



Ciências
ULisboa

Modelação Numérica

Aula 19

Exercícios

Exercício 1

O vetor T , contém observações horárias de temperatura durante 10 anos, sem falhas, num dado local. Diga como procederia para (indique as linhas de código apropriadas em cada caso):

- (a) Calcular e representar graficamente o seu espectro de amplitude, incluindo as legendas e escalas adequadas;
- (b) Estimar a amplitude do seu ciclo diurno;
- (c) Estimar a amplitude do seu ciclo anual;
- (d) Estimar a sua tendência.

Dados sintéticos

```
import numpy as np
import matplotlib.pyplot as plt
import datetime
plt.close('all'); datas=[]
data=datetime.datetime(2010,1,1)
datas.append(data)
onehour=datetime.timedelta(hours=1)
year=data.year
while year<2021:
    data=data+onehour
    year=data.year
    datas.append(data)
datas=np.array(datas)
N=len(datas)
dt=1/24
t=np.arange(0,N*dt,dt)
T=10*np.sin(2*np.pi*t/365.25)+12*np.sin(2*np.pi*t/1)
N2=(N+1)//2;
```

Espectro e picos

```
fT=np.fft.fft(T); #transformada de Fourier
ampT=np.abs(fT[0:N2+1]/N2); fNyq=1/(2*dt);df=fNyq/N2;
freq=np.arange(0,fNyq+df,df)

kdiurnal=np.argwhere(freq>1)[0]-1
kannual=np.argwhere(freq>1/365.25)[0]-1

fig,ax=plt.subplots(nrows=1)
ax.plot(freq,ampT); #gráfico
ax.scatter(freq[kdiurnal],ampT[kdiurnal])
ax.scatter(freq[kannual],ampT[kannual])

ax.set_xlabel(r'$f$ (dia-1)$');
ax.set_ylabel('Amplitude (K)');
ax.set_xscale('log')

print(ampT[kdiurnal],ampT[kannual])
```

Exercício 2

A difusão de uma nuvem de poluente na ausência de vento pode representar-se pela equação de difusão:

$$\frac{\partial \chi}{\partial t} = -D \left(\frac{\partial^2 \chi}{\partial x^2} + \frac{\partial^2 \chi}{\partial y^2} + \frac{\partial^2 \chi}{\partial z^2} \right)$$

- (a) Discretize essa equação utilizando um esquema implícito;
- (b) Esquematize o algoritmo de solução, indicando o código essencial;

Discretize essa equação utilizando um esquema implícito;

O esquema mais simples consiste em usar diferenças avançadas no tempo para a derivada temporal e diferenças centradas para a segunda derivada espacial.

Como se trata de um problema 3D+tempo, o campo χ será discretizado numa malha com 4 dimensões (3 espaciais e 1 temporal).

Designamos por i, j, k os índices espaciais nas direções x, y, z , respetivamente, e por n o índice temporal. O intervalo de discretização será $\Delta x, \Delta y, \Delta z, \Delta t$.

Assim a discretização será

$$\chi(x, y, z, t) \sim \chi_{i,j,k}^n$$

Derivada temporal

$$\frac{\partial \chi}{\partial t} \sim \frac{\chi_{i,j,k}^{n+1} - \chi_{i,j,k}^n}{\Delta t}$$

Derivada espacial implícita

$$\left(\frac{\partial^2 \chi}{\partial x^2} \right) \sim \frac{\chi_{i+1,j,k}^{n+1} + \chi_{i-1,j,k}^{n+1} - 2\chi_{i,j,k}^{n+1}}{\Delta x^2}$$

(idêntico para as outras direções)

Notar que o esquema será implícito porque estamos a calcular a derivada especial no “futuro”

$$\frac{\chi_{i,j,k}^{n+1} - \chi_{i,j,k}^n}{\Delta t} = -D \left(\frac{\partial^2 \chi}{\partial x^2} + \frac{\partial^2 \chi}{\partial y^2} + \frac{\partial^2 \chi}{\partial z^2} \right)$$

$$+ D \left[\frac{\chi_{i+1,j,k}^{n+1} + \chi_{i-1,j,k}^{n+1} - 2\chi_{i,j,k}^{n+1}}{\Delta x^2} + \frac{\chi_{i,j+1,k}^{n+1} + \chi_{i,j-1,k}^{n+1} - 2\chi_{i,j,k}^{n+1}}{\Delta y^2} + \frac{\chi_{i,j,k+1}^{n+1} + \chi_{i,j,k-1}^{n+1} - 2\chi_{i,j,k}^{n+1}}{\Delta z^2} \right] = 0$$

O que corresponde a uma equação implícita para $\chi_{i,j,k}^{n+1}$, que se pode escrever na forma de um sistema de (MUITAS) equações lineares

$$\chi_{i,j,k}^{n+1} + \Delta t D \left[\frac{\chi_{i+1,j,k}^{n+1} + \chi_{i-1,j,k}^{n+1} - 2\chi_{i,j,k}^{n+1}}{\Delta x^2} + \frac{\chi_{i,j+1,k}^{n+1} + \chi_{i,j-1,k}^{n+1} - 2\chi_{i,j,k}^{n+1}}{\Delta y^2} + \frac{\chi_{i,j,k+1}^{n+1} + \chi_{i,j,k-1}^{n+1} - 2\chi_{i,j,k}^{n+1}}{\Delta z^2} \right] = \chi_{i,j,k}^n$$

Que se pode escrever na forma

$$M\vec{\chi}^{n+1} = \vec{\chi}^n$$

Onde $\vec{\chi}^n$ é um vector gigante $(nx \times ny \times nz)$, e M uma matriz enorme $[(nx \times ny \times nz) \times (nx \times ny \times nz)]$

Notar que o cálculo da solução no passo de tempo $n + 1$ é feito em simultâneo para todos os pontos.

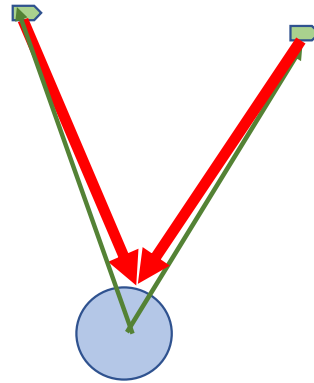
Localização 3D por GPS

Vamos esquecer a geometria esférica.

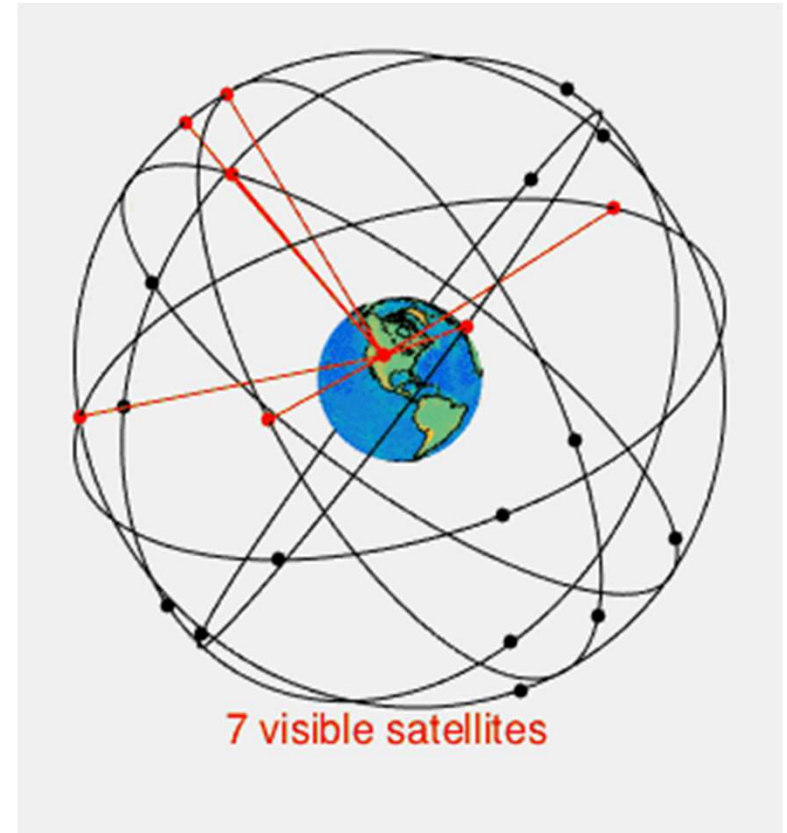
Vamos considerar 2 ângulos:
altura e azimute.

Altura: Ângulo com a horizontal

Azimute: Ângulo com o eixo dos
xx



$$t = \frac{\sqrt{x^2 + y^2 + z^2}}{c} + \varepsilon$$



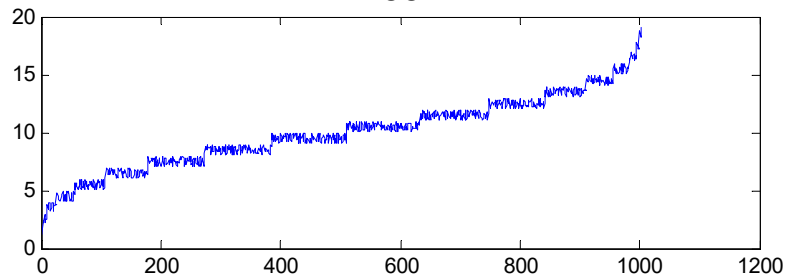
wikipedia

Função de custo para ajuste do histograma bin a bin

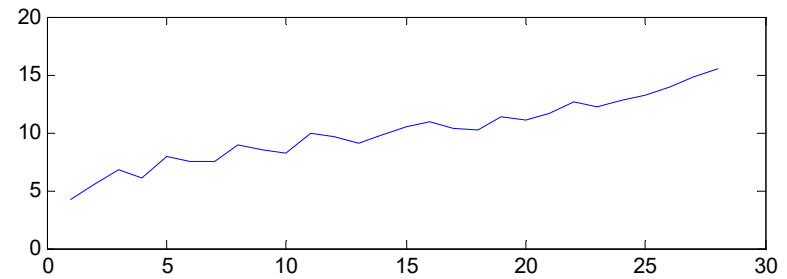
```
def iniCOST(tipo,dados)
    global tE wE wH nE
    shape=3.6;scale=11;threshold=0;nE=10000;tE=[0.5:49.5];nbin=length(tE);
    wB=nearest(weibull(shape,scale,threshold,tE)*nE);wB=wB'; %N° amostras
    nE=sum(wB);wE=zeros(nE,1);kE=0; %correção nE (nearest)
    for ibin=1:nbin
        New=wB(ibin);
        wE(kE+1:kE+New)=tE(ibin)+rand(New,1)-0.5;
        kE=kE+New;
    wH=hist(wE,tE); %Histograma sintético
end
function custo=cost(V)
global tE wE wH nE
custo=mean((weibull(shape,scale,threshold,tE)*nE-wH).^2);
end
function w=weibull(shape,scale,threshold,t)
w=real(shape/scale*((t-threshold)/scale).^(shape-1)...
    .*exp(-((t-threshold)/scale).^shape));
end
```

Sensibilidade à dimensão da amostra

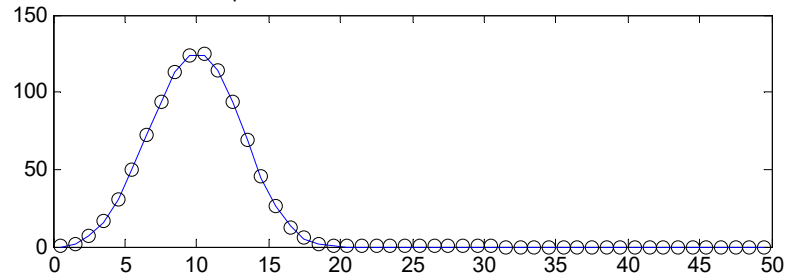
1002



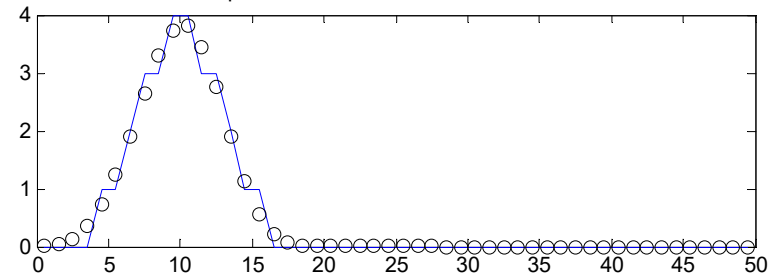
28



shape=3.5968 scale=10.9996 th=0 nE=1002



shape=3.9303 scale=10.9547 th=0 nE=28



Ajuste por máxima verosimilhança (*maximum likelihood*)

Escolher a solução mais provável (o conjunto de parâmetros a que tem maior probabilidade de produzir a amostra)

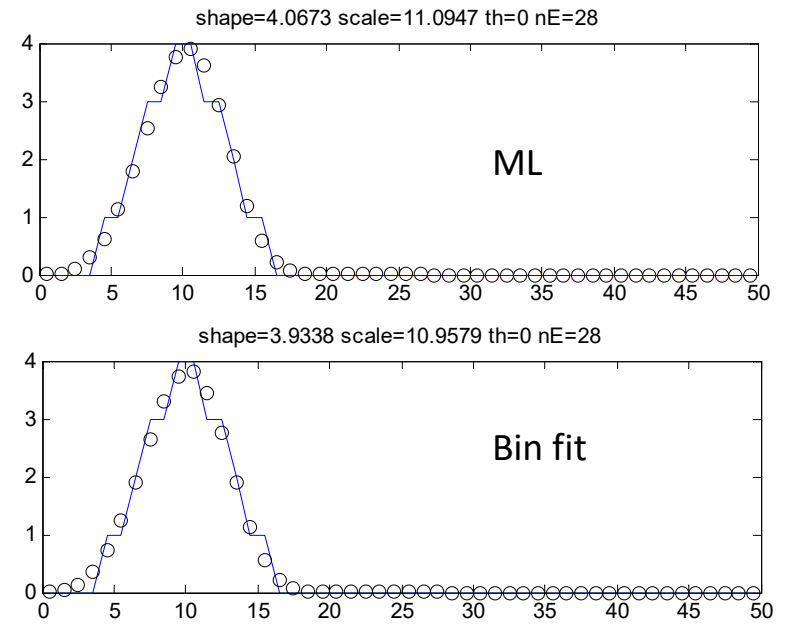
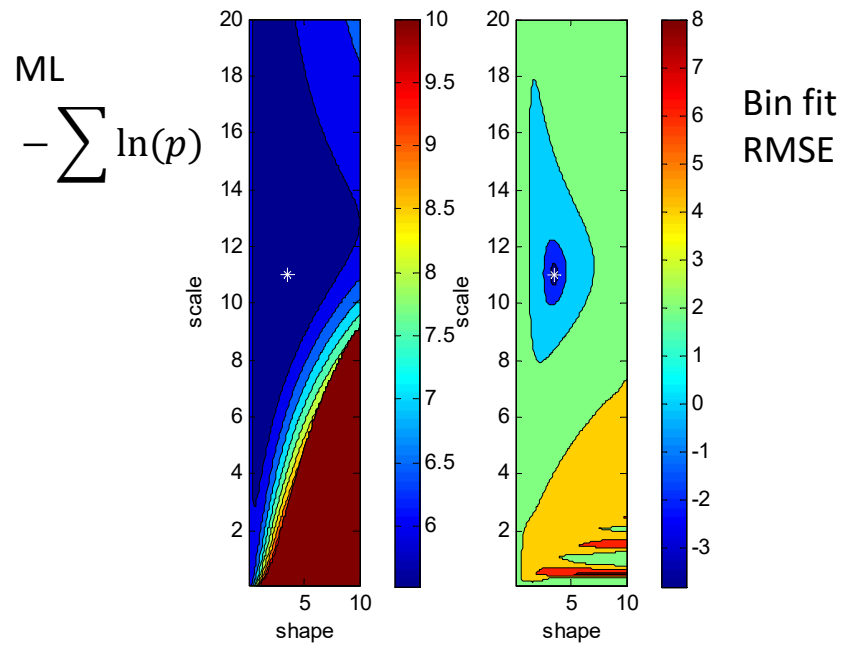
Maximizar: $J(\lambda, k, x) = \prod p(\lambda, k|x) \Rightarrow$

$$\text{Minimizar: } (-\ln(J)) = -\sum \ln(p(x))$$

```
function custo=cost(V)
global tE wE wH nE
ww=weibull(shape, scale, threshold, wE);
custo=-sum(log(ww));
end
```

wE é a amostra, não o cálculo por bin (1000 termos e não 50)

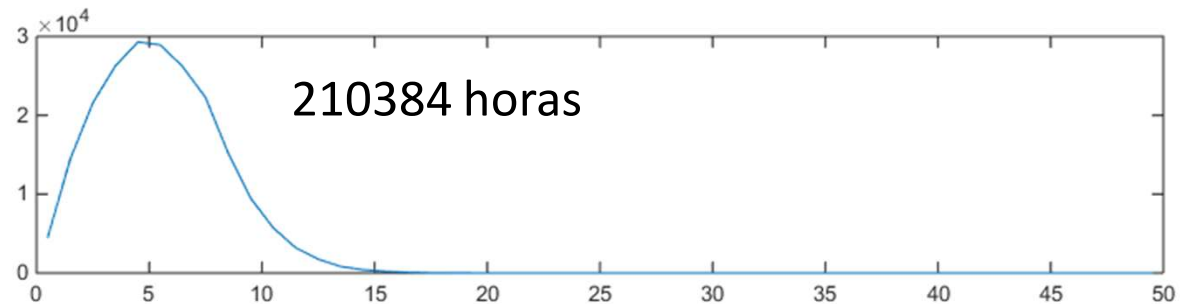
Comparação ML vs bin fit



Estima de k , λ , τ : por ajuste a mínimos quadrados do histograma

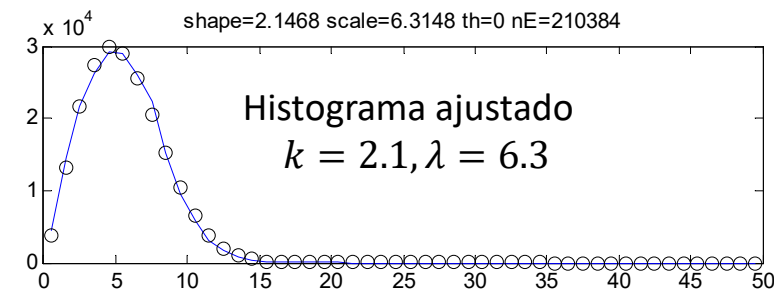
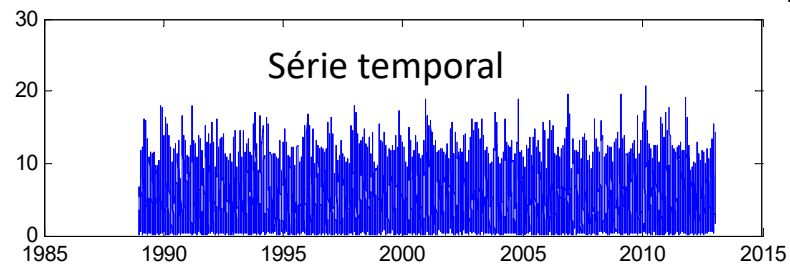
Vamos considerar uma **amostra** realista de dados de vento constituída por observações horárias durante 20 anos.

Vamos admitir que conhecemos o **domínio** do histograma: entre 0 e uma velocidade máxima, possivelmente superior à velocidade máxima registada na amostra. O nosso objetivo é calcular a curva de Weibull que melhor se ajusta ao histograma da amostra. Vamos usar o método de optimização já descrito a 3 dimensões.

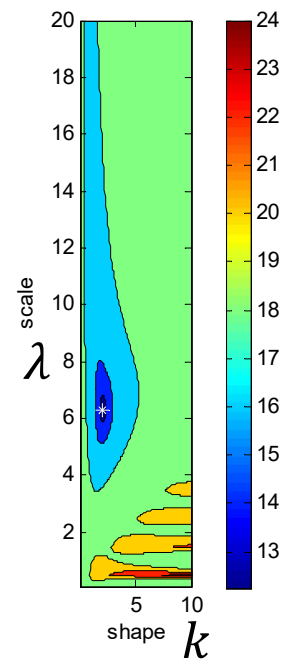


Ajuste de dados “reais”

```
function iniCOST(tipo,dados)
global tE wE wH nE %measurements
global windE windB binW
wrfdata=load('wrf_9km_IGIDL.dat'...
    ,'-ascii');
windE=sqrt(wrfdata(:,3).^2...
    +wrfdata(:,4).^2);
nE=length(windE);
binW=[0.5:49.5]'; %bins
windB=hist(windE,binW)';
clear wrfdata;
end
function custo=cost(V)
global windE windB binW nE
shape=V(1);scale=V(2);threshold=V(3);
custo=mean((weibull(shape,scale,threshold,binW)*nE-windB).^2);
end
```

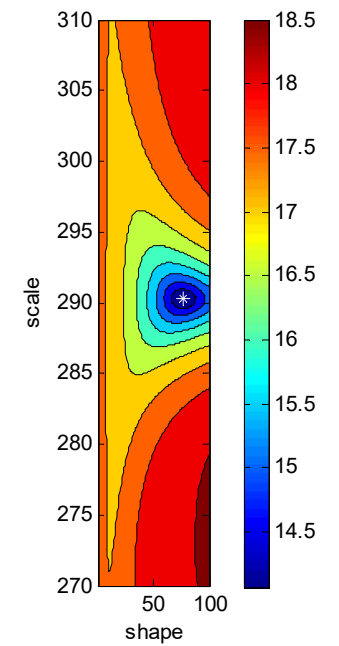
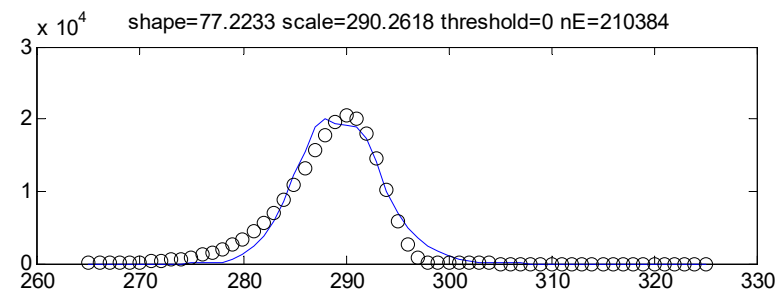
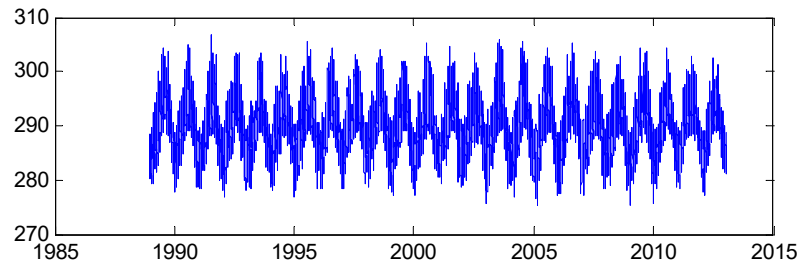


Função de custo



Dados de temperatura

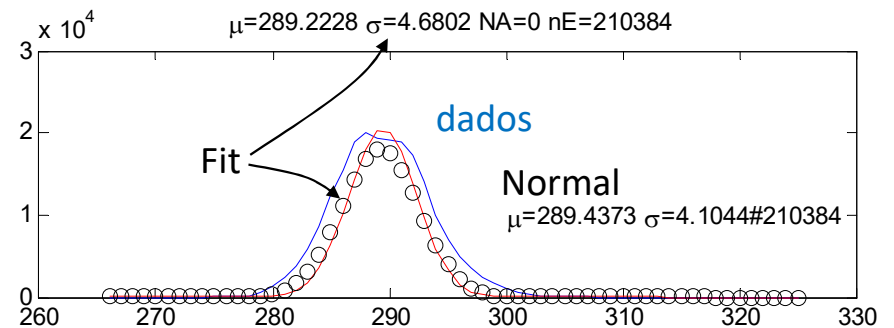
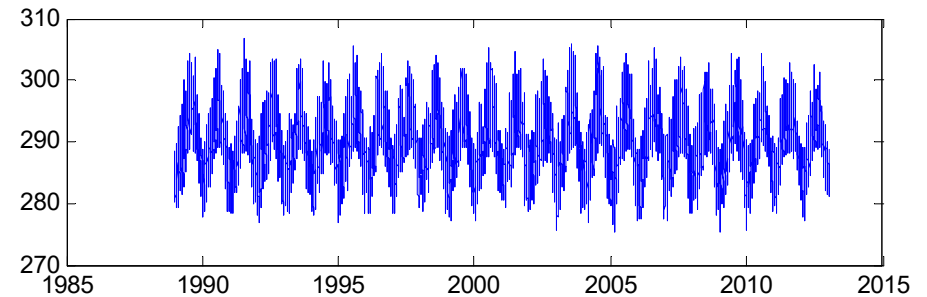
Ajuste de Weibull



Ajuste de distribuição normal

```
function custo=cost(V)
global tipo subtipo
global windE windB binW
global tE wE wH nE % measurements
mu=V(1);sigma=V(2);
custo=mean((normal(mu,sigma,binW)...
    *nE-windB).^2);
end
```

```
function gauss=normal(mu,sigma,x)
gauss=1/(sigma*sqrt(2*pi))*exp(-((x-mu)/sigma).^2);
end
```

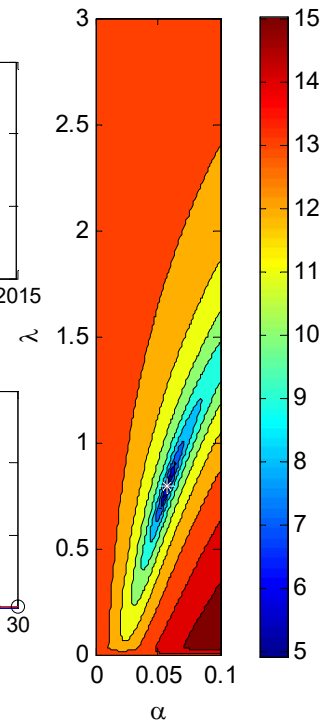
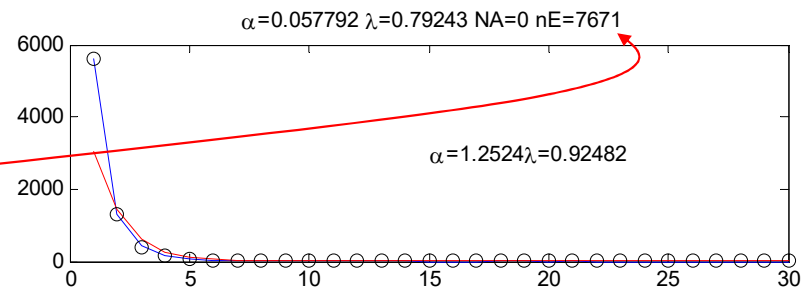
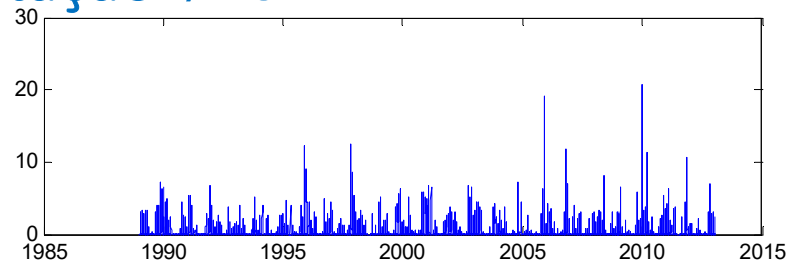


Ajuste de uma pdf Γ a precipitação $\neq 0$

```
function custo=cost(V)
global windE windB binW
global tE wE wH nE % measurements
alpha=V(1);lambda=V(2);
custo=mean((gamadens(alpha,...
    lambda,binW)*nE-windB).^2);
end
```

Deita-se fora o 1º bin

```
function gd=gamadens(alpha,lambda,x)
gd=lambda^alpha*x.^(alpha-1).*exp(-lambda*x)/gamma(alpha);
for ix=1:length(x)
    if x(ix)<=0; gd(ix)=0; end
end
end
```



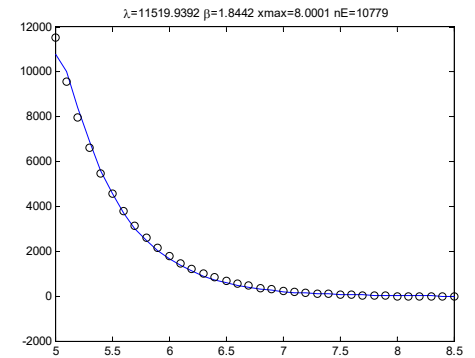
Optimização a n dimensões (1/2)

```
function [V,J,kHIT,kPERT]=annealing(vmin,vmax,Jmin,...
    minvstep,maxITER,maxPERT,COOL,kappa,V0,T,outITER)
ndim=length(vmin); vstep=2*(vmax-vmin);
if isnan(V0); V0=vmin+(vmax-vmin).*rand(ndim,1); end
V=V0;J=cost(V);
iTER=0;kHIT=1;kPERT=0;
while (iTER<maxITER && J>Jmin && vstep(1)>minvstep(1) ) || nHIT>0
    iP=0;
    while iP<maxPERT
        kPERT=kPERT+1;
        for idim=1:ndim
            rr=rand-0.5;
            VI(idim)=V(idim)+vstep(idim)*rr;
            while VI(idim)<vmin(idim) || VI(idim)>vmax(idim)
                rr=rand-0.5;
                VI(idim)=V(idim)+vstep(idim)*rr;
            end
        end
        end
        JI=cost(VI);
```

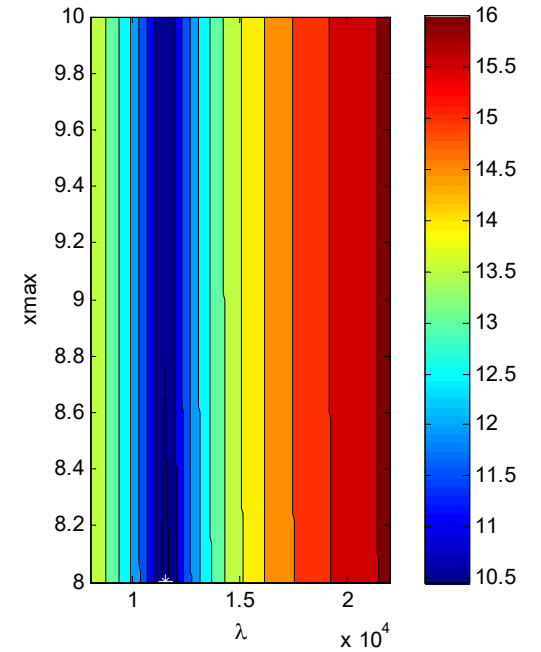
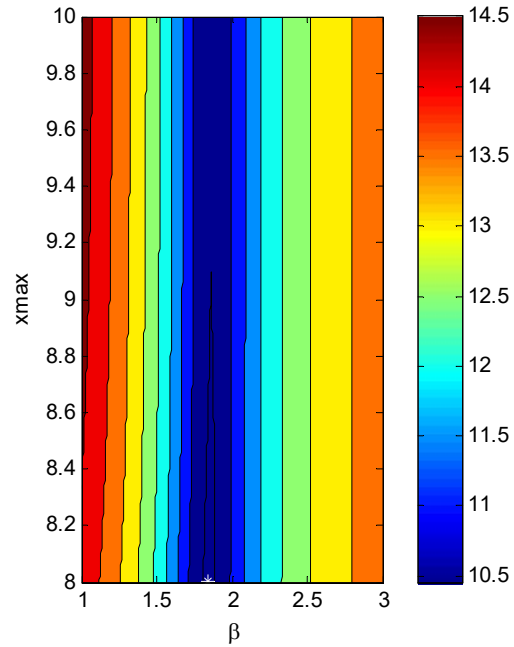
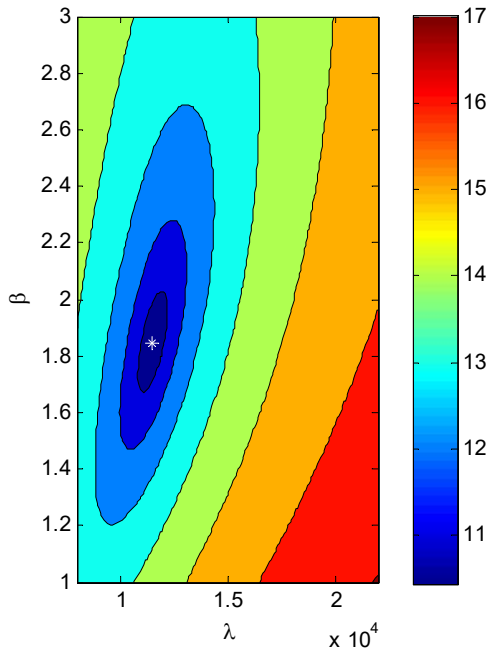
Optimização a n dimensões (2/2)

```
    JI=cost (VI) ;
    dJ=JI-J;
    if dJ<0
        J=JI;
        V=VI;
        kHIT=kHIT+1;
    end
    iP=iP+1;
end
iTER=iTER+1;
T=T*COOL;
vstep=max (minvstep, vstep*exp (-kappa/T) ) ;
end
end
```

Solução numérica: x_{Max} está mal condicionado!
 Apesar de o fit parecer excelente
 (mas não o semilogy...)



log(cost)



Refinando a função de custo

Os eventos de grande magnitude são raros mas contribuem muito para a Energia total dos sismos $\propto 10^{1.5 M}$. Isto sugere uma reformulação da função de custo.

Versão anterior:

```
custo=mean((guttrich(lambda,beta,xmin,xmax,grB))-(grF)).^2);
```

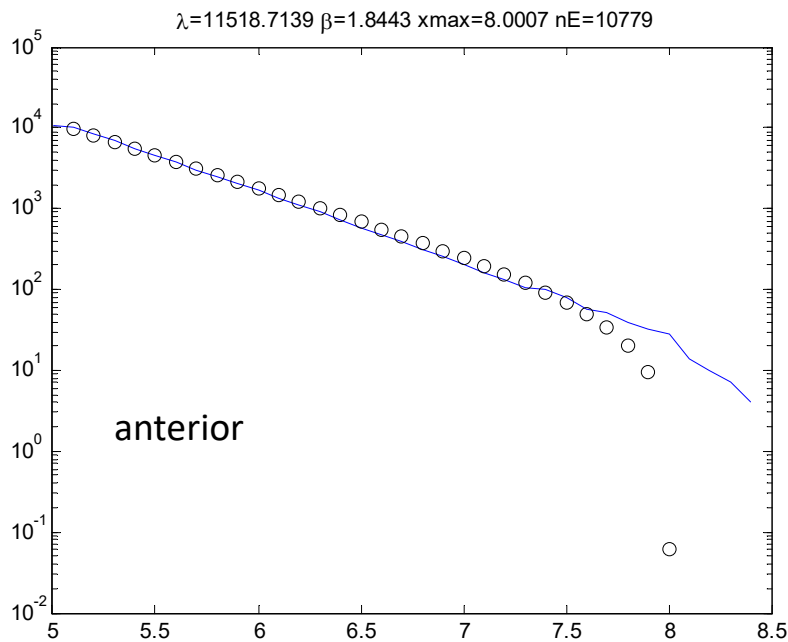
Nova versão

```
custo=mean((guttrich(lambda,beta,xmin,xmax,grB).*10.^grB-grF).*10.^grB).^2);
```

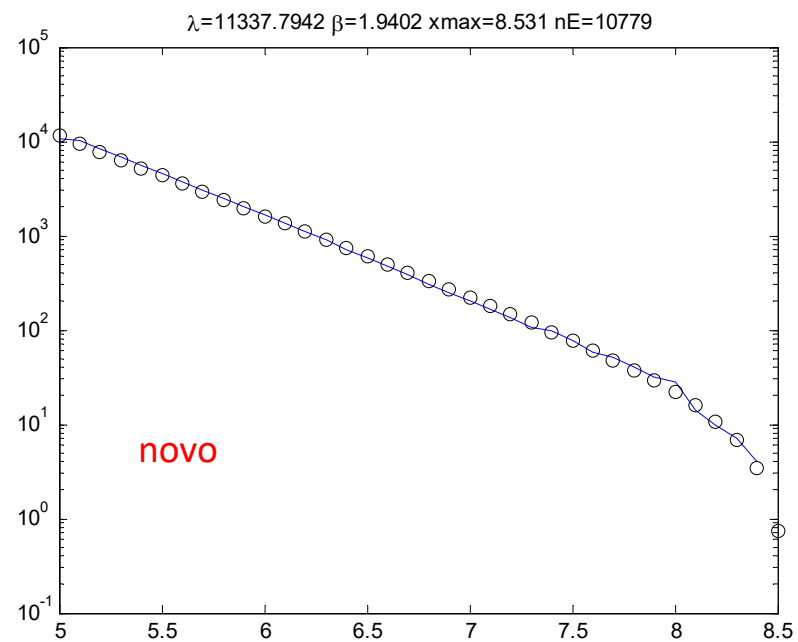
Notar que grB são os valores da magnitude no histograma (bins) variando entre 5 e 8.5

Nota: usou-se 10^M como ponderação e não $10^{1.5 M}$, para não exagerar o peso dos eventos de maior magnitude. Tentem com $10^{1.5 M}$.

Comparação entre os dois métodos



```
custo=mean((guttrich(lambda,beta  
,xmin,xmax,grB-grF).^2);
```



```
custo=mean((guttrich(lambda,beta,xmin,  
xmax,grB) .*10.^grB-grF .*10.^grB).^2);
```


Estrutura da nova função de custo (log(cost))

```
custo=mean((guttrich(lambda,beta,xmin,xmax,grB) .*10.^grB-grF.*10.^grB) .^2);
```

