

UNIVERSIDADE FEDERAL DE JUIZ DE FORA  
INSTITUTO DE CIÊNCIAS EXATAS  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**Estudo de variantes da Decomposição em  
Modos Dinâmicos aplicadas à dinâmica de  
sedimentos em escoamentos.**

**Breno Lino Prado**

JUIZ DE FORA  
JANEIRO, 2026

# **Estudo de variantes da Decomposição em Modos Dinâmicos aplicadas à dinâmica de sedimentos em escoamentos.**

BRENO LINO PRADO

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Bacharelado em Ciência da Computação

Orientador: Prof. D.Sc. José Jerônimo Camata

JUIZ DE FORA

JANEIRO, 2026

ESTUDO DE VARIANTES DA DECOMPOSIÇÃO EM MODOS  
DINÂMICOS APLICADAS À DINÂMICA DE SEDIMENTOS EM  
ESCOAMENTOS.

Breno Lino Prado

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS  
EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTE-  
GRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE  
BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Prof. D.Sc. José Jerônimo Camata  
Departamento de Ciência da Computação/UFJF

Prof. D.Sc. Bernardo Martins Rocha  
Departamento de Ciência da Computação/UFJF

Prof. D.Sc. Ruy Freitas Reis  
Departamento de Ciência da Computação/UFJF

JUIZ DE FORA  
23 DE JANEIRO, 2026

# Resumo

Este trabalho investiga a aplicação da Decomposição em Modos Dinâmicos (DMD) e de suas variantes sequenciais na modelagem e reconstrução da dinâmica temporal de escoamentos com transporte de sedimentos. Considerando a alta dimensionalidade dos dados gerados por simulações de Dinâmica dos Fluidos Computacional (CFD), modelos de ordem reduzida surgem como uma alternativa eficiente para a análise e aproximação da dinâmica dominante do sistema.

Foram implementados em C++ o DMD clássico, o *Online DMD* e o *Streaming DMD*, sendo os métodos aplicados a diferentes janelas temporais de uma simulação numérica de escoamento com sedimentos, a partir de *snapshots* bidimensionais em escala de cinza. A comparação entre os métodos foi conduzida de forma controlada, adotando postos reduzidos consistentes e estratégias uniformes de inicialização.

A avaliação do desempenho foi realizada exclusivamente por meio de métricas quantitativas, incluindo o erro quadrático médio (MSE), o *Peak Signal-to-Noise Ratio* (PSNR), bem como o custo computacional em termos de tempo de execução e uso de memória. Os resultados evidenciam diferenças relevantes entre as abordagens, destacando o compromisso entre precisão, robustez temporal e eficiência computacional na escolha do método mais adequado para aplicações sequenciais envolvendo transporte de sedimentos.

**Palavras-chave:** Decomposição em Modos Dinâmicos; Modelos de Ordem Reduzida; Transporte de Sedimentos.

# Abstract

This work investigates the application of Dynamic Mode Decomposition (DMD) and its sequential variants for modeling and reconstructing the temporal dynamics of sediment-laden flows. Due to the high dimensionality of data generated by Computational Fluid Dynamics (CFD) simulations, reduced-order models represent an efficient alternative for capturing the dominant system dynamics at lower computational cost.

The classical DMD, Online DMD, and Streaming DMD methods were implemented in C++ and applied to different temporal windows of a numerical sediment transport simulation, using two-dimensional grayscale snapshots as input data. The comparison between methods was conducted under controlled conditions, adopting consistent reduced-order ranks and uniform initialization strategies.

Performance evaluation was carried out exclusively using quantitative metrics, including the mean squared error (MSE), the Peak Signal-to-Noise Ratio (PSNR), as well as computational cost in terms of execution time and memory usage. The results highlight relevant differences among the approaches, emphasizing the trade-off between accuracy, temporal robustness, and computational efficiency when selecting an appropriate DMD-based method for sequential applications involving sediment transport.

**Keywords:** Dynamic Mode Decomposition; Reduced-Order Models; Sediment Transport.

# Conteúdo

<b>Lista de Figuras</b>	<b>4</b>
<b>Lista de Tabelas</b>	<b>5</b>
<b>Lista de Abreviações</b>	<b>6</b>
<b>1 Introdução</b>	<b>7</b>
1.1 Apresentação do Tema . . . . .	7
1.2 Contextualização . . . . .	8
1.3 Motivação e Justificativa . . . . .	9
1.4 Descrição do Problema . . . . .	10
1.5 Objetivos . . . . .	10
1.5.1 Objetivo Geral . . . . .	10
1.5.2 Objetivos Específicos . . . . .	11
1.6 Organização do Texto . . . . .	11
<b>2 Fundamentação Teórica</b>	<b>12</b>
2.1 Dinâmica dos Fluidos Computacional e Modelos de Ordem Reduzida . . .	12
2.2 Decomposição em Modos Dinâmicos (DMD) . . . . .	13
2.2.1 Formulação matemática do DMD clássico . . . . .	14
2.2.2 Limitações do DMD clássico e variantes . . . . .	19
<b>3 Metodologia</b>	<b>21</b>
3.1 Visão geral da metodologia . . . . .	21
3.2 Origem e caracterização dos dados . . . . .	22
3.2.1 Organização temporal e definição dos <i>snapshots</i> . . . . .	23
3.3 Implementação computacional . . . . .	24
3.3.1 Arquitetura geral da biblioteca em C++ . . . . .	24
3.3.2 Bibliotecas e ferramentas utilizadas . . . . .	25
3.4 Implementação do DMD clássico . . . . .	26
3.4.1 Definição do posto reduzido . . . . .	28
3.5 Implementação do <i>Online DMD</i> . . . . .	29
3.6 Implementação do <i>Streaming DMD</i> . . . . .	31
3.7 Métricas de avaliação . . . . .	33
<b>4 Resultados e Discussões</b>	<b>36</b>
4.1 Estudo de Caso: Corrente de Sedimentos . . . . .	36
4.2 Resultados nas Simulações com Sedimentos . . . . .	39
4.2.1 Experimento E1: Imagens de 1000 a 1500 . . . . .	39
4.2.2 Experimento E2: Imagens de 2000 a 2500 . . . . .	42
4.2.3 Experimento E3: Imagens de 3000 a 3500 . . . . .	45
4.3 Comparação entre os Métodos . . . . .	48
<b>5 Conclusões</b>	<b>51</b>
<b>Bibliografia</b>	<b>53</b>

## Lista de Figuras

3.1	Campo de sedimentos obtido diretamente da simulação de CFD. . . . .	23
4.1	Evolução temporal do campo de concentração de sedimentos em instantes representativos da simulação. . . . .	37
4.2	Comparação entre o campo de sedimentos original e as reconstruções obtidas pelos diferentes métodos no instante $t = 1250$ (Experimento E1). . . .	40
4.3	Evolução do erro quadrático médio ao longo do tempo para o DMD clássico no Experimento E1. . . . .	41
4.4	Evolução do erro quadrático médio ao longo do tempo para o <i>Online DMD</i> no Experimento E1. . . . .	41
4.5	Evolução do erro quadrático médio ao longo do tempo para o <i>Streaming DMD</i> no Experimento E1. . . . .	42
4.6	Comparação entre o campo de sedimentos original e as reconstruções obtidas pelos diferentes métodos no instante $t = 2250$ (Experimento E2). . . .	43
4.7	Evolução do erro quadrático médio ao longo do tempo para o DMD clássico no Experimento E2. . . . .	44
4.8	Evolução do erro quadrático médio ao longo do tempo para o <i>Online DMD</i> no Experimento E2. . . . .	44
4.9	Evolução do erro quadrático médio ao longo do tempo para o <i>Streaming DMD</i> no Experimento E2. . . . .	45
4.10	Comparação entre o campo de sedimentos original e as reconstruções obtidas pelos diferentes métodos no instante $t = 3250$ (Experimento E3). . . .	46
4.11	Evolução do erro quadrático médio ao longo do tempo para o DMD clássico no Experimento E3. . . . .	47
4.12	Evolução do erro quadrático médio ao longo do tempo para o <i>Online DMD</i> no Experimento E3. . . . .	47
4.13	Evolução do erro quadrático médio ao longo do tempo para o <i>Streaming DMD</i> no Experimento E3. . . . .	48
4.14	Comparação do erro quadrático médio entre os métodos em um intervalo temporal recortado, anterior ao crescimento acentuado do erro do <i>Online DMD</i> . . . . .	49

## Lista de Tabelas

4.1	Janelas temporais avaliadas nos experimentos com sedimentos. . . . .	38
4.2	Valores de PSNR (em dB) das reconstruções no instante $t = 1250$ no Experimento E1. . . . .	42
4.3	Valores de PSNR (em dB) das reconstruções no instante $t = 2250$ no Experimento E2. . . . .	45
4.4	Valores de PSNR (em dB) das reconstruções no instante $t = 3250$ no Experimento E3. . . . .	48
4.5	Tempo total de execução dos métodos nos experimentos realizados. . . .	50
4.6	Pico de memória RAM consumida pelos métodos nos experimentos realizados.	50



## Lista de Abreviações

CFD	Dinâmica dos Fluidos Computacional
ROM	Modelo de Ordem Reduzida
POD	Decomposição Ortogonal Própria
DMD	Decomposição em Modos Dinâmicos
SVD	Decomposição em Valores Singulares
MSE	Erro Quadrático Médio ( <i>Mean Squared Error</i> )
PSNR	<i>Peak Signal-to-Noise Ratio</i>
STL	<i>Standart Template Library</i>
RAM	<i>Random Access Memory</i>

# 1 Introdução

## 1.1 Apresentação do Tema

A Dinâmica dos Fluidos Computacional (do inglês *Computational Fluid Dynamics* – CFD) dedica-se ao estudo e à simulação numérica de escoamentos por meio da combinação de modelos físicos, métodos numéricos e técnicas computacionais. Nas últimas décadas, essa abordagem tornou-se essencial para investigar fenômenos complexos em que a interação entre fases e mecanismos de transporte desempenha papel central, como em problemas com partículas ou sedimentos em suspensão (ZHANG et al., 2024b; SCHMID, 2022).

Escoamentos com dinâmica de sedimentos tipicamente exibem forte não linearidade, múltiplas escalas temporais e espaciais e regimes transientes, o que dificulta tanto a modelagem quanto a análise de dados provenientes de simulações de alta fidelidade. Nesses cenários, técnicas capazes de extrair estruturas coerentes e representar a dinâmica dominante de forma compacta tornam-se particularmente relevantes (LIBERO et al., 2024; BAO et al., 2019).

Entre as abordagens baseadas em dados, destaca-se a *Dynamic Mode Decomposition* (DMD), que decompõe séries temporais de campos de alta dimensionalidade em modos espaciais associados a frequências e taxas de crescimento ou decaimento. Essa decomposição permite identificar padrões dominantes e interpretar mecanismos dinâmicos presentes na evolução do sistema (GIORGI; FICARELLA; FONTANAROSA, 2018; SCHMID, 2022). Para ampliar a aplicabilidade do método em situações práticas, diversas variantes têm sido propostas para lidar com ruído, com dados em fluxo contínuo e com dinâmicas não estacionárias, incluindo formulações *online/streaming* e estratégias de regularização (SCHMID, 2022; LIBERO et al., 2024).

## 1.2 Contextualização

A simulação de sistemas multifásicos com partículas ou sedimentos em suspensão permanece entre os desafios mais relevantes em CFD, dada a combinação de alta dimensionalidade, acoplamentos físico-matemáticos e elevados custos computacionais (GRAVE; CAMATA; COUTINHO, 2020; ZHANG et al., 2024b). Como consequência, análises paramétricas, estudos de incerteza e aplicações em tempo quase real podem ser limitados pelo tempo de processamento e pelo grande volume de dados gerado.

Nesse contexto, modelos de ordem reduzida (*Reduced-Order Models* – ROMs) têm se mostrado ferramentas eficazes para capturar a dinâmica dominante do sistema com menor custo computacional. Em termos gerais, ROMs buscam identificar subespaços de baixa dimensão que preservem o comportamento essencial do sistema, evitando a necessidade de resolver integralmente o modelo de alta fidelidade a cada nova consulta (BERKOOZ; HOLMES; LUMLEY, 1993; BENNER; GUGERCIN; WILLCOX, 2015; ROWLEY; DAWSON, 2017; ROWLEY; COLONIUS; MURRAY, 2004). Exemplos clássicos incluem métodos baseados em decomposição ortogonal própria (POD), projeções de Galerkin e suas extensões (por exemplo, Petrov–Galerkin), amplamente utilizados em dinâmica dos fluidos e em problemas de transporte (BERKOOZ; HOLMES; LUMLEY, 1993; ROWLEY; COLONIUS; MURRAY, 2004).

Além de técnicas puramente projetivas, abordagens híbridas vêm combinando ROMs por projeção com estratégias baseadas em dados para aumentar robustez e precisão, especialmente em regimes turbulentos e/ou com modelos de fechamento. Um exemplo é a combinação de POD–Galerkin com aproximações orientadas por dados para representar termos associados à viscosidade turbulenta em formulações por volumes finitos (HIJAZI et al., 2020). Trabalhos recentes também propõem estruturas híbridas projeção/dados para equações de Navier–Stokes parametrizadas (ZHANG et al., 2024a) e discutem aplicações em sistemas multifásicos, como escoamentos gás–sólido, empregando aprendizado profundo para modelar a dinâmica modal (LI; DUAN; SAKAI, 2024).

Entre as abordagens de ROMs fundamentadas em dados, o DMD ocupa posição de destaque por extrair diretamente, a partir de séries temporais, modos associados a características dinâmicas, sendo amplamente utilizado em análise modal e em construção

de modelos preditivos em fluidos (TAIRA et al., 2017). Entretanto, o DMD clássico pode apresentar limitações em presença de ruído e quando a dinâmica apresenta mudanças estruturais ao longo do tempo. Por isso, variantes do DMD têm sido desenvolvidas para melhorar estabilidade numérica e robustez, bem como para lidar com dados *streaming*, janelas deslizantes e mudanças de regime (SCHMID, 2022; LIBERO et al., 2024).

Dado o amplo espectro de variantes existentes e a diversidade de comportamentos presentes em escoamentos com transporte de sedimentos, torna-se relevante compreender como essas abordagens se comportam em cenários distintos, especialmente no que diz respeito à reconstrução temporal, à previsão e à robustez a ruídos e variações topológicas.

### 1.3 Motivação e Justificativa

A análise de escoamentos com transporte de sedimentos demanda ferramentas capazes de representar, com eficiência e precisão, sistemas caracterizados por múltiplas escalas, acoplamento entre fases e dinâmica transiente. Em aplicações ambientais, industriais e geotécnicas — como rios, canais, reatores e dutos — essa dinâmica impacta diretamente a previsão de processos de erosão, deposição, desgaste e transporte de partículas.

Embora simulações numéricas de alta fidelidade forneçam descrições detalhadas do sistema, seu custo computacional pode inviabilizar varreduras paramétricas, rotinas de otimização, quantificação de incerteza ou uso operacional. Nesse cenário, ROMs e decomposições modais baseadas em dados oferecem alternativas promissoras ao permitir representações compactas do comportamento dominante.

Apesar do avanço recente de variantes do DMD, ainda há lacunas quanto ao seu desempenho em sistemas multifásicos com sedimentos, especialmente quando há ruído, transientes e mudanças estruturais no escoamento. Em particular, são necessários estudos que avaliem, de forma comparativa e quantitativa, a eficácia dessas variantes para reconstrução, previsão e robustez em cenários representativos.

Diante desse contexto, torna-se relevante investigar abordagens que conciliem redução de custo computacional com a capacidade de preservar as principais características dinâmicas do escoamento multifásico com sedimentos. Assim, este trabalho adota técnicas de redução de ordem e decomposições modais baseadas em dados como es-

estratégia para estudar o desempenho de variantes da Decomposição em Modos Dinâmicos.

## 1.4 Descrição do Problema

Sistemas com transporte de sedimentos podem apresentar mudanças topológicas, regimes transientes e sensibilidade a ruídos, tornando desafiadora a obtenção de representações compactas que sejam, ao mesmo tempo, interpretáveis e preditivas. A DMD e suas variantes são candidatas naturais para análise, reconstrução e previsão, porém ainda é pouco compreendido como diferentes formulações se comportam especificamente em cenários envolvendo partículas em suspensão.

Além disso, faltam comparações padronizadas que permitam avaliar, sob condições controladas, a capacidade das diferentes variantes em (i) reconstruir campos temporais, (ii) prever estados futuros e (iii) manter robustez frente a ruídos e variações estruturais. Para viabilizar tais comparações de maneira reprodutível, é desejável dispor de uma implementação unificada, modular e extensível que permita configurar experimentos, instrumentar desempenho e reproduzir resultados.

Dessa forma, o problema central deste trabalho envolve tanto a investigação do desempenho de variantes do DMD em escoamentos com transporte de sedimentos quanto o desenvolvimento de uma implementação computacional que viabilize tais análises de forma eficiente, padronizada e reprodutível.

## 1.5 Objetivos

### 1.5.1 Objetivo Geral

Desenvolver e avaliar variantes da decomposição em modos dinâmicos (DMD) aplicadas a escoamentos com transporte de sedimentos, por meio da implementação de uma biblioteca modular em C++ (disponível em: <https://github.com/brenolino/dmd-variations>) e da análise comparativa de sua capacidade de reconstrução e robustez frente a ruídos e mudanças topológicas.

### 1.5.2 Objetivos Específicos

Para alcançar o objetivo geral, este trabalho propõe os seguintes objetivos específicos:

1. Implementar em C++ o DMD clássico e as variantes *Online DMD* e *Streaming DMD* em uma infraestrutura unificada, modular e extensível, possibilitando experimentos reprodutíveis e controlados.
2. Aplicar essas variantes a conjuntos de dados provenientes de simulações numéricas de escoamentos com partículas em suspensão.
3. Avaliar a capacidade de cada variante em reconstruir campos do sistema.
4. Comparar quantitativamente os métodos com base em métricas padronizadas, identificando vantagens, limitações e cenários de melhor aplicabilidade.

## 1.6 Organização do Texto

Além desta introdução, este trabalho está organizado da seguinte forma: o Capítulo 2, *Fundamentação Teórica*, apresenta os conceitos de CFD e modelos de ordem reduzida, com ênfase na Decomposição em Modos Dinâmicos e em suas principais variantes. O Capítulo 3, *Metodologia*, descreve a abordagem adotada, a origem e caracterização dos dados, a infraestrutura computacional implementada em C++ e as variantes consideradas (DMD clássico, *Online DMD* e *Streaming DMD*), bem como as métricas de avaliação. O Capítulo 4, *Resultados e Discussões*, reúne a validação das implementações e os resultados dos estudos de caso, incluindo a comparação entre os métodos. Por fim, o Capítulo 5, *Conclusões*, apresenta as conclusões do trabalho e possibilidades de continuidade.

## 2 Fundamentação Teórica

### 2.1 Dinâmica dos Fluidos Computacional e Modelos de Ordem Reduzida

A Dinâmica dos Fluidos Computacional desempenha um papel central na análise de escoamentos complexos, especialmente em sistemas que envolvem interação entre fases, como aqueles com transporte de sedimentos. Esses escoamentos frequentemente apresentam não linearidades pronunciadas, múltiplas escalas temporais e espaciais, além de regimes transientes que demandam elevado custo computacional para serem simulados em alta fidelidade (GRAVE; CAMATA; COUTINHO, 2020). Em muitos casos, a realização de análises paramétricas ou estudos em larga escala torna-se inviável devido ao custo associado às simulações completas.

Com o intuito de mitigar essas limitações, os Modelos de Ordem Reduzida têm sido empregados como alternativas eficientes para representar os principais comportamentos dinâmicos do sistema utilizando um número reduzido de graus de liberdade. Em geral, esses modelos são construídos a partir de dados provenientes de simulações ou experimentos, permitindo capturar estruturas coerentes e relevantes para a dinâmica do escoamento (BEIT-SADI; KROL; WYNN, 2021; GIORGI; FICARELLA; FONTANAROSA, 2018).

Entre as técnicas mais utilizadas na construção de ROMs destacam-se a Decomposição Ortogonal Própria e o Dynamic Mode Decomposition. Enquanto a POD busca modos que maximizam a energia representada nos dados, o DMD fornece uma visão espectral da dinâmica ao associar modos espaciais a frequências e taxas de crescimento ou decaimento, tornando-o particularmente adequado para a análise de fenômenos oscilatórios, instabilidades e padrões transientes observados em escoamentos complexos (SCHMID, 2022).

Outra característica importante do DMD é sua natureza baseada em dados, o que permite aplicá-lo como ferramenta de pós-processamento para extrair informações

dinâmicas de simulações CFD sem a necessidade de acessar explicitamente as equações governantes do escoamento (CALMET et al., 2020; NADERI; EIVAZI; ESFAHANIAN, 2019). Essa flexibilidade possibilita a construção de modelos compactos capazes de reproduzir a evolução temporal do sistema e identificar modos dominantes associados à sua dinâmica global.

A literatura recente demonstra um amplo espectro de aplicações do DMD e de suas variantes em escoamentos turbulentos, cavitantes e multifásicos reforçando seu potencial como ferramenta versátil na análise modal de sistemas fluidodinâmicos (LIBERO et al., 2024; ZHANG et al., 2024b).

Nas seções seguintes, apresenta-se a formulação matemática do método DMD clássico, bem como os princípios que fundamentam sua utilização na construção de modelos de ordem reduzida empregados em CFD.

## 2.2 Decomposição em Modos Dinâmicos (DMD)

A *Decomposição em Modos Dinâmicos* é uma técnica de análise modal que extrai, a partir de dados temporais, estruturas espaciais coerentes associadas a comportamentos dinâmicos bem definidos. O método identifica modos espaciais e suas respectivas taxas de crescimento, decaimento ou oscilação, permitindo caracterizar a evolução temporal do sistema sem recorrer explicitamente às equações governantes do escoamento (SCHMID, 2022; CALMET et al., 2020).

A ideia central do DMD é que a dinâmica observada pode, em muitos casos, ser aproximada por um operador linear que descreve a evolução temporal dos dados. A partir dessa aproximação, o método decompõe o comportamento do sistema em modos dinâmicos, cada um representando uma estrutura espacial característica e um padrão temporal associado (GIORGI; FICARELLA; FONTANAROSA, 2018).

Além de sua aplicação em análise modal e redução de ordem, o DMD também permite descrever a evolução temporal do sistema a partir dos modos identificados. Isso possibilita tanto a representação dos estados já observados quanto a extrapolação de seu comportamento para instantes futuros, o que o torna especialmente relevante para aplicações em previsão e reconstrução de campos fluidodinâmicos (QIN et al., 2019; BAO



et al., 2019).

A interpretação temporal do modelo deriva dos autovalores do operador dinâmico aproximado, que codificam informações sobre frequência e sobre taxas de crescimento ou amortecimento associadas a cada modo. Esses elementos resultam em uma descrição compacta e interpretável da dinâmica, capaz de capturar instabilidades, padrões recorrentes e comportamentos transientes.

De forma conceitual, o DMD parte da aproximação

$$\mathbf{x}_{k+1} \approx \mathbf{A} \mathbf{x}_k, \quad (2.1)$$

na qual um operador linear desconhecido  $\mathbf{A}$  relaciona estados consecutivos do sistema. O método constrói, a partir dos dados, uma representação reduzida desse operador que permite extrair seus autovalores e autovetores associados. Os autovetores projetados definem os modos dinâmicos, enquanto os autovalores determinam seu comportamento temporal.

A próxima subseção apresenta a formulação matemática clássica do DMD, detalhando as etapas necessárias para sua aplicação em problemas de fluidodinâmica computacional.

### 2.2.1 Formulação matemática do DMD clássico

#### Organização dos dados

Considere uma sequência de  $m$  estados observados de um sistema dinâmico, obtidos em instantes igualmente espaçados  $t_1, t_2, \dots, t_m$ , com passo temporal  $\Delta t$ . Cada estado é representado por um vetor coluna

$$\mathbf{x}_k \in \mathbb{R}^n,$$

onde  $n$  corresponde ao número de graus de liberdade do sistema. No contexto deste trabalho, cada  $\mathbf{x}_k$  é obtido pela vetorização de um campo bidimensional (simulação CFD) em um único vetor de dimensão  $n$ .

Para formular o DMD, esses vetores são organizados em duas matrizes defasadas:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_{m-1} \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} \mathbf{x}_2 & \mathbf{x}_3 & \cdots & \mathbf{x}_m \end{bmatrix}. \quad (2.2)$$

com  $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n \times (m-1)}$ . O objetivo do DMD é encontrar um operador linear  $\mathbf{A} \in \mathbb{R}^{n \times n}$  tal que, para cada par de estados consecutivos, tenhamos a aproximação

$$\mathbf{x}_{k+1} \approx \mathbf{A} \mathbf{x}_k, \quad k = 1, \dots, m-1. \quad (2.3)$$

o que, em forma matricial, se escreve como

$$\mathbf{Y} \approx \mathbf{A} \mathbf{X}. \quad (2.4)$$

A matriz  $\mathbf{A}$  representa o operador dinâmico que aproxima a evolução temporal do sistema no intervalo amostrado. Entretanto, sua dimensão  $n \times n$  é tipicamente muito elevada em aplicações de fluidodinâmica, tornando inviável a computação direta desse operador. Assim, o DMD utiliza técnicas de redução de ordem para obter uma representação equivalente de baixa dimensão que capture os principais padrões dinâmicos do sistema.

### Estimativa do operador dinâmico via mínimos quadrados

Os vetores  $\mathbf{x}_k$  que compõem as matrizes  $\mathbf{X}$  e  $\mathbf{Y}$  representam amostras do estado do sistema em instantes sucessivos. Esses vetores são frequentemente referidos na literatura como *snapshots*.

Partindo da organização dos dados discutida anteriormente, a evolução temporal observada pode ser aproximada por meio da relação linear Eq. (2.4) na qual  $\mathbf{A}$  atua como um operador dinâmico que aproxima a transição entre estados consecutivos. A estimativa desse operador pode ser formulada como o problema de mínimos quadrados

$$\min_{\mathbf{A}} \|\mathbf{Y} - \mathbf{A} \mathbf{X}\|_F. \quad (2.5)$$

cujas solução é dada pela pseudoinversa de Moore–Penrose:

$$\mathbf{A} = \mathbf{Y} \mathbf{X}^\dagger. \quad (2.6)$$

Apesar de sua simplicidade conceitual, essa formulação não é adequada para aplicações de fluidodinâmica, pois o operador  $\mathbf{A}$  teria dimensão  $n \times n$ , onde  $n$  corresponde ao número total de graus de liberdade do campo espacial. Como  $n$  é tipicamente muito grande em problemas reais, a computação direta de  $\mathbf{A}$  torna-se inviável em termos de custo computacional e armazenamento.

Por essa razão, o DMD emprega uma representação reduzida do operador dinâmico, obtida por meio da decomposição em valores singulares (SVD), apresentada a seguir.

### Redução de ordem via decomposição em valores singulares

Para tornar viável a estimativa do operador dinâmico, o DMD utiliza uma representação reduzida do sistema baseada na decomposição em valores singulares da matriz  $\mathbf{X}$ . A SVD permite identificar os componentes espaciais mais relevantes dos dados, possibilitando a projeção da dinâmica em um subespaço de dimensão muito menor que  $n$ .

A decomposição é dada por

$$\mathbf{X} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^\top \quad (2.7)$$

onde  $\mathbf{U} \in \mathbb{R}^{n \times n}$  e  $\mathbf{V} \in \mathbb{R}^{(m-1) \times (m-1)}$  são matrizes ortogonais, e  $\mathbf{\Sigma}$  é uma matriz diagonal contendo os valores singulares de  $\mathbf{X}$  em ordem decrescente.

Em aplicações práticas, retêm-se apenas os  $r$  maiores valores singulares, resultando na aproximação truncada

$$\mathbf{X} \approx \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^\top \quad (2.8)$$

na qual  $\mathbf{U}_r \in \mathbb{R}^{n \times r}$ ,  $\mathbf{\Sigma}_r \in \mathbb{R}^{r \times r}$  e  $\mathbf{V}_r \in \mathbb{R}^{(m-1) \times r}$ . Essa truncção define o subespaço dominante no qual a dinâmica será projetada.

Substituindo essa aproximação na relação  $\mathbf{Y} \approx \mathbf{A} \mathbf{X}$  e multiplicando à esquerda por  $\mathbf{U}_r^\top$ , obtém-se

$$\mathbf{U}_r^\top \mathbf{Y} \approx \mathbf{U}_r^\top \mathbf{A} \mathbf{U}_r \boldsymbol{\Sigma}_r \mathbf{V}_r^\top. \quad (2.9)$$

Como o objetivo é representar a dinâmica no subespaço gerado por  $\mathbf{U}_r$ , define-se

$$\tilde{\mathbf{A}} = \mathbf{U}_r^\top \mathbf{A} \mathbf{U}_r. \quad (2.10)$$

o que permite reescrever a expressão anterior como

$$\mathbf{U}_r^\top \mathbf{Y} \approx \tilde{\mathbf{A}} \boldsymbol{\Sigma}_r \mathbf{V}_r^\top. \quad (2.11)$$

Multiplicando à direita por  $\mathbf{V}_r \boldsymbol{\Sigma}_r^{-1}$ , obtém-se então

$$\tilde{\mathbf{A}} \approx \mathbf{U}_r^\top \mathbf{Y} \mathbf{V}_r \boldsymbol{\Sigma}_r^{-1}. \quad (2.12)$$

que é o operador dinâmico reduzido associado ao subespaço dominante da SVD. A matriz  $\tilde{\mathbf{A}} \in \mathbb{R}^{r \times r}$  captura a dinâmica essencial do sistema e torna o problema computacionalmente viável, permitindo calcular explicitamente seus autovalores e autovetores na etapa seguinte.

### Autovalores e modos dinâmicos

Uma vez obtida a matriz reduzida  $\tilde{\mathbf{A}} \in \mathbb{R}^{r \times r}$ , a dinâmica do sistema pode ser analisada por meio de sua decomposição espectral. Os autovalores e autovetores de  $\tilde{\mathbf{A}}$  satisfazem

$$\tilde{\mathbf{A}} \mathbf{W} = \mathbf{W} \boldsymbol{\Lambda}. \quad (2.13)$$

onde  $\mathbf{W} \in \mathbb{C}^{r \times r}$  contém os autovetores à direita e  $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_r)$  é a matriz diagonal de autovalores. Cada autovalor  $\lambda_j$  descreve o comportamento temporal associado a um modo dinâmico: seu módulo indica a taxa de crescimento ou decaimento, enquanto seu argumento determina a frequência de oscilação.

Para recuperar os modos no espaço original, projeta-se os autovetores de  $\tilde{\mathbf{A}}$  de volta ao espaço de dimensão  $n$  por meio da expressão

$$\Phi = \mathbf{Y} \mathbf{V}_r \Sigma_r^{-1} \mathbf{W} = \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \phi_r \end{bmatrix}. \quad (2.14)$$

em que cada coluna  $\phi_j \in \mathbb{C}^n$  representa um modo DMD. Esses modos correspondem a estruturas espaciais coerentes cuja evolução temporal é governada pelo autovalor associado  $\lambda_j$ . Modos com  $|\lambda_j| > 1$  indicam crescimento, enquanto  $|\lambda_j| < 1$  indicam amortecimento. Modos com autovalores complexos apresentam comportamento oscilatório.

A coleção formada por  $(\phi_j, \lambda_j)$  fornece uma descrição compacta e interpretável da dinâmica essencial do sistema, permitindo reconstruir sua evolução temporal a partir de combinações lineares desses modos. Essa reconstrução é apresentada na subseção seguinte.

### Evolução temporal e reconstrução

A partir dos modos dinâmicos  $\Phi$  e dos autovalores associados  $\lambda_j$ , é possível descrever a evolução temporal aproximada do sistema no subespaço dominante identificado pelo DMD. O primeiro passo consiste em determinar os coeficientes iniciais que expressam o estado inicial como combinação dos modos:

$$\mathbf{x}_1 \approx \Phi \mathbf{b} \quad \Rightarrow \quad \mathbf{b} = \Phi^\dagger \mathbf{x}_1 \quad (2.15)$$

onde  $\Phi^\dagger$  denota a pseudoinversa de Moore–Penrose e  $\mathbf{b} = [b_1, \dots, b_r]^\top$  contém as amplitudes iniciais associadas a cada modo.

Como os autovalores  $\lambda_j$  representam a evolução temporal discreta dos modos, costuma-se introduzir a forma contínua

$$\omega_j = \frac{\ln(\lambda_j)}{\Delta t} \quad (2.16)$$

na qual  $\text{Re}(\omega_j)$  descreve a taxa de crescimento ou decaimento e  $\text{Im}(\omega_j)$  fornece a frequência angular associada ao modo  $\phi_j$ . Com isso, a solução aproximada do sistema pode ser expressa como

$$\mathbf{x}(t) \approx \sum_{j=1}^r b_j \phi_j e^{\omega_j t} \quad (2.17)$$

o que permite reconstruir os estados observados e, se desejado, extrapolar a dinâmica para instantes futuros. A qualidade dessa reconstrução depende da escolha do posto  $r$ , da riqueza dos dados e da adequação da aproximação linear ao comportamento real do sistema.

A representação modal fornecida pelo DMD concentra a dinâmica essencial do sistema em um conjunto reduzido de modos coerentes, possibilitando analisar padrões temporais, identificar instabilidades e construir modelos compactos capazes de reproduzir a evolução do escoamento de forma eficiente. Essa formulação serve de base para as variantes do DMD apresentadas posteriormente e para as implementações desenvolvidas neste trabalho.

### 2.2.2 Limitações do DMD clássico e variantes

Apesar de sua formulação consolidada, a decomposição em modos dinâmicos apresenta limitações estruturais que motivaram o surgimento de variantes mais robustas e adaptativas. Em sua configuração clássica, o DMD depende da disponibilidade simultânea de todos os pares de *snapshots*, o que o torna inadequado para aplicações em que os dados chegam de forma sequencial ou quando o volume de informações excede a capacidade de armazenamento. Além disso, a estimação do operador dinâmico é sensível ao condicionamento das matrizes de dados, reduzindo sua robustez frente a ruídos e perturbações — condição comum em medições experimentais ou simulações extensas.

Outro fator limitante é a suposição de que a dinâmica global pode ser representada por um operador linear fixo. Em sistemas não estacionários ou sujeitos a mudanças topológicas, essa hipótese deixa de ser válida, comprometendo a capacidade do método de acompanhar a evolução temporal do sistema (ZHANG et al., 2019). Adicionalmente, o custo computacional associado à decomposição em valores singulares torna-se proibitivo em aplicações de Dinâmica dos Fluidos Computacional, onde cada *snapshot* pode conter dezenas ou centenas de milhares de graus de liberdade (HEMATI; WILLIAMS; ROWLEY, 2014).

Para superar tais limitações, diferentes variantes do DMD têm sido propostas. Algumas introduzem regularização para melhorar a estabilidade frente a ruídos; outras

ampliam a capacidade de representação por meio de expansões não lineares no espaço de observáveis; e há ainda métodos que incorporam informações físicas ao processo de decomposição. Particular relevância é dada às abordagens adaptativas e incrementais, capazes de atualizar o modelo continuamente à medida que novos dados chegam, evitando o custo e a rigidez do processamento em lotes.

Entre essas abordagens, destaca-se o *DMD Online*, que realiza uma atualização recursiva do operador dinâmico reduzido à medida que novos *snapshots* são observados, dispensando o armazenamento de toda a matriz de dados. Essa formulação possibilita acompanhar sistemas cuja dinâmica é explicitamente dependente do tempo e permite incorporar fatores de esquecimento que priorizam informações recentes (ZHANG et al., 2019).

Outra técnica relevante é o *Streaming DMD*, que utiliza uma formulação de baixa memória para atualizar incrementalmente as bases modais e as projeções dinâmicas a cada novo *snapshot*. Esse método reduz a necessidade de armazenamento, incorpora mecanismos de compressão baseados em decomposição modal e mantém apenas as direções dinamicamente mais relevantes, mesmo na presença de ruídos significativos (HEMATI; WILLIAMS; ROWLEY, 2014).

Essas variantes tornam-se particularmente adequadas para aplicações com grandes volumes de dados, dinâmica variável e presença de ruídos — características típicas de escoamentos com transporte de sedimentos. As próximas seções apresentam os algoritmos empregados neste trabalho, bem como sua implementação em C++.

## 3 Metodologia

### 3.1 Visão geral da metodologia

A metodologia adotada neste trabalho tem como objetivo viabilizar a análise comparativa de variantes da decomposição em modos dinâmicos aplicadas a escoamentos com transporte de sedimentos, combinando a utilização de dados provenientes de simulações numéricas com o desenvolvimento de uma infraestrutura computacional própria em C++.

Inicialmente, são utilizados conjuntos de dados gerados por simulações de Dinâmica dos Fluidos Computacional, nos quais a evolução temporal do escoamento é representada por campos espaciais discretizados ao longo do tempo. Esses campos são disponibilizados na forma de imagens previamente processadas, as quais são organizadas sequencialmente e interpretadas como amostras do estado do sistema dinâmico.

A partir dessas amostras, os dados são estruturados de modo a permitir a aplicação dos métodos de decomposição modal. Em particular, cada imagem é convertida em um vetor de estado, e as sequências temporais são organizadas conforme a formulação do DMD apresentada no capítulo de fundamentação teórica. Essa organização possibilita a aplicação direta do DMD clássico, bem como de suas variantes incrementais.

No âmbito computacional, é desenvolvida uma biblioteca modular em C++, projetada para implementar o DMD clássico, o *Online DMD* e o *Streaming DMD* de forma unificada. Essa infraestrutura permite a execução de experimentos sob condições controladas, garantindo consistência metodológica e reprodutibilidade dos resultados.

Por fim, as diferentes variantes são avaliadas com base em métricas que contemplam a qualidade da reconstrução temporal e aspectos computacionais relevantes. As etapas seguintes deste capítulo detalham a origem e a caracterização dos dados utilizados, bem como as estratégias de implementação adotadas para cada algoritmo.



## 3.2 Origem e caracterização dos dados

Os dados utilizados neste trabalho foram obtidos a partir de simulações numéricas de Dinâmica dos Fluidos Computacional envolvendo escoamentos com transporte de sedimentos. Nessas simulações, o domínio físico do problema é representado por uma malha computacional baseada em elementos finitos, a partir da qual são obtidos campos espaciais que descrevem a evolução temporal do escoamento. Esses campos resultam da resolução numérica das equações que governam o sistema, realizada por ferramentas de simulação apropriadas, produzindo informações definidas sobre a malha ao longo do tempo.

A partir desse processo, são gerados campos bidimensionais representativos do estado do sistema em instantes temporais sucessivos. Tais campos correspondem a amostras igualmente espaçadas no tempo e descrevem grandezas de interesse associadas à dinâmica do transporte de sedimentos. Para fins de análise, essas informações foram exportadas no formato de imagens, preservando a estrutura espacial necessária para a aplicação das técnicas de decomposição modal.

As imagens utilizadas neste trabalho foram disponibilizadas juntamente com a biblioteca associada ao projeto, sendo obtidas por meio da execução de um script automatizado que realiza o download e a organização dos dados. Importante destacar que essas imagens já se encontram previamente processadas, incluindo segmentação da região de interesse e conversão para escala de cinza. Dessa forma, não foi necessário realizar etapas adicionais de pré-processamento sobre os dados brutos antes da aplicação dos métodos estudados.

Para fins de contextualização, a Figura 3.1 apresenta um exemplo do campo original gerado pela simulação de CFD antes das etapas de segmentação da região de interesse e conversão para escala de cinza. Essa visualização ilustra a extensão completa do domínio computacional e o tipo de informação espacial disponível nos dados brutos.

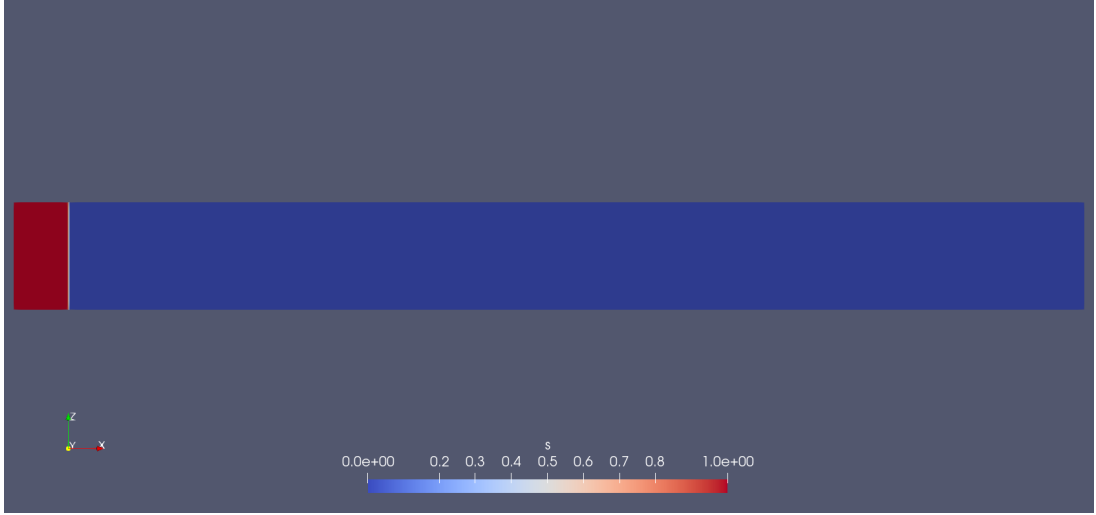


Figura 3.1: Campo de sedimentos obtido diretamente da simulação de CFD.

Cada imagem representa, portanto, um estado do sistema dinâmico em um dado instante de tempo e é interpretada como uma amostra da evolução temporal do escoamento. Essas amostras são utilizadas diretamente na construção das sequências temporais analisadas pelos métodos de decomposição em modos dinâmicos. A organização temporal dessas imagens e sua conversão para vetores de estado, são detalhadas na seção seguinte.

### 3.2.1 Organização temporal e definição dos *snapshots*

Para a aplicação dos métodos de decomposição em modos dinâmicos, os dados devem ser organizados como uma sequência temporal de estados do sistema. No contexto deste trabalho, cada estado é representado por uma imagem bidimensional proveniente da simulação numérica, correspondente a um instante específico da evolução do escoamento.

Cada imagem é interpretada como uma amostra do estado do sistema dinâmico e, seguindo a terminologia adotada na literatura, é denominada um *snapshot*. Essas imagens são ordenadas temporalmente de acordo com o instante em que foram geradas, assumindo um intervalo de tempo constante entre amostras consecutivas.

Para viabilizar a aplicação dos algoritmos de DMD, cada imagem bidimensional é convertida em um vetor coluna por meio de um processo de vetorização, no qual os valores de intensidade dos pixels são organizados sequencialmente em um único vetor. Dessa forma, cada *snapshot* é representado por um vetor de estado em um espaço de alta dimensão, preservando a informação espacial originalmente contida na imagem.

A sequência de vetores de estado é então organizada em duas matrizes defasadas no tempo, conforme a formulação clássica do DMD. A primeira matriz é composta pelos *snapshots* correspondentes aos instantes iniciais, enquanto a segunda contém os *snapshots* deslocados temporalmente em uma unidade de tempo. Essa organização estabelece a relação temporal necessária para a estimativa do operador dinâmico que aproxima a evolução do sistema.

Essa estrutura de dados é utilizada de forma consistente em todas as variantes estudadas neste trabalho. No caso do DMD clássico, as matrizes completas são construídas previamente. Já nas variantes incrementais, como o *Online DMD* e o *Streaming DMD*, os *snapshots* são incorporados progressivamente ao longo do tempo, respeitando a mesma lógica de organização temporal. Os detalhes específicos de cada abordagem são apresentados nas seções seguintes.

### 3.3 Implementação computacional

#### 3.3.1 Arquitetura geral da biblioteca em C++

Com o objetivo de viabilizar experimentos controlados e reprodutíveis, foi desenvolvida uma biblioteca em C++ para a implementação do DMD clássico e de suas variantes *Online* e *Streaming*. A biblioteca foi projetada com uma arquitetura modular, de modo a reduzir redundâncias entre implementações, facilitar a extensão para novas variantes e garantir consistência entre os procedimentos numéricos empregados em cada método.

A organização do código separa claramente as responsabilidades relacionadas (i) à leitura e preparação dos dados, (ii) à execução dos algoritmos de decomposição, e (iii) ao cálculo de medidas necessárias para avaliação posterior. Essa separação permite que os diferentes métodos sejam aplicados sobre a mesma estrutura de dados e com interfaces consistentes, isolando o efeito das variações algorítmicas e evitando interferências decorrentes de diferenças de implementação.

Em termos de funcionamento, a biblioteca opera sobre uma sequência temporal de *snapshots* vetorizados, a partir da qual são calculados os elementos necessários para a análise dinâmica do sistema. Esses elementos incluem os modos dinâmicos associados,

bem como as informações temporais utilizadas nos processos de reconstrução da dinâmica temporal.

A configuração dos algoritmos é realizada por meio de parâmetros definidos explicitamente, como o posto de truncamento e as estratégias de atualização adotadas em cada variante. Essa abordagem garante que diferentes métodos sejam avaliados sob condições controladas e comparáveis, preservando a consistência metodológica entre os experimentos realizados.

As seções seguintes apresentam em detalhes a implementação do DMD clássico e de suas variantes, destacando as particularidades algorítmicas e computacionais de cada abordagem.

### 3.3.2 Bibliotecas e ferramentas utilizadas

A implementação da biblioteca foi realizada integralmente em C++, com o objetivo de garantir eficiência computacional, controle explícito de memória e flexibilidade na integração dos diferentes algoritmos de decomposição estudados. Para viabilizar as operações numéricas necessárias, foram utilizadas bibliotecas amplamente consolidadas no contexto científico e de computação de alto desempenho.

Em particular, a biblioteca Eigen<sup>1</sup> foi empregada para a realização de operações de álgebra linear, como multiplicações matriciais, decomposição em valores singulares e resolução de sistemas lineares. A escolha da Eigen deve-se à sua eficiência, interface expressiva baseada em templates e ampla utilização em aplicações científicas, além de oferecer suporte direto a matrizes densas e operações necessárias para a implementação dos métodos de DMD.

Além disso, foram utilizadas estruturas padrão da biblioteca padrão do C++ (STL) para gerenciamento de dados, controle de fluxo e organização das estruturas internas, contribuindo para a portabilidade e manutenção do código. Essa combinação de ferramentas permitiu desenvolver uma infraestrutura computacional eficiente e modular, adequada ao processamento de dados de alta dimensionalidade provenientes de simulações numéricas.

---

<sup>1</sup><https://libeigen.gitlab.io/>

O uso dessas bibliotecas fornece uma base sólida para a implementação consistente das diferentes variantes do DMD, permitindo que as diferenças observadas nos resultados estejam associadas às características algorítmicas de cada método, e não a limitações da infraestrutura computacional adotada.

### 3.4 Implementação do DMD clássico

A implementação do DMD clássico foi estruturada em duas etapas complementares: (i) o **ajuste do modelo** a partir de uma matriz de *snapshots* e (ii) a **reconstrução** temporal a partir do modelo ajustado. Essas etapas correspondem diretamente aos métodos `DMD::fit()` e `DMD::predict()`, respectivamente.

No ajuste, a entrada é a matriz  $\mathbf{X} \in \mathbb{R}^{n \times m}$ , onde cada coluna representa um *snapshot* vetorizado do sistema. O objetivo é extrair uma representação modal de baixa dimensão associada a um operador linear que aproxima a evolução temporal observada nos dados. Para isso, a sequência é organizada em duas matrizes defasadas no tempo,  $\mathbf{X}_1$  e  $\mathbf{X}_2$ , contendo estados consecutivos do sistema, o que estabelece a relação temporal necessária para a aproximação do operador dinâmico. Em seguida, calcula-se a decomposição em valores singulares de  $\mathbf{X}_1$  e aplica-se truncamento para um posto  $r$ , reduzindo a dimensionalidade do problema e tornando viável a construção do operador dinâmico reduzido. Nesta implementação, o parâmetro  $r$  pode ser fornecido explicitamente ou determinado automaticamente a partir do espectro de valores singulares, dependendo da configuração adotada no experimento.

Com a base reduzida determinada, constrói-se o operador reduzido  $\tilde{\mathbf{A}}$ , que aproxima a dinâmica no subespaço dominante. A decomposição espectral desse operador fornece os autovalores  $\boldsymbol{\lambda}$  e autovetores associados, que são utilizados para reconstruir os modos dinâmicos  $\boldsymbol{\Phi}$  no espaço original. Por fim, calcula-se o vetor de amplitudes iniciais  $\mathbf{b}$  por meio de uma pseudoinversa, ajustando a combinação modal ao primeiro estado observado. A conversão dos autovalores discretos para frequências contínuas  $\boldsymbol{\omega}$  é realizada por  $\boldsymbol{\omega} = \ln(\boldsymbol{\lambda})/\Delta t$ , o que permite avaliar a solução em instantes arbitrários no tempo.

O Algoritmo 1 resume o fluxo computacional implementado no método `DMD::fit()`.

Uma vez ajustado o modelo, a reconstrução temporal é realizada a partir da

**Algoritmo 1:** Ajuste do Modelo DMD (`DMD::fit()`)

---

**Entrada:**  $\mathbf{X} \in \mathbb{R}^{n \times m}$ ,  $\Delta t$ , posto  $r$  (ou seleção automática)  
**Saída:**  $\Phi$ ,  $\lambda$ ,  $\mathbf{b}$ ,  $\omega$

```

1 início
2   Calcule dimensões de  $\mathbf{X}$ :  $(n, m) \leftarrow \text{shape}(\mathbf{X})$ ;
3   Calcule matriz defasada  $\mathbf{X}_1$ :  $\mathbf{X}_1 \leftarrow \mathbf{X}[:, 1 : (m - 1)]$ ;
4   Calcule matriz defasada  $\mathbf{X}_2$ :  $\mathbf{X}_2 \leftarrow \mathbf{X}[:, 2 : m]$ ;
5   Calcule estado inicial:  $\mathbf{x}_0 \leftarrow \mathbf{X}_1[:, 1]$ ;
6   Calcule SVD de  $\mathbf{X}_1$ :  $(\mathbf{U}, \Sigma, \mathbf{V}) \leftarrow \text{SVD}(\mathbf{X}_1)$ ;
7   Calcule truncamento para posto  $r$ :  $(\mathbf{U}_r, \Sigma_r, \mathbf{V}_r) \leftarrow \text{truncar}(\mathbf{U}, \Sigma, \mathbf{V})$ ;
8   Calcule operador dinâmico reduzido:  $\tilde{\mathbf{A}} \leftarrow \mathbf{U}_r^\top \mathbf{X}_2 \mathbf{V}_r \Sigma_r^{-1}$ ;
9   Calcule autovalores e autovetores de  $\tilde{\mathbf{A}}$ :  $(\lambda, \mathbf{W}) \leftarrow \text{eig}(\tilde{\mathbf{A}})$ ;
10  Calcule modos dinâmicos:  $\Phi \leftarrow \mathbf{X}_2 \mathbf{V}_r \Sigma_r^{-1} \mathbf{W}$ ;
11  Calcule amplitudes iniciais:  $\mathbf{b} \leftarrow \Phi^\dagger \mathbf{x}_0$ ;
12  Calcule frequências contínuas:  $\omega \leftarrow \ln(\lambda) / \Delta t$ ;
13  retorna  $\Phi, \lambda, \mathbf{b}, \omega$ ;
14 fim
```

---

mesma formulação modal. Para cada instante  $t_j$ , calcula-se o vetor de contribuições temporais  $\mathbf{d}(t_j) = \mathbf{b} \odot \exp(\omega t_j)$  e obtém-se o estado correspondente por  $\hat{\mathbf{x}}(t_j) = \Phi \mathbf{d}(t_j)$ .

No escopo deste trabalho, essa expressão é avaliada exclusivamente nos instantes temporais associados ao conjunto de dados utilizado no ajuste do modelo, caracterizando o regime de reconstrução *one-step-ahead*. Como a implementação envolve quantidades complexas ao longo dos cálculos, o estado final retornado é obtido tomando-se apenas a parte real da solução.

O Algoritmo 2 descreve o procedimento implementado no método `DMD::predict()`.

**Algoritmo 2:** Reconstrução com DMD (`DMD::predict()`)

---

**Entrada:**  $\Phi$ ,  $\mathbf{b}$ ,  $\omega$ ,  $\mathbf{t} = [t_1, \dots, t_p]$   
**Saída:**  $\hat{\mathbf{X}} \in \mathbb{R}^{n \times p}$

```

1 início
2   para  $j \leftarrow 1$  até  $p$  faça
3     Calcule contribuição temporal:  $\mathbf{d}_j \leftarrow \mathbf{b} \odot \exp(\omega t_j)$ ;
4     Calcule estado no instante  $t_j$ :  $\hat{\mathbf{x}}_j \leftarrow \Phi \mathbf{d}_j$ ;
5     Atribua coluna  $j$  de  $\hat{\mathbf{X}}$ :  $\hat{\mathbf{X}}[:, j] \leftarrow \hat{\mathbf{x}}_j$ ;
6   fim para
7   retorna  $\hat{\mathbf{X}}$ ;
8 fim
```

---

### 3.4.1 Definição do posto reduzido

A definição do posto reduzido  $r$  é uma etapa essencial na aplicação do DMD, pois afeta diretamente a capacidade de representar a dinâmica dominante do sistema e o custo computacional do método. Valores muito baixos de  $r$  podem levar à perda de informações relevantes, enquanto valores elevados tendem a aumentar o custo computacional sem ganhos significativos de precisão.

Nesta implementação, o posto reduzido pode ser fornecido explicitamente ou determinado de forma automática a partir do espectro de valores singulares da matriz de dados. No caso da seleção automática, utiliza-se um critério baseado em um limiar ótimo aplicado aos valores singulares, conforme descrito em (GAVISH; DONOHO, 2014).

Seja  $\mathbf{X}_1 \in \mathbb{R}^{n \times (m-1)}$  a matriz de *snapshots* defasados no tempo, cuja decomposição em valores singulares é dada por

$$\mathbf{X}_1 = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^\top, \quad (3.1)$$

com  $\mathbf{\Sigma} = \text{diag}(s_1, s_2, \dots, s_p)$  contendo os valores singulares ordenados de forma decrescente. Define-se a razão de aspecto da matriz como

$$\beta = \frac{\min(n, m-1)}{\max(n, m-1)}. \quad (3.2)$$

A partir dessa razão, é definido um limiar de truncamento

$$\tau = \omega(\beta) s_{\text{med}}, \quad (3.3)$$

onde  $s_{\text{med}}$  representa a mediana dos valores singulares e  $\omega(\beta)$  é uma função dependente apenas da razão de aspecto da matriz, conforme especificado em (GAVISH; DONOHO, 2014). O posto reduzido é então obtido como o número de valores singulares que satisfazem  $s_i > \tau$ .

O valor de  $r$  obtido por esse procedimento fornece uma estimativa do número de componentes necessárias para representar a dinâmica dominante do sistema a partir do espectro de valores singulares dos dados. Esse critério é adotado nesta implementação

como uma opção automática para a definição do posto reduzido, podendo ser utilizada conforme a configuração do método.

### 3.5 Implementação do *Online DMD*

A variante *Online DMD* foi implementada com o objetivo de permitir a atualização incremental do modelo dinâmico à medida que novos *snapshots* são observados, evitando a recomputação completa em lote. Essa abordagem é adequada a cenários em que os dados são processados sequencialmente e, em particular, a sistemas cuja dinâmica pode variar ao longo do tempo.

No presente trabalho, o método é aplicado em um subespaço de dimensão reduzida, obtido a partir de um conjunto inicial de *snapshots* (*warmup*). Esse conjunto é utilizado para construir uma base espacial  $\mathbf{U} \in \mathbb{R}^{n \times r}$  por meio de uma decomposição em valores singulares truncada, onde  $n$  representa a dimensão original do estado e  $r$  a dimensão reduzida escolhida. Cada estado  $\mathbf{x}_k \in \mathbb{R}^n$  é projetado no subespaço reduzido por

$$\mathbf{z}_k = \mathbf{U}^\top \mathbf{x}_k, \quad (3.4)$$

A dinâmica do sistema passa a ser modelada no espaço reduzido, estimando-se um operador reduzido  $\mathbf{A}_r \in \mathbb{R}^{r \times r}$  tal que

$$\mathbf{z}_{k+1} \approx \mathbf{A}_r \mathbf{z}_k. \quad (3.5)$$

A inicialização do método consiste em: (i) construir a base  $\mathbf{U}$  a partir do *warmup*; (ii) projetar pares consecutivos  $(\mathbf{x}_k, \mathbf{x}_{k+1})$  no subespaço reduzido, obtendo  $(\mathbf{z}_k, \mathbf{z}_{k+1})$ ; e (iii) inicializar o operador reduzido  $\mathbf{A}_r$  no espaço reduzido. No código, essa etapa é realizada no procedimento de inicialização a partir do *warmup*, que prepara  $\mathbf{U}$  e o estado inicial do modelo antes do processamento sequencial.

O Algoritmo 3 resume a etapa de inicialização a partir do *warmup* (implementação `OnlineDMDReduced::initialize_from_warmup()`).

Após a inicialização, o método opera de forma sequencial. A cada instante,



---

**Algoritmo 3:** Inicialização do *Online DMD* a partir do *warmup*  
(`OnlineDMDReduced::initialize_from_warmup()`)

---

**Entrada:**  $\mathbf{X}_w = [\mathbf{x}_1, \dots, \mathbf{x}_W] \in \mathbb{R}^{n \times W}$ , posto  $r$ , fator de esquecimento  $w$   
**Saída:**  $\mathbf{U} \in \mathbb{R}^{n \times r}$ , operador reduzido inicial  $\mathbf{A}_r \in \mathbb{R}^{r \times r}$

```

1 início
2   Calcule SVD do warmup:  $(\mathbf{U}_{\text{full}}, \mathbf{\Sigma}, \mathbf{V}) \leftarrow \text{SVD}(\mathbf{X}_w)$ ;
3   Calcule base truncada:  $\mathbf{U} \leftarrow \mathbf{U}_{\text{full}}[:, 1:r]$ ;
4   Calcule número de pares:  $P \leftarrow W - 1$ ;
5   para  $k \leftarrow 1$  até  $P$  faça
6     Calcule projeção do estado:  $\mathbf{z}_k \leftarrow \mathbf{U}^\top \mathbf{x}_k$ ;
7     Calcule projeção do próximo estado:  $\mathbf{z}_{k+1} \leftarrow \mathbf{U}^\top \mathbf{x}_{k+1}$ ;
8     Atribua coluna  $k$  de  $\mathbf{X}_p$ :  $\mathbf{X}_p[:, k] \leftarrow \mathbf{z}_k$ ;
9     Atribua coluna  $k$  de  $\mathbf{Y}_p$ :  $\mathbf{Y}_p[:, k] \leftarrow \mathbf{z}_{k+1}$ ;
10  fim para
11  Inicialize operador reduzido  $\mathbf{A}_r$  por mínimos quadrados com fator de
    esquecimento  $w$ ;
12 fim
```

---

realiza-se primeiramente uma predição causal de um passo à frente (*one-step-ahead*) e atualizações incrementais do operador reduzido à medida que novos dados são observados. Em seguida, após a observação do estado real subsequente, o operador reduzido  $\mathbf{A}_r$  é atualizado de forma incremental com o novo par projetado  $(\mathbf{z}_k, \mathbf{z}_{k+1})$ . Essa estratégia permite que  $\mathbf{A}_r$  se adapte progressivamente a alterações na dinâmica do sistema.

O Algoritmo 4 descreve o ciclo de predição e atualização incremental adotado.

---

**Algoritmo 4:** Predição causal e atualização incremental do *Online DMD*

---

**Entrada:** Base  $\mathbf{U}$ , operador reduzido  $\mathbf{A}_r$ , sequência  $\{\mathbf{x}_k\}_{k=1}^N$   
**Saída:** Estados previstos  $\hat{\mathbf{x}}_{k+1}$

```

1 início
2   para  $k \leftarrow 1$  até  $N - 1$  faça
3     Calcule projeção do estado atual:  $\mathbf{z}_k \leftarrow \mathbf{U}^\top \mathbf{x}_k$ ;
4     Calcule predição no espaço reduzido:  $\hat{\mathbf{z}}_{k+1} \leftarrow \mathbf{A}_r \mathbf{z}_k$ ;
5     Calcule estado previsto no espaço original:  $\hat{\mathbf{x}}_{k+1} \leftarrow \mathbf{U} \hat{\mathbf{z}}_{k+1}$ ;
6     Calcule projeção do próximo estado real:  $\mathbf{z}_{k+1} \leftarrow \mathbf{U}^\top \mathbf{x}_{k+1}$ ;
7     Atualize operador reduzido:  $\mathbf{A}_r \leftarrow \text{update}(\mathbf{A}_r, \mathbf{z}_k, \mathbf{z}_{k+1}, w)$ ;
8   fim para
9 fim
```

---

A atualização incremental do operador reduzido  $\mathbf{A}_r$  é baseada em uma regra de mínimos quadrados recursivos com fator de esquecimento  $w$ , o que enfatiza amostras recentes e torna o ajuste mais responsivo a variações temporais. No código, essa etapa corresponde ao núcleo de atualização realizado sobre os estados reduzidos, mantendo

também uma matriz auxiliar  $\mathbf{P}$  associada à forma recursiva do problema.

O Algoritmo 5 apresenta a regra de atualização recursiva utilizada no ajuste de  $\mathbf{A}_r$ .

---

**Algoritmo 5:** Atualização recursiva do operador reduzido  $\mathbf{A}_r$   
(`OnlineDMD::update()`)

---

**Entrada:**  $\mathbf{z}_k, \mathbf{z}_{k+1}$ , operador  $\mathbf{A}_r$ , matriz auxiliar  $\mathbf{P}$ , fator de esquecimento  $w$

**Saída:** Operador atualizado  $\mathbf{A}_r$

1 **início**

2     Calcule ganho escalar:  $\gamma \leftarrow (1 + \mathbf{z}_k^\top \mathbf{P} \mathbf{z}_k)^{-1}$ ;  
 3     Calcule resíduo de predição:  $\mathbf{r} \leftarrow \mathbf{z}_{k+1} - \mathbf{A}_r \mathbf{z}_k$ ;  
 4     Atualize operador reduzido:  $\mathbf{A}_r \leftarrow \mathbf{A}_r + \mathbf{r} (\gamma \mathbf{P} \mathbf{z}_k)^\top$ ;  
 5     Atualize matriz auxiliar:  $\mathbf{P} \leftarrow (\mathbf{P} - \gamma \mathbf{P} \mathbf{z}_k \mathbf{z}_k^\top \mathbf{P})/w$ ;

6 **fim**

---

É importante notar que, diferentemente do DMD clássico, o *Online DMD* implementado neste trabalho não reconstrói uma trajetória completa a partir de uma expansão modal do tipo  $\Phi \exp(\omega t)$  e amplitudes globais. Em vez disso, a reconstrução empregada é causal e de um passo à frente: em cada instante, o estado  $\mathbf{x}_{k+1}$  é aproximado diretamente pela aplicação do operador reduzido  $\mathbf{A}_r$  ao estado atual projetado, seguida da reconstrução via  $\mathbf{U}$ . Assim, a mesma rotina é utilizada para a reconstrução temporal em esquema causal *one-step-ahead*: a cada iteração, a entrada  $\mathbf{x}_k$  corresponde ao estado real observado do sistema, a partir do qual se obtém a estimativa do estado imediatamente subsequente. Essa formulação permite avaliar a capacidade do modelo em reproduzir a dinâmica local do sistema a partir de dados observados, sem recorrer a extrapolações temporais ou realimentação autoregressiva de estados previstos.

### 3.6 Implementação do *Streaming DMD*

A variante *Streaming DMD* foi implementada para processar *snapshots* de forma sequencial, mantendo um modelo dinâmico atualizado sem armazenar todas as matrizes completas do método clássico. Diferentemente do *Online DMD* utilizado neste trabalho, que opera em um subespaço fixo obtido no *warmup*, o *Streaming DMD* permite **adaptar dinamicamente** as bases reduzidas ao longo do tempo, adicionando novas direções quando

a projeção passa a representar mal os dados e realizando compressões quando a dimensão reduzida cresce além de um limite.

A implementação mantém duas bases reduzidas no espaço original,  $\mathbf{U}_x \in \mathbb{R}^{n \times r_x}$  e  $\mathbf{U}_y \in \mathbb{R}^{n \times r_y}$ , associadas, respectivamente, aos estados atuais  $\mathbf{x}_k$  e aos estados seguintes  $\mathbf{x}_{k+1}$ . Para cada par consecutivo, definem-se as projeções reduzidas

$$\tilde{\mathbf{x}}_k = \mathbf{U}_x^\top \mathbf{x}_k, \quad \tilde{\mathbf{y}}_k = \mathbf{U}_y^\top \mathbf{x}_{k+1}. \quad (3.6)$$

A dinâmica é representada por meio de matrizes de regressão reduzidas  $\mathbf{Q}$ ,  $\mathbf{P}_x$  e  $\mathbf{P}_y$ , atualizadas incrementalmente com fator de esquecimento  $\rho \in (0, 1]$ . O operador reduzido utilizado na predição é dado por

$$\tilde{\mathbf{A}} = \mathbf{Q} \mathbf{P}_x^\dagger \quad (3.7)$$

que aproxima a dinâmica entre estados consecutivos no subespaço reduzido.

A inicialização do método é realizada a partir de um conjunto inicial de *snapshots* (*warmup*), conforme descrito no Algoritmo 6.

<b>Algoritmo 6:</b>	Inicialização do <i>Streaming DMD</i>
(StreamingDMD::StreamingDMD())	
<b>Entrada:</b> Warmup $\mathbf{X}_w$ , $r_{\min}$ , $r_{\max}$ , limiar $\tau$ , meia-vida $h$	
<b>Saída:</b> $\mathbf{U}_x$ , $\mathbf{U}_y$ , $\mathbf{Q}$ , $\mathbf{P}_x$ , $\mathbf{P}_y$ , $\rho$	
<b>1 início</b>	
<b>2</b>	Calcule fator de esquecimento $\rho$ ;
<b>3</b>	Construa pares $\mathbf{X}$ e $\mathbf{Y}$ ;
<b>4</b>	Calcule bases iniciais $\mathbf{U}_x$ e $\mathbf{U}_y$ por SVD truncada;
<b>5</b>	Projete dados: $\tilde{\mathbf{X}} \leftarrow \mathbf{U}_x^\top \mathbf{X}$ , $\tilde{\mathbf{Y}} \leftarrow \mathbf{U}_y^\top \mathbf{Y}$ ;
<b>6</b>	Inicialize $\mathbf{Q}$ , $\mathbf{P}_x$ e $\mathbf{P}_y$ ;
<b>7 fim</b>	

Após a inicialização, o modelo passa a ser atualizado incrementalmente à medida que novos pares  $(\mathbf{x}_k, \mathbf{x}_{k+1})$  são observados. Nessa etapa, o método avalia a qualidade da projeção dos dados nas bases atuais, permitindo expansão adaptativa das bases e aplicação de compressão quando necessário. Para evitar crescimento descontrolado da dimensão reduzida, uma etapa de compressão é aplicada sempre que o rank ultrapassa um valor máximo permitido.

O Algoritmo 7 descreve o procedimento completo de atualização incremental.

---

<b>Algoritmo 7:</b>	Atualização incremental do <i>Streaming DMD</i>
(StreamingDMD::update())	
<hr/>	
<b>Entrada:</b>	$(\mathbf{x}_k, \mathbf{x}_{k+1}), \mathbf{U}_x, \mathbf{U}_y, \mathbf{Q}, \mathbf{P}_x, \mathbf{P}_y, \rho, \tau, r_{\max}$
<b>Saída:</b>	Bases e matrizes reduzidas atualizadas
1	<b>início</b>
2	Projete estados nas bases atuais;
3	Calcule resíduos de projeção;
4	<b>se</b> <i>erro relativo</i> $> \tau$ <b>então</b>
5	Adicione nova direção à base correspondente;
6	Atualize matrizes associadas;
7	<b>fim se</b>
8	<b>se</b> <i>rank</i> $> r_{\max}$ <b>então</b>
9	Execute compressão POD;
10	<b>fim se</b>
11	Atualize $\mathbf{Q}$ , $\mathbf{P}_x$ e $\mathbf{P}_y$ com esquecimento;
12	<b>fim</b>

---

A predição causal de um passo à frente (*one-step-ahead*) segue a mesma lógica adotada na variante anterior: o estado corrente é projetado no subespaço reduzido, a dinâmica é aplicada por meio do operador  $\tilde{\mathbf{A}}$  e o resultado é reconstruído no espaço original. O Algoritmo 8 apresenta esse procedimento.

---

<b>Algoritmo 8:</b>	Predição <i>one-step-ahead</i> no <i>Streaming DMD</i>
(StreamingDMD::predict_next())	
<hr/>	
<b>Entrada:</b>	$\mathbf{x}_k, \mathbf{U}_x, \mathbf{U}_y, \mathbf{Q}, \mathbf{P}_x$
<b>Saída:</b>	$\hat{\mathbf{x}}_{k+1}$
1	<b>início</b>
2	Calcule $\tilde{\mathbf{A}} \leftarrow \mathbf{Q} \mathbf{P}_x^\dagger$ ;
3	Projete estado: $\tilde{\mathbf{x}} \leftarrow \mathbf{U}_x^\top \mathbf{x}_k$ ;
4	Prediga no espaço reduzido: $\tilde{\mathbf{y}} \leftarrow \tilde{\mathbf{A}} \tilde{\mathbf{x}}$ ;
5	Reconstrua no espaço original: $\hat{\mathbf{x}}_{k+1} \leftarrow \mathbf{U}_y \tilde{\mathbf{y}}$ ;
6	<b>fim</b>

---

### 3.7 Métricas de avaliação

A avaliação dos métodos implementados foi realizada a partir de três critérios principais: (i) qualidade de reconstrução, (ii) tempo de processamento e (iii) uso de memória. Essas métricas permitem analisar tanto a precisão dos modelos quanto o custo computacional associado a cada abordagem.

(i) **Qualidade de reconstrução.** A qualidade dos resultados foi avaliada por meio do erro quadrático médio (*mean squared error*, MSE) e do *Peak Signal-to-Noise Ratio* (PSNR), calculados no esquema causal *one-step-ahead*. Em cada iteração, compara-se o estado real  $\mathbf{x}_{k+1}$  com a predição  $\hat{\mathbf{x}}_{k+1}$  fornecida pelo modelo, obtendo-se o vetor diferença

$$\mathbf{d} = \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}. \quad (3.8)$$

O erro quadrático médio é definido por

$$\text{MSE} = \frac{\|\mathbf{d}\|_2^2}{n}, \quad (3.9)$$

onde  $n$  representa a dimensão do estado. Essa métrica quantifica o erro médio por componente do vetor de estado e permite avaliar a evolução temporal da qualidade da reconstrução.

Como complemento à análise baseada em MSE, a qualidade visual das reconstruções foi avaliada por meio do *Peak Signal-to-Noise Ratio*, amplamente utilizado em comparação de imagens. O PSNR é definido como

$$\text{PSNR} = 10 \log_{10} \left( \frac{I_{\max}^2}{\text{MSE}} \right), \quad (3.10)$$

onde  $I_{\max}$  representa o valor máximo possível de intensidade do sinal da imagem. Valores mais elevados de PSNR indicam maior similaridade entre a imagem reconstruída e a imagem original no mesmo instante temporal.

Nos experimentos realizados, o MSE foi analisado ao longo do tempo para avaliar o comportamento dinâmico do erro, enquanto o PSNR foi empregado como métrica pontual para quantificar a qualidade visual das reconstruções em instantes específicos.

(ii) **Tempo de processamento.** O tempo total de execução de cada algoritmo foi medido utilizando a biblioteca `std::chrono` da linguagem C++. O instante inicial é registrado imediatamente antes da execução do método avaliado e o instante final é obtido após o término completo do processamento. A diferença entre esses dois instantes fornece o tempo total de execução, expresso em segundos.

Essa medida engloba todas as etapas realizadas pelo algoritmo correspondente, incluindo inicialização, ajuste do modelo, atualizações incrementais, quando aplicáveis, e etapa de reconstrução, permitindo uma comparação direta do custo computacional global entre os métodos.

**(iii) Uso de memória.** O uso de memória foi avaliado a partir do pico de memória residente do processo durante a execução do algoritmo. Para isso, utiliza-se uma chamada ao sistema operacional via `getrusage`, que fornece estatísticas de uso de recursos do processo corrente.

O valor considerado corresponde ao máximo de memória residente observado ao longo da execução. Essa métrica fornece uma estimativa do impacto de cada método em termos de consumo de memória, permitindo comparar abordagens em lote, online e streaming sob a perspectiva de eficiência de recursos.

## 4 Resultados e Discussões

### 4.1 Estudo de Caso: Corrente de Sedimentos

Os experimentos foram conduzidos a partir de *snapshots* bidimensionais em tons de cinza, provenientes de uma simulação de escoamento com transporte de sedimentos. A utilização de imagens em escala de cinza permite representar o campo de concentração por um único valor escalar por pixel, reduzindo a faixa dinâmica dos dados e simplificando o tratamento numérico, o que contribui para a eficiência computacional dos métodos aplicados.

Cada imagem representa um campo escalar associado à concentração de sedimentos no domínio em um instante discreto. Para a aplicação dos métodos baseados em DMD, cada imagem foi vetorizada e organizada como uma coluna da matriz de dados  $\mathbf{X} = [\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{N-1}]$ .

Nesta seção, apresentam-se os resultados obtidos ao aplicar o DMD clássico, o *Online DMD* e o *Streaming DMD* sobre diferentes janelas temporais da simulação, mantendo-se uma comparação justa ao utilizar, em cada experimento, o mesmo conjunto de dados de entrada para todos os métodos. As métricas de avaliação seguem a definição apresentada anteriormente neste trabalho.

A Figura 4.1 apresenta *snapshots* reais do campo de concentração de sedimentos em instantes representativos da simulação, com o objetivo de contextualizar visualmente a dinâmica espacial e temporal do sistema que se busca modelar nos experimentos subsequentes.

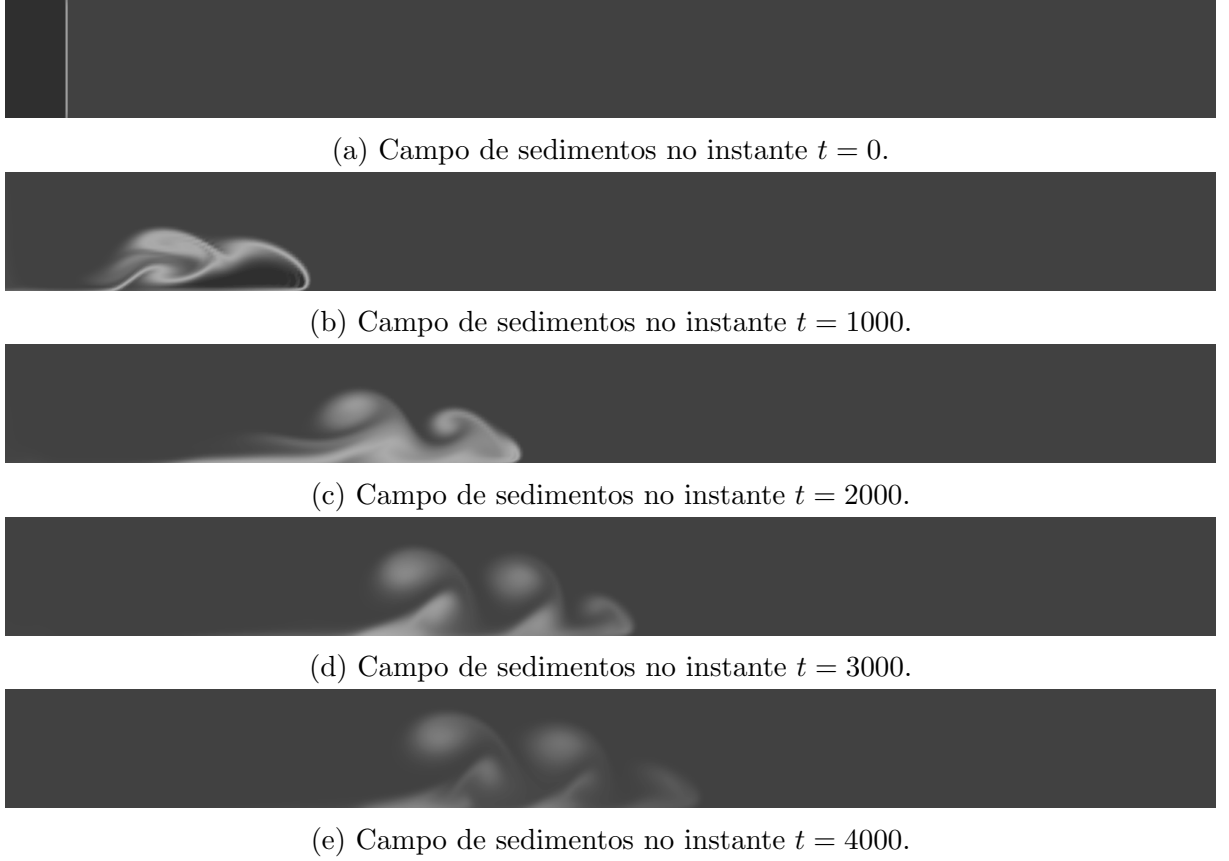


Figura 4.1: Evolução temporal do campo de concentração de sedimentos em instantes representativos da simulação.

## Configuração dos Experimentos

Foram considerados diferentes intervalos temporais da simulação, todos com passo de tempo constante  $\Delta t = 1$ . As janelas foram definidas de modo a conter conjuntos de 500 *snapshots*, permitindo avaliar o desempenho dos métodos em diferentes faixas temporais do escoamento e verificar sua capacidade de reconstrução sob dinâmicas possivelmente distintas.

A escolha de intervalos com 500 imagens também está associada a considerações de custo computacional e viabilidade prática dos experimentos. O processamento e armazenamento de grandes quantidades de *snapshots* bidimensionais impõem demandas significativas de tempo e memória, de modo que a adoção de janelas de tamanho moderado permite a realização de múltiplos experimentos comparativos de forma consistente, sem comprometer a estabilidade da execução no ambiente computacional disponível.

A Tabela 4.1 resume as janelas avaliadas e o número de *snapshots* utilizado em cada experimento.



Tabela 4.1: Janelas temporais avaliadas nos experimentos com sedimentos.

Experimento	Intervalo	$N$ ( <i>snapshots</i> )
E1	1000, 1500	500
E2	2000, 2500	500
E3	3000, 3500	500

Conforme ilustrado anteriormente na Figura 4.1, a simulação apresenta mudanças visuais relevantes ao longo do tempo. Assim, a escolha de múltiplas janelas visa testar os métodos sob condições dinâmicas distintas, enquanto o intervalo estendido (E4) permite observar a evolução do erro e a robustez numérica em horizontes maiores.

No caso do *Online DMD*, que requer um período inicial para a construção da base espacial reduzida, foi adotada uma estratégia uniforme em todos os experimentos: o tamanho da janela de *warmup* foi definido como metade do número total de *snapshots* do intervalo analisado. Assim, para os experimentos com 500 imagens, foram utilizados 250 *snapshots* iniciais exclusivamente para a construção da base  $\mathbf{U}$ , sendo toda a sequência temporal posteriormente processada pelo método ao longo da etapa de reconstrução.

Essa escolha permite um compromisso entre a qualidade inicial da base reduzida e a quantidade de dados disponíveis para avaliação do comportamento online do método, mantendo consistência entre os diferentes intervalos analisados.

## Definição do Posto Reduzido

Para garantir uma comparação justa entre os métodos avaliados, o posto reduzido utilizado nos experimentos foi definido de forma consistente para todas as abordagens. Em cada intervalo temporal considerado, o posto foi inicialmente estimado a partir do DMD clássico, por meio da análise do espectro de valores singulares da matriz de snapshots correspondente.

Esse procedimento foi realizado apenas uma vez para cada intervalo, com o objetivo exclusivo de determinar o valor de posto representativo da dinâmica dominante naquela janela temporal. Após essa etapa preliminar, o valor obtido foi fixado e fornecido manualmente como parâmetro de entrada para todas as execuções subsequentes, incluindo o próprio DMD clássico. Dessa forma, evita-se que o custo computacional associado ao cálculo automático do posto influencie as métricas de tempo e memória analisadas nos

resultados.

O mesmo valor de posto foi empregado no DMD clássico e no *Online DMD*. No caso do *Streaming DMD*, devido à presença de mecanismos adaptativos de expansão e compressão da base reduzida, foi definido um intervalo admissível em torno do posto estimado. Especificamente, para um posto médio  $r$ , foram adotados os limites  $r_{\min} = r - 10$  e  $r_{\max} = r + 10$ , permitindo variações controladas na dimensão reduzida sem comprometer a comparabilidade com os demais métodos.

Como o posto estimado varia de acordo com o intervalo temporal considerado, valores distintos foram obtidos para cada experimento. Esses valores são explicitamente indicados na apresentação dos resultados correspondentes a cada janela temporal.

## 4.2 Resultados nas Simulações com Sedimentos

### 4.2.1 Experimento E1: Imagens de 1000 a 1500

Neste experimento, os métodos foram aplicados ao intervalo temporal compreendido entre as imagens de índice 1000 e 1500 da simulação, totalizando 500 *snapshots*. Esse intervalo foi adotado como um caso base de referência, pois apresenta uma extensão temporal adequada para avaliar simultaneamente a qualidade de reconstrução *one-step-ahead* e o comportamento temporal do erro, sem os efeitos associados a janelas muito curtas ou excessivamente longas.

Foi adotado um mesmo posto reduzido  $r = 116$  para o DMD clássico e para o *Online DMD*. Esse valor foi determinado a partir da análise do espectro singular do DMD clássico aplicada a esse intervalo, utilizada exclusivamente para a estimativa do posto reduzido, e posteriormente fixado manualmente nos experimentos, de modo a evitar que o custo computacional da seleção de posto influenciasse as comparações de desempenho. No caso do *Streaming DMD*, foi adotado um intervalo adaptativo de postos, com  $r_{\min} = 106$  e  $r_{\max} = 126$ , centrado no valor médio utilizado pelos demais métodos.

A Figura 4.2 apresenta uma comparação visual entre o campo de sedimentos original e as reconstruções obtidas pelos diferentes métodos no instante  $t = 1250$ . São mostrados, respectivamente: (a) o *snapshot* original da simulação, (b) a reconstrução

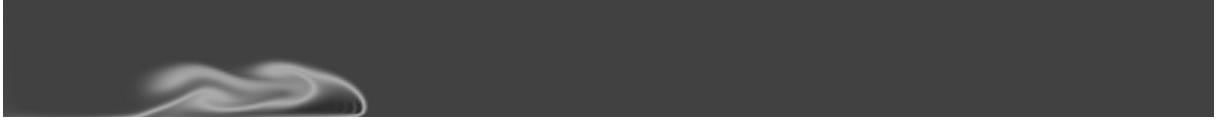
obtida pelo DMD clássico, (c) a reconstrução produzida pelo *Online DMD* e (d) a reconstrução obtida pelo *Streaming DMD*, todas avaliadas no mesmo instante temporal.



(a) Campo de sedimentos original no instante  $t = 1250$ .



(b) Reconstrução pelo DMD clássico no instante  $t = 1250$ .



(c) Reconstrução pelo *Online DMD* no instante  $t = 1250$ .



(d) Reconstrução pelo *Streaming DMD* no instante  $t = 1250$ .

Figura 4.2: Comparação entre o campo de sedimentos original e as reconstruções obtidas pelos diferentes métodos no instante  $t = 1250$  (Experimento E1).

Além da análise visual, foi avaliado o comportamento temporal do erro por meio do erro quadrático médio. As Figuras 4.3, 4.4 e 4.5 apresentam a evolução do MSE ao longo do tempo para o DMD clássico, o *Online DMD* e o *Streaming DMD*, respectivamente, ao longo do intervalo considerado.

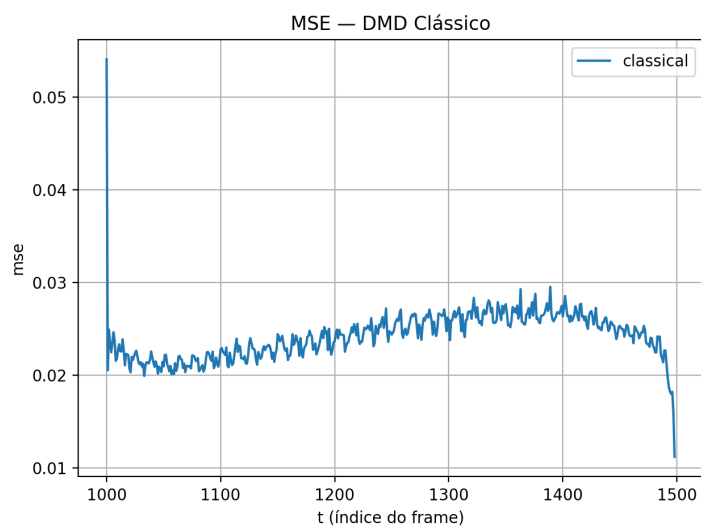


Figura 4.3: Evolução do erro quadrático médio ao longo do tempo para o DMD clássico no Experimento E1.

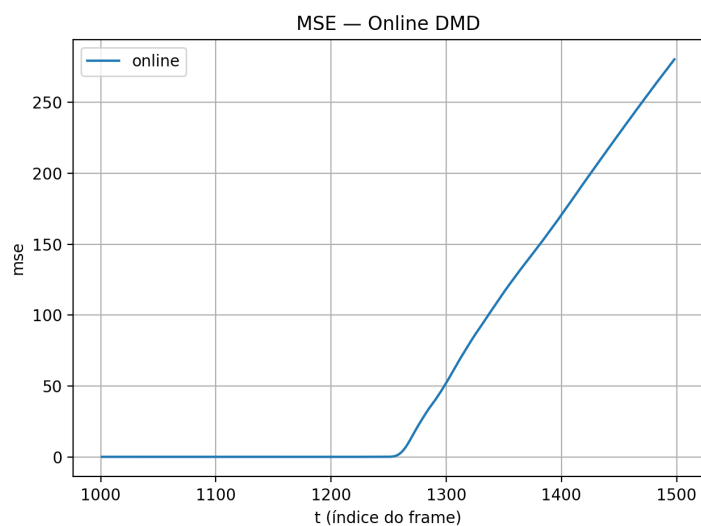


Figura 4.4: Evolução do erro quadrático médio ao longo do tempo para o *Online DMD* no Experimento E1.

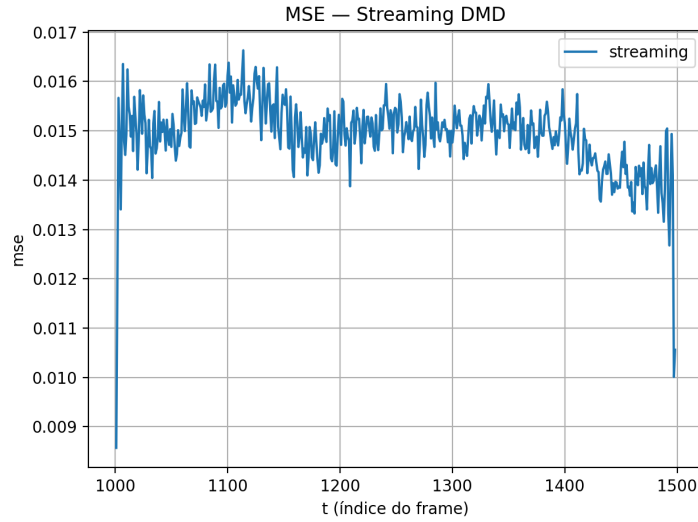


Figura 4.5: Evolução do erro quadrático médio ao longo do tempo para o *Streaming DMD* no Experimento E1.

Complementarmente, a qualidade visual das reconstruções foi quantificada por meio do *Peak Signal-to-Noise Ratio*, calculado para o instante  $t = 1250$ , correspondente às reconstruções apresentadas na Figura 4.2.

Tabela 4.2: Valores de PSNR (em dB) das reconstruções no instante  $t = 1250$  no Experimento E1.

Método	PSNR (dB)
DMD clássico	63.678923
<i>Online DMD</i>	58.353677
<i>Streaming DMD</i>	65.823686

Os resultados apresentados nesta subseção fornecem uma visão inicial do desempenho dos métodos na reconstrução *one-step-ahead* do campo de sedimentos em um intervalo temporal representativo da simulação. A comparação entre os métodos ao longo de diferentes janelas temporais é aprofundada nas subseções seguintes.

#### 4.2.2 Experimento E2: Imagens de 2000 a 2500

Neste experimento, os métodos foram aplicados ao intervalo temporal compreendido entre as imagens de índice 2000 e 2500 da simulação, totalizando 500 *snapshots*. Assim como no Experimento E1, essa janela temporal foi utilizada para avaliar o desempenho dos métodos na reconstrução e na evolução temporal do erro, agora em uma faixa distinta da simulação.

O posto reduzido adotado neste experimento foi  $r = 147$  para o DMD clássico e o *Online DMD*. Para o *Streaming DMD*, utilizou-se um intervalo adaptativo de postos, com  $r_{\min} = 137$  e  $r_{\max} = 157$ .

A Figura 4.6 apresenta uma comparação visual entre o campo de sedimentos original e as reconstruções obtidas pelos diferentes métodos no instante  $t = 2250$ . São mostrados, respectivamente: (a) o *snapshot* original da simulação, (b) a reconstrução obtida pelo DMD clássico, (c) a reconstrução produzida pelo *Online DMD* e (d) a reconstrução obtida pelo *Streaming DMD*, todas avaliadas no mesmo instante temporal.



(a) Campo de sedimentos original no instante  $t = 2250$ .



(b) Reconstrução pelo DMD clássico no instante  $t = 2250$ .



(c) Reconstrução pelo *Online DMD* no instante  $t = 2250$ .



(d) Reconstrução pelo *Streaming DMD* no instante  $t = 2250$ .

Figura 4.6: Comparação entre o campo de sedimentos original e as reconstruções obtidas pelos diferentes métodos no instante  $t = 2250$  (Experimento E2).

De forma análoga ao experimento anterior, foi avaliado o comportamento temporal do erro por meio do erro quadrático médio (MSE). As Figuras 4.7, 4.8 e 4.9 apresentam a evolução do MSE ao longo do tempo para os diferentes métodos no intervalo considerado.

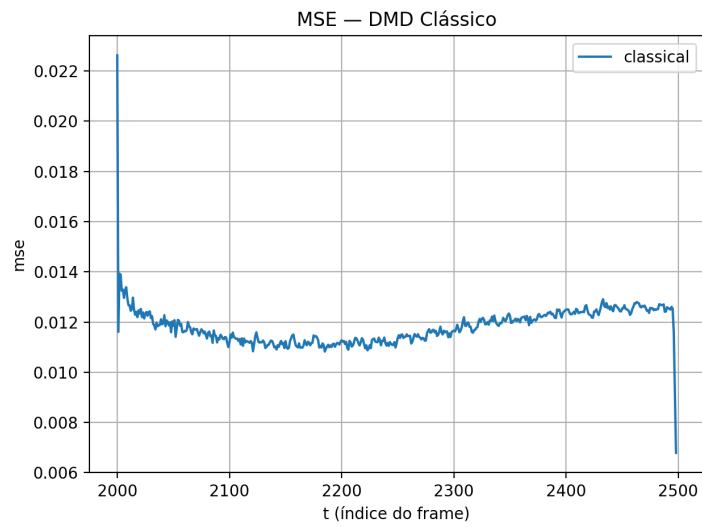


Figura 4.7: Evolução do erro quadrático médio ao longo do tempo para o DMD clássico no Experimento E2.

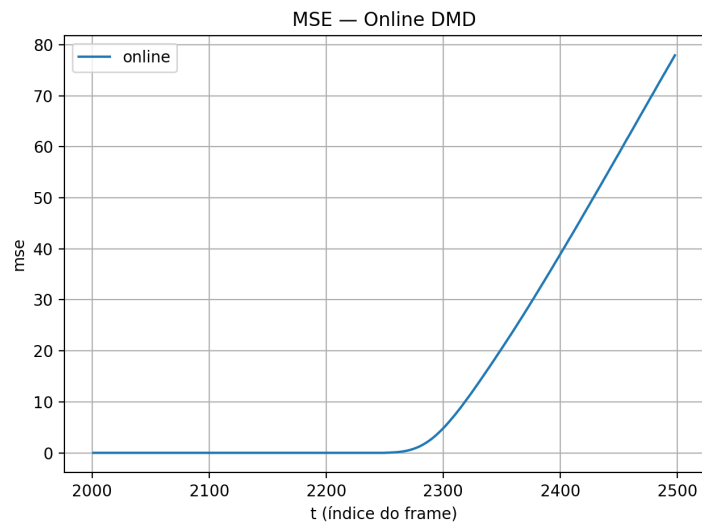


Figura 4.8: Evolução do erro quadrático médio ao longo do tempo para o *Online DMD* no Experimento E2.

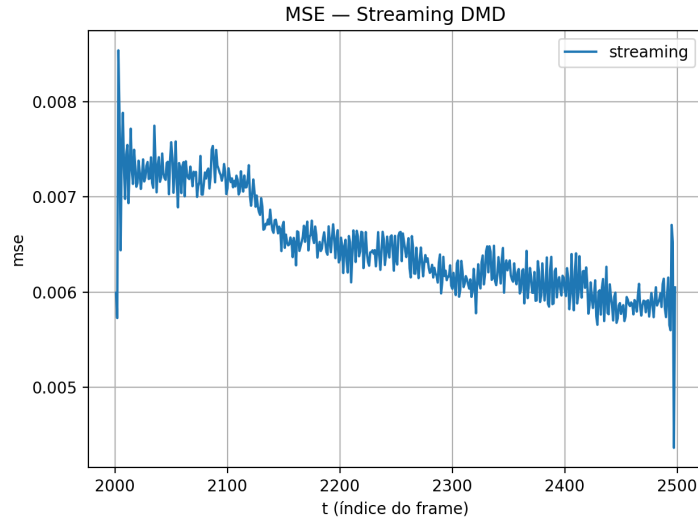


Figura 4.9: Evolução do erro quadrático médio ao longo do tempo para o *Streaming DMD* no Experimento E2.

A Tabela 4.3 apresenta os valores de PSNR calculados para o instante  $t = 2250$ , correspondente às reconstruções exibidas na Figura 4.6.

Tabela 4.3: Valores de PSNR (em dB) das reconstruções no instante  $t = 2250$  no Experimento E2.

Método	PSNR (dB)
DMD clássico	67.422187
<i>Online DMD</i>	63.604047
<i>Streaming DMD</i>	70.620878

Os resultados deste experimento reforçam as observações obtidas no Experimento E1, permitindo avaliar a consistência do desempenho dos métodos em uma faixa temporal distinta da simulação.

### 4.2.3 Experimento E3: Imagens de 3000 a 3500

Neste experimento, os métodos foram aplicados ao intervalo temporal compreendido entre as imagens de índice 3000 e 3500 da simulação, totalizando 500 *snapshots*. Essa janela temporal corresponde a uma fase posterior da simulação, permitindo avaliar a consistência do desempenho dos métodos em uma faixa distinta daquela analisada nos experimentos anteriores.

O posto reduzido adotado neste experimento foi  $r = 156$  para o DMD clássico e o *Online DMD*. Para o *Streaming DMD*, utilizou-se um intervalo adaptativo de postos,

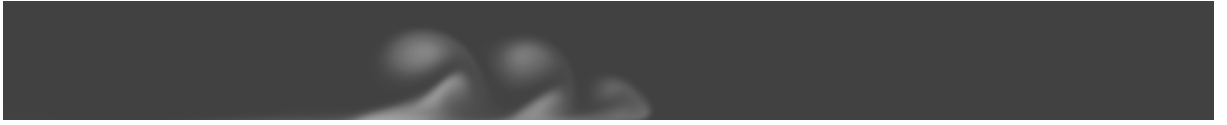


com  $r_{\min} = 146$  e  $r_{\max} = 166$ .

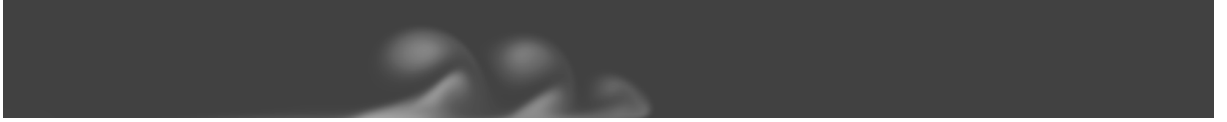
A Figura 4.10 apresenta uma comparação visual entre o campo de sedimentos original e as reconstruções obtidas pelos diferentes métodos no instante  $t = 3250$ . São mostrados, respectivamente: (a) o *snapshot* original da simulação, (b) a reconstrução obtida pelo DMD clássico, (c) a reconstrução produzida pelo *Online DMD* e (d) a reconstrução obtida pelo *Streaming DMD*, todas avaliadas no mesmo instante temporal.



(a) Campo de sedimentos original no instante  $t = 3250$ .



(b) Reconstrução pelo DMD clássico no instante  $t = 3250$ .



(c) Reconstrução pelo *Online DMD* no instante  $t = 3250$ .



(d) Reconstrução pelo *Streaming DMD* no instante  $t = 3250$ .

Figura 4.10: Comparação entre o campo de sedimentos original e as reconstruções obtidas pelos diferentes métodos no instante  $t = 3250$  (Experimento E3).

O comportamento temporal do erro foi avaliado por meio do erro quadrático médio (MSE). As Figuras 4.11, 4.12 e 4.13 apresentam a evolução do MSE ao longo do tempo para os diferentes métodos no intervalo considerado.

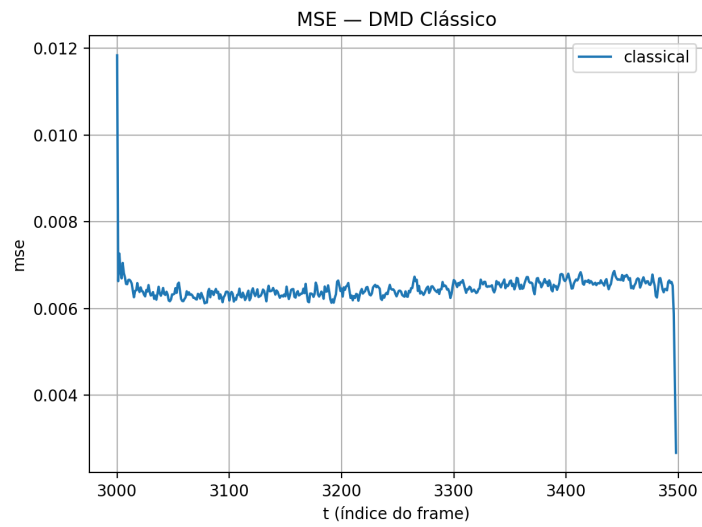


Figura 4.11: Evolução do erro quadrático médio ao longo do tempo para o DMD clássico no Experimento E3.

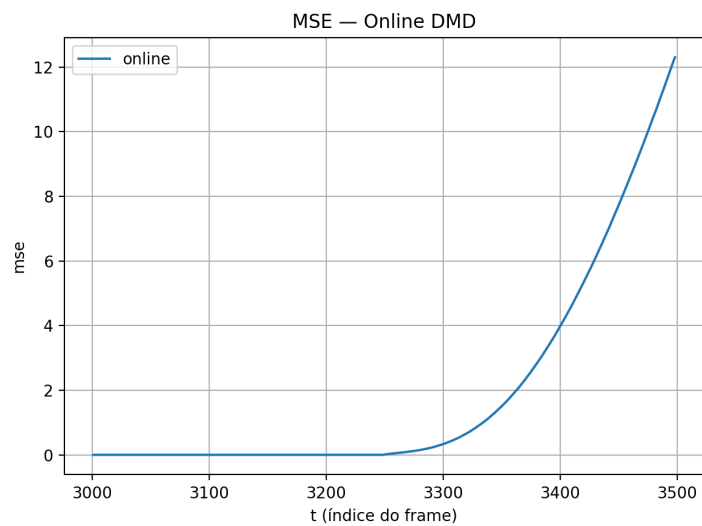


Figura 4.12: Evolução do erro quadrático médio ao longo do tempo para o *Online DMD* no Experimento E3.

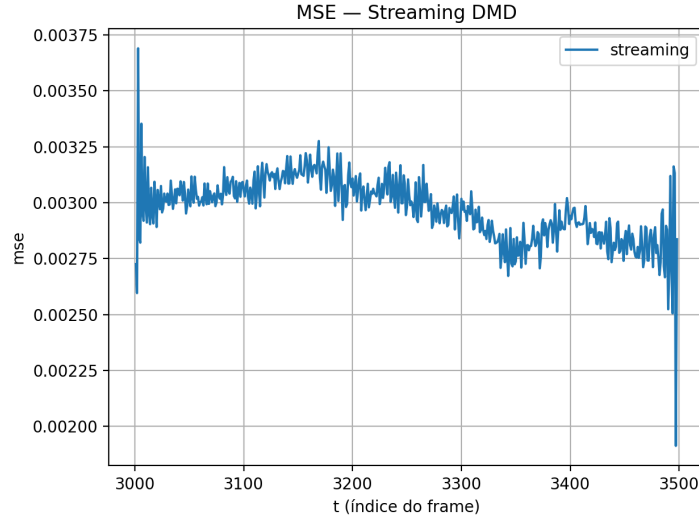


Figura 4.13: Evolução do erro quadrático médio ao longo do tempo para o *Streaming DMD* no Experimento E3.

A Tabela 4.4 apresenta os valores de PSNR calculados para o instante  $t = 3250$ , correspondente às reconstruções exibidas na Figura 4.10.

Tabela 4.4: Valores de PSNR (em dB) das reconstruções no instante  $t = 3250$  no Experimento E3.

Método	PSNR (dB)
DMD clássico	69.665698
<i>Online DMD</i>	66.815148
<i>Streaming DMD</i>	74.114566

Os resultados deste experimento permitem verificar a manutenção do desempenho dos métodos em uma fase mais avançada da simulação, complementando as análises realizadas nos experimentos anteriores.

### 4.3 Comparação entre os Métodos

Nesta seção é apresentada uma comparação integrada entre o DMD clássico, o *Online DMD* e o *Streaming DMD*, com base nos resultados obtidos nos experimentos descritos anteriormente. A análise considera simultaneamente a evolução do erro de reconstrução, a estabilidade temporal dos métodos e o custo computacional associado, mantendo condições equivalentes de dados e parâmetros sempre que possível.

A comparação do desempenho em termos de precisão é baseada no erro quadrático

médio calculado ao longo do tempo. Conforme observado nos experimentos individuais, o *Online DMD* apresenta um comportamento característico: enquanto a dinâmica do sistema permanece bem representada pela base espacial construída no período de *warmup*, o erro de reconstrução mantém-se controlado e competitivo. No entanto, à medida que a dinâmica evolui para regiões menos bem representadas por essa base fixa, observa-se um crescimento acentuado do erro, evidenciando a limitação do método em horizontes temporais mais longos.

Com o objetivo de evidenciar o desempenho dos métodos em um regime comparável, a Figura 4.14 apresenta a comparação do MSE em um intervalo temporal recortado, anterior ao crescimento abrupto do erro do *Online DMD*. Nesse intervalo, verifica-se que o *Online DMD* apresenta os menores valores de MSE entre os métodos avaliados, indicando excelente capacidade de reconstrução enquanto a base reduzida permanece adequada. O *Streaming DMD* apresenta comportamento estável ao longo do tempo, beneficiando-se do mecanismo de atualização adaptativa da base espacial, enquanto o DMD clássico mantém desempenho consistente, conforme esperado para um método offline ajustado a um conjunto fixo de dados.

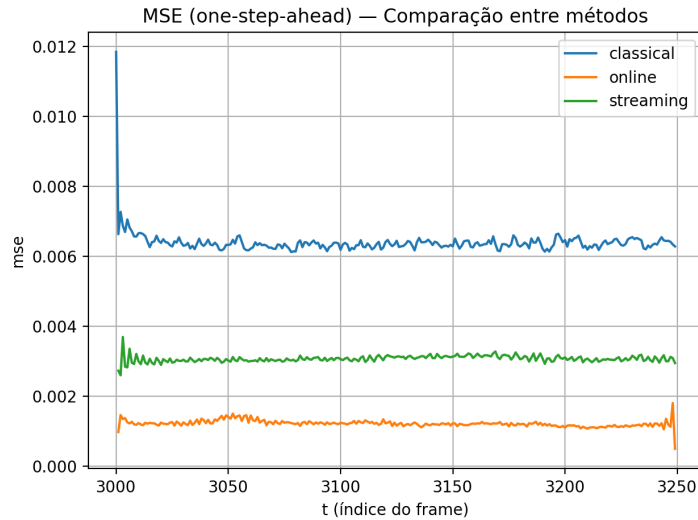


Figura 4.14: Comparação do erro quadrático médio entre os métodos em um intervalo temporal recortado, anterior ao crescimento acentuado do erro do *Online DMD*.

Além da precisão, foram analisados o tempo total de execução e o pico de memória consumida pelos métodos em cada experimento. Essas métricas fornecem uma visão complementar do custo associado a cada abordagem, o que é essencial em aplicações de

grande escala ou com restrições de recursos computacionais.

Os valores apresentados nas tabelas de custo computacional correspondem à média obtida a partir de múltiplas execuções independentes de cada método. Para cada experimento, cada algoritmo foi executado dez vezes sob as mesmas condições, e os tempos de execução e picos de memória foram posteriormente agregados por meio do valor médio, de modo a reduzir a influência de variações pontuais associadas ao sistema operacional e à execução dos algoritmos.

A Tabela 4.5 apresenta o tempo total de execução dos métodos nos diferentes experimentos, enquanto a Tabela 4.6 resume os valores de pico de memória observados durante as execuções.

Tabela 4.5: Tempo total de execução dos métodos nos experimentos realizados.

Experimento	DMD clássico (s)	<i>Online DMD</i> (s)	<i>Streaming DMD</i> (s)
E1	371	57	169
E2	472	52	166
E3	566	56	185

Tabela 4.6: Pico de memória RAM consumida pelos métodos nos experimentos realizados.

Experimento	DMD clássico (MB)	<i>Online DMD</i> (MB)	<i>Streaming DMD</i> (MB)
E1	6294	3184	7219
E2	6539	3184	7135
E3	6718	3192	7251

De forma geral, os resultados evidenciam diferenças claras entre as abordagens. O DMD clássico apresenta desempenho estável em termos de erro, ao custo de maior consumo de memória e natureza offline. O *Online DMD* demonstra excelente desempenho enquanto opera dentro do regime representado pelo *warmup*, mas perde precisão à medida que a dinâmica se afasta dessa base fixa. O *Streaming DMD*, por sua vez, mostra maior robustez temporal, mantendo o erro controlado ao longo de janelas mais extensas devido à adaptação contínua da base espacial.

## 5 Conclusões

Neste trabalho, investigou-se a aplicação de modelos de ordem reduzida baseados na Decomposição em Modos Dinâmicos para a modelagem e reconstrução da dinâmica temporal de um escoamento com transporte de sedimentos. O objetivo principal consistiu em avaliar o desempenho do DMD clássico e de duas variantes sequenciais em um cenário caracterizado por dados de alta dimensionalidade e evolução temporal complexa.

Os três métodos foram implementados em C++, com foco na eficiência computacional e no tratamento consistente de dados sequenciais. Em seguida, foram aplicados a diferentes janelas temporais de uma simulação de escoamento com transporte de sedimentos, utilizando *snapshots* bidimensionais em escala de cinza. A comparação foi conduzida de forma sistemática, adotando postos reduzidos compatíveis entre os métodos e avaliando os resultados por meio de métricas quantitativas (MSE e PSNR), análise visual das reconstruções e medidas de custo computacional.

Os resultados obtidos evidenciaram que o DMD clássico é capaz de produzir reconstruções de alta qualidade, porém apresenta elevado custo computacional e elevado consumo de memória, decorrentes de sua formulação offline e da necessidade de acesso completo ao conjunto de dados. Esses aspectos limitam sua aplicabilidade em cenários sequenciais ou de larga escala, como o considerado neste trabalho.

O *Online DMD* destacou-se como a abordagem mais eficiente do ponto de vista computacional, apresentando os menores tempos de execução e menor consumo de memória em todos os experimentos realizados. Conforme observado nas análises de erro, o método apresenta excelente desempenho durante o regime associado à base construída no período de *warmup*, com valores de MSE reduzidos e PSNR elevados, chegando a superar os demais métodos em determinados intervalos temporais. Entretanto, à medida que a dinâmica do sistema evolui para além do subespaço inicialmente capturado, observa-se um crescimento acentuado do erro, refletindo a limitação imposta pela base espacial fixa adotada pelo método.

Por sua vez, o *Streaming DMD* apresentou comportamento mais robusto ao longo

do tempo, mantendo níveis de erro mais estáveis em janelas temporais prolongadas. Essa robustez está associada aos mecanismos adaptativos de atualização e compressão da base espacial, que permitem ao método acomodar mudanças graduais na dinâmica do sistema. Em contrapartida, tal flexibilidade resulta em maior custo computacional e maior consumo de memória, como evidenciado pelos tempos de execução e picos de memória observados nos experimentos.

A análise comparativa de custo computacional confirmou essas tendências: o DMD clássico apresentou os maiores tempos de execução e elevado uso de memória, o *Streaming DMD* ocupou uma posição intermediária, enquanto o *Online DMD* mostrou-se a alternativa mais eficiente, desde que aplicado em um horizonte temporal compatível com a base reduzida inicialmente construída. Esses resultados reforçam a importância da escolha do método em função do compromisso entre precisão, custo computacional e horizonte temporal de interesse.

Como limitações deste trabalho, destaca-se a restrição no tamanho das janelas temporais avaliadas, imposta pelo custo computacional associado ao processamento e armazenamento de grandes volumes de imagens. Como perspectivas futuras, destaca-se a investigação de critérios mais robustos para a escolha do tamanho do *warmup* no *Online DMD*, bem como o estudo do impacto dessa escolha na estabilidade e na precisão das reconstruções ao longo do tempo. Além disso, a aplicação das abordagens estudadas a outros cenários de escoamento multifásico, bem como a incorporação de informações físicas adicionais aos modelos de ordem reduzida, configuram direções promissoras para trabalhos futuros.

De forma geral, os resultados obtidos demonstram que as variantes sequenciais do DMD constituem ferramentas promissoras para a análise de escoamentos com transporte de sedimentos, desde que suas limitações e características algorítmicas sejam adequadamente consideradas no contexto da aplicação.

## Bibliografia

- BAO, A. et al. Data-driven model reduction for coupled flow and geomechanics based on DMD methods. v. 4, n. 3, p. 138, 2019. ISSN 2311-5521. Disponível em: <https://www.mdpi.com/2311-5521/4/3/138>.
- BEIT-SADI, M.; KROL, J.; WYNN, A. Data-driven feature identification and sparse representation of turbulent flows. v. 88, p. 108766, 2021. ISSN 0142727X. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0142727X2031095X>.
- BENNER, P.; GUGERCIN, S.; WILLCOX, K. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, v. 57, n. 4, p. 483–531, 2015.
- BERKOOZ, G.; HOLMES, P.; LUMLEY, J. L. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, v. 25, p. 539–575, 1993.
- CALMET, H. et al. Dynamic mode decomposition analysis of high-fidelity CFD simulations of the sinus ventilation. v. 105, n. 3, p. 699–713, 2020. ISSN 1386-6184, 1573-1987. Disponível em: <https://link.springer.com/10.1007/s10494-020-00156-8>.
- GAVISH, M.; DONOHO, D. L. *The Optimal Hard Threshold for Singular Values is  $4/\sqrt{3}$* . 2014. Disponível em: <https://arxiv.org/abs/1305.5870>.
- GIORGI, M. G. D.; FICARELLA, A.; FONTANAROSA, D. Mode decomposition methods for the analysis of cavitating flows in turbomachinery. v. 148, p. 924–931, 2018. ISSN 18766102. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S1876610218303862>.
- GRAVE, M.; CAMATA, J. J.; COUTINHO, A. L. Residual-based variational multiscale 2d simulation of sediment transport with morphological changes. *Computers Fluids*, 2020.
- HEMATI, M. S.; WILLIAMS, M. O.; ROWLEY, C. W. Dynamic mode decomposition for large and streaming datasets. *Physics of Fluids*, AIP Publishing, v. 26, n. 11, 2014.
- HIJAZI, S. et al. Data-driven pod–galerkin reduced order model for turbulent flows. *Journal of Computational Physics*, v. 416, p. 109513, 2020.
- LI, S.; DUAN, G.; SAKAI, M. On reduced-order modeling of gas–solid flows using deep learning. *Physics of Fluids*, v. 36, n. 3, p. 033340, 2024.
- LIBERO, G. et al. Extended dynamic mode decomposition for model reduction in fluid dynamics simulations. v. 36, n. 6, p. 061911, 2024. ISSN 1070-6631, 1089-7666. Disponível em: <https://pubs.aip.org/pof/article/36/6/061911/3298528/Extended-dynamic-mode-decomposition-for-model>.
- NADERI, M. H.; EIVAZI, H.; ESFAHANIAN, V. New method for dynamic mode decomposition of flows over moving structures based on machine learning (hybrid dynamic mode decomposition). v. 31, n. 12, p. 127102, 2019. ISSN 1070-6631, 1089-7666. Disponível em: <https://pubs.aip.org/pof/article/31/12/127102/1075985/New-method-for-dynamic-mode-decomposition-of-flows>.



- QIN, W. et al. Investigation of in-cylinder engine flow quadruple decomposition dynamical behavior using proper orthogonal decomposition and dynamic mode decomposition methods. v. 141, n. 8, p. 081004, 2019. ISSN 0742-4795, 1528-8919. Disponível em: <https://asmedigitalcollection.asme.org/gasturbinespower/article/doi/10.1115/1.4042725/374477/Investigation-of-InCylinder-Engine-Flow-Quadruple>.
- ROWLEY, C. W.; COLONIUS, T.; MURRAY, R. M. Model reduction for compressible flows using pod and galerkin projection. *Physica D: Nonlinear Phenomena*, v. 189, n. 1–2, p. 115–129, 2004.
- ROWLEY, C. W.; DAWSON, S. T. M. Model reduction for flow analysis and control. *Annual Review of Fluid Mechanics*, v. 49, n. 1, p. 387–417, 2017.
- SCHMID, P. J. Dynamic mode decomposition and its variants. v. 54, n. 1, p. 225–254, 2022. ISSN 0066-4189, 1545-4479. Disponível em: <https://www.annualreviews.org/doi/10.1146/annurev-fluid-030121-015835>.
- TAIRA, K. et al. Modal analysis of fluid flows: An overview. *AIAA Journal*, v. 55, n. 12, p. 4013–4041, 2017.
- ZHANG, H. et al. Online dynamic mode decomposition for time-varying systems. *SIAM Journal on Applied Dynamical Systems*, SIAM, v. 18, n. 3, p. 1586–1609, 2019.
- ZHANG, T. et al. A new hybrid reduced order modeling for parametrized navier–stokes equations in stream-vorticity formulation. *Physics of Fluids*, 2024. Disponível em: <https://api.semanticscholar.org/CorpusID:270411030>.
- ZHANG, Z. et al. Physics-informed dynamic mode decomposition for reconstruction and prediction of dense particulate pipe flows. v. 36, n. 11, p. 113374, 2024. ISSN 1070-6631, 1089-7666. Disponível em: <https://pubs.aip.org/pof/article/36/11/113374/3322479/Physics-informed-dynamic-mode-decomposition-for>.