



INSTITUTO
TECNOLÓGICO
DE AERONÁUTICA

Introdução à Física
Experimental

Prof. Douglas Leite
www.lpp.ita.br/leite

Ajuste de modelos matemáticos não lineares em dados experimentais usando o Mathematica®

Aula 3

Vídeo 1 – Introdução e atividades da aula

Vídeo 2 – Tutorial: Vídeo-análise no Mathematica



Objetivos da Aula

Objetivo Principal

- Experimento de pêndulo amortecido lançado de altos ângulos e modelagem matemática sem aproximações*

Objetivos Secundários

- Video-Tracking com o Mathematica (Computer Vision)
- Estudo dos diferentes regimes de amortecimento em relação à velocidade do corpo (aerodinâmica)
- Entendimento dos limites das aproximações: “corpo pontual” e “baixos ângulos”

Desafios

- * Pêndulo simples ou pêndulo físico?



Atividades da aula

Assistir aos Tutoriais

- Vídeo 2 – Video-análise (tracking) com o Mathematica

Fazer Pesquisa Bibliográfica

- Pêndulo amortecido lançado de altos ângulo – façam sua própria pesquisa
- Artigo de F. J Arnold et al. (2011) – pêndulo simples amortecido - fornecido pelo professor
- Artigo de C. B. C. Silva et al. (2020) – pêndulo físico - fornecido pelo professor

Entregável

- Seguir roteiro do slide a seguir
- Entregar no Classroom até a próxima segunda-feira às 12hs:
 - Notebook Mathematica com o nome ***T#_G§§_aula3.nb*** (# = num turma, §§ = num grupo)
(Não esquecer de incluir CABEÇALHO NO NOTEBOOK COM OS NOMES E E-MAIL DOS INTEGRANTES DO GRUPO)
 - Vídeo do experimento com o nome ***T#_G§§_aula3.mp4*** (# = num turma, §§ = num grupo)
 - Arquivo TXT dos dados do rastreo com o nome ***T#_G§§_aula3.txt*** (# = num turma, §§ = num grupo)



Roteiro

Experimento:

1. Material necessário

- a) Linha 10 ou linha de pesca ou linha de costura
- b) Esfera leve com volume apreciável (pode ser bola de ping-pong, papel amassado, ou qualquer bola de plástico oca ou de isopor)
- c) Balança, régua e/ou trena e/ou paquímetro
- d) Câmera filmadora ou celular

2. Montagem do pêndulo

- a) Comprimento do pêndulo acima de 50 cm
- b) Cor da esfera deve ser homogênea (sem marcas ou manchas) e destoar da cor do fundo da filmagem
- c) Fazer marcação visível do ponto de fixação do pêndulo (o qual deve aparecer na filmagem)
- d) Afixar referência de escala para o vídeo (sugestão: usar algo maior que 50 cm para aumentar precisão)

3. Realização do experimento

- a) **IMPORTANTE:** usar tripé ou câmera fixa sem trepidação
- b) Iniciar a filmagem com o corpo em repouso para referência de vertical
- c) Todo o pêndulo e seu movimento deve ser enquadrado, incluindo o ponto de fixação deste
- d) Lançar pêndulo de ângulo maior que 45 graus
- e) Filmar até a oscilação se limitar a ângulos abaixo de 5 graus, ou pelo menos 2 minutos
- f) Medir massa e diâmetro da esfera assim como o comprimento do pêndulo, com as respectivas incertezas
- g) Importar vídeo e realizar o tracking seguindo as dicas do tutorial e exportar os dados em arquivo TXT para posterior análise



Roteiro

No Mathematica:

1. Seção de Introdução, Metodologia e Dados Experimentais: valores e forma de obtenção de todos os dados, incertezas, etc... (sucinto)
2. Seção do Modelo Matemático: apresentar, desenvolver e gerar gráfico juntamente com a curva experimental
3. Desenvolver rotina para ajuste automático e análise da qualidade do ajuste (partir de chute do Manipulate)
4. Obter o valor da aceleração da gravidade (g) e das constantes de amortecimento (b e c) com as respectivas incertezas
5. Discutir:
 - a) Qualidade do ajuste, criticando o modelo matemático, os dados experimentais, e o ajuste em si
 - b) Possíveis fontes de erro que levaram às incertezas e inexatidão dos parâmetros ajustados e nos resíduos do ajuste
 - c) A origem física de cada componente de amortecimento, apontando os regimes de dominância de cada
 - d) Os limites em que as aproximações por MHS ou MHA passariam a ser interessantes
6. Comentar e formatar o notebook (utilizando células de texto e divisões de seções em locais apropriados)
(explicar o que foi feito em cada célula, explicar os gráficos e interpretar os resultados e citar referências utilizadas)
7. Desafio: (em seção separada no final do notebook):

Obter a real distância entre o ponto de fixação do pêndulo e o centro de oscilação (Q) (tratar como pêndulo físico) e comparar com o que seria obtido matematicamente. Discutir as possíveis influências. Apontar possibilidades de modelo matemático para a substituição do modelo utilizado

Pesos:

1) 1,0 pontos; 2) 2,0 pontos; 3) 2,0 pontos; 4) 2,0 pontos; 5) 2,0 pontos; 6) 1,0 pontos 7) 2,0 pontos (adicionais)



Materiais

no Lab de Mecânica (sala 2650) – Téc. Mário

- Esferas (bolas) e linhas → cada Grupo de cada Turma pode pegar individualmente e devolver no final do módulo
- Réguas e Paquímetros → 1 kit contendo: 2 paquímetros 2 réguas grandes para cada Turma. Devolver no final do módulo
- Balança → usar no local (Lab Mecânica)

Download (material da aula)

- Notebook Mathematica com programa de rastreo (tracking)
- Artigo de F. J Arnold et al. (2011) – pêndulo simples amortecido
- Artigo de C. B. C. Silva et al. (2020) – pêndulo físico

Material Próprio

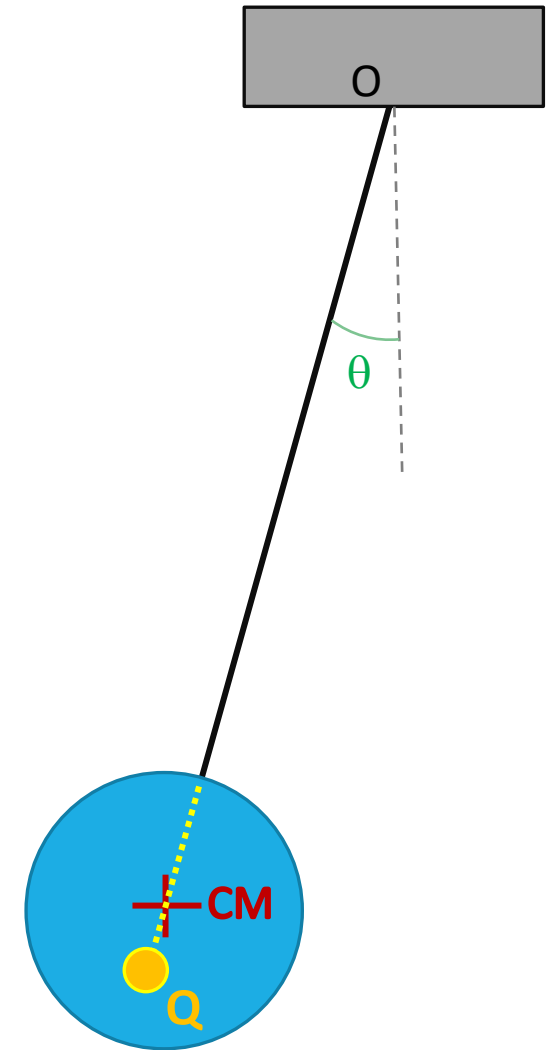
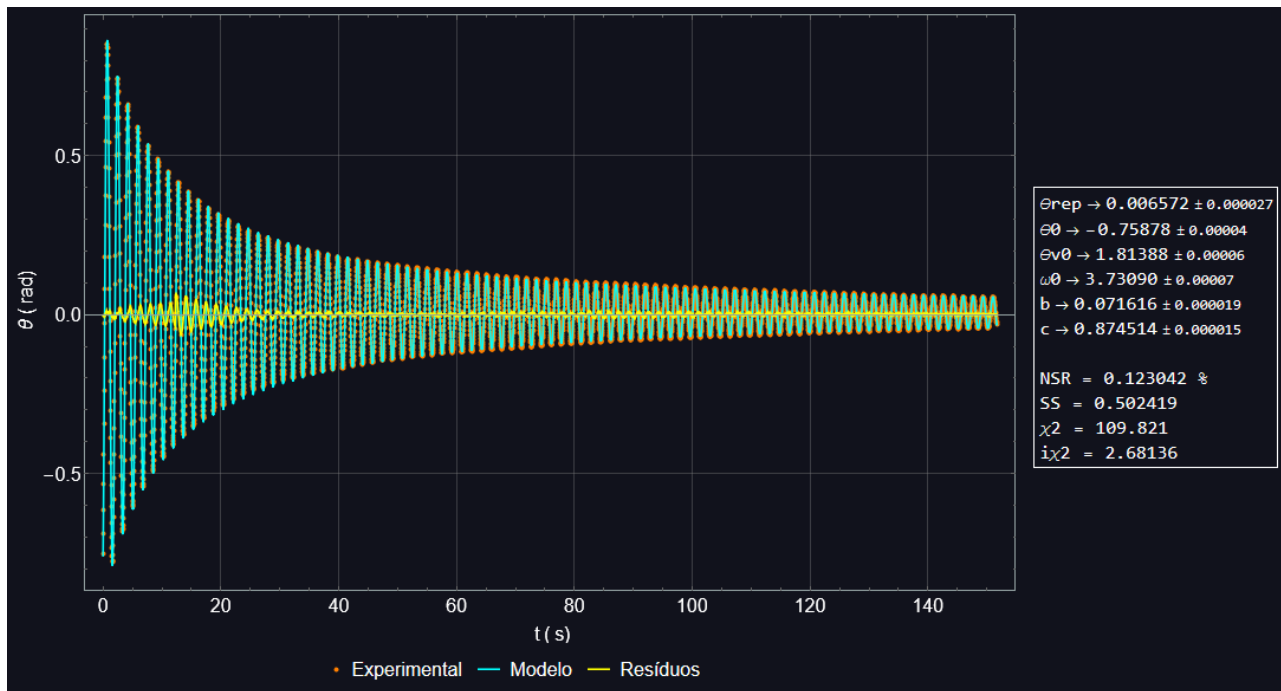
- Celular ou filmadora



Dicas

Modelos

$$\frac{d^2\theta}{dt^2} + \frac{b}{m} \frac{d\theta}{dt} + \frac{Lc}{m} \left| \frac{d\theta}{dt} \right| \frac{d\theta}{dt} + \frac{g}{L} \sin\theta = 0.$$



Dicas

Comando FindFit para ajuste automático

FindFit

(encontra parâmetros de ajuste)

UPDATED IN 12

`FindFit[data, expr, pars, vars]`

finds numerical values of the parameters *pars* that make *expr* give a best fit to *data* as a function of *vars*.

`FindFit[data, {expr, cons}, pars, vars]`

finds a best fit subject to the parameter constraints *cons*.

▼ Properties & Relations (6)

`FindFit` gives the parameter estimates:

```
In[1]:= data = Table[{i, Exp[RandomReal[{i - 1, i}]]}, {i, 10}];
```

```
In[2]:= FindFit[data, Exp[a + b x], {a, b}, x]
```

```
Out[2]:= {a → -2.72603, b → 1.24764}
```

`NonlinearModelFit` allows for extraction of additional information about the fitting:

```
In[3]:= nlm = NonlinearModelFit[data, Exp[a + b x], {a, b}, x]
```

```
Out[3]:= FittedModel[
$$e^{-2.72603 + 1.24764 x}$$
]
```

Extract the parameter estimates:

```
In[4]:= nlm["BestFitParameters"]
```

```
Out[4]:= {a → -2.72603, b → 1.24764}
```

Extract additional results and diagnostics:

```
In[5]:= nlm[{"BestFit", "FitResiduals", "ParameterTable"}]
```

```
Out[5]:= {
$$e^{-2.72603 - 1.24764 x}$$
, {1.08219, 3.12207, 16.2154, 24.9605, 104.08, 222.406, 588.695, -66.6804, -132.326, 27.8251},

|   | Estimate | Standard Error | t Statistic | P-Value                  |
|---|----------|----------------|-------------|--------------------------|
| a | -2.72603 | 0.410164       | -6.64618    | 0.000161446              |
| b | 1.24764  | 0.0413678      | 30.1597     | $1.58529 \times 10^{-9}$ |

}
```



Dicas

Comando FindFit para ajuste automático

NonlinearModelFit (ajusta a um modelo não linear)

`NonlinearModelFit[{y1, y2, ...}, form, {β1, ...}, x]`

constructs a nonlinear model with structure *form* that fits the y_i for successive x values 1, 2, ... using the parameters $\beta_1, \dots$

`NonlinearModelFit[{{x11, x12, ..., y1}, {x21, x22, ..., y2}, ...}, form, {β1, ...}, {x1, ...}]`

constructs a nonlinear model where *form* depends on the variables x_k .

`NonlinearModelFit[data, {form, cons}, {β1, ...}, {x1, ...}]`

constructs a nonlinear model subject to the parameter constraints *cons*.

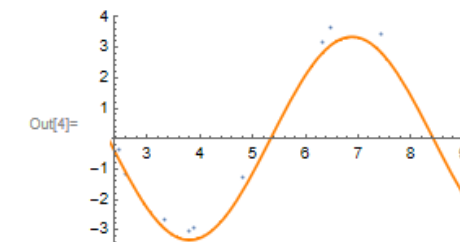
Generalizations & Extensions (2)

Fit data to a model defined by a numerical operation:

```
In[1]:= data = {{6.47, 3.65}, {7.43, 3.45}, {3.9, -2.94}, {4.8, -1.29}, {2.48, -0.35},  
             {6.32, 3.16}, {2.59, -1.19}, {9.13, -2.}, {3.81, -3.04}, {3.33, -2.68}};  
  
In[2]:= model[a_?NumberQ, b_?NumberQ, c_?NumberQ] := Module[{y, x},  
    First[y /. NDSolve[{y''[x] + a y[x] == 0, y[0] == b, y'[0] == c}, y, {x, 0, 10}]]]  
  
In[3]:= nlm = NonlinearModelFit[data, model[a, b, c][x], {a, b, c}, x, Method -> "Gradient"]
```

Out[3]= FittedModel[InterpolatingFunction[ Domain: {{0., 10.}}
Output: scalar][x]]

```
In[4]:= Show[ListPlot[data], Plot[nlm[x], {x, 0, 10}, PlotStyle -> Orange]]
```



Make the computation much faster by caching solutions of the differential equation:

```
In[5]:= Clear[model];  
model[a_?NumberQ, b_?NumberQ, c_?NumberQ] := (model[a, b, c] = Module[{y, x},  
    First[y /. NDSolve[{y''[x] + a y[x] == 0, y[0] == b, y'[0] == c}, y, {x, 0, 10}]]])  
  
In[7]:= nlm = NonlinearModelFit[data, model[a, b, c][x], {a, b, c}, x, Method -> "Gradient"]
```

Out[7]= FittedModel[InterpolatingFunction[ Domain: {{0., 10.}}
Output: scalar][x]]



Dicas

Comando FindFit para ajuste automático

NonlinearModelFit (ajusta a um modelo não linear)

`NonlinearModelFit[{y1, y2, ...}, form, {β1, ...}, x]`

constructs a nonlinear model with structure *form* that fits the y_i for successive x values 1, 2, ... using the parameters $\beta_1, \dots$

`NonlinearModelFit[{{x11, x12, ..., y1}, {x21, x22, ..., y2}, ...}, form, {β1, ...}, {x1, ...}]`

constructs a nonlinear model where *form* depends on the variables x_k .

`NonlinearModelFit[data, {form, cons}, {β1, ...}, {x1, ...}]`

constructs a nonlinear model subject to the parameter constraints *cons*.

▼ Sums of Squares (1)

Fit a nonlinear model to some data:

```
In[1]:= data = Flatten[Table[{x, y, .2 Log[5 x + .1 y] + RandomReal[.1]},  
  {x, RandomReal[5, 10]}, {y, RandomReal[5, 10]}], 1];  
nlm = NonlinearModelFit[data, a Log[b x + c y], {a, b, c}, {x, y}]
```

```
Out[1]= FittedModel[0.180782 Log[8.8973 x - 0.0288818 y]]
```

Extract the estimated error variance:

```
In[2]:= nlm["EstimatedVariance"]
```

```
Out[2]= 0.000951786
```

Obtain the analysis of variance table:

```
In[3]:= nlm["ANOVATable"]
```

	DF	SS	MS
Model	3	28.2083	9.40277
Error	97	0.0923232	0.000951786
Uncorrected Total	100	28.3006	
Corrected Total	99	1.6317	

▼ Goodness-of-Fit Measures (1)

Obtain a table of goodness-of-fit measures for a nonlinear model:

```
In[5]:= data = {{6.6, 9.4, 14.4}, {0.1, 5.6, 11.1}, {0.1, 3.1, 6.5}, {9.5, 4.7, 11.1},  
  {6.6, 2., 6.8}, {7., 6.2, 11.1}, {7., 6.9, 12.4}, {7., 1.5, 8.9},  
  {6.6, 0.2, 8.6}, {6.6, 2.1, 11.4}, {6.6, 9.3, 11.8}, {9.5, 2.5, 10.9}};
```

```
In[7]:= nlm = NonlinearModelFit[data, Exp[a x + b y], {a, b}, {x, y}]
```

```
Out[7]= FittedModel[e0.224818 x + <<20>> y]
```

```
In[8]:= Grid[Transpose[{#, nlm[#]} &[{"AdjustedRSquared", "AIC", "BIC", "RSquared"}],  
  Alignment → Left]
```

```
AdjustedRSquared 0.8331  
Out[8]= AIC      73.1501  
        BIC      74.6048  
        RSquared 0.860916
```



Dicas

Comando FindFit para ajuste automático

NonlinearModelFit (ajusta a um modelo não linear)

`NonlinearModelFit[{y1, y2, ...}, form, {β1, ...}, x]`

constructs a nonlinear model with structure *form* that fits the y_i for successive x values 1, 2, ... using the parameters $\beta_1, \dots$

`NonlinearModelFit[{x11, x12, ..., y1}, {x21, x22, ..., y2}, ..., form, {β1, ...}, {x1, ...}]`

constructs a nonlinear model where *form* depends on the variables x_k .

`NonlinearModelFit[data, {form, cons}, {β1, ...}, {x1, ...}]`

constructs a nonlinear model subject to the parameter constraints *cons*.

▼ Applications (1)

Simulate some data:

```
In[1]:= data = BlockRandom[SeedRandom[12345];  
Table[{x, Exp[-2.3 x / (11 + .4 x + x^2)] + RandomReal[{- .5, .5}]},  
{x, RandomReal[{1, 15], 20}]]];
```

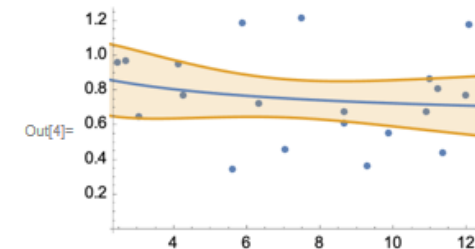
Fit a nonlinear model to the data:

```
In[2]:= nlm = NonlinearModelFit[data, Exp[a x / (b + c x)], {a, b, c}, x]
```

Out[2]= FittedModel
$$\left[\frac{7.27565 x}{e^{-72.8613 + \ll 19 \gg x}} \right]$$

Obtain and visualize 90% confidence bands for the fit:

```
In[3]:= bands90[x_] = nlm["MeanPredictionBands", ConfidenceLevel -> .9];  
Show[ListPlot[data], Plot[{nlm[x], bands90[x]}, {x, 1, 15}, Filling -> {2 -> {1}}]]
```



Obtain 95%, 99%, and 99.9% confidence bands:



INSTITUTO
TECNOLÓGICO
DE AERONÁUTICA

Introdução à Física
Experimental

Prof. Douglas Leite
www.lpp.ita.br/leite

Ajuste de modelos matemáticos não lineares em dados experimentais usando o Mathematica®

Aula 3

Vídeo 1 – Introdução e atividades da aula

Vídeo 2 – Tutorial: Vídeo-análise no Mathematica

