Determinação numérica da taxa interna de retorno: confronto entre os algoritmos de Boulding e de Wild *

Clovis de Faro **

Há quase 50 anos, Boulding apresentou como geral, e sem nenhuma justificativa formal das propriedades alegadas, um algoritmo para a determinação numérica da taxa interna de retorno associada a um dado projeto. Mais recentemente, em 1976, e sem fazer referência ao trabalho de Boulding, Wild apresentou como novo um outro procedimento numérico com a mesma finalidade. No presente artigo é efetuada uma investigação teórica e empírica dos dois algoritmos, sendo evidenciado que o procedimento de Wild é tão-somente um caso particular do formulado por Boulding.

1. Introdução

Considere-se um projeto caracterizado pela seqüência, com ao menos uma variação de sinal, de fluxos de caixa líquidos $\{a_0, a_1, \ldots, a_n\}$, onde, sem perda de generalidade, $a_0 < 0$ e $a_n \neq 0$. Sendo $i > -1$ uma taxa de juros cujo período coincida com o intervalo de tempo entre fluxos de caixa consecutivos, defina-se a função valor atual associada ao projeto:

$$V(i) = \sum_{j=0}^{n} a_j (1+i)^{-j} \qquad (1)$$

Para a avaliação econômica do projeto considerado, um dos critérios mais populares é o da chamada taxa interna de retorno. Por este critério, sendo i^* uma taxa de juros que anule a função atual, isto é $V(i^*) = 0$, taxa esta chamada de eficiência marginal do capital ou taxa interna de retorno, e sendo r uma certa taxa de juros tomada para comparação, o projeto será considerado como economicamente viável se $i^* > r$.

* Agradece-se ao patrocínio do PNPE.
** Da Escola de Pós-Graduação em Economia da Fundação Getulio Vargas.

A par de certos requisitos de existência e de unicidade de i^*, bem como de consistência com o chamado método do valor atual, como discutidos em de Faro & Mello e Souza (1975) e de Faro & Soares (1976), a implementação do critério da taxa interna de retorno resume-se, então, ao problema da determinação de seu valor numérico. Fazendo-se $x = (1 + i)^{-1}$, a determinação de i^*é equivalente à busca das raízes positivas do polinômio

$$F(x) = \sum_{j=0}^{n} a_j x^j \qquad (2)$$

um problema que, em geral, para $n > 4$, só pode ser resolvido por métodos iterativos. Dentre estes, além do confiável, mas lento, método da bisseção (Barros Santos, 1982, p. 46-8), destaca-se o chamado algoritmo de Newton-Raphson (cf. Barros Santos, 1982, p. 62-74, e Henrici, 1964, p. 77-86). Este último, embora eficiente, nem sempre é aplicável, pois sua convergência exige certas condições de monotonicidade e de ausência de mudança de concavidade de $F(x)$ no intervalo relevante.[1]

As deficiências dos métodos clássicos têm levado a que, para a finalidade precípua de determinação numérica de i^*, tenham sido propostos vários métodos específicos. Dentre estes, além de fórmulas aproximadas para certos casos particulares, como as discutidas em de Faro (1981; 1982) e Hawawini & Vora (1982), salientam-se os algoritmos respectivamente propostos por Boulding (1936) e por Wild (1976).

O propósito do presente trabalho é o de confrontar os dois algoritmos específicos citados. Assim, no item 2 é apresentado o algoritmo de Boulding. O algoritmo de Wild é descrito no item 3, que contém também uma comparação entre os dois procedimentos. O item 4 apresenta algumas considerações teóricas. Os resultados de uma investigação empírica constituem o item 5. Finalmente, no item 6 são resumidas as principais conclusões do estudo efetuado.

2. O algoritmo de Boulding

Há mais de 40 anos, Boulding (1936) desenvolveu o conceito de *centro temporal*, a partir do qual construiu um procedimento, dito ser geral, para a determinação numérica da taxa interna de retorno i^*associada a um dado projeto. Entretanto, apesar de interessantes características, tal procedimento parece ter sido quase que completamente negligenciado pelos estudiosos do assunto. Assim, a não ser na literatura alemã, no trabalho de Witten & Zimmermann (1977), não encontramos nenhuma referência moderna sobre o assunto que faça menção ao algoritmo de Boulding.

[1] No contexto de determinação de taxa interna de retorno, o uso do algoritmo de Newton-Raphson parece ter sido originalmente sugerido por Fisher (1966), sendo que Kaplan (1967) evidenciou a sua não aplicabilidade geral. Para certas classes de projetos com aplicabilidade garantida, ver os estudos de Faro (1976; 1978) e de Faro & Soares (1976; 1978).

2.1 *O conceito de centro temporal*

Dado o projeto $\{a_0, a_1, \ldots, a_n\}$, definam-se as seqüências de benefícios $\{b_0, b_1, \ldots, b_n\}$ e de custo $\{c_0, c_1, \ldots, c_n\}$, a ele associadas, de tal forma que:

$$b_j = \begin{cases} a_j, \text{ se } a_j \geq 0 \\ 0, \text{ se } a_j < 0 \end{cases} \quad j = 0, 1, \ldots, n \tag{3}$$

e

$$c_j = \begin{cases} -a_j, \text{ se } a_j < 0 \\ 0, \text{ se } a_j \geq 0 \end{cases} \quad j = 0, 1, \ldots, n \tag{4}$$

Então, observando-se que $a_j = b_j - c_j$, $j = 0, 1, \ldots, n$, segue-se que i^* será uma taxa interna de retorno associada ao projeto em causa se for verificada a seguinte igualdade:

$$V_b(i^*) = \sum_{j=0}^{n} b_j (1 + i^*)^{-j} = \sum_{j=0}^{n} c_j (1 + i^*)^{-j} = V_c(i^*) \tag{5}$$

Por outro lado, denotando-se por T_b e por T_c os respectivamente chamados centros temporais de benefícios e de custos, tem-se, por definição, que devem ser tais que:

$$(1 + i^*)^{T_b} \quad V_b(i^*) = V_b(0) \tag{6}$$

e

$$(1 + i^*)^{T_c} \quad V_c(i^*) = V_c(0) \tag{7}$$

Dividindo-se (6) por (7) e tendo em vista (5), decorre que:

$$(1 + i^*)^{T_b - T_c} = \frac{V_b(0)}{V_c(0)} \tag{8}$$

ou

$$i^* = (V_b(0)/V_c(0))^{1/(T_b - T_c)} - 1 \tag{8'}$$

Isto é, a expressão (8') indica que, se forem conhecidos os centros temporais de benefícios e de custos, podemos determinar a taxa interna i^*. Porém, em princípio, isso não parece ser de grande ajuda, pois T_b e T_c só podem ser determinados se conhecermos i^*.

2.2 *Justificativa heurística do algoritmo*

O princípio do procedimento consiste em determinar aproximações para T_b e T_c. Isto posto, o emprego de (8') fornece uma aproximação para i^*.

Tomando a taxa i como variável, defina a função

$$H_j(i) = b_j (1+i)^{T_b - j} \tag{9}$$

cuja derivada com respeito a i é:

$$H_j'(i) = (T_b - j) b_j (1+i)^{T_b - j - 1} \tag{10}$$

Então, dado que $H_j(0) = b_j$ e $H_j'(0) = (T_b - j)b_j$, segue-se que, desenvolvendo-se $H_j(i)$ em série de Maclaurin e desprezando-se os termos do segundo grau, em i, em diante, podemos escrever:

$$(1+i^*)^{T_b} V_b(i^*) = \sum_{j=0}^{n} H_j(i^*) \simeq \sum_{j=0}^{n} b_j [1 + (T_b - j) i^*] \tag{11}$$

Logo, tendo em vista a (6), segue-se que, em termos aproximados:

$$\sum_{j=0}^{n} b_j [1 + (T_b - j) i^*] = \sum_{j=0}^{n} b_j \tag{12}$$

Por conseguinte, desde que $i^* \neq 0$, podemos deduzir de (12) a seguinte aproximação inicial T_{b_1} para o centro temporal de benefícios:[2]

$$T_{b_1} = \sum_{j=0}^{n} j b_j / \sum_{j=0}^{n} b_j \tag{13}$$

Similarmente, temos também a aproximação:

$$T_{c_1} = \sum_{j=0}^{n} j c_j / \sum_{j=0}^{n} c_j \tag{14}$$

[2] O caso onde $i^* = 0$ é trivial, pois então o valor exato da taxa interna é sinalizado pelo fato de que $\sum_{j=0}^{n} a_j = 0$.

A partir de (13) e de (14), e fazendo uso de (8'), determina-se então uma primeira aproximação i_1 para i^*. Se $| V_b(i_1) - V_c(i_1) | < \delta$, onde $\delta > 0$, e tão pequeno quanto se queira, é a tolerância prefixada para o valor atual, i_1 será tomada como aproximação final. Caso contrário, Boulding (1936) sugere que, partindo de i_1, se obtenha uma nova aproximação i_2. Para tanto, fazendo-se uso respectivamente de (6) e de (7), defina as aproximações T_{b_2} e T_{c_2} de forma que:

$$(1+i_1)^{T_{b_2}} V_b(i_1) = V_b(0) \Rightarrow T_{b_2} = \frac{\log (V_b(0)/V_b(i_1))}{\log (1+i_1)} \tag{15}$$

e

$$(1+i_1)^{T_{c_2}} V_c(i_1) = V_c(0) \Rightarrow T_{c_2} = \frac{\log (V_c(0)/V_c(i_1))}{\log (1+i_1)} \tag{16}$$

A seguir, entrando com as novas aproximações T_{b_2} e T_{c_2} em (8'), determina-se a segunda aproximação i_2 para a taxa interna i^*. Se necessário, repita-se, partindo de i_2, o procedimento, e assim por diante.

2.3 *Procedimento iterativo*

Sumariando, uma vez conhecido i_1, e sendo i_k a aproximação obtida na k-ésima iteração, o algoritmo faz uso da seguinte recursão, para k = 1,2. . . :

$$i_{k+1} = (V_b(0)/V_c(0))^{1/(T_{b_{k+1}} - T_{c_{k+1}})} - 1 \tag{17}$$

onde

$$T_{b_{k+1}} - T_{c_{k+1}} = \frac{\log (V_b(0)/V_b(i_k)) - \log (V_c(0)/V_c(i_k))}{\log (1 + i_k)} \tag{18}$$

3. O algoritmo de Wild

Em um artigo publicado há não muito tempo, no qual não consta referência ao trabalho de Boulding, Wild (1976) apresentou como novo um procedimento para a determinação da taxa interna de retorno. Tal procedimento, que também faz uso do conceito de centro temporal, completa-se em até três passos.

De acordo com Wild (1976), o primeiro passo, que coincide com a iteração inicial do algoritmo de Boulding, fornece resultados no intervalo compreendido entre 69,3% e 99,95% do verdadeiro valor, sendo que em média chega-se a 91,4%.

É também alegado que o segundo passo apresenta uma acuidade média de 99,1% e que o terceiro e último passo provê uma acuidade média de 99,999%. Nada é dito quanto à existência de casos onde o procedimento não funcione.

Designando por $\hat{i}_k$ a aproximação associada ao k-ésimo passo, tem-se:

1º passo:

$$\hat{i}_1 = (V_b(0)/V_c(0))^{1/(T_{b_1} - T_{c_1})} - 1 \tag{19}$$

2º passo:

Sendo

$$\alpha_k = 1 - \frac{\log (V_b(\hat{i}_k)/V_c(\hat{i}_k))}{\log (V_b(0)/V_c(0))} \tag{20}$$

tome:

$$\hat{i}_2 = (1 + \hat{i}_1)^{1/\alpha_1} - 1 \tag{21}$$

3º passo:

Determine

$$\hat{i}_3 = (1 + \hat{i}_2)^{1/\alpha_2} - 1 \tag{22}$$

e tome como aproximação final

$$\hat{i}^* = \hat{i}_1 + \frac{(\hat{i}_2 - \hat{i}_1)^2}{2\hat{i}_2 - \hat{i}_1 - \hat{i}_3} \tag{23}$$

3.1 *Interpretação do algoritmo de Wild como caso particular do de Boulding*

Como iremos aqui evidenciar, a aproximação final sugerida por Wild nada mais é do que uma média, provavelmente sugerida por alguma forma não-linear de interpolação, dos resultados respectivamente obtidos nas três primeiras iterações do algoritmo de Boulding.

Como já apontado, temos que $\hat{i}_1 \equiv i_1$. Por outro lado, de (17) e (18) decorre que:

$$\log(1 + i_{k+1}) =$$

$$= \left\{ \frac{\log(1 + i_k)}{\log(V_b(0)/V_b(i_k)) - \log(V_c(0)/V_c(i_k))} \right\} \log(V_b(0)/V_c(0))$$

$$= \frac{\log(1 + i_k) \;\; \log(V_b(0)/V_c(0))}{\log(V_b(0)/V_c(0)) - \log(V_b(i_k)/V_c(i_k))}, k = 1,2,\ldots \tag{24}$$

E de (20) e (21), ou (22), tem-se:

$$\log(1 + \hat{i}_{k+1}) =$$

$$= \left\{ \frac{\log(V_b(0)/V_c(0))}{\log(V_b(0)/V_c(0)) - \log(V_b(\hat{i}_k)/V_c(\hat{i}_k))} \right\} \log(1 + \hat{i}_k), k = 1,2 \tag{25}$$

Logo, comparando-se (24) e (25), decorre imediatamente que $\hat{i}_k \equiv i_k$ também para $k = 2,3$. Por conseguinte, o algoritmo de Wild não é realmente novo. Não só as aproximações obtidas nos dois primeiros passos coincidem como as respectivamente obtidas nas duas primeiras iterações do procedimento de Boulding, como a aproximação final de Wild é média dos resultados das três primeiras aproximações de Boulding.

4. Desempenho dos algoritmos

A análise efetuada no item 3 permite-nos concluir que, do ponto de vista teórico, só cabe efetivamente um estudo do algoritmo de Boulding. Entretanto, quanto ao aspecto computacional, é interessante que se inclua na investigação a aproximação final de Wild, pois o uso desta pode, eventualmente, vir a acelerar o processo de convergência do procedimento devido a Boulding.

Dado que, regra geral, a determinação da taxa interna de retorno associa-se à implementação do critério de avaliação econômica de mesmo nome, a aplicação do algoritmo de Boulding deve ser analisada tendo em vista tal finalidade. Deste modo, segue-se que sua aplicação não deve ser feita de maneira indiscriminada, mas somente para projetos que atendam a certos requisitos relativos ao emprego do critério da taxa interna, tais como discutidos em de Faro & Soares (1976).

Assim, por exemplo, para o projeto $\{-10,15,-16\}$, a aplicação do algoritmo de Boulding produzirá a seguinte absurda seqüência de aproximações: $i_1 =$

$= -0{,}2275$, $i_2 = -0{,}4033$ e $i_3 > 2{,}7 \times 10^{10}$. Tal absurdo é devido ao fato de que, para o projeto considerado, como se verifica facilmente, as raízes referentes ao polinômio do segundo grau, definido por (2), são números complexos. Ou seja, no caso, inexiste a taxa interna de retorno.

Por outro lado, do ponto de vista prático, estamos usualmente interessados tão-somente no campo das taxas de juros positivas. Isto acarreta um possível problema, pois, por exemplo, se o projeto considerado for do que chamaremos do "tipo Jean", isto é com exatamente duas variações de sinal na seqüência de fluxos de caixa líquidos e tal que $\sum_{j=0}^{n} a_j > 0$, sabe-se que (cf. Jean, 1968) existirá exatamente uma solução $i^* > 0$ e outra em $(0,-1)$. Deste modo, é possível que o algoritmo de Boulding venha a convergir para a raiz negativa, que não é a que nos interessaria. Tal é o caso do projeto $-1, 5 -3$, cujas taxas internas negativa e positiva são, respectivamente, $i_1^* \cong -30{,}28\%$ e $i_2^* \cong 330{,}28\%$. Para este projeto, a primeira iteração do algoritmo de Boulding produz a aproximação $i_1 = -36{,}00$, indicando assim convergência para a raiz negativa.[3]

Para projetos tais como o do exemplo anterior, uma possível linha de ação é a de, fazendo uso do teorema de Vincent, como discutido em de Faro (1983), delimitar inicialmente, com alguma precisão, um intervalo que contenha a taxa positiva. A seguir, tomando-se como aproximação inicial o ponto médio do intervalo, faz-se uso do algoritmo de Boulding. No caso do exemplo considerado, dado que $i^* \in (3,5)$, temos que tomando-se $i_1 = 4$ o algoritmo de Boulding produzirá a aproximação $i_2 = 1{,}91$, indicando possível convergência para a raiz positiva desejada. Esta estratégia será comentada no item 5, que relata nossa investigação empírica.

4.1 *Análise teórica*

Do ponto de vista teórico, só cabe analisar formalmente a convergência do algoritmo de Boulding uma vez assegurada a existência e unicidade da taxa interna de retorno em todo o campo de significação econômica (definido por $i > -1$). Ora, *a priori*, sem que se aponham condições adicionais, somente a classe dos projetos ditos convencionais, que são aqueles que apresentam uma única variação de sinal na seqüência de fluxos de caixa líquidos, satisfaz os necessários requisitos. Concentrando atenção nesta classe, propriedades do algoritmo de Boulding serão apresentadas para dois casos particulares.

a) $a_j = 0$, $j = 1,2,\ldots,n-1$ e $a_n > 0$

[3] Na realidade, pode-se verificar que, já na terceira iteração, a aproximação será igual a $-30{,}43\%$.

Para este caso, é imediato que o valor exato da taxa interna de retorno é dado por:

$$i^* = (-a_n/a_0)^{1/n} - 1 \tag{26}$$

Por outro lado, observando-se que $T_{b_1} = n$ e $T_{c_1} = 0$, segue-se de (8') que a primeira iteração do procedimento de Boulding produzirá exatamente o valor procurado i^* dado por (26).

b) $a_j \geqslant 0, j = 1,2, \ldots, n$

Temos agora a classe dos projetos ditos de investimento simples. Para esta classe, é mostrado formalmente no anexo, de autoria de Paulo Klinger, que o algoritmo de Boulding produz uma seqüência de aproximações que converge para o verdadeiro valor i^*.

Para o caso geral de projetos convencionais, não dispomos de resultados formais. Porém, sendo corroborado por investigações empíricas efetuadas pelo autor, conjetura-se que também existe convergência.[4]

Entretanto, ao menos quando se requer unicidade da taxa interna de retorno somente no campo das taxas de juros não-negativas,[5] que costuma ser o caso de real interesse prático, nem sempre o algoritmo de Boulding será aplicável. Isto é, podem ocorrer situações onde as expressões matemáticas que são utilizadas não sejam definidas. De fato, tendo presente (13), (14) e (19), um caso de colapso formal do algoritmo de Boulding ocorre quando

$$\frac{\sum_{j=0}^{n} jb_j}{\sum_{j=0}^{n} b_j} = \frac{\sum_{j=0}^{n} jc_j}{\sum_{j=0}^{n} c_j} \tag{27}$$

Tal possibilidade é exemplificada pelo projeto cuja seqüência de fluxos de caixa líquidos é $\{-1, 5, 5, 1\}$. Fazendo uso de (3) e de (4), é fácil verificar que a relação (27) é satisfeita. Assim, embora se trate de um projeto do tipo Jean, com a única taxa interna positiva sendo aproximadamente igual a 482,84% por período (com a taxa negativa de cerca de – 82,84% por período), o algoritmo de Boulding não é aplicável, pois o valor numérico da primeira iteração fica indefinido.

[4] A propósito, vejam-se os casos dos projetos P23, P24, P25, P26, P58, P59, P60, P61, P62, P63, P64, P65, P66, P67, P68, P69 e P70, listados no quadro 1.

[5] Em nossas investigações, não conseguimos encontrar nenhum caso de colapso formal de algoritmo, para projetos com uma única taxa interna de retorno no campo geral das taxas com significação econômica, e tais que $a_n > 0$.

Embora menos provável de acontecer na prática, devido a erros de aproximação, um outro caso de colapso do algoritmo de Boulding ocorre quando, tendo em vista (20):

$$\frac{V_b(0)}{V_c(0)} = \frac{V_b(i_k)}{V_c(i_k)} \tag{28}$$

Conquanto seja um exemplo artificial, o caso do projeto $\{-10, 20, 40, -32\}$, também do tipo Jean (com $i^*_1 \simeq -34{,}44\%$ e $i^*_2 \simeq 198{,}16\%$), ilustra tal possibilidade. Basta tomar $i_k = 1$.

5. Investigação empírica

Tendo presente o apresentado em de Faro (1978), excluiremos de nossa análise empírica, para evitar duplicação, o caso dos projetos ditos de investimento simples. Para esta classe, embora considerando somente as três primeiras iterações do algoritmo de Boulding e a interpolação de Wild, constatou-se que, em média e com menos esforço computacional, são obtidas aproximações mais precisas do que via o emprego do mesmo número de iterações do procedimento de Newton-Raphson.

Para fins de avaliação geral do desempenho do algoritmo de Boulding, consideraram-se 86 diferentes projetos, que são listados no quadro 1. Neste quadro, para cada projeto, caracterizado por sua respectiva seqüência de fluxos de caixa líquidos, são apresentadas suas distintas taxas internas de retorno, bem como suas multiplicidades.[6]

[6] Devido ao fato mencionado de que, do ponto de vista prático, só interessam as taxas internas positivas, os projetos considerados, à exceção dos do tipo convencional, satisfazem sempre a condição $\sum_{j=0}^{n} a_j > 0$.

Quadro 1
Caracterização dos projetos

Projeto		Taxa(s) interna(s)[1]
Nº	Fluxo de caixa	(%)
P1	{ -2,2,2, -4, -4, 5,5, -6, -6,7,7	19,05
P2	{ -300,500,400,0, -300	-29,49 e 118,15
P3	{ -300,500,400, -300,100	110,95
P4	{ -300,0,400, -800,1000	19,68
P5	{ -300,900, -1200,0,3000	114,72
P6	{ -100,50, -50,100, -50,150	20,34
P7	{ -100, -150,200, -100,200	8,04
P8	{ -220,550, -320,10	-96,69; -15,19 e 61,88
P9	{ -1000,3010, -3030,10258	210,00
P10	{ -1000,1900, -2000,4400	100,00
P11	{ -10,60, -120,80	100,00 (3)
P12	{ -100,50,50,50, -10,60	32,00
P13	{ -10,30, -10,50	218,00
P14	{ -60,70, -20,240	100,00
P15	{ -80,40, -10,40,40, -10	-79,21 e 9,57
P16	{ -450, -200,700, -60,2000, -50	-97,50 e 58,43
P17	{ -100, -300,800, -300	-52,86 e 32,63
P18	{ -500,250,1020, -1000,3920	100,00
P19	{ -200,400, -100	-70,71 e 70,71
P20	{ -200,1200, -2400,1600	100,00 (3)
P21	{ -90,230, -100	-44,44 e 100,00
P22	{ -100,50,200, -200,800	100,00
P23	{ -10, -20, -30,0,50,100,20,0,0,0,100	38,34
P24	{ -100,0,0, -200,500,0,0,0,0,0,0,1,00, 0,0,0,100,100,100,100,100,100	29,23
P25	{ -1000, -2000, - 3000,0,0, - 10000, 500,500, 500,500,500	-35,57
P26	{ -100, -500,0,0,100,400	-4,49
P27	{ -2560,9600, -12640,7400, -3000,1250	25,00 (3)
P28	{ -640,1120,1160, -2350, -500,1250	25,00 (3)
P29	{ -100,400, -500,300, -700,500,2100	132,84
P30	{ -100,500, -700,300, -1000,600,2400	126,25
P31	{ -100,250, -200,300	120,94
P32	{ -300,350, -100,350, -100	-70,86 e 40,17
P33	{ -800,400, -100,400,400, -100	-79,21 e 9,57
P34	{ -110,40,40,40, -20,40, -10	-73,87 e 7,69
P35	{ -300,150,160, -20,100, -10	-89,97 e 13,02
P36	{ -100,50,50,50, -20,60	29,98
P37	{ -100,70,60, -10,60	34,08
P38	{ -30,10,10,10,10, -20,80	33,33
P39	{ -100,200,100, -300,1000	153,93
P40	{ -100,40,40,40, -10,50, -10 }	-79,98 e 17,98

continua

P41	{ −200,100,100, −10,100, −10	−90,01 e 17,32
P42	{ −200,110,80, −10,100, −10	−89,99 e 15,51
P43	{ −720,140,140,140,140,140,30, −20, 140, −20	−85,53 e 4,00
P44	{ −720,140,140,140,140.140,30, −20,140,140	7,70
P45	{ −500,1000, −400	−44,72 e 44,72
P46	{ −800,300,300,300, −200,300, −100	−64,37 e 5,82
P47	{ −1000,800,210. −20,830, −20	−97,59 e 31,06
P48	{ −200,800, −900,700	185,26
P49	{ −300,800, −1600,800,3200	100,00
P50	{ −300, −1800,2400,4000, −3200, −9600,38400	100,00
P51	{ −150,240, −50,80	60,00
P52	{ −600,1300, −600	−33,33 e 50,00
P53	{ −200,600, −500,200	100,00
P54	{ −100,600, −1300,1400,−1 200,800	100,00 (3)
P55	{ −800,600,2000, −800,1600	100,00
P56	{ −40,130, −100,100,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,1000	166,38
P57	{ −1,5, −3	−30,28 e 330,28
P58	{ −1, −1,0,0, −10.0,0,0,0,5,5	2,96
P59	{ −1, −1, −1, −1, 1,1,1,1,5	16,49
P60	{ −5,0,0,0, −5, −5, −5, −5,30	4,99
P61	{ −1,0,0,0,0,0,0,0,0, −1,10	24,23
P62	{ −5, −5,10,10,10.10	65,29
P63	{ −5, −5,10,10,10.10,10,10,10,	71,90
P64	{ −5, −5, −5,10,10,10,10,10,10,	39,80
P65	{ −5, −5,−5, −5, −5, 10,10,10,10,10,	23,02
P66	{ −5, −5, −5, −5, −5,10,10,10,10,	10,91
P67	{ −5, −5, −5, −5, −5, −5, 10,10,20	6,16
P68	{ −5, −5, −5, −5, −5, −5, −5,10,10	−12,32
P69	{ −5, −5, −5, −5, −5, −5, −5, −5,10	−32,34
P70	{ −1, −1,5	79,13
P71	{ −1,2, −4,8	100,00
P72	{ −1, −2,3,0, −8,56	100,00
P73	{ −4,6, −8,3	−50,00
P74	{ −1,3, −4,2, −16,40	100,00
P75	{ −8,16, −64,29	−50,00
P76	{ −20,10, −2, −2,20	9,22
P77	{ −2,5, −2	−50,00 e 100,00
P78	{ −4,12, −9,2	−50,00 (2) e 100,00
P79	{ −5,8.4,4, −8	−18,77 e 100,00
P80	{ −1,1, −1, −1,14	100,00
P81	{ −1,11,90, −100,1	−0,93 e 1518,05
P82	{ −1, −3,8, −3 }	−52,85 e 32,63

continua

continuação

P83	{ −2,18, −54,54	200,00 (3)
P84	{ −200,400,100, −800,696	59,95
P85	{ −16,72, −108,54	50,00 (3)
P86	{ −2,8, −22,40, −40,30, −30,20, −2 }	−88,09 e 40,95

[1] Se a taxa não for simples, indica-se entre parênteses sua multiplicidade.

No quadro 2, para cada um desses 86 projetos são fornecidos os valores numéricos obtidos nas três primeiras iterações do algoritmo de Boulding e na interpolação de Wild, bem como os respectivos erros percentuais com relação à única taxa interna positiva (ou com relação à taxa única, mesmo que negativa, no caso de projetos do tipo convencional.[7] Adicionalmente, além de uma coluna de observações, são também apresentados no quadro 2 os números de iterações necessárias para que, tanto fazendo uso da interpolação de Wild, como não (entre parênteses), fosse alcançada uma aproximação com precisão até a segunda casa decimal da taxa percentual.

Com relação ao caso geral dos projetos do tipo convencional, os resultados obtidos indicam um excelente desempenho para o algoritmo de Boulding. Assim, em média, não mais do que quatro iterações, fazendo uso da interpolação de Wild, foram necessárias para a obtenção de aproximação com erro não superior a 1%. Mesmo nos casos de convergência mais lenta, como em P24, os erros já eram bastante pequenos na quarta iteração (que, na realidade, corresponde à primeira iteração adicional a partir da interpolação de Wild).

Para as demais classes de projetos, os resultados nem sempre são satisfatórios. Em especial, como evidenciado pelos projetos P52 e P77, que foram aleatoriamente gerados, o algoritmo de Boulding entrou em colapso, pois foi satisfeita a igualdade dada por (27).

Tivemos também um fraco desempenho, no sentido de uma convergência muito lenta (mais de 203 iterações), para os projetos em que a respectiva taxa interna, embora única no campo geral, apresentava multiplicidade maior do que 1. Tal foi o caso dos projetos P11, P20, P27, P28, P54, P83 e P85.

[7] Para obtenção desses resultados fizemos uso de programa, escrito em Fortran, implementado em um computador Polymax da série 201 DP.

Quadro 2
Desempenho do algoritmo de Boulding

Nº do projeto	Aproximação (%)/erro (%)[1]				Nº da iteração final[2]	Observação
	1ª	2ª	3ª	Wild		
P1	16,66	18,73	19,01	19,05	3	
	(–12,55)	(–1,73)	(–0,21)	(–0–)	(4)	
P2	–51,80	–	–	–	–	Convergiu para a taxa
	–	–	–	–	–	negativa
P3	1 186,01	68,73	146,42	141,37	16	
	(968,96)	(–38,05)	(31,97)	(27,42)	(16)	
P4	21,34	19,55	19,69	19,68	3	
	(8,43)	(–0,66)	(0,05)	(–0–)	(4)	
P5	74,99	100,54	109,81	115,08	10	
	(–34,63)	(–12,36)	(–4,28)	(0,31)	(9)	
P6	20,58	20,34	–	–	–	
	(1,18)	(–0–)	–	–	(2)	
P7	8,10	8,04	–	–	–	
	(0,75)	(–0–)	–	–	(2)	
P8	–21,59	–	–	–	–	Convergiu para a taxa
	–	–	–	–		–15,19%
P9	213,60	210,03	210,00	–	–	
	(1,71)	(0,01)	(–0–)	–	(3)	
P10	100,90	100,00	–	–	–	
	(0,90)	(–0–)	–	–	(2)	
P11	28,37	41,47	49,41	61,58	>203	Não convergiu após 203
	(–71,63)	(–58,63)	(–50,59)	(–38,42)	(>203)	iterações
P12	29,61	31,81	31,98	32,00	3	
	(–7,47)	(–0,59)	(–0,06)	(–0–)	(4)	

continua

continuação

Nº do projeto	Aproximação (%)/erro (%)[1]				Nº da iteração final[2]	Observação
	1ª	2ª	3ª	Wild		
P13	203,14	216,79	217,90	218,00	3	
	(–6,82)	(–0,56)	(–0,05)	(–0–)	(4)	
P14	93,73	99,41	99,94	100,00	3	
	(–6,27)	(–0,59)	(–0,06)	(–0–)	(4)	
P15	9,71	9,57		–	–	
	(1,46)	(–0–)		–	(2)	
P16	61,27	58,45	58,43	–	–	
	(4,86)	(0,03)	(–0–)	–	–	
P17	59,58	25,50	35,74	33,37	8	
	(82,59)	(–21,85)	(9,53)	(2,27)	(10)	
P18	133,68	97,43	100,29	100,08	5	
	(33,68)	(–2,27)	(0,29)	(0,08)	(5)	
P19	137,04	57,99	75,26	72,17	8	
	(93,81)	(–17,99)	(6,43)	(2,06)	(9)	
P20	28,37	41,47	49,41	61,58	>203	Não convergiu após 203 iterações
	(–71,63)	(–58,53)	(–50,59)	(–38,42)	(>203)	
P21	–97,35	–	–	–	–	Convergiu para a taxa negativa
	–	–	–	–	–	
P22	133,65	97,48	100,28	100,08	5	
	(33,65)	(–2,52)	(0,28)	(0,08)	(5)	
P23	32,07	37,54	38,25	38,35	4	
	(–16,35)	(–2,09)	(–0,23)	(0,03)	(5)	
P24	16,01	23,32	26,74	29,74	9	
	(–45,23)	(–20,22)	(–8,52)	(1,74)	(12)	
P25	–34,58	–35,57		–	–	
	(–2,78)	(–0–)		–	(2)	
P26	–4,49			–	–	
	(–0–)			–	(1)	

continua

continuação

Nº do projeto	Aproximação (%)/erro (%)[1]				Nº da iteração final[2]	Observação
	1ª	2ª	3ª	Wild		
P27	7,79	11,22	13,24	16,13	>203	Não convergiu após 203
	(–68,84)	(–55,12)	(–47,04)	(–35,48)	(>203)	iterações
P28	7,79	11,21	13,22	16,09	>203	Não convergiu após 203
	(–68,84)	(–55,16)	(–47,12)	(–35,64)	(>203)	iterações
P29	57,64	82,04	97,70	125,73	22	
	(–56,61)	(–38,24)	(–26,45)	(–5,35)	(26)	
P30	46,77	65,20	76,69	95,72	69	
	(–62,95)	(–48,36)	(–39,26)	(–24,18)	(72)	
P31	122,58	120,83	120,94	–	–	
	(1,36)	(0,09)	(–0–)	–	(3)	
P32	52,29	38,82	40,35	40,16	5	
	(30,17)	(–3,36)	(0,45)	(–0,02)	(5)	
P33	9,71	9,57		–	–	
	(1,46)	(–0–)		–	(2)	
P34	7,92	7,68	7,69	–	–	
	(2,99)	(–0,13)	(–0–)	–	(3)	
P35	12,90	13,02		–	–	
	(–0,92)	(–0–)		–	(2)	
P36	29,11	29,94	29,98	–	–	
	(–2,90)	(–0,13)	(–0–)	–	(3)	
P37	31,60	33,91	34,07	34,08	3	
	(–7,28)	(–0,50)	(–0,03)	(–0–)	(4)	
P38	36,20	33,35	33,33	–	–	
	(8,61)	(0,06)	(–0–)	–	(3)	
P39	182,59	153,96	153,93	–	–	
	(18,62)	(0,02)	(–0–)	–	(3)	

continua

continuação

Nº do projeto	Aproximação (%)/erro (%)[1] 1ª	2ª	3ª	Wild	Nº da iteração final[2]	Observação
P40	18,53	17,98		–	–	
	(3,06)	(–0–)		–	(2)	
P41	17,05	17,31	17,32	–	–	
	(–1,56)	(–0,06)	(–0–)	–	(3)	
P42	15,25	15,51		–	–	
	(–1,68)	(–0–)		–	(2)	
P43	3,95	4,00		–	–	
	(–1,25)	(–0–)		–	(2)	
P44	7,28	7,67	7,70	–	–	
	(–5,45)	(–0,39)	(–0–)	–	(2)	
P45	158,12	23,47	64,03	54,62	16	
	(253,58)	(–47,52)	(43,18)	(22,18)	(18)	
P46	6,07	5,81	5,82	–	–	
	(4,30)	(–0,17)	(–0–)	–	(3)	
P47	27,97	30,75	31,02	31,06	3	
	(–9,95)	(–1,00)	(–0,13)	(–0–)	(6)	
P48	184,17	185,95	184,83	185,26	3	
	(–0,59)	(0,37)	(–0,23)	(–0–)	(11)	
P49	75,41	93,70	98,40	100,03	5	
	(–24,59)	(–6,30)	(–1,60)	(0,03)	(8)	
P50	130,75	96,46	100,54	100,11	5	
	(30,75)	(–3,54)	(0,54)	(0,11)	(6)	
P51	60,00			–	–	
	(–0–)			–	(1)	
P52	–					O algoritmo ficou
	–					indefinido

continua

continuação

Nº do projeto	Aproximação (%)/erro (%)[1]				Nº da iteração final[2]	Observação
	1ª	2ª	3ª	Wild		
P53	548,47	29,90	301,40	208,10	>203	Convergência lenta e
	(448,47)	(–70,10)	(201,40)	(108,10)	(>203)	oscilante
P54	28,37	41,47	49,41	–61,58	>203	Não convergiu após 203
	(–71,63)	(–58,53)	(–50,59)	(–38,42)	(>203)	iterações
P55	136,89	96,82	100,38	100,09	5	
	(36,89)	(–3,18)	(0,38)	(0,09)	(6)	
P56	103,05	115,42	126,53	–	–	Tomando 90% como apro
	(–38,06)	(–30,63)	(–23,95)	–	(28)	ximação inicial
P57	–36,00	–	–	–	–	Convergiu para a taxa
	–	–	–	–	–	negativa
P58	–2,95	–2,96		–	–	
	(–0,34)	(–0–)		–	(2)	
P59	16,24	16,48	16,49	–	–	
	(–1,52)	(–0,06)	(–0–)	–	(3)	
P60	5,19	4,98	4,99	–	–	
	(4,01)	(–0,20)	(–0–)	–	(3)	
P61	33,99	22,70	24,55	24,29	6	
	(40,28)	(–6,31)	(1,32)	(0,25)	(6)	
P62	58,74	64,75	65,25	65,29	3	
	(–10,03)	(–0,83)	(–0,06)	(–0–)	(4)	
P63	54,10	68,07	71,11	71,96	5	
	(–24,76)	(–5,33)	(–1,10)	(0,08)	(7)	
P64	36,08	39,49	39,77	39,80	3	
	(–9,35)	(–0,78)	(–0,08)	(–0–)	(5)	
P65	22,58	23,01	23,02	–	–	
	(–1,91)	(–0,04)	(–0–)	–	(3)	

continua

continuação

Nº do projeto	Aproximação (%)/erro (%)[1]				Nº da iteração final[2]	Observação
	1ª	2ª	3ª	Wild		
P66	11,01	10,91		–	–	
	(0,92)	(–0–)		–	(2)	
P67	6,24	6,15	6,16	–	–	
	(1,30)	(–0,16)	(–0–)	–	(3)	
P68	–11,69	–12,29	–12,32	–	–	
	(–5,11)	(–0,24)	(–0–)	–	(3)	
P69	–26,51	–31,05	–32,04	–32,33	4	
	(–18,03)	(–3,99)	(–0,93)	(–0,03)	(6)	
P70	84,20	78,91	79,14	79,13	3	
	(6,41)	(–0,28)	(0,01)	(–0–)	(4)	
P71	100,00			–	–	
	(–0–)			–	(1)	
P72	160,18	93,39	101,39	100,54	6	
	(60,18)	(–6,61)	(1,39)	(0,54)	(7)	
P73	–57,81	–47,17	–50,93	–49,95	6	
	(15,62)	(–5,66)	(1,86)	(–0,10)	(8)	
P74	87,17	96,98	99,27	99,97	5	
	(–12,83)	(–3,02)	(–0,73)	(–0,03)	(5)	
P75	–60,13	–47,93	–50,43	–50,00	3	
	(20,26)	(–4,14)	(0,86)	(–0–)	(5)	
P76	9,02	9,21	9,22	–	–	
	(–2,17)	(–0,11)	(–0–)	–	(3)	
P77	–					O algoritmo ficou indefinido
	–					
P78	–52,73					Convergiu, pobremente, para a taxa negativa
	–					

continua

continuação

P79	−25,31					Convergiu para a taxa negativa
	–					
P80	112,64	100,39	100,01	100,00	3	
	(12,64)	(0,39)	(0,01	(−0−)	(4)	
P81	−0,93					Convergiu para a taxa negativa
	–					
P82	59,58	25,50	35,74	33,37	8	
	(82,59)	(−21,85)	(9,53)	(2,27)	(10)	
P83	55,24	80,60	96,07	120,25	>203	Não convergiu após 203 iterações
	(−72,38)	(−59,70)	(−51,97)	(−39,88)	(>203)	
P84	51,71	60,66	59,88	59,94	4	
	(−13,74)	(1,18)	(−0,12)	(−0,02)	(8)	
P85	14,90	21,63	25,65	31,60	>203	Não convergiu após 203 iterações
	(−70,20)	(−56,74)	(−48,70)	(−36,80)	(>203)	
P86	23,25	35,93	40,23	42,44	7	
	(−43,22)	(−12,26)	(−1,76)	(3,64)	(5)	

[1] Com relação à taxa interna única, se existir, ou com relação à taxa positiva.

[2] Usando a aproximação de Wild. Entre parênteses, sem a aproximação.

Para os demais casos de taxa interna única no campo geral, à exceção do caso do projeto P53, os resultados foram, em regra, bastante satisfatórios. Já para P53, a convergência revelou-se extremamente lenta e oscilante. Na iteração de ordem 202 a aproximação foi igual a 90,88%, assumindo o valor 110,07% na iteração seguinte[8] ao passo que o valor exato é $i^* = 100\%$.

Quanto aos casos de única taxa interna positiva, havendo porém também taxa negativa, os resultados foram de caráter misto. Além dos casos já apontados de colapso formal, o algoritmo, por diversas vezes, como anteriormente discutido, convergiu para a indesejada taxa negativa. Isso ocorreu para os projetos P2, P8, P21, P57, P78, P79 e P81. Por outro lado, para certos casos bastante semelhantes, como para os projetos P15, P16, P17, P19, P32, P33, P34, P35, P40, P41, P42, P43, P45, P46, P47, P52, P82 e P86, o algoritmo convergiu para a taxa positiva de interesse.

Quanto à estratégia de, no evento de convergência para a taxa negativa indesejada, tomar como ponto de partida uma taxa positiva, infelizmente obtivemos resultados que a desaconselham. Assim, no caso de P57, tomando como ponto de partida $i_1 = 4$, obtivemos $i_2 = 1{,}91$ e $i_3 = -0{,}99$, convergindo a seguir, novamente, para a taxa negativa $-0{,}3028$. Fato semelhante ocorreu quando foi feito $i_1 = 3.5$

Pior resultado ainda ocorreu com o projeto P2. Tanto tomando-se $i_1 = 1$, como $i_1 = 1, 2$, o algoritmo entrou em um ciclo infinito, repetindo sempre o par (0,554903; 8,667192).

6. Conclusão

Evidenciou-se aqui que o procedimento sugerido por Wild (1976) nada mais é do que uma versão estritamente finita do algoritmo originalmente proposto por Boulding (1936). Especificamente, o procedimento de Wild consiste tão-somente em tomar como aproximação final uma média das aproximações respectivamente obtidas nas três primeiras iterações do algoritmo de Boulding. De outro lado, a aplicação estrita do método de Boulding requer que sejam efetuadas tantas iterações quanto as que forem necessárias para que seja alcançada a precisão desejada.

Do ponto de vista teórico, concluiu-se, formalmente, que o algoritmo de Boulding, para o caso de projetos do tipo investimento simples, gera uma seqüência de aproximações que converge para o verdadeiro valor da taxa interna de retorno. Conquanto seja conjeturado que tal resultado estenda-se ao caso geral dos projetos do tipo convencional, verificou-se que o mesmo nem sempre acontece para a classe dos projetos do tipo não-convencional. Em particular, foram identificados casos de colapso formal do algoritmo.

Em uma análise empírica, constatou-se que, na maioria dos casos, ocorreu um desempenho satisfatório. Ainda mais, regra geral, tal desempenho foi tanto

[8] Foram necessárias 34 iterações para passar do par (112,23%; 89,24%) ao par final indicado.

melhor quando, a partir da terceira iteração, tomou-se como ponto de partida a aproximação da média sugerida por Wild. Entretanto, detectou-se não só a existência de casos de convergência excessivamente lenta, como também a possibilidade de, independentemente da especificação da iteração inicial, obter-se sempre uma indesejada solução negativa.

Uma interessante indagação, que será objeto de um outro estudo, é a que diz respeito a uma comparação com o método de Newton-Raphson. Em especial, para as classes de projetos não-convencionais em que este último seja aplicável, qual seria o algoritmo mais eficiente?

Em suma e dependendo da resposta a esta indagação, o algoritmo de Boulding parece ser, *a priori,* uma boa opção. Deve-se, porém, não só estar atento aos casos de colapso formal, como também aos de convergência lenta e de indesejada convergência para uma taxa interna negativa. Nestas últimas eventualidades, sugere-se que seja feito uso do método da bisseção.

Anexo 1

Convergência do algoritmo de Boulding

Paulo Klinger

Será aqui formalmente evidenciado que, para a classe de projetos ditos do tipo investimento simples, o algoritmo de Boulding produz uma seqüência de aproximações que converge para o verdadeiro valor da única taxa interna de retorno i^*.

1. Resultados auxiliares

Defina-se a função Φ: $(0, \infty) \to \mathbb{R}$, monótona e contínua, tal que:

a) $\Phi(x) = 0 \Leftrightarrow x = 1$;

b) existe um único $\zeta > 0$ para o qual $\Phi(\zeta) = 1$.

Então, sendo $A = (0, \infty) - \{1\}$ e $f: A \to (0, \infty)$ definida como

$f(x) = x^{1/\Phi(x)}$

segue-se que f é contínua em A.

Lema 1. A imagem de A pela função f está contida em A.

Demonstração:

Se $x \in A$ e $f(x) = 1$, tem-se que $\log(f(x)) = 0$. Logo, $\log x/\Phi(x) = 0 \Rightarrow \log x \cdot x = 0 \Rightarrow x = 1$. O que é uma contradição com $1 \notin A$.

c.q.d.

Teorema 1

Se $x \in A$ então $\begin{cases} x \leqslant \zeta \Rightarrow f(x) \geqslant x \\ x \geqslant \zeta \Rightarrow f(x) \leqslant x \end{cases}$

e

se $\begin{cases} \Phi \text{ é não-decrescente} \Rightarrow f(A) \subset (1, \infty) \\ \Phi \text{ é não-decrescente} \Rightarrow f(A) \subset (0, 1) \end{cases}$

Demonstração:

a) supondo que Φ seja não-decrescente.

Temos que $\zeta > 1$, pois $(\zeta) = 1 > \Phi(1) = 0$. Por outro lado, tem-se que:

a.1) se $0 < x < 1$, tem-se $\Phi(x) < 0 \Rightarrow 1/\Phi(x) < 0$ e, portanto, $f(x) = x^{1/\Phi(x)} > 1 > x$. Ou seja, nesse caso $f(x) > 1$ e $f(x) > x$;

a.2) se $1 < x \leqslant \zeta$, teremos $0 < \Phi(x) \leqslant 1$. Logo $1/\Phi(x) \geqslant 1$ e, portanto, $f(x) \geqslant x > 1$.

Assim, de *a.1* e *a.2*, decorre que se $x \leqslant \zeta \Rightarrow f(x) \geqslant x$ e $f(x) > 1$;

a.3) se $x \geqslant \zeta$, segue-se que $\Phi(x) \geqslant 1$, logo $0 < 1/\Phi(x) \leqslant 1$, e, portanto, $1 < f(x) = x^{1/\Phi(x)} \leqslant x$.

Deste modo, conclui-se que $f(A) \subset (1, \infty)$ e que $x \leqslant \zeta \Rightarrow f(x) \geqslant x$ e $x \geqslant \zeta \Rightarrow f(x) \leqslant x$;

b) supondo que Φ seja não-crescente.

Nesse caso teremos $\zeta < 1$, pois $\Phi(\zeta) = 1 > \Phi(1) = 0$;

b.1) se $0 < x \leqslant \zeta$, segue-se que $\Phi(x) \geqslant 1 \Rightarrow 1/\Phi(x) \leqslant 1$ e, portanto, $x \leqslant f(x) = x^{1/\Phi(x)} < 1$;

b.2) se $\zeta \leqslant x < 1$, tem-se $0 < \Phi(x) \leqslant 1 \Rightarrow 1/\Psi(x) \geqslant 1$ e, portanto,
$f(x) = x^{1/(x)} \leqslant x < 1$;

b.3) se $x > 1$, tem-se $\Phi(x) < 0 \Rightarrow 1/\Phi(x) < 0$. Logo, teremos
$f(x) = x^{1/(x)} < 1 < x$.

Por conseguinte, sendo Φ não-crescente, ter-se-á sempre $f(x) < 1$ se $x \in A$. Adicionalmente, $x \leqslant \zeta \Rightarrow f(x) \geqslant x$ e $x \geqslant \zeta \Rightarrow f(x) \leqslant x$.

c.q.d.

Teorema 2

Sendo $s \geqslant 0$, seja $x_0 \in A$ e defina-se $x_{s+1} = f(x_s)$. Então, para $(x_s)_{s \geqslant 0}$, que está bem definida pois $f(A) \subset A$, são válidas as seguintes propriedades:

1) se existe $s_0 \geqslant 0$ tal que $x_{s_0} = \zeta$, então $x_s = \zeta$ para todo $s \geqslant s_0$;

2) se $x_s \geqslant \zeta$ para todo $s \geqslant s_1$, para algum $s_1 \geqslant 0$, então existe $\bar{x} = \lim\limits_{s \to \infty} x_s$ e $\bar{x} = \zeta$;

3) se $x_s \leqslant \zeta$ para todo $s \geqslant s_2$, para algum $s_2 \geqslant 0$, então existe $\hat{x} = \lim\limits_{s \to \infty} x_s$ e $\hat{x} = \zeta$.

Demonstração:

1) A prova será efetuada por indução em s, $s \geqslant s_0$.

Por hipótese, $x_s = \zeta$ para $s \geqslant s_0$. Logo, por definição, tem-se $x_{s+1} = f(x_s) = f(\zeta) = \zeta^{1/\Phi(\zeta)} = \zeta$. Por conseguinte, $x_{s+1} = \zeta$ e a indução está completa.

2) Suponha que $x_s \geqslant \zeta$ para todo $s \geqslant s_1$. Então, pelo teorema 1, segue-se que $x_{s+1} = f(x_s) \leqslant x_s$. Logo, $(x_s)_{s \geqslant s_1}$ é monótona e limitada e, por conseguinte, existe o limite $\bar{x} = \lim\limits_{s \to \infty} x_s$. Além do mais, $\bar{x} \geqslant \zeta$, pois que $x_s \geqslant \zeta$ para todo $s \geqslant s_1$.

Observe-se que, se $\zeta > 1$, teremos $\bar{x} > 1$. Por outro lado, se $\zeta < 1$, caso em que Φ é não-crescente com $f(A) \subset (0, 1)$, segue-se que $\bar{x} \leqslant x_{s_1+1} < 1$.

Portanto, em qualquer caso, $\bar{x} > 0$ e $\bar{x} \neq 1$. Assim, $\bar{x} = \lim\limits_{s \to \infty} x_{s+1} = \lim\limits_{s \to \infty} f(x_s) = f(\bar{x})$. Ou seja:

$$\bar{x} = f(\bar{x}) \Rightarrow \log \bar{x} = \frac{\log \bar{x}}{\Phi(\bar{x})} \Rightarrow \Phi(\bar{x}) = 1, \text{ pois } \bar{x} \neq 1.$$

Logo, $\bar{x} = \zeta$.

3) Prova análoga à da segunda propriedade.

c.q.d.

Exemplo:

Seja $\theta(x) = -1 + \sum_{j=1}^{n} a_j x^{-j}$

com $a_j \geqslant 0$, $j = 1, \ldots n-1$ e $a_n > 0$ e, tal que $\theta(1) \neq 0$.

Observe-se que
$\lim_{x \to \infty} \theta(x) = -1$ e $\lim_{x \to 0+} \theta(x) = +\infty$

Então, como $\theta(x)$ é contínua para $x \in (0, \infty)$ e estritamente decrescente nesse intervalo, segue-se que existe um único $\zeta > 0$ tal que $\theta(\zeta) = 0$. Ainda mais, como $\theta(1) \neq 0$, decorre que $\zeta \neq 1$. Da monotonicidade de $\theta(x)$, tem-se também que:

$$\zeta > 1 \Leftrightarrow \theta(1) > 0$$
$$\zeta < 1 \Leftrightarrow \theta(1) < 0$$

Note-se ainda que, como $\theta(\zeta) = 0$, vem que $\sum_{j=1}^{n} a_j \zeta^{-j} = 1$.

Defina-se agora a função:

$$\Phi(x) = 1 - \frac{\log(\theta(x) + 1)}{\log(\theta(1) + 1)}, \quad x > 0.$$

A função Φ está bem definida, pois que $0 < \theta(1) + 1 \neq 1$ e $\theta(x) + 1 = \sum_{j=1}^{n} a_j x^{-j} > 0$. Tem-se as seguintes propriedades:

a) Φ é estritamente crescente se, e somente se, $\theta(1) > 0$;
b) Φ é estritamente decrescente se, e somente se, $\theta(1) < 0$, pois θ é estritamente decrescente e $\theta(1) + 1 = \sum_{j=1}^{n} a_j > 0$;
c) $\Phi(1) = 0$;
d) $\Phi(\zeta) = 1$ e Φ é contínua, pois θ é contínua.

Lema 2

Se $\zeta > 1$, então $x \geqslant \zeta \Rightarrow f(x) \geqslant \zeta$.
Se $\zeta < 1$, então $x \leqslant \zeta \Rightarrow f(x) \leqslant \zeta$.

Demonstração:

Considerem-se as funções

$$\alpha(x) = \frac{\log x}{\log \zeta} + \frac{\log(\theta(x) + 1)}{\log(\theta(1) + 1)}, \; x > 0$$

e

$$\beta(x) = \frac{\log x}{\log \zeta} + \frac{\sum_{j=1}^{n} ja_j \zeta^{-j} (\log(\zeta/x)}{\log(\sum_{j=1}^{n} a_j)}, \; x > 0.$$

Como log x é côncava, $\sum_{j=1}^{n} a_j \zeta^{-j} = 1$ e $a_j \geqslant 0$, $j = 1, \ldots, n$, tem-se que:

$$(1) \quad \log(\sum_{j=1}^{n} a_j x^{-j}) = \log(\sum_{j=1}^{n} \frac{a_j}{\zeta j} (\frac{\zeta}{x})^j) \geqslant \sum_{j=1}^{n} \frac{ja_j}{\zeta^j} \log(\zeta/x), \; x > 0.$$

Observe-se que podemos também escrever:

$$\beta(x) = \frac{\log x}{\log \zeta} + \frac{\log \zeta - \log x}{\log(\sum_{j=1}^{n} aj)} \sum_{j=1}^{n} ja_j \zeta^{-j} =$$

$$= \frac{\log \zeta}{\log(\sum_{j=1}^{n} a_j)} \sum_{j=1}^{n} ja_j \zeta^{-j} + \log x \left[\frac{1}{\log \zeta} - \frac{1}{\log(\sum_{j=1}^{n} a_j)} \sum_{j=1}^{n} ja_j \zeta^{-j} \right]$$

Particularizando-se (1) para o caso onde x = 1, tem-se:

$$\log\left(\sum_{j=1}^{n} a_j\right) - \sum_{j=1}^{n} j a_j \zeta^{-j} \log \zeta \geqslant 0 .$$

Ainda mais, como $\log \zeta \cdot \log\left(\sum_{j=1}^{n} a_j\right) > 0$, já que $\sum_{j=1}^{n} a_j > 1 \Rightarrow \zeta > 1$ e $\sum_{j=1}^{n} a_j < 1 \Rightarrow \zeta < 1$, a expressão acima pode ser escrita como:

$$\frac{1}{\log \zeta} - \frac{1}{\log\left(\sum_{j=1}^{n} a_j\right)} \sum_{j=1}^{n} j a_j \xi^{-j} \geqslant 0.$$

Por conseguinte, dado que a primeira parcela na expressão de β (x) é uma constante e que log x é crescente, conclui-se que β (x) é crescente. Logo, se $x \geqslant \zeta$ teremos $\beta(x) \geqslant \beta(\zeta) = 1$. Se $x \leqslant \zeta$, então $\beta(x) \leqslant 1$.

Caso $\zeta > 1$, o que ocorre se e somente se $\sum_{j=1}^{n} a_j > 1$, se $x \geqslant \zeta$, usando-se a propriedade de concavidade da função logaritmo, vem que:

$$\alpha(x) = \frac{\log x}{\log \zeta} + \frac{\log\left(\sum_{j=1}^{n} a_j x^{-j}\right)}{\log\left(\sum_{j=1}^{n} a_j\right)} \geqslant$$

$$\geqslant \frac{\log x}{\log \zeta} + \frac{\sum_{j=1}^{n} j a_j \zeta^{-j} \log(\zeta/x)}{\log\left(\sum_{j=1}^{n} a_j\right)} = \beta(x) \geqslant 1$$

Caso $\zeta < 1$, o que ocorre se e somente $\sum_{j=1}^{n} a_j < 1$, se $x \leqslant \zeta$ teremos $\alpha(x) \leqslant \beta(x) \leqslant 1$.

Finalmente, suponha-se $\zeta > 1$ e $x \geqslant \zeta$. Nesse caso, tendo presente a definição de $f(x)$, temos:

$$f(x) \geqslant \zeta \Leftrightarrow \frac{\log x}{\Phi(x)} \geqslant \log \zeta \Leftrightarrow \frac{\log x}{\log \zeta} \geqslant \Phi(x) \Leftrightarrow$$

$$\Leftrightarrow \frac{\log x}{\log \zeta} \geqslant 1 - \frac{\log(\theta(x)+1)}{\log(\theta(1)+1)} \Leftrightarrow \alpha(x) \geqslant 1.$$

Como quando $\zeta > 1$ se $x \geqslant \zeta$ temos $\alpha(x) \geqslant 1$, decorre então que $f(x) \geqslant \zeta$. Suponha-se agora $\zeta < 1$ e $x \leqslant \zeta$. Nesse caso, vem:

$$f(x) \leqslant \zeta \Leftrightarrow \frac{\log x}{\log \zeta} \geqslant \log \zeta \Leftrightarrow \frac{\log x}{\log \zeta} \leqslant \Phi(x) \Leftrightarrow \alpha(x) \leqslant 1.$$

Portanto, como quando $\zeta < 1$ se $x \leqslant \zeta$ temos $\alpha(x) \leqslant 1$, conclui-se que $f(x) \leqslant \zeta$.

c.q.d.

Teorema 3

Para $s \in \mathbb{N} \cup \{0\}$, seja a seqüência (x_s) como definida no teorema 2. Então, $\lim_{s \to \infty} x_s = \zeta$.

Demonstração:

a) caso onde $\zeta > 1$.

Se $x_s \leqslant \zeta$ para todo *s*, pelo item 3 do teorema 2 tem-se que $x_s \to \zeta$. Por outro lado, se existir algum $s_0 \geqslant 0$ tal que $x_{s_0} \geqslant \zeta$, fazendo-se uso do lema 2 e de indução finita obtém-se $x_s \geqslant \zeta$ para todo $s \geqslant s_0$. Portanto, $x_s \to \zeta$.

b) caso onde $\zeta < 1$.

A prova é análoga.

c.q.d.

Teorema 4

Nas hipóteses do exemplo.

$$\text{Se} \begin{cases} \zeta > 1 \text{ e } x_0 \geqslant \zeta \text{ então } x_s \downarrow \zeta \\ \zeta < 1 \text{ e } x_0 \leqslant \zeta \text{ então } x_s \uparrow \zeta \end{cases}$$

Demonstração:

a) caso onde $\zeta > 1$ e $x_0 \geqslant \zeta$.

De acordo com o lema 2, e aplicando indução finita, teremos $x_s \geqslant \zeta$ para todo $s \geqslant 0$. De outro lado, pelo teorema 1, tem-se que $x_{s+1} \leqslant x_s$ para todo $s \in \mathbb{N}$;

b) caso onde $\zeta < 1$ e $x_0 \leqslant \zeta$.

A demonstração é análoga.

c.q.d.

2. Aplicação: convergência do algoritmo de Boulding

Seja a função valor atual $V(i) = \sum_{j=0}^{n} a_j(1+i)^{-j}$, para o caso de um projeto dito do tipo investimento simples; isto é, tal que $a_0 < 0$, $a_j \geqslant 0$, $j = 1, \ldots, n-1$ e $a_n > 0$.

No caso não-trivial de taxa interna de retorno diferente de zero, caracterizado pelo fato de que $\sum_{j=0}^{n} a_j \neq 0$, dividindo-se $V(i)$ por $|a_0| > 0$, podemos sempre recair no caso especial em que $a_0 = -1$. Este será o caso considerado.

Defina-se, como no exemplo, a função

$$\theta(x) = -1 + \sum_{j=1}^{n} a_j x^{-j}, \; x = 1 + i, \; x > 0 \text{ (ou } i > -1)$$

e tome-se

$$\Phi(x) = 1 - \frac{\log\left(\sum_{j=1}^{n} a_j x^{-j}\right)}{\log\left(\sum_{j=1}^{n} a_j\right)}$$

Considere-se $x_0 > 0$, $x_0 \neq 1$. Por exemplo, como no passo inicial do algoritmo de Boulding, tome-se

$$x_0 = \left(\sum_{j=1}^{n} a_j\right)^{\sum_{j=1}^{n} a_j / \sum_{j=1}^{n} j a_j}$$

Recursivamente, considere-se

$$x_{s+1} = x_s^{1/\Phi(x_s)}, \; s \in \mathbb{N} \cup \{0\}$$

Então, se $\zeta = 1 + i^*$, onde i^* é tal que $V(i^*) = 0$, do teorema 3 decorre que $x_s \to \zeta$. Ainda mais, se, para o projeto inicial, tivermos $\sum_{j=1}^{n} a_j / |a_0| > 1$, se $x_0 \geqslant \zeta$ ter-se-á $x_s \downarrow \zeta$.

Anexo 2

Programa usado na investigação empírica

```
   45 FORMAT(/,10X,'DEU ALFA IGUAL A ZERO')
      GO TO 1
   48 TAXA1 = (BZERO/CZERO) ** (1./ALFA) - 1.
      V = 1./(1. + TAXA1)
      CALL VALOR(V,BJ,CJ,NF,VB,VC)
      TAXAP = 100. * TAXA1
      WRITE(MSAI,50)
   50 FORMAT(/,10X,'PRIMEIRAS 3 ITERACOES ATE WILD',/)
      ITER = 1
      WRITE(MSAI,60) ITER,TAXAP
   60 FORMAT(/,5X,'ITERACAO = ',I1,5X,'TAXA PERCENTUAL = ',F10.6)
      DO 80 L = 1,2
      BETA = 1. - FLOG(VB,VC)/CONST
      TAXA =(1. + TAXAP/100.)**(1./BETA) - 1.
      V = 1./(1. + TAXA)
      CALL VALOR(V,BJ,CJ,NF,VB,VC)
      TAXAP = 100. * TAXA
      ITER  = ITER + 1
      WRITE(MSAI,60) ITER,TAXAP
      APRX(L) = TAXA
   80 CONTINUE
      TAXAW = TAXA1 + (APRX(1)-TAXA1)**2/(2.*APRX(1)-TAXA1-APRX(2
      TAXAP = 100. * TAXAW
      WRITE(MSAI,90) TAXAP
   90 FORMAT(10X,'APROXIMACAO PERCENTUAL DE WILD = ',F10.6)
      VBOULD = VB - VC
      BETAB = 1. - FLOG(VB,VC)/CONST
      V = 1./(1.+TAXAW)
      CALL VALOR(V,BJ,CJ,NF,VB,VC)
      BETAW = 1. - FLOG(VB,VC)/CONST
      VWILD = VB - VC
      WRITE(MSAI,95) VBOULD,VWILD
   95 FORMAT(5X,'VBOULD = ',F15.6,/,5X,'VWILD = ',F15.6)
      IF(ABS(VBOULD).LE.TOL1 .OR.ABS(VWILD).LE.TOL1) GO TO 1
      TAXAB = APRX(2)
         ITERACOES ADICIONAIS
  100 MAX = 200
      IF(TAXAI .EQ. 0.0) GO TO 110
      TAXAB = TAXAI
      V = 1./(1.+TAXAB)
      CALL VALOR (V,BJ,CJ,NF,VB,VC)
      BETAB = 1.-FLOG(VB,VC)/CONST
      TAXAW = TAXAI
      V = 1./(1.+TAXAW)
      CALL VALOR(V,BJ,CJ,NF,VB,VC)
      BETAW = 1.- FLOG(VB,VC)/CONST
  110 KCONT = KCONT + 1
      IF(KCONT .GT. MAX) GO TO 1
      TAXABP = 100.* TAXAB
      TAXAWP = 100.* TAXAW
      TAXAB = (1.+ TAXAB)**(1./BETAB) - 1.
      DTAXAB =TAXAB - TAXABP/100.
      V = 1./(1.+ TAXAB)
      CALL VALOR(V,BJ,CJ,NF,VB,VC)
      VBOULD = VB - VC
      BETAB = 1. - FLOG(VB,VC)/CONST
```

```
      PROGRAM BD*WD
C        COMPARACAO ENTRE BOULDING E WILD
      DIMENSION AJ(101),BJ(101),CJ(101),APRX(2)
      FLOG(X1,X2) = ALOG10(X1/X2)
      TOL1 = 0.0001
      TOL2 = 0.000001
      MENT = 1
      MSAI = 2
    1 WRITE(1,11)
   11 FORMAT('1','  NOME   N    TAXAI',/,' ========iiiffffffffff',
     1/)
      READ(MENT,2) CNOME,CONT,N,TAXAI
C        N E' O No. DE PERIODO DE VIDA DO PROJETO
C        TAXAI, SE DIFERENTE DE ZERO POR CENTO, SERA TOMADA COMO PONTO
C         DE PARTIDA
    2 FORMAT(2A4,I3,F10.0)
      IF(N.LE.0) GO TO 1000
      NF = N + 1
      WRITE(MSAI,5) CNOME,CONT,TAXAI
    5 FORMAT('1',15X,'PROJETO ',2A4,' COM TAXA INICIAL = ',F10.4,' POR C
     1ENTO POR PERIODO',//)
      BZERO = 0.0
      CZERO = 0.0
      BBAR  = 0.0
      CBAR  = 0.0
      TAXAI = TAXAI/100.
      WRITE(1,22)
   22 FORMAT(9X,'fffF10.0ff')
      DO 40 K = 1,NF
      I = K - 1
      WRITE(1,33) I
   33 FORMAT(' AJ(',I3,')=')
      READ(MENT,10) AJ(K)
   10 FORMAT(F10.0)
      WRITE(MSAI,15) I,AJ(K)
   15 FORMAT(10X,'AJ(',I3,') = ',F10.2)
      IF(AJ(K).LT.0.0) GO TO 20
      BJ(K) = AJ(K)
      CJ(K) = 0.0
      GO TO 30
   20 BJ(K) = 0.0
      CJ(K) = -AJ(K)
   30 BZERO = BZERO + BJ(K)
      CZERO = CZERO + CJ(K)
      BBAR = BBAR + I * BJ(K)
   40 CBAR = CBAR + I * CJ(K)
      CONST = FLOG(BZERO,CZERO)
      KCONT = 0
      IF(TAXAI.NE.0.0) GO TO 100
      ALFA = BBAR/BZERO - CBAR/CZERO
      IF(ALFA.NE.0.0) GO TO 48
      WRITE(MSAI,45)
```

```
      TAXAW = (1.+ TAXAW)**(1./BETAW) - 1.
      DTAXAW = TAXAW - TAXAWP/100.
      V = 1./(1.+ TAXAW)
      CALL VALOR(V,BJ,CJ,NF,VB,VC)
      VWILD = VB - VC
      BETAW = 1. - FLOG(VB,VC)/CONST
      TAXB = 100.* TAXAB
      TAXW = 100.* TAXAW
      WRITE(MSAI,120) KCONT,TAXB,TAXW
  120 FORMAT(/,5X,'ITERACAO = ',I3,5X,'TAXAB = ',F10.6,5X,'TAXAW = ',
     1F10.6)
      WRITE(MSAI,95) VBOULD,VWILD
      IF(ABS(VBOULD).LE.TOL1 .OR. ABS(VWILD).LE.TOL1) GO TO 1
      IF(ABS(DTAXAB).LE.TOL2 .OR. ABS(DTAXAW).LE.TOL2) GO TO 1
      GO TO 110
 1000 WRITE(MSAI,1001)
 1001 FORMAT(///,15X,'FIM DE PAPO')
      STOP
      END
      SUBROUTINE VALOR(V,B,C,N,VB,VC)
      DIMENSION B(N),C(N)
      VB = B(N)
      VC = C(N)
      N1 = N - 1
      DO 50 K=1,N1
      J = N - K
      VB = V*VB + B(J)
   50 VC = V*VC + C(J)
      RETURN
      END
A>
```

Referências bibliográficas

Barros Santos, Vitoriano Ruas. *Curso de cálculo numérico*. 4. ed. Rio de Janeiro, Livros Técnicos e Científicos, 1982.

Boulding, K. E. Time and investment. *Economica, 3* (10): 196-200, May 1936.

de Faro, Clovis. O critério da taxa interna de retorno e o caso dos projetos do tipo investimento puro. *Revista de Administração de Empresas, 16* (5): 57-63, set./out. 1976.

__________. *Determinação da taxa de juros implícita em esquemas genéricos de financiamento;* comparação entre os algoritmos de Wild e de Newton-Raphson. Rio de Janeiro, Escola de Pós-Graduação em Economia do Instituto Brasileiro de Economia da Fundação Getulio Vargas, 1978.

__________. Closed-form expressions for the approximate evaluation of interest rates extensions to the geometric sequence of payments case. *The Engineering Economist, 27* (1): 80-9, Fall 1981.

__________. Determinação da taxa de rentabilidade de letras de câmbio. *Revista Brasileira de Mercado de Capitais, 8* (24): 191-9, set./dez. 1982.

__________. O Teorema de Vincent e o problema de multiplicidade de taxas internas de retorno. *Revista Brasileira de Economia, 37* (1): 55-76, jan./mar. 1983.

__________ & de Mello e Souza, Alberto. O uso do critério da taxa interna de retorno e sua aplicação em investimentos educacionais. *Estudos Econômicos, 5* (3): 37,64, set./dez. 1975.

__________ & Soares, Luiz. A aplicabilidade do critério da taxa interna de retorno. *Pesquisa e Planejamento Econômico, 6* (3): 587-617, dez. 1976.

__________ & __________. A flexible sufficient condition for a unique nonnegative internal rate of return. *The Engineering Economist 23* (2): 117-27, Winter 1978.

Fisher, Lawrence. An algorithm for finding exact rates of return. *The Journal of Business of the University of Chicago, 39* (1): 111-8, part II, Jan. 1966.

Hawawini, Gabriel A. & Vora, Ashok. Yeld approximations: a historical perspective. *The Journal of Finance, 37* (1): 145-56, Mar. 1982.

Henrici, Peter. *Elements of numerical analysis.* New York, John Wiley, 1964.

Jean, William H. On multiple rates of return. *The Journal of Finance, 23* (1): 187-91, Mar. 1968.

Kaplan, Seymour. Computer algorithms for finding exact rates of return. *The Journal of Business of the University of Chicago, 40* (4): 389-92, Oct. 1967.

Wild, N. H. Return on investment made easy. *Chemical Engineering,* (12): 153-4, Apr. 1976.

Witten, Peer & Zimmermann, Horst Günther. Zur Eindeutigkeit des internen Zinssatzes und seiner numerischen Bestimmung. *Zeitschrift für Betriebswirtschaft, 47* (2): 99-114, Feb. 1977.