

A multiblock joint PDF finite-volume hybrid algorithm for the computation of turbulent flows in complex geometries

B. Rembold^{*}, P. Jenny

Institute of Fluid Dynamics, Department of Mechanical Engineering, ETH Zurich, Sonneggstrasse 3, CH-8092 Zurich, Switzerland

Received 3 October 2005; received in revised form 28 April 2006; accepted 2 May 2006

Available online 22 June 2006

Abstract

A new hybrid joint probability density function (JPDF) solution algorithm for turbulent flows in complex 3D geometries is presented. The main focus is to demonstrate the applicability of JPDF methods for complex flows as observed in industrial applications. All elements of the algorithm are explained in detail and extensive validation studies are presented. A multiblock finite-volume solver, capable of handling globally unstructured, locally structured grids, was implemented together with a Lagrangian particle method. Efficient and robust particle management and accurate schemes for estimation and interpolation of particle statistics have been developed. For numerical efficiency particle sub-time stepping and an implicit finite-volume solver are applied. A fast coupling strategy was developed together with a multigrid method that allows very fast convergence on refined grids. Comparison with an established JPDF code for a bluff-body stabilized flow shows very good agreement. Furthermore, robustness and consistency of the algorithm for turbulent flow simulations with complex geometry is demonstrated.

© 2006 Elsevier Inc. All rights reserved.

Keywords: PDF methods; Particle methods; Monte Carlo; Hybrid algorithm; Turbulence modeling; Finite-volume; Complex geometries; Multiblock

1. Introduction

Accurate simulations of chemical reactions in complex turbulent environments are crucial for many studies of engineering applications. In addition to reliable turbulence models, major challenges include general descriptions of turbulence-reaction interaction and mixing. Compared with traditional turbulence models [1–3], which are based on Reynolds averaging techniques and yield modeled equations for statistical moments, joint velocity-composition probability density function (JPDF) methods have the crucial advantage that turbulence-reaction interaction and turbulent convection (which includes turbulent dispersion of scalars) appear in closed form. Molecular mixing, dissipation and redistribution of turbulent kinetic energy

^{*} Corresponding author.

E-mail addresses: rembold@ifd.mavt.ethz.ch (B. Rembold), jenny@ifd.mavt.ethz.ch (P. Jenny).

on the other hand require modeling, which is an active research area. An overview of theory and modeling of turbulent flow is given in [4] and a comprehensive review of modeling aspects with respect to reacting flows is presented in [5].

In terms of better models for probability density function methods (PDF) significant progress has been made during the past decades. For example, more advanced joint velocity-frequency-composition PDF methods, which also provide a model for the turbulent time scale [6,7], were adopted and sophisticated modeling techniques developed for second-moment closures [8–11] were incorporated. Relevant for turbulent combustion simulations are recent improvements for efficient computation of complex reaction chemistry [12] and better molecular mixing models. Due to these improvements, PDF methods could be successfully applied for many inert [13–16] and reactive flow studies [17–24]. To solve modeled PDF transport equations typically Monte Carlo methods, i.e. stand-alone particle-mesh methods, are employed. This is due to the high dimensional space in which the PDF evolves [25–27]. In such particle methods each particle represents a realization of the flow at its position and their evolution is described by stochastic differential equations. These equations are constructed such that the particle number density can be interpreted as the probability density in sample space, provided enough particles are employed. However, the large number of particles which is required for robust and accurate simulations makes stand-alone particle-mesh methods relatively expensive. Therefore, despite their intrinsic modeling advantages, PDF methods are not widely used in industry. Motivated by this deficiency a consistent particle-finite-volume hybrid algorithm to solve the JPFD transport equation for turbulent reactive flows was developed [28,29]. The algorithm is a combination of a finite-volume scheme and a particle method, where the finite-volume scheme is used to solve the Reynolds averaged Navier–Stokes (RANS) equations and the particle method to compute the JPFD transport equation. This allows to use a much smoother mean (Favre averaged) velocity field, which is computed by the finite-volume scheme, in the particle method. Vice versa, the turbulent fluxes and the mean reaction term are extracted from the particle field and can be used to close the RANS equations. It could be demonstrated that results computed with this hybrid method contain much less bias and statistical errors than solutions obtained with stand-alone particle-mesh methods (for the same number of particles). This allows to perform simulations with much less particles and results in a significant improvement of the computational efficiency [13]. For the application of JPFD methods for complex, large problems, e.g. in industry, this is very promising. However, that hybrid algorithm was developed for orthogonal 2D (plane or axi-symmetric) grids only and cannot deal with the geometrical complexities typical for practical problems.

In this paper, we present a new hybrid algorithm for complex 3D geometries based on locally structured, globally unstructured multiblock grids [30,31]. Compared with fully unstructured grids, such grids have numerous algorithmic advantages and can still honor very complex 3D geometries. The basic concept of our JPFD method is based on the ideas of the consistent hybrid method for orthogonal 2D grids, but to accommodate for the much more general grids it was necessary to address and resolve various additional algorithmic issues, most of them related to an efficient particle management in general hexahedron multiblock topologies. We also present a number of schemes, which we developed to further enhance the overall efficiency, e.g. particle sub-time stepping for multiblock topologies, a multigrid algorithm for solutions on very fine grids, and an algebraic proxy turbulence model for a good initial solution of the flow field. The implementation of the multiblock JPFD algorithm is based on our Lagrangian–Eulerian multiblock simulation kit (*LEMBSK*), which was designed as a general platform for the quick implementation of coupled particle/finite-volume methods. Compared to previous work on PDF methods for more complex geometries [32–35] the present algorithm has the advantage that it consistently solves for the joint fluctuating-velocity-frequency-composition PDF and is explicitly designed for arbitrary non-orthogonal multiblock grids. The new JPFD solver was first validated for a bluff-body test case with simple geometry. It is shown that the results are in excellent agreement with those of the established 2D hybrid method. The capability to deal with complex geometries was demonstrated with simulations of turbulent flow in a generic combustion chamber. It is shown that the level of internal consistency which can be achieved numerically is good.

The layout of the paper is as follows. First, we describe the governing equations in Section 2. Then, all details of the solution algorithm are described in Section 3 and in Section 4, a careful validation is presented. Finally, conclusions are given in Section 5.

2. Hybrid JPDF method

In this section, we describe the governing equations for the JPDF method. In the framework of the hybrid algorithm described here, a transport equation for the joint PDF of fluctuating velocity and composition is solved (Section 2.1). Since this JPDF contains no information about the mean velocity, the Reynolds-averaged Navier–Stokes (RANS) equations are solved simultaneously (Section 2.2).

2.1. JPDF transport equation

The dependent variable in the JPDF transport equation is the mass density function (MDF) $\mathcal{G} = \langle \rho \rangle(\mathbf{x}, t) \tilde{g}(\mathbf{v}, \mathbf{C}; \mathbf{x}, t)$, which is the product of the mean fluid density, $\langle \rho \rangle$, and the mass weighted JPDF, \tilde{g} , of the fluctuating velocity and composition at each location, \mathbf{x} , and time, t . The vectors \mathbf{v} and \mathbf{C} are the sample space variables of the fluctuating velocity, \mathbf{u} , and composition, \mathbf{c} , respectively. The transport equation for \mathcal{G} reads

$$\frac{\partial \mathcal{G}}{\partial t} + \frac{\partial \mathcal{G}(\tilde{U}_j + v_j)}{\partial x_j} + \frac{\partial}{\partial v_i} \left(\mathcal{G} \left\langle \frac{du_i}{dt} \middle| \mathbf{v}, \mathbf{C} \right\rangle \right) + \frac{\partial}{\partial C_\alpha} \left(\mathcal{G} \left\langle \frac{dc_\alpha}{dt} \middle| \mathbf{v}, \mathbf{C} \right\rangle \right) = 0, \quad (1)$$

where $\tilde{\mathbf{U}} = \mathbf{U} - \mathbf{u}$ is the Favre-averaged fluid velocity. Favre-averaged quantities are denoted by $\tilde{\cdot}$, Reynolds-averaged quantities by $\langle \cdot \rangle$ and $\langle \cdot | \cdot \rangle$ are conditional expectations. The second, third and fourth terms on the left-hand side denote transport in physical, velocity and composition space, respectively. The substantial derivatives

$$\frac{du_i}{dt} = \frac{dU_i}{dt} - \frac{d\tilde{U}_i}{dt} = -u_j \frac{\partial \tilde{U}_i}{\partial x_j} + \frac{1}{\langle \rho \rangle} \frac{\partial \langle \rho \rangle \widetilde{u_i u_j}}{\partial x_j} - \frac{1}{\rho} \frac{\partial p}{\partial x_i} + \frac{1}{\langle \rho \rangle} \frac{\partial \langle p \rangle}{\partial x_i} + \frac{1}{\rho} \frac{\partial \tau_{ij}}{\partial x_j} - \frac{1}{\langle \rho \rangle} \frac{\partial \langle \tau_{ij} \rangle}{\partial x_j} \quad (2)$$

and

$$\frac{dc_\alpha}{dt} = -\frac{1}{\langle \rho \rangle} \frac{\partial J_\alpha^z}{\partial x_i} + S_\alpha \quad (3)$$

are derived from the Navier–Stokes equations, where S_α denotes the chemical source term and \mathbf{J}^α is the diffusive flux of composition α . With Eqs. (1)–(3), the transport equation for \mathcal{G} becomes

$$\begin{aligned} & \frac{\partial \mathcal{G}}{\partial t} + \frac{\partial \mathcal{G}(\tilde{U}_j + v_j)}{\partial x_j} - \frac{\partial \tilde{U}_i}{\partial x_j} \frac{\partial \mathcal{G} v_j}{\partial v_i} + \frac{1}{\langle \rho \rangle} \frac{\partial \langle \rho \rangle \widetilde{u_i u_j}}{\partial x_j} \frac{\partial \mathcal{G}}{\partial v_i} + \frac{\partial \mathcal{G} S_\alpha}{\partial C_\alpha} \\ & = \frac{\partial}{\partial v_i} \left(\mathcal{G} \left\langle \frac{1}{\rho} \frac{\partial p}{\partial x_i} - \frac{1}{\langle \rho \rangle} \frac{\partial \langle p \rangle}{\partial x_i} - \frac{1}{\rho} \frac{\partial \tau_{ij}}{\partial x_j} + \frac{1}{\langle \rho \rangle} \frac{\partial \langle \tau_{ij} \rangle}{\partial x_j} \middle| \mathbf{v}, \mathbf{C} \right\rangle \right) + \frac{\partial}{\partial C_\alpha} \left(\mathcal{G} \left\langle \frac{1}{\langle \rho \rangle} \frac{\partial J_\alpha^z}{\partial x_i} \middle| \mathbf{v}, \mathbf{C} \right\rangle \right). \end{aligned} \quad (4)$$

Note that the terms on the left-hand side appear in closed form. A particular advantage of PDF methods is that the chemical source term and turbulent convection do not have to be modeled. However, models are required to close the terms on the right-hand side, i.e. to account for the effects of the fluctuating viscous stress tensor, $1/\rho \partial \tau_{ij} / \partial x_j - 1/\langle \rho \rangle \partial \langle \tau_{ij} \rangle / \partial x_j$, fluctuating pressure gradient, $1/\rho \partial p / \partial x_j - 1/\langle \rho \rangle \partial \langle p \rangle / \partial x_j$, and diffusive fluxes, \mathbf{J}^α . For simplicity, but without loss of generality, the algorithmic issues addressed in this paper are discussed without considering reactions or compositions. The simplified Langevin model (SLM) [4] is employed to close Eq. (4), which results in the modeled transport equation

$$\frac{\partial \mathcal{G}}{\partial t} + \frac{\partial \mathcal{G}(\tilde{U}_j + v_j)}{\partial x_j} - \frac{\partial \tilde{U}_i}{\partial x_j} \frac{\partial \mathcal{G} v_j}{\partial v_i} + \frac{1}{\langle \rho \rangle} \frac{\partial \langle \rho \rangle \widetilde{u_i u_j}}{\partial x_j} \frac{\partial \mathcal{G}}{\partial v_i} = -\frac{\partial}{\partial v_i} \left(\mathcal{G} \left(\frac{1}{2} + \frac{3}{4} C_0 \right) \Omega v_i + \frac{1}{2} C_0 k \Omega \frac{\partial \mathcal{G}}{\partial v_i} \right), \quad (5)$$

where Ω is the conditional turbulence frequency, $k = \widetilde{u_i u_i} / 2$ the turbulent kinetic energy, and C_0 and C_Ω are model constants given in Table 1. Eq. (5) is a Fokker–Planck equation and can be solved using a Lagrangian particle method (PM) (see Section 3.2.1).

Table 1
Constants used in the JPDF models

Constant	Suggested value	Used in
C_0	2.1	SLM
C_Ω	0.6893	Definition of omega
$C_{\omega 1}$	0.65	Turbulence frequency model
$C_{\omega 2}$	0.9	Turbulence frequency model
C_3	1.0	Turbulence frequency model
C_4	0.25	Turbulence frequency model

2.2. RANS equations

To close Eq. (4), the Favre-averaged velocity, \tilde{U}_i , has to be known. Therefore, in the hybrid approach under consideration, the RANS equations, i.e.

$$\begin{aligned} \frac{\partial}{\partial t} \langle \rho \rangle + \frac{\partial}{\partial x_i} (\langle \rho \rangle \tilde{U}_i) &= 0, \\ \frac{\partial}{\partial t} (\langle \rho \rangle \tilde{U}_i) + \frac{\partial}{\partial x_j} (\langle \rho \rangle \tilde{U}_i \tilde{U}_j + \langle p \rangle \delta_{ij}) &= -\frac{\partial}{\partial x_j} (\langle \rho \rangle \widetilde{u_i u_j}), \\ \frac{\partial}{\partial t} (\langle \rho \rangle \tilde{E}_s) + \frac{\partial}{\partial x_i} (\tilde{U}_i (\langle \rho \rangle \tilde{E}_s + \langle p \rangle)) &= \langle \rho \dot{Q} \rangle - \frac{\partial}{\partial x_i} (\langle \rho \rangle \widetilde{u_i h''_s}) - \frac{\partial}{\partial x_i} \left(\frac{\langle \rho \rangle}{2} \widetilde{u_i u_j u_j} \right) - \frac{\partial}{\partial x_i} (\tilde{U}_j \langle \rho \rangle \widetilde{u_i u_j}), \end{aligned} \quad (6)$$

are solved, where \tilde{E}_s is the Favre-averaged total energy per unit volume. Note that the molecular fluxes are neglected, which is justified for high Reynolds numbers. The terms on the right-hand side of Eq. (6) can be computed from the JPDF \tilde{g} and, therefore, Eqs. (5) and (6) combined provide a closed system, which is the basis for the hybrid algorithm. There, the RANS equations (6) are solved with a finite-volume method (FVM), while a particle method is used to solve Eq. (5).

3. Solution algorithm

The basis of the new 3D multiblock algorithm is a coupled Lagrangian/Eulerian simulation platform. It consists of a grid based finite-volume method and a particle method. Whereas the FVM is more or less standard, most of the algorithmic details are centered around the PM. A flow chart of the algorithm is depicted in Fig. 1. The coupling between the FVM and the PM is achieved via particle field estimation and interpolation of mean flow quantities. Before explaining the details of the PM in Section 3.2, which also includes the coupling issues, we describe the FVM.

3.1. Finite-volume method

The RANS equations (6) are solved on a multiblock grid using a FVM. The global block topology can be unstructured, but each block consists of a structured grid. Coupling between adjacent blocks is achieved by introducing ghost cells coinciding with the corresponding neighboring cells. For illustration, a single structured block is shown in Fig. 2. Note that it consists of arbitrarily shaped hexahedra (details of the geometry description are explained in Section 3.2.2). The numerical fluxes, \mathbf{F} , at a volume interface are computed with a characteristic-based approximate Riemann solver [36]. Spatial discretization is second order accurate using a MUSCL reconstruction [37] and the minmod limiter. In order to overcome the CFL restrictions of explicit methods, especially for low Mach number flows, a fully implicit scheme is used for time integration. Each time step, a nonlinear problem has to be solved, which is achieved by employing the Newton–Raphson method within each block. After each Newton–Raphson iteration the information in the ghost cells is updated. Thereby, the vector of conservative variables, \mathcal{U} , in volume (i, j, k) at the new time level $n + 1$ is computed as

$$\mathcal{U}_{i,j,k}^{n+1} = \mathcal{U}_{i,j,k}^n - \frac{\Delta t}{V_{i,j,k}^n} \{ \mathbf{F}_{i+1/2,j,k}^{n+1} - \mathbf{F}_{i-1/2,j,k}^{n+1} + \mathbf{F}_{i,j+1/2,k}^{n+1} - \mathbf{F}_{i,j-1/2,k}^{n+1} + \mathbf{F}_{i,j,k+1/2}^{n+1} - \mathbf{F}_{i,j,k-1/2}^{n+1} \}, \quad (7)$$

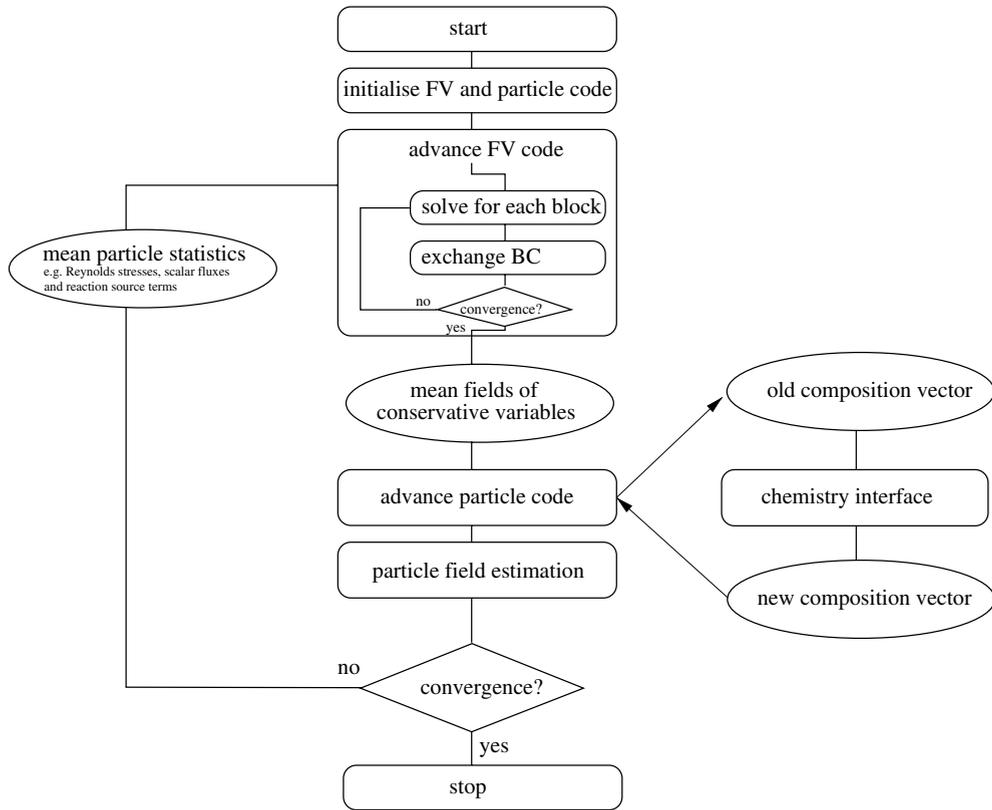


Fig. 1. Hybrid algorithm: flow chart.

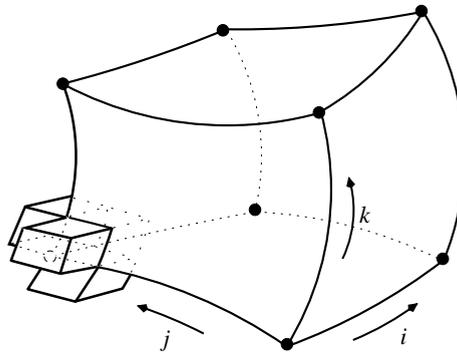


Fig. 2. Multiblock paradigm: structured block with exemplary ghost cells.

where the numerical flux, \mathbf{F}^{n+1} , is approximated by the linearization

$$\mathbf{F}^{n+1} \approx \mathbf{F}^n + \frac{\partial \mathbf{F}^n}{\partial \mathcal{U}} (\mathcal{U}^{n+1} - \mathcal{U}^n). \quad (8)$$

The superscripts n and $n + 1$ denote the old and new time levels, respectively. Note that the numerical flux \mathbf{F} also contains the area of the corresponding face of the hexahedron cell (dimension of \mathbf{F} is dimension of \mathcal{U} times m^3/s). $V_{i,j,k}^c$ denotes the volume of the cell, as defined later by Eq. (15). The Jacobian, $\partial \mathbf{F}^n / \partial \mathcal{U}$, is computed analytically (1st order) or numerically (1st or 2nd order). For the numerical computation of the Jacobian, the approximation

$$\frac{\partial \mathbf{F}^n}{\partial \mathcal{U}_k} \approx \frac{\mathbf{F}(\mathcal{U} + \Delta \mathcal{U}) - \mathbf{F}(\mathcal{U})}{\Delta \mathcal{U}_k} \quad (9)$$

is used, where $\Delta \mathcal{U}$ is a small perturbation, where $\Delta \mathcal{U}_{j|j \neq k} = 0$. Due to the ghost cells, block interfaces or boundaries require no special treatment, except that the values in the ghost cells need to be updated after each iteration (Schwarz overlap technique). However, to achieve consistency, one has to ensure that each ghost cell is geometrically identical with its corresponding cell in the neighbor block. The implicit solver was validated and compared with an explicit Runge–Kutta scheme. It was confirmed that the implicit solver allows for much larger time steps (several orders of magnitude), especially for test cases with small Mach numbers or highly heterogeneous grids. According to our experience it makes no difference whether the Jacobian is computed analytically or numerically (Eq. (9)). The numerical version, however, is much easier to implement and more flexible, e.g. it is straightforward to incorporate viscous fluxes or to extend the scheme to higher order. Moreover, it was found that approximately the same convergence rate is achieved, if the higher order extension is treated explicitly. This results in linear systems with smaller bandwidth, which can be solved more efficiently. For small problems, we use an efficient direct LU decomposition solver, which makes use of the band-structure of the matrix. For larger problems, a preconditioned iterative solver is applied. The terms on the right-hand side of Eq. (6) are extracted from the particle fields; more will be explained in Section 3.2. The fluxes have to be interpolated from the cell nodes to the face-midpoints. Here, the average value of the four nodes adjacent to a cell interface is taken to approximate the correct flux across an interface. To impose boundary conditions, the same ghost cell technique as for the block coupling is employed. At inflow boundaries, \mathcal{U} is enforced for all components but the total energy per unit volume, $\rho \tilde{E}_s$, which is extrapolated from the computational domain. At outflow boundaries, all components of \mathcal{U} except $\rho \tilde{E}_s$, which is enforced, are extrapolated from the computational domain to the ghost cells. Slip boundary conditions are imposed by reflecting the velocity vector at the boundary face. It is important to note that the ghost cell geometry is an exact mirror of its adjacent cell inside the computational domain, otherwise the flux solver does not guarantee that the mass flux across the boundary is zero. Note that it is also possible to deal with more complex boundary conditions using the same ghost cell approach.

3.2. Lagrangian particle method

The PM and its coupling with the FVM are the most delicate aspects of the algorithm. In this section, we first describe the modeling issues on the particle side. Then, the algorithmic details of the PM are addressed, which includes particle management in complex 3D multiblock grids, extraction of mean quantities from the particle field, and interpolation of data to particle locations. Moreover, critical aspects with respect to computational efficiency are discussed.

3.2.1. Modeled PDF transport equation

Eq. (4) is solved using a Lagrangian particle method. An ensemble of notional particles represents the mass density function, \mathcal{G} . Each particle carries information about its position, \mathbf{x}^* , weight, m^* , Favre-fluctuating velocity, \mathbf{u}^* , turbulence frequency, ω^* , and composition, \mathbf{c}^* . The superscript $*$ denotes a notional particle property. In physical space, the particle evolution is described by

$$dx_i^*(t) = (\tilde{U}_i + u_i^*) dt \quad (10)$$

and in fluctuating velocity space, particle evolution is governed by Eq. (2). The first two terms on the right-hand side of Eq. (2) are in closed form, but the effect of fluctuating pressure and fluctuating viscous stresses needs to be modeled. Therefore, as mentioned earlier, we here employ the SLM [4], which leads to the stochastic differential equation

$$du_i^*(t) = \frac{1}{\langle \rho \rangle} \frac{\partial \langle \rho \rangle \widetilde{u_i u_j}}{\partial x_j} dt - u_j^* \frac{\partial \tilde{U}_i}{\partial x_j} dt - \left(\frac{1}{2} + \frac{3}{4} C_0 \right) \Omega u_i^*(t) dt + (C_0 k \Omega)^{\frac{1}{2}} dW_i(t) \quad (11)$$

for the fluctuating particle velocity. Note that $dW_i(t)$ is a Wiener process, where $dW_i(t) = W_i(t+dt) - W_i(t)$ is normally distributed with $\langle dW_i(t) \rangle = 0$ and $\langle dW_i(t) dW_j(t) \rangle = dt \delta_{ij}$.

To compute the conditional turbulence frequency,

$$\Omega = C_\Omega \frac{\langle \rho^* \omega^* | \omega^* \geq \tilde{\omega} \rangle}{\langle \rho \rangle}, \quad (12)$$

a further stochastic differential equation, which is explained in Ref. [29], is solved for ω^* .

If reacting flow is considered, which is not discussed in this paper, also molecular mixing and chemical source terms have to be computed. Those two terms are responsible for the particle evolution in composition space.

3.2.2. Particle management, interpolation, and estimation

For a fast solution algorithm, efficient particle management within the 3D multiblock framework is crucial. First, the particles have to be assigned to a cell in the multiblock grid. Second, mean quantities, e.g. Reynolds stresses, have to be extracted from the ensemble of particles and stored at the grid nodes. Such mean quantities are used for three purposes: to close the particle evolution equations, to close the RANS equations solved by the FVM, and to present results. In order to access mean quantities (including gradients and fields computed by the FVM) in the particle evolution equations, the data have to be interpolated from the grid nodes to the particle positions.

Next, we describe how the mean particle data are extracted at the grid nodes and how data are interpolated from the grid nodes to the particle locations.

3.2.2.1. Geometry definition. Each hexahedron grid cell, \mathcal{C} , is defined by eight nodes, \mathbf{x}_i ($i \in \{0, \dots, 7\}$), which determine the cell's midpoint (Fig. 3a)

$$\mathbf{x}^m = \frac{1}{8} \sum_{i=0}^7 \mathbf{x}_i \quad (13)$$

and the midpoint of a cell face Υ_i (Fig. 3b)

$$\mathbf{x}^{\Upsilon_i} = \frac{1}{4} \sum_{j|\mathbf{x}_j \in \Upsilon_i} \mathbf{x}_j = \frac{1}{4} \sum_{j=0}^3 \mathbf{x}_{i,j}, \quad (14)$$

where $\mathbf{x}_{i,j}$ are the nodes of face Υ_i . Each hexahedron cell \mathcal{C} is subdivided into six pyramids, one for each face, with the common node \mathbf{x}^m . Finally, each pyramid is divided into four tetrahedra, $\Gamma_{i,j}$ ($i \in \{0, \dots, 5\}$ and $j \in \{0, \dots, 3\}$), with the common nodes \mathbf{x}^m and \mathbf{x}^{Υ_i} , where i is the index of the pyramid and j the index of the tetrahedron. The cell volume is defined by

$$V^\mathcal{C} = \sum_{i=0}^5 \sum_{j=0}^3 V^{\Gamma_{i,j}}, \quad (15)$$

where the volume of the tetrahedron $\Gamma_{i,j}$ is

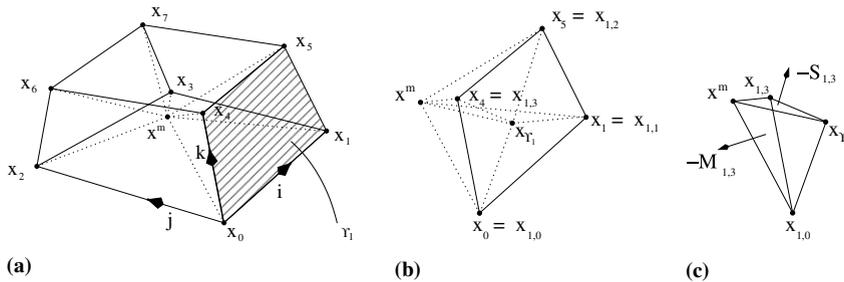


Fig. 3. Geometrical definitions: (a) notation of nodes in \mathcal{C} , (b) pyramid at face Υ_1 consisting of tetrahedra $\Gamma_{1,j}, j \in \{0, \dots, 3\}$, and (c) isolated tetrahedron $\Gamma_{1,3}$.

$$V^{\Gamma_{i,j}} = \frac{1}{6} ((\mathbf{x}^{T_i} - \mathbf{x}_{i,j}) \times (\mathbf{x}^{T_i} - \mathbf{x}_{i,\text{mod}_4(j+1)})) \cdot (\mathbf{x}^m - \mathbf{x}^{T_i}). \quad (16)$$

Note that $\text{mod}_4(\cdot)$ is the modulo function. The face normal vectors \mathbf{N}_i of Υ_i are computed as

$$\mathbf{N}_i = \frac{\mathbf{n}_i}{2|\mathbf{n}_i|^2} \sum_{j=0}^3 |(\mathbf{x}^{T_i} - \mathbf{x}_{i,j}) \times (\mathbf{x}_{i,j} - \mathbf{x}_{i,\text{mod}_4(j+1)}) \cdot \mathbf{n}_i| \quad (17)$$

with

$$\mathbf{n}_i = (\mathbf{x}_{i,0} - \mathbf{x}_{i,2}) \times (\mathbf{x}_{i,1} - \mathbf{x}_{i,3}). \quad (18)$$

Note that the absolute value of the normal vector \mathbf{n}_i corresponds to the area of face Υ_i .

3.2.2.2. Definition of basis functions. Within each tetrahedron, $\Gamma_{i,j}$, we introduce a piecewise tri-linear basis functions, $w_l(\xi)$, for each node $\mathbf{x}_l \in \mathcal{C}$. These basis functions fulfill

$$w_l(\xi = \mathbf{x}_k) = \delta_{lk}, \quad l, k = 0, \dots, 7 \quad (19)$$

and

$$w_l(\xi = \mathbf{x}^m) = 0.125, \quad l = 0, \dots, 7. \quad (20)$$

Here, ξ denotes the position within the cell. The basis functions have compact support, i.e. $w_l(\xi) = 0 \quad \forall \xi \notin \mathcal{C}$ and

$$\sum_{l=0}^7 w_l(\xi) = 1 \quad \forall \xi \in \mathcal{C}. \quad (21)$$

The gradient $\nabla w_l(\xi)$ is constant within $\Gamma_{i,j}$ and satisfies

$$(\mathbf{x}_{i,\text{mod}_4(j+k)} - \mathbf{x}^m) \cdot \nabla w_l = w_l(\xi = \mathbf{x}_{i,\text{mod}_4(j+k)}) - 0.125, \quad k = 0, 1 \quad (22)$$

and

$$(\mathbf{x}^{T_i} - \mathbf{x}^m) \cdot \nabla w_l = 0.125. \quad (23)$$

The latter equation implies that the basis functions have the value 0.25 at the face midpoints adjacent to the node where their value is one.

3.2.2.3. Extraction from particles to nodes. The weighted particle contribution of the quantity Ψ^* at node \mathbf{x}_l is

$$\hat{\Psi}_l = \sum_{\forall \text{ particles}} m^* w_l(\mathbf{x}^*) \Psi^*, \quad (24)$$

where

$$w_l(\mathbf{x}^*) = 0.125 + \nabla w_l(\mathbf{x}^*) \cdot (\mathbf{x}^* - \mathbf{x}^m) \quad (25)$$

is the corresponding weight.

3.2.2.4. Interpolation from nodes to particle locations. To interpolate a quantity Φ from the cell corners to the particle position, \mathbf{x}^* , the same weights as for the extraction are used. The interpolation then reads

$$\Phi(\mathbf{x}^*) = \sum_{l=0}^7 \Phi_l w_l(\mathbf{x}^*), \quad (26)$$

where Φ_l denotes the value of Φ at the node \mathbf{x}_l . Similarly, the gradient $\nabla \Phi$ can be computed as

$$\nabla \Phi(\mathbf{x}^*) = \sum_{l=0}^7 \Phi_l \nabla w_l(\mathbf{x}^*). \quad (27)$$

Note that the geometrical data, \mathbf{x}^m , \mathbf{x}^\top , \mathbf{N} and ∇w_l , have to be computed only once, but for each particle it has to be determined in which tetrahedron it is located. This procedure is described in [Appendix B](#).

3.2.3. Extraction of particle statistics

An estimate of Ψ at node \mathbf{x}_l is obtained by

$$\tilde{\Psi}_l = \frac{\hat{\Psi}_l}{M_l}, \tag{28}$$

where

$$M_l = \sum_{\forall \text{ particles}} m^* w_l(\mathbf{x}^*). \tag{29}$$

In addition, some particle data are represented as an average over a grid cell $\mathcal{C}_{i,j,k}$, i.e. by

$$\tilde{\Psi}_{i,j,k} = \frac{\hat{\Psi}_{i,j,k}}{M_{i,j,k}} \tag{30}$$

with

$$\hat{\Psi}_{i,j,k} = \sum_{\forall \text{ particles} \in \mathcal{C}_{i,j,k}} m^* \Psi^* \tag{31}$$

and

$$M_{i,j,k} = \sum_{\forall \text{ particles} \in \mathcal{C}_{i,j,k}} m^*. \tag{32}$$

[Table 2](#) shows the quantities which have to be extracted and indicates where they are used. For the extraction of statistical particle data at the nodes of a multiblock grid it is important to maintain consistency across block boundaries. Therefore, for nodes at block interfaces, particles from the neighboring blocks have to be taken into account. In the current implementation this is considered for extraction at block interfaces, vertices and edges and it is ensured that no discontinuities in the extracted fields at block-interface nodes can occur (note that extracted quantities at such nodes are stored separately for each block). This is important for stability reasons.

At symmetry boundaries, one has to be careful with the boundary conditions for the Reynolds stresses, e.g. in the case of a normal vector pointing in the direction of x_1 , $\widetilde{u_1 u_2}$ has to be exactly zero at a symmetry plane. This cannot be obtained, if particles are extracted only from one side of the symmetry plane. Therefore, ghost particles are introduced for each computational particle contributing to the extraction at this boundary ([Fig. 4](#)). Note that the ghost particles are not added to the particle cloud; they are only used during the extraction procedure. The velocity of the ghost particle is mirrored at the block boundary. All other ghost particle quantities, such as mass, m^* , composition vector, \mathbf{c}^* , and turbulence frequency, ω^* , are identical to those of the corresponding computational particle.

Table 2
Extracted statistics from the particles

Quantity	Description	Location	Used by
$\widetilde{u_i u_j}$	Reynolds stress	Grid node	FVM, PM
$\widetilde{u_i h''_s}$	Turbulence enthalpy flux	Grid node	FVM
$\widetilde{\omega}$	Turbulence frequency	Grid node	PM
Ω	Conditional turbulence frequency	Grid node	PM
$\widetilde{u_i u_k u_k}$	Triple correlation	Grid node	FVM
$\langle \rho \dot{Q} \rangle$	Source term	Cell center	FVM
ρ_p	Particle number density	Cell center	Position correction
$\widetilde{u_i}$	Mean fluctuating velocity	Grid node	Velocity correction
$\langle \mathbf{c} \rangle$	Composition vector	Grid node	PM

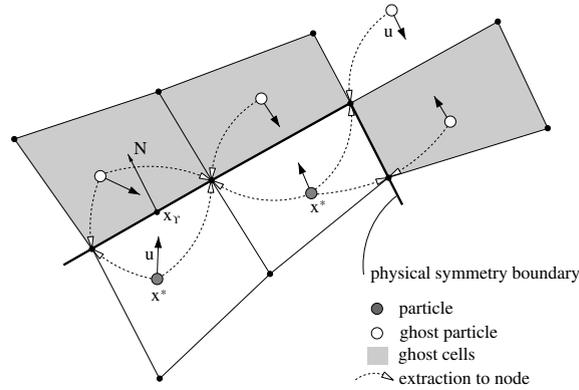


Fig. 4. Particle extraction at symmetry boundaries.

However, if there exist multiple symmetry planes adjacent to a node (at block edges or corners), it is not possible to obtain the desired symmetry by introducing a separate ghost particle for each symmetry plane. In such cases, only one ghost particle with a velocity reflected at the edge or corner is introduced (Fig. 4).

3.2.4. Particle time integration: sub-time stepping

Particle transport in physical space is governed by Eq. (10), where a CFL criterion has to be fulfilled for accuracy reason. Without special treatment, however, fast particles in small cells can impose a severe time step size limitation for all particles, which clearly has a negative impact on the efficiency of the overall solution algorithm. This is a concern in particular for grids with local refinement. To overcome this problem, a sub-time stepping algorithm similar to previously used schemes [35] was devised and implemented for multiblock grids. Eq. (10) is integrated for each particle using sub-time steps fulfilling the local CFL criterion. First, the global time step Δt_{PM} is defined and then each particle is evolved by a consistent number of sub-time steps

$$\Delta t_{PM}^s = CFL \min_i \left(\frac{\Delta x_i}{\tilde{U}_i(\mathbf{x}^*) + u_i^*} \right). \quad (33)$$

The maximum extension of the cell in which the particle is located is denoted by $\Delta \mathbf{x}$. Typically, we use $CFL \approx 0.5$. Each sub-time step, the mean quantities needed in the stochastic differential equations are interpolated to the particle positions (Eqs. (26) and (27)). The extraction of particle data is performed after each global time step. Note that sub-time stepping requires special care at the inflow boundary (see Section 3.2.5).

To achieve second-order accuracy for the time integration, the midpoint rule is applied [38]. The first half-step reads

$$\mathbf{x}^{*n+\frac{1}{2}} = \mathbf{x}^{*n} + \frac{\Delta t_{PM}^s}{2} (\tilde{\mathbf{U}}^n(\mathbf{x}^{*n}) + \mathbf{u}^{*n}). \quad (34)$$

Then, \mathbf{u}^{*n+1} is computed at the estimated mid-point, $\mathbf{x}^{*n+\frac{1}{2}}$, and is used to compute the new particle position

$$\mathbf{x}^{*n+1} = \mathbf{x}^{*n} + \Delta t_{PM}^s \left(\tilde{\mathbf{U}}^n \left(\mathbf{x}^{*n+\frac{1}{2}} \right) + \frac{1}{2} (\mathbf{u}^{*n} + \mathbf{u}^{*n+1}) \right), \quad (35)$$

where the superscripts n and $n+1$ denote the old and new time levels, respectively. Interpolation of the mean velocity, $\tilde{\mathbf{U}}$, is discussed in Appendix A.

3.2.5. Boundary conditions

Currently, four types of boundary conditions are considered for the particles, i.e. slip wall, symmetry boundary, inflow and outflow. If a particle crosses a symmetry or a slip wall boundary, it is reflected. The only difference between symmetry and slip wall boundary condition is that for the slip wall no ghost particles for extraction are created (see Section 3.2.3). In Fig. 5, reflection is illustrated in a sketch. It is guaranteed that the numerical particle flux across the boundary is exactly zero. The reflection of a particle implies mirroring its

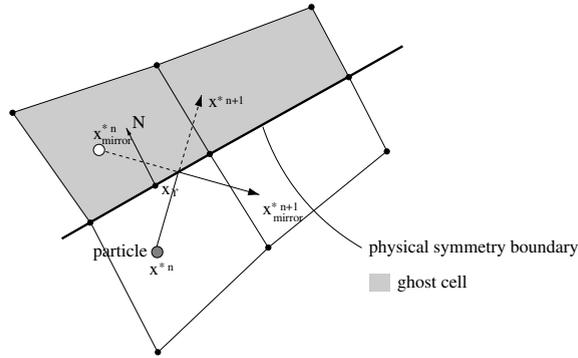


Fig. 5. Particle boundary condition: reflection at symmetry boundaries.

new position, \mathbf{x}^{*n+1} , velocity vector, \mathbf{u}^* , and origin, \mathbf{x}^{*n} , at the interface Υ after a sub-time step. Note that near edges a particle may be reflected several times. At the inflow boundary, the first plane of ghost cells is populated with particles, which have appropriate properties. If particle sub-time stepping is applied (see Section 3.2.4), it has to be guaranteed that a new particle is generated for each particle moving from a ghost cell into the computational domain. The position of the new particle is chosen randomly within the ghost cell. If a particle crosses an outflow boundary, it is removed. At the end of each time step, all ghost cell particles are collected and the inflow ghost cells are repopulated.

3.2.6. Particle number control

In order to keep the statistical and the deterministic bias errors small, a sufficiently large number of particles is required in each volume. On the other hand, however, the total number of particles used for a computation is tightly related to the computational cost. Therefore, for efficiency reason, it is important to control the particle number density. While increasing the number of particles is simply achieved by splitting heavy particles into lighter ones (this can be done without violating the conservation laws), reducing the particle number in a volume is not straight forward. It is not possible to fulfill all, mass, momentum and kinetic energy budget, in single events. In the present algorithm, first the two lightest particles in a cell are selected. Then, by random choice, one is eliminated. The weight of the eliminated particle is added to the other one to conserve mass. Even though conservation of kinetic energy and momentum may be violated in single events, this elimination procedure satisfies the conservation laws statistically.

It can be demonstrated that the expectation of any function $Q(\Phi^*)$, where Φ^* is the particle property vector, is preserved by this elimination procedure by considering the conditional expectation of a pair of particles

$$Q(\Phi)_{w_1, w_2} = \frac{\langle Q(\Phi_1^*)w_1^* + Q(\Phi_2^*)w_2^* \rangle}{\langle w_1^* + w_2^* \rangle} = \frac{w_1^*}{w_1^* + w_2^*} \langle Q(\Phi) |_{w_1^*} \rangle + \frac{w_2^*}{w_1^* + w_2^*} \langle Q(\Phi) |_{w_2^*} \rangle, \quad (36)$$

where w_1^* and w_2^* are the weights of the two particles. Note that the expression (36) is based on fixed weights w_1^* and w_2^* . The probability that particle one (two) is eliminated is p ($q = 1 - p$). The expectation after the elimination is

$$E = p \frac{w_2^* + w_1^*}{w_1^* + w_2^*} \langle Q(\Phi) |_{w_2^*} \rangle + q \frac{w_1^* + w_2^*}{w_1^* + w_2^*} \langle Q(\Phi) |_{w_1^*} \rangle. \quad (37)$$

If $p = w_2^*/(w_1^* + w_2^*)$ and consistently $q = w_1^*/(w_1^* + w_2^*)$, then E is identical to the conditional expectation given by Eq. (36). This proves that in the mean the elimination procedure preserves any (since Q is general) statistics derived from the particle properties. Moreover, note that the total mass is conserved exactly. The motivation for taking the lightest particles is simply to account for the stronger statistical representation by the heavy particles.

Note that with controlled splitting and elimination as described above the efficiency of the overall particle algorithms can be improved dramatically.

3.3. Coupling and solution strategy

One of the major challenges in the development of the current solution algorithm was a robust and efficient coupling strategy. Information has to be exchanged between the two components of the code, i.e. the mean velocity from the finite-volume scheme to the particle method and statistical moments like the turbulent fluxes and the energy source term vice versa. Essentially there are two coupling aspects. First, the convergence rate is significantly influenced by the coupling method, and second, inappropriate coupling can lead to instabilities. Next, the most important coupling issues are explained and discussed.

3.3.1. Moving time averaging

In order to reduce the statistical and the deterministic bias errors [39] of the extracted data like Reynolds stresses, we employ exponentially weighted moving time averaging. Note that this is only admissible, if statistically stationary solutions are considered. For example, let $\hat{\Phi}_l^{n+1}$ be some extracted data at node \mathbf{x}_l and time level $n + 1$. Employing exponentially weighted moving time averaging leads to the quantity

$$\bar{\Phi}^{n+1} = v_\phi \bar{\Phi}^n + (1 - v_\phi) \hat{\Phi}^{n+1}, \quad (38)$$

which can be used to replace $\hat{\Phi}_l^{n+1}$ in the equations. The main advantage is that $\bar{\Phi}^{n+1}$ is much smoother than $\hat{\Phi}_l^{n+1}$. As a result, statistical and bias errors are reduced and the coupling between particle and finite-volume methods becomes more robust. Note that the memory factor, v_ϕ , has to be between zero and one and determines the level of smoothing.

3.3.2. Algebraic proxy model

In order to take full advantage of the implicit finite-volume scheme and the particle sub-time stepping strategy, it is important to start with a reasonable initial condition. This can be achieved by initially employing a simple algebraic turbulence model with an ad hoc and case dependent value for the turbulent viscosity. During this phase, the particles evolve as usual, but their influence on the finite-volume solver is ignored or damped. For example, the Reynolds stresses in Eq. (6) are replaced by

$$\langle \rho \rangle \widetilde{\widetilde{u_i u_j}} = -v_{\text{RS}} \mu_t \left[\left(\frac{\partial \tilde{U}_i}{\partial x_j} + \frac{\partial \tilde{U}_j}{\partial x_i} \right) - \frac{2}{3} (\nabla \cdot \tilde{\mathbf{U}}) \delta_{ij} \right] + (1 - v_{\text{RS}}) \langle \rho \rangle \widetilde{u_i u_j}, \quad (39)$$

where μ_t is the turbulent viscosity and $v_{\text{RS}} \in [0, 1]$ is a blending factor. The computation is started with $v_{\text{RS}} = 1$ using large time steps. Like that, full advantage of the implicit solver is taken. Once an approximately stationary solution is obtained, v_{RS} is slowly relaxed to zero, e.g. as

$$v_{\text{RS}}^n = v_{\text{RS}}^0 \alpha_v^n, \quad (40)$$

where n is the time level and $\alpha_v = 0.99$. Finally, after a sufficient number of time steps, all unclosed terms in the RANS equations come from the particle statistics. According to our experience, the use of such an algebraic turbulence model during the initial phase of a PDF simulation can make a big difference, in particular for geometrically complex simulations. Note that this idea can be improved, e.g. by replacing the algebraic model by a Reynolds stress model.

3.3.3. Multigrid strategy

In order to obtain grid converged solutions, one has to perform computation on grids with different resolution. For a grid with N cells in each direction, the cost of the PM is proportional to N^4 (employing a constant number of particles per cell). The cost of the FVM is proportional to $N^{1+3\beta}$, where β depends on the linear solver used for the matrix inversion. For more efficient computations on very fine grids, we devised and implemented a multigrid algorithm. The strategy is to start with a simulation on a coarse grid and to project the solution to subsequently finer grids. Properties of this re-gridding algorithm are that

- a multiblock grid solution can be projected onto a new grid with arbitrary resolution, while grid curvature is retained,
- the new grid is generated by interpolation from the old grid,
- all finite-volume and particle data are interpolated onto the new grid,
- linear or cubic spline interpolation can be used for interpolation,
- the stretching factors of the original grid are retained within each block,
- original block connectivities are honored for arbitrary topologies,
- all particles are redistributed, and
- local or global refinement is possible.

The recursive algorithm can be outlined as follows:

```

for each block do
  for each logical direction of the block do
    if necessary to re-grid
      - parameterize grid density function
      - interpolate cell centered data to nodes
      - parameterize node based data
      - generate new grid by distributing the desired
        number of nodes using parameterized grid density function
      - interpolate node based data to inner nodes
      - interpolate node based data back to cell centers
        by node averaging
    end
  end
end

```

Various interpolation kernels, e.g. linear or cubic splines, can be used. Projection is applied to node-based quantities. First, the grid lines are parameterized, and the new grid is created along these grid lines, leaving the first and last node of an interpolation direction untouched, see Fig. 6. Then, the quantities are interpolated similarly. Cell centered quantities Φ^m are interpolated to node \mathbf{x}_n by averaging over its adjacent cells \mathcal{C}_k ($k \in [0, 7]$), i.e.

$$\Phi_n = \frac{\sum_{k=0}^7 \Phi_k^m / |\mathbf{x}_n - \mathbf{x}_k^m|}{\sum_{k=0}^7 1 / |\mathbf{x}_n - \mathbf{x}_k^m|}, \quad (41)$$

where \mathbf{x}_k^m is the mid-point of cell \mathcal{C}_k . After projection, interpolation to the cell centers is achieved by averaging over the cell nodes. In a last step, the particles are reassigned to the new grid cells and particle number control (see Section 3.2.6) ensures that the particle number is within the defined limits when the simulation is restarted with the new grid.

3.4. Consistency

While the JPDF method is fully consistent at the level of the governing equations, this is not always guaranteed numerically. In this section, two correction schemes dealing with consistency between particle and finite-volume data are presented.

3.4.1. Particle position correction

In the hybrid code, the mean fluid density is computed twice, i.e. $\langle \rho \rangle$ by the finite-volume scheme and the particle number density, ρ_p , by the particle method. For a robust, stable and accurate solution algorithm it is essential that these two fields are consistent. This is guaranteed at the analytical level, but in general not numerically. Several methods have been proposed to numerically enforce this consistency, e.g. by Zhang

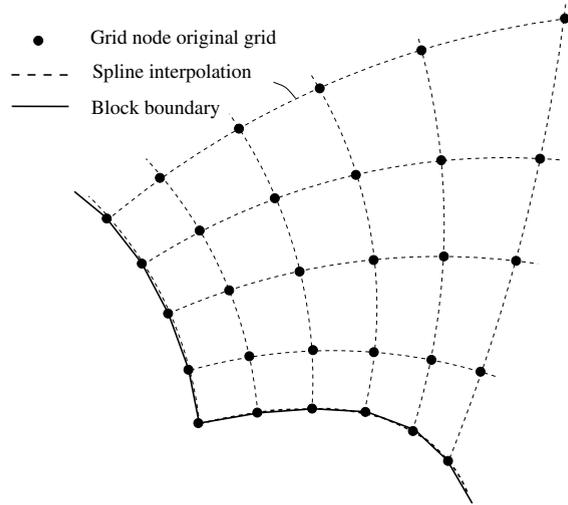


Fig. 6. Multigrid projection: The gridlines between the nodes are parameterized by, e.g., cubic splines and the new grid is generated according to the desired resolution along these lines.

and Haworth [32] or Muradoglu et al. [40]. Here we devise a simpler particle position correction scheme based on a particle drift flux to reduce $|\langle \rho \rangle - \rho_p|$. Given are the mass discrepancies dM_i and dM_j in two adjacent cells, where

$$dM_i = (\rho_p - \langle \rho \rangle)_i V_i^{\mathcal{C}}. \quad (42)$$

The volume of cell i is denoted by $V_i^{\mathcal{C}}$. Now the correction flux

$$A_{i,j} = \frac{1}{2\Delta t_{\text{PM}}} (dM_i - dM_j) \quad (43)$$

across the interface between the two cells is computed and added to the fluxes obtained from the finite volume solver, which are also defined at the cell interfaces. The interpolation to the particle positions is then obtained by Eq. (A.4), where the velocities u^{T_i} at the cell interfaces in Eq. (A.2) are now obtained from the sum of the fluxes from the FVM and the position correction. Despite its simplicity, in all our studies this correction procedure proved to be very effective and robust.

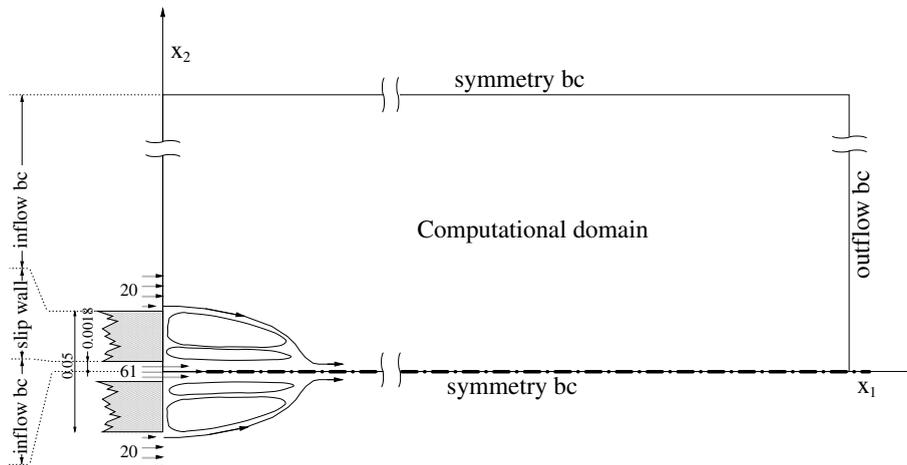


Fig. 7. Validation: sketch of the bluff body flow test case.

3.4.2. Particle velocity correction

Analytically, the stochastic differential equation (11) guarantees that the expectation of the fluctuating particle velocity, \mathbf{u}^* , is always zero everywhere, if it was zero initially. However, numerically this is not automatically fulfilled. To enforce that, first the fluctuating velocity is extracted at each grid node. By employing time averaging (Eq. (38)) using the memory factor v_{fluct} and subsequent interpolation to the particle position, one obtains the vector $\mathbf{u}_{\text{fluct}}$ which is then used to correct the particle velocity as

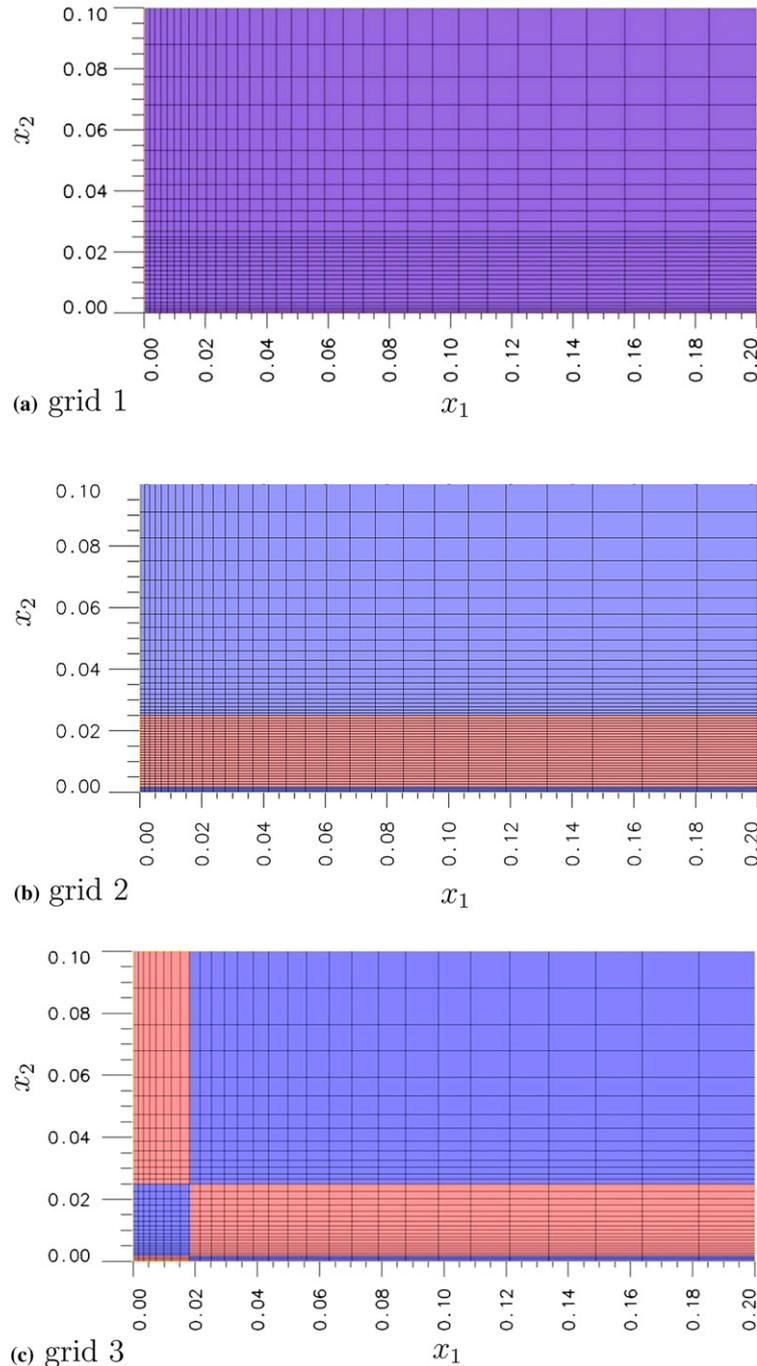


Fig. 8. Validation: three different grids with (a) one block (1024 cells), (b) three blocks (1456 cells), and (c) six blocks (1008 cells); different blocks are visualized by different colors. This figure appears in color in the online version of this paper.

$$\mathbf{u}_{\text{corrected}}^* = \mathbf{u}_{\text{uncorrected}}^* - \mathbf{u}_{\text{fluct}}. \quad (44)$$

This simple velocity correction technique is applied after each time step and according to our experience it works very well, i.e. $\tilde{\mathbf{u}}_l$ could always be reduced to insignificant values.

4. Results

The entire implementation of the new JPDF solution algorithm, which is based on our Lagrangian/Eulerian multiblock simulation kit (*LEMBSK*), was validated with a number of test cases using different multiblock grids. In the following sections, comparisons with the established hybrid code *pdf2dfv* [29] for a bluff-body stabilized flow are presented. The various algorithmic components and their influence on the overall performance are analyzed. Moreover, it is demonstrated that the new JPDF solution algorithm can deal with complex geometries, which is crucial for most practical applications. Note that, if not mentioned otherwise, SI units are used.

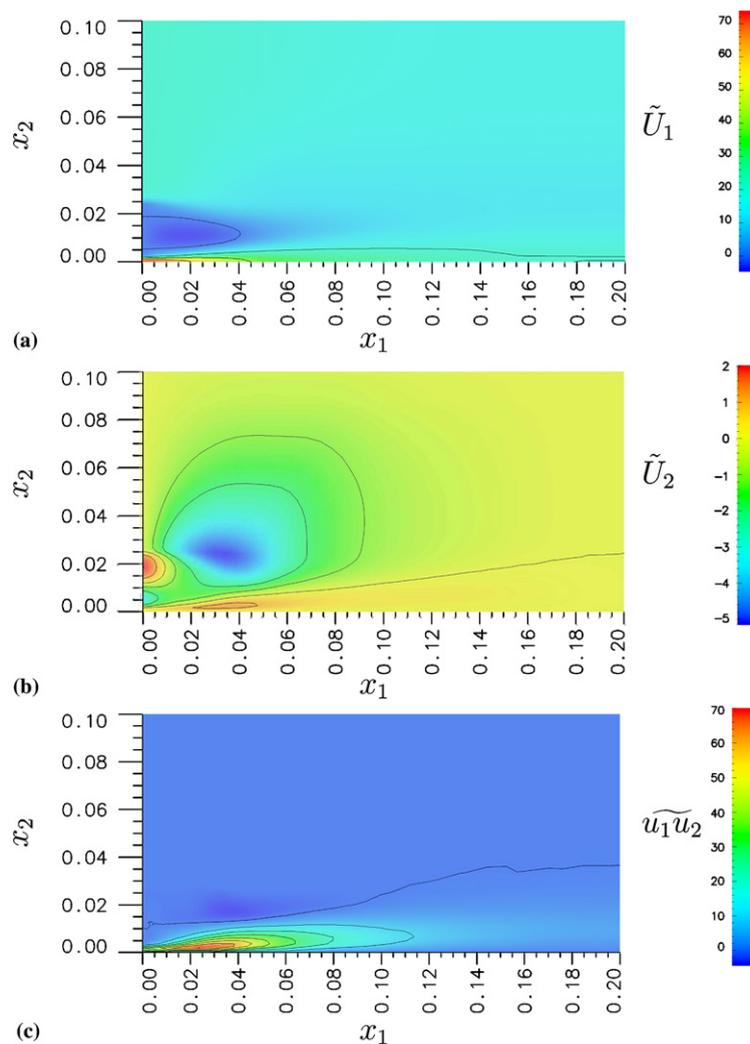


Fig. 9. Validation with grid 1: (a) \tilde{U}_1 , (b) \tilde{U}_2 , (c) $\widetilde{u_1 u_2}$; the contour increment in is (a) 20, in (b) 1, and in (c) 10. This figure appears in color in the online version of this paper.

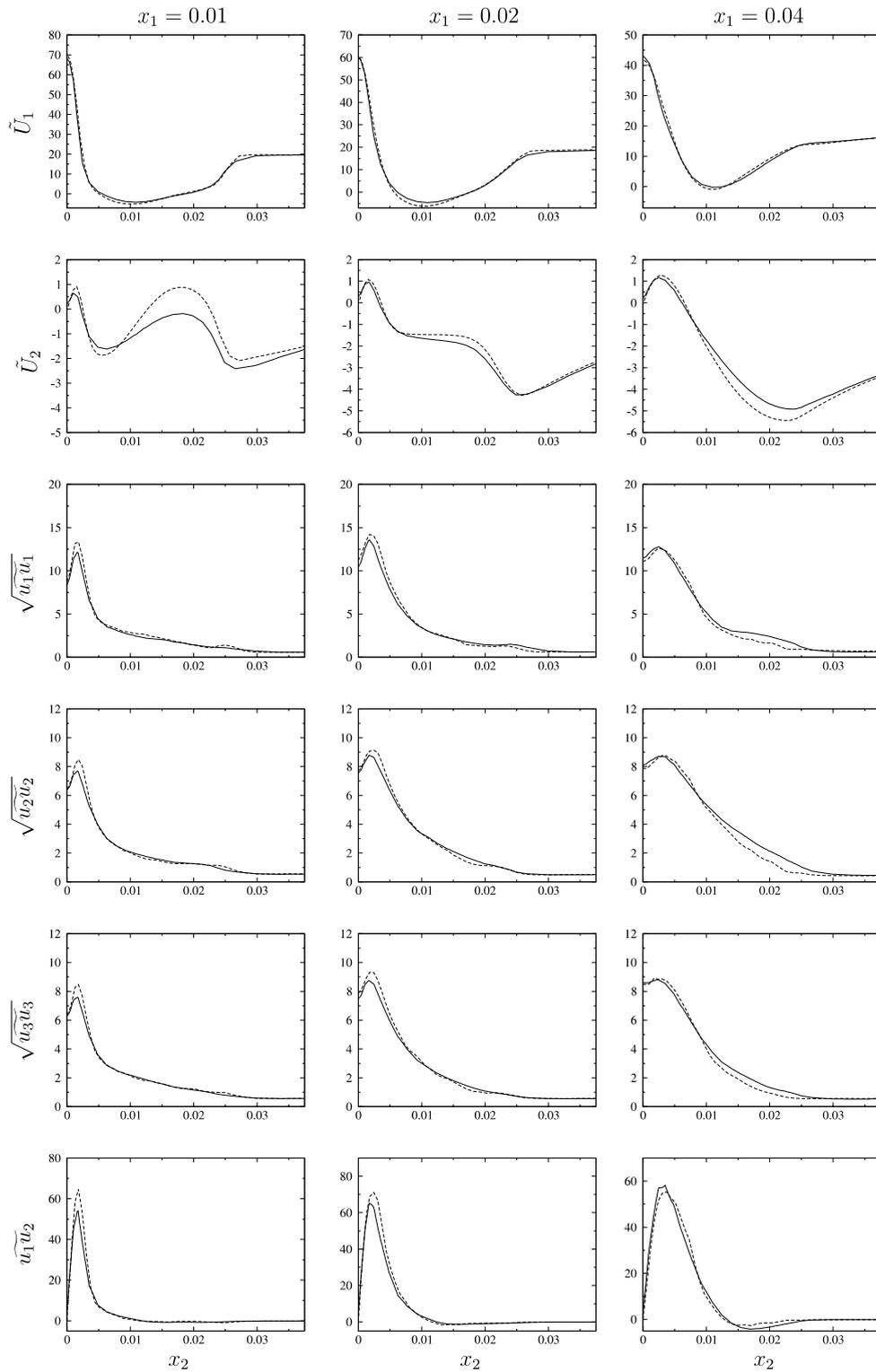
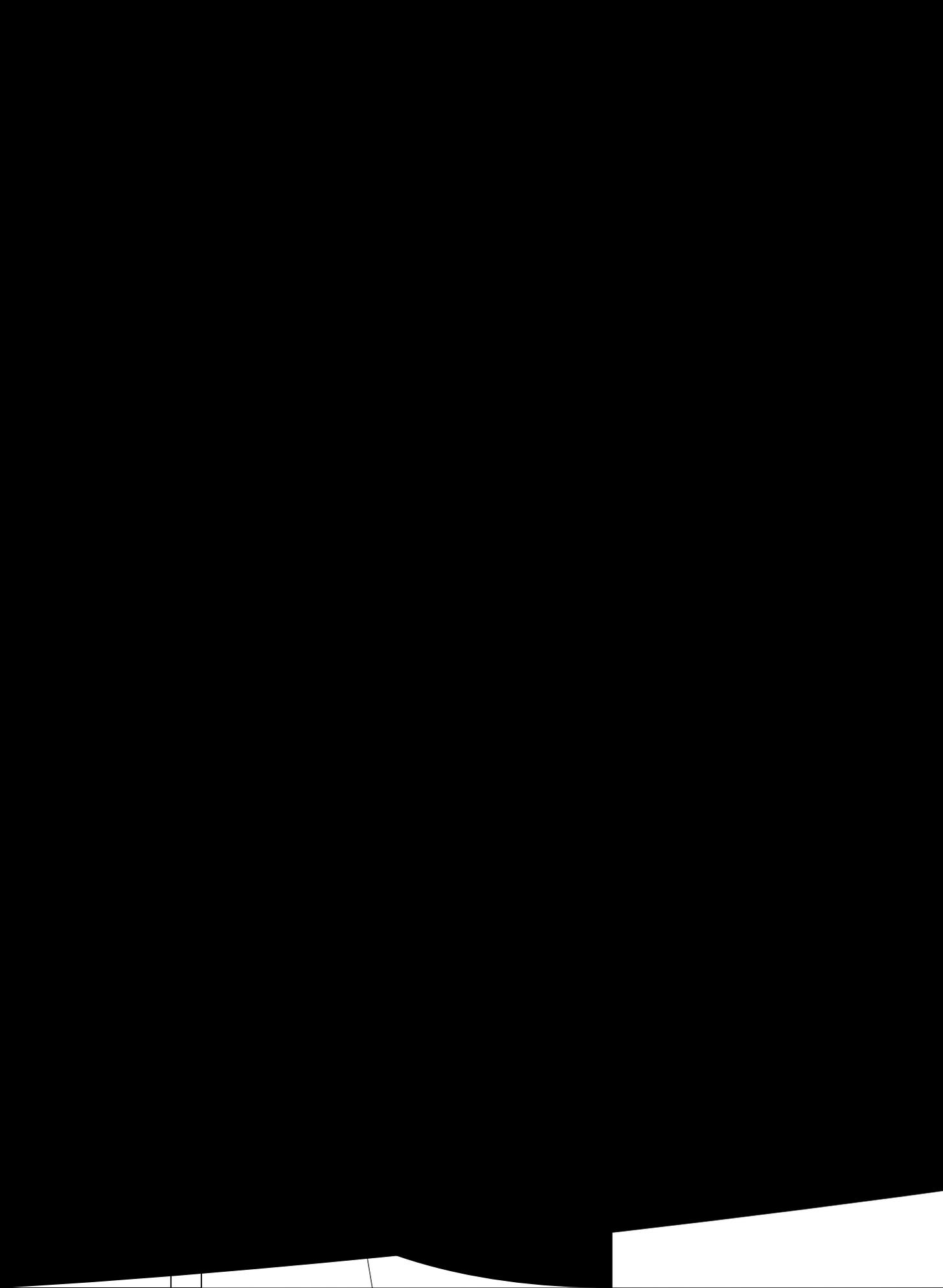


Fig. 10. Validation against *pdf2dfv*: comparison for various quantities at three downstream locations; *LEMBSK* results for grid 1 — and reference results by *pdf2dfv* ----.



4.1. Description of the bluff-body test case

Bluff-body stabilized flows have extensively been studied experimentally and theoretically [41–46,13] and various data-sets are available for validation. Here, the test case is described only briefly; for details and experimental data we refer to [41,46].

In Fig. 7, a sketch of the axi-symmetric test case with the computational domain is shown. For simplicity, however, we used an equivalent, but plane configuration and consider non-reacting flow. The jet and the bluff-body diameters are 0.0036 and 0.05, respectively. Both, jet and co-flow consist of air with constant density. Bulk velocities of jet and co-flow are 61 and 20. At the inflow boundary, profiles of the mean and rms velocities are imposed [13] and at the centerline and far-field boundaries, symmetry boundary conditions are applied (see Fig. 7). In the bluff-body region, slip-wall boundary conditions and at the right boundary, outflow conditions are employed (Sections 3.2.3 and 3.1).

4.2. Algorithm validation

In this section, it is shown for three different grids (Fig. 8) that *LEMBSK* and *pdf2dfv* solutions are in good agreement. Note that, since the topic of the paper is the solution algorithm and not turbulence modeling, all comparative simulations were performed with the same models, boundary conditions and grids. Finally, in Section 4.4, it is demonstrated that *LEMBSK* is capable of handling non-orthogonal, complex multiblock grids.

In Fig. 9, the single block (32 × 32 grid) *LEMBSK* solution is depicted, i.e. the mean velocity components, \tilde{U}_1 and \tilde{U}_2 , and the shear stress $\tilde{u}_1\tilde{u}_2$. In Fig. 10, *LEMBSK* and *pdf2dfv* are compared, i.e. profiles of \tilde{U}_1 , \tilde{U}_2 , $\tilde{u}_1\tilde{u}_1$, $\tilde{u}_2\tilde{u}_2$, $\tilde{u}_3\tilde{u}_3$ and $\tilde{u}_1\tilde{u}_2$ are shown at three different down-stream locations. In general, there exists very good

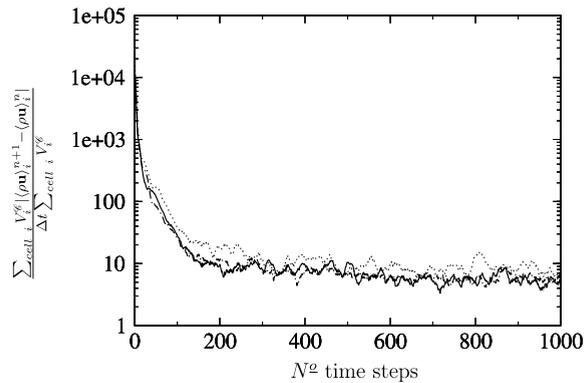


Fig. 12. Validation: residual for grid 1 —, grid 2 ---, and grid 3 ···.

Table 3

Algorithm efficiency: implicit solver and particle number control (runs A.i) and multigrid algorithm (runs B.i)

	FVM	Δt	No. cells	Particle number control	CPU-time/ Δt (%)	Re-gridding
A.1	expl.		1024	No	100	No
A.2	impl.	4×10^{-4}	1024	No	14	No
A.3	impl.	4×10^{-4}	1024	Yes	7	No
B.1	impl.	5×10^{-5}	4032	Yes	100	No
B.2	impl.	2×10^{-4}	252	Yes	2	No
B.3	impl.	1×10^{-4}	1008	Yes	9	Yes (from B.2)
B.4	impl.	5×10^{-5}	4032	Yes	100	Yes (from B.3)
B.5	impl.	1×10^{-4}	1008	Yes	9	No

agreement, except in the case of \tilde{U}_2 , where a discrepancy can be observed. This is due to first order treatment of the boundary conditions in $pdf2dfv$, which are treated second order in *LEMBSK*.

Fig. 11 shows the same profiles for the three different grids. Despite the fact that all grids are rather coarse and only similar, but not identical, the solutions are in good agreement. Again, the largest discrepancies are observed for \tilde{U}_2 , which is a very sensitive quantity with respect to grid resolution. Since its absolute value is only about one percent of \tilde{U}_1 , even very small deviations of \tilde{U}_1 can lead to large changes of \tilde{U}_2 . In particular, the good agreement for the multiblock grid cases is remarkable. The large gradients at the block boundaries near the jet-inflow are correctly captured. Note that these gradients are important for the development of $\widehat{u_1 u_2}$, which largely determines the spreading rate.

For stationary solutions, where time averaging is used to improve the statistics, it has to be ensured that statistical stationarity is reached. In Fig. 12, the residual of the FVM is plotted as a function of time steps. It can be observed for all three grids that the residual drops rapidly and then becomes stationary. Note that the convergence rate is approximately the same for all three grids.

4.3. Efficiency of the algorithm

Algorithmical efficiency is one of the major concerns for JPDF simulations with complex grids and most of the efficiency issues are related to the PM, e.g. particle sub-time stepping (Section 3.2.4) and particle number control (Section 3.2.6). On the FVM side it is the block-implicit MBG solver (Section 3.1) that allows for large time steps. Effective coupling depends on the exponentially weighted moving time averaging (Section 3.3.1), the algebraic proxy model (Section 3.3.2) and the multigrid algorithm (Section 3.3.3).

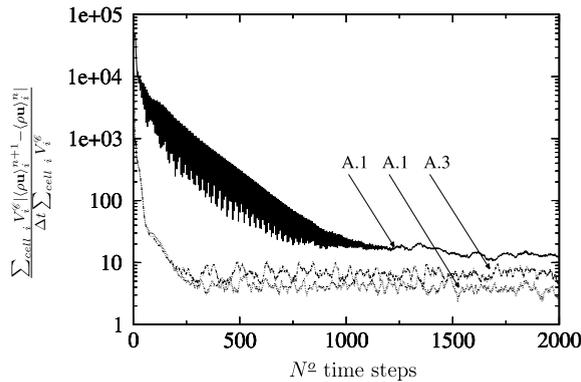


Fig. 13. Efficiency implicit solver and particle number control: residual for *LEMBSK*; case A.1 —, A.2 ···, A.3 -·-·.

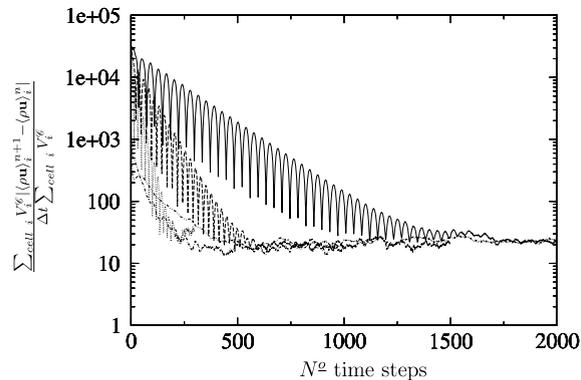
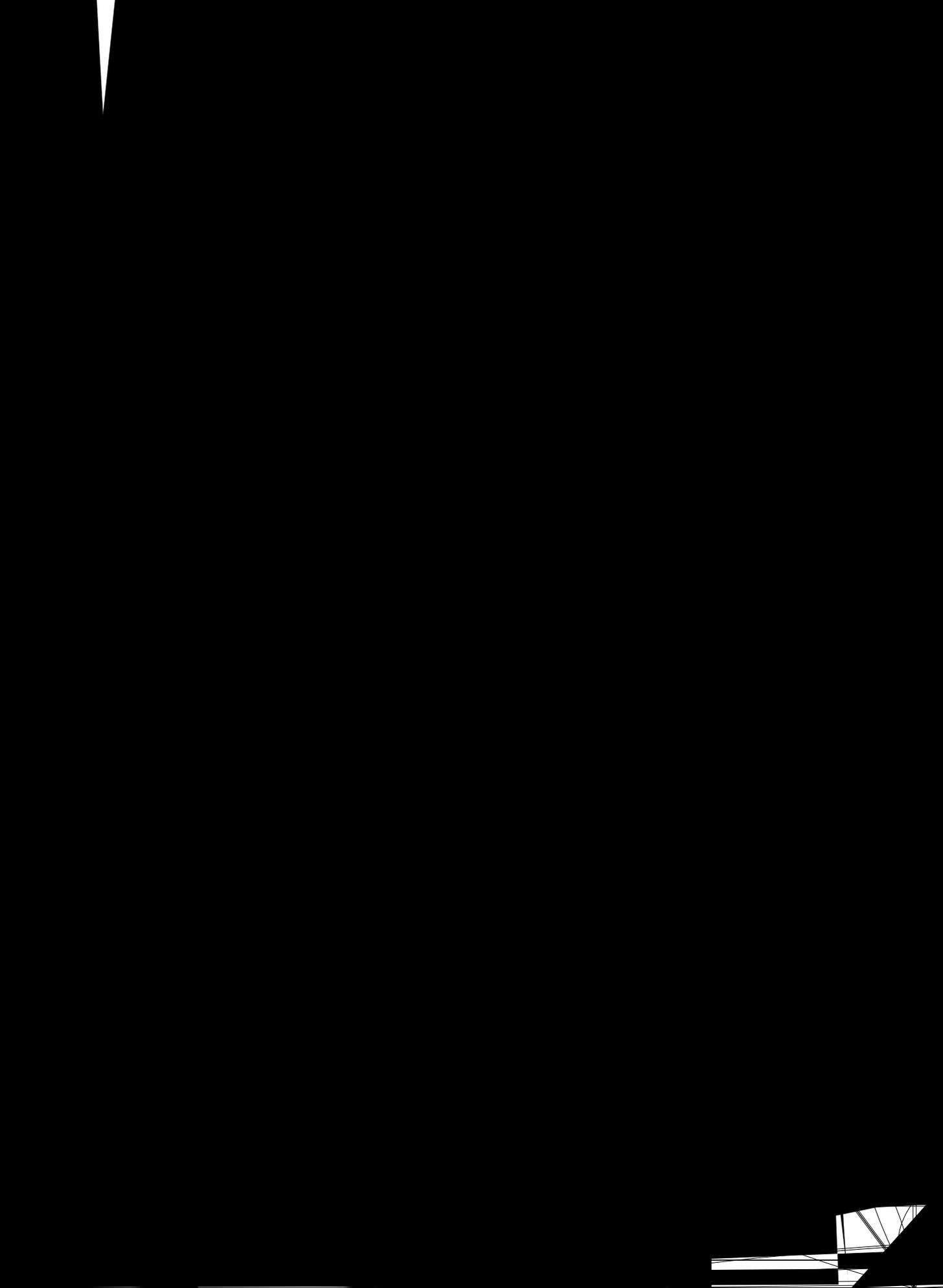


Fig. 14. Efficiency multigrid algorithm: residual for *LEMBSK*; case B.1 —, B.2 ···, B.3 -·-·, B.4 - - -, B.5 - - - -.







The effectiveness of the multigrid strategy is assessed with the runs B.1–B.5. B.1 was initialized with no flow in the entire domain and is the reference solution on the finest grid. B.2 was also initialized with no flow in the entire domain and run for 300 time steps. The solution of B.2 was then projected onto the grid of B.3 using the multigrid algorithm and after 1000 time steps further projected to the grid of B.4. It can be seen in Fig. 14 that after restarting, simulations B.3 and B.4 converge significantly faster than B.1. Moreover, after re-gridding the extracted particle data already agrees well with the converged values, which reduces the required duration of time averaging. B.5 was only added to demonstrate that even with one step already a gain in efficiency is obtained. Note that the solutions of B.1 and B.4 are slightly different than those of A.i, which is due to the higher grid resolutions. If not started on regrided data, the simulations are initialized using no flow in the entire domain. The results of the different runs are plotted in Fig. 16.

More difficult is the quantitative assessment of the improvement due to the algebraic turbulence model. According to our experience the Reynolds stresses extracted from the particle field can become very distorted at the beginning of a simulation. This is mainly due to the very large time steps (we employ the same time steps for the PM as for the FVM), which can lead to an inconsistent particle distribution in physical space, if the mean velocity field changes rapidly (which is the case at the beginning of a simulation). This problem could be reduced by employing an algebraic proxy turbulence model. Note, however, that the algebraic proxy model approach only has a significant impact in complex simulations.

In all simulations the exponentially weighted moving time averaging technique was applied. This is very important for efficient simulations, since it dramatically reduces the number of particles which have to be employed.

4.4. Complex grid test case

Here we demonstrate that the new JPDF solution algorithm can deal with geometrically complex cases. Therefore, the flow in a generic combustion chamber with a bluff body and a secondary inlet is considered, and a multiblock grid consisting of 29 blocks and 7424 cells was generated. In Fig. 17 a sketch of the test case, the boundary conditions and the multiblock grid is shown. Only half of the symmetric flow scenario was simulated and approximately 20 particles were employed per cell. Walls are simply treated by slip boundary conditions, i.e. no special wall modeling was applied. At the left inflow boundary a mean velocity profile with a

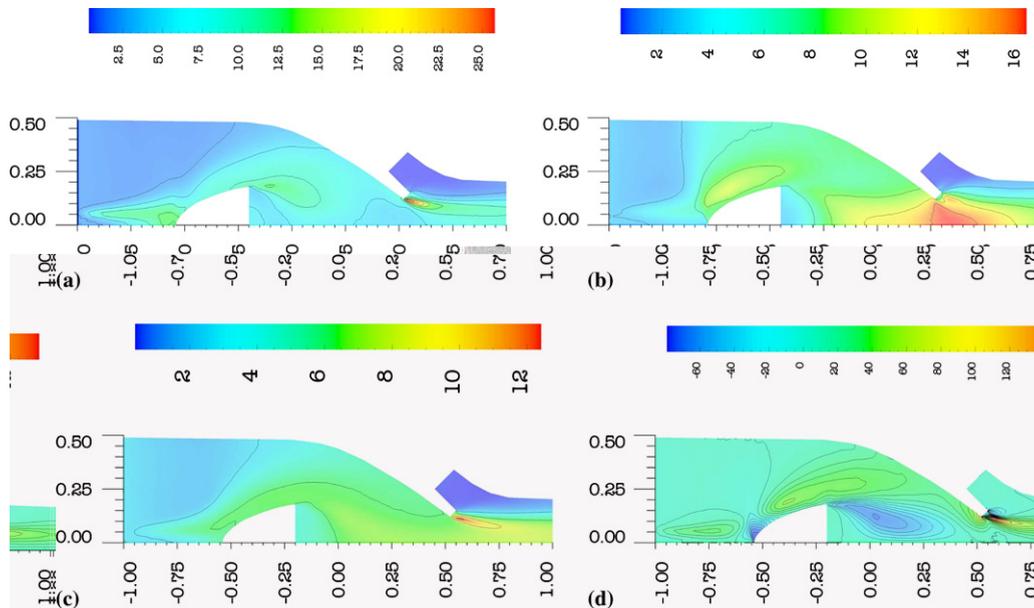


Fig. 20. Complex grid test case; contours of (a) $\sqrt{u_1 u_1}$, (b) $\sqrt{u_2 u_2}$, (c) $\sqrt{u_3 u_3}$ (contour increment is 3), and (d) $u_1 u_2$ (contour increment is 10). This figure appears in color in the online version of this paper.

maximum value of 50 at the centerline was imposed. An additional inlet provides a secondary air stream (the mean inlet velocity is 60) and at the right boundary, outflow conditions are specified. Moreover, mixing of a cold fuel stream coming from the left inlet boundary is simulated, but for the present test case no reactions are considered. Note that the objective of this study was not a quantitative assessment of the device, but to show that the algorithm can be applied for geometrically complex cases, that it reaches steady state and that the resulting particle statistics is consistent with the finite-volume data.

For the fine grid simulation we employed the re-gridding strategy. Initially, a solution on a coarse grid consisting of 1856 cells was computed. This solution was then projected onto a fine grid consisting of 7424 cells (this corresponds to a re-gridding factor of two in each direction) before the fine grid run was restarted. The corresponding convergence history is shown in Fig. 18; the time of re-gridding is indicated by a vertical dotted line. Note that all the following results are presented on the fine grid.

Figs. 19 and 20 show contour plots of the mean velocity components, \tilde{U}_1 and \tilde{U}_2 , and the turbulent velocity statistics. It can be seen that a recirculation zone develops behind the bluff-body and that the fluid accelerates near the outlet. Turbulence production (indicated by large Reynolds stress values) peaks in the shear layer near the inflow and in the shear region, where the secondary flow joins the main stream. Note that the flow is highly anisotropic and inhomogeneous.

The mean of the fluctuating particle velocities, $\langle \mathbf{u} \rangle$, and the normalized deviation of the time averaged particle number density from the fluid density, i.e. $(\rho_p - \langle \rho \rangle) / \langle \rho \rangle$, are plotted in Fig. 21. It can be seen that $\langle \mathbf{u} \rangle$ remains within the range of a few percent of the mean velocity. Moreover, in most parts of the domain the

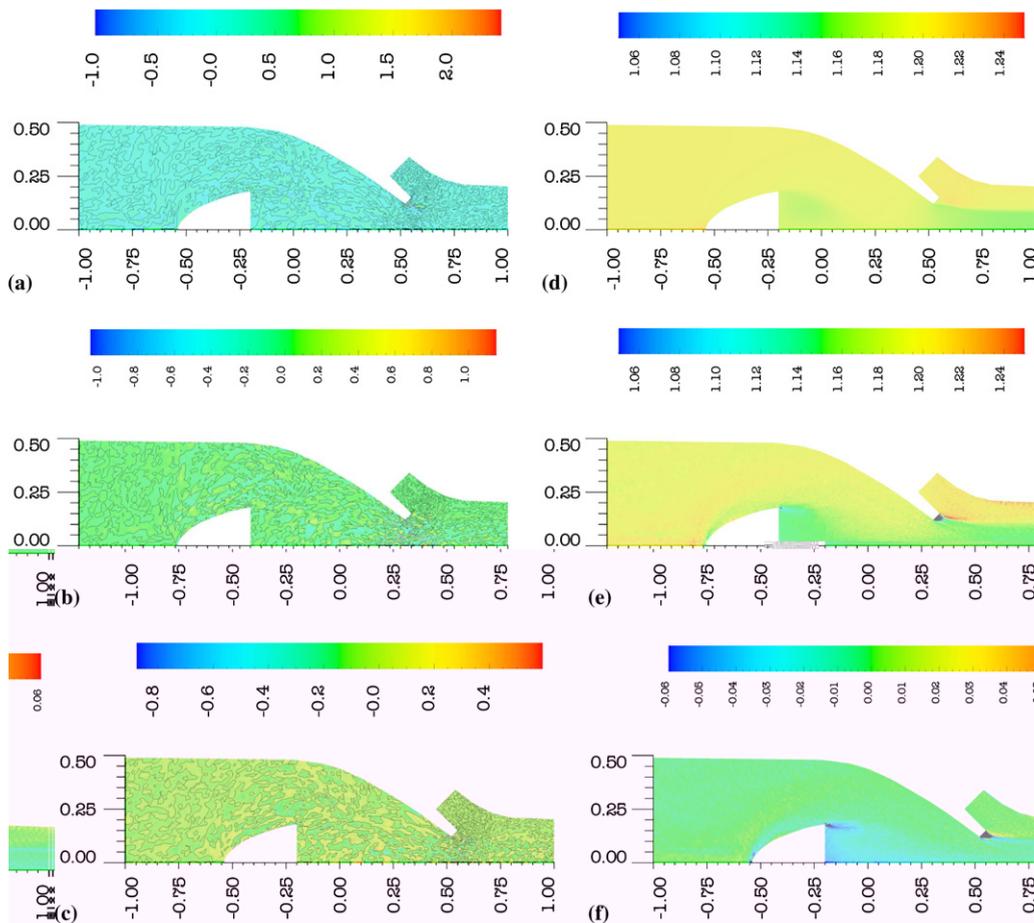


Fig. 21. Complex grid test case; consistency: mean particle velocity fluctuations: (a) $\langle u_1 \rangle$, (b) $\langle u_2 \rangle$ and (c) $\langle u_3 \rangle$ (the contour increment is 2); (d) $\langle \rho \rangle$, (e) ρ_p , and (f) the relative density discrepancy $(\rho_p - \langle \rho \rangle) / \langle \rho \rangle$. This figure appears in color in the online version of this paper.

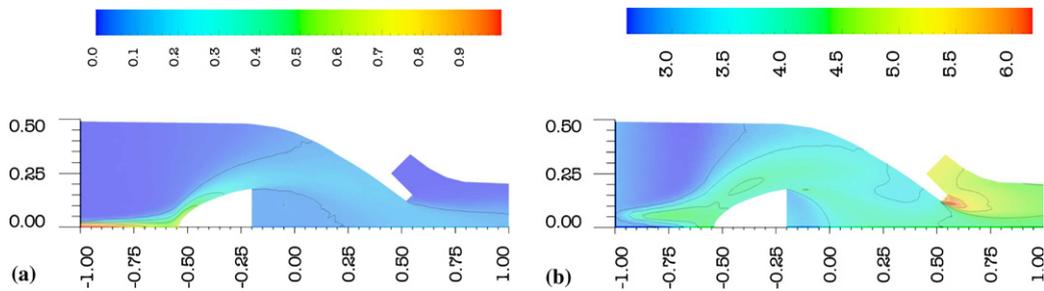


Fig. 22. Complex grid test case; (a) mean passive scalar (contour increment is 0.2), (b) logarithm of conditional turbulence frequency, i.e. $\log(\Omega)$ (contour increment is 0.5). This figure appears in color in the online version of this paper.

difference between the mean particle density, ρ_p , and $\langle \rho \rangle$ is less than five percent of the mean fluid density. This result is of particular importance, since it demonstrates that the presented method is numerically consistent and that the JPDF can accurately be represented.

Finally, in Fig. 22 the distribution of a passive scalar (in this case of the cold fuel), which is injected near the centerline at the inflow boundary, and of the conditional turbulence frequency, Ω , are shown.

5. Conclusion

A new hybrid algorithm to solve the joint velocity-frequency-composition PDF transport equation for complex grids has been developed, implemented and tested. The main goal was to demonstrate that it is feasible to apply JPDF methods for applications with complicated 3D geometries. The basis is our Lagrangian/Eulerian multiblock simulation kit (*LEMBSK*), which is characterized by the following properties:

- Complex 3D geometries are handled by locally structured, globally unstructured multiblock grids.
- Conservative particle tracking based on efficient particle management and accurate interpolation and extraction schemes makes it easy to implement different coupled Lagrangian/Eulerian (PM/FVM) algorithms.
- Efficiency is improved by particle sub-time stepping and particle number control.
- Due to the modular, object oriented code design (C++), *LEMBSK* is very flexible and can easily be adapted for different problems.

During the development of the JPDF solution algorithm, various important issues, most of them related to computational efficiency and numerical consistency, had to be resolved or adapted. Examples are

- an implicit FVM for 3D multiblock grids to allow for larger time steps,
- exponentially weighted moving time averaging to reduce statistical and bias errors,
- an algebraic proxy turbulence model to get reasonable initial conditions and faster convergence,
- a multigrid method for efficient fine grid simulations,
- a particle position correction scheme to achieve numerical consistency between fluid and particle number density, and
- a particle velocity correction scheme to ensure numerically that the expectation of the fluctuating particle velocity remains zero.

The algorithm is general and also applicable to non-stationary problems and non-constant density flows, however, at this point only results for constant density flows were presented. For a bluff-body stabilized flow test case it was shown that the results are in very good agreement with the solutions obtained with the established JPDF code *pdf2dfv* and it could be demonstrated that the algorithm can handle applications with complex geometries. This is very promising for JPDF methods in general and important for their broader use in practical studies, e.g. in industry. Naturally, the next steps will include the integration of various combustion and mixing models, and the development of a massive parallel version.

Appendix A. Velocity interpolation

For the time integration of the particles in physical space the Favre averaged velocity field $\tilde{\mathbf{U}}$ has to be interpolated to the particle locations. In the previous work by Jenny et al. [29] much effort was spent to develop a second order, conservative velocity interpolation scheme. Recently, it was demonstrated by Meyer and Jenny [47] that it is justified to use a less rigorous, but simpler velocity interpolation, which we extended for multi-block grids. As the scheme by Jenny et al., it is based on the finite-volume fluxes at the cell faces and is conservative for constant density flow.

Let us consider a particle with position \mathbf{x}^* in a grid cell \mathcal{C} . The velocity at face Υ_i , which is obtained from the finite-volume fluxes, is u^{r_i} , and the corresponding face normal vector is \mathbf{N}_i . These quantities are then interpolated to the particle position as

$$\mathbf{N}_i^* = W_i \mathbf{N}_i + W_{i+3} \mathbf{N}_{i+3} \quad \forall i \in \{0, 1, 2\} \quad (\text{A.1})$$

and

$$u^{r_{i,*}} = W_i u^{r_i} + W_{i+3} u^{r_{i+3}} \quad \forall i \in \{0, 1, 2\}, \quad (\text{A.2})$$

where

$$W_i = \sum_{l|\mathbf{x}_l \in \Upsilon_i} w_l. \quad (\text{A.3})$$

Finally, the interpolated mean velocity at the particle position \mathbf{x}^* is obtained by solving the system

$$\tilde{\mathbf{u}}^* \cdot \mathbf{N}_i^* = u^{r_{i,*}} \quad \forall i \in \{0, 1, 2\}. \quad (\text{A.4})$$

For the computation of the velocity gradients at the particle locations we use the interpolation scheme (27), where the velocities at the nodes are estimated as averages of their surrounding cells (Eq. (41)).

Appendix B. Geometrical data for the particle management

Here we address the efficiency of the particle management of Section 3.2.2. In order to determine within which tetrahedron $\Gamma_{i,j}$ of cell \mathcal{C} a particle with position \mathbf{x}^* is located, the vectors

$$\mathbf{M}_{i,j} = (\mathbf{x}_{i,j} - \mathbf{x}^m) \times (\mathbf{x}_{i,\text{mod}_4(j+1)} - \mathbf{x}^m) \quad (\text{B.1})$$

and

$$\mathbf{S}_{i,j} = (\mathbf{x}_{i,j} - \mathbf{x}^{Y_i}) \times (\mathbf{x}^m - \mathbf{x}^{Y_i}) \quad (\text{B.2})$$

are stored. Note that these are computed once only for each tetrahedron and used throughout the whole simulation. If

$$M_{i,j} \cdot (\mathbf{x}^* - \mathbf{x}^m) > 0 \quad (\text{B.3})$$

is fulfilled $\forall j \in \{0, \dots, 3\}$, then the particle is located inside the pyramid consisting of cell face Υ_i and cell midpoint \mathbf{x}^m . Finally, the corresponding tetrahedron, $\Gamma_{i,j}$, is the one for which

$$S_{i,\text{mod}_4(j+k)} \cdot (\mathbf{x}^* - \mathbf{x}^m)(-1)^k > 0 \quad \forall k \in \{0, 1\} \quad (\text{B.4})$$

is satisfied. This localization procedure only is efficient since $\mathbf{M}_{i,j}$ and $\mathbf{S}_{i,j}$ do not have to be computed more than once.

References

- [1] B.E. Launder, D.B. Spalding, *Mathematical Models of Turbulence*, Academic Press, London, 1972.
- [2] B.E. Launder, Phenomenological modelling: present.. and future? in: J.L. Lumley (Ed.), *Whither Turbulence? Turbulence at the Crossroads*, Lecture Notes in Physics, vol. 357, Springer, Berlin, 1990, pp. 439–485.
- [3] D.C. Wilcox, *Turbulence Modeling for CFD*, DCW Industries, La Cañada, CA, 1993.
- [4] S.B. Pope, *Turbulent Flows*, Cambridge University Press, Cambridge, 2000.

- [5] R.O. Fox, *Computational Models for Turbulent Reacting Flows*, Cambridge University Press, Cambridge, 2003.
- [6] S.B. Pope, Y.L. Chen, The velocity-dissipation probability density function model for turbulent flows, *Phys. Fluids A* 2 (1990) 1437–1449.
- [7] P.R.V. Slooten, Jayesh, S.B. Pope, Advances in PDF modeling for inhomogeneous turbulent flows, *Phys. Fluids* 10 (1998) 246–265.
- [8] D.C. Haworth, S.B. Pope, A generalized Langevin model for turbulent flows, *Phys. Fluids* 29 (1986) 387–405.
- [9] S.B. Pope, Lagrangian PDF methods for turbulent flows, *Annu. Rev. Fluid Mech.* 26 (1994) 23–63.
- [10] S.B. Pope, On the relationship between stochastic Lagrangian models of turbulence and second-moment closures, *Phys. Fluids* 6 (1994) 973–985.
- [11] P.R.V. Slooten, S.B. Pope, PDF modeling of inhomogeneous turbulence with exact representation of rapid distortions, *Phys. Fluids* 9 (1997) 1085–1105.
- [12] S.B. Pope, Computationally efficient implementation of combustion chemistry using in situ adaptive tabulation, *Combust. Theory Modelling* 1 (1997) 41–63.
- [13] P. Jenny, M. Muradoglu, K. Liu, S.B. Pope, D.A. Caughey, PDF simulations of a bluff-body stabilized flow, *J. Comput. Phys.* 169-1 (2001) 1–23.
- [14] M.S. Anand, A.T. Hsu, S.B. Pope, Calculations of swirl combustors using joint velocity-scalar probability density function method, *AIAA J.* 35 (1997) 1143–1150.
- [15] B.J. Delarue, S.B. Pope, Application of PDF methods to compressible turbulent flows, *Phys. Fluids* 9 (1997) 2704–2715.
- [16] J.P. Minier, J. Pozorski, Analysis of a PDF model in a mixing layer case, in: *Tenth Symposium on Turbulent Shear Flows*, 1995, pp. 26.25–26.30.
- [17] K. Liu, S.B. Pope, D.A. Caughey, Calculations of bluff-body stabilized flames using a joint probability density function model with detailed chemistry, *Combust. Flame* 141 (2005) 89–117.
- [18] J. Xu, S.B. Pope, PDF calculations of turbulent nonpremixed flames with local extinction, *Combust. Flame* 123 (2000) 281–307.
- [19] Q. Tang, J. Xu, S.B. Pope, Probability density function calculations of local extinction and no production in piloted-jet turbulent methane/air flames, *Proc. Combust. Inst.* 28 (2000) 133–139.
- [20] V. Raman, R.O. Fox, A.D. Harvey, Hybrid finite-volume/transported PDF simulations of a partially premixed methane–air flame, *Combust. Flame* 136 (2004) 327–350.
- [21] M. Muradoglu, K. Liu, S. Pope, PDF modeling of a bluff-body stabilized turbulent flame, *Combust. Flame* 132 (2003) 115–137.
- [22] P.A. Nooren, H.A. Wouters, T.W.J. Peeters, D. Roekaerts, U. Maas, D. Schmidt, Monte Carlo PDF simulation of a turbulent natural-gas diffusion flame, in: *Twenty-Sixth Symp. (Int'l) on Combust.*, Combustion Institute, Pittsburgh, 1996, pp. 79–96.
- [23] A.R. Masri, S.B. Pope, PDF calculations of piloted turbulent non-premixed flames of methane, *Combust. Flame* 81 (1990) 13–29.
- [24] V. Saxena, S.B. Pope, PDF calculations of major and minor species in a turbulent piloted jet flame, in: *Combustion Institute*, Pittsburgh, 1998, pp. 1081–1086.
- [25] C. Dopazo, Recent developments in PDF methods, in: P.A. Libby, F.A. Williams (Eds.), *Turbulent Reacting Flows*, Academic Press, London, 1994, pp. 375–474 (Chapter 7).
- [26] S.B. Pope, PDF methods for turbulent reactive flows, *Prog. Energy Combust. Sci.* 11 (1985) 119–192.
- [27] S.B. Pope, Computations of turbulent combustion: progress and challenges, in: *Twenty-Third Symp. (Int'l) on Combust.*, Combustion Institute, Pittsburgh, 1990, pp. 591–612.
- [28] M. Muradoglu, P. Jenny, S.B. Pope, D.A. Caughey, A consistent hybrid finite-volume/particle method for the PDF equations of turbulent reactive flows, *J. Comput. Phys.* 154 (1999) 342–371.
- [29] P. Jenny, S.B. Pope, M. Muradoglu, D.A. Caughey, A hybrid algorithm for the joint PDF equation of turbulent reactive flows, *J. Comput. Phys.* 166 (2001) 218–252.
- [30] P. Jenny, C. Wolfsteiner, S.H. Lee, L.J. Durlofsky, Modeling flow in geometrically complex reservoirs using hexahedral multi-block grids, *SPE J.* (2002) 149–157.
- [31] S.H. Lee, P. Jenny, H. Tchelepi, A finite-volume method with hexahedral multiblock grids for modeling flow in porous media, *Comput. Geosci.* 6 (3) (2002) 353–379.
- [32] Y.Z. Zhang, D.C. Haworth, A general mass consistency algorithm for hybrid particle/finite-volume PDF methods, *JCP* 194 (2004) 156–193.
- [33] Y.Z. Zhang, E.H. Kung, D.C. Haworth, A PDF method for multidimensional modeling of HCCI engine combustion: effects of turbulence/chemistry interactions on ignition timing and emissions, *Proc. Combust. Inst.* 30 (2005) 2763–2771.
- [34] H. Möbus, P. Gerlinger, D. Brüggemann, Scalar and joint scalar-velocity-frequency Monte Carlo PDF simulation of supersonic combustion, *Combust. Flame* 132 (2003) 3–24.
- [35] S. James, M.S. Andand, S.B. Pope, The Lagrangian PDF transport method for simulations of gas turbine combustor flows, in: *38th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, no. AIAA 2002-4017, Indianapolis, IN, 2002.
- [36] P. Jenny, B. Müller, H. Thomann, Riemann solver for compressible combustion codes, in: *Proceedings of First International Symposium on Finite Volumes for Complex Applications*, Rouen, France, 1996, pp. 705–715.
- [37] C. Hirsch *Numerical Computation of Internal and External Flows*, vol. 2, Wiley, New York, 2000.
- [38] S.B. Pope, PDF2DV: a Fortran code to solve the modelled joint PDF equations for two-dimensional recirculating flows, Cornell University, 1994, unpublished.
- [39] J. Xu, S.B. Pope, Assessment of numerical accuracy of PDF/Monte Carlo methods for turbulent reacting flows, *J. Comput. Phys.* 152 (1999) 192–230.
- [40] M. Muradoglu, S.B. Pope, D.A. Caughey, The hybrid method for the PDF equations of turbulent reactive flows: consistency conditions and correction algorithms, *J. Comput. Phys.* 172 (2001) 841–878.

- [41] B.B. Dally, D.F. Fletcher, A.R. Masri, Flow and mixing fields of turbulent bluff-body jets and flames, *Combust. Theory Modelling* 2 (1998) 193–219.
- [42] S.M. Correa, S.B. Pope, Comparison of a Monte Carlo PDF finite-volume mean flow model with bluff-body Raman data, in: *Twenty-Fourth Symp. (Int'l) on Combust.*, Combustion Institute, Pittsburgh, 1992, pp. 279–285.
- [43] S.M. Correa, S.B. Pope, Measurements and modeling of a bluff-body stabilized flame, *Combust. Flame* 89 (1992) 195–213.
- [44] W.M. Roquemore, R.L. Britton, S.S. Sandhu, Investigation of the dynamic behaviour of a bluff-body diffusion flame using flame emission, *AIAA J.* 21 (10) (1983) 1410–1417.
- [45] A.R. Masri, R.W. Dibble, R.S. Barlow, The structure of turbulent nonpremixed flames revealed by Raman–Rayleigh–LIF measurements, *Prog. Energy Combust. Sci.* 22 (4) (1996) 307–362.
- [46] A.R. Masri, The University of Sydney. Available from: <<http://www.mech.eng.usyd.edu.au/research/energy/>>.
- [47] D.W. Meyer, P. Jenny, Conservative velocity interpolation for PDF methods, in: *Proc. Appl. Math. Mech.*, Dresden, Germany, 2004.