Cálculo Numérico / Métodos Numéricos

Solução de equações polinomiais
Briot-Ruffini-Horner

26 Sep 2008 . 15:49

Métodos iterativos

- Há vários métodos iterativos para resolver

$$f(x) = 0 \qquad (1)$$

  - Bissecção
  - Posição Falsa
  - Iterativo linear (ponto fixo)
  - Newton

- Obviamente, todos podem ser usado para resolver equações polinomiais:

$$P(x) = a_n x_n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

Equações polinomiais

- O que vamos aprender é que para equações polinomiais, existem métodos mais eficientes:

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$$

- Em um primeiro momento, calculemos quanto gastamos computacionalmente para calcular P(x) para um dado x:

$x^n = x \cdot x^{n-1}$ precisa de (N-1) multiplicações

Há N somas

N multiplicações dos coeficientes pelos termos em x

Equações polinomiais

$$P(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

$a_4x^4 = a_4 \cdot x \cdot x \cdot x \cdot x$ (4 multiplicações)

$+$

$a_3x^3 = a_3 \cdot x \cdot x \cdot x$ (3 multiplicações)

$+$

$a_2x^2 = a_2 \cdot x \cdot x$ (2 multiplicações)

$+$

$a_1x = a_1 \cdot x$ (1 multiplicação)

$+$

a_0

(4 somas)

reescrevendo

$$P(x) = (((a_4 \cdot x + a_3) \cdot x + a_2) \cdot x + a_1) \cdot x + a_0$$

4 multiplicações e 4 somas

Equações polinomiais

no caso geral:

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x + a_0$$

$n^2/2 + n/2$ multiplicações

n adições

$$P(x) = ((\dots(a_n x + a_{n-1})x + a_{n-2})x + \dots + a_1)x + a_0$$

n multiplicações

n adições

Algoritmo de Briot-Ruffini

$$P(x) = \underbrace{\underbrace{\underbrace{\underbrace{(((a_4}_{b_4}\cdot x+a_3)}_{b_3}\cdot x+a_2)}_{b_2}\cdot x+a_1)}_{b_1}\cdot x + a_0$$

$$b_0 = P(x)$$

De maneira geral:

$$\begin{cases} b_n = a_n \\ b_{n-k} = b_{n-k+1}x + a_{n-k} \end{cases}$$

Algoritmo de Briot-Ruffini (exemplo)

$$P(x) = 3x^4 + 2x^3 - x^2 + x + 5$$

$$P(x) = (((3 \cdot x+2) \cdot x-1) \cdot x+1) \cdot x + 5$$

Quanto vale P(3) ?

$b_4 = a_4 = 3$

$b_3 = b_4 \cdot x + a_3 = 3 \cdot 3 + 2 = 11$

$b_2 = b_3 \cdot x + a_2 = 11 \cdot 3 - 1 = 32$

$b_1 = b_2 \cdot x + a_1 = 32 \cdot 3 + 1 = 97$

$b_0 = b_1 \cdot x + a_0 = 97 \cdot 3 + 5 = 296$

$= P(3)$

Esquema prático

Calcular $P(x) = a_4x^4 + a_3x^3 + a_2 x^2 + a_1 x + a_0$

em $x = z$:

$b_4 = a_4$

$b_3 = b_4z + a_3$

$b_2 = b_3z + a_2$

$b_1 = b_2z + a_1$

$b_0 = b_1z + a_0 \longrightarrow P(z)$

	a_4	a_3	a_2	a_1	a_0	
z		$+$ zb_4	$+$ zb_3	$+$ zb_2	$+$ zb_1	
b_i	b_4	b_3	b_2	b_1	b_0	$P(z)$

Esquema prático aplicado ao exemplo

Calcular $P(x) = 3x^4 + 2x^3 \;\; -1\,x^2 + x + 5$
em $x = 3$:

	3	2	-1	1	5	
3		+ 9	+ 33	+ 96	+ 291	
b_i	3	11	32	97	296	P(z)

No caso geral

		a_n	a_{n-1}	a_{n-2}	...	a_2	a_1	a_0	
z			$+$ zb_n	$+$ zb_{n-1}	...	$+$ zb_3	$+$ zb_2	$+$ zb_1	
	b_i	b_n	b_{n-1}	b_{n-2}	...	b_2	b_1	b_0	$P(z)$

ponto onde queremos calcular $P(x)$ e $P'(x)$

Calcule:

- $P(x) = 2x^5 + 3x^3 - 2x^2 + x - 10$ em $x=2$

  $P(2) = 72$

- $P(x) = x^6 - 3x^5 - 2x^2 - x + 50$ em $x=3$

  $P(3) = 29$

Obtendo a derivada

- Uma vez que sabemos calcular a derivada de um polinômio de maneira bem automatizada, podemos tentar analisar alguma forma de obtê-la também com alguma economia computacional. De fato:

$$P(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

$$P'(x) = 4a_4x^3 + 3a_3x^2 + 2a_2x + a_1$$

Lembremos que para um dado $x = c$:

$$\begin{aligned} b_4 &= a_4 \\ b_3 &= b_4c + a_3 \\ b_2 &= b_3c + a_2 \\ b_1 &= b_2c + a_1 \\ b_0 &= b_1c + a_0 \end{aligned} \quad \longrightarrow \quad \begin{aligned} a_4 &= b_4 \\ a_3 &= b_3-b_4c \\ a_2 &= b_2-b_3c \\ a_1 &= b_1-b_2c \\ a_0 &= b_0-b_1c \end{aligned}$$

IDÉIA (para eficiência computacional)

- No método de Newton, a cada iteração, precisamos calcular $P(x)$ e $P'(x)$ em um dado ponto x_k (vamos chamá-lo de *c*).
- Suponha que já calculamos $P(c)$
- Para calcularmos $P'(c)$ precisamos:

$$P'(c) = 4a_4c^3 + 3a_3c^2 + 2a_2c + a_1$$

Substituindo

$a_4 = b_4$; $a_3 = b_3 - b_4c$; $a_2 = b_2 - b_3c$; $a_1 = b_1 - b_2c$;

$$P'(c) = 4b_4c^3 + 3(b_3 - b_4c)c^2 + 2(b_2 - b_3c)\,c + (b_1 - b_2c)$$

$$\boxed{\mathbf{P'(c) = b_4c^3 + b_3c^2 + b_2c + b_1}}$$

Como calcular $P'(c)$ da maneira mais eficiente que conhecemos ?

IDÉIA (para eficiência computacional)

$$\boxed{P'(c) = b_4c^3 + b_3c^2 + b_2c + b_1}$$

Como calcular P'(c) da maneira mais eficiente que conhecemos ?

- Usamos a mesma estratégia:

$c_4 = b_4$

$c_3 = b_4c + b_3$

$c_2 = b_3c + b_2$

$\boxed{c_1} = b_2c + b_1$

P'(c)

Caso geral

De maneira geral:

$$\begin{cases} c_n = b_n \\ c_{n-k} = c_{n-k+1}x + b_{n-k} \end{cases}$$

$$k=1\ldots\, n-1$$

$$\boxed{c_1 = P'(c)}$$

Esquema prático

	a_n	a_{n-1}	a_{n-2}	...	a_2	a_1	a_0	
z		$+$ zb_n	$+$ zb_{n-1}	...	$+$ zb_3	$+$ zb_2	$+$ zb_1	
b_i	b_n	b_{n-1}	b_{n-2}	...	b_2	b_1	b_0	$P(z)$
		$+$ zc_n	$+$ zc_{n-1}	...	$+$ zc_3	$+$ zc_2		
c_i	c_n	c_{n-1}	c_{n-2}	...	c_2	c_1		$P'(z)$

ponto onde queremos calcular $P(x)$ e $P'(x)$

Algoritmo de Newton

- $\delta x = x_0$
- Para k=1 ... it_{max}
  - $b = a_n$
  - $c = b$
  - Para i=(n-1)...1
    - $b = a_i + bx$
    - $c = b + cx$
  - se $|b| < \varepsilon_1$ FIM
  - $\delta x = b/c$
  - $x = x - \delta x$
  - Se $|\delta x| < \varepsilon_2$ FIM
- Não houve convergência no n. de iterações fixado (FIM)

Quais são os critérios de parada nesse caso ?

- a função está suficientemente próxima de zero

ou

- o passo é suficientemente pe-queno.

Exemplo (1/4)

- Calcular a raiz de $P(x) = x^3 + 2x^2 - 0.85x - 1.7$ na proximidade de $x=0.9$, com erro relativo menor que 10^{-2}.

	1	2	-0.85	-1.7		
		+ 0.9	+ 2.61	+ 1.584		
b_i	1	2.9	1.76	-0.116	$\longrightarrow$	P(0.9)
		+ 0.9	+ 3.42			
c_i	1	3.8	5.18		$\longrightarrow$	P'(0.9)

podemos calcular x_1

Exemplo (2/4)

$x_1 = x_0 - P(x_0)/P'(x_0)$

$= 0.9 - (-0.116)/1.584 = 0.9224$

Erro relativo:

$|x_1 - x_0|/|x_1| \approx 0.02 > 10^{-2}$

Continuamos!

Exemplo (3/4)

- $x_1 = 0.9224$

	1	2	-0.85	-1.7	
		+ 0.9224	+ 2.6956	+ 1.7024	
b_i	1	2.9224	1.8456	0.0024	→ P(0.9224)
		+ 0.9224	+ 3.5464		
c_i	1	3.8448	5.392		→ P'(0.9224)

podemos calcular x_2

Exemplo (4/4)

$x_2 = x_1 - P(x_1)/P'(x_1)$

$= 0.9224 - (0.0024)/5.392 = 0.9220$

Erro relativo:

$|x_2 - x_1|/|x_2| \approx 0.0004 < 10^{-2}$

FIM!

Obtendo outras raízes

- Note que:

$$Q(x) = b_n x^{n-1} + b_{n-1} x^{n-2} + \dots + b_2 x + b_1 = P(x)/(x-z)$$

De fato:

$$(b_n x^{n-1} + b_{n-1} x^{n-2} + \dots + b_2 x + b_1)\,(x-z)$$
$$= b_n x^n + (b_{n-1} - z b_n) x^{n-1} + \dots + (b_1 - z b_2) x + (b_0 - z b_1)$$
$$= a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

<u>O que isso significa ?</u>

- Podemos pegar a linha dos b_n's na tabela e recomeçar o procedimento para conseguir mais uma raiz.

Voltando ao exemplo

Tabela para a raiz: 0.9220

	1	2	-0.85	-1.7
		+ 0.9220	+ 2.6941	
	0.9220			
b_i	1	2.9220	1.844	
	b_3	b_2	b_1	

$$Q(x) = x^2 + 2.9220x + 1.844$$

Novo polinômio.

As raizes de Q(x) são as raízes ainda não calculadas de P(x)