

# EP3 - LABORATÓRIO DE MÉTODOS NUMÉRICOS

## MACO210

Breno Moura e Tiago Nápoli

27/05/2017

## 1 Introdução

Neste trabalho, continuação do EP2, será implementada a Interpolação Bicúbica utilizando aproximação numérica das derivadas parciais e mistas. Será feito o embasamento teórico e diversos experimentos constatando o funcionamento correto do programa e comprovando alguns resultados teóricos. A Interpolação Bicúbica, juntamente com a Bilinear (do EP2), serão utilizadas para compressão de imagens com perdas, e alguns resultados serão apresentados.

## 2 Cálculo das derivadas

### 2.1 Caso com 1 variável

Vamos deduzir as fórmulas de diferenças de 3 pontos para aproximar as primeiras derivadas para o caso 1D. Depois vamos estender as fórmulas para o caso 2D. Para cada dedução, seja  $f$  uma função com pelo menos até terceira derivada e com terceira derivada limitada.

#### 2.1.1 Fórmula centrada de 3 pontos

Por Taylor temos:

$$f(x_0 - h) = f(x_0) - hf'(x_0) + \frac{h^2}{2}f''(x_0) - \frac{h^3}{6}f'''(\zeta_1)$$

$$f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{h^2}{2}f''(x_0) + \frac{h^3}{6}f'''(\zeta_2)$$

Com  $\zeta_1$  entre  $x_0$  e  $x_0 - h$  e  $\zeta_2$  entre  $x_0 - h$  e  $x_0$ .

Subtraindo a segunda pela primeira, temos:

$$f(x_0 + h) - f(x_0 - h) = 2hf'(x_0) + \frac{h^3}{6}[f'''(\zeta_2) + f'''(\zeta_1)]$$

Isolando  $f'(x_0)$ :

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0 - h)}{2h} - \frac{h^2}{12}[f'''(\zeta_1) + f'''(\zeta_2)]$$

Supondo  $f'''$  contínua, temos que, pelo teorema do Valor Intermediário, existe  $\zeta_3$  entre  $x_0 - h$  e  $x_0 + h$  tal que  $2f'''(\zeta_3) = f'''(\zeta_1) + f'''(\zeta_2)$ . Assim:

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0 - h)}{2h} - \frac{h^2}{6}f'''(\zeta_3)$$

Aproximando  $f'(x_0)$  por  $\frac{f(x_0+h)-f(x_0-h)}{2h}$ , temos que o erro será, em módulo, já que  $f'''$  foi suposta limitada:

$$\frac{h^2}{6}|f'''(\zeta_3)| = \mathcal{O}(h^2)$$

### 2.1.2 Fórmula para frente de 3 pontos

Por Taylor temos:

$$\begin{aligned}f(x_0 + h) &= f(x_0) + hf'(x_0) + \frac{h^2}{2}f''(x_0) + \frac{h^3}{6}f'''(\zeta_1) \\f(x_0 + 2h) &= f(x_0) + 2hf'(x_0) + \frac{4h^2}{2}f''(x_0) + \frac{8h^3}{6}f'''(\zeta_2)\end{aligned}$$

Com  $\zeta_1$  entre  $x_0$  e  $x_0 + h$  e  $\zeta_2$  entre  $x_0 + h$  e  $x_0 + 2h$ .

Subtraindo a segunda por 4 vezes a primeira, temos:

$$f(x_0 + 2h) - 4f(x_0 + h) = -3f(x_0) - 2hf'(x_0) + \frac{4h^3}{6}[2f'''(\zeta_2) - f'''(\zeta_1)]$$

Isolando  $f'(x_0)$ :

$$f'(x_0) = \frac{-3f(x_0) + 4f(x_0 + h) - f(x_0 + 2h)}{2h} + \frac{h^2}{3}[2f'''(\zeta_2) - f'''(\zeta_1)]$$

Supondo  $f'''$  derivável e limitada, e  $f''''$  limitada, existem  $M_1$  e  $M_2$  reais tais que:

$$\begin{aligned}|f'''(x)| &\leq M_1 \\|f''''(x)| &\leq M_2\end{aligned}$$

Temos, por Taylor ao redor de  $\zeta_2$ :

$$f'''(\zeta_2) = f'''(\zeta_1) + (\zeta_2 - \zeta_1)f''''(\zeta_3)$$

2 Com  $\zeta_3$  entre  $\zeta_1$  e  $\zeta_2$ , ou ainda  $\zeta_3$  entre  $x_0$  e  $x_0 + 2h$ .

Aproximando  $f'(x_0)$  por  $\frac{-3f(x_0)+4f(x_0+h)-f(x_0+2h)}{2h}$ , temos que o módulo do erro será (sabendo que  $|\zeta_2 - \zeta_1| \leq 2h$ ):

$$\begin{aligned}\left| \frac{h^2}{3}[2f'''(\zeta_2) - f'''(\zeta_1)] \right| &\leq \left| \frac{h^2}{3}[2f'''(\zeta_1) + 2(\zeta_2 - \zeta_1)f''''(\zeta_3) - f'''(\zeta_1)] \right| \leq \\&\left| \frac{h^2}{3}[f'''(\zeta_1) + 2(\zeta_2 - \zeta_1)f''''(\zeta_3)] \right| \leq \\&\left| \frac{h^2}{3}[f'''(\zeta_1) + 2(\zeta_2 - \zeta_1)f''''(\zeta_3)] \right| \leq \\&|\frac{h^2}{3}f'''(\zeta_1)| + |\frac{2h^2}{3}(\zeta_2 - \zeta_1)f''''(\zeta_3)| \leq \\&\frac{h^2}{3}|M_1| + \frac{4h^3}{3}|M_2|\end{aligned}$$

Logo, o erro será  $\mathcal{O}(h^2)$ .

### 2.1.3 Fórmula para trás de 3 pontos

Por Taylor temos:

$$\begin{aligned}f(x_0 - h) &= f(x_0) - hf'(x_0) + \frac{h^2}{2}f''(x_0) - \frac{h^3}{6}f'''(\zeta_1) \\f(x_0 - 2h) &= f(x_0) - 2hf'(x_0) + \frac{4h^2}{2}f''(x_0) - \frac{8h^3}{6}f'''(\zeta_2)\end{aligned}$$

Com  $\zeta_1$  entre  $x_0$  e  $x_0 - h$  e  $\zeta_2$  entre  $x_0$  e  $x_0 - 2h$ .

Subtraindo a segunda por 4 vezes a primeira, temos:

$$f(x_0 - 2h) - 4f(x_0 - h) = -3f(x_0) + 2hf'(x_0) - \frac{4h^3}{6}[2f'''(\zeta_2) - f'''(\zeta_1)]$$

Isolando  $f'(x_0)$ :

$$f'(x_0) = \frac{f(x_0 - 2h) - 4f(x_0 - h) + 3f(x_0)}{2h} + \frac{h^2}{3}[2f'''(\zeta_2) - f'''(\zeta_1)]$$

Daí, aproximaremos:

$$f'(x_0) \approx \frac{f(x_0 - 2h) - 4f(x_0 - h) + 3f(x_0)}{2h}$$

O cálculo do erro será análogo aos casos anteriores e dará  $\mathcal{O}(h^2)$ .

## 2.2 Caso com 2 variáveis

Vamos fixar como notação:

$$\begin{aligned} f_x(x, y) &= \frac{\partial f(x, y)}{\partial x} \\ f_y(x, y) &= \frac{\partial f(x, y)}{\partial y} \\ f_{xy}(x, y) &= \frac{\partial^2 f(x, y)}{\partial xy} \end{aligned}$$

Temos, do Cálculo, que as derivadas parciais são conseguidas fixando a outra coordenada e calculando a derivada como se houvesse somente uma coordenada variando. Então podemos, usando essa ideia, aplicar as fórmulas deduzidas na seção anterior para o caso com duas variáveis. Assim teremos:

$$\begin{aligned} f_x(x_i, y_j) &\approx \frac{f(x_i + h, y_j) - f(x_i - h, y_j)}{2h} = \frac{f(x_{i+1}, y_j) - f(x_{i-1}, y_j)}{2h} \\ f_y(x_i, y_j) &\approx \frac{f(x_i, y_j + h) - f(x_i, y_j - h)}{2h} = \frac{f(x_i, y_{j+1}) - f(x_i, y_{j-1})}{2h} \end{aligned}$$

Para a derivada mista, vamos considerar a função da classe  $C^2$ . Assim:

$$\frac{\partial f_x(x, y)}{\partial y} = \frac{\partial f_y(x, y)}{\partial x}$$

Daí podemos, usando as fórmulas da seção anterior:

$$f_{xy}(x_i, y_j) \approx \frac{f_y(x_{i+1}, y_j) - f_y(x_{i-1}, y_j)}{2h}$$

### 2.2.1 Casos Especiais: Bordas

Nas bordas, as fórmulas de diferenciação centrada não podem ser usadas. Poderemos, entretanto, usar as fórmulas para trás de 3 pontos ou para frente de 3 pontos, dependendo se a borda estiver para cima, baixo, esquerda ou direita.

Borda na direita (não há  $x_{i+1}$ ):

$$f_x(x_i, y_j) = \frac{f(x_i - 2h, y_j) - 4f(x_i - h, y_j) + 3f(x_i, y_j)}{2h} = \frac{f(x_{i-2}, y_j) - 4f(x_{i-1}, y_j) + 3f(x_i, y_j)}{2h}$$

$$f_{xy}(x_i, y_j) = \frac{f_y(x_i - 2h, y_j) - 4f_y(x_i - h, y_j) + 3f_y(x_i, y_j)}{2h} = \frac{f_y(x_{i-2}, y_j) - 4f_y(x_{i-1}, y_j) + 3f_y(x_i, y_j)}{2h}$$

Borda na esquerda (não há  $x_{i-1}$ ):

$$f_x(x_i, y_j) = \frac{-3f(x_i, y_j) + 4f(x_i + h, y_j) - f(x_i + 2h, y_j)}{2h} = \frac{-3f(x_i, y_j) + 4f(x_{i+1}, y_j) - f(x_{i+2}, y_j)}{2h}$$

$$f_{xy}(x_i, y_j) = \frac{-3f_y(x_i, y_j) + 4f_y(x_i + h, y_j) - f_y(x_i + 2h, y_j)}{2h} = \frac{-3f_y(x_i, y_j) + 4f_y(x_{i+1}, y_j) - f_y(x_{i+2}, y_j)}{2h}$$

Borda em cima (não há  $y_{j-1}$ ):

$$f_y(x_i, y_j) = \frac{-3f(x_i, y_j) + 4f(x_i, y_j + h) - f(x_i, y_j + 2h)}{2h} = \frac{-3f(x_i, y_j) + 4f(x_i, y_{j+1}) - f(x_i, y_{j+2})}{2h}$$

Borda embaixo (não há  $y_{j+1}$ ):

$$f_y(x_i, y_j) = \frac{f(x_i, y_j - 2h) - 4f(x_i, y_j - h) + 3f(x_i, y_j)}{2h} = \frac{f(x_i, y_{j-2}) - 4f(x_i, y_{j-1}) + 3f(x_i, y_j)}{2h}$$

### 3 Implementação

Para implementação fizemos as funções `aproxdf`, `compress` e `ErroAproxdf`. A função `aproxdf` aproxima as derivadas de  $f(x, y)$  dado uma malha de  $N_x$  por  $N_y$  pontos utilizando as fórmulas discutidas e tratando devidamente os casos da borda. A função `ErroAproxdf` calcula o erro máximo da aproximação de cada derivada, utilizamos a função para testar empiricamente o erro e verificar se era, de fato,  $\mathcal{O}(h^2)$ . A função `compress` recebia o nome do arquivo que continha a imagem, o tipo de interpolação e a quantidade de pontos na malha (Sendo que a quantidade de pontos na malha menos 1 deve dividir a quantidade de pixels menos 1, para que as bordas estejam contidas na malha). A função pega pontos igualmente espaçados entre 1 e o número de pixels e os usa para fazer as matrizes de interpolação (Explicado no EP2), usando `aproxdf` para aproximar as derivadas. Com essas matrizes a função interpola os outros pontos, gerando uma imagem comprimida com perdas.

## 4 Experimentos

A fim de constatar o funcionamento correto das aproximações e do programa criado, foram realizados vários experimentos, que estão documentados abaixo.

### 4.1 Erro das derivadas

O primeiro teste realizado foi o teste do erro das derivadas. Queria-se constatar que o erro da aproximação das derivadas seguia o resultado conseguido na teoria:  $\mathcal{O}(h^2)$ .

Para esta e as próximas seções, 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  $h$ .

Para isso foram testadas várias funções: os resultados de duas delas foram colocados abaixo. Para os testes, o domínio foi  $[0, 10] \times [0, 10]$ , e foi variado a distância  $h$ . Para cada malha, todos os pontos eram

visitados e calculava-se o valor real e o aproximado da derivada, conseguindo-se o erro da aproximação. Foi considerado, então, o valor máximo entre esses erros, para cada malha.

1. Para a função  $f(x, y) = x^5 + y^3 + x^2y + y^3x$ :

Comportamento Erro $f_x(x, y)$			Comportamento Erro $f_y(x, y)$		
h	Erro	Fator de redução do erro	h	Erro	Fator de redução do erro
1.0000	1714.000		1.0000	22.000	
0.5000	463.375	$\approx 0.270$	0.5000	5.500	$\approx 0.25$
0.2500	120.367	$\approx 0.259$	0.2500	1.375	$\approx 0.25$
0.1250	30.667	$\approx 0.254$	0.1250	0.343	$\approx 0.25$
0.0625	7.739	$\approx 0.252$	0.0625	0.085	$\approx 0.25$

Comportamento Erro $f_{xy}(x, y)$		
h	Erro	Fator de redução do erro
1.0000	2.000000	
0.5000	0.500000	$\approx 0.25$
0.2500	0.125000	$\approx 0.25$
0.1250	0.031250	$\approx 0.25$
0.0625	0.007812	$\approx 0.25$

Observando as tabelas para essa função, ficou fácil de observar que a ordem do erro é de  $\mathcal{O}(h^2)$ , conforme previsto na parte teórica, pois multiplicando  $h$  por 0.5, o erro multiplicava por, aproximadamente,  $0.5^2 = 0.25$ .

2. Para a função  $g(x, y) = y \sin(x)$ :

Comportamento Erro $g_x(x, y)$			Comportamento Erro $g_{xy}(x, y)$		
h	Erro	Fator de redução do erro	h	Erro	Fator de redução do erro
1.0000	3.065180		1.0000	0.306518	
0.5000	0.802688	$\approx 0.26$	0.5000	0.080269	$\approx 0.26$
0.2500	0.203806	$\approx 0.25$	0.2500	0.020381	$\approx 0.25$
0.1250	0.051799	$\approx 0.25$	0.1250	0.005180	$\approx 0.25$
0.0625	0.013003	$\approx 0.25$	0.0625	0.001300	$\approx 0.25$

Nesse caso, não foi colocada a tabela para  $g_y(x, y)$  porque o erro para ela foi 0 para todos os casos, uma vez que  $g_y(x, y) = \sin(x)$ , assim, se observarmos a fórmula para cálculo das derivadas (que mantém o  $x$  e varia o  $y$ ), ele de fato deve dar 0. Foi possível observar claramente a ordem do erro, pois, multiplicando  $h$  por 0.5 o erro multiplicava por, aproximadamente,  $0.5^2 = 0.25$ , mostrando um erro da ordem de  $\mathcal{O}(h^2)$  para as derivadas, conforme previsto na parte teórica.

3. Função  $k(x, y) = e^{x+y}$

Comportamento do Erro: $k_x(x, y)$		
h	Erro	Fator de redução do erro
1.0000	3702.455964	
0.5000	1282.891801	$\approx 0.34$
0.2500	382.047841	$\approx 0.30$
0.1250	104.566441	$\approx 0.27$
0.0625	27.374257	$\approx 0.26$

Comportamento do Erro: $k_{xy}(x, y)$		
h	Erro	Fator de redução do erro
1.0000	6782.561512	
0.5000	2491.063882	$\approx 0.36$
0.2500	757.469083	$\approx 0.30$
0.1250	208.636472	$\approx 0.27$
0.0625	54.714494	$\approx 0.26$

Nesse caso,  $k_x(x, y) = k_y(x, y)$ , assim, as tabelas deles ficaram iguais, foi colocado somente um deles. Nota-se que o fator de redução do erro converge a 0.25.

## 4.2 Verificação de Interpolação

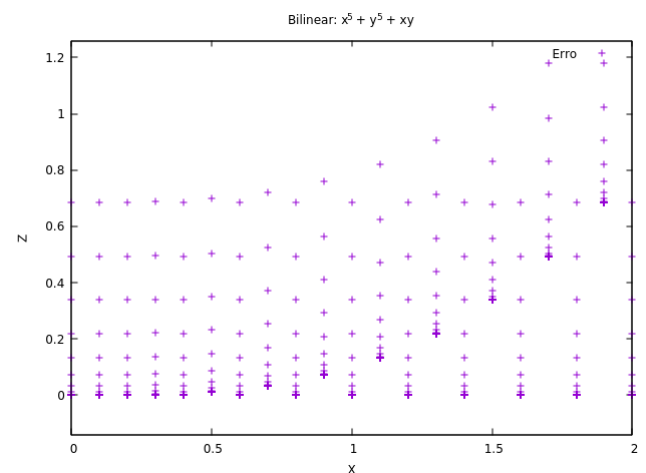
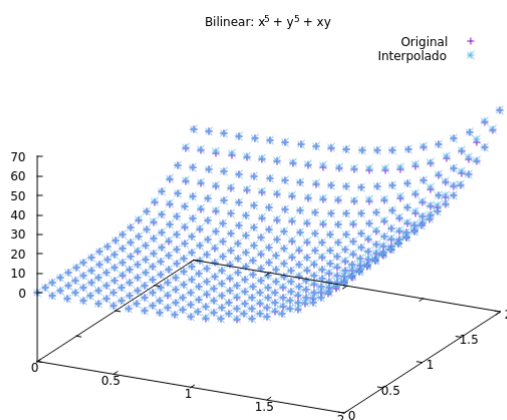
Como feito no EP2, a verificação de interpolação deve constatar que:

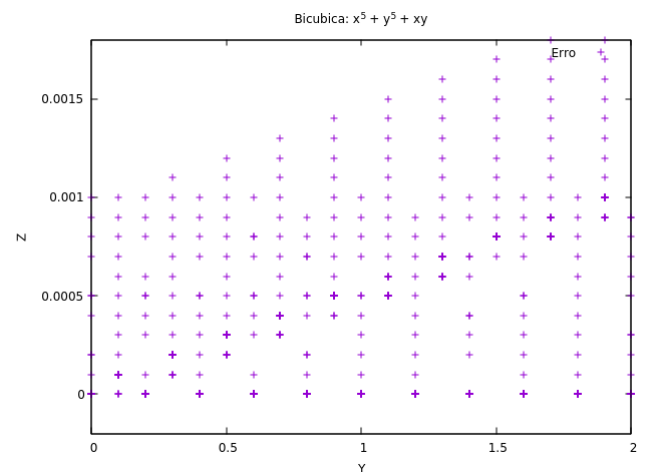
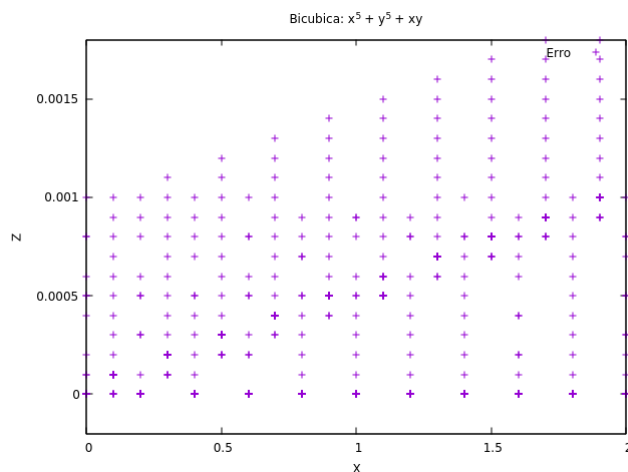
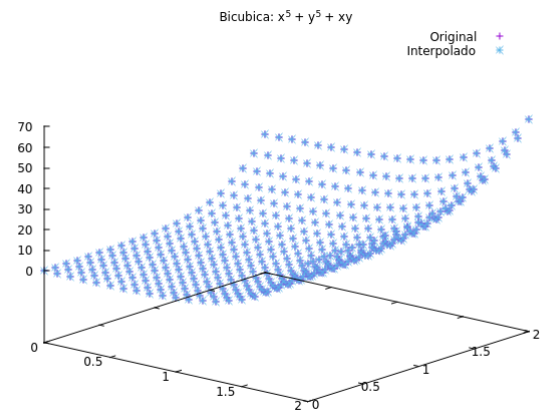
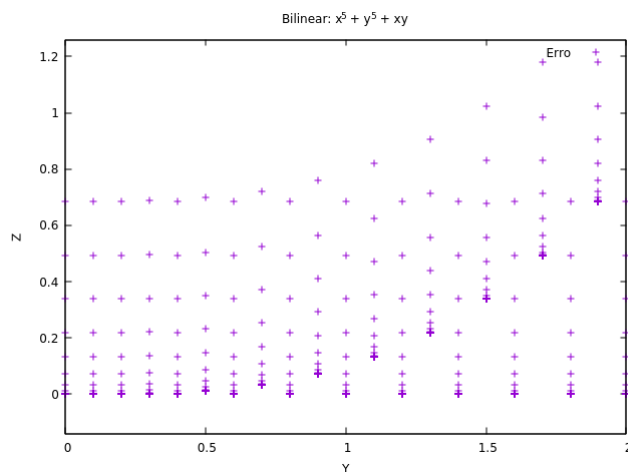
$$v(x_i, y_j) = f(x_i, y_j)$$

Sendo  $v(x, y)$  o polinômio interpolador, e  $f(x, y)$  a função interpolada, e  $x_i, y_j$  pontos pertencentes à malha interpoladora.

Para isso, a função *DataMain* percorre, com um passo determinado (nos experimentos foi 0.1), o *grid* em que a função foi interpolada. Em cada ponto  $(x, y)$  visitado, calculava-se o valor real da função  $f(x, y)$  e o valor aproximada pelo polinômio interpolador. Assim, calculava-se o valor do módulo do erro. Com esses valores, então, foram feitos gráficos dos valores da função, do polinômio e do erro. Foram feitos testes com as mesmas funções do EP2, para que pudéssemos observar como aproximar as derivadas interfeririam na interpolação bicúbica (essa comparação será comentada na seção *Erro de Interpolação*).

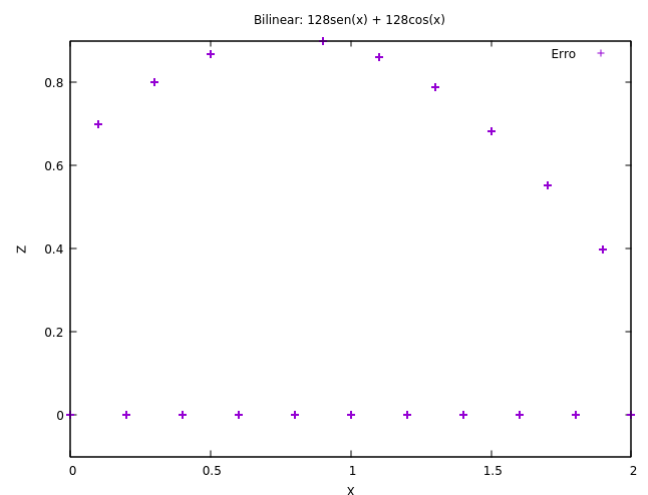
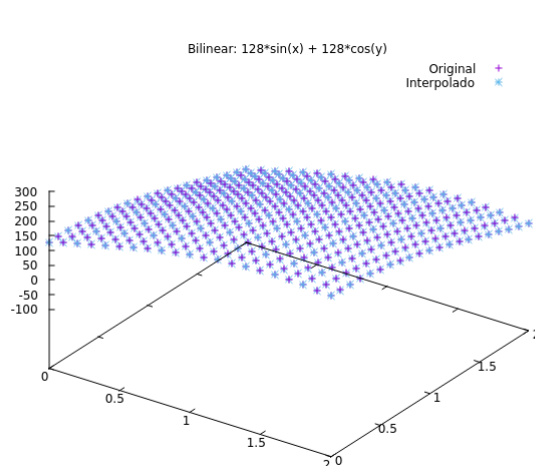
1. Função:  $x^5 + y^5 + xy$ . A região de *plot* foi  $[0, 2] \times [0, 2]$ , passo = 0.1, com  $11 \times 11$  pontos usados na interpolação ( $N_x = N_y = 10$ ).

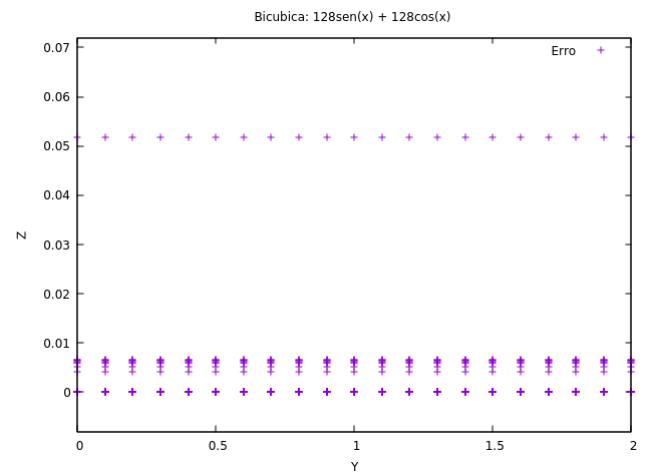
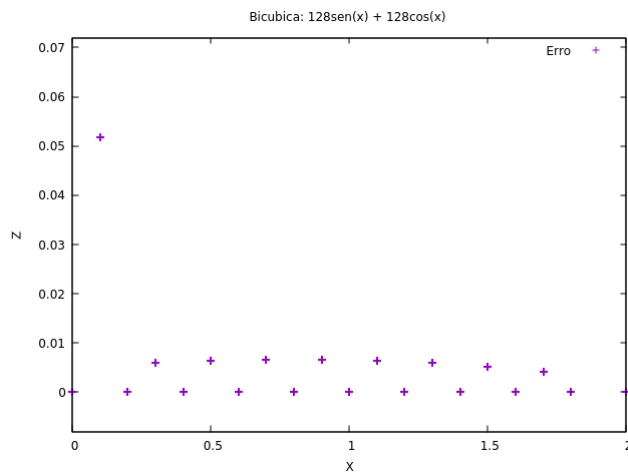
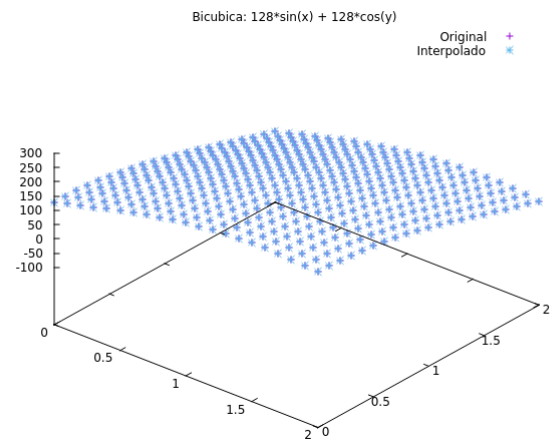
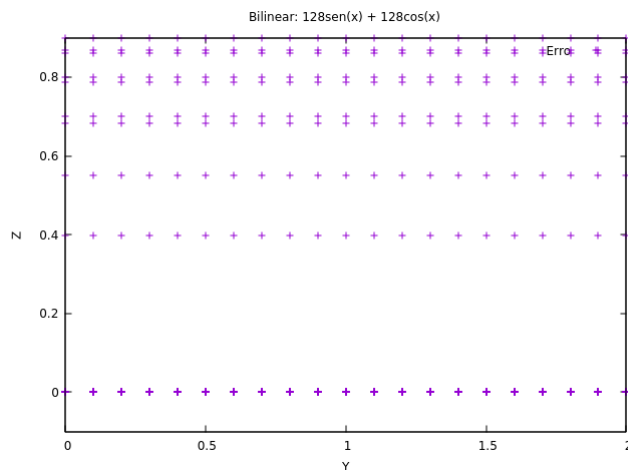




Analisando os gráficos é possível observar que ambas interpolações aproximam bem o comportamento da função original. A interpolação bicúbica, mesmo usando derivadas aproximadas, se mostra novamente muito melhor que a interpolação bilinear, ou seja com ordem do erro bem menor. É possível observar, no gráfico do erro, que há vários pontos em que o erro se anula, vários desses pontos os da malha interpoladora, constatando que o polinômio interpola a função.

- Função  $128 \sin(x) + 128 \cos(x)$ . 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.

### 4.3 Erro de Interpolação

Este experimento consistiu em procurar determinar, na prática, o comportamento do erro de interpolação. Para isso, variava-se  $N$  de modo a diminuir  $h$ . Algumas funções foram testadas e seus resultados foram compilados nas seguintes tabelas:

1. Função  $f(x, y) = x^5 + y^5 + xy$ , no domínio  $[0, 5] \times [0, 5]$ .

Comportamento do Erro: Bilinear, $x^5 + y^5 + xy$		
h	Erro	Fator de redução do erro
1.0000	458.437500	
0.5000	130.145640	$\approx 0.28$
0.2500	34.947364	$\approx 0.27$
0.1250	8.338389	$\approx 0.24$
0.0625	2.211596	$\approx 0.26$

Comportamento do Erro: Bicúbica $x^5 + y^5 + xy$		
h	Erro	Fator de redução do erro
1.0000	133.1404	
0.5000	20.1206	$\approx 0.15$
0.2500	2.7512	$\approx 0.14$
0.1250	0.1787	$\approx 0.065$
0.0625	0.0109	$\approx 0.0613$



2. Função  $g(x, y) = e^{x+y}$ :

Comportamento do Erro: Bilinear $e^{x+y}$		
h	Erro	Fator de redução do erro
1.0000	2398.8321	
0.5000	873.9084	$\approx 0.36$
0.2500	267.3688	$\approx 0.30$
0.1250	55.9025	$\approx 0.21$
0.0625	16.9826	$\approx 0.30$

Comportamento do Erro: Bicúbica $e^{x+y}$		
h	Erro	Fator de redução do erro
1.0000	917.175322	
0.5000	191.897527	$\approx 0.21$
0.2500	32.357665	$\approx 0.17$
0.1250	2.214165	$\approx 0.068$
0.0625	0.119656	$\approx 0.054$

3. Função  $h(x, y) = x^5 + y^3 + x^2y + y^3x$ .

Comportamento do Erro: Bilinear $x^5 + y^3 + x^2y + y^3x$		
h	Erro	Fator de redução do erro
1.0000	248.906250	
0.5000	70.337220	$\approx 0.28$
0.2500	18.843707	$\approx 0.27$
0.1250	4.500938	$\approx 0.24$
0.0625	1.191770	$\approx 0.26$

Comportamento do Erro: Bicúbica $x^5 + y^3 + x^2y + y^3x$		
h	Erro	Fator de redução do erro
1.0000	68.720640	
0.5000	10.338720	$\approx 0.15$
0.2500	1.411035	$\approx 0.14$
0.1250	0.091568	$\approx 0.065$
0.0625	0.005620	$\approx 0.0613$

Em relação ao EP2, o cálculo da interpolação bilinear não mudou, pois não se usa as derivadas no cálculo desta interpolação. Observa-se, para a Bilinear, que o erro tende a  $\mathcal{O}(h^2)$ . Para o caso da interpolação Bicúbica, como o cálculo das derivadas foi aproximado, houve uma propagação de erro para a interpolação, aumentando os valores do erro em relação ao EP2 (comparando com as tabelas do relatório passado), o que era esperado. Quanto aos fatores de redução do erro, para os valores maiores de  $h$ , os fatores de redução são piores em relação do EP2. Entretanto, diminuindo  $h$ , o erro do cálculo das derivadas vai diminuindo, e então o fator de redução do erro começa a tender a  $0.625 = 0.5^4$ , ou  $\mathcal{O}(h^4)$ , que foi o resultado obtido no EP2. Além disso, pode-se claramente notar que o erro máximo vai diminuindo conforme vamos diminuindo  $h$ , algo esperado pela teoria.

## 5 Compressão de imagens

### 5.1 Imagens geradas a partir de funções

Primeiramente foram testadas imagens geradas a partir de funções de classe  $C^2$ , assim como no EP2. Os resultados, comparando com o EP2, foram semelhantes. Para a interpolação bilinear não houve diferença, pois ela não foi mudada. Para a interpolação bicúbica, que agora utiliza aproximações das derivadas, tivemos resultados ligeiramente piores que o EP2, mas muito satisfatórios e ainda bem melhores que a interpolação bilinear.

1. Função:  $f(x, y) = 128 \sin(x) + 128 \cos(x)$ :

Figura 1: Original

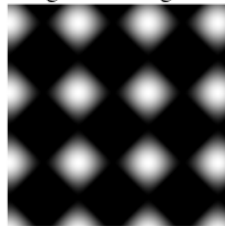


Figura 2: Bicúbica 10x10    Figura 3: Bicúbica 15x15    Figura 4: Bicúbica 18x18

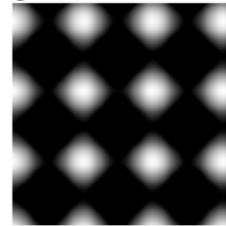
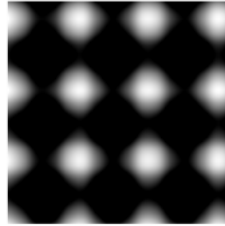
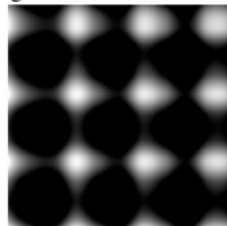
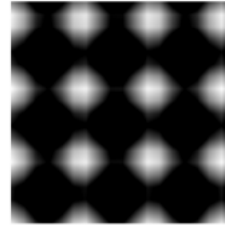
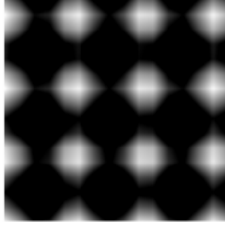
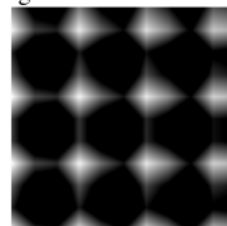


Figura 5: Bilinear 10x10    Figura 6: Bilinear 15x15    Figura 7: Bilinear 18x18



2. Função:  $f(x, y) = 255 \sin(x)$ :

Figura 8: Original



Figura 9: Bicúbica 10x10



Figura 10: Bicúbica 12x12



Figura 11: Bicúbica 15x15

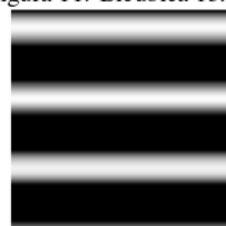


Figura 12: Bilinear 10x10



Figura 13: Bilinear 12x12



Figura 14: Bilinear 15x15



## 5.2 Imagens quaisquer

Por fim, foram feitos testes de compressão para imagens quaisquer, não obtidas necessariamente a partir de funções contínuas. Por isso, espera-se que as imagens comprimidas tenham pior qualidade, e para obter boa nitidez será necessário uma malha maior. Os resultados, entretanto, estão muito ligados a cada imagem. (Obs: Compressão, no contexto das legendas abaixo, significa que os pontos originais foram comprimidos a x%, ou seja, foram usados x% dos pontos originais).

### 1. Card

Figura 1: Original



Figura 2: Bicúbica - Compressão: 25%



Figura 3: Bicúbica - Compressão: 12%



Figura 4: Bicúbica - Compressão: 5%



Figura 5: Bicúbica - Compressão: 0.2%



## 2. Gato

Figura 1: Original



Figura 2: Bicúbica - Compressão: 2.6880%    Figura 3: Bilinear - Compressão: 2.6880%



## 3. Meninas superpoderosas

Figura 1: Original

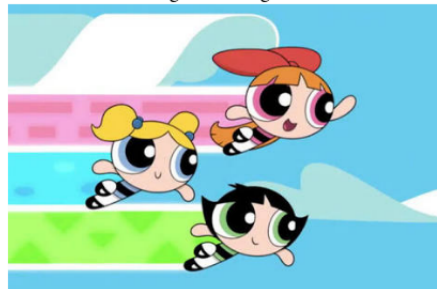
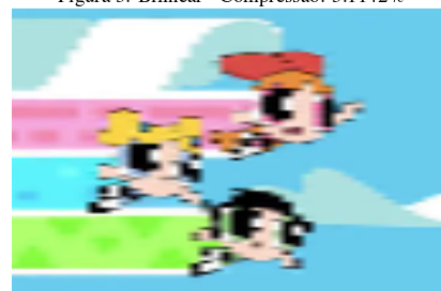


Figura 2: Bicúbica - Compressão: 3.1142%



Figura 3: Bilinear - Compressão: 3.1142%



## 4. Quadrados

Figura 1: Quadrados

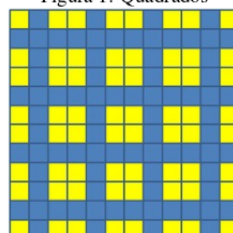
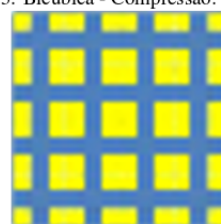
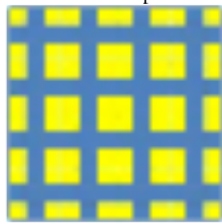


Figura 2: Bilinear - Compressão: 2.1361%    Figura 3: Bicúbica - Compressão: 2.1361%



### 5.3 Conclusões

A conclusão que tivemos é que para comprimir imagens com perdas utilizando interpolação, aproximadamente 20% a 25% dos pontos da imagem original devem estar contidos na malha, para pelo menos termos uma imagem razoável (Não muito embaçada). Com 5% a 10% a imagem já é reconhecível na maioria dos casos, contudo a imagem fica pouco nítida, dificultando a identificação de detalhes. Entretanto, tudo depende da imagem escolhida, e os resultados para cada imagem variarão. Até mesmo a escolha do método, bicúbico ou bilinear, deve ser feito de acordo com os resultados práticos obtidos com cada um, uma vez que, como para imagens quaisquer as funções não são nem contínuas necessariamente, os efeitos em cada tipo de interpolação serão inesperados.

A compressão de imagens com perdas é um exemplo de como podemos aplicar os métodos de interpolação aprendidos em sala de aula em situações reais, obtendo resultados muito satisfatórios. Ver os métodos de interpolação de maneira aplicada ajudou a entender a importância de entender de fato a teoria, para somente assim termos um claro entendimento sobre como a interpolação é feita e por que ela funciona.