# INSTITUTO TECNOLÓGICO DE AERONÁUTICA

# CCI-22 - Professor Victor Curtis


# Pedro Luchiari de Carvalho - Turma 4


# Relatório 6 - Ajuste de Curvas

## Questão 1

É proposta a linearização de uma curva exponencial. Para tal, considere que a função seja da forma y = alfa*exp(beta*x). Ao aplicarmos o logaritmo natural em ambos os membros da equação, obtemos log(y) = log(alfa) + beta*x. Sendo log(y) = z, log(alfa) = a0 e beta = a1, obtemos z = a0 + a1*x, que é uma equação de reta. Através de um algoritmo advindo do método dos mínimos quadrados, calculamos os coeficientes e plotamos os gráficos propostos.

```
format long
x=[1 2 3 4 5 6];
y=[1.487 2.958 5.602 8.003 11.452 13.021];
%Linearização:
z=log(y);
n=length(x);
soma_x=sum(x);
soma_z=sum(z);
soma_x_quad=sum(x.^2);
soma_xz=sum(x.*z);
x_med=soma_x/n;
z_med=soma_z/n;
%item a:
a1=(n*soma_xz-soma_x*soma_z)/(n*soma_x_quad-(soma_x)^2);
fprintf("a1 = %.4f\n", a1);
```

```
a1 = 0.4362
```

```
a0=z_med-a1*x_med;
fprintf("a0 = %.4f\n", a0);
```

```
a0 = 0.1882
```

```
%item b:
alfa=exp(a0);
beta=a1;
fprintf("alfa = %.4f\n", alfa);
```

```
alfa = 1.2070
```

```
fprintf("beta = %.4f\n", beta);
```

```
beta = 0.4362
```

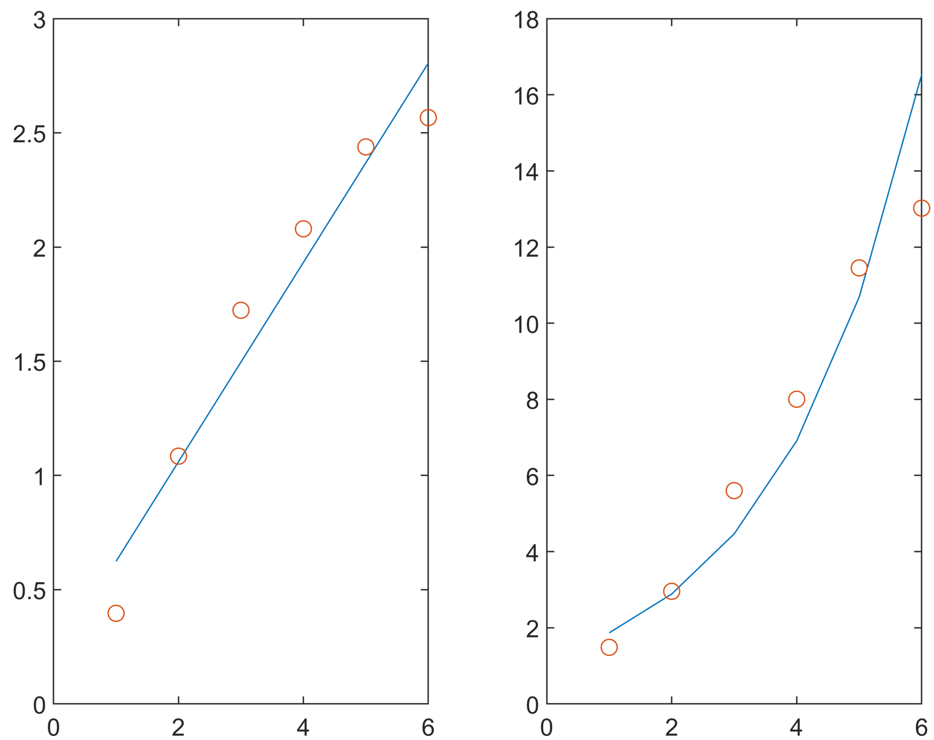
```
y1=a1*x+a0;
y2=alfa*exp(beta*x);
St=sum((z-z_med).^2);
Sr=sum((z-a0-a1*x).^2);
Sy=sqrt(St/(n-1));
Sy_x=sqrt(Sr/(n-2));
%item c:
Cd=(St-Sr)/St;
fprintf("Cd = %.8f\n", Cd);
```

```
Cd = 0.94683203
```

```
%item d:
r=sqrt(Cd);
fprintf("r = %.7f\n", r);
```

```
r = 0.9730529
```

```
%item e:
subplot(1,2,1);
plot(x,y1); %impressão da linearização
hold on
plot(x,z,'o');
hold off
subplot(1,2,2);
plot(x,y2); %impressão da função potência
hold on
plot(x,y,'o');
hold on
```




## Questão 2:

Novamente, o algoritmo dos mínimos quadrados é aplicado, mas agora o algoritmo nos leva a uma função polinomial de grau 2.

```
m=2;
n=length(x);
soma_x=sum(x);
soma_y=sum(y);
soma_x2=sum(x.^2);
soma_x3=sum(x.^3);
soma_x4=sum(x.^4);
soma_xy=sum(x.*y);
soma_x2y=sum((x.^2).*y);
x_med=soma_x/n;
y_med=soma_y/n;
Mc=[n soma_x soma_x2; soma_x soma_x2 soma_x3; soma_x2 soma_x3 soma_x4];
Mb=[soma_y; soma_xy; soma_x2y];
a=Mc\Mb; %resolução do Sistema Linear
y3=a(1,1)+a(2,1)*x+a(3,1)*x.^2;
%item a:
fprintf("a0 = %.4f\n", a(1,1));
```

```
a0 = -0.8498
```

```
fprintf("a1 = %.4f\n", a(2,1));
```

```
a1 = 1.9806
```

```
fprintf("a3 =%.4f\n", a(3,1));
```

```
a3 =0.0663
```

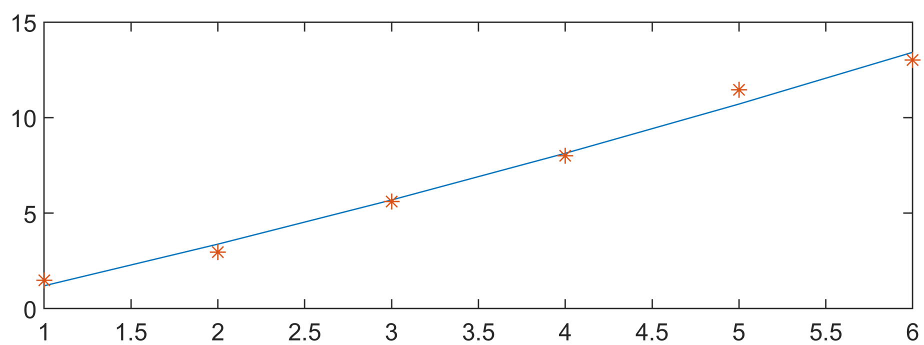
```


St=sum((y-y_med).^2);
Sr=sum((y-a(1,1)-a(2,1).*x-a(3,1).*x.^2).^2);
Sy_x=sqrt(Sr/(n-(m+1)));
%itens b e c:
Cd=(St-Sr)/St;
r=sqrt(Cd);
fprintf("Cd = %.4f\n", Cd);
```

```
Cd = 0.9906
```

```
fprintf("r = %.4f\n", r);
```

```
r = 0.9953
```

```
%item d:
subplot(2,1,1);
plot(x,y3);
hold on
plot(x,y,'*');
hold off
```



## Discussão:

Observa-se, a partir dos cálculos de Cd e r, um maior ajuste dos pontos dados por uma curva polinomial (parábola) do que através de uma curva exponencial. Em ambos os casos, entretanto, r tem valor superior a 95%, indicando relativa boa aproxiamação.