Sprawozdanie 1

Paweł Maciocha

1 Wprowadzenie

W niniejszym sprawozdaniu zajmiemy się wyceną opcji europejskich *call* oraz *put*. W tym celu posłużymy się modelami dyskretnymi, w których losowość wyrażana jest poprzez wystąpienie skończonej liczby możliwych scenariuszy.

Zakłada się przy tym, że rynek jest wielookresowy tzn. cena akcji zmienia się w chwilach $t = 1, 2, \ldots, T$ gdzie T to czas wygaśnięcia instrumentu. Najprostszym modelem wyceny opcji jest model dwumianowy, w którym to w każdym okresie może wystąpić jedno z dwóch zdarzeń - przyrost lub spadek ceny akcji. Dokładniej, jeśli w momencie t cena akcji wynosi S_t, to w momencie $t+1$ może wzrosnąć z prawdopodobieństwem p do $S_t u$, gdzie $u > 1$, lub spaść do $S_t d$, gdzie $0 < d < 1$.

Powyższy model można przedstawić graficznie za pomocą drzewa. Więcej o przedstawieniu graficznym przeczytać można w (...) strona 1. Podstawowym zagadnieniem związanym z modelem dwumianowym jest dobranie takich parametrów u, d oraz p, aby otrzymać jak najlepsze przybliżenie możliwych ruchów cen akcji.

2 Kalibracja modelu

W celu wyznaczenia parametrów u, d oraz p przyrównuje się średnią i wariancję ceny z końca okresu modelu dwumianowego do średniej i wariancji w modelu Blacka-Scholesa. Parametry spełniają wtedy następujące równania:

$$pu + (1-p)d = e^{(r-q)\Delta t} \tag{1}$$

$$pu^2 + (1-p)d^2 = e^{2(r-q+\sigma^2/2)\Delta t}. \tag{2}$$

Z powyższego otrzymujemy,

$$e^{(r-q)\Delta t}(u+d) - du = e^{2(r-q+\sigma^2/2\Delta t)}, \tag{3}$$

gdzie Δt jest długością okresu.

Aby jednoznacznie wyznaczyć u, d i p potrzebne jest dodatkowe równanie. Można je określić korzystając z modelu Coxa-Rossa-Rubinsteina lub Jarrowa-Ruda. Pierwszy z nich zakłada, że $ud = 1$, a drugi ustala wartość $p = \frac{1}{2}$.

3 Symulacje

W tej sekcji spróbujemy porównać oba modele w oparciu o symulacje, które przeprowadzimy w języku *Python*. Kod programu prezentuje się w sposób następujący

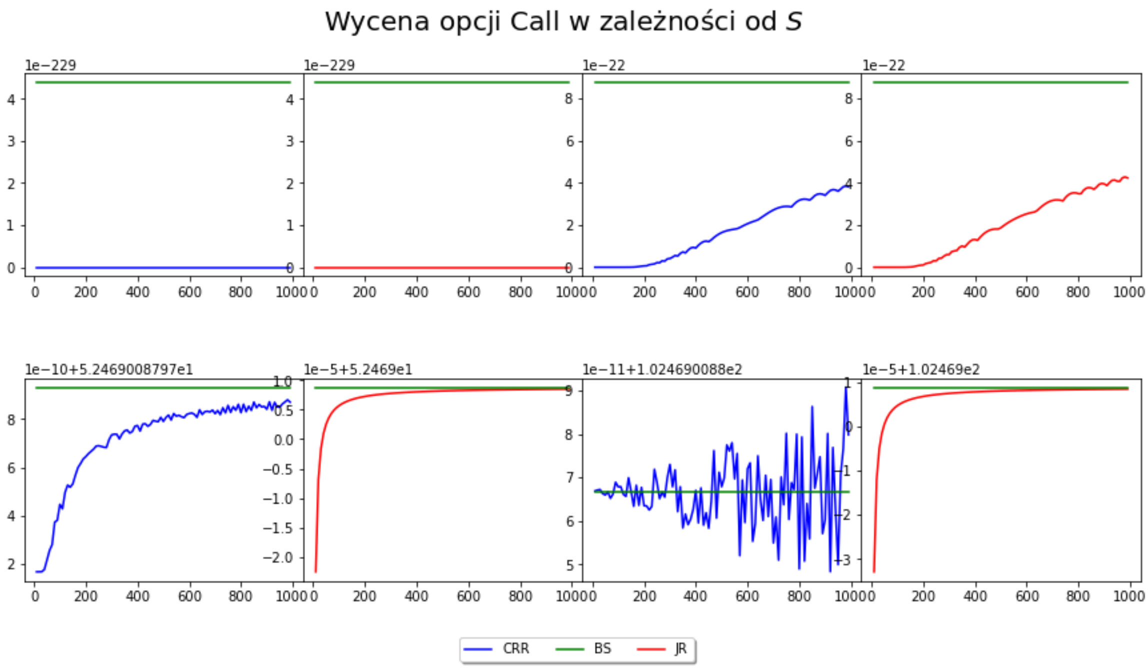
```
def Binomial_CRR(S, K, T, r, sigma, PutCall, n):
        dt = T/n
        u = np.exp(sigma*np.sqrt(dt))
        d = 1./u
        p = (np.exp(r*dt)-d) / (u-d)
        return Binomial_Body_Function(S, K, dt, r, PutCall, n, u, d, p)
```

```python
def Binomial_JR(S, K, T, r, sigma, PutCall, n):
        dt = T/n
        u = np.exp((r - sigma**2/2)*dt + sigma*np.sqrt(dt))
        d = np.exp((r - sigma**2/2)*dt - sigma*np.sqrt(dt))
        p = 0.5
        return Binomial_Body_Function(S, K, dt, r, PutCall, n, u, d, p)

def Binomial_Body_Function(S, K, dt, r, PutCall, n, u, d, p):
        stockvalue = np.zeros((n+1,n+1))
        stockvalue[0,0] = S
        for i in range(1,n+1):
                stockvalue[i,0] = stockvalue[i-1,0]*u
                for j in range(1,i+1):
                        stockvalue[i,j] = stockvalue[i-1,j-1]*d

        optionvalue = np.zeros((n+1,n+1))
        for j in range(n+1):
                if PutCall=="Call":
                        optionvalue[n,j] = max(0, stockvalue[n,j]-K)
                elif PutCall=="Put":
                        optionvalue[n,j] = max(0, K-stockvalue[n,j])

        for i in range(n-1,-1,-1):
                for j in range(i+1):
                        if PutCall=="Put":
                                optionvalue[i,j] = max(0, np.exp(-r*dt)*
                                (p*optionvalue[i+1,j] + (1-p)*optionvalue[i+1,j+1]))
                        elif PutCall=="Call":
                                optionvalue[i,j] = max(0, np.exp(-r*dt)*
                                (p*optionvalue[i+1,j] +(1-p)*optionvalue[i+1,j+1]))
        return optionvalue[0,0]
```

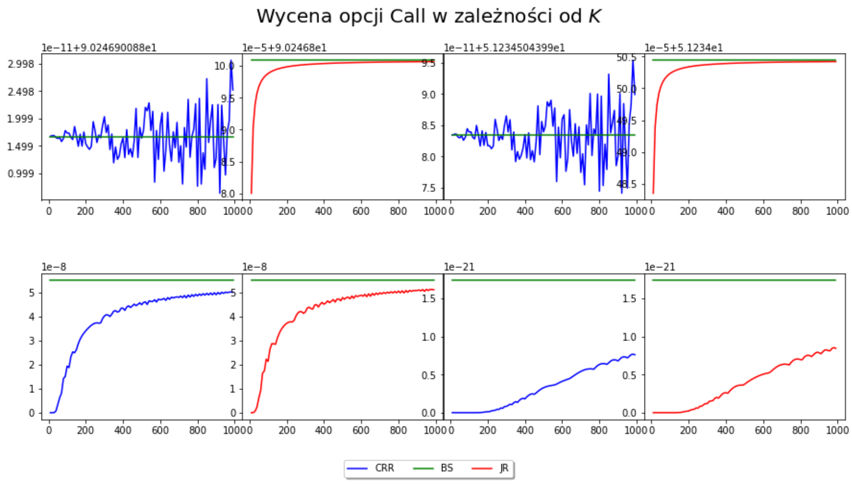
Rozważamy $t = 5, 10, \dots, 1000$. Z racji iż przestrzeń parametrów jest nieskończona musimy się na jakieś zdecydować. Zatem niech

$$S = 100, \quad K = 100, \quad T = \frac{1}{2}, \quad r = 0.05, \quad \sigma = 0.1. \tag{4}$$

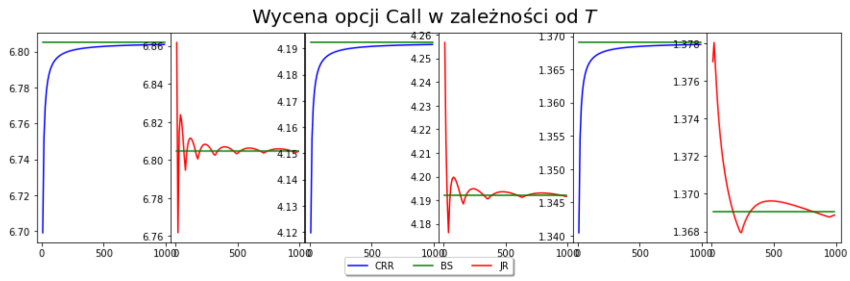
Będziemy badać szybkość zbieżności cen modeli do ceny Blacka-Scholesa (BS). Gdy $S \in \{10, 50, 150, 200\}$.



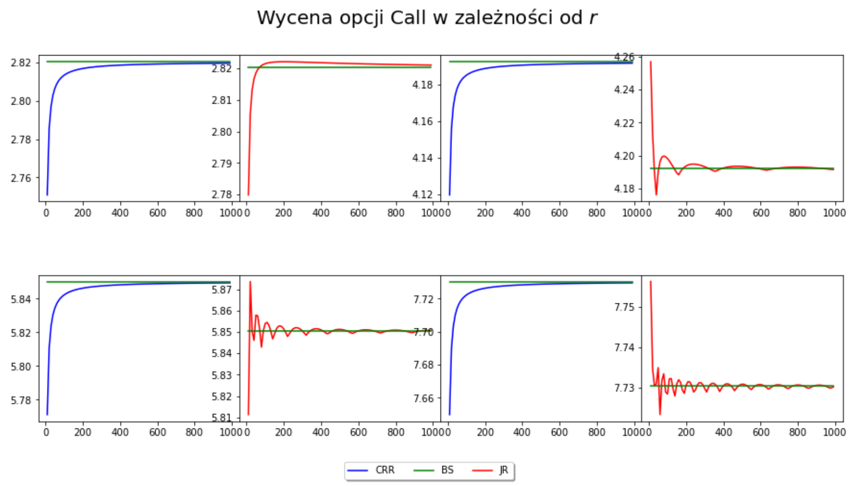
Z powyższego rysunku zauważyć możemy, że model JR cechuje z jednej strony większą stabilnością z drugiej natomiast obarczony jest większym błędem szczególnie dla małych n. Na kolejnych rysunkach rozważamy $K \in \{10, 50, 150, 200\}$, $T \in \{1, 0.5, 1/12\}$, $r \in \{0, 0.05, 0.1, 0.15\}$ oraz $\sigma \in \{0.01, 0.1, 1, 10\}$.



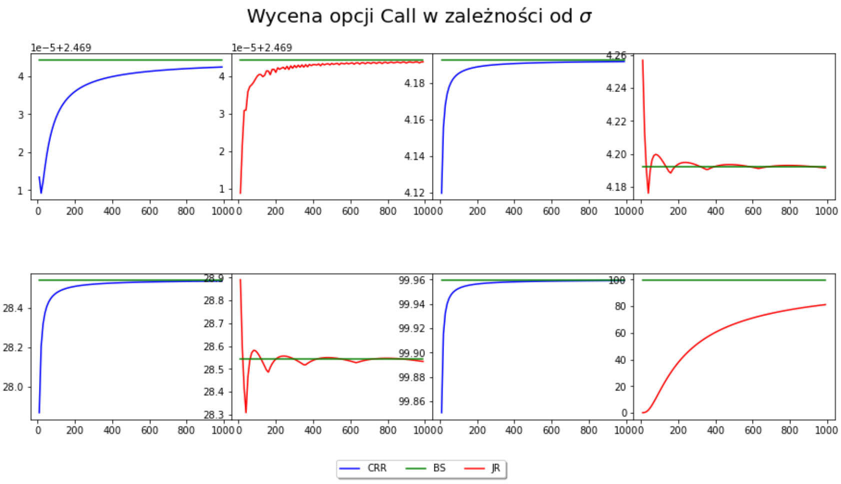
W przypadku gdy zmienną jest K sytuacja jest analogiczna do tej gdy zmienną jest S.



Zauważmy, że wraz ze wzrostem argumentu T model JR się stabilizuje, jest on dokładniejszy od modelu CRR, natomiast jest mniej stabilny. Być może, że wraz ze wzrostem parametru T (lub n) się stabilizuje się całkowicie.



Widzimy, że wraz ze wzrostem spoty procentowej model JR traci na stabilności ale jednocześnie jego dokładność względem CRR się zwiększa.



Zauważyć możemy, że w przypadku gdy σ przyjmuje wartości bardzo małe to model JR jest bezdyskusyjnie lepszy pomimo mniejszej stabilności. Jednakże gdy σ przyjmuje wartości coraz to większe model JR staje się coraz gorszy.

Nie będziemy porównywać modeli dla opcji *put*. Wnioski nie będą się wiele różnić z faktu iż algorytmy wyznaczający ceny instrumentów różnią się tylko i wyłącznie ceną wypłaty, a zatem nie może to znacznie wpłynąć na stabilność czy zbieżność do ceny BS omawianych modeli. Z wykresami dla opcji *put* można się zapoznać na końcu artykułu.

4 Wpływ parametrów u i d na wycenę

Poprzednio przyjmowaliśmy,

$$u = \exp(\sigma\sqrt{dt}), \quad d = \frac{1}{u}, \quad p = \frac{\exp(rdt) - d}{u - d}, \tag{5}$$

$$u = \exp\{(r - \frac{\sigma^2}{2})dt + \sigma\sqrt{dt}\}, \quad d = \exp\{(r - \frac{\sigma^2}{2})dt - \sigma\sqrt{dt}\}, \quad p = \frac{1}{2} \tag{6}$$

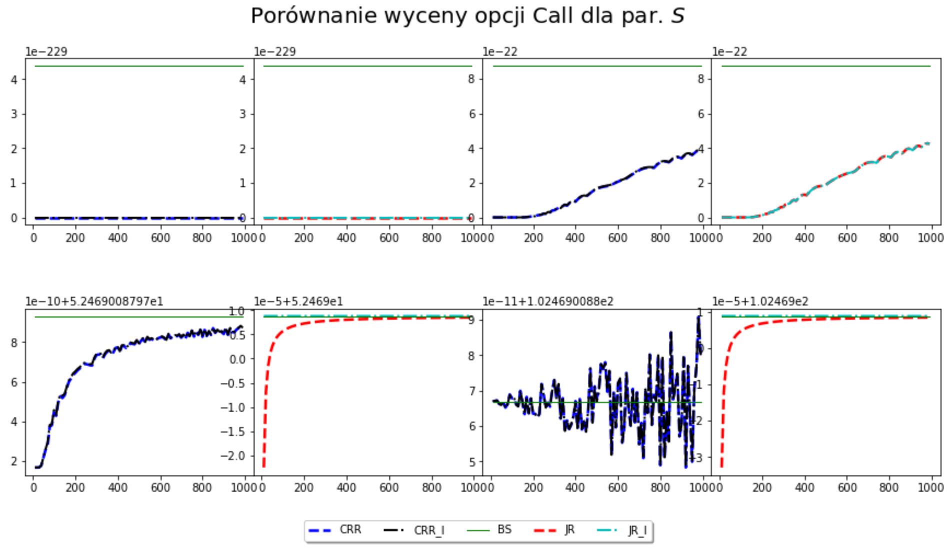
odpowiednio dla CRR oraz JR. Niech teraz

$$u_+/d_- = \frac{1}{2}\Big((e^{rdt+\sigma^2 dt} + e^{-rdt}) \pm \sqrt{(e^{rdt+\sigma^2 dt} + e^{-rdt})^2 - 4}\Big), \tag{7}$$

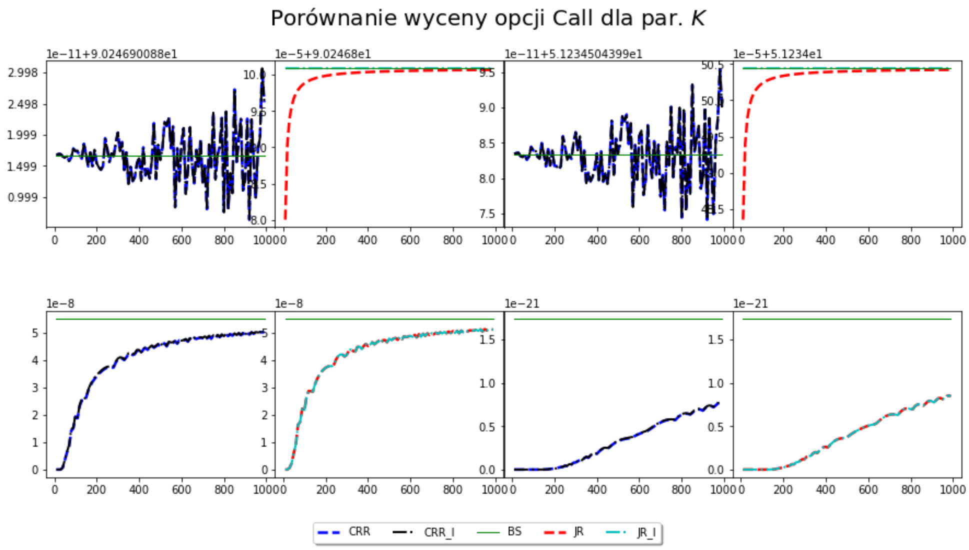
$$u_+/d_- = e^{rdt}(1 \pm \sqrt{e^{\sigma^2 dt} - 1}), \tag{8}$$

dla CRR oraz JR, p dane jest przez takie same wzory jak poprzednio (z uwzględnieniem nowych wartości u i d).

Ponownie przeprowadzając symulacje porównamy modele ze starym zestawem zmiennych oraz te z nowym. Wartości parametrów S, K, T, r, σ przyjmujemy takie jak poprzednio.

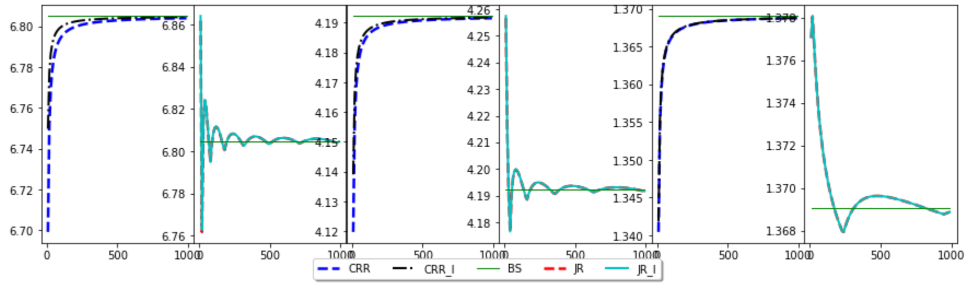


Poprawę obserwujemy przy $S = 150, 200$ dla modelu JR widzimy, że wartość jest bardzo dokładne dla małych n.



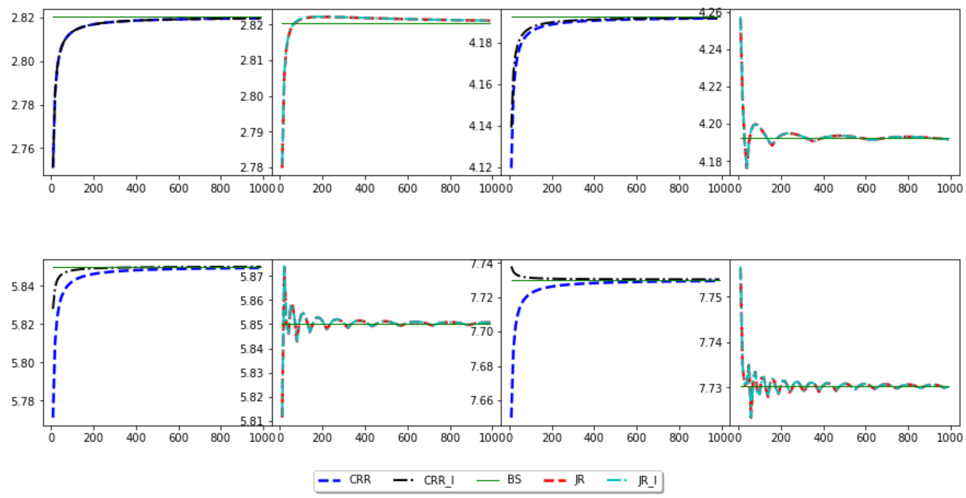
Zauważmy zależność, model JR z nowymi parametrami poprzednio poprawiał dokładność gdy początkowa cena akcji była duża, teraz dokładność jest lepsza gdy cena wykonania jest względnie mała.

Porównanie wyceny opcji Call dla par. T



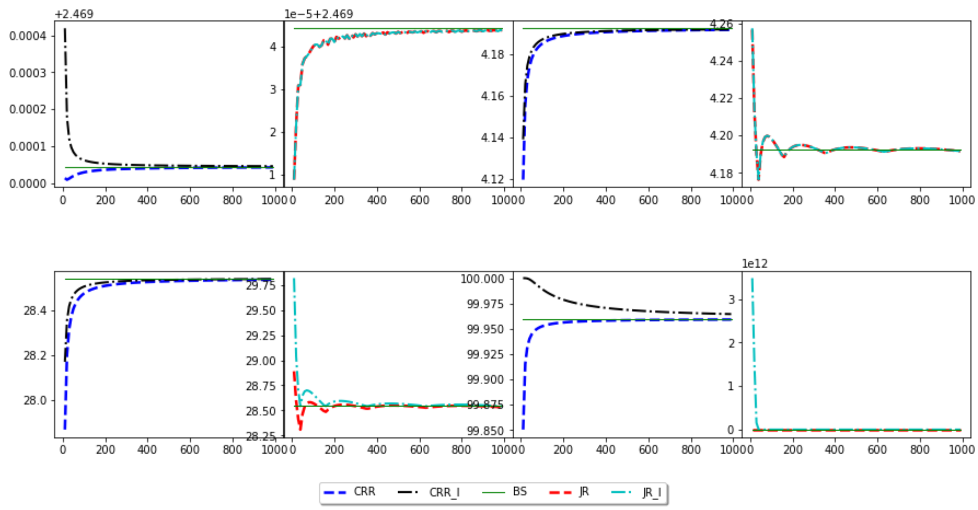
Widoczną poprawę odnotować możemy dla modelu CRR gdy $T = 1$ oraz $1/2$. Być może im krótszy okres inwestycji tym lepiej sprawuje się model CRR z nowymi parametrami.

Porównanie wyceny opcji Call dla par. r



Ponownie lepsze wyniki uzyskujemy dla CRR w przypadku modelu JR nie widać poprawy. Odnotujmy, że przy zerowej stopie procentowej poprawy nie ma.

Wycena opcji Call w zależności od σ



Dla parametru σ wyniki nie są aż tak jednoznaczne.