,,

# EP2 - Laboratório de Métodos Numéricos
## MAC0210

Breno Moura e Tiago Nápoli 27/05/2017

# 1 Introdução

# 2 Cálculo dos polinômios interpoladores

## 2.1 Interpolação bilinear

Na interpolação bilinear, dado um retângulo com vértices $(x_i, y_j)$, $(x_{i+1}, y_j)$, $(x_i, y_{j+1})$,$(x_{i+1}, y_{j+1})$, aproximamos $f(x, y)$, com $x$ e $y$ no interior deste retângulo, por:

$$s_{ij}^L = a_{00} + a_{01}\frac{(x - x_i)}{h_x} + a_{10}\frac{(y - y_i)}{h_y} + a_{11}\frac{(x - x_i)}{h_x}\frac{(y - y_i)}{h_y}$$

A expressão acima pode ser escrita na forma matricial:

$$s_{ij}(x, y)^L = \begin{bmatrix} 1 & \frac{(x-x_i)}{h_x} \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} 1 \\ \frac{(y-y_j)}{h_y} \end{bmatrix}$$

Para a interpolação bilinear temos seguintes condições de interpolação:

$$s_{ij}^L(x_i, y_j) = f(x_i, y_j)$$
$$s_{ij}^L(x_i, y_{j+1}) = f(x_i, y_{j+1})$$
$$s_{ij}^L(x_{i+1}, y_j) = f(x_{i+1}, y_j)$$
$$s_{ij}^L(x_{i+1}, y_{j+1}) = f(x_{i+1}, y_{j+1})$$

A forma matricial destas condições são:

$$s_{ij}^L(x_i, y_j)^L = \begin{bmatrix} 1 & \frac{(x_i-x_i)}{h_x} \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} 1 \\ \frac{(y_j-y_j)}{h_y} \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$s_{ij}^L(x_{i+1}, y_j)^L = \begin{bmatrix} 1 & \frac{(x_{i+1}-x_i)}{h_x} \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} 1 \\ \frac{(y_j-y_j)}{h_y} \end{bmatrix} = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$s_{ij}^L(x_i, y_{j+1})^L = \begin{bmatrix} 1 & \frac{(x_i-x_i)}{h_x} \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} 1 \\ \frac{(y_{j+1}-y_j)}{h_y} \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$s_{ij}^L(x_{i+1}, y_{j+1})^L = \begin{bmatrix} 1 & \frac{(x_{i+1}-x_i)}{h_x} \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} 1 \\ \frac{(y_{j+1}-y_j)}{h_y} \end{bmatrix} = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Note que podemos escrever essas expressões de uma forma mais compacta da seguinte maneira (usando também as condições de interpolação):

$$\begin{bmatrix} f(x_i, y_j) & f(x_i, y_{j+1}) \\ f(x_{i+1}, y_j) & f(x_{i+1}, y_{j+1}) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

Logo, podemos calcular facilmente os coeficientes de $s_{ij}^L(x, y)$:

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} f(x_i, y_j) & f(x_i, y_{j+1}) \\ f(x_{i+1}, y_j) & f(x_{i+1}, y_{j+1}) \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}^{-1}$$

## 2.2 Interpolação bicúbica

Na interpolação bicúbica, além dos valores da função nos vértices do retângulo, também são considerados o valores da derivadas parciais $\frac{\partial f(x,y)}{\partial x}$, $\frac{\partial f(x,y)}{\partial y}$, e o valor da derivada mista $\frac{\partial f(x,y)}{\partial x \partial y}$ nesses pontos. Nesse tipo de interpolação, a função $f(x, y)$ será aproximada por:

$$s_{ij}^C(x, y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} \Big(\frac{x - x_i}{h_x}\Big)^i \Big(\frac{y - y_j}{h_y}\Big)^j$$

Na forma matricial:

$$s_{ij}(x, y)^C = \begin{bmatrix} 1 & \left(\frac{x-x_i}{h_x}\right) & \left(\frac{x-x_i}{h_x}\right)^2 & \left(\frac{x-x_i}{h_x}\right)^3 \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} 1 \\ \left(\frac{y-y_j}{h_y}\right) \\ \left(\frac{y-y_j}{h_y}\right)^2 \\ \left(\frac{y-y_j}{h_y}\right)^3 \end{bmatrix}$$

Note que há 16 coeficientes a serem determinados, então serão necessários também 16 equações. Os coeficientes devem ser tais que $s_{ij}^C$ satisfaz as condições de interpolação:

$$\begin{aligned} s_{ij}^C(x_i, y_j) &= f(x_i, y_j) \\ s_{ij}^C(x_i, y_{j+1}) &= f(x_i, y_{j+1}) \\ s_{ij}^C(x_{i+1}, y_j) &= f(x_{i+1}, y_j) \\ s_{ij}^C(x_{i+1}, y_{j+1}) &= f(x_{i+1}, y_{j+1}) \end{aligned}$$

Por enquanto só temos 4 equações, mais 12 são necessárias. Considerando as derivadas parciais teremos mais 8 equações:

$$\begin{aligned} \partial_x s_{ij}^C(x_i, y_j) &= \partial_x f(x_i, y_j) \\ \partial_y s_{ij}^C(x_i, y_j) &= \partial_y f(x_i, y_j) \\ \partial_x s_{ij}^C(x_{i+1}, y_j) &= \partial_x f(x_{i+1}, y_j) \\ \partial_y s_{ij}^C(x_{i+1}, y_j) &= \partial_y f(x_{i+1}, y_j) \\ \partial_x s_{ij}^C(x_i, y_{j+1}) &= \partial_x f(x_i, y_{j+1}) \\ \partial_y s_{ij}^C(x_i, y_{j+1}) &= \partial_y f(x_i, y_{j+1}) \\ \partial_x s_{ij}^C(x_{i+1}, y_{j+1}) &= \partial_x f(x_{i+1}, y_{j+1}) \\ \partial_y s_{ij}^C(x_{i+1}, y_{j+1}) &= \partial_y f(x_{i+1}, y_{j+1}) \end{aligned}$$

As últimas 4 equações serão dadas pelas condições impostas pelas derivadas mistas:

$$\partial^2_{xy} s^C_{ij}(x_i, y_j) = \partial^2_{xy} f(x_i, y_j)$$
$$\partial^2_{xy} s^C_{ij}(x_{i+1}, y_j) = \partial^2_{xy} f(x_{i+1}, y_j)$$
$$\partial^2_{xy} s^C_{ij}(x_i, y_{j+1}) = \partial^2_{xy} f(x_i, y_{j+1})$$
$$\partial^2_{xy} s^C_{ij}(x_{i+1}, y_{j+1}) = \partial^2_{xy} f(x_{i+1}, y_{j+1})$$

Com as 16 equações dadas, os coeficientes estão determinados, basta resolver o sistema linear com as equações acima, entretanto, isso não será eficiente. A fim de encontrar um método mais eficiente, pode-se desenvolver as equações dadas. Entretanto, para simplificar os cálculos, usemos as seguintes relações:

$$A = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$\partial_x s_{ij}(x,y)^C = \begin{bmatrix} 0 & \frac{1}{h_x} & \frac{2}{h_x}\left(\frac{x-x_i}{h_x}\right) & \frac{3}{h_x}\left(\frac{x-x_i}{h_x}\right)^2 \end{bmatrix} A \begin{bmatrix} 1 \\ \left(\frac{y-y_j}{h_y}\right) \\ \left(\frac{y-y_j}{h_y}\right)^2 \\ \left(\frac{y-y_j}{h_y}\right)^3 \end{bmatrix}$$

$$\partial_y s_{ij}(x,y)^C = \begin{bmatrix} 1 & \left(\frac{x-x_i}{h_x}\right) & \left(\frac{x-x_i}{h_x}\right)^2 & \left(\frac{x-x_i}{h_x}\right)^3 \end{bmatrix} A \begin{bmatrix} 0 \\ \frac{1}{h_y} \\ \frac{2}{h_y}\left(\frac{y-y_j}{h_y}\right) \\ \frac{3}{h_y}\left(\frac{y-y_j}{h_y}\right)^2 \end{bmatrix}$$

$$\partial^2_{xy} s_{ij}(x,y)^C = \begin{bmatrix} 0 & \frac{1}{h_x} & \frac{2}{h_x}\left(\frac{x-x_i}{h_x}\right) & \frac{3}{h_x}\left(\frac{x-x_i}{h_x}\right)^2 \end{bmatrix} A \begin{bmatrix} 0 \\ \frac{1}{h_y} \\ \frac{2}{h_y}\left(\frac{y-y_j}{h_y}\right) \\ \frac{3}{h_y}\left(\frac{y-y_j}{h_y}\right)^2 \end{bmatrix}$$

Agora, as equações acima podem ser escritas na forma matricial, substituindo $x$ e $y$ na forma matricial de $s^C_{ij}$, resultando em:

$$s_{ij}(x_i, y_j)^C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} A \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad s_{ij}(x_{i+1}, y_j)^C = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} A \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$s_{ij}(x_i, y_{j+1})^C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} A \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad s_{ij}(x_{i+1}, y_{j+1})^C = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} A \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Derivadas parciais em $x$:

$$\partial_x s_{ij}(x_i, y_j)^C = \begin{bmatrix} 0 & \frac{1}{h_x} & 0 & 0 \end{bmatrix} A \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \partial_x s_{ij}(x_{i+1}, y_j)^C = \begin{bmatrix} 0 & \frac{1}{h_x} & \frac{2}{h_x} & \frac{3}{h_x} \end{bmatrix} A \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\partial_x s_{ij}(x_i, y_{j+1})^C = \begin{bmatrix} 0 & \frac{1}{h_x} & 0 & 0 \end{bmatrix} A \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \partial_x s_{ij}(x_{i+1}, y_{j+1})^C = \begin{bmatrix} 0 & \frac{1}{h_x} & \frac{2}{h_x} & \frac{3}{h_x} \end{bmatrix} A \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Derivadas parciais em $y$:

$$\partial_y s_{ij}(x_i, y_j)^C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} A \begin{bmatrix} 0 \\ \frac{1}{h_y} \\ 0 \\ 0 \end{bmatrix}, \quad \partial_y s_{ij}(x_{i+1}, y_j)^C = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} A \begin{bmatrix} 0 \\ \frac{1}{h_y} \\ 0 \\ 0 \end{bmatrix}$$

$$\partial_y s_{ij}(x_i, y_{j+1})^C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} A \begin{bmatrix} 0 \\ \frac{1}{h_y} \\ \frac{2}{h_y} \\ \frac{3}{h_y} \end{bmatrix}, \quad \partial_y s_{ij}(x_{i+1}, y_{j+1})^C = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} A \begin{bmatrix} 0 \\ \frac{1}{h_y} \\ \frac{2}{h_y} \\ \frac{3}{h_y} \end{bmatrix}$$

Derivada mista:

$$\partial^2_{xy} s_{ij}(x_i, y_j)^C = \begin{bmatrix} 0 & \frac{1}{h_x} & 0 & 0 \end{bmatrix} A \begin{bmatrix} 0 \\ \frac{1}{h_y} \\ 0 \\ 0 \end{bmatrix}, \quad \partial^2_{xy} s_{ij}(x_{i+1}, y_j)^C = \begin{bmatrix} 0 & \frac{1}{h_x} & \frac{2}{h_x} & \frac{3}{h_x} \end{bmatrix} A \begin{bmatrix} 0 \\ \frac{1}{h_y} \\ 0 \\ 0 \end{bmatrix}$$

$$\partial^2_{xy} s_{ij}(x_i, y_{j+1})^C = \begin{bmatrix} 0 & \frac{1}{h_x} & 0 & 0 \end{bmatrix} A \begin{bmatrix} 0 \\ \frac{1}{h_y} \\ \frac{2}{h_y} \\ \frac{3}{h_y} \end{bmatrix}, \quad \partial^2_{xy} s_{ij}(x_{i+1}, y_{j+1})^C = \begin{bmatrix} 0 & \frac{1}{h_x} & \frac{2}{h_x} & \frac{3}{h_x} \end{bmatrix} A \begin{bmatrix} 0 \\ \frac{1}{h_y} \\ \frac{2}{h_y} \\ \frac{3}{h_y} \end{bmatrix}$$

A partir dessas equações, e usando as condições de intepolação e as condições aplicadas às derivadas mistas e parciais, podemos criar uma matriz mais compacta, analogamente ao que foi feito para a Interpolação Bilinear. Consultando a bibliografia, temos que:

$$\begin{bmatrix} f(x_i, y_j) & f(x_i, y_{j+1}) & \partial_y f(x_i, y_j) & \partial_y f(x_i, y_{j+1}) \\ f(x_{i+1}, y_j) & f(x_{i+1}, y_{j+1}) & \partial_y f(x_{i+1}, y_j) & \partial_y f(x_{i+1}, y_{j+1}) \\ \partial_x f(x_i, y_j) & \partial_x f(x_i, y_{j+1}) & \partial_{xy} f(x_i, y_j) & \partial_{xy} f(x_i, y_{j+1}) \\ \partial_x f(x_{i+1}, y_j) & \partial_x f(x_{i+1}, y_{j+1}) & \partial_{xy} f(x_{i+1}, y_j) & \partial_{xy} f(x_{i+1}, y_{j+1}) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & \frac{1}{h_x} & 0 & 0 \\ 0 & \frac{1}{h_x} & \frac{2}{h_x} & \frac{3}{h_x} \end{bmatrix} A \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & \frac{1}{h_y} & \frac{1}{h_y} \\ 0 & 1 & 0 & \frac{2}{h_y} \\ 0 & 1 & 0 & \frac{3}{h_y} \end{bmatrix}$$

Por fim, a matriz de coeficientes é calculada eficientemente com:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & \frac{1}{h_x} & 0 & 0 \\ 0 & \frac{1}{h_x} & \frac{2}{h_x} & \frac{3}{h_x} \end{bmatrix}^{-1} \begin{bmatrix} f(x_i, y_j) & f(x_i, y_{j+1}) & \partial_y f(x_i, y_j) & \partial_y f(x_i, y_{j+1}) \\ f(x_{i+1}, y_j) & f(x_{i+1}, y_{j+1}) & \partial_y f(x_{i+1}, y_j) & \partial_y f(x_{i+1}, y_{j+1}) \\ \partial_x f(x_i, y_j) & \partial_x f(x_i, y_{j+1}) & \partial_{xy} f(x_i, y_j) & \partial_{xy} f(x_i, y_{j+1}) \\ \partial_x f(x_{i+1}, y_j) & \partial_x f(x_{i+1}, y_{j+1}) & \partial_{xy} f(x_{i+1}, y_j) & \partial_{xy} f(x_{i+1}, y_{j+1}) \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & \frac{1}{h_y} & \frac{1}{h_y} \\ 0 & 1 & 0 & \frac{2}{h_y} \\ 0 & 1 & 0 & \frac{3}{h_y} \end{bmatrix}$$

# 3 Implementação

## 3.1 Função constroiv

A função constroiv foi feita de forma que recebia os valores Nx, Ny (Quantidade de pontos na malha), Ax, Ay (Pontos iniciais da malha), Bx, By (Pontos finais da malha), M (Uma matrix Nx por Ny por 4, com 3 dimensões sendo que cada tripla (x, y, 1) correspondia aos valores da função nos pontos de f(x, y) na malha, (x, y, 2) correpondia a derivada parcial de x de f(x, y) , (x, y, 3) correspondia a derivada parcial de y de f(x, y) e (x, y, 4) corependia a segunda derivada parcial de x e y de f(x, y) e um valor type que dizia se a interpolação a ser feita era bilinear ou bicúbica. A função retornava uma matriz com matrizes de coeficientes da interpolação (Como mostrada nas seções acima), cada índice da matriz retornada continha uma matriz de coeficientes (que são os coeficientes daquele retângulo da malha).

Caso fosse o caso bilinear multiplicava a matriz dos valores da função nos pontos pelas matrizes que foram deduzidas, como mostrada na seção 1. Caso fosse bicúbica multiplicava a matriz com as o valor das funções e as derivadas nos pontos pela matrizes que foram deduzidas, como mostrada na seção 2.

## 3.2 Função avaliav

A função avaliav foi feita de forma que recebia os valores x, y (o valor dos pontos a serem avaliados), A (A matriz dos coeficientes da interpolação, calculada pela constroiv, como mostrada nas seções acima), Nx, Ny (Quantidade de pontos na malha), Ax, Ay (Pontos iniciais da malha), Bx, By (Pontos finais da malha) e type, o tipo da interpolação a ser calculada. A função avaliav via qual "subretangulo"da minha malha meu ponto se enquadrava, e calculava o valor da interpolação de acordo com o valor dos coeficientes da matriz daquele "subretangulo".

## 3.3 Outras funções

Além da avaliav e da constroiv, foram feitas algumas funções auxiliares, como a que calcula o erro, as funções main (que chama a constroi e avaliav de acordo com uma função passada) e uma função draw, que desenha as funções como pedido no enunciado e uma função Data que gera os dados a serem usados pela função que desenha.

## 3.4 Casos para serem rodados

Como pedido pelo monitor, deixamos entre os arquivos passados alguns casos teste exemplo para serem rodados, estes estão no arquivo TESTE, sendo necessário apenas copiar as linhas indicadas

# 4 Experimentos

Foram feitos diversos experimentos a fim de constatar que as funções implementadas de fato funcionavam. Nas seções a seguir são descritos os objetivos, meios e resultados dos experimentos realizados

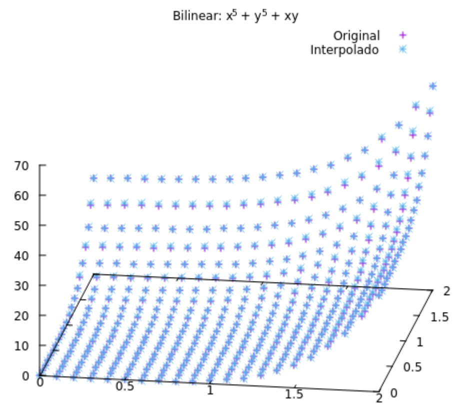
## 4.1 Teste de Interpoalação

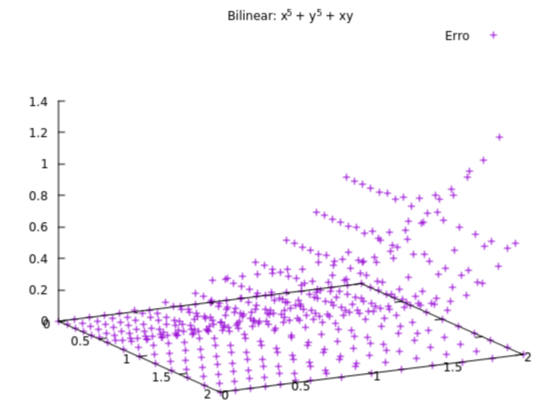
O primeiro teste consistiu em conferir se, para algumas funções inventadas, os polinômios criados satisfaziam as condições de interpolação:

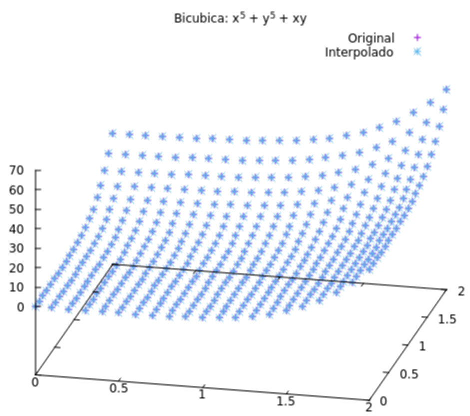
$$s(x_i, y_i) = f(x_i, y_i)$$

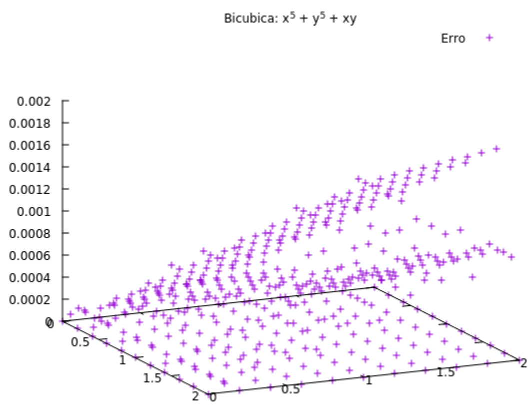
Para isso, a função criada *DataMain* percorre, com um passo determinado (no caso dos experimentos foi 0.1), o *grid* em que a função foi interpolada. Em cada ponto $(x, y)$ visitado nesse processo, calculava-se o valor real da função $f(x, y)$ e o valor aproximado pelo polinômio interpolador. Calculava-se também, assim, o módulo do erro. Com esses dados em mãos, foi possível fazer gráficos do valores da função, da aproximação do polinômio, e do erro. Após os testes com algumas funções, alguns dos resultados estão a seguir:

1. Função: $x^5 + y^5 + xy$. A região de interesse foi $[0, 2] \times [0, 2]$, passo $= 0.1$, com $11 \times 11$ pontos usados para interpolação ($N_x = N_y = 10$).



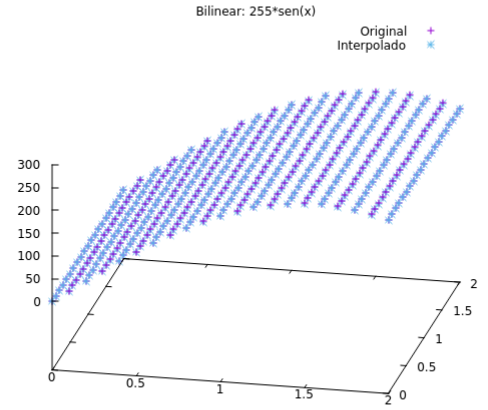



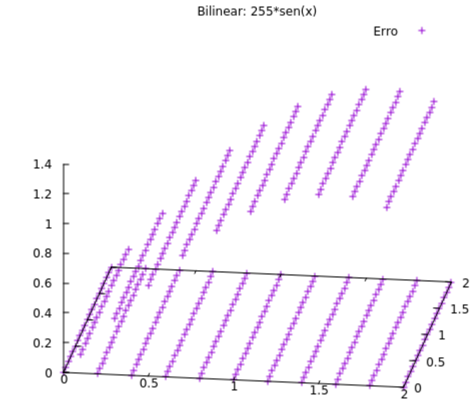



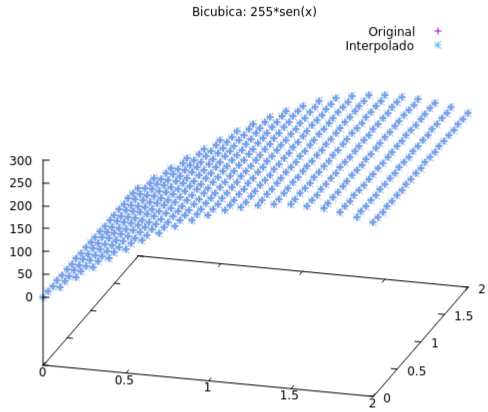



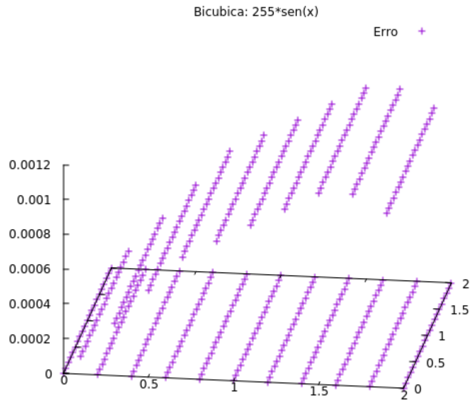

Para a interpolação bicúbica note como a ordem do erro é pequena em relação ao da interpolação bilinear, por isso os pontos calculados por essa interpolação parecem mais próximos à função original em comparação aos calculados pela interpolação bilinear. Isso é ligeiramente observável nos gráficos de cada interpolação. Algo importante a se notar, e esse era outro objetivo do experimento (além de checar se o polinômio interpolador se aproximava ao comportamento da função), é que a condição de interpolação é, de fato, satisfeita, pois é possível ver no gráfico do erro que em diversos pontos $(x, y)$, o valor do erro se anula. Esses são os pontos usados na interpolação.

2. Função $255\sin(x)$. A região de interesse foi $[0, 2] \times [0, 2]$, passo $= 0.1$, com $11 \times 11$ pontos usados para interpolação ($N_x = N_y = 10$).




As observações do item anterior são novamente aplicáveis a este teste.

### 4.2 Comportamento do Erro

O segundo teste realizado consistiu em procurar determinar, na prática, o comportamento do erro. Para esta seção, definamos $N_x = N_y = N$, $h_x = \frac{B_x - A_x}{N_x}$, $h_y = \frac{B_y - A_y}{N_y}$, e $h_x = h_y = h$. Para avaliar o comportamento do erro, variava-se $N$ de modo a diminuir $h$. Os resultados conseguidos foram compilados nas seguintes tabelas.

| Comportamento do Erro: Bilinear, $x^5 + y^5 + xy$ | | |
|---|---|---|
| h | Erro | Fator de redução do erro |
| 1.0000 | 458.43 | |
| 0.5000 | 130.14 | $\approx 0.28$ |
| 0.2500 | 34.94 | $\approx 0.27$ |
| 0.1250 | 8.33 | $\approx 0.24$ |
| 0.0625 | 2.21 | $\approx 0.26$ |

Tabela 1: Comportamento do Erro para Interpolação Bilinear: $x^5 + y^5 + xy$

| Comportamento do Erro: Bicúbica, $x^5 + y^5 + xy$ | | |
|---|---|---|
| h | Erro | Fator de redução do erro |
| 1.0000 | 2.81250 | |
| 0.5000 | 0.17136 | $\approx 0.0609$ |
| 0.2500 | 0.01098 | $\approx 0.0640$ |
| 0.1250 | 0.00067 | $\approx 0.0615$ |
| 0.0625 | 0.00004 | $\approx 0.0636$ |

Tabela 2: Comportamento do Erro para Interpolação Bicúbica: $x^5 + y^5 + xy$

| Comportamento do Erro: Bilinear, $255\sin(x)$ | | |
|---|---|---|
| h | Erro | Fator de redução do erro |
| 1.0000 | 31.13 | |
| 0.5000 | 7.60 | $\approx 0.24$ |
| 0.2500 | 1.90 | $\approx 0.25$ |
| 0.1250 | 0.47 | $\approx 0.25$ |
| 0.0625 | 0.11 | $\approx 0.25$ |

Tabela 3: Comportamento do Erro para Interpolação Bilinear: $255\sin(x)$

| Comportamento do Erro: Bicúbica, $255\sin(x)$ | | |
|---|---|---|
| h | Erro | Fator de redução do erro |
| 1.0000 | 0.651433 | |
| 0.5000 | 0.038075 | $\approx 0.0584$ |
| 0.2500 | 0.002385 | $\approx 0.0626$ |
| 0.1250 | 0.000149 | $\approx 0.0624$ |
| 0.0625 | 0.000009 | $\approx 0.0604$ |

Tabela 4: Comportamento do Erro para Interpolação Bicúbica: $255\sin(x)$

Analisando os dados nas tabelas é possível prever o comportamento do erro. Note que a cada linha de cada tabela, $h$ é multiplicado por $0.5$. Ao observarmos os valores do erro, é possível observar que, no caso da interpolação Bicúbica, o fator de redução varia em torno de $0.625 = 0.5^4$, enquanto que no caso da interpolação Bilinear, ele varia em torno de $0.25 = 0.5^2$. Essa constatação permite criar a hipótese de que o erro na interpolação Bilinear vale $\mathcal{O}(h^2)$ e na interpolação Bicúbica vale $\mathcal{O}(h^4)$.

### 4.3 Teste em imagens

A última parte dos testes consistiu em aplicar o que foi feito à compressão de imagens com perdas. Foram inventadas várias funções contínuas e com derivadas contínuas pelo menos até a derivada mista e, a partir delas, criadas imagens preto e branco com 128x128 pixels. A compressão consistiu em, usando os métodos de interpolação, diminuir a malha de pontos necessária para representar a mesma imagem. Para os testes rodados, a bicúbica precisava de malhas de cerca de 5x5 a 8x8 para já ter uma boa nitidez. Já a bilinear precisava de malhas de cerca de 15x15 a 20x20 para ter uma nitidez razoável. Além disso, as imagens geradas com a interpolação bicúbica eram muito mais suaves, por conta do uso das derivadas também como condição da interpolação. Na bilinear era possível ver "divisões"no meio do desenho, que eram regiões de separação dos quadrados da malha de interpolação. Isso porque a interpolação bilinear já não é tão suave,pois não leva em conta as derivadas. Observe esses detalhes citados nos exemplos abaixo.

1. $255\sin(x)$

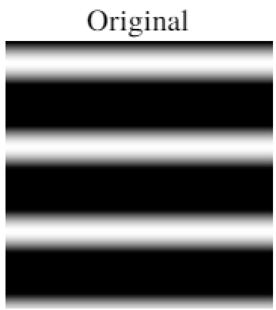


Figura 1: Bilinear 5x5

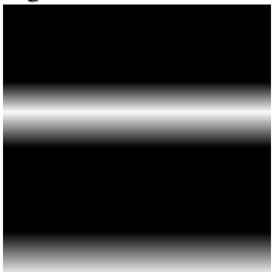


Figura 2: Bilinear 8x8

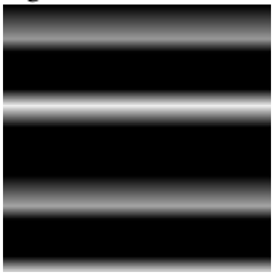


Figura 3: Bilinear 12x12

Figura 4: Bilinear 20x20

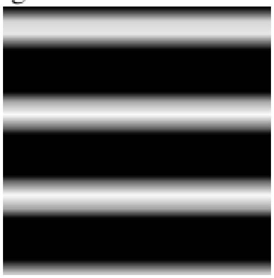


Figura 5: Bicúbica 5x5

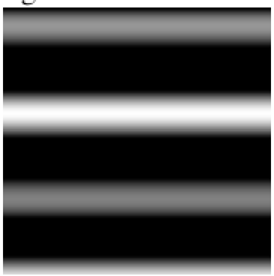
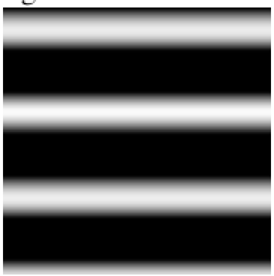


Figura 6: Bicúbica 8x8



2. $128\sin(x) + 128\cos(y)$

Original

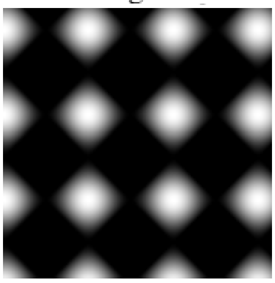


Figura 7: Bilinear 8x8

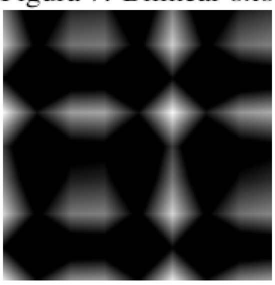


Figura 8: Bilinear 12x12

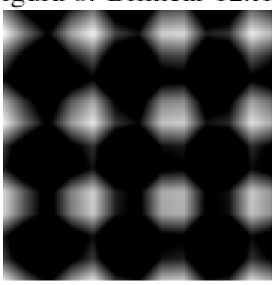


Figura 9: Bilinear 20x20

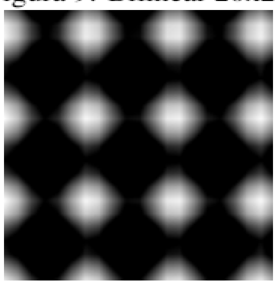


Figura 10: Bilinear 25x25

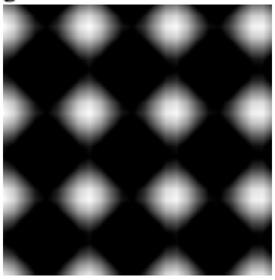


Figura 11: Bicúbica 8x8

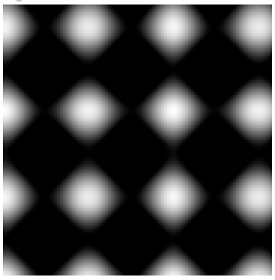


Figura 12: Bicúbica 12x12

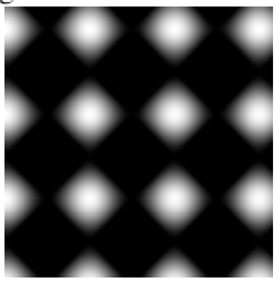
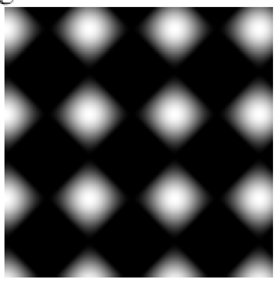


Figura 13: Bicúbica 20x20



Em todos os casos, pode-se notar que a taxa de compressão foi muito satisfatória, girando em torno de 97% para a bilinear e 99% para a bicúbica, a fim de conseguir boa nitidez. É claro, porém, que variando a função, esses valores vão mudar, por conta da necessidade de malhas de interpolação menores ou maiores para conseguir boa nitidez.