

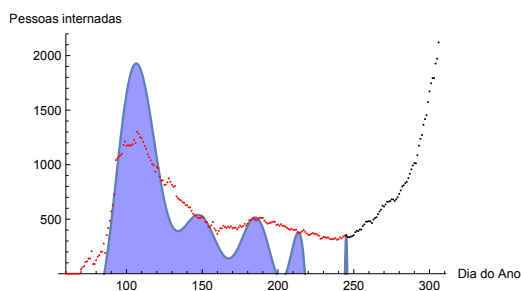
1º Trabalho Computacional - Análise Numérica (2020/21)

Relatório-resumo do Grupo: 24

Nome: Vasco Pearson (nº97015)

Questão 1.a):

[R1a1] Os pontos a vermelho representam os valores reais conhecidos do número de pessoas internadas e os pontos a preto são os valores reais verificados posteriormente, enquanto que o gráfico interpolador encontra-se a azul, estendendo até ao dia 306 (1 de Novembro). O gráfico foi cortado em $y = 0$ pois não faz sentido termos um número negativo de pessoas internadas.



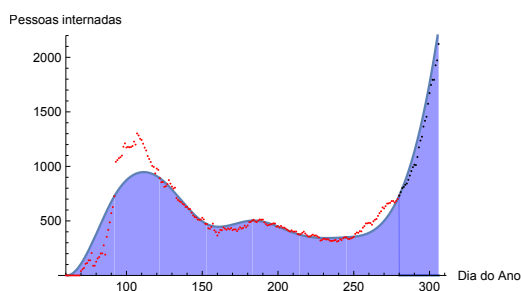
[R1a2] Construiu-se a seguinte tabela:

x_k	Valores exatos	Estimados	Erro Abs.
75	139	-223	362
76	139	-243	382
107	1302	1925	623
137	657	411	246
168	423	140	283
198	476	40	436
229	325	-6696	7021
260	482	-1107543	1108025
306	2122	-1732919298	1732921420

A interpolação polinomial leva a problemas de instabilidade quando consideramos um número de nós muito elevado, o que se verificou neste caso. Obtemos um polinómio de grau muito elevado e devido a isso, nos pontos apresentados, obtemos um erro absoluto também elevado o que indica uma má aproximação do polinómio interpolador. Podemos ainda notar que o erro nos valores interpolados foi muito menor do que nos valores extrapolados (onde se verificou um erro absoluto máximo de 1732922122).

Questão 1.b)

[R1b1] O mesmo que anteriormente, agora com o gráfico obtido por interpolação por splines cúbicos encastrados.



[R1b2] A nova tabela obtida é a seguinte:

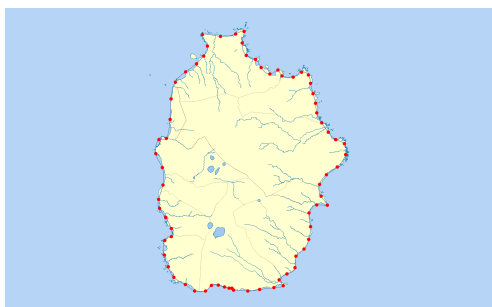
x_k	Valores exatos	Estimados	Erro Abs.
75	139	230	91
76	139	258	119
107	1302	936	366
137	657	676	19
168	423	459	36
198	476	459	17
229	325	343	18
260	482	390	92
295	1272	1418	146
306	2122	2262	140

Reparamos agora que os valores apresentam um erro muito menor do que na alínea anterior, o que indica que os valores estimados são muito mais próximos da realidade. Ao fazer uma interpolação por splines cúbicos estamos a considerar funções seccionalmente polinomiais de grau três e não um só polinómio de grau muito elevado. Isto leva a uma melhor aproximação que, neste caso, se prolonga aos valores extrapolados, pois o crescimento acentuado previsto pelos valores conhecidos foi acompanhado pela função interpoladora.

Questão 2.a)

[R2a1]

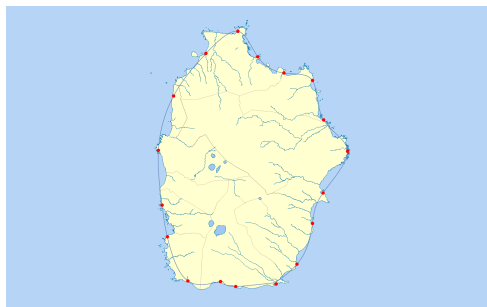
A ilha atribuída ao meu grupo foi a ilha das Flores, pertencente ao arquipélago dos Açores. Foram selecionados 72 pontos onde o primeiro é igual ao último e foi construída uma função interpoladora por splines cúbicos encastrados para as coordenadas x e y separadamente. Em baixo encontra-se a ilha e a função parametrizada com os pontos utilizados.



Observação: As derivadas no primeiro e último ponto (estas coincidem) foram calculadas pela fórmula das diferenças centradas.

[R2a2]

Aqui considerámos 19 pontos em que o primeiro é igual ao último. Este foi o menor número de nós possível, de forma a manter o aspeto geral da ilha. Podemos observar que mesmo com poucos pontos, a função ajusta-se bem ao contorno da ilha.



Questão 2.b)

Nesta pergunta foram usadas as seguintes fórmulas para a Área e para o Perímetro da ilha:

$$\text{Área} = \int_a^b y(t)x'(t) dt \quad \text{Perímetro} = \int_a^b \sqrt{y'(t)^2 + x'(t)^2} dt$$

Foi utilizado o método de Simpson para aproximar o integral com um método de quadratura. Este tem a seguinte expressão:

$$Q(f) = \frac{b-a}{6}(f(a) + 4f(\frac{a+b}{2}) + f(b))$$

Os resultados obtidos foram:

Pontos	Área (km ²)	Perímetro (km)
72	146.742	54.1011
19	146.379	48.9603

e comparando com a área oficial 143 Km², concluímos que a aproximação é boa, apesar de neste caso poder haver um ligeiro desvio nos resultados devido ao ajuste de pontos e devido à escala. Observamos ainda que quando se considera apenas os 19 pontos o perímetro varia mais, mas que no entanto a área obtida neste caso está ligeiramente mais próxima do valor da área real, o que representa de novo a capacidade dos splines se ajustarem a um contorno.

– Código Mathematica –

Questão 1.a):

```
DifDivididas =
Function[{lista},
Module[{m, N, a, tabela = {}, out = {}, aux, i, k, DD, A, max},
m = Length[lista] - 1;
a = Map[Function[w, Length[w] - 2], lista];
A = Apply[Plus, a];
(*Preencher a tabela auxiliar*)
For[k = 1, k <= m + 1, k++,
For[i = 1, i <= a[[k]] + 1, i++,
tabela = Append[tabela, lista[[k]]]];
N = Length[tabela];
(*Calcular as DD de ordem 1*)
out = Append[out, Map[Function[w, w[[2]]], tabela]];
(*Calcular as restantes DD*)max = m + A + 1;
For[k = 2, k <= max, k++, aux = {}];
(*Calcular as DD de ordem k*)
For[i = 1, i <= N - k + 1, i++,
If[tabela[[i, 1]] === tabela[[i + k - 1, 1]],
DD = 1/(k - 1)! tabela[[i, k + 1]],
DD = (out[[k - 1, i + 1]] -
out[[k - 1, i]])/(tabela[[i + k - 1, 1]] - tabela[[i, 1]]);
aux = Append[aux, DD]];
out = Append[out, aux]];
out]];
Hermite =
Function[input,
Module[{DD, N, x = {}, m, a, k, i},
a = Map[Function[w, Length[w] - 2], input];
m = Length[input] - 1;
N = m + Total[a];
For[k = 1, k <= m + 1, k++,
For[i = 1, i <= a[[k]] + 1, i++, x = Append[x, input[[k, 1]]]];
(*Queremos apenas as DD úteis para a interpolação:*)
DD = Map[First, DifDivididas[input]];
Sum[DD[[k + 1]] Product[(t - x[[i + 1]]), {i, 0, k - 1}], {k, 0,
N}]]];
dyk[yk_, ykm1_, ykm2_] := yk - (ykm1 + ykm2)/2 (*yk'*);
listaNos = {{60, 0, 0}, {61, 0, 0}, {92, 726,
dyk[726, 627, 571]}, {122, 892, dyk[892, 968, 980]}, {153, 471,
dyk[471, 474, 514]}, {183, 503, dyk[503, 491, 489]}, {214, 375,
dyk[375, 381, 403]}, {245, 350, dyk[350, 349, 341]}};
listaValoresY = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 40, 57, 69, 107, 114,
139, 139, 206, 89, 89, 126, 156, 169, 201, 203, 276, 191, 354, 418,
486, 571, 627, 726, 1042, 1058, 1075, 1084, 1099, 1180, 1211,
1173, 1179, 1175, 1177, 1187, 1227, 1200, 1302, 1284, 1253, 1243,
1208, 1172, 1146, 1095, 1068, 1040, 1005, 995, 936, 980, 968, 892,
855, 856, 813, 818, 838, 874, 842, 815, 797, 805, 709, 692, 680,
```

```

673, 657, 649, 628, 629, 609, 608, 576, 550, 536, 531, 513, 510,
512, 529, 514, 474, 471, 432, 428, 445, 475, 414, 398, 366, 394,
417, 415, 440, 428, 419, 431, 423, 435, 416, 422, 422, 407, 424,
441, 429, 436, 457, 442, 458, 489, 491, 503, 510, 495, 489, 504,
513, 511, 512, 487, 471, 459, 462, 467, 472, 478, 476, 447, 452,
439, 454, 439, 439, 431, 420, 410, 403, 414, 402, 403, 403, 381,
375, 378, 390, 401, 384, 369, 356, 357, 366, 374, 365, 367, 358,
348, 320, 325, 336, 336, 329, 334, 321, 316, 317, 321, 325, 311,
317, 334, 324, 341, 349, 350, 337, 334, 339, 345, 354, 381, 394,
391, 406, 404, 438, 452, 477, 478, 482, 480, 465, 497, 511, 518,
546, 571, 588, 624, 615, 635, 659, 661, 666, 682, 682, 668, 682,
701, 732, 764, 801, 811, 831, 843, 877, 916, 957, 993, 1015, 1012,
1086, 1174, 1237, 1272, 1365, 1418, 1455, 1574, 1672, 1747, 1794,
1794, 1927, 1972, 2122};
listaValores = {};
For[i = 1, i <= Length[listaValoresY], i++,
  AppendTo[listaValores, {i + 59, listaValoresY[[i]]}]];
polinomio = Hermite[listaNos];
Plot[polinomio, {t, 60, 306}, PlotRange -> {0, 2200},
  AxesLabel -> {"Dia do Ano", "Pessoas internadas"},
  Epilog -> {PointSize[0.005], Red,
    Map[Point, listaValores[[1 ;; 245 - 59]]], Black,
    Map[Point, listaValores[[245 - 59 ;;]]], Filling -> Axis,
    FillingStyle -> Opacity[0.4, Blue]] \[NewLine]
Tabela = Function[{f, lval, lnos, perg},
  Module[{xk = {}, vexatos = {}, vinterp = {}, erroabs = {}, i},
    For[i = 1, i <= Length[lnos], i++,
      AppendTo[xk, lnos[[i]][[1]] + 15]];
    AppendTo[xk, 306];
    For[i = 1, i <= Length[xk], i++,
      AppendTo[vexatos, lval[[xk[[i]] - 59]][[2]]];
    For[i = 1, i <= Length[xk], i++,
      AppendTo[vinterp, IntegerPart[f /. t -> xk[[i]]]];
    If[perg == "R1b1",
      vinterp[[-1]] = IntegerPart[f[[1]][[8]][[1]] /. t -> xk[[-1]]];
      vinterp[[-2]] = IntegerPart[f[[1]][[8]][[1]] /. t -> xk[[-2]]];
    For[i = 1, i <= Length[xk], i++,
      AppendTo[erroabs, Abs[vexatos[[i]] - vinterp[[i]]]];
    TableForm[{xk, vexatos, vinterp, erroabs}, TableDirections -> Row,
      TableAlignments -> Center,
      TableHeadings -> {"xk", "Valores Exatos", "Valores Estimados",
        "Erro Abs"}]]];
Tabela[polinomio, listaValores, listaNos, ""]

```

Questão 1.b):

```

DifDivididas>(*Definido na pergunta anterior*)
Splines =
Function[{input, df0, dfn},
  Module[{n, x, f, h, A, DD, b, dds, ds, s}, n = Length[input] - 1;

```

```

x = Map[First, input];
f = Map[Last, input];
h = Join[{0}, Table[x[[i + 1]] - x[[i]], {i, 1, n}], {0}];
A = SparseArray[{{i_, j_} /; i == j :> (h[[i]] + h[[i + 1]])/
  3, {i_, j_} /; i == j + 1 :>
  h[[i]]/6, {i_, j_} /; i + 1 == j :> h[[i + 1]]/6}, {n + 1,
  n + 1}];
h = Most[Rest[h]];
DD = DifDivididas[input][[2]];(*São as únicas necessárias*)
b = Join[{DD[[1]] - df0},
  Table[DD[[i]] - DD[[i - 1]], {i, 2, n}], {dfn - DD[[-1]]}];
dds = LinearSolve[A, b];
ds = Table[
  DD[[k]] - h[[k]] (2 dds[[k]] + dds[[k + 1]])/6, {k, 1, n}];
s = Table[{f[[k]] + (t -
  x[[k]]) ds[[k]] + ((t - x[[k]])^2) dds[[k]]/
  2 + ((t -
  x[[k]])^3) ((dds[[k + 1]] - dds[[k]])/(x[[k + 1]] -
  x[[k]])))/6, x[[k]] <= t <= x[[k + 1]]}, {k, 1, n}];
Piecewise[s, 0]];
listaNos2 = {{60, 0}, {61, 0}, {92, 726}, {122, 892}, {153,
  471}, {183, 503}, {214, 375}, {245, 350}, {280, 732}};
listaValores;(*Definido na alínea anterior*)
spline = Splines[listaNos2, 0, 732 - 701];
prolongamento = spline[[1]][[8]][[1]];(*Prolongar o gráfico até 306*)
Show[
  Plot[spline, {t, 60, 306}, PlotRange -> {0, 2200},
  AxesLabel -> {"Dia do Ano", "Pessoas internadas"},
  Epilog -> {PointSize[0.005], Red,
  Map[Point, listaValores[[1 ;; 280 - 59]]], Black,
  Map[Point, listaValores[[280 - 59 ;;]]], Filling -> Axis,
  FillingStyle -> Opacity[0.4, Blue]],
  Plot[prolongamento, {t, 280, 306}, PlotRange -> {0, 2200},
  Filling -> Axis, FillingStyle -> Opacity[0.4, Blue]]]\[NewLine]
Tabela;(*Definido na pergunta anterior*)
Tabela[spline, listaValores, listaNos2, "R1b1"]

```

Questão 2.a):

```

DifDivididas;(*Definido na pergunta anterior*)
Splines;(*Definido na pergunta anterior*)
link1 = "https://upload.wikimedia.org/wikipedia/"
link2 = "commons/1/16/Locator_map_Azores_Flores.png"
FloresMapa = Import[StringJoin[a, b]];
Coord = {{-46, -360}, {0, -362}, {38, -358}, {84, -352}, {114, -346},
{101, -327}, {126, -308}, {152, -288}, {155, -251}, {181, -228},
{196, -197}, {202, -156}, {197, -112}, {222, -86}, {256, -87}, {236,
-58}, {231, -20}, {253, 11}, {288, 36}, {315, 76}, {311, 111},
{283, 124}, {259, 148}, {238, 178}, {222, 210}, {218, 241}, {210, 272},
{202, 305}, {195, 332}, {173, 341}, {146, 325}, {110, 329}, {97, 348},

```

```

{71, 335}, {43, 356}, {25, 382}, {-5, 395}, {-18, 435}, {-12, 472},
{-39, 464}, {-88, 456}, {-146, 462}, {-130, 425}, {-142, 393}, {-165, 368},
{-200, 342}, {-233, 309}, {-247, 255}, {-250, 189}, {-262, 128},
{-286, 124}, {-296, 79}, {-275, 34}, {-273, -22}, {-287, -68},
{-284, -97}, {-264, -126}, {-246, -162}, {-246, -188}, {-267, -200},
{-269, -247}, {-256, -285}, {-238, -319}, {-201, -342}, {-171, -362},
{-136, -362}, {-117, -345}, {-95, -344}, {-75, -350}, {-60, -354}, {-51, -354},
{-46, -360}};
For[i = 1, i <= Length[Coord], i++, Coord[[i]] = Coord[[i]]/48.85];
CoordX = {};
For[i = 1, i <= Length[Coord], i++,
  AppendTo[CoordX, {i, Coord[[i]][[1]]}]];
CoordY = {};
For[i = 1, i <= Length[Coord], i++,
  AppendTo[CoordY, {i, Coord[[i]][[2]]}]];
f0[Coordenadas_] := (Coordenadas[[-2, 2]] - Coordenadas[[2, 2]])/2;
(*Calcular a derivada f0=fn*)
splineX = Splines[CoordX, f0[CoordX], f0[CoordX]];
splineY = Splines[CoordY, f0[CoordY], f0[CoordY]];
ImageCompose[FloresMapa, {ParametricPlot[{splineX, splineY}, {t, 1,
  72}, PlotRange -> {-8.3, 11.7},
  Epilog -> {PointSize[0.01], Red, Color -> Red, Map[Point, Coord]},
  Axes -> False], 1}, Scaled[{.549, .526}]] \[NewLine]
CoordX4 = {};
For[i = 1, i <= Length[Coord]/4, i++,
  AppendTo[CoordX4, {i, Coord[[4 i]][[1]]}]]
AppendTo[CoordX4, {72/4 + 1,
  Coord[[4]][[1]]}];(*de forma a que o primeiro ponto e o último sejam iguais*)
CoordY4 = {};
For[i = 1, i <= Length[Coord]/4, i++,
  AppendTo[CoordY4, {i, Coord[[4 i]][[2]]}]]
AppendTo[CoordY4, {72/4 + 1, Coord[[4]][[2]]}];
splineX4 = Splines[CoordX4, f0[CoordX4], f0[CoordX4]];
splineY4 = Splines[CoordY4, f0[CoordY4], f0[CoordY4]];
ImageCompose[FloresMapa, {ParametricPlot[{splineX4, splineY4}, {t, 1,
  72/4 + 1}, PlotRange -> {-8.3, 11.8},
  Epilog -> {PointSize[0.01], Red, Color -> Red,
  Map[Point, Take[Coord, {4, 72, 4}]]}, Axes -> False], 1},
  Scaled[{.553, .532}]]

```

Questão 2.b):

```

Derivada[spline_, n_] :=
Piecewise[
  Table[{Sum[
    j*Coefficient[spline[[1, i, 1]], t, j]*t^(j - 1), {j, 1, 3}],
    spline[[1, i, 2]]}, {i, 1, n - 1}]]
DsplineX = Derivada[splineX, 72];
DsplineY = Derivada[splineY, 72];
DsplineX4 = Derivada[splineX4, 72/4 + 1];

```

```

DsplineY4 = Derivada[splineY4, 72/4 + 1];
area72pontos =
Abs[Sum[(1/6)*(((splineY*D splineX) /. t -> k) +
4*((splineY*D splineX) /.
t -> (k + 1/2)) + ((splineY*D splineX) /. t -> (k + 1))), {k,
1, 71}]] \[NewLine]
perimetro72pontos =
Sum[(1/6)*(((D splineX^2) + (D splineY^2))^(1/2) /. t -> k) +
4*((D splineX^2) + (D splineY^2))^(1/2) /.
t -> (k + 1/2)) + (((D splineX^2) + (D splineY^2))^(1/2) /.
t -> k + 1)), {k, 1, 71}] \[NewLine]
area19pontos =
Abs[Sum[(1/6)*(((splineY4*D splineX4) /. t -> k) +
4*((splineY4*D splineX4) /.
t -> (k + 1/2)) + ((splineY4*D splineX4) /.
t -> (k + 1))), {k, 1, 71}]] \[NewLine]
perimetro19pontos =
Sum[(1/6)*(((D splineX4^2) + (D splineY4^2))^(1/2) /. t -> k) +
4*((D splineX4^2) + (D splineY4^2))^(1/2) /.
t -> (k + 1/2)) + (((D splineX4^2) + (D splineY4^2))^(1/2) /.
t -> k + 1)), {k, 1, 71}]

```

Observação: Devido ao tamanho ocupado pelos dados, foi utilizada esta página para finalizar o relatório