

KYAMOS Software - Lattice Boltzmann oil spill simulations

Antonios P. Papadakis^{1,2}, Nicholas Antoniou¹, Sofia Nikolaidou¹, Vasilis Hadjigeorgiou¹, Sotiris Pigiotis¹, Nektaria Mila¹ and Wasif Almady^{1,2}

¹KYAMOS LTD, 37 Polyneikis Street, Strovolos, 2047, Nicosia, Cyprus

²Frederick University, Pallouriotissa, 1036, Nicosia, Cyprus

Abstract— This study utilizes the sophisticated Lattice Boltzmann method (LBM) enhanced by Multiple Relaxation Time (LB-MRT) to simulate the complex dynamics of oil spills. Chosen for its ability to accurately manage interactions between oil and water, the LBM allows for realistic modeling of oil behavior following marine spills. The MRT model significantly enhances the LBM by offering improved stability, accuracy, and flexibility for simulating complex fluid dynamics in oil spill scenarios. Its application provides a detailed and realistic portrayal of spill dynamics, crucial for developing effective response strategies and advancing environmental simulation technologies.

Our simulations provide detailed visualizations of initial oil dispersion and subsequent spread, offering crucial insights for emergency response planning and environmental impact minimization. Future enhancements will focus on boosting computational efficiency and incorporating complex interactions such as chemical degradation and biological effects, as well as detailed geographic features.

By improving predictive capabilities and broadening applicability, this research enhances our ability to effectively manage and mitigate the consequences of oil spills, demonstrating the significant benefits of advanced computational methods in environmental disaster management.

Keywords—Lattice Boltzmann; Multiple Relaxation Time; incompressible flows; Oil-Spills, High Performance Computing

I. INTRODUCTION

Oil spills stand as one of the most critical environmental disasters, causing widespread damage to marine life, coastal communities, and local economies. The immediate consequences, such as habitat contamination, are visible and alarming, but also the long-term effects, harming biodiversity and economic activities, pose a lingering threat. The challenge in addressing oil spills lies in their complexity, influenced by factors like the type of oil spilled, the amount, the environmental conditions at the time, and the specifics of the affected coastal areas.

A clear example of such a disaster is the 2010 Deepwater Horizon spill in the Gulf of Mexico, which lasted 87 days and released over 3 million barrels of oil [1]. This incident led to the death of thousands of marine animals and birds, and economic losses estimated in the billions, impacting the fishing and tourism industries. Moreover, the full extent of the spill's impact on public health and the quality of life for millions remains uncertain. Factors such as currents, wind and the oil's chemical makeup critically influence how spilled oil behaves and changes over time. Processes like evaporation, mixing with water, breaking down, and being consumed by microbes make the spilled oil's impact more complex and challenging to mitigate.

In this context, computational modeling has become a key strategy for managing oil spill impacts. By simulating how oil spills evolve in marine environments, these models help predict the best response strategies, guiding efficient clean-up efforts and protecting vulnerable areas. They play a crucial role in both immediate responses, helping to quickly contain and clean up spills, and long-term planning, improving strategies for dealing with spills and informing policies to prevent future disasters. Essentially, computational models help make informed decisions faster, reducing the overall damage of oil spills and aiding in the recovery of affected ecosystems and economies.

The Lattice-Boltzmann Method

The choice of the Lattice Boltzmann Method (LBM) over traditional computational fluid dynamics (CFD) techniques such as Finite Difference (FD), Finite Element (FE), and Finite Volume (FV) methods for simulating oil spills is driven by several compelling factors that align with the unique challenges posed by such environmental phenomena. These factors include the inherent suitability of LBM for multiphase flows, its ability to handle complex boundary conditions, and its scalability on modern computational architectures, among others.

One of the critical challenges in simulating oil spills is accurately modeling the interaction between multiple phases – oil, water, and possibly air. LBM inherently excels in dealing with multiphase and multicomponent flows, thanks to its mesoscopic nature and the ease of implementing complex interfacial dynamics.

Oil spill simulations frequently involve complex geometries and dynamic interactions with the environment, such as coastlines, vegetation, and man-made structures. LBM's discrete lattice structure and local collision dynamics facilitate the straightforward handling of complex boundaries and moving objects. In contrast, FD, FE, and FV methods typically require intricate meshing strategies and often struggle with stability issues near boundaries.

Additionally, turbulence plays a significant role in the dispersion of oil spills. LBM directly simulates the Navier-Stokes equations at a mesoscopic level, enabling the capture of turbulent flow features with a higher degree of accuracy and reliability than the traditional CFD methods. The intrinsic kinetic approach of LBM allows for a more natural representation of the turbulent eddies and flow structures characteristic of oil spill scenarios.

Furthermore, the parallel nature of LBM, where computations are localized to the lattice nodes, makes it highly scalable and efficient on modern high-performance computing architectures, including GPUs, as demonstrated in this project. This scalability is important for simulating the vast spatial extents and fine details required in oil spill simulations. While FD, FE, and FV methods can also be parallelized, LBM's simplicity in data structure and local communication patterns results in superior computational efficiency and easier implementation on massively parallel systems.

II. LITERATURE REVIEW

In the literature, authors commonly employ either the Lagrangian (microscopic) or the Eulerian (macroscopic) approach to simulate the propagation of oil spills in the sea. Recently, a novel technique, the Lattice Boltzmann (mesoscopic) method, has emerged. This approach bridges the gap between the two traditional methods by utilizing particle velocity distribution functions, enabling the transition from microscopic to macroscopic variables.

Lattice Boltzmann has already been used in the simulation of oil-spills. Zhang et al [2-3] have used the Lattice Boltzmann advection-diffusion equation to study the oil-spill that occurred at the Gulf of Mexico using real ocean current data from the Unified Wave Interface-Coupled model. They have validated their model against GNOME and involved the simulation of a Gaussian hill using linear currents and realistic ocean currents and were found to be in good agreement.

Meslo et al. [4] have used LB simulations to conduct large scale oil-spill modeling for the Lebanon oil spill. They have investigated the stability of single and two relaxation time models and have included flux limiting interpolation techniques on the velocity in the Lattice Boltzmann to avoid oscillations.

Keramea et al. [5] have provided an extensive review regarding oil-spill simulations created from marine traffic, petroleum production, or other sources. The models range from simplistic models to three-dimensional models, which can be coupled to meteorological, hydrodynamic, and wave models, able to forecast in high-resolution, the transportation of oil. They list eighteen, state-of-the-art models for oil-spill simulations and signify spreading, advection, diffusion, evaporation, emulsification, and dispersion as the most significant processes involved in oil-spill simulations. Some of the more important software are CDOG, OSCAR, OSIS, OILMAP, OILMAPDEEP, SIMAP, TAMOC, BLO-SOM, MOTHY, OILTOX, MOHID, POSEIDONOSM, MEDS-LIK, GNOME, OILTRANS, OSERIT, MEDSLIK-II and OPENOIL.

Li et. al. [6] utilizes a hydrodynamic model to analyze the oil-spill in the Luanjiakou District, near the Port of Yantai. In the mathematical model, they include the spreading of an oil slick on its edge, diffusion and drift, evaporation and spreading thickness of an oil slick in its interior, and the adsorption and emulsification near shorelines and islands. In another paper by Li et al. [7], they have used a coupled numerical method for the direct simulation of shallow water dynamics and pollutant transport. For this purpose, the shallow water dynamics equations and the convection-diffusion equations are solved using the Lattice Boltzmann method. In a paper by Zhang et al. [8], they use the Lattice Boltzmann method for oil-spill simulation by comparing the Navier-Stokes model vs the advection-diffusion equation models. Ha et al. [9] have used the Lattice Boltzmann method to simulate the advection-diffusion of the spread of oil-slick on the sea. Li et al. [10] have used the D2Q5 and D2Q9 lattices to simulate the convection-diffusion equation for scalar transport. Banda et al. [11] have used the Lattice Boltzmann method to simulate pollutant dispersion by shallow water flows. The mass, momentum and transport equations are obtained from the nine-velocity distributions (D2Q9) of hydraulic flow and pollutant concentration variables for the Strait of Gibraltar.

In a book by Fingas [12], oil spill modeling is introduced, where predictions are given for key processes such as evaporation, water-in-oil emulsification, spontaneous dispersion, and dissolution. Other processes such as photo-oxidation, sedimentation and oil-micromaterial interactions are also

discussed. An important behavioral process is evaporation. Light oil can lose 30-60% of its mass in 2 days due to evaporation. Algorithms for calculating the amount of evaporation are summarized. Another important behavior is hydration. Oil can absorb water in one of five ways: as water soluble, no significant uptake or instability, entrained water, meso-stable emulsion or stable emulsion. The last three mechanisms are important as they have a large impact on the further behavior of the oil and have a large impact on countermeasures. Fingas stresses also the effect of environmental factors such as weathering on the oil spill development, and the need for accurate oceanic currents, as well as environmental weather conditions.

Given the increasing frequency and severity of oil spill incidents, there's an immediate need to improve the reliability and accuracy of the methods employed in oil spill detection and mapping [13-19]. The complexities of detecting, monitoring, and categorizing oil spills in oceans and seas pose significant challenges [13-14]. Consequently, researchers have implemented machine learning algorithms to address these issues. Numerous machine learning approaches have been applied to detect oil spills, including decision trees [20], support vector machines [21-22], random forests [23-25], and artificial neural networks [26-30]. Among these techniques, deep learning has garnered increased attention [30-34]. Broadly classified as a subset of machine learning methods [35], deep learning distinguishes itself by learning directly from data instead of relying on predetermined features [32,34,35]. Recent advancements in deep learning architectures have significantly impacted the detection of oil spills [30,34]. Therefore, providing an overview of the current state and trends in using deep learning for oil spill detection and mapping becomes crucial to consolidate practical analytical methods and approaches for identification and monitoring. This effort contributes to the enhancement of knowledge and the evolution of this scientific domain [30-31].

III. METHODOLOGY

A. Diffusion Equation

We utilize the advection-diffusion equation as a cornerstone for modelling the dispersion of oil spills in marine environments. The advection-diffusion equation for sea water and oil, represented as

$$\frac{\partial C}{\partial t} + u \cdot \nabla C = D \Delta C$$

, where C denotes the concentration of oil at a given point in space and time, $\partial C/\partial t$ signifies the rate of change of concentration with respect to time, u represents the velocity field of the fluid, D is the diffusion coefficient, and ΔC represents the Laplacian operator acting on the concentration field, serves as the foundation for understanding the combined effects of advection and diffusion on the movement of oil in water. By accounting for both the transport of oil by fluid currents (advection) and its spread due to random molecular motion (diffusion), this equation enables us to predict the spatial and temporal evolution of oil slicks. Our methodology involves solving this partial differential equation numerically, considering environmental factors such as sea currents, to accurately simulate the complex dynamics of oil spill dispersion.

B. Mathematical model

The lattice Boltzmann method originated from the Ludwig Boltzmann's kinetic theory of gases, which was proposed by James Clerk Maxwell and others between the 1860s to 1890s

[36]. The lattice Boltzmann method has garnered increasing interest since it emerged from lattice gas models in the late 1980s [37]. Additionally, the lattice Boltzmann method (LBM) with the multiple relaxation time (MRT) collision operator was developed by a group of researchers led by Prof. Shiyi Chen and Prof. Hao Chen, in the late 1990s. The LB-MRT method adjusts multiple relaxation time parameters to offer higher stability and accuracy than the LB-BGK model [38].

1) Equations

The Lattice-Boltzmann-MRT Method consists of several main stages. These include the initialization of distribution functions, the streaming step, the collision step, and the applied boundary conditions. For the first step of initializing the distribution functions, the initial guess for the distribution function f_i can be set to f_i^{eq} at each lattice node:

$$f_i(x, t) = f_i^{eq}(x, t)$$

The Equilibrium Distribution function for fluid simulations can be defined as follows:

$$f_i^{eq}(x, t) = w_i \rho \left(1 + \frac{\mathbf{u} \cdot \mathbf{c}_i}{c_s^2} + \frac{(\mathbf{u} \cdot \mathbf{c}_i)^2}{2c_s^4} - \frac{\mathbf{u} \cdot \mathbf{u}}{2c_s^2} \right)$$

where w_i are weights, ρ is the local fluid density, \mathbf{u} is the fluid velocity, and c_s is the speed of sound. Then, we calculate the macroscopic variables of fluid density and velocity, from the distribution functions. We proceed with updating the distribution functions by applying the collision operator at the collision step:

$$f_i^*(x, t) = f_i(x, t) - M^{-1} S M [f_i(x, t) - f_i^{eq}(x, t)]$$

$$f_i^*(x, t) = f_i(x, t) - M^{-1} S [m_i - m_i^{eq}]$$

, where m_i are the moments calculated as follows:

$$m_i = M f_i, \text{ for } i = [0,8]$$

Regarding the D2Q9 model in the context of MRT, the M matrix is as follows:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & 2 \\ 4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & -2 & 0 & 2 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \\ 0 & 0 & -2 & 0 & 2 & 1 & 1 & -1 & -1 \\ 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \end{pmatrix}$$

and the M^{-1} matrix is as follows:

$$\begin{pmatrix} 1/9 & -1/9 & 1/9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/9 & -1/36 & -1/18 & 1/6 & -1/6 & 0 & 0 & 1/4 & 0 \\ 1/9 & -1/36 & -1/18 & 0 & 0 & 1/6 & -1/6 & -1/4 & 0 \\ 1/9 & -1/36 & -1/18 & -1/6 & 1/6 & 0 & 0 & 1/4 & 0 \\ 1/9 & -1/36 & -1/18 & 0 & 0 & -1/6 & 1/6 & -1/4 & 0 \\ 1/9 & 1/18 & 1/36 & 1/6 & 1/12 & 1/6 & 1/12 & 0 & 1/2 \\ 1/9 & 1/18 & 1/36 & -1/6 & -1/12 & 1/6 & 1/12 & 0 & -1/2 \\ 1/9 & 1/18 & 1/36 & -1/6 & -1/12 & -1/6 & -1/12 & 0 & 1/2 \\ 1/9 & 1/18 & 1/36 & 1/6 & 1/12 & -1/6 & -1/12 & 0 & -1/2 \end{pmatrix}$$

The nine velocity directions in two-dimensions are given as follows:

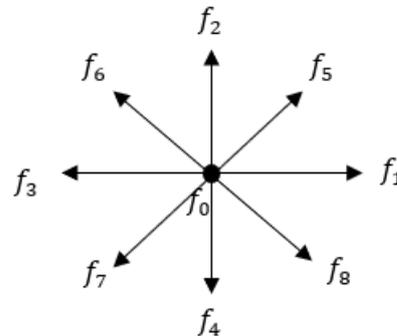
$$\begin{pmatrix} 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \end{pmatrix}$$

And the weights associated with the above directions are:

$$w_i = \begin{cases} \frac{4}{9}, & \text{for } i = 0 \\ \frac{1}{9}, & \text{for } i = [1,4] \\ \frac{1}{36}, & \text{for } i = [5,8] \end{cases}$$

Additionally, the streaming step takes place, where we propagate the distribution functions to neighbouring lattice nodes. The distribution functions at each lattice node $f_i(x, t)$ move to adjacent nodes in the direction of the corresponding velocity vector e_i , as shown in the following equation:

$$f_i(x + e_i * \Delta t, t + \Delta t) = f_i^*(x, t)$$



2) Reynolds number: bulk and dynamic viscosity

The Reynolds number, denoted as Re, is a fundamental concept in fluid dynamics that becomes particularly pivotal when simulating phenomena such as oil spills. This dimensionless quantity is defined by the formula: $Re = \rho U L / \mu$, where ρ represents the fluid's density, U is the characteristic flow velocity, L refers to a characteristic length (size of the oil spill), and μ is the dynamic viscosity of the fluid. Understanding and accurately applying the Reynolds number in simulations is critical for several reasons, especially when dealing with the complex interactions between oil, water, and possibly other substances present in an oil spill scenario.

The dynamic viscosity (μ) of a fluid is essentially a measure of its resistance to gradual deformation by shear or tensile stress. In the context of oil spills, this property significantly influences how the oil behaves once it enters the water. The higher the dynamic viscosity, the slower the oil tends to spread and mix with the water, affecting everything from the spill's initial dispersion to its long-term environmental impact. Bulk viscosity, on the other hand, is a measure of a fluid's resistance to compression. It becomes important in situations where the fluid undergoes rapid changes in volume or pressure. This can happen in turbulent conditions or when an oil spill interacts with obstacles, such as underwater topography or man-made structures.

One of the key applications of the Reynolds number in oil spill simulations is its ability to indicate the transition between laminar and turbulent flow regimes. Laminar flows are smooth and orderly, typically occurring at lower Reynolds numbers, while turbulent flows are chaotic and mixed, occurring at higher Reynolds numbers. Understanding this transition is crucial for predicting the behavior of an oil spill. For example, in laminar flow conditions, oil may spread in a relatively uniform manner, allowing for more straightforward containment measures. In contrast, turbulent conditions can lead to unpredictable dispersion patterns, making containment and clean-up efforts more challenging.

3) Conversion from physical to LB parameters and characteristic scales

Translating physical parameters into their Lattice Boltzmann (LB) equivalents is a critical process in the simulation of oil spills, to ensure a realistic representation of actual phenomena. This conversion is fundamental in the LB method, where physical phenomena are modelled in a lattice framework rather than continuous space.

In LB simulations, fluid properties like density and viscosity are expressed in lattice units, which necessitates their conversion from real-world values. The density in lattice units is normalized for simplicity. The viscosity is recalculated based on the lattice's relaxation time (τ) using the formula:

$$\mu = \rho c^2 * (\tau - 0.5)\Delta t$$

where c represents the speed of sound within the lattice framework, and Δt is the time step used. This equation bridges the gap between the macroscopic physical properties and their mesoscopic representations within the LB simulation. It ensures that the fluid's behavior in the simulation mirrors real-world behavior.

The scaling of velocity and length from their physical measurements to lattice units is also important for preserving the integrity of the simulation. This step involves adjusting the reference velocity and length scale from the real scenario to fit the discrete nature of the lattice framework, ensuring that the simulation remains faithful to physical reality. A crucial aspect of this conversion is maintaining the Mach number ($Ma = U/c$) significantly below 1, below 0.3, which is essential for the incompressibility assumption that underpins oil spill simulations. This assumption simplifies the simulations and is particularly relevant for flows at lower speeds, which are common in environmental and industrial applications.

C. Methodology of LB – MRT

1) Initial Conditions (Water and Crude Oil such as density and viscosity)

The computational domain that has been simulated is of size 40 x 40 LB nodes. The density of the crude oil was taken to be 800 kgm⁻³ and of the sea water 1000 kgm⁻³. An inlet velocity of 0.1 ms⁻¹ for the incoming sea wave was introduced on the west boundary at the beginning of the simulation, assuming that the overall initial seawater velocity was zero.

2) Collisions - Equilibrium Moments and Moments

The moment vector \mathbf{m} is as follows:

$$\mathbf{m} = (\rho, e, \varepsilon, j_x, q_x, j_y, q_y, P_{xx}, P_{xy})^T$$

The equilibrium moments m_i^{eq} are as follows:

$$m_i = \begin{cases} \rho \\ -2\rho + \frac{3(j_x j_x + j_y j_y)}{\rho} \\ \rho + \frac{3(j_x j_x + j_y j_y)}{\rho} \\ j_x \\ -j_x \\ j_y \\ -j_y \\ \frac{(j_x j_x - j_y j_y)}{\rho} \\ \frac{(j_x j_y)}{\rho} \end{cases}$$

The nine moments are defined as ρ being the fluid density, e is the energy, ε is related to the square of energy, j_x is the momentum of the fluid in the x-direction, q_x is moment related to flux in the x-direction, j_y is the momentum in the y-direction, q_y is

the moment related to flux in the y-direction, P_{xx} is the moment related to the stress-rate tensor term in the xx-direction and P_{xy} is the stress-rate tensor term related to the xy-direction. Moments ρ , j_x and j_y are the conserved moments, whereas e , ε , q_x , q_y , P_{xx} , P_{xy} are the non-conserved moments.

3) Relaxation Parameters

The relaxation parameter of the sea water is calculated by:

$$\omega_f = \frac{1}{3 * \nu_f * \frac{Ct}{Cl * Cl} + \frac{1}{2}}$$

, where ν_f is the kinematic viscosity, $\nu_f = 1e-6 \text{ m}^2\text{s}^{-1}$.

Relaxation parameter for the crude oil is:

$$\omega_a = \frac{1}{3 * D_a * \frac{Ct}{Cl * Cl} + \frac{1}{2}}$$

, where D_a is the diffusion coefficient of the crude oil, $D_a = 1e-5 \text{ m}^2\text{s}^{-1}$.

D. Boundary Conditions and Source Terms

Boundary conditions play a crucial role in ensuring both stability and precision in numerical solutions. In the Lattice Boltzmann method, it's essential to appropriately handle the discrete distribution functions at boundaries to accurately represent the macroscopic conditions of the fluid. For this purpose of oil-spill simulations, the outflow and Dirichlet inlet boundary conditions were used.

1) Dirichlet boundary conditions

In the context of oil spills, Dirichlet boundary conditions can be applied as inlet or no-slip conditions on solid surfaces to model the spread and behaviour of oil on the surface of water. In our oil spill simulations, the inflow boundary condition is crucial for realistically introducing seawater movement that will affect the conditions of an actual spill. By testing multiple inflow velocity conditions, it allows us for a detailed examination of how oil disperses and behaves in response to various environmental and physical sea wave input currents.

2) Neumann Boundary Conditions

Neumann boundary conditions are another type of boundary condition commonly used in mathematical modeling, including in the context of oil spills. Neumann boundary conditions specify the flux of a quantity across the boundary of the domain rather than directly specifying the value of the quantity at the boundary, as Dirichlet boundary conditions do and are used for slip boundary conditions.

3) Outflow Boundary Conditions

Outflow boundary conditions in oil spill simulations define how oil and contaminated water exit the computational domain, critically influencing the simulation's portrayal of oil movement and dispersion into larger bodies of water. Achieving realistic and reliable simulations hinges on correctly establishing these conditions, which govern the end-state dynamics of the modelled substances. The methodologies for implementing outflow boundary conditions can vary based on the simulation's specifics and the environmental characteristics being modelled. A common method involves specifying the gradients of variables such as velocity or oil concentration at the boundaries. This approach controls how these properties change as they transition out of the

simulation domain, ensuring that their variation reflects realistic outflow dynamics. Alternatively, convective outflow conditions might be employed, where the properties of the fluids at the boundary are set in accordance with the flow characteristics at those points. Outflow conditions hold particular importance in oil spill simulations, especially in open water contexts. The modeling of these conditions directly affects predictions regarding the spread and dispersion of oil as it moves beyond the initially impacted area. For instance, in simulations where oil spills into an ocean, setting appropriate outflow conditions at the domain boundaries can realistically simulate the spread of oil into unaffected regions. Accurately modeling this spread is critical for assessing the potential reach of the oil, evaluating possible impacts on distant ecosystems, and aiding the planning of extensive response and mitigation efforts. Insights gained from these outflow boundary conditions can help predict long-term environmental effects and guide comprehensive cleanup operations across expansive oceanic regions. Consequently, outflow boundary conditions are integral to comprehensive oil spill modeling, effectively linking localized spill dynamics with broader environmental and ecological outcomes. This connection is vital for developing strategic responses and understanding the far-reaching consequences of oil spills, thereby informing both immediate and long-term environmental protection efforts.

4) Source terms – Mass Flow Rate

We have successfully incorporated source terms in the LBM simulations including its formulation. In our case, the source term is the influx of crude oil in the sea water, originating from a spillage. We assume that an initial mass of oil is released and subsequently spreads and propagates due to diffusion and incoming advection sea waves respectively.

Formulation of source term involves the incorporation of distribution function terms for all velocity directions of the flow rate, multiplied by the corresponding weights. Then, each distribution function of the source flow rate is added as a separate term to the post-collision distribution functions, prior to streaming.

The Mass Flow Rate in these simulations, has been assigned the value of $1 \text{ kgm}^{-2}\text{s}^{-1}$ per dt (chosen time-step) over an area of 400 nodes, positioned in the centre of the domain, of size 20×20 .

E. LB Mesh - Partitioning

The effectiveness of the LBM in parallel computing environments is affected by the efficiency of the mesh partitioning algorithms. For this purpose, we delve into the intricacies of a specialized parallel partition algorithm software tailored for LBM mesh simulations. Our mesh partitioning algorithm is based on splitting nodes of the parent partition into contiguous partitions. The algorithm also manages the communication for shared nodes by taking advantage of the properties of parallel computing. This is achieved by efficiently distributing nodes across different processors, ensuring seamless communication and optimizing computational efficiency.

For our project, we utilized METIS, a set of partitioning graphs programs. The METIS algorithms are based on the multilevel recursive-bisection, multilevel k -way and multi-constraint partitioning schemes [39]. METIS partitions are better than those produced by spectral partitioning algorithms and faster than other widely used partitioning algorithms. METIS reduces the storage and computational requirements of sparse matrix factorization, and its elimination trees are suitable for parallel direct factorization.

A commonly used mesh partitioning library of METIS is *mpmetis*. However, we identified limitations in existing partitioning library in balancing the load of the partitions, therefore alternatives were utilized. These include *m2gmetis* for graph creation, and graph partitioning METIS (*gpmets*) for partitioning.

The graph partitioning METIS (*gpmets*) is a graph partitioning library that employs a multi-level approach [40]. The algorithm initially coarsens the graph, merging vertices to create a smaller representation, while preserving the structural properties of the graph. Then, a fast-partitioning algorithm is utilized to partition the coarsened graph. Additionally, it refines the partitioning by iteratively moving vertices between partitions to reduce the edge-cut, which is the number of edges connecting different partitions. Finally, the refined partitioning is projected to the original graph. *gpmets* utilizes sophisticated techniques to balance the load across partitions and minimize communication overhead, making it efficient for partitioning large graphs in various applications of computing and parallel processing. This strategic integration of tools optimizes the partitioning process, enabling efficient utilization of computational resources and accelerating simulation times.

F. Parallel Libraries: MPI, NCCL, and Thrust

Parallel computing libraries play a pivotal role in optimizing the performance and scalability of computational tasks in various domains, including high-performance computing (HPC), data analytics, and machine learning. The prominent parallel libraries employed in our project include Message Passing Interface (MPI), NVIDIA Collective Communications Library (NCCL), and Thrust (CUDA).

1) Message Passing Interface (MPI)

MPI is a standardized and widely used parallel programming model and library for distributed memory systems. It enables efficient communication and coordination among processes running on different compute nodes in a parallel computing environment. MPI provides a rich set of functionalities for point-to-point and collective communication, allowing developers to design and implement parallel algorithms for diverse applications. In our project, MPI facilitates seamless communication and data exchange between computational nodes, thereby enabling the efficient distribution of workload and parallelization of tasks.

2) NVIDIA Collective Communications Library (NCCL)

NCCL is a high-performance communication library specifically designed for NVIDIA GPUs and multi-GPU systems. It offers optimized collective communication operations, such as all-gather, all-reduce, and broadcast, tailored for GPU-accelerated computations. NCCL leverages the underlying hardware architecture of NVIDIA GPUs to achieve high throughput and low latency communication, making it well-suited for e.g. deep learning training or GPU-accelerated applications. In our project, NCCL enhances the scalability and performance of parallel algorithms by efficiently utilizing the computational power of multiple GPUs within a distributed system.

3) Thrust:

Thrust is a parallel algorithms library of CUDA provided by NVIDIA, offering a collection of high-level abstractions and optimized primitives for parallel computing on GPUs. It simplifies the

development of parallel algorithms by providing familiar interfaces and idioms inspired by the C++ Standard Template Library (STL). While thrust does not directly handle communication between GPUs like NCCL, it supports a wide range of parallel operations, including sorting, scanning, and reduction, enabling developers to express complex parallel computations concisely and efficiently. In our project, thrust provides a high-level interface for expressing parallel algorithms on NVIDIA GPUs, leveraging its optimized primitives and algorithms to achieve superior performance and scalability.

In summary, the integration of MPI, NCCL, and Thrust in our project empowers us to leverage parallel computing techniques effectively, optimizing performance, scalability, and efficiency across distributed and GPU-accelerated computing environments. The execution of parallel algorithms offers a significantly time-efficient solution using computational advancements in our research endeavours.

IV. RESULTS/ DISCUSSION

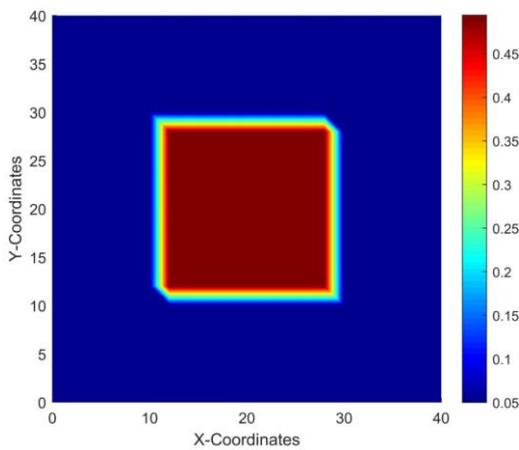


Figure 1. Density of the oil at the time of the deployment of the oil spill as a square.

In the initial phase of the simulation, a square oil spill profile was introduced at the center of the domain, amid the advancing water wave traveling from left to right. As illustrated in Fig. 1, there was a notable spike in the total system density upon the deployment of the oil spill, signifying the infusion of oil into the previously uniform water medium. This abrupt density increase serves as a clear indication of the initiation of the oil spill event within the simulated environment. Subsequent dynamics observed in the simulation, characterized by the gradual dispersion of the oil spill towards the right boundary of the domain under the influence of the water flow, are mirrored in the evolving density patterns depicted in the subsequent figures.

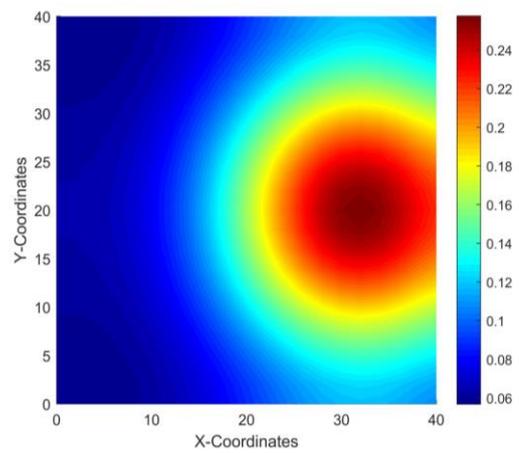


Figure 2. Oil density distribution after being transported by the wave and undergoing diffusion, resulting in its radial spread from the center

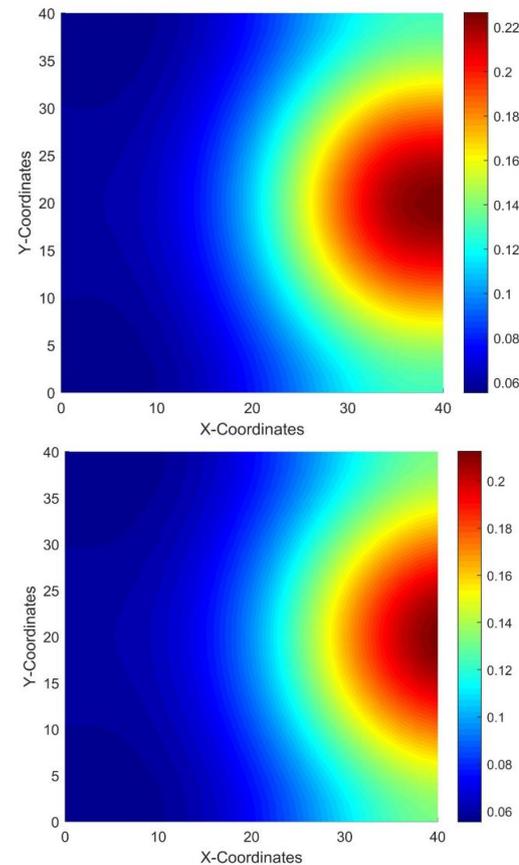


Figure 3. The transformation of the oil density profile from a square shape to a more circular and slightly elliptical distribution as it travels towards the right boundary. This evolution signifies the spreading and smoothing of the oil spill due to the combined effects of advection and diffusion processes.

As the simulation progresses, we witness the interaction between fluid dynamics and diffusion, influencing the movement and diffusion of the oil density within the domain. Initially characterized by a well-defined square profile, depicted in Fig. 2, the oil density gradually diffuses and spreads under the influence of the flowing water. This phenomenon, governed by the advection-diffusion equation solved in our model, is indicative of the lateral spread of the oil density towards the domain's right boundary. Figure 3 illustrate the evolving density distribution as the oil propagates from left to right.

Throughout the simulation duration, the combined effects of advection and diffusion play pivotal roles in shaping the spatial evolution and spreading dynamics of the oil spill. The advection-driven movement of the oil density, coupled with diffusion-induced homogenization, leads to a gradual dispersion and attenuation of the initially well-defined oil profile. Such behaviour is consistent with the diffusion equation and reflects the expected outcome in real-world oil spill scenarios. Importantly, the profile of the oil density becomes circular and then slightly elliptical during its propagation to the right, a phenomenon evident from the observed patterns in Fig. 3. By accurately simulating these phenomena, our model demonstrates its capability to capture the intricacies of oil spill dynamics and provides valuable insights for understanding and managing such environmental events effectively.

V. CONCLUSIONS/FUTURE WORK

A comprehensive simulation framework was developed capable of predicting the behaviour of oil spills with a high degree of accuracy. Throughout this project, we have leveraged a combination of computational fluid dynamics (CFD) methods to create a versatile and robust modeling platform. By integrating techniques such as the Lattice Boltzmann method with Multiple Relaxation Time (LB-MRT), we have achieved a level of sophistication that allows for the prediction of oil spill dynamics under varying environmental conditions.

Summing up, state-of-the-art methods, algorithms and techniques were implemented in KYAMOS software for simulating the oil spills. The LB-MRT model was implemented in software programming language C++ and was validated against well-known benchmarks. After being validated, it was applied to a real oil-spill simulation model. Having validated the oil-spill model, the solver was converted to state-of-the-art, CUDA aware MPI protocol for increased speed and accuracy. In future work, multiple results will be produced and used as input to a deep-learning process for training and validation of the AI model, that will be fine-tuned to instantly predict oil-spill simulations.

Moreover, this project represents a significant step forward in the ongoing effort to develop proactive measures for preventing and mitigating the impact of oil spills. By providing decision-makers with timely and accurate information, our simulations empower them to make informed choices that can minimize harm to ecosystems and coastal communities. Utilizing the above methods, we have achieved a model that not only captures the complex fluid dynamics involved but also provides real-time insights crucial for effective decision-making in spill response efforts.

VI. ACKNOWLEDGMENT

This work was co-funded by the European Regional Development Fund of the European Union and the Republic of Cyprus through the Research and Innovation Foundation (Project: CONCEPT-OILSPILLS/0722/0066).

REFERENCES

[1] "Deepwater Horizon – BP Gulf of Mexico Oil Spill | US EPA." US EPA, 14 Aug. 2023, www.epa.gov/enforcement/deepwater-horizon-bp-gulf-mexico-oil

- spill#:~:text=On%20April%202020%2C%202010%2C%20the,of%20marine%20oil%20drilling%20operations.
- [2] Z. Zhang, M. E. Kress, and T. Schäfer, "A lattice boltzmann advection diffusion model for ocean oil spill surface transport prediction," in 2020 Winter Simulation Conference (WSC), 2020, pp. 680-691: IEEE.
- [3] Z. Zhang, T. Schaefer, and M. E. Kress, "GNOME and LBM Model Evaluation on Ocean Oil Spill Far-Field Impacts to Highly Sensitive Areas," arXiv preprint arXiv:2105.05193, 2021.
- [4] A. Maslo, J. Panjan, and D. Žagar, "Large-scale oil spill simulation using the lattice Boltzmann method, validation on the Lebanon oil spill case," Marine pollution bulletin, vol. 84, no. 1-2, pp. 225-235, 2014.
- [5] P. Keramea, K. Spanoudaki, G. Zodiatis, G. Gikas, and G. Sylaios, "Oil Spill Modeling: A Critical Review on Current Trends, Perspectives, and Challenges," Journal of Marine Science and Engineering, vol. 9, no. 2, p. 181, 2021.
- [6] D. Li, X. Tang, Y. Li, X. Wang, and H. Zhang, "Mathematical Modeling of Marine Oil Spills in the Luanjiakou District, near the Port of Yantai," Discrete Dynamics in Nature and Society, vol. 2018, p. 2736102, 2018/01/17 2018.
- [7] Y. Li and P. Huang, "A coupled lattice Boltzmann model for the shallow water-contamination system," International journal for numerical methods in fluids, vol. 59, no. 2, pp. 195-213, 2009.
- [8] Z. Zhang, T. Schaefer, and M. E. Kress, "9.5 LATTICE BOLTZMANN METHOD FOR OCEAN OIL SPILL PROPAGATION MODEL AND SIMULATION."
- [9] S. Ha, N. Ku, and K.-Y. Lee, "Lattice Boltzmann Simulation For the Prediction of Oil Slick Movement And Spread In Ocean Environment," in The Twenty-second International Offshore and Polar Engineering Conference, 2012, vol. All Days, ISOPE-I-12-155.
- [10] L. Li, R. Mei, and J. F. Klausner, "Lattice Boltzmann models for the convection-diffusion equation: D2Q5 vs D2Q9," International Journal of Heat and Mass Transfer, vol. 108, pp. 41-62, 2017.
- [11] M. K. Banda and M. Seaid, Lattice Boltzmann simulation for shallow water flow applications. INTECH Open Access Publisher, 2012.
- [12] Fingas, Mervin. Oil Spill Science and Technology. Gulf Professional Publishing, 2016.
- [13] Fingas, M.; Brown, C. Review of Oil Spill Remote Sensing. Mar. Pollut. Bull. 2014, 83, 9–23.
- [14] Fingas, M.; Brown, C.E. A Review of Oil Spill Remote Sensing. Sensors 2018, 18, 91.
- [15] Alpers, W.; Holt, B.; Zeng, K. Oil Spill Detection by Imaging Radars: Challenges and Pitfalls. In Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Fort Worth, TX, USA, 23–28 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1522–1525.

- [16] Topouzelis, K.N. Oil Spill Detection by SAR Images: Dark Formation Detection, Feature Extraction and Classification Algorithms. *Sensors* 2008, 8, 6642–6659.
- [17] Brekke, C.; Solberg, A.H.S. Oil Spill Detection by Satellite Remote Sensing. *Remote Sens. Environ.* 2005, 95, 1–13.
- [18] Gens, R. Oceanographic Applications of SAR Remote Sensing. *Glsci. Remote Sens.* 2008, 45, 275–305.
- [19] Leifer, I.; Lehr, W.J.; Simecek-Beatty, D.; Bradley, E.; Clark, R.; Dennison, P.; Hu, Y.; Matheson, S.; Jones, C.E.; Holt, B.; et al. State of the Art Satellite and Airborne Marine Oil Spill Remote Sensing: Application to the BP Deepwater Horizon Oil Spill. *RemotemSens. Environ.* 2012, 124, 185–209.
- [20] Topouzelis, K.N.; Psyllos, A. Oil Spill Feature Selection and Classification Using Decision Tree Forest on SAR Image Data. *ISPRS J. Photogramm. Remote Sens.* 2012, 68, 135–143.
- [21] Li, K.; Yu, H.; Xu, Y.; Luo, X. Detection of Marine Oil Spills Based on HOG Feature and SVM Classifier. *J. Sens.* 2022, 2022, 3296495.
- [22] Dong, Z.-M.; Guo, L.-X.; Zeng, J.-K.; Zhou, X.-B. Oil-Spills Detection in Net-Sar Radar Images Using Support Vector Machine. *Open Autom. Control. Syst. J.* 2015, 7, 1958–1962.
- [23] Conceição, M.R.A.; Mendonça, L.F.F.; Lentini, C.A.D.; Lima, A.T.C.; Lopes, J.M.; Vasconcelos, R.N.; Gouveia, M.B.; Porsani, M.J. Sar Oil Spill Detection System through Random Forest Classifiers. *Remote Sens.* 2021, 13, 2044.
- [24] Lentini, C.A.D.; de Mendonça, L.F.F.; Conceição, M.R.A.; Lima, A.T.C.; de Vasconcelos, R.N.; Porsani, M.J. Comparison between Oil Spill Images and Look-Alikes: An Evaluation of SAR-Derived Observations of the 2019 Oil Spill Incident along Brazilian Waters. *An. Acad. Bras. Cienc.* 2022, 94, 1.
- [25] Vasconcelos, R.N.; Lentini, C.A.D.; Cunha Lima, A.T.; Mendonça, L.F.F.; Miranda, G.V.; Cambuí, E.C.B.; Costa, D.P.; Duverger, S.G.; Gouveia, M.B.; Lopes, J.M.; et al. Oil Spill Detection Based on Texture Analysis: How Does Feature Importance Matter in Classification? *Int. J. Remote Sens.* 2022, 43, 4045–4064.
- [26] Del Frate, F.; Petrocchi, A.; Lichtenegger, J.; Calabresi, G. Neural Networks for Oil Spill Detection Using ERS-SAR Data. *IEEE Trans. Geosci. Remote Sens.* 2000, 38, 2282–2287.
- [27] Krestenitis, M.; Orfanidis, G.; Ioannidis, K.; Avgerinakis, K.; Vrochidis, S.; Kompatsiaris, I. Oil Spill Identification from Satellite Images Using Deep Neural Networks. *Remote Sens.* 2019, 11, 1762.
- [28] Garcia-Pineda, O.; MacDonald, I.R.; Li, X.; Jackson, C.R.; Pichel, W.G. Oil Spill Mapping and Measurement in the Gulf of Mexico with Textural Classifier Neural Network Algorithm (TCNNA). *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 2013, 6, 2517–2525.
- [29] Singha, S.; Bellerby, T.J.; Trieschmann, O. Satellite Oil Spill Detection Using Artificial Neural Networks. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 2013, 6, 2355–2363.
- [30] Yang, Y.-J.; Singha, S.; Mayerle, R. A Deep Learning Based Oil Spill Detector Using Sentinel-1 SAR Imagery. *Int. J. Remote Sens.* 2022, 43, 4287–4314.
- [31] Al-Ruzouq, R.; Gibril, M.B.A.; Shanableh, A.; Kais, A.; Hamed, O.; Al-Mansoori, S.; Khalil, M.A. Sensors, Features, and Machine Learning for Oil Spill Detection and Monitoring: A Review. *Remote Sens.* 2020, 12, 3338.
- [32] Shaban, M.; Salim, R.; Abu Khalifeh, H.; Khelifi, A.; Shalaby, A.; El-Mashad, S.; Mahmoud, A.; Ghazal, M.; El-Baz, A. A Deep-Learning Framework for the Detection of Oil Spills from SAR Data. *Sensors* 2021, 21, 2351.
- [33] Jha, M.N.; Levy, J.; Gao, Y. Advances in Remote Sensing for Oil Spill Disaster Management: State-of-the-Art Sensors Technology for Oil Spill Surveillance. *Sensors* 2008, 8, 236–255. [CrossRef]
- [34] Huby, A.A.; Sagban, R.; Alubady, R. Oil Spill Detection Based on Machine Learning and Deep Learning: A Review. In *Proceedings of the 2022 5th International Conference on Engineering Technology and its Applications (IICETA)*, Al-Najaf, Iraq, 31 May–1 June 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 85–90.
- [35] Cresson, R. *Deep Learning for Remote Sensing Images with Open Source Software*, 1st ed.; CRC Press: Boca Raton, FL, USA, 2020; Volume 1, ISBN 9781003020851.
- [36] Bao, Yuanxun & Meskas, J., *Lattice Boltzmann Method for Fluid Simulations*, ResearchGate, 2014
- [37] T. Krüger et al. *The Lattice Boltzmann Method Principles and Practice*, Springer, Graduate Texts in Physics, 2016.]
- [38] Shan, Xiaowen, and Hudong Chen. "Lattice Boltzmann Model for Simulating Flows With Multiple Phases and Components." *Physical Review E*, vol. 47, no. 3, American Physical Society (APS), Mar. 1993, pp. 1815–19. Crossref, <https://doi.org/10.1103/physreve.47.1815>.]
- [39] METIS - Serial Graph Partitioning and Fill-reducing Matrix Ordering, Karypis Lab. glaros.dtc.umn.edu/gkhome/metis/metis/overview.]
- [40] Gpmetis: Partitions a Graph Into a Specified Number of Parts. Metis Commands, Man Pages, Mankier. www.mankier.com/1/gpmetis