A computational slip boundary condition for near-wall turbulence modelling

Computers and Fluids 246 (2022) 105628

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

A computational slip boundary condition for near-wall turbulence modeling

Shiyu Lyu [a], Sergey Utyuzhnikov [a,b,*]

[a] Department of Mechanical, Aerospace and Civil Engineering, University of Manchester, Manchester M13 9PL, UK
[b] Moscow Institute of Physics and Technology, Dolgoprudny 141700, Russia

ARTICLE INFO

Keywords:
Near-wall turbulence modeling
RANS models
Non-overlapping domain decomposition
Interface boundary condition
Low-Reynolds-number model
Code openFOAM

ABSTRACT

Near-wall turbulence modeling represents one of challengers in the computational fluid dynamics. To tackle this problem, the near-wall non-overlapping domain decomposition (NDD) method proved to be very efficient. It has been successfully used with different Reynolds-averaged Navier–Stokes models. In NDD the computational domain is split into two non-overlapping sub-domains: an inner region near the wall, which is characterized by high gradients, and the outer region. To simplify the solution, in the inner region the thin-layer model can be used. In this case, NDD represents a trade-off between the accuracy and computational time. It has been demonstrated on numerous test cases that NDD is able to save up to one order of computational time while retaining practically high accuracy. A practical drawback of the algorithm is a need to split the computational domain into two regions. In the present paper, for the first time the NDD is realized implicitly. For this purpose, specific boundary conditions of Robin type are derived at the wall. Such boundary conditions reduce the gradients of the solution near the wall. The key property is that the solution can be obtained on a relatively coarse grid and then be recalculated in the inner region. In the approach it is guaranteed that the original outer and updated inner solutions are linked smoothly. The algorithm with the new boundary conditions can be easily implemented in standard codes. This is demonstrated with the code OpenFOAM. In addition, in the paper a more accurate approach to obtain the solution in the inner region is realized. The efficiency of the entire technique is demonstrated on test cases with modeling turbulent flows in a channel and asymmetric diffuser.

1. Introduction

The problem of modeling near-wall turbulent flows represents one of the major challenges in fluid dynamics for many years. In the numerical simulation, the major part of the computational time is taken by the resolution of a thin layer, which includes a laminar sublayer adjacent to the wall. The thickness of this layer is usually very small and no more than about 1% of the typical size of the flow in the normal direction [1]. The use of Large Eddy Simulation (LES) in industrial design is still prohibitively expensive if solid boundaries are present. In this case the problem becomes more complicated since it is problematic to extend the LES approach immediately to the wall. At the same time, the engineering approaches based on the Reynolds-averaged Navier–Stokes equations (RANS) closed with the use of high-Reynolds number (HRN) models often do not satisfy modern requirements to the accuracy. The problem is that in these models the effect of the wall is only counted via semi-empirical boundary conditions of Dirichlet type called the wall functions [2]. The use of low-Reynolds number (LRN) models improves the accuracy since such models resolve the structure of the entire turbulent boundary layer. However, this leads to a multiscale problem with high gradients of the solution in a very thin layer that always exists due to the no-slip boundary condition and damping effect of the wall. It is to be noted that its resolution has a significant influence on the prediction of the entire flow. As a result, an accurate enough resolution of the near-wall region requires a very fine mesh and takes up to 90% of the total computing time [3] and even more [1].

HRN models are still widely used in the industrial community. The drawbacks of this approach are well-known. The governing equations do not describe the transition region from laminar flow in the sublayer to the fully turbulent regime. Thus, HRN models do not resolve the sub-layer region. The numerical solution can be essentially mesh sensitive. The standard wall functions are semi-empirical and only applicable to very simple near-wall flows [1]. Attempts to extend their applicability lead to some free parameters to be tuned. It is to be noted that there are some advanced approaches such as the scalable wall functions [4], analytical and numerical wall functions [5], and adaptive wall functions [6]. In these models, the sub-layer region is partially resolved in the near-wall cell. However, the advanced wall functions

* Corresponding author at: Department of Mechanical, Aerospace and Civil Engineering, University of Manchester, Manchester M13 9PL, UK.
E-mail address: s.utyuzhnikov@manchester.ac.uk (S. Utyuzhnikov).

https://doi.org/10.1016/j.compfluid.2022.105628
Received 7 February 2022; Received in revised form 22 June 2022; Accepted 8 August 2022
Available online 12 August 2022

mentioned above only partially improve the entire approach which has a relatively small basis due to the assumptions behind the HRN models. In particular, the sensitivity of the solution to the size of the near-wall cell retains although it is essentially weaker than that for the conventional wall functions (see, e.g., [1,4,7]).

The near-wall domain decomposition (NDD) approach first proposed in [8,9] for HRN models can also be applied to LRN models [10] straightforward. The main idea of the approach is based on the transfer of the boundary conditions from the wall to the interface boundary between the inner and outer regions to be set. It turns out the boundary condition can be transferred exactly under some general conditions. If a locally one-dimensional model is used in the inner region, then any Dirichlet condition at the wall is exactly mapped onto a Robin boundary condition at the interface boundary [11]. As noted in [8], this boundary condition represents a slip boundary condition from the physical point of view. In a multidimensional case, it turns out that in a strict formulation the interface boundary condition (IBC) must be nonlocal [12]. The IBCs are always formulated in a mesh-independent form and do not contain free parameters. As was demonstrated on numerous test cases [13,14], the NDD allows the computational time to be reduced by one order of magnitude while retaining a good enough accuracy even for quite complex geometries [15,16]. Recently, the approach was successfully extended to essentially unsteady problems in [17]. It turns out that for essentially unsteady problems the IBCs should be nonlocal over time. It is to be noted that the original NDD approach implies the use of the thin boundary layer equations (TBLE) near the wall. For this reason, it is also called the approximate NDD (ANDD) approach in contrast to the exact NDD proposed in [16]. It is worth noting here that the TBLE model is also used in Wall-Modeled LES [18–20].

A practical drawback of the NDD approach is related to its relative algorithmic complexity. It is necessary to split the domain into two regions. Then, IBCs are set at the interface boundary. The solution should be obtained first in the outer region. Then, the solution in the inner region can be achieved with the boundary conditions determined from the solution in the outer region. This algorithm is essentially simplified in the present paper. The idea is the following. The IBC for the tangential velocity is transferred back to the wall in the framework of a simplified model with frozen coefficients. The new boundary condition at the wall is of Robin type and do not coincide with the original boundary condition. It represents a slip boundary condition at the wall which leads to smaller gradients of the solution near the wall. Next, the solution in the entire region can first be obtained on a relatively coarse mesh with the new boundary condition at the wall. Then, the solution in the inner region can be recalculated immediately with the TBLE model and original boundary condition at the wall and Robin-type IBCs at the interface boundary. It turns out that the solution in the inner and outer regions are always smoothly merged. In this approach there is no need in an explicit decomposition with the split of the computational domain into two sub-domains. The solution is obtained in the entire region and updated in the inner region. The last procedure can be realized in quadratures. In contrast to the standard ANDD, in the present paper the turbulent viscosity coefficient is calculated from the turbulent model used rather than approximated in advance. This allows us to enhance the accuracy of prediction in the inner region by the cost of some extra computational time.

The paper is organized as follows. In the next section, the original NDD method is described. Then, the new slip boundary condition is derived in Section 3. In this section the modified NDD algorithm is also introduced. It does not require implementation of domain decomposition explicitly. Next, Section 4 presents the derivation of the interface boundary conditions for the key variables, including the velocity, turbulent kinetic energy and its dissipation in the LRN $k-\varepsilon$ turbulence model, followed by the detailed implementation of the ANDD and slip boundary condition algorithms into the OpenFOAM. The efficiency of the new approach is demonstrated on a one-dimensional channel flow and two-dimensional diffuser flow in Section 5, where the numerical results of the test cases are discussed in detail. This is followed by the conclusion in Section 6.

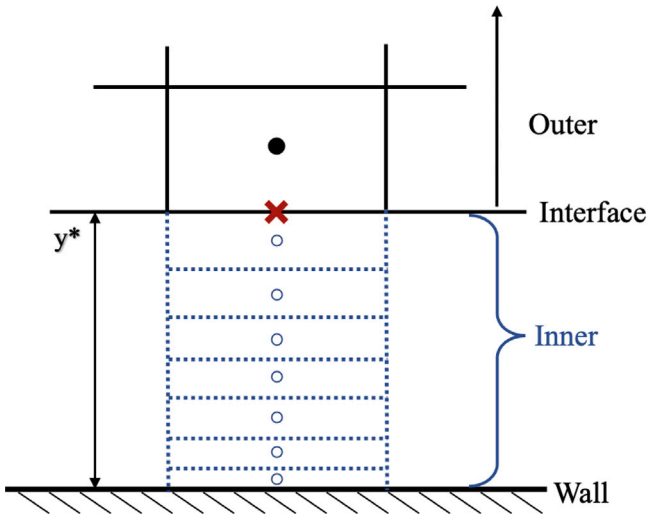


Fig. 1. Sketch of domain decomposition. Domain $0 < y < y^*$ corresponds to the inner region.

2. Near-wall domain decomposition

Consider the key idea of NDD approach in application to a Dirichlet boundary value problem (BVP) for a 1D equation defined in interval $\Omega := [0,\ y_e]$:

$$\mathcal{L}_y(U) = R_U, \tag{1}$$

$$U(0) = U_0, \tag{2}$$

$$U(y_e) = U_1,$$

where $\mathcal{L}_y$ is a differential operator of second order: $\mathcal{L}_y := \frac{\partial}{\partial y}\mu\frac{\partial}{\partial y}$.

Let split interval Ω into two subintervals $\Omega^- := [0,\ y^*]$ and $\Omega^+ := [y^*,\ y_e]$ with the interface at $y = y^*$ as shown in Fig. 1. The boundary condition from the wall ($y = 0$) can be exactly transferred to the interface boundary $y = y^*$ via the Dirichlet-to-Neumann map [10]:

$$\frac{dU}{dy} = \Psi(U, R_U). \tag{3}$$

Here, Ψ is a pseudo-differential operator which is nonlocal. It is worth noting that this statement is valid even in a multi-dimensional formulation. In 1D case, IBC (3) can be presented explicitly in the form of a Robin boundary condition [11,16]:

$$U(y^*) = f_1\frac{dU}{dy}(y^*) + f_2 + U_0, \tag{4}$$

where $f_1 = \int_0^{y^*}\frac{\mu^*}{\mu}dy$, $f_2 = \frac{1}{\mu^*}(I_{S_2} - f_1 I_{S_1})$, $I_{S1} = \int_0^{y^*} R_U dy$, $I_{S2} = \int_0^{y^*}\frac{\mu^*}{\mu}\int_0^{y'} R_U dy dy'$. Here and further below, the asterisk corresponds to the value of a function at $y = y^*$: $\mu^* = \mu(y^*)$.

IBC (4) fully replaces the solution of the problem in the inner region Ω^- for the outer region Ω^+. This means the solution of BVP (1), (2) must satisfy Eq. (4). In turn, one can prove that condition (4) is unique. In the linear case, Eq. (4) is totally independent from the solution in the outer region. This means the statements above are valid even if the operator L_y is both multidimensional and nonlinear in Ω^+. In particular, all these statements can be applied to RANS equations under the assumption of known coefficient μ in Ω^-.

It is to be noted that it is obviously impossible to obtain the solution in the outer region, Ω^+, without the solution in the inner region, Ω^-. However, one can transfer the boundary condition from the wall ($y = 0$) to the interface boundary without knowledge of the solution in the outer region. It is worth noting that the solution in Ω^- with boundary condition (4) is not unique since this condition is exactly derived from

the governing equation and boundary condition at the opposite end. It is clear that the solution of BVP in the inner region becomes unique once the external boundary condition is based on the solution of BVP in the outer region.

Finally, one can prove that the composite solution must be smooth. This means both solutions, the inner and outer, coincide at the interface boundary up to the first derivative. The proof immediately follows from the fact that the IBC is fully equivalent to the governing equation in the inner region and the boundary condition at the wall. In addition, the solution in the outer region is unique.

In application to the RANS equations, the TBLE are considered in the inner region. As such, the turbulent viscosity coefficient depends on the solution in Ω^+, so does the IBC. In the formulation described above, the IBCs for BVP in the outer region can be formulated in a universal form (4) for all variables but the normal to the wall velocity. With the latter variable the problem is that for it there is no leading equation of the second order. In [16], a Robin boundary condition is derived for the normal velocity from the continuity equation and no-flux boundary condition. In contrast to (4), this boundary condition is local.

Thus, we arrive at the following procedure to realize the NDD. First, the solution in the outer region Ω^+ with boundary condition (4) at the left-hand end should be obtained. Then, a BVP should be solved in the inner region with the Dirichlet (or Neumann/ Robin) boundary condition at the right-hand end. This boundary condition can be immediately obtained from the solution of BVP in the outer region. It is to be noted that the solution in the inner region can be easily obtained even in quadratures since this problem is locally one-dimensional. Moreover, if only a flux to the wall is needed, then it can be obtained immediately from the governing equation:

$$\tau_w = \mu^* \frac{dU}{dy}(y^*) - \int_0^{y^*} R_U \, dy, \tag{5}$$

where τ_w is the friction at the wall.

In the governing equations $\mu = \mu_l + \mu_t$ is the efficient viscosity coefficient which is the sum of the laminar and turbulent coefficients, respectively. Thereby, it depends on the solution. To simplify the use of boundary condition (4), one can implement available approximations for the turbulent viscosity μ_t near the wall such as the Van-Driest profile [21]. Some examples of approximation for μ_t are provided in [13]. In all of them $\mu_t = \mu_t(\tau_w)$. Thus, the IBC retains to be nonlinear, although the nonlinear dependence, which requires an iterative procedure, is weak.

As a result, the entire algorithm reads:

1^0. Select an approximation for the near-wall turbulent viscosity coefficient μ_t.

2^0. Select an initial approximation for τ_w.

3^0. Calculate IBC (4).

4^0. Solve BVP in the outer region with IBC (4).

5^0. Update τ_w via (5).

6^0. Update the near-wall turbulent viscosity profile.

7^0. Repeat the procedure from step 3.

De facto, the problem is numerically solved in the outer region since in the inner region the solution is then obtained straightforward:

$$U(y) = \int_0^y \frac{\mu^*}{\mu(\eta)} d\eta \frac{dU}{dy}(y^*) - \int_0^y \frac{\int_\eta^{y^*} R_U \, d\eta'}{\mu(\eta)} d\eta + U_0. \tag{6}$$

Thus, in the inner region a special mesh generation is not required since the quadratures can be calculated on a local sub-grid.

A drawback of this algorithm is related to some complexity with its implementation in existing computer codes. It is necessary to split the computational domain into two non-overlapping sub-domains. Then, the problem is effectively solved only in the outer region.

The entire algorithm can be essentially simplified with the development of effective boundary conditions at the wall such that the problem can be solved in the outer computational domain on a relatively coarse mesh and then recalculated only in the inner region. For that we transfer the IBC for the tangential velocity back to the wall with the use of a truncated TBLE model with frozen coefficients. The objective of the new boundary conditions is the following. In the outer region the solution of BVP obtained in the entire region with these boundary conditions should coincide with the solution of the original problem in the framework of NDD. Since the IBCs are derived with the use of LRN model, while the new boundary conditions at the wall are based on the model with frozen coefficients, the original and new boundary conditions at the wall do not coincide. Inevitably, we should obtain a slip boundary condition. In addition, since the slip boundary condition can essentially reduce the gradients of the solution, a relatively coarse mesh might be used at that stage. It is worth noting here that in [22,23] the dynamic slip boundary conditions are derived for LES but they are obtained on different principles, from the LES filtering procedure.

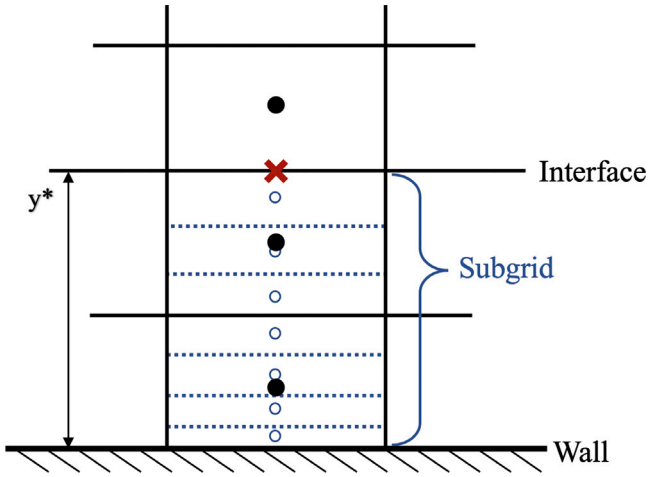


Fig. 2. Sketch of coupling in the near-wall region: black nodes and solid lines correspond to main grid; blue circles and dot lines, fine sub-grid for recalculation; red cross, information exchange point at the interface.

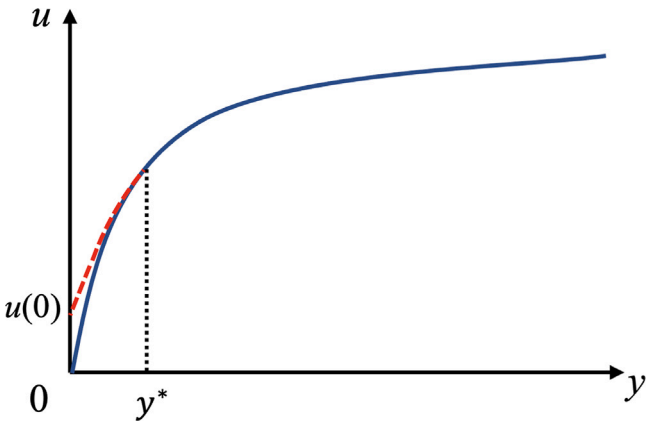


Fig. 3. Sketch of velocity profile obtained in two stages. The dashed line corresponds to the profile corrected at the first stage.

3. Slip boundary condition and implicit NDD

To derive the new boundary condition at the wall, let fix the coefficient μ and the right-hand side R_U at y^* in (1). Such a model is further called the truncated TBLE model.

Next, for the solution $\widetilde{U}$ with the slip boundary condition in the inner region we have

$$\widetilde{U}(y) \approx \widetilde{U}(0) + y\widetilde{U}'(0) + \frac{y^2}{2}\widetilde{U}''(0).$$

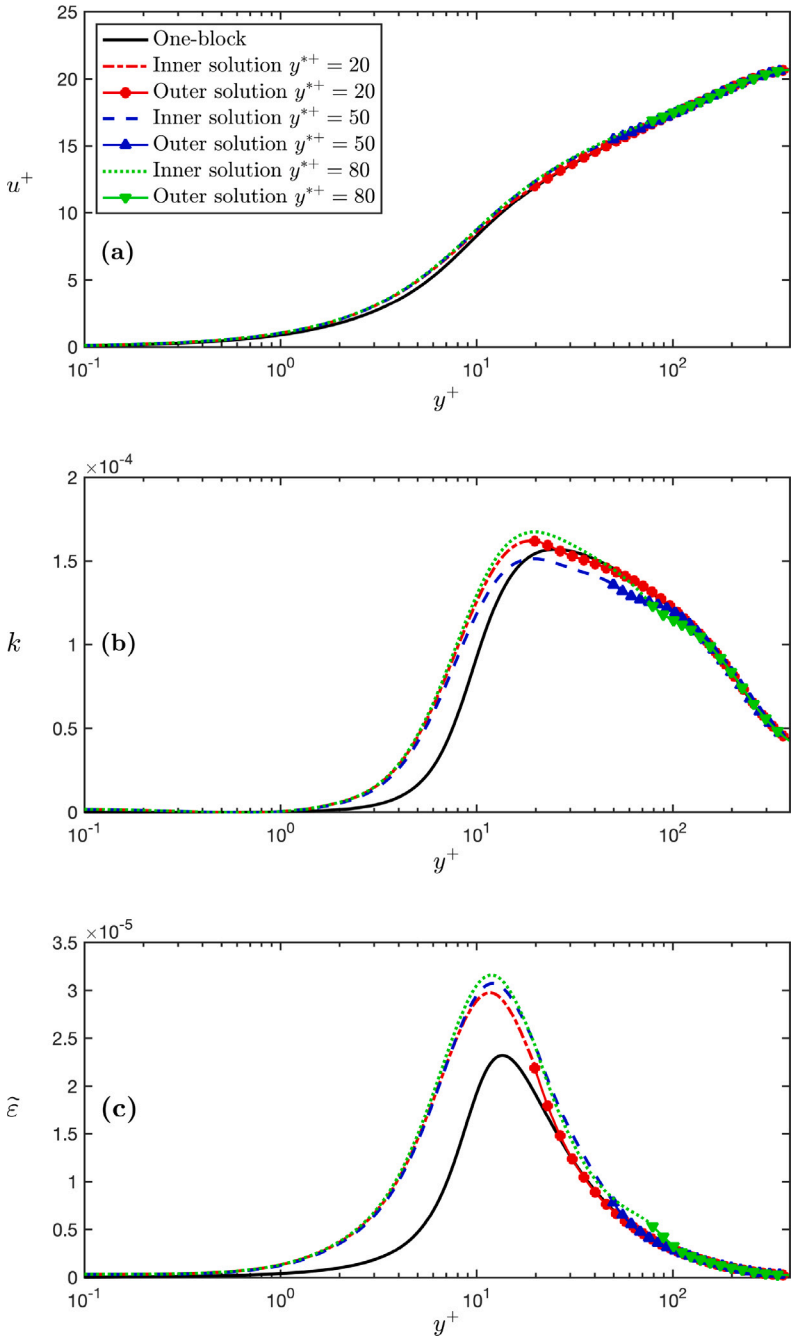


Fig. 4. Channel flow, $Re_\tau = 395$. Comparison of explicit NDD (ANDD) results ($y^{*+} = 20$ in red line, 50 in blue line and 80 in green line) with one-block LRN solution.

Then

$$\widetilde{U}(y^*) = \widetilde{U}(0) + y^*\widetilde{U}'(0) + \frac{y^{*2}R_U^*}{2\mu(y^*)}.$$

Here, the second derivative is immediately obtained from the simplified governing equation where $R_U^* = R_U(y^*)$.

In addition,

$$\widetilde{U}'(y^*) = \widetilde{U}'(0) + \frac{y^*R_U^*}{\mu^*}. \tag{7}$$

On the other hand, we require that the new solution must satisfy IBC (4):

$$\widetilde{U}(y^*) = f_1\widetilde{U}'(y^*) + \widetilde{f_2}, \tag{8}$$

where $\widetilde{f_2} = f_2 + U_0$.

Then, immediately we arrive at the boundary condition of Robin type at the wall for variable $\widetilde{U}$:

$$\widetilde{U}(0) = f_{w1}\widetilde{U}'(0) + f_{w2}, \tag{9}$$

where $f_{w1} = f_1 - y^*$, $f_{w2} = \widetilde{f_2} + \frac{y^*R_U^*}{\mu^*}(f_1 - \frac{y^*}{2})$.

It is easy to prove that $f_{w1} \geq 0$ since μ_t is a monotonic function near the wall. Hence, boundary condition (9) is always well-posed. In addition, as can be seen, if y^* tends to zero, then boundary condition (9) transforms to the no-slip boundary condition. Moreover, it immediately becomes the no-slip boundary condition as long as y^* is situated in the laminar sub-layer. Indeed, in this case $f_1 = y^*$ and $f_2 = -\frac{R_U^*}{\mu*}\frac{y^{*2}}{2}$ as such $f_{w1} = 0$ and $f_{w2} = U_0$.

From the derivation it follows that IBC (8) (and (4)) is fully equivalent to the simplified governing equation in the inner region and

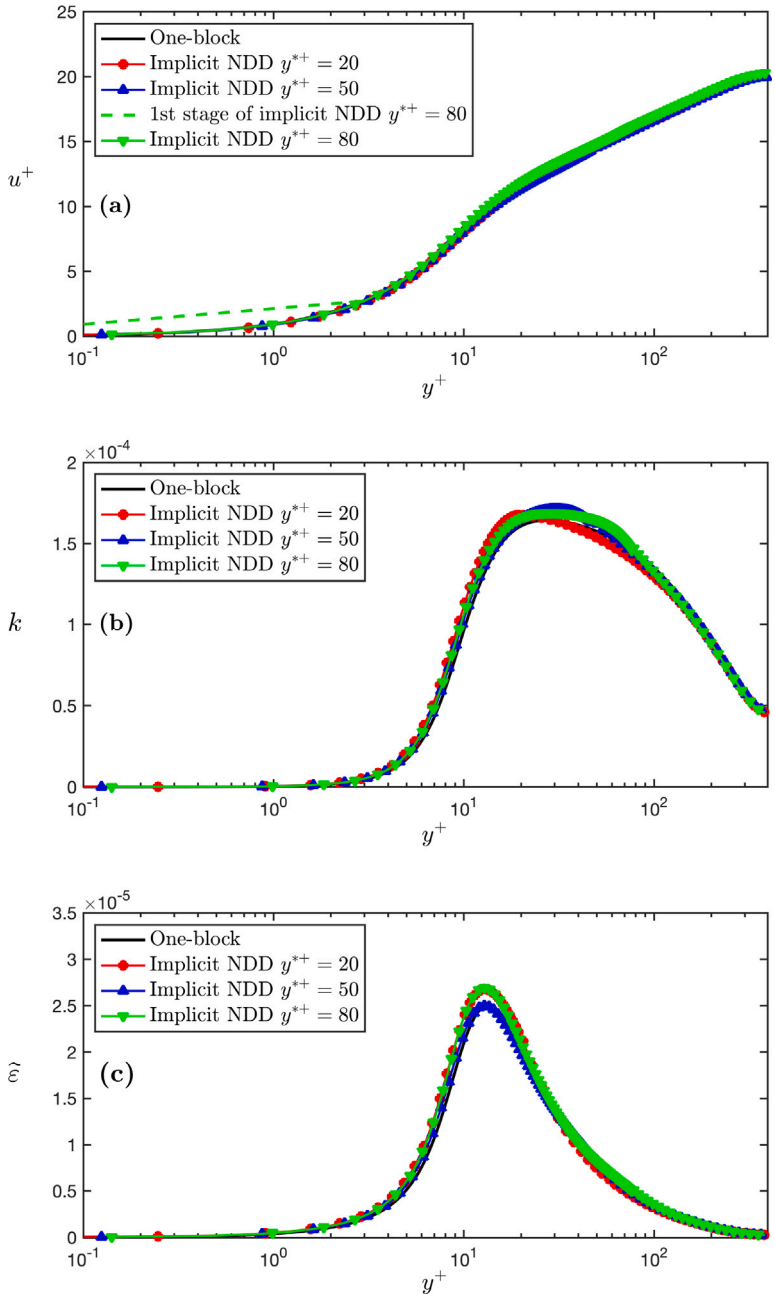


Fig. 5. Channel flow, $Re_\tau = 395$. Comparison of implicit NDD results ($y^{*+} = 20$ in red line, 50 in blue line and 80 in green line) with one-block LRN solution. The dashed line represents the velocity profile after the 1st stage with $y^{*+} = 80$.

slip boundary condition (9). Thus, at the interface boundary the inner solutions obtained with the original and truncated TBLE models must coincide up to the first derivative.

In this way, the computational algorithm becomes straightforward. First, the solution is obtained in the entire region with the slip boundary condition. Then, in the inner region the solution is updated according to (6). If we are interested only in a flux to the wall, then only the flux is recalculated according to (5).

As can be seen, with the new approach the NDD is realized only implicitly. There is no need in the split of the computational domain explicitly. The problem can be solved with a LRN model with the slip boundary condition at the wall. This solution can be obtained on a relatively coarse grid since it does not have high gradients. Then, the solution in the inner region is corrected. It is smoothly adjusted to the solution in the outer region. As the original NDD, the new approach represents a trade-off between the accuracy and computational time depending on interface boundary y^*. In further consideration, the NDD approach with the slip boundary condition (9) is called the implicit NDD.

As an additional modification, in contrast to the conventional ANDD, in the implicit NDD we do not specify the turbulent viscosity coefficient in the inner region. Instead, it is determined from the RANS model used. This issue, is discussed in more detail in the next section. It is to be noted that the suggested latter modification is not inherent to the implicit algorithm of NDD based on the slip boundary condition and can be dropped. However, we presume that it is potentially able to improve the accuracy of the solution by the cost of some extra computational time.

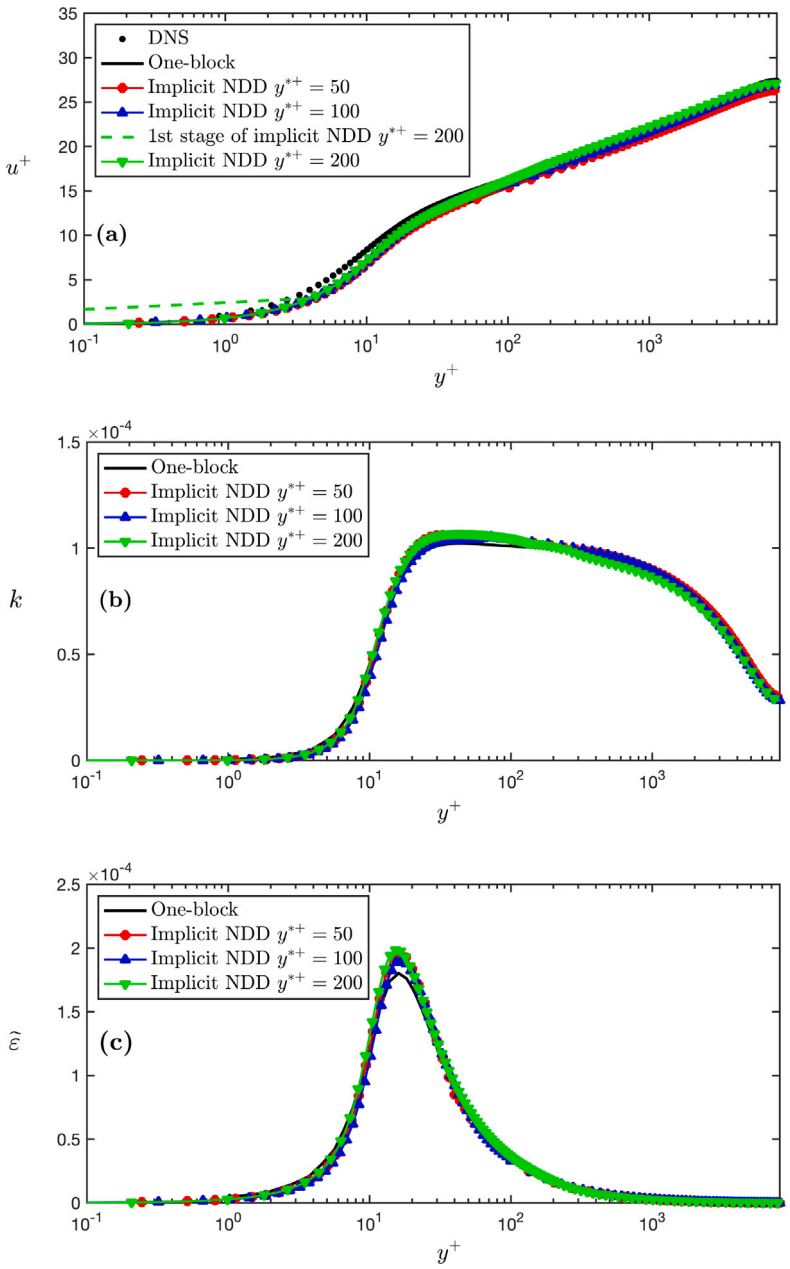


Fig. 6. Channel flow, $Re_\tau = 8000$. Comparison of implicit NDD results ($y^{*+} = 50$ in red line, 100 in blue line and 200 in green line) with one-block LRN solution. The dashed line represents the velocity profile after the 1st stage with $y^{*+} = 200$. The dotted line corresponds to DNS data [24].

4. Application to LRN turbulence model

The LRN $k-\varepsilon$ model of Launder–Sharma [25] is chosen to close the governing equations. The TBLE equations utilized in the inner region can be written as follows:

$$\begin{aligned}
&\frac{\partial}{\partial y}\left[(\nu+\nu_t)\frac{\partial u}{\partial y}\right] = \frac{1}{\rho}\frac{\partial p}{\partial x},\\
&\frac{\partial}{\partial y}\left[\left(\nu+\frac{\nu_t}{\sigma_k}\right)\frac{\partial k}{\partial y}\right] = \varepsilon - \nu_t\left(\frac{\partial u}{\partial y}\right)^2,\\
&\frac{\partial}{\partial y}\left[\left(\nu+\frac{\nu_t}{\sigma_\varepsilon}\right)\frac{\partial \tilde{\varepsilon}}{\partial y}\right] = C_{\varepsilon 2}\overline{f_2}\frac{\tilde{\varepsilon}^2}{k} - C_{\varepsilon 1}\overline{f_1}\frac{\tilde{\varepsilon}}{k}\nu_t\left(\frac{\partial u}{\partial y}\right)^2 - E,
\end{aligned} \tag{10}$$

where

$$\nu_t = C_\mu f_\mu k^2/\tilde{\varepsilon}, \quad C_\mu = 0.09, \quad f_\mu = e^{\frac{-3.4}{(1+Re_t/50)^2}}, \quad \varepsilon = \varepsilon_0 + \tilde{\varepsilon},$$

$$\varepsilon_0 = 2\nu\left(\frac{\partial\sqrt{k}}{\partial y}\right)^2, \quad \overline{f_1} = 1, \quad \overline{f_2} = 1 - 0.3e^{-Re_t^2}, \quad Re_t = \frac{k^2}{\tilde{\varepsilon}\nu},$$

$$C_{\varepsilon 1} = 1.44, \quad C_{\varepsilon 2} = 1.92, \quad \sigma_k = 1, \quad \sigma_\varepsilon = 1.3, \quad E = -2\nu\nu_t\left(\frac{\partial^2 u}{\partial y^2}\right)^2.$$

Here, x is the coordinate along the wall; p, the pressure; k, the turbulent kinetic energy, ε, its dissipation; ν, the dynamic viscosity coefficient; ν_t, the eddy viscosity coefficient.

As can be seen, the convective terms are not included in the TBLE equations. Their possible contribution depends on y^* and briefly discussed in the next subsection.

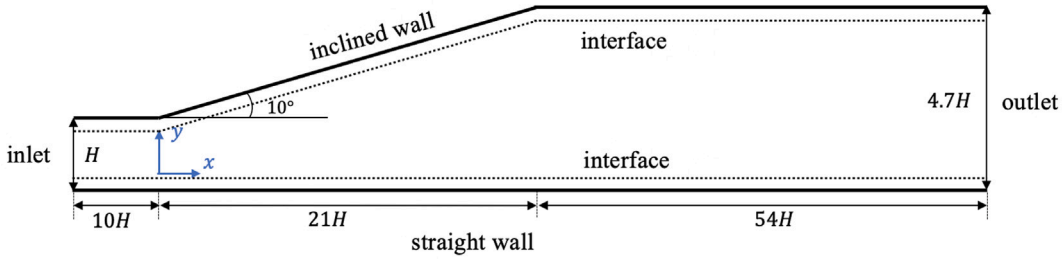


Fig. 7. Geometry of asymmetric diffuser.

4.1. Contribution of convective terms to IBC

To estimate the contribution of the convective terms, consider the momentum equation

$$\frac{\partial}{\partial y}\left[(\nu+\nu_t)\frac{\partial u}{\partial y}\right]=\frac{1}{\rho}\frac{\partial(p+\rho u^2)}{\partial x}+\frac{1}{\rho}\frac{\partial \rho u v}{\partial y}.$$

In this case, IBC (4) reads

$$u(y^*)=f_1\frac{du}{dy}(y^*)+f_2',$$

where $f_2'=-(u^*v^*/v^*)f_1+\int_0^{y^*}uv/\nu dy-\int_0^{y^*}\frac{1}{\nu}\int_y^{y^*}(p+\rho u^2)_x dy' dy$, v is the normal to the wall component of the velocity.

Next, we introduce the following approximation for the velocity $u\approx(y/y^*)u^*$ and $v\approx(y/y^*)^2v^*$. Then, the second term in f_2' is estimated as

$$\int_0^{y^*}\frac{uv}{\nu}dy\approx\frac{u^*v^*}{v^*}\int_0^{y^*}\frac{v^*}{\nu}(\frac{y}{y^*})^3dy<\frac{u^*v^*}{4\nu^*}f_1.$$

Thus, the contribution of the convective terms becomes essential if $\rho^*u^{*2}\sim p$ and $u^*v^*\sim u_\tau^2$, where $u_\tau=\sqrt{\tau_w/\rho}$. The latter estimate is reduced to $u^{+*}v^{+*}\sim 1$, where $u^+=u/u_\tau$ and $v^+=v/u_\tau$.

4.2. IBCs in ANDD

The right-hand side term R_U in compliance with (1) can be expressed for u, k and $\tilde{\varepsilon}$ equations as

$$\begin{aligned}R_u&=\frac{1}{\rho}\frac{\partial p}{\partial x},\\ R_k&=\varepsilon-\nu_t\left(\frac{\partial u}{\partial y}\right)^2,\\ R_{\tilde{\varepsilon}}&=C_{\varepsilon 2}f_2\frac{\tilde{\varepsilon}^2}{k}-C_{\varepsilon 1}f_1\frac{\tilde{\varepsilon}}{k}\nu_t\left(\frac{\partial u}{\partial y}\right)^2-E.\end{aligned}\tag{11}$$

The IBCs for variables u, k, and $\tilde{\varepsilon}$ are used in Robin-type form (4). Term R_U is essential for the calculation of coefficient f_2. For the tangential velocity u, the stream-wise pressure gradient is assumed to be constant and evaluated immediately at the interface position y^* from the outer region, such that $R_u=dp/dx(y^*)/\rho$.

In a finite volume code, IBC (4) can be approximated in the outer region via the value of U at the interface boundary U^* and at the center of the cell adjoint to the interface boundary U_c. Then, we arrive at a discrete IBC:

$$U^*=\frac{f_1U_c+\delta\widetilde{f_2}}{\delta+f_1}.\tag{12}$$

Here, δ is the distance between the center of the cell and interface boundary.

In contrast to the tangential velocity, the IBC for the normal component of velocity can be derived from the no-flux boundary condition and continuity equation using the Taylor expansion [16]. It is also formulated as a Robin-type boundary condition:

$$v^*=\frac{y^*}{3}\frac{dv}{dy}(y^*).\tag{13}$$

In turn, in the discrete form it reads

$$v^*=\frac{y^*}{3\delta+y^*}v_c^*,\tag{14}$$

where v^* and v_c are the values of normal velocity at the interface boundary and interface-adjacent cell center in the outer region, respectively.

4.3. Determination of turbulent viscosity coefficient in the inner region

The only parameter required to implement IBCs (4) is the turbulent viscosity profile in the inner region. In the conventional ANDD approach, it can be specified in one way or another from existing near-wall approximations. For example, the non-linear profile of Duprat et al. [26] can be used, which takes into account the stream-wise pressure gradient:

$$\nu_t(\xi)=\nu\kappa\xi^+\left[\alpha+\xi^+(1-\alpha)^{3/2}\right]^\beta\left[1-\exp\left(\frac{-\xi^+}{1+A\alpha^3}\right)\right]^2,\tag{15}$$

where $\kappa=0.41$, $A=17$, $\beta=0.78$, $\alpha=u_\tau^2/u_{\tau p}^2$, $\xi^+=yu_{\tau p}/\nu$, $u_{\tau p}=\sqrt{u_\tau^2+u_p^2}$, $u_\tau=\sqrt{|\tau_w|/\rho}$, $u_p=|\nu\partial p/\partial x/\rho|$.

The turbulent viscosity profile (15) depends on the surface friction and stream-wise pressure gradient. It allows us to predict flows with a boundary layer separation [17]. In the algorithm, τ_w can be updated each iteration by (5). The stream-wise pressure gradient $\partial p/\partial x$ can be taken at the interface boundary from the outer region. Discussion of other possible approximations in application to NDD can be found in [14,27].

In contrast to this approach, in the implicit NDD we determine the turbulent viscosity from TBLE (10) for k and $\tilde{\varepsilon}$ with Dirichlet boundary conditions at the interface boundary and original homogeneous boundary conditions for k and $\tilde{\varepsilon}$ at the wall.

4.4. Implementation of implicit NDD

As described in Section 3, the implicit NDD effectively presumes two stages. At the first stage, the solution is obtained on a relatively coarse grid in the entire region. Then, at the next stage, the solution in the inner region is recalculated on a sub-grid depicted in Fig. 2. It is to be noted that the sub-grids for any fixed coordinate x along the wall are entirely local and can be not related to each other.

At the first stage, the slip boundary condition (9) is implied to the tangential velocity at the wall boundary. The no-flux boundary condition is set for the normal velocity component. At the second stage, for the sub-domain recalculation, the Dirichlet boundary condition is used at the interface boundary and no-slip boundary condition is used at the wall. The calculations are used on a local fine sub-grid. Computing the tangential velocity profile is illustrated by the sketch presented in Fig. 3 where the dashed line corresponds to the preliminary velocity profile in the inner region obtained with the slip boundary condition.

The implicit NDD algorithm can be summarized as follows:

1^0. Initialize the flow fields for both the main (coarse) grid and sub-grid.

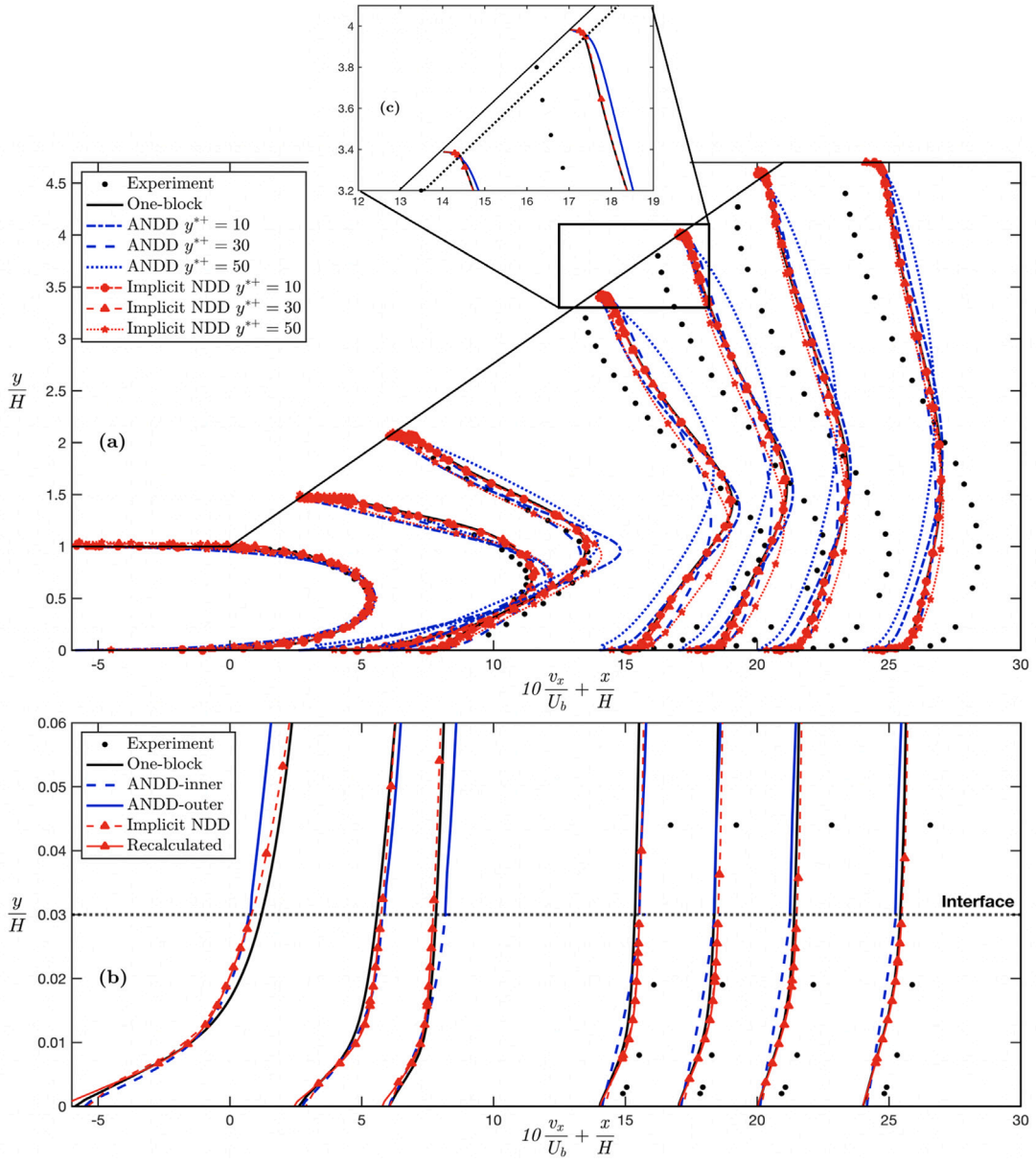


Fig. 8. Comparison of mean stream-wise velocities between the results from experiment [28], one-block solution, explicit and implicit NDD. Subfigure (a) shows profiles of v_x at different sampling positions in asymmetric diffuser; plot (b), enlargement of profiles within near-wall region up to $y/H = 0.06$; plot (c), enlargement of the box region in subfigure (a).

2^0. Calculate the coefficients f_{w1} and f_{w2} based on the sub-grid solution to impose the slip boundary condition at the wall.

3^0. Solve the governing equations on the main grid with the slip boundary condition.

4^0. Recalculate the solution in the inner region on the sub-grid region with the interface value obtained on the main grid.

5^0. Recalculate the turbulent viscosity based on the corrected velocity profile.

6^0. Repeat the procedure from step 2.

The entire algorithm can be easily implemented in the existing codes. For this purpose, the slip boundary conditions should be set at the wall followed by the re-calculation procedure in the inner region that is entirely local for any fixed coordinate x. The described approach has been realized in the OpenFOAM [29], an advanced open-source toolbox based on the finite-volume method. With the help of multi-region functionality in the OpenFOAM, the separate governing equations, incorporated with the corresponding IBCs, are effectively solved in different regions.

5. Test cases

Next, the proposed algorithm is tested on the one-dimensional channel flow and two-dimensional flow in an asymmetric diffuser.

In each simulation, mesh-independence is firstly reached on a grid for the LRN one-block solution to verify the numerical results. In the outer region, the grid retains the same for both the explicit (ANDD) and implicit NDD. In turn, in the inner region a very coarse mesh is used at the first stage of the implicit NDD. Then, at the second stage, a fine local mesh is used to recalculate the solution in the inner region according to (6). The same mesh is used for the explicit NDD.

Finally, we compare the results obtained with the use of thee approaches: one-block solution on a fine mesh, explicit and implicit NDD.

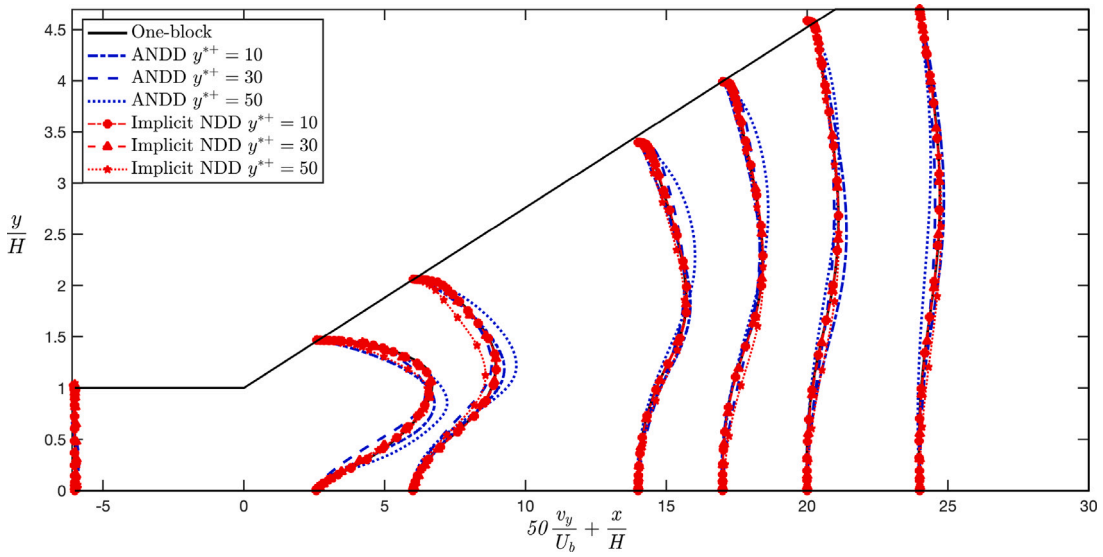


Fig. 9. Comparison of velocity component v_y between the results from one-block solution, explicit and implicit NDD for different y^{*+}.

5.1. Plane channel flow

A fully developed channel flow with the friction Reynolds number $Re_\tau = 395$ is considered as the first test case. Here, Re_τ is defined in terms of the friction velocity u_τ and the half of the channel height h: $Re_\tau = u_\tau h/\nu$.

The periodic boundary conditions are used in the OpenFOAM for the inlet and outlet boundaries to exactly map the variations at the outlet back to the inlet, eventually reaching a fully-developed flow. The front and back boundary conditions are set as empty that means no solution is required in that direction for the one- or two-dimensional case. The pressure gradient along the stream-wise direction is fixed as a constant driving force.

Different cases are simulated with the interface located at $y^*/h = 0.05$, 0.126 and 0.20, which correspond to the y^{*+} varied from 20 to 80. As portrayed in Fig. 4, the predicted profile of u^+ (defined as u/u_τ) with the explicit ANDD method is agreed well enough with the one-block LRN solution for different y^{*+}. Both k and $\tilde{\varepsilon}$ profiles remain continuous at the interface positions that is a property of the Robin boundary condition. However, there is a notable discrepancy for the turbulent kinetic energy k when compared with the one-block solution as shown in Fig. 4(b). In addition, in subfigure (c), ANDD method largely overpredicts $\tilde{\varepsilon}$ around the peak area, especially at $y^{*+} = 80$. This is mainly due to the error introduced by the approximated turbulent viscosity in the inner region.

As can be seen from Fig. 5(a), with the implicit NDD method the dimensionless velocity u^+ for each interface position almost perfectly matches with the benchmark solution. The velocity profile after the first stage of implicit NDD is also shown for $y^+ = 80$. It is noticeably shifted up because of the slip boundary condition implied. Then, the solution is refined in the inner region at the second stage. Besides, Fig. 5(b) and 5(c) clearly demonstrate that k and $\tilde{\varepsilon}$ profiles with the inner-layer correction are much closer to the benchmark solutions when compared with the corresponding ANDD results. The disagreement that appeared in the ANDD is effectively resolved, due to the fact that the turbulent viscosity in the inner region is reconstructed immediately from the original turbulence model instead of an analytical approximation. The overall near-wall solution is determined by the original model reduced to the TBLE. Thus, the facilitation of the original governing equations in the inner region does not cause any essential error. The same tendency retains for higher Reynolds numbers. Fig. 6 demonstrates the results for the test case from [24] with $Re = 8000$. These results correspond to greater values of $y^{*+} := 50, 100, 200$. However, the obtained solutions are also low sensitive to the variation of y^* and well coincide with the benchmark one-block solution. Similar to the previous example, the velocity profile is essentially corrected in the inner region after the first stage as can be seen in Fig. 6(a). It is worth noting that the combined velocity profile coincides well enough with the DNS data from [24].

5.2. Two-dimensional asymmetric diffuser

Now, consider the 2D asymmetric diffuser test case. The geometry of the 2-D asymmetric diffuser, according the layout in [28], is shown in Fig. 7. The inlet is located at $x/H = -10$ and the outlet, at $x/H = 75$. The corners of the diffuser situated at $x/H = 0$ and $x/H = 21$ are smoothed with the radius of curvature equal to $9.7H$. The diffuser angle is 10°. The inlet boundary condition is imposed as the fully developed channel flow corresponding to Reynolds number $Re = 1.8 \times 10^4$ which is based on the bulk velocity U_b and channel height $Re = U_b H/\nu$.

In Fig. 8, the predictions of the stream-wise velocity component v_x with the explicit and implicit NDD are compared with the one-block LRN results and experimental data [28] at $x/H = -5.87$, 2.59, 5.98, 13.56, 16.93 and 20.32. Different simulations with the explicit and implicit NDD are carried out at the interface positions $y^* = 0.01$, 0.03 and 0.05, which correspond to $y^{*+} = 10$, 30 and 50, respectively. As can be seen, there is a growing discrepancy of the velocity profiles between ANDD predictions and one-block LRN solutions as y^{*+} increases that is caused by the approximate viscosity profile and TBLE model used in the inner region. The velocity profiles obtained with the implicit NDD are in much better agreement with the benchmark LRN solutions. This is basically because with the implicit NDD the turbulent viscosity is determined from the TBLE model. It can also be seen that with the implicit NDD the velocity profiles with $y^{*+} = 50$ become noticeably different than the profiles with $y^{*+} = 10$ and 30. This is caused by the error of the TBLE model that is more essential in the two-dimensional case considered.

The near-wall regions nearby the straight and inclined walls with the interface located at $y^*/H = 0.03$ are enlarged in Fig. 8(b) and Fig. 8(c), respectively. The velocity curves located at $x/H = 14$ and 17 are chosen to represent the velocity profiles near the inclined wall. The results with the explicit and implicit NDD approaches are demonstrated in the outer and inner regions. Subfigures (b) and (c) clearly demonstrate that the velocity profiles obtained with the implicit NDD provide notably more accurate results than the ANDD prediction.

The velocity component along the y-axis v_y is plotted in Fig. 9 at the same sampling positions as v_x. The profiles are scaled up five times

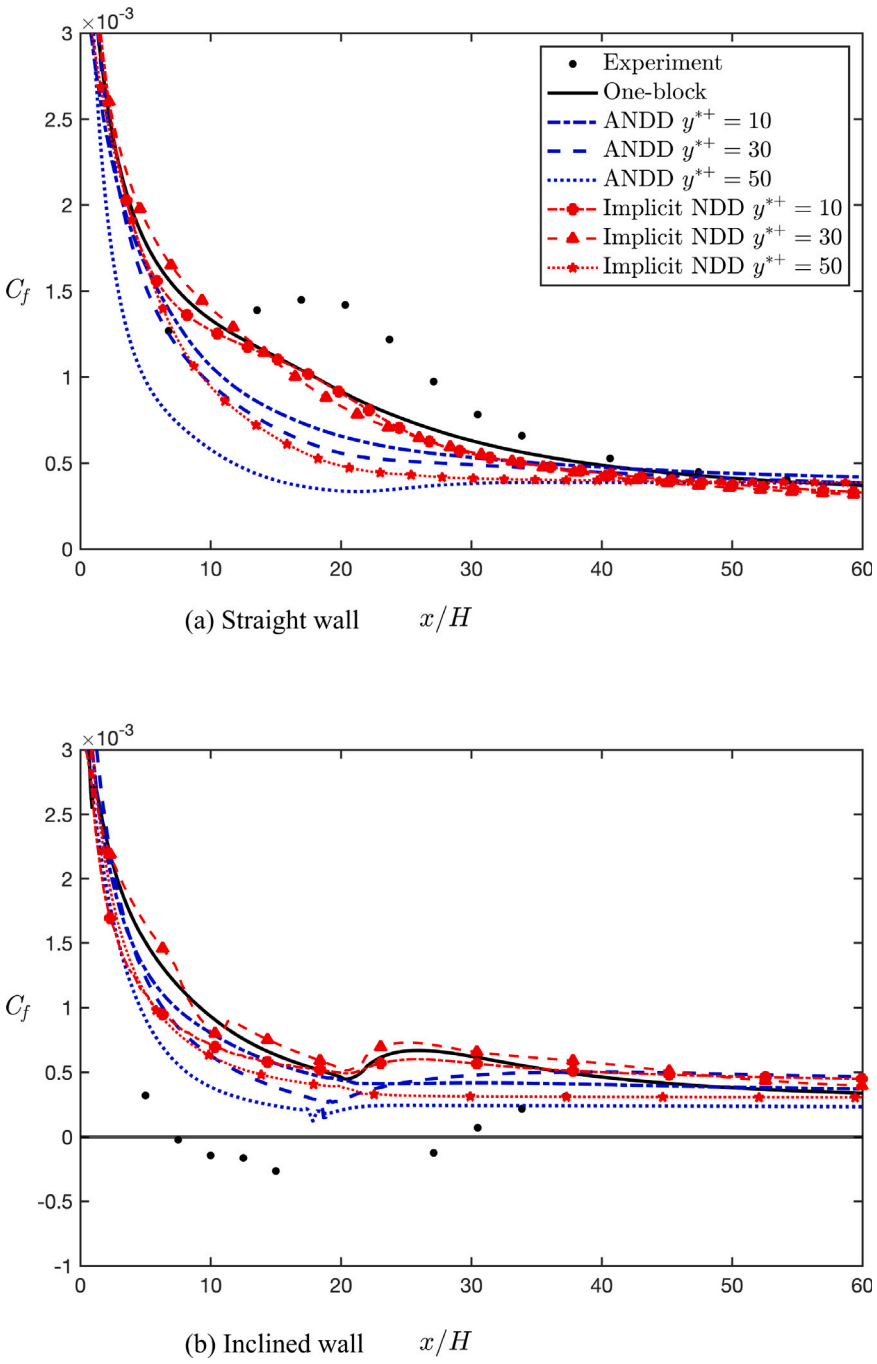
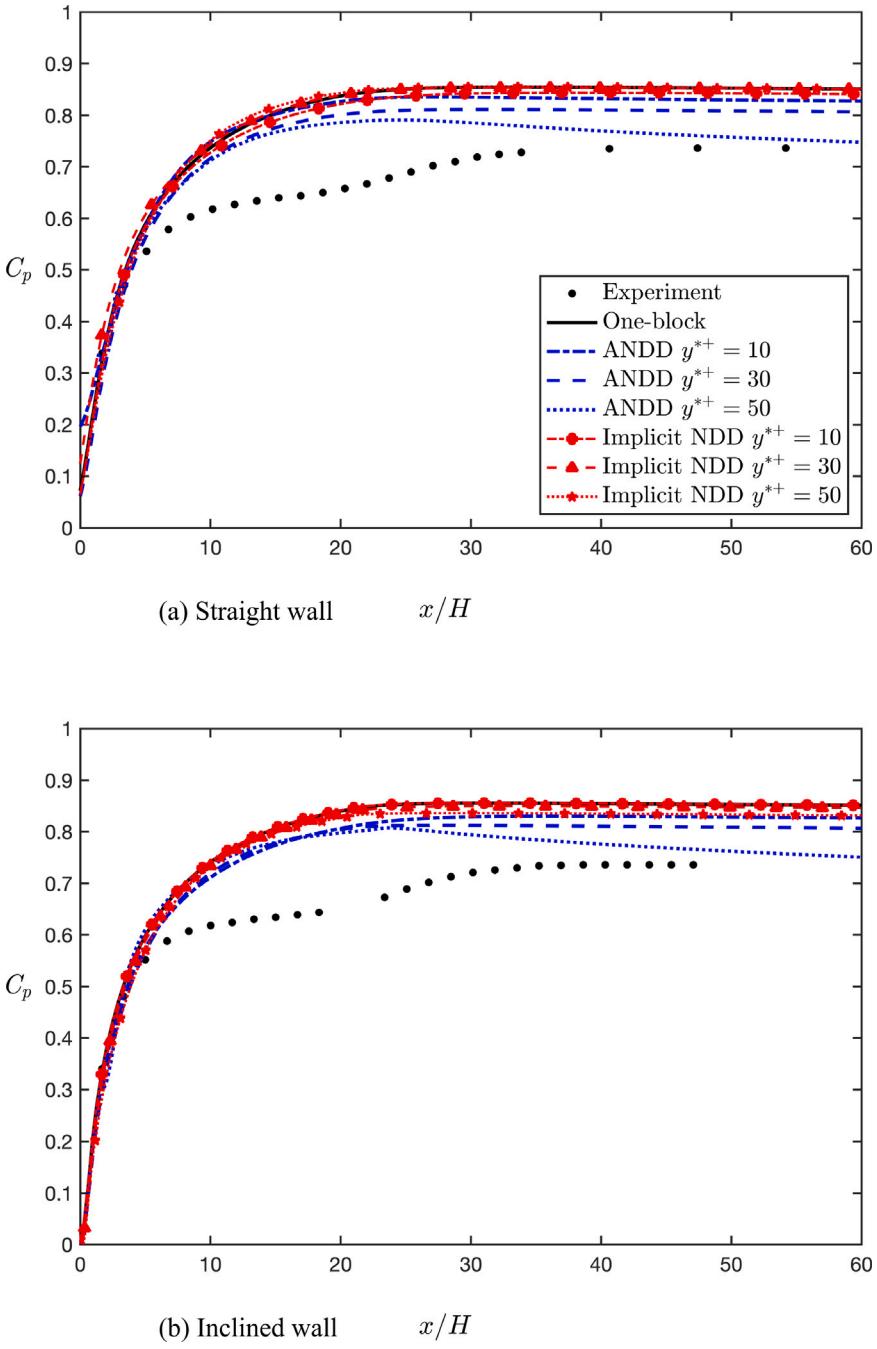


Fig. 10. Comparison of friction coefficients C_f along the straight and inclined walls of the diffuser, obtained from experiment [28], one-block solution, explicit and implicit NDD.

larger than the steam-wise velocity in the previous figure. It can be seen that the velocity v_y obtained with the implicit NDD is less sensitive to y^* and better matches with the one-block solution compared to the ANDD results.

The skin friction coefficient $C_f = 2\tau_w/(\rho U_b^2)$ is shown along the straight wall in Fig. 10(a) and along the inclined wall in Fig. 10(b). C_f curves obtained with ANDD show progressively worse agreement along both the straight and inclined walls as y^* increased. The plots of C_f demonstrate that the results obtained with the implicit NDD are less sensitive to the interface locations y^* than the solutions obtained with ANDD. Only the case of $y^{*+} = 50$ deviates noticeably from the LRN one-block solution. The surface friction curves obtained with the implicit NDD are well improved in comparison with the predictions based on ANDD. The reason of the improvement is that τ_w in the slip boundary condition is obtained from a good enough prediction in the inner region. In turn, this provides more accurate mean velocity profiles. However, in subfigure (a) the secondary peak of C_f along the straight wall located around $x/H = 15$ is observed in the experimental data, whereas both the explicit and implicit NDD based on the LRN $k-\varepsilon$ model fail to predict it. In addition, it should be noted from subfigure (b) that the ANDD and implicit NDD with the LRN $k-\varepsilon$ model fail to predict the recirculation region. This is caused by the drawback of the original LRN $k-\varepsilon$ model which is unable to correctly capture any recirculation region in the diffuser flow [30]. However, it is worth noting here that as shown in [14], the recirculation bubble can be effectively predicted with the ANDD in case the Spalart–Allmaras model is used.



(a) Straight wall

(b) Inclined wall

Fig. 11. Comparison of pressure coefficients C_p along the straight and inclined walls of the diffuser, experiment [28], one-block solution, explicit and implicit NDD.

The pressure coefficient is computed as $C_p = 2(p - p_{ref})/(\rho U_b^2)$. Here, the reference pressure p_{ref} is evaluated at the mid-point of the inlet channel. The C_p predictions with both the explicit and implicit NDD are close to the one-block LRN solution, as can be seen in Fig. 11. Similarly to C_f, the pressure coefficient profiles across the diffuser obtained with the implicit NDD are essentially more accurate than those for the ANDD thanks to more accurate prediction of the wall shear stress τ_w.

Fig. 12 shows the computational times for ANDD and implicit NDD with y^{*+} between 10 to 50. The computational times are normalized by the time needed for the one-block simulation. It can be seen that the computational cost of both ANDD and implicit NDD is significantly lower than that of the one-block solution. The reason of that is threefold. With an adaptive to the wall mesh the condition number drastically decreases in both regions. Thus, the algorithm works as a preconditioner. In the inner region, the computing time is almost negligible since the solution can be obtained analytically in quadrature. Finally, in the outer region the time-relaxation parameter might be significantly increased since the minimal space step can be essentially larger than that in the one-block approach. As y^{*+} increases the computational time decreases for both ANDD and implicit NDD. The maximum time occurs in the case of $y^{*+} = 10$ with the implicit NDD, which is roughly one-third of the time for the one-block solution. As mentioned above, the accuracy of the ANDD and implicit NDD is reduced as the interface is far away from the wall. Thus, there is a trade-off between the accuracy and computational time for both the explicit and implicit NDD approaches. As can be expected, the implicit NDD needs slightly more time that the conventional NDD because in the former approach the viscosity coefficient is calculated from the RANS model used.

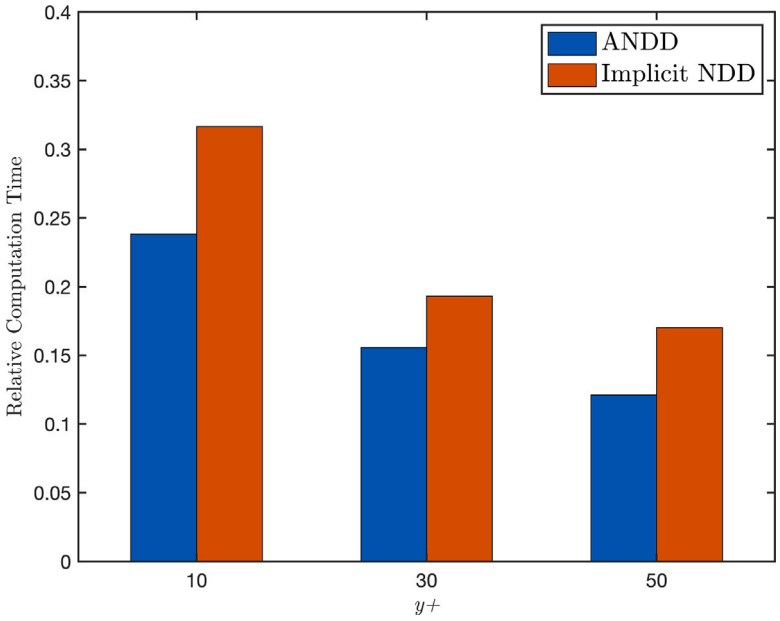


Fig. 12. Relative computational times with the one-block solution, explicit and implicit NDD. The times are normalized by that of one-block solution.

6. Conclusions

An implicit near-wall domain decomposition approach has been proposed for the first time. It is based on a specially constructed slip boundary condition for the velocity. The algorithm allows us to calculate a solution in the entire region on a relatively coarse grid. Then, the solution is locally recalculated in the inner region. As a result, the composite solution is smooth up to the first derivative. For the first time, in the inner region the turbulent viscosity is calculated from the RANS model rather than prescribed by an analytical formulae. This allows us to enhance the accuracy of prediction. The entire algorithm along with the conventional ANDD approach has been successfully implemented in the OpenFOAM code and applied to test cases. The considered test cases contain a 1D channel flow and 2D flow in an asymmetric diffuser flow. The obtained results demonstrated the efficiency of the developed algorithm. It can be easily implemented in the existing codes for different turbulence models and has essentially higher accuracy than the original approach to the near-wall domain decomposition.

CRediT authorship contribution statement

Shiyu Lyu: Software, Computations, Validation, Visualization. **Sergey Utyuzhnikov:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Shiyu Lyu reports financial support was provided by China Scholarship Council.

Data availability

No data was used for the research described in the article.

Acknowledgments

The first author was financially supported by the China Scholarship Council (grant No. 201708060474). The second author was partially supported by grant EPSRC, UK EP/V038249/1. The authors are grateful to the referees for useful remarks which essentially improved the quality of the paper.

References

[1] Craft T, Gant S, Iacovides H, Launder B. A new wall function strategy for complex turbulent flows. Numer Heat Transfer B 2004;45(4):301–18.
[2] Launder BE, Spalding DB. The numerical computation of turbulent flows. Comput Methods Appl Mech Engrg 1974;3(2):269–89.
[3] Knopp T. On grid-independence of RANS predictions for aerodynamic flows using model-consistent universal wall-functions. in: Proceedings of ECCOMAS CFD; 2006.
[4] Grotjans H, Menter F. Wall functions for general application CFD codes. Comput. Fluid Dynam. Eur. Conf. 1998;1(2):1112–7.
[5] Gant SE. Development and application of a new wall function for complex turbulent flows (Ph.D. thesis), University of Manchester Institute of Science and Technology, Manchester, UK; 2002.
[6] Kalitzin G, Medic G, Iaccarino G, Durbin P. Near-wall behavior of RANS turbulence models and implications for wall functions. J Comput Phys 2005;204(1):265–91.
[7] Craft T, Gerasimov A, Iacovides H, Launder B. Progress in the generalization of wall-function treatments. Int J Heat Fluid Flow 2002;23(2):148–60.
[8] Utyuzhnikov S. Some new approaches to building and implementation of wall-functions for modeling of near-wall turbulent flows. Comput & Fluids 2005;34(7):771–84.
[9] Utyuzhnikov S. The method of boundary condition transfer in application to modeling near-wall turbulent flows. Comput & Fluids 2006;35(10):1193–204.
[10] Utyuzhnikov S. Interface boundary conditions in near-wall turbulence modeling. Comput & Fluids 2012;68:186–91.
[11] Utyuzhnikov S. Robin-type wall functions and their numerical implementation. Appl Numer Math 2008;58(10):1521–33.
[12] Utyuzhnikov S. Domain decomposition for near-wall turbulent flows. Comput & Fluids 2009;38(9):1710–7.
[13] Jones A, Utyuzhnikov S. Application of a near-wall domain decomposition method to turbulent flows with heat transfer. Comput & Fluids 2015;119:87–100.
[14] Jones A, Utyuzhnikov S. A near-wall domain decomposition approach in application to turbulent flow in a diffuser. Appl Math Model 2016;40(1):329–42.
[15] Jones A, Utyuzhnikov S. Efficient computation of turbulent flow in ribbed passages using a non-overlapping near-wall domain decomposition method. Comput Phys Comm 2017;217:1–10.
[16] Petrov M, Utyuzhnikov S, Chikitkin A, Titarev V. On extension of near-wall domain decomposition to turbulent compressible flows. Comput & Fluids 2020;210:1–9.

[17] Chikitkin A, Utyuzhnikov S, Petrov M, Titarev V. Non-overlapping domain decomposition for modeling essentially unsteady near-wall turbulent flows. Comput & Fluids 2020;202:1–9.
[18] Johnson PL, Bassenne M, Moin P. Turbophoresis of small inertial particles: theoretical considerations and application to wall-modelled large-eddy simulations. J Fluid Mech 2020;883.
[19] De Vanna F, Cogo M, Bernardini M, Picano F, Benini E. Unified wall-resolved and wall-modeled method for large-eddy simulations of compressible wall-bounded flows. Phys Rev Fluids 2021;6(3):034614.
[20] Mukha T, Rezaeiravesh S, Liefvendahl M. A library for wall-modelled large-eddy simulation based on openfoam technology. Comput Phys Comm 2019;239:204–24.
[21] Van Driest ER. On turbulent flow near a wall. J Aeronaut Sci 1956;23(11):1007–11.
[22] Bose S, Moin P. A dynamic slip boundary condition for wall-modeled large-eddy simulation. Phys Fluids 2014;26:1–9.
[23] Bae H, Lozano-Durán A, Bose S, Moin P. Dynamic slip wall model for large-eddy simulation. J Fluid Mech 2019;859:400–32.
[24] Yamamoto Y, T. Y. Numerical evidence of logarithmic regions in channel flow at R e τ=8000. Phys Rev Fluids 2018;3.1:012602.
[25] Launder BE, Sharma BI. Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc. Lett Heat Mass Transf 1974;1(2):131–7.
[26] Duprat C, Balarac G, Métais O, Congedo PM, Brugière O. A wall-layer model for large-eddy simulations of turbulent flows with/out pressure gradient. Phys Fluids 2011;23(1):015101.
[27] Jones A. Deng thesis "development of a near-wall domain decomposition method for turbulent flows". Manchester University; 2015, p. 271.
[28] Buice CU, Eaton JK. Experimental investigation of flow through an asymmetric plane diffuser. Centre Turbul Res Annu Res Brief 1995 1995;117–20.
[29] Weller HG, Tabor G, Jasak H, Fureby C. A tensorial approach to computational continuum mechanics using object-oriented techniques. Comput Phys 1998;12(6):620–31.
[30] Hellsten A, Rautaheimo P, Orpana M. 8th ERCOFTAC/IAHR/COST workshop on refined turbulence modelling. ERCOFTAC. Finland: Helsinki University of Technology, Espoo; 1999.