Gil Sales M. Neto & João Victor de Fonseca

Projeto I
MAE001 - Modelagem Mat. em Finanças I
O Modelo Binomial

Universidade Federal do Rio de Janeiro

Instituto de Matemática

Bacharelado em Matemática Aplicada

Prof.: Marco Cabral

Brasil

Maio, 2019

Sumário

1 Os Algoritmos

Esta seção tem como objetivo apresentar todos os códigos utilizados nas simulações.

1.1 Simulação do modelo binomial

Utilizamos a linguagem Python3 para a implementação do algoritmo que simula os caminhos da ação

```
In [2]: ## Função para simular os valores da ação dados os parametros
        def binomial(S0, T, dt, u, d, p):
            Si = S0
            S = []
            t = np.arange(0,T,dt)
            for ti in t:
                rnd = np.random.rand()
                if rnd < p:
                    Si *= u
                else:
                    Si *= d
                S.append(Si)
            return S
```

Esse algoritmo tem depedência do pacote Numpy do Python, e os gráficos foram plotados com o pacote MatPlotLib.pyplot

1.2 Plotando o gráfico da questão 1

Foram plotados dois gráficos, os quais seguem os códigos

1.2.1 Plot usual

In [3]:
```
## Definição dos parametros
T = 20
dt = 0.5
S0 = 5
u = 120/110
d = 110/120
p = 0.5

## Construção da lista de valores da ação pelo tempo
x = [binomial(S0, T, dt, u, d, p) for i in range(0,20)]

## Inserir o valor inicial em cada lista
for i in range(0,20):
    x[i].insert(0,S0)

## Discretização do tempo
ts = np.linspace(0,20,41)

## Plot do gráfico
plt.figure(figsize=(20,10))
for i in range(0,20):
    plt.plot(ts,x[i],label="{0:.5f}".format(x[i][-1]))
plt.title('Simulação com 20 caminhos de valor da ação')
plt.legend(title='Valor final', loc='upper center',
↪ fancybox=True, shadow=True,
           ncol=7, bbox_to_anchor=(0.5,1.05))
plt.xlabel('Tempo')
plt.ylabel('Valor da Ação.')
plt.show()
```

1.2.2 Plot com escala Log em Y

In [4]:
```
plt.figure(figsize=(20,10))
for i in range(0,20):
    plt.semilogy(ts,x[i],label="{0:.5f}".format(x[i][-1]))
plt.legend(title='Valor final', loc='upper center',
↪  fancybox=True, shadow=True, ncol=7,
↪  bbox_to_anchor=(0.5,1.05))
plt.title('Caminhos de valor da ação')
plt.show()
```

1.3 Boxplots da questão 2

In [54]:
```
vs = [[],[]]
for j in [250,500,1000,2000]:
    vs[0].append([binomial(S0, T, dt, u, d, p)[-1] for i in range(0,j)])
plt.boxplot(vs[0])
plt.title('Boxplot com simulações')
plt.xlabel('Número de caminhos executados na simulação')
plt.xticks([1,2,3,4],[250,500,1000,2000])
plt.show()
```

1.4 Boxplots da questão 3

In [55]:
```
u_n = np.sqrt(u)
d_n = np.sqrt(d)
dt_n = dt/2
for j in [250,500,1000,2000]:
    vs[1].append([binomial(S0, T, dt_n, u_n, d_n, p)[-1] for i in range(0,j)])

plt.boxplot(vs[1])

plt.title('Boxplot com simulações')
plt.ylabel('Valores das ações ao tempo final')
plt.xlabel('Número de caminhos executados na simulação')
plt.xticks([1,2,3,4],[250,500,1000,2000])
```

1.5 Função para calculo da esperança

In [56]:
```
## Função para calcular a esperança
def esperanca(p_e,u_e,d_e,S0_e,n_e):
    return ((p_e*u_e+(1-p_e)*d_e)**n_e)*S0_e
```

1.6 Plot dos gráficos de erro

In [58]:
```
ts = np.linspace(250,2000,4)

ys = [[],[]]
for j in [0,1]:
    for i in vs[j]:
        ys[j].append(abs((np.mean(i)-e[0])))
plt.plot(ts,ys[0])
plt.plot(ts,ys[1])
plt.title('Gráfico do erro em relação ao número de simulações')
plt.xlabel('Número de simulações')
plt.ylabel('Erro relativo')
plt.show()
plt.semilogy(ts,ys[0])

plt.semilogy(ts,ys[1])
plt.title('Gráfico do erro em relação ao número de simulações escala log')
plt.xlabel('Número de simulações')
plt.ylabel('Erro relativo')
plt.show()
```

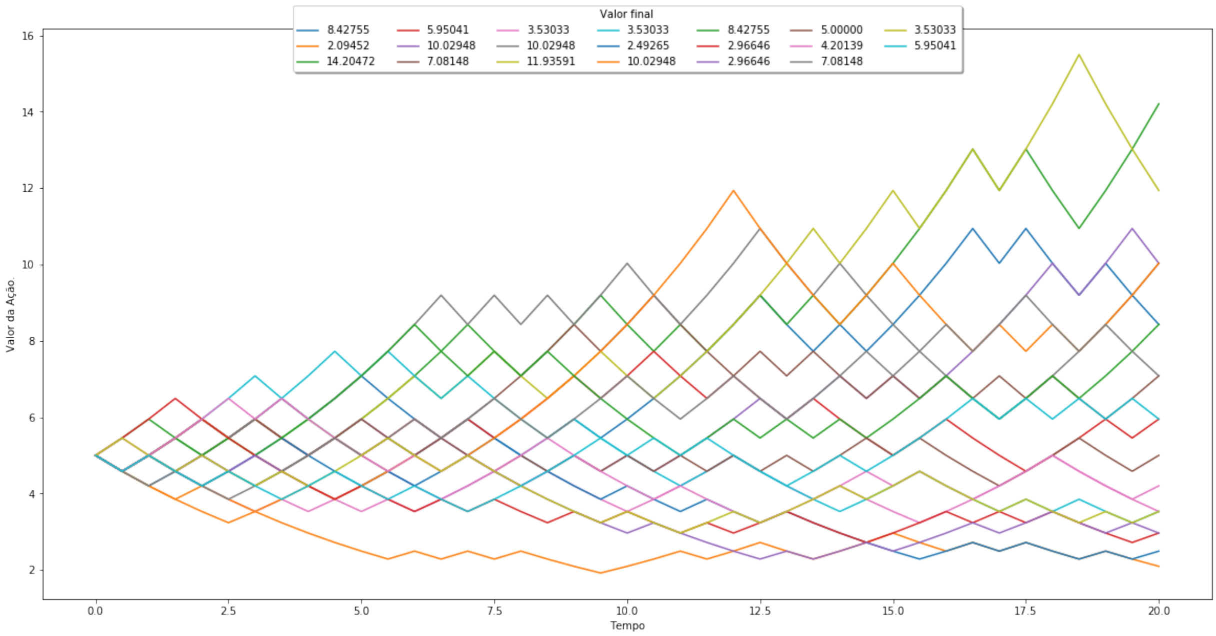
2 Questão 1

Foi pedido para fixar os parametros $T = 20$ e $\Delta t = 0.5$ e escolher os outros, nossos parâmetros então ficaram:

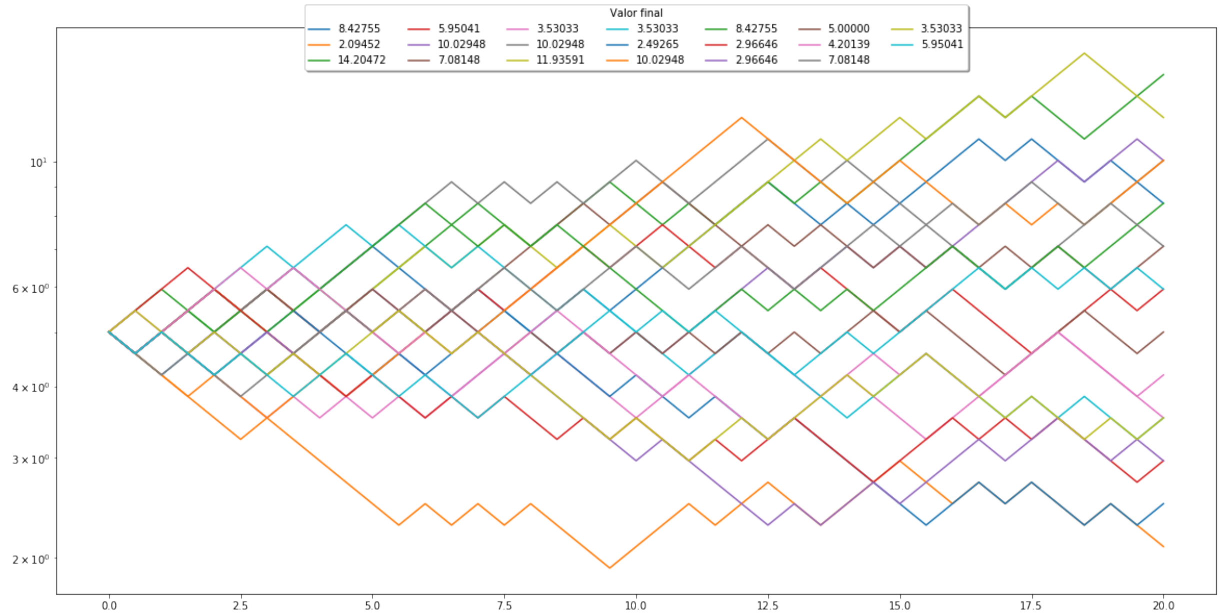
- $T = 20$
- $\Delta t = 0.5$
- $S_0 = 5$
- $u = \frac{120}{110}$
- $d = \frac{110}{120}$
- $p = 0.5$

Como estamos tratando de um modelo binomial que lida com exponenciais, faz-se necessário também visualizar os valores com escala log.

2.0.1 Plot com escala comum

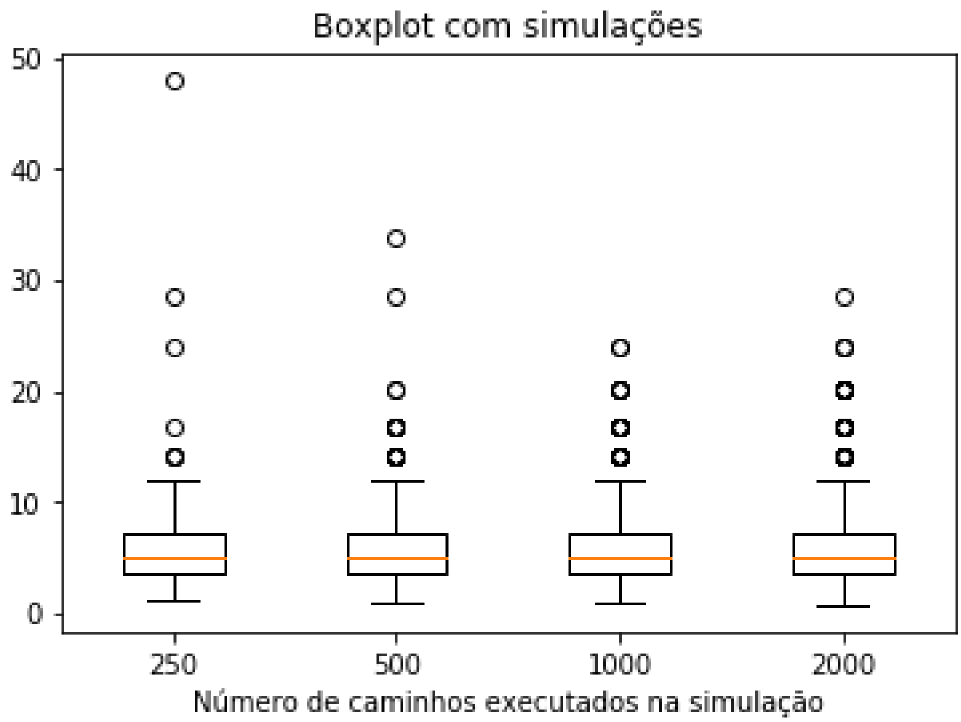


2.0.2 Plot com escala log



3 Questão 2

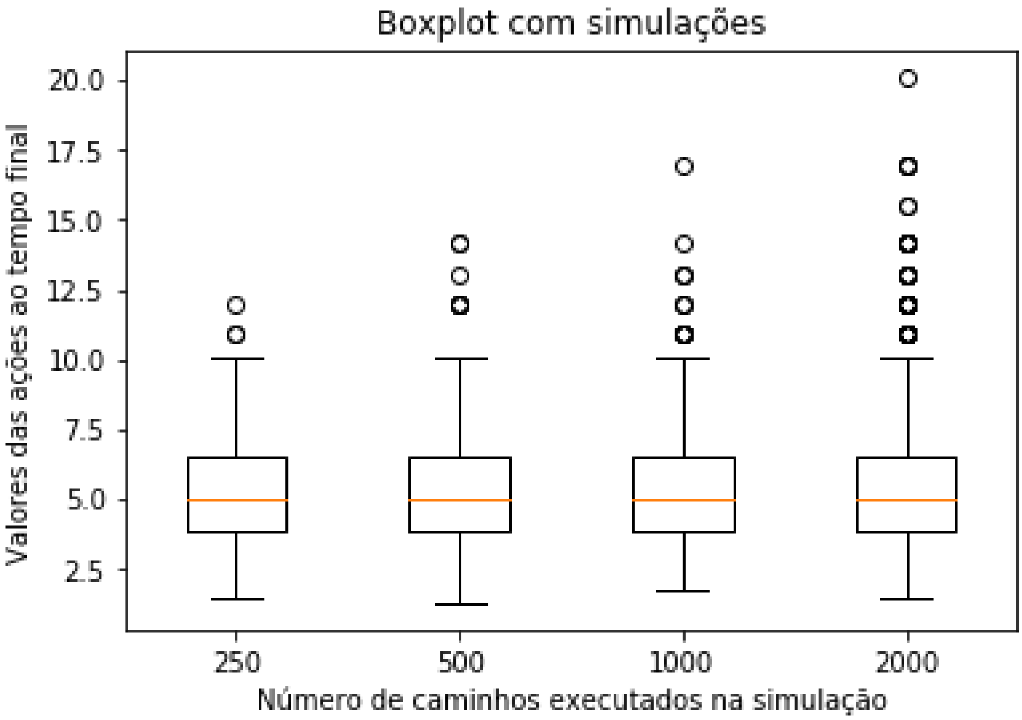
Foi pedido um boxplot para 250,500,1000,2000 caminhos de valor de ação



4 Questão 3

Foi pedido um mesmo boxplot, mas com novos parametros, que ficaram:

- $T = 20$
- $\Delta t = 0.5$
- $S_0 = 5$
- $u = \sqrt{\frac{120}{110}}$
- $d = \sqrt{\frac{110}{120}}$
- $p = 0.5$



Podemos ver que quando fazemos mais passos com up e down menores, os valores finais ficam bem mais 'comportados', por exemplo os outliers aqui chegaram a no máximo 20, enquanto no caso anterior chegavam a 50.

5 Questão 4

A média esperada da binomial pode ser dada pela seguinte fórmula:

$$E[X] = (p \cdot u + q \cdot d)^n \cdot S_0$$

onde $q = (1 - p), n = T$, e os outros são os parametros do algoritmo
A média do conjunto de dados é obtida pela média aritmetica, utilizando o método mean do numpy.

5.0.1 Valor esperado para o primeiro caso

$$\begin{aligned} E[X] &= \left(0.5 \cdot \frac{120}{110} + 0.5 \cdot \frac{110}{120}\right)^{20} \cdot 5 \\ &\approx (1.0038)^{20} \cdot 5 \\ &\approx (1.079) \cdot 5 \\ &\approx 5.393 \end{aligned}$$

5.0.2 Valor esperado para o segundo caso

$$\begin{aligned} E[X] &= \left(0.5 \cdot \sqrt{\frac{120}{110}} + 0.5 \cdot \sqrt{\frac{110}{120}}\right)^{20} \cdot 5 \\ &\approx (1.0095)^{20} \cdot 5 \\ &\approx (1.019) \cdot 5 \\ &\approx 5.095 \end{aligned}$$

5.0.3 Média dos valores para o primeiro caso

Simulações	Média
250	5.828
500	5.965
1000	5.643
2000	5.762

5.0.4 Média dos valores para o segundo caso

Simulações	Média
250	5.332
500	5.342
1000	5.326
2000	5.415

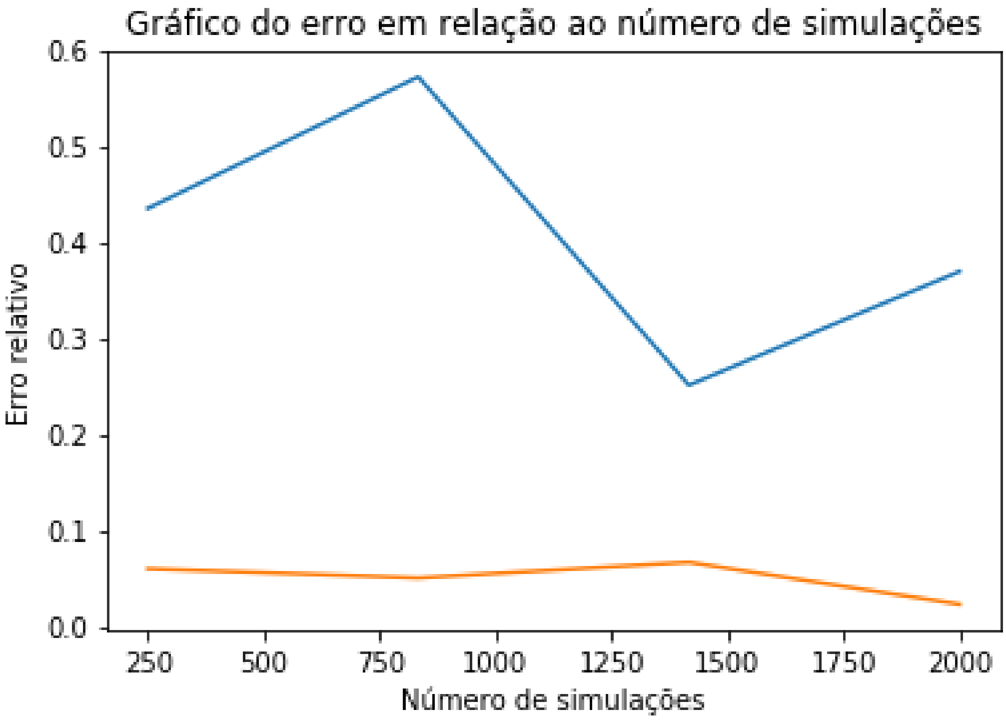
5.0.5 Conclusão

Podemos ver que as médias se aproximam das esperanças a medida que o número de simulações aumenta

6 Questão 5

6.0.1 Plot dos gráficos em escala usual

A curva azul representa o caso em que $u = \frac{120}{110}$ e a curva laranja o caso onde $u = \sqrt{\frac{120}{110}}$



6.0.2 Plot dos gráficos em escala log

A curva azul representa o caso em que $u = \frac{120}{110}$ e a curva laranja o caso onde $u = \sqrt{\frac{120}{110}}$