



NUMERICAL METHODS IN FLUID MECHANICS – AN OVERVIEW

Wojciech Sobieski¹, Božidar Šarler²

¹ORCID: 0000-0003-1434-5520

Department of Mechanical Engineering and Fundamentals of Machine Design
University of Warmia and Mazury in Olsztyn

²ORCID: 0000-0003-0618-6873

Faculty of Mechanical Engineering
University of Ljubljana, Ljubljana, Slovenia

Received 21 July 2023, accepted 9 November 2023, available online 10 November 2023.

Key words: CFD, numerical modelling, theoretical basics, mesh methods, meshless methods.

Abstract

The article presents in a review way the most important numerical methods used in modern fluid mechanics. The individual chapters discuss Finite Difference Method, Finite Volume Method, Lattice Boltzmann Method, Discrete Element Method and Smoothed Particle Hydrodynamics. The aim of the article is to familiarize the reader with the most important concepts, features and mathematical equations used in particular methods. The article is intended mainly for people who want to get acquainted with the current possibilities of numerical modelling in the field of broadly understood fluid mechanics. The material is intended to facilitate the decision on how to implement the planned research.

Introduction

In the field of science, regardless of the specific discipline, numerical modelling has become one of the essential methods for investigating or predicting the behaviour of various systems. The primary challenge is the extensive knowledge

Correspondence: Wojciech Sobieski, Katedra Mechaniki i Podstaw Konstrukcji Maszyn, Uniwersytet Warmińsko-Mazurski, ul. Oczapowskiego 11, 10-736 Olsztyn, e-mail: wojciech.sobieski@uwm.edu.pl

required to effectively utilize numerical modelling. Over the years, a wide range of techniques has been developed to address various classes of problems.

A typical numerical modelling process consists of the following stages:

- preparation of the numerical model for calculations (pre-processing):
 - a) determining the geometry of the computational domain, b) discretizing the computational domain, c) selecting an appropriate mathematical model, d) defining a closure set, e) specifying properties of materials used, f) setting boundary conditions, g) establishing initial conditions;
- performing calculations (solving):
 - a) defining numerical techniques, b) determining monitoring parameters for the calculation process, c) configuring recording settings, d) defining termination criteria for the calculations, e) executing simulations and monitoring the progress;
- processing and analysis of the resulting data (post-processing):
 - a) visualization of the simulation results, b) in-depth analysis of the data, c) data processing to extract meaningful information and insights.

In the context of numerical analysis, a computational domain signifies a virtual representation of the geometry, encompassing either its entirety or a partial depiction of a real-world system. This representation is custom-tailored to suit the specific demands of numerical simulations. The term “closure” in numerical modelling refers to a mathematical model that elucidates a specific phenomenon or process. It augments the fundamental set of mathematical equations, enabling them to be effectively solved. The 13 stages of numerical modelling may seem straightforward at first glance, but each stage entails a wealth of extensive and intricate knowledge. Consider, for instance, the process of discretizing the computational domain. This process encompasses various methods, including the utilization of nodes, pixels, voxels, elements, or cells. These objects can form structured or unstructured meshes and may be node-centered, cell-centered, multi-nodal, or multi-cellular. Moreover, the selection of discretization can be further categorized into structured, unstructured, hybrid, multi-block, multi-zone, overlapping, adaptive, movable (with rotary or reciprocating movement), and various other techniques. Each type of domain discretization necessitates distinct generation techniques and algorithms, each characterized by its unique aspects.

Another example is the algorithm required for solving sets of linear equations, a common necessity during the solving stage. While developing numerical methods, it became apparent that classical methods typically described in textbooks were inadequate. New, significantly more intricate techniques had to be devised, such as relaxation methods, methods founded on Krylov space, or multi-grid methods. Another challenge was determining the most effective way to implement these methods. The implementation process, in turn, drove the advancement of parallel programming and computational techniques that could harness the

power of graphics processors. It is evident that contemporary numerical modelling represents a fusion of knowledge from various fields, including mathematics, physics, algorithmic, and programming.

It is important to acknowledge that various numerical methods were developed during different eras, each with its own set of technological circumstances, including operating systems and programming languages. Over the years, programming languages like FORTRAN and C were commonly utilized, and later on, C++ gained popularity. In the present day, Python has also emerged as a widely-used language. Python can be employed directly in the implementation of numerical methods or indirectly as an environment that connects various software or libraries, often written in FORTRAN, C, or C++. These languages continue to dominate the field of numerical modelling due to their high-speed computational capabilities, which make them well-suited for complex mathematical operations.

All of the factors mentioned above, when considered collectively, make it challenging for a single researcher to master all the numerical techniques currently in use. Typically, researchers tend to specialize in one particular numerical method, especially when their research is focused on a specific type of matter, such as solids, fluids, or granular materials. However, in cases involving multiphase systems that demand the integration of different numerical techniques, researchers may employ multiple methods. This high degree of specialization often results in limited knowledge of alternative approaches, even those that could be potentially applied within the same research area. It is important to note that this is a generalized perspective and may not account for unique situations or exceptions.

Having a broader understanding allows for more strategic planning and the more efficient allocation of resources to achieve specific objectives. The primary goal of the research described in this paper is to introduce selected numerical methods, which are frequently employed in the realm of Computational Fluid Dynamics (CFD). This introduction encompasses the fundamental principles underlying these methods, the primary equations they employ, their key characteristics, areas of application, as well as their strengths and limitations. The content is designed to cater to two primary groups of readers: those who are new to numerical modelling and those who possess some experience in one or two specific areas and wish to expand their knowledge in this domain.

It should be emphasized that the starting point for the considerations presented in the article is the Finite Volume Method, which is described in the most detail for this reason. Other methods either serve as potential alternatives (such as the Lattice Boltzmann Method and the Smoothed Particle Hydrodynamics) or complement it (such as the Discrete Element Method). The Finite Difference Method is described because it is historically the first numerical method used

in fluid mechanics and is still used there in some areas today. Some of the methods described in the article, especially the Discrete Element Method and the Smoothed Particle Hydrodynamics, can be used to model fluid-solid systems or materials located at the interface between solids and fluids. In this area, they often serve as better tools than methods typically dedicated to fluids. A key to understanding the choice of numerical methods described in the article is presented further in Figure 5, summarizing the strategies for modelling multiphase media. All the numerical methods described in the article can be used independently or in combination to implement one of the visible variants. It should be noted that the aforementioned figure does not include solid matter, which explains why the Finite Element Method is not described in the article. Of course, there are issues where this method is coupled with a method designed for fluids. Examples include fluid flow in flexible pipes or shells, fluid interaction with deformable membranes, etc. It should be added that the Finite Element Method can also be used to directly solve flow problems, as evidenced by the capabilities of programs such as SolidWorks, COMSOL, or Abaqus, but it has never been the focus of the authors, who consider the Finite Volume Method to be more suitable for modelling fluid flows than the Finite Element Method. The article only describes numerical methods with which the authors had direct practical experience, not just theoretical knowledge. For the same reason, many other methods or their variants, such as the Discontinuous Galerkin Method, the Boundary Element Method, or the Immersed Boundary Method, are not described. For those interested in the application of the Finite Element Method in the analysis of fluid flow problems, it is recommended to explore publications such as HUGHES and ZIENKIEWICZ (1979), WEINAN and LIU (2000) or ZIENKIEWICZ et al. (2005).

Finite Difference Method

The Finite Difference Method (FDM) (COURANT et al. 1928, CAUSON, MINGHAM 2010, LANGTANGEN, LINGE 2017) is an Eulerian method that involves approximating the derivative of a function using finite differences. It relies on a previously defined spatial discrete mesh of nodes. The starting point is the Newton-Raphson method, where the derivative of the function at a point equals the slope of the tangent line to the function (curve) at that point (IYENGAR, JAIN 2009, BURDEN, FAIRES 2011). This concept is illustrated in Figure 1. However, this definition is not unambiguous because it can be implemented in at least three ways: the so-called forward scheme (1), the backward scheme (2), or the central scheme (3):

$$f'(x) \approx \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} \tag{1}$$

$$f'(x) \approx \frac{f(x_0) - f(x_0 - \Delta x)}{\Delta x} \tag{2}$$

$$f'(x) \approx \frac{f(x_0 + \Delta x) - f(x_0 - \Delta x)}{2\Delta x} \tag{3}$$

In equations (1-3), only three nodes are utilized ($i - 1, i, i + 1$). However, higher-order numerical schemes can also be structured, involving 5 or more nodes to approximate a derivative. The second derivative of the function $f(x)$ at the point x_0 is obtained by applying the difference scheme three times. Depending on the schemes employed, there can be multiple variations of the second derivative. It should be noted that if the equation includes a time-dependent term, time must also be discretized. The maximum allowable time step depends on the numerical scheme utilized and is sometimes constrained by additional parameters, primarily the CFL condition.

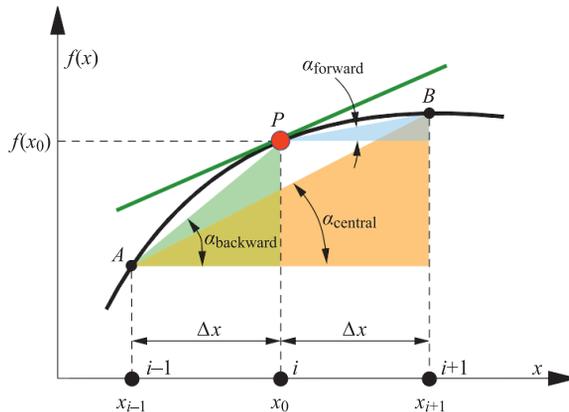


Fig. 1. Illustration to the main idea of the FDM

An excellent example of the FDM can be found in the one-dimensional thermal diffusion equation (LANGTANGEN, LINGE 2017):

$$\frac{\partial T}{\partial t} - \frac{\lambda}{\rho c_s} \frac{\partial^2 T}{\partial x^2} = 0 \tag{4}$$

where:

- T – temperature [K],
- t – time [s],
- λ – thermal conductivity [W/(m·K)],
- ρ – density [kg/m³],
- c_s – specific heat [J/(kg·K)],
- x – coordinate [m].

By applying the central difference scheme for the second derivative and the explicit Euler scheme for time discretization, equation (4) takes the following form (LANGTANGEN, LINGE 2017):

$$T_i^{n+1} = T_i^n + \frac{\lambda}{\rho c_s} \frac{T_{i-1}^n - 2T_i^n + T_{i+1}^n}{(\Delta x)^2} \Delta t \quad (5)$$

where:

n and $n + 1$ represent the current and the next time levels, respectively.

Equation (5) is valid for nodes from to $N - 1$, where N is the number of nodes in the numerical grid. Boundary conditions need to be specified for the extreme nodes. For instance, if equation (4) is assumed to describe the heat flow through a single-layer homogeneous wall with Newton's law of cooling on both sides, then:

$$T_1^{n+1} = \frac{\lambda T_2^{n+1} + \alpha \Delta x T_l}{\alpha \Delta x + \lambda} \quad \text{and} \quad T_N^{n+1} = \frac{\lambda T_{N-1}^{n+1} + \alpha \Delta x T_r}{\alpha \Delta x + \lambda} \quad (6)$$

where:

- α – heat transfer coefficient [W/(m²·K)],
- T_l – outside temperature on the left side of the wall [K],
- T_r – outside temperature on the right side of the wall [K].

It soon became evident that the FDM is well-suited for solving relatively simple differential equations (examples shown in Fig. 2). However, the method proved to be inadequate for both solid and fluid mechanics applications. Its primary limitations included challenges in adapting the mesh to complex geometries, issues related to calculation stability, difficulties in handling boundary conditions, often resulting in a loss of accuracy, the necessity for uniform grid spacing, and the associated requirement for a large number of nodes. These challenges prompted the search for alternative numerical modelling methods that could better meet the specific needs of various physics departments. In modern Computational Fluid Dynamics, the FDM is rarely employed. Nevertheless, it holds a significant place in the history of numerical modelling, representing an important milestone in the development of simulation techniques.

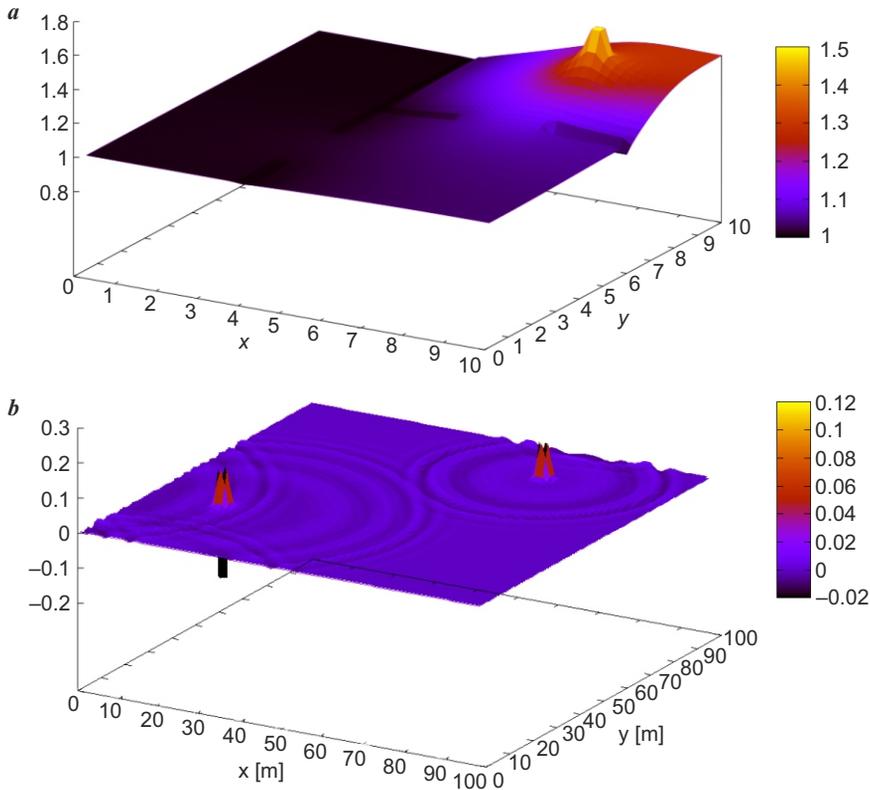


Fig. 2. Examples of using the FDM: *a* – resolving the diffusion equation in 2D space and modelling of the spread of smell in the layout of laboratory rooms, *b* – resolving the wave equation in 2D and modelling of waves propagation on a surface with random falling drops

Finite Volume Method

The Finite Volume Method (FVM) (MCDONALD 1971, MACCORMACK, PAULLAY 1972, MOUKALLED et al. 2016) is designed for modelling fluid flows within a discretized computational domain (Ω) using the Eulerian approach. This method relies on a mesh-based approach, where a fundamental role is played by the concept of the Finite Volume or Control Volume (CV). A Control Volume is a volume $V \in \Omega$ enclosed by a closed surface S . The orientation of this surface is defined by the outward-directed vector \vec{n} (as depicted in Fig. 3). In numerical models, control volumes represent the cells of the computational grid. The FVM is based on two types of balance equations: surface and volumetric balance equations.

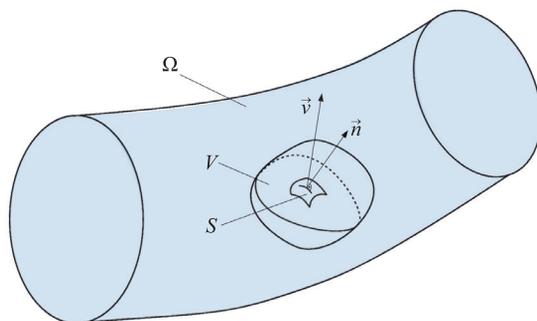


Fig. 3. Schematic representation of a Control Volume in a calculation domain

The surface balance describes the possibility to exchange the value of with the surroundings through the fluxes flowing through the surface:

$$\left. \frac{\partial \Theta}{\partial t} \right|_S = \int_S (\Theta \vec{v} \cdot \vec{n}) dS \quad (7)$$

where:

Θ – any physical quantity, e.g. mass, momentum or energy,

\vec{v} – fluid velocity,

\vec{n} – surface directional vector.

An example of a quantity that requires surface balancing is heat flux. The resulting value is determined by summing up the flows entering (with a negative sign) and exiting (with a positive sign) through the surface S of a Control Volume (CV).

The volumetric balance equation describes the ability to change the value of Θ within the volume V :

$$\left. \frac{\partial \Theta}{\partial t} \right|_V = \int_V (s_\Theta) dV \quad (8)$$

where:

s_Θ – source of the Θ quantity.

An example of a quantity that requires volumetric balancing is the heat generated in a combustion process. In this case, the source term has a positive sign, but in general, a source term may increase or decrease the value of the balance quantity in a control volume, depending on its physical meaning. After applying the surface and volume balances to the entire computational mesh within the domain Ω , the following set of balance equations is obtained (MOUKALLED et al. 2016):

$$\frac{\partial}{\partial t}(\rho) + \text{div}(\rho\vec{v}) = 0 \tag{9}$$

$$\frac{\partial}{\partial t}(\rho\vec{v}) + \text{div}(\rho\vec{v}\vec{v}) = \text{div}(-p\vec{I} + \vec{\tau}^l + \vec{\tau}^t) + \rho\vec{s}_m \tag{10}$$

$$\frac{\partial}{\partial t}(\rho e) + \text{div}(\rho\vec{v}e) = \text{div}[(-p\vec{I} + \vec{\tau}^l + \vec{\tau}^t)\vec{v} + \vec{q}^l + \vec{q}^t] + \rho\vec{s}_e \tag{11}$$

where:

- ρ – density [kg/m³],
- p – pressure [Pa],
- \vec{I} – unit tensor [-],
- $\vec{\tau}^l$ – laminar stress tensor [Pa],
- $\vec{\tau}^t$ – turbulent stress tensor [Pa],
- $\rho\vec{s}_m$ – momentum source term [N/m³],
- e – unitary energy (a sum of internal and kinetic energies) [J/kg],
- \vec{q}^l – laminar heat flux [J/(m²·s)],
- \vec{q}^t – turbulent heat flux [J/(m²·s)],
- $\rho\vec{s}_e$ – energy source term [J/(m³·s)].

Equations (9-11) represent the Mass Balance Equation (MaBE), the Momentum Balance Equation (MoBE), and the Energy Balance Equation (EBE), respectively. These equations can be expressed in what is known as vector form:

$$\frac{\partial}{\partial t} \begin{bmatrix} \rho \\ \rho\vec{v} \\ \rho e \end{bmatrix} + \text{div} \begin{bmatrix} \rho\vec{v} \\ \rho\vec{v}\vec{v} + p\vec{I} \\ \rho\vec{v}e + p\vec{I}\vec{v} \end{bmatrix} = \text{div} \begin{bmatrix} 0 \\ \vec{\tau}^l + \vec{\tau}^t \\ (\vec{\tau}^l + \vec{\tau}^t)\vec{v} + \vec{q}_s^l + \vec{q}_s^t \end{bmatrix} + \begin{bmatrix} 0 \\ \rho\vec{s}_m \\ \rho\vec{s}_e \end{bmatrix} \tag{12}$$

where the similarities between specific balance equations become much more apparent (please note that the terms containing pressure are now positioned on the left side).

The first part of equation (12) represents the time-dependent term, which describes the rate of change of the balanced quantity with respect to time. For steady-state flows this term is equal to zero. The second term in equation (12) is the so-called convection term, which describes the exchange of the balanced quantity between the current control volume (CV) and neighbouring cells. The components of this term are reversible, meaning they can be converted. An example is the transfer of pressure energy during compression and release during fluid expansion. The third term in equation (12) is the dissipative term, which describes irreversible phenomena. This encompasses all phenomena related to viscosity, turbulence, diffusion, radiation, etc. The final part of equation (12)

is the source term, which accounts for the possibility of changing the balance quantity within a CV, such as due to the influence of external forces.

The FVM is based on cell-centered meshes, which are typically structured or unstructured. The mesh is generated within the domain, and its boundaries coincide with the domain boundary. However, in the literature another approach may be found, the so-called Immersed Boundary Method (PESKIN 1977), in which the domain is immersed in the grid and does not conform with the boundary. In the immersed boundary method, special treatment has to be taken at the boundary to incorporate the boundary conditions. The main advantage of the Immersed Boundary Method is the use of structured meshes, which are easy to generate, and the calculations performed on them are faster. The greatest difficulty of the method is the imposition of boundary conditions.

The equations set (9-11) has to be supplemented by other mathematical dependencies (closures mentioned before). Two of them are in particular critical: the laminar stress tensor, in which the key role plays the choice of the appropriate fluid model (rheology is the branch of physics that deals with this issue) and the turbulent stress tensor, which can be described in many different ways. In fact, turbulence modelling is the most fundamental issue in the fluid mechanics still not fully solved.

In Figure 4, an overview of applied turbulence models is visible. Reynolds-Averaged Navier-Stokes (RANS) is a group of turbulent flow simulation methods based on the concept of decomposing the Reynolds velocity field, in which the

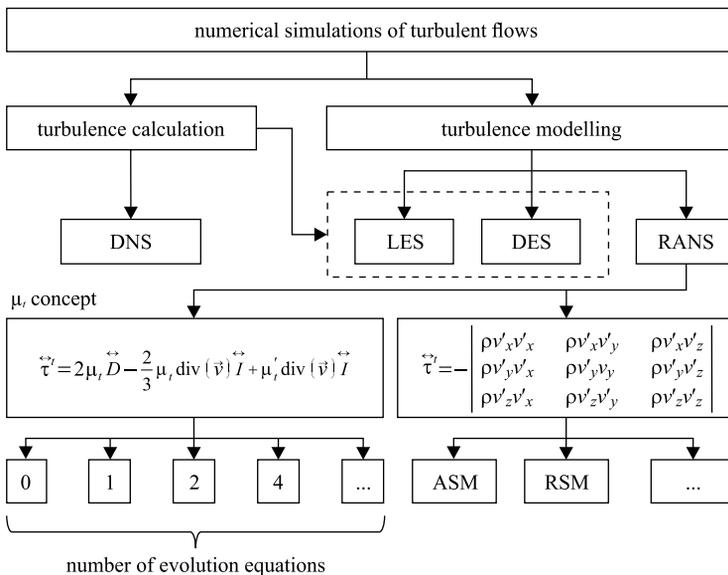


Fig. 4. Main concepts in turbulence modelling

velocity and pressure fields are separated into mean values and fluctuation components. In frames of this group two main trends can be distinguished:

- based on the concept of turbulent viscosity – in this approach, turbulent stresses are proportional to the deformation rate, analogous to viscous stresses (the turbulent stress tensor serves as a form of correction to the viscous stress tensor);

- not based on the turbulent viscosity concept – in this approach, the components of the turbulent stress tensor are calculated directly from additional equations, algebraic (ASM), differential (RSM) or evolution.

The evolution equations are added to the system of equations (12) (they must have the same structure) and are solved using the same numerical techniques. The evolution equations are not used to determine a new balance quantity but a variable necessary to solve the mass, momentum, or energy balance equation. Typically, evolution equations are needed to determine the turbulent stress tensor. Equations from the RSM group are not included in the set (12) and are solved independently of them.

Direct Numerical Simulation (DNS), as proposed by Orszag in 1970, is a method for simulating turbulent flows. It involves the direct solution of the Navier-Stokes equations without any simplifications, which means that all turbulence scales are considered in the calculations. Calculations performed with the DNS method enable the correct reconstruction of the dynamics of all spatial and temporal scales of turbulence, which, unlike the RANS methods, are not modelled, but are the result of a numerical solution. The DNS method requires enormous computing power and for this reason is still not applicable in practical issues.

Large Eddy Simulation (LES), as introduced by Smagorinsky in 1963, is a method for simulating turbulent flows. It involves the separation of vortices into two scales through the use of specific filters. The large-scale fluid movement is calculated based on the Navier-Stokes equations, while the small-scale structures (smaller than the width of the filter used), in which kinetic energy is dissipated by the action of viscous forces, are modelled. The smaller the filter width, the closer the LES method becomes to DNS.

Detached Eddy Simulation (DES), developed by Spalart in 1997, is a method for simulating turbulent flows that combines elements of both Large Eddy Simulation and Reynolds-Averaged Navier-Stokes. The RANS approach is applied in the boundary layers (where large vortices are absent), while the LES approach is employed in the core of the flow. Achieving a seamless transition of fields between the LES and RANS regions in DES necessitates the use of high-quality numerical grids.

It is currently believed that the most versatile model suitable for engineering modelling of typical fluid flows is the $k - \omega$ SST (MENTER 1993) model from the RANS group. This model accurately reflects phenomena both in the

boundary layer and far away from it. In some areas, other RANS models are recommended, such as the Spalart-Allmaras model, specifically designed for aerospace applications. LES or DES are used in cases where various turbulence scales occur, such as large coherent structures in vortex flows, which cannot be adequately modeled with the RANS approach.

The equations set (9-11) is valid for single-component fluids (SOBIESKI, GRZYGO 2019). However, with appropriate modifications and extensions, it can also be used for a wide range of multiphase flows. In Figure 5, the one-fluid approach means that the occurrence of certain phenomena may be predicted from scalar or vector fields obtained for a single-phase flow. An example of this is the barotropic cavitation model, in which the cavitation phenomenon is predicted based on the pressure field. Another way to introduce a second phase is by incorporating some effects of its presence. Good examples of this approach are the Porous Media Model or the Porous Jump Model, in which flow resistance is modelled by a source term in the momentum balance equation within a chosen volume or on a surface, respectively.

In the homogeneous approach, the mixture is considered as a substance composed of any number of phases and is modelled using only one set of balance equations (NIEDŹWIEDZKA et al. 2016). Mass or volume fractions of each individual phase are calculated using additional balance equations, which increases the number of variables in the vectors seen in equation (12). Depending on further details, the phase interface may or may not be tracked. The first case, known in the literature as the Volume of Fluid model, is used in simulations of fluid flows with a free surface or in flows involving two non-mixing liquids. Each of the components can be a single fluid or a homogeneous mixture of any number

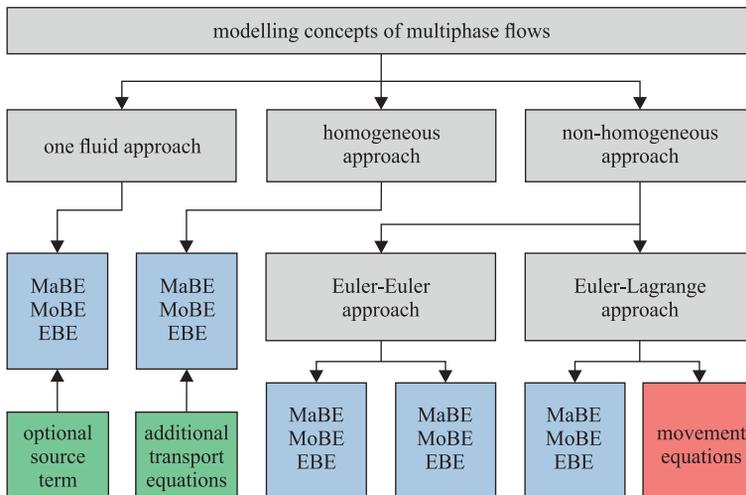


Fig. 5. Main concepts in modelling multiphase flows

of components. The homogeneous approach is typical for modelling selected mixing phenomena, flows involving evaporation and condensation, or flows with chemical reactions.

In the non-homogeneous approach, each component of the mixture has its own set of balance equations (SOBIESKI 2009). Phase coupling occurs through pressure and interfacial coefficients related to mass, momentum, and energy exchange. The description of this interaction primarily depends on the type of flow, whether it is liquid-liquid, liquid-gas, liquid-solid, or gas-solid. This approach

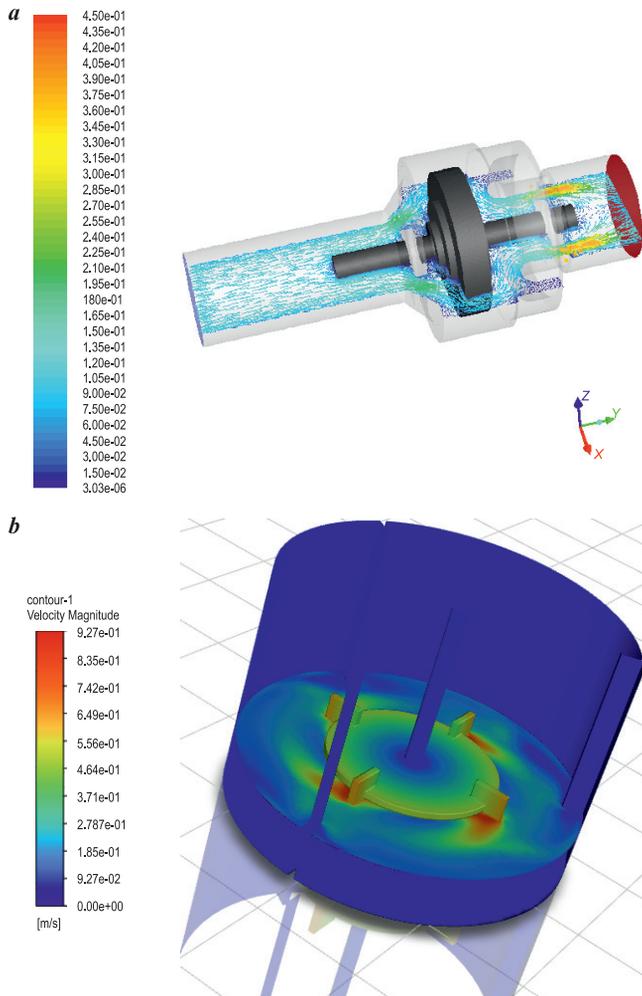


Fig. 6. Examples of using the Finite Volume Method: *a* – determination of the hydrodynamic thrust force acting on a valve plug, *b* – determination of mixing power in a mixer with a Rushton turbine

is primarily intended for flows with one background phase (liquid or gas) and one or more dispersed phases (bubbles, droplets, or solid particles). The fluid phase is described using the Eulerian approach, while the dispersed phase can be treated either in the Eulerian manner (as in the Eulerian Multiphase Model) or in the Lagrangian manner (as in the Discrete Phase Model). In the latter case, appropriate movement equations must be added to the model. In practice, this task is accomplished by coupling the FVM (or another method designed for fluids), responsible for managing the continuous phase, with the Discrete Element Method, utilized in this context for modelling the dispersed phase. The Discrete Element Method is likewise described in the article.

The FVM is highly popular, and examples of its application can be found in every field of modern science. Some examples from mechanical engineering are shown in Figure 6. The primary limitation of this method remains the insufficient computing power of currently available computers.

Lattice Boltzmann Method

The Lattice Boltzmann Method (LBM) (BHATNAGAR et al. 1954, SUKOP, THORNE 2006) is used to model the behaviour of a lattice gas in a state of thermodynamic imbalance, employing a Lagrangian approach. The foundational concept in the LBM is the phase space, where each dimension corresponds to a distinct physical quantity. In classical mechanics, the phase space typically encompasses position, velocity or momentum, and time (as shown in Fig. 7) (KRÜGER et al. 2017):

$$f(\vec{x}, \vec{v}, t) \tag{13}$$

where:

- f – distribution function of gas density in the phase space,
- \vec{x} – position vector (here in Cartesian coordinates),
- t – time.

The function $f(\vec{x}, \vec{v}, t)$ represents the probability of finding a given particle at a given location in the space \vec{x} and time t with a given velocity \vec{v} .

If there are no particle collisions, then during the movement in the phase space, all those particles that, at time $t + dt$, are in the phase space element described by the coordinates $\vec{x} + d\vec{x}$ and $\vec{v} + d\vec{v}$, were at time t within the phase space element with the coordinates \vec{x} and \vec{v} (SUCCI 2001, SUKOP, THORNE 2006, MOHAMAD 2011):

$$f(\vec{x} + d\vec{x}, \vec{v} + d\vec{v}, t + dt) = f(\vec{x}, \vec{v}, t) \tag{14}$$

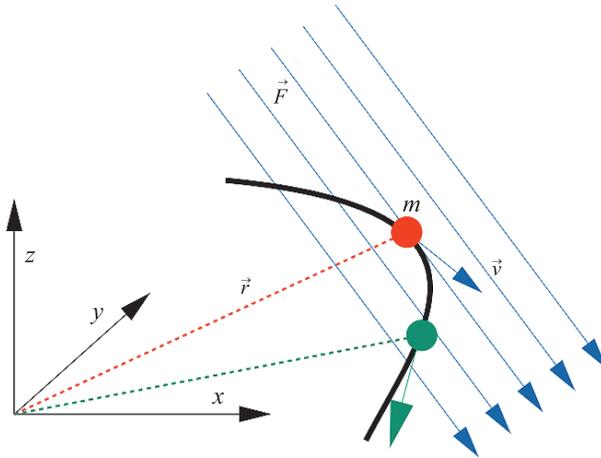


Fig. 7. Schematic representation of the phase space

Applying Newton’s laws of motion, this equation can be written as

$$f\left(\vec{x} + \vec{v}dt, \vec{v} + \frac{\vec{F}}{m}dt, t + dt\right) = f(\vec{x}, \vec{v}, t) \tag{15}$$

or

$$\frac{\partial f}{\partial t} + \vec{v} \cdot \nabla_{\vec{x}}f + \frac{\vec{F}}{m} \cdot \nabla_{\vec{v}}f = 0 \tag{16}$$

where:

- \vec{F} – external force acting on gas particles [N],
- m – mass of gas particles [kg].

If particle collisions occur, meaning that particles move not only away from a given point in space but also towards it, the derivative does not equal zero:

$$\frac{\partial f}{\partial t} + \vec{v} \cdot \nabla_{\vec{x}}f + \frac{\vec{F}}{m} \cdot \nabla_{\vec{v}}f = \left(\frac{\partial f}{\partial t}\right)_{\text{coll}} \tag{17}$$

The simplest model describing the collision term is the BGK linear model (BHATNAGAR et al. 1954):

$$\left(\frac{\partial f}{\partial t}\right)_{\text{coll}} = \frac{f - f^{\text{eq}}}{\tau} \tag{18}$$

where:

- τ – the so-called relaxation time.

This parameter determines how quickly the f function may obtain the equilibrium value, which depends on the assumed viscosity of the fluid. In turn, f^{eq} means the distribution function of gas density in the equilibrium state:

$$f^{\text{eq}} = \left[\frac{m}{2\pi k_B T} \right]^{\frac{3}{2}} e^{-\frac{1}{2} \frac{mv^2}{k_B T}} \quad (19)$$

where:

- T – absolute temperature [K],
- k_B – Boltzmann constant [J/K].

If the values of the distribution function $f(\vec{x}, \vec{v}, t)$ in each i^{th} direction are known, different macroscopic parameters may be calculated. For example the density and velocity of lattice gas may be determined with the following formulas (SUKOP, THORNE 2006):

$$\rho = \sum_{i=0}^{n_i} f_i \quad (20)$$

and

$$\mathbf{v} = \frac{1}{\rho} \sum_{i=0}^{n_i} f_i \mathbf{e}_i \quad (21)$$

where:

- n_i – the number of spatial directions in the model (equal 8 in the D2Q9 model; 0 is referred to the current lattice node),
- \mathbf{e}_i – the unit vector of the i^{th} velocity direction (depends on the chosen lattice model).

A notable characteristic of the LBM is that the computational domain must be represented using a binary geometry. This kind of geometry can be directly prepared, for example, using random techniques or indirectly converted from another type of geometry. A conversion scheme from a vector geometry to a binary geometry is depicted in Figure 8. In this process, the original vector geometry (a) is overlaid by a lattice of nodes (b), and each node is assigned a value of 1 (solid node) or 0 (fluid node) based on its location. This method of defining the computational domain enables the relatively easy preparation of even highly complex geometries. Indeed, it is the primary advantage of this method and indicates its fundamental areas of application. However, the major drawback is that the quality of the resulting numerical model strongly depends on the assumed lattice resolution. Increasing the node density improves the model's quality but also increases the computational power requirements and the time needed to obtain results with the same level of convergence.

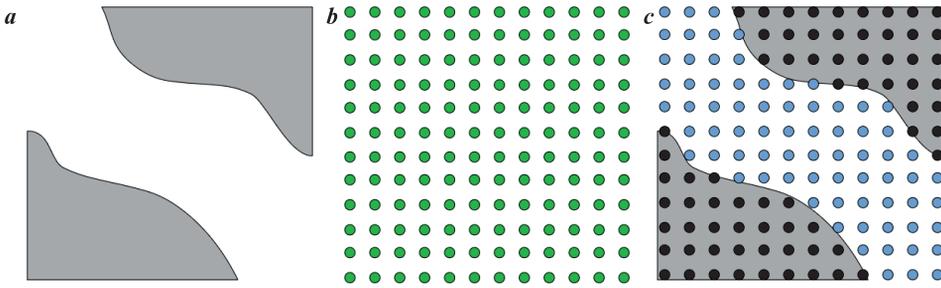


Fig. 8. Schematic representation of geometry conversion from a vector to a binary geometry; description in the text

Another feature of the method is that the movement of the lattice gas can only occur along strictly defined directions, and the probability of selecting a given direction is determined by appropriate weights. The weight value is highest for the current node, meaning that a significant number of molecules always stay in the same place. The most popular lattice models are D2Q9, D3Q15 (limited range of stability, low computational effort), D3Q19 (a compromise between stability and computational effort), and D3Q27 (good stability, high computational effort). D2 or D3 represents the number of space dimensions, and Q9, Q15, Q19, or Q27 represents the number of possible directions of gas movement. The lattice models described here, along with the weight values of individual directions, are presented in Figure 9 (SUKOP, THORNE 2006, WAGNER 2008).

In Figure 10, the main steps of the LBM algorithm are presented. In stages *a* and *b*, formulas (20-21) and (19) are used in a discrete form, respectively. In step *c*, collisions between particles and between particles and walls are included. The collision process involves swapping values of the appropriate components of the $f(\vec{x}, \vec{v}, t)$ function. The specific details depend on the assumed collision model, whether it includes slippage or not. On this stage, periodic conditions may also be implemented. The symbol f^* represents the f function after the collision stage but before the streaming procedure. In step *d*, the lattice gas is moved (streamed) along individual directions of the assumed lattice model, and this step includes the previously mentioned weights.

The model described in this chapter is applicable to single-phase laminar fluid flow. However, the LBM has now reached an advanced stage, enabling the modelling of a wide range of fluid flows, including the same fluid models, turbulence models, and multiphase flow concepts as in the FVM. The areas of application are also similar (e.g., Fig. 11*a*), but the LBM is particularly useful for modelling flows in highly complex geometries (e.g., Fig. 11*b*). Interesting examples of applications can be found in medicine (KRAUSE 2010, ABAS 2016), chemical engineering (XIE et al. 2018), agricultural engineering (KORN, HERLITZIUS

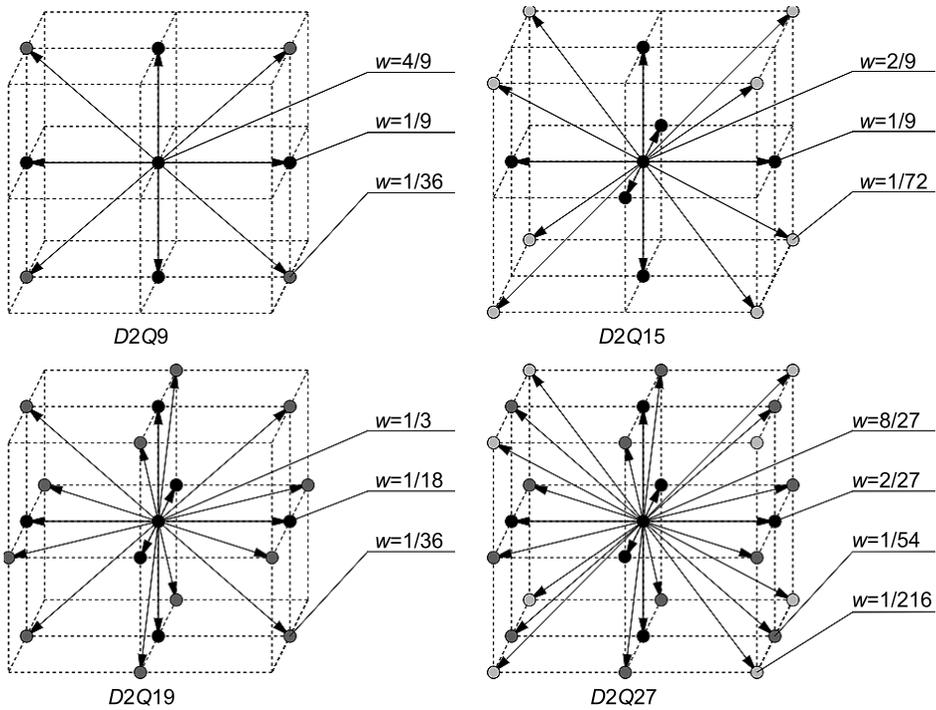


Fig. 9. The most popular lattice models in the LBM

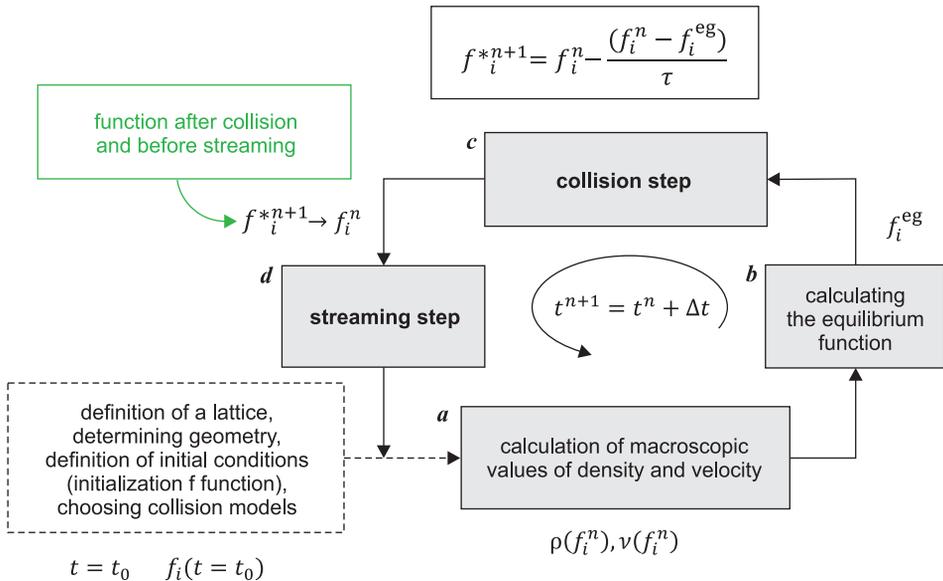


Fig. 10. Main steps of the LBM algorithm; description in the text

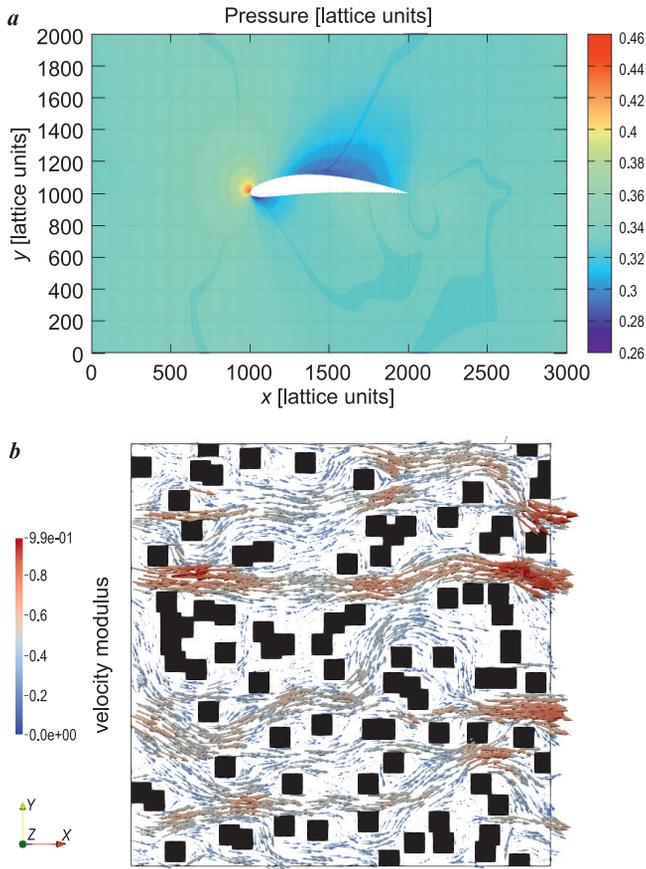


Fig. 11. Examples of using the LBM: *a* – modelling a flow around the NACA6412 airfoil, *b* – investigation of the hydraulic tortuosity in random generated pore structures

2017), energetics (SCHUBIGER et al. 2020), and other fields. The LBM is applied independently, as exemplified by the cited works of KRAUSE (2010) or ABAS et al. (2016), or in combination with another numerical method, primarily the Discrete Element Method (KORN, HERLITZIUS 2017, XIE et al. 2018).

The main drawback of the LBM is its significant computational power requirement, which limits its practical application in many cases. To expand its applicability, much attention is devoted to parallelization techniques or the use of graphical processing units (GPUs) instead of CPUs in computations.

Discrete Element Method

The Discrete Element Method (DEM) (CUNDALL, STRACK 1979) is employed for analysing the dynamics of solid systems in a Lagrangian approach. This method is founded on the classical principles of Newtonian dynamics. Typically, two types of solids are considered: particles, often representing granular materials, and walls, typically representing various types of technical infrastructure. Originally, particles could only have a spherical shape, but nowadays other simple shapes (such as cylinders, cuboids, polyhedrons) or complex shapes consisting of either rigidly connected bodies with simple shapes or closed surfaces in the form of a mesh are utilized. In the context of CFD, the DEM is increasingly being employed in conjunction with the FVM or the LBM to model multiphase flow problems in an Euler-Lagrange approach (see Fig. 5).

The discrete element method employs dynamic equations of motion (LABRA et al. 2012), which consider both translational:

$$m \frac{d\vec{v}}{dt} = \vec{F} \quad (22)$$

and rotational motion:

$$I \frac{d\vec{\omega}}{dt} = \vec{M} \quad (23)$$

where:

- m – mass [kg],
- I – moment of inertia [$\text{kg} \cdot \text{m}^2$],
- v – linear velocity [m/s],
- ω – angular velocity [1/s],
- F – resultant force [N],
- M – resultant moment of force [N·m].

The resultant force can be expressed as the sum of two types of actions:

$$\vec{F} = \vec{F}_c + \vec{F}_e \quad (24)$$

where:

- F_c – forces resulting from contacts (close interactions) [N],
- F_e – external forces (distant interactions) [N].

In turn, contact forces consist of two components:

$$\vec{F}_c = \vec{F}_n + \vec{F}_t \quad (25)$$

where:

- F_n – normal forces [N],
- F_t – tangential forces [N].

In the case of the i^{th} circular or spherical body in contact with other similar bodies, equations (22) and (23) will take the form:

$$m_i \frac{d\vec{v}_i}{dt} = \sum_{j=1}^{n_c} (\vec{F}_{n,ji} + \vec{F}_{t,ji}) + \vec{F}_e \tag{26}$$

and

$$I_i \frac{d\vec{\omega}_i}{dt} = \sum_{j=1}^{n_c} (r_i \times \vec{F}_{t,ji} + f_{ij} \times \vec{F}_{n,ji}) \tag{27}$$

where:

- $F_{n,ji}$ – normal force acting on the i^{th} body in contact with the j^{th} body [F],
- $F_{t,ji}$ – tangential force acting on the i^{th} body in contact with the j^{th} body [F],
- r_i – radius of the i^{th} body [m],
- $f_{ij} = f_{ji}$ – distance between the direction of the resultant normal force acting between the bodies i and j , and the centre of rotation of the i^{th} body [m].

If the contacting circular or spherical bodies are non-deformable, the contact zone is limited to a point, and the distance $f_{ij} = f_{ji}$ is equal to zero (Fig. 12a). The distance between the centres of the particles i and j is equal to $r_i + r_j$. This type of contact in the literature is called hard contact. If the bodies are deformable (Fig. 12b), then instead of a point contact, a line (in 2D) or a surface (in 3D) contact will be created. In this case, the resulting point of action of the normal force depends on the distribution of normal stresses on that line or surface. Consequently, $f_{ij} = f_{ji} > 0$, and the normal force will generate an additional moment of force. The particles are deformable, therefore the distance between their centres is smaller than the sum of the radii r_i and r_j . In this way, the degree of particle overlap becomes a measure of their deformation. The contact of deformable particles in the literature is called soft contact.

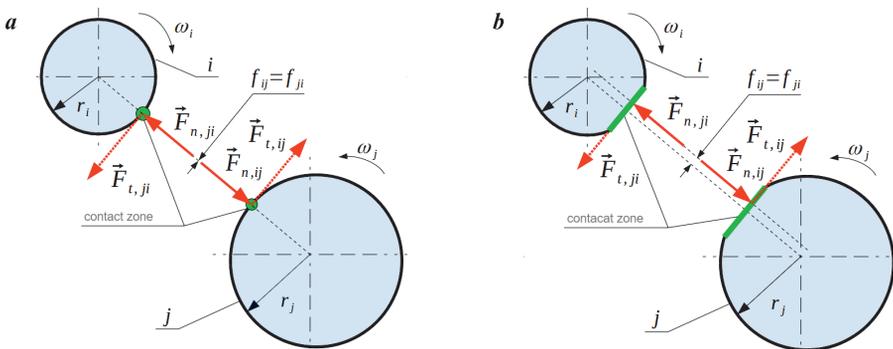


Fig. 12. Collision diagram of two circular or spherical bodies:
 a – non-deformable, b – deformable

In summary, in the DEM, two main tasks can be distinguished:

- searching for all contacts between objects;
- calculating forces and moments of forces at all points of contact based on adopted contact models.

Contact models are based on different material models, and additional factors may also be introduced, such as forces resulting from the existence of a liquid layer on the particle surfaces. The most popular material models include:

- elastic model – a model in which the strain is reversible and does not depend on the relative velocity between bodies. A distinction is made between a linear elastic model, according to Hooke's law, and non-linear models, e.g. the Hertz model (HERTZ 1881):

$$F_{n,ji}(\delta_n) = k_n \delta_n^\alpha \quad (28)$$

where:

- δ_n – absolute deformation at the contact point in the normal direction [m],
- k_n – modulus of elasticity in the normal direction [N/m^α],
- α – model exponent [-].

For mentioned Hooke and Hertz models is equal to 1 and 3/2, respectively;

- viscoelastic model – a model in which the deformation of a body is assumed to be a partially irreversible process (due to internal damping). Linear and non-linear viscoelastic models are distinguished:

$$F_{n,ji}(\delta_n) = k_n \delta_n^\alpha + C_n \delta_n^\beta \dot{\delta}_n \quad (29)$$

where:

- C_n – damping coefficient in the normal direction [kg/s],
- $\dot{\delta}_n$ – speed of absolute deformation at the contact point [m/s],
- β – model exponent [-];

– elasto-plastic model – a model in which it is assumed that in the first phase of loading, a body behaves elastically, and after a certain limit (yield point) is exceeded, the behaviour of the body becomes typical for a plastic material. This model can be described by various combinations of linear and nonlinear elastic or plastic models.

When determining the tangential force, two scenarios are usually considered. If the tangential force is relatively small, the tangential deformation is calculated. In the simplest approach, similar formulas are used as when calculating the normal force. If the tangential force exceeds the static friction force, sliding occurs with a constant dynamic friction force. The displacement in the normal direction is only dependent on material properties, not time. In contrast, if the colliding bodies are in a state of sliding, the relative displacement in the tangential direction depends on time.

In Figure 13, the main steps of the DEM algorithm are presented. To simplify the diagram, only forces are shown, but moments of forces are calculated in an analogous way.

A distinctive feature of the DEM is that the computational domain has no explicit boundaries. This means that the domain size depends on the current location of the objects involved in the simulation. Alternatively, the maximum range of coordinates for a simulation may be defined.

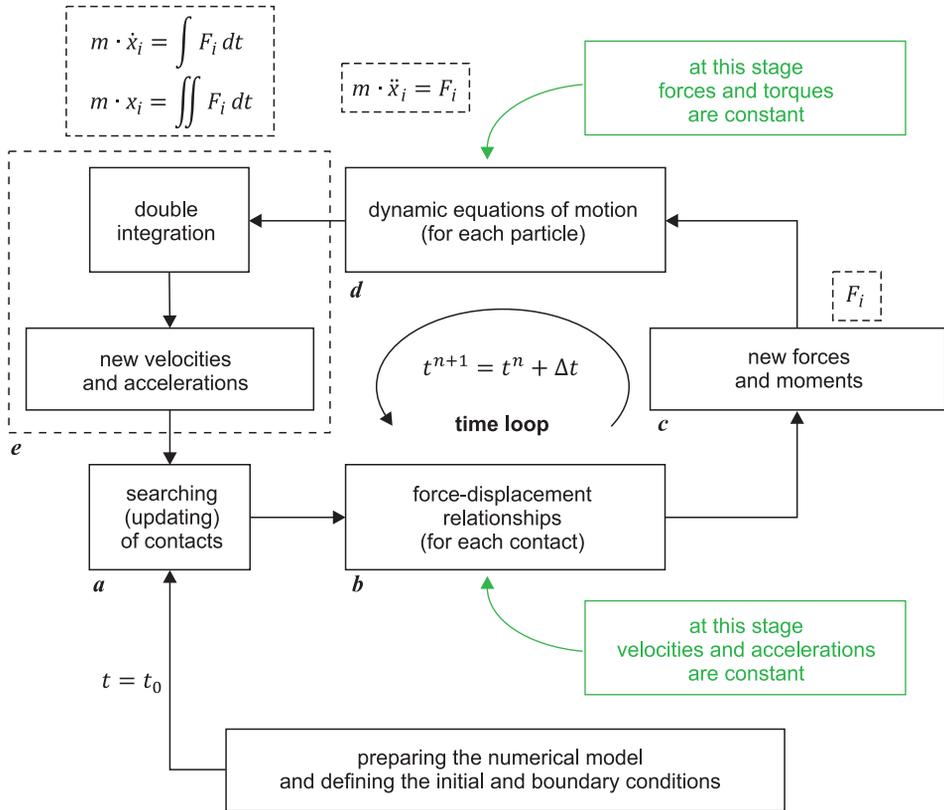


Fig. 13. Main steps of the DEM algorithm

A typical area of application for the DEM is the modelling of loose materials' behaviour in various processes, including transportation, loading and unloading, tank filling and emptying, collection, spreading, separation, and many others. An example from this field is shown in Figure 14a.

The DEM also bridges the gap between granular mechanics and classical solid mechanics. By utilizing carefully selected contact models, it can simulate phenomena such as cracking, delamination, crushing, grinding, pelleting, and

other processes in which the internal cohesion of materials is disrupted or formed. Additionally, the DEM has significant applications in modelling the mechanics of elastic and highly deformable bodies, such as ropes, membranes, foils, nets, fibres, rubber, silicone, and more.

In the context of CFD, the most interesting application of the DEM is its combination with FVM or LBM for modelling fluid-solid systems. Examples in this field include fluidized beds (FULCHINI et al. 2019, LIU, ZHAO 2021), cyclones (CHU 2010), soil erosion (ZOU et al. 2020), and more. The method can also be utilized to generate the geometry of granular porous media, which is subsequently employed, for instance, in the analysis of fluid flows within the pore space (SOBIESKI et al. 2018). In the first of the areas mentioned here, the coupling of DEM with FVM or DEM with LBM is performed iteratively within a single computational loop. In the second case, the connection is sequential (Fig. 14b). The DEM can also be used independently, without combining it with other numerical methods, to model the behaviour of semi-liquid materials such as pastes (PLATZER, FIMBINGER 2021), concretes (GAO et al. 2023), 3D printer filaments (KHAN, KOÇ 2022), and so on.

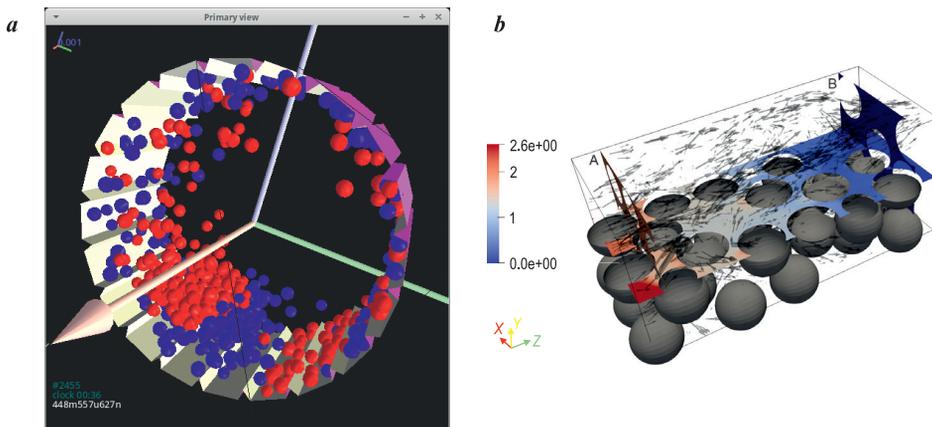


Fig. 14. Examples of using the DEM: *a* – mixing particles in a rotating tank (own calculation of an example available in the YADE DEM code (*The YADE code* 2023)), *b* – predicting the flow resistance by the use of the FVM in a granular porous bed created by the DEM
Source: own calculation of an example available in the YADE DEM code (2023).

Smoothed Particle Hydrodynamics

Smoothed Particle Hydrodynamics (SPH) (LIU, LIU 2003, LI, LIU 2008, LIND et al. 2020) is a mesh-less Lagrangian method for analysing the behaviour of solids or fluids in which continuous matter is represented by a set of particles.

Each particle represents a small volume of the matter and contains individual parameters such as mass, density, pressure, velocity, etc. The starting point for SPH comes from the observation that a function operating on a domain may be expressed in integral form as follows:

$$f(\vec{x}) = \int_{\Omega} f(\vec{x}')d\vec{x}' \tag{30}$$

where:

- \vec{x} – position vector of the point for which the value of the function is sought,
- \vec{x}' – position vector of a point where the value of the function is known.

In equation (30), if the Ω space is reduced to a single point, then the value of the function $f(\vec{x})$ will be equal to the value of $f(\vec{x}')$ at that point. Assuming that Ω space is continuous, a value will always be obtained for each \vec{x} location (see Fig. 15a). However, if Ω is a discrete space, two scenarios are possible:

- if the \vec{x} location coincides with one of the \vec{x}' locations, then the correct value of the $f(\vec{x})$ function can be obtained;
- in the other case, the $f(\vec{x})$ function will be zero (see Fig. 15b). This can be expressed in the following form:

$$f(\vec{x}) = \int_{\Omega} f(\vec{x}')\delta(\vec{x} - \vec{x}')d\vec{x}', \quad \text{where} \quad \delta(\vec{x} - \vec{x}') = \begin{cases} 1, & \text{for } \vec{x} = \vec{x}' \\ 0, & \text{for } \vec{x} \neq \vec{x}' \end{cases} \tag{31}$$

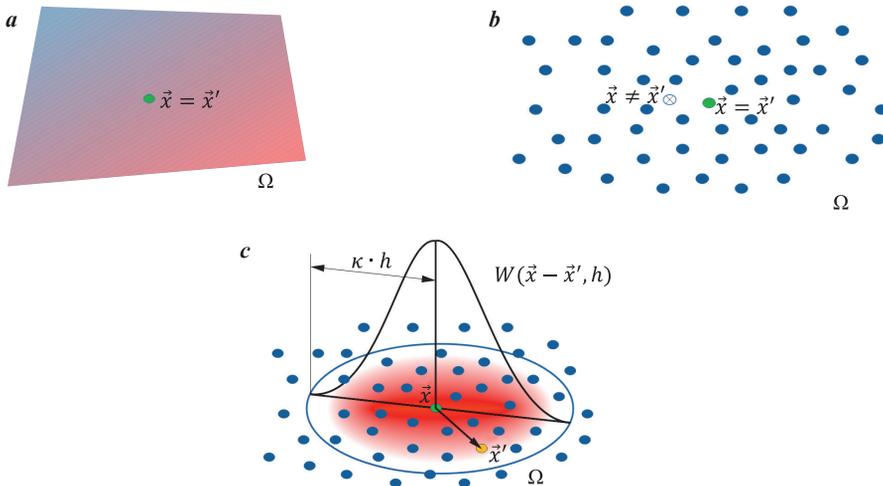


Fig. 15. Determining a function value in: a – a continuous space, b – a discrete space based on Dirac function, c – a discrete space based on smoothing function

The main question is how to obtain a value for any location in a discrete system. To address this issue, the following approximation can be applied (as shown in Fig. 15c)

$$f(\vec{x}) \approx \langle f(\vec{x}) \rangle = \int_{\Omega} f(\vec{x}') W(\vec{x} - \vec{x}', h) d\vec{x}' \quad (32)$$

where:

$W(\vec{x} - \vec{x}', h)$ – a weighting function (also known as a smoothing function or kernel function) that determines the degree of influence of a given location \vec{x}' on the approximation result at point \vec{x} . h is referred to as the smoothing length, which is typically constant for all particles and is responsible for defining the extent of the approximation area. The smoothing length is often multiplied by a factor κ , typically falling in the range between 1 and 3. In the literature, the space within the circle or sphere κh is referred to as the support domain.

The smoothing function must satisfy several important conditions, the most significant of which are as follows:

- normalization condition (eq. 33): This condition ensures that the influence of discretization points within the support domain sums up to 1. In other words, each discretization point contributes a percentage share to the calculation of the approximated value;

- Dirac function property (eq. 34): This property dictates that when the support domain is reduced to a single point, the smoothing function behaves like the Dirac delta function;

- compactness conditions (eq. 35): These conditions ensure that the value of the smoothing function is zero outside the support domain.

The smoothing function must meet a number of conditions, the most important of which are:

- normalization condition (eq. 33), what means that the influence of the discretization points in the support domain sums up to 1 (each of the discretization points has its percentage share in the calculation of the approximated value);

- Dirac function property (eq. 34), what means that in the case of reducing the size of the support domain to a point, the smoothing function becomes the Dirac delta;

- compact conditions (eq. 35), what means that the value of the smoothing function outside the support domain is zero:

$$\int_{\Omega} W(\vec{x} - \vec{x}', h) d\vec{x}' = 1 \quad (33)$$

$$\lim_{h \rightarrow 0} W(\vec{x} - \vec{x}', h) = \delta(\vec{x} - \vec{x}') \quad (34)$$

$$W(\vec{x} - \vec{x}', h) = 0, \text{ for } |\vec{x} - \vec{x}'| \geq \kappa h \quad (35)$$

The aforementioned conditions are used to derive various forms of smoothing functions. It is often assumed that the initial form of such a function is a polynomial. The most popular kernel functions include:

- Gaussian kernel (1977) for $\kappa = \infty$;
- Bell-shaped kernel (1977) for $\kappa = 1$;
- Quadratic kernel (1996) for $\kappa = 2$;
- Quintic kernel (1995) for $\kappa = 2$.

A crucial aspect of this method is that when calculating the derivative of a smoothing function, the ∇ operator is transferred from the function $f(\vec{x})$ to the smoothing function $W(\vec{x} - \vec{x}', h)$:

$$\nabla \cdot f(\vec{x}) \approx \langle \nabla \cdot f(\vec{x}) \rangle = - \int_{\Omega} f(\vec{x}') \cdot \nabla W(\vec{x} - \vec{x}', h) d\vec{x}' \quad (36)$$

Such a state significantly facilitates calculations. An example of a kernel function (Gaussian) and its derivative is provided in equations (37) and (38) (GOFFIN 2013). The term α_d depends on the kernel function and the number of spatial dimensions assumed. In the Gaussian kernel, it is equal to $1/\pi h^2$ in 2D and $1/\pi^{3/2}h^3$ in 3D. The shapes of both curves are presented in Figure 16:

$$W = \alpha_d e^{-\left(\frac{r}{h}\right)^2} \quad (37)$$

$$\frac{dW}{dr} = \frac{\alpha_d}{h} \left(-2 \frac{r}{h} e^{-\left(\frac{r}{h}\right)^2} \right) \quad (38)$$

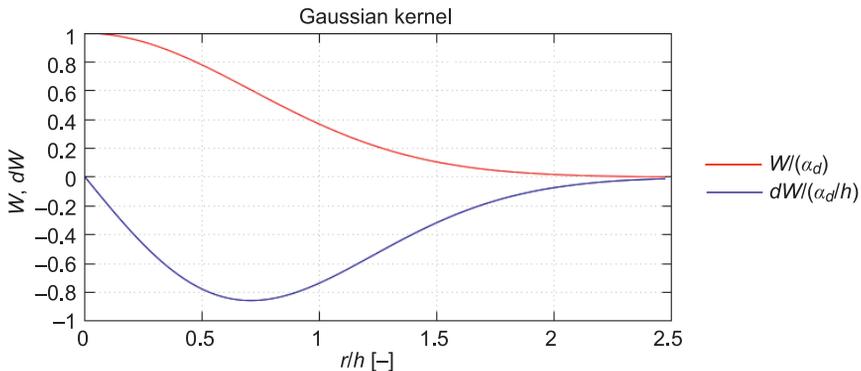


Fig. 16. Gaussian kernel function and its derivative

SPH is among others dedicated to the analysis of fluid flows. The fundamental equations of fluid dynamics can be expressed in the following form (LIU, LIU 2003, LI, LIU 2008):

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \vec{v} \quad (39)$$

$$\frac{d\vec{v}}{dt} = -\frac{1}{\rho} \nabla p + \frac{1}{\rho} \nabla \cdot \vec{\tau} + \vec{f} \quad (40)$$

After applying the previously described method, the mass balance equation (39) will be discretized as follows (LIU, LIU 2003, GESTEIRA et al. 2010):

$$\frac{d\rho_i}{dt} = \sum_{j=1}^N m_j (\vec{v}_i - \vec{v}_j) \cdot \nabla_i W_{ij} \quad (41)$$

where:

i – particle at location \vec{x} ,

j – particles at locations \vec{x}' ,

N – the number of particles belonging to the support domain of the i -th particle,

W_{ij} – a shorter notation for the smoothing function.

The discrete form of the momentum balance equation (40) can be expressed as follows (LIU and LIU 2003, GESTEIRA et al. 2010):

$$\frac{d\vec{v}_i}{dt} = - \sum_{j=1}^N m_j f(p, \rho) \nabla_i W_{ij} + \sum_{j=1}^N m_j f(\vec{\tau}, \rho) \nabla_i W_{ij} + \vec{f} \quad (42)$$

The function $f(p, \rho)$ can take the form of either $\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2}$ or $\frac{p_i + p_j}{\rho_i \rho_j}$, depending on the chosen symmetrization method. Symmetrization is necessary to satisfy Newton's third law. Similarly, the function $f(\vec{\tau}, \rho)$ can be expressed analogically as $\frac{\vec{\tau}_i}{\rho_i^2} + \frac{\vec{\tau}_j}{\rho_j^2}$ or $\frac{\vec{\tau}_i + \vec{\tau}_j}{\rho_i \rho_j}$. In this case, the shear tensor is calculated based on physical viscosity. Another popular approach is to use artificial viscosity, where the $f(\vec{\tau}, \rho)$ function is represented by the artificial viscosity, typically denoted as Π_{ij} . The advantages of using artificial viscosity are to ensure the stability of calculations when particles move apart, ensure the correct direction of momentum transfer, and facilitate implementation.

To calculate the term $f(p, \rho)$, the equation of state must be applied. For gases, the ideal gas law may be used (ARCHAMBEAU 2013, LIU, LIU 2003). In the literature, two forms of this equation may be found:

$$p = \left[\left(\frac{\rho}{\rho_0} - 1 \right) \rho_u \right] \frac{R_u T}{M} \tag{43}$$

and

$$p = (\gamma - 1)\rho e \tag{44}$$

where:

$\rho_u = 1$ – unitary density (needed to ensure that the units in the equation are consistent) [kg/m³],

γ – adiabatic exponent [-],

e – specific internal energy [J/kg].

For liquids, the artificial compressibility equation is typically used (ARCHAMBEAU 2013, ULRICH 2013):

$$p = B \left[\left(\frac{\rho}{\rho_0} \right)^\gamma - 1 \right] \tag{45}$$

The symbols used here have the following meanings:

ρ_0 – the reference density [kg/m³],

R_u – the universal gas constant [J/(K·mol)],

M – molar weight [kg/mol],

$B = \frac{c_s^2 \rho_0}{\gamma}$ – a constant dependent on the speed of sound [Pa],

c_s – the speed of sound [m/s],

γ – model exponent (usually equal to 7) [-].

Due to the fact that a high value of the speed of sound requires a very small time step, in practice, a lower value is often assumed. A popular expression is $c_s = 10v_{\max}$, where v_{\max} represents the maximum expected velocity in the flow. This adjustment keeps the error at an acceptable level while significantly reducing calculation time.

With the derivatives (41) and (42), the new values of the main variables can be calculated as follows (using the Euler scheme):

$$\rho_i^{n+1} = \rho_i^n + \frac{d\rho_i}{dt} \Delta t \tag{46}$$

$$\vec{v}_i^{n+1} = \vec{v}_i^n + \frac{d\vec{v}_i}{dt} \Delta t \tag{47}$$

$$\vec{x}_i^{n+1} = \vec{x}_i^n + \vec{v}_i^{n+1} \Delta t \tag{48}$$

where:

n – the current iteration [-],

$n + 1$ – the next iteration [-],

Δt – time step [s].

The time step is limited by conditions that take into account the intensity of mass forces and the intensity of viscous dissipation.

In Figure 17, the main SPH algorithm is presented. It is worth noting that nested loops are executed during a single time step: one loop iterates over all fluid particles, and another loop iterates over all particles belonging to the support domain of every fluid particle.

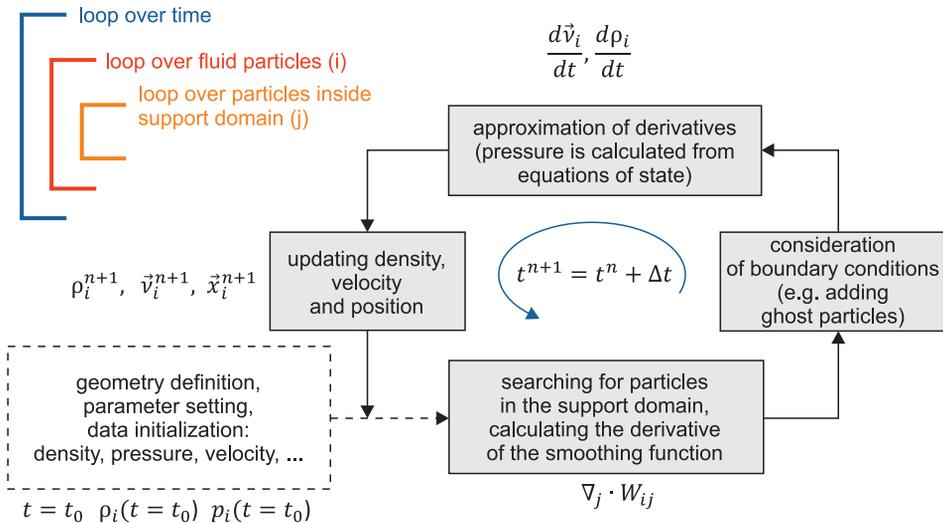


Fig. 17. Main steps of the SPH algorithm

When there are walls within the calculation domain, appropriate boundary conditions must be introduced. The three most popular variants are as follows: 1) Adding an artificial repulsive force to prevent liquid particles from penetrating the wall; 2) Introducing so-called ghost particles outside the wall with the same parameters as fluid particles but with the opposite direction of velocity. This results in velocity components on the wall being equal to zero; 3) Treating wall particles in the same way as fluid particles, requiring the definition of two layers of non-moving wall particles. Conditions 1 and 2 are commonly used. The third condition is sensitive to the initial distance between the fluid and the wall. When the initial distance is too great, the fluid may move towards the boundary, and when it is too small, the fluid molecule will experience suction as it moves away from the boundary.

The SPH method is primarily employed for modelling systems in which matter undergoes significant deformations. It finds applications in fracture mechanics (DOUILLET-GRELLIER et al. 2016), metal processing (NGUYEN, HOJNY 2022), additive manufacturing (TRUSHNIKOV et al. 2019), materials engineering

(KOUKOUVINIS et al. 2018), geomechanics (BUI, NGUYEN 2021), and various other fields. Of course, SPH can also be used to model conventional fluid dynamics, such as analysing coastal problems (see Fig. 18).

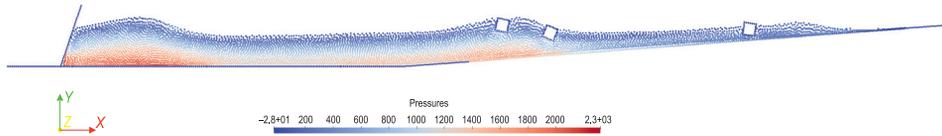


Fig. 18. Example of using SPH to modelling movement of objects floating on the water surface near a coast
Source: own calculation of an example available in the SPHYSICS_2D code (GESTEIRA et al. 2010).

Summary

Over the past few decades, numerous specialized numerical modelling methods have been developed. These techniques have been refined to the extent that they now constitute fundamental research tools in almost every field of science. Initially, there was a clear distinction between methods designed for modelling the behaviour of solids, fluids, or dispersed matter. Often, choosing one approach made it impossible to incorporate other options. However, the progress of science and current trends are expanding the possibilities of simultaneously using multiple methods and their variants. In particular, simulations that combine two or more numerical methods are becoming more prevalent. For example, the FVM or LBM can be combined with the DEM (e.g., for modelling fluidization, sedimentation, etc.), or the FVM can be coupled with the Finite Element Method (not described in this paper, e.g., for modelling fluid flow in flexible conduits). This development allows researchers to address complex problems that require a more comprehensive and integrated approach.

It is worth emphasizing that the range of possible variants in a given numerical model of physical phenomena is vast. The palette of techniques and variations is also highly diverse, particularly evident in the case of multiphase systems or domains with moving boundaries. This diversity often leads to communication difficulties between researchers involved in experimental studies and numerical modelling. Even a minor change in assumptions, concepts, or research objectives may necessitate the use of entirely different mathematical frameworks or computational techniques. Since researchers specializing in simulation studies typically focus on specific problem classes, such changes can significantly delay the entire task due to the need to familiarize themselves with new theories, relevant literature, and the practical application of newly acquired concepts in selected software. Moreover, the time required for

conducting a single simulation can extend to days or weeks. Considering that the development of new models often involves testing multiple variants (e.g., during the search for new closures), obtaining initial, qualitatively correct results may take months or even years. This situation often leads to misunderstandings, as experimenters or engineers usually expect rapid results. The necessity for laborious preparation when approaching modelling new problem classes is one of the factors limiting the widespread application of numerical modelling at the engineering level.

Another critical factor is the current inadequacy and insufficient computational power of today's computers. It can be speculated that a new chapter in numerical research will begin with the advent of long-awaited quantum computers. Furthermore, some currently employed approaches are expected to become less attractive: Reynolds-Averaged Navier-Stokes may be replaced by Direct Numerical Simulation; diffusion-convection equations may be superseded by full mass, momentum, and energy balance equations; macro-scale Porous Media Models may be substituted by micro-scale modelling of fluid flow in porous channels, and so on. It is also important to note that greater computational power will enable modelling on a significantly larger scale, such as simulating the entire flow system in a hydroelectric power plant rather than focusing solely on a single interblade channel. This development will hold paramount practical significance.

The publication was written as a result of the author's internship in the Ljubljana University in Slovenia, co-financed by the European Union under the European Social Fund (Operational Program Knowledge Education Development), carried out in the project Development Program at the University of Warmia and Mazury in Olsztyn (POWR.03.05. 00-00-Z310/17).

References

- ABAS A., MOKHTAR N.H., ISHAK H.H.H., ABDULLAH M.Z., TIAN A.H. 2016. *Lattice Boltzmann Model of 3D Multiphase Flow in Artery Bifurcation Aneurysm Problem*. Computational and Mathematical Methods in Medicine, 2016, article ID 6143126. <https://doi.org/10.1155/2016/6143126>
- ARCHAMBEAU P., PIROTON M., DEWALS B., DUCHÊNE L., MOUZELARD T. 2013. *Development of a didactic SPH model*. Université de Liège – Faculté des Sciences Appliquées.
- BHATNAGAR P.L., GROSS E.P., KROOK M. 1954. *A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems*. Physical Review, 94: 511-525.
- BUI H.H., NGUYEN G.D. 2021. *Smoothed particle hydrodynamics (SPH) and its applications in geomechanics: From solid fracture to granular behaviour and multiphase flows in porous media*. Computers and Geotechnics, 138: 104315.
- BURDEN R.L., FAIRES J.D. 2011. *Numerical Analysis*. 9th Edition. Brooks/Cole, Pacific Grove.
- CAUSON D.M., MINGHAM C.G. 2010. *Introductory Finite Difference Methods for PDEs*. Ventus Publishing ApS.

- CHU K. 2010. *CFD-DEM simulation of complex particle-fluid flows*. PhD Thesis. The University of New South Wales.
- COURANT R., FRIEDRICHS K., LEWY H. 1928. *On the Partial Difference Equations of Mathematical Physics*. *Mathematische Annalen*, 100: 32-74.
- CUNDALL P., STRACK O. 1979. *A discrete element model for granular assemblies*. *Géotechnique*, 29: 47-65.
- DOUILLET-GRELLIER T., JONES B.D., PRAMANIK R., PAN K., ALBAIZ A., WILLIAMS J.R. 2016. *Mixed-mode fracture modeling with smoothed particle hydrodynamics*. *Computers and Geotechnics*, 79: 73-85.
- FULCHINI F., GHADIRI M., BORISSOVA A., AMBLARD B., BERTHOLIN S., CLOUPET A., YAZDANPANAHI M. 2019. *Development of a methodology for predicting particle attrition in a cyclone by CFD-DEM*. *Powder Technology*, 357: 21-32.
- GAO Y., CHENG Y., CHEN J. 2023. *Experimental Study and 3-D Meso-Scale Discrete Element Modeling on the Compressive Behavior of Foamed Concrete*. *Buildings*, 13: 674.
- GESTEIRA M.G., ROGERS B.B., DALRYMPLE R.A., CRESPO A.J.C., NARAYANASWAMY M. 2010. *User Guide for the SPHysics code*. Retrieved from https://wiki.manchester.ac.uk/sphysics/index.php/SPHYSICS_Home_Page (20.05.2023).
- GOFFIN L. 2013. *Development of a didactic SPH model*. MSc Thesis, Université de Liège (Belgium).
- HERTZ H. 1881. *Über die berührung fester elastischer Körper*. *Journal für die Reine und Angewandte Mathematik*, 92: 156-171.
- HUGHES T.J.R., ZIENKIEWICZ O.C. 1979. *Finite elements in fluid mechanics: an introduction to the Galerkin method*. Pergamon Press, Oxford.
- IYENGAR S.R.K., JAIN R.K. 2009. *Numerical Methods*. New Age International Publishers, New Delhi.
- KHAN S.A., KOÇ M. 2022. *Numerical modelling and simulation for extrusion-based 3D concrete printing: The underlying physics, potential, and challenges*. *Results in Materials*, 16: 100337.
- KORN Ch., HERLITZIUS T. 2012. *Coupled CFD-DEM simulation of separation process in combine harvester cleaning devices*. *Landtechnik*, 72(5): 247-261.
- KOUKOUVINIS P., KYRIAZIS N., GAVAISES M. 2018. *Smoothed particle hydrodynamics simulation of a laser pulse impact onto a liquid metal droplet*. *PLoS ONE*, 13(9): e0204125.
- KRAUSE M.J. 2010. *Fluid Flow Simulation and Optimisation with Lattice Boltzmann Methods on High Performance Computers – Application to the Human Respiratory System*. PhD Thesis. Karlsruhe Institute of Technology.
- KRÜGER T., KUSUMAATMAJA H., KUZMIN A., SHARDT O., SILVA G., VIGGEN E.M. 2017. *The Lattice Boltzmann Method – Principles and Practice*. Springer, Berlin.
- LABRA C.A., ONATE E., ROJEK J. 2012. *Advances in the development of the discrete element method for excavation processes*. Monograph CIMNE, N°-132: 1-213.
- LANGTANGEN H.P., LINGE S. 2017. *Finite Difference Computing with PDEs – A Modern Software Approach*. Springer Open, Berlin.
- LI S., LIU W.K. 2007. *Meshfree Particle Methods*. Springer, Berlin.
- LIND S.J., ROGERS B.D., STANSBY P.K. 2020. *Review of smoothed particle hydrodynamics: towards converged Lagrangian flow modelling*. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 476(2241).
- LIU G.R., LIU M.B. 2003. *Smoothed Particle Hydrodynamics*. World Scientific Publishing, Singapore.
- LIU Z., MA H., ZHAO Y. 2021. *CFD-DEM Simulation of Fluidization of Polyhedral Particles in a Fluidized Bed*. *Energies*, 14: 4939.
- MACCORMACK R.W., PAULLAY A.J. 1972. *Computational efficiency achieved by time splitting of finite difference operators*. AIAA Paper, 72-154, San Diego.
- MCDONALD P.W. 1971. *The Computation of Transonic Flow Through Two-Dimensional Gas Turbine Cascades*. *Proceedings of the ASME 1971 International Gas Turbine Conference and Products Show*. ASME 1971 International Gas Turbine Conference and Products Show, Houston, Texas, USA. March 28-April 1.

- MENTER F.R. 1993. *Zonal Two Equation k - ω Turbulence Models for Aerodynamic Flows*. AIAA Paper, 93-2906.
- MOHAMAD A.A. 2011. *Lattice Boltzmann Method. Fundamentals and Engineering Applications with Computer Codes*. Springer, Berlin.
- MOUKALLED F., MANGANI F., DARWISH M. 2016. *The Finite Volume Method in Computational Fluid Dynamics*. Springer, Berlin.
- NGUYEN T.T.T., HOJNY M. 2022. *Application of Smoothed Particle Hydrodynamics Method in Metal Processing: An Overview*. Archives of Foundry Engineering, 3: 67-80.
- NIEDŹWIEDZKA A., SCHNERR G.H., SOBIESKI W. 2016. *Review of numerical models of cavitating flows with the use of the homogeneous approach*. Archives of Thermodynamics, 37(2): 71-88.
- ORSZAG S.A. 1970. *Analytical Theories of Turbulence*. Journal of Fluid Mechanics, 41: 363-386.
- PESKIN C.S. 1977. *Numerical Analysis of Blood Flow in the Heart*. Journal of Computational Physics, 25: 220-252.
- PLATZER F., FIMBINGER E. 2021. *Modelling Pasty Material Behaviour Using the Discrete Element Method*. Multiscale Science and Engineering, 3: 119-128.
- SCHUBIGER A., BARBER S., NORDBORG H. 2020. *Evaluation of the lattice Boltzmann method for wind modelling in complex terrain*. Wind Energy Science, 5: 1507-1519.
- SOBIESKI W. 2009. *Switch function and sphericity coefficient in the Gidaspow drag model for modeling solid-fluid systems*. Drying Technology, 27(2): 267-280.
- SOBIESKI W., GRYGO D. 2019. *Fluid flow in the impulse valve of a hydraulic ram*. Technical Sciences, 22(3): 205-220.
- SOBIESKI W., MATYKA M., GOLEMBIEWSKI J., LIPÍŃSKI S. 2018. *The Path Tracking Method as an alternative for tortuosity determination in granular beds*. Granular Matter, 20: 72.
- SMAGORINSKY J. 1963. *General circulation experiments with the primitive equations I. the basic experiment*. Monthly Weather Review, 91(3): 99-164.
- SPALART P.R. 1997. *Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach*. In Proceedings of first AFOSR international conference on DNS/LES, Greyden Press.
- SUCCI S. 2001. *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond*. Clarendon Press, Oxford.
- SUKOP M.C., THORNE D.T. 2006. *Lattice Boltzmann Modeling – An Introduction for Geoscientists and Engineers*. Springer, Berlin.
- The YADE code. Retrieved from <https://yade-dem.org/doc/> (20.05.2023).
- TRUSHNIKOV D.N., KOLEVA E.G., DAVLYATSHIN R.P., GERASIMOV R.M., BAYANDIN Y.V. 2019. *Mathematical modeling of the electronbeam wire deposition additive manufacturing by the smoothed particle hydrodynamics method*. Mechanics of Advanced Materials and Modern Processes, 5: 4.
- URLICH Ch. 2013. *Smoothed-Particle-Hydrodynamics Simulation of Port Hydrodynamic Problems*. TUHH, Hamburg.
- WAGNER A.J. 2008. *A Practical Introduction to the Lattice Boltzmann Method*. North Dakota State University, Fargo.
- WEINAN E., LIU J.-G. 2000. *Gauge finite element method for incompressible flows*. International Journal for Numerical Methods in Fluids, 24(8): 701-710.
- XIE Y., LIU Y., LI L., XU Ch., LI B. 2018. *Simulation of different gas-solid flow regimes using a drag law derived from lattice Boltzmann simulations*. The Journal of Computational Multiphase Flows, 10(4): 202-204.
- ZIENKIEWICZ O.C., TAYLOR R.L., NITHIARASU P. 2005. *The Finite Element Method for Fluid Dynamics*. Butterworth-Heinemann, Oxford (UK).
- ZOU Y., CHEN Ch., ZHANG L. 2020. *Simulating Progression of Internal Erosion in Gap-Graded Sandy Gravels Using Coupled CFD-DEM*. International Journal of Geomechanics, 20(1): 04019135-1.