

Modelação Numérica 2017

Aula 5, 1/Mar

- Projecto 1
- Preparação dos dados.
- Tendências.
- Espectro

<http://modnum.ucs.ciencias.ulisboa.pt>

Modelação Numérica 2017

Aula 6, 7/Mar

- FFT
- Convolução
- Correlação

<http://modnum.ucs.ciencias.ulisboa.pt>

Projecto 1.

Análise de séries temporais. Análise espectral. Filtragem.

1. Dados de pressão no fundo do oceano na região do Alasca.
2. Sismo em Samatra (2012.04.11) observado no Alasca.
3. Ruído oceânico observado por um sismómetro em Portugal num dia de ondulação intensa.
4. Radiação solar medida no campus da FCUL.
5. Evolução da temperatura e da concentração atmosférica de CO₂ em bolhas de gelo em Vostok.
6. Clima simulado em Lisboa (1989-2012).
7. Evolução do clima num cenário de aquecimento global.
8. Vento e potência num parque eólico.

Projecto 1.

Método

- **Inspecionar** os dados; representá-los graficamente. Se necessários, fazer zoom dos dados em várias janelas.
- Verificar se os dados foram **amostrados regularmente** e a que passo. Identificar eventuais erros ou falhas na amostragem. Verificar e, se necessário, **corrigir as unidades** utilizadas.
- **Corrigir erros**, eliminando-os. Se necessário **interpolarmos** os dados para recuperar ou construir uma **base regular de amostragem**.
- Tendo em conta a amostragem e comprimento da série, calcular a **frequência mínima e máxima** dos processos que podemos estudar com as séries em questão.
- **Calcular o espectro de amplitude** dos dados. Se for conveniente, pode-se truncar a dimensão da série a um número favorável para a FFT (e.g. na forma 2^n , ...).
- **Identificar tendências** nos dados, i.e. variações cujo período aparente é superior à duração do sinal. Se existir uma tendência eliminá-la por regressão linear ou por média móvel de longo período.

Projecto 1.

Método

- Identificar ciclos dominantes nos dados (anual, diurno, etc) e desenhar filtros capazes de os remover.
- Analisar os dados filtrados: inspeção da série no domínio do tempo e cálculo do novo espectro.
- Se for apropriado, calcular o espectro em janelas parcelares, no sentido de identificar oscilações transientes, ex: um tsunami ou um sismo.
- No caso de séries temporais relacionadas, calcular a correlação entre os sinais. Calcular as diferenças de fase.
- Interpretar.
- Discutir.
- Inovar...

Input de dados em Python.

```
129 # elevation:_FillValue : -999.999
130 # elevation:standard_name : elevation
131 # elevation:long_name : surface_elevation_in_meters_above_sea_level
132 # elevation:units : m
133 # elevation:comment : Surface elevation in meters above sea level. See provider_comment if available.
134 # intake_height:_FillValue : -999.999
135 # intake_height:long_name : sample_intake_height_in_meters_above_ground_level
136 # intake_height:units : m
137 # intake_height:comment : Sample intake height in meters above ground level (magl). See provider_comm
138 # qcflag:_FillValue : NA
139 # qcflag:long_name : quality_control_flag
140 # qcflag:comment : This quality control flag is provided by the contributing PIs. See provider_commen
141 # qcflag:provider_comment : This is the NOAA 3-character quality control flag. Column 1 is the REJEC
142 #
143 # VARIABLE ORDER
144 #
145 # site_code year month day hour minute second value value_unc nvalue latitude longitude altitude elev
146 MLO 1974 1 1 0 0 0 -999.99 -99.99 0 19.536 -155.576 3437.0 3397.0 40.0 NA *..
147 MLO 1974 2 1 0 0 0 -999.99 -99.99 0 19.536 -155.576 3437.0 3397.0 40.0 NA *..
148 MLO 1974 3 1 0 0 0 -999.99 -99.99 0 19.536 -155.576 3437.0 3397.0 40.0 NA *..
149 MLO 1974 4 1 0 0 0 -999.99 -99.99 0 19.536 -155.576 3437.0 3397.0 40.0 NA *..
150 MLO 1974 5 1 0 0 0 333.157 0.286 15 19.536 -155.576 3437.0 3397.0 40.0 NA ...
151 MLO 1974 6 1 0 0 0 332.059 0.317 27 19.536 -155.576 3437.0 3397.0 40.0 NA ...
152 MLO 1974 7 1 0 0 0 330.988 0.505 31 19.536 -155.576 3437.0 3397.0 40.0 NA ...
153 MLO 1974 8 1 0 0 0 329.165 0.659 31 19.536 -155.576 3437.0 3397.0 40.0 NA ...
154 MLO 1974 9 1 0 0 0 327.407 0.548 29 19.536 -155.576 3437.0 3397.0 40.0 NA ...
155 MLO 1974 10 1 0 0 0 327.201 0.314 29 19.536 -155.576 3437.0 3397.0 40.0 NA ...
156 MLO 1974 11 1 0 0 0 328.325 0.401 29 19.536 -155.576 3437.0 3397.0 40.0 NA ...
157 MLO 1974 12 1 0 0 0 329.501 0.463 31 19.536 -155.576 3437.0 3397.0 40.0 NA ...
158 MLO 1975 1 1 0 0 0 -999.99 0.491 31 19.536 -155.576 3437.0 3397.0 40.0 NA ...
159 MLO 1975 2 1 0 0 0 -999.99 0.407 28 19.536 -155.576 3437.0 3397.0 40.0 NA ...
160 MLO 1975 3 1 0 0 0 -999.99 0.243 26 19.536 -155.576 3437.0 3397.0 40.0 NA ...
161 MLO 1975 4 1 0 0 0 -999.99 0.472 30 19.536 -155.576 3437.0 3397.0 40.0 NA ...
162 MLO 1975 5 1 0 0 0 333.912 0.353 30 19.536 -155.576 3437.0 3397.0 40.0 NA ...
163 MLO 1975 6 1 0 0 0 333.398 0.434 29 19.536 -155.576 3437.0 3397.0 40.0 NA ...
164 MLO 1975 7 1 0 0 0 331.783 0.801 27 19.536 -155.576 3437.0 3397.0 40.0 NA ...
165 MLO 1975 8 1 0 0 0 329.877 0.946 31 19.536 -155.576 3437.0 3397.0 40.0 NA ...
```

Input de dados em Python.

```
import matplotlib.pyplot as plt
import numpy as np
import datetime

# %% Importar dados

fin='0_co2_mlo_surface-insitu_1_ccgg_MonthlyData-susana.txt'

data = np.loadtxt(fin, usecols=range(1,8), skiprows=145)

datet=[datetime.datetime(int(data[i,0]), int(data[i,1]), int(data[i,2]),
                        int(data[i,3]), int(data[i,4]), int(data[i,5]))
                        for i in range(len(data))]
juld=[dt.timetuple().tm_yday for dt in datet]
date = [data[i,0] + float(juld[i])/365. for i in range(len(datet))]
date = np.array(date)

co2=data[:,6]

# %% Plot

plt.plot(datet, co2)
plt.grid()
plt.xlabel('Time (years)')
plt.ylabel('CO2')
plt.title('CO2 mensal em Mauna Loa (Hawaii)')
```

Input de dados em Python.

```
import matplotlib.pyplot as plt
import numpy as np
import datetime

# %% Importar dados

fin='0_co2_mlo_surface-insitu_1_ccg

data = np.loadtxt(fin, usecols=rang

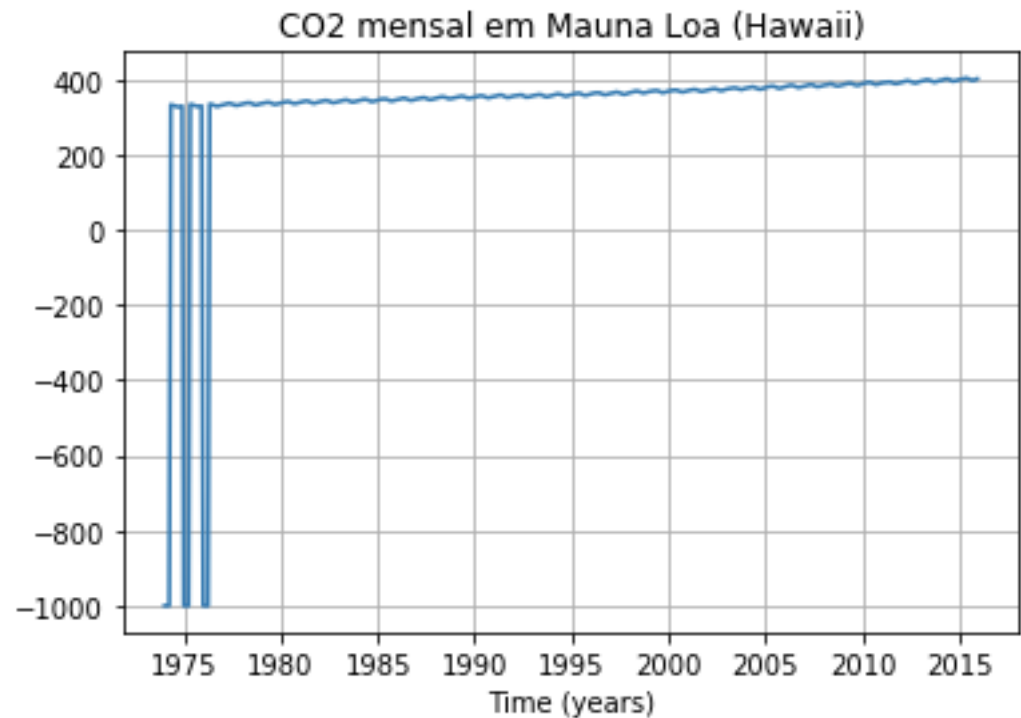
datet=[datetime.datetime(int(data[i
                                int(data[i,
                                for i in ra

juld=[dt.timetuple().tm_yday for dt
date = [data[i,0] + float(juld[i])/
date = np.array(date)

co2=data[:,6]

# %% Plot

plt.plot(datet, co2)
plt.grid()
plt.xlabel('Time (years)')
plt.ylabel('CO2')
plt.title('CO2 mensal em Mauna Loa (Hawaii)')
```



Input de dados em Python.

```
import matplotlib.pyplot as plt
import numpy as np
import datetime

# %% Importar dados

fin='0_co2_mlo_surface-insitu_1_ccg

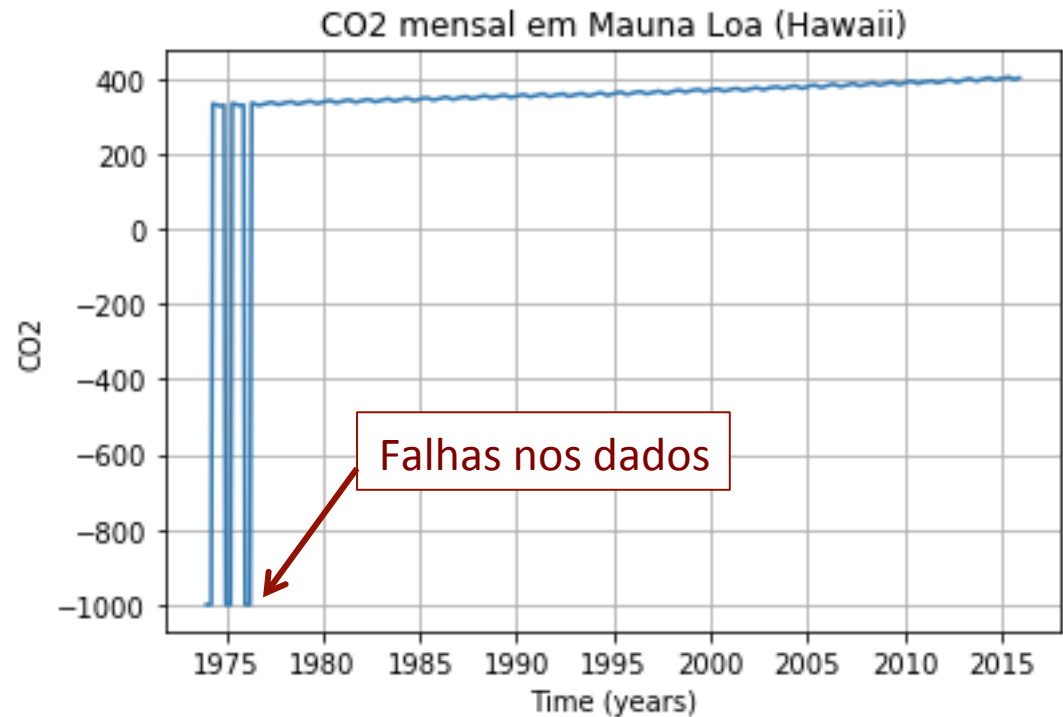
data = np.loadtxt(fin, usecols=rang

datet=[datetime.datetime(int(data[i
                                int(data[i,
                                for i in ra
juld=[dt.timetuple().tm_yday for dt
date = [data[i,0] + float(juld[i])/
date = np.array(date)

co2=data[:,6]

# %% Plot

plt.plot(datet, co2)
plt.grid()
plt.xlabel('Time (years)')
plt.ylabel('CO2')
plt.title('CO2 mensal em Mauna Loa (Hawaii)')
```



Input de dados em Python.

```
import matplotlib.pyplot as plt
import numpy as np
import datetime

# %% Importar dados

fin='0_co2_mlo_surface-insitu_1_ccg

data = np.loadtxt(fin, usecols=rang

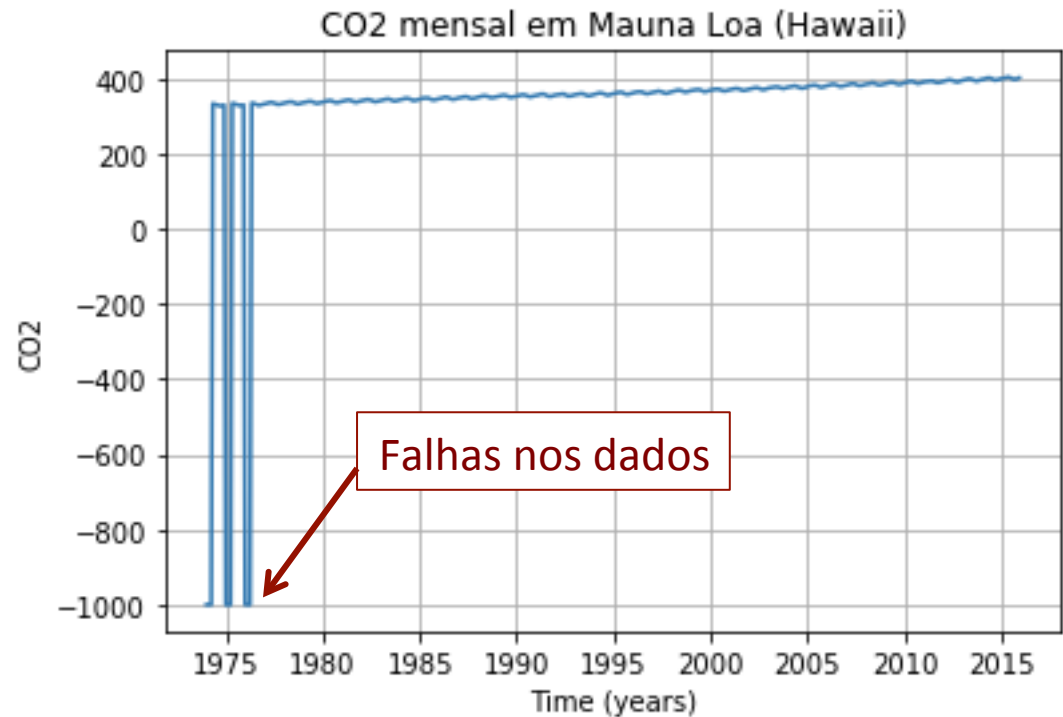
datet=[datetime.datetime(int(data[i
                                int(data[i,
                                for i in ra

juld=[dt.timetuple().tm_yday for dt
date = [data[i,0] + float(juld[i])/
date = np.array(date)

co2=data[:,6]

# %% Plot

plt.plot(datet, co2)
plt.grid()
plt.xlabel('Time (years)')
plt.ylabel('CO2')
plt.title('CO2 mensal em Mauna Loa (Hawaii)')
```



Dados regulares
(1/mês)

Input de dados em Python.

```
129 # elevation:_FillValue : -999.999
130 # elevation:standard_name : elevation
131 # elevation:long_name : surface_elevation_in_meters_above_sea_level
132 # elevation:units : m
133 # elevation:comment : Surface elevation in meters above sea level. See provider_comment if available.
134 # intake_height:_FillValue : -999.999
135 # intake_height:long_name : sample_intake_height_in_meters_above_ground_level
136 # intake_height:units : m
137 # intake_height:comment : Sample intake height in meters above ground level (magl). See provider_comm
138 # qcflag:_FillValue : NA
139 # qcflag:long_name : quality_control_flag
140 # qcflag:comment : This quality control flag is provided by the contributing PIs. See provider_commen
141 # qcflag:provider_comment : This is the NOAA 3-character quality control flag. Column 1 is the REJEC
142 #
143 # VARIABLE ORDER
144 #
145 # site_code year month day hour minute second value value_unc nvalue latitude longitude altitude elev
146 MLO 1974 1 1 0 0 0 -999.99 -99.99 0 19.536 -155.576 3437.0 3397.0 40.0 NA *..
147 MLO 1974 2 1 0 0 0 -999.99 -99.99 0 19.536 -155.576 3437.0 3397.0 40.0 NA *..
148 MLO 1974 3 1 0 0 0 -999.99 -99.99 0 19.536 -155.576 3437.0 3397.0 40.0 NA *..
149 MLO 1974 4 1 0 0 0 -999.99 -99.99 0 19.536 -155.576 3437.0 3397.0 40.0 NA *..
150 MLO 1974 5 1 0 0 0 333.157 0.286 15 19.536 -155.576 3437.0 3397.0 40.0 NA ...
151 MLO 1974 6 1 0 0 0 332.059 0.317 27 19.536 -155.576 3437.0 3397.0 40.0 NA ...
152 MLO 1974 7 1 0 0 0 330.988 0.505 31 19.536 -155.576 3437.0 3397.0 40.0 NA ...
153 MLO 1974 8 1 0 0 0 329.165 0.659 31 19.536 -155.576 3437.0 3397.0 40.0 NA ...
154 MLO 1974 9 1 0 0 0 327.407 0.548 29 19.536 -155.576 3437.0 3397.0 40.0 NA ...
155 MLO 1974 10 1 0 0 0 327.201 0.314 29 19.536 -155.576 3437.0 3397.0 40.0 NA ...
156 MLO 1974 11 1 0 0 0 328.325 0.401 29 19.536 -155.576 3437.0 3397.0 40.0 NA ...
157 MLO 1974 12 1 0 0 0 329.501 0.463 31 19.536 -155.576 3437.0 3397.0 40.0 NA ...
158 MLO 1975 1 1 0 0 0 -999.99 0.491 31 19.536 -155.576 3437.0 3397.0 40.0 NA ...
159 MLO 1975 2 1 0 0 0 -999.99 0.407 28 19.536 -155.576 3437.0 3397.0 40.0 NA ...
160 MLO 1975 3 1 0 0 0 -999.99 0.243 26 19.536 -155.576 3437.0 3397.0 40.0 NA ...
161 MLO 1975 4 1 0 0 0 -999.99 0.472 30 19.536 -155.576 3437.0 3397.0 40.0 NA ...
162 MLO 1975 5 1 0 0 0 333.912 0.353 30 19.536 -155.576 3437.0 3397.0 40.0 NA ...
163 MLO 1975 6 1 0 0 0 333.398 0.434 29 19.536 -155.576 3437.0 3397.0 40.0 NA ...
164 MLO 1975 7 1 0 0 0 331.783 0.801 27 19.536 -155.576 3437.0 3397.0 40.0 NA ...
165 MLO 1975 8 1 0 0 0 329.877 0.946 31 19.536 -155.576 3437.0 3397.0 40.0 NA ...
```

Eliminar erros, interpolar

```
### Verifica número de valores inválidos
### Constrói vector de valores válidos

datev=np.array([]); co2v=np.array([])      # matrizes de valores válidos
inval=0

for i in range(len(co2)):
    if co2[i] < 0.:
        inval = inval+1
    else:
        datev=np.append(datev,date[i])
        co2v=np.append(co2v,co2[i])

print(inval)

### Interpola

datei=date
co2i=np.interp(datei, datev, co2v, left=None, right=None, period=None)
```

Eliminar erros, interpolar

```
#%% Interpola
```

```
datei=date  
co2i=np.interp(datei, datev, co2v, left=None, right=None, period=None)
```

```
#%% Plot
```

```
plt.close()
```

```
plt.subplot(2,1,1)  
plt.plot(date, co2, 'r')  
plt.plot(datei, co2i, '--b')  
plt.title('Mole fraction of CO2 in dry air')
```

```
plt.subplot(2,1,2)  
plt.plot(datei, co2i, 'b')  
plt.grid()  
plt.xlabel('Year')
```

```
plt.tight_layout()
```

Eliminar erros, interpolar

```
### Interpola
```

```
datei=date  
co2i=np.interp(datei, datev, co2v, left=None, right=None, period=None)
```

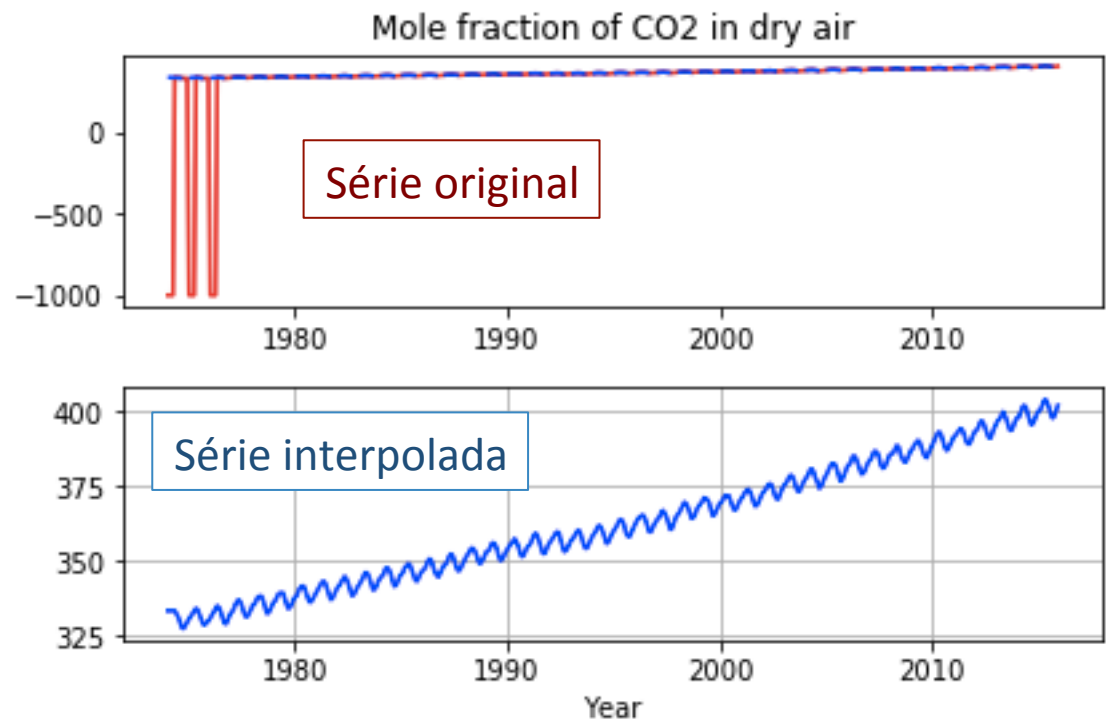
```
### Plot
```

```
plt.close()
```

```
plt.subplot(2,1,1)  
plt.plot(date, co2, 'r')  
plt.plot(datei, co2i, '--b')  
plt.title('Mole fraction of CO2
```

```
plt.subplot(2,1,2)  
plt.plot(datei, co2i, 'b')  
plt.grid()  
plt.xlabel('Year')
```

```
plt.tight_layout()
```



Espectro sem processamento

```
## Calcular espectro
import numpy.fft as fft

dt=1.                # espaçamento (em meses)
N=len(co2i)          # comprimento do vector de dados
N2=N/2               # metade do comprimento do vector de dados
fNyq=1./(2.*dt);    # frequência de Nyquist
df=1./(dt*N);       # espaçamento do espectro
freq=np.arange(0, fNyq, df); # vector de frequências
Fout=fft.fft(co2i); # calcular o espectro com FFT
FoutA=np.abs(Fout[:N2])/N2; # amplitudes do espectro

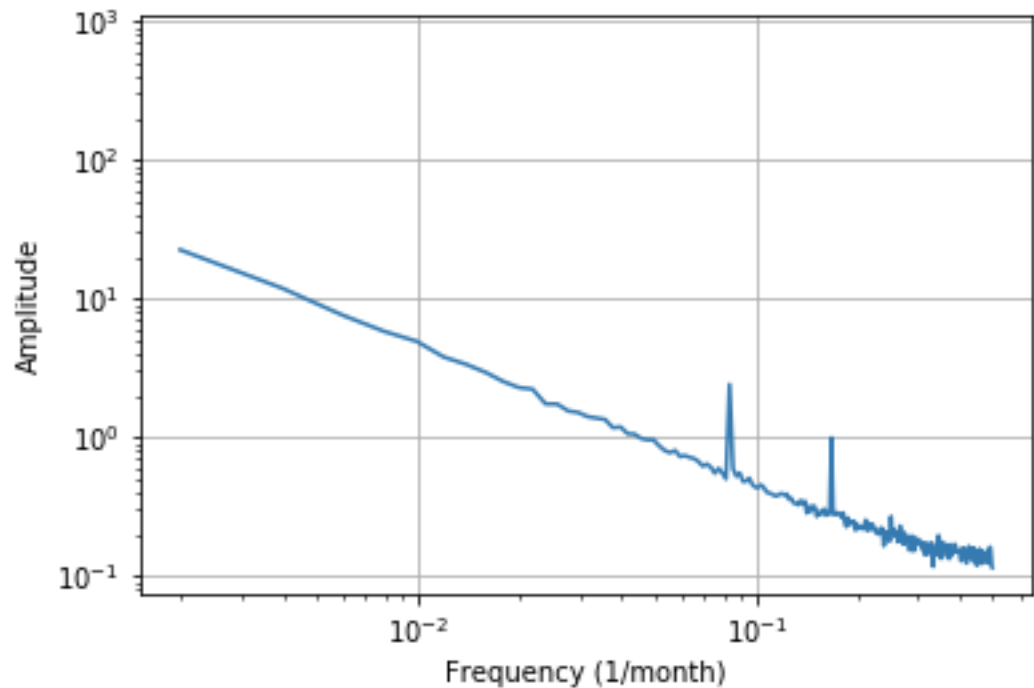
plt.close()
plt.loglog(freq, FoutA);
plt.axis('tight')
plt.grid()
plt.xlabel('Frequency (1/month)')
plt.ylabel('Amplitude')
```

Espectro sem processamento

```
## Calcular espectro
import numpy.fft as fft

dt=1.                                # espaçamento (em meses)
N=len(co2i)                           # comprimento do vector de dados
N2=N/2                                # metade do comprimento do vector de dados
fNyq=1./(2.*dt);                      # frequência de Nyquist
df=1./(dt*N);                         # espaçamento do espectro
freq=np.arange(0, fNyq, df);          # vector de frequências
Fout=fft.fft(co2i);                  # calcular o espectro com FFT
FoutA=np.abs(Fout[:N2])/N2;          # amplitude

plt.close()
plt.loglog(freq, FoutA);
plt.axis('tight')
plt.grid()
plt.xlabel('Frequency (1/month)')
plt.ylabel('Amplitude')
```

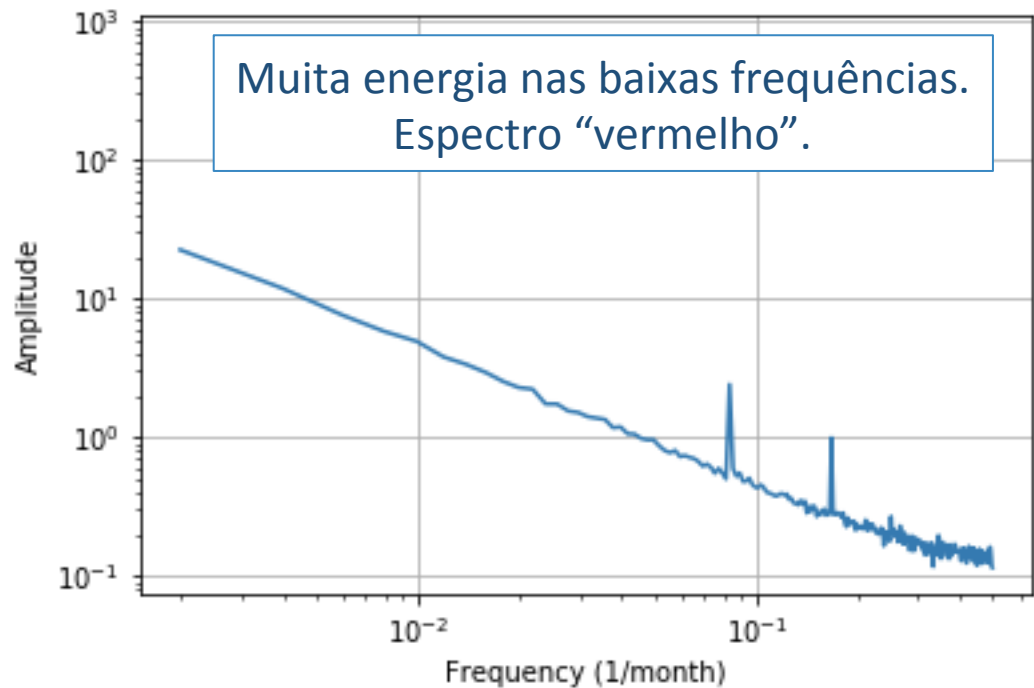


Espectro sem processamento

```
## Calcular espectro
import numpy.fft as fft

dt=1. # espaçamento (em meses)
N=len(co2i) # comprimento do vector de dados
N2=N/2 # metade do comprimento do vector de dados
fNyq=1./(2.*dt); # frequência de Nyquist
df=1./(dt*N); # espaçamento do espectro
freq=np.arange(0, fNyq, df); # vector de frequências
Fout=fft.fft(co2i); # calcular o espectro com FFT
FoutA=np.abs(Fout[:N2])/N2; # amplitude

plt.close()
plt.loglog(freq, FoutA);
plt.axis('tight')
plt.grid()
plt.xlabel('Frequency (1/month)')
plt.ylabel('Amplitude')
```

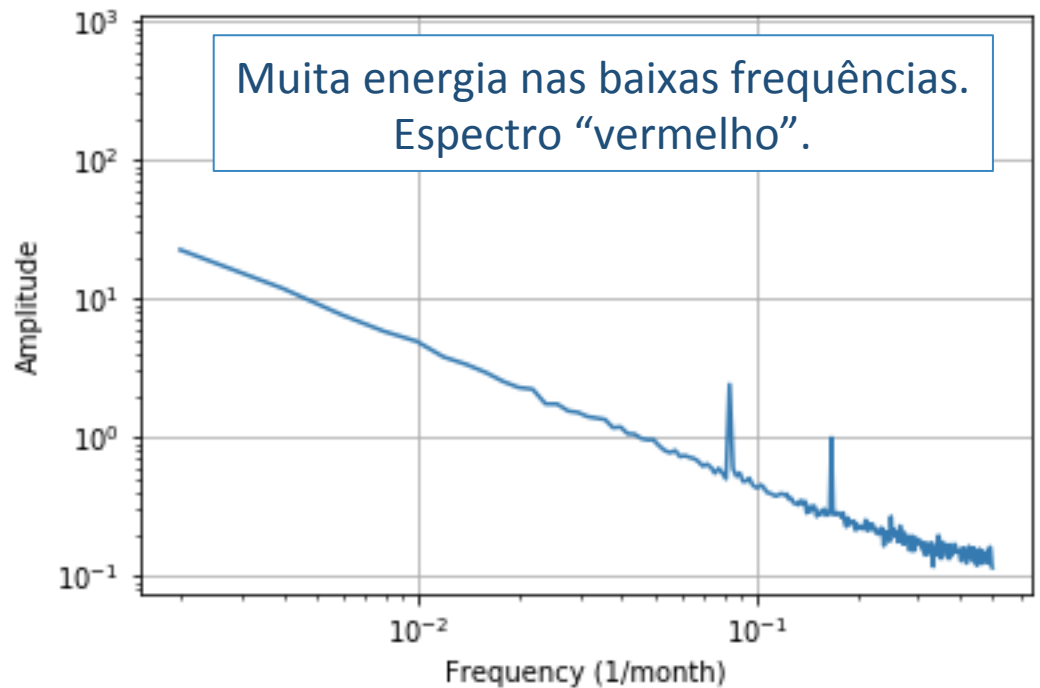
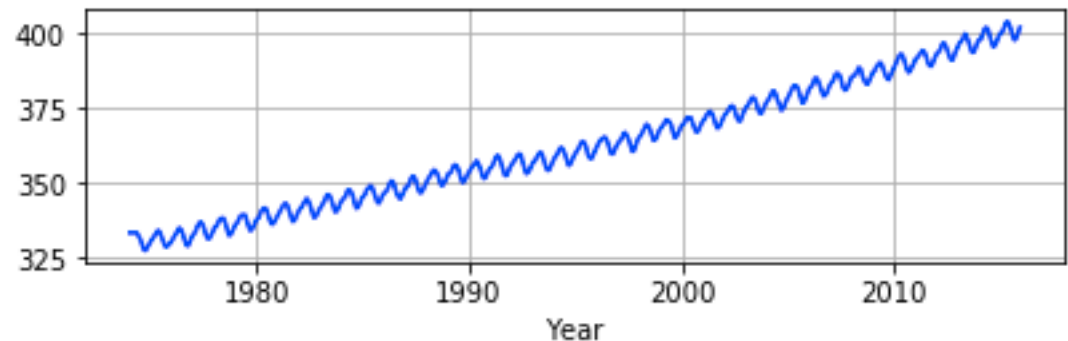


Espectro sem processamento

```
%% Calcular espectro
import numpy.fft as fft

dt=1.
N=len(co2i)
N2=N/2
fNyq=1./(2.*dt);
df=1./(dt*N);
freq=np.arange(0, fNyq, df);
Fout=fft.fft(co2i);
FoutA=np.abs(Fout[:N2])/N2; # an

plt.close()
plt.loglog(freq, FoutA);
plt.axis('tight')
plt.grid()
plt.xlabel('Frequency (1/month)')
plt.ylabel('Amplitude')
```



Remover a tendência

```
## Remove a tendência: ajuste a uma recta
```

```
P = np.polyfit(datei, co2i, 1)      # ajusta os dados a uma recta  
co2trend=P[0]                     # declive da recta  
co20=P[1]                         # ordenada na origem da recta
```

```
co2fit=co2trend*datei + co20      # recta que ajusta os dados  
#co2fit=P[0]*datei + P[1]       # recta que ajusta os dados
```

```
plt.close()  
plt.plot(datei, co2i, 'b')  
plt.plot(datei, co2fit, 'red')  
plt.grid()  
plt.xlabel('Year')
```

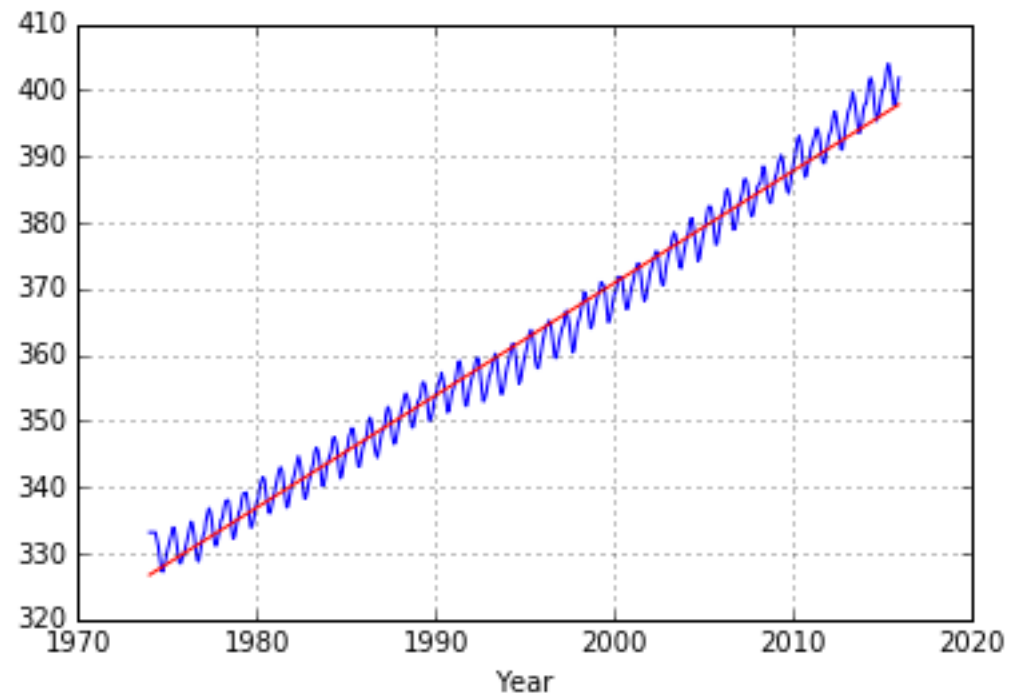
Remover a tendência

```
## Remover a tendência: ajuste a uma recta
```

```
P = np.polyfit(datei, co2i, 1) # ajusta os dados a uma recta  
co2trend=P[0] # declive da recta  
co20=P[1] # ordenada na origem da recta
```

```
co2fit=co2trend*datei + co20 # recta que ajusta os dados  
# co2fit=P[0]*datei + P[1] # recta que ajusta os dados
```

```
plt.close()  
plt.plot(datei, co2i, 'b')  
plt.plot(datei, co2fit, 'red')  
plt.grid()  
plt.xlabel('Year')
```



Remover a tendência

```
### Remover a tendência: subtração da tendência
```

```
co2_detrend = co2i - co2fit
```

```
plt.close()  
plt.subplot(3,1,1)  
plt.plot(datei, co2_detrend, 'b')  
plt.xlabel('Year')  
plt.grid()
```

```
Fout=fft.fft(co2_detrend);           # calcular o espectro com FFT  
FoutA=np.abs(Fout[:N2])/N2;         # amplitudes do espectro
```

```
plt.subplot(3,1,2)  
plt.loglog(freq,FoutA);  
plt.xlabel('Frequency (1/month)')  
plt.grid(); plt.axis('tight')  
plt.ylim([1e-3,10])
```

```
plt.subplot(3,1,3)  
plt.loglog(1./freq,FoutA);  
plt.xlabel('Period (months)')  
plt.grid(); plt.axis('tight')
```

```
plt.tight_layout()
```

Remover a tendência

Remover a tendência: subtração da tendência

```
co2_detrend = co2i - co2fit
```

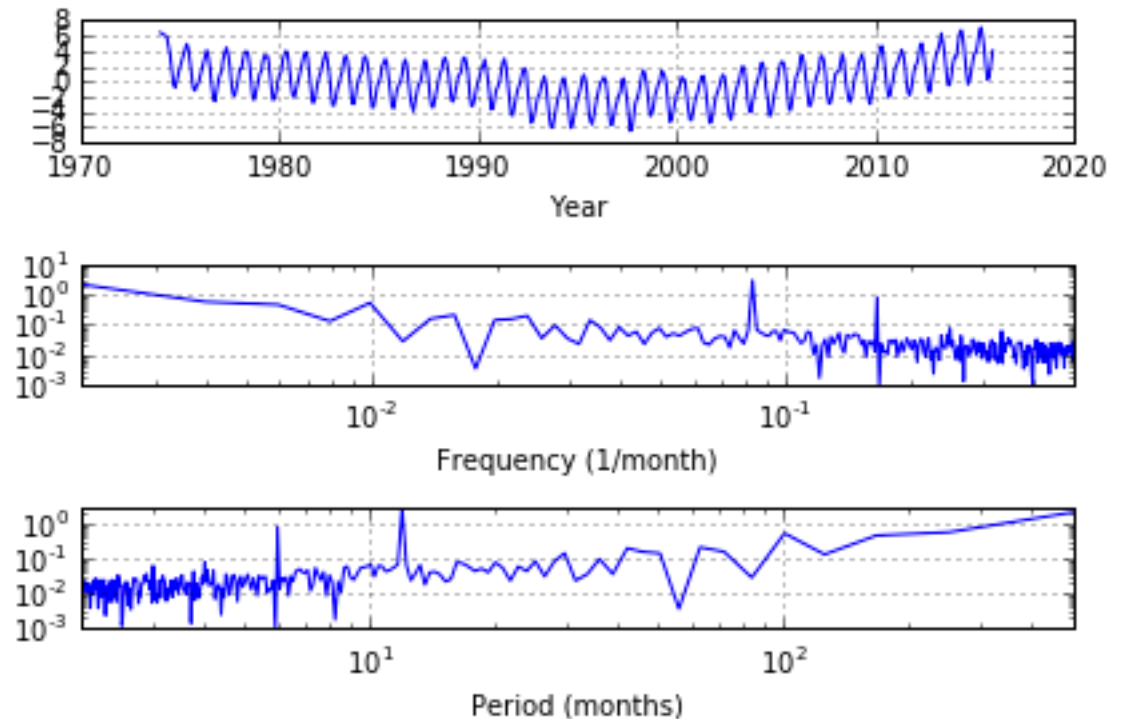
```
plt.close()  
plt.subplot(3,1,1)  
plt.plot(datei, co2_detrend, 'b')  
plt.xlabel('Year')  
plt.grid()
```

```
Fout=fft.fft(co2_detrend);  
FoutA=np.abs(Fout[:N2])/N2;
```

```
plt.subplot(3,1,2)  
plt.loglog(freq,FoutA);  
plt.xlabel('Frequency (1/month)');  
plt.grid(); plt.axis('tight')  
plt.ylim([1e-3,10])
```

```
plt.subplot(3,1,3)  
plt.loglog(1./freq,FoutA);  
plt.xlabel('Period (months)');  
plt.grid(); plt.axis('tight')
```

```
plt.tight_layout()
```



Remover a tendência

Remover a tendência: subtração da tendência

```
co2_detrend = co2i - co2fit
```

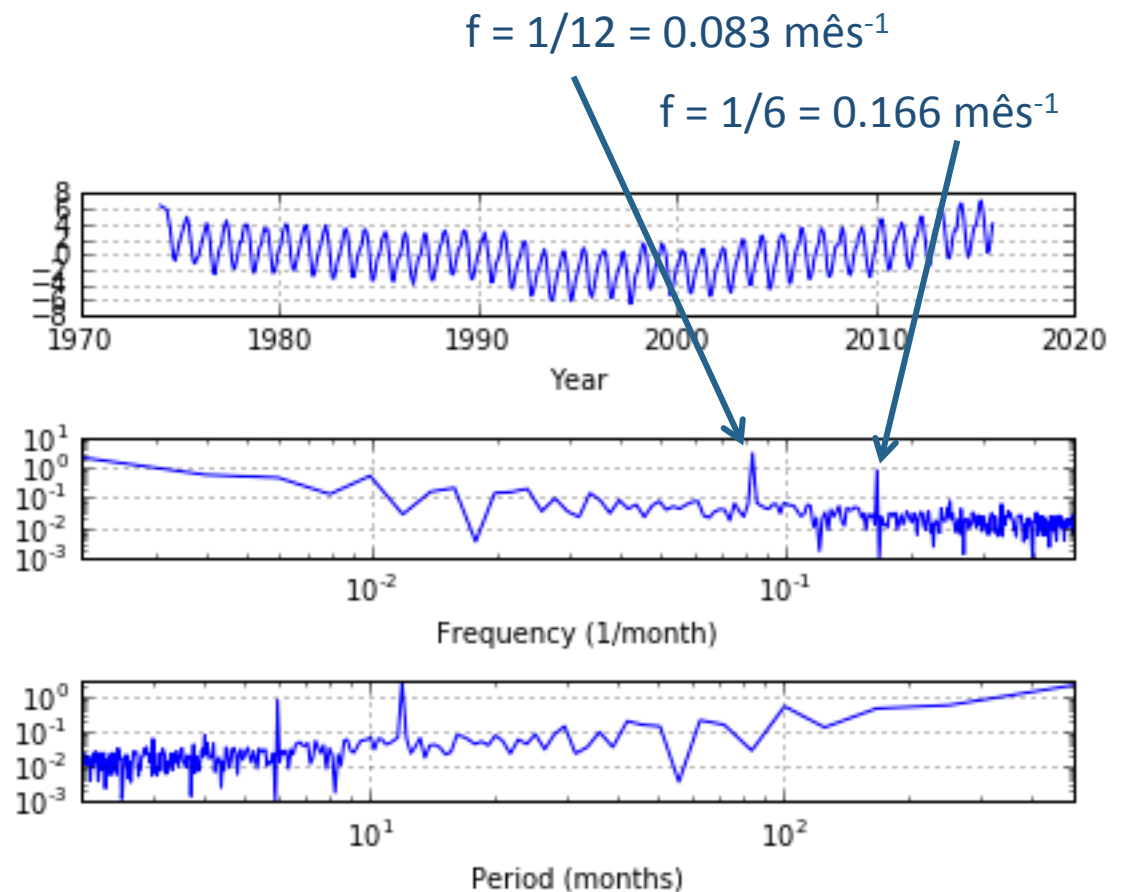
```
plt.close()
plt.subplot(3,1,1)
plt.plot(datei, co2_detrend, 'b')
plt.xlabel('Year')
plt.grid()
```

```
Fout=fft.fft(co2_detrend);
FoutA=np.abs(Fout[:N2])/N2;
```

```
plt.subplot(3,1,2)
plt.loglog(freq,FoutA);
plt.xlabel('Frequency (1/month)');
plt.grid(); plt.axis('tight')
plt.ylim([1e-3,10])
```

```
plt.subplot(3,1,3)
plt.loglog(1./freq,FoutA);
plt.xlabel('Period (months)');
plt.grid(); plt.axis('tight')
```

```
plt.tight_layout()
```



Remover a tendência

%% Remover a tendência: subtração da tendência

```
co2_detrend = co2i - co2fit
```

```
plt.close()
plt.subplot(3,1,1)
plt.plot(datei, co2_detrend, 'b')
plt.xlabel('Year')
plt.grid()
```

```
Fout=fft.fft(co2_detrend);
FoutA=np.abs(Fout[:N2])/N2;
```

```
plt.subplot(3,1,2)
plt.loglog(freq,FoutA);
plt.xlabel('Frequency (1/month)');
plt.grid(); plt.axis('tight')
plt.ylim([1e-3,10])
```

```
plt.subplot(3,1,3)
plt.loglog(1./freq,FoutA);
plt.xlabel('Period (months)');
plt.grid(); plt.axis('tight')
```

```
plt.tight_layout()
```

