ECONOMIA COMPUTACIONAL COM O USO DO *MATHEMATICA* ™

POLINÔMIOS E EQUAÇÕES

Prof. Marcelo de Oliveira Passos
Departamento de Economia – Universidade Federal de Pelotas
Mestrado em Organizações e Mercados – Economia Aplicada

Importante

- Aconselha-se utilizar a versão paga do programa Mathematica™ ou a versão de teste por 15 dias.
- Evite a pirataria.
- A versão de teste está disponível em https://www.wolfram.com/mathematica/trial/
- Os preços das versões pagas estão disponíveis em https://www.wolfram.com/mathematica/pricing/

Matemática simbólica (álgebra literal)

- As expressões algébricas manuscritas são parecidas com as que são escritas no Mathematica.
- As diferenças estão no uso de alguns símbolos do teclado (como ^, para as potências, * para as multiplicações, / para os quocientes e assim por diante).
- Após digitar a expressão, pressione as teclas [SHIFT]+[ENTER] no teclado normal (mais usado em laptops, notebooks e netbooks) ou pressione [ENTER] no teclado numérico (esta forma é mais usada em desktops com teclados maiores do que os dos notebooks).

Matemática simbólica (álgebra literal)

- Se a expressão não for compreendida, isto é, no caso de o programa não puder calcular ou simplificar, a mesma expressão digitada **In[n]** será retornada como resposta **Out[n]**.
- É importante notar que nas potências, x^2y significa x^2y. Mas x^(2y) equivale a x^{2y}

- **>x^3+4x^2-6x+16**
- $16 - 6x + 4x^2 + x^3$

Matemática simbólica (álgebra literal)

- O *Mathematica* calcula expressões simbólicas tão bem quanto executa cálculos numéricos.
- As operações aritméticas devem ser agrupadas dentro de parênteses.
- É importante ressaltar que o produto de a e b é escrito como a b.
- Sem um espaço definido ab é um produto univariado.

Comandos para operações algébricas: visão geral

- **Expand[expr]:** desenvolve os fatores da expressão algébrica.
- **Factor[expr]:** fatora a expressão onde for possível.
- **Simplify[expr]:** retorna a forma mais simples da expressão.
- **ExpandAll[expr]:** desenvolve produtos e potências da expressão.

Comandos para operações algébricas: visão geral

- **Together[expr]:** põe toda a expressão sobre um denominador (transforma a expressão no numerador de uma fração algébrica).
- **Cancel[expr]:** permite cancelar todos os fatores comuns da expressão.
- %: utiliza novamente a expressão imediatamente anterior.
- %%: Utiliza novamente a expressão imediatamente anterior à anterior

Comandos para equações algébricas

- **Solve[equações, variáveis]:** tenta resolver as equações *para as variáveis*.
- As raízes determinadas por **Solve** são expressas como uma lista da seguinte forma:
- $\{\{x \rightarrow x_1\},\{x \rightarrow x_2\},\ldots\}$
- A notação $x \rightarrow x_1$ indica que a solução x é x_1, mas x não é substituído por este valor.
- Se a equação possui raízes de múltiplos m>1, cada uma delas é repetida m vezes.

Comandos para equações algébricas

- Se somente uma delas estiver presente, as variáveis podem ser omitidas.
- Ex:
- > **Solve[10x+5== 5x+2.5]**
- {{x → -0.5}}
- Se resolvermos a equação ax = b para x, **Solve** nos dirá que x = b/a.
- Mas se a = b = 0, então qualquer número x é uma solução.

Comandos para equações algébricas

- Neste caso, o comando **Reduce** pode ser usado para descrever todas as soluções possíveis.
- **Reduce[equações, variáveis]**: simplifica equações tentando resolvê-las para variáveis.
- Se o termo "equações" for uma identidade, **Reduce** retorna o valor `True`. Se ele for uma contradição, o valor `False` é retornado.
- **Reduce** usa os símbolos && (o "e" lógico ou "$\wedge$") e o || ("ou" lógico ou "$\vee$"). O símbolo && tem precedência sobre o ||.

Comandos para equações algébricas

- Exs:
- **Solve[x^2-12==-3,x]**
- {{x → -3}, {x → 3}}
- **Reduce[x^2-9==(x-3)*(x+3),x]**
- True
- **Reduce[x^2-12==(x-3)*(x+3),x]**
- False

Comandos para equações algébricas

- **Solve[x^2+y^2==12&&x+y==4]**
- $\{\{x \to 2 - \sqrt{2}, y \to 2 + \sqrt{2}\}, \{x \to 2 + \sqrt{2}, y \to 2 - \sqrt{2}\}\}$

- O *Mathematica* retorna as soluções das equações como uma lista agrupada ordenadamente.
- Ela não pode ser usada diretamente como input para outra estrutura matemática.
- Mas é possível acessar estes valores sem desnecessariamente digitar ou colar usando /.

Comandos para equações algébricas

- Se nós quisermos calcular o valor de uma expressão usando soluções obtidas a partir de Solve, nós podemos usar o /. (operador de substitução) e o *Mathematica* substituirá os valores apropriados.
- Ex: Resolver o sistema com as equações x+y=10 e x+y^2=52 e calcular a expressão 2x+2y.
- **>solucoes=Solve[{x+y==10,x+y^2==52},{x,y}]**
- {{x → 16, y → -6}, {x → 3, y → 7}}
- **>2x+2y/.solucoes**
- {20, 20}

Comandos para equações algébricas

- O comando **Solve** foi projetado para resolver equações algébricas.
- Mas ele pode às vezes ser usado para encontrar soluções limitadas de equações transcendentais.
- Uma mensagem de alerta aparece quando nem todas as soluções puderem ser encontradas.
- Ex: >**Solve[Cos[x]==1/2,x]**
- $\{\{x \to \text{ConditionalExpression}[-\frac{\pi}{3} + 2\pi\, C[1],\ C[1] \in \text{Integers}]\},$
  $\{x \to \text{ConditionalExpression}[\frac{\pi}{3} + 2\pi\, C[1],\ C[1] \in \text{Integers}]\}\}$

Comandos para equações algébricas

- Para obter uma solução mais geral para essa equação, use **Reduce.**
- **>Reduce[Cos[x]==1/2,x]**
- $C[1] \in \text{Integers} \;\&\&\; \left(x == -\frac{\pi}{3} + 2\,\pi\, C[1] \;||\; x == \frac{\pi}{3} + 2\,\pi\, C[1] \right)$
- Se as equações a serem resolvidas forem inconsistentes o *Mathematica* retorna uma lista vazia.
- O comando **Nsolve** obtém uma aproximação numérica para a expressão.

Comandos para equações algébricas

- Ex: **> equacao=x^4-16x^3+61x^2-22x-12==0**
- $-12-22x+61x^2-16x^3+x^4==0$
- **>Solve[equacao]**
- $\{\{x\to 3-\sqrt{5}\},\{x\to 3+\sqrt{5}\},\{x\to 5-2\sqrt{7}\},\{x\to 5+2\sqrt{7}\}\}$
- **> NSolve[equacao]**
- $\{\{x\to -0.291503\},\{x\to 0.763932\},\{x\to 5.23607\},\{x\to 10.2915\}\}$
- Uma solução "estranha" é aquela que tecnicamente não é uma solução da equação, mas que emerge do processo de solução. Soluções de cálculos com radiciações podem gerar tais soluções.

Comandos para equações algébricas

- O comando **VerifySolutions** determina se o *Mathematica* deve verificar se as soluções obtidas são estranhas.
- O default do comando é **VerifySolutions -> True**, que elimina as soluções estranhas da lista de soluções.
- Mas se as soluções estranhas forem desejadas, o comando **VerifySolutions->False** deve ser usado.
- Ex: **>Solve[x+x^(1/2)==5]**
- $\left\{\left\{x \to \frac{1}{2}\left(11 - \sqrt{21}\right)\right\}\right\}$

Comandos para equações algébricas

- > **Solve[x+x^(1/2)==5,VerifySolutions→False]**
- $\left\{\left\{x \to \frac{1}{2}\left(11 - \sqrt{21}\right)\right\}, \left\{x \to \frac{1}{2}\left(11 + \sqrt{21}\right)\right\}\right\}$
- Verifica-se que o segundo termo da solução acima é uma solução estranha.

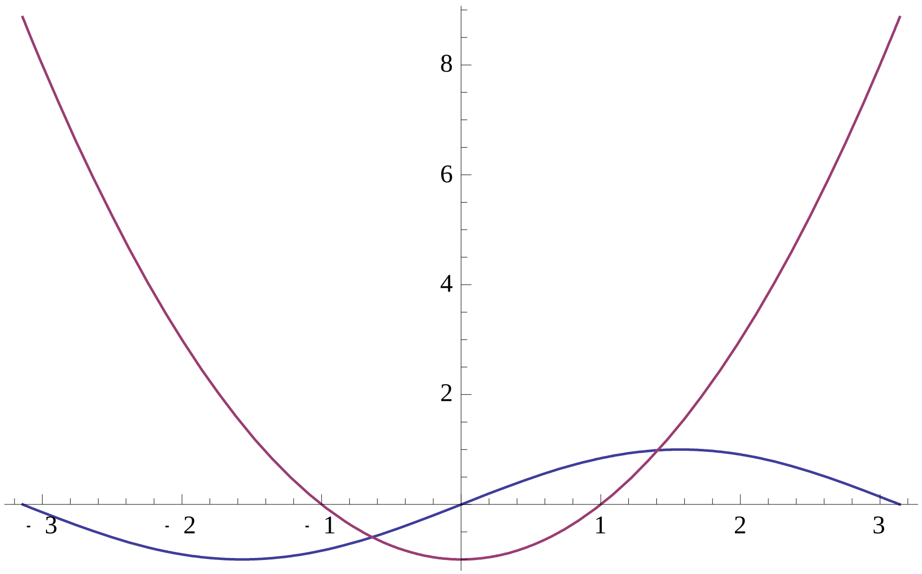
Comandos para equações transcendentais

- As equações transcendentais são aquelas que não são equações algébricas.
- O *Mathematica* tem o comando **FindRoot** para lidar com estas equações.
- O **FindRoot** usa métodos iterativos para encontrar soluções.
- Deve-se especificar um valor inicial, também chamado de estimativa inicial (*initial guess*).
- Para melhores resultados, a estimativa inicial deve ser tão próxima da raiz desejada quanto for possível.

Comandos para equações transcendentais

- **FindRoot[lde==lee, {x,x_0}]** : resolve a equação lde (lado direito da equação) == lee (lado esquerdo da equação) usando o método de Newton com valor inicial igual a x_0.
- **FindRoot[lde==lee, {x,{x_0,x_1}}]** : resolve a equação lde == lee usando uma variante do método da secante com valores iniciais iguais a x_0 e a x_1.
- **FindRoot[lde==lee, {x,x_0,xmin, xmax}}]** : tenta resolver a equação lde == lee, mas para se a iteração for para fora do intervalo {xmin, xmax}.

Comandos para equações transcendentais

- Se uma função for especificada no lugar da equação lde==lee, o comando **FindRoot** irá calcular o zero da função (o número x tal que f(x)=0)
- **>Plot[{Sin[x], x^2-1},{x,-π,π}]**
- 

Comandos para equações transcendentais

- O gráfico das duas funções mostra que a interseção delas é próxima de x= -1 e x=1.
- **>FindRoot[Sin[x]==x^2-1,{x,-1}]**
- {x → -0.636733}
- **>FindRoot[Sin[x]==x^2-1,{x,1}]**
- {x → 1.40962}

Outros exemplos

- > **Expand[(a b-c d)^2+(a d+b c)^2]**
- $a^2 b^2 + b^2 c^2 + a^2 d^2 + c^2 d^2$

- >**Factor[%]**
- $(a^2 + c^2) (b^2 + d^2)$

- A função **Simplify** utiliza vários métodos para reduzir expressões às suas formas mais simples.

Exemplos

- > **Simplify[Cos[x]^4-Sin[x]^4]**
- Cos[2 x]
- **FullSimplify** é uma versão mais poderosa de **Simplify**, mas ela pode demorar mais para dar uma resposta.
- > **FullSimplify[Gamma[z] Gamma[1-z]]**
- π Csc [π z]

Atribuindo valores à variáveis

- Para atribuir um valor numérico ou literal (letra) a uma variável, escreva o valor da variável e pressione [ENTER]
- Escrever em seguida a expressão e pressione [SHIFT]+[ENTER] no teclado comum ou [ENTER] no teclado numérico.
- Use o formato "nome_da_variável = expressão”.

Atribuindo valores às variáveis

- O valor atribuído à variável permanecerá em todas as expressões inseridas depois dela
- **>y=-5**
- -5
- **>y^3+6y^2-3y**
- 40

Atribuindo valores à variáveis

- Outro exemplo: atribuir um valor literal à variável y. Como a variável y já foi utilizada vamos limpá-la antes da nova atribuição. O programa dará a resposta na forma que considera mais simples.
- **>y=.**
- $1 + x^2$
- **>y=x^2+1**
- $-3\,y + 3\,y^2 + y^3$

Atribuindo valores à variáveis

- É sempre recomendável limpar os valores atribuídos a uma variável antes de continuar com os cálculos na mesma sessão. Assim, o próximo passo é limpar a variável y.
- **>Clear [y]**
- O símbolo de igualdade = é utilizado para atribuir valores às variáveis.

Atribuindo valores à variáveis

- É possível colocar vários comandos sobre uma linha separando-as com sinais de ponto-e-vírgula.
- Os comandos são executados em seqüência, com o output sendo o resultado do último comando.

- **> x=3;y=x+1; Sqrt[x^2+y^2]**
- 5

Atribuindo valores à variáveis

- Os valores atribuídos às variáveis são mantidos até que eles sejam explicitados claramente – ou até que o usuário conclua a sessão do *Mathematica*.
- Abaixo, espera-se uma expressão algébrica, mas em vez de obtê-la, obtém-se como resultado um número, uma vez que x e y ainda possuem valores numéricos que foram atribuídos acima.

- **> Expand[(x+y)^5]**
- 16807

Atribuindo valores à variáveis

- Para evitar este comportamento inesperado, convém apagar os valores das variáveis tão logo se finalize o uso delas.
- **> Clear[x,y];Expand[(x+y)^5]**
- $x^5 + 5x^4 y + 10x^3 y^2 + 10x^2 y^3 + 5x y^4 + y^5$

Atribuindo valores à variáveis

- No *Mathematica*, você pode evitar todas estas questões pelo uso de regras de transformação. A regra $x \longrightarrow 3$ representa "x é redefinido como 3".
- Abaixo o /. aplica esta regra à expressão anterior, mas não atribui um valor a x neste processo.

- **>1+x^2+x^4+x^5/.x->3**
- 334

Limpando os valores das variáveis

- Para cancelar o valor da variável, isto é, para limpar a letra ou valor numérico associado a ela, você pode utilizar os comandos: nome_da_variável =. ou simplesmente **Clear [nome_da_variável]**
- Para limpar mais de uma variável de uma só vez, utilize o comando **Clear [var1, var2, ..., vark].**

Limpando valores das variáveis

- O valor atribuído à variável permanecerá em todas as expressões inseridas depois dela
- Para cancelar o valor da variável, isto é, para limpar a letra ou valor numérico associado a ela, você pode utilizar os comandos: nome_da_variável =.   ou simplesmente **Clear [nome_da_variável]**

Exemplos

- **>(a+b)*(a-b)/((2-x)*(x^3+y^2))**
- $$\frac{(a-b)\ (a+b)}{(2-x)\ (x^3+y^2)}$$
- Desenvolvendo os fatores da expressão acima com o comando Expand [%]:
- **>Expand[%]**
- $$\frac{a^2}{(2-x)\ (x^3+y^2)} - \frac{b^2}{(2-x)\ (x^3+y^2)}$$

Exemplos

- Para desenvolver os produtos e as potências da expressão, utiliza-se o comando **ExpandAll [%%]:**
- **>ExpandAll [%%]**
- $$\frac{a^2}{2\,x^3 - x^4 + 2\,y^2 - x\,y^2} - \frac{b^2}{2\,x^3 - x^4 + 2\,y^2 - x\,y^2}$$
- Fatorando ao máximo a expressão:
- **>Factor[%]**
- $$-\frac{(a-b)\,(a+b)}{(-2+x)\,(x^3+y^2)}$$

Exemplo

- Simplificando ao máximo a expressão:
- **>Simplify[%]**
- $$\frac{-a^2 + b^2}{(-2 + x)\ (x^3 + y^2)}$$

Exemplo

- Outro exemplo: atribuir um valor literal à variável y.
- Como a variável y já foi utilizada vamos limpá-la antes da nova atribuição.
- O programa dará a resposta na forma que considera mais simples.

- **>y=.**
- **>y=x^2+1**
- $1 + x^2$

Exemplo

- **>y^3+3y^2-3y**
- $-3\,y + 3\,y^{2} + y^{3}$
- É sempre recomendável limpar os valores atribuídos a uma variável antes de continuar com os cálculos na mesma sessão.
- Assim, o próximo passo é limpar a variável y.
- **>Clear [y]**

Exemplo

- O símbolo de igualdade = é utilizado para atribuir valores às variáveis.
- É possível colocar vários comandos sobre uma linha separando-as com sinais de ponto-e-vírgula.
- Os comandos são executados em seqüência, com o output sendo o resultado do último comando.

- **> x=3;y=x+1; Sqrt[x^2+y^2]**
- 5

Exemplo

- No *Mathematica*, você pode evitar todas estas questões pelo uso de regras de transformação.
- A regra $x \longrightarrow 3$ representa "x é redefinido como 3".
- Abaixo o /. aplica esta regra à expressão anterior, mas não atribui um valor a x neste processo.

- **>1+x^2+x^4+x^5/.x->3**
- 334

Exemplos

- **>(a+b)*(a-b)/((2-x)*(x^3+y^2))**
- $$\frac{(a-b)\ (a+b)}{(2-x)\ (x^3+y^2)}$$
- Desenvolvendo os fatores da expressão acima com o comando Expand [%]:
- **>Expand[%]**
- $$\frac{a^2}{(2-x)\ (x^3+y^2)} - \frac{b^2}{(2-x)\ (x^3+y^2)}$$

Exemplos

- Para desenvolver os produtos e as potências da expressão, utiliza-se o comando **ExpandAll [%%]**
- **>ExpandAll [%%]**
- $$\frac{a^2}{2x^3 - x^4 + 2y^2 - xy^2} - \frac{b^2}{2x^3 - x^4 + 2y^2 - xy^2}$$
- Fatorando ao máximo a expressão:
- **>Factor[%]**
- $$-\frac{(a-b)(a+b)}{(-2+x)(x^3+y^2)}$$

Exemplos

- Simplificando ao máximo a expressão:
- **>Simplify[%]**
- $$\frac{-a^2+b^2}{(-2+x)\ (x^3+y^2)}$$

Polinômios

- A seguir está uma lista com os principais comandos do *Mathematica* para trabalhar com polinômios.
- Designaremos “pol” como abreviatura de polinômio.
- Inicialmente digita-se o polinômio dando-lhe um nome na forma de "poln = expr".
- Após cada digitação, pressionar [ENTER].

Polinômios: comandos

- **Exponent[pol, x]:** mostra o grau do polinômio.
- **PolynomialGCD[pol1, pol2]**: máximo divisor comum entre pol1 e pol2.
- **PolynomialLCM[pol1, pol2]**: mínimo múltiplo comum entre pol1 e pol2.
- **PolynomialQuotient[pol1, pol2, var]**: retorna o quociente da divisão de pol1 por pol2.
- **PolynomialRemainder[pol1, pol2, var]**: fornece o resto da divisão de pol1 por pol2.

Polinômios: exemplos

- A abreviatura var é utilizada para significar "variável do polinômio".
- Para expandir uma expressão polinomial, utilize o comando abaixo:
- **>pol1=Expand[(x+3)^2+(x-1)^3]**
- $8 + 9x - 2x^2 + x^3$
- Inserindo um novo polinômio para operarmos com dois polinômios

Polinômios: exemplos

- **>pol2=2x^3+4x-2**
- $-2 + 4\,x + 2\,x^3$
- Para achar o quociente entre pol1 e pol2:
- **>PolynomialQuotient[pol1,pol2,x]**
- 1/2

Polinômios: exemplos

- Calculando o resto da divisão de pol1 por pol2:
- **>PolynomialRemainder[pol1,pol2,x]**
- $9 + 7x - 2x^2$

Equações na forma simbólica

- As equações são escritas com a notação ==.
- O segundo argumento de **Solve** é a variável incógnita.
- As soluções são fornecidas pelo *Mathematica* como regras da forma $x \longrightarrow$ *value*.

- **>Solve[x^2+x==a,x]**
- $$\left\{\left\{x \to \frac{1}{2}\left(-1-\sqrt{1+4a}\right)\right\}, \left\{x \to \frac{1}{2}\left(-1+\sqrt{1+4a}\right)\right\}\right\}$$

Equações na forma simbólica

- O operador = é utilizado para atribuir valores às variáveis.
- Se o usuário utilizar = em vez de ==, a função Solve irá responder com um *beep* de alerta.
- Ocorre que, involuntariamente, o usuário ainda pode fixar o valor de uma variável.
- O usuário pode resolver equações envolvendo mais de uma única variável.

Equações na forma simbólica

- No output, cada sublista representa uma solução.
- >**Solve[{x^2-k^2==0,y^2==x^2},{x,y}**]

$$\{\{x \to -k, y \to -k\}, \{x \to -k, y \to k\}, \{x \to k, y \to -k\}, \{x \to k, y \to k\}\}$$

- A função Solve pode também manipular equações que envolvam funções mais complicadas.
- >**Solve[Log[x+Sqrt[a+x^2]]==b,x]**
- $\{\{x \to \frac{1}{2} e^{-b} (-a + e^{2b})\}\}$

Resolvendo equações numericamente

- Você pode usar NSolve para encontrar aproximações numéricas para as raízes de uma equação polinomial.

- **> solutions=NSolve[x^3+x+1==0,x]**

$$\{\{x \to -0.6823278038280193\},\{x \to 0.3411639019140964 - 1.161541399997252i\},\{x \to 0.3411639019140964 + 1.161541399997252i\}\}$$

Resolvendo equações numericamente

- A função NSolve fornece uma lista de regras.
- Para obter uma lista de números para um x qualquer, em vez de uma lista de regras, você deve apenas aplicar estas regras para x usando o operador /.

- **> x/.solutions**

$$\{-0.6823278038280193, 0.3411639019140096 - 1.161541399997252 i, 0.3411639019140096 + 1.161541399997252 i\}$$

Resolvendo equações numericamente

- A função Nsolve também encontra soluções numéricas para sistemas de equações simultâneas.

- **> NSolve[{x+y==2,x-3y+z==3,x-y+z==0},{x,y,z}]**

- $\{\{x \to 3.5, y \to -1.5, z \to -5.\}\}$

Resolvendo equações numericamente

- Para equações não polinomiais, você pode usar a função FindRoot para encontrar soluções.
- O argumento {x, 1} pede ao *Mathematica* para encontrar uma solução numérica iniciando em x = 1.
- **> FindRoot[3Cos[x]==Log[x],{x,1}]**
- $\{x \to 1.447258617277903\}$

Combinando padrões

- O *Mathematica* se baseia no princípio unificador de que tudo pode ser tratado como uma expressão simbólica.
- Todas as operações no *Mathematica* – sejam números, símbolos, expressões ou quaisquer outras – podem ser pensadas como transformações de expressões.

Combinando padrões

Objeto	Exemplo	Representação
Uma lista de elementos	{a,b,c}	List[a,b,c]
Uma expressão algébrica	x^2+5	Plus[5,Power[x,2]]
Uma equação	x==Sin[x]	Equal[x,Sin[x]]

Combinando padrões

- O símbolo _ (chamado de “blank” ou “espaço”) pode representar qualquer expressão. Por exemplo, o padrão _^_ representa qualquer potência.
- A função **Cases** fornece os elementos de uma lista que corresponde a um padrão específico.
- > **Cases[{1,x^2,E^3,2^t^3,4},_^_]**
- $\{x^2, e^3, 2^{t^3}\}$

Combinando padrões

- É possível nomear um pedaço de um padrão para retirá-lo de uma expressão.
- No padrão _^n_ o expoente é chamado n.
- > **Cases[{1,x^2,E^3,2^t^3,4},_^n_→n]**
- $\{2,3,t^3\}$

Combinando padrões

- Isto mostra que é possível usar padrões para transformar expressões. Aqui, o padrão x + y permanece para a soma de dois termos.

- **>{a+b,a+b+c}/.x+y→x^2+y^2**
- $\{a^2 + b^2, a^2 + (b+c)^2\}$

Referências

- BLACHMAN, Nancy. *Mathematica: a practical approach*. New Jersey: Prentice-Hall, 1992.
- DON, Eugene. *Mathematica - 750 exercises with answers .2nd ed. Schaum`s Outlines Series*. New York: McGraw-Hill, 2009.
- STINESPRING, John Robert. *Mathematica for Microeconomics*. New York:Harcourt/Academic Press, 2002.

Referências sobre Polinômios, Equações e aplicações à Economia

- BALDANI, J.; BRADFIELD, J. & TURNER, R.W. *Mathematical Economics*. 2nd ed. Chulo Alto, CA: Southwestern College Publications, 2004.
- CHIANG, A. & WAINWRIGHT, K. *Matemática para economistas*. 4ª ed. Rio de Janeiro: Campus, 2005.
- DOWLING, E. T. *Introduction to Mathematical Economics*. 3rd ed. New York: McGraw-Hill, 2001. Schaum`s Outline Series.

Referências sobre Polinômios, Equações e aplicações à Economia

- IEZZI, G. *Complexos, Polinômios e Equações: Fundamentos de Matemática Elementar - Vol. 6.* São Paulo: Atual, 1985.
- LIPSCHUTZ, S. *Matemática Finita (Coleção Schaum)*. São Paulo: McGraw-Hill, 1972 (tradução de Adalberto Panobianco Bergamasco).
- SIMON, C.P. & BLUME, L. *Mathematics for Economists*. New York: W.W. Norton & Company Inc., 1994