

# Diferenciação e Integração numérica

Prof. Wagner Hugo Bonat

Curso de Especialização em  
Data Science & Big Data  
Universidade Federal do Paraná

4 de maio de 2018



# Conteúdo

# Conteúdo

## 1. Diferenciação numérica.

- 1.1 Aproximação da derivada por diferenças finitas;
- 1.2 Fórmulas de diferenças finitas usando expansão em série de Taylor;
- 1.3 Erros na diferenciação numérica;
- 1.4 Extrapolação de Richardson;
- 1.5 Diferenciação parcial numérica;
- 1.6 Funções residentes do R para diferenciação numérica.

## 2. Integração numérica.

- 2.1 Método Trapezoidal;
- 2.2 Método de Simpson 1/3;
- 2.3 Quadratura Gaussiana;
- 2.4 Quadratura Gaussiana Adaptativa;
- 2.5 Aproximação de Laplace;
- 2.6 Integração Monte Carlo;
- 2.7 Funções residentes do R para integração numérica.

# Diferenciação numérica

# Diferenciação numérica

- ▶ Derivada dá uma medida da taxa na qual a variável  $y$  muda devido a uma mudança na variável  $x$ .
- ▶ A função a ser diferenciada pode ser dada por uma função  $f(x)$ , ou apenas por um conjunto de pontos  $(y_i, x_i)$ .
- ▶ Quando devemos usar derivadas numéricas?
  1.  $f'(x)$  é difícil de obter analiticamente.
  2.  $f'(x)$  é caro para calcular computacionalmente.
  3. Quando a função é especificada apenas por um conjunto de pontos.
- ▶ Abordagens para a diferenciação numérica
  1. Aproximação por diferenças finitas;
  2. Aproximar a função por uma outra função de fácil derivação.

# Aproximação da derivada por diferenças finitas

- ▶ Derivada  $f'(x)$  de uma função  $f(x)$  no ponto  $x = a$  é definida como:

$$f'(a) = \lim_{x \rightarrow a} \frac{f(x) - f(a)}{x - a}.$$

- ▶ Derivada é o valor da inclinação da reta tangente à função em  $x = a$ .
- ▶ Escolhe-se um ponto  $x$  próximo a  $a$  e calcula-se a inclinação da reta que conecta os dois pontos.
- ▶ A precisão do cálculo aumenta quando  $x$  aproxima de  $a$ .
- ▶ Aproximação numérica: função será avaliada em diferentes pontos próximos a  $a$  para aproximar  $f'(a)$ .

# Aproximação da derivada por diferenças finitas

## ► Fórmulas para diferenciação numérica:

1. Diferença progressiva: Inclinação da reta que conecta os pontos  $(x_i, f(x_i))$  e  $(x_{i+1}, f(x_{i+1}))$ :

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}.$$

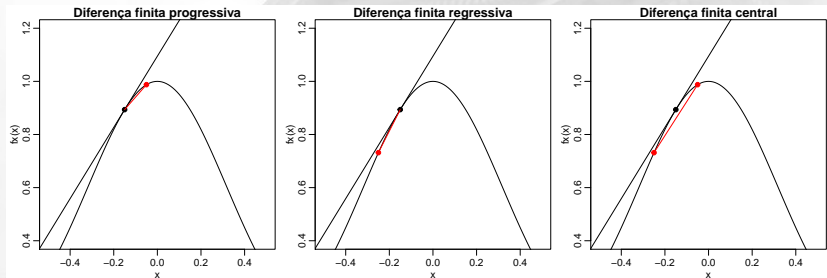
2. Diferença regressiva: Inclinação da reta que conecta os pontos  $(x_{i-1}, f(x_{i-1}))$  e  $(x_i, f(x_i))$ :

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}.$$

3. Diferença central: Inclinação da reta que conecta os pontos  $(x_{i-1}, f(x_{i-1}))$  e  $(x_{i+1}, f(x_{i+1}))$ :

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1}))}{x_{i+1} - x_{i-1}}.$$

# Ilustração: Derivada por diferenças finitas





# Implementação: Aproximação da derivada por diferenças finitas

## ► Diferença progressiva

```
dif_prog <- function(fx, x, h) {  
  df <- (fx(x + h) - fx(x))/(x + h - x)  
  return(df)  
}
```

## ► Diferença regressiva

```
dif_reg <- function(fx, x, h) {  
  df <- (fx(x) - fx(x - h))/(x - (x - h))  
  return(df)  
}
```

## ► Diferença central

```
dif_cen <- function(fx, x, h) {  
  df <- (fx(x + h) - fx(x - h))/(x + h - (x - h))  
  return(df)  
}
```

# Exemplo: Aproximação da derivada por diferenças finitas

- ▶ Considere  $f(x) = x^3$ , assim  $f'(x) = 3x^2$ .
- ▶ Numericamente temos

```
fx <- function(x) x^3
# Diferença progressiva
dif_prog(fx, x = 2, h = 0.001)

## [1] 12.006

# Diferença regressiva
dif_reg(fx, x = 2, h = 0.001)

## [1] 11.994

# Diferença central
dif_cen(fx, x = 2, h = 0.001)

## [1] 12

# Exata
3*2^2

## [1] 12
```

# Fórmulas de diferenças finitas usando expansão em série de Taylor

- ▶ As fórmulas anteriores podem ser deduzidas usando expansão em série de Taylor.
- ▶ O número de pontos para aproximar a derivada pode mudar.
- ▶ Vantagem da dedução por série de Taylor é que ela fornece uma estimativa do erro de truncamento.

# Diferença finita progressiva com dois pontos

- Aproximação de Taylor para o ponto  $x_{i+1}$

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f'''(x_i)}{3!}h^3 + \dots,$$

onde  $h = x_{i+1} - x_i$ .

- Fixando dois termos e deixando os outros termos como um resíduo, temos

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(\xi)}{2!}h^2.$$

- Resolvendo para  $f'(x_i)$ , temos

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} - \frac{f''(\xi)}{2!}h^2.$$

- Erro de truncamento,

$$-\frac{f''(\xi)}{2!}h^2 = O(h).$$

# Diferença finita regressiva com dois pontos

- Aproximação de Taylor para o ponto  $x_{i-1}$

$$f(x_{i-1}) = f(x_i) - f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f'''(x_i)}{3!}h^3 + \dots,$$

onde  $h = x_i - x_{i-1}$ .

- Fixando dois termos e deixando os outros termos como um resíduo, temos

$$f(x_{i-1}) = f(x_i) - f'(x_i)h + \frac{f''(\xi)}{2!}h^2.$$

- Resolvendo para  $f'(x_i)$ , temos

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h} + \frac{f''(\xi)}{2!}h^2.$$

- Erro de truncamento,

$$\frac{f''(\xi)}{2!}h^2 = O(h).$$

# Diferença finita central com dois pontos

- Aproximação de Taylor para o ponto  $x_{i+1}$

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f'''(\xi_1)}{3!}h^3,$$

onde  $\xi_1$  está entre  $x_i$  e  $x_{i+1}$ .

- Aproximação de Taylor para o ponto  $x_{i-1}$

$$f(x_{i-1}) = f(x_i) - f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f'''(\xi_2)}{3!}h^3,$$

onde  $\xi_2$  está entre  $x_{i-1}$  e  $x_i$ .

- Subtraindo as equações acima, temos

$$f(x_{i+1}) - f(x_{i-1}) = 2f'(x_i)h + \frac{f'''(\xi_1)}{3!}h^3 + \frac{f'''(\xi_2)}{3!}h^3.$$

- Resolvendo para  $f'(x_i)$ , temos

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1}))}{2h} + O(h^2).$$

# Diferença finita progressiva com três pontos

- ▶ Aproxima  $f'(x_i)$  avaliando a função no ponto e nos dois pontos seguintes  $x_{i+1}$  e  $x_{i+2}$ .
- ▶ Aproximação de Taylor em  $x_{i+1}$  e  $x_{i+2}$ ,

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f'''(\xi_1)}{3!}h^3, \quad (1)$$

$$f(x_{i+2}) = f(x_i) + f'(x_i)2h + \frac{f''(x_i)}{2!}(2h)^2 + \frac{f'''(\xi_2)}{3!}(2h)^3. \quad (2)$$

- ▶ Equações 1 e 2 são combinadas de forma que os termos com derivada segunda desapareçam.
- ▶ Multiplicando Eq. 1 por 4 e subtraindo Eq. 2, temos

$$4f(x_{i+1}) - f(x_{i+2}) = 3f(x_i) + 2f'(x_i)h + \frac{4f'''(\xi_1)}{3!}h^3 - \frac{f'''(\xi_2)}{3!}(2h)^3.$$

# Diferença finita com três pontos

- Resolvendo em  $f'(x_i)$ , temos

$$f'(x_i) = \frac{-3f(x_i) + 4f(x_{i+1}) - f(x_{i+2}))}{2h} + O(h).$$

- Diferença finita regressiva com três pontos

$$f'(x_i) = \frac{f(x_{i-2}) - 4f(x_{i-1}) + 3f(x_i))}{2h} + O(h).$$



# Fórmulas de diferenças finitas para a segunda derivada

- ▶ Usando as mesmas idéias podemos aproximar a derivada segunda de uma função qualquer por diferenças finitas.
- ▶ A derivação das fórmulas são idênticas, porém mais tediosas.
- ▶ Fórmula diferença central com três pontos para a derivada segunda

$$f''(x_i) = \frac{f(x_{i-1}) - 2f(x_i) + f(x_{i+1}))}{h^2} + O(h^2).$$

- ▶ Diferença central com quatro pontos

$$f''(x_i) = \frac{-f(x_{i-2}) + 16f(x_{i-1}) - 30f(x_i) + 16f(x_{i+1}) - f(x_{i+2}))}{12h^2} + O(h^4)$$

# Fórmulas de diferenças finitas para a segunda derivada

- ▶ Diferença progressiva com três pontos

$$f''(x_i) = \frac{f(x_i) - 2f(x_{i+1}) + f(x_{i+2}))}{h^2} + O(h).$$

- ▶ Diferença regressiva com três pontos

$$f''(x_i) = \frac{f(x_{i-2}) - 2f(x_{i-1}) + f(x_i))}{h^2} + O(h).$$

- ▶ Uma infinidade de fórmulas de várias ordens estão disponíveis.
- ▶ Fórmulas de diferenciação podem ser obtidas usando polinômios de Lagrange.

# Erros na diferenciação numérica

- ▶ Em todas as fórmulas o erro de truncamento é função de  $h$ .
- ▶  $h$  é o espaçamento entre os pontos, i.e.  $h = x_{i+1} - x_i$ .
- ▶ Fazendo  $h$  pequeno o erro de truncamento será pequeno.
- ▶ Em geral usa-se a precisão da máquina, algo como  $1e^{-16}$ .
- ▶ O erro de arredondamento depende da precisão finita de cada computador.
- ▶ Mesmo que  $h$  possa ser tão pequeno quanto desejado o erro de arredondamento pode crescer quando se diminui  $h$ .

# Extrapolação de Richardson

- ▶ Extrapolação de Richardson é usada para obter uma aproximação mais precisa da derivada a partir de duas aproximações menos precisas.
- ▶ Considere o valor  $D$  de uma derivada (desconhecida) calculada pela fórmula

$$D = D(h) + k_2 h^2 + k_4 h^4, \quad (3)$$

onde  $D(h)$  aproxima  $D$  e  $k_2$  e  $k_4$  são termos de erro.

- ▶ O uso da mesma fórmula, porém com espaçamento  $h/2$  resulta

$$D = D\left(\frac{h}{2}\right) + k_2 \left(\frac{h}{2}\right)^2 + k_4 \left(\frac{h}{2}\right)^4. \quad (4)$$

# Extrapolação de Richardson

- A Eq. 4 pode ser rescrita (após multiplicar por 4):

$$4D = 4D\left(\frac{h}{2}\right) + k_2 h^2 + k_4 \frac{h^4}{4}. \quad (5)$$

- Subtraindo 3 de 5 elimina os termos com  $h^2$  e fornece

$$3D = 4D\left(\frac{h}{2}\right) + D(h) - k_4 \frac{3h^4}{4}. \quad (6)$$

- Resolvendo 6, temos

$$D = \frac{1}{3} \left( 4D\left(\frac{h}{2}\right) + D(h) \right) - k_4 \frac{h^4}{4}. \quad (7)$$

# Extrapolação de Richardson

- ▶ O erro na Eq. 7 é agora  $O(h^4)$ . O valor de  $D$  é aproximado por

$$D = \frac{1}{3} \left( 4D\left(\frac{h}{2}\right) + D(h) \right) + O(h^4).$$

- ▶ A partir de duas aproximações de ordem inferiores, obtemos uma aproximação de  $O(h^4)$  mais precisa.
- ▶ Procedimento a partir de duas aproximações com erro  $O(h^4)$  mostra que

$$D = \frac{1}{15} \left( 16D\left(\frac{h}{2}\right) + D(h) \right) + O(h^6).$$

- ▶ Aproximação ainda mais precisa.

# Exemplo: Extrapolação de Richardson

- ▶ Calcule a derivada de  $f(x) = \frac{2^x}{x}$  no ponto  $x = 2$ .
- ▶ Solução exata:  $\frac{\log(2)2^x}{x} - \frac{2^x}{x^2}$ .
- ▶ Solução numérica usando diferença central

```
fx <- function(x) (2^x)/x
fpx <- function(x)(log(2)*(2^x))/x - (2^x)/x^2
erro <- fpx(x = 2)/dif_cen(fx = fx, x = 2, h = 0.2)
(erro-1)*100

## [1] 0.345544
```

- ▶ Extrapolação de Richardson

```
D2 <- dif_cen(fx = fx, x = 2, h = 0.2/2)
D <- dif_cen(fx = fx, x = 2, h = 0.2)
der <- (1/3)*( 4*D2 - D)
erro2 <- fpx(x = 2)/der
(erro2-1)*100

## [1] -0.001585268

c("Exata" = fpx(x = 2), "Richardson" = der,
  "Central" = dif_cen(fx = fx, x = 2, h = 0.2))

##      Exata Richardson      Central
## 0.3862944 0.3863005 0.3849641
```

# Derivadas parciais

- ▶ Para funções com muitas variáveis, a derivada parcial da função em relação a uma das variáveis representa a taxa de variação da função em relação a essa variável, mantendo as demais constantes.
- ▶ Assim, as fórmulas de diferenças finitas podem ser usadas no cálculo das derivadas parciais.
- ▶ As fórmulas são aplicadas em cada uma das variáveis, mantendo as outras fixas.
- ▶ A mesma ideia se aplica para derivadas de mais alta ordem.



# Implementação: Derivadas parciais

- ▶ Derive  $f(\beta_0, \beta_1) = \sum_{i=1}^n |y_i - (\beta_0 + \beta_1 x_i)|$ .
- ▶ Fórmula dois pontos central

```
dif_cen <- function(fx, pt, h, ...) {  
  df <- (fx(pt + h, ...) - fx(pt - h, ...))/( (pt + h) - (pt - h))  
  return(df)  
}
```

- ▶ Função a ser diferenciada

```
fx <- function(par, y, x1) {sum ( abs( y - (par[1] + par[2]*x1)) )}
```

- ▶ Gradiente usando diferenças finita.

```
grad_fx <- function(fx, par, h, ...) {  
  fbeta0 <- function(beta0, beta1, y, x) fx(par = c(beta0, beta1), y = y, x = x)  
  fbeta1 <- function(beta1, beta0, y, x) fx(par = c(beta0, beta1), y = y, x = x)  
  db0 <- dif_cen(fx = fbeta0, pt = par[1], h = h, beta1 = par[2], y = y, x = x)  
  db1 <- dif_cen(fx = fbeta1, pt = par[2], h = h, beta0 = par[1], y = y, x = x)  
  return(c(db0, db1))  
}
```

# Exemplo: Derivadas parciais

- ▶ Simulando  $y_i$ 's e  $x_i$ 's.

```
set.seed(123)
x <- runif(100)
y <- rnorm(100, mean = 2 + 3*x, sd = 1)
```

- ▶ Gradiente numérico

```
grad_fx(fx = fx, par = c(2, 3), h = 0.001, y = y, x1 = x)

## [1] 6.000000 2.272805
```

- ▶ Gradiente analítico

```
c(sum(((y - 2 - 3*x)/abs(y - 2 - 3*x))*(-1)),
  sum(((y - 2 - 3*x)/abs(y - 2 - 3*x))*(-x)))

## [1] 6.000000 2.272805
```

# Uso de funções residentes do R para diferenciação numérica.

- ▶ Pacote `numDeriv` implementa derivadas por diferença finita.
- ▶ Gradiente

```
require(numDeriv)
args(grad)

## function (func, x, method = "Richardson", method.args = list(),
##      ...)
## NULL
```

- ▶ Hessiano

```
args(hessian)

## function (func, x, method = "Richardson", method.args = list(),
##      ...)
## NULL
```

- ▶ Exemplo de aplicação

```
grad(func = fx, x = c(2, 3), y = y, x1 = x)

## [1] 6.000000 2.272805

hessian(func = fx, x = c(2, 3), y = y, x1 = x)

##           [,1]      [,2]
## [1,] 58.91271 29.53710
## [2,] 29.53710 48.86648
```



# Integração numérica

# Integração numérica

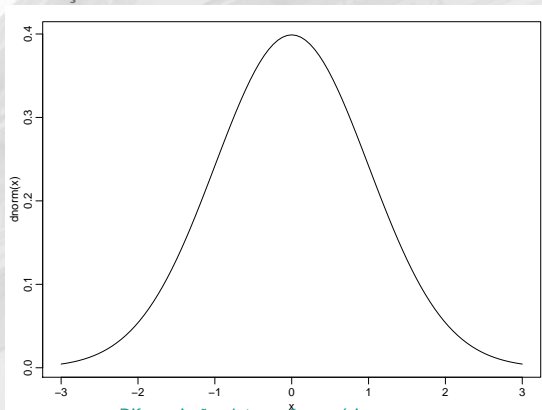
- ▶ Integrais aparecem com frequência em cálculo de probabilidades.
- ▶ A probabilidade de um evento é a área abaixo de uma curva.
- ▶ Em modelos complicados a integral pode não ter solução analítica.
- ▶ Os métodos de integração numérica, podem ser divididos em três grupos:
  1. Métodos baseados em soma finita.
  2. Aproximar a função por uma outra de fácil integração.
  3. Estimar o valor da integral.

# Integração numérica

- Considere a distribuição Gaussiana com função densidade probabilidade dada por

$$f(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x - \mu)^2\right).$$

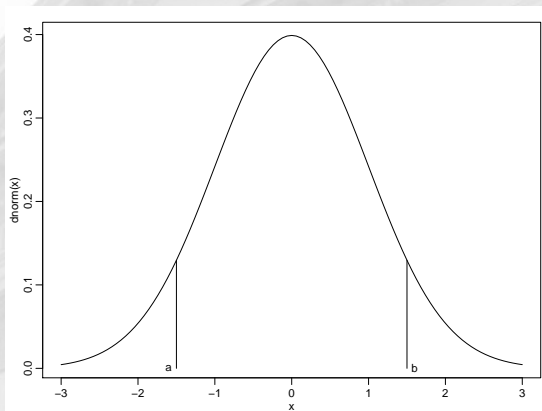
- Gráfico da função



# Integração numérica

- O cálculo de uma probabilidade qualquer baseado nesta distribuição é dado pela seguinte integral

$$P(a < x < b) = \int_a^b \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x - \mu)^2\right) dx.$$



# Integração numérica: Método Trapezoidal

- ▶ Usa uma função linear para aproximar o integrando.
- ▶ O integrando pode ser aproximado por Série de Taylor

$$f(x) \approx f(a) + (x - a) \left[ \frac{f(b) - f(a)}{b - a} \right].$$

- ▶ Integrando analiticamente essa aproximação, tem-se

$$\begin{aligned} I(f) &\approx \int_a^b f(a) + (x - a) \left[ \frac{f(b) - f(a)}{b - a} \right] dx \\ &= f(a)(b - a) + \frac{1}{2}[f(b) - f(a)](b - a). \end{aligned}$$

- ▶ Simplificando, obtém-se

$$I(f) \approx \frac{[f(a) + f(b)]}{2}(b - a).$$



# Implementação: Método Trapezoidal

## ► Função em R.

```
trapezio <- function(integrando, a, b, ...){  
  Int <- ((integrando(a, ...) + integrando(b, ...))/2)*(b-a)  
  return(Int)  
}
```

## ► Exemplo: Calcule $\int_2^3 x^2 dx$

```
fx <- function(x) x^2  
trapezio(integrando = fx, a = 2, b = 3)  
  
## [1] 6.5
```

## ► Solução exata: $\int_2^3 x^2 dx = \frac{x^3}{3} \Big|_2^3 = \frac{3^3}{3} - \frac{2^3}{3} = 6.34$

# Integração numérica: Método de Simpson 1/3

- ▶ Aproxima o integrando por um polinômio de segunda ordem.
- ▶ Pontos finais  $x_1 = a$ ,  $x_3 = b$ , e o ponto central,  $x_2 = (a + b)/2$ .
- ▶ O polinômio pode ser escrito na forma:

$$p(x) = \alpha + \beta(x - x_1) + \lambda(x - x_1)(x - x_2) \quad (8)$$

onde  $\alpha$ ,  $\beta$  e  $\lambda$  são constantes desconhecidas.

- ▶ Impomos a condição que o polinômio deve passar por todos os pontos,  $p(x_1) = f(x_1)$ ,  $p(x_2) = f(x_2)$  e  $p(x_3) = f(x_3)$ .

# Integração numérica: Método de Simpson 1/3

- ▶ Isso resulta em:

$$\alpha = f(x_1), \quad \beta = [f(x_2) - f(x_1)]/(x_2 - x_1) \quad \text{e}$$

$$\lambda = \frac{f(x_3) - 2f(x_2) + f(x_1)}{2(h)^2}$$

onde  $h = (b - a)/2$ .

- ▶ Substituindo em 8 e integrando  $p(x)$ , obtém-se

$$I = \int_a^b f(x)dx \approx \int_a^b p(x)dx = \frac{h}{3} \left[ f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right].$$

# Implementação: Método de Simpson 1/3

- A integral é facilmente calculada com apenas três avaliações da função.

```
simpson <- function(integrando, a, b, ...){  
  h <- (b-a)/2  
  x2 <- (a+b)/2  
  integral <- (h/3)*(integrando(a,...) +  
                 4*integrando(x2, ...) + integrando(b, ...))  
  return(integral)  
}
```

- Exemplo: Calcule  $\int_2^3 x^2 dx$

```
fx <- function(x) x^2  
simpson(integrando = fx, a = 2, b = 3)  
## [1] 6.333333
```

- Solução exata:  $\int_2^3 x^2 dx = \frac{x^3}{3} \Big|_2^3 = \frac{3^3}{3} - \frac{2^3}{3} = 6.34$

# Quadratura de Gauss

- ▶ Método trapezoidal e Simpson são muito simples.
- ▶ Aproximam o integrando por um polinômio de fácil integração.
- ▶ Resolvem a integral aproximada.
- ▶ Pontos são igualmente espaçados.
- ▶ Simples e intuitivos, porém de difícil generalização.
- ▶ Quadratura Gaussiana é um dos métodos mais populares de integração numérica.
- ▶ Aplicações: Modelos mistos não-lineares, análise de dados longitudinais, medidas repetidas, modelos lineares generalizados mistos, etc.

# Quadratura de Gauss

- Forma geral da quadratura de Gauss:

$$\int_a^b f(x)dx \approx \sum_{i=1}^n C_i f(x_i), \quad (9)$$

onde  $C_i$  são pesos e  $x_i$  são os pontos de Gauss em  $[a, b]$ .

- Exemplo 1: Para  $n = 2$  a Eq. 9 tem a forma:

$$\int_a^b f(x)dx \approx C_1 f(x_1) + C_2 f(x_2).$$

- Exemplo 2: Para  $n = 3$  a Eq. 9 tem a forma:

$$\int_a^b f(x)dx \approx C_1 f(x_1) + C_2 f(x_2) + C_3 f(x_3).$$

# Quadratura de Gauss

- ▶ Coeficientes  $C_i$  e a localização dos pontos  $x_i$  depende dos valores de  $n$ ,  $a$  e  $b$ .
- ▶  $C_i$  e  $x_i$  são determinados de forma que o lado direito da Eq. 9 seja igual ao lado esquerdo para funções  $f(x)$  especificadas.
- ▶ A especificação de  $f(x)$  vai depender do domínio de integração.
- ▶ Diferentes domínios levam a diferentes variações do método.
- ▶ Domínios comuns:
  1. Gauss-Legendre, Gauss-Jacobi e Gauss-Chebyshev

$$\int_a^b f(x) dx.$$

2. Gauss-Laguerre

$$\int_0^{\infty} f(x) e^{-x} dx.$$

3. Gauss-Hermite

$$\int_{-\infty}^{\infty} f(x) e^{-x^2} dx.$$

# Quadratura de Gauss

- ▶ No domínio  $[-1, 1]$  a forma da quadratura de Gauss é

$$\int_{-1}^1 f(x)dx \approx \sum_{i=1}^n C_i f(x_i).$$

- ▶  $C_i$  e  $x_i$  são determinados fazendo com que a Eq. 9 seja exata quando  $f(x) = 1, x, x^2, x^3, \dots$
- ▶ O número de casos depende do valor de  $n$ .
- ▶ Para  $n = 2$ , tem-se

$$\int_{-1}^1 f(x)dx \approx C_1 f(x_1) + C_2 f(x_2). \quad (10)$$



# Quadratura de Gauss-Legendre

- ▶ As quatro constantes  $C_1, C_2, x_1$  e  $x_2$  são determinadas fazendo Eq. 10 exata quando aplicada aos quatro casos:

$$\text{Caso 1} \quad f(x) = 1 \quad \int_{-1}^1 1dx = 2 = C_1 + C_2$$

$$\text{Caso 2} \quad f(x) = x \quad \int_{-1}^1 xdx = 0 = C_1x_1 + C_2x_2$$

$$\text{Caso 3} \quad f(x) = x^2 \quad \int_{-1}^1 x^2dx = \frac{2}{3} = C_1x_1^2 + C_2x_2^2$$

$$\text{Caso 4} \quad f(x) = x^3 \quad \int_{-1}^1 x^3dx = 0 = C_1x_1^3 + C_2x_2^3$$

- ▶ Sistema não-linear de quatro equações e quatro incógnitas.
- ▶ Podem existir múltiplas soluções.
- ▶ Uma solução particular é obtido por impor que  $x_1 = -x_2$ .
- ▶ Pela equação 2, implica que  $C_1 = C_2$  e a solução é

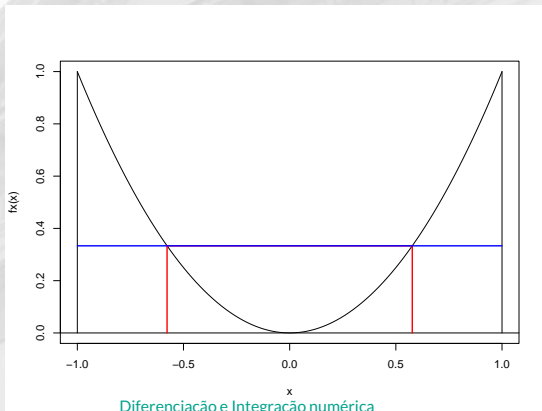
$$C_1 = 1, \quad C_2 = 1, \quad x_1 = -\frac{1}{\sqrt{3}} \quad \text{e} \quad x_2 = \frac{1}{\sqrt{3}}.$$

## Exemplo: Quadratura de Gauss-Legendre

- ▶ Calcule  $\int_{-1}^1 x^2 dx$ .
- ▶ Usando Gauss-Legendre com dois pontos, tem-se

$$\int_{-1}^1 x^2 dx = 1 \left( \frac{-1}{\sqrt{3}} \right)^2 + 1 \left( \frac{1}{\sqrt{3}} \right)^2 = \frac{2}{3}.$$

- ▶ Ilustração



## Exemplo: Quadratura de Gauss-Legendre

- ▶ Quando  $f(x)$  é uma função diferente de  $f(x) = 1, f(x) = x, f(x) = x^2$  ou  $f(x) = x^3$  ou qualquer combinação linear destas a aproximação é exata.
- ▶ Caso contrário o procedimento fornece uma aproximação.
- ▶ Exemplo:  $f(x) = \cos(x)$  valor exato é

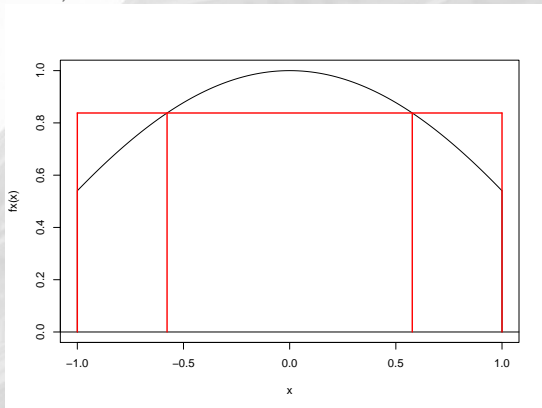
$$\int_{-1}^1 \cos(x) dx = \sin(x) \Big|_{-1}^1 = \sin(1) - \sin(-1) = 1.682841.$$

- ▶ Usando Quadratura de Gauss-Legendre com  $n = 2$ , tem-se

$$\int_{-1}^1 \cos(x) dx \approx \cos(-1/\sqrt{3}) + \cos(1/\sqrt{3}) = 1.675823.$$

# Exemplo: Quadratura de Gauss-Legendre

- Graficamente, tem-se



# Quadratura de Gauss-Legendre

- ▶ O número de pontos de integração controla a precisão da aproximação.
- ▶ Em R o pacote pracma fornece os pesos e pontos de integração.
- ▶ Exemplo

```
require(pracma)
gaussLegendre(n = 2, a = -1, b = 1)

## $x
## [1] -0.5773503  0.5773503
##
## $w
## [1] 1 1
```

- ▶ Baseado nos pontos e pesos de integração é fácil construir funções genéricas para integração numérica.

# Implementação: Quadratura de Gauss-Legendre

## ► Função genérica

```
gauss_legendre <- function(integrando, n.pontos, a, b, ...){  
  pontos <- gaussLegendre(n.pontos, a = a, b = b)  
  integral <- sum(pontos$w*integrando(pontos$x,...))  
  return(integral)  
}
```

## ► Exemplo: $\int_{-1}^1 \cos(x)dx$ .

```
# n = 2  
gauss_legendre(integrando = cos, n.pontos = 2, a = -1, b = 1)  
## [1] 1.675824  
  
# n = 10  
gauss_legendre(integrando = cos, n.pontos = 10, a = -1, b = 1)  
## [1] 1.682942
```

# Quadratura de Gauss-Laguerre

- ▶ Gauss-Laguerre resolve integrais do tipo:

$$\int_0^{\infty} e^{-x} f(x) dx.$$

- ▶ Integral é aproximada por uma soma ponderada.

$$\int_0^{\infty} e^{-x} f(x) dx \approx \sum_{i=1}^n w_i f(x_i)$$

- ▶ Função é avaliada nos pontos de Gauss e pesos de integração.
- ▶ Os pesos e pontos de integração são obtidos de forma similar ao caso de Gauss-Legendre, porém baseado no polinômio de Laguerre.

# Implementação: Quadratura de Gauss-Laguerre

## ► Função genérica para integração de Gauss-Laguerre

```
gauss_laguerre <- function(integrando, n.pontos, ...){  
  pontos <- gaussLaguerre(n.pontos)  
  integral <- sum(pontos$w*integrando(pontos$x,...)  
                /exp(-pontos$x))  
  return(integral)  
}
```

## ► Exemplo: $\int_0^{\infty} \lambda \exp(-\lambda x) dx$ .

```
fx <- function(x, lambda) lambda*exp(-lambda*x)  
# n = 2  
gauss_laguerre(integrando = fx, n.pontos = 2, lambda = 10)  
## [1] 0.04381233  
  
# n = 10  
gauss_laguerre(integrando = fx, n.pontos = 10, lambda = 10)  
## [1] 0.8981046  
  
# n = 100  
gauss_laguerre(integrando = fx, n.pontos = 100, lambda = 10)  
## [1] 1
```



# Quadratura de Gauss-Hermite

- ▶ Gauss-Hermite resolve integrais do tipo:

$$\int_{-\infty}^{\infty} e^{-x^2} f(x) dx.$$

- ▶ Integral é aproximada por uma soma ponderada.

$$\int_{-\infty}^{\infty} e^{-x^2} f(x) dx \approx \sum_{i=1}^n w_i f(x_i)$$

- ▶ Função é avaliada nos pontos de Gauss e pesos de integração.
- ▶ Os pesos e pontos de integração são obtidos de forma similar ao caso de Gauss-Legendre, porém baseado no polinômio de Hermite.

# Implementação: Quadratura de Gauss-Hermite

## ► Função genérica para integração de Gauss-Hermite

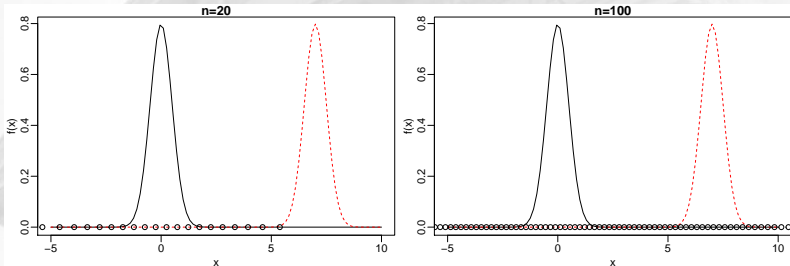
```
gauss_hermite <- function(integrando, n.pontos, ...){  
  pontos <- gaussHermite(n.pontos)  
  integral <- sum(pontos$x*integrando(pontos$x,...)  
                /exp(-pontos$x^2))  
  return(integral)  
}
```

## ► Exemplo: $\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right) dy.$

```
# n = 2  
gauss_hermite(integrando = dnorm, n.pontos = 2)  
## [1] 0.9079431  
  
# n = 10  
gauss_hermite(integrando = dnorm, n.pontos = 10)  
## [1] 0.9999876  
  
# n = 100  
gauss_hermite(integrando = dnorm, n.pontos = 100)  
## [1] 1
```

# Limitações: Quadratura de Gauss

- ▶ Quadratura de Gauss apresenta duas grandes limitações:
  1. Os pontos são escolhidos baseado no polinômio escolhido, ignorando a função a ser integrada.
  2. Número de pontos necessários para a integração cresce como uma potência da dimensão da integral. 20 pontos em uma dimensão demanda  $20^2 = 400$  pontos em duas dimensões.



- ▶ Espalhar os pontos de forma mais inteligente diminui o número de pontos necessários.

# Quadratura de Gauss-Hermite Adaptativa

- ▶ Os pontos de integração são centrados e escalonados como se  $f(x)e^{-x^2}$  fosse a distribuição Gaussiana.
- ▶ A média da aproximação Gaussiana será a moda  $\hat{x}$  de  $\ln[f(x)e^{-x^2}]$ .
- ▶ A variância da aproximação Gaussiana será

$$\left[ -\frac{\partial^2}{\partial x^2} \ln[f(x)e^{-x^2}] \Big|_{x=\hat{x}} \right]^{-1}.$$

- ▶ Novos pontos de integração adaptados serão dados por

$$x_i^+ = \hat{x} + \left[ -\frac{\partial^2}{\partial x^2} \ln[f(x)e^{-x^2}] \Big|_{x=\hat{x}} \right]^{-1/2} x_i$$

com correspondentes pesos,

$$w_i^+ = \left[ -\frac{\partial^2}{\partial x^2} \ln[f(x)e^{-x^2}] \Big|_{x=\hat{x}} \right]^{-1/2} \frac{e^{x_i^+}}{e^{-x_i}} w_i.$$

# Quadratura de Gauss-Hermite Adaptativa

- ▶ Como antes, a integral é aproximada por

$$\int f(x)e^{-x^2}dx \approx \sum_{i=1}^n w_i^+ f(x_i^+).$$

- ▶ Problema!! Como encontrar a moda e o hessiano de  $\ln[f(x)e^{-x^2}]$ ?
- ▶ Analiticamente ou numericamente.
- ▶ Caso especial Gauss-Hermite Adaptativa com  $n = 1 \rightarrow$  Aproximação de Laplace.

# Aproximação de Laplace

- Denote  $f(x)e^{-x^2}$  por  $Q(x)$ .
- Como  $n = 1$ ,  $x_1 = 0$  e  $w_1 = 1$ , obtemos  $x_1^+ = \hat{x}$ .
- Pesos de integração são iguais a

$$w_1^+ = |Q''(\hat{x})|^{-1/2} \frac{e^{-\hat{x}}}{e^{-0}} = (2\pi)^{n/2} |Q''(\hat{x})|^{-1/2} \frac{e^{Q(\hat{x})}}{f(\hat{x})}.$$

- Assim, a aproximação fica dada por

$$\begin{aligned} \int f(x)e^{-x^2} dx &= \int e^{Q(x)} dx \\ &\approx w_1^+ f(x_1^+) = (2\pi)^{n/2} |Q''(\hat{x})|^{-1/2} e^{Q(\hat{x})}. \end{aligned}$$

# Implementação: Aproximação de Laplace

- Função `optim()` encontra o máximo e o hessiano de  $Q(x)$ .

```
laplace <- function(funcao, otimizador, n.dim, ...){  
  integral <- -999999  
  inicial <- rep(0, n.dim)  
  temp <- try(optim(inicial, funcao, ..., method=otimizador,  
                    hessian=TRUE, control=list(fnscale=-1)))  
  if(class(temp) != "try-error"){  
    integral <- exp(temp$value) * (exp((n.dim/2)*log(2*pi) -  
                                     0.5*determinant(-temp$hessian)$modulus))}  
  return(integral)  
}
```

- Exemplo:  $\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right) dy$ .

```
laplace(dnorm, otimizador = "BFGS", n.dim = 1, log = TRUE)  
## [1] 1  
## attr("logarithm")  
## [1] TRUE
```

# Integração Monte Carlo

- ▶ Método simples e geral para resolver integrais.
- ▶ Objetivo: estimar o valor da integral de uma função  $f(x)$  em algum domínio  $D$  qualquer, ou seja,

$$I = \int_D f(x) dx \quad (11)$$

- ▶ Seja  $p(x)$  uma fdp cujo domínio coincide com  $D$ .
- ▶ Então, a integral em 11 é equivalente a

$$I = \int_D \frac{f(x)}{p(x)} p(x) dx.$$

- ▶ A integral corresponde a  $E\left(\frac{f(x)}{p(x)}\right)$ .



# Algoritmo: Integração Monte Carlo

## ► Algoritmo: Integração Monte Carlo

1. Gere números aleatórios de  $p(x)$ ;
2. Calcule  $m_i = f(x_i)/p(x_i)$  para cada amostra,  $i = 1, \dots, n$ .
3. Calcule a média  $\sum_{i=1}^n \frac{m_i}{n}$ .

## ► Implementação para funções com $D = \mathbb{R}$ .

```
monte.carlo <- function(funcao, n.pontos, ...) {  
  pontos <- rnorm(n.pontos)  
  norma <- dnorm(pontos)  
  integral <- mean(funcao(pontos,...)/norma)  
  return(integral)  
}
```

## ► Exemplo: $\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right) dy$ .

```
# Integrando a Normal padrão  
monte.carlo(funcao = dnorm, n.pontos = 1000)  
  
## [1] 1  
  
# Integrando distribuição t com df = 30  
monte.carlo(funcao = dt, n.pontos = 1000, df = 30)  
  
## [1] 0.9982789
```

# Função do R para integração numérica

- ▶ Função `integrate()` implementa integração numérica usando quadratura adaptativa.
- ▶ O algoritmo depende do tipo da função.

```
args(integrate)

## function (f, lower, upper, ..., subdivisions = 100L, rel.tol = .Machine$double.eps^0.25,
##      abs.tol = rel.tol, stop.on.error = TRUE, keep.xy = FALSE,
##      aux = NULL)
## NULL
```

- ▶ Exemplo 1:  $\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right) dy$ .

```
integrate(f = dnorm, lower = -Inf, upper = Inf)

## 1 with absolute error < 9.4e-05
```

- ▶ Exemplo 2:  $\int_{-1}^1 x^2 dx$ .

```
fx <- function(x)x^2
integrate(f = fx, lower = -1, upper = 1)

## 0.6666667 with absolute error < 7.4e-15
```

# Discussão

- ▶ Integração numérica aparece com frequência em modelos mistos não Gaussianos.
- ▶ Método de Gauss-Hermite (GH) é muito popular.
- ▶ GH é limitado a integrais de baixa dimensão  $n < 10$ .
- ▶ GH é computacionalmente caro.
- ▶ GH adaptativo é mais eficiente, porém ainda limitado.
- ▶ Aproximação de Laplace excelente para integrando simétrico.
- ▶ Laplace resolve problemas em alta dimensão.
- ▶ Pode ser inacurada para integrando assimétricos.
- ▶ Integração Monte Carlo depende da escolha da *proposal*.
- ▶ Computacionalmente intensivo.
- ▶ Resolve problemas de alta dimensão a um alto custo computacional.