RELATÓRIO - EP3

Renan Fichberg - **NUSP:** 7991131

Laboratório de Métodos Numéricos - MAC0210 - 2017/1

Professor: Ernesto G. Birgin

Monitor: Lucas Magno

1 Arquivos e diretórios

Neste exercício programa, estão sendo entregues os seguintes arquivos e diretórios:

- `/docs` - Diretório contendo este relatório.
- `/docs/relatorio.pdf` - Este documento.
- `/images` - Diretório que contém a imagem original e as geradas (se for rodar o programa que manipula imagens ao invés de funções).
- `/images/purple_tentacle.jpg` - Imagem original. Está sendo entregue como exemplo de entrada do programa de imagens para efeitos ilustrativos.
- `/images/compressed_red.jpg` - Imagem gerada pela compressão da imagem `purple_tentacle.jpg`. Está sendo entregue (e o programa a escreve no disco) como exemplo de saída do programa para efeitos ilustrativos.
- `/images/compressed_blue.jpg` - Idem a imagem `compressed_red.jpg`.
- `/images/compressed_green.jpg` - Idem as imagens `compressed_red.jpg` e `compressed_blue.jpg`.
- `/images/decompressed_red.jpg` - Imagem gerada pela decompressão da imagem `decompressed_red.jpg` Está sendo entregue como exemplo de saída do programa para efeitos ilustrativos. Retorna a imagem ao estado anterior (isto é, restaura a imagem `compressed_red.jpg` através dos métodos de interpolação bicúbica ou bilinear nos *pixels* de `compressed.jpg`), com perdas.
- `/images/decompressed_blue.jpg` - Idem a imagem `decompressed_red.jpg`.
- `/images/decompressed_green.jpg` - Idem as imagens `decompressed_red.jpg` e `decompressed_blue.jpg`
- `/images/final.jpg` - Junção dos canais para formar a imagem original (a qualidade está ruim por causa de um problema com a biblioteca `GraphicsMagick`).
- `/src` - Diretório contendo os códigos-fonte do Exercício Programa 3.
- `/src/argument_checker.m` - Código-fonte do Exercício Programa 3 para leitura dos argumentos da CLI, chamado pelo *script* principal selecionado (um dos dois listados nos itens abaixo).
- `/src/bivariate_interpolation.m` - Código-fonte do Exercício Programa 3 para manipulação de uma *imagem*.

- `/src/bivariate_interpolation_test.m` - Código-fonte do Exercício Programa 3 para manipulação de uma *função conhecida f(x, y)*.

2 Invocação dos Programas

O exercício programa foi dividido em dois *scripts* devido a minha experiência obtida com o EP2, na qual eu achei que o código acabou tendo muitas condições de desvio que acabaram aumentando a complexidade da função principal, tornando-o difícil (e chato) de ler. Para resolver este problema, achei melhor desenvolver dois *scripts* independentes. O código-fonte de ambos difere de muita pouca coisa, sendo que o que foi pedido tanto no EP2 quanto no EP3 está implementado da mesma forma em ambos os *scripts*. Todas as funções implementadas, porém, são encontradas em ambos os códigos-fonte com a mesma assinatura, diferindo apenas nos parâmetros (no de imagem, os parâmetros referentes às malhas são triplicados por causa dos três canais de cores. Idem eventuais retornos das malhas pelas funções).

2.1 Manipulação de imagens

Para rodar o programa no modo imagem, é necessário estar no diretório do código-fonte '`/src`' e usar o seguinte comando:

```
$ ./bivariate_interpolation.m <parametros>
```

Onde os `<parâmetros>` disponíveis são os descritos abaixo (todos antecedidos de *dois* caracters "-"):

- –**image**: parâmetro **mandatório**. Deve ser uma **imagem**. Os testes foram realizados na imagem que está sendo fornecida, com a extensão `.jpg`. A imagem deve **obrigatoriamente** estar no diretório '`/images`'.
- –**bilinear**: parâmetro **semi-mandatório**. Passe este parâmetro, sem argumentos, se quiser rodar o programa no modo de interpolação bilinear. Este parâmetro está classificado como "semi-mandatório" pois ele não é obrigado estar presente, desde que o parâmetro para o modo bicúbico esteja. **Um dos dois parâmetros de modo deve ser obrigatoriamente passado**.
- –**bicubic**: parâmetro **semi-mandatório**. Idem '`--bilinear`', só que para rodar o programa no modo bicúbico. Apenas um dos parâmetros 'semi-mandatórios' pode ser passado por vez.
- –**cr**: parâmetro **mandatório**. Deve ser um **número inteiro** $\geq$ **5**. É a taxa de compressão da imagem (**c**ompression **r**ate).

Exemplo válido de invocação do *script* de imagem:

```
$ ./bivariate_interpolation.m --image purple_tentacle.jpg --bicubic --cr 5
```

2.2 Manipulação de funções e rotinas de teste

Para rodar o programa no modo de funções, é necessário estar no diretório do código-fonte '`/src`' e usar o seguinte comando:

```
$ ./bivariate_interpolation_test.m <parametros>
```

Onde os `<parâmetros>` disponíveis são os descritos abaixo (todos antecedidos de *dois* caracters "-"):

- **−nx**: parâmetro **mandatório**. Deve ser um **número inteiro e positivo**, conforme especificado no enunciado.
- **−ny**: parâmetro **mandatório**. Deve ser um **número inteiro e positivo**, conforme especificado no enunciado.
- **−ax**: parâmetro **mandatório**. Deve ser um **número real**, conforme especificado no enunciado.
- **−bx**: parâmetro **mandatório**. Deve ser um **número real**, conforme especificado no enunciado. Ainda, deve satisfazer a condição $a_x < b_x$.
- **−ay**: parâmetro **mandatório**. Deve ser um **número real**, conforme especificado no enunciado.
- **−by**: parâmetro **mandatório**. Deve ser um **número real**, conforme especificado no enunciado. Ainda, deve satisfazer a condição $a_y < b_y$.
- **−x**: parâmetro **mandatório**. Deve ser um **número real contido na malha**.
- **−y**: parâmetro **mandatório**. Deve ser um **número real contido na malha**.
- **−bilinear**: parâmetro **semi-mandatório**. Passe este parâmetro, sem argumentos, se quiser rodar o programa no modo de interpolação bilinear. Este parâmetro está classificado como "semi-mandatório" pois ele não é obrigado estar presente, desde que o parâmetro para o modo bicúbico esteja. **Um dos dois parâmetros de modo deve ser obrigatoriamente passado**.

- –**bicubic**: parâmetro **semi-mandatório**. Idem '`--bilinear`', só que para rodar o programa no modo bicúbico. Apenas um dos parâmetros 'semi-mandatórios' pode ser passado por vez.

Exemplo válido de invocação do *script* de funções:

```
$ ./bivariate_interpolation_test.m --nx 150 --ny 150 --ax -5 --bx 5 --ay -5
--by 5 --x 3.11 --y 1.42 --bicubic
```

2.2.1 Testes realizados neste modo

Mais informações sobre os testes descritos abaixo podem ser encontradas na seção 7.

1. Teste de verificação: o programa irá tentar mostrar ao usuário que, para os valores de parâmetros escolhidos, v interpola f nos pontos da malha. Ou seja, será mostrado que para todos os pontos (x, y) da malha, $|f(x, y) - v(x, y)| = 0$.

2. Teste de comportamento do erro: se o programa detectar que a área é quadrada, isto é, $h_x = h_y$, serão *duplicados* os valores dos parâmetros n_x e n_y de forma a mostrar que o erro diminui com o aumento da quantidade de pontos na malha fina (em outras palavras, quando tornamos a malha fina mais densa, fazendo h se aproximar mais de zero. $h_x = h_y = h \to 0$) . O objetivo deste teste é tentar mostrar que existe uma relação entre o erro e os valores de h_x e h_y.

   Consideração importante: o teste de comportamento de erro irá re-executar o programa com valores superiores para os parâmetros n_x e n_y. A depender do seu valor inicial, a execução do programa pode ficar consideravelmente demorada, pois o número de pontos na malha para a realização deste teste será **superior**. A explicação do porque isso é feito já foi fornecida acima e, conforme já foi mencionado, mais informações sobre este teste pode ser encontrado na seção 7.

3 Dedução dos métodos de interpolação: cálculo eficiente dos coeficientes

Esta seção abordará detalhes do que foi feito na implementação para resolver o problema do EP2. Todas as definições e suposições feitas no enunciado já estão sendo consideradas para as explicações que se seguem. Para a resolução dos problemas do EP3, por favor, pule para a próxima seção. Ainda, há uma explicação adicional no final do caso bilinear (sobre a não necessidade de calcular o sistema linear apresentado mas apenas usar diretamente o seu resultado diretamente).

3.1 Caso Bilinear

Para resolver o problema de interpolação no caso bilinear, precisamos apenas considerar os valores da função $f(x, y)$ em cada vértice do *pixel* da malha definida pelos 6 parâmetros mandatórios de entrada n_x, n_y, a_x, b_x, a_y e b_y. Para todos os efeitos, vamos sempre considerar que o ponto (x_i, y_j) refere-se ao vértice *inferior esquerdo* do *pixel*, que o ponto (x_{i+1}, y_j) refere-se ao vértice *inferior direito* do *pixel*, que o ponto (x, y_{j+1}) refere-se ao vértice *superior esquerdo* do *pixel* e, finalmente, que o ponto (x_{i+1}, y_{j+1}) refere-se ao vértice do *pixel* que sobrou, o *superior direito*.

O polinômio dado no enunciado por

$$s_{ij}^L(x, y) = a_{00} + a_{10}(x - x_i) + a_{01}(y - y_j) + a_{11}(x - x_i)(y - y_j)$$

deve interpolar f, e portanto, temos que a condição $f(x, y) \approx s_{ij}^L(x, y)$ deve ser satisfeita. Ainda, nos pontos da malha, temos que $f(x, y) = s_{ij}^L(x, y)$, que nos leva às condições de interpolação que são fornecidas no enunciado e reescritas abaixo:

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

Que podemos expandi-las para cada um dos quatro vértices do *pixel*:

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

E podemos reescrever o que está representado acima matricialmente por:

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

E, finalmente, obtemos uma maneira de calcular os coeficientes eficientemente:

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

Assim, os 4 coeficientes a_{ij} de cada polinômio interpolador de $v(x, y)$ podem ser calculados diretamente fazendo as seguintes computações na ordem explicitada abaixo:

i. $a_{00} = f(x_i, y_j)$

ii. $a_{10} = \frac{f(x_{i+1}, y_j) - a_{00}}{h_x}$ e $a_{01} = \frac{f(x_i, y_{j+1}) - a_{00}}{h_y}$

iii. $a_{11} = \frac{f(x_{i+1}, y_{j+1}) - a_{00} - h_x a_{10} - h_y a_{01}}{h_x h_y}$

3.2 Caso Bicúbico

O caso bicúbico compartilha semelhanças com o caso bilinear explicado na subseção anterior. A diferença é que agora precisamos usar os valores das derivadas de primeira ordem com relação às variáveis x e y de cada vértice do *pixel*, bem como as derivadas mistas de segunda ordem **também**, pois temos 12 coeficientes a mais a serem determinados. Naturalmente, é suposto que a função admita a existência das derivadas parciais com relação às variáveis x e y, bem como a existência da derivada mista de segunda ordem. Em outras palavras, f deve ser de classe C^2. **Observação**: o Exercício Programa **não tem implementações para realizar tais checagens**. Está a cargo do usuário colocar uma função que respeite tais condições.

Analogo ao processo que fizemos na seção do caso bilinear, conjuntamente com a suposição da existência das derivadas parciais já mencionadas, reescrevemos todas as 16 condições de interpolação do enunciado abaixo:

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

$$
\begin{aligned}
s_{ij}^C(x_{i+1}, y_{j+1}) &= f(x_{i+1}, y_{j+1}) \\
\frac{\partial s_{ij}^C}{\partial y}(x_i, y_j) &= \frac{\partial f}{\partial y}(x_i, y_j) \\
\frac{\partial s_{ij}^C}{\partial x}(x_i, y_j) &= \frac{\partial f}{\partial x}(x_i, y_j) \\
\frac{\partial s_{ij}^C}{\partial y}(x_i, y_{j+1}) &= \frac{\partial f}{\partial y}(x_i, y_{j+1}) \\
\frac{\partial s_{ij}^C}{\partial x}(x_i, y_{j+1}) &= \frac{\partial f}{\partial x}(x_i, y_{j+1}) \\
\frac{\partial s_{ij}^C}{\partial y}(x_{i+1}, y_j) &= \frac{\partial f}{\partial y}(x_{i+1}, y_j) \\
\frac{\partial s_{ij}^C}{\partial x}(x_{i+1}, y_j) &= \frac{\partial f}{\partial x}(x_{i+1}, y_j) \\
\frac{\partial s_{ij}^C}{\partial y}(x_{i+1}, y_{j+1}) &= \frac{\partial f}{\partial y}(x_{i+1}, y_{j+1}) \\
\frac{\partial s_{ij}^C}{\partial x}(x_{i+1}, y_{j+1}) &= \frac{\partial f}{\partial x}(x_{i+1}, y_{j+1}) \\
\frac{\partial^2 s_{ij}^C}{\partial x \partial y}(x_i, y_j) &= \frac{\partial^2 f}{\partial x \partial y}(x_i, y_j) \\
\frac{\partial^2 s_{ij}^C}{\partial x \partial y}(x_i, y_{j+1}) &= \frac{\partial^2 f}{\partial x \partial y}(x_i, y_{j+1}) \\
\frac{\partial^2 s_{ij}^C}{\partial x \partial y}(x_{i+1}, y_j) &= \frac{\partial^2 f}{\partial x \partial y}(x_{i+1}, y_j) \\
\frac{\partial^2 s_{ij}^C}{\partial x \partial y}(x_{i+1}, y_{j+1}) &= \frac{\partial^2 f}{\partial x \partial y}(x_{i+1}, y_{j+1})
\end{aligned}
$$

E reescrevamos as condições de interpolação no seguinte formato matricial:

$$
\begin{pmatrix}
s_{ij}^C(x_i, y_j) & s_{ij}^C(x_i, y_{j+1}) & \frac{\partial s_{ij}^C}{\partial y}(x_i, y_j) & \frac{\partial s_{ij}^C}{\partial y}(x_i, y_{j+1}) \\
s_{ij}^C(x_{i+1}, y_j) & s_{ij}^C(x_{i+1}, y_{j+1}) & \frac{\partial s_{ij}^C}{\partial y}(x_{i+1}, y_j) & \frac{\partial s_{ij}^C}{\partial y}(x_{i+1}, y_{j+1}) \\
\frac{\partial s_{ij}^C}{\partial x}(x_i, y_j) & \frac{\partial s_{ij}^C}{\partial x}(x_i, y_{j+1}) & \frac{\partial^2 s_{ij}^C}{\partial x \partial y}(x_i, y_j) & \frac{\partial^2 s_{ij}^C}{\partial x \partial y}(x_i, y_{j+1}) \\
\frac{\partial s_{ij}^C}{\partial x}(x_{i+1}, y_j) & \frac{\partial s_{ij}^C}{\partial x}(x_{i+1}, y_{j+1}) & \frac{\partial^2 s_{ij}^C}{\partial x \partial y}(x_{i+1}, y_j) & \frac{\partial^2 s_{ij}^C}{\partial x \partial y}(x_{i+1}, y_{j+1})
\end{pmatrix}
=
\begin{pmatrix}
f(x_i, y_j) & f(x_i, y_{j+1}) & \frac{\partial f}{\partial y}(x_i, y_j) & \frac{\partial f}{\partial y}(x_i, y_{j+1}) \\
f(x_{i+1}, y_j) & f(x_{i+1}, y_{j+1}) & \frac{\partial f}{\partial y}(x_{i+1}, y_j) & \frac{\partial f}{\partial y}(x_{i+1}, y_{j+1}) \\
\frac{\partial f}{\partial x}(x_i, y_j) & \frac{\partial f}{\partial x}(x_i, y_{j+1}) & \frac{\partial^2 f}{\partial x \partial y}(x_i, y_j) & \frac{\partial^2 f}{\partial x \partial y}(x_i, y_{j+1}) \\
\frac{\partial f}{\partial x}(x_{i+1}, y_j) & \frac{\partial f}{\partial x}(x_{i+1}, y_{j+1}) & \frac{\partial^2 f}{\partial x \partial y}(x_{i+1}, y_j) & \frac{\partial^2 f}{\partial x \partial y}(x_{i+1}, y_{j+1})
\end{pmatrix}
$$

Consideremos, então, a forma matricial do polinômio $s_{ij}^C(x, y)$ fornecida pelo enunciado, reproduzida abaixo:

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

Temos, finalmente, para cada um dos 16 polinômios $s_{ij}^C(x, y)$:

1) $s_{ij}^C(x_i, y_j)$

$$
= \begin{pmatrix} 1 & (x_i - x_i) & (x_i - x_i)^2 & (x_i - x_i)^3 \end{pmatrix}
\begin{pmatrix}
a_{00} & a_{01} & a_{02} & a_{03} \\
a_{10} & a_{11} & a_{12} & a_{13} \\
a_{20} & a_{21} & a_{22} & a_{23} \\
a_{30} & a_{31} & a_{32} & a_{33}
\end{pmatrix}
\begin{pmatrix}
1 \\ (y_j - y_j) \\ (y_j - y_j)^2 \\ (y_j - y_j)^3
\end{pmatrix}
$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$= a_{00}$$

2) $s_{ij}^C(x_i, y_{j+1})$

$$= \begin{pmatrix} 1 & (x_i - x_i) & (x_i - x_i)^2 & (x_i - x_i)^3 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 1 \\ (y_{j+1} - y_j) \\ (y_{j+1} - y_j)^2 \\ (y_{j+1} - y_j)^3 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 1 \\ h_y \\ h_y^2 \\ h_y^3 \end{pmatrix}$$

$$= a_{00} + a_{01}h_y + a_{02}h_y^2 + a_{03}h_y^3$$

3) $s_{ij}^C(x_{i+1}, y_j)$

$$= \begin{pmatrix} 1 & (x_{i+1} - x_i) & (x_{i+1} - x_i)^2 & (x_{i+1} - x_i)^3 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 1 \\ (y_j - y_j) \\ (y_j - y_j)^2 \\ (y_j - y_j)^3 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & h_x & h_x^2 & h_x^3 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$= a_{00} + a_{10}h_x + a_{20}h_x^2 + a_{30}h_x^3$$

4) $s_{ij}^C(x_{i+1}, y_{j+1})$

$$= \begin{pmatrix} 1 & (x_{i+1} - x_i) & (x_{i+1} - x_i)^2 & (x_{i+1} - x_i)^3 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 1 \\ (y_{j+1} - y_j) \\ (y_{j+1} - y_j)^2 \\ (y_{j+1} - y_j)^3 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & h_x & h_x^2 & h_x^3 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 1 \\ h_y \\ h_y^2 \\ h_y^3 \end{pmatrix}$$

$$= a_{00} + a_{01}h_y + a_{02}h_y^2 + a_{03}h_y^3 + a_{10}h_x + a_{11}h_xh_y + a_{12}h_xh_y^2 + a_{13}h_xh_y^3 + a_{20}h_x^2 + a_{21}h_x^2h_y + a_{22}h_x^2h_y^2 + a_{23}h_x^2h_y^3 + a_{30}h_x^3 + a_{31}h_x^3h_y + a_{32}h_x^3h_y^2 + a_{33}h_x^3h_y^3$$

5) $\frac{\partial s_{ij}^C}{\partial y}(x_i, y_j)$

$$= \begin{pmatrix} 1 & (x_i - x_i) & (x_i - x_i)^2 & (x_i - x_i)^3 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 2(y_j - y_j) \\ 3(y_j - y_j)^2 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$= a_{01}$$

6) $\frac{\partial s_{ij}^C}{\partial y}(x_i, y_{j+1})$

$$= \begin{pmatrix} 1 & (x_i - x_i) & (x_i - x_i)^2 & (x_i - x_i)^3 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 2(y_{j+1} - y_j) \\ 3(y_{j+1} - y_j)^2 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 2h_y \\ 3h_y^2 \end{pmatrix}$$

$$= a_{01} + 2a_{02}h_y + 3a_{03}h_y^2$$

7) $\frac{\partial s_{ij}^C}{\partial y}(x_{i+1}, y_j)$

$$= \begin{pmatrix} 1 & (x_{i+1} - x_i) & (x_{i+1} - x_i)^2 & (x_{i+1} - x_i)^3 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 2(y_j - y_j) \\ 3(y_j - y_j)^2 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & h_x & h_x^2 & h_x^3 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$= a_{01} + a_{11}h_x + a_{21}h_x^2 + a_{31}h_x^3$$

8) $\frac{\partial s_{ij}^C}{\partial y}(x_{i+1}, y_{j+1})$

$$= \begin{pmatrix} 1 & (x_{i+1} - x_i) & (x_{i+1} - x_i)^2 & (x_{i+1} - x_i)^3 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 2(y_{j+1} - y_j) \\ 3(y_{j+1} - y_j)^2 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & h_x & h_x^2 & h_x^3 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 2h_y \\ 3h_y^2 \end{pmatrix}$$

$$= a_{01} + a_{11}h_x + a_{21}h_x^2 + a_{31}h_x^3 + 2(a_{02}h_y + a_{12}h_xh_y + a_{22}h_x^2h_y + a_{32}h_x^3h_y) + 3(a_{03}h_y^2 + a_{13}h_xh_y^2 + a_{23}h_x^2h_y^2 + 3a_{33}h_x^3h_y^2)$$

9) $\frac{\partial s_{ij}^C}{\partial x}(x_i, y_j)$

$$= \begin{pmatrix} 0 & 1 & 2(x_i - x_i) & 3(x_i - x_i)^2 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 1 \\ (y_j - y_j) \\ (y_j - y_j)^2 \\ (y_j - y_j)^3 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$= a_{10}$$

10) $\frac{\partial s_{ij}^C}{\partial x}(x_i, y_{j+1})$

$$= \begin{pmatrix} 0 & 1 & 2(x_i - x_i) & 3(x_i - x_i)^2 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 1 \\ (y_{j+1} - y_j) \\ (y_{j+1} - y_j)^2 \\ (y_{j+1} - y_j)^3 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 1 \\ h_y \\ h_y^2 \\ h_y^3 \end{pmatrix}$$

$$= a_{10} + a_{11}h_y + a_{12}h_y^2 + a_{13}h_y^3$$

11) $\frac{\partial s_{ij}^C}{\partial x}(x_{i+1}, y_j)$

$$= \begin{pmatrix} 0 & 1 & 2(x_{i+1} - x_i) & 3(x_{i+1} - x_i)^2 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 1 \\ (y_j - y_j) \\ (y_j - y_j)^2 \\ (y_j - y_j)^3 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 1 & 2h_x & 3h_x^2 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$= a_{10} + 2a_{20}h_x + 3a_{30}h_x^2$$

12) $\frac{\partial s_{ij}^C}{\partial x}(x_{i+1}, y_{j+1})$

$$= \begin{pmatrix} 0 & 1 & 2(x_{i+1} - x_i) & 3(x_{i+1} - x_i)^2 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 1 \\ (y_{j+1} - y_j) \\ (y_{j+1} - y_j)^2 \\ (y_{j+1} - y_j)^3 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 1 & 2h_x & 3h_x^2 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 1 \\ h_y \\ h_y^2 \\ h_y^3 \end{pmatrix}$$

$$= a_{10} + a_{11}h_y + a_{12}h_y^2 + a_{13}h_y^3 + 2(a_{20}h_x + a_{21}h_xh_y + a_{22}h_xh_y^2 + a_{23}h_xh_y^3) + 3(a_{30}h_x^2 + a_{31}h_x^2h_y + a_{32}h_x^2h_y^2 + a_{33}h_x^2h_y^3)$$

13) $\frac{\partial^2 s_{ij}^C}{\partial x \partial y}(x_i, y_j)$

$$= \begin{pmatrix} 0 & 1 & 2(x_i - x_i) & 3(x_i - x_i)^2 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 2(y_j - y_j) \\ 3(y_j - y_j)^2 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$= a_{11}$$

14) $\frac{\partial^2 s_{ij}^C}{\partial x \partial y}(x_i, y_{j+1})$

$$= \begin{pmatrix} 0 & 1 & 2(x_i - x_i) & 3(x_i - x_i)^2 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 2(y_{j+1} - y_j) \\ 3(y_{j+1} - y_j)^2 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 2h_y \\ 3h_y^2 \end{pmatrix}$$

$$= a_{11} + 2a_{12}h_y + 3a_{13}h_y^2$$

15) $\frac{\partial^2 s_{ij}^C}{\partial x \partial y}(x_{i+1}, y_j)$

$$= \begin{pmatrix} 0 & 1 & 2(x_{i+1} - x_i) & 3(x_{i+1} - x_i)^2 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 2(y_j - y_j) \\ 3(y_j - y_j)^2 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 1 & 2h_x & 3h_x^2 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$= a_{11} + 2a_{21}h_x + 3a_{31}h_x^2$$

16) $\frac{\partial^2 s_{ij}^C}{\partial x \partial y}(x_{i+1}, y_{j+1})$

$$= \begin{pmatrix} 0 & 1 & 2(x_{i+1} - x_i) & 3(x_{i+1} - x_i)^2 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 2(y_{j+1} - y_j) \\ 3(y_{j+1} - y_j)^2 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 1 & 2h_x & 3h_x^2 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 2h_y \\ 3h_y^2 \end{pmatrix}$$

$$= a_{11} + 2a_{21}h_x + 3a_{31}h_x^2 + 2(a_{12}h_y + 2a_{22}h_xh_y + 3a_{32}h_x^2h_y) + 3(a_{13}h_y^2 + 2a_{23}h_xh_y^2 + 3a_{33}h_x^2h_y^2)$$

Uma vez com estes 16 valores obtidos, podemos substituir as instâncias de $s_{ij}^C(x,y)$ na nossa matriz e obter a seguinte expressão matricial:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & h_x & h_x^2 & h_x^3 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2h_x & 3h_x^2 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & h_y & 1 & 1 \\ 0 & h_y^2 & 0 & 2h_y \\ 0 & h_y^3 & 0 & 3h_y^2 \end{pmatrix} = \begin{pmatrix} f(x_i,y_j) & f(x_i,y_{j+1}) & \frac{\partial f}{\partial y}(x_i,y_j) & \frac{\partial f}{\partial y}(x_i,y_{j+1}) \\ f(x_{i+1},y_j) & f(x_{i+1},y_{j+1}) & \frac{\partial f}{\partial y}(x_{i+1},y_j) & \frac{\partial f}{\partial y}(x_{i+1},y_{j+1}) \\ \frac{\partial f}{\partial x}(x_i,y_j) & \frac{\partial f}{\partial x}(x_i,y_{j+1}) & \frac{\partial^2 f}{\partial x \partial y}(x_i,y_j) & \frac{\partial^2 f}{\partial x \partial y}(x_i,y_{j+1}) \\ \frac{\partial f}{\partial x}(x_{i+1},y_j) & \frac{\partial f}{\partial x}(x_{i+1},y_{j+1}) & \frac{\partial^2 f}{\partial x \partial y}(x_{i+1},y_j) & \frac{\partial^2 f}{\partial x \partial y}(x_{i+1},y_{j+1}) \end{pmatrix}$$

Precisamos isolar os coeficientes. Finalmente, supondo que a matrizes que estão multiplicando a matriz de coeficientes a_{ij} sejam invertíveis (como uma é transposta da outra, apesar dos termos h_x e h_y, basta que apenas uma seja invertível que a outra também será, então a rigor fazemos apenas uma suposição, e não duas), temos que:

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{-3}{h_x^2} & \frac{3}{h_x^2} & \frac{-2}{h_x} & \frac{-1}{h_x} \\ \frac{2}{h_x^3} & \frac{-2}{h_x^3} & \frac{1}{h_x^2} & \frac{1}{h_x^2} \end{pmatrix} \begin{pmatrix} f(x_i,y_j) & f(x_i,y_{j+1}) & \frac{\partial f}{\partial y}(x_i,y_j) & \frac{\partial f}{\partial y}(x_i,y_{j+1}) \\ f(x_{i+1},y_j) & f(x_{i+1},y_{j+1}) & \frac{\partial f}{\partial y}(x_{i+1},y_j) & \frac{\partial f}{\partial y}(x_{i+1},y_{j+1}) \\ \frac{\partial f}{\partial x}(x_i,y_j) & \frac{\partial f}{\partial x}(x_i,y_{j+1}) & \frac{\partial^2 f}{\partial x \partial y}(x_i,y_j) & \frac{\partial^2 f}{\partial x \partial y}(x_i,y_{j+1}) \\ \frac{\partial f}{\partial x}(x_{i+1},y_j) & \frac{\partial f}{\partial x}(x_{i+1},y_{j+1}) & \frac{\partial^2 f}{\partial x \partial y}(x_{i+1},y_j) & \frac{\partial^2 f}{\partial x \partial y}(x_{i+1},y_{j+1}) \end{pmatrix} \begin{pmatrix} 1 & 0 & \frac{-3}{h_y^2} & \frac{2}{h_y^3} \\ 0 & 0 & \frac{3}{h_y^2} & \frac{-2}{h_y^3} \\ 0 & 1 & \frac{-2}{h_y} & \frac{1}{h_y^2} \\ 0 & 0 & \frac{-1}{h_y} & \frac{1}{h_y^2} \end{pmatrix}$$

Uma maneira eficiente de calcular os coeficientes para realizarmos o método da interpolação bicúbica.

4 Dedução das derivadas parciais em x, em y e mistas.

Neste capítulos deduziremos formas de calcular as derivadas parciais $\frac{\partial f}{\partial x}(x, y)$ e $\frac{\partial f}{\partial y}(x, y)$ e a derivada parcial mista $\frac{\partial^2 f}{\partial x \partial y}(x, y)$ a partir da expansão do polinômio de Taylor em torno do ponto (x_i, y_j). Para tal, serão usadas as expansões de 3 pontos: centrada, para frente, para trás, para cima e para baixo, a depender do caso em que o ponto (x_i, y_j) está em alguma borda da imagem ou não.

4.1 Expansão do polinômio de Taylor para funções de duas variáveis

Conforme mencionado anteriormente, será usada a expansão do polinômio de Taylor para função de duas variáveis para aproximar as derivadas. Abaixo é apresentada a forma do polinômio P_n com a expansão em torno do ponto (x_i, y_j).

$$
\begin{gathered}
P_n(x,y) = f(x_i,y_j) + \frac{\partial f}{\partial x}(x_i,y_j)(x-x_i) + \frac{\partial f}{\partial y}(x_i,y_j)(y-y_j) + \\
+\frac{1}{2}\left(\frac{\partial^2 f}{\partial x^2}(x_i,y_j)(x-x_i)^2 + 2\frac{\partial^2 f}{\partial x \partial y}(x_i,y_j)(x-x_i)(y-y_j) + \frac{\partial^2 f}{\partial y^2}(x_i,y_j)(y-y_j)^2\right) + \\
+ \cdots + \frac{1}{n!}\sum_{k=0}^{n}\binom{n}{k}\frac{\partial^n f}{\partial x^{n-k}\partial y^k}(x_i,y_j)(x-x_i)^{n-k}(y-y_j)^k
\end{gathered}
$$

Fonte: https://en.wikipedia.org/wiki/Taylor_series

4.2 Derivada parcial em x

Começaremos com a derivada parcial $\frac{\partial f}{\partial x}(x, y)$, considerando a nossa malha de pontos no plano cartesiano. A depender da posição do ponto (x_i, y_j) que queremos aproximar a derivada parcial na malha, devemos considerar um dos três casos abaixo:

1. O ponto (x_i, y_j) não pertence a uma borda da imagem.

2. O ponto (x_i, y_j) pertence à borda esquerda da imagem.

3. O ponto (x_i, y_j) pertence à borda direita da imagem.

Como estamos considerando $\frac{\partial f}{\partial x}(x, y)$ no plano cartesiano, podemos desconsiderar os casos do ponto pertencer às bordas superior ou inferior da imagem. A seguir, demonstraremos os três itens enumerados acima.

4.2.1 Caso 1: o ponto não pertence a uma borda

Usando o ponto (x, y_j) pois o "deslocamento" é exclusivamente horizontal, temos:

I) $f(x+h_x, y_j) = f(x, y_j) + h_x \left(\frac{\partial f}{\partial x}(x, y_j)(x - x_i) + \frac{\partial f}{\partial y}(x, y_j)(y_j - y_j) \right) +$
$+ \frac{h_x^2}{2} \left(\frac{\partial^2 f}{\partial x^2}(x, y_j)(x - x_i)^2 + 2\frac{\partial^2 f}{\partial x \partial y}(x, y_j)(x - x_i)(y_j - y_j) + \frac{\partial^2 f}{\partial y^2}(x, y_j)(y_j - y_j)^2 \right) + O(h_x^3)$
$= f(x, y_j) + h_x \frac{\partial f}{\partial x}(x, y_j)(x - x_i) + \frac{h_x^2}{2} \frac{\partial^2 f}{\partial x^2}(x, y_j)(x - x_i)^2 + O(h_x^3)$

II) $f(x-h_x, y_j) = f(x, y_j) - h_x \left(\frac{\partial f}{\partial x}(x, y_j)(x - x_i) + \frac{\partial f}{\partial y}(x, y_j)(y_j - y_j) \right) +$
$+ \frac{h_x^2}{2} \left(\frac{\partial^2 f}{\partial x^2}(x, y_j)(x - x_i)^2 + 2\frac{\partial^2 f}{\partial x \partial y}(x, y_j)(x - x_i)(y_j - y_j) + \frac{\partial^2 f}{\partial y^2}(x, y_j)(y_j - y_j)^2 \right) + O(h_x^3)$
$= f(x, y_j) - h_x \frac{\partial f}{\partial x}(x, y_j)(x - x_i) + \frac{h_x^2}{2} \frac{\partial^2 f}{\partial x^2}(x, y_j)(x - x_i)^2 + O(h_x^3)$

e de **I - II** segue que:

$f(x+h_x, y_j) - f(x-h_x, y_j) = 2h_x \frac{\partial f}{\partial x}(x, y_j)(x - x_i) + O(h_x^3)$

e portanto:

$\frac{\partial f}{\partial x}(x, y_j)(x - x_i) = \frac{f(x+h_x, y_j) - f(x-h_x, y_j)}{2h_x} + O(h_x^2)$

4.2.2 Caso 2: o ponto pertence à borda esquerda

Usando o ponto (x, y_j) pois o "deslocamento" é exclusivamente horizontal, temos:

I) $f(x+h_x, y_j) = f(x, y_j) + h_x \left(\frac{\partial f}{\partial x}(x, y_j)(x - x_i) + \frac{\partial f}{\partial y}(x, y_j)(y_j - y_j) \right) +$
$+ \frac{h_x^2}{2} \left(\frac{\partial^2 f}{\partial x^2}(x, y_j)(x - x_i)^2 + 2\frac{\partial^2 f}{\partial x \partial y}(x, y_j)(x - x_i)(y_j - y_j) + \frac{\partial^2 f}{\partial y^2}(x, y_j)(y_j - y_j)^2 \right) + O(h_x^3)$
$= f(x, y_j) + h_x \frac{\partial f}{\partial x}(x, y_j)(x - x_i) + \frac{h_x^2}{2} \frac{\partial^2 f}{\partial x^2}(x, y_j)(x - x_i)^2 + O(h_x^3)$

II) $f(x+2h_x, y_j) = f(x, y_j) + 2h_x \left(\frac{\partial f}{\partial x}(x, y_j)(x - x_i) + \frac{\partial f}{\partial y}(x, y_j)(y_j - y_j) \right) +$
$+ 2h_x^2 \left(\frac{\partial^2 f}{\partial x^2}(x, y_j)(x - x_i)^2 + 2\frac{\partial^2 f}{\partial x \partial y}(x, y_j)(x - x_i)(y_j - y_j) + \frac{\partial^2 f}{\partial y^2}(x, y_j)(y_j - y_j)^2 \right) + O(h_x^3)$
$= f(x, y_j) + 2h_x \frac{\partial f}{\partial x}(x, y_j)(x - x_i) + 2h_x^2 \frac{\partial^2 f}{\partial x^2}(x, y_j)(x - x_i)^2 + O(h_x^3)$

e de **4I - II** segue que:

$4f(x+h_x, y_j) - f(x+2h_x, y_j) = 3f(x, y_j) + 2h_x \frac{\partial f}{\partial x}(x, y_j)(x - x_i) + O(h_x^3)$

e portanto:

$\frac{\partial f}{\partial x}(x, y_j)(x - x_i) = \frac{-3f(x, y_j) + 4f(x+h_x, y_j) - f(x+2h_x, y_j)}{2h_x} + O(h_x^2)$

4.2.3 Caso 3: o ponto pertence à borda direita

Usando o ponto (x, y_j) pois o "deslocamento" é exclusivamente horizontal, temos:

I) $f(x-h_x, y_j) = f(x, y_j) - h_x\left(\frac{\partial f}{\partial x}(x, y_j)(x - x_i) + \frac{\partial f}{\partial y}(x, y_j)(y_j - y_j)\right) +$
$+\frac{h_x^2}{2}\left(\frac{\partial^2 f}{\partial x^2}(x, y_j)(x - x_i)^2 + 2\frac{\partial^2 f}{\partial x \partial y}(x, y_j)(x - x_i)(y_j - y_j) + \frac{\partial^2 f}{\partial y^2}(x, y_j)(y_j - y_j)^2\right) + O(h_x^3)$
$= f(x, y_j) - h_x\frac{\partial f}{\partial x}(x, y_j)(x - x_i) + \frac{h_x^2}{2}\frac{\partial^2 f}{\partial x^2}(x, y_j)(x - x_i)^2 + O(h_x^3)$

II) $f(x-2h_x, y_j) = f(x, y_j) - 2h_x\left(\frac{\partial f}{\partial x}(x, y_j)(x - x_i) + \frac{\partial f}{\partial y}(x, y_j)(y_j - y_j)\right) +$
$+2h_x^2\left(\frac{\partial^2 f}{\partial x^2}(x, y_j)(x - x_i)^2 + 2\frac{\partial^2 f}{\partial x \partial y}(x, y_j)(x - x_i)(y_j - y_j) + \frac{\partial^2 f}{\partial y^2}(x, y_j)(y_j - y_j)^2\right) + O(h_x^3)$
$= f(x, y_j) - 2h_x\frac{\partial f}{\partial x}(x, y_j)(x - x_i) + 2h_x^2\frac{\partial^2 f}{\partial x^2}(x, y_j)(x - x_i)^2 + O(h_x^3)$

e de **4I - II** segue que:

$4f(x-h_x, y_j) - f(x-2h_x, y_j) = 3f(x, y_j) - 2h_x\frac{\partial f}{\partial x}(x, y_j)(x - x_i) + O(h_x^3)$

e portanto:

$\frac{\partial f}{\partial x}(x, y_j)(x - x_i) = \frac{3f(x,y_j) - 4f(x-h_x,y_j) + f(x-2h_x,y_j)}{2h_x} + O(h_x^2)$

4.3 Derivada parcial em y

Tomemos agora a derivada parcial $\frac{\partial f}{\partial y}(x, y)$, considerando a nossa malha de pontos no plano cartesiano. A depender da posição do ponto (x_i, y_j) que queremos aproximar a derivada parcial na malha, devemos considerar um dos três casos abaixo:

1. O ponto (x_i, y_j) não pertence a uma borda da imagem.

2. O ponto (x_i, y_j) pertence à borda inferior da imagem.

3. O ponto (x_i, y_j) pertence à borda superior da imagem.

Como estamos considerando $\frac{\partial f}{\partial y}(x, y)$ no plano cartesiano, podemos desconsiderar os casos do ponto pertencer às bordas laterais da imagem. A seguir, demonstraremos os três itens enumerados acima.

4.3.1 Caso 1: o ponto não pertence a uma borda

Usando o ponto (x_i, y) pois o "deslocamento" é exclusivamente vertical, temos:

I) $f(x_i, y+h_y) = f(x_i, y) + h_y\left(\frac{\partial f}{\partial x}(x_i, y)(x_i - x_i) + \frac{\partial f}{\partial y}(x_i, y)(y - y_j)\right) +$
$+\frac{h_y^2}{2}\left(\frac{\partial^2 f}{\partial x^2}(x_i, y)(x_i - x_i)^2 + 2\frac{\partial^2 f}{\partial x \partial y}(x_i, y)(x_i - x_i)(y - y_j) + \frac{\partial^2 f}{\partial y^2}(x_i, y)(y - y_j)^2\right) + O(h_y^3)$
$= f(x_i, y) + h_y\frac{\partial f}{\partial y}(x_i, y)(y - y_j) + \frac{h_y^2}{2}\frac{\partial^2 f}{\partial y^2}(x_i, y)(y - y_j)^2 + O(h_y^3)$

II) $f(x_i, y-h_y) = f(x_i, y) - h_y\left(\frac{\partial f}{\partial x}(x_i, y)(x_i - x_i) + \frac{\partial f}{\partial y}(x_i, y)(y - y_j)\right) +$
$+\frac{h_y^2}{2}\left(\frac{\partial^2 f}{\partial x^2}(x_i, y)(x_i - x_i)^2 + 2\frac{\partial^2 f}{\partial x \partial y}(x_i, y)(x_i - x_i)(y - y_j) + \frac{\partial^2 f}{\partial y^2}(x_i, y)(y - y_j)^2\right) + O(h_y^3)$
$= f(x_i, y) - h_y\frac{\partial f}{\partial y}(x_i, y)(y - y_j) + \frac{h_y^2}{2}\frac{\partial^2 f}{\partial y^2}(x_i, y)(y - y_j)^2 + O(h_y^3)$

e de **I - II** segue que:

$f(x_i, y + h_y) - f(x_i, y - h_y) = 2h_y \frac{\partial f}{\partial y}(x_i, y)(y - y_j) + O(h_y^3)$

e portanto:

$\frac{\partial f}{\partial y}(x_i, y)(y - y_j) = \frac{f(x_i, y+h_y) - f(x_i, y-h_y)}{2h_y} + O(h_y^2)$

4.3.2 Caso 2: o ponto pertence à borda inferior

Usando o ponto (x_i, y) pois o "deslocamento" é exclusivamente vertical, temos:

I) $f(x_i, y + h_y) = f(x_i, y) + h_y \left(\frac{\partial f}{\partial x}(x_i, y)(x_i - x_i) + \frac{\partial f}{\partial y}(x_i, y)(y - y_j) \right) +$
$+ \frac{h_y^2}{2} \left(\frac{\partial^2 f}{\partial x^2}(x_i, y)(x_i - x_i)^2 + 2\frac{\partial^2 f}{\partial x \partial y}(x_i, y)(x_i - x_i)(y - y_j) + \frac{\partial^2 f}{\partial y^2}(x_i, y)(y - y_j)^2 \right) + O(h_y^3)$
$= f(x_i, y) + h_y \frac{\partial f}{\partial y}(x_i, y)(y - y_j) + \frac{h_y^2}{2} \frac{\partial^2 f}{\partial y^2}(x_i, y)(y - y_j)^2 + O(h_y^3)$

II) $f(x_i, y + 2h_y) = f(x_i, y) + 2h_y \left(\frac{\partial f}{\partial x}(x_i, y)(x_i - x_i) + \frac{\partial f}{\partial y}(x_i, y)(y - y_j) \right) +$
$+ 2h_y^2 \left(\frac{\partial^2 f}{\partial x^2}(x_i, y)(x_i - x_i)^2 + 2\frac{\partial^2 f}{\partial x \partial y}(x_i, y)(x_i - x_i)(y - y_j) + \frac{\partial^2 f}{\partial y^2}(x_i, y)(y - y_j)^2 \right) + O(h_y^3)$
$= f(x_i, y) + 2h_y \frac{\partial f}{\partial y}(x_i, y)(y - y_j) + 2h_y^2 \frac{\partial^2 f}{\partial y^2}(x_i, y)(y - y_j)^2 + O(h_y^3)$

e de **4I - II** segue que:

$4f(x_i, y + h_y) - f(x_i, y + 2h_y) = 3f(x_i, y) + 2h_y \frac{\partial f}{\partial y}(x_i, y)(y - y_j) + O(h_y^3)$

e portanto:

$\frac{\partial f}{\partial y}(x_i, y)(y - y_j) = \frac{-3f(x_i, y) + 4f(x_i, y+h_y) - f(x_i, y+2h_y)}{2h_y} + O(h_y^2)$

4.3.3 Caso 3: o ponto pertence à borda superior

Usando o ponto (x_i, y) pois o "deslocamento" é exclusivamente vertical, temos:

I) $f(x_i, y - h_y) = f(x_i, y) - h_y \left(\frac{\partial f}{\partial x}(x_i, y)(x_i - x_i) + \frac{\partial f}{\partial y}(x_i, y)(y - y_j) \right) +$
$+ \frac{h_y^2}{2} \left(\frac{\partial^2 f}{\partial x^2}(x_i, y)(x_i - x_i)^2 + 2\frac{\partial^2 f}{\partial x \partial y}(x_i, y)(x_i - x_i)(y - y_j) + \frac{\partial^2 f}{\partial y^2}(x_i, y)(y - y_j)^2 \right) + O(h_y^3)$
$= f(x_i, y) - h_y \frac{\partial f}{\partial y}(x_i, y)(y - y_j) + \frac{h_y^2}{2} \frac{\partial^2 f}{\partial y^2}(x_i, y)(y - y_j)^2 + O(h_y^3)$

II) $f(x_i, y - 2h_y) = f(x_i, y) - 2h_y \left(\frac{\partial f}{\partial x}(x_i, y)(x_i - x_i) + \frac{\partial f}{\partial y}(x_i, y)(y - y_j) \right) +$
$+ 2h_y^2 \left(\frac{\partial^2 f}{\partial x^2}(x_i, y)(x_i - x_i)^2 + 2\frac{\partial^2 f}{\partial x \partial y}(x_i, y)(x_i - x_i)(y - y_j) + \frac{\partial^2 f}{\partial y^2}(x_i, y)(y - y_j)^2 \right) + O(h_y^3)$
$= f(x_i, y) - 2h_y \frac{\partial f}{\partial y}(x_i, y)(y - y_j) + 2h_y^2 \frac{\partial^2 f}{\partial y^2}(x_i, y)(y - y_j)^2 + O(h_y^3)$

e de **4I - II** segue que:

$4f(x_i, y - h_y) - f(x_i, y - 2h_y) = 3f(x_i, y) - 2h_y \frac{\partial f}{\partial y}(x_i, y)(y - y_j) + O(h_y^3)$

e portanto:

$\frac{\partial f}{\partial y}(x_i, y)(y - y_j) = \frac{3f(x_i, y) - 4f(x_i, y-h_y) + f(x_i, y-2h_y)}{2h_y} + O(h_y^2)$

4.4 Derivada parcial mista

Tomemos agora a derivada parcial mista $\frac{\partial^2 f}{\partial x \partial y}(x, y)$, considerando a nossa malha de pontos no plano cartesiano. A depender da posição do ponto (x_i, y_j) que queremos aproximar a derivada parcial na malha, devemos considerar um dos três casos abaixo:

1. O ponto (x_i, y_j) não pertence a uma borda da imagem.

2. O ponto (x_i, y_j) pertence à borda esquerda da imagem.

3. O ponto (x_i, y_j) pertence à borda direita da imagem.

Como estamos considerando $\frac{\partial^2 f}{\partial x \partial y}(x, y)$ no plano cartesiano, podemos desconsiderar os casos do ponto pertencer às bordas superior ou inferior da imagem. A seguir, demonstraremos os três itens enumerados acima.

Observação: perceba que estamos calculando a derivada parcial com relação à x na função $\frac{\partial f}{\partial y}(x, y)$, e como isso significa considerar o eixo das abcissas, não é necessário trabalhar os casos das bordas superior e inferior.

4.4.1 Caso 1: o ponto não pertence a uma borda

Usando o ponto (x, y_j) pois o "deslocamento" é exclusivamente horizontal, temos:

I) $\frac{\partial f}{\partial y}(x + h_x, y_j) = \frac{\partial f}{\partial y}(x, y_j) + h_x \left(\frac{\partial^2 f}{\partial x \partial y}(x, y_j)(x - x_i) + \frac{\partial^2 f}{\partial y^2}(x, y_j)(y_j - y_j) \right) +$
$+ \frac{h_x^2}{2} \left(\frac{\partial^3 f}{\partial x^2 \partial y}(x, y_j)(x - x_i)^2 + 2\frac{\partial^3 f}{\partial x \partial y^2}(x, y_j)(x - x_i)(y_j - y_j) + \frac{\partial^3 f}{\partial y^3}(x, y_j)(y_j - y_j)^2 \right) + O(h_x^3)$
$= \frac{\partial f}{\partial y}(x, y_j) + h_x \frac{\partial^2 f}{\partial x \partial y}(x, y_j)(x - x_i) + \frac{h_x^2}{2} \frac{\partial^3 f}{\partial x^2 \partial y}(x, y_j)(x - x_i)^2 + O(h_x^3)$

II) $\frac{\partial f}{\partial y}(x - h_x, y_j) = \frac{\partial f}{\partial y}(x, y_j) - h_x \left(\frac{\partial^2 f}{\partial x \partial y}(x, y_j)(x - x_i) + \frac{\partial^2 f}{\partial y^2}(x, y_j)(y_j - y_j) \right) +$
$+ \frac{h_x^2}{2} \left(\frac{\partial^3 f}{\partial x^2 \partial y}(x, y_j)(x - x_i)^2 + 2\frac{\partial^3 f}{\partial x \partial y^2}(x, y_j)(x - x_i)(y_j - y_j) + \frac{\partial^3 f}{\partial y^3}(x, y_j)(y_j - y_j)^2 \right) + O(h_x^3)$
$= \frac{\partial f}{\partial y}(x, y_j) - h_x \frac{\partial^2 f}{\partial x \partial y}(x, y_j)(x - x_i) + \frac{h_x^2}{2} \frac{\partial^3 f}{\partial x^2 \partial y}(x, y_j)(x - x_i)^2 + O(h_x^3)$

e de **I - II** segue que:
$\frac{\partial f}{\partial y}(x + h_x, y_j) - \frac{\partial f}{\partial y}(x - h_x, y_j) = 2h_x \frac{\partial^2 f}{\partial x \partial y}(x, y_j)(x - x_i) + O(h_x^3)$
e portanto:
$\frac{\partial^2 f}{\partial x \partial y}(x, y_j)(x - x_i) = \frac{\frac{\partial f}{\partial y}(x + h_x, y_j) - \frac{\partial f}{\partial y}(x - h_x, y_j)}{2h_x} + O(h_x^2)$

4.4.2 Caso 2: o ponto pertence à borda esquerda

Usando o ponto (x, y_j) pois o "deslocamento" é exclusivamente horizontal, temos:

I) $\frac{\partial f}{\partial y}(x+h_x, y_j) = \frac{\partial f}{\partial y}(x, y_j) + h_x \left(\frac{\partial^2 f}{\partial x \partial y}(x, y_j)(x - x_i) + \frac{\partial^2 f}{\partial y^2}(x, y_j)(y_j - y_j) \right) +$

$+\frac{h_x^2}{2} \left(\frac{\partial^3 f}{\partial x^2 \partial y}(x, y_j)(x - x_i)^2 + 2\frac{\partial^3 f}{\partial x \partial y^2}(x, y_j)(x - x_i)(y_j - y_j) + \frac{\partial^3 f}{\partial y^3}(x, y_j)(y_j - y_j)^2 \right) + O(h_x^3)$

$= \frac{\partial f}{\partial y}(x, y_j) + h_x \frac{\partial^2 f}{\partial x \partial y}(x, y_j)(x - x_i) + \frac{h_x^2}{2} \frac{\partial^3 f}{\partial x^2 \partial y}(x, y_j)(x - x_i)^2 + O(h_x^3)$

II) $\frac{\partial f}{\partial y}(x+2h_x, y_j) = \frac{\partial f}{\partial y}(x, y_j) + 2h_x \left(\frac{\partial^2 f}{\partial x \partial y}(x, y_j)(x - x_i) + \frac{\partial^2 f}{\partial y^2}(x, y_j)(y_j - y_j) \right) +$

$+2h_x^2 \left(\frac{\partial^3 f}{\partial x^2 \partial y}(x, y_j)(x - x_i)^2 + 2\frac{\partial^3 f}{\partial x \partial y^2}(x, y_j)(x - x_i)(y_j - y_j) + \frac{\partial^3 f}{\partial y^3}(x, y_j)(y_j - y_j)^2 \right) + O(h_x^3)$

$= \frac{\partial f}{\partial y}(x, y_j) + 2h_x \frac{\partial^2 f}{\partial x \partial y}(x, y_j)(x - x_i) + 2h_x^2 \frac{\partial^3 f}{\partial x^2 \partial y}(x, y_j)(x - x_i)^2 + O(h_x^3)$

e de **4I - II** segue que:

$4\frac{\partial f}{\partial y}(x+h_x, y_j) - \frac{\partial f}{\partial y}(x+2h_x, y_j) = 3\frac{\partial f}{\partial y}(x, y_j) + 2h_x \frac{\partial^2 f}{\partial x \partial y}(x, y_j)(x - x_i) + O(h_x^3)$

e portanto:

$\frac{\partial^2 f}{\partial x \partial y}(x, y_j)(x - x_i) = \frac{-3\frac{\partial f}{\partial y}(x, y_j) + 4\frac{\partial f}{\partial y}(x+h_x, y_j) - \frac{\partial f}{\partial y}(x+2h_x, y_j)}{2h_x} + O(h_x^2)$

4.4.3 Caso 3: o ponto pertence à borda direita

I) $\frac{\partial f}{\partial y}(x-h_x, y_j) = \frac{\partial f}{\partial y}(x, y_j) - h_x \left(\frac{\partial^2 f}{\partial x \partial y}(x, y_j)(x - x_i) + \frac{\partial^2 f}{\partial y^2}(x, y_j)(y_j - y_j) \right) +$

$+\frac{h_x^2}{2} \left(\frac{\partial^3 f}{\partial x^2 \partial y}(x, y_j)(x - x_i)^2 + 2\frac{\partial^3 f}{\partial x \partial y^2}(x, y_j)(x - x_i)(y_j - y_j) + \frac{\partial^3 f}{\partial y^3}(x, y_j)(y_j - y_j)^2 \right) + O(h_x^3)$

$= \frac{\partial f}{\partial y}(x, y_j) - h_x \frac{\partial^2 f}{\partial x \partial y}(x, y_j)(x - x_i) + \frac{h_x^2}{2} \frac{\partial^3 f}{\partial x^2 \partial y}(x, y_j)(x - x_i)^2 + O(h_x^3)$

II) $\frac{\partial f}{\partial y}(x-2h_x, y_j) = \frac{\partial f}{\partial y}(x, y_j) - 2h_x \left(\frac{\partial^2 f}{\partial x \partial y}(x, y_j)(x - x_i) + \frac{\partial^2 f}{\partial y^2}(x, y_j)(y_j - y_j) \right) +$

$+2h_x^2 \left(\frac{\partial^3 f}{\partial x^2 \partial y}(x, y_j)(x - x_i)^2 + 2\frac{\partial^3 f}{\partial x \partial y^2}(x, y_j)(x - x_i)(y_j - y_j) + \frac{\partial^3 f}{\partial y^3}(x, y_j)(y_j - y_j)^2 \right) + O(h_x^3)$

$= \frac{\partial f}{\partial y}(x, y_j) - 2h_x \frac{\partial^2 f}{\partial x \partial y}(x, y_j)(x - x_i) + 2h_x^2 \frac{\partial^3 f}{\partial x^2 \partial y}(x, y_j)(x - x_i)^2 + O(h_x^3)$

e de **4I - II** segue que:

$4\frac{\partial f}{\partial y}(x-h_x, y_j) - \frac{\partial f}{\partial y}(x-2h_x, y_j) = 3\frac{\partial f}{\partial y}(x, y_j) - 2h_x \frac{\partial^2 f}{\partial x \partial y}(x, y_j)(x - x_i) + O(h_x^3)$

e portanto:

$\frac{\partial^2 f}{\partial x \partial y}(x, y_j)(x - x_i) = \frac{3\frac{\partial f}{\partial y}(x, y_j) - 4\frac{\partial f}{\partial y}(x-h_x, y_j) + \frac{\partial f}{\partial y}(x-2h_x, y_j)}{2h_x} + O(h_x^2)$

4.5 Verificação do erro

Foi verificado na prática que, ao dobrar o valor de h_x ou h_y, o erro é quadruplicado. Apesar de não estar presente no código, o método feito para verificar foi usar os valores conhecidos das funções $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial y}$ e $\frac{\partial^2 f}{\partial x \partial y}$ e comparar com os aproximados, vendo por quanto errava. Depois, dobrar os valores de h_x e h_y e verificar que o erro era aproximadamente 4 vezes maior (o que faz sentido, dado que a aproximação tem grandeza quadrática).

5 Implementação: construindo e avaliando a v(x, y)

Para construir os polinômios interpoladores $v(x, y)$, foi usada uma matriz V do tamanho da malha, na qual cada posição (x, y) conhecida da malha tem seu polinômio interpolador $v(x, y)$ na mesma posição (x, y) de V. Tal matriz V contendo os polinômios $v(x, y)$ é construída usando a estrutura de dados *Struct Array* do *GNU-Octave*, de tal forma que os coeficientes são guardados de forma limpa e eficiente para cada polinômio. Assim, por exemplo, um valor de $f(x, y)$ que, na malha, pertença ao quadrado $Q = \overline{P_{ij}P_{(i+1)j}P_{(i+1)(j+1)}P_{i(j+1)}}$, formado pelos pontos $P_{ij}, P_{(i+1)j}, P_{(i+1)(j+1)}$ e $P_{i(j+1)}$, será interpolado pelo polinômio cujo os coeficientes estão guardados em V_{ij}.

A função `constroiv` recebe os parâmetros que definem a malha, a própria malha com os valores de **conhecidos** $f(x, y)$ e o modo (bicúbico ou bilinear) e internamente chama as funções `bilinear_method` ou `bicubic_method` que devolverão a matriz V contendo cada $v(x, y)$ com o número de coeficientes esperados (4 ou 16) para cada caso já computados. Em particular, se o modo escolhido for o bicúbico, antes das chamadas à função `bicubic_method` será feita uma chamada à função `aproxdf`, que computará as derivadas parciais discretas com as aproximações deduzidas na seção anterior, com erros possuindo uma ordem de grandeza $O(h_x^2)$ ou $O(h_y^2)$, a depender da derivada em questão. Estas derivadas são retornadas em matrizes do tamanho da malha, com seus valores computados para cada ponto conhecido, e são usadas de entrada para a função do modo bicúbico.

Já para realizar a avaliação de um determinado ponto (x, y) qualquer, simplesmente buscamos o quadrado $Q = \overline{P_{ij}P_{(i+1)j}P_{(i+1)(j+1)}P_{i(j+1)}}$, formado pelos *pixels* $P_{ij}, P_{(i+1)j}, P_{(i+1)(j+1)}$ e $P_{i(j+1)}$, para podermos recuperar os coeficientes do polinômio interpolador esperado. Encontramos tal quadrado Q usando os parâmetros $h_x = \frac{b_x - a_x}{n_x}$ e $h_y = \frac{b_y - a_y}{n_y}$ definidos no enunciado, acumulando h_x e h_y em duas variáveis (uma para cada eixo) até que estas encontrem a 'área' a qual o ponto (x, y) pertence na malha, recuperando também os índices que devem ser usadas na matriz V para obter o polinômio interpolador $v(x, y)$. Uma vez com o polinômio, basta realizar o cálculo normalmente considerando os valores de x e de y passados pelo usuário.

Observação: os valores de x e de y devem ser passados no programa de função, pois ali o usuário define a malha. No modo imagem, conforme dúvida tirada com o monitor, a imagem é a própria malha e seus parâmetros são descobertos durante a execução. É importante ressaltar aqui que as funções de ambos os programas possuem as mesmas implementações, com uma ou outra exceção tendo uma pequena variação (para tratamento de eventuais parâmetros distintos, como por exemplo o fato do modo imagem trabalhar com uma malha para cada canal de cor).

6 Compressão de decompressão de imagens

O programa de modo de manipulação de imagem tem a diferença de aplicar a teoria para compressão e decompressão de imagens. Nesta seção será abordado como isso foi realizado no EP3.

Nota: nesta seção, sempre que usarmos o termo $f(x, y)$, na verdade é para considerar que existem três funções: $f_R(x, y), f_G(x, y)$ e $f_B(x, y)$, uma para cada canal de cor.

6.1 Compressão

Quando o programa de modo de manipulação de imagem inicializado, a primeira coisa que ele faz é buscar a imagem no diretório '`./images`' que foi especificada na linha de comando para lê-la. Diferentemente do programa de função, que define diretamente a malha pelos parâmetros passados pela linha de comando, o programa de imagens irá antes fazer a compressão da imagem para então considerar a imagem comprimida como a função $f(x, y)$.

A ação de compressão (e conseqüentemente decompressão) está diretamente relacionado à opção do usuário, uma vez que o parâmetro '`--cr`' passado pela linha de comando irá definir o quanto da imagem será comprimida. Na versão que está sendo entregue, o valor mínimo para tal parâmetro é 5, conforme especificado na segunda seção deste documento. Ainda, está a cargo do usuário passar uma imagem de dimensões condizentes com o valor passado no parâmetro. A imagem que está sendo entregue de exemplo, '`./images/purple_tentacle.jpg`', foi testada extensivamente com a taxa de compressão $cr = 5$.

A maneira que a taxa de compressão é usada pelo programa para comprimir a imagem é bastante simples: para um valor de $cr = n$ e uma imagem de tamanho $W \times L$, o que será feito é remover todas as linhas i e colunas j que são divisíveis por n da imagem. Ou seja, construímos uma nova imagem de tamanho $w \times l$, onde $w = W - \lfloor \frac{W}{n} \rfloor$ e $l = L - \lfloor \frac{L}{n} \rfloor$. As únicas linhas e colunas que não removemos, porém, são as bordas, pois não há como saber que elas foram removidas durante a decompressão. Isso significa que a imagem comprimida pode ter também as dimensões $(W - \lfloor \frac{W}{n} \rfloor + 1) \times (L - \lfloor \frac{L}{n} \rfloor)$, $(W - \lfloor \frac{W}{n} \rfloor) \times (L - \lfloor \frac{L}{n} \rfloor + 1)$ ou ainda $(W - \lfloor \frac{W}{n} \rfloor + 1) \times (L - \lfloor \frac{L}{n} \rfloor + 1)$, a depender do valor da dimensão da imagem original e da tava de compressão cr.

Abaixo é fornecido um exemplo ilustrativo da compressão. Considere $W = L = 5$ e $cr = 2$ e a cada célula de índice (i, j). Note que a dimensão final será dada por $w = (5 - \lfloor \frac{5}{2} \rfloor) = 3$ e $l = (5 - \lfloor \frac{5}{2} \rfloor) = 3$.

(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)
(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)
(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)
(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)
(5, 1)	(5, 2)	(5, 3)	(5, 4)	(5, 5)

Imagem Original

(1, 1)	(1, 2)	(1, 3)	(1, 2)	(1, 3)
(2, 1)	(2, 2)	(2, 3)	(2, 2)	(2, 3)
(3, 1)	(3, 2)	(3, 3)	(3, 2)	(3, 3)
(2, 1)	(2, 2)	(2, 3)	(2, 2)	(2, 3)
(3, 1)	(3, 2)	(3, 3)	(3, 2)	(3, 3)

Imagem Original com as fileiras a serem removidas

(1, 1)	(1, 3)	(1, 5)
(3, 1)	(3, 3)	(3, 5)
(5, 1)	(5, 3)	(5, 5)

Imagem Comprimida $f(x, y)$

6.2 Decompressão

A decompressão é o processo que busca recuperar a imagem original a partir da imagem comprimida $f(x, y)$ usando o método de interpolação escolhido. isso significa que, quando o programa atinge este estágio, ele já calculou todas as derivadas parciais discretas da imagem e está preparado para tentar recuperar a imagem original.

Para recuperar a imagem original, a primeira coisa que o programa faz é ampliar a imagem, inserindo as fileiras de *pixels* nas posições que foram removidas, usando a taxa de compressão cr. Tal ação é realizada pela função '`enlarge`'. Para que isso funcione como esperado, a primeira tarefa que esta função faz é tentar descobrir o tamanho da imagem original, e a maneira que ela faz isso é pelas expressões:

- $W' = round(\frac{(w-1)\times cr}{cr-1})$
- $L' = round(\frac{(l-1)\times cr}{cr-1})$

É possível que existam perdas nesta transformação, a depender dos valores de cr, w e l. É

por isso que tomamos o cuidado de chamar as dimensoes recuperadas de W' e L'. Um exemplo onde há perda é para os valores $W = 12345$ e $cr = 7$, que resultará em $w = 12344$.

Em seguida, após ter acontecido a expansão da imagem comprimida, calculamos o valor dos *pixels* das fileiras adicionadas a partir da interpolação. Buscamos o polinômio interpolador em V apenas percorrendo os índices da imagem expandida, e sempre que chegamos em uma linha com índice p ou uma coluna com índice q divisível por cr, obtemos os índices da linha i e da coluna j de V a partir da seguinte expressão:

- $i = p - \lfloor \frac{p}{cr} \rfloor$
- $j = q - \lfloor \frac{q}{cr} \rfloor$

Assim, o polinômio interpolador de coeficientes guardados em V_{ij} será usado para atribuir o valor do novo *pixel* adicionado na decompressão. Além disso, para obter um resultado melhor, as informações da vizinhança do *pixel* também são usadas, de tal forma que usamos os polinômios interpoladores vizinhos para tentar conseguir um resultado mais próximo do esperado. Abaixo exemplos gráficos para ilustrar a decompressão.

(1, 1)	(1, 3)	(1, 5)
(3, 1)	(3, 3)	(3, 5)
(5, 1)	(5, 3)	(5, 5)

Imagem Comprimida $f(x, y)$. $V(x, y)$ tem a mesma dimensão

(1, 1)	(1, 2)	(1, 3)	(1, 2)	(1, 3)
(2, 1)	(2, 2)	(2, 3)	(2, 2)	(2, 3)
(3, 1)	(3, 2)	(3, 3)	(3, 2)	(3, 3)
(2, 1)	(2, 2)	(2, 3)	(2, 2)	(2, 3)
(3, 1)	(3, 2)	(3, 3)	(3, 2)	(3, 3)

Imagem Comprimida expandida. *Pixels* adicionados em azul

Agora, a depender da posição do *pixel* desonhecido, o seu novo valor será a **média aritmética do valor dos polinômios interpoladores dos seus vizinhos de cor cinza**, chutando para cada *pixel cinza* um valor (x, y) que pertença àquele quadrado e chamando a nossa função `avaliav`. Finalmente, devemos ter algo teoricamente similar ao representado na última ilustração, logo abaixo.

(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)
(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)
(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)
(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)
(5, 1)	(5, 2)	(5, 3)	(5, 4)	(5, 5)

Imagem Descomprimida

Assim, finalmente, entendendo como a decompressão funciona, é notável que se o *cr* eliminasse fileiras vizinhas, a informação seria totalmente perdida, pois na hora de reconstruir a imagem nós não teríamos *pixels* vizinhos conhecidos para retirar informação para interpolar. Ainda, recorde-se que como já foi mostrado anteriormente, há valores em que parte da informação é perdida na hora que fazemos a expansão para determinar as dimensões da imagem decomprimida.

Note que a maneira que foi implementada no EP3 não resolve os casos para todas as imagens. Para cada pixel, são avaliados apenas os seus vizinhos mais próximos e isso significa que se tivessemos uma imagem onde cada linha ou coluna tem duas cores intercaladas, a informação recuperada estaria errada. Para endereçar este problema, o programa também deveria considerar informações dos vizinhos dos vizinhos, ou seja, ao invés de uma área 2×2, uma área 4×4. Isto não foi implementado pois aumentaria a complexidade da função '`pixel_neighborhood_mean`', mas estou deixando registrado aqui no relatório esta observação feita a partir do estudo deste caso, que por si só já demonstra que o exercício programa cumpriu com o seu objetivo. Abaixo um exemplo ilustrativo de uma imagem que daria errado:

(1, 1)	(1, 2)	(1, 3)	(1, 2)	(1, 3)
(2, 1)	(2, 2)	(2, 3)	(2, 2)	(2, 3)
(3, 1)	(3, 2)	(3, 3)	(3, 2)	(3, 3)
(2, 1)	(2, 2)	(2, 3)	(2, 2)	(2, 3)
(3, 1)	(3, 2)	(3, 3)	(3, 2)	(3, 3)

Imagem de cores intercaladas

Perceba que a compressão com $cr = 2$ deixaria apenas os *pixels* de cor cinza e não recuperaríamos nenhum de cor verde. Logicamente que este problema pode ser extendido de tal forma que se intercalássemos as cores duas a duas, ou três a três até n a n (o exemplo acima é uma intercalação uma a uma), o número de "vizinhos de vizinhos"que precisaríamos visitar

aumentaria. Claro que aqui surge naturalmente uma questão: qual é o vizinho mais distante que podemos visitar? Pois ir muito longe também significa se distanciar do valor real do *pixel* a depender da imagem. A resposta do que é o melhor varia de imagem para imagem.

7 Testes e análises de resultados

Nesta seção serão discutidos os testes realizados e serão fornecidas análises dos seus respectivos resultados para explicar os valores das suas saídas. Todos os testes já estão implementados no arquivo '`bivariate_interpolation_test.m`', de tal forma que basta o usuário rodar este *script* para obter os resultados dos testes implementados.

7.1 Teste de verificação de interpolação

Este teste verifica se a interpolação funciona como deveria, isto é, se v interpola f nos pontos da malha. A maneira que ele foi implementado é bastante simples: todos os pontos (x, y) conhecidos da malha são lidos um a um e são calculados os seus valores em $v(x, y)$ e $f(x, y)$. O valor usado em $v(x, y)$, pensando no nosso quadrado $Q = \overline{P_{ij}P_{(i+1)j}P_{(i+1)(j+1)}P_{i(j+1)}}$, é o do vértice inferior esquerdo Pij. Finalmente, este ponto Pij é avaliado pela função '`avaliav`', que retornará $v(x, y)$ e em seguida será feito $|f(x, y) - v(x, y)|$.

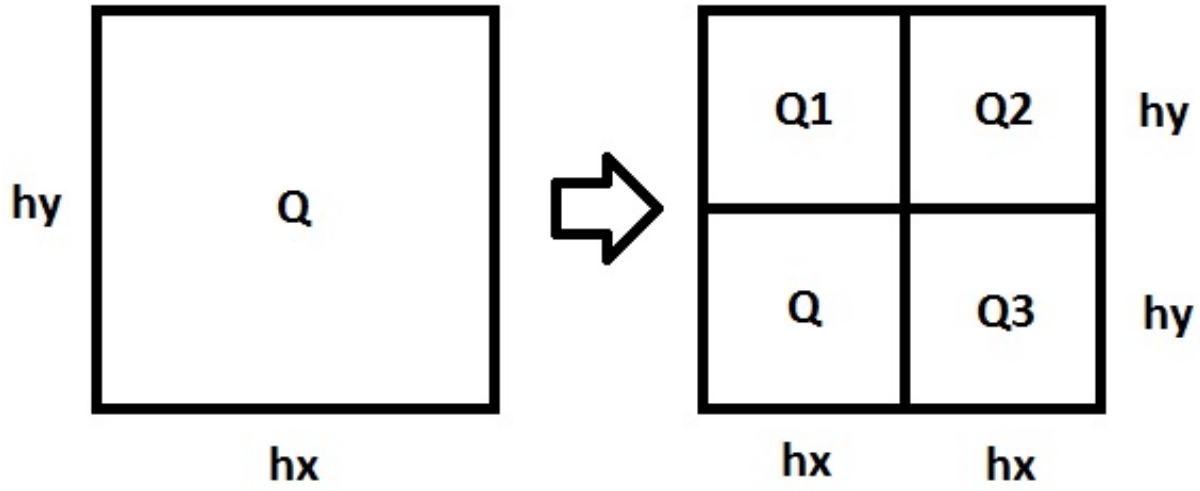
É esperado que $|f(x, y) - v(x, y)| = 0$, uma vez que considerando a forma de cada polinômio interpolador (descrita no enunciado e na seção 3 deste documento) e os pontos (x, y), o $v(x, y) = a_{00} = f(x, y)$. Nota que como os pontos (x, y) são conhecidos, todos os outros termos são anulados ou por $(x - x_i)$ com $x = x_i$ ou por $(y - y_j)$ com $y = y_j$, restando apenas o coeficiente $a_{00} = f(x, y)$.

7.2 Teste do comportamento do erro com relação a h_x e h_y

Este teste verifica se há uma relação entre o erro $|f(x, y) - v(x, y)|$ e os valores de h_x e h_y. É importante notificar o usuário que **este teste só será executado se** $h_x = h_y = h$, ou seja, se a malha tiver um formato quadrado (e não retangular). Escolha os parâmetros cuidadosamente se quiser rodar este teste. O exemplo de invocação fornecido na seção 2.2 respeita esta condição e pode ser reproduzido.

Pelas experiências realizadas com a implementação deste teste, há claramente uma relação entre ambos, de tal forma que ao duplicarmos os valores de n_x e n_y, dividimos pela metade os valores de h_x e h_y e verificamos que o valor do erro $|f(x, y) - v(x, y)|$ também diminui. Uma explicação para este fenômeno é que estamos tomando os limites $h_x \to 0$ e $h_y \to 0$ e, mais uma vez no nosso quadrado $Q = \overline{P_{ij}P_{(i+1)j}P_{(i+1)(j+1)}P_{i(j+1)}}$, isso equivale a diminuir pela metade o tamanho das suas arestas e inserir 3 novos quadrados Q_1, Q_2 e Q_3 no espaço até então ocupado por Q. Repetir este processo indefinidamente significa diminuir as dimensões do quadrado Q, aproximando seus vértices $P_{(i+1)j}, P_{(i+1)(j+1)}$ e $P_{i(j+1)}$ de P_{ij}. Isso significa que cada vez mais,

a área que o ponto (x, y) cair é mais próxima dos pontos conhecidos, e portanto o erro deve ser menor. Veja o exemplo ilustrativo abaixo:



Perceba que um valor de $f(x, y)$ que antes caia em Q (da esquerda) agora pode cair em Q_1, Q_2 e Q_3 e que a distância de (x, y) até o inferior esquerdo deste quadrado é menor do que a distância deste mesmo ponto vértice inferior esquerdo de Q, diminuindo portanto o valor do erro.

8 Considerações Finais (e pessoais)

Do EP2 para este, resolvi alguns problemas de falta de conhecimento que tinha com o *GNU-Octave* para oferecer um código mais limpo e eficiente. A qualidade do código com relação ao último EP está incomparavelmente superior, e desta vez eu estou entregando o EP satisfeito com os resultados obtidos e a qualidade do código. Ainda, este relatório traz informações a respeito de **todo trabalho que foi feito**, sem exceções, e responde todas as perguntas presentes no enunciado do EP3 ao longo das 7 seções.

Com relação à manipulação de imagens, o problema que eu tive com os índices no EP2 foi aparentemente resolvido neste EP, dado que eu consegui atingir resultados deveras satisfatórios para a interpolação da imagem exemplo. Este EP3 não foi feito copiando e colando o EP2 e 'melhorando as partes ruins'. Tudo foi replanejado e reimplementado, ora até usando outras estruturas de dados, tal como *Struct Arrays*.

Ainda, a maneira que eu aproximei as imagens no EP2 não usou nas bordas as expansões de Taylor para 3 pontos para frente, para trás, para cima e para baixo (apenas fazia a média aritmética entre o ponto da borda e o *pixel* vizinho conhecido do eixo correspondente), diferentemente do que foi feito neste EP3.

Finalmente, como o EP3 sugeriu uma idéia de continuidade do EP2 (e pensando em boas práticas de desenvolvimento de *software*), eu resolvi aproveitar a explicação da dedução dos cálculos dos coeficientes dos métodos de interpolação nessa documentação apenas porque me pareceu a coisa mais correta a se fazer, mesmo que isso não tivesse sido pedido no enunciado. Com isso a documentação fica, de fato, completa.