V European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2010
J. C. F. Pereira and A. Sequeira (Eds)
Lisbon, Portugal,14-17 June 2010

FAST MULTIPOLE BOUNDARY ELEMENT METHOD WITH LAGRANGIAN PARTICLE TRACKING FOR VISCOUS FLOWS

Jure Ravnik[*,†], **Matjaž Hriberšek**[*] **and Leopold Škerget**[*]

[*]University of Maribor, Faculty of Mechanical Engineering
Smetanova ulica 17, SI-2000 Maribor, Slovenia
[†]e-mail: jure.ravnik@uni-mb.si

Key words: Dispersed two phase flow, Lagrangian particle tracking, Lid driven cavity, Boundary Element Method, Fast Multipole Method

Abstract. *An algorithm for the solution of the velocity-vorticity formulation of the Navier-Stokes equations will be presented. Boundary element method (BEM) based numerical technique is used to solve the governing equations. Single domain BEM is employed to solve for vorticity on the boundary. Solution of vorticity and energy transport equations is obtained via sub-domain BEM. After discretization the use of single domain BEM results in fully populated system of linear equations, whose parameters need to be stored in fully populated matrices. In order to avoid storage and calculation of fully populated matrices, we employ the fast multipole method (FMM). Using FMM we obtain a data sparse representation of the fully populated matrices and thus reduce storage, CPU time requirements and the complexity of the problem from* $\mathcal{O}(n^2)$ *to* $\mathcal{O}(n \log n)$.

The developed code has been coupled with a Lagrangian particle tracking solver. The coupled algorithms are capable of simulation of dilute suspensions of particles in viscous flows taking into account gravity, buoyancy, drag, pressure gradient and added mass. The Lagrangian solver uses the Newton-Raphson method to determine the velocity of the fluid at the location of the particle and the fourth order Runge-Kutta method to advance the particle location and velocity through time.

The developed method was used to study behaviour of slightly buoyant and non-buoyant particles in a lid driven cavity. We aimed at discovering cases when particles leave the primary vortex and enter into secondary vortices, a phenomenon described in previous experimental work. A parametric study of this phenomenon was performed. We confirmed and extended experimental observations. It was found, that the diameter of the particle is the crucial factor for the particle to gain entrance in the secondary vortex. The narrow streamline corridors along the top and side walls prevent entrance of larger particles. The limiting diameter was found to increase with the increase of the flows Reynolds number value.

1 INTRODUCTION

Dispersed two phase flows are commonly encountered in environmental flows and engineering practice, especially in process engineering. Environmental flows commonly include dispersion of solid or liquid particles in atmosphere and solid, liquid and gas particles in hydrosphere. In process engineering, the main attraction of implementing dispersed multiphase flow systems is in a large contact area between the continuous and dispersed phase, thus increasing the effective heat and mass transport between particles and fluid. On the other hand, dispersed multiphase systems arise in pneumatic and slurry transport systems, enabling effective transportation of granular material inside a process system, as well as inside chemical reactors, as for example in crystallization and extraction processes. In the latter case, mixing vessel with Rushton impellers is a frequently encountered process equipment. A simplification of such a system is a driven cavity flow in a cubic cavity. Existence of recirculation areas in corners of the cavity can effect the quality of the product, as these areas posses characteristics of low heat and mass transfer. It is therefore an important question, when and how do particles of various sizes and densities enter these areas.

In this work we coupled a fluid flow solver with a Lagrangian particle tracking module. We used the developed numerical algorithm to study particle movement in a lid driven cavity flow. The case was experimentally studied by Tsorng et al.[1] and offers an excellent combination of a standard benchmark test case for fluid flow with praxis relevant study of particle movement in the flow. The practical aspects refer to flow in various mixing devices.

2 SIMULATION OF FLUID FLOW

Laminar viscous flow was simulated using a code developed by Ravnik et al.[2, 3, 4]. The code solves the velocity-vorticity formulation of Navier-Stokes equations using the boundary-domain integral method. A combination of sub-domain and single domain technique is used. Incompressible viscous Newtonian fluid with constant material properties is considered. In velocity-vorticity formulation vorticity $\vec{\omega}$ is defined as the curl of the velocity $\vec{\omega} = \vec{\nabla} \times \vec{u}$. Both velocity and vorticity fields are divergence free. The viscous fluid flow is governed by the kinematics equation

$$\nabla^2 \vec{u} + \vec{\nabla} \times \vec{\omega} = 0, \tag{1}$$

which links the velocity and vorticity fields for every point in space and time. The kinetic aspect of fluid movement is governed by the vorticity transport equation, written in non-dimensional form:

$$\frac{\partial \vec{\omega}}{\partial t} + (\vec{u} \cdot \vec{\nabla})\vec{\omega} = (\vec{\omega} \cdot \vec{\nabla})\vec{u} + \frac{1}{Re}\nabla^2 \vec{\omega}, \tag{2}$$

with the Reynolds number denoted by Re. Equation (2) equates the advective vorticity transport on the left hand side with the vortex twisting and stretching term and the diffusion term on the right hand side.

The system of equations (1) and (2) is solved in a nonlinear loop of three steps. In the first step, boundary vorticity values are calculated by solving the kinematics equation by single domain BEM. The second step is the calculation of domain velocity values by solving the kinematics equation by subdomain BEM and the final step is the solution of vorticity transport equation for domain vorticity values using the boundary values from the solution of the kinematics equation by subdomain BEM.

The boundary condition required to obtain the solution is the prescribed velocity on the boundary. The unknown boundary conditions for the vorticity transport equation are calculated as a part of the algorithm using single domain BEM.

2.1 Fast multipole single domain BEM

Consider an arbitrary domain Ω with a position vector $\vec{r} \in \mathbb{R}^3$ and its boundary $\Gamma = \partial\Omega$. The integral form of the kinematics equation without derivatives of the velocity and vorticity fields takes the following form:

$$c(\vec{\xi})\vec{v}(\vec{\xi}) + \int_\Gamma \vec{v}\,\vec{\nabla}u^\star \cdot \vec{n}d\Gamma = \int_\Gamma \vec{v} \times (\vec{n} \times \vec{\nabla})u^\star d\Gamma + \int_\Omega (\vec{\omega} \times \vec{\nabla}u^\star)d\Omega, \tag{3}$$

where $\vec{\xi}$ is the collocation point, $u^\star = 1/4\pi|\vec{r} - \vec{\xi}|$ is the fundamental solution of the Laplace equation and $\vec{n}$ is the unit normal. In order to use the kinematics equation to obtain boundary vorticity values, we must rewrite the equation (3) in a tangential form by multiplying it with a normal in the source point $\vec{n}(\vec{\xi})$:

$$\begin{aligned} c(\vec{\xi})\vec{n}(\vec{\xi}) \times \vec{v}(\vec{\xi}) + \vec{n}(\vec{\xi}) \times \int_\Gamma \vec{v}\vec{\nabla}u^\star \cdot \vec{n}d\Gamma \\ = \vec{n}(\vec{\xi}) \times \int_\Gamma \vec{v} \times (\vec{n} \times \vec{\nabla})u^\star d\Gamma + \vec{n}(\vec{\xi}) \times \int_\Omega (\vec{\omega} \times \vec{\nabla}u^\star)d\Omega. \end{aligned} \tag{4}$$

In order to write a linear system of equations for the unknown boundary vorticity values, we set the source point into every boundary node of the whole computational domain.

In this work we use meshes made up of hexahedral domain elements and parallelepipedial boundary elements. Shape functions are used to describe function variation in elements. Quadratic interpolation is achieved by using 27 nodes per hexahedra and domain shape functions Φ_i. Boundary shape functions φ_i having 9 nodes are used within each boundary element.

To be able to write a discrete form of the kinematics equation, the following integrals must be calculated:

$$[H] = \int_\Gamma \varphi_i \vec{\nabla}u^\star \cdot \vec{n}d\Gamma, \quad [\vec{H}^t] = \int_\Gamma \varphi_i(\vec{n} \times \vec{\nabla})u^\star d\Gamma, \quad [\vec{D}] = \int_\Omega \Phi_i \vec{\nabla}u^\star d\Omega. \tag{5}$$

The square brackets denote integral matrices. Each collocation point location yields one row in these matrices. The collocation point is placed into every boundary node, making

the number of rows of matrices (5) equal to the number of boundary nodes. The number of columns of matrices $[H]$ and $[\vec{H}^t]$ is equal to the number of boundary nodes. The domain matrices $[\vec{D}]$ have the number of columns equal to the number of all nodes.

A discrete form of equation (4) is

$$[\vec{n}] \times [H]\{\vec{v}\} = [\vec{n}] \times \left(\{\vec{v}\} \times [\vec{H}^t]\right) + [\vec{n}] \times \left(\{\vec{\omega}\} \times [\vec{D}]\right), \tag{6}$$

where the curly brackets denote vectors of nodal values, $[\vec{n}]$ is the matrix of nodal values of $\vec{n}(\vec{\xi})$ and the calculated $c(\vec{\xi})$ are added to the diagonal terms of the $[H]$ matrix. In order to use equation (6) to calculate boundary vorticity values, it must be rearranged so that the boundary part of domain matrices $[\vec{D}_\Gamma]$ are moved to the left hand side and all other matrices form the right hand side.

$$[\vec{n}] \times \left(\{\vec{\omega}_\Gamma\} \times [\vec{D}_\Gamma]\right) = -[\vec{n}] \times [H]\{\vec{v}\} + [\vec{n}] \times \left(\{\vec{v}\} \times [\vec{H}^t]\right) + [\vec{n}] \times \left(\{\vec{\omega}_\Omega\} \times [\vec{D}_\Omega]\right), \tag{7}$$

The domain matrices $[\vec{D}_\Omega]$ on the right hand side take up most storage space and their multiplication with vectors take up most CPU time. The number of elements in $[\vec{D}_\Omega]$ is equal to $n_b \cdot n_d$, where n_b is the number of boundary nodes and n_d is the number of domain nodes. Using the FMM we will set up an approximation of the domain matrices $[\vec{D}'_\Omega] \approx [\vec{D}_\Omega]$ whose complexity will scale as $\mathcal{O}(n_d)$.

Let us consider the domain integrals in equation (5), which must be calculated in order to set up $[\vec{D}_\Omega]$. Since for each collocation point $\vec{\xi}$ integrals for all domain cells must be evaluated, we are obviously faced by a problem of quadratic complexity. The FMM, described below, is used to reduce this complexity.

2.1.1 Series expansion

The FMM is based on the fact that it is possible to separate the variables (i.e. the collocation point $\vec{\xi}$ and the domain integration point $\vec{r}$) of the integral kernel of equation (5) by series expansion. The gradient of the Laplace fundamental solution of equation (5) is expanded into a spherical harmonics series in the following manner:

$$\begin{aligned}
\vec{\nabla} u^\star = \vec{\nabla} \frac{1}{4\pi|\vec{r}-\vec{\xi}|} = \\
= \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \frac{(-1)^m}{2l+1} \frac{1}{\xi^{l+1}} Y_l^{-m}(\theta_\xi, \varphi_\xi) \vec{\nabla} \left[r^l Y_l^m(\theta_r, \varphi_r) \right] = \\
= \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \frac{(-1)^m}{2l+1} \frac{1}{\xi^{l+1}} Y_l^{-m}(\theta_\xi, \varphi_\xi) \left\{ l Y_l^m(\theta_r, \varphi_r) r^{l-2} \vec{r} + r^l \vec{\nabla} Y_l^m(\theta_r, \varphi_r) \right\},
\end{aligned} \tag{8}$$

where Y_l^m are spherical harmonics in polar coordinate system; $\vec{r} = (r, \varphi_r, \theta_r)$ and $\vec{\xi} = (\xi, \varphi_\xi, \theta_\xi)$.

In order for the series (8) to converge we must have $r/\xi < 1$. This condition is not satisfied for all $\vec{\xi}$ and $\vec{r}$ combinations in an arbitrary domain. However, since the integral kernel depends only on the distance between the collocation and domain points, we may swap $\vec{\xi}$ and $\vec{r}$ in order to meet the convergence criteria. Furthermore, it is also possible to move the origin of the coordinate system so that the series convergence is improved.

Using the above expansion, the domain integrals of equation (5) may now be written with separate variables, as

$$[\vec{D}_\Omega] \approx [\vec{D}'_\Omega] = \sum_{l=0}^{L} \sum_{m=-l}^{l} F_l^m(\vec{\xi}) \int_\Omega G_l^m(\vec{r}) d\Omega, \tag{9}$$

where F and G represent the above derived relationships. We are able to approximately calculate each entry in the domain matrices with the above sum. The number of expansion terms $n_{exp} = (L+1)^2$ in the series controls the accuracy of the approximation.

Using the series (9) instead of the direct evaluation of the integral kernel does not by itself bring a reduction of memory. Only when (9) is used on a cluster of collocation points and domain cells it is possible to form a data sparse approximation of a part of the domain matrix. The clusters are formed and organized in a hierarchical tree-like structure, which is described below.

2.1.2 Cluster trees

Let us consider a cluster of n_r nearby collocation points and a cluster of n_c nearby domain cells, as illustrated in Figure 1. These correspond to a $n_r \times n_c$ matrix block,

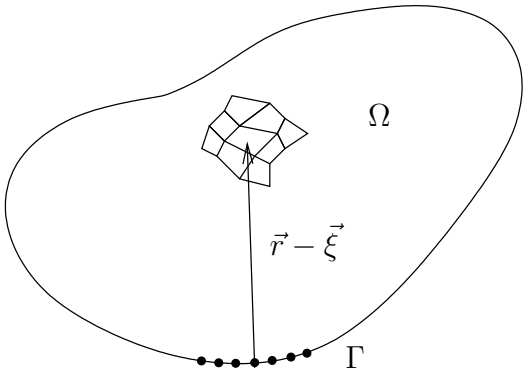

Figure 1: A problem domain shown with a cluster of collocation points $\vec{\xi}$ and a cluster of domain cells.

which is a part of the domain matrix. Since the variables in equation (9) are separated, it is possible to evaluate two lower order matrix blocks $(n_r \times n_{exp})$ and $(n_{exp} \times n_c)$ instead of the full matrix block $(n_r \times n_c)$. In the first lower order matrix block expansion terms F are

evaluated for all collocation points. In the second one integrals of expansion terms G are evaluated for all domain cells. Multiplication of the two lower order matrix blocks gives the full $n_r \times n_c$ matrix block up to an expansion error, which is defined by the number of terms in the expansion. But this is never done; namely we store the two lower order matrices instead of the full matrix. This technique saves memory if the amount of data, that must be stored in the two lower order matrices, is smaller than the amount of data in the full matrix block, i.e.

$$2(n_r n_{exp} + n_c n_{exp}) < n_r n_c; \tag{10}$$

the factor 2 on the left hand side is due to the fact that spherical harmonics are complex and must be stored as such, while real values are stored in the full matrix. As long as the collocation node cluster and the domain cells cluster are far apart from each other the integral kernels are slowly varying functions, so we can expect a low number of expansion terms to yield a suitable approximation. When the clusters are nearby, they should be smaller and a larger number of expansion terms must be used. When the clusters coincide, i.e. the collocation nodes are a part of the integration cells, the kernels are singular. Such cluster pairs are called inadmissible and the corresponding matrix block is evaluated in full, not approximated with two lower order matrices.

In order to be able to build a sparse approximation of the whole domain matrix, we must divide the collocation points and domain cells into clusters. We constructed a tree of collocation point clusters and a tree of clusters of domain cells. The trees were constructed in a recursive hierarchical manner. The problem domain was enclosed by a parallelepiped. All collocation points and all of the domain cells are within this root parallelepiped. They make up root clusters of both trees. The parallelepiped is cut in half by a plane, breaking the root clusters into two. The cutting process is repeated recursively, so the clusters on each level have less and less collocation points and domain cells. Each branch in the tree of clusters has two child branches corresponding to the cluster's domain being cut in half. The cutting planes are parallel to the coordinate system axes, a sequence of $x - y$, $x - z$ and $y - z$ is used. Thus three cuts are needed to cut a cube into eight equal parts. The cutting sequence is stopped, when the number of collocation nodes and domain cells in the cluster is so small, that the condition (10) can no longer be satisfied.

With both cluster trees in place, the next step is to pair them, so a tree of pairs of clusters can be formed. Each branch of the collocation tree is paired with each branch of the domain cells tree on the same level and with each branch of the domain cells tree on the next level thus forming branches on the tree of pairs of clusters. For each pair a decision is taken based on the admissibility criterion whether a sparse approximation for this cluster pair is possible or not. If the pair is admissible, the branch on the tree becomes an admissible leaf, where the two low order matrices will be calculated. If admissibility criterion is not reached until the last level of the tree, such cluster pairs are inadmissible and will be calculated in full and not with the sparse approximation.

2.1.3 Admissibility criterion

The admissibility criterion is devised as follows. Let us consider one branch of the tree of pairs of clusters, which has a cluster of collocation points and a cluster of domain cells. Firstly, we try to find an origin of the coordinate system in nodes within the domain cells of the cluster. We choose such origin that the ratio r/ξ is minimal for all pairs of collocation nodes and domain cells so the series will converge as fast as possible. If the minimal ratio is above one, series expansion for this pair of clusters is not possible. Thus this pair is not admissible. Secondly, if the r/ξ ratio is below one, we calculate the number of expansion terms needed to have the accuracy of calculation of the integral kernel less than user's prescribed criteria ϵ. If the number of expansion terms is low enough, so that condition (10) is fulfilled, this cluster pair is admissible. At this point the tree of pairs of clusters gets a leaf - no further branching is necessary.

2.1.4 Implementation

A set of routines was written to form a hierarchical tree structure with evaluated matrices on each of the admissible or inadmissible leaves. They are capable of constructing an approximation of the domain matrices $[\vec{D}'_\Omega]$, which are used to evaluate the right hand side of the discrete kinematics system of equations (7). The advantages of using the approximation $[\vec{D}'_\Omega]$ instead of the fully populated $[\vec{D}_\Omega]$ are summarized in the following points.

- Since we have taken careful care of that the amount of data required to store matrices in all admissible leaves is smaller than data storage of their fully populated counterparts, we know that the memory required to store the FMM sparse approximation of the domain matrices $[\vec{D}'_\Omega]$ will be less than the memory required to store $[\vec{D}_\Omega]$.

- Evaluation of the two lower order matrices for each admissible leaf is computationally less expensive than the computation of their full matrix block counterpart. For one, there are less matrix elements to evaluate; secondly one of the low-order matrices holds values only (see equation (9)) and not integrals. The elements of the second low-order matrix are integrals of slowly varying functions, thus less effort is needed to evaluate them.

- We are approximating three domain matrices, each holding integrals of one component of the gradient of the fundamental solution. Since one of the low-order matrices in admissible leaves holds values independent of $\vec{r}$, they are also independent of the gradient direction. Thus only one set of these matrices, which is common for all three directions, needs to be stored. This saves additional data storage space.

3 LAGRANGIAN PARTICLE TRACKING

The equation of particle motion is given by

$$\frac{d^2\vec{r}'}{dt'^2} = \vec{a}'(\vec{v}', \vec{u}'), \tag{11}$$

where $\vec{r}'$ is the location of the particle and $\vec{a}'$ is its acceleration, which depends on the particle velocity $\vec{v}'$ and on the fluid velocity $\vec{u}'$.

The importance of the forces acting on particles in laminar and turbulent flow, which contribute to its acceleration, is a topic of recent discussions. Armenio and Fiorotto[5], for example, analysed the importance of forces acting on a particle for different particle over fluid density ratios. We considered the equation for particle acceleration as was given by Maxey and Riley[6]:

$$\frac{d\vec{v}'}{dt'} = \vec{g} - \frac{\rho}{\rho_p}\vec{g} + \frac{\vec{u}' - \vec{v}'}{\tau_p} + \frac{\rho}{\rho_p}\frac{D\vec{u}'}{Dt'} + \frac{\rho}{2\rho_p}\left(\frac{d\vec{u}'}{dt'} - \frac{d\vec{v}'}{dt'}\right), \tag{12}$$

where $d/dt' = \partial/\partial t' + (\vec{v}' \cdot \vec{\nabla})$ and $D/Dt' = \partial/\partial t' + (\vec{u}' \cdot \vec{\nabla})$. ρ, ν are the fluid density and viscosity. ρ_p, d_p, τ_p are the particle density, particle diameter and $\tau_p = \rho_p d_p^2/\rho 18\nu$ is the particle relaxation time. The terms included in the equation are gravity, buoyancy, drag, pressure gradient term and added mass term. The equation is rewritten in non-dimensional form with u_0 and L being characteristic fluid velocity scale and characteristic problem length scale, yielding $\vec{u} = \vec{u}'/u_0$, $\vec{v} = \vec{v}'/u_0$, $t = t'u_0/L$:

$$\vec{a} = \frac{d\vec{v}}{dt} = \frac{A}{St}\left\{\vec{v}_s + \vec{u} - \vec{v}\right\} + \frac{3}{2}R\frac{\partial\vec{u}}{\partial t} + R\{(\vec{u} + \frac{1}{2}\vec{v}) \cdot \vec{\nabla}\}\vec{u}, \tag{13}$$

where the Stokes number is defined as

$$St = \frac{\rho_p d_p^2 u_0}{\rho 18\nu L}, \tag{14}$$

the settling velocity is

$$\vec{v}_s = \frac{d_p^2}{18\nu u_0}\left(\frac{\rho_p}{\rho} - 1\right)\vec{g} \tag{15}$$

parametes R and A are

$$R = \frac{\rho}{\rho_p + \frac{1}{2}\rho}, \qquad A = \frac{\rho_p}{\rho_p + \frac{1}{2}\rho} = \frac{1}{1 + \frac{1}{2}R\left(1 + \frac{\rho}{2\rho_p}\right)}. \tag{16}$$

In the case of very light particles ($\rho_p \ll \rho$, bubble limit) the parameters R and A tend to $R \to 2$, $A \to 0$. In the aerosol limit ($\rho_p \gg \rho$), the parameters are $R \to 0$, $A \to 1$, rendering the pressure gradient and added mass terms negligible. For fluid particles ($\rho_p = \rho$), we have $R = A = 2/3$.

With the acceleration of the particle given in equation (13) we may solve the particle equation of motion (11) by employing the 4^{th} order Runge-Kutta method (Press et al.[7]).We integrate the following six equations simultaneously:

$$\frac{dx}{dt} = v_x, \quad \frac{dv_x}{dt} = a_x, \qquad \frac{dy}{dt} = v_y, \quad \frac{dv_y}{dt} = a_y, \qquad \frac{dz}{dt} = v_z, \quad \frac{dv_z}{dt} = a_z \tag{17}$$

The unknowns are the particle location x, y, z and particle velocity v_x, v_y, v_z. The initial particle location and velocity must be known. We require a subroutine calculating the right hand sides of the six equations. In order to calculate the acceleration on the right hand side, the velocity of the fluid $\vec{u}$ at the location of the particle is needed.

Flow simulations were performed on a mesh consisting of hexahedral elements. The first task to be performed in order to find the velocity of the fluid at the location of the particle is to find the mesh element in which the particle resides. This problem has been considered by Zhou and Leschziner[8] as well as by Cheng et al.[9]. We used the technique proposed by Marchioli et al.[10]. We calculate outward pointing normal of each face of the mesh element. Then, we calculate dot product between the normal and vector connecting the element face and the particle location; $\vec{r} \cdot \vec{n}$. If dot products for all element faces are negative, then the particle is located in the element. See Fig. 2.

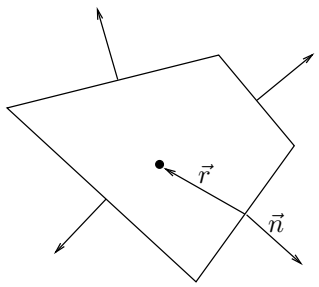

Figure 2: A 2D representation of a hexahedral mesh element. Element face normals $\vec{n}$ and particle position vector $\vec{r}$ are shown.

To avoid searching through the whole grid, the algorithm examines the mesh element in which the particle was located in the previous time step, secondly examines elements neighbouring this element and finally loops through all elements. When using adequate time steps, looping through all elements is necessary only in the case when particle leaves the computational domain. The proposed method works only for convex elements.

The next step is to interpolate the fluid velocity, which is known in the element's nodes, to the location of the particle. The solution of this problem in 2D was given in Ravnik et al.[11] and for 3D curvilinear grids by Marchioli et al.[10]. We extended the algorithm proposed in Ravnik et al.[11] to three dimensions.

Consider single element in a mesh made out of arbitrary six sided parallelepipedial elements and its counterpart in the local coordinate system (Fig. 3).

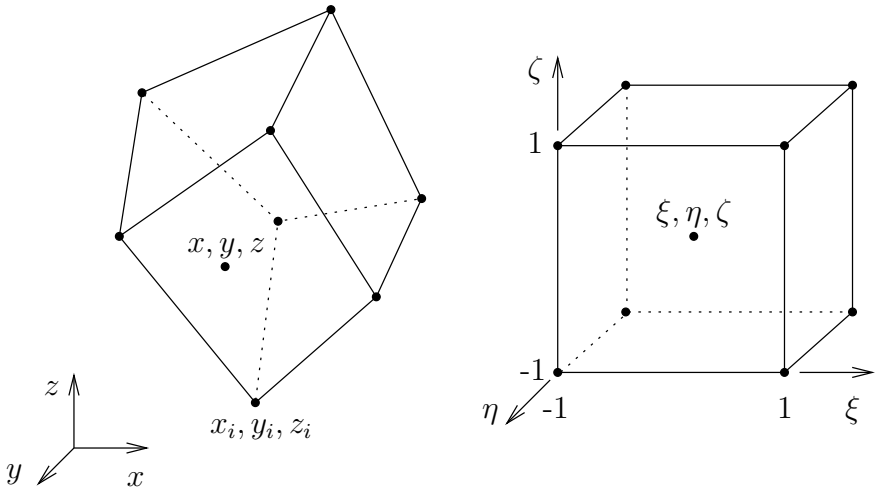

Figure 3: A 3D representation of a hexahedral mesh element and its counterpart in the local coordinate system

A point inside the element (x, y, z) may be written in local coordinate system (ξ, η, ζ) by using shape functions $\Phi_i(\xi, \eta, \zeta) = f(1, \xi, \eta, \zeta, \xi\eta, \xi\zeta, \eta\zeta)$ as

$$x(\xi, \eta, \zeta) = \sum_{i=1}^{8} \Phi_i(\xi, \eta, \zeta) x_i, \tag{18}$$

$$y(\xi, \eta, \zeta) = \sum_{i=1}^{8} \Phi_i(\xi, \eta, \zeta) y_i, \tag{19}$$

$$z(\xi, \eta, \zeta) = \sum_{i=1}^{8} \Phi_i(\xi, \eta, \zeta) z_i. \tag{20}$$

We are interested in exactly the inverse transformation; based on a known location (x, y, z) find the coordinates (ξ, η, ζ) in the local coordinate system. When (ξ, η, ζ) are known, we can interpolate any field function, which was the result of a CFD simulation, to the location (x, y, z).

Analytical solution for (ξ, η, ζ) given (x, y, z) for the above system of equations can not be found. The problem is solved numerically with the Newton-Raphson method[7]. The method uses first order Taylor expansion to write a linear system of equations for corrections of each unknown variable. Iteratively, the initial guesses for (ξ, η, ζ) are advanced towards the correct values. About a hundred iterations are needed to converge the solution to accuracy of 10^{-6}. This part of the particle tracking algorithm requires most CPU time.

The particle tracking algorithm can be summarized in the following points.

- Before simulation, pre-process mesh connectivity to make lists of neighbours for each mesh element. This greatly speeds up finding of the mesh element within which the particle is located.
- Use the element face normal times particle location dot product to find the mesh element within which the particle is located. Remember the particle's element of the previous time step. Check it and its neighbours first.
- Use the Newton-Raphson method to determine the velocity of the fluid at the location of the particle.
- Use the fourth order Runge-Kutta method to advance the particle location and velocity through time.

4 TEST CASE

In the lid driven cavity, the moving top lid induces a large primary vortex in the centre of a cubical enclosure. The size of the vortex increases with Reynolds number. Secondary vortices appear in the corners of the cavity, their position and strength changing with Reynolds number. Figure 4 shows the boundary conditions of the lid driven cavity graphically. The Reynolds number for this case is defined with the length of cavity's edge L and the top wall velocity u_0. The particle experiments performed by Tsorng et al.[1] were done at $Re = 130$, $Re = 470$ and $Re = 860$, thus we simulated the flow at these conditions.

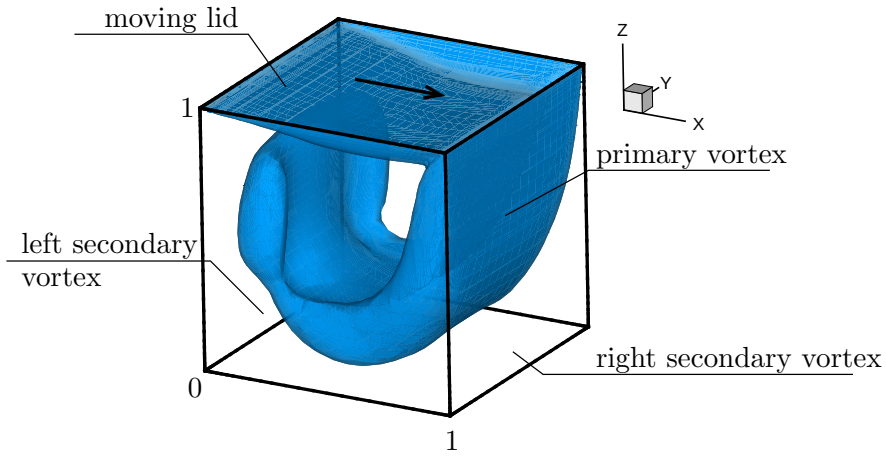

Figure 4: Geometry and boundary conditions of the lid driven cavity test case.

Study of behaviour of particles in lid driven cavity flows a topic of recent research. Kosinski et al.[12] numerically studied behaviour of dust clouds taking into account particle-particle interactions and two-way coupling. Tsorng et al.[1] studied behaviour of macro-

scopic rigid particles suspended in a fully three-dimensional lid driven cavity flow field. They used 2D PIV experimental technique to examine the particle motion. Our aim was to repeat and extend these experiments using the developed numerical technique. Tsorng et al.[1] reported most of their findings at $Re = 470$ and some at $Re = 130$ and $Re = 860$. We chose the same Reynolds numbers for our numerical experiments.

Particle boundary condition on the walls of the cavity was implemented as an elastic collision. A particle which is, after a time step, found outside of the problem domain, is mirrored back inside into the domain. Its velocity is also mirrored across a plane tangential to the wall.

We consider macroscopic particles with diameter $d_p = 3mm$ in a $L = 10cm$ lid driven cavity. The particles density was $\rho_p = 1210kg/m^3$. The particles were inserted into two fluids. The density of the first fluid was $\rho = 1210.605kg/m^3$ making the particles slightly buoyant. The density of the second fluid was $\rho = 1209.153kg/m^3$ making the particles non-buoyant with a small settling velocity. Flow Reynolds numbers $Re = 130$, $Re = 470$ and $Re = 860$ were considered. Lists of Stokes numbers and lid driving velocities are given in Tables 1 and 2. The simulation parameters were chosen such that the particle Stokes numbers are equal for both - buoyant and non-buoyant cases.

Table 1: Parameters used for Lagrangian particle tracking in lid driven cavity - case of non-buoyant macro particles. Fluid properties: $\rho = 1209.153kg/m^3$, $\nu = 17.3mm^2/s$; particle data: $d_p = 3mm$, $\rho_p = 1210kg/m^3$; cavity size $L = 10cm$; equation of motion (13) parameters: $A = 0.666822259$, $R = 0.666355483$, $v_s = -0.19mm/s$.

case	Re	$u_0[mm/s]$	St
d)	130	22.49	$6.50 \cdot 10^{-3}$
e)	470	81.31	$23.51 \cdot 10^{-3}$
f)	860	148.78	$43.03 \cdot 10^{-3}$

Table 2: Parameters used for Lagrangian particle tracking in lid driven cavity - case of slightly buoyant macro particles. Fluid properties: $\rho = 1210.6053kg/m^3$, $\nu = 37.2mm^2/s$; particle data: $d_p = 3mm$, $\rho_p = 1210kg/m^3$; cavity size $L = 10cm$; equation of motion (13) parameters: $A = 0.666555519$, $R = 0.666888963$, $v_s = 0.0659mm/s$.

case	Re	$u_0[mm/s]$	St
a)	130	48.36	$6.49 \cdot 10^{-3}$
b)	470	174.84	$23.49 \cdot 10^{-3}$
c)	860	319.92	$42.97 \cdot 10^{-3}$

Tsorng et al.[1] examined the reasons for the fact that the macro ($d_p = 3mm$) particles remain entrained in the primary vortex and do not enter the secondary corner vortices.

They argued that the particle size (i.e. diameter) is the crucial factor for this behaviour considering the chosen particle and fluid densities. By examining the flow field, they observed that only a narrow streamline corridor close to the top ($z = 1$) and right walls ($x = 1$) lead to the secondary vortices. They concluded that the $3mm$ particles were too large to enter this corridor and thus do not enter the secondary corner vortices. Their findings are verified by careful experimental work, whereas they only used a CFD approach for simulation of micro (fluid) particles paths, which were identical to fluid streamlines.

In order to complete the comparison, we examined the remaining situations using previously described and verified numerical algorithm for flow simulation and particles tracking. We inserted 1000 particles into flow fields. Initially, particles were randomly distributed within a cube in the centre of the cavity ($0.3 < x, y, z < 0.7$). Particle diameter was varied between $0.3mm$ and $2mm$, since we have already confirmed in previous section, that $3mm$ particles do not enter the secondary vortices. Both non-buoyant (Table 1) and buoyant (Table 2) macro particles were considered. Initial particle velocity was the set to be equal to the fluid velocity. Particles were tracked until all of them made at least ten revolutions in the primary vortex. Particle traces were recorded and examined to assert whether a particle did or did not enter the secondary vortices. We considered two secondary vortices. The right secondary vortex is considered to be in the bottom right region of the cavity, i.e. where $x \approx 1$ and $z \approx 0$, while the left secondary vortex is in the bottom left part of the cavity, i.e. where $x \approx 0$ and $z \approx 0$. Locations of secondary vortices are schematically shown in Figure 4. Results of entrainment of particles into secondary vortices are shown in Table 3 and described below.

At $Re = 130$ we observed that regardless of the diameter size, none of the buoyant particles enter the secondary vortices. All of the particles remain in the main vortex. Particles with a small diameter can be found throughout the primary vortex. Particles with a larger diameter move closer to the vortex core, but none actually enter the core. Figure 5 shows locations of particles of three diameters, demonstrating this fact.

Non-buoyant particles at $Re = 130$ with diameters less than $d_p \leq 0.5mm$ do not enter the secondary vortices. Most of the particles remain entrapped in the primary vortex, while some settle on the ground of the cavity. Particles with larger diameters enter the left secondary vortex, but they are not entrapped within. They rather fall through the slow flow of the left secondary vortex and settle on the ground. The larger the particle diameter, the more particles settle on the ground. None of the particles are able to enter the right secondary vortex.

Buoyant particles at $Re = 470$ with small diameter $d_p \leq 0.75mm$ enter the secondary vortices. Most of them enter the right vortex around the central plane. Thus, particles may be found in the entire cavity. An interesting phenomenon occurs with $d_p = 1mm$ particles. Although they do not enter the core of the right secondary vortex around the central $y = 0.5$ plane, they are sucked into the right corners of the cavity. Cavity with particle positions demonstrating this effect is shown in Figure 6. Particles with diameters $d_p = 1.25mm$ and $d_p = 1.5mm$ enter only the left vortex, while $d_p = 2mm$ particles do

not enter any of the secondary vortices.

Non-buoyant particles at $Re = 470$ with small diameter $d_p \leq 0.75mm$ enter the secondary vortices after first making a few revolutions in the main vortex. At $d_p = 0.75mm$ some particles settle on the ground of the cavity. At $d_p = 1mm$ the situation is similar to the buoyant case. The particles do not enter the right secondary vortex in the centre of the cavity, but do reach the corners of the cavity. However, since they are non-buoyant they settle on the bottom of the cavity. Cavity with particle positions demonstrating this effect is shown in Figure 6. Particles with diameter $d_p \geq 1.25mm$ enter only the left secondary vortex. Some of the particles, which enter the left secondary vortex, settle to the ground. As the particle diameter increases the number of particles entering the left vortex and the number of particles settling to the ground diminishes.

Buoyant particles at $Re = 860$ with diameters $d_p \leq 1mm$ enter the secondary vortices and can be, after a long time, found in the whole cavity. For $d_p = 1.25mm$ and $d_p = 1.5mm$ particles can be found only in the left secondary vortex, while none enter the right vortex. At $d_p = 2mm$ all particles remain in the primary vortex. Figure 7 shows these differences graphically.

Small non-buoyant particles at $Re = 860$ enter both secondary vortices when their diameter is less than $d_p \leq 1mm$. There is no settling of particles with $d_p = 0.3mm$. With increasing diameter more and more particles settle on the ground. At $d_p = 1.25mm$ and $d_p = 1.5mm$ the particles are able to reach only the left secondary vortex. Most of the particles, which enter the secondary vortex, settle to the ground. However at $d_p = 2mm$ no settling occurs at all. All of the particles are kept in the primary vortex, none enter the secondary vortices and none settle to the ground.

In summary, we discovered that only particles with diameters below a certain limit are able to enter in both secondary vortices. This limit depends on the Reynolds number. Particles with diameters above this limit are able to enter into the left secondary vortex only. As the particle diameter is increased even further, all particles remain in the primary vortex. The difference between buoyant and non-buoyant particles is expressed above all in the fact that the non-buoyant particles tend to settle to the ground. Since the secondary vortices are located at the bottom of the cavity, the non-buoyant particles are able to enter the secondary vortices with higher diameters than the buoyant particles.

5 CONCLUSIONS

A Lagrangian particle tracking algorithm was presented for simulation of dilute suspensions of particles in viscous flow. The physical model includes gravity, buoyancy, drag, pressure gradient and added mass effects. The particle equation of motion was solved and advanced through time by the Runge-Kutta method. Interpolation of fluid velocity to the location of the particle in the computational mesh was done by solving a non-linear system of equations using the Newton-Raphson method. Boundary element method based numerical algorithm for the solution of velocity-vorticity form of Navier-Stokes equations was used to simulate viscous flow.

Table 3: Entrance of particles in secondary vortices in the bottom right and left part of the cavity. Particles do not enter the secondary vortices (-), particles enter both secondary vortices (+), (l) particles enter the left secondary vortex only.

	$Re = 130$		$Re = 470$		$Re = 860$	
$d_p[mm]$	buoy.	non-buoy.	buoy.	non-buoy.	buoy.	non-buoy.
0.3	-	-	+	+	+	+
0.5	-	-	+	+	+	+
0.75	-	l	+	+	+	+
1.0	-	l	l	l	+	+
1.25	-	l	l	l	l	l
1.5	-	l	l	l	l	l
2.0	-	l	-	l	-	-

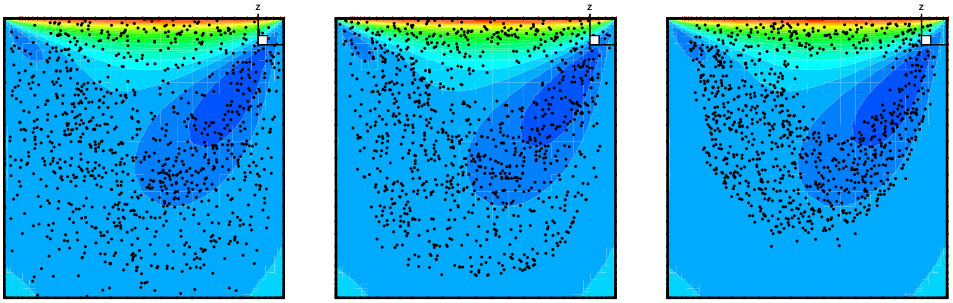

Figure 5: Buoyant particles with $d_p = 0.3mm$ (left), $d_p = 1mm$ (middle) and $d_p = 2mm$ (right) at $Re = 130$ after a long time shown on the $x - z$ plane. None of the particles enter the secondary vortices. Smaller particles find their way into the whole primary vortex, while larger are kept in a torus shape by the flow. Contours of u_x flow velocity component are shown in a slice at $y = 0.94$.

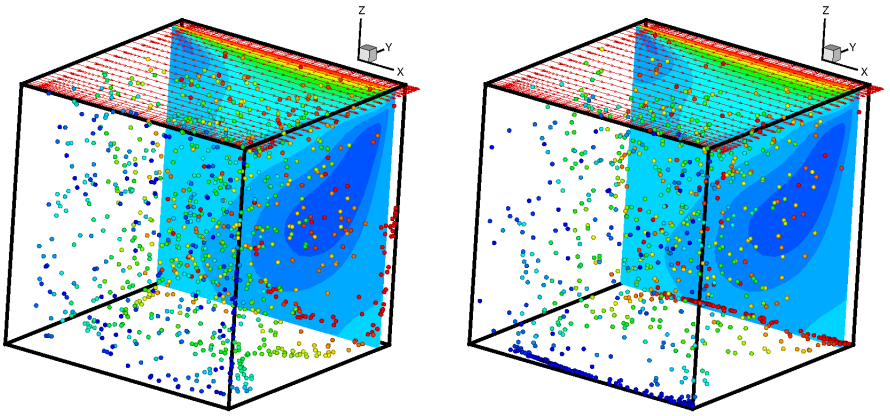

Figure 6: Buoyant (left) and non-buoyant (right) $d_p = 1mm$ particle positions after a long time in a cavity at $Re = 470$. None of the particles enter the right secondary vortex at $x = 1$, $y = 0.5$, while they are entrained into both $x = 1$ corners. Particle colour refers to its y coordinate. Contours of u_x flow velocity component are shown in a slice at $y = 0.94$.

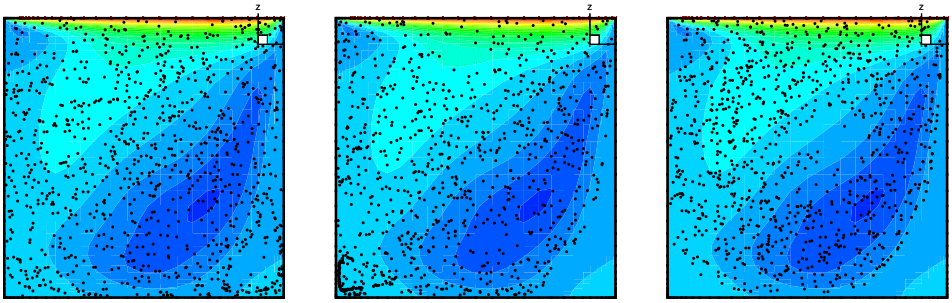

Figure 7: Buoyant particles with $d_p = 1mm$ (left), $d_p = 1.5mm$ (middle) and $d_p = 2mm$ (right) at $Re = 860$ after a long time shown on the $x - z$ plane. While the small particles enter both secondary vortices, the middle size particles enter only the left secondary vortex and the large particles do not enter any. Contours of u_x flow velocity component are shown in a slice at $y = 0.94$.

The particle tracking code was validated by revisiting a problem of particle movement in a cellular flow field. Very good agreement between present results and previously published results of other authors was observed.

The developed algorithm was used to study the behaviour of macroscopic slightly buoyant and slightly non-buoyant particles in a lid driven cavity. The flow in a lid driven cavity is fully three-dimensional and features a primary vortex in the main part of the cavity as well as secondary vortices in the corners of the cavity. We investigated the particle movement and discovered, that particles above a certain size, can not enter secondary vortices. The size limit increases with Reynolds number value. For the selected diameter data range, in case of Re=470 the limit is $d_p = 0.75$mm, and in case of Re=860 the limit is $d_p = 1.0$mm, both limits valid for the case of entering both secondary vortices. The physical reason for such behaviour was found to be the flow structure. Only small streamline paths lead to the secondary vortices, which can not be entered by larger particles. Thus the larger particles remain trapped in the primary vortex, while smaller particles are able to enter into the secondary vortices in the corners of the cavity.

REFERENCES

[1] S. J. Tsorng, H. Capart, D.C. Lo, J.S. Lai, and D.L. Young. Behaviour of macroscopic rigid spheres in lid-driven cavity flow. *Int. J. Multiphase Flow*, 34:76–101, 2008.

[2] J. Ravnik, L. Škerget, and Z. Žunič. Combined single domain and subdomain BEM for 3D laminar viscous flow. *Eng. Anal. Bound. Elem.*, 33:420–424, 2009.

[3] J. Ravnik, L. Škerget, and Z. Žunič. Comparison between wavelet and fast multipole data sparse approximations for Poisson and kinematics boundary – domain integral equations. *Comput. Meth. Appl. Mech. Engrg.*, 198:1473–1485, 2009.

[4] J. Ravnik, L. Škerget, and Z. Žunič. Fast single domain–subdomain BEM algorithm for 3D incompressible fluid flow and heat transfer. *Int. J. Numer. Meth. Engng.*, 77:1627–1645, 2009.

[5] V. Armenio and V. Fiorotto. The importance of the forces acting on particles in turbulent flows. *Phys. Fluids*, 13:2437–2440, 2001.

[6] M.R. Maxey and J.J. Riley. Equation of motion for a small rigid sphere in a nonuniform flow. *Phys. Fluids*, 26:883–889, 1983.

[7] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes - The Art of Scientific computing, Second Edition*. Cambridge University Press, 1997.

[8] Q. Zhou and M.A. Leschziner. An improved particle-locating algorithm for eulerian-lagrangian computations of two-phase flows in general coordinates. *Int. J. Multiphase Flow*, 25:813–825, 1999.

[9] H.-P. Cheng, J.-R. Chend, and G.-T. Yeh. A particle tracking technique for the Lagrangian-Eulerian finite element method in multi-dimensions. *Int. J. Numer. Meth. Engng.*, 39:1115–1136, 1996.

[10] C. Marchioli, V. Armenio, and A. Soldati. Simple and accurate scheme for fluid velocity interpolation for Eulerian-Lagrangian computation of dispersed flows in 3D curvilinear grids. *Comput. & Fluids*, 36:1187–1198, 2007.

[11] J. Ravnik, L. Škerget, M. Hriberšek, and Z. Žunič. Numerical simulation of dilute particle laden flows by wavelet BEM-FEM. *Comput. Meth. Appl. Mech. Engrg.*, 197/6-8:789–805, 2008.

[12] P. Kosinski, A. Kosinska, and AC Hoffmann. Simulation of solid particles behaviour in a driven cavity flow. *Powder Technology*, 191:327–339, 2009.