金融財務研究会
アメリカンオプション入門
(Excel編)

２０２１年４月９日

森谷博之

Quasars22 Private Limited

オプション取引

オプション・プレミアム

買い方と売り方の需給でオプション・プレミアムは決まる。
そのもとになる価値は理論的に5つの要素で決まる。

原資産価格
一般的に原資産価格が上昇すればコールが高くなり、プットは安くなる。
逆に原資産価格が下降すればコールは安くなり、プットは高くなる。
権利行使価格
コールもプットもOTMならば権利行使価格に近づくほど高くなる。
逆に権利行使価格から離れるほど低くなる。ITMに入ると逆になる。
満期までの時間
満期までの時間が長ければ、原資産が権利行使価格に達する確率が高くなり、プレミアムは高くなる。
金利・配当(外国金利)
金利が上がればプレミアムは下がり、配当が高ければプレミアムは上がる。
ボラティィリティ
ボラティィリティが高ければ、プレミアムは高くなる。

オプション取引

ブラックショールズ方程式の一般化

原資産の価格Sは

$$\frac{dS}{S} = \mu dt + \sigma dz$$

にしたがうとする。ここでdzはウィナー過程、これにキャリーコストを含むと

$$\frac{dF}{F} = (\mu - b)dt + \sigma dz$$

となる。したがって、オプションの価格Vの時間発展を表現する偏微分方程式は

$$2\sigma^2 S^2 V_{ss} + bSV_s - rV + V_t = 0$$

となる。

ブラック・ショールズ・モデル

一般化

スポット価格	s
行使価格	k
ボラティリティ	σ
資金調達費用	r
配当、外国金利等	q
キャリーコスト	b=r-q
満期・行使日までの期間	T

b=r	ブラック株価オプション
b=r-q	連続配当付き株価オプション
b=0	先物オプション
b=0,r=0	マージン先物オプション
$b=r-r_f$	通貨オプション

ブラック・ショールズ・モデル

$$C(s,k,\sigma,r,b,T)=se^{-(b-r)T}N(d_1)-ke^{-rT}N(d_2)$$

$$P(s,k,\sigma,r,b,T)=ke^{-rT}N(-d_2)-se^{-(b-r)T}N(-d_1)$$

$$\mathrm{N}(z)=z=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{z}e^{-\frac{y^2}{2}}\mathrm{d}y$$

$$d_1=\frac{\log\frac{s}{k}+(b+\sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_2=\frac{\log\frac{s}{k}+(b-\sigma^2/2)T}{\sigma\sqrt{T}}=d_1-\sigma\sqrt{T}$$

ブラック・ショールズ・モデル

スポット価格	s
行使価格	k
ボラティリティ	σ
資金調達費用	r
配当、外国金利等	q
キャリーコスト	b=r-q
満期・行使日までの期間	T

bnm

	A	B
1	s	100
2	k	105
3	t	0.5
4	r	0.08
5	b	0.08
6	v	0.3
7	BSCALL	8.050102

bscall

一般化

bnm

```
Function bscall(s, k, t, r, b, v) As Double
    Dim d1, d2, ss, kk As Double
    d1 = (Log(s / k) + (b + v ^ 2 / 2) * t) / (v * Sqr(t))
    d2 = d1 - v * Sqr(t)
    ss = s * Exp((b - r) * t) * Application.NormSDist(d1)
    kk = k * Exp(-r * t) * Application.NormSDist(d2)
    bscall = ss - kk
End Function
```

$$\mathrm{N}(z) = z = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{z} e^{-\frac{y^2}{2}}\mathrm{d}y$$

$$C(s,k,\sigma,r,b,T)=se^{-(b-r)T}N(d_1)-ke^{-rT}N(d_2)$$

オプション価格モデル

- 基本的な考え方
  - 無リスクな裁定取引が不可能
    - 裁定取引が不可能であれば、ある株式について
      - 現在の株価
      - 期末の株価の取るべき値
      - 行使価格
      - 借入の費用

という所与の情報があれば、コールの価値を導き出すことができる。

オプション価格モデル

- 二項ランダムウォーク
  - ベルヌーイ過程：価格が独立に上昇するか下落するかだけからなり、その独立試行を繰り返すこと。

  - 二項過程：X_iを上昇・下落ならなるベルヌーイ試行としたときに、その部分和
    $S_0=0$
    $S_1= X_1$
    $S_2= X_1 + X_2$
    $S_n= X_1 +,,,+ X_n$
    のこと。
  - 二項分布： 確率変数S_nがしたがう確率分布のこと。

オプション価格モデル

- 二項ランダムウォーク

	A	B	C	D	E	F	G
1							
2	S: 株価の初期値						
3	u: 株価の上昇率+1						
4	d: 株価の下落率+1						
5						Su^5	
6					Su^4		
7				Su^3		Sd^1u^4	
8			Su^2		Sd^1u^3		
9		Su^1		Sd^1u^2		Sd^2u^3	
10	S		Su^1d^1		Sd^2u^2		
11		Sd^1		Sd^2u^1		Sd^3u^2	
12			Sd^2		Sd^3u^1		
13				Sd^3		Sd^4u^1	
14					Sd^4		
15						Sd^5	
16							

$X_n \geq u^a d^{n-a}$となる確率は

$$\Phi[a;n,q] = \sum_{j=a}^{n} \left(\frac{n!}{j!\,(n-j)!} \right) q^j \,(1-q)^{n-j}$$

となる。 jは上昇の数$n-j$は下落の数、 総数は$n = j + n - j$

オプション価格モデル

- 二項オプションモデル

	A	B	C	D	E	F	G
1							
2	S: 株価の初期値						
3	u: 株価の上昇率+1						
4	d: 株価の下落率+1						
5						Su^5	
6					Su^4		
7				Su^3		Sd^1u^4	
8			Su^2		Sd^1u^3		
9		Su^1		Sd^1u^2		Sd^2u^3	
10	S		Su^1d^1		Sd^2u^2		
11		Sd^1		Sd^2u^1		Sd^3u^2	
12			Sd^2		Sd^3u^1		
13				Sd^3		Sd^4u^1	
14					Sd^4		
15						Sd^5	
16							

満期時の
$\mathrm{Max}(Su^j d^{n-j}-k,0)$
の期待値が
オプション価格
と関連する。

コールオプションの価格は

$$C = \sum_{j=a}^{n} \left(\frac{n!}{j!\,(n-j)!} \right) q^j\,(1-q)^{n-j} \left[u^j d^{n-j} S - K \right] / r^n$$

となる。

オプション価格モデル

- 株式の期待収益率とリスク
  - 期待収益率　$m_s \equiv qu + (1-q)d$
  - 分散　$v_s^2 \equiv q(u-m_s)^2 + (1-q)(d-m_s)^2$
  - 標準偏差 $v_s = [q(1-q)\ (u-d)^2]^{1/2}$
- リスク中立確率
  - 株価がSuまたはSdになるとき、無リスク金利$r-1>0$、$Su > rS > Sd$であれば、上昇時のリスク中立確率は

$$\mathrm{p} = \frac{rS - Sd}{Su - Sd}$$

  - 投資家がリスク中立な場合pはqの均衡値

オプション価格モデル

- 二項ツリーヨーロピアンオプションモデル

	A	B	C	D	E	F	G
1							
2	S: 株価の初期値						
3	u: 株価の上昇率+1						
4	d: 株価の下落率+1						
5						Su^5	
6					Su^4		
7				Su^3		Sd^1u^4	
8			Su^2		Sd^1u^3		
9		Su^1		Sd^1u^2		Sd^2u^3	
10	S		Su^1d^1		Sd^2u^2		
11		Sd^1		Sd^2u^1		Sd^3u^2	
12			Sd^2		Sd^3u^1		
13				Sd^3		Sd^4u^1	
14					Sd^4		
15						Sd^5	
16							

満期時のみ評価すればよい

オプション価格モデル

- 二項ツリーヨーロピアンオプションモデル

```
Function bnmcall(s, k, t, r, b, v, n) As Double
    Dim u, d, p, sum, dt, a As Double
    Dim j As Long
    dt = t / n
    u = Exp(v * Sqr(dt))
    d = 1 / u
    p = (Exp(b * dt) - d) / (u - d)
    a = Int(Log(k / (s * d ^ n)) / Log(u / d)) + 1
    sum = 0
    For j = a To n
        sum = sum + Application.Combin(n, j) * p ^ j * (1 - p) ^ (n - j) * (s * u ^ j * d ^ (n - j) - k)
    Next

    bnmcall = sum * Exp(-r * t)

End Function
```

dt=t/n

$$u = \exp\left(v\sqrt{t}\right)$$

$$d = \exp\left(-v\sqrt{t}\right) = \frac{1}{u}$$

$$p = \frac{\exp(b \times dt) - d}{u - d}$$

	A	B	C	D	E
1	二項ツリーオプションモデル				
2	s	100	100	100	100
3	k	105	105	105	105
4	t	0.5	0.5	0.5	0.5
5	r	0.08	0.08	0.08	0.08
6	b	0.08	0.08	0.08	0.08
7	v	0.3	0.3	0.3	0.3
8	n	5	10	100	1000
9		8.214507	8.213075	8.048668	8.051718
10	解析解	8.050102			

bnmcall

オプション価格モデル

- 二項ツリーヨーロピアンオプションモデル

	A	B	C	D	E	F	G
1							
2	S: 株価の初期値						
3	u: 株価の上昇率+1						
4	d: 株価の下落率+1						
5						Su^5	
6					Su^4		
7				Su^3		Sd^1u^4	
8			Su^2		Sd^1u^3		
9		Su^1		Sd^1u^2		Sd^2u^3	
10	S		Su^1d^1		Sd^2u^2		
11		Sd^1		Sd^2u^1		Sd^3u^2	
12			Sd^2		Sd^3u^1		
13				Sd^3		Sd^4u^1	
14					Sd^4		
15						Sd^5	
16							

bnm

```
Function bnm2call(s, k, t, r, b, v, n) As Double
    Dim ovalue() As Double
    ReDim ovalue(0 To n + 1)
    Dim u, d, p, sum, dt, df As Double
    Dim i, j As Long
    dt = t / n
    u = Exp(v * Sqr(dt))
    d = 1 / u
    p = (Exp(b * dt) - d) / (u - d)
    df = Exp(-r * dt)
    For i = 0 To n
        ovalue(i) = Application.Max(s * u ^ i * d ^ (n - i) - k, 0)
    Next
    For j = n - 1 To 0 Step -1
        For i = 0 To j
            ovalue(i) = (p * ovalue(i + 1) + (1 - p) * ovalue(i)) * df
        Next
    Next
    bnm2call = ovalue(0)
End Function
```

ovalue(0) = (p * ovalue(1) + (1 - p) * ovalue(0)) * df

後ろから前に計算する

オプション価格モデル

- 二項ツリーヨーロピアンオプションモデル

	A	B	C	D	E	F	G
1							
2	S: 株価の初期値						
3	u: 株価の上昇率+1						
4	d: 株価の下落率+1						
5						Su^5	
6					Su^4		
7				Su^3		Sd^1u^4	
8			Su^2		Sd^1u^3		
9		Su^1		Sd^1u^2		Sd^2u^3	
10	S		Su^1d^1		Sd^2u^2		
11		Sd^1		Sd^2u^1		Sd^3u^2	
12			Sd^2		Sd^3u^1		
13				Sd^3		Sd^4u^1	
14					Sd^4		
15						Sd^5	
16							

bnm

	A	B	C	D	E	F
1	二項ツリーオプションモデル					
2	s	100	100	100	100	100
3	k	105	105	105	105	105
4	t	0.5	0.5	0.5	0.5	0.5
5	r	0.08	0.08	0.08	0.08	0.08
6	b	0.08	0.08	0.08	0.08	0.08
7	v	0.3	0.3	0.3	0.3	0.3
8	n	5	10	100	1000	10000
9		8.214507	8.213075	8.048668	8.051718	8.050295
10	解析解	8.050102				

bnm2call

ovalue(0) = (p * ovalue(1) + (1 - p) * ovalue(0)) * df

後ろから前に計算する

オプション価格モデル

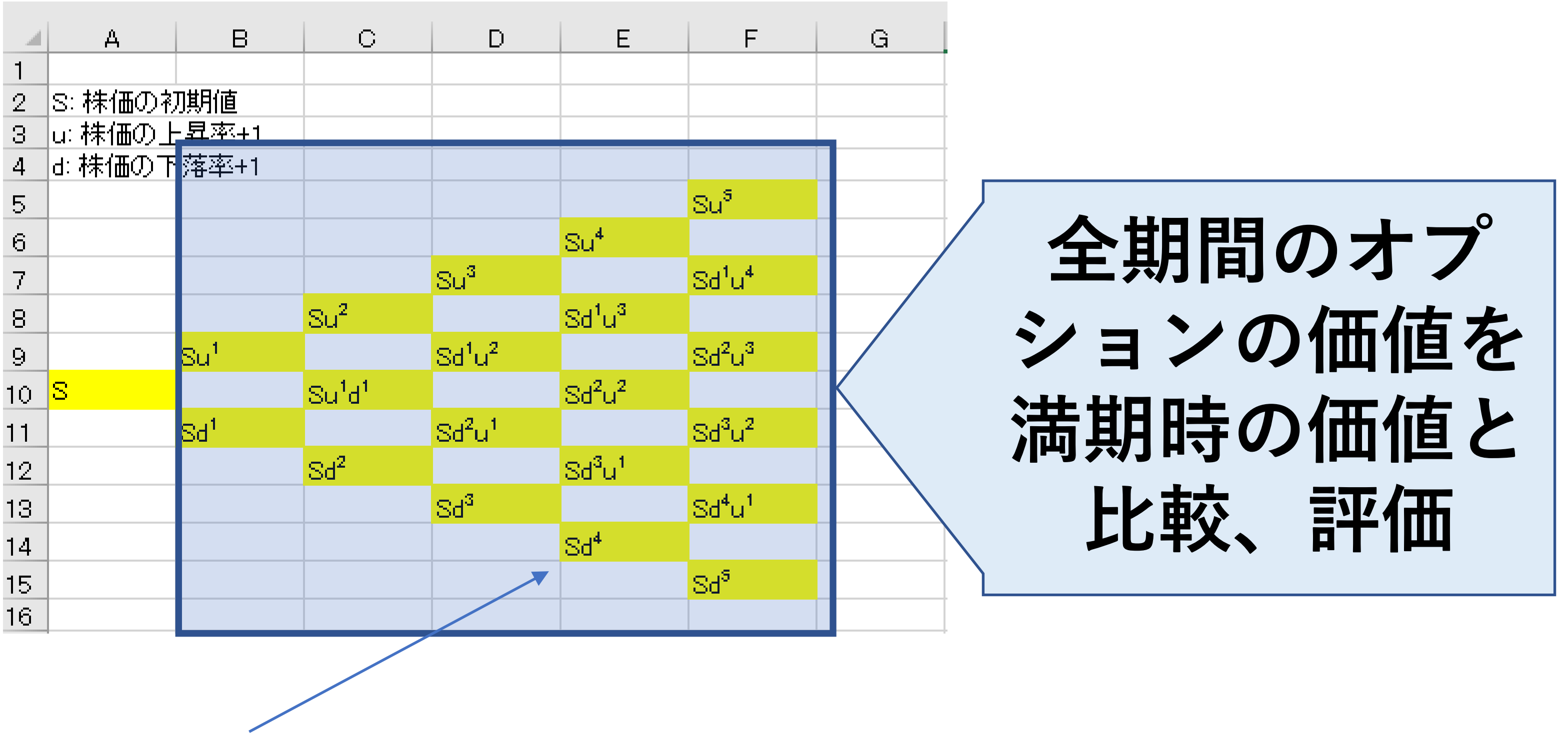
- アメリカンオプション
  - アメリカンオプションは、満期前であればいつでも行使できる。
  - $b < r$のときに多くのオプションは取引されている。
  - 原資産の価格Ｓが行使価格Kに対して著しく高い場合、$N(d_1)$と$N(d_2)$ は１に近づきヨーロピアンの価値は
$$Se^{(b-r)T} - Ke^{-rT}$$
に近づく。ヨーロピアンのオプション価値よりも高い$S - K$を得るためにアメリカンオプションはその場で行使される。したがって、アメリカンオプションには早期行使のプレミアムが付く。
  - $b \geq r$のとき、たとえば、配当が無く$b = r$となる株式において、どのような価格で早期行使しても、それにより得られるものよりもヨーロピアンオプションの価格の下限が高いので行使されることはない。

オプション価格モデル

- 二項ツリーアメリカンオプションモデル
  - Cox-Ross-Rubinstein （１９７９）’Option pricing: A simplified approach’ Journal of Financial Economics
  - アメリカンオプションの価格の評価は、満期前であればいつでも行使できる自由度から、ヨーロピアンオプションに比べて、難しくなる。
  - アメリカンオプションの解析解をもともめることは特別な場合を除いて不可能。
  - アメリカンオプションの近似解は実用上便利であるために良く用いられる。

オプション価格モデル

- 二項ツリーアメリカンオプションモデル



$[pSd^4u^1 + (1-p)Sd^5]$ か$Sd^4u^0 - k$のどちらか大きいほう ； j=4,i=0

オプション価格モデル

- 二項ツリーアメリカンオプションモデル

bnm

```
Function bnmamrcall(s, k, t, r, b, v, n) As Double
    Dim ovalue() As Double
    ReDim ovalue(0 To n + 1)
    Dim u, d, p, sum, dt, df, x, y As Double
    Dim i, j As Long
    dt = t / n
    u = Exp(v * Sqr(dt))
    d = 1 / u
    p = (Exp(b * dt) - d) / (u - d)
    df = Exp(-r * dt)
    For i = 0 To n
        ovalue(i) = Application.Max((s * u ^ i * d ^ (n - i) - k), 0)
    Next
    For j = n - 1 To 0 Step -1
        For i = 0 To j
            x = (s * u ^ i * d ^ (j - i) - k)
            y = (p * ovalue(i + 1) + (1 - p) * ovalue(i)) * df
            ovalue(i) = Application.Max(x, y)
        Next
    Next
    bnmamrcall = ovalue(0)
End Function
```

満期時のペイオフ。

J=4から始まります。

i=0から始まります。

xかyの大きいほうを選択

オプション価格モデル

- 二項ツリーアメリカンオプションモデル

$[p \times \mathrm{ovalue}(0) + (1-p) \times \mathrm{ovalue}(1)]$か $Sd^3u^0 - k$のどちらか大きいほう=ovalue(0)

	A	B	C	D	E	F	G
1							
2	S: 株価の初期値						
3	u: 株価の上昇率+1						
4	d: 株価の下落率+1						
5						Su^5	
6					Su^4		
7				Su^3		Sd^1u^4	
8			Su^2		Sd^1u^3		
9		Su^1		Sd^1u^2		Sd^2u^3	
10	S		Su^1d^1		Sd^2u^2		
11		Sd^1		Sd^2u^1		Sd^3u^2	
12			Sd^2		Sd^3u^1		
13				Sd^3		Sd^4u^1	
14					Sd^4		
15						Sd^5	
16							

全期間のオプションの価値を満期時の価値と比較、評価

$[pSd^4u^1 + (1-p)Sd^5]$ か$Sd^4u^0 - k$のどちらか大きいほう=ovalue(0)

オプション価格モデル

- 二項ツリーアメリカンオプションモデル

bnm

	A	B	C	D	E	F
1	二項ツリーオプションモデル					
2	s	100	100	100	100	100
3	k	105	105	105	105	105
4	t	0.5	0.5	0.5	0.5	0.5
5	r	0.08	0.08	0.08	0.08	0.08
6	b	0.08	0.08	0.08	0.08	0.08
7	v	0.3	0.3	0.3	0.3	0.3
8	n	5	10	100	1000	10000
9		8.214507	8.213075	8.048668	8.051718	8.050295
10	解析解	8.050102				

bnmamrcall

オプション価格モデル

- 二項ツリーアメリカンオプションモデル
  - Rendelman and Bartter(1979)'Two-State Option Pricing,' Journal of Finance
  - $u = e^{\left(b - \frac{\sigma^2}{2}\right)\Delta t + \sigma\sqrt{\Delta t}}$
  - $d = e^{\left(b - \frac{\sigma^2}{2}\right)\Delta t - \sigma\sqrt{\Delta t}}$
  - $p = 0.5$

オプション価格モデル

- 二項ツリーアメリカンオプションモデル
  - Leisen-Reimer(1996)'Binomial Models for Option Valuation Examining and Improving Convergence,' Applied Mathematical Finance
  - $u = e^{b\Delta t \frac{h(d_1)}{h(d_2)}}$
  - $d = \frac{e^{b\Delta t} - pu}{1-p}$
  - $p = h(d_2)$
  - $h(x) = 0.5 + \gamma \sqrt{0.25 - 0.25 \exp\left[-\left(\frac{x}{n+\frac{1}{3}}\right)^2 \left(n + \frac{1}{6}\right)\right]}$
  - $x \geq 0$, then γ=1 else γ=-1

オプション価格モデル

- アメリカンオプションの解析解
  - Barone-Adesi and Whaley(1987) ‘Efficient Analytic Approximationof American Option Values’, Journal of Finance
    - Stoll and Whaley(1986)’The New Option Instruments: Arbitragable Linkages and Valuation.’ Advances in Futures and Options Research I
      - 早期オプション行使の条件
    - MacMillan(1986)’Analytic Approximation for American Put Option.’ Advances in Futures and Options Research 1
      - アメリカンプットオプションの近似解
  - Bjerksund andStenland(1993)’Closed-Form Approximation of American Options’ Scandinavian Journal of Management
    - より正確な評価を実現
  - Bjerksund andStenland(2002)’Closed-Form Valuation of American Options’, Working paper NHH
    - さらに正確な評価を実現

オプション価格モデル

- Barone-Adesi and Whaleyアメリカンオプションの解析解
  - $C(S,K,T) = \begin{cases} C_{BSM}(S,K,T) + A_2(\frac{S}{S^*})^{q_2} \text{ when } S < S^* \\ S - K \text{ when } S \geq S^* \end{cases}$
  - $A_2 = \frac{S^*}{q_2}\{1 - \exp[(b-r)T]\, N[d_1(S^*)]\}$
  - $d_1(S) = \frac{\log\frac{s}{k} + (b + \sigma^2/2)T}{\sigma\sqrt{T}}$
  - $q_2 = \frac{-(2b/\sigma^2 - 1) + \sqrt{(2b/\sigma^2-1)^2 + 8r/\sigma^2/(1-\exp(-rT)}}{2}$

analytic_call

	A	B	C	D	E	F	G
1	s	100	100	100	100	100	100
2	k	100	100	100	100	100	100
3	t	0.1	0.1	0.1	0.5	0.5	0.5
4	r	0.1	0.1	0.1	0.1	0.1	0.1
5	b	0	0	0	0	0	0
6	v	0.15	0.25	0.35	0.15	0.25	0.35
7	BSCALL	1.876925	3.127687	4.377673	4.084124	6.801348	9.510313

BAWamrcall

オプション価格モデル

- Barone-Adesi and Whaleyアメリカンオプションの解析解
  - $C(S,K,T) = \begin{cases} C_{BSM}(S,K,T) + A_2(\frac{S}{S^*})^{q_2} \text{ when } S < S^* \\ S - K \text{ when } S \geq S^* \end{cases}$
  - $A_2 = \frac{S^*}{q_2}\{1 - \exp[(b-r)T]\, N[d_1(S^*)]\}$
  - $d_1(S) = \frac{\log\frac{S}{k} + \left(b + \sigma^2/2\right)T}{\sigma\sqrt{T}}$
  - $q_2 = \frac{-\left(2b/\sigma^2 - 1\right) + \sqrt{(2b/\sigma^2-1)^2 + 8r/\sigma^2/(1-\exp(-rT)}}{2}$
  - $s^* - K = c(S^*, K, T) + \{1 - \exp[(b-r)T]\, N[d_1(S^*)]\}S^* \frac{1}{q_2}$
  - $LHS(S_i) = S_i - K$
  - $RHS(S_i) = c(S_i, K, T) + \{1 - \exp[(b-r)T]\, N[d_1(S_i)]\}S_i \frac{1}{q_2}$

これらの方程式の解はNewton-Raphsonアルゴリズムにより求める。

オプション価格モデル

- Barone-Adesi and Whaleyアメリカンオプションの解析解

```
Function BAWamrcall(s, k, t, r, b, v) As Double
    Dim sk, n, x As Double
    Dim d1, q2, a2 As Double
    If b >= r Then
        BAWamrcall = bscall(s, k, t, r, b, v)
    Else
        sk = kc(k, t, r, b, v)
        n = 2 * b / v ^ 2
        x = 2 * r / (v ^ 2 * (1 - Exp(-r * t)))
        d1 = (Log(sk / k) + (b + v ^ 2 / 2) * t) / (v * Sqr(t))
        q2 = (-(n - 1) + Sqr((n - 1) ^ 2 + 4 * x)) / 2
        a2 = (sk / q2) * (1 - Exp((b - r) * t) * Application.NormSDist(d1))
        If s < sk Then
            BAWamrcall = bscall(s, k, t, r, b, v) + a2 * (s / sk) ^ q2
        Else
            BAWamrcall = s - k
        End If
    End If
End Function
```

オプション価格モデル

- Barone-Adesi and Whaleyアメリカンオプションの解析解

```
Function kc(x, t, r, b, v) As Double
    Dim n, m, su, si, h2, k As Double
    Dim d1, q2, q2u As Double
    Dim lhs, rhs As Double
    Dim bi, ee As Double
    n = 2 * b / v ^ 2
    m = 2 * r / v ^ 2
    q2u = (-(n - 1) + Sqr((n - 1) ^ 2 + 4 * m)) / 2
    su = x / (1 - 1 / q2u)
    h2 = -(b * t + 2 * v * Sqr(t)) * x / (su - x)
    si = x + (su - x) * (1 - Exp(h2))
    k = 2 * r / (v ^ 2 * (1 - Exp(-r * t)))
    d1 = (Log(si / x) + (b + v ^ 2 / 2) * t) / (v * Sqr(t))
    q2 = (-(n - 1) + Sqr((n - 1) ^ 2 + 4 * k)) / 2
    lhs = si - x
    rhs = bscall(si, x, t, r, b, v) + (1 - Exp((b - r) * t) * Application.NormSDist(d1)) * si / q2
    bi = Exp((b - r) * t) * Application.NormSDist(d1) * (1 - 1 / q2) + (1 - Exp((b - r) * t) * Application.NormSDist(d1) / (v * Sqr(t))) / q2
    ee = 0.000001
    While Abs(lhs - rhs) / x > ee
        si = (x + rhs - bi * si) / (1 - bi)
        d1 = (Log(si / x) + (b + v ^ 2 / 2) * t) / (v * Sqr(t))
        lhs = si - x
        rhs = bscall(si, x, t, r, b, v) + (1 - Exp((b - r) * t) * Application.NormSDist(d1)) * si / q2
        bi = Exp((b - r) * t) * Application.NormSDist(d1) * (1 - 1 / q2) + (1 - Exp((b - r) * t) * Application.NormSDist(d1) / (v * Sqr(t))) / q2
    Wend
    kc = si
End Function
```

オプション価格モデル

- モンテカルロシミュレーションによるアメリカンオプション
  - Boyle, Broadie,and Glasserman(1997)'Monte Carlo Mehods for Security Pricing,' Journal ofEconomics Dynamics and Controlのモンテカルロシミュレーションを試みる。
  - モデルはMeier(2000)'Implementing the Broadie-Glasserman Approachfor Pricing Multi-Asset American Options Using Monte Carlo Simulation,' Semesterarbeit, University of Zurichによりプログラムされたもの。
  - 早期行使できる機会をｍ
  - 枝分かれできる機会をbranch
  - シミュレーションの回数をｎとする。

オプション価格モデル

- モンテカルロシミュレーションによるアメリカンオプション

```
Function BGmcamrcall(s, k, t, r, b, v As Double, m, branches, n As Long) As Double
    Dim drift, vsqrdt, discdt As Double
    Dim i, j, i1, i2 As Long
    Dim estimator, simulation As Long
    Dim estimatorsum, sum1, sum2 As Double
    Dim w() As Long
    Dim ss() As Double
    Dim estimators() As Double
    ReDim w(1 To m) As Long
    ReDim ss(1 To branches, 1 To m) As Double
    ReDim estimators(1 To 2) As Double
    drift = (b - v ^ 2 / 2) * t / (m - 1)
    vsqrdt = v * Sqr(t / (m - 1))
    discdt = Exp(-r * t / (m - 1))
    For estimator = 1 To 2
        estimatorsum = 0
        For simulation = 1 To n
            ss(1, 1) = s
            w(1) = 1
            For j = 2 To m
                ss(1, j) = ss(1, j - 1) * Exp(drift + vsqrdt * Application.NormSInv(Rnd))
                w(j) = 1
            Next j
            j = m
            Do While j > 0
                If j = m Then
                    ss(w(j), j) = Application.Max((ss(w(j), j) - k), 0)
                    If w(j) < branches Then
                        ss(w(j) + 1, j) = ss(w(j - 1), j - 1) * Exp(drift + vsqrdt * Application.NormSInv(Rnd))
                        w(j) = w(j) + 1
                    ElseIf w(j) = branches Then
                        w(j) = 0
                        j = j - 1
                    End If
                ElseIf j < m Then
                    If estimator = 1 Then
                        sum1 = 0
                        For i1 = 1 To branches
                            sum1 = sum1 + discdt * ss(i1, j + 1)
                        Next i1
                        ss(w(j), j) = Application.Max(Application.Max((ss(w(j), j) - k), 0), sum1 / branches)
                    ElseIf estimator = 2 Then
                        sum1 = 0
                        For i1 = 1 To branches
                            sum2 = 0
                            For i2 = 1 To branches
                                If i2 <> i1 Then sum2 = sum2 + discdt * ss(i2, j + 1)
                            Next i2
                            If Application.Max((ss(w(j), j) - k), 0) >= sum2 / (branches - 1) Then
                                sum1 = sum1 + Application.Max((ss(w(j), j) - k), 0)
                            Else
                                sum1 = sum1 + discdt * ss(i1, j + 1)
                            End If
                        Next i1
                        ss(w(j), j) = sum1 / branches
                    End If
```

オプション価格モデル

- モンテカルロシミュレーションによるアメリカンオプション

```
                    If w(j) < branches Then
                        If j > 1 Then
                            ss(w(j) + 1, j) = ss(w(j - 1), j - 1) * Exp(drift + vsqrdt * Application.NormSInv(Rnd))
                            w(j) = w(j) + 1
                            For i = j + 1 To m
                                ss(1, i) = ss(w(j), j) * Exp(drift + vsqrdt * Application.NormSInv(Rnd))
                                w(i) = 1
                            Next i
                            j = m
                        Else
                            j = 0
                        End If
                    ElseIf w(j) = branches Then
                        w(j) = 0
                        j = j - 1
                    End If
                End If
            Loop
            estimatorsum = estimatorsum + ss(1, 1)
        Next simulation
        estimators(estimator) = estimatorsum / n
    Next estimator
    BGmcamrcall = 0.5 * Application.Max(Application.Max((s - k), 0), estimators(2)) + 0.5 * estimators(1)
End Function
```

options_vba2

	A	B
1	s	100
2	k	100
3	t	1
4	r	0
5	b	0
6	v	0.1
7	m	2
8	branches	20
9	n	1000
10	BSCALL	3.976732

BGmcamrcall

オプション価格モデル

- モンテカルロによるコーラブルオプション
  - モンテカルロシミュレーションの最大の特徴は経路依存型のオプションの価値を評価できることである。
  - 経路依存型のオプションの１つがコーラブルオプションである。
  - コーラブルコールオプションの保有者は原資産がある一定のレベルを超えるとオプションの行使を強要される。
  - コーラブルプットの保有者は原資産がある一定のレベルを下回るとオプションの行使を強要される。
- コーラブルボンド、プッタブルボンドとしての実用が可能

オプション価格モデル

- モンテカルロによるコーラブルオプション

```
Function callablecall(s, k, h, t, r, b, v, m, mm, n) As Double
    Dim i, j, nn, counter As Long
    Dim dt, st, sum As Double
    Dim drift, vsqrt As Double
    Dim barrierhitprob As Double
    nn = m * t
    dt = t / nn
    drift = (b - v * v * 0.5) * dt
    vsqrt = v * Sqr(dt)
    sum = 0
    For j = 1 To n
        barrierhitprob = 0
        st = s
        counter = 0
        For i = 2 To nn
            st = st * Exp(drift + vsqrt * Application.NormSInv(Rnd))
            If st > h Then
                counter = counter + 1
            Else
                counter = 0
            End If
            If counter = mm Then
                sum = sum + Exp(-r * (i * dt)) * Application.Max(st - k, 0)
                barrierhitprob = 1
                Exit For
            End If
        Next i
        sum = sum + Exp(-r * t) * (1 - barrierhitprob) * Application.Max(st - k, 0)
    Next j
    callablecall = sum / n
End Function
```

options_vba2

	A	B	C	D	E
1	s	100	100	100	100
2	k	100	100	100	100
3	h	105	105	105	105
4	t	1	1	1	1
5	r	0	0	0	0
6	b	0	0	0	0
7	v	0.1	0.2	0.1	0.2
8	m	250	250	250	250
9	mm	1	1	10	10
10	n	1000	1000	1000	1000
11	callablecall	3.2878679	4.4371844	3.653361	5.992194

callablecall

オプション価格モデル

- エクゼクティブストックオプション
- Jennergren and Naslund(1993)'A Comment on Valuation of Executive Stock Options and the FASB Proposal,' The Accounting Review

$$c = e^{-\lambda T}\left[Se^{-(b-r)T}N(d_1) - Xe^{-rT}N(d_2)\right]$$
$$p = e^{-\lambda T}\left[Xe^{-rT}N(-d_2) - Se^{-(b-r)T}N(-d_1)\right]$$

$$d_1 = \frac{\log(S/X)+\left(b+\sigma^2/2\right)T}{\sigma\sqrt{T}},\ d_2 = d_1\text{-}\sigma\sqrt{T}$$

オプション価格モデル

- エクゼクティブストックオプション
- Carpenter(1998)'The exercise and valuation of executive stock options,' Journal of Financial Economics
- 二項価格評価モデル
- $u = e^{(\hat{\mu}-\delta)h - \log(\cosh \sigma\sqrt{h}) + \sigma\sqrt{h}}$
- $d = e^{(\hat{\mu}-\delta)h - \log(\cosh \sigma\sqrt{h}) - \sigma\sqrt{h}}$
- $\hat{\mu} = \log(1+\mu)\,, n = the\ number\ of\ periods\ per\ year, h = \frac{1}{n}$
- $\tilde{p} = (e^{(r-\delta)h} - d)/(u-d)$