



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE DO CAMPUS ARARANGUÁ  
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

Leandro Seiti Kakimoto Silva

**Análise de Aproximações em Malhas com Refinamento Adaptativo**

Araranguá  
2022

Leandro Seiti Kakimoto Silva

## **Análise de Aproximações em Malhas com Refinamento Adaptativo**

Trabalho de Conclusão de Curso do Curso de Graduação em Engenharia de Computação do Centro de Ciências, Tecnologias e Saúde do Campus Araranguá da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Engenharia de Computação.

Orientadora: Profa. Priscila Cardoso Calegari, Dra.

Araranguá

2022

Ficha de identificação da obra elaborada pelo autor,  
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Silva, Leandro Seiti Kakimoto  
Análise de Aproximações em Malhas com Refinamento  
Adaptativo / Leandro Seiti Kakimoto Silva ; orientadora,  
Priscila Cardoso Calegari, 2022.  
29 p.

Trabalho de Conclusão de Curso (graduação) -  
Universidade Federal de Santa Catarina, Campus Araranguá,  
Graduação em Engenharia de Computação, Araranguá, 2022.

Inclui referências.

1. Engenharia de Computação. 2. Dinâmica de Fluidos. 3.  
Discretização. 4. Equações Diferenciais Parciais. 5. Mínimos  
Quadrados Móveis. I. Cardoso Calegari, Priscila. II.  
Universidade Federal de Santa Catarina. Graduação em  
Engenharia de Computação. III. Título.

Leandro Seiti Kakimoto Silva

## **Análise de Aproximações em Malhas com Refinamento Adaptativo**

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de “Bacharel em Engenharia de Computação” e aprovado em sua forma final pelo Curso de Graduação em Engenharia de Computação.

Araranguá, 14 de dezembro de 2022.

---

Profa. Analucia Schiaffino Morales, Dra.  
Coordenador do Curso

### **Banca Examinadora:**

---

Profa. Priscila Cardoso Calegari, Dra.  
Orientadora

---

Prof. Antônio Carlos Sobieranski, Dr.  
Avaliador  
Universidade Federal de Santa Catarina

---

Prof. Marcelo Zannin da Rosa, Dr.  
Avaliador  
Universidade Federal de Santa Catarina

# Análise de Aproximações em Malhas com Refinamento Adaptativo

## *Analysis of Approximations on Meshes with Adaptive Refinement*

Leandro Seiti Kakimoto Silva <sup>\*</sup>      Priscila Cardoso Calegari <sup>†</sup>

2022, Dezembro

### Resumo

A discretização de Equações Diferenciais Parciais e de um domínio computacional são elementos muito importantes na Dinâmica de Fluidos Computacional, que visa obter soluções numéricas para problemas de escoamento. Uma das técnicas comuns nesta área é o refinamento adaptativo de malhas, que, geralmente, exige a aproximação de dados nas interfaces entre níveis de refinamento. Uma aproximação local é proposta onde necessário sobre um estêncil definido por Diferenças Finitas, utilizando o Método de Mínimos Quadrados Móveis. Esta aproximação permite a utilização de malhas com configurações mais complexas, mas mantendo a ordem de aproximação obtida em malhas mais simples. É realizado um experimento comparativo entre métodos tradicionais de aproximação por Mínimos Quadrados e o Método de Mínimos Quadrados Móveis, e esta variação do método é julgada como superior com base na precisão da aproximação. A técnica é avaliada para a discretização da equação de Poisson bidimensional, um sistema linear é construído a partir do estêncil de Diferenças Finitas para cada elemento de uma malha adaptativa, utilizando as aproximações nos elementos necessários. O sistema linear obtido e o grau de dificuldade da sua resolução são estudados com base em seu número de condicionamento.

**Palavras-chaves:** Mínimos quadrados; Aproximações por mínimos quadrados móveis; Discretização; Malhas adaptativas

---

<sup>\*</sup>leandro.kakimoto@grad.ufsc.br

<sup>†</sup>priscila.calegari@ufsc.br

# Análise de Aproximações em Malhas com Refinamento Adaptativo

## *Analysis of Approximations on Meshes with Adaptive Refinement*

Leandro Seiti Kakimoto Silva <sup>\*</sup>      Priscila Cardoso Calegari <sup>†</sup>

2022, Dezembro

### Abstract

The discretization of both Partial Differential Equations and a computational domain are important elements in the Computation Fluid Dynamics research area, which seeks numerical solutions for flow problems. A common technique within this area is the adaptive mesh refinement. This paper proposes a local approximation over a stencil defined by the Finite Difference Method, for points where an approximation is required, using the Moving Least Squares Method. This approximation allows the usage of complex meshes, but preserving the approximation order obtained for simpler meshes. An experiment comparing traditional Least Squares and Moving Least Squares approximation methods is done, and the latter is judged as superior from the perspective of approximation accuracy. The technique is evaluated on the discretized Poisson's equation, a linear system is built from the Finite Differences stencil for each and every element of the adaptive mesh, using approximations on elements where it is necessary. This linear system and the difficulty of solving it are studied based on its condition number.

**Key-words:** Least squares; Moving least squares approximation; Discretization; Adaptive meshes

---

<sup>\*</sup>leandro.kakimoto@grad.ufsc.br

<sup>†</sup>priscila.calegari@ufsc.br

# 1 Introdução

Equações Diferenciais Parciais (EDPs) são um importante elemento de modelos matemáticos de fenômenos físicos, como por exemplo, as Leis de Conservação. Na área de Dinâmica de Fluidos Computacional, a solução numérica de EDPs é essencial para as simulações numéricas que envolvem escoamentos. Tais simulações frequentemente utilizam técnicas de discretização do domínio (geração de malhas, (CUNHA, 2010), (FORTUNA, 2000)), e discretização das equações (Método de Diferenças Finitas, Método do Volumes finitos, entre outros. Mais detalhes podem ser encontrados nas obras de (BRENNER; SCOTT, 2008) e (LEVEQUE, 2007)). Tanto a discretização do domínio quanto das equações regentes do modelo são necessárias para que seja possível solucionar o problema numericamente.

Uma técnica de discretização do domínio computacional é o refinamento adaptativo de malhas. Com critérios de refinamento previamente estabelecidos, é possível concentrar o poder de processamento em regiões de interesse. Assim obtemos menos pontos na discretização para a criação destas malhas. Não sendo necessário resolver um problema em uma malha irregular, ou em uma malha excessivamente discretizada, como apontado por (BABISHCHEVICH, 1989).

O refinamento adaptativo de malhas é versátil, uma vez que também podem ser utilizados em problemas dinâmicos, ou seja, que evoluem no tempo. A obra de Berger e Oliger (BERGER; OLIGER, 1984) descreve métodos de discretização do domínio e das equações em malhas adaptativas bloco estruturadas para EDPs do tipo hiperbólicas (caracterizadas pela variação no tempo e propagação da solução como uma onda, descritas em (LEVEQUE, 2007)). Além disso, Berger posteriormente prova a estabilidade de malhas adaptativas nas interfaces entre refinamentos para estas equações (BERGER, 1985).

A vantagem desta técnica é que os operadores diferenciais discretos mantêm a mesma estrutura independente do nível de refinamento. Porém são necessários operadores de interpolação para a comunicação dos dados entre os diferentes níveis de refinamento. Para isso, a geração das malhas com refinamento precisa atender algumas regras. Por exemplo, a regra do aninhamento, onde cada elemento de um nível de refinamento deve compartilhar arestas somente com elementos do mesmo nível, elementos do próximo nível de refinamento ou do nível de refinamento anterior. Isto é feito para evitar diferenças de refinamento muito abruptas na interface entre dois elementos, uma vez que isto pode afetar a solução (BERGER; OLIGER, 1984), (LEVEQUE, 2007). Uma técnica muito útil de geração é a de malhas compostas, que tal como argumentado por (BABISHCHEVICH, 1989) e (BERGER; OLIGER, 1984), permite utilizar o conceito de divisão e conquista para resolver problemas em domínios menores, e por vezes em malhas regulares, ao invés de grandes regiões irregulares.

O trabalho de (MIN; GIBOU, 2006) apresenta uma técnica de representação de malhas, nomeada *quadtrees* (para 2 dimensões) e *octrees* (para 3 dimensões), dado que o domínio é dividido em quatro elementos iguais, e um destes elementos é novamente dividido, seguindo esta abordagem recursivamente. A estrutura de dados permite armazenar esta configuração como uma árvore, tal que o vértice raiz da árvore define o domínio completo. Os critérios de divisão do domínio desta maneira podem incluir, por exemplo, uma variação de propriedades físicas do problema dentro do domínio (como a temperatura), ou quando o erro de aproximação está acima de uma tolerância pré definida.

A fim de obter uma malha que possa ser definida com menos restrições, este

trabalho visa analisar um método que permita aproximar os valores de uma função em elementos de diferentes níveis dentro de uma malha, caso isso não seja possível, cada elemento que possui vizinhas com diferentes tamanhos teria de ser tratada de maneira independente.

Um método popular que realiza aproximações a partir de uma nuvem de pontos, além de obter o comportamento geral de uma solução numérica em uma determinada região, é o chamado método de mínimos quadrados (MMQ). Tal como é descrito nas obras de (ZHANG; KWON; YOUN, 2005) e (SOUSA *et al.*, 2019), o método de mínimos quadrados pode ser utilizado para aproximar valores em pontos que não pertencem a algum elemento da malha dentro do domínio conhecido.

Os trabalhos citados no parágrafo anterior utilizam uma variação do MMQ, conhecido como método de mínimos quadrados móveis (MLS, do inglês *Moving Least Squares*). Esta técnica permite que a aproximação de um elemento que não existe na malha seja definido como uma combinação linear dos elementos na vizinhança, ou a uma distância arbitrária, permitindo que a aproximação seja fiel ainda que em uma configuração complexa.

Tanto a obra de (ZHANG; KWON; YOUN, 2005) quanto a de (SOUSA *et al.*, 2019) utilizam métodos *meshless*, isto é, uma técnica de simulação computacional que não envolve malhas, e que é frequentemente utilizada na presença de muitas descontinuidades no domínio, estas técnicas utilizam interpolações envolvendo uma nuvem de pontos a uma distância do ponto de aproximação. Demonstrando uma aplicação prática que utiliza o MLS, (LE; KHOO, 2017) utilizam a técnica na reconstrução dos valores para a velocidade nos pontos da vizinhança da camada limite (FREIRE, 2019), evitando cálculos que envolvem a velocidade nos limites deformáveis do domínio.

É desejado que caso exista uma malha adaptativa, e uma operação seja feita na interface de duas células de níveis diferentes, o MLS possa ser utilizado para aproximar o valor na célula que não está presente. Desta maneira, a operação pode ser realizada como se a malha fosse uniforme dentro daquela região.

Assim, o objetivo deste trabalho é estudar as aproximações realizadas sobre um domínio discreto, de maneira a preservar as características do estêncil independente da configuração da malha em que este será construído, e removendo algumas restrições dos métodos de geração de malhas adaptativas no contexto de análise e resolução de EDPs. Para atingir este objetivo, o MLS é utilizado para aproximar os valores em pontos que não pertencem a malha. Alguns dos trabalhos presentes na literatura (como (ZHANG; KWON; YOUN, 2005) e (SOUSA *et al.*, 2019)) também utilizam do MLS para realizar aproximações sobre domínios discretos, porém estas são estudadas para métodos *meshless*, ou seja, as aproximações são realizadas independente da maneira que o domínio foi discretizado. Dito isso, a utilização de aproximações com MLS sobre uma malha adaptativa ainda é pouco explorada, e o presente trabalho propõe o estudo desse tipo de aproximação em malhas adaptativas estruturadas.

A Seção 2 descreve as técnicas de discretização de um domínio computacional, e de equações diferenciais através de Diferenças Finitas. Após essa definições, o conceito de malhas adaptativas é explicado. Na Seção 3 é descrita a fundamentação teórica do MMQ para duas dimensões, utilizada para desenvolver o MLS. Uma aplicação prática para a técnica proposta é realizada na Seção 4. Por fim, são apresentados os resultados na Seção 5, que são discutidos na Seção 6 juntamente com as propostas para trabalhos futuros.



## 2 Definições da malha e operadores diferenciais discretos

### 2.1 Discretização do domínio

A motivação principal ao realizar uma discretização é, na maioria das vezes, aproximar a solução de uma Equação Diferencial Parcial, conforme (LEVEQUE, 2007), (FORTUNA, 2000), (CUNHA, 2010). A solução da equação é uma função contínua que satisfaz relações com suas derivadas, e na maioria das vezes não é simples encontrar uma solução analítica para este tipo de problema.

Para um método computacional, é inviável obter soluções sobre uma região contínua. O domínio, onde a equação é definida, é dividido em elementos (também comumente chamados de células), e estes elementos podem ser representados por diversas formas geométricas (como por exemplo, retângulos ou triângulos, dependendo da geometria do problema e suas características físicas (FORTUNA, 2000)), que são utilizadas na obtenção da solução aproximada, por meio do Método de Diferenças Finitas, (mais detalhes na Seção 2.1.1) além do Método de Volumes Finitos, entre outros.

Para o caso bidimensional, cada um dos pontos onde a função é avaliada corresponde a um par ordenado  $(x, y)$ . Uma malha é então definida como o conjunto formado pela união das células que compõem o (CUNHA, 2010). Uma malha é dita uniforme quando todos os pares  $(x, y)$  estão igualmente espaçados entre si em todas as direções. Uma malha é estruturada quando os elementos apresentam uma distribuição regular de seus pontos sobre o espaço, como explicado por (FORTUNA, 2000).

Dentre as principais formas de discretização utilizadas na literatura, podemos destacar duas. A primeira caracterizada pela posição dos pontos pertencentes a malha nos vértices das células, ou seja, diretamente sobre as divisões do domínio, (presente nos trabalhos de (LE; KHOO, 2017) e (MIN; GIBOU, 2006)). Na segunda, as variáveis pertencentes a malha se situam no centro da célula da (SOUSA *et al.*, 2019), (BERGER; OLIGER, 1984) e (ALVAREZ, 2013). Neste trabalho utiliza-se a segunda abordagem para a implementação das aproximações na malha. A Figura 1 é uma representação das duas abordagens em um domínio bidimensional.

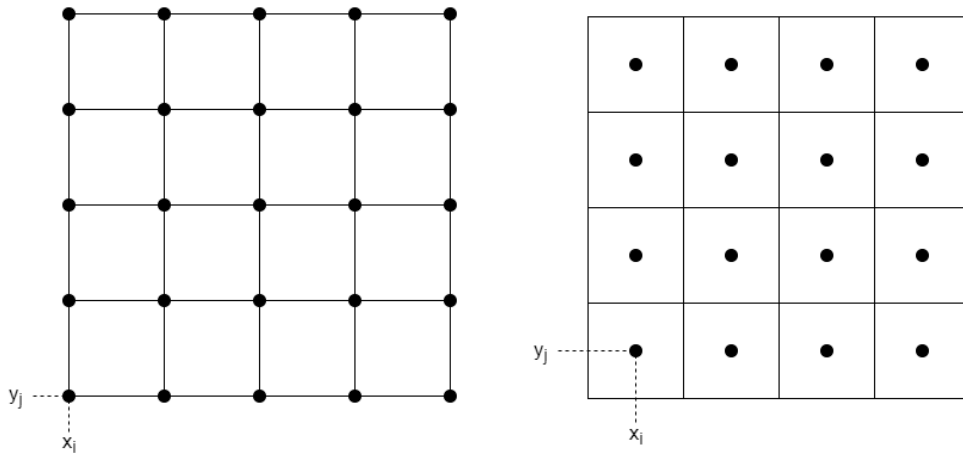


Figura 1 – Discretização com variáveis sobre os vértices das células (esquerda) e nos centros das células (direita). Fonte: Autor, 2022.

### 2.1.1 Método das Diferenças Finitas

O método de Diferenças Finitas é um método baseado na Série de Taylor, para a aproximação de derivadas de uma função (LEVEQUE, 2007). A definição para a aproximação da primeira derivada de uma função  $u(x)$  é dada por

$$D_+u(\bar{x}) = \frac{u(\bar{x} + h) - u(\bar{x})}{h}, \quad (1)$$

sendo  $h = \frac{(b-a)}{N}$ , com  $N$  igual ao número de subintervalos da discretização, e  $a$  e  $b$  como os limites inferior e superior do domínio, respectivamente, veja Figura 2. É interessante notar que, quando se toma o limite de  $h \rightarrow 0$ , a equação (1) se torna a definição de derivada de uma função contínua de uma variável.

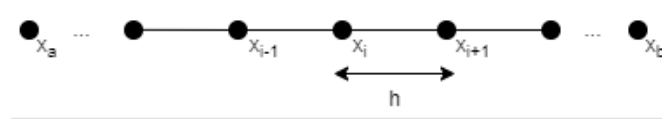


Figura 2 – Estêncil da aproximação de uma derivada de uma função unidimensional. Fonte: Autor, 2022.

A equação (1) é chamada de Diferenças Finitas Progressivas, uma vez que ela aproxima  $u'(x)$  somente pela direita. De maneira análoga é possível definir uma aproximação à esquerda, conhecida como Diferenças Finitas Regressivas, e dada por

$$D_-u(\bar{x}) = \frac{u(\bar{x}) - u(\bar{x} - h)}{h}. \quad (2)$$

As equações (1) e (2) são aproximações de primeira ordem, ou seja, o erro cometido ao realizar a aproximação é proporcional a  $h$ . Uma aproximação central é dada com base nas definições das aproximações laterais, e é dado por

$$D_0u(\bar{x}) = \frac{u(\bar{x} + h) - u(\bar{x} - h)}{2h} = \frac{1}{2}(D_+u(\bar{x}) + D_-u(\bar{x})). \quad (3)$$

A equação (3) é uma aproximação de segunda ordem, ou seja, o erro é proporcional a  $h^2$ .

Seguindo esta mesma metodologia, é possível definir uma aproximação para a derivada de segunda ordem  $u''(x)$ . A aproximação das diferenças finitas centrais para a derivada de segunda ordem é dada por

$$D^2u(\bar{x}) = \frac{1}{h^2}[u(\bar{x} - h) - 2u(\bar{x}) + u(\bar{x} + h)]. \quad (4)$$

Esta aproximação é obtida com base na soma das séries de Taylor de  $u(\bar{x} + h)$  e  $u(\bar{x} - h)$ , o que pode ser visualizado na equação (5).

$$\begin{aligned} u(\bar{x} + h) &= u(\bar{x}) + hu'(\bar{x}) + \frac{h^2}{2}u''(\bar{x}) + \frac{h^3}{6}u^{(3)}(\bar{x}) + \frac{h^4}{24}u^{(4)}(\bar{x}) + \dots, \\ u(\bar{x} - h) &= u(\bar{x}) - hu'(\bar{x}) + \frac{h^2}{2}u''(\bar{x}) - \frac{h^3}{6}u^{(3)}(\bar{x}) + \frac{h^4}{24}u^{(4)}(\bar{x}) - \dots, \end{aligned} \quad (5)$$

Somando as expressões na equação (5), obtém-se

$$u(\bar{x} + h) + u(\bar{x} - h) = 2u(\bar{x}) + h^2 u''(\bar{x}) + \frac{h^4}{12} u^{(4)}(\bar{x}) + \dots, \quad (6)$$

sendo

$$u''(\bar{x}) = D^2 u(\bar{x}) = \frac{1}{h^2} [u(\bar{x} - h) - 2u(\bar{x}) + u(\bar{x} + h)] + \frac{O(h^4)}{h^2} \quad (7)$$

onde a ordem do erro cometido é  $\frac{O(h^4)}{h^2}$  ou simplesmente  $O(h^2)$ , ou seja, também é proporcional a  $h^2$ .

Para uma função  $u(x, y)$ , com duas variáveis, são definidas as aproximações para as derivadas em relação a  $x$  e  $y$ . Para simplificar a notação, o ponto  $\bar{x} - h_x$  será denotado como  $i - 1$ , o ponto  $\bar{y} - h_y$  será denotado como  $j - 1$ , e assim por diante. As distâncias entre os pontos da malha, em cada uma das direções, são  $h_x$  e  $h_y$ , respectivamente. O estêncil da aproximação das segundas derivadas na posição  $i, j$  é apresentado na Figura 3.

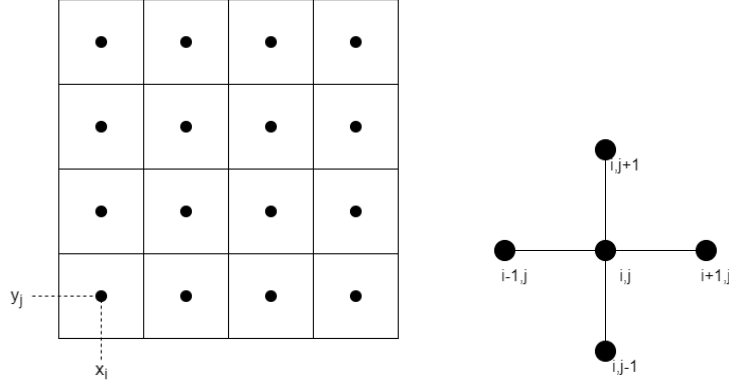


Figura 3 – Domínio discreto (esquerda) e representação do estêncil bidimensional de diferenças finitas (direita). Fonte: Autor, 2022.

Deste modo, as aproximações para as segundas derivadas, em cada direção, são

$$\begin{aligned} D_x^2 u(x_i, y_j) &= \frac{1}{h_x^2} [u_{i-1,j} - 2u_{i,j} + u_{i+1,j}], \\ D_y^2 u(x_i, y_j) &= \frac{1}{h_y^2} [u_{i,j-1} - 2u_{i,j} + u_{i,j+1}]. \end{aligned} \quad (8)$$

### 2.1.2 Malhas Adaptativas

Segundo (BABISHCHEVICH, 1989), o crescimento de problemas que necessitam de malhas adaptativas é notável em inúmeros estudos que tratam de problemas físicos ou matemáticos, como por exemplo, problemas com domínios complexos e irregulares ou problemas com soluções não suaves.

A aproximação de EDPs pode ser feita utilizando métodos com baixo custo computacional em grande parte de seu domínio. Por exemplo, definindo uma malha uniforme

sobre o domínio espacial e utilizando o método de Diferenças Finitas (seção 2.1.1). Porém, é comum encontrar nos domínios onde as EDPs são definidas, pontos de inflexão, descontinuidades ou variações abruptas nos valores das funções.

LeVeque, em (LEVEQUE, 2007) mostra o impacto na escolha do nível de refinamento do domínio. Caso seja desejada uma solução com alta precisão em uma malha uniforme, é necessário utilizar um grande número de pontos na malha. Caso uma região seja descontínua entre dois elementos de posição  $i$  e  $i + 1$ , por exemplo, a aproximação de uma derivada sobre estes elementos se torna difícil de ser obtida, se a malha empregada não possuir um nível suficiente de refinamento.

Porém, existem casos em que essa metodologia não é ideal, como por exemplo, se uma equação possui uma solução que se comporta de maneira suave em uma região e varia abruptamente em outra região do domínio analisado. Nesse caso as aproximações realizadas sobre esta subregião em que a solução é suave utilizam um recurso computacional excessivo, ainda que seja necessário empregar este processamento sobre a região onde a solução possui um comportamento complexo.

Uma solução interessante para este problema é a possibilidade de adicionar malhas com células menores nestas regiões de interesse. Esta abordagem é conhecida como refinamento adaptativo de malhas, que foi proposta por (BERGER; OLIGER, 1984).

Na obra de (BERGER; OLIGER, 1984), que propõe a utilização do refinamento adaptativo, as malhas são geradas de maneira que, quando a estimativa de erro da solução aumenta além de um limiar desejado, ou ocorre a aparição das fronteiras do domínio, uma malha refinada é empregada diretamente sobre estas regiões. Ainda é possível refinar mais os elementos desta malha, a geração recursiva, como é chamada, permite a criação de malhas dentro de outras malhas, ajustando a discretização do domínio da melhor maneira possível.

A Figura 4 apresenta um exemplo de domínio discretizado que se assemelha ao lado direito da Figura 1. Neste caso, o domínio foi refinado em uma região. Isto fornece mais precisão ao aproximar um valor que está contido em uma das células menores quando comparado a uma célula maior. De maneira análoga, é possível visualizar cada uma das células maiores como um refinamento de uma célula ainda maior, e assim por diante.

É importante notar que as aproximações aqui realizadas são feitas para um domínio bidimensional, mas obras como (STARIUS, 1980), (BERGER; OLIGER, 1984) e (BISHCHEVICH, 1989) mostram que a técnica pode ser generalizada para algum espaço de ordem mais alta. Utilizando o chamado método de malhas compostas, é possível dividir o domínio em subdomínios e fazer com que cada subdomínio possua sua própria malha.

É possível definir regras para a geração de uma malha para reduzir o impacto das diferenças de nível na utilização das diferenças finitas. Uma regra comum, que facilita uso da interpolação, é a chamada regra do aninhamento, já citada neste trabalho.

Para um domínio discretizado de maneira uniforme, o menor valor que  $h$  pode assumir em um método de diferenças finitas é também tamanho de uma célula da malha, porém quando existe um refinamento adaptativo, este tamanho varia juntamente com o ponto de referência na aproximação. Se o ponto a ser utilizado na aproximação da derivada estiver em uma célula que possui uma vizinha com refinamento, existe um problema fundamental no cálculo da aproximação, bem como na garantia da ordem do erro.

Assim, caso uma célula possua vizinhas em diferentes níveis de refinamento, é

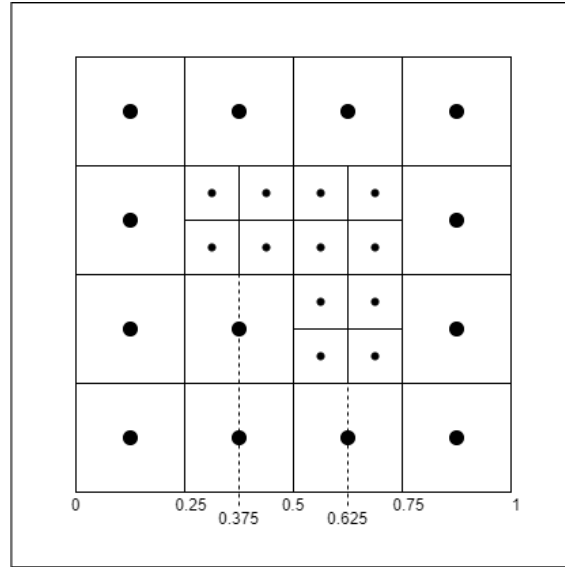


Figura 4 – Exemplo de discretização com malha refinada e variáveis no centro da célula.  
Fonte: Autor, 2022.

necessária uma aproximação local para que o estêncil não seja alterado, veja Figura 5. Para essas aproximações será utilizado o método de mínimos quadrados.

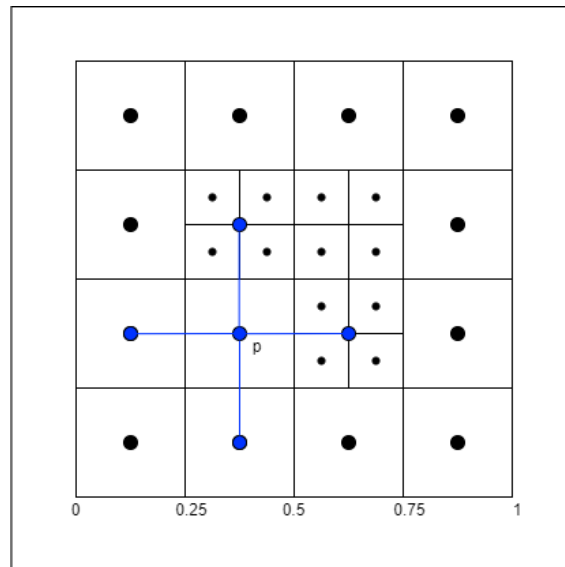


Figura 5 – Estêncil de discretização para o ponto  $p$  de uma malha refinada. Fonte: Autor, 2022.

### 3 Método dos Mínimos Quadrados

O método de mínimos quadrados (MMQ) é famoso na Estatística e na Física, sendo amplamente utilizado em técnicas de regressão linear. Em resumo, dada uma nuvem de pontos obtidos empiricamente é possível encontrar uma função que se ajusta aos pontos, de

forma que o erro seja mínimo. Dessa forma, os valores nos pontos que não foram obtidos na amostra do experimento podem ser estimados.

### 3.1 Mínimos Quadrados em duas dimensões

Em uma abordagem semelhante a utilizada por (BABISHCHEVICH, 1989), o trabalho de (NEALEN, 2004) mostra a expansão do MMQ, normalmente utilizado para aproximar funções unidimensionais, para mais de uma dimensão. A técnica utiliza uma base que corresponde a dimensão da aproximação, além do fato de que a função objetivo da aproximação também é bidimensional, tal que para o caso de uma função do  $\mathbb{R}^2$  é possível utilizar uma base polinomial como  $[1, x, y]$  ou  $[1, x, y, xy, x^2, y^2]$ .

Ainda que seja possível utilizar uma base polinomial de ordens mais altas, o erro de aproximação por Diferenças Finitas, que é onde as aproximações por MMQ serão utilizadas, será proporcional  $h^2$ , logo, bases de ordens muito altas aumentariam tempo necessário para calcular a aproximação, sem impactar de maneira drástica o erro da aproximação.

Dados  $m$  pontos  $(x_i, y_j) \in \mathbb{R}^2$  e uma função  $f(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}$  avaliada nestes pontos, o objetivo é obter uma função  $g(x, y)$  que melhor aproxima  $f$  nos  $m$  pontos. Esta função  $g$  é representada pelo produto escalar da base com um vetor de coeficientes  $c$ , tal que

$$g(x, y) = b(x, y)^T c = b(x, y) \cdot c, \quad (9)$$

onde  $b(x, y) = [b_1(x, y), \dots, b_n(x, y)]$  é uma base polinomial pertencente a dimensão  $n$  em questão e  $c = [c_1, \dots, c_n]$  é o vetor de coeficientes desconhecidos a ser aproximado. Em outras palavras, estamos aproximando uma função qualquer  $f(x_i, y_j)$  por um polinômio que pertence a dimensão em que será realizada o método. Alguns exemplos de base polinomial do  $\mathbb{R}^2$  são  $b(x, y) = [1, x, y]$  e  $b(x, y) = [1, x, y, xy, x^2, y^2]$ .

No método dos Mínimos Quadrados deseja-se realizar a minimização do resíduo dado por

$$J = \min \sum_{i,j}^m [g(x_i, y_j) - f_{ij}]^2. \quad (10)$$

Para minimizar a função quadrática em questão, realiza-se o cálculo das derivadas parciais para cada coeficiente da base e iguala-se cada resultado a zero, cada uma destas equações em conjunto define um sistema linear que, ao ser resolvido, resulta no vetor de coeficientes que multiplica a base e define a aproximação final. Uma destas derivadas para o  $n$ -ésimo termo da base  $b$  é apresentada na equação (11)

$$\frac{\partial J}{\partial c_n} = 2 \sum_{i,j}^m b_n(x_i, y_j) [b(x_i, y_i)^T c - f_{ij}] = 0. \quad (11)$$

Utilizando a notação matricial, o vetor de coeficientes  $c$  pode ser definido utilizando o conceito de produto de uma matriz por sua inversa sendo igual a matriz identidade, logo, dado o sistema linear da equação (12)

$$\sum_{i,j}^m [b(x_i, y_j) b(x_i, y_j)^T c] = \sum_{i,j}^m b(x_i, y_j) f_{ij}, \quad (12)$$

o problema pode ser resolvido para o vetor de coeficientes  $c$ , tal que:

$$c = \sum_{i,j}^m [b(x_i, y_j)b(x_i, y_j)^T]^{-1} \sum_{i,j}^m b(x_i, y_j) f_{ij}, \quad (13)$$

onde  $A = \sum_{i,j}^m [b(x_i, y_j)b(x_i, y_j)^T]$  e  $B = [b(x_1, y_1), b(x_2, y_2), \dots, b(x_m, y_m)]$ , tal que cada  $b(x_i, y_j)$  representa uma linha da matriz B avaliada no ponto  $(x_i, y_j)$ , logo, por fim:

$$c = A^{-1}Bf. \quad (14)$$

Assim como demonstrado por (NEALEN, 2004), a forma matricial deste sistema linear quadrado (uma vez que  $[b(x_i, y_j)b(x_i, y_j)^T]$  garante uma matriz quadrada), e para a base polinomial  $(1, x, y, x^2, xy, y^2)$  é:

$$\sum_{i,j} \begin{bmatrix} 1 & x_i & y_j & x_i^2 & x_i y_j & y_j^2 \\ x_i & x_i^2 & x_i y_j & x_i^3 & x_i^2 y_j & x_i y_j^2 \\ y_j & x_i y_j & y_j^2 & x_i^2 y_j & x_i y_j^2 & y_j^3 \\ x_i^2 & x_i^3 & x_i^2 y_j & x_i^4 & x_i^3 y_j & x_i^2 y_j^2 \\ x_i y_j & x_i^2 y_j & x_i y_j^2 & x_i^3 y_j & x_i^2 y_j^2 & x_i y_j^3 \\ y_j^2 & x_i y_j^2 & y_j^3 & x_i^2 y_j^2 & x_i y_j^3 & y_j^4 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{bmatrix} = \sum_{i,j} \begin{bmatrix} 1 \\ x_i \\ y_j \\ x_i^2 \\ x_i y_j \\ y_j^2 \end{bmatrix} f_{i,j}.$$

Além do MMQ tradicional, existem outras técnicas de aproximação baseadas em mínimos quadrados, como por exemplo a obra desenvolvida por (THOMPSON; PINSKY, 1995). O trabalho trata de uma variação do MMQ bidimensional, utilizando uma formulação conhecida como enunciação de Galerkin (explicada por (BELYTSCHKO *et al.*, 1996)), e empregando os chamados mínimos quadrados de Galerkin (GLS, do inglês *Galerkin Least Squares*) para ajustar o desvio de fase de uma solução numérica para uma EDP.

Outra variante do MMQ tradicional é o método chamado de Mínimos Quadrados Móveis (MLS), que realiza uma aproximação de mínimos quadrados, ponderada por uma função peso e que é computada avaliando pontos da região da malha que estão geometricamente próximos ao ponto que se deseja aproximar, esta variante é utilizada na obra de (SOUSA *et al.*, 2019), por exemplo.

### 3.2 Método dos Mínimos Quadrados Móveis

O MLS também é um método de reconstrução de uma função contínua a partir de um conjunto ou uma nuvem de pontos arbitrários, a maior diferença em relação ao MMQ padrão é que a minimização de quadrados é direcionada, ou seja, o ponto a ser aproximado é movido por toda a região do domínio discretizada, como descrito por (NEALEN, 2004) e (SOUSA *et al.*, 2019).

Isso indica, em essência, que a aproximação neste caso é válida para a região da vizinhança, além do ponto arbitrário escolhido para a aproximação. Frequentemente o MLS é utilizado para a reconstrução de superfícies complexas (LIU *et al.*, 2021) e (MOHAMMADI; SOTIROPOULOS; BRINKERHOFF, 2018).

Para entender o MLS primeiro é necessário empregar o conceito do MMQ ponderado, que consiste em um método de mínimos quadrados onde para cada um dos termos  $x_i, y_j$  a serem avaliados, a função resíduo quadrática a ser minimizada é também multiplicada por

um peso  $\lambda(||(x - x_i, y - y_j)||)$ , que frequentemente é baseado na distância entre o ponto que está sendo aproximado e os valores dos pontos nas células vizinhas da célula em que o ponto está.

O método foi proposto inicialmente por (LANCASTER; SALKASKAS, 1981) que ainda especificam o conceito, dividindo entre MLS interpolante e não interpolante. O MLS não interpolante é uma aproximação feita por MLS baseada na suposição de que, dado um domínio  $\Omega = \{(x, y) | a \leq x, y \leq b\} \subset \mathbb{R}^2$ , a função  $\phi : \Omega \rightarrow \mathbb{R}$  e os valores de  $\phi$  são conhecidos em  $\Omega$ , assim é possível definir um aproximante global  $G\phi$  para  $\phi$  a partir de uma aproximação local para um ponto único ponto  $\hat{z}$ .

O MLS interpolante se baseia no conceito de tornar o peso  $\lambda(||(x - x_i, y - y_j)||)$  uma singularidade no ponto  $\hat{z}$  caso a aproximação global esteja interpolando naquele ponto, assim, no ponto  $\hat{z}$  o valor da aproximação local e da função devem ter erro zero, ou seja, devem ser exatamente iguais (WENDLAND, 2001).

### 3.2.1 O esquema de aproximação por mínimos quadrados móveis

A enunciação do MLS não interpolante, que é a técnica empregada neste trabalho é deduzida através dos métodos apresentados por (ZHANG; KWON; YOUN, 2005) e (ZHANG; GUO *et al.*, 2015), em que o aproximante global (para a região de interesse) será escrito como uma combinação do produto entre pesos  $w_{ij}(x, y)$  (para cada elemento  $x_i, y_j$  da vizinhança de um ponto  $\hat{z}$ ), e o valor exato conhecido da função  $\phi(x_i, y_j)$  nestes mesmos elementos.

É possível também escrever a aproximação global como uma combinação linear de coeficientes  $c_i$  com os elementos da base utilizada na aproximação, como na obra de (BALE *et al.*, 2021), e como demonstrado anteriormente para o MMQ simples (Seção 3.1), entretanto a forma descrita no parágrafo anterior permite maior flexibilidade para a utilização prática neste caso (seção 4).

Assim, a aproximação por MLS  $\phi^h(x, y)$  de uma função  $\phi$  contínua no domínio  $\Omega \subset \mathbb{R}^2$  é definida de maneira semelhante a do MMQ padrão, com a adição do peso  $\lambda(x, y)$  nos produtos que geram as matrizes A e B, e que pode ser escrita como:

$$\phi^h(x, y) = \sum_k^n b(x_i, y_j) c_k(x, y) \equiv b(x, y)^T c(x, y), \quad (15)$$

assim como no MMQ padrão  $b(x, y) = [b_1(x, y), \dots, b_n(x, y)]$  é uma base polinomial de  $n$  elementos e  $c = [c_1, \dots, c_n]$  é o vetor de coeficientes.

Para obter os coeficientes  $c$ , o erro quadrático deve ser minimizado, mas neste caso, a soma quadrática está ponderada, o que resulta em:

$$J = \min \sum_{i,j}^m \lambda_{ij}(x, y) [b(x_i, y_j)^T c(x, y) - \phi_{ij}]^2, \quad (16)$$

os elementos  $\lambda_{ij}$  são os pesos associados a cada um dos elementos do domínio que serão utilizados na aproximação,  $m$  é o número de pontos na vizinhança do ponto  $\hat{z}$ .

Com relação aos pesos  $\lambda$ , que são utilizados para a montagem das matrizes de aproximação, as funções mais comuns que se baseiam na distância entre o ponto  $z = (\hat{x}_p, \hat{y}_p)$



arbitrário e os pontos  $(x_i, y_j)$  na aproximação são a função que utiliza o inverso da distância  $r = \sqrt{(x_p - x_i)^2 + (y_p - y_j)^2}$ :

$$\lambda_{ij}(x, y) = \frac{1}{r^2 + \epsilon^2}, \quad (17)$$

sendo  $\epsilon$  é um parâmetro para evitar a singularidade no caso de  $x_p = x_i$  e  $y_p = y_j$ , tal que  $\epsilon > 0$ , ou ainda a função de peso com *splines* (CUNHA, 2010) cúbicas, ou de ordem 3

$$\lambda_{ij}(x, y) = \begin{cases} \frac{2}{3} - 4r^2 + 4r^3 & \text{se } r \leq \frac{1}{2} \\ \frac{4}{3} - 4r + 4r^2 - \frac{4}{3}r^3 & \text{se } \frac{1}{2} < r \leq 1 \\ 0 & \text{se } r > 1. \end{cases} \quad (18)$$

Esta é uma função de suporte compacto, isso significa que a função peso só possui valor diferente de zero para uma vizinhança próxima ao ponto de aproximação  $\hat{z}$ , esta função é explicada e utilizada nos trabalhos de (LE; KHOO, 2017) e (MOHAMMADI; SOTIROPOULOS; BRINKERHOFF, 2018).

Continuando com a enunciação do MLS, no presente trabalho, a vizinhança de uma célula é definida por todas as células que compartilham vértices ou arestas com a célula em questão. A Figura 6 retrata esta relação, com a vizinhança do ponto  $\hat{z}$  na região tracejada. Para fins práticos de implementação, o próprio ponto  $\hat{z}$  é incluído no conjunto da vizinhança. É importante notar que, na Figura 6, o ponto a ser aproximado é o ponto  $p$ , que está contido na mesma célula do ponto  $\hat{z}$ .

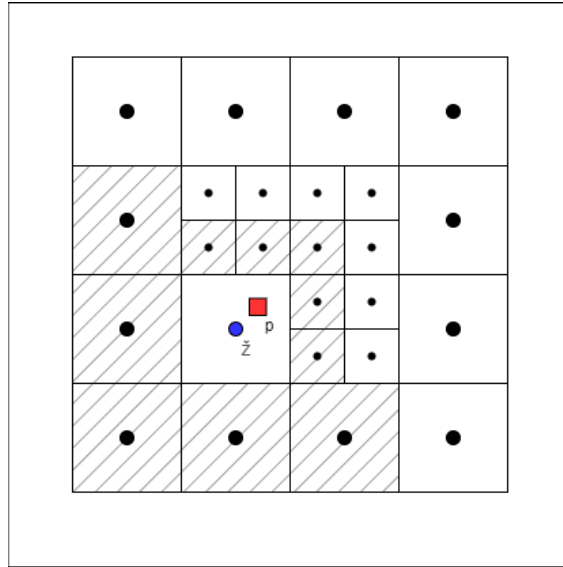


Figura 6 – Definição de vizinhança para um ponto arbitrário  $\hat{z}$ . Fonte: Autor, 2022.

Os coeficientes  $c(x, y)$  podem ser expressados por:

$$c(x, y) = A^{-1}(x, y)B(x, y)\hat{\phi}, \quad (19)$$

onde

$$\begin{aligned}
A(x, y) &= \sum_{i,j}^m \lambda_{ij}(x, y) b(x_i, y_j) b(x_i, y_j)^T \\
B &= [\lambda_1(x, y) b_1(x_1, y_1), \lambda_2(x, y) b_2(x_2, y_2), \dots, \lambda_m(x, y) b_m(x_m, y_m)] \\
\hat{\phi}^T &= [\phi_1, \phi_2, \dots, \phi_m].
\end{aligned}$$

Uma vez que  $\phi^h(x, y) = b^T c = (c^T b)^T$ , a aproximação pode ser escrita como:

$$\phi^h(x, y) = c^T b = \hat{\phi}^T \underbrace{B^T A^{-T} b}_w \quad (20)$$

$$\phi^h(x, y) = \sum_{i,j}^m w_{ij}(x, y) \phi(x_i, y_j). \quad (21)$$

A partir da equação 21, é possível extrair uma importante informação a respeito da aproximação, como demonstrado por (SOUSA *et al.*, 2019), os pesos  $w_{ij}$  possuem a propriedade de recuperação da função base  $\phi$  em sua forma exata, ou ainda,  $\sum_{i,j}^m w_{ij} = 1$ . Esses pesos  $w_{ij}$  ponderam cada um dos elementos da vizinhança da célula onde está contido o ponto aproximado.

## 4 Discretização da equação de Poisson

Esta seção apresenta a discretização de uma equação de Poisson, para ilustrar as aproximações de dados em malhas com refinamento. O domínio é  $\Omega = \{(x, y) | a \leq x, y \leq b\} \subset \mathbb{R}^2$  e a solução analítica é  $\phi(x, y) = \sin(2\pi x) \sin(2\pi y)$ . A equação de Poisson é dada por

$$-\frac{\partial^2 \phi}{\partial x^2} - \frac{\partial^2 \phi}{\partial y^2} = f(x, y), \quad (22)$$

sendo consideradas condições de fronteira Dirichlet, ou seja, a solução é conhecida na fronteira do domínio e  $f(x, y)$  é um dado do problema. Neste caso,  $f(x, y)$  é dada por  $8\pi^2 \sin(2\pi x) \sin(2\pi y)$ .

Como apresentado em (SOUSA *et al.*, 2019), a equação (22) é discretizada por Diferenças Finitas. Em uma malha uniforme, para cada ponto da malha, temos uma equação discreta da forma:

$$-\frac{1}{h_x^2} (\phi_{i-1,j} - 2\phi_{i,j} + \phi_{i+1,j}) - \frac{1}{h_y^2} (\phi_{i,j-1} - 2\phi_{i,j} + \phi_{i,j+1}) = f_{i,j}, \quad (23)$$

sendo  $0 \leq i, j \leq N$  os valores para  $\phi_{i-1,j}, \phi_{i,j}, \phi_{i+1,j}, \phi_{i,j-1}, \phi_{i,j+1}$  serão renomeados daqui em diante para  $\phi_L, \phi_C, \phi_R, \phi_B$  e  $\phi_U$ , respectivamente, para tornar a notação mais simples.

A Figura 7 representa o estêncil formado para esta equação discreta em uma malha adaptativa. Nesse estêncil definido sobre o ponto escolhido (que está contido na célula de  $\phi_C$ ), é possível notar a ausência de uma célula específica para  $\phi_U$  e  $\phi_R$  nesta discretização e dada esta configuração de malha. Em outras palavras, os valores destas variáveis são desconhecidos, desta forma deseja-se aproximar essas duas incógnitas a fim de utilizá-las no estêncil.

De fato, quando o ponto central do estêncil se encontra na interface entre diferentes níveis das células da malha, como é o caso de  $\phi_C$ , alguns elementos do estêncil não possuem valor definido, nesse caso  $\phi_U$  e  $\phi_R$ . Porém é possível empregar uma aproximação da função

objetivo nesses pontos, utilizando os valores de células vizinhas, de maneira ponderada, para obter esses valores.

É possível então escrever uma combinação linear dos valores da função conhecidos na região da vizinhança do ponto a ser aproximado, tal que para descobrir os coeficientes desta combinação linear, o MLS é utilizado, como descrito na seção 3.2.1.

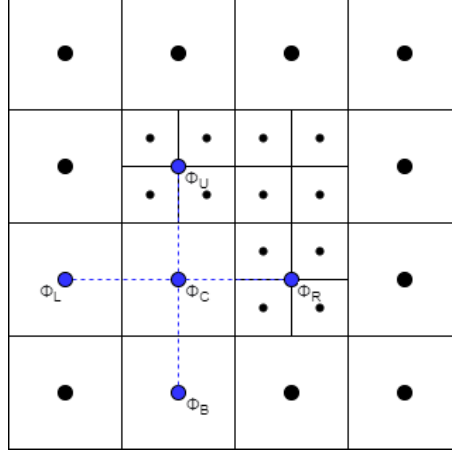


Figura 7 – Estêncil de discretização sobre um ponto arbitrário de uma malha adaptativa.  
Fonte: Autor, 2022.

Assim, uma vez expressando:

$$\phi_U = \sum_j^u w_j \phi_j \quad (24)$$

$$\phi_R = \sum_i^r w_i \phi_i \quad (25)$$

onde os limites da soma  $u$  e  $r$  se referem ao número de células na vizinhança da aproximação para cada um dos pontos, a equação (23) se torna:

$$-\frac{1}{h_x^2}(\phi_L - 2\phi_C - \sum_i^r w_i \phi_i) + \frac{1}{h_y^2}(\phi_B - 2\phi_C + \sum_j^u w_j \phi_j) = f_{i,j}. \quad (26)$$

Apesar de este trabalho não possuir como principal objetivo demonstrar as técnicas de geração de malhas adaptativas, mas apresentar ênfase na utilização de malhas adaptativas, os elementos das malhas utilizadas foram padronizados. Então é definido que as malhas devem ser estruturadas (cada nível de refinamento faz com que o tamanho de um elemento possua metade do tamanho de um elemento de nível anterior), e que os elementos são retângulos que possuem uma mesma orientação.

Logo, para a equação (26),  $h_x = h_y$ , fazendo com que a forma final da equação de Poisson por Diferenças Finitas seja:

$$-\frac{1}{h^2}(\phi_L - 4\phi_C + \sum_i^r w_i \phi_i + \phi_B + \sum_j^u w_j \phi_j) = f_{i,j}. \quad (27)$$

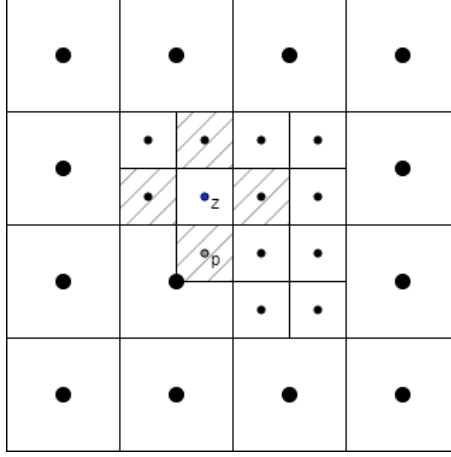


Figura 8 – Estêncil de discretização sobre um ponto refinado de uma malha adaptativa.  
Fonte: Autor, 2022.

A Figura 8 apresenta nas células tracejadas o estêncil em torno do ponto  $\hat{z}$  refinado. O ponto  $p$  desejado para ser incluído no sistema de equações de Diferenças Finitas não está presente na malha, logo deve ser aproximado pelo MLS, com um procedimento semelhante ao realizado para obter a equação (27).

#### 4.1 Montagem da matriz da equação de Poisson

Uma vez definida a discretização, será definido o método de construção do sistema linear para uma equação de Poisson bidimensional, com condição de contorno Dirichlet, esta que é definida pela equação 28 assim como bem descrita por (WANG *et al.*, 2016):

$$\begin{aligned}
 \phi(x_i, a_2) &= \phi_{i,0}, \\
 \phi(x_i, b_2) &= \phi_{i,N}, \\
 \phi(a_1, y_j) &= \phi_{0,j}, \\
 \phi(b_1, y_j) &= \phi_{N,j},
 \end{aligned} \tag{28}$$

onde  $a_i$  e  $b_i$  são os limites do domínio, neste caso para um domínio quadrado de lado unitário, os valores de  $a$  e  $b$  são iguais para  $x$  e  $y$ , tal que  $a_1 = a_2 \equiv a = 0$  e  $b_1 = b_2 \equiv b = 1$ ,  $N$  é o número de subintervalos do domínio discretizado, e o  $h$  base para a discretização de Poisson é definido por  $h_x = h_y \equiv h$ .

##### 4.1.1 Impacto das condições de fronteira e posição das variáveis

Neste tópico será descrito o impacto na discretização ao utilizar o conjunto de variáveis ao centro ou sobre os vértices das células do domínio. Considerando uma equação de Poisson bidimensional, com as variáveis sobre os vértices, a matriz de Poisson associada ao sistema de equações associadas a equação (23), com  $N = 4$  e condições de contorno Dirichlet, as equações para as quatro fronteiras do domínio são:

$$\begin{aligned}
-\phi_{2,j} + 4\phi_{1,j} - \phi_{1,j-1} - \phi_{1,j+1} &= h^2 f_{1,j} + \phi_{0,j}, \\
-\phi_{N-2,j} + 4\phi_{N-1,j} - \phi_{N-1,j-1} - \phi_{N-1,j+1} &= h^2 f_{N-1,j} + \phi_{N,j}, \\
-\phi_{i-1,1} - \phi_{i+1,1} + 4\phi_{i,1} - \phi_{i,2} &= h^2 f_{i,1} + \phi_{i,0}, \\
-\phi_{i-1,N-1} - \phi_{i+1,N-1} + 4\phi_{i,N-1} - \phi_{i,N-2} &= h^2 f_{i,N-1} + \phi_{i,N},
\end{aligned} \tag{29}$$

alguns termos da equação (29) são conhecidos, devidas as condições de contorno apresentadas na equação (28), assim o sistema completo equações pode ser representado na forma matricial, como:

$$\begin{bmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} \phi_{11} \\ \phi_{21} \\ \phi_{31} \\ \phi_{12} \\ \phi_{22} \\ \phi_{32} \\ \phi_{13} \\ \phi_{23} \\ \phi_{33} \end{bmatrix} = \begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \\ b_{12} \\ b_{22} \\ b_{32} \\ b_{13} \\ b_{23} \\ b_{33} \end{bmatrix},$$

sendo  $b$  um vetor que contém  $h^2 f_{ij}$  mais os valores na fronteira.

Já com as variáveis no centro da célula, os valores nos extremos do domínio aparecem nas equações do sistema linear, ao invés de serem tratados como valores conhecidos dadas as condições de contorno e adicionados no vetor  $b$ . As equações para os elementos da fronteira inferior do domínio, são da forma:

$$\underbrace{-(2\phi(x_1, a) - \phi_{1,1})}_{-\phi_{0,1}} - \phi_{2,1} + 4\phi_{1,1} - \underbrace{(2\phi(a, y_1) - \phi_{1,1})}_{-\phi_{1,0}} - \phi_{1,2} = h^2 f_{1,1}, \tag{30}$$

que, ao ser simplificada se torna

$$6\phi_{1,1} - \phi_{2,1} - \phi_{1,2} = h^2 f_{1,1} + 2\phi(x_1, a) + 2\phi(a, y_1), \tag{31}$$

a equação (31) descreve somente o canto inferior esquerdo do domínio, a equação que descreve o restante da borda inferior do domínio é

$$\underbrace{-(2\phi(x_1, a) - \phi_{1,j})}_{-\phi_{0,j}} - \phi_{2,j} + 4\phi_{1,j} - \phi_{1,j-1} - \phi_{1,j+1} = h^2 f_{1,j}, \tag{32}$$

ou na forma simplificada

$$5\phi_{1,j} - \phi_{2,j} - \phi_{1,j-1} - \phi_{1,j+1} = h^2 f_{1,j} + 2\phi(x_1, a). \tag{33}$$

As equações para as demais fronteiras do domínio podem ser deduzidas de maneira similar.

A Figura 9 apresenta apenas os elementos da matriz diferentes de zero. Esta matriz é classificada como uma matriz esparsa, o que significa que possui uma grande quantidade de zeros em seus elementos, esta característica faz com que resolver um sistema com

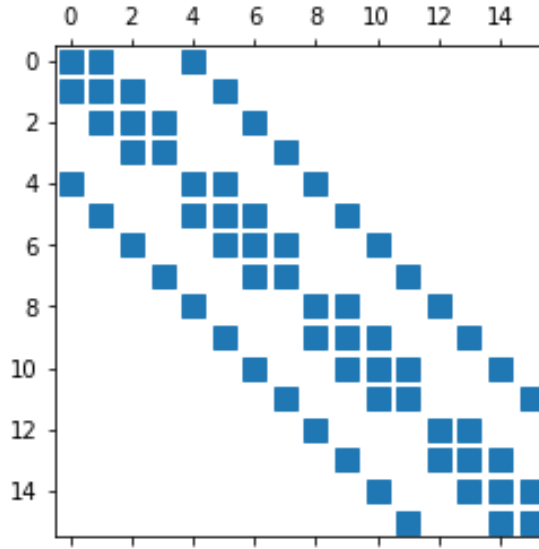


Figura 9 – Representação gráfica dos coeficientes não nulos da matriz de Poisson. Fonte: Autor, 2022.

este tipo de matriz seja muito interessante. Primeiro, é possível economizar memória ao armazenar essa matriz, uma vez que é possível armazenar somente os valores não nulos e suas posições, e reconstruir a estrutura da matriz quando necessário. Além disso, há diversas bibliotecas, como o pacote *sparse* na biblioteca *scipy* da linguagem *Python*, que possui métodos específicos para resolver sistemas, realizar produtos ou obter informações gerais como o determinante deste tipo de matriz.

A matriz obtida neste caso só possui essa forma devido a enumeração utilizada para os as incógnitas da matriz, uma vez que para diferentes enumerações, diferentes tipos de matriz são obtidos (ALVAREZ, 2013). No presente trabalho, a enumeração da malha é feita fixando um valor de  $y$ , e variando todos os valores de  $x$  em ordem crescente.

É importante também notar que a matriz com as variáveis sobre os vértices possui tamanho  $(N - 1)^2$ , já a matriz com as variáveis no centro das células possui tamanho  $N^2$ , considerando uma malha estruturada uniforme. Assim como indicado anteriormente, este trabalho utiliza a discretização com variáveis ao centro, e o sistema linear obtido se refere ao formado pelo conjunto de equações similares as equações (31) e (33).

## 4.2 Sistema de equações com aproximação por MLS

O sistema linear de equações consiste em um sistema similar a matriz discreta da equação de Poisson, mas onde em cada ponto que deveria possuir uma variável para uma malha uniforme, agora é acrescentado em 3 pontos adicionais para cada nível de refinamento presente na malha.

A matriz então deixa de possuir um tamanho  $N^2 \times N^2$  e passa a ser de tamanho  $M \times M$ , onde  $M$  é o número de células da malha, o número de incógnitas também aumenta para  $M$ , uma vez que agora ao invés de um valor para uma  $\phi_{ij}$  existem incógnitas adicionais oriundas do refinamento.

Os termos  $h$  para o lado direito do sistema são referidos como  $h_l$  neste momento,

uma vez que o nível de refinamento presente em cada célula altera o valor de  $h$  para o lado direito da equação.

Cada um dos elementos da matriz é ponderado por um peso  $w_{ij}$ , que se origina da aproximação por MLS caso o estêncil formado por um ponto apresente diferenças de nível. Como os pesos são gerados para cada célula da vizinhança do ponto aproximado, cada uma destas células agora compõe este ponto na matriz, juntamente com seus pesos.

A Figura 10 apresenta uma matriz obtida para este sistema, dada uma malha com  $M = 25$ , com enumeração dos elementos das incógnitas da matriz seguindo a ordem  $x$  fixo e variando valores de  $y$ , com o nível  $h_l$  possuindo a menor influência na ordenação.

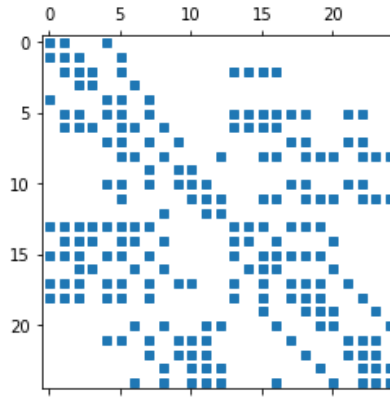


Figura 10 – Malha obtida para o sistema da equação de Poisson em malha refinada, com aproximações por MLS. Fonte: Autor, 2022.

## 5 Resultados numéricos

Para a implementação das aproximações por mínimos quadrados, primeira foi utilizada a função *least\_squares* da biblioteca *scipy.optimize* da linguagem *Python*. Na próxima fase de implementação, foi empregado o MMQ ponderado pela função peso baseada no inverso da distância entre dois pontos (o ponto a ser aproximado e cada um dos pontos da vizinhança), dada pela equação 17.

Foram avaliados o erro de aproximação na resolução do MMQ ponderado através do método iterativo e o erro de aproximação da função *least\_squares*, e por fim foi realizada a implementação do MLS para a obtenção dos pesos  $w_{ij}$  utilizados na matriz do sistema para solucionar a equação de Poisson.

A função *least\_squares* foi amplamente utilizada nos estágios iniciais da implementação, esta função resolve um problema não linear de mínimos quadrados através da minimização de uma função resíduo  $\frac{1}{2} \sum_i^m (f(x_i) - g(x_i))^2$  dados  $m$  pontos no domínio  $\Omega$  em  $\mathbb{R}^2$ .

A chamada da função é realizada na forma *least\_squares(fun, x0, args=())*, tal que *fun* é a chamada função resíduo dada pela soma do quadrado das diferenças entre o valor da função  $f$  e do valor inicial da função  $g$ ,  $x0$  é o conjunto de coeficientes que são utilizados como palpite inicial na aproximação, e *args* é o conjunto de argumentos adicionais que podem ser passados para a função resíduo.

É possível generalizar a função *least\_squares* para um domínio bidimensional, mas é importante que na passagem dos argumentos adicionais da função seja passada a lista de pares ordenados, ou a lista de pontos que será utilizada pelo método.

O retorno da função é um objeto que contém diversas informações sobre o processo de mínimos quadrados, tais como o vetor de coeficientes  $c$ , que em conjunto com a base definida compõem a aproximação por MMQ, além de informações como custo de operação, grau de otimalidade, e os resíduos da solução.

## 5.1 Fases de implementação

Nesta seção serão descritas as implementações que foram desenvolvidas ao longo do trabalho para a aproximação por mínimos quadrados, iniciando na implementação de um MMQ simples bidimensional, e culminando na implementação do sistema linear de Poisson para uma malha adaptativa com aproximações por MLS.

### 5.1.1 Primeira implementação

O código trabalha com uma malha uniforme com  $N$  subdivisões, com o objetivo de aproximar o valor da função  $f(x, y) = \sin(2\pi x)\sin(2\pi y)$  (Figura 11) em um ponto arbitrário, dentro do domínio representado por um quadrado de lado 1 ( $a = 0, b = 1$ ). Após a geração da malha e dado um ponto arbitrário  $\hat{z} = (x_p, y_p)$ , o algoritmo encontra em qual célula está localizado o ponto  $\hat{z}$ , e busca todas as células entre este ponto e uma distância qualquer, que neste caso foi escolhida a distância  $h$ , tal que  $h = \frac{(b-a)}{N}$ .

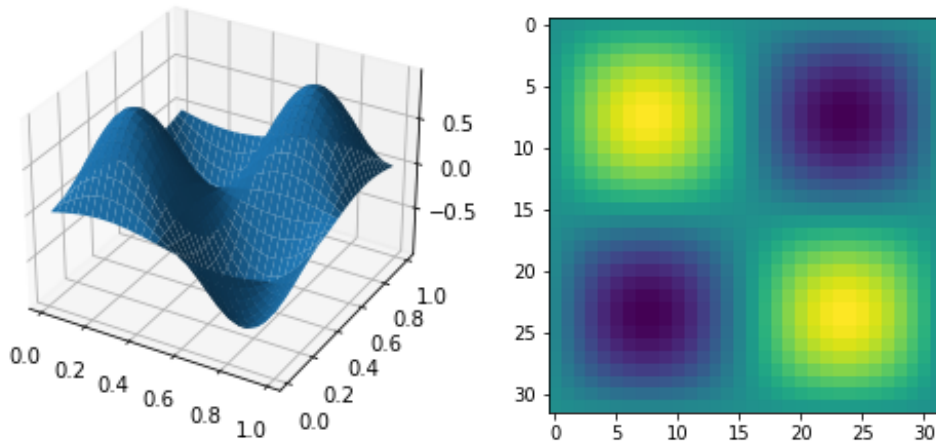


Figura 11 – Função a ser aproximada e sua projeção bidimensional. Fonte: Autor, 2022.

Com o ponto e as células vizinhas, o algoritmo passa para a função *least\_squares* o conjunto de pares ordenados que serão utilizados na aproximação, a função resíduo dada pela diferença entre o valor da função  $f$  e a combinação linear do palpite inicial dos coeficientes e os elementos da base  $b$ , que é a base polinomial bidimensional  $[1, x, y, xy, x^2, y^2]$ . Os coeficientes obtidos pela minimização realizada pelo algoritmo são utilizados para obter o valor da aproximação por MMQ no ponto escolhido.



### 5.1.2 Segunda implementação

Nesta implementação é iniciada a aproximação através de MMQ ponderado em malhas uniformes, foi utilizada a função peso da equação (17) multiplicando a função resíduo que é minimizada pela função *least\_squares*. O procedimento realizado é similar ao do MMQ simples da primeira implementação, com a adição da função peso.

Para esta implementação, os resultados da aproximação podem ser visualizados graficamente, uma vez que é obtida uma função de  $x$  e  $y$  do produto da base e coeficientes obtidos pela função da biblioteca *scipy*. Foi escolhido arbitrariamente o ponto  $\hat{z} = (0.7, 0.8)$ , com  $N = 16$ , os gráficos da Figura 12 descrevem a superfície gerada pela solução aproximada.

A Tabela 1 a ordem dos erros cometidos para aproximações no ponto  $\hat{z} = (0.7, 0.8)$  para malhas com diferentes tamanhos.

N	erro
8	$4.131 \times 10^{-2}$
16	$1.187 \times 10^{-3}$
32	$5.976 \times 10^{-5}$
64	$8.445 \times 10^{-5}$

Tabela 1 – Erros de aproximação com MMQ ponderados para diversas malhas uniformes. Fonte: Autor, 2022.

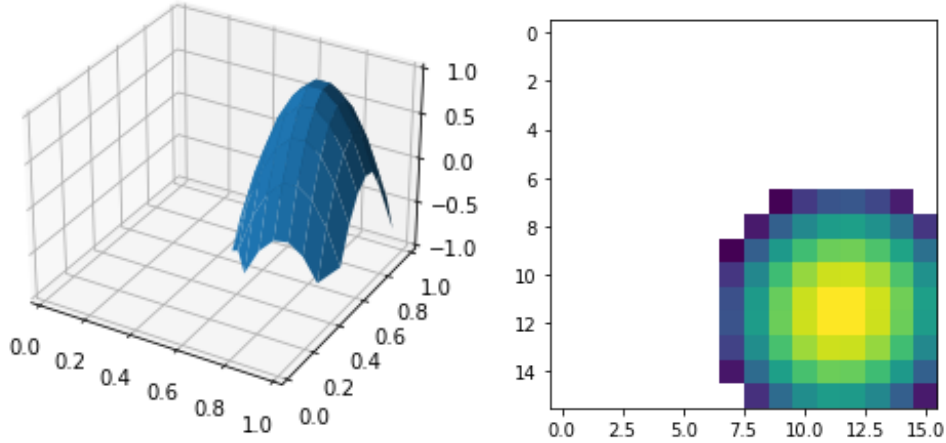


Figura 12 – Aproximação local de  $\sin(2\pi x)\sin(2\pi y)$  para uma malha uniforme e sua projeção. Fonte: Autor, 2022.

### 5.1.3 Terceira implementação

Nesta etapa de implementação foram iniciados os testes com malhas adaptativas, ainda utilizando a função *least\_squares*. A partir desta etapa, ao invés de realizar a combinação dos pares ordenados para dois vetores que definem  $x$  e  $y$ , devido a restrição de refinamento da malha que pode estar contida em uma região com qualquer número de células, a malha é passada como uma lista que contém os valores de  $x$ ,  $y$  e  $l$ , sendo  $l$  um valor que representa o nível de refinamento da malha naquele ponto.

A Figura 13 apresenta os gráficos referentes a aproximação local da função  $f(x, y)$  utilizando a função *least\_squares* para um MMQ ponderado em torno do ponto  $\hat{z} = (0.8, 0.8)$ . O ponto escolhido apresenta interface entre dois níveis de refinamento de células para a malha utilizada na simulação. O número de subdivisões  $N$  da malha é igual a 8 nesta aproximação, e o erro de aproximação é de  $4.13 \times 10^{-2}$ , o que corresponde ao erro para a malha uniforme. O resíduo minimizado, ou a otimalidade do método, pode ser visualizado para diferentes tamanho de malha (com base  $N$ ) na Tabela 2.

N	otimalidade
4	$7.389 \times 10^{-9}$
8	$8.938 \times 10^{-10}$
16	$1.667 \times 10^{-9}$
32	$4.527 \times 10^{-11}$

Tabela 2 – Fator de otimalidade da função *least\_squares* para diferentes tamanhos de malha. Fonte: Autor, 2022.

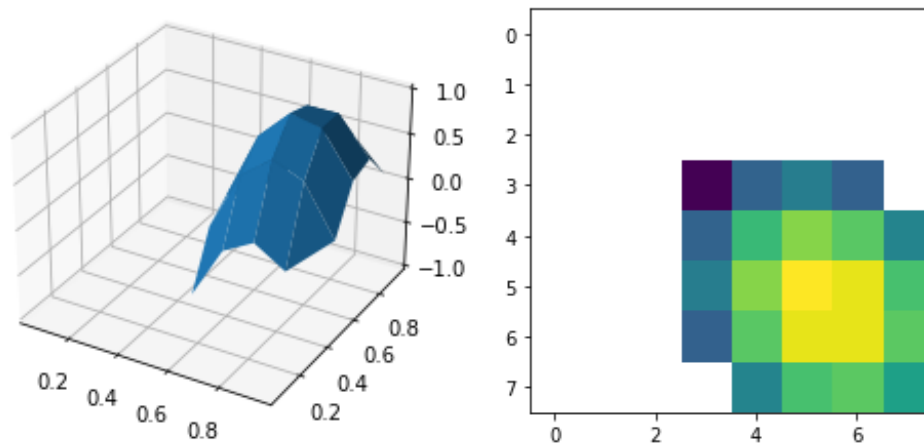


Figura 13 – Aproximação local de  $\sin(2\pi x)\sin(2\pi y)$  para uma malha adaptativa utilizando a função *least\_squares*. Fonte: Autor, 2022.

#### 5.1.4 Quarta implementação

Foi implementado o MMQ ponderado através da definição matricial do vetor de coeficientes  $c$ , assim como descrito na Seção 3.1, realizando o produto do sistema pela inversa da matriz  $A$ , a fim de isolar o vetor de coeficientes na equação, esta técnica também é demonstrada em (ZHANG; KWON; YOUN, 2005), (BALE *et al.*, 2021) e (LE; KHOO, 2017).

A fim de comparar os resultados do método da matriz inversa com a função *least\_squares*, foi utilizada a mesma malha adaptativa da Seção 5.1.3, e a aproximação foi feita para o mesmo ponto  $\hat{z} = (0.8, 0.8)$ . A Figura 14 representa a superfície e região geradas para esta aproximação. Além disso, o erro de aproximação para esta abordagem é similar ao erro de aproximação obtido utilizando a função *least\_squares*, sendo  $4.38 \times 10^{-2}$  nesta aproximação.

Para as duas técnicas de implementação do MMQ ponderado para malhas adaptativas (método iterativo e função *least\_squares*), a ordem do erro para malhas com maiores valores de  $N$  (e consequentemente mais elementos) corresponde a ordem presente na Tabela 1 para malhas uniformes. O valor exato dos erros para os dois métodos e diferentes tamanhos de malhas (com base  $N$ ) pode ser visualizado na Tabela 3.

$N$	erro <i>least_squares</i>	erro matriz inversa
8	$4.131 \times 10^{-2}$	$4.38 \times 10^{-2}$
16	$1.187 \times 10^{-3}$	$1.212 \times 10^{-3}$
32	$5.976 \times 10^{-5}$	$6.03 \times 10^{-5}$

Tabela 3 – Comparação de erros para as duas implementações de MMQ ponderado. Fonte: Autor, 2022.

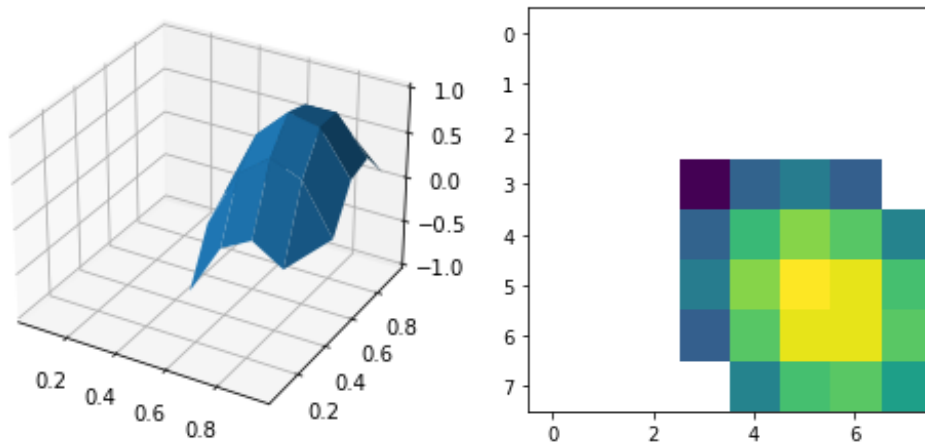


Figura 14 – Aproximação local de  $\sin(2\pi x)\sin(2\pi y)$  para uma malha adaptativa utilizando o método da matriz inversa. Fonte: Autor, 2022.

#### 5.1.5 Implementação de mínimos quadrados móveis

A partir da comparação das implementações descritas nas Seções 5.1.3 e 5.1.4, foi realizada a decisão da técnica de implementação do MLS. Para diversas malhas adaptativas e em pontos arbitrários, o erro local de aproximação da implementação através do método da matriz inversa possui a mesma ordem do erro obtido utilizando a função *least\_squares*, o que indica que ambos os métodos possuem um desempenho satisfatório.

A técnica empregada para o MLS então utiliza um método derivado da técnica da matriz inversa, como apresentado na Seção 3.2.1 unindo as metodologias de (ZHANG; GUO *et al.*, 2015) e (ZHANG; KWON; YOUN, 2005). Para esta implementação, a função peso utilizada passa a ser a função *spline* de ordem 3, utilizada também nas obras de (MOHAMMADI; SOTIROPOULOS; BRINKERHOFF, 2018) e (LE; KHOO, 2017), com suporte compacto (o que significa que a função peso só poderá ser diferente de zero para uma vizinhança próxima do ponto  $\hat{z}$ , como mostrado por (LIEW; YAN; ZHANG, 2017)).

É construída a matriz quadrada  $A$  a partir da soma ponderada dos elementos da matriz quadrada obtida através do produto da base por sua transposta.  $B$  é uma matriz

que contém os elementos da base ponderados, e avaliados sobre cada um dos elementos da vizinhança, e os pesos  $w_{ij}$  são definidos através do produto entre a matriz  $B$  transposta, a inversa da matriz  $A$  também transposta, e a base polinomial utilizada, avaliada no ponto de aproximação.

O procedimento descrito acima depende do ponto  $\hat{z}$  já que a matriz  $B$  e o valor da base polinomial também dependem do ponto  $\hat{z}$ , logo, a aproximação é válida para a região da vizinhança ao redor de  $\hat{z}$ , tal que cada célula vizinha possui seu próprio peso no cálculo da aproximação.

Os resultados numéricos das simulações realizada com o MLS demonstram que esta é uma técnica superior em relação ao MMQ tradicional e o MMQ ponderado. Dadas as três aproximações realizadas sobre um mesmo ponto da malha adaptativa. Quando o ponto escolhido está em uma subregião uniforme dentro da malha, o MLS obtém um erro na mesma ordem dos erros para as outras técnicas. Porém, quando o ponto se encontra na interface entre diferentes níveis, a aproximação por MLS apresenta um erro de aproximação menor do que as outras técnicas, em alguns casos sendo até uma ordem inteira inferior (ou dez vezes menor).

A Tabela 4 apresenta a comparação entre os erros dos três métodos, avaliados sobre diferentes pontos em uma malha com refinamento e base  $N = 8$ , para o domínio empregado até agora (quadrado de lado 1 com  $a = 0$  e  $b = 1$ ). Tal que na tabela a função *least\_squares* é chamada de Método 1, e o método da matriz inversa é chamado de Método 2.

$\hat{z}$	Método 1	Método 2	MLS
(0.8,0.8)	$4.131 \times 10^{-2}$	$4.38 \times 10^{-2}$	$4.4 \times 10^{-2}$
(0.1,0.6)	$5.8 \times 10^{-2}$	$7.7 \times 10^{-2}$	$7.4 \times 10^{-2}$
(0.6,0.55)	$2.73 \times 10^{-2}$	$2.89 \times 10^{-2}$	$8.38 \times 10^{-3}$
(0.65,0.55)	$2.49 \times 10^{-2}$	$2.49 \times 10^{-2}$	$1.5 \times 10^{-2}$

Tabela 4 – Comparação de erros das três implementações sobre diferente pontos. Fonte: Autor, 2022.

É possível notar que os dois últimos pontos avaliados, ou as duas últimas linhas da Tabela 4 possuem uma aproximação melhor através do MLS. Esses dois pontos estavam muito próximos a uma interface entre dois níveis de refinamento.

## 5.2 Segundo experimento numérico

Para implementar a matriz que descreve o sistema de equações discretas da equação de Poisson, é gerada uma matriz  $M \times M$  tal como enunciado na Seção 4.2. Cada elemento que existia na malha uniforme antes do refinamento, é dividido agora entre 4 novos elementos que são representados diretamente na matriz. O valor dos coeficientes que ponderam cada um destes novos elementos é definido pelo MLS seguindo os passos da Seção 4.

A Figura 15 apresenta a configuração das matrizes obtidas para diferentes malhas, com nível base da malha  $N = 4$ ,  $N = 8$  e  $N = 16$  (referindo-se ao número de subdivisões do domínio para as células menos refinadas da malha), ou ainda,  $M = 25$ ,  $M = 88$  e  $M = 352$ , respectivamente. A enumeração utilizada foi a mesma que na Seção 4.2.

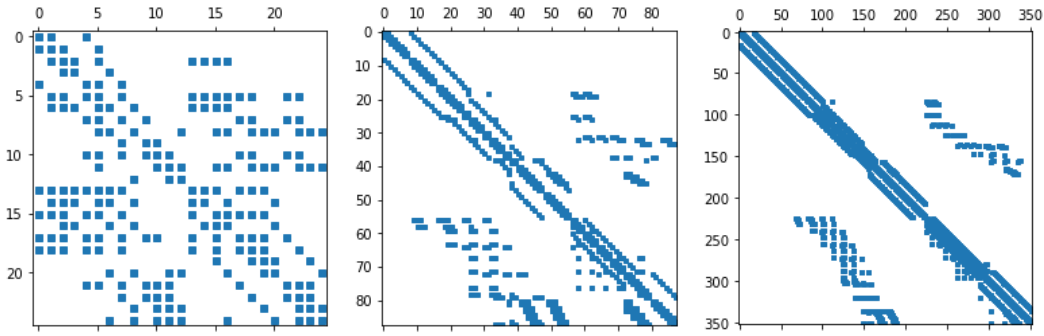


Figura 15 – Matrizes da equação de Poisson discretizada por diferenças finitas em uma malha adaptativa. Fonte: Autor, 2022.

O lado direito do sistema é um vetor com tamanho  $M$  (número de incógnitas) e é inicializado para cada elemento  $x_i, y_j$  da malha, o vetor recebe  $h_l^2 f_{ij}$ . Novamente, como é tratada uma malha adaptativa, o valor de  $h$  muda dependendo do nível da célula, e por isso é utilizada a notação  $h_l$ . Após essa inicialização, as condições de contorno são adicionadas para a correspondência com os extremos da matriz, a exemplo do lado direito das equações (31) e (33).

Para a solução deste sistema, foi utilizado o método *solve* do conjunto de álgebra linear da biblioteca *numpy* da linguagem *Python*. Para os três sistemas estudados, o erro obtido na aproximação da solução da EDP não foi satisfatório, obtendo erros de aproximação de ordens altas. Além disso, foi possível notar que, caso as incógnitas do sistema estejam dispostas de maneiras diferentes, o valor do erro obtido se altera, isso é explicado através do número de condição da matriz associada ao sistema.

Como explicado na obra de (PYZARA; BYLINA; BYLINA, 2011), o número de condição é um número associado a uma matriz que estima o quão sensível a perturbações nos dados um dado sistema é, ou seja, o número indica a dificuldade da resolução deste sistema. Este número pode ser calculado utilizando técnicas de decomposição de uma matriz, ou avaliando o produto da norma de uma matriz com a norma de sua inversa, como demonstrado em (CLINE *et al.*, 1979).

Em sua tese, (ALVAREZ, 2013) avalia a influência da enumeração dos elementos da matriz sobre a solução do sistema. Assim, além do fato dos sistemas possuírem grandes tamanhos, também existem os erros numéricos e limitações computacionais da máquina utilizada.

Um sistema é dito difícil de ser solucionado por métodos tradicionais (como métodos diretos alguns métodos iterativos que não convergem), caso o número de condição seja superior a 1. Para os sistemas obtidos da discretização da equação de Poisson em malhas refinadas, o menor sistema apresentou um número de condicionamento da ordem de  $10^1$ , e o maior dos sistemas estudados possui um número de condicionamento da ordem de  $10^2$ .

É possível utilizar técnicas e bibliotecas para alterar a enumeração dos elementos do sistema (além de técnicas de pré condicionamento), e melhorar a ordem das aproximações, como foi realizado por (ALVAREZ, 2013). Porém, o foco deste trabalho não reside na solução destes sistemas com configuração complexa, por isso, visto que uma técnica mais robusta de enumeração dos elementos era necessária, este trabalho não abrange esta vertente.

## 6 Considerações Finais e Trabalho Futuros

A aproximação de elementos de um estêncil de Diferenças Finitas, para malhas adaptativas, através do Método De Mínimos Quadrados Móveis (MLS) foi apresentada neste trabalho. Diversas formas do Método De Mínimos Quadrados (MMQ) foram implementadas. As implementações foram baseadas na forma tradicional do MMQ para uma malha uniforme, e a base polinomial utilizada para aproximar a função objetivo é  $b = [1, x, y, xy, x^2, y^2]$  em todos os casos. Iniciando pela implementação de um MMQ ponderado para malhas uniformes, que foi testado numericamente. Em seguida a implementação foi generalizada para uma malha adaptativa e seu erro numérico comparado ao da implementação anterior. Diferentes formas de implementar o MMQ foram comparadas, e a técnica tradicional que utiliza de uma matriz inversa foi empregada para o MLS.

O desempenho do MLS foi avaliado como superior em relação ao métodos tradicionais implementados. O erro de aproximação do MLS em comparação ao MMQ ponderado é da mesma ordem para uma região geral de uma malha, e apresenta uma melhoria para uma interface entre diferentes níveis de refinamento. Dadas essas considerações, conclui-se que é possível utilizar o MLS para aproximar um estêncil de Diferenças Finitas dada uma configuração complexa em uma malha adaptativa, apresentando resultados satisfatórios para todos os testes numéricos realizados.

Utilizando a implementação do MLS, o estêncil de Diferenças Finitas em malhas adaptativas foi construído para a equação de Poisson. Os pesos gerados pelo MLS foram utilizados para ponderar células vizinhas de um ponto que não pertence a malha, que teve seu valor aproximado no sistema. O sistema linear foi construído e sua solução avaliada, sendo constatado um problema de condicionamento da matriz que afeta a solução do sistema.

Dito isso, e analisando o desenvolvimento dos métodos e algoritmos empregados até aqui, é sugerido como um acréscimo a este trabalho que pode beneficiar muito a aplicação, o emprego de técnicas de pré condição do sistema linear, de forma a diminuir o número de condicionamento da matriz associada a equação de Poisson.

Outro ponto interessante a ser abordado em trabalhos futuros é o estudo e implementação de técnicas robustas para resolver um sistema linear, em específico técnicas que apresentam uma melhor taxa de convergência, ainda que para sistemas de alta ordem, evitando recorrer a métodos ou funções tradicionais presentes em algumas linguagens de programação.

Também é interessante que sejam estudados os graus de complexidade dos métodos implementados, como o MLS, buscando otimizações, e realizando comparações com outras técnicas que podem possuir características mais simples.

## Referências

- ALVAREZ, Catalina Maria Rua. Simulação computacional adaptativa de escoamentos bifásicos viscoelásticos. **Instituto de Matemática e Estatística**, Universidade de São Paulo, 2013. DOI: <10.11606/T.45.2013.tde-07082013-112937>.
- BABISHCHEVICH, P.N. Composite adaptive meshes in problems of mathematical physics. **USSR Computational Mathematics and Mathematical Physics**, v. 29, n. 3, p. 180–187, 1989. ISSN 0041-5553. DOI: <https://doi.org/10.1016/0041-5553(89)90165-1>.
- BALE, Rahul *et al.* A one-sided direct forcing immersed boundary method using moving least squares. **Journal of Computational Physics**, v. 440, p. 110359, 2021. ISSN 0021-9991. DOI: <https://doi.org/10.1016/j.jcp.2021.110359>.
- BELYTSCHKO, T. *et al.* Meshless methods: An overview and recent developments. **Computer Methods in Applied Mechanics and Engineering**, v. 139, n. 1, p. 3–47, 1996. ISSN 0045-7825. DOI: <https://doi.org/10.1016/S0045-7825(96)01078-X>.
- BERGER, Marsha J; OLIGER, Joseph. Adaptive mesh refinement for hyperbolic partial differential equations. **Journal of Computational Physics**, v. 53, n. 3, p. 484–512, 1984. ISSN 0021-9991. DOI: <https://doi.org/10.1016/0021-9991(84)90073-1>.
- BERGER, Marsha J. Stability of Interfaces with Mesh Refinement. **Mathematics of Computation**, American Mathematical Society, v. 45, n. 172, p. 301–318, 1985. ISSN 00255718, 10886842. DOI: <https://doi.org/10.2307/2008126>. Acesso em: 7 dez. 2022.
- BRENNER, Susanne C.; SCOTT, Larkin R. **The Mathematical Theory of Finite Element Methods**. [S.l.]: Springer, 2008. v. 15. (Texts in Applied Mathematics). ISBN 9780387759333. DOI: <10.1007/978-0-387-75934-0>.
- CLINE, A. K. *et al.* An Estimate for the Condition Number of a Matrix. **SIAM Journal on Numerical Analysis**, v. 16, n. 2, p. 368–375, 1979. DOI: <10.1137/0716029>.
- CUNHA, Maria Cristina C. **Métodos Numéricos**. [S.l.]: Unicamp, 2010. ISBN 9788526808775.
- FORTUNA, Armando de Oliveira. **Técnicas computacionais para dinâmica dos fluidos: conceitos básicos e aplicações**. [S.l.]: USP, 2000.
- FREIRE, Atila Pantaleão Silva. **Teoria de camada limite**. [S.l.]: Núcleo Interdisciplinar de Dinâmica dos Fluidos - UFRJ, 2019.
- LANCASTER, Peter; SALKAUSKAS, Kestutis. Surfaces generated by moving least squares methods. **Mathematics of Computation**, v. 37, p. 141–158, 1981. DOI: <https://doi.org/10.2307/2007507>.
- LE, Duc-Vinh; KHOO, Boo-Cheong. A Moving-Least-Square Immersed Boundary Method for Rigid and Deformable Boundaries in Viscous Flow. **Communications in Computational Physics**, Cambridge University Press, v. 22, n. 4, p. 913–934, 2017. DOI: <10.4208/cicp.OA-2016-0164>.
- LEVEQUE, Randall J. **Finite Difference Methods for Ordinary and Partial Differential Equations**. [S.l.]: Society for Industrial e Applied Mathematics, 2007. DOI: <10.1137/1.9780898717839>.

LIEW, K.M.; YAN, Jian-Wei; ZHANG, Lu-Wen. Chapter 4 - Atomistic-Continuum Theory. In: LIEW, K.M.; YAN, Jian-Wei; ZHANG, Lu-Wen (Ed.). **Mechanical Behaviors of Carbon Nanotubes**. [S.l.]: Elsevier, 2017. (Micro and Nano Technologies). P. 141–248. ISBN 978-0-323-43137-8. DOI: <<https://doi.org/10.1016/B978-0-323-43137-8.00004-4>>. Disponível em: <<<https://www.sciencedirect.com/science/article/pii/B9780323431378000044>>>.

LIU, Shi-Lin *et al.* Deep Implicit Moving Least-Squares Functions for 3D Reconstruction. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. [S.l.: s.n.], 2021. DOI: <10.1109/CVPR46437.2021.00183>.

MIN, Chohong; GIBOU, Frédéric. A second order accurate projection method for the incompressible Navier–Stokes equations on non-graded adaptive grids. **Journal of Computational Physics**, v. 219, n. 2, p. 912–929, 2006. ISSN 0021-9991. DOI: <<https://doi.org/10.1016/j.jcp.2006.07.019>>.

MOHAMMADI, M.; SOTIROPOULOS, F.; BRINKERHOFF, Joshua. Moving Least Squares Reconstruction for Sharp Interface Immersed Boundary Methods: MLS reconstruction in the immersed boundary method. **International Journal for Numerical Methods in Fluids**, v. 90, dez. 2018. DOI: <10.1002/fld.4711>.

NEALEN, Andrew. An As-Short-As-Possible Introduction to the Least Squares, Weighted Least Squares and Moving Least Squares Methods for Scattered Data Approximation and Interpolation, jan. 2004.

PYZARA, Anna; BYLINA, Beata; BYLINA, Jarosław. The influence of a matrix condition number on iterative methods' convergence. In: 2011 Federated Conference on Computer Science and Information Systems (FedCSIS). [S.l.: s.n.], 2011. P. 459–464. ISBN 978-83-60810-22-4.

SOUSA, F.S. *et al.* A finite difference method with meshless interpolation for incompressible flows in non-graded tree-based grids. **Journal of Computational Physics**, v. 396, p. 848–866, 2019. ISSN 0021-9991. DOI: <<https://doi.org/10.1016/j.jcp.2019.07.011>>.

STARIUS, Göran. On composite mesh difference methods for hyperbolic differential equations. **Numerische Mathematik**, v. 35, p. 241–255, jan. 1980. DOI: <10.1007/BF01396411>. Disponível em: <<<https://doi.org/10.1007/BF01396411>>>.

THOMPSON, Lonny L.; PINSKY, Peter M. A Galerkin least-squares finite element method for the two-dimensional Helmholtz equation. **International Journal for Numerical Methods in Engineering**, v. 38, n. 3, p. 371–397, 1995. DOI: <<https://doi.org/10.1002/nme.1620380303>>.

WANG, Hanquan *et al.* An efficient implementation of fourth-order compact finite difference scheme for Poisson equation with Dirichlet boundary conditions. **Computers & Mathematics with Applications**, v. 71, n. 9, p. 1843–1860, 2016. ISSN 0898-1221. DOI: <<https://doi.org/10.1016/j.camwa.2016.02.022>>.

WENDLAND, Holger. Local polynomial reproduction and moving least squares approximation. **IMA Journal of Numerical Analysis**, v. 21, n. 1, p. 285–300, 2001. ISSN 0272-4979. DOI: <10.1093/imanum/21.1.285>.

ZHANG, Huaqing; GUO, Chunxian *et al.* Measurement Data Fitting Based on Moving Least Squares Method. **Mathematical Problems in Engineering**, v. 2015, p. 1–10, 2015. DOI: <<http://dx.doi.org/10.1155/2015/195023>>.



ZHANG, Xiang Kun; KWON, Kie-Chan; YOUN, Sung-Kie. The least-squares meshfree method for the steady incompressible viscous flow. **Journal of Computational Physics**, v. 206, n. 1, p. 182–207, 2005. ISSN 0021-9991. DOI: <<https://doi.org/10.1016/j.jcp.2004.11.033>>.