

**MSDM5058 Information Science**  
**Computational Project I: Time Series Analysis**  
**with Financial Data**

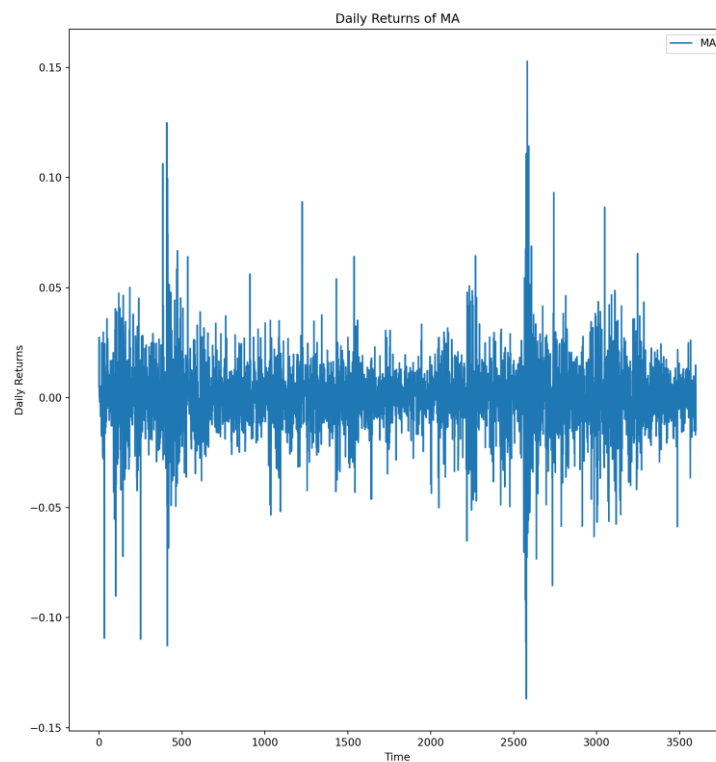
Zhang Mingtao

## 1 Data Preprocessing

We choose stocks Mastercard(MA) and Microsoft(MSFT), download their daily closing prices from Yahoo Finance which start from 2009-01-01 and end at 2024-04-11.

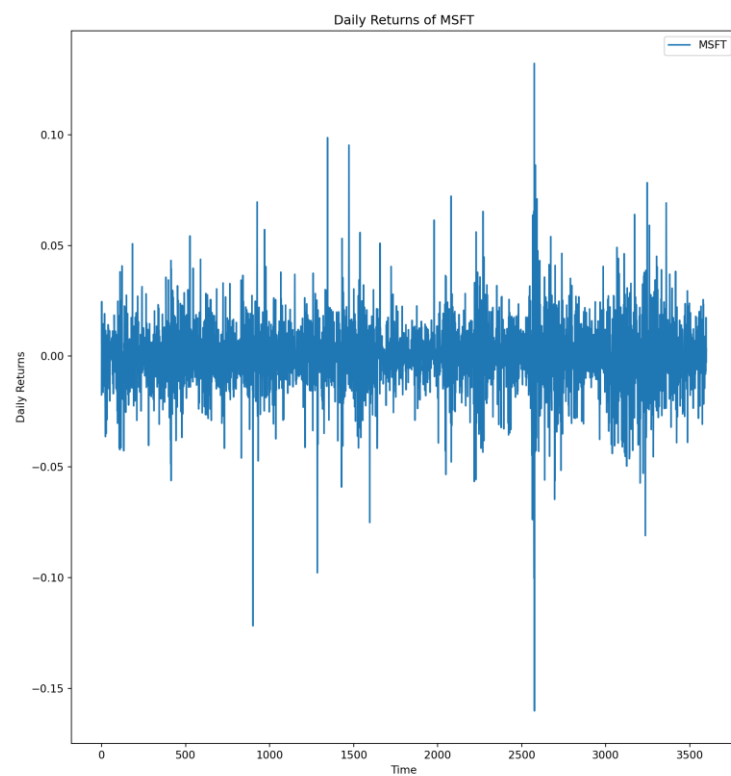
### 1.1 Update $X(t)$ from the daily return

Two  $X(t)$  plots:



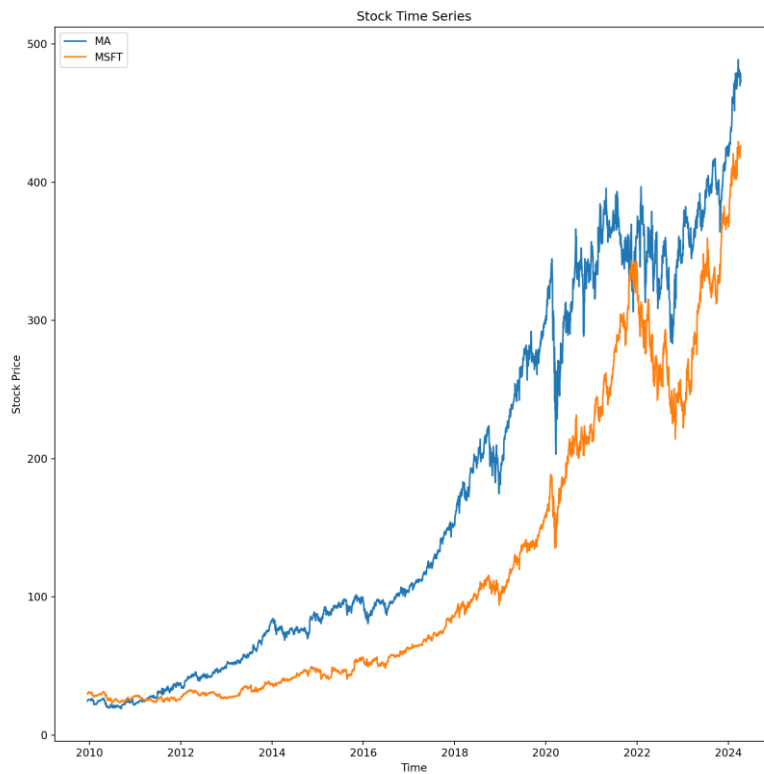
**Fig. 1.1.1.** Daily returns of MA

word 可编辑



**Fig. 1.1.2.** Daily returns of MSFT

## 1.2 Plot $S(t)$



**Fig. 1.2.** Stock prices

## 2 Stationarity and Autocorrelation

We now have two daily-return time series  $X_1$  and  $X_2$ , each with a mean equal to zero.

word 可编辑

## 2.1 Dickey-Fuller test

```
# Perform the augmented Dickey-Fuller test on X to judge its stationarity
result_MA = adfuller(ts_MA_update)
print('Result of ADF test for ts_MA_update:')

## Result of ADF test for ts_MA_update:

print('ADF Statistic:', result_MA[0]) #Large -ve stats --> reject null, time series stationary

## ADF Statistic: -13.268421834087434

print('p-value:', result_MA[1]) #p-value smaller than 0.05 --> reject null, time series stationary

## p-value: 8.129435490642646e-25

print('used lag:', result_MA[2]) #No. of lags used

## used lag: 25

print('critical values: ', result_MA[4]) #Critical values at 1%, 5%, 10%

## critical values: {'1%': -3.432181002494954, '5%': -2.8623490340799274, '10%': -2.5672006619905186}
```

**Fig. 2.1.1.** D-F test for MA

```
result_MSFT = adfuller(ts_MSFT_update)
print('Result of ADF test for ts_MSFT_update:')

## Result of ADF test for ts_MSFT_update:

print('ADF Statistic:', result_MSFT[0]) #Large -ve stats --> reject null, time series stationary

## ADF Statistic: -21.3003560720134

print('p-value:', result_MSFT[1]) #p-value smaller than 0.05 --> reject null, time series stationary

## p-value: 0.0

print('used lag:', result_MSFT[2]) #No. of lags used

## used lag: 8

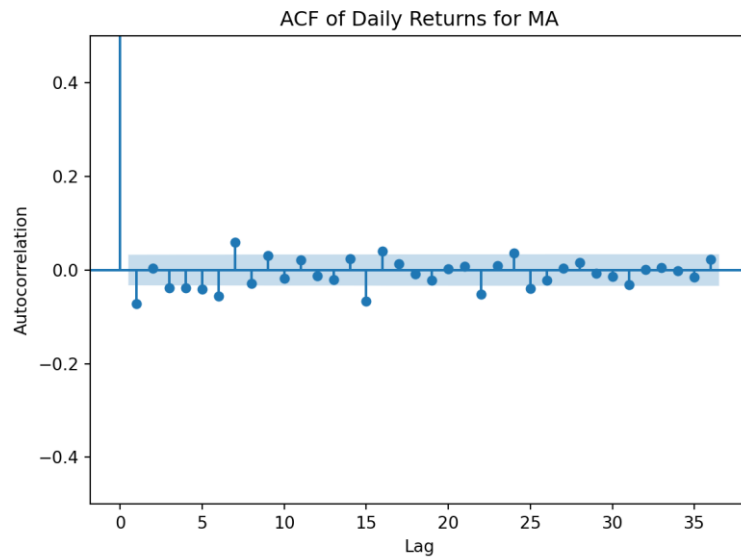
print('critical values: ', result_MSFT[4]) #Critical values at 1%, 5%, 10%

## critical values: {'1%': -3.4321723282160366, '5%': -2.8623452024961766, '10%': -2.567198622175818}
```

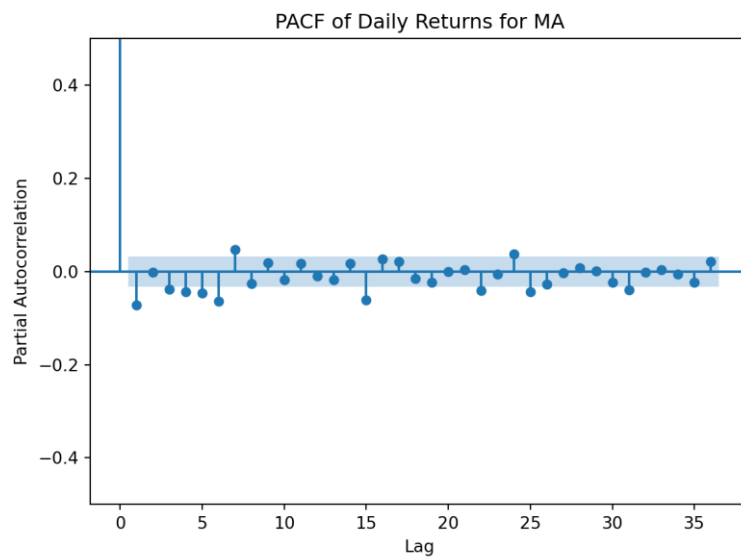
**Fig. 2.1.2.** D-F test for MSFT

Both stock time series are stable.

## 2.2 ACF & PACF plots



**Fig. 2.2.1.** ACF plot for MA



**Fig. 2.2.2.** PACF plot for MA

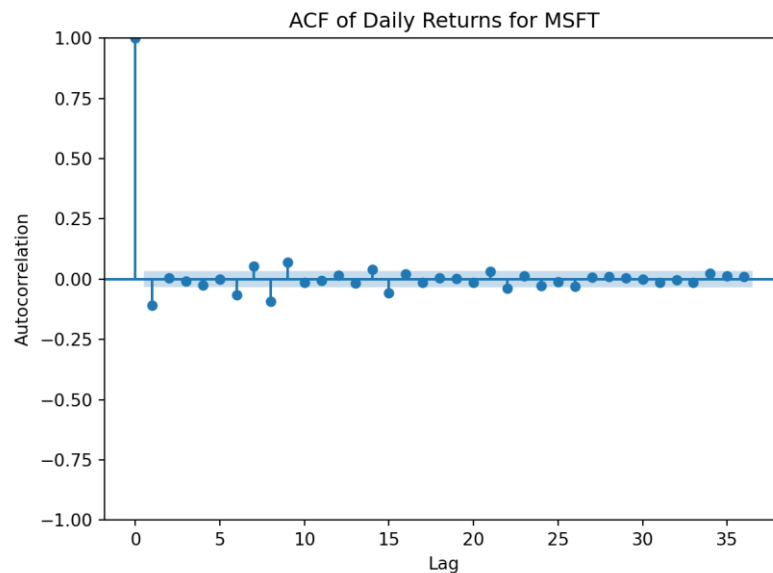
word 可编辑

```
# MA: ARMA(1,1)
# bestfit:
modell = auto_arma(ts_MA_update, start_p=0, start_q=0, seasonal=False, max_p=5, max_q=5, trace=True)

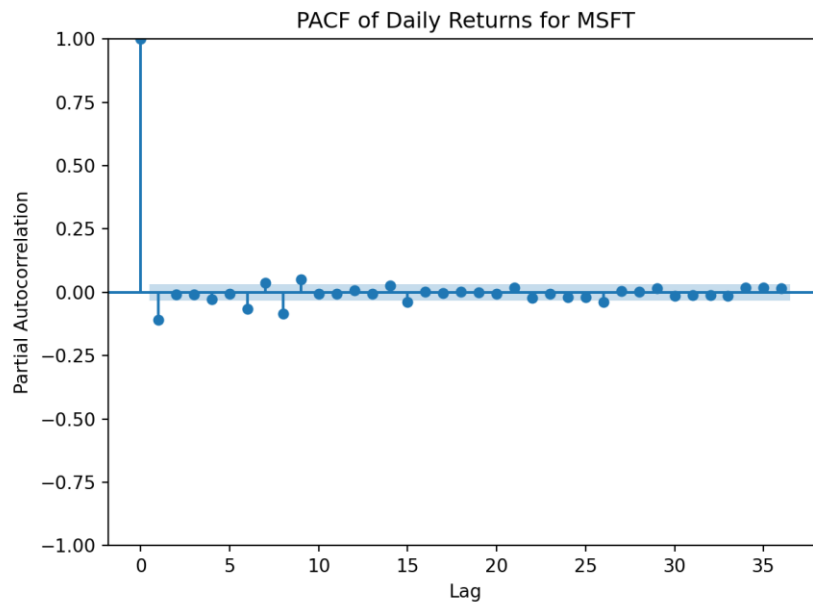
## Performing stepwise search to minimize aic
## ARIMA(0,0,0)(0,0,0)[0] : AIC=-18934.854, Time=0.12 sec
## ARIMA(1,0,0)(0,0,0)[0] : AIC=-18951.881, Time=0.08 sec
## ARIMA(0,0,1)(0,0,0)[0] : AIC=-18951.941, Time=0.09 sec
## ARIMA(1,0,1)(0,0,0)[0] : AIC=-18949.899, Time=0.11 sec
## ARIMA(0,0,2)(0,0,0)[0] : AIC=-18949.959, Time=0.24 sec
## ARIMA(1,0,2)(0,0,0)[0] : AIC=-18954.464, Time=0.12 sec
## ARIMA(2,0,2)(0,0,0)[0] : AIC=-18964.888, Time=1.28 sec
## ARIMA(2,0,1)(0,0,0)[0] : AIC=-18947.892, Time=0.30 sec
## ARIMA(3,0,2)(0,0,0)[0] : AIC=-18962.797, Time=1.49 sec
## ARIMA(2,0,3)(0,0,0)[0] : AIC=-18962.634, Time=1.49 sec
## ARIMA(1,0,3)(0,0,0)[0] : AIC=-18957.689, Time=0.30 sec
## ARIMA(3,0,1)(0,0,0)[0] : AIC=-18951.410, Time=0.12 sec
## ARIMA(3,0,3)(0,0,0)[0] : AIC=-18961.290, Time=2.24 sec
## ARIMA(2,0,2)(0,0,0)[0] intercept : AIC=-18943.932, Time=1.37 sec
##
## Best model: ARIMA(2,0,2)(0,0,0)[0]
## Total fit time: 9.322 seconds
```

**Fig. 2.2.3.** ARMA model for MA

We guess ARMA(1,1) model from the ACF and PACF plots for MA, and the best model from the calculation by AIC is ARMA(2,2).



**Fig. 2.2.4.** ACF plot for MSFT



**Fig. 2.2.5.** PACF plot for MSFT

```
# MSFT: ARMA(1,1)
# bestfit:
model2 = auto_arima(ts_MSFT_update, start_p=0, start_q=0, seasonal=False, max_p=5, max_q=5, trace=True)

## Performing stepwise search to minimize aic
## ARIMA(0,0,0)(0,0,0)[0] : AIC=-19437.141, Time=0.14 sec
## ARIMA(1,0,0)(0,0,0)[0] : AIC=-19476.958, Time=0.20 sec
## ARIMA(0,0,1)(0,0,0)[0] : AIC=-19477.222, Time=0.08 sec
## ARIMA(1,0,1)(0,0,0)[0] : AIC=-19475.221, Time=0.10 sec
## ARIMA(0,0,2)(0,0,0)[0] : AIC=-19475.226, Time=0.26 sec
## ARIMA(1,0,2)(0,0,0)[0] : AIC=-19473.217, Time=0.29 sec
## ARIMA(0,0,1)(0,0,0)[0] intercept : AIC=-19475.222, Time=0.19 sec
##
## Best model: ARIMA(0,0,1)(0,0,0)[0]
## Total fit time: 1.264 seconds
```

**Fig. 2.2.6.** ARMA model for MSFT

We guess ARMA(1,1) model from the ACF and PACF plots for MSFT, and the best model from the calculation by AIC is ARMA(0,1).



2.3 ACF plots of absolute value

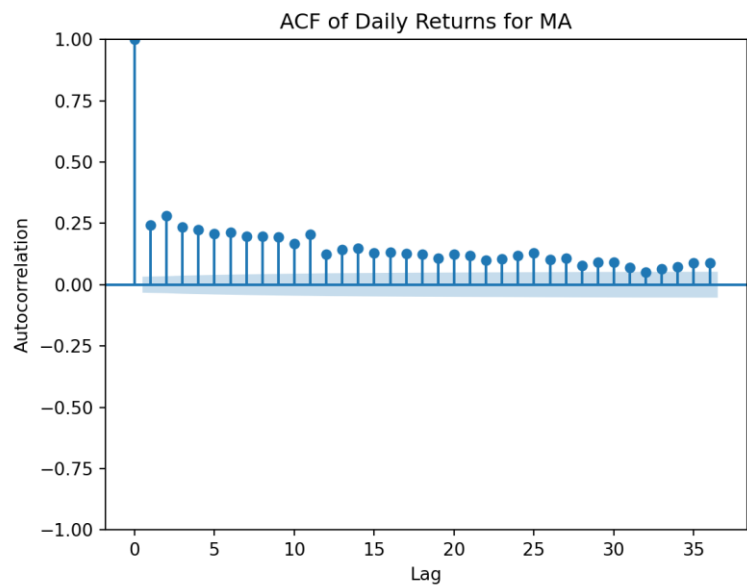


Fig. 2.3.1. ACF plot of absolute value for MA

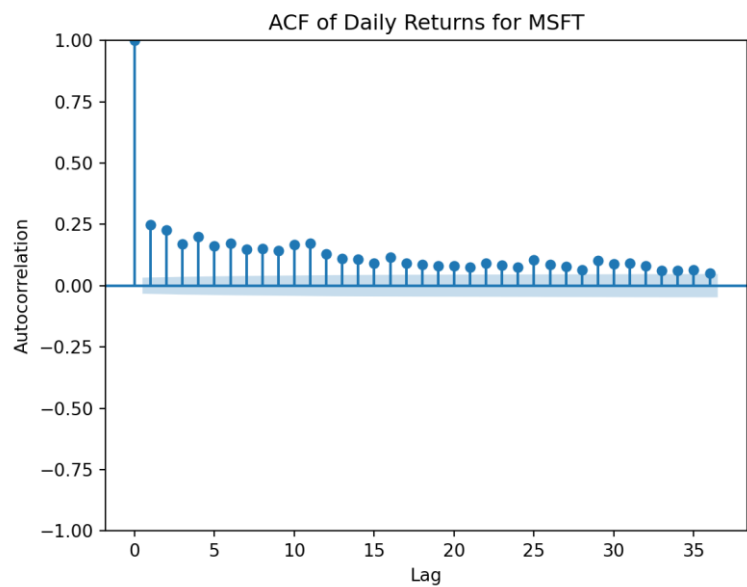


Fig. 2.3.2. ACF plot of absolute value for MSFT

Absolute value eliminates the direction of returns and focuses only on the magnitude of the change. It captures the dependence between the magnitude of X and its past magnitudes.

Also it helps identify non-linear patterns in data.

## 3 Fractal Behaviour of Time Series

### 3.1 Hurst Exponent

```
import nolds
from sklearn.linear_model import LinearRegression
np.random.seed(1234)
H1 = nolds.hurst_rs(ts_MA_update)
print(H1)
# H1=0.451 < 0.5, the ts_MA has a bit of negative effect, indicating that the future trend is opposite to the past.
```

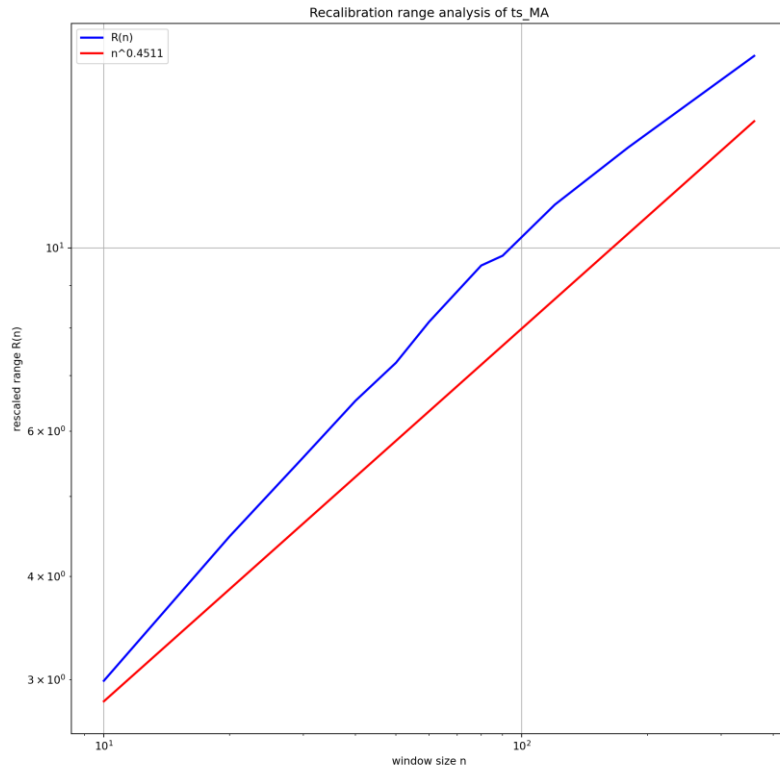
```
## 0.45109852525343097
```

```
np.random.seed(111)
H2 = nolds.hurst_rs(ts_MSFT_update)
print(H2)
# H2=0.481 < 0.5, the ts_MA has a bit of negative effect, indicating that the future trend is opposite to the past.
```

$H1(MA)=0.451 < 0.5$ , the  $ts\_MA$  has a bit of negative effect, indicating that the future trend is opposite to the past.

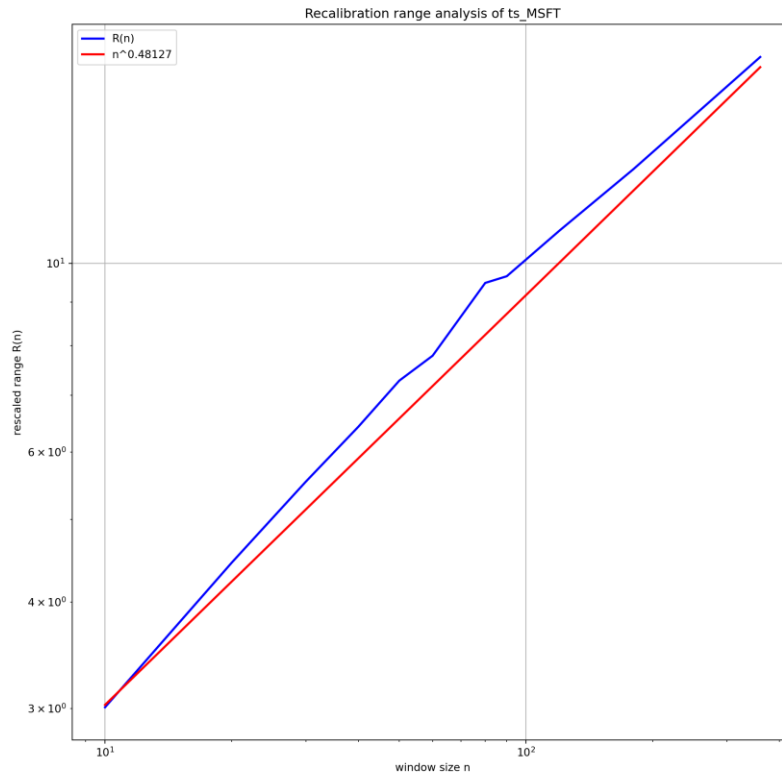
$H2(MSFT)=0.481 < 0.5$ , the  $ts\_MA$  has a bit of negative effect, indicating that the future trend is opposite to the past.

word 可编辑



**Fig. 3.1.1.** Recalibration range analysis of MA

The function written manually shows the R/S exponent of MA is 0.491523, a little bit more than the function in the python library, which is caused by the different window ranges and the finite length of time series.

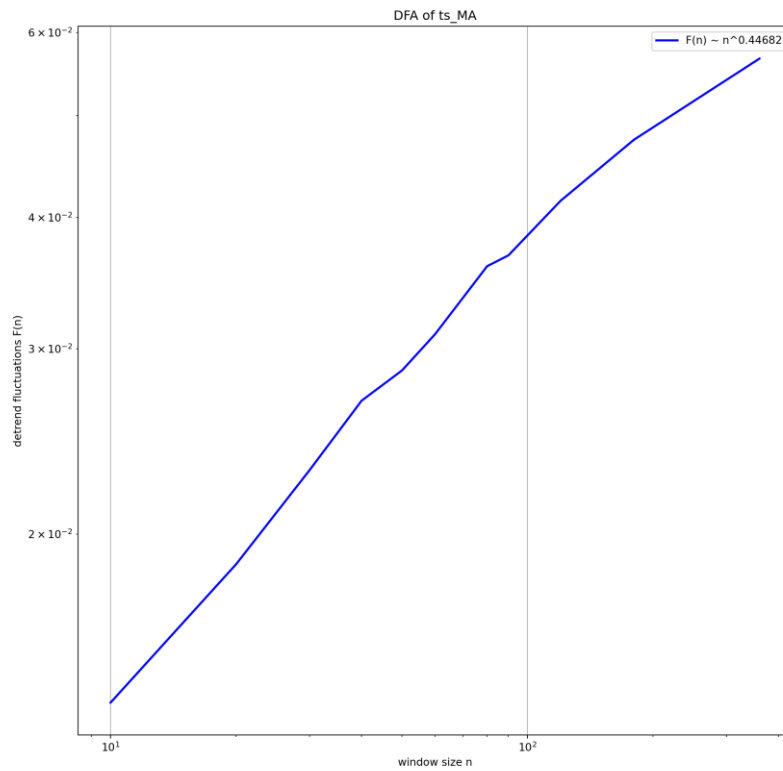


**Fig. 3.1.2.** Recalibration range analysis of MSFT

The function written manually shows the R/S exponent of MSFT is 0.49073, a little bit more than the function in the python library, which is also caused by the different window ranges and the finite length of time series.

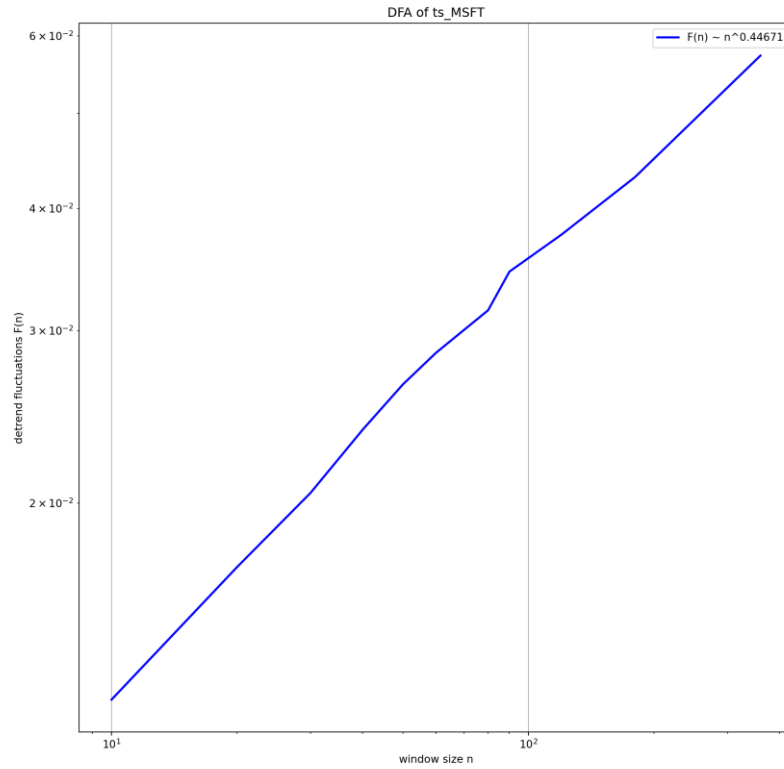
word 可编辑

### 3.2 Detrended Fluctuation Analysis (DFA)



**Fig. 3.2.1.** DFA plot of MA

DFA exponent (MA) is 0.4468 , which is consistent with the Hurst exponent (MA) = 0.451.



**Fig. 3.2.2.** DFA plot of MSFT

DFA exponent (MSFT) is 0.4467 , which is consistent with the Hurst exponent (MSFT) = 0.481, although there exists some inevitable errors caused by different window ranges and the finite length of time series.

word 可编辑

### 3.3 Multifractality

```
## 3.3 Multifractality

def calculate_M(q, tau_values, Y):
    N = len(Y)
    M = np.zeros(tau_values.shape)
    for i, t in enumerate(tau_values):
        if t >= 0:
            M[i] = np.mean(np.abs(Y[t:] - Y[:N-t]) ** q)
        else:
            M[i] = np.mean(np.abs(Y[:N+t] - Y[-t:]) ** q)
    return M

q_values = np.linspace(1, 5, 9)
tau_values = np.arange(1, 50)
```

Fig. 3.3.1. Main function to compute  $M(q, \tau)$

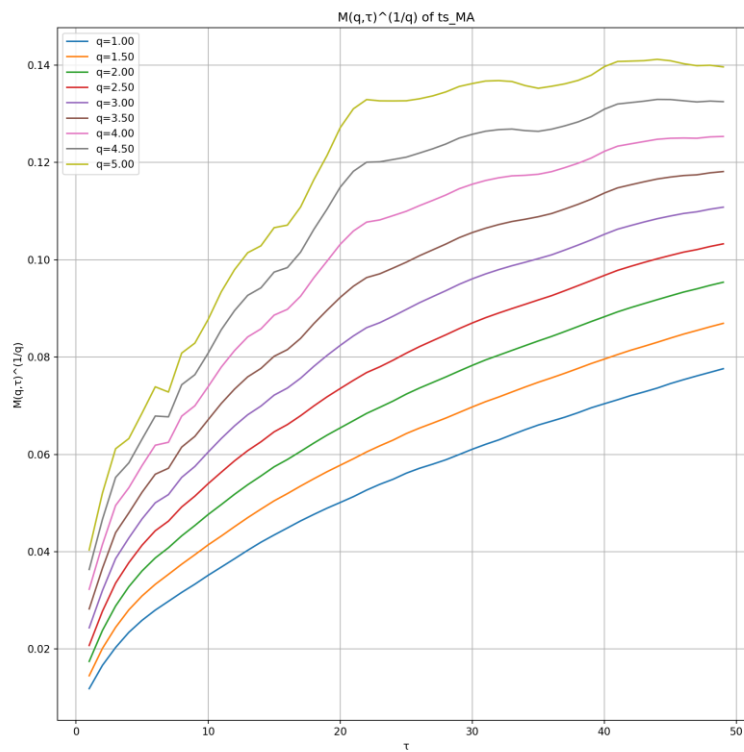
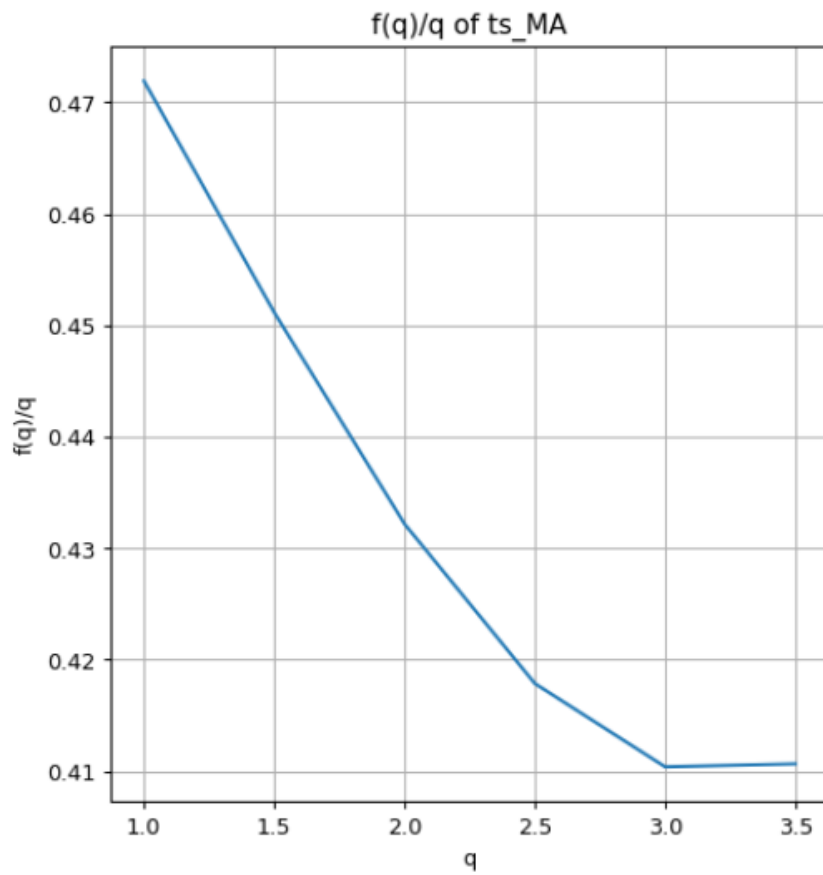


Fig. 3.3.2.  $M(q, \tau)^{1/q}$  plot of MA



**Fig. 3.3.3.**  $f(q)/q$  plot of MA

When  $q = 1$ ,  $f(1) \approx 0.472$ , which is a little bit higher than  $H1 = 0.451$ , but is still less than 0.5 so there exists multifractal.



word 可编辑

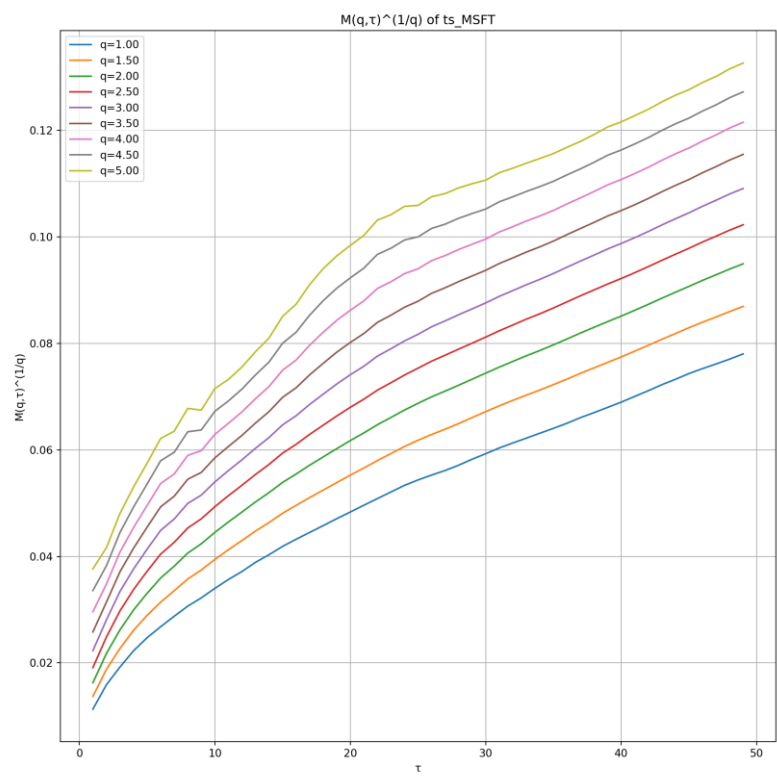
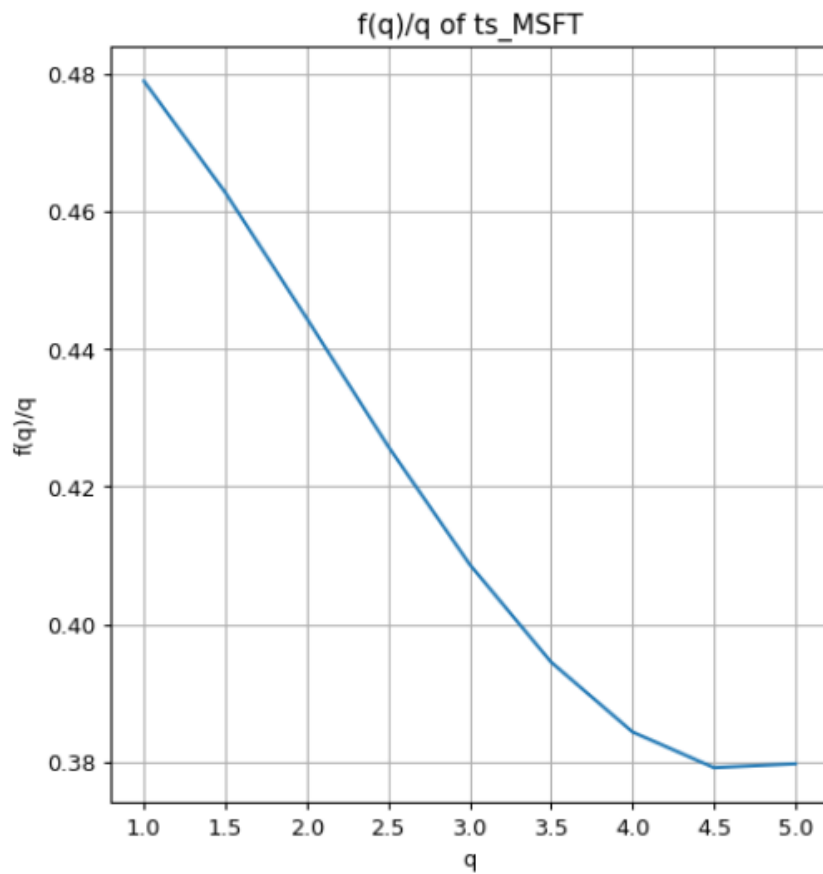


Fig. 3.3.4.  $M(q,\tau)^{1/q}$  plot of MSFT



**Fig. 3.3.5.**  $f(q)/q$  plot of MSFT

When  $q = 1$ ,  $f(1) \approx 0.479$ , which is a little bit smaller than  $H_2 = 0.481$ , but is still less than 0.5 so there exists multifractal.

Both values of  $f(1)$  are roughly around the Hurst exponents.

word 可编辑

Then we consider multifractal detrended fluctuation analysis (MDFA), which generalizes DFA:

```
# MDFA
def MDFA(X, q, n):
    L = len(X)//n
    nf = int(L*n)

    y = np.cumsum(X - np.mean(X))
    y_hat = []
    for i in range(int(L)):
        x = np.arange(1, n+1, 1)
        y_temp = y[int(i*n+1)-1:int((i+1)*n)]
        coef = np.polyfit(x, y_temp, 1)
        y_hat.append(np.polyval(coef, x))
    fn = (sum((np.asarray(y)-np.asarray(y_hat).reshape(-1))**q)/nf)**(1/q)
    return fn

q_values = np.array([1, 2, 4, 6, 8, 10])
```

Fig. 3.3.6. Main function to compute MDFA

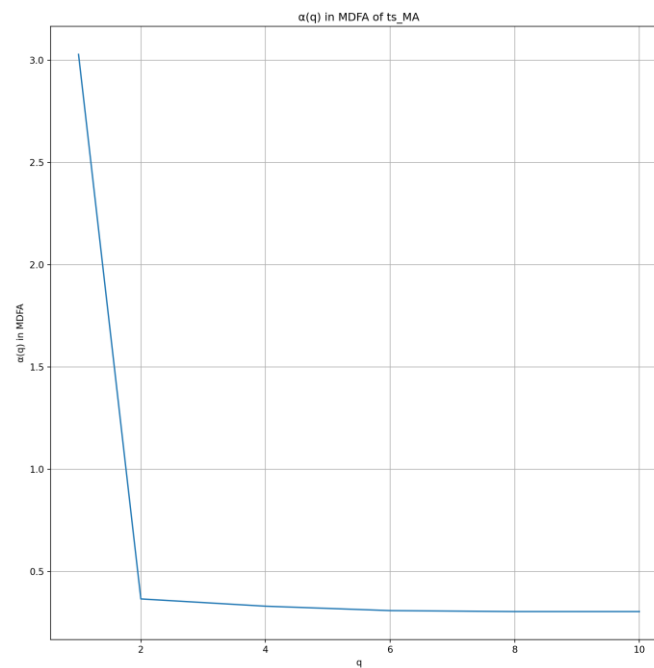
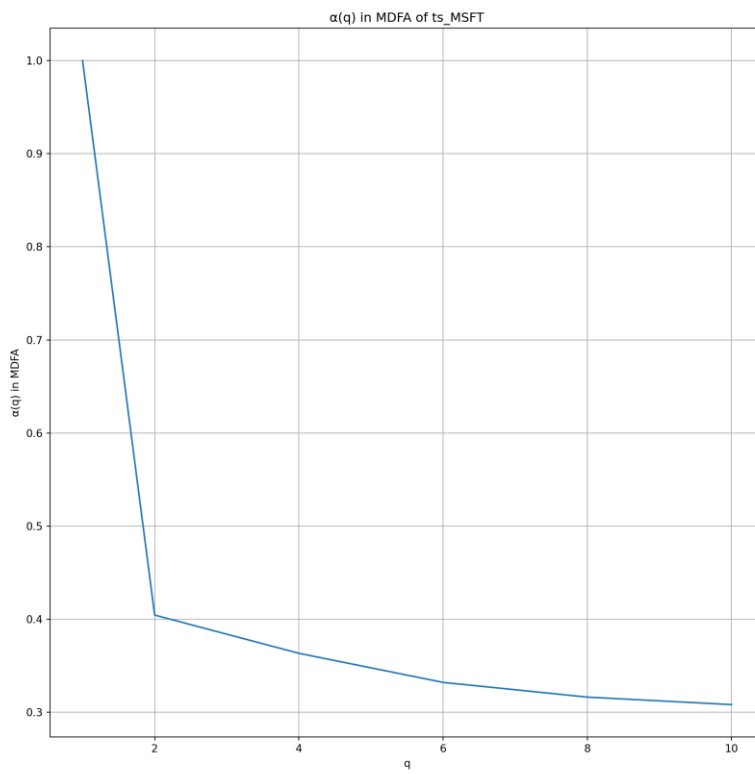


Fig. 3.3.7.  $\alpha(q)$  in MDFA of MA



**Fig. 3.3.8.**  $\alpha(q)$  in MDFA of MSFT

Still both are consistent with the ones obtained before.

word 可编辑

## 4 Granger Causality

Next, we are going to study the causal relation between X1 and X2 by fitting a VARMA model to them:

```
### 4 Granger Causality
from statsmodels.tsa.statespace.varmax import VARMAX
merged_data = pd.concat([pd.Series(ts_MA_update, name='MA'), pd.Series(ts_MSFT_update, name='MSFT')], axis=1)
train_size = int(len(merged_data))-3
train_data, test_data = merged_data[:train_size], merged_data[train_size:]

best_aic = np.inf
best_order = None
best_model = None
pq_range = range(2) # 取值范围
for p in pq_range:
    for q in pq_range:
        try:
            model = VARMAX(train_data, order=(p, q))
            result = model.fit()
            aic = result.aic
            if aic < best_aic:
                best_aic = aic
                best_order = (p, q)
                best_model = result
        except:
            continue

## D:\app\python\Lib\site-packages\statsmodels\tsa\statespace\varmax.py:161: EstimationWarning: Estimation of VARMA(p,q) models
is not generically robust, due especially to identification issues.
## warn('Estimation of VARMA(p,q) models is not generically robust,'

print('Best order:', best_order)

## Best order: (1, 0)

print('Best AIC:', best_aic)

## Best AIC: -39740.64107274744
```

**Fig. 4.1.** Main function to choose best VARMA model

We can write codes to determine the best VARMA model orders by aic. The coefficients of this best VARMA(1,0) are all significant, except intercept.MA and intercept.MSFT. Other coefficients' p-values are < 0.05.

```
coefficients = best_model.params
print(coefficients)
```

```
## intercept.MA      -0.000003
## intercept.MSFT    -0.000007
## L1.MA.MA          -0.044291
## L1.MSFT.MA        -0.053631
## L1.MA.MSFT        -0.046009
## L1.MSFT.MSFT      -0.079570
## sqrt.var.MA       0.017384
## sqrt.cov.MA.MSFT  0.009028
## sqrt.var.MSFT     0.013402
## dtype: float64
```

**Fig. 4.2.** Coefficients in Model VARMA(1,0)

```
>>> best_model.summary()
<class 'statsmodels.iolib.summary.Summary'>
"""
                    Statespace Model Results
=====
Dep. Variable:      ['MA', 'MSFT']    No. Observations:      3597
Model:              VAR(1)            Log Likelihood         19879.321
                   + intercept        AIC                     -39740.641
Date:              周六, 20 4月 2024   BIC                     -39684.950
Time:              22:28:09           HQIC                    -39720.793
Sample:            0
                   - 3597
Covariance Type:    opg
=====
Ljung-Box (L1) (Q):      0.00, 0.00   Jarque-Bera (JB):      10337.14, 8463.40
Prob(Q):                0.96, 0.97   Prob(JB):              0.00, 0.00
Heteroskedasticity (H):  1.25, 1.15   Skew:                  -0.04, -0.09
Prob(H) (two-sided):    0.00, 0.01   Kurtosis:              11.30, 10.51
=====
                    Results for equation MA
=====
              coef    std err          z      P>|z|      [0.025    0.975]
-----
intercept -2.915e-06    0.000     -0.010    0.992     -0.001     0.001
L1.MA      -0.0443     0.014     -3.118    0.002     -0.072     -0.016
L1.MSFT    -0.0536     0.017     -3.124    0.002     -0.087     -0.020
=====
                    Results for equation MSFT
=====
              coef    std err          z      P>|z|      [0.025    0.975]
-----
intercept -7.295e-06    0.000     -0.027    0.979     -0.001     0.001
L1.MA      -0.0460     0.015     -3.094    0.002     -0.075     -0.017
L1.MSFT    -0.0796     0.014     -5.568    0.000     -0.108     -0.052
=====
                    Error covariance matrix
=====
              coef    std err          z      P>|z|      [0.025    0.975]
-----
sqrt.var.MA      0.0174    9.8e-05   177.381    0.000     0.017     0.018
sqrt.cov.MA.MSFT 0.0090    0.000    62.597    0.000     0.009     0.009
sqrt.var.MSFT    0.0134    7.63e-05  175.603    0.000     0.013     0.014
=====
```

**Fig. 4.3.** Significance of coefficients in Model VARMA(1,0)

Then we do F-tests to judge the Granger causality between X1 and X2:

```
Granger Causality
number of lags (no zero) 1
ssr based F test:      F=6.1889 , p=0.0129 , df_denom=3593, df_num=1
ssr based chi2 test:   chi2=6.1941 , p=0.0128 , df=1
likelihood ratio test: chi2=6.1888 , p=0.0129 , df=1
parameter F test:      F=6.1889 , p=0.0129 , df_denom=3593, df_num=1

Granger Causality
number of lags (no zero) 2
ssr based F test:      F=3.3324 , p=0.0358 , df_denom=3590, df_num=2
ssr based chi2 test:   chi2=6.6741 , p=0.0355 , df=2
likelihood ratio test: chi2=6.6679 , p=0.0357 , df=2
parameter F test:      F=3.3324 , p=0.0358 , df_denom=3590, df_num=2

Granger Causality
number of lags (no zero) 3
ssr based F test:      F=2.3702 , p=0.0686 , df_denom=3587, df_num=3
ssr based chi2 test:   chi2=7.1243 , p=0.0680 , df=3
likelihood ratio test: chi2=7.1173 , p=0.0683 , df=3
parameter F test:      F=2.3702 , p=0.0686 , df_denom=3587, df_num=3

Granger Causality
number of lags (no zero) 4
ssr based F test:      F=1.7206 , p=0.1425 , df_denom=3584, df_num=4
ssr based chi2 test:   chi2=6.8998 , p=0.1413 , df=4
likelihood ratio test: chi2=6.8932 , p=0.1416 , df=4
parameter F test:      F=1.7206 , p=0.1425 , df_denom=3584, df_num=4

Granger Causality
number of lags (no zero) 5
ssr based F test:      F=1.9892 , p=0.0771 , df_denom=3581, df_num=5
ssr based chi2 test:   chi2=9.9764 , p=0.0759 , df=5
likelihood ratio test: chi2=9.9625 , p=0.0763 , df=5
parameter F test:      F=1.9892 , p=0.0771 , df_denom=3581, df_num=5

Granger Causality
number of lags (no zero) 6
ssr based F test:      F=1.8176 , p=0.0917 , df_denom=3578, df_num=6
ssr based chi2 test:   chi2=10.9452 , p=0.0901 , df=6
likelihood ratio test: chi2=10.9285 , p=0.0906 , df=6
parameter F test:      F=1.8176 , p=0.0917 , df_denom=3578, df_num=6
```

**Fig. 4.4.1.** Granger causality result 1

```

Granger Causality
number of lags (no zero) 7
ssr based F test:      F=2.3657 , p=0.0207 , df_denom=3575, df_num=7
ssr based chi2 test:   chi2=16.6297 , p=0.0199 , df=7
likelihood ratio test: chi2=16.5913 , p=0.0202 , df=7
parameter F test:     F=2.3657 , p=0.0207 , df_denom=3575, df_num=7

Granger Causality
number of lags (no zero) 8
ssr based F test:      F=2.4338 , p=0.0127 , df_denom=3572, df_num=8
ssr based chi2 test:   chi2=19.5628 , p=0.0121 , df=8
likelihood ratio test: chi2=19.5097 , p=0.0124 , df=8
parameter F test:     F=2.4338 , p=0.0127 , df_denom=3572, df_num=8

Granger Causality
number of lags (no zero) 9
ssr based F test:      F=2.2812 , p=0.0151 , df_denom=3569, df_num=9
ssr based chi2 test:   chi2=20.6405 , p=0.0143 , df=9
likelihood ratio test: chi2=20.5814 , p=0.0146 , df=9
parameter F test:     F=2.2812 , p=0.0151 , df_denom=3569, df_num=9

Granger Causality
number of lags (no zero) 10
ssr based F test:      F=2.0727 , p=0.0234 , df_denom=3566, df_num=10
ssr based chi2 test:   chi2=20.8486 , p=0.0222 , df=10
likelihood ratio test: chi2=20.7883 , p=0.0226 , df=10
parameter F test:     F=2.0727 , p=0.0234 , df_denom=3566, df_num=10

Granger Causality
number of lags (no zero) 11
ssr based F test:      F=1.9223 , p=0.0323 , df_denom=3563, df_num=11
ssr based chi2 test:   chi2=21.2820 , p=0.0306 , df=11
likelihood ratio test: chi2=21.2191 , p=0.0312 , df=11
parameter F test:     F=1.9223 , p=0.0323 , df_denom=3563, df_num=11

Granger Causality
number of lags (no zero) 12
ssr based F test:      F=1.7986 , p=0.0429 , df_denom=3560, df_num=12
ssr based chi2 test:   chi2=21.7345 , p=0.0406 , df=12
likelihood ratio test: chi2=21.6689 , p=0.0414 , df=12
parameter F test:     F=1.7986 , p=0.0429 , df_denom=3560, df_num=12

```

**Fig. 4.4.2.** Granger causality result 2



word 可编辑

```
Granger Causality
number of lags (no zero) 13
ssr based F test:      F=1.7320 , p=0.0484 , df_denom=3557, df_num=13
ssr based chi2 test:   chi2=22.6866 , p=0.0456 , df=13
likelihood ratio test: chi2=22.6151 , p=0.0465 , df=13
parameter F test:      F=1.7320 , p=0.0484 , df_denom=3557, df_num=13

Granger Causality
number of lags (no zero) 14
ssr based F test:      F=1.5677 , p=0.0803 , df_denom=3554, df_num=14
ssr based chi2 test:   chi2=22.1268 , p=0.0760 , df=14
likelihood ratio test: chi2=22.0587 , p=0.0774 , df=14
parameter F test:      F=1.5677 , p=0.0803 , df_denom=3554, df_num=14

Granger Causality
number of lags (no zero) 15
ssr based F test:      F=1.4687 , p=0.1077 , df_denom=3551, df_num=15
ssr based chi2 test:   chi2=22.2227 , p=0.1021 , df=15
likelihood ratio test: chi2=22.1540 , p=0.1038 , df=15
parameter F test:      F=1.4687 , p=0.1077 , df_denom=3551, df_num=15

Granger Causality
number of lags (no zero) 16
ssr based F test:      F=1.4522 , p=0.1083 , df_denom=3548, df_num=16
ssr based chi2 test:   chi2=23.4520 , p=0.1022 , df=16
likelihood ratio test: chi2=23.3756 , p=0.1041 , df=16
parameter F test:      F=1.4522 , p=0.1083 , df_denom=3548, df_num=16

Granger Causality
number of lags (no zero) 17
ssr based F test:      F=1.5051 , p=0.0831 , df_denom=3545, df_num=17
ssr based chi2 test:   chi2=25.8395 , p=0.0774 , df=17
likelihood ratio test: chi2=25.7467 , p=0.0792 , df=17
parameter F test:      F=1.5051 , p=0.0831 , df_denom=3545, df_num=17

Granger Causality
number of lags (no zero) 18
ssr based F test:      F=1.5246 , p=0.0718 , df_denom=3542, df_num=18
ssr based chi2 test:   chi2=27.7292 , p=0.0663 , df=18
likelihood ratio test: chi2=27.6224 , p=0.0680 , df=18
parameter F test:      F=1.5246 , p=0.0718 , df_denom=3542, df_num=18
```

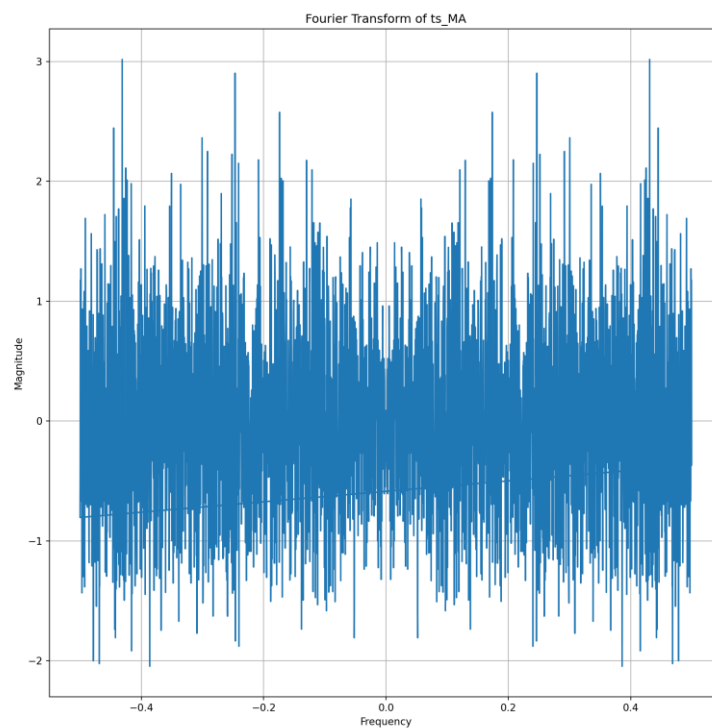
**Fig. 4.4.3. Granger causality result 3**

When  $p\text{-value} < 0.05$ , we refuse  $H_0$ , there exists Granger Causality at corresponding lags. Specifically, they are lag 1,2,7,8,9,10,11,12,13.

## 5 Fourier Transform and Power Spectrum

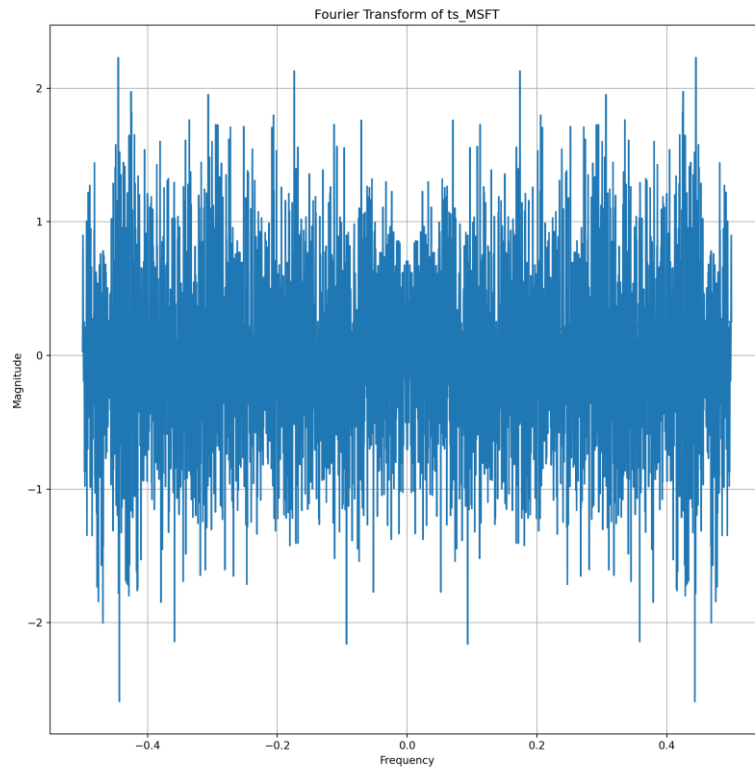
### 5.1 Fourier Transform

We compute the Fourier transform of  $X$ , then plot the magnitude of its Fourier coefficients against frequency.



**Fig. 5.1.1.** Fourier Transform of MA

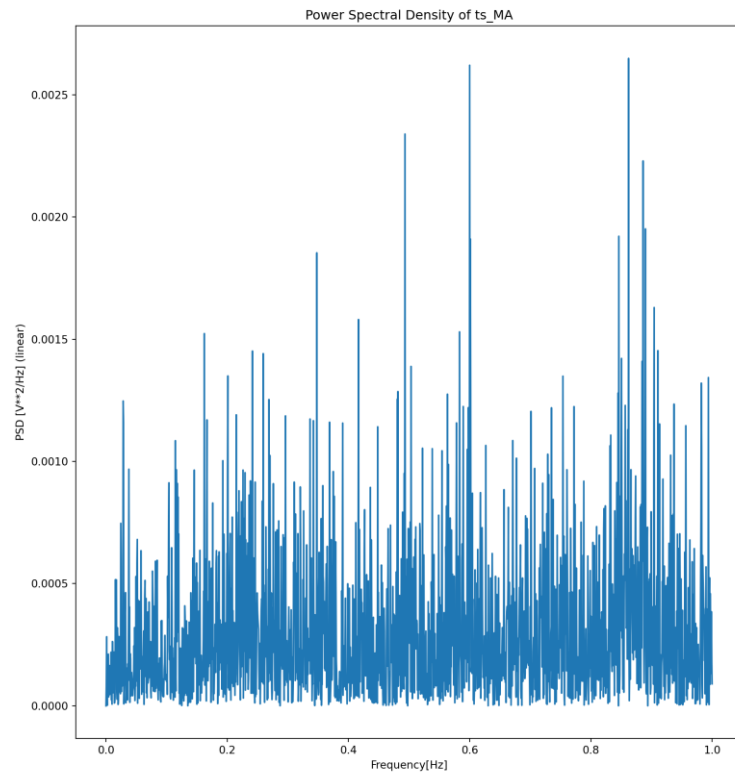
word 可编辑



**Fig. 5.1.2.** Fourier Transform of MSFT

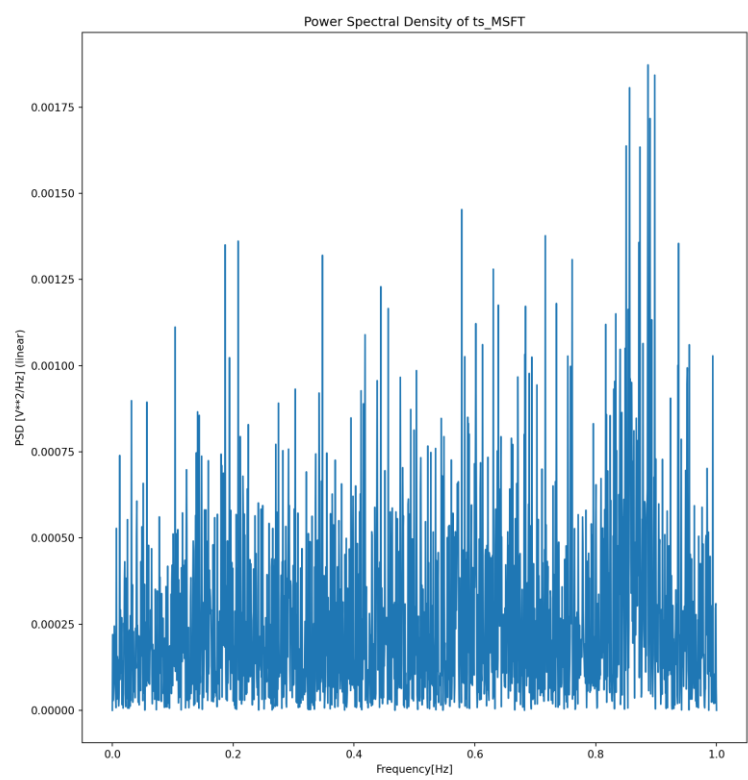
## 5.2 Power Spectral Density

Then we plot the power spectral density (PSD) of  $X$  against frequency, with doubled sampling rate.



**Fig. 5.2.1.** Power Spectral Density of MA

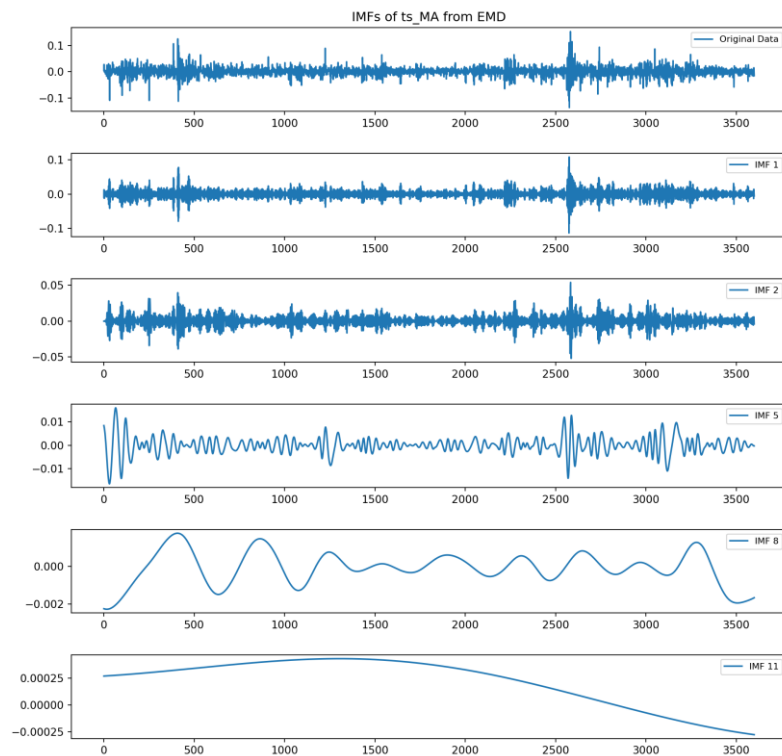
word 可编辑



**Fig. 5.2.2.** Power Spectral Density of MSFT

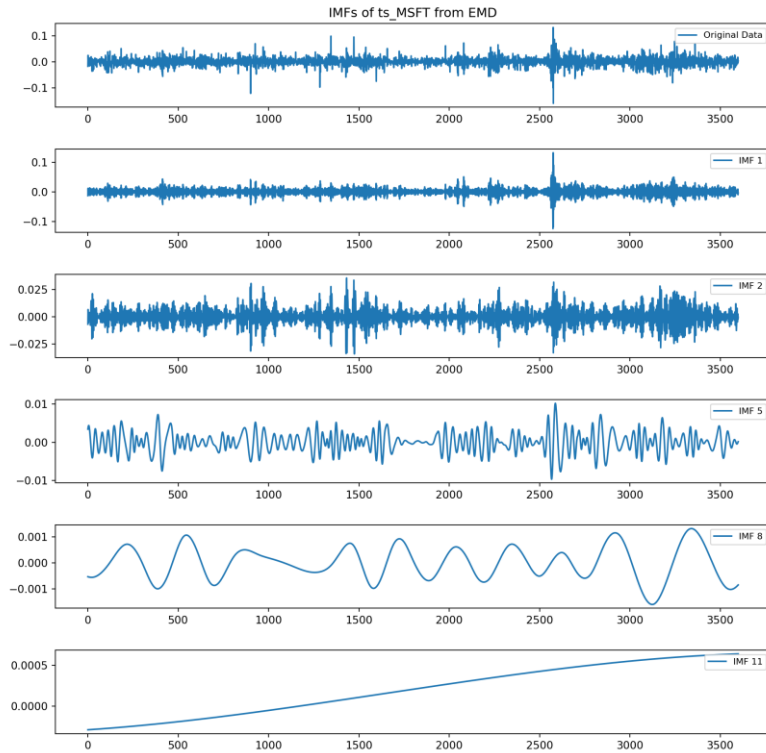
## 6 Empirical Mode Decomposition

First we decompose  $X$  into its IMFs:



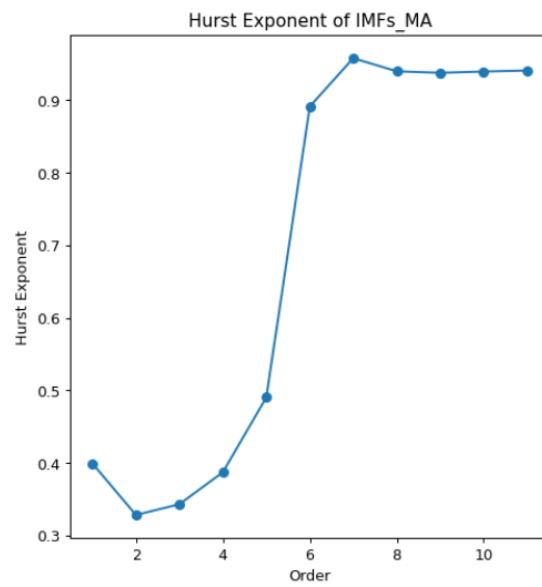
**Fig. 6.1.** IMFs of MA from EMD method

word 可编辑

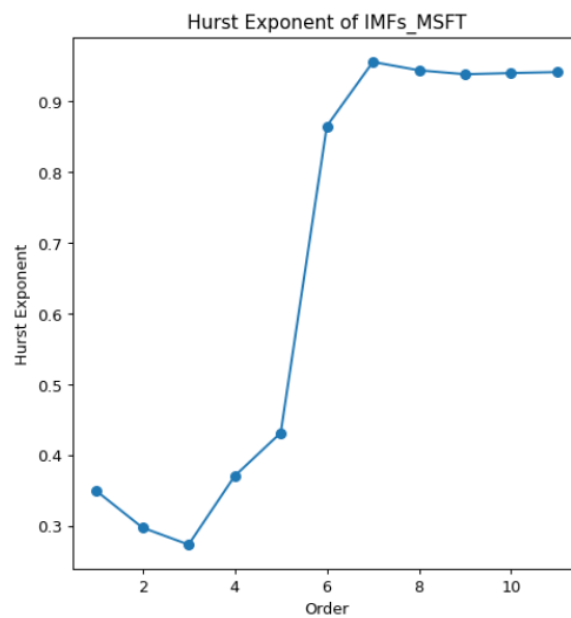


**Fig. 6.2.** IMFs of MSFT from EMD method

Then we compute the Hurst exponent of each of the IMFs, and plot their Hurst exponents against their orders.



**Fig. 6.3.** Hurst Exponent of IMFs\_MA



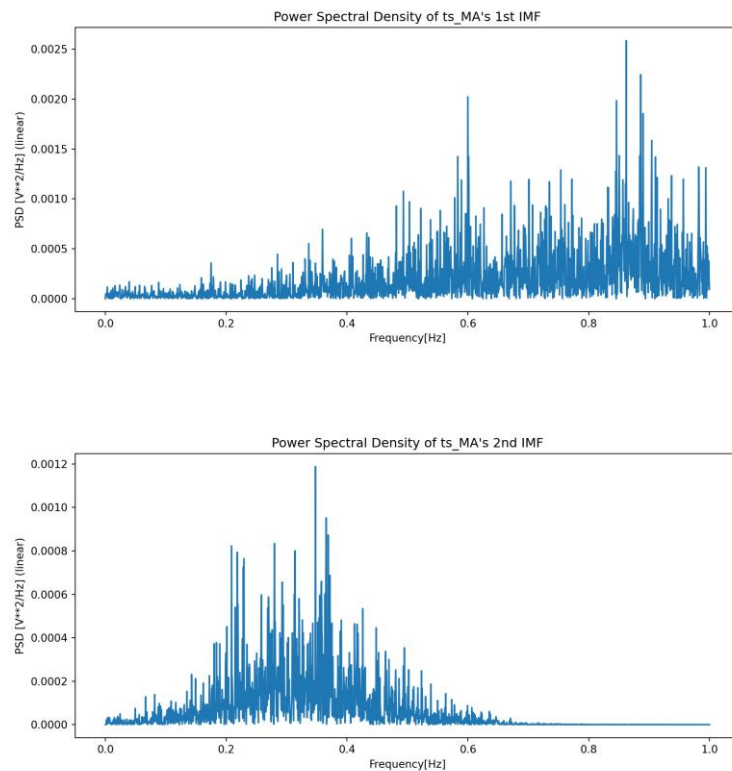
**Fig. 6.4.** Hurst Exponent of IMFs\_MSFT



word 可编辑

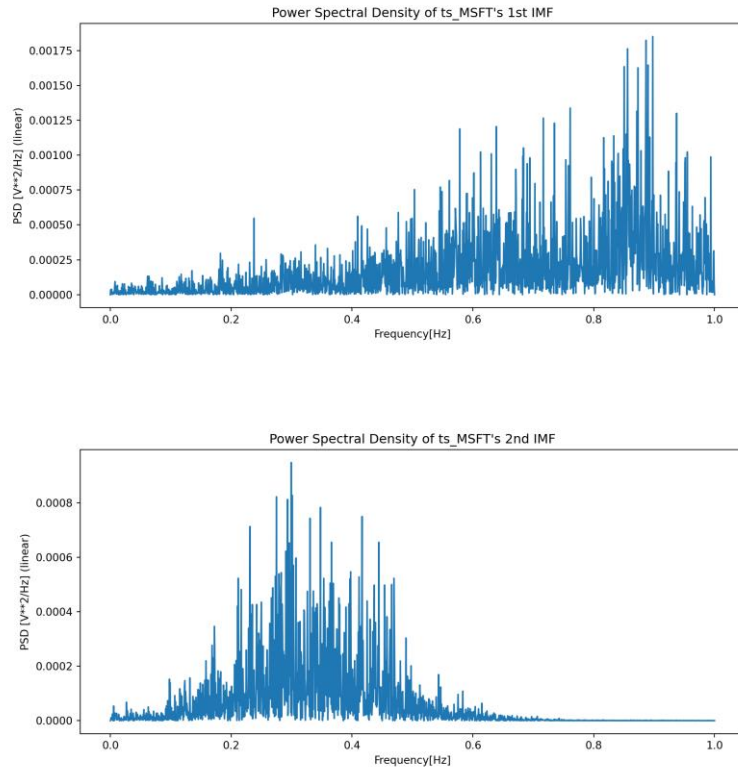
Both show linear gradual downward trend with the IMF orders growing.

Now consider the first two IMFs. Plot their PSD against frequency:



**Fig. 6.5.** Power Spectral Density of MA's first 2 IMFs

1st IMF's PSD concentrates on low frequency range; 2nd IMF's PSD concentrates on high frequency range.

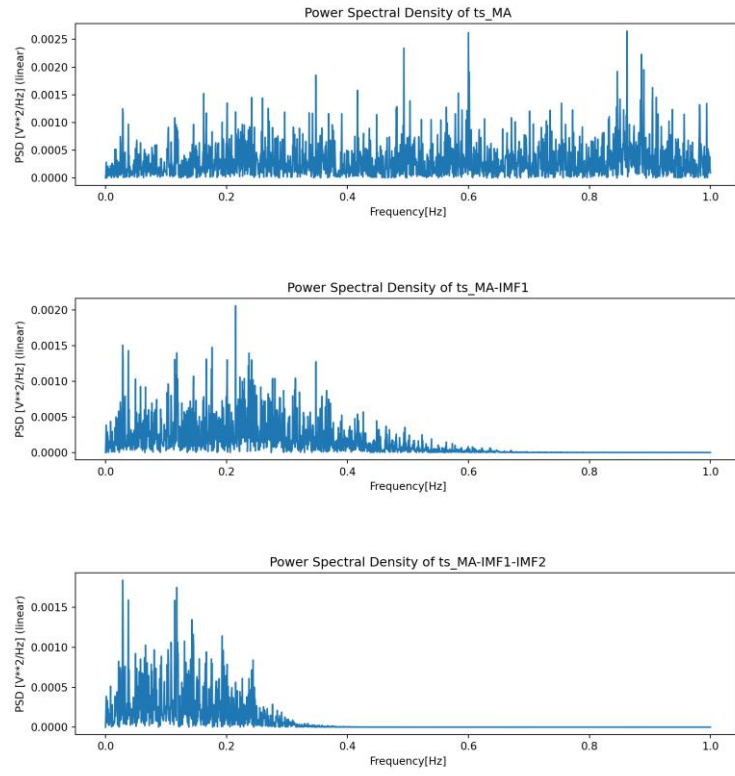


**Fig. 6.6.** Power Spectral Density of MSFT's first 2 IMFs

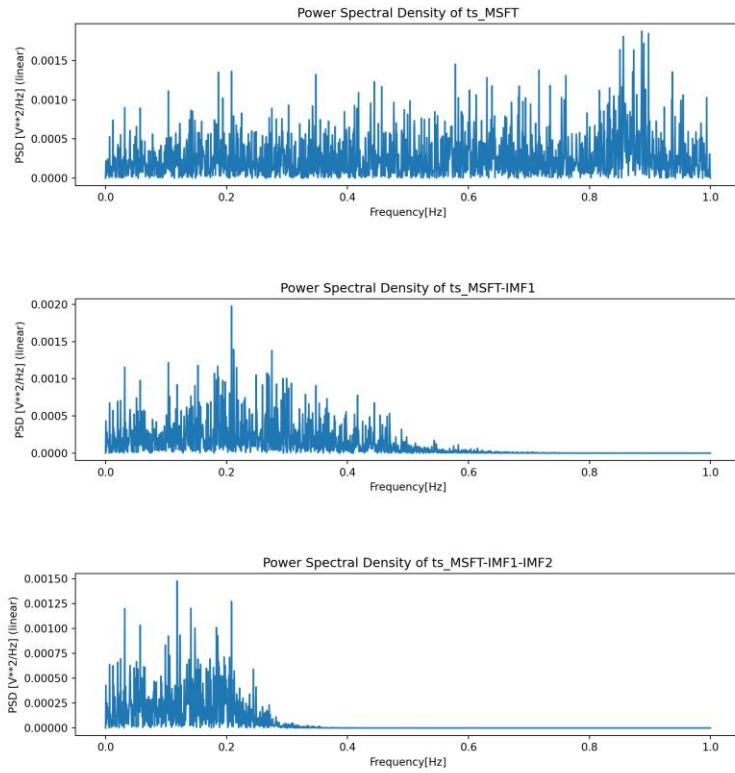
Also, 1st IMF's PSD concentrates on low frequency range; 2nd IMF's PSD concentrates on high frequency range.

Finally, let us plot the PSDs of X-c1 and X-c1-c2 against frequency compare the spectra of them to that of X.

word 可编辑



**Fig. 6.7.** PSD of MA, MA-IMF1 and MA-IMF1-IMF2



**Fig. 6.8.** PSD of MSFT, MSFT-IMF1 and MSFT-IMF1-IMF2

We can observe that the higher order IMFs minused by original signal, the lower frequency range remaining signal concentrates at.