura 19.3 Arquivo de resultados do programa19a.f90 para os dados da Fig. 19.2.

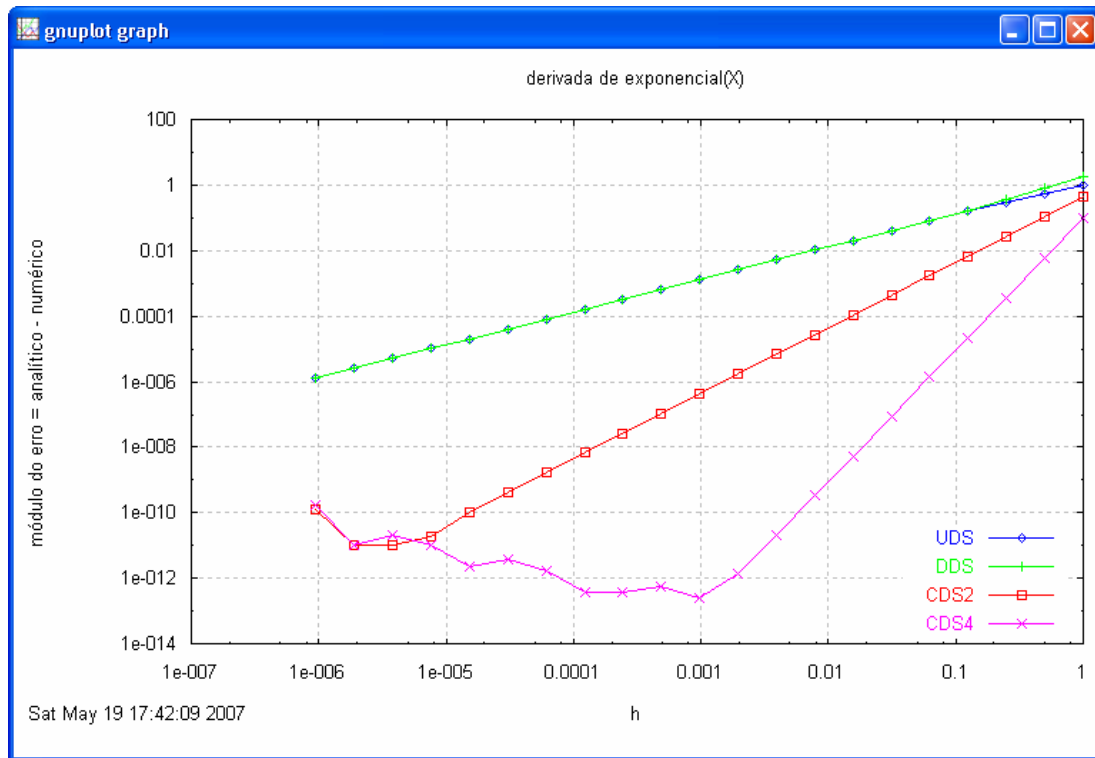


Figura 19.4. Gráfico com resultados do programa19a.f90 para os dados da Fig. 19.2.

## 19.2 programa19b.f90

- 1) Objetivo do programa: calcular integrais numéricas com três tipos de aproximações numéricas: regras do retângulo, trapézio e Simpson.
- 2) Nesta seção será usado o mesmo projeto da seção anterior. Portanto, deve-se executar o seguinte no Fortran:
  - a) **Clicar** sobre o nome do programa-fonte.
  - b) **Edit, Cut** para retirar o programa-fonte do projeto.
  - c) **Clicar** dentro do campo de edição de programa-fonte.
  - d) **File, Close**.
- 3) **Acessar** o site [ftp://ftp.demec.ufpr.br/Disciplinas/Tm784/projeto\\_19](ftp://ftp.demec.ufpr.br/Disciplinas/Tm784/projeto_19)
- 4) **Clicar** com o botão do lado direito do mouse sobre o arquivo programa19b.f90
- 5) **Escolher** a opção Copiar para pasta... (Save Target As)
- 6) **Localizar** a pasta do projeto
- 7) **Clicar** no botão OK
- 8) **Repetir** os itens 3 a 7, acima, para os arquivos dados19b.txt e comandos19b.gnu
- 9) No Fortran, seguindo o procedimento-padrão, **inserir** no projeto o programa-fonte **programa19b.f90**, mostrado na **Tabela 19.2**.
- 10) **Estudar o programa-principal** considerando os comentários do item 11, abaixo.



Tabela 19.2 Programa19b.f90

```

program integrais

use portlib

implicit none

integer :: i      ! número do refino dos elementos de integração
integer :: j      ! número do elemento de integração
integer :: Nmin ! número mínimo de elementos de integração
integer :: L      ! número de refinamentos de elementos de integração
integer :: r      ! razão de aumento de elementos de integração
integer :: ver    ! auxílio de System
integer :: tipo_funcao ! tipo de função a derivar

real*8 :: h      ! tamanho dos elementos de integração
real*8 :: exato ! solução analítica da derivada

call dados ! lê os dados do programa

open(6,file="saida19b.txt") !

write(6,10)
10 format("#", t5,"h", t20,"E(retângulo)", t35,"E(trapézio)", t50,"E(simpson)")

select case ( tipo_funcao )
  case ( 1 )
    call exponencial
  case ( 2 )
    call potencia
end select

close(6)

ver = system ("Notepad saida19b.txt")

call grafico

!-----

contains

!-----

```

```

subroutine dados

ver = system ("Notepad dados19b.txt")

open(5,file="dados19b.txt")
read(5,*) tipo_funcao
read(5,*) Nmin
read(5,*) r
read(5,*) L
close(5)

end subroutine dados

!-----

subroutine exponencial

real*8 :: xp, fp, ret, trap, xw, fw, simp, xe, fe

! calcula a integral da exponencial de X entre 0 e 1

exato = dexp(1.0d0) - 1

do i = 0, L

    h = 1.0d0 / ( Nmin * (r ** i) )

    ret = 0.0d0
    trap = 0.0d0

    do j = 1, Nmin*(r**i)

        xp = h*(j-0.5d0)
        fp = dexp(xp)
        ret = ret + fp * h

        xp = h*j
        xw = xp - h
        fw = dexp(xw)
        fp = dexp(xp)
        trap = trap + h*(fw+fp)/2

    end do

```

```

simp = 0.0d0

do j = 2, Nmin*(r**i), 2

    xp = h*(j-1)
    xw = xp - h
    xe = xp + h
    fw = dexp(xw)
    fe = dexp(xe)
    fp = dexp(xp)
    simp = simp + h*(fw+4*fp+fe)/3

end do

write(6,10) h, dabs(exato-ret), dabs(exato-trap), dabs(exato-simp)

10 format( 4(1pe15.5) )

end do

end subroutine exponencial

!-----

subroutine potencia

real*8 :: xp, fp, ret, trap, xw, fw, simp, xe, fe

! calcula a integral de X**5 entre 0 e 1

exato = 1 / 6.0d0

do i = 0, L

    h = 1.0d0 / ( Nmin * ( r ** i ) )

    ret = 0.0d0
    trap = 0.0d0

    do j = 1, Nmin*(r**i)

        xp = h*(j-0.5d0)

```

```

    fp = xp**5
    ret = ret + fp * h

    xp = h*j
    xw = xp - h
    fw = xw**5
    fp = xp**5
    trap = trap + h*(fw+fp)/2

end do

simp = 0.0d0

do j = 2, Nmin*(r**i), 2

    xp = h*(j-1)
    xw = xp - h
    xe = xp + h
    fw = xw**5
    fe = xe**5
    fp = xp**5
    simp = simp + h*(fw+4*fp+fe)/3

end do

write(6,10) h, dabs(exato-ret), dabs(exato-trap), dabs(exato-simp)

10 format( 4(1pe15.5) )

end do

end subroutine potencia

!-----

subroutine grafico

integer k

open(9,file="comandos19b.gnu")
do k = 1, 11
    read(9,*)
end do

```

```

select case ( tipo_funcao )
  case ( 1 )
    write(9,*) "set title 'integral de exponencial(X) entre 0 e 1'"
  case ( 2 )
    write(9,*) "set title 'integral de X**5 entre 0 e 1'"
end select

write(9,*) "replot"

close(9)

ver = system ("Wgnuplot comandos19b.gnu")

end subroutine grafico

!-----

end program integrais

```

11) Comentários sobre o programa:

- (a) O programa19b.f90 é composto pelo programa-principal e quatro sub-rotinas.
- (b) A sub-rotina DADOS é usada para ler os dados do programa do arquivo dados19b.txt.
- (c) A sub-rotina EXPONENCIAL é usada para calcular a integral numérica da função  $f = e^x$  com três tipos de aproximações numéricas:

$$\int_0^1 f dx \approx \sum_{k=1}^N [f(x_k)h] \quad (\text{regra do retângulo}) \quad (19.5)$$

onde  $h = \Delta x = 1/N$ ;  $N =$  número de intervalos de integração;  $x_k = h*(k-1/2)$ .

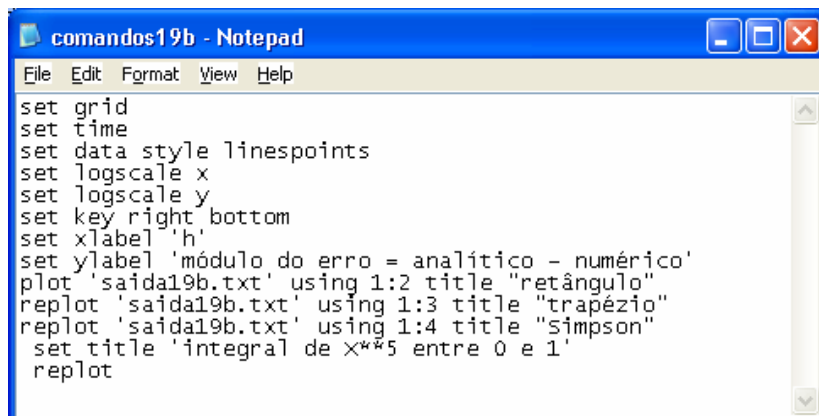
$$\int_0^1 f dx \approx \sum_{k=1}^N \left[ \frac{f(x_{k-1}) + f(x_k)}{2} h \right] \quad (\text{regra do trapézio}) \quad (19.6)$$

onde  $x_k = h*k$ .

$$\int_0^1 f dx \approx \sum_{k=2,4,\dots}^N \left[ \frac{f(x_{k-2}) + 4f(x_{k-1}) + f(x_k)}{3} h \right] \quad (\text{regra de Simpson}) \quad (19.7)$$

onde  $x_k = h*k$ . Esta sub-rotina também calcula o erro de cada uma das três aproximações numéricas em relação à solução analítica da integral que é  $e-1$ .

- (d) A sub-rotina POTENCIA é usada para calcular a integral numérica da função  $f = x^5$  com os três tipos de aproximações numéricas das Eqs. (19.5) a (19.7). Esta sub-rotina também calcula o erro de cada uma das três aproximações numéricas em relação à solução analítica da integral que é  $1/6$ .
- (e) A sub-rotina GRAFICO abre o arquivo comandos19b.gnu e desce as primeiras onze linhas deste arquivo. Na linha seguinte, escreve um comentário para representar o título do gráfico de acordo com a função escolhida para calcular a integral. E na linha seguinte, escreve outro comentário, que é o comando replot do Wgnuplot. Finalmente, esta sub-rotina executa o programa Wgnuplot para fazer o gráfico log-log do erro do cálculo da integral da função escolhida, para cada uma das três aproximações numéricas, em função do número ( $N$ ) de intervalos ( $h$ ) de integração definido com os dados do programa.
- (f) O programa-principal: (1) define suas variáveis globais; (2) chama a sub-rotina DADOS; (3) cria o arquivo saida19b.txt; (4) 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); (5) dependendo do tipo de função escolhida pelo usuário, é chamada a sub-rotina adequada; (6) o Notepad abre o arquivo de saída; e, (7) é chamada a sub-rotina para fazer o gráfico dos erros com os comandos mostrados na Figura 19.5.



```
File Edit Format View Help
set grid
set time
set data style linespoints
set logscale x
set logscale y
set key right bottom
set xlabel 'h'
set ylabel 'módulo do erro = analítico - numérico'
plot 'saida19b.txt' using 1:2 title "retângulo"
replot 'saida19b.txt' using 1:3 title "trapézio"
replot 'saida19b.txt' using 1:4 title "simpson"
set title 'integral de x**5 entre 0 e 1'
replot
```

Figura 19.5 Arquivo de comandos para o aplicativo Wgnuplot do programa19b.f90.

- 12) Executar **Build, Compile** para compilar o programa.
- 13) Gerar o programa-executável fazendo **Build, Build**.
- 14) Executar o programa através de **Build, Execute. Usar, os dados mostrados na Figura 19.6**.
- 15) **Analisar os resultados** mostrados nas Figuras 19.7 e 19.8. Conforme previsto pela teoria, as aproximações das regras do retângulo e trapézio devem ter erros similares, enquanto que o erro da

regra de Simpson deve ser o menor, para um mesmo  $h$ . Teoricamente, o erro de cada aproximação deveria sempre se reduzir com a redução de  $h$ , e deveria atingir o valor nulo para  $h = 0$ . Porém, na prática, os erros de arredondamento impõem um limite mínimo ao erro de cada aproximação, como mostrado na Figura 19.8 para a regra de Simpson.

- 16) **Executar** novamente o programa usando **tipo de função = 1** e **analisar** os novos resultados.
- 17) **Executar** novamente o programa usando outros dados e **analisar** os novos resultados.
- 18) No Fortran, para fechar o projeto atual, executar **File, Close Workspace**.

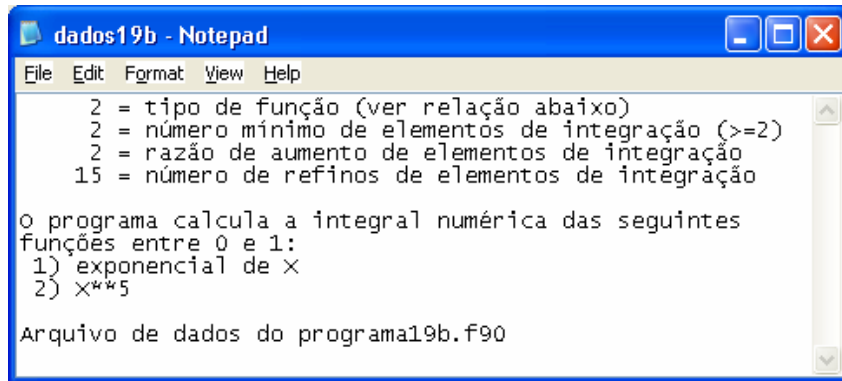


Figura 19.6 Arquivo de dados do programa19b.f90.

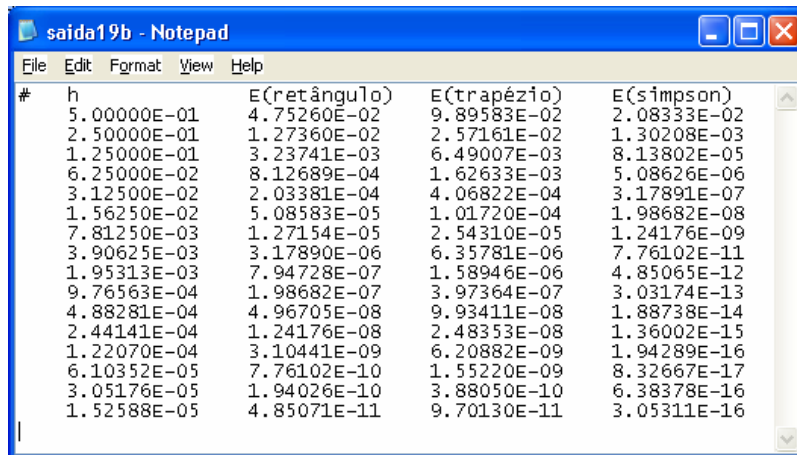


Figura 19.7 Arquivo de resultados do programa19b.f90 para os dados da Fig. 19.6.

## 19.3 EXERCÍCIOS

### Exercício 19.1

Acrescentar uma sub-rotina no programa19a.f90 para calcular a derivada de  $f = 3e^x$ . Incluir a nova opção de função no arquivo de dados e no programa. Apresentar arquivo com o erro dos quatro tipos de aproximações e respectivo gráfico.

### **Exercício 19.2**

Acrescentar uma sub-rotina no programa19a.f90 para calcular a derivada de  $f = x^4$ . Incluir a nova opção de função no arquivo de dados e no programa. Apresentar arquivo com o erro dos quatro tipos de aproximações e respectivo gráfico.

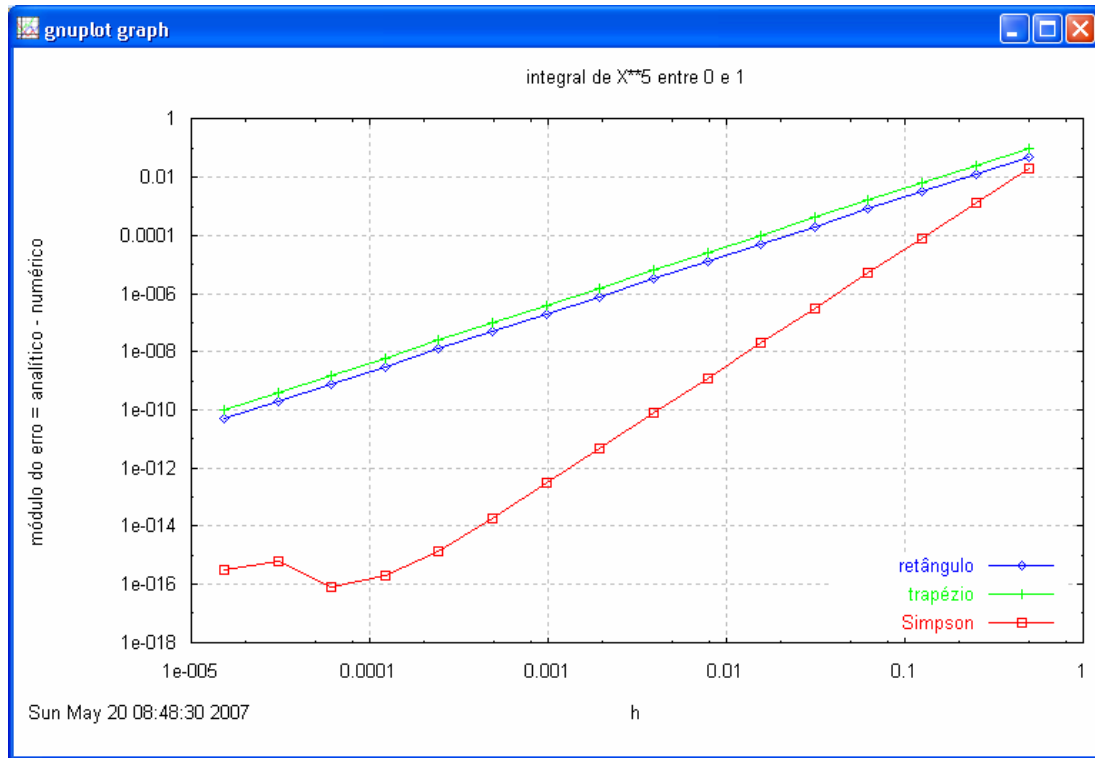


Figura 19.8. Gráfico com resultados do programa19b.f90 para os dados da Fig. 19.6.

### **Exercício 19.3**

Acrescentar uma sub-rotina no programa19b.f90 para calcular a integral de  $f = 3e^x$ . Incluir a nova opção de função no arquivo de dados e no programa. Apresentar arquivo com o erro dos três tipos de aproximações e respectivo gráfico.

### **Exercício 19.4**

Acrescentar uma sub-rotina no programa19b.f90 para calcular a integral de  $f = x^4$ . Incluir a nova opção de função no arquivo de dados e no programa. Apresentar arquivo com o erro dos três tipos de aproximações e respectivo gráfico.