Application of a near-wall domain decomposition method to turbulent flows with heat transfer

Adam Jones [a,*], Sergey Utyuzhnikov [a,b]

[a] School 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

ABSTRACT

A near-wall domain decomposition method for use in turbulence modelling is applied to the $k - \omega$ SST, Spalart–Allmaras and BL-$\overline{v^2}/k$ turbulence models. The near-wall region is excluded from the main computational mesh. This eliminates the expense of computing the solution in the viscous sub layer and reduces the total computation time.

A one-dimensional boundary layer equation is used to transfer the wall boundary condition from the wall to an interface located within the flow domain. The boundary conditions imposed on the interface are of Robin type and are written in mesh-independent form. The boundary layer equation can contain source terms such as the pressure gradient or near-wall damping terms. Scalar boundary conditions can be calculated using the same formalism as the boundary conditions for the velocity.

The implementation of the boundary conditions is tested on a channel flow, two heated annulus flows and a two-dimensional, asymmetric diffuser. For each case, different locations of the interface boundary are tested. The results are not sensitive to the location of the interface. Friction factors and heat transfer data calculated with the domain decomposition approach compare well to the results obtained with the fully-resolved forms of the respective turbulence models.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Fluid flows in practical engineering applications often have very large Reynolds numbers. In such flows, a boundary layer develops near to the walls in which the velocity of the fluid changes rapidly with distance from the wall. This poses a significant computational challenge. In order to capture the large gradients in the boundary layer, the computational mesh must be very fine near the wall. This leads to a large computation time.

In large eddy simulation (LES), the computational requirement of resolving the near-wall layer often makes a wall-resolved LES computation impractical. However as computing power increases, it is becoming more common to fully resolve the boundary layers in a computational fluid dynamics (CFD) calculation, particularly with Reynolds averaged Navier–Stokes (RANS) models. Turbulence models that resolve the boundary layer are called low Reynolds number (LRN) models. However the time requirements of resolving the near-wall layer can be as much as 90% of the total computation time. This is unappealing for industrial applications, where a solution is often required in a short space of time.

For these reasons, various approaches have been developed to simplify the modelling of the near-wall regions of flows. In RANS computations, wall functions are often used. These are semi-empirical correlations that link the velocity at the cell nearest to the wall with the shear stress at the wall. This allows a coarse grid to be used near to the wall whilst retaining an adequate approximation of the wall shear stress. The coarser grid reduces the computational cost of the simulation. Models that use wall functions are called high-Reynolds number (HRN) models. The wall function represents the inner (near-wall) region while the turbulence model represents the outer region [16].

The earliest wall functions are based on the log law, which is valid under certain circumstances in the fully turbulent region of the boundary layer. The logarithmic relationship between wall distance and flow velocity is used to calculate the wall shear stress. However the log law is only valid in the logarithmic region of the boundary layer. If the near-wall cell is located in the viscous sub layer then the wall function fails. Furthermore in many flows, such as those with separation regions, the logarithmic region of the boundary layer does not exist at all. Such wall functions are commonly used in engineering applications today, even though they are of limited accuracy, especially if the near-wall cell centre does not lie in the logarithmic region of the boundary layer.

* Corresponding author. Tel.: +44 161 306 3707.
E-mail addresses: adam.jones-2@manchester.ac.uk (A. Jones), s.utyuzhnikov@manchester.ac.uk (S. Utyuzhnikov).

An improvement to this type of wall function is the scalable wall function (SWF) [18]. With this approach, if the near-wall cell happens to be in the viscous sub layer, boundary conditions are imposed as if the cell were at the very edge of the viscous sub layer. The log law is still used to compute the wall shear stress, however the requirement that the near-wall cell centre lie in the logarithmic region of the boundary layer has been removed. For this reason, and also because of its simplicity, the SWF is a commonly used wall function. However the wall function cannot account for pressure gradients, and is invalid in flows where separation occurs.

In recent years, more sophisticated wall functions have been developed to try to further improve the range of validity of wall functions. One such example is the analytical wall function [12]. This wall function assumes a piecewise linear variation of the turbulent viscosity near to the wall, which permits analytical integration of the momentum boundary layer equations across the boundary layer. This wall function does not explicitly use the log law. In addition, the pressure gradient is included in the integration, which improves the accuracy of the boundary conditions compared to log law-based approaches. This wall function is not widely used in industrial CFD codes because the analytical expressions are rather long and cumbersome, especially since special treatment is required if the near-wall cell centre lies in the viscous sublayer. Moreover it is difficult to generalise this approach to unstructured codes.

Another example of an improved wall function is the numerical wall function (NWF) [11], which does not assume a viscosity profile. A one-dimensional numerical grid is used to solve a simplified version of the momentum equations in every near wall cell. This means that the NWF is, in general, more accurate than the AWF. However, this wall function suffers from stability issues [17]. Implementation of this approach also requires significant modifications to the underlying CFD code, and is fraught with technical issues in unstructured codes.

The alternative to using wall functions is to use a near-wall domain decomposition (NDD) method [36]. With NDD approaches, the computational domain is split into an outer region and the inner region, which is near to the wall. In contrast to wall functions, which are intrinsically mesh-dependent [16], the NDD approach seamlessly merges the solution in the inner and outer regions via mesh-independent interface boundary conditions (IBCs).

In this paper a NDD approach is used in which IBCs [1,33–35,37] are applied on the interface between the inner and outer regions. IBCs are calculated using a boundary layer equation in the inner region to transfer the wall boundary condition to the interface. This leads to a Robin type boundary condition at the interface. The full form of the pressure gradient and other source terms can be included in the calculation of the IBCs.

IBCs can be calculated for both HRN and LRN models. In the case of HRN models, the interface should be located sufficiently far from the wall so that the HRN model is applicable in the outer region. This restriction is analogous to the requirement with wall functions that the near-wall cell centre be in the fully turbulent part of the boundary layer. In the case of LRN models, there is no lower restriction on the location of the interface boundary. As the interface boundary approaches the wall, the IBCs tend to the original wall boundary conditions without any modification.

The LRN solution is optimal in that it offers the highest accuracy available with RANS modelling, however the LRN solution takes the most computation time to calculate. On the other hand, the HRN solution is optimal in that the simulation is fastest, however the accuracy is in general the worst. The NDD solution allows a trade-off to be made between accuracy and computation time. As the distance from the interface boundary to the wall increases the computational time reduces since the mesh size decreases,

however the accuracy of the solution also decreases. Hence as y^* changes, NDD provides all Pareto solutions, whilst the LRN and HRN solutions represent the two extreme (ideal) solutions.

IBCs are expressed in mesh-independent form and contain no free parameters. The only requirement is an approximation of the turbulent viscosity in the inner region. The more accurate this profile is, so the more accurate the NDD solution will be. Many near-wall turbulent viscosity profiles are based on the wall shear stress [8,2] and produce accurate results in one-dimensional regions of flow. However if there are separation points in a flow, then these profiles fail and other profiles must be used. In this work, a turbulent viscosity profile is chosen that depends on the pressure gradient [14]. This is shown to produce accurate results even in regions of flow recirculation.

Previously IBCs have only been applied to the $k - \varepsilon$ model [33,35,37]. The $k - \varepsilon$ model is one of the earliest turbulence models and is known to perform poorly in many situations, such as in regions with adverse pressure gradients or streamline curvature, rotating flows and flows with heat transfer [28,31]. In this paper, IBCs are applied to the $k - \omega$ SST [26], Spalart–Allmaras [32] and BL-$\overline{v^2}/k$ [4] models for the first time.

The paper begins in Section 2 by deriving the IBCs used with the NDD approach. Implementation of IBCs with the three turbulence models used is explained in Section 3. Details of the computational code used in this work are given in Section 4. The four test cases studied in this work are discussed in Section 5. These are: a plane channel flow, two different annulus flows and an asymmetric, two-dimensional diffuser. In the case of the two annuli, heat transfer data are reported. A comparison of the computation time for the NDD approach and conventional LRN models with and without wall functions is given for the case of the diffuser. The conclusions are given in Section 6.

2. Near-wall domain decomposition and interface boundary conditions

The governing RANS equation for a function Φ can usually be written

$$\frac{\partial (\rho \Phi)}{\partial t} + \nabla \cdot (\rho \mathbf{U} \Phi) = \nabla \cdot (\Gamma_\Phi \nabla \Phi) + F, \qquad (1)$$

where ρ is the density, $\mathbf{U}$ is the velocity and F is any source terms. For brevity, the diffusion coefficient for the function Φ is written as

$$\Gamma_\Phi \equiv \frac{\mu}{\sigma_\Phi} + \frac{\mu_t}{\sigma_{t,\Phi}}, \qquad (2)$$

where μ is the dynamic viscosity, μ_t is the turbulent viscosity, σ_Φ is the Prandtl number for Φ and $\sigma_{t,\Phi}$ is the turbulent Prandtl number for Φ. The function Φ could be a velocity component, transported scalar or function from a turbulence model. It is assumed that Φ obeys a Dirichlet boundary condition at the wall.

In the near-wall region, $0 \leqslant y \leqslant y^*$, Eq. (1) can be written as

$$\frac{\partial}{\partial y}\left(\Gamma_\Phi \frac{\partial \Phi}{\partial y}\right) = R_\Phi(y). \qquad (3)$$

The full form of the right hand side, $R_\Phi(y)$, contains wall-parallel diffusion, the convection terms and any source terms. In many situations, it is a good-enough approximation to ignore wall-parallel diffusion and convection in the inner region [12,23,30]. This approximation is made in this work. Under such approximations, the form of R_Φ for the U velocity component is $R_U(y) = P_x$, where P_x is the streamwise pressure gradient. It is stressed, however, that convection and wall-parallel diffusion need not be neglected. In contrast, conventional wall functions, such as the SWF, typically require that $R_\Phi = 0$.