

---

```
% Take Home 5 - Mezzetti anita
% part e

clc
clear all
close all

% parameters:
sigma = 0.15;
lambda = 0.4;
alpha = -0.5;
beta = 0.4;
S0 = 1;

T = 0.5;
a_min = 0.7;
a_max = 1.3;

eta = -1;
L = 50;

a = linspace(a_min,a_max,100); % strike prices vector

% 1) Fourier transform formula
price_fourier = zeros(length(a),1);

tic
for j = 1:length(a)
    price_fourier(j) = MertonDigitalEurOptPricing ...
        (lambda, sigma, alpha, beta, a(j), S0, T, eta, L);
end
fourier_time = toc;

% 2) Chebychev interpolation
n = [2:30]; % interpolation order
%n = [30:5:150]; % interpolation order

cheb_time = zeros(length(n),1);
f = @(a) MertonDigitalEurOptPricing (lambda, sigma, alpha, beta, a,
    S0, T, eta, L);

price_cheb = zeros(length(a),length(n));
for j = 1:length(n)
    tic
    price_cheb(:,j) = ChebInterpol(f, a, n(j), a_min, a_max);
    cheb_time(j) = toc;
end

% plot the two prices:
figure
plot(a, price_fourier, 'r' , a, price_cheb(:,end) , 'b--')
title('European digital option pricing')
```

---

---

```

xlabel('Strike price')
ylabel('Option price')
legend('Fourier transform','Chebyshev interpolation N = 30')

% plot Cheb prices:
figure
plot(a, price_fourier, '*', 'DisplayName', 'Fourier');
hold on
for j = 1:length(n)
    hold on
    tt = sprintf('n = %f ', n(j));
    plot(a, price_cheb(:,j), 'DisplayName', tt);
end
title('Chebychev prices for different order of interpolation')
xlabel('Strike price')
ylabel('Option price')
legend show

% maximum absolute error for each n:
error = zeros(1, length(n));
for j = 1:length(n)
    error(j) = max(abs(price_fourier - price_cheb(:,j)));
end

% plot the error:
figure
semilogy(n, exp(-n), 'k--')
hold on
semilogy(n,error)
grid on
title('Maximal absolute error')
legend('O(exp(-N))', 'max abs err')
xlabel('Interpolation order n')
ylabel('Error')

% plot execution times
figure
plot(linspace(n(1),n(end)), fourier_time*ones(100,1))
hold on
plot(n, cheb_time, '*')
grid on
title('Execution time')
legend('Fourier', 'Chebyshev')
xlabel('Interpolation order n')
ylabel('Execution Time (seconds)')

% estimate rho
P = polyfit(n,log(error),1);
rho = exp(-P(1));
disp('Estimated rho:')
disp(rho)

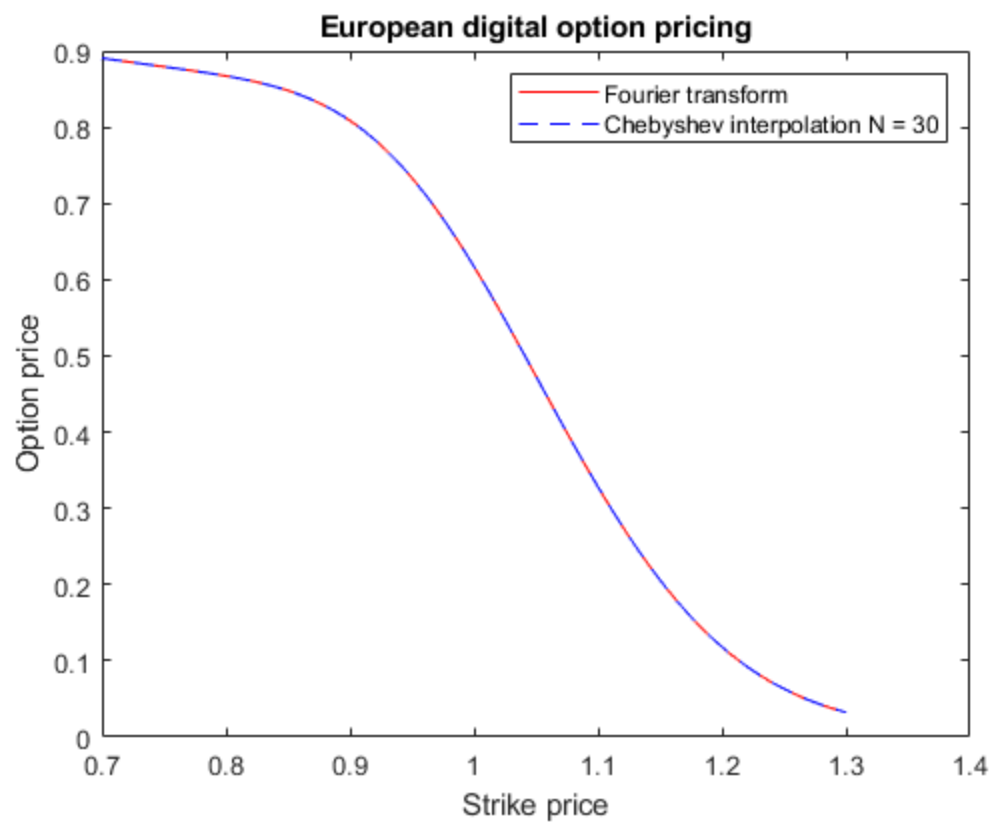
Estimated rho:

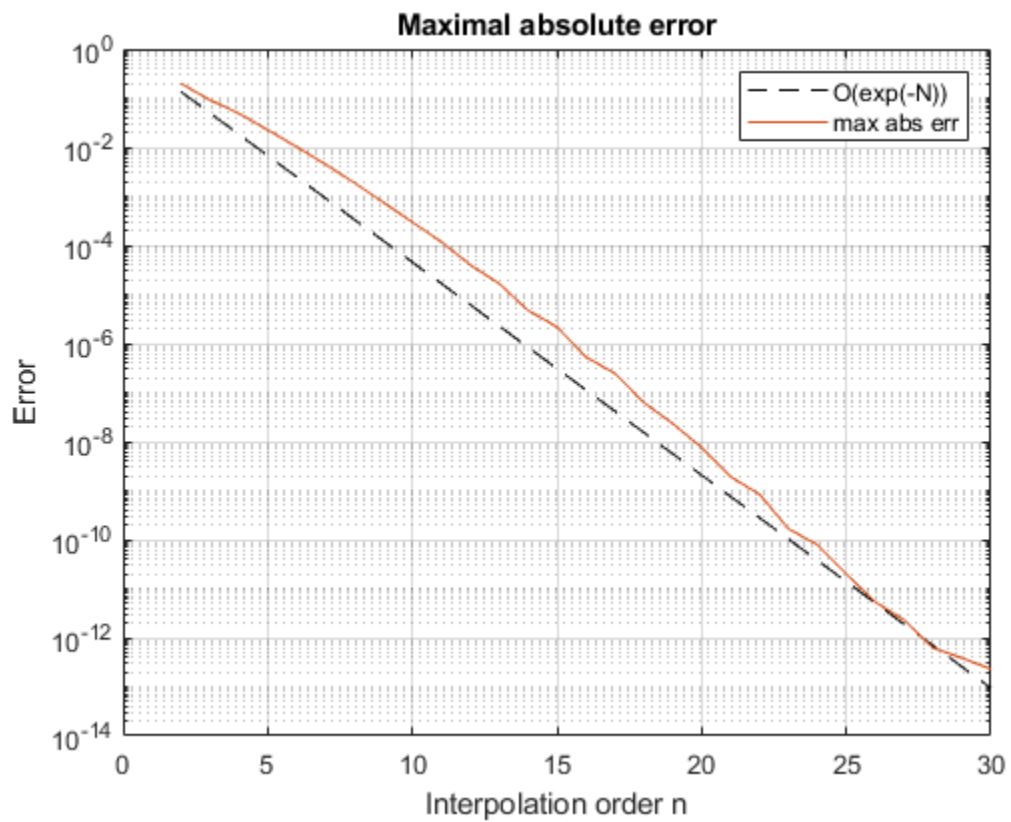
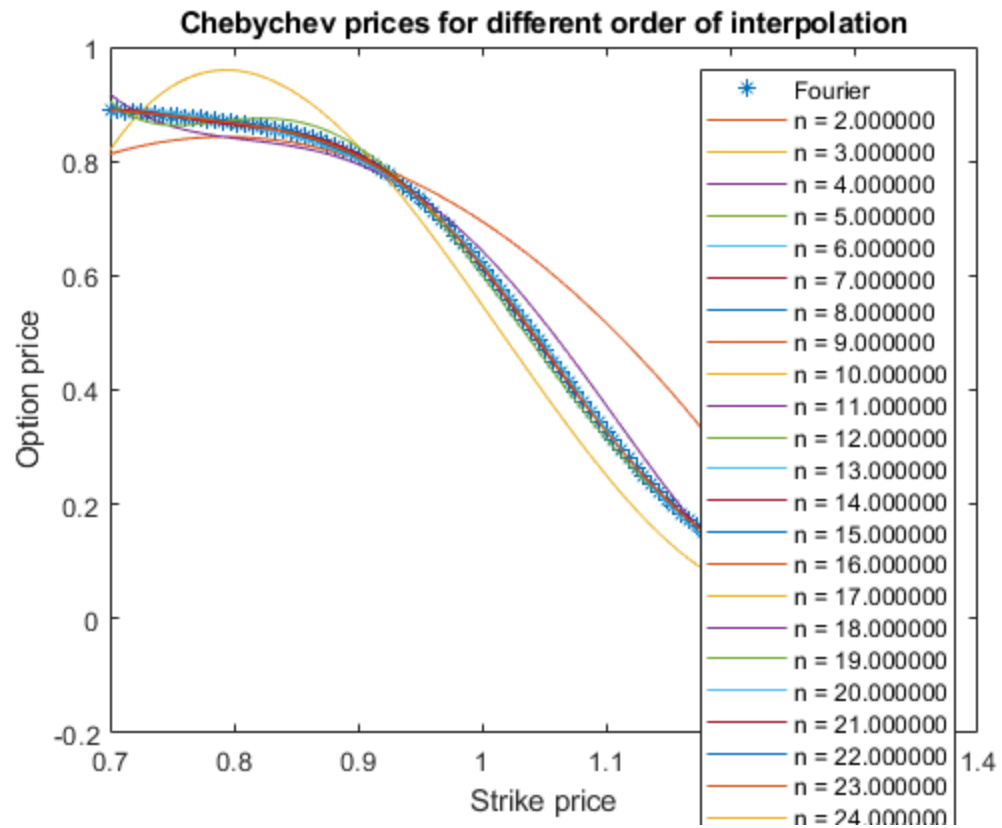
```

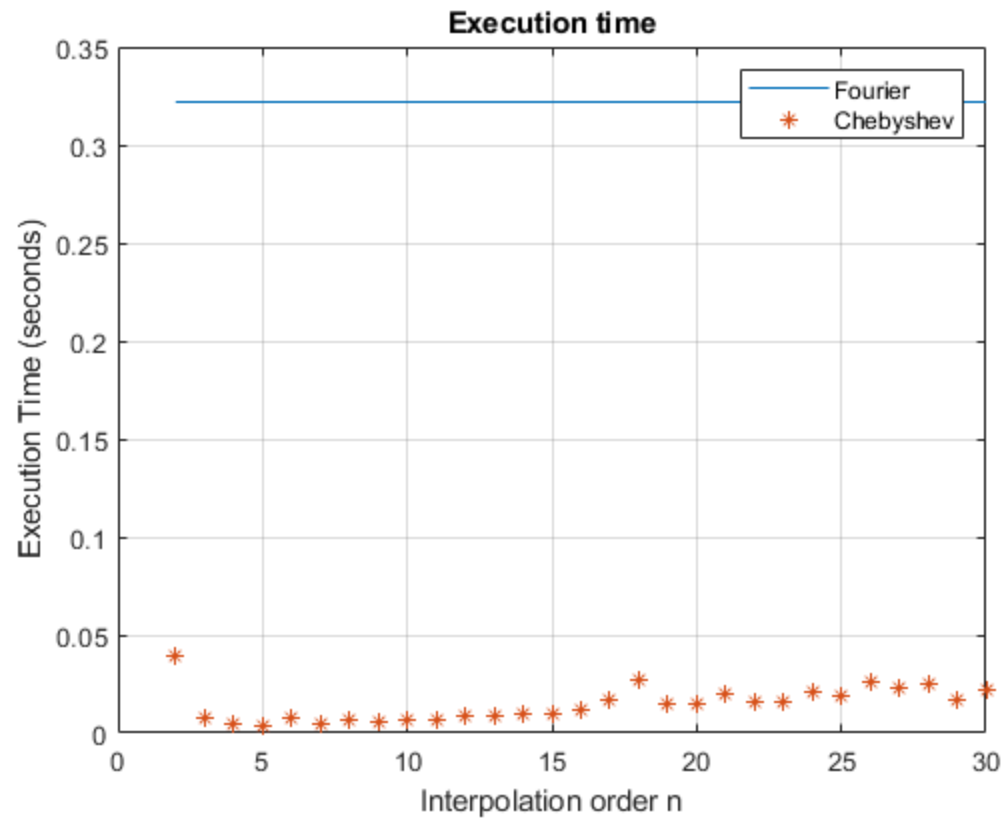
---

---

2.8223







*Published with MATLAB® R2020a*