Notas em Interpolação Polinomial

Matheus do Ó Santos Tiburcio

Sumário

1. Introdução

1.a. Definição de interpolação polinomial:

Meu trabalho tem como tema interpolação polinomial. Acho importante começar com uma breve definição de o que é interpolação polinomial através da ideia de uma função interpoladora.

Definição: Dado um número finito de pontos, a função interpoladora vai ser aquela que melhor se aproximar de todos esses pontos dados em um dado grau pré-estabelecido.

Entrada:

- Pontos;
- Grau g da função interpoladora.

Saída:

- Uma função de grau g.

Vale lembrar que o grau também é definido. Desse modo, para o mesmo grupo de pontos você pode ter um função interpoladora de grau 1, 2 e assim em diante. O que é interessante da interpolação é que ela tem solução única, ou seja, mesmo que você faça o mesmo cálculo ou, no meu caso, rode o programa mais de uma vez com o mesmo grupo de pontos, o resultado gerado vai sempre ser o mesmo naquele grau selecionado.

Um ponto legal de falar é que há mais de um jeito de fazer uma interpolação polinomial e o método que utilizei é só um deles.

1.b. Método utilizado:

O método que eu usei foi basicamente mínimos quadrados até dizer chega. O modo de achar a função interpoladora basicamente consistia na solução do sistema $A^TAx = A^Tb$. Mas o que seria cada coisa?

A matriz A:

O A é uma matriz rxn* e consiste na clássica matriz de Vandermonde, que basicamente posiciona em forma crescente ou decrescente as potências de um dado x.
Exemplo:
Seja x1 = 2, x2 = 5
Teremos a matriz de Vandermonde como:
(1)

$$\begin{bmatrix} 2^n & 2^{n-1} & 2^{n-2} & 2^{n-3} & \dots & 2^0 \\ 5^n & 5^{n-1} & 5^{n-2} & 5^{n-3} & \dots & 5^0 \end{bmatrix}$$

Onde o n será discutido mais à frente.

As linhas dessa matriz remetem às restrições, que nesse caso seriam o número de pontos dados. Nesse sentido, se 10 pontos foram dados, teremos 10 linhas nessa matriz, em (1) podemos ver que x_1 e x_2 estão cada um posicionado em uma linha, você pode ler eles como coordenada x do ponto 1 e coordenada x do ponto 2 respectivamente. Já as colunas remetem às potências dos x em questão e estão diretamente ligadas a que grau o polinômio vai ter.

O vetor x:

O vetor x é é um vetor nx1 e é onde estão os coeficientes. Por ser uma função polinomial, temos coeficientes e esses coeficientes que serão o brilho da interpolação no futuro.

O vetor b:

Outro vetor rx1 e, diferentemente da matriz A que tem os x dos pontos, o vetor b vai guardar as coordenadas y de todos os pontos. Desse modo a coordenada y do ponto 1 vai estar na linha 1, a do 2 na linha 2 e assim por diante.

**Escolhi r para representar as restrições e n para representar o número de potências que teremos.*

E por que isso funciona?

Primeiro acho legal vermos como uma função polinomial de uma variável é escrita:

$p(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + ... + a_nx^0$

Onde a representa seus coeficientes.

Se olharmos bem para essa fórmula vemos que não é nada mais do que combinações entre os coeficientes e as potências de x, desse modo podemos escrever isso como a multiplicação de vetores abaixo:

(2)

$$\begin{bmatrix} x^n & x^{n-1} & x^{n-2} & x^{n-3} & .... & x^0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ .... \\ a_n \end{bmatrix} = p(x) = y$$

, onde n representará também o grau do polinômio!

E voi là!

Se generalizarmos isso para n $p(x)$ chegamos na Matriz A, que é uma matriz de Vandermonde:

$$\begin{bmatrix} {x_0}^n & {x_0}^{n-1} & {x_0}^{n-2} & {x_0}^{n-3} & .... & {x_0}^0 \\ {x_1}^n & {x_1}^{n-1} & {x_1}^{n-2} & {x_1}^{n-3} & .... & {x_1}^0 \\ {x_2}^n & {x_2}^{n-1} & {x_2}^{n-2} & {x_2}^{n-3} & .... & {x_2}^0 \\ .... & .... & .... & .... & .... & .... \\ .... & .... & .... & .... & .... & .... \\ {x_{r-1}}^n & {x_{r-1}}^{n-1} & {x_{r-1}}^{n-2} & {x_{r-1}}^{n-3} & .... & {x_{r-1}}^0 \end{bmatrix}$$

E no vetor b:

$$\begin{bmatrix} p(x)_0 \\ p(x)_1 \\ p(x)_2 \\ p(x)_3 \\ \cdots\cdot \\ p(x)_{r-1} \end{bmatrix}$$

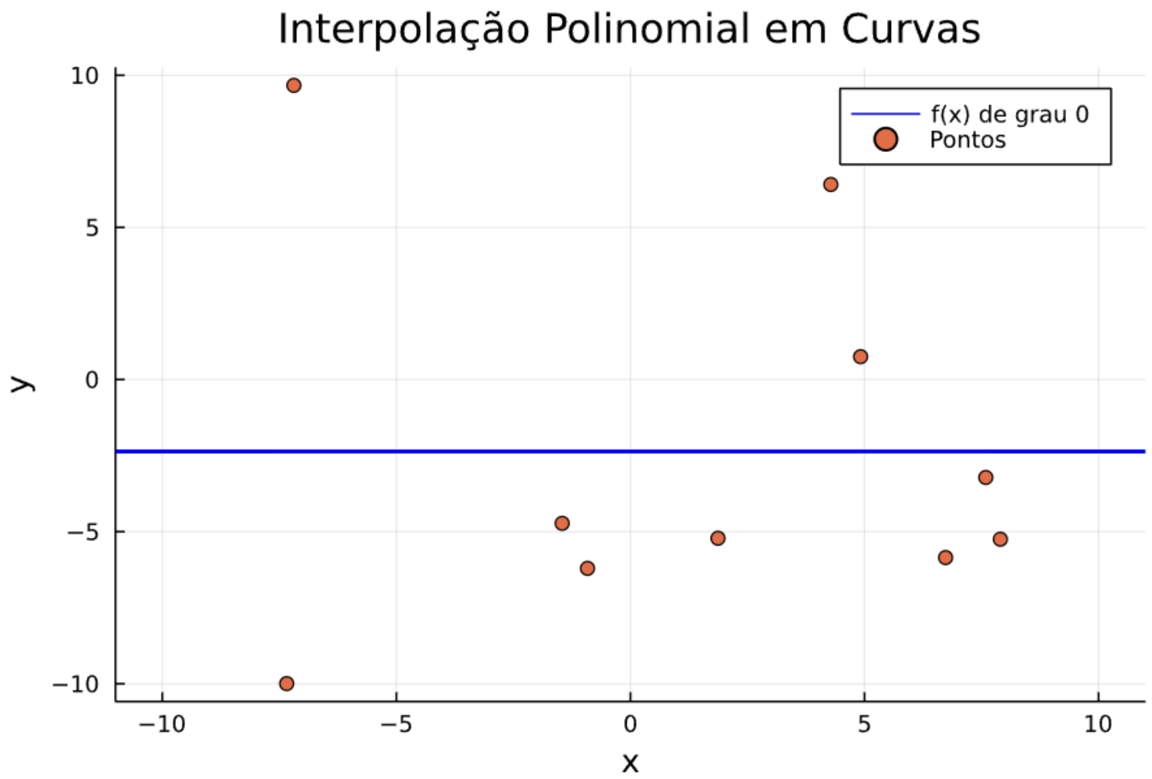
O vetor x será igual ao vetor de coeficientes mostrado em (2).

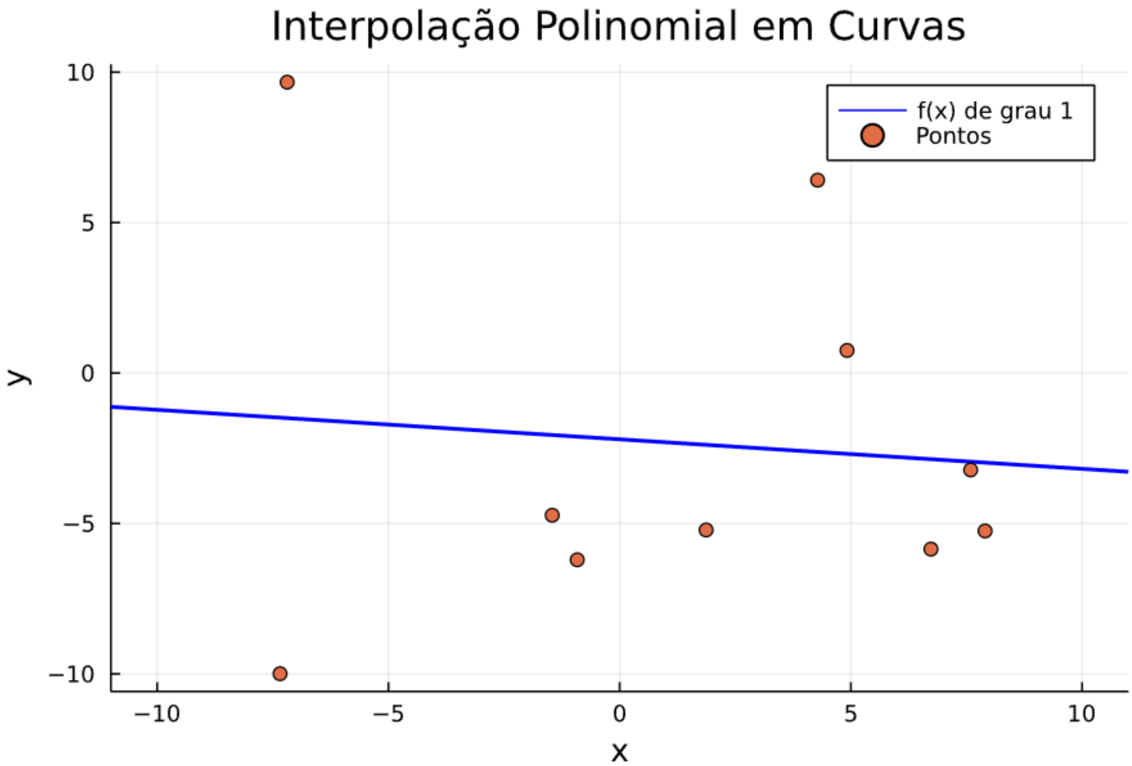
2. Interpolação polinomial em R^2:

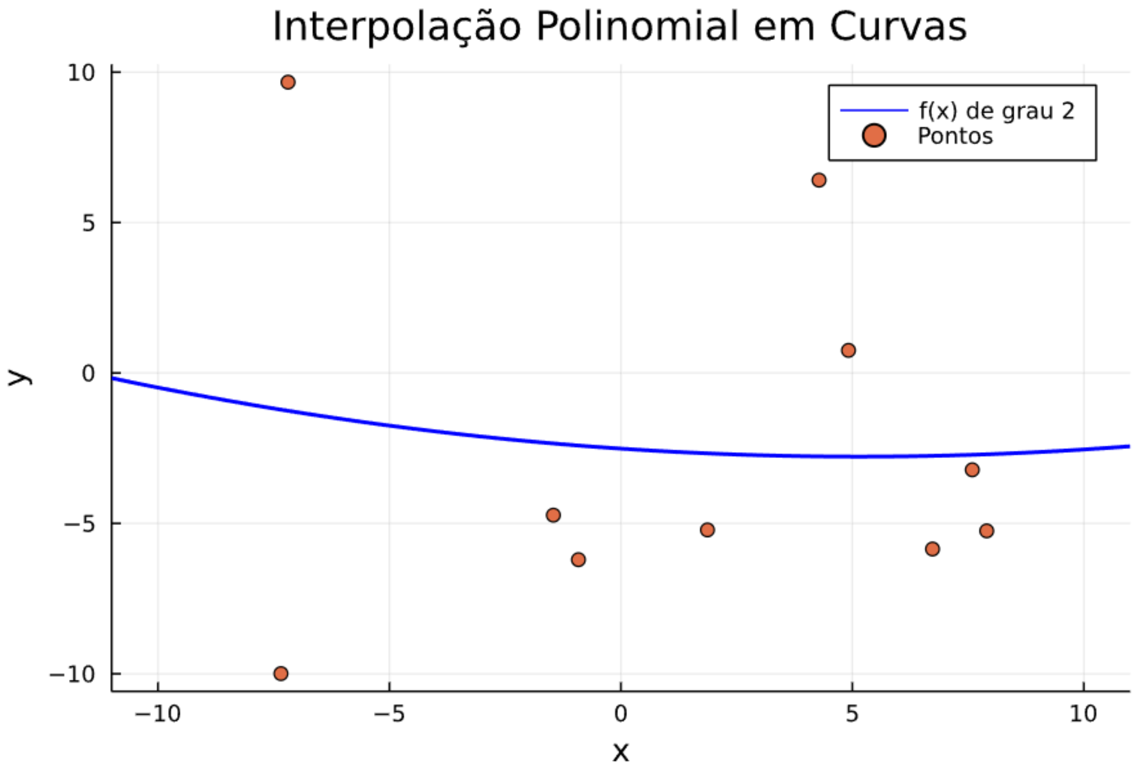
Uma parte do meu projeto diz respeito a interpolar pontos (x, y) quaisquer em R2. Abaixo vou colocar os resultados de alguns testes que fiz:

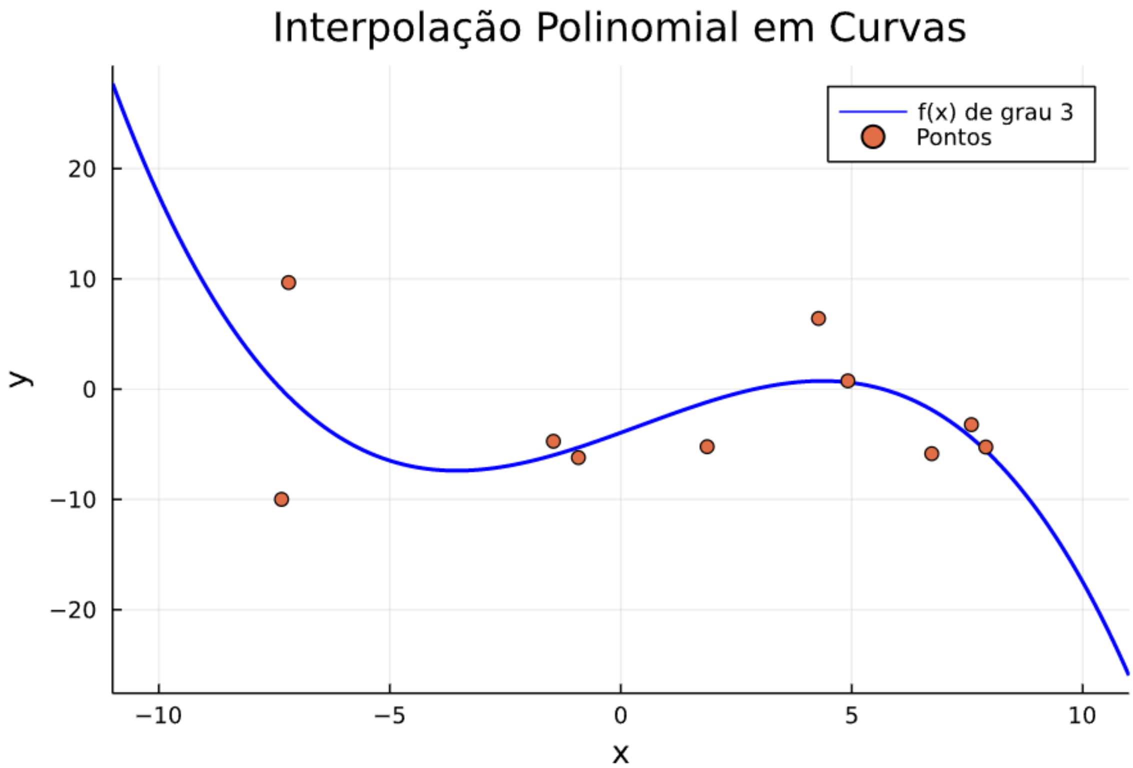
2.a. Gráficos

Interpolações feitas com o conjunto de pontos B1:*

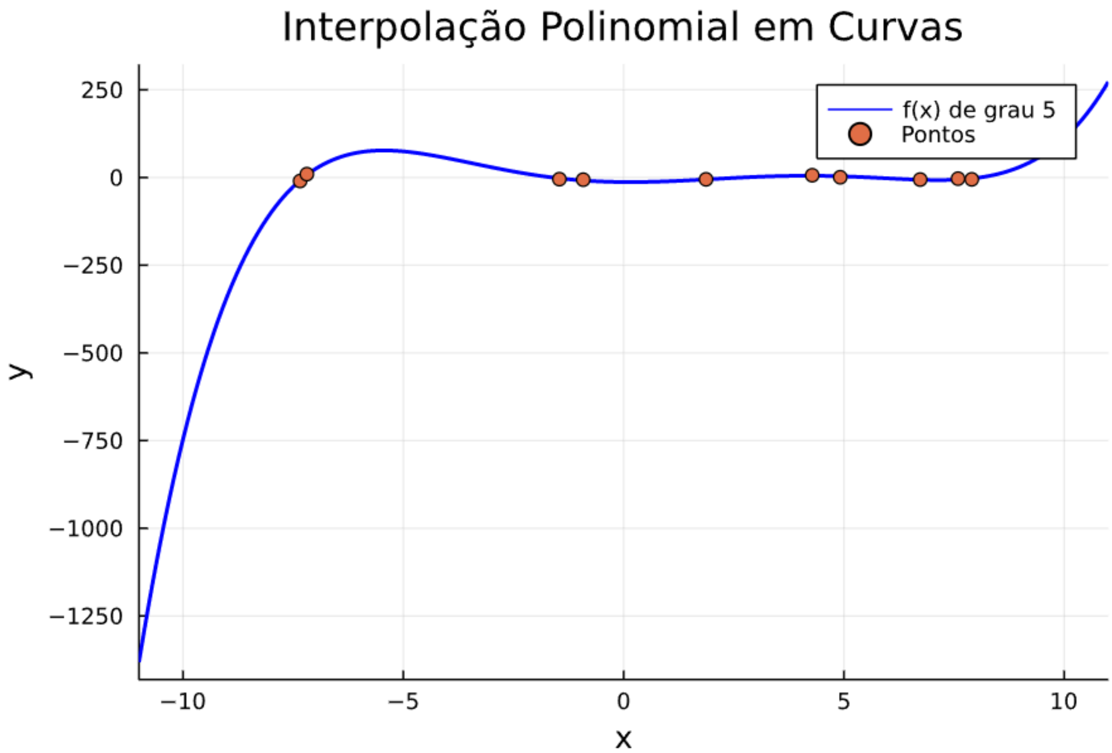



Interpolação Polinomial em Curvas
f(x) de grau 1
Pontos
y
x
10
5
0
−5
−10
−10
−5
0
5
10

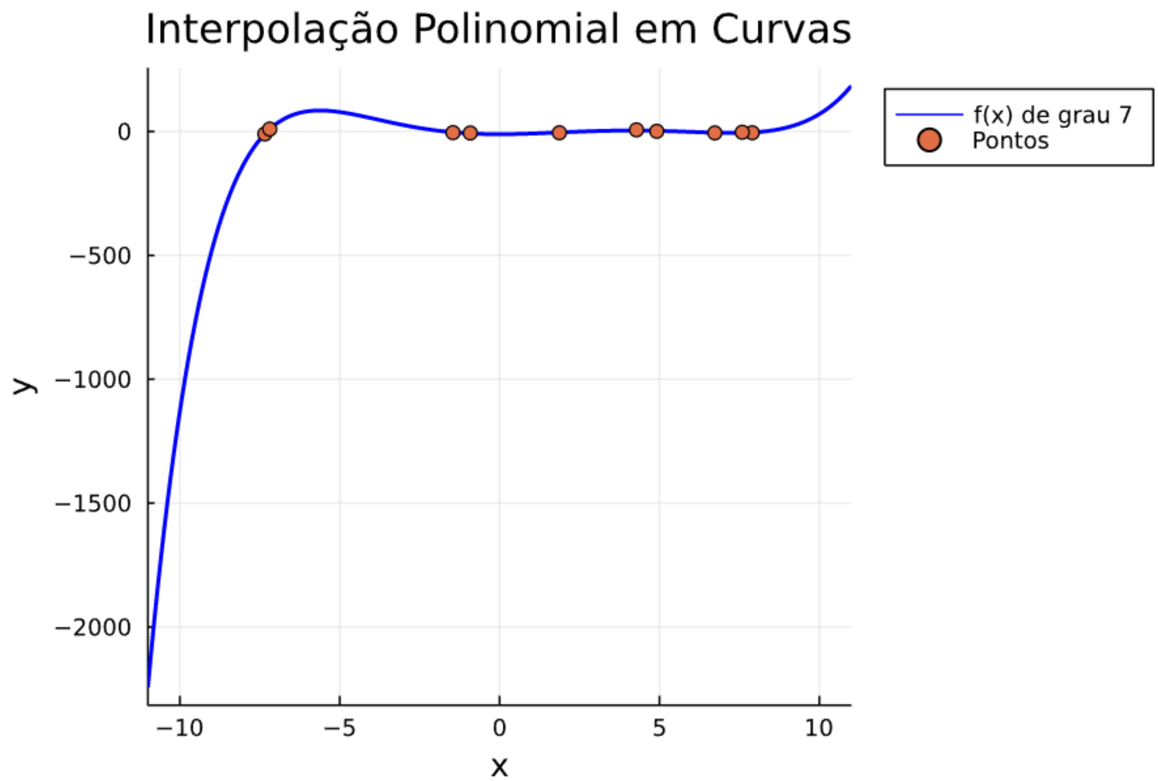

Interpolação Polinomial em Curvas
f(x) de grau 2
Pontos
y
x
10
5
0
−5
−10
−10
−5
0
5
10



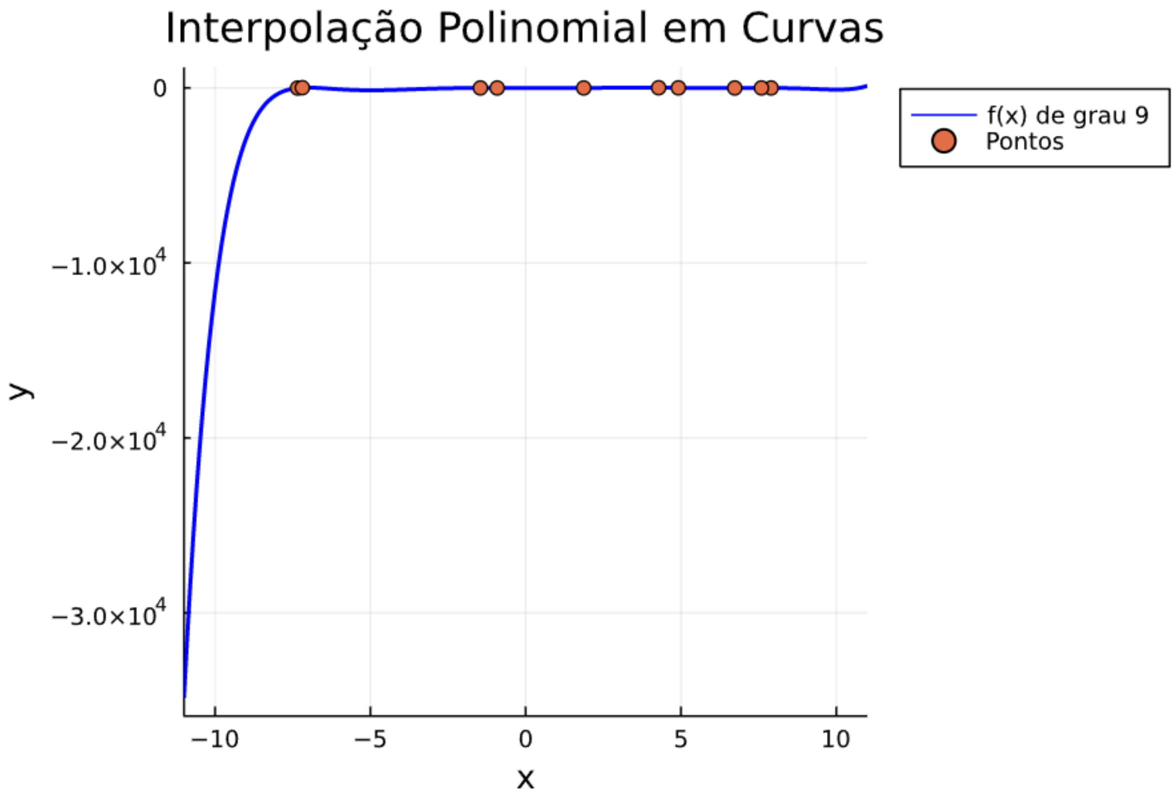
Interpolação Polinomial em Curvas
f(x) de grau 3
Pontos
y
x
20
10
0
−10
−20
−10
−5
0
5
10



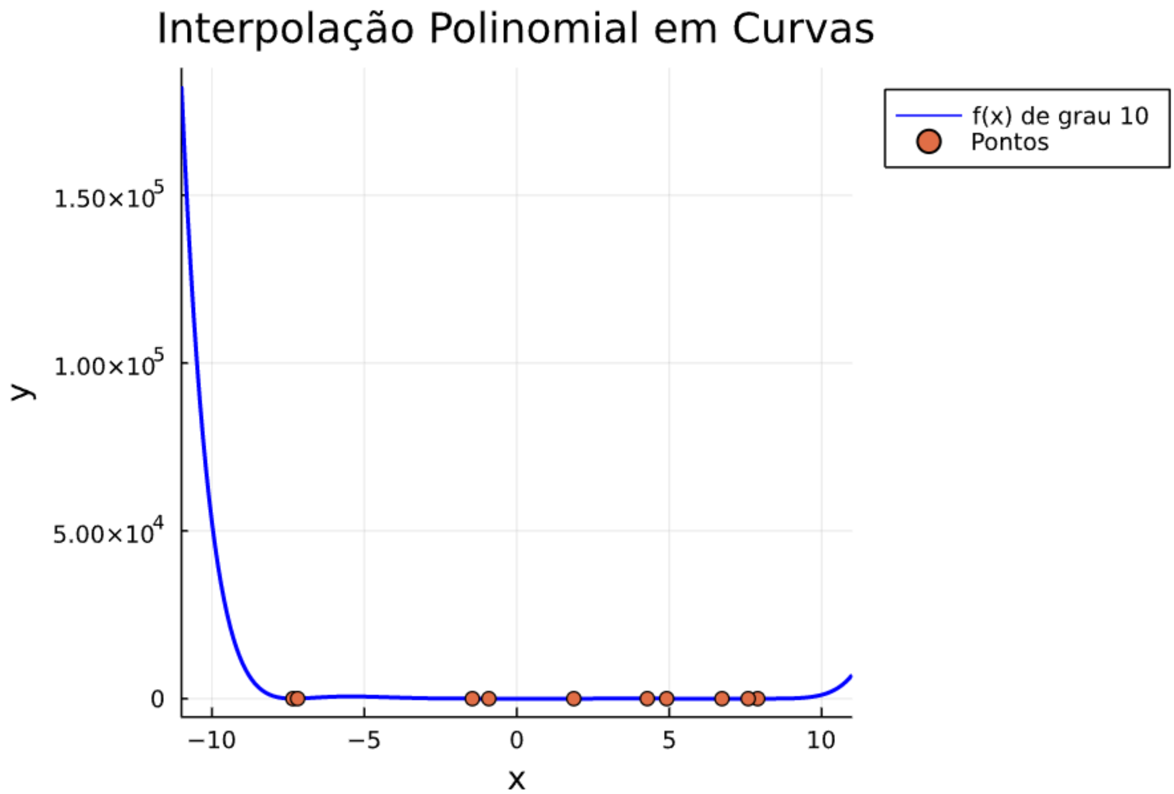
Interpolação Polinomial em Curvas
f(x) de grau 5
Pontos
y
x
250
0
−250
−500
−750
−1000
−1250
−10
−5
0
5
10



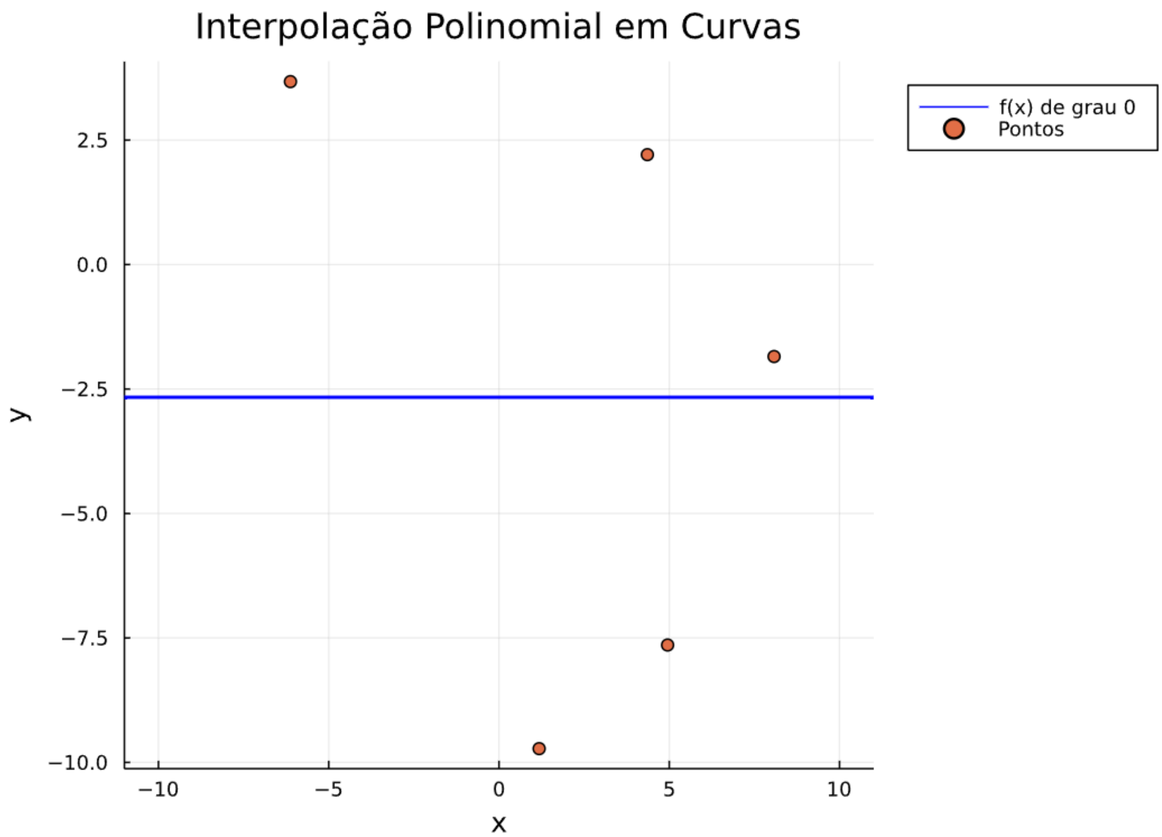
Interpolação Polinomial em Curvas
f(x) de grau 7
Pontos
0
−500
−1000
−1500
−2000
y
−10
−5
0
5
10
x

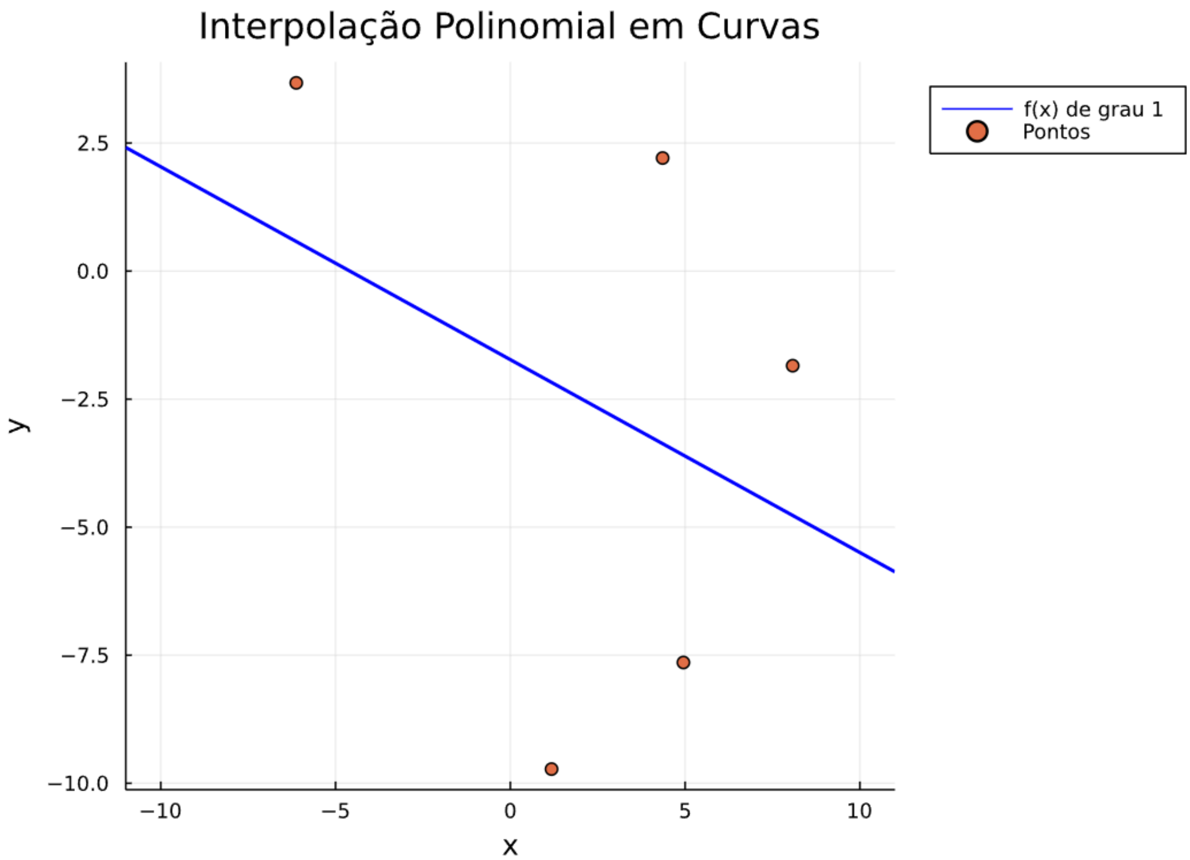


Interpolação Polinomial em Curvas
f(x) de grau 9
Pontos
0
−1.0×10^4
−2.0×10^4
−3.0×10^4
y
−10
−5
0
5
10
x

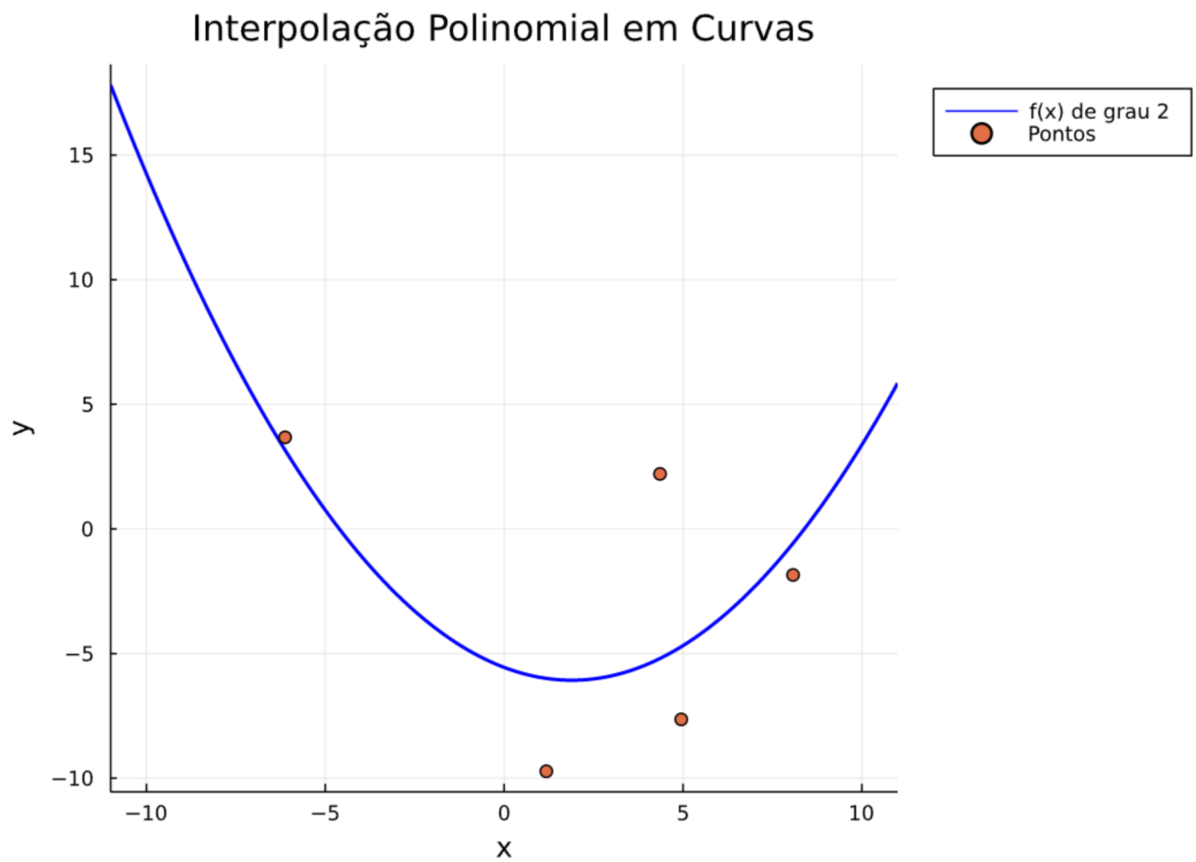


Interpolação feita com o conjunto de pontos B2:

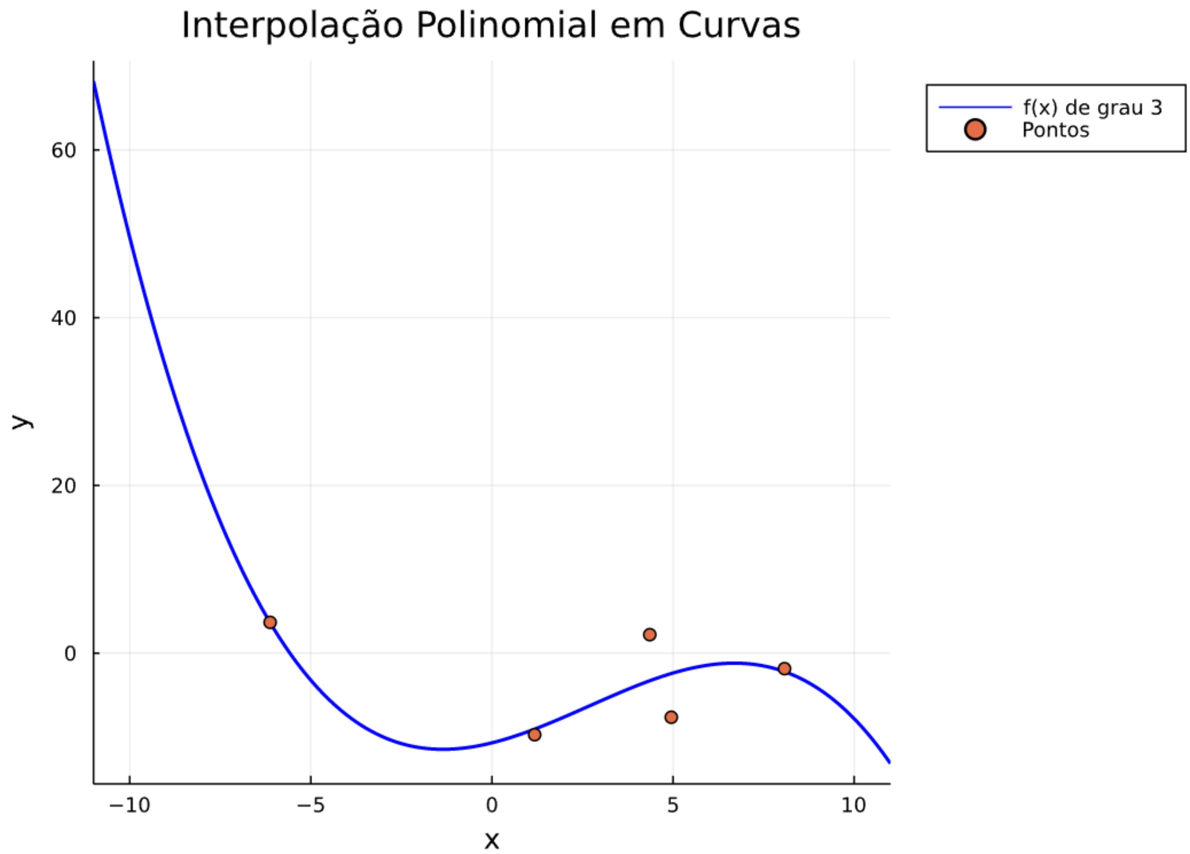


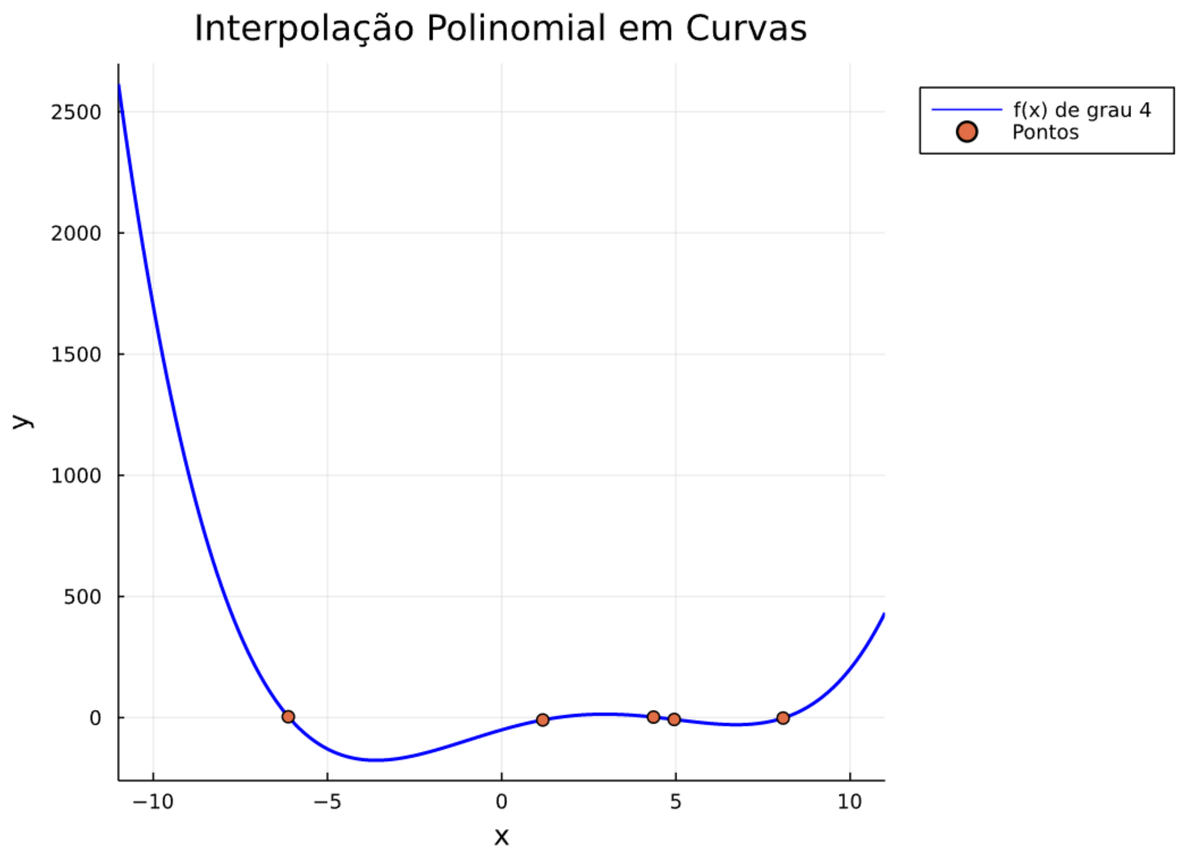


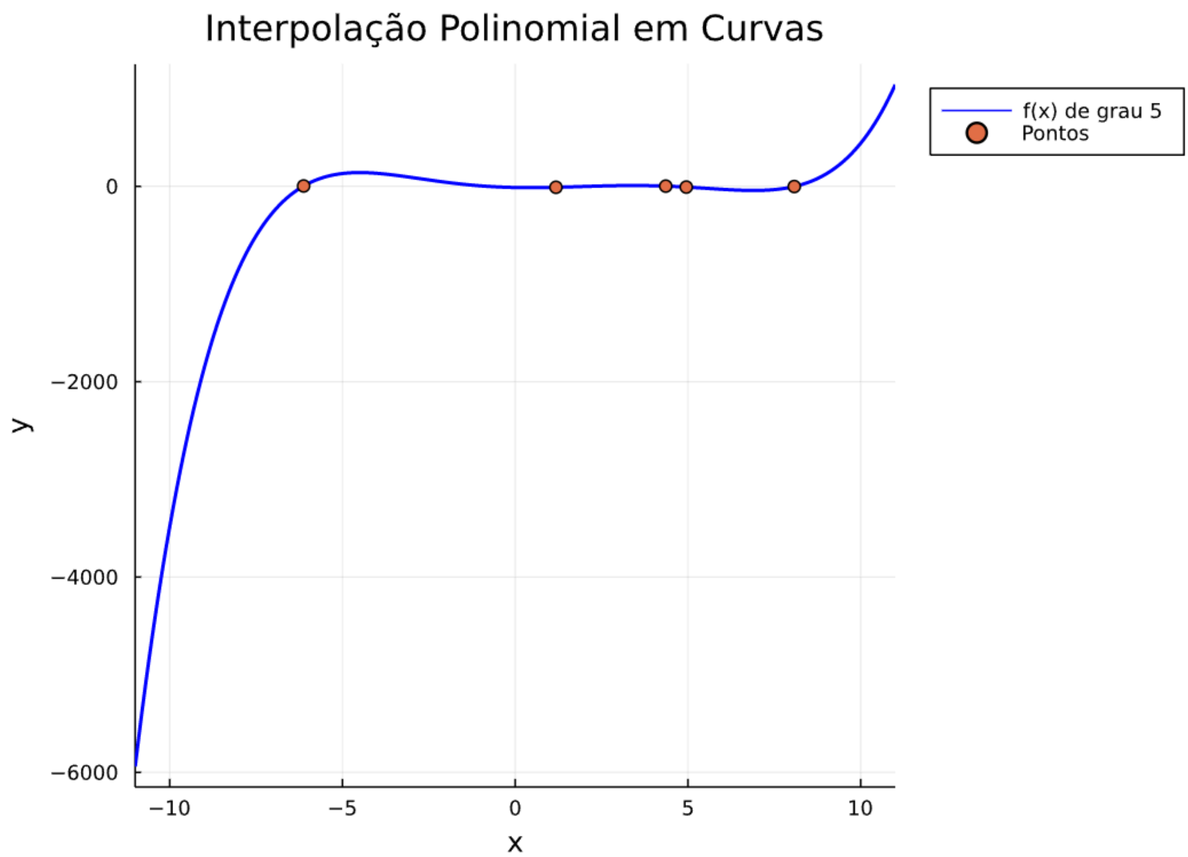
Interpolação Polinomial em Curvas
f(x) de grau 1
Pontos
y
x
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−10
−5
0
5
10



Interpolação Polinomial em Curvas
f(x) de grau 2
Pontos
y
x
15
10
5
0
−5
−10
−10
−5
0
5
10


Interpolação Polinomial em Curvas
f(x) de grau 3
Pontos
60
40
20
0
y
−10
−5
0
5
10
x


Interpolação Polinomial em Curvas
f(x) de grau 4
Pontos
2500
2000
1500
1000
500
0
y
−10
−5
0
5
10
x



**Grupo de pontos B1, B2, C1 e C2 podem ser vistos no arquivo .txt que enviarei junto desse pdf.*

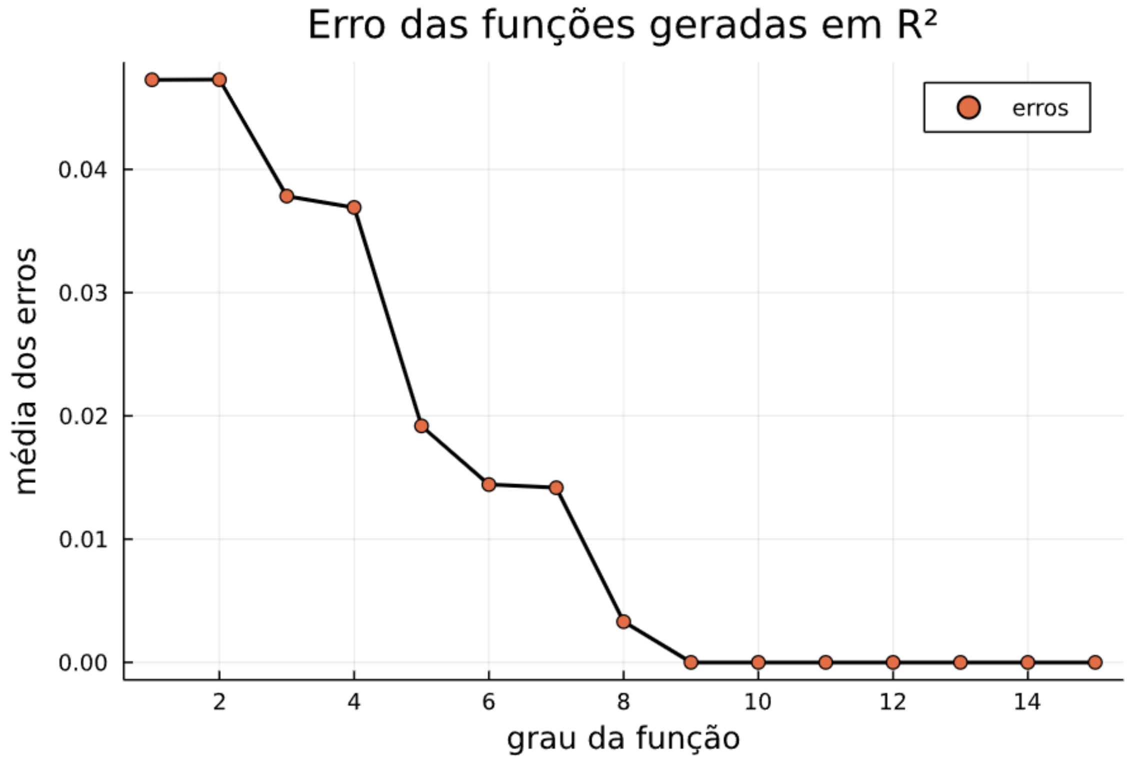
À medida que o grau vai elevando, dá para visualmente perceber ou entender que a função parece estar ficando cada vez mais próxima dos pontos, diminuindo a distância entre eles e a própria função. Isso de fato é um comportamento padrão e faz sentido! Afinal, o número de restrições (ou o número de pontos dados) permanece o mesmo, porém conforme o grau vai aumentando, vamos ganhando mais variáveis que podemos utilizar para resolver o sistema.

Como as variáveis podem assumir quaisquer valores, podemos entender elas como facilitadoras para nós. Isso cria uma matriz A "larga", com muito mais variáveis que restrições, facilitando o encontro de uma solução mais satisfatória ao sistema em questão. O oposto seria, por exemplo, matrizes "altas", onde se tem muitas restrições e poucas variáveis, dificultando o sistema de ter solução exata e exigindo a solução aproximada do mínimos quadrados.

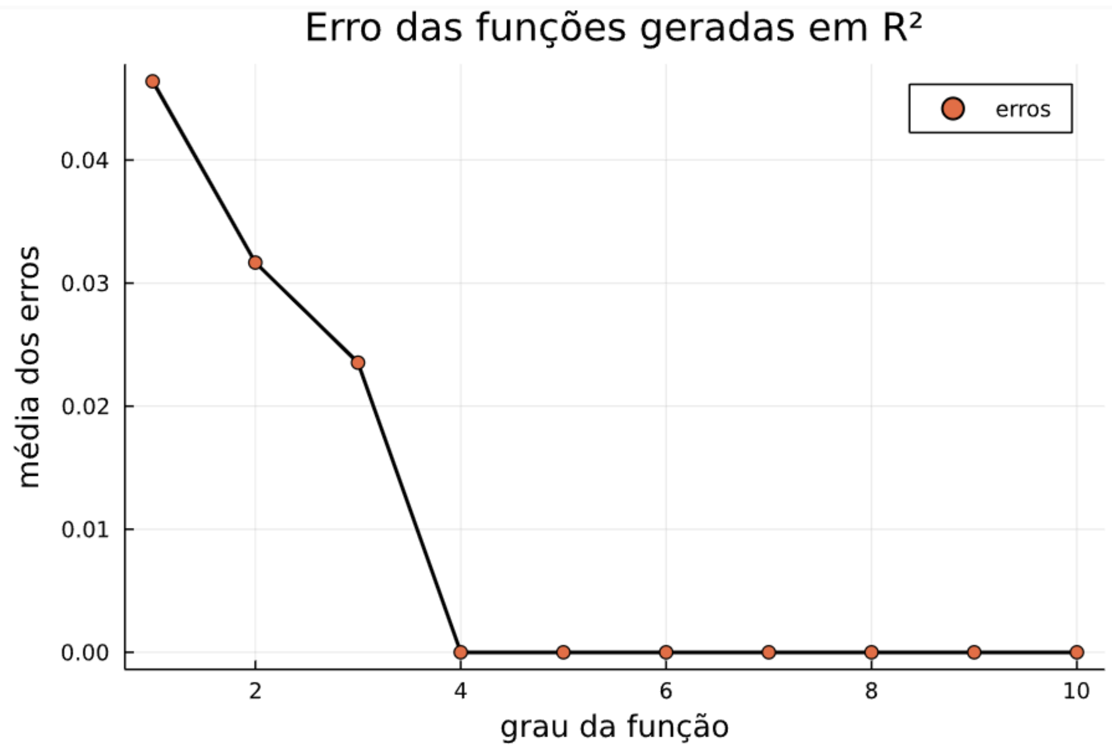
Abaixo temos um gráfico de grau por erro que mostra bem como a média de erro despenca conforme o grau aumenta:

2.b. Erro

Erros com o grupo de pontos B1:



Erros com o grupo de pontos B2:



O erro foi calculado do seguinte modo:

Peguei os pontos dados e comparei os valores de p(x) e o y do ponto naquele x, o erro vinha do valor em módulo da diferença entre y no ponto x e p(x), ou seja, $|y_k - p(x_k)|$ com $0 \leq k \leq n - 1$. Nesse caso, se x fosse 2, faria a diferença em módulo do ponto (2, y) com p(2). Esse teste ocorre com todos os pontos e o erro final é a soma de todas essas distâncias dividida pelo número de pontos. Em adicional, eu dividi esse valor por 100 para passar para o gráfico.

O legal é notar que a maior despencada costuma ocorrer no grau n-1, onde n seria o número de pontos dados. Esse seria o exato momento em que temos uma matriz quadrada, ou seja, mesmo número de restrições e variáveis. Nesse caso teremos uma matriz de dimensões, por exemplo, rxr e exatos r pivôs, pois cada coluna dessa matriz de Vandermonde seria linearmente independente, pois tendo inúmeros valores distintos de x, não haveria combinação possível que te levasse de $[x_0\, x_1\, x_2 \ldots\, x_{r-1}]$ para $[x_0^{\ 2}\, x_1^{\ 2}\, x_2^{\ 2} \ldots\, x_{r-1}^{\ 2}]$. Já deixaria de ser uma coisa linear.

Apesar de partirmos da ideia de que quanto maior o grau, melhor, podemos muitas vezes facilmente parar nesse grau n-1, pois ele se encontra tão próximo dos pontos e os saltos para os graus seguintes são tão menores, que pode valer mais a pena utilizar essa função interpoladora para qualquer aplicação que você queira fazer com ela. Nesses testes muitas vezes o erro nessas funções chegava a casa de 10^{-10} ou até menos! Sem contar que é muito melhor computacionalmente trabalhar com um grau menor, pois em geral teríamos menos operações para realizar na hora de calcular um p(x) qualquer, por exemplo.

3. Interpolação polinomial em R^3:

Aqui as coisas ficam um pouco mais divertidas.

Como seria uma função em R^3? E pior, como seria uma função de grau n em R^3?

Diferente de R^2, agora temos duas variáveis sendo passadas, ou seja, temos algo como f(x, y) para alguma função f. E para definirmos a "cara" de uma função nesse nosso novo mundo, temos que entender como lidar com x e y.

Por exemplo, em R^2, uma função de grau 2 podia ser escrita como $f(x) = ax^2 + bx + c$, onde x^2 está "variando duas vezes", pois estamos variando x e depois x de novo. Mas em R^3 podemos variar x e depois y, por exemplo, chegando em x^2, y^2 e xy (3) "variando duas vezes". Ou seja, uma função de grau n em R^3 vai ter todas as combinações das duas variáveis que variem o número de vezes do grau que estamos. Em outras palavras, talvez mais intuitivas, queremos todas as combinações possíveis de x e y em que o valor da soma de seus expoentes dêem o nosso grau, note que em (3) todas as combinações tem a soma das potências = 2. Vale ressaltar que consideraremos xy = yx, por exemplo, pois no fim, nesse nosso mundo, são a mesma coisa pela propriedade de comutatividade, portanto quando somados ou subtraídos vão se tornar uma coisa só.

Vale lembrar que isso é um processo recursivo, repetitivo, isto é, se queremos uma função f(x, y) de grau 2, temos que ter todas as combinações que somem 2 na potência, todas que somem 1 e todas que somem 0.*

**Todo número elevado a 0 é 1, então no fim teremos sempre uma constante. Isso ocorre em R^2 também com o c na função de 2 grau, por exemplo, e aqui com o f de f(x, y) no exemplo.*

Assim chegamos na função de grau 2 em R^3:

$f(x, y) = ax^2 + by^2 + cxy + dx + ey + f$

Sobre o sistema Ax = b, a única diferença significativa seria que a matriz de Vandermonde ficaria um pouco mais complicada, pois não teríamos somente as potências de x sozinho, mas também de y e das combinações de x e y
Exemplo:

$$\begin{bmatrix} {x_0}^2 & {y_0}^2 & x_0 y_0 & x_0 & y_0 & 1 \\ {x_1}^2 & {y_1}^2 & x_1 y_1 & x_1 & y_1 & 1 \\ {x_2}^2 & {y_2}^2 & x_2 y_2 & x_2 & y_2 & 1 \\ {x_3}^2 & {y_3}^2 & x_3 y_3 & x_3 & y_3 & 1 \end{bmatrix}$$

Matriz de Vandermonde de um polinômio p(x, y) de grau 2.

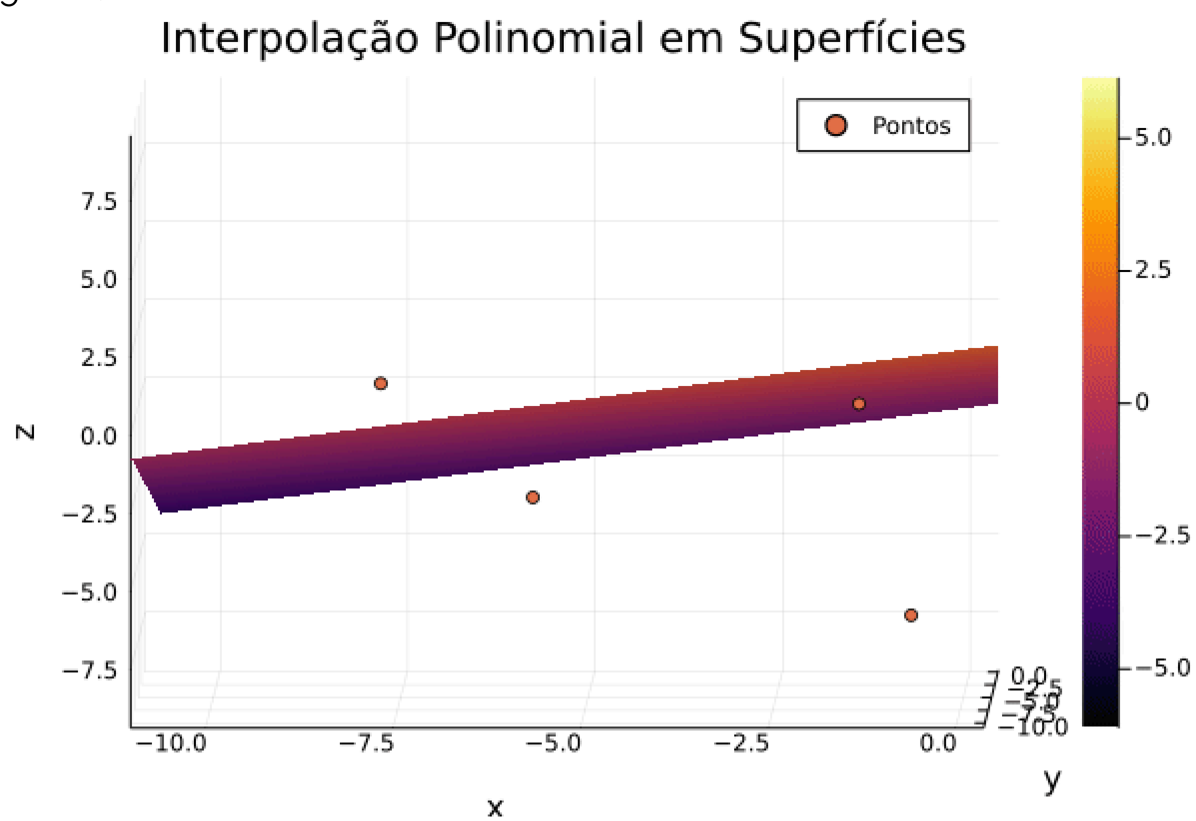
Só com isso já dá pra notar que o número de coeficientes vai crescer mais rápido que em R^2 (ainda vai piorar, acredite) e de fato cresce!

Agora a parte legal que é os gráficos:

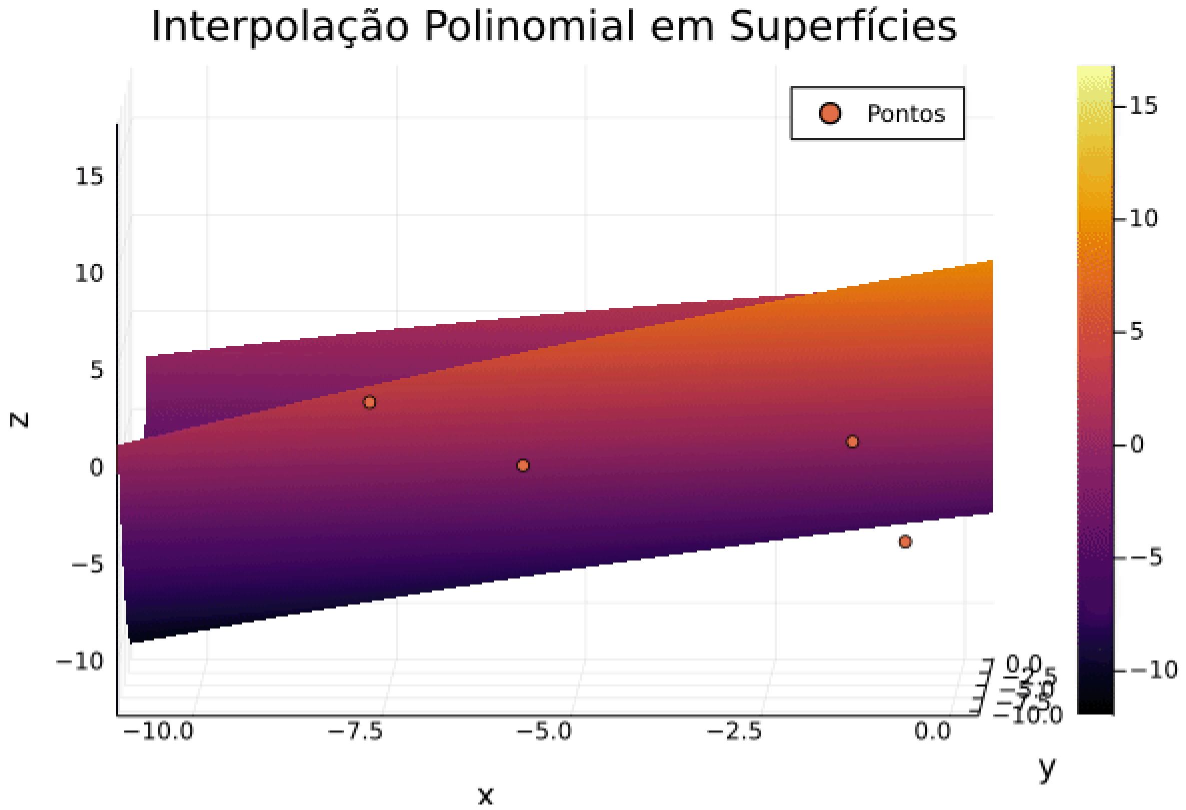
3.a. Gráficos

Interpolações feitas com o conjunto de pontos C1:

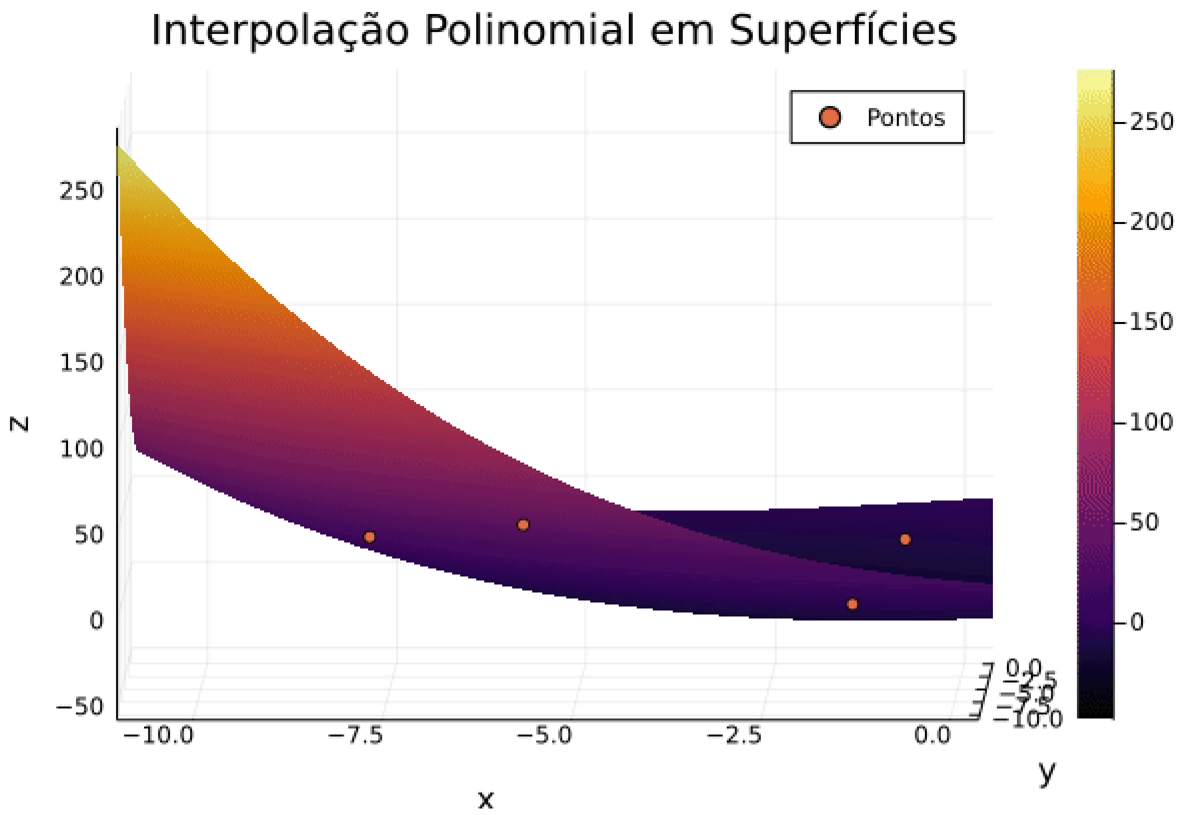
grau 1:



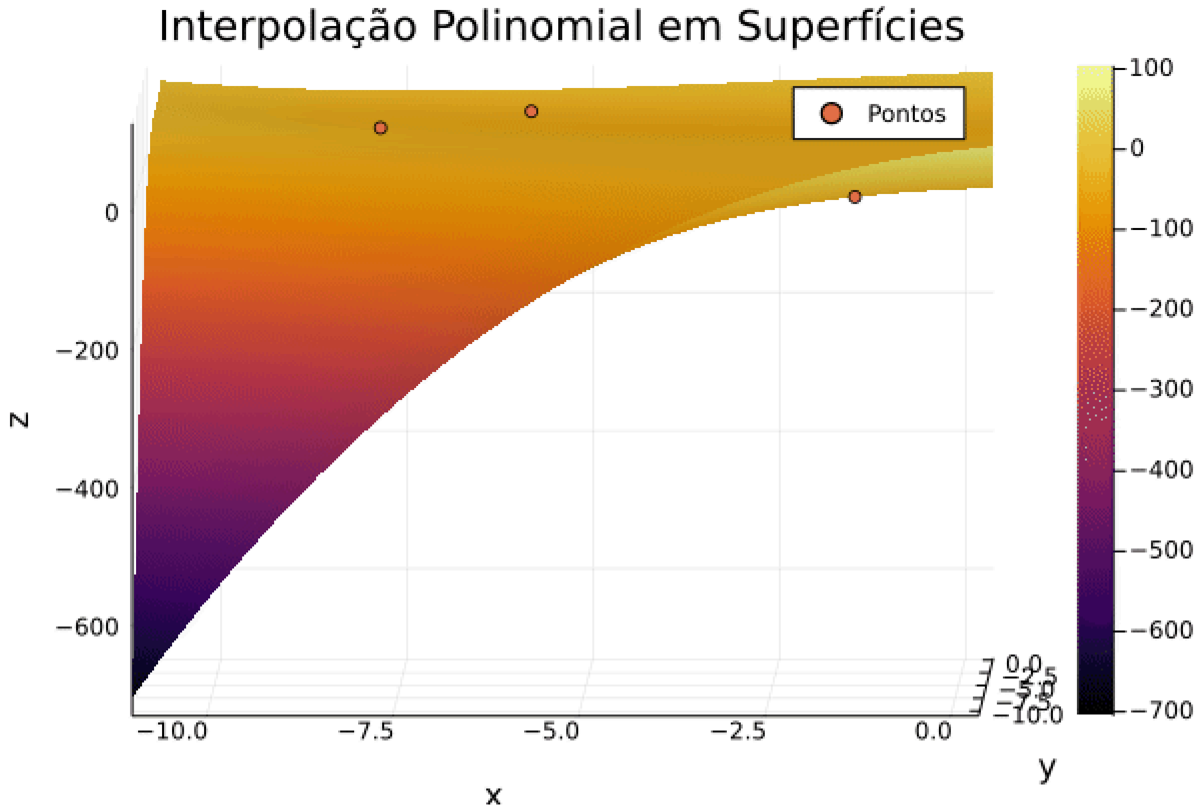
grau 2:


Interpolação Polinomial em Superfícies
Pontos
z
x
y
15
10
5
0
−5
−10
−10.0
−7.5
−5.0
−2.5
0.0

grau 3:

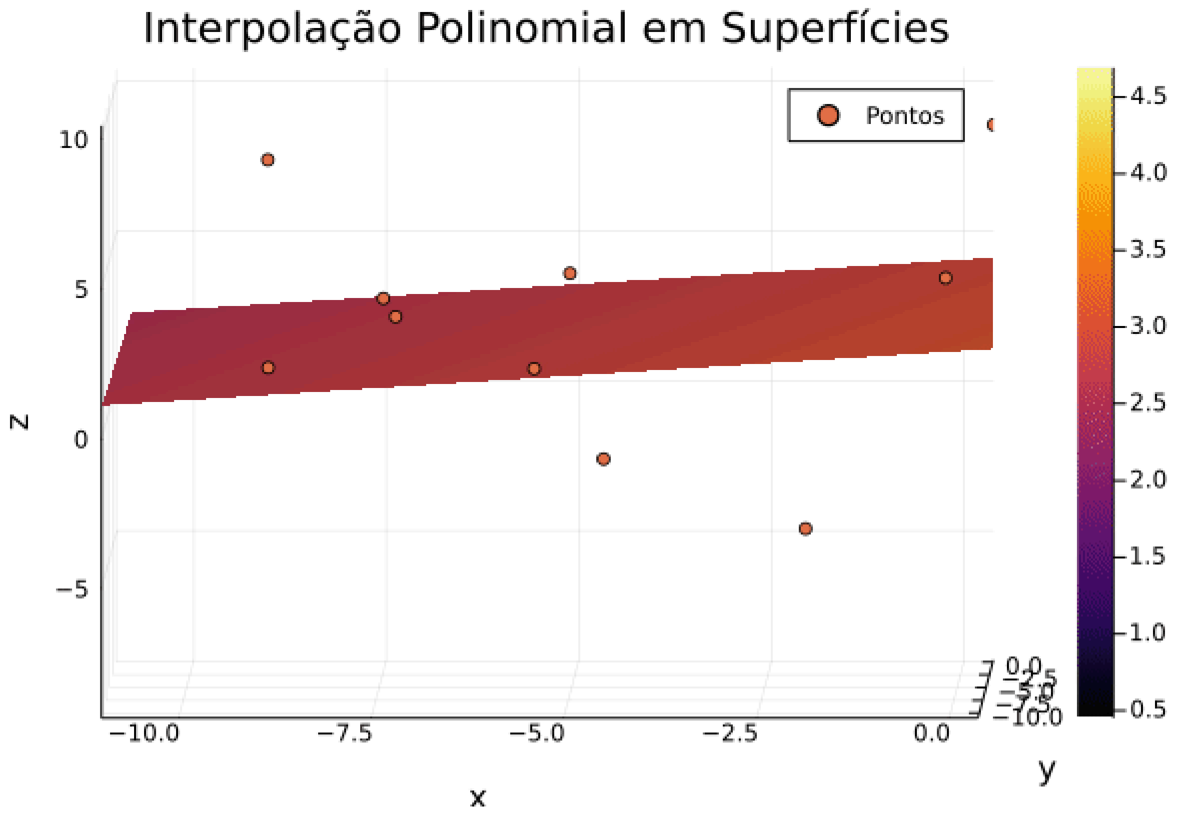

Interpolação Polinomial em Superfícies
Pontos
z
x
y
250
200
150
100
50
0
−50
−10.0
−7.5
−5.0
−2.5
0.0

grau 4:

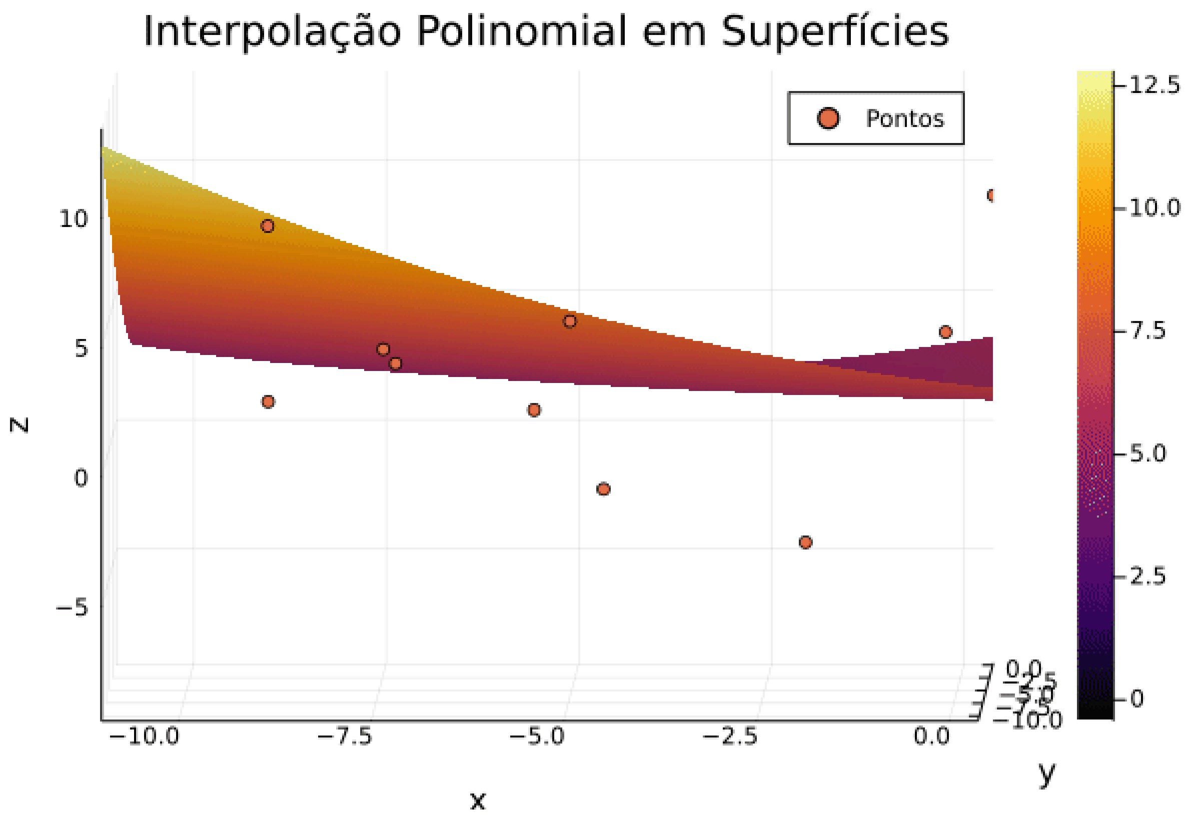


Interpolações feitas com o conjunto de pontos C2:

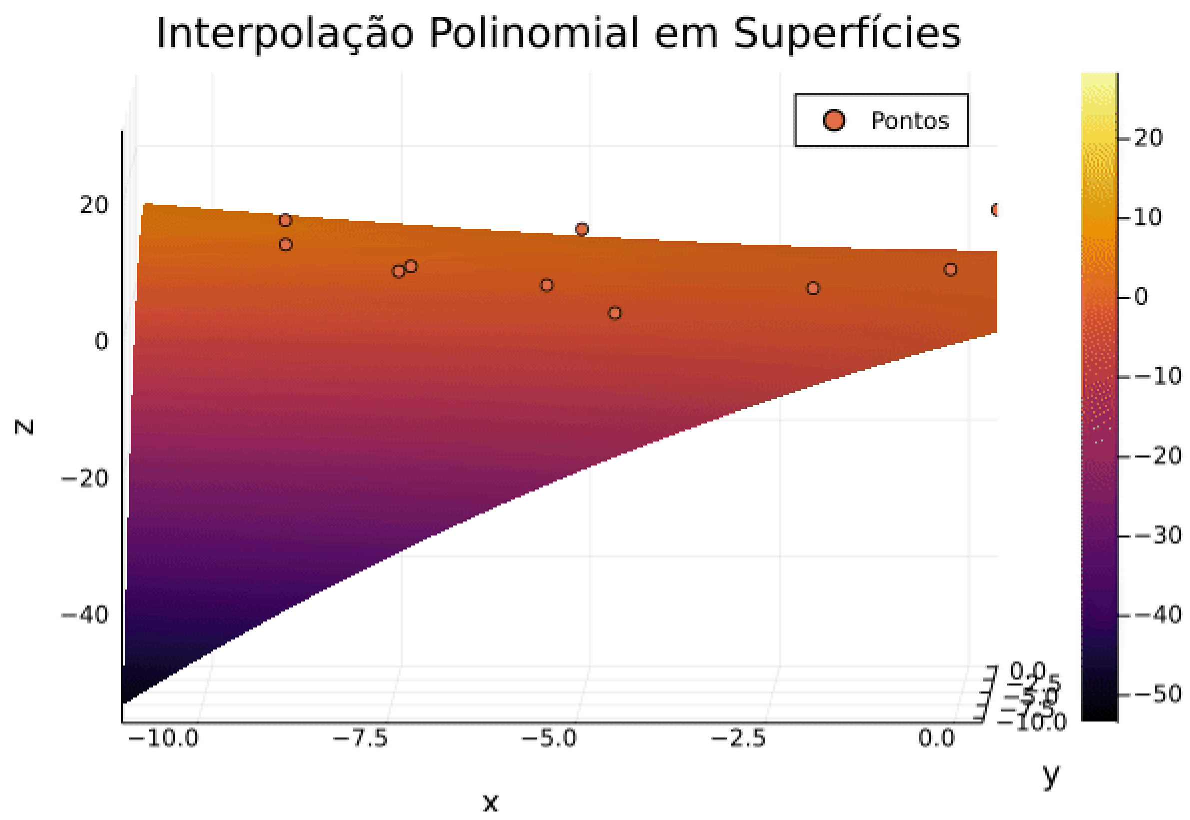
grau 1:


Interpolação Polinomial em Superfícies
Pontos
z
x
y

grau 2:

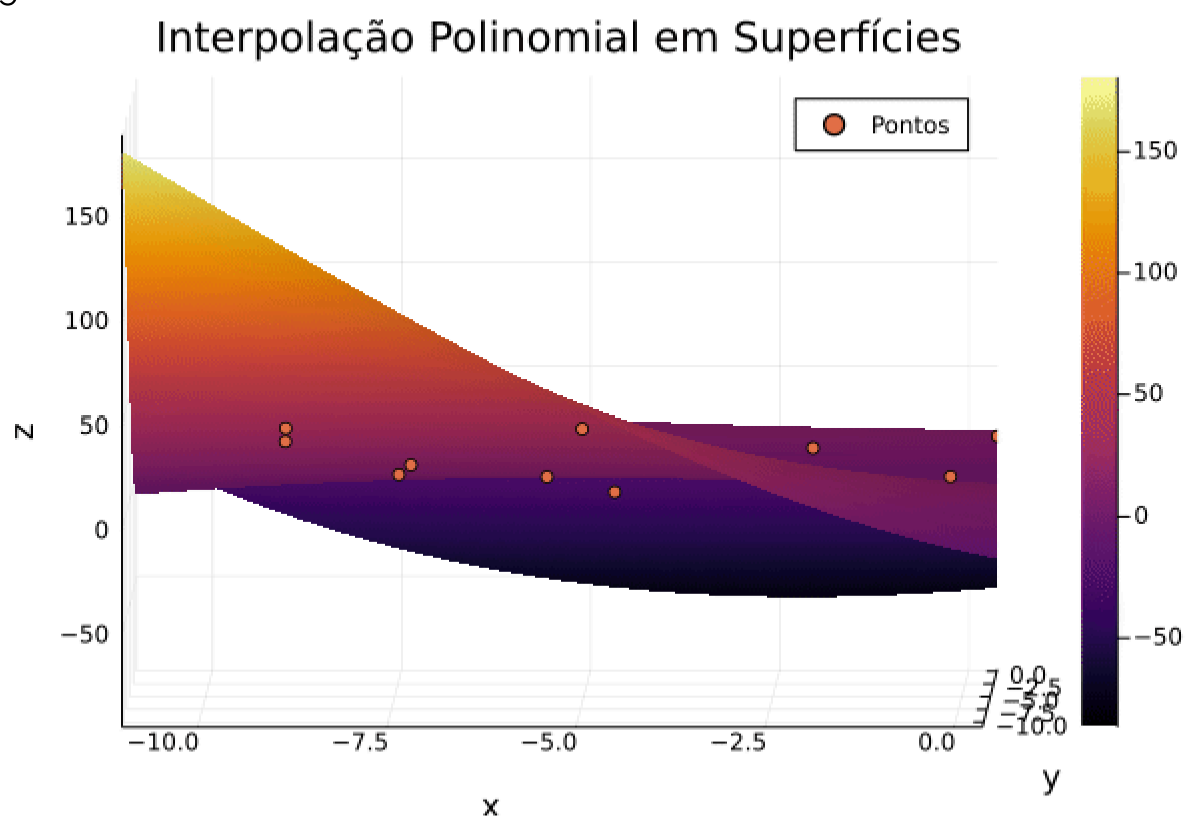

Interpolação Polinomial em Superfícies
Pontos
z
x
y

grau 3:



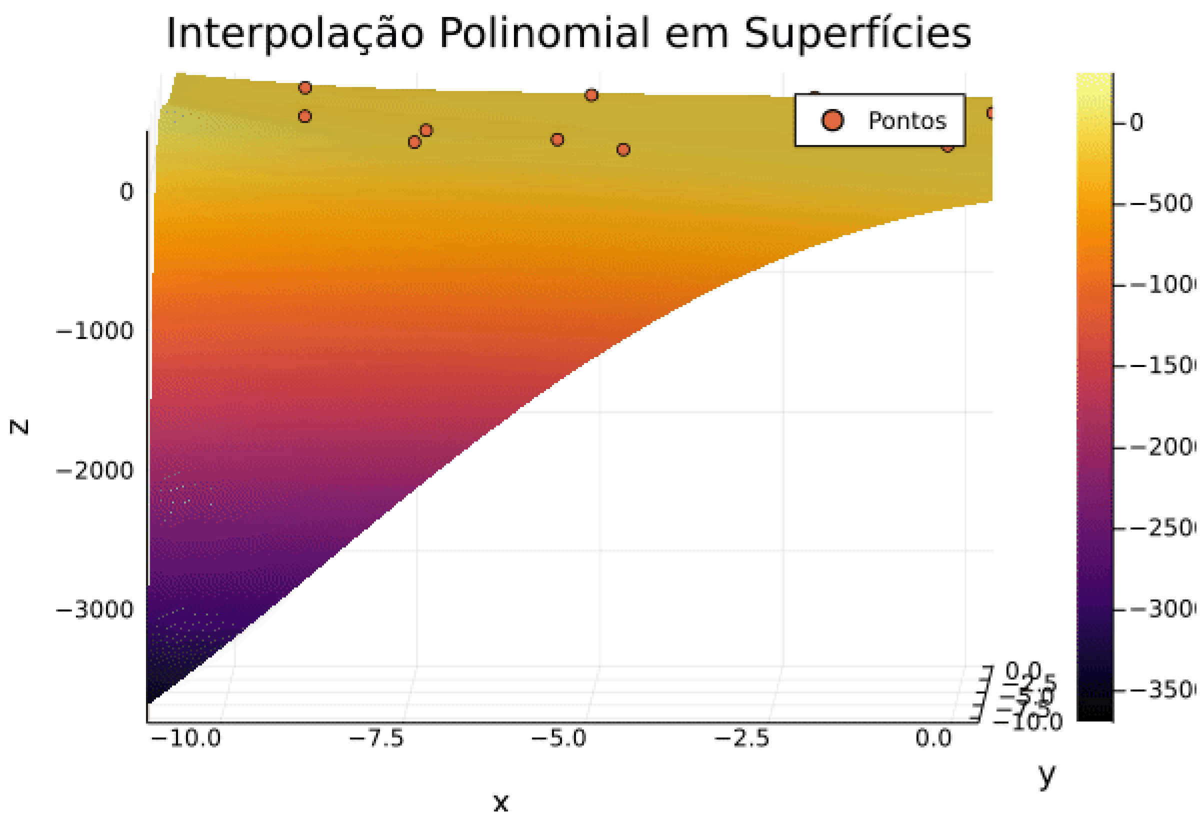
Interpolação Polinomial em Superfícies
Pontos
z
x
y
20
0
−20
−40
−10.0
−7.5
−5.0
−2.5
0.0
10
−10
−30
−50

grau 4:



Interpolação Polinomial em Superfícies
Pontos
z
x
y
150
100
50
0
−50
−10.0
−7.5
−5.0
−2.5
0.0

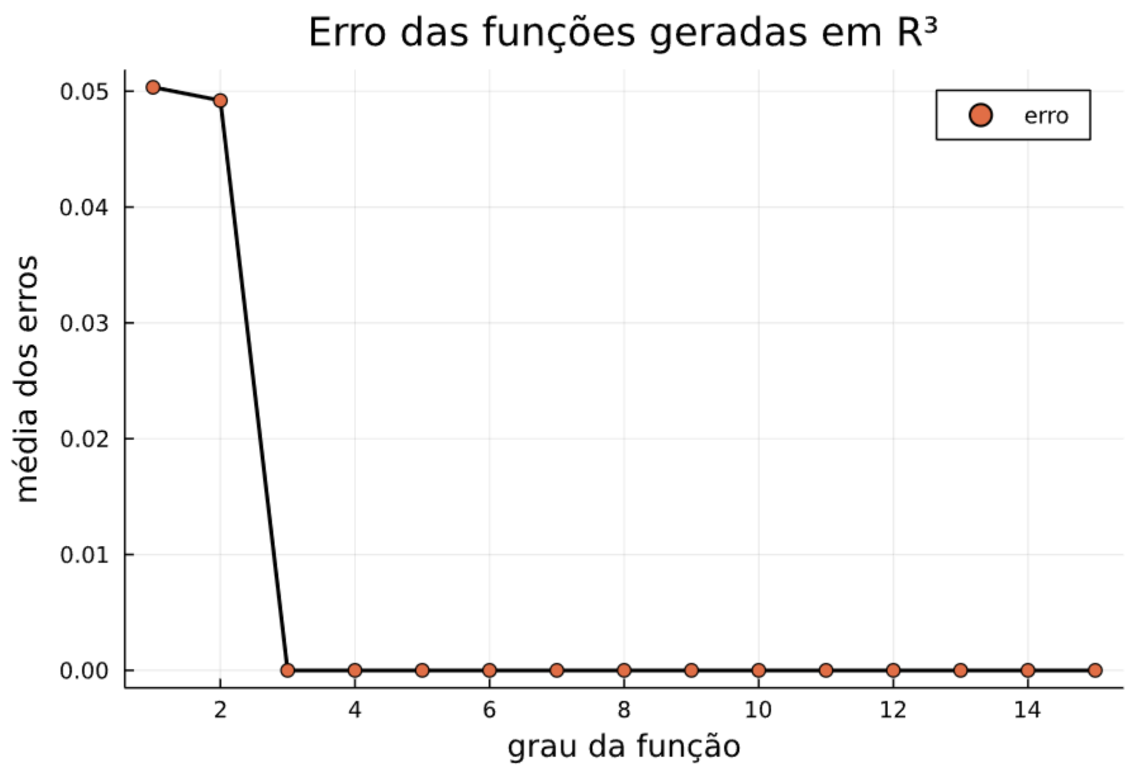
grau 5:



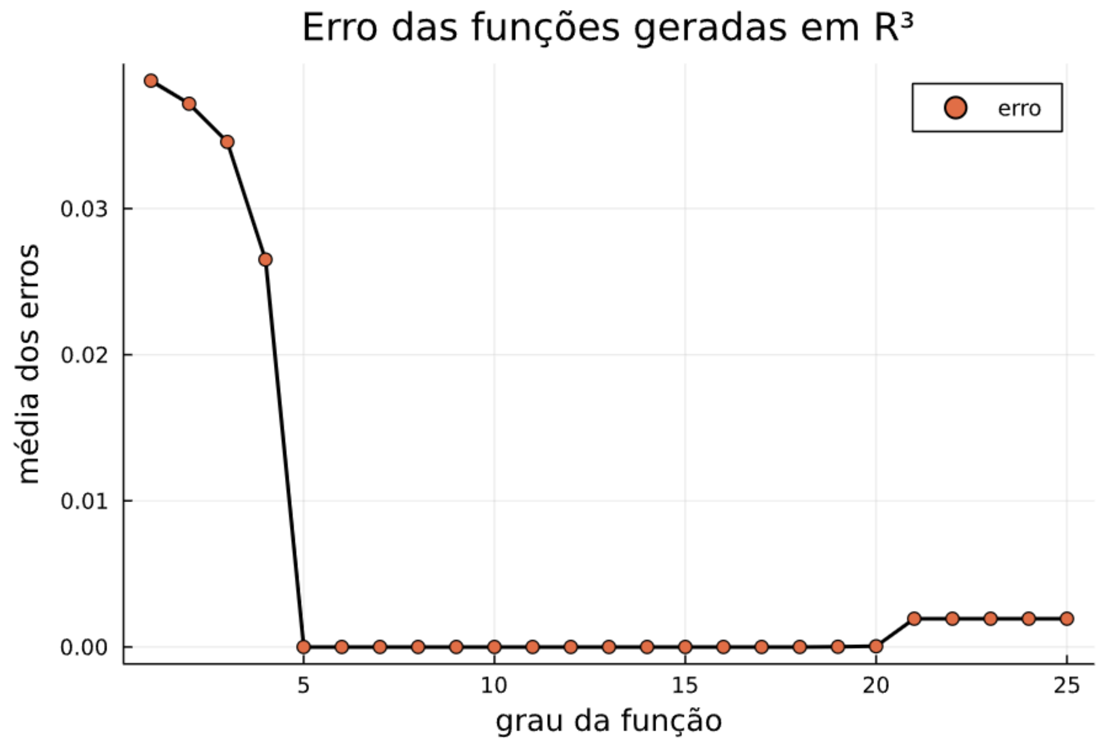
3.b. Erros

O cálculo do erro foi o mesmo usado em R^2, mas como estamos em R^3 as coordenadas agora são do tipo (x, y, z), portanto comparei os z de cada ponto e novamente dividi por 100 para passar para o gráfico abaixo:

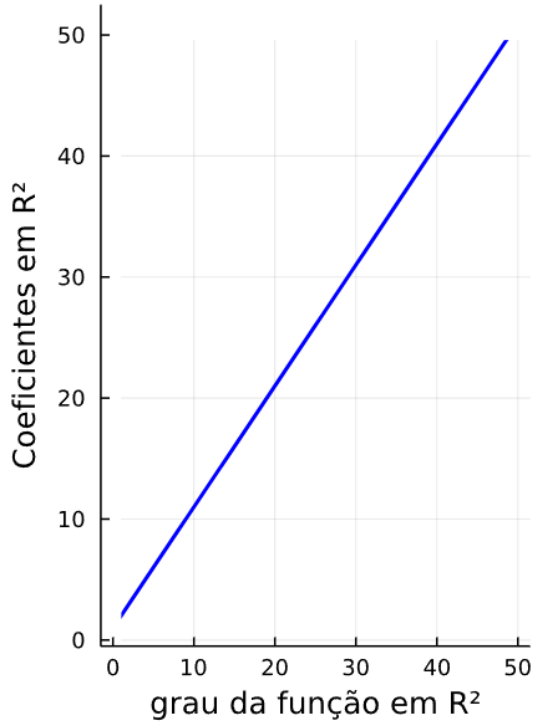
Erros com o grupo de pontos C1:

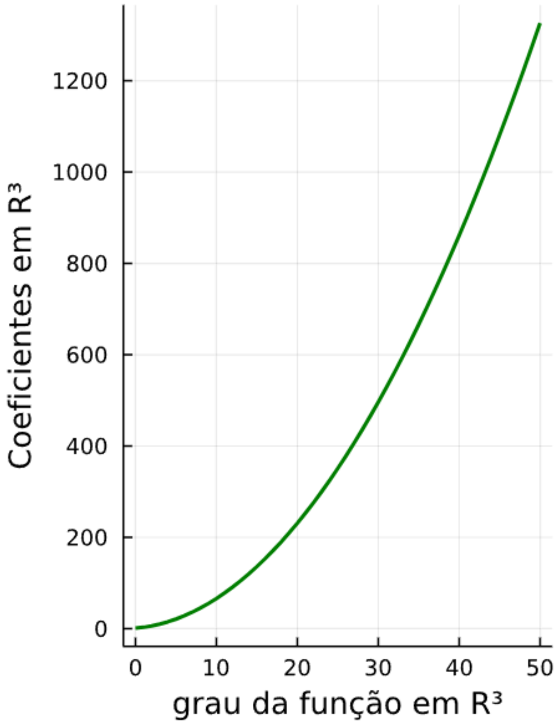


Erros com o grupo de pontos C2:



E agora temos aqui um gráfico comparativo entre R^2 e R^3 referente ao número de coeficientes que cada função tem neles por grau.





Aqui notamos que em R^2 existe um padrão. Uma função de grau n vai ter exatamente n+1 coeficientes, já em R^3 as coisas crescem uma pouco mais.... Desse modo podemos ver um função de grau 50 em R^2 tem 51 coeficientes e em R^3 lindos 1326.

4. Interpolação polinomial em R^4:

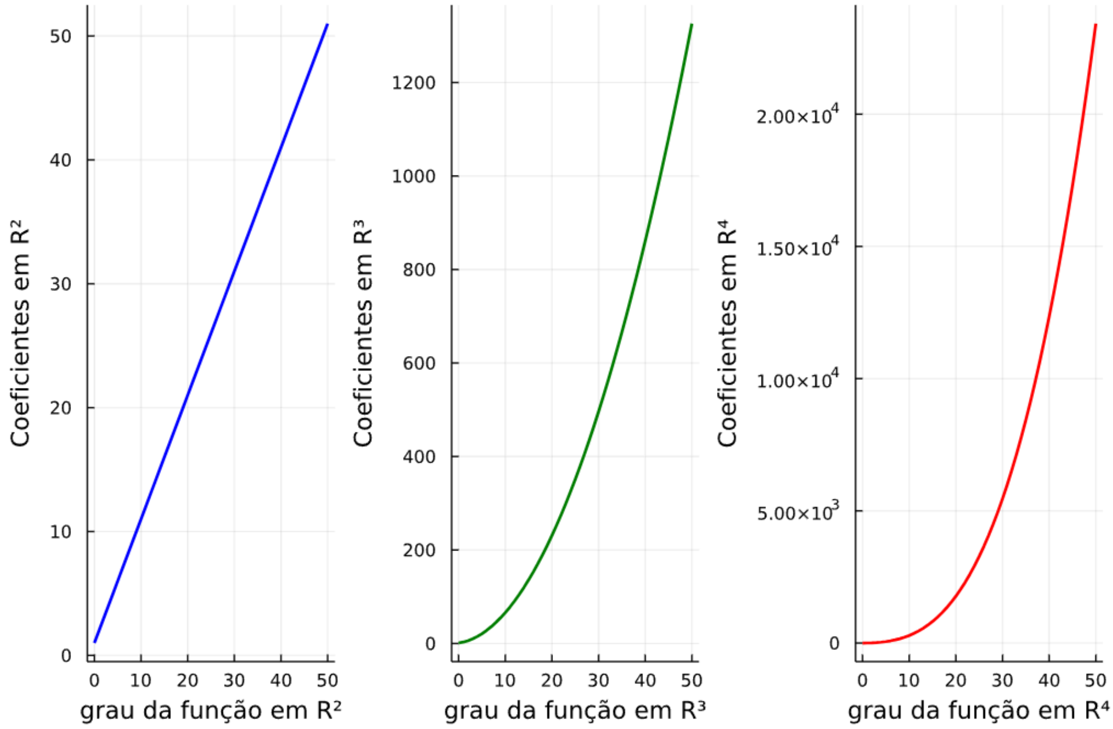
Infelizmente aqui as coisas já começaram a desandar um pouco. Não consegui bolar um algoritmo que de fato fizesse certamente o que gostaria. Na verdade, ao que tudo indica meu código conseguia de fato achar um polinômio, porém a medição de erro parece estar imprecisa ou com um comportamento estranho, creio que seja devido ao modo como programei a função para resolver o polinômio, isto é, dar um x, y e z para o polinômio já conhecido e receber um w. Mas fica de curiosidade a matriz de Vandermonde para um polinômio p(x, y, z) de grau 2:

$$\begin{bmatrix} {x_0}^2 & {y_0}^2 & {z_0}^2 & x_0 y_0 & x_0 z_0 & y_0 z_0 & x_0 & y_0 & z_0 & 1 \\ {x_1}^2 & {y_1}^2 & {z_1}^2 & x_1 y_1 & x_1 z_1 & y_1 z_1 & x_1 & y_1 & z_1 & 1 \\ {x_2}^2 & {y_2}^2 & {z_2}^2 & x_2 y_2 & x_2 z_2 & y_2 z_2 & x_2 & y_2 & z_2 & 1 \\ {x_3}^2 & {y_3}^2 & {z_3}^2 & x_3 y_3 & x_3 z_3 & y_3 z_3 & x_3 & y_3 & z_3 & 1 \end{bmatrix}$$

Porém a lógica seguiria a mesma que nos R anteriores, mas agora teríamos 3 variáveis para checar as combinações e, de novo, você pode imaginar que os coeficientes cresceriam ainda mais rápido.... E de fato crescem! Há uma fórmula de combinatória que encontrei (link nas referências, com a devida prova) que nos diz quantos coeficientes dada função de n variáveis teria em um grau g, dada por:

$$\frac{(n+g)!}{g!(n+g-g)!} = \frac{(n+g)!}{g!n!}$$

Com isso podemos ver quantos coeficientes uma função em R^4 teria em algum grau g, apesar de não conseguirmos calcular seu erro. Abaixo temos 3 gráficos comparando as 3 dimensões R^2, R^3 e R^4 respectivamente:



5. Interpolação polinomial em R^n!:

Assim como não pude terminar com qualidade e segurança em R4, infelizmente não consegui bolar um programa que conseguisse generalizar a ideia de interpolação polinomial em qualquer dimensão que quisesse. Apesar disso, com o que aprendemos podemos prever como seria a matemática. Se tivéssemos n variáveis, e quiséssemos grau g, teríamos de aplicar as ideias já discutidas. A matriz A seria todas as combinações das variáveis que somassem o grau g, o grau g-1, até o grau 0, sendo que em cada linha teríamos isso aplicado às variáveis de um único ponto. Como discutido antes, cada linha da matriz de Vandermonde remete a um ponto em específico. O vetor x seria os coeficientes e o b os valores de $f(x_0, x_1, ..., x_{n-1})$ para cada ponto.

Referências:

https://pt.m.wikipedia.org/wiki/Interpola%C3%A7%C3%A3o_polinomial
https://pt.m.wikipedia.org/wiki/Matriz_de_Vandermonde
https://www-users.cse.umn.edu/~olver/n_/mv.pdf*
https://math.stackexchange.com/questions/380116/number-of-coefficients-of-multivariable-polynomial

*Acabei não aproveitando muito esse, mas parece bem interessante.