

The_Binomial_Model_and_The_Greeks

March 22, 2020

1 Radosław Matulewicz 404685

2 The Binomial Model and the Greeks

Naszym celem jest dobranie takiej liczby węzłów M , aby uzyskać wymaganą dokładność. Przeanalizujemy więc zachowanie się ceny europejskiej opcji put, oraz jej greckich parametrów, przy zwiększającej się liczbie węzłów. Jakby się mogło wydawać cena opcji powinna być monotoniczna przy wzroście M , jednak w tym przypadku tak się nie dzieje. Ku zaskoczeniu, cena opcji waha się (patrz Rys. 1). Podobna sytuacja jest licząc jej Greckie parametry (patrz Rys. 2,3,4). Widać więc, że liczenie dla pojedynczej wartości M mija się z celem, ponieważ nie wiadomo, w której części krzywej się znajdziemy. Niemniej jednak, z rysunków można wywnioskować, że amplituda wartości zmniejsza się wraz z liczbą węzłów M . Dokonując zatem obliczeń dla dużej liczby wartości M jesteśmy w stanie znaleźć górną i dolną granicę tej oscylacji. Powiększone fragmenty odpowiednich wartości pozwalają nam bardziej zrozumieć tą sytuację (patrz Rys. 1a, 2a, 3a, 4a). Możemy więc dobrać taką liczbę węzłów M , dla której wahania te będą mniejsze niż dokładność, z którą chcemy tą cenę obliczyć. Poniżej przedstawione zostało krok po kroku dojście do powyższych wniosków.

```
[1]: import numpy as np
import matplotlib.pyplot as plt
```

```
[2]: def CRR_put(S0, r, sigma, T, K, M):
    dt = T/M
    U = np.exp(sigma*np.sqrt(dt))
    D = 1/U
    p = (np.exp(r*dt)-D)/(U-D)
    DF = np.exp(-r*dt)

    S = np.zeros([M+1, M+1])

    for i in range(M+1):
        for j in range(i+1):
            S[j,i] = S0*(U**(i-j))*(D**j)

    Ve = np.zeros([M+1, M+1])
    Va = np.zeros([M+1, M+1])
    Ve[:,M] = np.maximum(np.zeros(M+1), K - S[:, M])
    Va[:,M] = np.maximum(np.zeros(M+1), K - S[:, M])
```

```

for i in np.arange(M-1, -1, -1):
    for j in np.arange(0, i+1):
        Ve[j, i] = DF * (p*Ve[j, i+1] + (1-p)*Ve[j+1, i+1])
        Va[j, i] = np.maximum(np.maximum(0, K - S[j, i]), DF * (p*Va[j, i+1]
↪+ (1-p)*Va[j+1, i+1]))

# Greckie parametry
# Delta:
delta_E = (Ve[0,1] - Ve[1,1]) / ((S0 * U) - (S0 * D))
delta_A = (Va[0,1] - Va[1,1]) / ((S0 * U) - (S0 * D))

# Gamma:
h = 0.5 * (S[0,2] - S[2,2])
delta1_E = (Ve[1,2] - Ve[2,2]) / (S0 - S[2,2])
delta2_E = (Ve[0,2] - Ve[1,2]) / (S[0,2] - S0)
gamma_E = (delta2_E - delta1_E) / h
delta1_A = (Va[1,2] - Va[2,2]) / (S0 - S[2,2])
delta2_A = (Va[0,2] - Va[1,2]) / (S[0,2] - S0)
gamma_A = (delta2_A - delta1_A) / h

# Theta
theta_E = (Ve[1,2] - Ve[0,0]) / (2 * dt)
theta_A = (Va[1,2] - Va[0,0]) / (2 * dt)

return Ve, Va, delta_E, delta_A, gamma_E, gamma_A, theta_E, theta_A

```

```

[3]: cena_europejska = np.array([])
delta_europejska = np.array([])
gamma_europejska = np.array([])
theta_europejska = np.array([])

podzial = range(5,1000,5)

for i in podzial:
    [Ve, Va, delta_E, delta_A, gamma_E, gamma_A, theta_E, theta_A] =
↪CRR_put(100,0.03,0.2,1.1,95,i)
    cena_europejska = np.append(cena_europejska, Ve[0,0])
    delta_europejska = np.append(delta_europejska, delta_E)
    gamma_europejska = np.append(gamma_europejska, gamma_E)
    theta_europejska = np.append(theta_europejska, theta_E)

```

```

[4]: import matplotlib.pyplot as plt

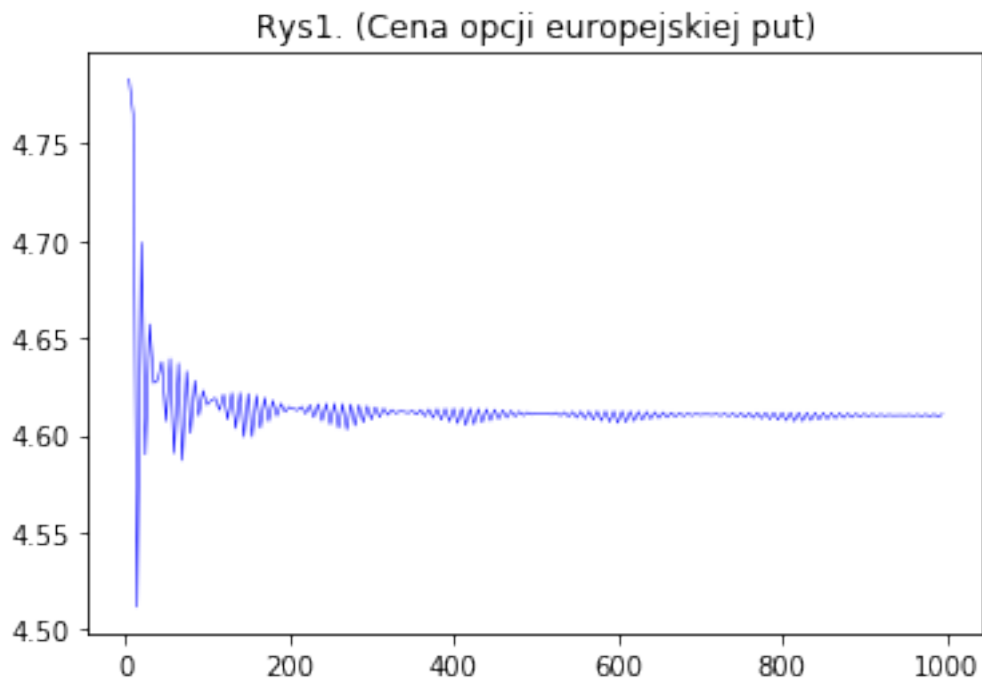
plt.plot(podzial, cena_europejska, "-b", linewidth = 0.5)
plt.title('Rys1. (Cena opcji europejskiej put)')
plt.show()

```

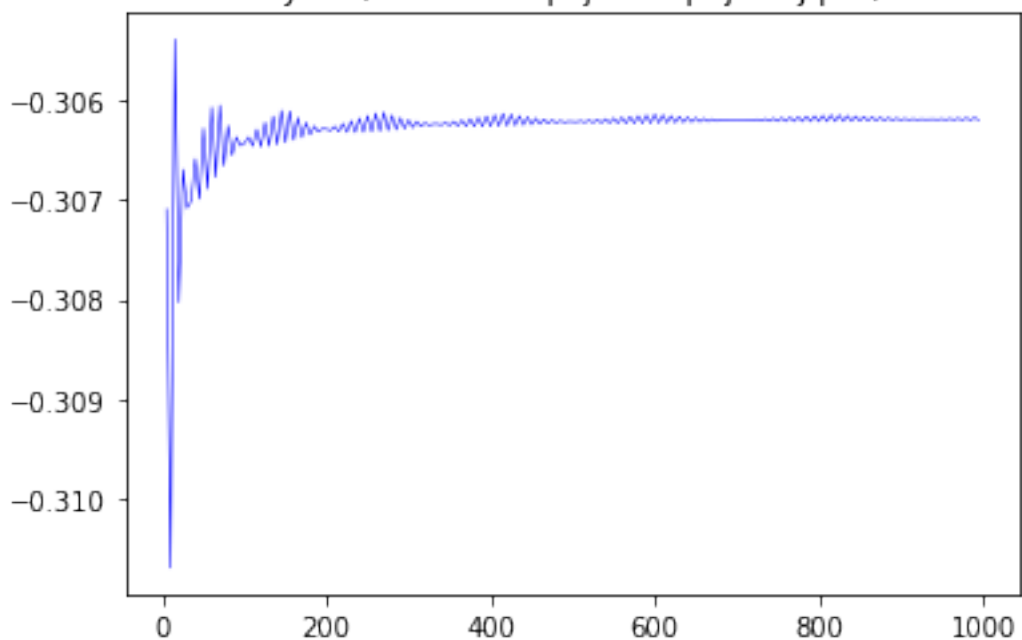
```

plt.plot(podzial, delta_europejska, "-b", linewidth = 0.5)
plt.title('Rys2. (Delta dla opcji europejskiej put)')
plt.show()
plt.plot(podzial, gamma_europejska, "-b", linewidth = 0.5)
plt.title('Rys3. (Gamma dla opcji europejskiej put)')
plt.show()
plt.plot(podzial, theta_europejska, "-b", linewidth = 0.5)
plt.title('Rys4. (Theta dla opcji europejskiej put)')
plt.show()

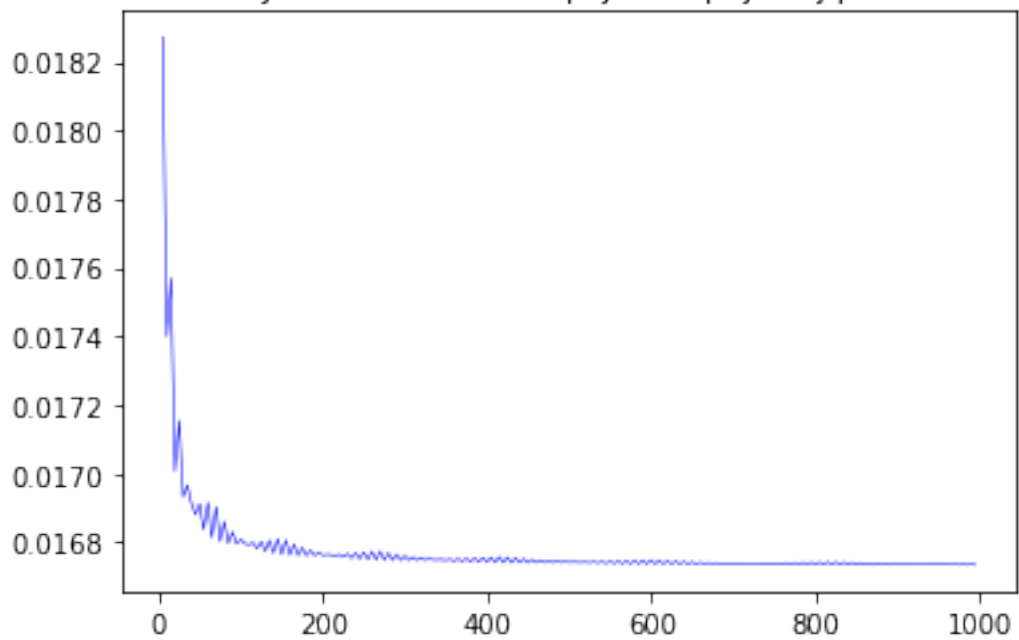
```

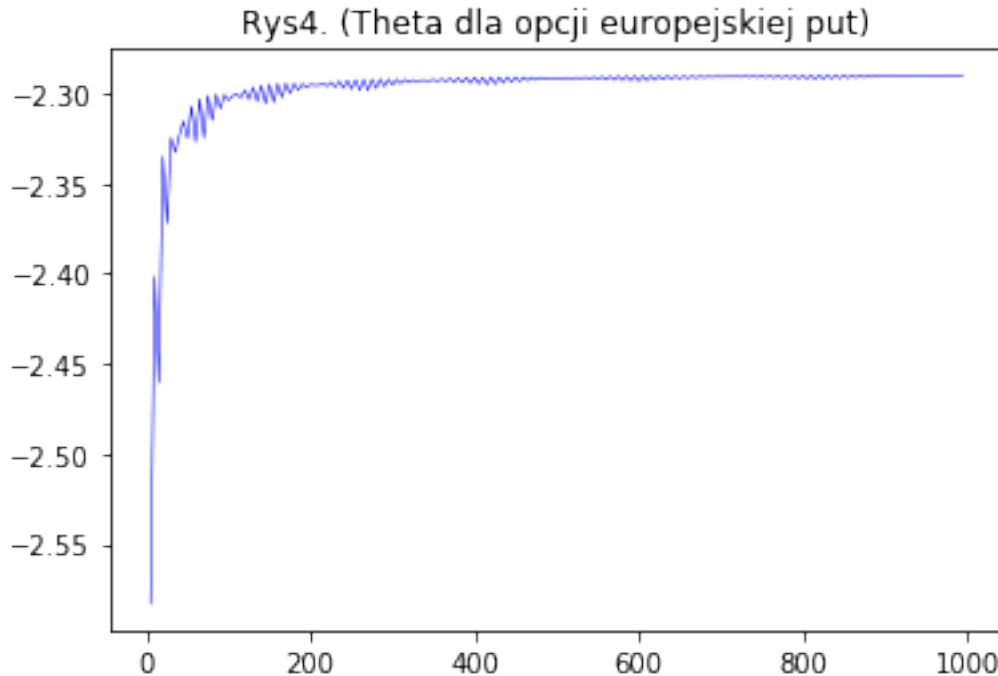


Rys2. (Delta dla opcji europejskiej put)



Rys3. (Gamma dla opcji europejskiej put)





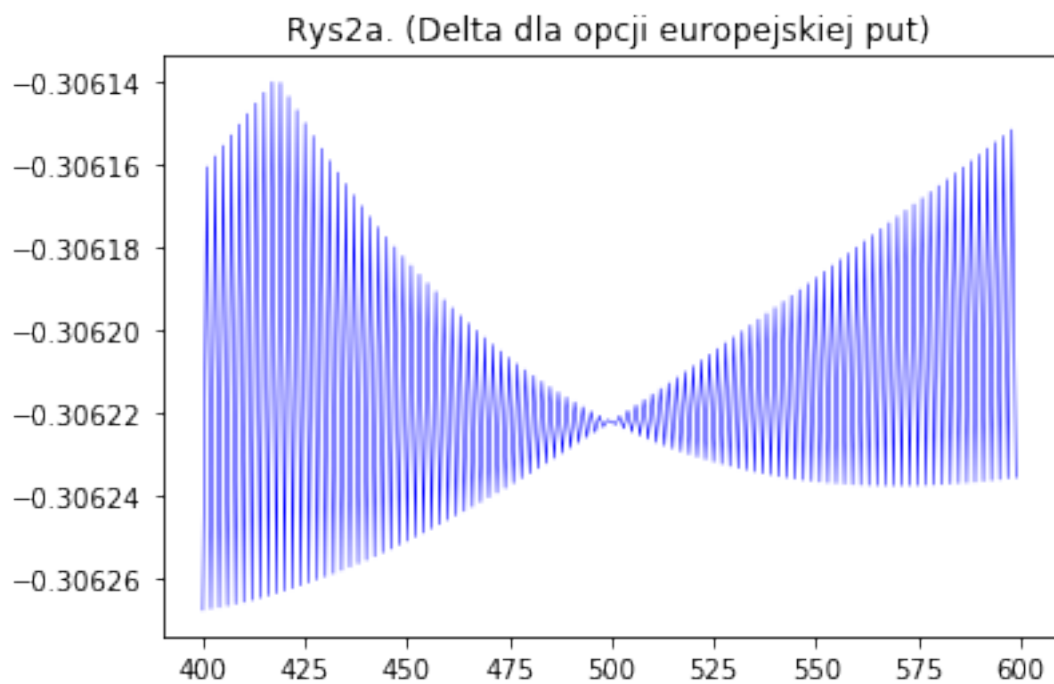
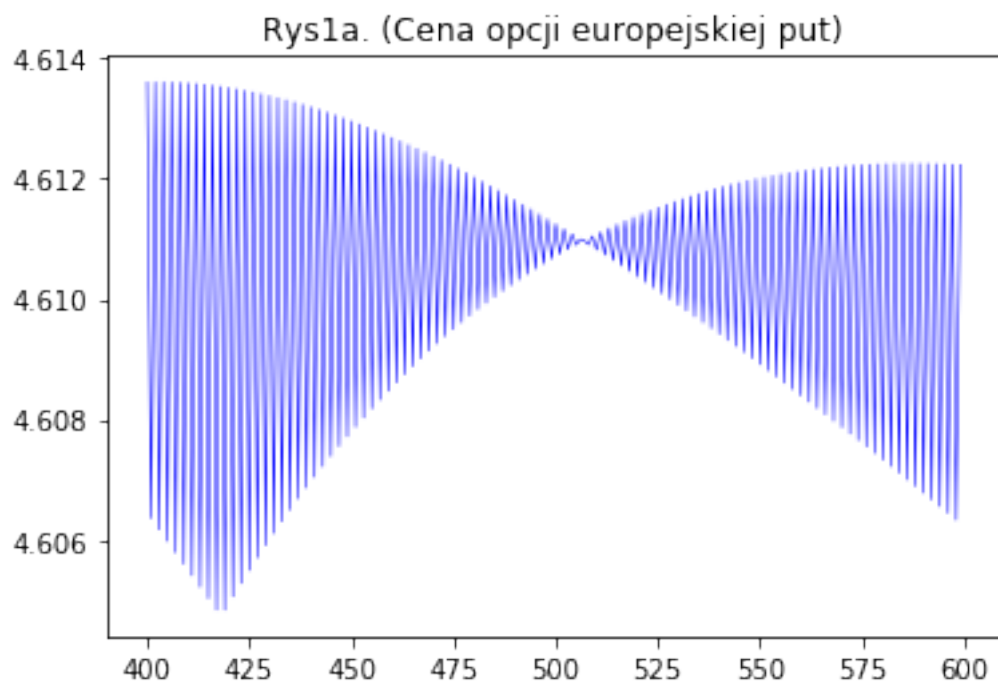
```
[5]: cena_europejska_dokladnie = np.array([])
delta_europejska_dokladnie = np.array([])
gamma_europejska_dokladnie = np.array([])
theta_europejska_dokladnie = np.array([])

powiekszenie = range(400,600,1)

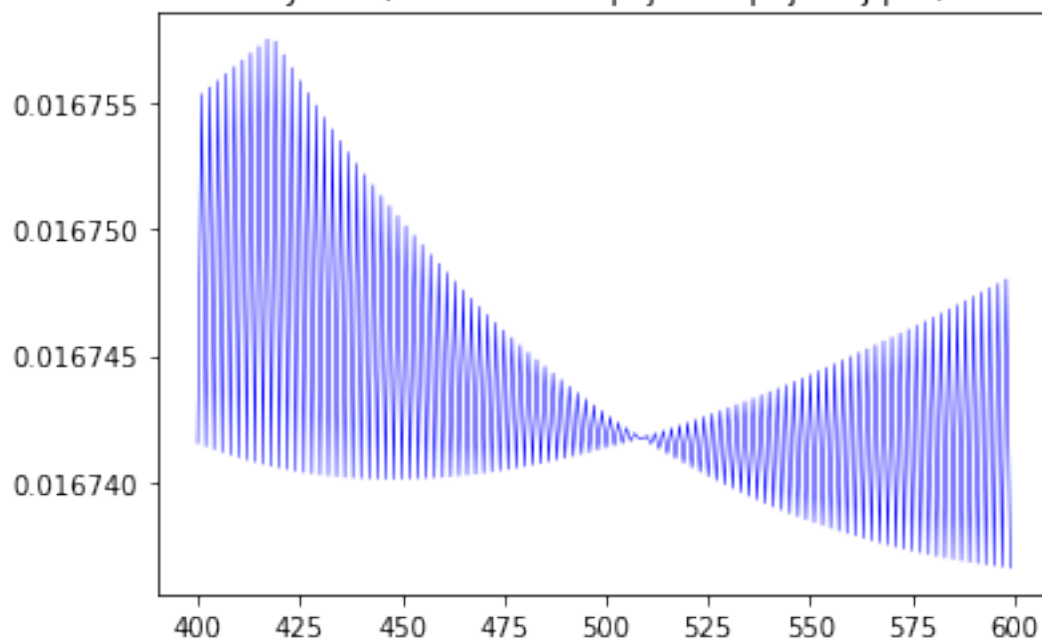
for i in powiekszenie:
    [Ve1, Va1, delta_E1, delta_A1, gamma_E1, gamma_A1, theta_E1, theta_A1] = u
    ↳CRR_put(100,0.03,0.2,1.1,95,i)
    cena_europejska_dokladnie = np.append(cena_europejska_dokladnie, Ve1[0,0])
    delta_europejska_dokladnie = np.append(delta_europejska_dokladnie, delta_E1)
    gamma_europejska_dokladnie = np.append(gamma_europejska_dokladnie, gamma_E1)
    theta_europejska_dokladnie = np.append(theta_europejska_dokladnie, theta_E1)
```

```
[6]: plt.plot(powiekszenie, cena_europejska_dokladnie, "-b", linewidth = 0.5)
plt.title('Rys1a. (Cena opcji europejskiej put)')
plt.show()
plt.plot(powiekszenie, delta_europejska_dokladnie, "-b", linewidth = 0.5)
plt.title('Rys2a. (Delta dla opcji europejskiej put)')
plt.show()
plt.plot(powiekszenie, gamma_europejska_dokladnie, "-b", linewidth = 0.5)
plt.title('Rys3a. (Gamma dla opcji europejskiej put)')
plt.show()
```

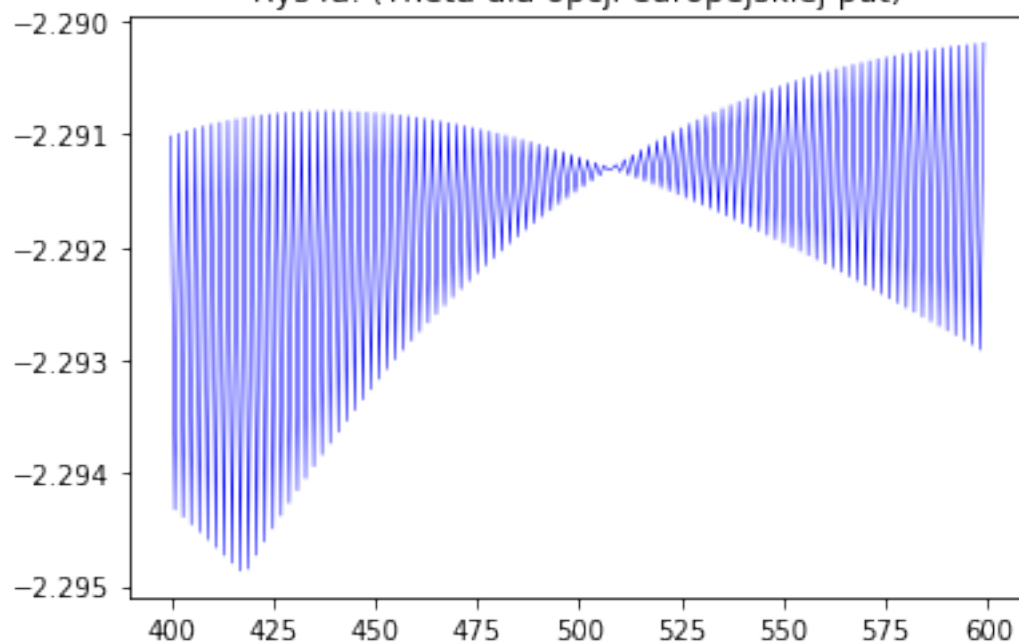
```
plt.plot(powiekszenie, theta_europejska_dokladnie, "-b", linewidth = 0.5)
plt.title('Rys4a. (Theta dla opcji europejskiej put)')
plt.show()
```



Rys3a. (Gamma dla opcji europejskiej put)



Rys4a. (Theta dla opcji europejskiej put)



[]: