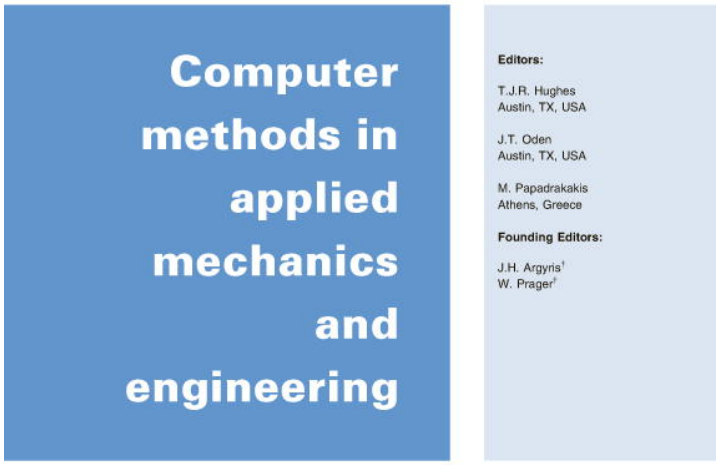
Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.

Volume 197, Issues 6–8, Published 15 January 2008 ISSN 0045-7825



Available online at
ScienceDirect
www.sciencedirect.com
http://www.elsevier.com/locate/cma

This article was published in an Elsevier journal. The attached copy is furnished to the author for non-commercial research and education use, including for instruction at the author's institution, sharing with colleagues and providing to institution administration.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/copyright

Author's personal copy

Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 197 (2008) 789–805
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Numerical simulation of dilute particle laden flows by wavelet BEM–FEM

J. Ravnik *, L. Škerget, M. Hriberšek, Z. Žunič

Faculty of Mechanical Engineering, University of Maribor, Smetanova ulica 17, SI-2000 Maribor, Slovenia

Received 25 January 2007; received in revised form 24 July 2007; accepted 12 September 2007
Available online 21 September 2007

Abstract

A wavelet transform based BEM and FEM numerical scheme was used to simulate laminar viscous flow. The velocity–vorticity formulation of the Navier–Stokes equations was used. The flow simulation algorithm was coupled with a Lagrangian particle tracking scheme for dilute suspensions of massless particles and particles without inertia. The proposed numerical approach was used to simulate flow and particle paths for two test cases: flow over a backward-facing step and flow past a circular cylinder. We present methods of calculating the pressure and stream function field at the end of each time step. The pressure field was used to calculate drag and lift coefficients, which enable qualitative comparison of our results with the benchmark. The stream function enabled the comparison of streamlines and massless particle paths in steady state low Reynolds number value flow fields, and thus provided an estimate on the accuracy of the particle tracking algorithm. Unsteady higher Reynolds number value flows were investigated in terms of particle distributions in vortex streets in the wake of the cylinder and behind the step. Sedimentation of particles without inertia was studied in the flow field behind a backward-facing step at Reynolds number value 5000.
© 2007 Elsevier B.V. All rights reserved.

PACS: 47.11.–j; 47.11.Hj; 47.11.Fg; 47.55.Kf

Keywords: Boundary element method; Velocity–vorticity formulation; Discrete wavelet transform; Lagrangian particle tracking; Backward-facing step; Bluff body flow; Dilute particle suspension

1. Introduction

Multiphase flows are frequently occurring in various fields, like unit operations in process engineering, flows in pipeline systems, environmental flows, etc. Dispersed flow consists of particles (drops, bubbles, solids), moving in a continuous phase (air, water). Particles interact with their surroundings (exchange of momentum, mass and energy), and in order to accurately compute interactions, particle locations must be known.

Development of algorithms for numerical simulation of particle laden flows has been a prime subject among researchers in the last decade. Lagrangian and Euler particle tracking methods have been combined with a wide variety of flow simulation techniques, for laminar [30] and turbulent [1] flows in a wide variety of engineering fields-from chemical, biological and nuclear to mechanical engineering.

Many times, the particles are small enough and have negligible inertia, so that their influence on the fluid may be neglected. Further simplification is possible, if we neglect the particle's mass, and thus simulate massless particles with Stokes number equal to zero. This approximation is the first to be considered, when combining a CFD code with a particle tracking algorithm, in order to assess the versatility and stability of the algorithm.

Flows over bluff bodies, such as flow over a circular cylinder, as well as flows in a channel with sudden expansion, i.e. flow over a backward-facing step, were investigated in

* Corresponding author. Tel.: +386 22207745; fax: +386 22207990.
E-mail addresses: jure.ravnik@uni-mb.si (J. Ravnik), leo@uni-mb.si (L. Škerget), matjaz.hribersek@uni-mb.si (M. Hriberšek), zoran.zunic@uni-mb.si (Z. Žunič).

0045-7825/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.cma.2007.09.007

Author's personal copy

this research. In both cases, steady state results exist with known benchmarks solutions [4,10]. These were used to check the validity of our flow simulation code. At the same time, since streamlines are identical to pathlines of massless particles, steady state simulation results were also used to check the accuracy and validity of our particle tracking algorithm. By increasing the Reynolds number values, we were able to track dilute suspensions of particles in the wake of the circular cylinder as well as in the vortex street behind the backward-facing step.

Slater and Young [26] developed a time-marching method for calculating two-dimensional, dilute, non-turbulent gas-particle flows using an Eulerian formulation and tested it on a flow over a circular cylinder. Liang et al. [17] examined particle mixing and reactive front dynamics in an open shallow flow of water past a circular cylinder. Wang et al. [32] simulated particle response to turbulence along its path, by comparing the properties of the particle phase and the gas phase using Lagrangian particle tracking for a turbulent flow over a backward-facing step. Yu et al. [35] simulated gas-particle flow in a single-side backward-facing step using LES for particles with Stokes numbers between 0.002 and 11.9. Jacobs et al. [14] work towards efficient tracking of inertial particles with high-order multidomain methods using backward-facing step for benchmarking. Siriboonluckul and Juntasaro [25] have investigated the performance of different turbulence models and source-term models for the turbulent separated particle-laden flows using the backward-facing step.

In this research we simulate laminar viscous Newtonian flow using the velocity–vorticity formulation of the incompressible Navier–Stokes equations. The velocity–vorticity formulation is an alternative to the more common primitive velocity–pressure equations. The pressure is eliminated from the system and replaced by the vorticity. The system is composed of a vector equation for the velocity field (kinematics equation) and a diffusion advection vector equation for the vorticity field. In two dimensions the vorticity is a scalar, so we need to solve a coupled system of three partial differential equations. In three dimensions, however, we are faced with six equations compared to four in the primitive velocity–pressure formulation. This apparent disadvantage is mitigated by our boundary element method (BEM) solution of the kinematics equations, which provides Dirichlet boundary conditions for the three vorticity equations yielding very well posed systems of linear equations, which may be solved efficiently using iterative solvers. The vorticity transport equation is written in a form directly applicable for first kind boundary value problems. The accurate calculation of boundary vorticity values is of utmost importance, since the vorticity is generated at the boundary. The solution of the transport equation only carries the vorticity, generated at the boundary, into the flow with diffusion and advection. Other authors proposed several different schemes for the calculation of boundary vorticity [34,36]. We propose to use BEM because of its unique advantage for solving the boundary vorticity values directly.

For the solution of the domain values Škerget et al. [28] and Ramšak et al. [21] proposed a subdomain BEM technique, while Hriberšek and Škerget [13] developed a segmentation technique. Although the subdomain technique results in sparse matrices, it still requires a considerable amount of computer memory and CPU time. These requirements were reduced by Žunič et al. [39], who proposed using FEM for the solution of the domain. We employed the latter approach.

In order to be able to perform simulations on dense grids, we used a wavelet compression algorithm on fully populated matrices, resulting from the BEM calculation of boundary vorticity, to further decrease the computer memory and CPU time requirements of the coupled BEM–FEM algorithm. A discrete wavelet transform for vectors of arbitrary length, developed by Ravnik et al. [23,24], was used. Ravnik et al. [24] developed a 2D large-eddy simulation code, that was based on the combination of wavelet transform, boundary element method and finite element method. Using a laminar version of this code enabled us to make unsteady simulations on dense grids with small time steps.

The rest of the paper is organized as follows. The second section states the governing equations: the kinematics equation, the vorticity transport equation, the pressure equation, the stream function equation and states presumptions and approximations for particle movement. The third section presents the numerical flow simulation and particle tracking algorithm and explains in detail the numerical method used to obtain a solution of each equation. Results of flow simulations and particle tracking for both numerical examples are given in the fourth section.

2. Governing equations

2.1. Assumptions and approximations

In this paper we assume an incompressible viscous Newtonian fluid with constant material properties: density $\rho = \rho_0$ and kinematic viscosity $\nu = \nu_0$. The continuity equation (mass conservation law) within this approximation

$$\vec{\nabla} \cdot \vec{v} = 0 \tag{1}$$

requires the velocity field $\vec{v}$ to be solenoidal, i.e. divergence free. The field functions were non-dimensionalized in the following manner: $\vec{v} \to \frac{\vec{v}}{v_0}$, $\vec{r} \to \frac{\vec{r}}{L}$, $\vec{\omega} \to \frac{\vec{\omega} L}{v_0}$, $t \to \frac{v_0 t}{L}$, $p \to \frac{p}{p_0}$, where p is pressure, L is the characteristic length and $\vec{\omega}$ is vorticity. With the above stated assumptions the momentum transport equation (momentum conservation law) in non-dimensional form reads as:

$$\frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \vec{\nabla})\vec{v} = -\frac{1}{Eu}\vec{\nabla} p + \frac{1}{Re}\nabla^2 \vec{v}, \tag{2}$$

with the Euler Eu number defined as $Eu = \frac{\rho_0 v_0^2}{p_0}$ and the Reynolds number being $Re = \frac{v_0 L}{\nu_0}$.

Author's personal copy

2.2. Velocity–vorticity formulation

Let us first sketch the derivation of the velocity–vorticity formulation of the Navier–Stokes equations. Vorticity $\vec{\omega}$ is defined as the curl of the velocity $\vec{\omega} = \vec{\nabla} \times \vec{v}$. By definition vorticity must be divergence free, $\vec{\nabla} \cdot \vec{\omega} = 0$. The continuity equation (1) is used to transform the vorticity definition into a kinematics equation. By applying the curl operator on the vorticity definition, we obtain

$$\nabla^2 \vec{v} + \vec{\nabla} \times \vec{\omega} = 0, \tag{3}$$

the kinematics equation, which relates the velocity and vorticity fields for every point in space and time. Both, the vorticity and the velocity fields must be solenoidal, in order for this equation to be fulfilled.

Taking a curl of the momentum equation (2) and after some algebraic manipulation (see [29] or [24] for details), the final form of the vorticity transport equation reads as

$$\frac{\partial \vec{\omega}}{\partial t} + (\vec{v} \cdot \vec{\nabla})\vec{\omega} = (\vec{\omega} \cdot \vec{\nabla})\vec{v} + \frac{1}{Re}\nabla^2 \vec{\omega}. \tag{4}$$

Eq. (4) equates the Stokes rate of change of vorticity on the left hand side with the vortex twisting and stretching term and the diffusion term on the right hand side. The vortex twisting and stretching term $(\vec{\omega} \cdot \vec{\nabla})\vec{v}$ vanishes in cases of planar flows, where the vorticity vector is always perpendicular to the plane of motion.

2.3. The pressure equation

In the velocity–vorticity formulation the pressure is eliminated from the momentum equation as a primary variable, while in the primitive variables approach, it appears in a gradient form, and as such it can cause numerical instabilities specially in solving incompressible viscous flow situations where the pressure can not be seen as a thermodynamic pressure, but rather as a force. Therefore pressure is calculated in a post-processor step after each time step has converged using calculated field functions. A pressure gradient is expressed from the momentum transport equation (2):

$$\vec{\nabla} p = Eu\left\{ -\frac{\partial \vec{v}}{\partial t} - (\vec{v} \cdot \vec{\nabla})\vec{v} + \frac{1}{Re}\nabla^2 \vec{v} \right\}. \tag{5}$$

Using the kinematics equation (3), the Laplace operator in Eq. (5) may be expressed with a curl of vorticity. By applying the divergence operator on Eq. (5) we obtain an elliptic partial differential equation for pressure of Poisson type:

$$\nabla^2 p = Eu\vec{\nabla} \cdot \left\{ -\frac{\partial \vec{v}}{\partial t} - (\vec{v} \cdot \vec{\nabla})\vec{v} - \frac{1}{Re}\vec{\nabla} \times \vec{\omega} \right\} = \vec{\nabla} \cdot \vec{f}_{\mathrm{p}}, \tag{6}$$

where we introduced a pressure vector $\vec{f}_{\mathrm{p}} = Eu\{-\frac{\partial \vec{v}}{\partial t} - (\vec{v} \cdot \vec{\nabla})\vec{v} - \frac{1}{Re}\vec{\nabla} \times \vec{\omega}\}$ to simplify notation.

2.4. The stream function equation

The stream function is also not a part of the non-linear system of equations in the $\vec{v} - \vec{\omega}$ formulation. Thus, it is calculated after the non-linear iteration loop at the end of each time step. The stream function is introduced as a vector potential $\vec{\psi}(\vec{r}, t)$ of the velocity field:

$$\vec{\nabla} \cdot \vec{v} = 0, \quad \vec{v} = \vec{\nabla} \times \vec{\psi}, \tag{7}$$

When we apply a curl to Eq. (7), take into account the definition of vorticity and choose $\vec{\nabla} \cdot \vec{\psi} = 0$, we obtain a Poisson type equation for the vector potential:

$$\nabla^2 \vec{\psi} = -\vec{\omega}, \tag{8}$$

which in 2D reduces to the stream function $\vec{\psi} = (0, 0, \psi)$.

2.5. Particle movement

In our simulations we focus on dilute suspensions, with particle volume fraction low enough, so that the particle–particle interaction can be neglected. The coupling between particle and fluid motion is dominated by momentum transfer. We will consider particles with diameter d_{p} much smaller than the length scales of the smallest flow structures. Furthermore, the density of the particles ρ_{p} is assumed to be much higher than the density of fluid so that the dominant force on an individual particle is the Stokes drag. The Stokes drag force for spherical particles with small particle Reynolds numbers is [19]

$$\vec{F} = 3\pi v \rho d_{\mathrm{p}}(\vec{v} - \vec{v}_{\mathrm{p}}), \tag{9}$$

where $(\vec{v} - \vec{v}_{\mathrm{p}})$ is the difference between the fluid velocity and the velocity of the particle $\vec{v}_{\mathrm{p}}$. Particles with negligible inertia move with a velocity that is equal to the sum of local fluid velocity and a settling velocity $\vec{v}_{\mathrm{s}}$. The settling velocity vector has the direction of the gravity force;

$$\vec{v}_{\mathrm{s}} = \frac{d_{\mathrm{p}}^2}{18v}\left(\frac{\rho_{\mathrm{p}}}{\rho} - 1\right)\vec{g}. \tag{10}$$

The particle Stokes number is defined by the following expression

$$St = \frac{d_{\mathrm{p}}^2 v_{\mathrm{o}} \rho_{\mathrm{p}}}{18 v \rho L} = \frac{\tau_{\mathrm{p}}}{\tau_{\mathrm{f}}}, \tag{11}$$

where L is the characteristic length of the flow, such as, the diameter of the cylinder or the height of the backward-facing step in our numerical examples. The τ_{p} is the relaxation time scale of the particle and $\tau_{\mathrm{f}} = L/v_{\mathrm{o}}$ is the time scale of the flow.

In our simulations we consider only one way coupling between particles and the fluid, i.e. only fluid flow affects the movement of particles.

3. Numerical scheme

The laminar flow of a viscous Newtonian fluid can be simulated by solving the non-linear system of two partial

Author's personal copy

differential equations: the kinematics equation (3) and the vorticity transport equation (4). At the end of each time step the pressure and stream function equations are solved as well as particle tracking is performed.

In this section we describe the complete numerical scheme, starting with the solution algorithm and explaining the solution of each individual equation in subsequent subsections.

3.1. Solution algorithm

The solution of the system of partial differential equations in vorticity–velocity formulation was obtained in planar geometry. Vorticity is a scalar quantity in 2D, thus the vorticity transport equation (4) is a scalar equation. The stream function is also a scalar quantity in 2D. The solution algorithm with references to sections of the paper is presented below:

- calculate integrals, that depend solely on the mesh geometry
- use wavelet compression on matrices of integrals
- set up starting positions for particle tracking
- begin time step loop
  - insert new particles into the flow
  - begin nonlinear loop
    - calculate boundary values by solving the kinematics equation by wavelet compressed BEM (Section 3.2)
    - calculate domain values by solving the kinematics equation by FEM (Section 3.4)
    - solve vorticity transport equation by FEM using the boundary values from BEM calculation (Section 3.3)
    - check convergence-repeat steps in the nonlinear loop until convergence is achieved
  - end nonlinear loop
  - calculate boundary pressure values by BEM (Section 3.5)
  - calculate domain pressure values by FEM (Section 3.6)
  - calculate boundary stream function values by method of integration (Section 3.7)
  - calculate domain stream function values by FEM (Section 3.7)
  - using the pressure, velocity and vorticity fields, calculate drag and lift coefficients (Section 3.8)
  - track particles (Section 3.9)
- end time step
- output results

3.2. Solution of the kinematics equation for boundary values by wavelet compressed BEM

The kinematics equation (3) is an elliptic Poisson type equation. The basic BEM derivation [33] yields the following boundary-domain integral form

$$c(\xi)\vec{v}(\xi)+\int_\Gamma \vec{v}(\vec{n}\cdot\vec{\nabla})u^*\,\mathrm{d}\Gamma = \int_\Gamma u^*(\vec{n}\cdot\vec{\nabla})\vec{v}\,\mathrm{d}\Gamma+\int_\Omega(\vec{\nabla}\times\vec{\omega})u^*\,\mathrm{d}\Omega,\quad \xi\in\Gamma \tag{12}$$

where u^* is the fundamental solution of the Laplace equation ($u^*=\frac{1}{2\pi}\ln\frac{1}{r}$ in 2D), $\vec{n}$ is the unit normal, ξ is the source point and $c(\xi)$ is the geometrical factor. By the use of algebraic manipulation, Gauss divergence clause and solenoidality of the fluid, Eq. (12) can be rewritten in a form without derivatives of the velocity and vorticity fields, i.e.

$$c(\xi)\vec{v}(\xi)+\int_\Gamma \vec{v}(\vec{n}\cdot\vec{\nabla})u^*\,\mathrm{d}\Gamma = \int_\Gamma \vec{v}\times(\vec{n}\times\vec{\nabla})u^*\,\mathrm{d}\Gamma+\int_\Omega(\vec{\omega}\times\vec{\nabla}u^*)\,\mathrm{d}\Omega. \tag{13}$$

Detailed derivation between (12) and (13) can be found in Škerget et al. [29] and in Ravnik et al. [24]. In order to obtain discrete solution of the integral kinematics equation (13), the boundary Γ is divided into E boundary elements Γ^e and the domain Ω is divided into C domain cells Ω_c with $\Gamma\approx\sum_{e=1}^{E}\Gamma_e$ and $\Omega\approx\sum_{c=1}^{C}\Omega_c$. Within each boundary element and each internal cell the field functions are approximated by boundary φ_i^{b} and domain φ_i^{d} shape functions $\vec{v}=\sum_{i=1}^{n_{\mathrm{b}}}\vec{v}^i\varphi_i^{\mathrm{b}}$ and $\vec{\omega}=\sum_{i=1}^{n_{\mathrm{d}}}\vec{\omega}^i\varphi_i^{\mathrm{d}}$, where n_{b} and n_{d} are the number of nodes in a boundary element and in a domain cell. In this paper we used three node quadratic boundary elements and nine node Lagrange domain cells. Inserting approximations into Eq. (13) we have

$$\begin{aligned}c(\xi)\vec{v}(\xi)+\sum_{e=1}^{E}\sum_{i=1}^{n_{\mathrm{b}}}\vec{v}^i\left(\int_{\Gamma_e}\varphi_i^{\mathrm{b}}(\vec{n}\cdot\vec{\nabla})u^*\,\mathrm{d}\Gamma\right)&=\sum_{e=1}^{E}\sum_{i=1}^{n_{\mathrm{b}}}\vec{v}^i\times\left(\int_{\Gamma_e}\varphi_i^{\mathrm{b}}(\vec{n}\times\vec{\nabla})u^*\,\mathrm{d}\Gamma\right)\\&+\sum_{c=1}^{C}\sum_{i=1}^{n_{\mathrm{d}}}\vec{\omega}^i\times\left(\int_{\Omega_c}\varphi_i^{\mathrm{d}}\vec{\nabla}u^*\,\mathrm{d}\Omega\right).\end{aligned} \tag{14}$$

There are three types of integrals in the above equation. In all, the integrand is a derivative of the fundamental solution multiplied by the shape function and the unit normal. The values of integrals depend solely on the mesh and the shape functions chosen, thus they can be calculated before starting the non-linear iterative procedure. The integrals are traditionally named as $h_{i,e}$, $\vec{h}^t_{i,e}$ and $\vec{d}_{i,c}$ respectively. In order to obtain a non-singular system of equations for boundary vorticities from Eq. (14), one must use its tangential form [29], obtained by taking a cross product of Eq. (14) with the unit normal in the source point:

$$\begin{aligned}c(\xi)\vec{n}(\xi)\times\vec{v}(\xi)+\vec{n}(\xi)\times\sum_{e=1}^{E}\sum_{i=1}^{n_{\mathrm{b}}}\vec{v}^i h_{i,e}&=\vec{n}(\xi)\times\sum_{e=1}^{E}\sum_{i=1}^{n_{\mathrm{b}}}\vec{v}^i\times\vec{h}^t_{i,e}+\vec{n}(\xi)\times\sum_{c=1}^{C}\sum_{i=1}^{n_{\mathrm{d}}}\vec{\omega}^i\times\vec{d}_{i,c}.\end{aligned} \tag{15}$$

Author's personal copy

We now define the planar geometry. The fluid flows in the x–y plane, vorticity has only one non-zero component in the z direction. Since we are dealing with velocity on the boundary, it is convenient to introduce normal velocity $v_n = n_x v_x + n_y v_y$ and tangential velocity $v_t = n_x v_y - n_y v_x$ and treat vorticity as a scalar quantity $\vec{\omega} \to \omega_z = \omega$. The integral vector $\vec{h}^t$ has also only one non-zero component and will be also treated as scalar h^t. When the cross products of Eq. (15) are calculated and the system of equations is simplified to planar geometry, we obtain a non-zero equation in z direction only:

$$c(\xi)v_t(\xi) + \sum_{e=1}^{E}\sum_{i=1}^{n_b} v_t^i h_{i,e} = -\sum_{e=1}^{E}\sum_{i=1}^{n_b} v_n^i h_{i,e}^t + \sum_{c=1}^{C}\sum_{i=1}^{n_d} \omega^i (n_x d_{i,c}^x + n_y d_{i,c}^y). \tag{16}$$

The source point ξ is set in all boundary nodes in order to obtain a system of linear equations. The field function values in nodes are represented by vectors $v_t^i \to \{v_t\}$, $v_n^i \to \{v_n\}$. The vorticity is divided into a boundary vector $\omega_{i\in\Gamma}^i \to \{\omega^\Gamma\}$ and a domain vector $\omega_{i\in\Omega\setminus\Gamma}^i \to \{\omega^{\Omega\setminus\Gamma}\}$. The integrals are assembled into matrices accordingly. With the boundary vorticities as unknowns, the matrix–vector form of Eq. (16) is

$$[D^\Gamma]\{\omega^\Gamma\} = ([C] + [H])\{v_t\} + [H^t]\{v_n\} - [D^{\Omega\setminus\Gamma}]\{\omega^{\Omega\setminus\Gamma}\}. \tag{17}$$

The matrices $[D^\Gamma]$, $[C]$, $[H]$ and $[H^t]$ are square, fully populated and unsymmetrical with the number of boundary nodes of rows and columns. Although full, the storage requirements for these matrices are not large, since the number of boundary nodes is very small compared to the number of domain nodes. On the other hand, the matrix $[D^{\Omega\setminus\Gamma}]$ is rectangular and also full and unsymmetrical and has dimensions of the number of boundary nodes rows and the number of domain nodes columns. The reason that the $[D^{\Omega\setminus\Gamma}]$ scales with the number of domain nodes is the fact that a fundamental solution of a Poisson type of equation does not exist and that we had to use the Laplace fundamental solution. This yielded a domain integral in the integral representation, which requires discretization of the domain and ultimately the storage of a matrix that scales with the number of nodes in the domain. Storing this matrix in memory requires huge amount of on-board memory and thus limits the size of meshes that can be used. To tackle this problem we employed a wavelet transform technique for rectangular matrices developed by Ravnik et al. [23]. The transform is based on Haar wavelets and the fast wavelet transform algorithm of Beylkin et al. [3]. The technique has already been successfully used for a planar LES of turbulent natural convection by Ravnik et al. [24].

In order to reduce the storage and CPU time requirements we will calculate the matrix vector product $[D^{\Omega\setminus\Gamma}]\{\omega^{\Omega\setminus\Gamma}\}$ using a wavelet compressed matrix of integrals. Since we will compress only the domain matrix, we will change only right hand side of our system of equations, leaving the system matrix unperturbed. By introducing an error estimation algorithm, we will be able to keep the error of right hand side calculation of the same order than the error of the solution of the system.

Let W be a wavelet transform matrix, which transforms a vector to a wavelet basis. It is set up using the Haar wavelet transform for vectors of arbitrary length [23] and is capable of transforming matrices of arbitrary size. The W transform is still in its essence the Haar wavelet transform. Before the Haar transformation, the vector is modified in such manner, that just the right number of wavelet coefficient end up zero. Not storing zeros makes it possible to apply the W transform to a vector with an arbitrary number of components and store only the same number of wavelet coefficients.

Since the product of wavelet matrix W and its transpose W^T is an identity, we may write

$$[D^{\Omega\setminus\Gamma}]\{\omega^{\Omega\setminus\Gamma}\} = W^T(\underbrace{W[D^{\Omega\setminus\Gamma}]W^T}_{[D_W^{\Omega\setminus\Gamma}]} W\{\omega^{\Omega\setminus\Gamma}\}). \tag{18}$$

The wavelet compressed matrix of integrals $[D_W^{\Omega\setminus\Gamma}] = W[D^{\Omega\setminus\Gamma}]W^T$ is the wavelet transform of all rows and columns of the original matrix of integrals $[D^{\Omega\setminus\Gamma}]$. Small (in absolute sense) elements of matrix $[D_W^{\Omega\setminus\Gamma}]$ may be zeroed without diminishing the accuracy of the matrix vector product in Eq. (18). Wavelet compression is preformed only once, prior to the iterative process. The matrices W and W^T are not actually stored in memory, but are rather calculated on the fly, thus omitting the need of additional storage. During the compression process we measure relative difference between full matrix times random vector product and compressed matrix times the same vector. The compression is stopped, when the relative difference reaches a prescribed criteria ϵ. In our previous work [24], we have determined that using compression with $\epsilon = 10^{-5}$ has virtually no influence on the accuracy of the flow simulation. The resulting sparse matrix is written in compressed row storage format to save computer memory. In each iteration we must instead of one full matrix time vector product perform a wavelet transform of the domain vorticity values $W\{\omega^{\Omega\setminus\Gamma}\}$, a sparse matrix times vector product $[D_W^{\Omega\setminus\Gamma}]W\{\omega^{\Omega\setminus\Gamma}\}$ and an inverse wavelet transform $W^T([D_W^{\Omega\setminus\Gamma}]W\{\omega^{\Omega\setminus\Gamma}\})$. All our tests showed that full matrix time vector product requires more CPU time, thus wavelet compression saves storage and CPU time. However, since the matrix times vector product (18) presents only a small fraction of the total CPU time requirements, the decrease of CPU time is almost negligible. On the other hand, the decrease of storage requirements is substantial and enables high density meshes to be used with our algorithm.

The final wavelet BEM based discrete form of the kinematics equation for the calculation of boundary vorticity values is

Author's personal copy

$$[D^{\Gamma}]\{\omega^{\Gamma}\} = ([C]+[H])\{v_t\} + [H^t]\{v_n\} - W^{\mathrm{T}}\left([D_W^{\Omega\setminus\Gamma}]W\{\omega^{\Omega\setminus\Gamma}\}\right). \quad (19)$$

Since the system matrix $[D^{\Gamma}]$ depends on the mesh geometry only and is constant throughout the simulation, the solution of the system is obtained using a direct solver with LU decomposition.

3.3. Solution of the diffusion advection equation by FEM

The vorticity transport equation (4) in 2D is a scalar partial differential equation of diffusion-advection type for vorticity ω. The partial time derivative is approximated by the following second order finite difference approximation

$$\frac{\partial\omega}{\partial t} \approx \frac{3\omega - 4\omega^n + \omega^{n-1}}{2\Delta t}, \quad (20)$$

where ω is the field function to be calculated in the next time step, ω^n is the field function in the present time step, and ω^{n-1} is the previous time step field function. The time step size is Δt. Having the approximation of the time derivative in mind, one can state the general scalar diffusion advection equation in the following form

$$\beta\omega + C_i\frac{\partial\omega}{\partial x_i} = D\nabla^2\omega + M \quad (21)$$

with β, C_i, D and M functions of time and location. Their values are set in accordance with Eqs. (20) and (4). Einstein summation notation is employed with $i = 1, 2$. The classical Garlekin FEM procedure is employed. One calculates integrals over each domain cell Ω_c using shape functions φ_k $(k = 1 \ldots n_d)$ as weighting functions. The order of the diffusion term is decreased by the Green's first theorem. Field functions are approximated across the cells: $\omega = \sum_{l=1}^{n_d}\varphi_l\omega_l$, $M = \sum_{l=1}^{n_d}\varphi_l M_l$. We use $n_d = 9$ node biquadratic shape functions to interpolate the field function behaviour in the Lagrangian cells Ω_c. Using these approximations, the final discrete form of the diffusion advection equation is

$$\begin{aligned}
&\sum_{l=1}^{n_d}\omega_l\left\{\int_{\Omega_c}\beta\varphi_k\varphi_l\,\mathrm{d}\Omega\right\} \\
&\quad+\sum_{l=1}^{n_d}\omega_l\left\{\int_{\Omega_c}\left[C_i+\frac{\partial D}{\partial x_i}\right]\varphi_k\frac{\partial\varphi_l}{\partial x_i}\,\mathrm{d}\Omega\right\} \\
&\quad=-\sum_{l=1}^{n_d}\omega_l\left\{\int_{\Omega_c}D\left(\frac{\partial\varphi_k}{\partial x_i}\frac{\partial\varphi_l}{\partial x_i}\right)\mathrm{d}\Omega\right\} \\
&\qquad+\sum_{l=1}^{n_d}\omega_l\left\{\int_{\Gamma_c}\varphi_k D\vec{\nabla}\varphi_l\cdot\mathrm{d}\vec{\Gamma}\right\} \\
&\qquad+\sum_{l=1}^{n_d}M_l\left\{\int_{\Omega_c}\varphi_k\varphi_l\,\mathrm{d}\Omega\right\}. \quad (22)
\end{aligned}$$

Solution of the resulting linear system of equations is obtained by the BI-CGSTAB solver [27]. The FEM solution method for the vorticity transport equation was developed by Žunič et al. [39] and tested in 2D by Ravnik et al. [24] and in 3D by Žunič et al. [37].

3.4. Solution of the kinematics equation for internal values by FEM

When the kinematics equations are solved by BEM, all boundary velocities and vorticities are known. It is now possible to make an explicit BEM calculation to get the internal velocities. However, for this, a new integral matrix is needed, which is fully populated and unsymmetrical and its size is number of all nodes squared, and as such limits the mesh size.

Alternatively, we have decided to use FEM instead of the explicit calculation by BEM. The kinematics equation (3) is of Poisson type and the BEM calculation provided the Dirichlet boundary conditions. The Poisson type equation is just a simplification of the general diffusion advection equation (21), solved in the previous section. Thus, solution may be obtained by setting $\beta = 0$, $C = 0$, $D = 1$ and $\vec{M} = \vec{\nabla}\times\vec{\omega}$ and solving Eq. (22) for each velocity component separately.

3.5. Calculation of pressure on the boundary by BEM

The pressure is not a part of the velocity–vorticity formulation. It can be calculated in a post-processing step, out of the converged flow fields for each time step. The computational time requirements for the calculation of pressure are virtually negligible compared to the requirements for the simulation of flow. The pressure equation (6) is a Poisson type equation. Its integral form [33] is:

$$\begin{aligned}
&c(\vec{\xi})p(\vec{\xi}) + \int_{\Gamma}p(\vec{n}\cdot\vec{\nabla})u^{\star}\,\mathrm{d}\Gamma \\
&\quad=\int_{\Gamma}u^{\star}(\vec{n}\cdot\vec{\nabla})p\,\mathrm{d}\Gamma - \int_{\Omega}(\vec{\nabla}\cdot\vec{f}_{\mathrm{p}})u^{\star}\,\mathrm{d}\Omega, \quad (23)
\end{aligned}$$

A vector equation $(\vec{\nabla}\cdot\vec{f}_{\mathrm{p}})u^{\star} = \vec{\nabla}\cdot(\vec{f}_{\mathrm{p}}u^{\star}) - \vec{f}_{\mathrm{p}}\cdot\vec{\nabla}u^{\star}$ and the Gauss clause are used to transform the domain integral in Eq. (23). The definition (5) stated that f_{p} is equal to the pressure gradient, thus the following is valid on the boundary $\vec{n}\cdot\vec{f}_{\mathrm{p}} = (\vec{n}\cdot\vec{\nabla})p$. Using this expression one can eliminate two boundary integrals and Eq. (23) is rewritten into

$$c(\vec{\xi})p(\vec{\xi}) + \int_{\Gamma}p(\vec{n}\cdot\vec{\nabla})u^{\star}\,\mathrm{d}\Gamma = \int_{\Omega}\vec{f}_{\mathrm{p}}\cdot\vec{\nabla}u^{\star}\,\mathrm{d}\Omega. \quad (24)$$

Since the boundary condition, which was used in the derivation, is undetermined to a constant, the solution of Eq. (24) is uniquely defined only, if the pressure in one node on the boundary is known. Thus, the reference pressure is prescribed in one of the nodes.

The integral equation (24) includes boundary as well as domain integrals. Using the same type of discretization as

Author's personal copy

for the kinematics equation ($p = \sum_{i=1}^{n_r} p^i \varphi_i^{\mathrm{b}}, \vec{f}_{\mathrm{p}} = \sum_{i=1}^{n_o} \vec{f}_{\mathrm{p}}^{\,i} \varphi_i^{\mathrm{d}}$), the discrete form of the pressure equation for the calculation of boundary pressure values may be stated as

$$c(\vec{\xi})p(\vec{\xi}) + \sum_{e=1}^{E} \sum_{i=1}^{n_r} p^i \left(\int_{\Gamma_e} \varphi_i^{\mathrm{b}} (\vec{n} \cdot \vec{\nabla}) u^{\star} \, \mathrm{d}\Gamma \right) = \sum_{c=1}^{C} \sum_{i=1}^{n_o} \vec{f}_{\mathrm{p}}^{\,i} \cdot \left(\int_{\Omega_c} \varphi_i^{\mathrm{d}} \vec{\nabla} u^{\star} \, \mathrm{d}\Omega \right). \tag{25}$$

The integrals depend on mesh geometry only, thus may be calculated only once. A collocation scheme is employed to obtain a system of equations for boundary pressure values. The matrices of integrals are large, full and unsymmetrical. Since the calculation can be performed in a post processing step, we omitted wavelet compression. Instead we wrote a MPI [9] based parallel code, which divided the matrices across processors, and made the calculation.

3.6. Calculation of pressure in the domain

With the boundary pressure values known from the BEM algorithm, we face a problem of calculating the domain pressure values. A Poisson type equation with Dirichlet boundary conditions must be solved. Similarly as with the kinematics equation, we used FEM method to do this calculation. The following values were used in Eq. (22): $\beta = 0$, $C = 0$, $D = 1$, $M = -\vec{\nabla} \cdot \vec{f}_{\mathrm{p}}$.

3.7. The stream function

In order to be able to solve the stream function equation (8) we must first find the boundary stream function values. Since the stream function is undetermined up to a constant, we may arbitrarily choose a value in one point on the boundary. From that point, the values on the boundary are calculated by means of integration of the stream function definition. In two dimensions the tangential derivative of the stream function is

$$\frac{\partial \psi}{\partial t} = n_x \frac{\partial \psi}{\partial y} - n_y \frac{\partial \psi}{\partial x} = n_x v_x + n_y v_y. \tag{26}$$

Starting from the point with a known boundary value, the following integral

$$\psi_{\Gamma} = \int_{\Gamma} n_x v_x + n_y v_y \, \mathrm{d}\Gamma \tag{27}$$

is used to obtain boundary stream function values. They are then used as Dirichlet type boundary conditions for a finite element method solution of the stream function equation (8). The following values were used in Eq. (22): $\beta = 0$, $C = 0$, $D = 1$, $M = \omega$.

3.8. Drag and lift coefficients

When dealing with flows over bluff bodies qualitative comparison of the results is usually given by comparing lift c_{L} and drag c_{D} coefficients. For a 2D case they are the non-dimensional force acting on the body, defined by

$$\left\| \begin{matrix} c_{\mathrm{D}} \\ c_{\mathrm{L}} \end{matrix} \right\| = \frac{1}{\rho L \frac{1}{2} v_0^2} \vec{F}, \quad \vec{F} = -\int_{\Gamma} p \, \mathrm{d}\vec{\Gamma} + \int_{\Gamma} \tau \cdot \mathrm{d}\vec{\Gamma}, \tag{28}$$

where Γ is the boundary of the bluff body and $\tau_{ij} = \eta(\partial v_i/\partial x_j + \partial v_j/\partial x_i)$ is the strain tensor for a Newtonian incompressible fluid. Taking into account the definitions of vorticity and strain tensor, the coefficients may be expressed by

$$\left\| \begin{matrix} c_{\mathrm{D}} \\ c_{\mathrm{L}} \end{matrix} \right\| = 2 \int_{\Gamma} \left(-\frac{1}{Eu} p \vec{n} + \frac{2}{Re} (\vec{n} \cdot \vec{\nabla}) \vec{v} + \frac{1}{Re} \left\| \begin{matrix} n_y \\ n_x \end{matrix} \right\| \omega \right) \mathrm{d}\Gamma, \tag{29}$$

where $\vec{n} = (n_x, n_y)$ is the unit normal on the boundary, pointing out of the flow domain. By looking at Eq. (6), we observe that the pressure is multiplied with the Euler number. In the equation for drag and lift coefficients (29) the pressure is divided by the Eu number value, thus c_{D} and c_{L} are independent of the Euler number, and depend solely on the Reynolds number value.

3.9. Particle tracking

We developed a Lagrangian particle tracking scheme, which requires that particle positions must be updated every time step using the simulated flow field and the particle settling velocity. Using small particle tracking time steps (usually 1/10 of the flow field simulation time step) we advanced the particle positions in time between subsequent flow fields by small displacements $\mathrm{d}\vec{s} = (\vec{v} + \vec{v}_{\mathrm{s}})\,\mathrm{d}t$. The velocity of the particle was set to be equal to the sum of the local fluid velocity $\vec{v}$ and particle settling velocity $\vec{v}_{\mathrm{s}}$ (Eq. (10)).

The main problem with the numerical implementation of this approach is the efficient calculation of the fluid velocity at the location of each particle. This task requires two actions: firstly, to find the mesh element, within which the particle is located, and secondly to interpolate the velocity flow field inside the element to find the flow velocity at the exact location of the particle.

In our numerical model, we use 9 node Lagrange elements with 4 node geometry. This means that four nodes are corner nodes, four nodes are located on straight lines between the corner nodes and the ninth node is inside the elements. Thus, mathematically the problem may be formulated as: there are four points in $\mathbb{R}^2$ connected with straight lines. We must figure out, whether the fifth (particle location) is inside or outside the element formed by the straight lines.

Fig. 1 shows such an element, defined by corner nodes (x_i, y_i) and its transformation to the local coordinate system (ξ, η). A point inside the element x, y may be written in local coordinate system (ξ, η) by using shape functions (30):

Author's personal copy

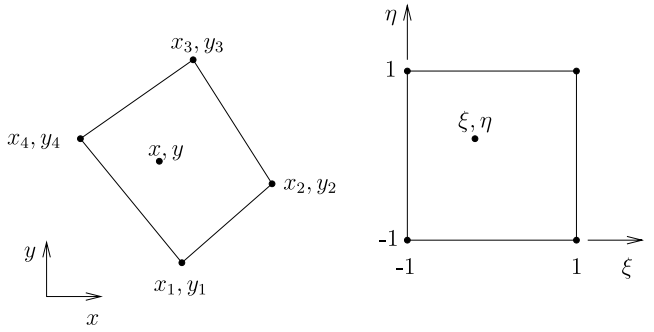


Fig. 1. An element with linear geometry and its transformation to the ξ, η coordinate system.

$$\varphi_1 = \frac{1}{4}(1-\eta)(1-\xi), \quad \varphi_2 = \frac{1}{4}(1-\eta)(1+\xi),$$
$$\varphi_3 = \frac{1}{4}(1+\eta)(1+\xi), \quad \varphi_4 = \frac{1}{4}(1+\eta)(1-\xi) \tag{30}$$

as

$$x(\xi,\eta) = \sum_{i=1}^{4} \varphi_i(\xi,\eta) x_i, \quad y(\xi,\eta) = \sum_{i=1}^{4} \varphi_i(\xi,\eta) y_i. \tag{31}$$

We are interested in exactly the inverse transformation; based on a known location (x, y) find the coordinates (ξ, η) in the local coordinate system. When (ξ, η) are known we can interpolate any field function, which was the result of a CFD simulation, to the location x, y. When shape functions (30) are inserted into Eq. (31) we obtain the following expressions for η

$$\eta = \frac{\{4x - (x_1 + x_2 + x_3 + x_4)\} + \xi\{x_1 - x_2 - x_3 + x_4\}}{\{x_4 - x_1 + x_3 - x_2\} + \xi\{x_1 - x_4 + x_3 - x_2\}} = \frac{A + B\xi}{C + D\xi}, \tag{32}$$

$$\eta = \frac{\{4y - (y_1 + y_2 + y_3 + y_4)\} + \xi\{y_1 - y_2 - y_3 + y_4\}}{\{y_4 - y_1 + y_3 - y_2\} + \xi\{y_1 - y_4 + y_3 - y_2\}} = \frac{E + F\xi}{G + H\xi}, \tag{33}$$

where capital letters stand for linear combinations of corner node coordinates. Eqs. (32) and (33) are used to solve for ξ. A second degree polynomial equation is obtained:

$$(BH - FD)\xi^2 + (AH + BG - ED - FC)\xi + (AG - EC) = 0. \tag{34}$$

If the discriminant of the polynomial (34) is negative or the equation is trivial, the element is not regular-for example, lines between corners cross each other. This does never happen, if the mesh is generated correctly. If both solutions for ξ are outside the interval $[-1 \ldots 1]$, the point in question (x, y) does not lie within the element. If at least one of the ξ is in the interval $[-1 \ldots 1]$, we use it to calculate η out of Eq. (32) or Eq. (33). The solution of Eq. (34) answers two questions at the same time-whether the point is located within the element and at the same time provides the local coordinates, which may be used for interpolation of flow field within the element.

The following summarises our particle tracking algorithm.

- The algorithm maintains a list of particles. It keeps record of position of each particle, and weather the particle is active or inactive. A particle becomes inactive, when the flow carries it out of the domain through the outflow boundary or when it settles down onto the bottom boundary of the domain. The particle active/inactive status is updated after each time step by comparing its position against the fluid computational mesh boundaries. All of the particles that hit the bottom of the domain settle there permanently, i.e. become inactive and are not released back into the flow.
- The algorithm divides each flow simulation time step to 10 particle path integration steps. It uses the interpolation routine to find the flow velocity at the particle location and integrates the particle path between two flow simulation time steps.
- When the next flow simulation time step is reached, the algorithm uses the new flow field and repeats the procedure.
- The flow simulation time step must be small enough, so that particles do not move for more than one element within one time step.

4. Numerical examples

To show the efficiency and accuracy of the proposed numerical scheme, flow simulation and particle tracking was examined for two numerical examples: flow past a circular cylinder and flow over a backward-facing step.

4.1. Flow past a circular cylinder

Consider a fluid flowing in a laminar fashion on a infinite x–y plane in x direction. A pressure gradient drives the fluid. The velocity profile is uniform, the streamlines are parallel lines $y = const$. The flow is two-dimensional, the domain extents infinitely in the z direction. In this section we will examine what happens, when such fluid encounters a circular obstacle. The nondimensional parameter that governs the flow fields in this case is the Reynolds number, defined as $Re = v_0 D / v_0$, where v_0 is the velocity of the unperturbed flow field, D is the diameter of the cylinder and v_0 is the fluid viscosity.

Since it is impossible the model an infinite domain, we chose the domain large enough to enable setting of boundary conditions. The domain dimensions are 16×32, the diameter of the cylinder is 1, its centre is located at (8, 8). The computational domain is presented in Fig. 2. The fluid enters the domain on the left side with constant velocity $\vec{v} = (1, 0)$. The boundary condition at the top and bottom walls assumes fully developed horizontal flow

Author's personal copy

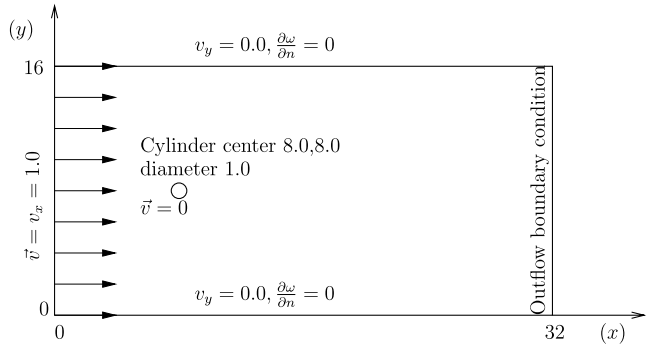


Fig. 2. Geometry and boundary conditions of the flow past a circular cylinder.

$(v_y = 0, \frac{\partial\omega}{\partial y} = 0)$, i.e. the upper and lower boundary of the domain are assumed far enough so the perturbation of the flow field, caused by the cylinder, does not cause any mass or vorticity flux through them. The cylinder is assumed to be solid with non-slip boundary conditions $(\vec{v} = 0)$. The flow exits the domain on the right hand side. Since vortices are formed behind the cylinder they must be transported through the outflow boundary without causing any disturbance of the flow field, such as cause reflections from the outflow back into the flow field. To achieve this, a convective outflow boundary condition [20,16] is used. A monochromatic wave equation

$$\frac{\partial\phi}{\partial t} + c\frac{\partial\phi}{\partial n} = 0, \tag{35}$$

is used to transport normal velocity and vorticity $(\phi = v_x, \omega)$ from the nodes nearest to the outflow boundary to the boundary nodes. We have used the inlet velocity for the wave speed $c = v_x^{\text{inlet}} = 1$.

Three mesh densities were employed to solve the problem. The coarsest mesh consisted of 4060 nodes (see Fig. 3), the other two had 8200 and 32,400 nodes respectively.

The flow at $Re = 20$ and $Re = 40$ is steady. A nondimensional time step of 1 was used; steady state was achieved after around 100 time steps. The flow is symmetric across $y = 8.0$, hence there is no lift; $c_L = 0.0$. Two additional parameters are measured at steady state. The separation angle Θ is the angle between the $y = 8$ line and the line connecting the cylinder centre and the point on the cylinder where vorticity vanishes. The recirculation length $\mathscr{L}$ is the length of the recirculation zone behind the cylinder measured along the $y = 8.0$ line. The steady state results for $Re = 20$ and $Re = 40$ are presented graphically in Figs. 4 and 5 respectively. Both the vorticity contours and the stream lines are shown in both figures.

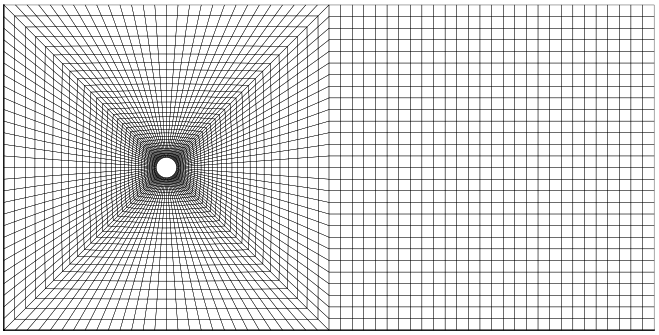


Fig. 3. The coarsest computational mesh with 4060 nodes for the flow past a circular cylinder.

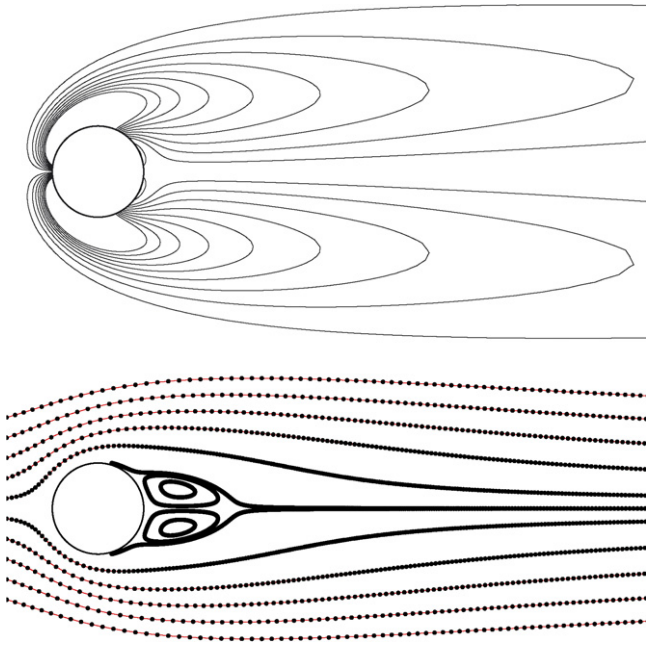


Fig. 4. Vorticity contours (top) and streamlines (bottom) for $Re = 20$. Calculated streamlines (lines) are in a very good agreement with the massless particles tracked through the flow field (circles).

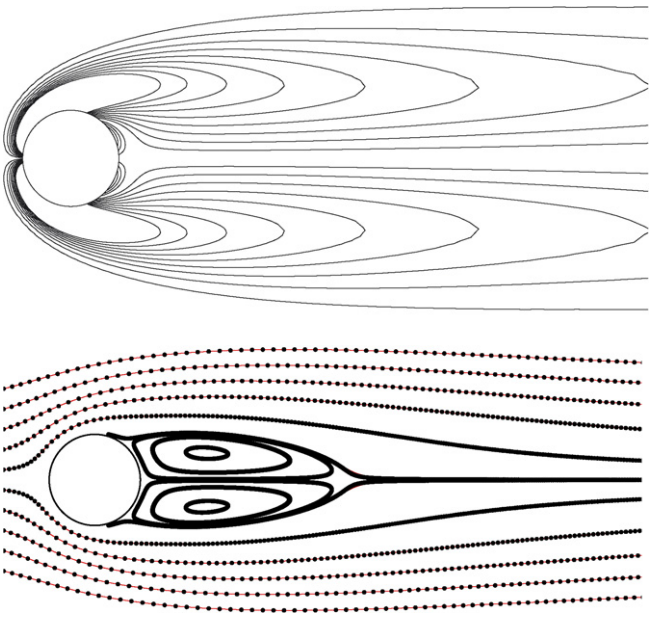


Fig. 5. Vorticity contours (top) and streamlines (bottom) for $Re = 40$. Calculated streamlines (lines) are in a very good agreement with the massless particles tracked through the flow field (circles).

Author's personal copy

In a steady state flow field, the streamlines are identical with pathlines, i.e. massless particle paths must coincide with the streamlines. These two steady state results were used to check the accuracy and validity of the particle tracking algorithm. For both Reynolds numbers, a nearly perfect match is obtained when comparing streamlines and particle paths. Comparison is shown in Figs. 4 and 5.

The calculated separation angle, recirculation length and drag coefficient are shown in Table 1. For comparison results of other authors are presented along with a FVM commercial code CFX. We can clearly see, that our present results are in very good agreement with the reference results.

At $Re = 100$ the flow behind the cylinder is unsteady. There is no need to artificially induce the instability. The system starts oscillating at around $t = 100$. Our calculation was preformed with a nondimensional time step of 0.1 until $t = 200$. The time variation of the lift coefficient is characterized by the Strouhal number $Str = \frac{D}{v_0 t_0}$, where t_0 is the shredding period of the lift. The vorticity contours and streamlines for $t = 200$ are shown in Fig. 6. The drag coefficient, the amplitude of the lift coefficient and Strouhal number are presented in Table 2. The time dependence of both coefficients is shown graphically in Fig. 7. It may be noted, that the frequency of oscillations of the drag coefficient is twice the oscillating frequency of the lift coefficient.

Analysing the results in Table 2 we can see, that the Strouhal number is in good agreement with reference results. The c_D value and the amplitude of c_L are slightly lower than the reference results, however in good agreement with the commercial FVM code CFX.

Massless particles were released into the $Re = 100$ flow field. Four particles were released each time step at location $x = 7$ and $y = 7.8, 7.9, 8.1, 8.2$. Bearing in mind that the cylinder centre is at $(8, 8)$ and its diameter is 1, we have released the particles just upstream of the cylinder. When the flow is still under development, the recirculation region is very long. This is illustrated with massless particles at $t = 50$ in Fig. 8. When the flow is fully developed,

Table 1
Flow past a circular cylinder (separation angle Θ, recirculation length $\mathscr{L}$ and drag coefficient c_D for $Re = 20$ and $Re = 40$)

	$Re = 20$			$Re = 40$		
	$\mathscr{L}$	Θ (°)	c_D	$\mathscr{L}$	Θ (°)	c_D
Tritton [31] (exp)	–	–	2.22	–	–	1.48
Coutanceau and Buoard [6] (exp)	0.73	42.3	–	1.89	52.8	–
Fornberg [8] (num)	0.91	–	2.00	2.24	–	1.50
Dennis and Chang [7] (num)	0.94	43.7	2.05	2.35	53.8	1.52
Calhoun [5] (num)	0.91	45.5	2.19	2.18	54.2	1.62
Ramšak et al. [21] (num)	0.94	43.0	2.14	2.27	53.3	1.59
CFX 5.7 $N = 4.060$	0.96	–	2.21	2.29	–	1.64
CFX 5.7 $N = 8.200$	0.95	–	2.22	2.32	–	1.65
CFX 5.7 $N = 32.400$	0.94	–	2.23	2.26	–	1.66
Present $N = 4.060$	0.93	43.59	2.16	2.23	53.93	1.57
Present $N = 8.200$	0.93	43.86	2.20	2.23	53.80	1.61
Present $N = 32.400$	0.94	44.04	2.23	2.24	53.99	1.65

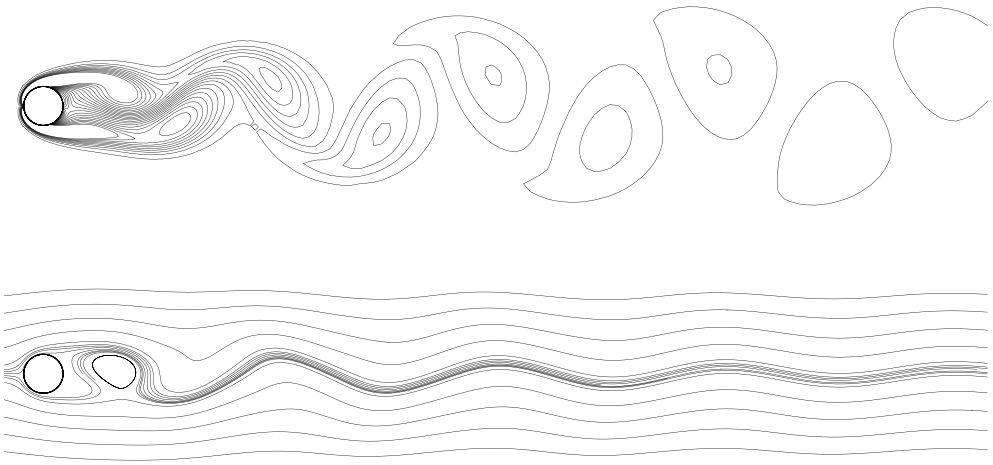


Fig. 6. Vorticity contours (top) and streamlines (bottom) for $Re = 100$.

Author's personal copy

Table 2
Flow past a circular cylinder (the drag coefficient c_D, the amplitude of the lift coefficient c_L and Strouhal number Str for $Re = 100$)

	$Re = 100$		
	c_D	c_L	Str
Calhoun [5] $N = 204.800$	1.330 ± 0.014	± 0.298	0.175
Braza et al. [4] (num)	1.364 ± 0.015	± 0.25	–
Liu et al. [18] (num)	1.350 ± 0.012	± 0.339	0.164
Ramšak et al. [21] $N = 112.500$	1.283 ± 0.007	± 0.149	0.168
CFX 5.7 $N = 4.060$	1.198 ± 0.0012	± 0.118	0.161
CFX 5.7 $N = 8.200$	1.335 ± 0.0038	± 0.210	0.161
CFX 5.7 $N = 32.400$	1.346 ± 0.042	± 0.225	0.161
CFX 5.7 $N = 129.600$	1.335 ± 0.033	± 0.216	0.161
Present $N = 4.060$	1.217 ± 0.003	± 0.186	0.159
Present $N = 8.200$	1.250 ± 0.0026	± 0.190	0.158
Present $N = 32.400$	1.300 ± 0.006	± 0.195	0.159

$t = 200$, the flow field is periodically symmetric, which can also be seen by examining the distribution of massless particles in Fig. 9.

4.2. Flow over a backward-facing step

Consider a two-dimensional channel. An incompressible fluid, driven by a pressure gradient, is flowing in a laminar fashion along the channel. The flow rate is constant; the streamlines are parallel to the channel walls. The velocity profile is parabolic, with zero velocity on the walls of the channel and the maximal velocity in the middle of the channel. In this section we examine what happens, when such flow encounters a one-sided sudden expansion of the channel, i.e. flows over a backward-facing step.

Flow over a two-dimensional backward-facing step is one of the classical examples, which is used to check the accuracy and stability of numerical schemes. Two decades ago, Armaly and co-workers [2] measured the velocity field using laser Doppler measurements. Gartling [10] made a numerical benchmark. Later, the backward-facing step flow at $Re = 800$ was used for validation of different numerical algorithms Keskar and Lyn [15] and Gresho et al. [11]. Among others Grigoriev and Dargush [12] and Ramšak et al. [22] checked the boundary element based code.

The height of the step is one half of the channel height. The characteristic length scale is the height of the channel H. At inflow a parabolic velocity profile is prescribed. It is non-dimensionalized by the velocity v_0. The velocity v_0 is chosen is such a manner that it ensures the same flow rate with an uniform profile. The Reynolds number is defined as $Re = \frac{Hv_0}{v_0}$. The ratio between the length and height of the channel was the same as in the benchmark solution; $L/H = 30$. The channel with boundary conditions is sketched in Fig. 10. The walls have a non-slip boundary condition. When we simulated a steady state solution, we employed a developed profile boundary condition at the outflow boundary, i.e. all normal fluxes were set to zero. When the simulation was unsteady, the convective outflow boundary condition was used.

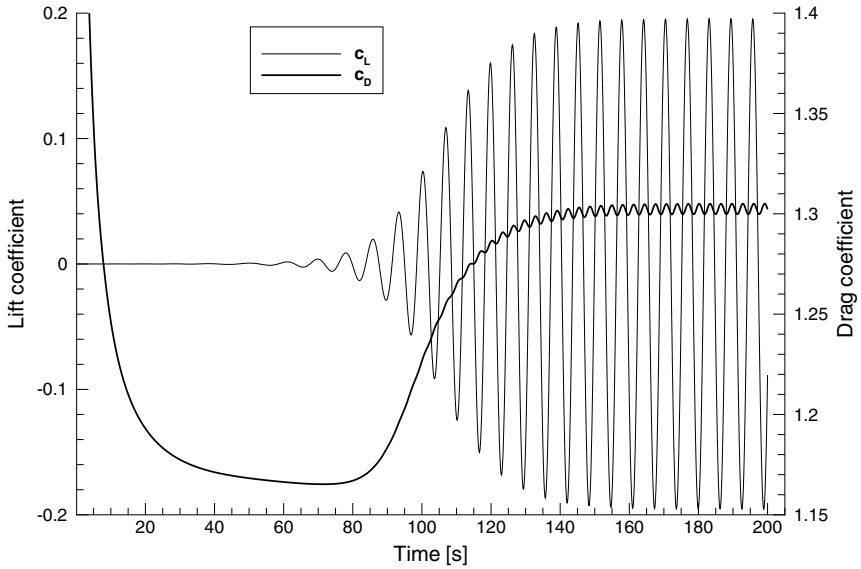


Fig. 7. Drag c_D and lift c_L coefficients with respect to time, $Re = 100$.

Fig. 8. Massless particle flow behind a circular cylinder at $Re = 100$ when the flow is still under development; $t = 50$.

Author's personal copy

Fig. 9. Massless particle flow behind a circular cylinder at $Re = 100$ in the developed flow stage; $t = 200$.

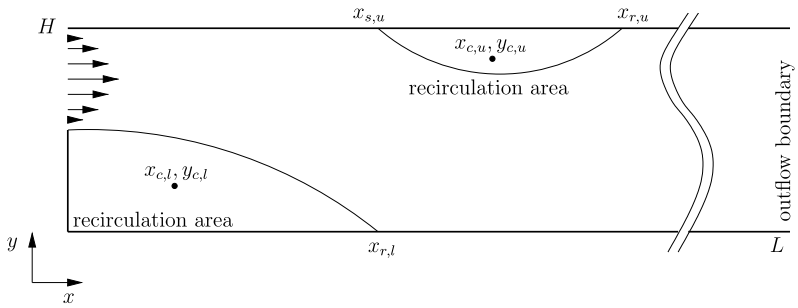


Fig. 10. Flow over a backward-facing step in a channel. The centre of the bottom eddy is marked with $x_{c,l}$, $y_{c,l}$, the recirculation length on the bottom wall is $x_{r,l}$. On the top wall, the flow separation takes place at $x_{s,u}$ and the recirculation is finished at $x_{r,u}$. The centre of the top eddy is at $x_{c,u}$, $y_{c,u}$.

We compared solutions with different wavelet compression ratios, with the aim to find a wavelet compression ratio, which does not affect the accuracy of simulation. Other authors claim, that the mesh with 500×50 nine-node Lagrange elements, which has 101,101 uniformly distributed nodes, is dense enough to solve the problem at $Re = 800$. Using this mesh, we changed the wavelet compression ratio and compared the solution with benchmark results of Gartling [10] and solutions of other authors. The comparison is shown in Tables 3 and 4. We noticed, that the difference to the benchmark solution is negligible for $\epsilon \leq 10^{-5}$. At higher compression ratios, the accuracy of the solution deteriorates. Fig. 11 shows velocity profiles for different compression ratios. We observe, that the profiles up to $\epsilon = 10^{-5}$ are virtually identical. At higher compression ratios, the difference between profiles is the largest in the area where recirculation regions end. At $\epsilon = 10^{-4}$ the lower eddy is too short, while the upper eddy is moved towards the left. Thus we conclude, that $\epsilon = 10^{-5}$ is the compression ratio, which may be used in simulation without wavelet compression diminishing the accuracy of results.

This finding is in accordance with findings for a lid driven cavity test case and for natural convection in a cavity test case, where $\epsilon = 10^{-5}$ was reported as the right compression ratio in our previous work [24].

Vorticity contours and streamlines with particles are shown in Figs. 12 and 13. Contour values are the same

Table 3
Laminar flow over a backward-facing step at $Re = 800$ (presentation of results of other authors and their methods)

Author	Grigoriev and Dargush [12]	Keskar and Lyn [15]	Gartling [10]	Gresho et al. [11]	Gresho et al. [11]	Ramšak et al. [22]	Žunič et al. [38]
Method	BEM	Spectral	FEM	Finite diff.	Spectral	BEM	BEM-FEM
Mesh	17.878	3737	E:129.681	1920×128	≈8.000	400.000	32.000
$x_{r,l}$	6.10	6.0964	6.10	6.082	6.10	6.10	6.11
$x_{c,l}$		3.392	3.350	3.375			
$y_{c,l}$		0.296	0.3	0.2968			
$\omega(x_{c,l}, y_{c,l})$		−2.2620	−2.283				
$\psi(x_{c,l}, y_{c,l})$		−0.03420	−0.0342	−0.034195	−0.0342		
$x_{r,u}$	10.47	10.4785	10.48	10.4648	10.49	10.48	10.49
$x_{s,u}$	4.85	4.8534	4.85	4.8388	4.86	4.86	4.87
$x_{c,u}$		7.447	7.400	7.4375			
$y_{c,u}$		0.815	0.8	0.8125			
$\omega(x_c, y_c)$		1.1527	1.322				
$\psi(x_c, y_c)$		0.50653	0.5064	0.50661	0.5065		

Author's personal copy

Table 4
Laminar flow over a backward-facing step at $Re = 800$ (presentation of present results for different compression ratios)

ϵ	10^{-4}	2×10^{-5}	10^{-5}	10^{-6}	10^{-7}
Compression	0.956	0.879	0.847	0.748	0.536
$x_{r,l}$	5.62	6.045	6.087	6.100	6.096
$x_{c,l}$	3.09	3.33	3.33	3.36	3.36
$y_{c,l}$	0.32	0.3	0.3	0.3	0.3
$\omega(x_{c,l}, y_{c,l})$	−2.420	−2.277	−2.269	−2.29	−2.29
$\psi(x_{c,l}, y_{c,l})$	0.0454	0.03509	0.03439	0.0340	0.0342
$x_{r,u}$	10.58	10.435	10.489	10.478	10.478
$x_{s,\ u}$	4.57	4.818	4.847	4.855	4.851
$x_{c,u}$	7.02	7.41	7.41	7.47	7.47
$y_{c,u}$	0.77	0.81	0.81	0.82	0.82
$\omega(x_c, y_c)$	1.407	1.187	1.197	1.09	1.10
$\psi(x_c, y_c)$	0.473	0.503727	0.505824	0.506964	0.506607

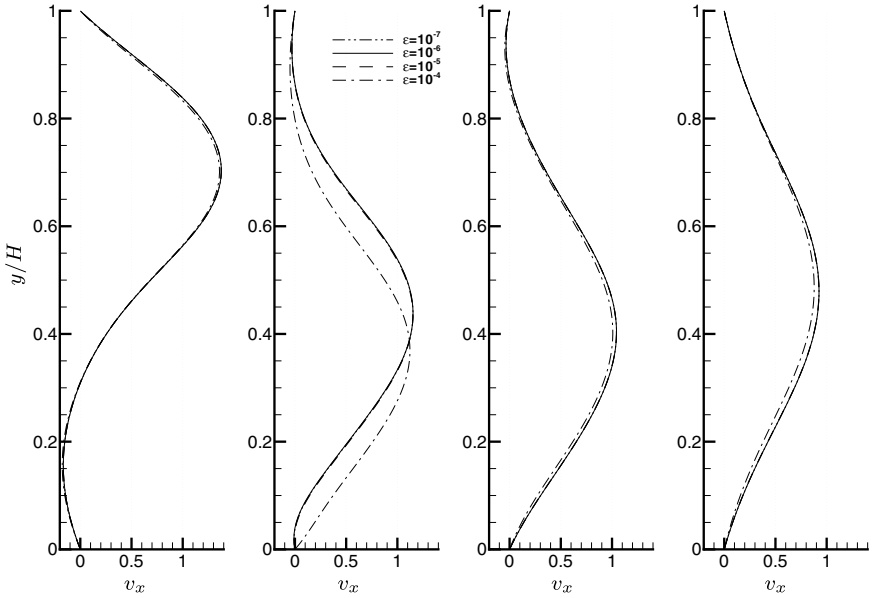


Fig. 11. Flow over a backward-facing step, $Re = 800$. Comparison of velocity profiles for different compression ratios. Profiles at $x/H = 3, 6, 9, 12$ (from left to right) are shown.

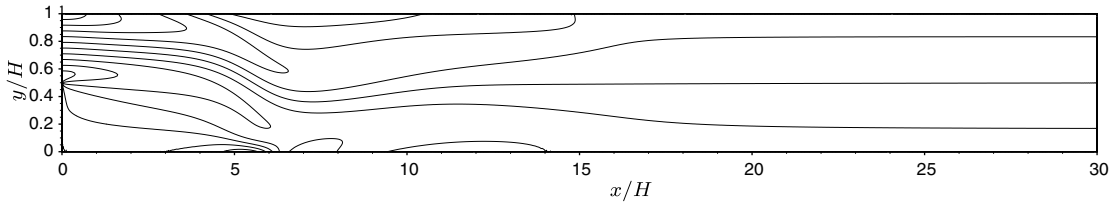


Fig. 12. Flow over a backward-facing step, $Re = 800$, $\epsilon = 10^{-6}$. Lines of constant vorticity (−8, −6, −4, 0, 2, 4, 6, 8, 10). The vertical axis is scaled by a factor of 4.

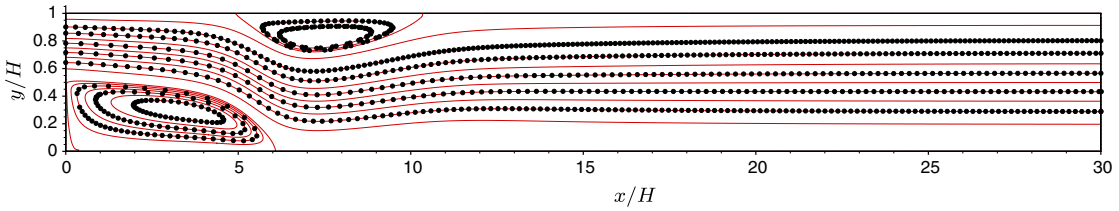


Fig. 13. Flow over a backward-facing step, $Re = 800$, $\epsilon = 10^{-6}$. Streamlines (−0.03, −0.025, −0.02, −0.015. −0.01, −0.005, 0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.502, 0.504). The vertical axis is scaled by a factor of 4.

Author's personal copy

as in the benchmark paper of Gartling [10]. By tracking particles in the steady flow field we were again able to confirm good agreement between streamlines and particle pathlines.

With the compression ratio established, we proceeded to increase the Reynolds number to have an unsteady flow situation which enables particle tracking. In order to be able to see formation of larger eddies, we increased the ratio between the height of the channel and the height of the step from 1 to 6. The Reynolds number, based on the height of the step, was $Re = 5000$.

Considering water as our fluid ($\rho = 10^3$ kg/m^3, $v = 10^{-6}$ m^2/s), particles with density $\rho_p = 3 \times 10^3$ kg/m^3, the step height 10 cm and inflow fluid velocity $v_0 =$ 5 cm/s, three cases were considered. The Stokes number was set to 3×10^{-4}, 1.2×10^{-5} and 1.2×10^{-7}. Corresponding particle diameters and settling velocities are shown in Table 5. Simulation duration was 10,000 time steps spanning 1000 time units. Fig. 14 shows the number of particles in the computational domain through time for the three Stokes numbers considered. In order to do statistical analysis on the same dataset, we decided to start analyses at $t = 400$. The first 4000 time steps were not used, since the flow has not yet reached self similar state in all three Stokes number cases. The average number of particles in the domain in each time step, $t > 400$, was $29{,}497 \pm 432$ for $St = 1.2 \times 10^{-7}$, $27{,}357 \pm 311$ for $St = 1.2 \times 10^{-5}$ and $18{,}583 \pm 171$ for $St = 3 \times 10^{-4}$. The average ratio between the area occupied by particles and the domain area was $\approx 10^{-7}$ for $St = 1.2 \times 10^{-7}$, $\approx 10^{-5}$ for $St = 1.2 \times 10^{-5}$ and $\approx 10^{-4}$ for $St = 3 \times 10^{-4}$, thus the assumption of dilute particle suspension is valid for all three cases.

We presume that the particle distribution at the inflow is statistically uniform, thus particles were inserted into the flow directly above the step with a uniform distribution. Let us define the number of particles per unit of length and time at the inflow as particle flux J_0. At the bottom of the domain, the sedimentation particle flux is J_s, while at the outflow of the domain, we have J_o. When simulation reached a statistically steady state regime ($t \geq 400$; number of particles in the domain does not change significantly over time), we measured relative particle flux in the sedimentation region $\chi_s = J_s/J_0$ and in the outflow region

Table 5
Particle diameters and Stokes numbers used for tracking a dilute suspension of particles in a flow field behind a backward-facing step

St	d_p (μm)	v_s (m/s)
3×10^{-4}	60	3.92×10^{-3}
1.2×10^{-5}	12	1.57×10^{-4}
1.2×10^{-7}	1.2	1.57×10^{-6}

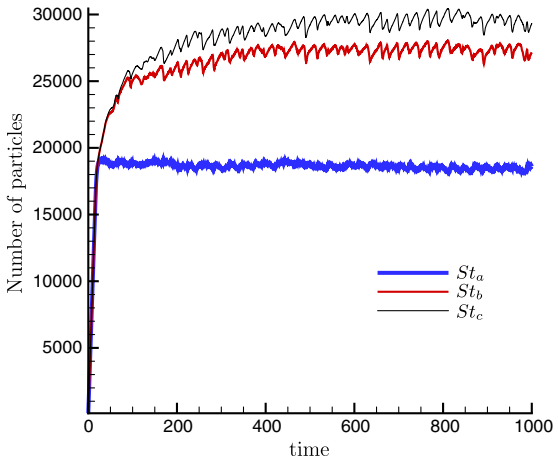


Fig. 14. Number of particles in the computational domain through time for Stokes numbers $St_a = 3 \times 10^{-4}$, $St_b = 1.2 \times 10^{-5}$ and $St_c = 1.2 \times 10^{-7}$. This graph was used to determine the start of the statistical analysis at $t = 400$.

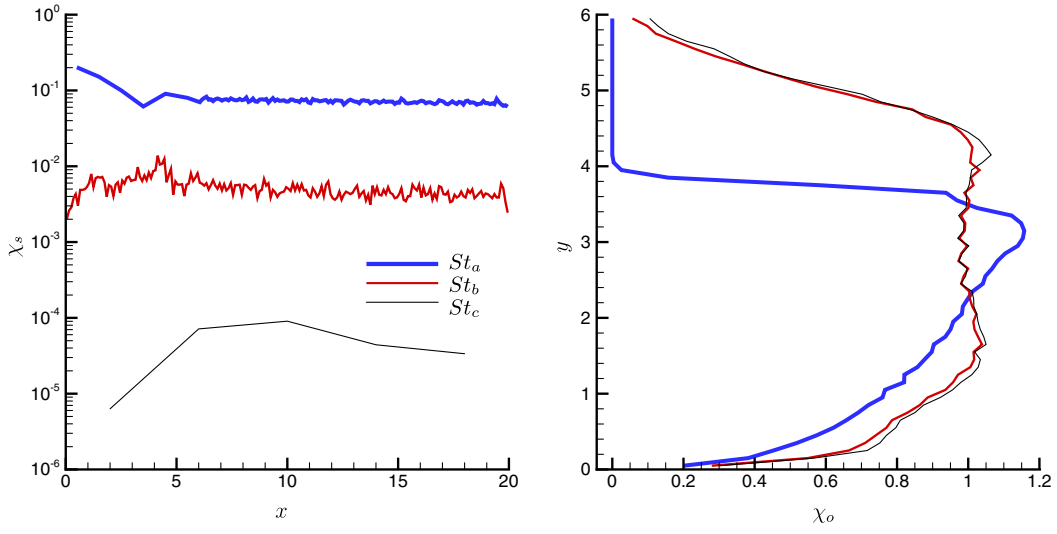


Fig. 15. Relative particle flux for Stokes numbers $St_a = 3 \times 10^{-4}$, $St_b = 1.2 \times 10^{-5}$ and $St_c = 1.2 \times 10^{-7}$, left graph shows the sedimentation region χ_s, right graph shows the outflow region, χ_o. The particle flux in areas, where relative particle flux is equal to one, is equal to the inflow particle flux.

Author's personal copy

$\chi_o = J_o/J_0$. Fig. 15 shows χ_s and χ_o for the three particle Stokes numbers. Almost all of the particles with the lowest Stokes number flow out of the domain through the outflow boundary. Only a small fraction $\chi_s \approx 10^{-4}$ settle on the lower boundary. On the other hand, a much larger share of particles at the highest Stokes number settle on the lower boundary $\chi_s \approx 10^{-1}$.

Fig. 16 shows particle distribution behind the step for all three Stokes numbers at the same time instant $t = 672$. In order to achieve clear representation of particles in the figure, the size of the particles has been vastly ($\approx 10^4$ times) exaggerated. The vortices forming in the wake of the step capture the particles and carry them through the flow field. The vortex forming on the top wall does not capture particles in the case of the largest Stokes number $St = 3 \times 10^{-4}$, since the particle settling velocity is too large and particles settle below the top wall vortex. The largest number of particles caught in the top wall vortex is, as expected, in the case of the lowest Stokes number. The lager the Stokes number the less effect does a fluid vortex have on particle dispersion. These findings are in accordance with the results provided by Yu et al. [35]. Since we are dealing with dilute particle suspension, where particles do not influence the fluid flow, we can observe exactly the same location of vortices in all three Stokes number cases, the only difference being the number of particles dispersed by the vortex.

5. Conclusions

We presented a numerical algorithm for simulation of 2D incompressible viscous Newtonian fluid flow with tracking of dilute suspensions of particles. The algorithm solves the velocity–vorticity formulation of Navier–Stokes equations using the wavelet transform, boundary element method and finite element method. The particles are tracked in a Lagrangian manner under the assumptions that they do not have an impact on the fluid and have negligible inertia, so that their velocity can be modelled as a sum of local fluid velocity and a settling velocity.

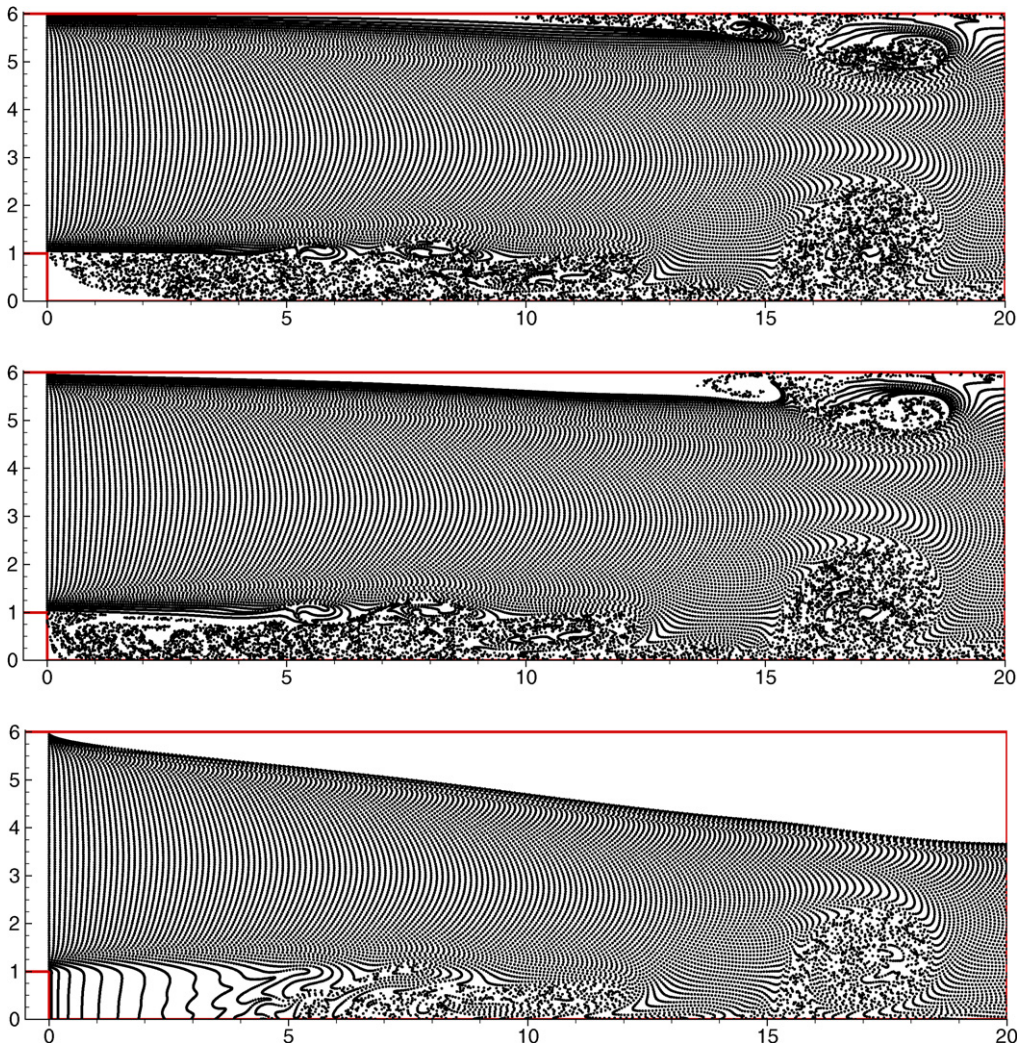


Fig. 16. Particle flow over a backward-facing step $Re = 5000$. Particle locations at $t = 672$ are shown for $St = 1.2 \times 10^{-7}$ (top), $St = 1.2 \times 10^{-5}$ (middle) and $St = 3 \times 10^{-4}$ (bottom).

Author's personal copy

Using the developed numerical scheme, we simulated incompressible viscous flow and particle tracking past a circular cylinder. The flow simulation results and comparison of particle paths and streamlines show exceptionally good agreement for the steady state ($Re = 20, 40$). We have calculated the drag coefficient and the amplitude of the lift coefficient and the Strouhal number for the unsteady flow at $Re = 100$. The Strouhal number is in good agreement with the reference values, while drag and lift coefficient amplitudes are slightly smaller than reference values, however in good agreement with the commercial code CFX. Particle tracking demonstrated that particles are caught in vortices in the wake of the cylinder and are thus carried downstream. Comparison of lines of constant stream function in the steady state example with massless particle paths demonstrated the accuracy of our particle tracking algorithm.

The steady state backward-facing step flow at $Re = 800$ was used to determine the optimal wavelet compression ratio. We were able to confirm our previous statement [24], which was obtained using different numerical examples. Locations of vortex centres and recirculation lengths were found to be in good agreement with reference results. Comparison of lines of constant stream function and massless particle paths also showed nearly perfect agreement.

By tracking a dilute particle suspension in a flow over a backward-facing step at $Re = 5000$ with three different particle Stokes numbers, we were able to compare particle flow behaviour and sedimentation in a very dynamic unsteady flow field. Comparing the settled particle flux and the outflow particle flux, we observed that the settled particle flux increases dramatically with the increasing Stokes number. We also found good correlation between the increasing Stokes number and the decreasing number of particles caught by vortices in the flow field.

References

[1] B. Arcen, A. Tainère, B. Oesterlé, On the influence of near wall forces in particle-laden channel flows, Int. J. Multiphase Flow 32 (2006) 1326–1339.

[2] B.F. Armaly, F. Durst, J.C.F. Pereira, B. Schönung, Experimental and theoretical investigation of backward-facing step flow, J. Fluid Mech. 127 (1983) 473–496.

[3] G. Beylkin, R. Coifman, V. Rokhlin, Fast wavelet transforms and numerical algorithms, Commun. Pure Appl. Math. 44 (1991) 141–183.

[4] M. Braza, P. Chassaing, H.H. Minh, Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder, J. Fluid Mech. 165 (1986) 79–130.

[5] D. Calhoun, A Cartesian grid method for solving the two-dimensional stream function-vorticity equations in irregular regions, J. Comput. Phys. 196 (2002) 231–275.

[6] M. Coutanceau, R. Buoard, Experimental determination of main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. Steady flow, J. Fluid Mech. 79 (1977) 231–256.

[7] S.C.R. Dennis, G. Chang, Numerical solution for steady flow past a circular cylinder at Reynolds number up to 100, J. Fluid Mech. 42 (1970) 471–489.

[8] B. Fornberg, A numerical study of steady viscous flow past a circular cylinder, J. Fluid Mech. 98 (1980) 819–855.

[9] M.P.I. Forum, MPI: A Message-Passing Interface Standard, version 1.1 ed., 1995.

[10] D. Gartling, A test problem for outflow boundary conditions-flow over a backward-facing step, Int. J. Numer. Methods Fluids 11 (1990) 953–967.

[11] P.M. Gresho, D.K. Gartling, J.R. Torczynski, K.A. Cliffe, K.H. Winters, T.J. Garratt, J.W. Goodrich, Is the steady viscous incompressible two-dimensional flow over a backward-facing step at $Re = 800$ stable? Int. J. Numer. Methods Fluids 17 (1993) 501–541.

[12] M.M. Grigoriev, G.F. Dargush, A poly-region boundary element method for incompressible viscous fluid flows, Int. J. Numer. Methods Engrg. 46 (1999) 1127–1158.

[13] M. Hriberšek, L. Škerget, Boundary domain integral method for high Reynolds viscous fluid flows in complex planar geometries, Comput. Methods Appl. Mech. Engrg. 194 (2005) 4196–4220.

[14] G. Jacobs, D. Kopriva, F. Mashayek, Towards efficient tracking of inertial particles with high-order multidomain methods, J. Comput. Appl. Math. 206 (2007) 392–408.

[15] J. Keskar, D.A. Lyn, Computations of laminar backward-facing step flow at $Re = 800$ with a spectral domain decomposition method, Int. J. Numer. Methods Fluids 29 (1999) 411–427.

[16] M. Kobayashi, J. Periera, J. Sousa, Comparison of several open boundary numerical treatments for laminar recirculating flows, Int. J. Numer. Methods Fluids 16 (1993) 403–419.

[17] Q. Liang, P.H. Taylor, A.G.L. Borthwick, Particle mixing and reactive front motion in unsteady open shallow flow-modelled using singular value decomposition, Comput. Fluids 36 (2007) 248–258.

[18] Z. Liu, X. Zheng, C.H. Sung, Preconditioned multigrid methods for unsteady incompressible flows, J. Comput. Phys. 160 (2000) 151–178.

[19] F. Necker, C. Hartel, L. Kleiser, E. Meiburg, High-resolution simulations of particle-driven gravity currents, J. Multiphase Flow 28 (2002) 279–300.

[20] I. Orlanski, A simple boundary condition for unbounded hyperbolic flows, J. Comput. Phys. 21 (1976) 251–269.

[21] M. Ramšak, L. Škerget, A subdomain boundary element method for high-Reynolds laminar flow using stream function-vorticity formulation, Int. J. Numer. Methods Fluids 46 (2004) 815–847.

[22] M. Ramšak, L. Škerget, M. Hriberšek, Z. Žunič, A multidomain boundary element method for unsteady laminar flow using stream function vorticity equations, Engrg. Anal. Bound. Elem. 29 (2005) 1–14.

[23] J. Ravnik, L. Škerget, M. Hriberšek, The wavelet transform for BEM computational fluid dynamics, Engrg. Anal. Bound. Elem. 28 (2004) 1303–1314.

[24] J. Ravnik, L. Škerget, M. Hriberšek, 2D velocity vorticity based LES for the solution of natural convection in a differentially heated enclosure by wavelet transform based BEM and FEM, Engrg. Anal. Bound. Elem. 30 (2006) 671–686.

[25] N. Siriboonluckul, V. Juntasaro, Turbulence modelling for wall-bounded particle-laden flow with separation, Int. Commun. Heat Mass Transf. 34 (2007) 331–338.

[26] S.A. Slater, J.B. Young, The calculation of inertial particle transport in dilute gas-particle flows, Int. J. Multiphase Flow 27 (2001) 61–87.

[27] G.L.G. Sleijpen, D.R. Fokkema, BICGSTAB(l) for linear equations involving unsymmetric matrices with complex spectrum, El. Trans. Numer. Anal. 1 (1993) 11–32.

[28] L. Škerget, M. Hriberšek, G. Kuhn, Computational fluid dynamics by boundary domain integral method, Int. J. Num. Methods Engrg. 46 (1999) 1291–1311.

[29] L. Škerget, M. Hriberšek, Z. Žunič, Natural convection flows in complex cavities by BEM, Int. J. Num. Methods Heat Fluid Flow 13 (2003) 720–735.

[30] A. Tomiyama, I. Žun, H. Higaki, Y. Makino, T. Sakaguchi, A three-dimensional particle tracking method for bubbly flow simulation, Nucl. Engrg. Des. 175 (1997) 77–86.

[31] D.J. Tritton, Experiments on the flow past a circular cylinder at low Reynolds numbers, J. Fluid Mech. 6 (1959) 547–567.

Author's personal copy

[32] B. Wang, H. Zhang, X. Wang, Large eddy simulation of particle response to turbulence along its trajectory in a backward-facing step turbulent flow, Int. J. Heat Mass Transfer 49 (2006) 415–420.
[33] L.C. Wrobel, The Boundary Element Method, John Willey & Sons, Ltd., 2002.
[34] D.L. Young, Y.H. Liu, T.I. Eldho, A combined BEM–FEM model for velocity–vorticity formulation of the Navier–Stokes equations in three dimensions, Engrg. Anal. Bound. Elem. 24 (2000) 307–314.
[35] K. Yu, K. Lau, C. Chan, Numerical simulation of gas-particle flow in a single-side backward-facing step flow, J. Comput. Appl. Math. 163 (2004) 319–331.
[36] B. Zhou, Finite volume solution of the Navier–Stokes equations in velocity–vorticity formulation, Int. J. Numer. Methods Fluids 48 (2005) 607–629.
[37] Z. Žunič, M. Hriberšek, L. Škerget, J. Ravnik, 3-D boundary element-finite element method for velocity–vorticity formulation of the Navier–Stokes equations, Engrg. Anal. Bound. Elem. 31 (2007) 259–266.
[38] Z. Žunič, L. Škerget, M. Hriberšek, J. Ravnik, Boundary element-finite element method for incompressible viscous fluid flow, in: 2. Workshop on Advanced Computational Engineering Mechanics, 2005.
[39] Z. Žunič, L. Škerget, M. Hriberšek, J. Ravnik, Boundary element-finite element method for velocity–vorticity formulation of Navier–Stokes equations, in: WIT transactions on Modelling and Simulation, vol. 41, 2005.