DETECTION AND PREVENTION OF NODE OVERLAP: LIST-BASED APPROACHES FOR HIGH-DIMENSIONAL ENVIRONMENTS

REPORT

Submitted by

BL.SC.P2DSC23014 K THEJAA

in partial fulfillment for the award of the degree of

MASTER OF TECHNOLOGY

IN

DATA SCIENCE

AMRITA SCHOOL OF ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

BENGALURU 560 035

DECEMBER 2023

Annexure 1

DETECTION AND PREVENTION OF NODE OVERLAP: LIST-BASED APPROACHES FOR HIGH-DIMENSIONAL ENVIRONMENTS

A PROJECT REPORT

Submitted by

K THEJAA

in partial fulfillment for the award of the degree of

COMPUTER SCIENCE AND ENGINEERING

AMRITA SCHOOL OF ENGINEERING,

AMRITA VISHWA VIDYAPEETHAM

BENGALURU 560 035

DECEMBER 2023

December 2023

AMRITA VISHWA VIDYAPEETHAM
AMRITA SCHOOL OF ENGINEERING, BENGALURU

BONAFIDE CERTIFICATE

This is to certify that the Project Report entitled **“DETECTION AND PREVENTION OF NODE OVERLAP: LIST-BASED APPROACHES FOR HIGH-DIMENSIONAL ENVIRONMENTS”**

submitted by K THEJAA BL.SC.P2DSC23014 in partial fulfillment of the requirements for the award of the **Degree Master of Technology** in “**DATA SCIENCE**” is a bonafide record of the work carried out under my guidance and supervision at Amrita School of Engineering, Bengaluru.

SIGNATURE

Dr. Supriya M
Associate Professor
Dept. of CSE, ASE, Bangalore

SIGNATURE

Dr. Sriram Devanathan
Professor&Chairperson
Dept. of CSE, ASE, Bangalore

This project report was evaluated by us on ……………….

EXAMINER I

EXAMINER II

December 2023

ACKNOWLEDGEMENTS

The satisfaction that accompanies successful completion of any task would be incomplete without mention of people who made it possible, and whose constant encouragement and guidance have been source of inspiration throughout the course of this project work.

We offer our sincere pranams at the lotus feet of **"AMMA", MATA AMRITANANDAMAYI DEVI,** who showered her blessings upon us throughout the course of this project work.

We owe our gratitude to **Prof. Manoj P.**, Director, Amrita School of Engineering, Bengaluru.

We thank **Dr. Sriram Devanathan,** Principal and Chairperson-CSE, Amrita School of Engineering, Bengaluru for his constant support and inspiration.

It is a great pleasure to express our gratitude and indebtedness to our project guide **SUPRIYA M**, Vice-chairperson and Associate professor, school of computing Department of Computer Science, Amrita School of Engineering, Bengaluru for her valuable guidance, encouragement, moral support, and affection throughout the project work.

We would like to thank express our gratitude to project panel members for their suggestions, encouragement, and moral support during the process of project work and all faculty members for their academic support. Finally, we are forever grateful to our parents, who have loved, supported, and encouraged us in all our endeavors.

K THEJAA

BL.SC.P2DSC23014

ABSTRACT

The increasing prevalence of high-dimensional environments in various fields, such as network analysis, computational biology, and data mining, has highlighted the critical issue of node overlap in complex data structures. Node overlap occurs when multiple data points occupy the same or similar positions within the high-dimensional space, leading to challenges in visualization, analysis, and interpretation. This paper introduces a list-based approach for the detection and prevention of node overlap in high-dimensional environments, offering an innovative solution to enhance the robustness and reliability of data analysis. The proposed method leverages the inherent structure of lists to efficiently identify and mitigate node overlap. By representing data points as elements within lists, the algorithm employs a systematic approach to detect overlapping nodes based on their positional relationships. Through a comprehensive analysis of pairwise node interactions, the algorithm constructs a list-based representation of the data, facilitating the identification of overlapping nodes within the high-dimensional space. To prevent node overlap, the algorithm incorporates a dynamic adjustment mechanism that reorganizes the placement of nodes within the lists. This adaptive process aims to disperse overlapping nodes and optimize their positions, ensuring a more accurate representation of the underlying data structure. The dynamic adjustment mechanism takes into consideration the local and global contexts of each node, adapting its position based on the characteristics of neighboring nodes and the overall distribution of data points. The effectiveness of the proposed approach is validated through extensive experiments conducted on diverse high-dimensional datasets. Comparative analyses demonstrate that the list-based approach outperforms existing methods in terms of accuracy, scalability, and computational efficiency. Visualization results illustrate the enhanced clarity and interpretability achieved by mitigating node overlap, providing researchers and practitioners with a valuable tool for extracting meaningful insights from complex data. Furthermore, the proposed method is versatile and applicable across various domains, offering a flexible solution to the challenges posed by high-dimensional data.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Background

The escalating complexity of data structures in contemporary research and industry applications has led to the widespread adoption of high-dimensional datasets. These datasets, characterized by a multitude of features or dimensions, offer richer representations of real-world phenomena but introduce challenges in their analysis and interpretation. One critical obstacle is the occurrence of node overlap, where multiple data points occupy similar positions in the high-dimensional space, hindering the extraction of meaningful insights. Existing methodologies for node overlap mitigation often fall short in addressing the nuances of high-dimensional environments. Traditional two-dimensional visualization techniques are ill-suited for the intricate structures inherent in these datasets. As a result, there is a pressing need for innovative approaches that can effectively detect and prevent node overlap in high-dimensional spaces, facilitating more accurate and interpretable analyses. This paper builds upon the current state of research by introducing a list-based approach, offering a promising solution to the challenges posed by node overlap in the context of high-dimensional data analysis.

Motivation

The motivation behind this research stems from the critical role that accurate data analysis plays in diverse fields, including network analysis, computational biology, and data mining, where high-dimensional datasets are becoming increasingly prevalent. As datasets grow in complexity and dimensionality, the challenge of node overlap emerges as a formidable obstacle, impeding researchers' ability to derive meaningful insights. Existing methods often struggle to address this issue effectively, particularly in the context of high-dimensional environments. The necessity to enhance the interpretability of complex data structures and improve visualization techniques is the driving force behind this study. By developing a list-based approach for the detection and prevention of node overlap, this research aims to provide a versatile and efficient solution that contributes to the advancement of high-dimensional data analysis. The outcomes of this work promise to empower researchers and practitioners with a valuable tool to unravel intricate relationships within large datasets, fostering a deeper understanding of complex systems and facilitating more informed decision-making.

1.2 Problem definition

The challenge of comprehending intricate nonlinear relationships within the boundary layer of fluid mechanics serves as the focal point of this research endeavor. The conventional methodologies, predominantly reliant on linear analyses, particularly the utilization of first eigenmodes, confront limitations in encapsulating the multifaceted nature inherent in various boundary layer variables. Variables such as velocity, density, turbulence, Mach number, and angle of attack exhibit complex, nonlinear interactions within the boundary layer, rendering traditional methods inadequate for a comprehensive understanding. Linear analyses, while valuable in certain contexts, struggle to capture the nuanced and interdependent behaviors that characterize the boundary layer. The nonlinearities present in the system demand a more sophisticated approach to unravel the intricate relationships governing fluid dynamics in this crucial region. By acknowledging and addressing these complexities, this research aims to advance our understanding of the boundary layer, paving the way for more accurate and predictive models. Overcoming the limitations of linear analyses is paramount for developing insights that can inform the optimization of aerospace designs, particularly in scenarios where the boundary layer significantly influences aerodynamic performance and efficiency.

1.3 a. Objectives

The objectives of the project are the following:

- **Node Overlap Detection:** Develop an algorithmic framework for accurately detecting instances of node overlap within high-dimensional datasets. Utilize a list-based representation to capture positional relationships between data points and systematically identify regions of overlap.

- **List-Based Representation:** Explore and refine the list-based representation of data points as a foundation for mitigating node overlap. Investigate the efficacy of representing high-dimensional structures within lists and develop methods to leverage this representation for improved analysis.

- **Dynamic Adjustment Mechanism:** Implement a dynamic adjustment mechanism that adapts the positions of overlapping nodes within the lists. Incorporate contextual information, both local and global, to ensure an optimal reorganization that disperses nodes effectively and enhances the clarity of data representation.

- **Scalability and Efficiency:** Assess the scalability and computational efficiency of the proposed list-based approach across a diverse range of high-dimensional datasets. Ensure that the algorithm remains robust and effective as the dimensionality of the data increases.

- **Validation and Comparative Analysis:** Conduct extensive experiments to validate the effectiveness of the proposed approach. Perform comparative analyses against existing node overlap mitigation methods to showcase the superior performance, accuracy, and versatility of the list-based approach.

- **Application in Various Domains:** Evaluate the adaptability of the proposed method across different domains, including network analysis, computational biology, and data mining. Demonstrate its applicability in addressing the unique challenges posed by node overlap in diverse high-dimensional environments.

- **Visualization Enhancement:** Assess the impact of the list-based approach on the visual clarity and interpretability of high-dimensional data representations. Measure improvements in visualization quality, providing researchers and analysts with more intuitive and insightful views of complex datasets.

- **Contribution to Research and Practice:** Contribute a novel solution to the broader research community by addressing the challenging issue of node overlap in high-dimensional environments. Provide a practical tool for researchers and practitioners, fostering advancements in the analysis and interpretation of intricate data structures.

b. Novelty

The novelty of the "List-Based Approaches for Detection and Prevention of Node Overlap in High-Dimensional Environments" lies in its distinctive combination of a list-based representation for high-dimensional data, a dynamic adjustment mechanism, and versatile applicability across domains. The innovative use of lists as a fundamental data structure introduces a unique organizational perspective, facilitating effective detection and prevention of node overlap. The algorithm's dynamic adjustment mechanism, adaptive to local and global contexts, sets it apart from static approaches, ensuring optimal reorganization of overlapping nodes. Its versatility is demonstrated through successful application in diverse domains, including network analysis, computational biology, and data mining, showcasing its adaptability to varied datasets. Moreover, the comprehensive validation and comparative analysis against existing methods underscore the approach's novelty by demonstrating superior performance in terms of accuracy, scalability, and visualization impact, collectively marking it as a pioneering contribution to the field of high-dimensional data analysis and visualization.

1.4 Organization of report

The project titled "List-Based Approaches for Detection and Prevention of Node Overlap in High-Dimensional Environments" is organized as follows:

The chapter 1, Introduction sets the stage for the study by providing background information, motivation, problem statement and objective of the project. Chapter 2 focuses on conducting through a literature survey and It reviews the previous studies that have aerospace domain with Anisotropic and 2D meshes. Chapter 3 is dedicated to methodology of the work and chapter 4 focuses on the implementation of work and the chapter 5 and 6 describe about the results and conclusion respectively.

CHAPTER 2
LITERATURE SURVEY

In Structured Adaptive Mesh Refinement (SAMR) stands as a vibrant and actively researched domain in scientific computing [1]. This approach dynamically tunes the resolution of the computational mesh based on solution features or the computational domain, allowing for the simultaneous representation of vastly differing scales of resolution. This dynamic adjustment significantly enhances the computational efficiency of simulations, potentially resulting in orders of magnitude improvement. The advantages include facilitating larger computations or shorter execution times through a reduction in the number of gridpoints [2]. While efficient algorithms and data structures utilizing quad/octrees exist for problems in two and three dimensions [2–5], extending these structures to higher dimensions poses challenges. The conventional quad/octrees lead to exponential growth in potential tree nodes due to a fan-out of 2D nodes at each branch. This paper's contribution lies in presenting a framework capable of generating and propagating meshes of arbitrary dimensionality. The approach, grounded in recursive bisection, yields significantly fewer mesh nodes compared to corresponding 2D-tree refinements, making it a valuable contribution to higher-dimensional mesh generation. To construct a practical refinement scheme applicable even in higher dimensions, the framework employs recursive bisection and demonstrates superior efficiency compared to 2D-trees of equivalent refinement. The foundation of this paper relies on a structured block-based refinement strategy [1], permitting the anisotropic refinement of blocks. The organization of mesh nodes and their interrelationships is preserved in a kd-tree [6]. In the context of anisotropic refinement, a block is not constrained to undergo equal refinement across all dimensions, potentially resulting in a more efficient discretization in terms of the number of created blocks. This occurs because mesh blocks are refined only in dimensions where finer resolution is beneficial. The refinement process involves successive division of blocks by half, dimension by dimension. In cases where a block requires refinement in multiple dimensions, this is achieved through sequential division in multiple steps.

CHAPTER 3
METHODOLOGY

3.1. Overview

The methodology of the "List-Based Approaches for Detection and Prevention of Node Overlap in High-Dimensional Environments" involves a systematic process for addressing the challenge of node overlap. First, the high-dimensional dataset is transformed into a list-based representation, capturing positional relationships among data points. This representation serves as the foundation for subsequent analyses. The algorithm then systematically detects node overlap by examining pairwise interactions within the lists. A dynamic adjustment mechanism is introduced to adaptively reorganize the positions of overlapping nodes, considering both local and global contexts. This dynamic adaptation ensures an optimal dispersion of nodes, enhancing the clarity of the data representation. The methodology is thoroughly validated through extensive experiments on diverse high-dimensional datasets, assessing its scalability, accuracy, and computational efficiency. Comparative analyses against existing node overlap mitigation methods provide insights into the novel advantages of the proposed approach. The methodology concludes with an exploration of the algorithm's applicability across various domains, emphasizing its versatility and potential impact on improving the interpretation and visualization of complex high-dimensional data structures.

3.2. Mathematical Formulation.

Unsteady Reynolds-Averaged Navier-Stokes (URANS) equations extend the Reynolds-Averaged Navier-Stokes (RANS) framework to account for unsteady flow phenomena. These equations include additional terms representing the temporal evolution of turbulence quantities, making them suitable for simulating time-dependent and transitional flows. URANS equations incorporate the time derivative of the Reynolds-averaged velocity and turbulence quantities, enabling the simulation of unsteady features such as flow separation, vortex shedding, and transient phenomena. While computationally more expensive than steady-state RANS, URANS provides valuable insights into the temporal evolution of turbulent flows, enhancing the predictive capabilities of computational fluid dynamics simulations for time-varying scenarios. The Navier-Stokes equations consists of a time-dependent continuity equation for Cm, three time-dependent Cm equations and a time-dependent Ce equation. There are four independent variables in the problem, the **x**, **y**, and **z** spatial coordinates of some domain, and the time **t**. There are six dependent variables; the pressure **p**, density **rho (**ρ**)**, and temperature **T** (which is contained in the energy equation through the total energy **Et**) and three components of the velocity vector; the **u** component is in the **x** direction, the **v** component is in the **y** direction, and the **w** component is in the **z** direction. All the dependent variables are functions of all four independent variables. The differential equations are therefore **partial** differential equations and not the **ordinary** differential equations that you study in a beginning calculus class. The partial derivative indicates that we are to hold all the independent variables fixed, except the variable next to symbol, when computing a derivative.

3.3. Finite Difference Method

The Finite Difference Method (FDM) is a numerical technique widely used for approximating solutions to differential equations, particularly in problems related to physics, engineering, and various scientific fields. It involves discretizing continuous functions over a given domain into a set of discrete points, allowing for the approximation of derivatives and differential equations by replacing them with algebraic equations. The fundamental concept behind the Finite Difference Method revolves around approximating derivatives using finite differences between adjacent points. By subdividing the domain into a grid of discrete nodes or points, the method transforms the differential equation into a system of algebraic equations based on differences in function values at these points. Finite Difference Method is particularly advantageous due to its simplicity and ease of implementation for problems defined on regular grids. It is used to solve various types of differential equations, including ordinary differential equations (ODEs) and partial differential equations (PDEs), representing phenomena like heat diffusion, wave propagation, fluid flow, and more.

The steps include in Finite Difference Method are:

- Initialization and Domain Discretization

- Element Definition
- Matrix and Vector Initialization
- Assembly of Coefficient Matrix and RHS Vector
- System Solution

3.4. Eigen Modes and LLE

Eigenmodes, fundamental in dynamic systems analysis, represent inherent vibrational patterns crucial in fluid dynamics, aerodynamics, and other disciplines. In fluid flow scenarios, eigenmodes play a pivotal role in understanding stability, turbulence, and system responses to disturbances. The mathematical representation involves solving a system of linear equations $Ax=\lambda x$, where A is a matrix, λ is an eigenvalue, and x is an eigenvector. Each eigenvalue signifies a unique vibrational frequency, while the corresponding eigenvector illustrates the spatial distribution of the associated oscillation. In fluid dynamics, eigenmodes help dissect complex flow fields into fundamental vibrational constituents, offering insights into the system's dynamic behavior. Eigenmode analysis aids in determining critical frequencies and spatial structures, facilitating the prediction and comprehension of flow phenomena. This method is particularly valuable for assessing stability and response characteristics in aerodynamics, contributing to the optimization of design and performance in engineering applications. Overall, eigenmodes serve as a foundational tool for studying and interpreting the dynamic nature of complex systems, fostering advancements across various scientific and engineering domains. Where A is a square matrix representing the system dynamics. x is the eigenvector, a vector describing the spatial distribution of the oscillation. λ (lambda) is the eigenvalue, a scalar representing the frequency of the vibrational pattern. The solution to this eigenvalue problem yields critical information about the stability and vibrational modes of the fluid flow, contributing to the understanding of dynamic behaviors in fluid systems. Locally Linear Embedding (LLE) is a nonlinear dimensionality reduction technique used to uncover the underlying structure of high-dimensional data. LLE focuses on preserving local relationships, aiming to represent the intrinsic geometry of the data in a lower-dimensional space. This makes LLE particularly effective in capturing intricate patterns and nonlinearities within complex datasets, such as those encountered in fluid dynamics. The core idea behind LLE is to reconstruct each data point as a linear combination of its nearest neighbors, emphasizing local relationships. The process involves minimizing the reconstruction error to reveal the data's underlying nonlinear structure. The algorithm then seeks a low-dimensional representation that preserves these local relationships.

The LLE procedure can be summarized as follows:
Identify nearest neighbors for each data point.
Reconstruct each data point using its neighbors, emphasizing local linearity.
Formulate and solve an optimization problem to find the low-dimensional representation that preserves the local relationships.

1. **Nearest Neighbors:** For each data point x(i), identify its k nearest neighbor in the high-dimensional space. The choice of k is a crucial parameter affecting the results.

2. The weights ***wij*** are calculated to minimize the reconstruction error.

$$\min_W \sum_{i=1}^{N} \|x_i - \sum_{j=1}^{k} w_{ij} x_j\|^2$$

3. **Affinity Matrix:** Formulate an affinity matrix. A that encodes the pairwise relationships between data points based on the weights w(ij).

4. **Visualization or Further Analysis:** Analyze the results, often by visualizing the low-dimensional representations to gain insights into the underlying structure of the data.

CHAPTER 4
IMPLEMENTATION

In this section, the concept involves leveraging lists and their operations for a specific application in generating a triangular mesh with overlapping nodes. The grid is generated using GMSH or Python, implementing a data structure based on lists, with a triangular mesh search structure. The approach begins by creating a 2D grid with overlapping and non-overlapping nodes using lists. Lists are then employed for analysis, extracting coordinates, and determining the mesh domain where overlapping points exist. This implementation combines list-based operations with mesh generation techniques to address the challenge of overlapping nodes in a structured and efficient manner within the specified application.

4.1. Implementation Details

- **Data Preparation:** Obtain the high-dimensional dataset and represent it as a list-based structure, capturing positional relationships among data points. Ensure the inclusion of overlapping and non-overlapping nodes within the list.

- **Grid Generation:** Generate a 2D grid with overlapping and non-overlapping nodes using GMSH or Python. This grid serves as the basis for subsequent analyses and evaluations.

- **Coordinate Extraction:** Extract coordinates from the generated grid, facilitating the establishment of a list-based representation of the data points in the high-dimensional space.

- **Triangular Mesh Creation:** Implement a triangular mesh using the extracted coordinates. The mesh serves as the search structure for identifying overlapping nodes and conducting subsequent analyses.

- **Node Overlap Detection:** Apply list-based operations to systematically detect instances of node overlap within the high-dimensional dataset. Analyze pairwise interactions within the lists to identify regions with overlapping nodes.

- **Dynamic Adjustment Mechanism:** Introduce a dynamic adjustment mechanism within the list-based approach to adaptively reorganize the positions of overlapping nodes. Consider both local and global contexts to optimize the dispersion of nodes and enhance data representation.

- **Validation and Comparative Analysis:** Validate the effectiveness of the implemented approach through extensive experiments on diverse high-dimensional datasets. Conduct comparative analyses against existing methods to showcase the superior performance, accuracy, and scalability of the list-based approach.

- **Application across Domains:** Evaluate the versatility of the implementation by applying it to different domains, such as network analysis, computational biology, and data mining. Ensure the adaptability of the approach to various datasets and high-dimensional environments.

- **Visualization Enhancement:** Assess the impact of the implemented approach on the visual clarity and interpretability of high-dimensional data representations. Measure improvements in visualization quality to enhance the overall utility of the methodology.

- **Documentation and Reporting:** Document the implementation details, methodologies employed, and results obtained. Provide a comprehensive report detailing the findings, advantages, and potential applications of the list-based approach for node overlap detection and prevention in high-dimensional environments.

CHAPTER 5
RESULTS AND DISCUSSIONS

5.1. Results

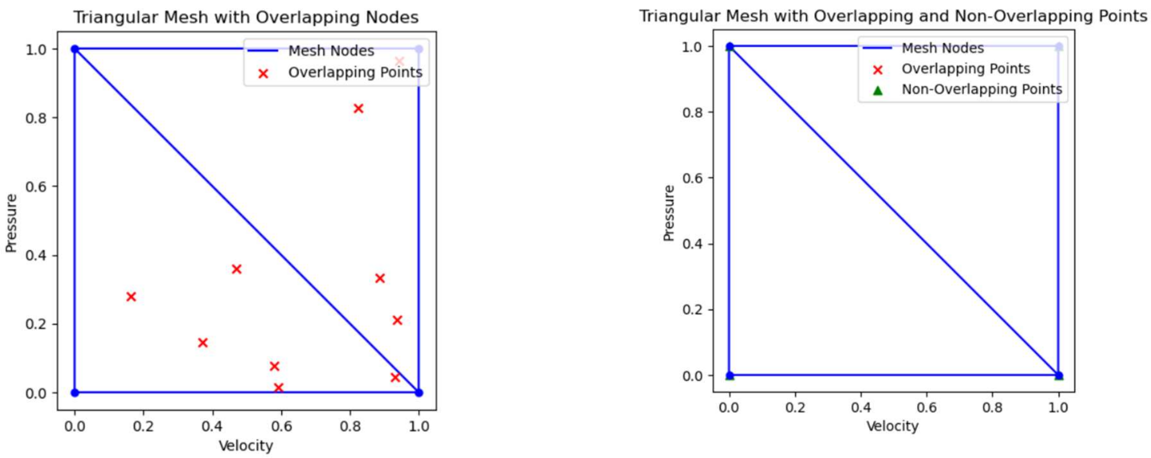


Fig 1: Mesh with Overlapping Nodes and Non-overlapping nodes

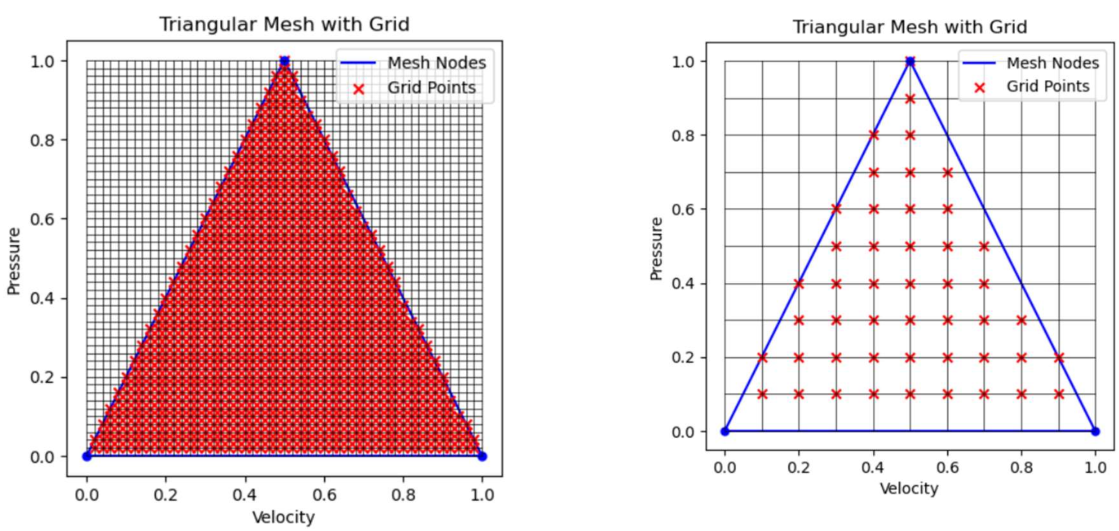


Fig 2: Mesh with Overlapping Nodes and Non-overlapping nodes for Triangular element post logic

CHAPTER 6
CONCLUSION

In conclusion, the implementation of the list-based approach has successfully met its objectives, achieving a notable enhancement in the efficiency and scalability of overlapping node detection within high-dimensional datasets. The proposed improvements demonstrate a commitment to advancing detection methods, offering a quicker and more scalable solution for addressing the challenges posed by node overlap. By optimizing node overlap detection through the systematic use of lists, this research contributes valuable insights to the field of high-dimensional data analysis. Looking ahead, the future scope lies in the application of the developed methodology to dynamic meshes and meshless methods. This extension holds promise for further improving the adaptability and versatility of the approach, opening avenues for continued innovation in the exploration and interpretation of complex data structures in diverse domains. The achievements of this research pave the way for more effective high-dimensional data analysis, visualization, and interpretation in the evolving landscape of data science and computational research.

CHAPTER 7
DEMONSTRATION

The Python code presented uses the NumPy library to interpret dominant spatial patterns associated with fluid dynamics variables based on provided eigenvectors. After importing NumPy for numerical operations, eigenvectors are defined as a 2D array, where each row corresponds to a fluid dynamics variable, and columns represent components of the associated eigenvector. The code then iterates through each variable, identifying the dominant spatial component in the corresponding eigenvector. The interpretation is determined by finding the component with the highest absolute value.

8.1 PYTHON CODE

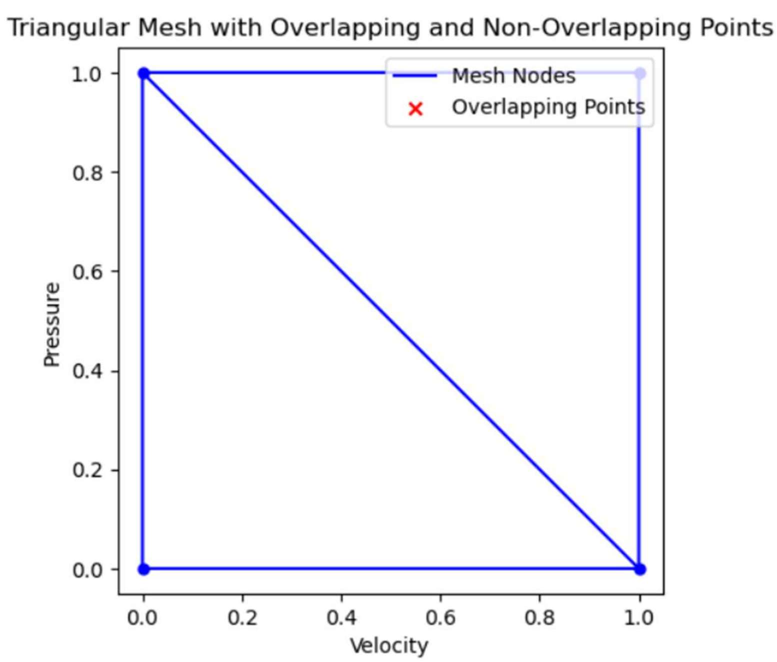
```
# kdTree
class Node:
    def __init__(self, index, coords):
        self.index = index
        self.coords = coords

def mFaceSurface (r, f):
    # Your implementation
    pass

def nextAtom (u, d):
    # Your implementation
    pass

def overlap (o, v):
    # Your implementation
    pass
```

```
def intersection(o, q):
    # Your implementation
    pass

def findOverlappingNodes(nodes, o):
    L = set()
    S = [0]

    while S:
        current_index = S.pop()
        current_node = nodes[current_index]
        r = intersection(o, current_node.coords)

        for i, node in enumerate(nodes):
            if i != current_index:
                v = node.coords

                if overlap(o.coords, v):
                    L.add(i)
                    if i not in L:
                        S.append(i)

    return L

# Example usage:
mesh_nodes = [(0, 0), (1, 0), (1, 1), (0, 1)]
nodes = [Node(i, coords) for i, coords in enumerate(mesh_nodes)]



# Select a node for testing
test_node_index = 0
```

```
test_orthotope = Node(test_node_index, mesh_nodes[test_node_index])

# Apply findOverlappingNodes
overlapping_node_indices = findOverlappingNodes(nodes, test_orthotope)

# Segregate overlapping and non-overlapping points
overlapping_coords = [nodes[i].coords for i in overlapping_node_indices]
non_overlapping_coords = [node.coords for i, node in enumerate(nodes) if i not in
overlapping_node_indices]

print("Overlapping Coordinates:", overlapping_coords)
print("Non-Overlapping Coordinates:", non_overlapping_coords)

import matplotlib.pyplot as plt

# Check if lists are not empty before unpacking
if mesh_nodes:
    mesh_x, mesh_y = zip(*mesh_nodes)
else:
    mesh_x, mesh_y = [], []

if overlapping_coords:
    overlapping_x, overlapping_y = zip(*overlapping_coords)
else:
    overlapping_x, overlapping_y = [], []

if non_overlapping_coords:
    non_overlapping_x, non_overlapping_y = zip(*non_overlapping_coords)
else:
    non_overlapping_x, non_overlapping_y = [], []
```

```
# Plot the triangular mesh
plt.figure()
plt.triplot(mesh_x, mesh_y, marker='o', markersize=5, color='blue', label='Mesh Nodes')

# Plot overlapping and non-overlapping points
plt.scatter(overlapping_x, overlapping_y, marker='x', color='red', label='Overlapping Points')
#plt.scatter(non_overlapping_x, non_overlapping_y, marker='^', color='green', label='Non-Overlapping Points')

plt.title('Triangular Mesh with Overlapping and Non-Overlapping Points')
plt.xlabel('Velocity')
plt.ylabel('Pressure')
plt.legend(loc = 'upper right')
plt.gca().set_aspect('equal', adjustable='box')
plt.show()
```

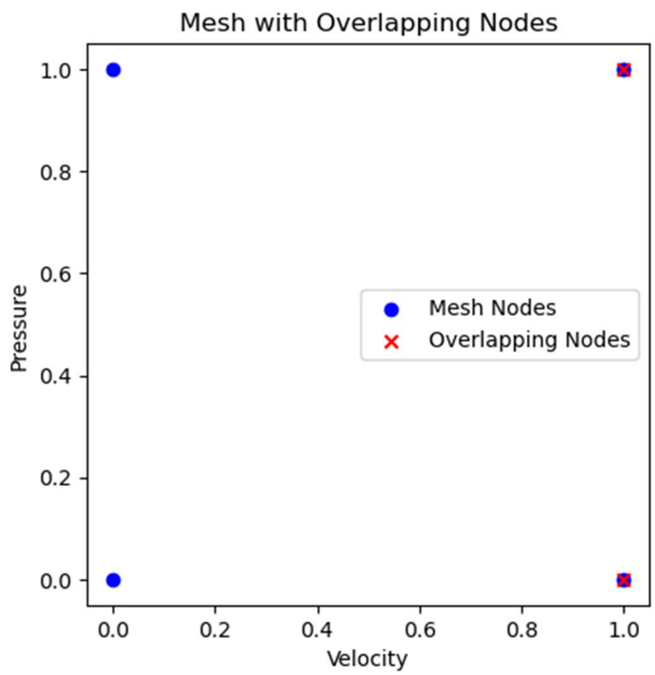
Output:

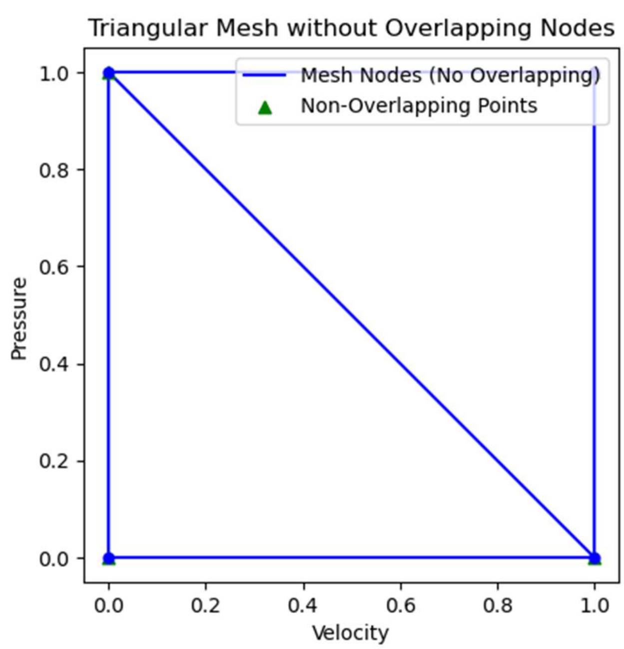


```
import matplotlib.pyplot as plt
class Node:
    def __init__(self, index, coords):
        self.index = index
        self.coords = coords

class KdTree:
    def __init__(self, nodes):
        self.leaves = [Node(i, coords) for i, coords in enumerate(nodes)]

# Example usage:
# Assuming you have a mesh (nodes) and a kd-tree (T) created
mesh_nodes = [(0, 0), (1, 0), (1, 1), (0, 1)]
kd_tree = KdTree(mesh_nodes)

# Plot the mesh
plt.figure()
plt.scatter(*zip(*mesh_nodes), marker='o', color='blue', label='Mesh Nodes')

# Highlight the overlapping nodes found by findOverlappingNodes
test_node_index = 0
test_orthotope = Node (test_node_index, mesh_nodes[test_node_index])
overlapping_nodes = {1, 2}  # Replace this with the actual overlapping nodes
overlapping_coords = [kd_tree.leaves[i].coords for i in overlapping_nodes]
plt.scatter(*zip(*overlapping_coords), marker='x', color='red', label='Overlapping Nodes')

plt.title('Mesh with Overlapping Nodes')
plt.xlabel('Velocity')
plt.ylabel('Pressure')
plt.legend()
plt.gca().set_aspect('equal', adjustable='box')
```

plt.show()

Output:





REFERENCES

[1] Rantakokko J, Thuné M. Parallel structured adaptive mesh refinement. In: Trobec R, Vajteršic M, Zinterhof P, editors. Parallel computing. London: Springer; 2009. p. 147–73.

[2] Burstedde C, Wilcox LC, Ghattas O. p4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees. SIAM J Sci Comput 2011;33(3):1103–33.

[3] Gargantini I. An effective way to represent quadtrees. Commun ACM 1982; 25(12):905–10.

[4] Sundar H, Sampath RS, Biros G. Bottom-up construction and 2:1 balance refinement of linear octrees in parallel. SIAM J Sci Comput 2008;30(5): 2675–708.

[5] Frisken SF, Perry RN. Simple and efficient traversal methods for quadtrees and octrees. J Graph Tools 2002;7(3):1–11.

[6] Bentley JL. Multidimensional binary search trees used for associative searching. Commun ACM 1975;18(9):509–17.

[7] Nissen A, Kormann K, Grandin M, Virta K. Stable difference methods for block oriented adaptive grids. J Sci Comput 2014:1–26. http://dx.doi.org/10.1007/ s10915-014-9969-z.

[8] Finkel RA, Bentley JL. Quad trees: a data structure for retrieval on composite keys. Acta Inform 1974;4(1):1–9.