Computer Physics Communications 217 (2017) 1–10

Contents lists available at ScienceDirect

Computer Physics Communications

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

Efficient computation of turbulent flow in ribbed passages using a non-overlapping near-wall domain decomposition method

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

Article history:
Received 3 March 2016
Received in revised form 26 January 2017
Accepted 24 February 2017
Available online 4 March 2017

Keywords:
Domain decomposition
Robin boundary condition
Ribbed passages
Design optimisation

ABSTRACT

Turbulent flow in a ribbed channel is studied using an efficient near-wall domain decomposition (NDD) method. The NDD approach is formulated by splitting the computational domain into an inner and outer region, with an interface boundary between the two. The computational mesh covers the outer region, and the flow in this region is solved using the open-source CFD code *Code_Saturne* with special boundary conditions on the interface boundary, called interface boundary conditions (IBCs). The IBCs are of Robin type and incorporate the effect of the inner region on the flow in the outer region. IBCs are formulated in terms of the distance from the interface boundary to the wall in the inner region. It is demonstrated that up to 90% of the region between the ribs in the ribbed passage can be removed from the computational mesh with an error on the friction factor within 2.5%. In addition, computations with NDD are faster than computations based on low Reynolds number (LRN) models by a factor of five. Different rib heights can be studied with the same mesh in the outer region without affecting the accuracy of the friction factor. This is tested with six different rib heights in an example of a design optimisation study. It is found that the friction factors computed with NDD are almost identical to the fully-resolved results. When used for inverse problems, NDD is considerably more efficient than LRN computations because only one computation needs to be performed and only one mesh needs to be generated.

1. Introduction

In industrial computational fluid dynamics (CFD), designers often need to study flows within a fixed geometry multiple times with different boundary conditions and flow scenarios. However, wall bounded flows are computationally expensive to resolve because of the boundary layers that are present near the walls. Thus, methods that can reduce the computation time are of significant benefit to industrial users. Moreover, when optimising the geometry of a design using CFD, a separate mesh must be created for each geometry. This increases the cost of generating a solution, as more time is spent creating the relevant geometries and meshes, and also delays the acquisition of useful results.

In this paper, turbulent flow in a ribbed channel is studied with a near-wall domain decomposition (NDD) method [1–5]. With NDD, the computational domain is split into an outer region, which must be meshed, and one or more inner regions. The flow structure in the inner region does not need to be resolved at each iteration. Instead, the response of the flow in the inner region to changes in the outer region is incorporated in special boundary conditions at the interface, called interface boundary conditions (IBCs). This leads to a reduction in the number of cells in the computational mesh, and a corresponding decrease in the simulation time. In addition, the IBCs depend only on a parameter, y^*, which is the distance from the interface boundary to the wall. Thus, different shapes of inner region can be studied with the same computational mesh of the outer region.

In this paper, for the first time, NDD is applied to turbulent flow in a ribbed channel, and it is demonstrated how different rib heights can be accurately studied using only a single mesh of the outer region. Thus, in a design optimisation study, the height of the rib can be optimised with only one computational mesh. An example of a study in which multiple rib heights are studied using only a single mesh is presented and it is found that the friction factors are in excellent agreement with the solutions based on wall-resolved, low Reynolds number (LRN) models.

1.1. Near-wall domain decomposition

In order to capture the large gradients that occur in boundary layers, small cells are needed near to the walls in wall-bounded flows. The thickness of the boundary layer decreases as

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

http://dx.doi.org/10.1016/j.cpc.2017.02.024
0010-4655/

the Reynolds number increases, which means that the problem is especially severe in the high Reynolds number flows that typically occur in engineering applications. Resolution of the boundary layers of a flow can be responsible for the vast majority of the total time required for a computation.

Reynolds averaged Navier–Stokes (RANS) models that are able to compute the entire boundary layer are referred to as LRN models. These can be contrasted with high Reynolds number (HRN) models, which do not compute the entire boundary layer. Instead, with HRN models a coarse mesh must be used with a large near-wall cell, and a wall function is used to determine the boundary condition at the wall. A wall function is usually a semi-empirical formula that links the wall shear stress with the fluid velocity in a particular region of the flow. Often wall functions use the logarithmic law of the wall [6,7]. These wall functions link the velocity in the logarithmic region of a turbulent boundary layer to the wall shear stress. In order to use such a wall function, the near-wall cell must be of an appropriate size such that the wall function correlations are valid. This condition is difficult to satisfy a priori, and several meshes may need to be tested before an appropriate mesh is found. In some flows the condition is impossible to satisfy everywhere. This is often the case in flows which exhibit complex flow physics, such as flow separation. In regions of separated flow, the logarithmic region of the boundary layer may not exist at all. Therefore modified wall functions were developed to remove the limitation of the validity of the log law to only the logarithmic region of the boundary layer.

Wall functions are still an active topic of current research as CFD is applied to more and more complex flows. Look-up table approaches can be used to develop a wall function for a particular turbulence model [8,9]. Alternatively, blending functions that merge analytical solutions in the viscous and turbulent regions of the boundary layer can be used to create a wall function that can be used at all points in the boundary layer [10]. In addition, wall functions that can incorporate momentum source terms have also been developed [11]. However, these wall functions are formulated in terms of the mesh. Thus, to have any saving in computation time, the accuracy of the solution must be decreased and the solution becomes mesh-dependent.

The work in this paper develops a recently-proposed NDD approach for near-wall modelling that does not use wall functions [4,5,12,13]. A boundary layer equation is assumed to hold in the inner regions of the flow and is used to transfer the boundary conditions from the wall to the interface boundary. The full RANS equations are solved only in the outer region. Hence the computational mesh need not include prism layers near to the wall. This reduces the size of the matrix system that must be solved at each time step and makes the method efficient. The method is also more accurate than many wall-function based approaches since the full form of any source terms can be included in the boundary layer equation. In contrast with most wall functions, the method does not contain any free parameters.

With NDD, the mesh in the outer region is completely independent of the inner region. Thus, a mesh-independent solution in the outer region is obtained with sufficient refinement. In addition, the size of the inner region is determined by a single parameter, y^*. It is therefore possible to study different sizes of inner regions by modifying only y^*, and without creating a new mesh in the outer region. This makes NDD particularly well-suited to design optimisation problems and inverse problems, where the geometry of a flow is the output of a computation and a parameter such as the friction factor is the input.

When implemented for a LRN model, there is no lower limit on the value of y^*. As $y^* \to 0$ the NDD solution asymptotes to the LRN solution in all cases [4,12]. Thus, with NDD the user is in control of the trade-off between accuracy and computation time.

The NDD approach with IBCs has previously been applied to one-dimensional flows with the $k-\varepsilon$, Spalart–Allmaras, $k-\omega$ SST and BL-$\overline{\upsilon^2}/k$ models [4,12,14]. In this paper NDD is used with the Spalart–Allmaras model to study two-dimensional flow in a ribbed channel.

1.2. Flows in ribbed channels

Flows through ribbed channels have been extensively studied over many years. The flow in between the ribs is complex and two-dimensional, which makes it a challenging flow for any wall function method. The ribs cause flow separation and reattachment, which enhances the heat transfer. This is widely exploited in industrial applications such as turbine blade cooling [15] and advanced gas-cooled reactor (AGR) fuel assemblies [16]. However, the increase in heat transfer must be balanced against the increase in pressure drop caused by the ribs.

The pressure drop in flows in ribbed channels is of particular importance in flows in AGR cores, where a carbon deposit forms on the ribbed fuel pins in the fuel assembly [17]. The deposit fills in part of the region between the ribs in the fuel pins, which changes the effective rib height. This introduces uncertainty into the pressure drop across a fuel assembly, which in turn introduces uncertainty into the computed channel power. This has important implications for assessment of decay heats, uncertainties in which can lead to delays in refuelling and increased costs.

Rau et al. [18] studied the flow of air through a square channel with square ribs arranged normal to the flow on either one wall or two opposite walls of the channel. The ratio of the rib pitch to rib width was varied and it was found that the friction factor and heat transfer enhancement caused by the ribs peaks at a rib pitch to width ratio of approximately 9. The same result was found in a study by Okamoto et al. [19], who found additionally that the optimum rib pitch to width ratio is approximately independent of the Reynolds number. Experimental studies of similar flows have been performed by Casarsa et al. [20] and Arts et al. [21], and numerical studies have been carried out in [15,16,22,23]. Rotating ribbed channels have been studied with Direct Numerical Simulation (DNS) in [24] up to Reynolds number of approximately 1.5×10^4. However, the computational costs of DNS are excessively high for design purposes, and RANS methods as well as large eddy simulation remain the methods of choice for industrial users, especially for design optimisation [25,26].

1.3. Structure of this paper

The structure of this paper is as follows. Firstly the computational code used in this work, *Code_Saturne* is introduced in Section 2, along with the governing equations of the Spalart–Allmaras turbulence model. The theory underlying the NDD method and the derivation of IBCs are described in Section 3. In Section 4, the ribbed channel flow of Rau et al. [18] is studied using NDD, to assess the maximum portion of the rib that can be removed from the computational mesh. Friction factors, computation times and profiles of the pressure and velocity from the NDD, LRN and experimental results are compared. Section 5 gives an example of a design optimisation study performed with NDD, where different rib heights are studied using the same computational mesh. Conclusions from the work in this paper are presented in Section 6.

2. Computational code

The computational code used for this work is the open source CFD code *Code_Saturne*, version 3.0.5. *Code_Saturne* is based on a colocated finite volume scheme for unstructured meshes in which