Computer Physics Communications ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Computer Physics Communications

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

An approach for accelerating incompressible turbulent flow simulations based on simultaneous modelling of multiple ensembles

Boris I. Krasnopolsky

Laboratory of General Aerodynamics, Institute of Mechanics, Lomonosov Moscow State University, 119192, Moscow, Michurinsky ave. 1, Russia

ARTICLE INFO

Article history:
Received 5 June 2017
Received in revised form 12 January 2018
Accepted 27 March 2018
Available online xxxx

Keywords:
Turbulent flow
Direct numerical simulation
Ensemble averaging
Multiple right-hand sides
Generalized sparse matrix–vector multiplication
High performance computing

ABSTRACT

The present paper deals with the problem of improving the efficiency of large scale turbulent flow simulations. The high-fidelity methods for modelling turbulent flows become available for a wider range of applications thanks to the constant growth of the supercomputers performance, however, they are still unattainable for lots of real-life problems. The key shortcoming of these methods is related to the need of simulating a long time integration interval to collect reliable statistics, while the time integration process is inherently sequential.

The novel approach with modelling of multiple flow states is discussed in the paper. The suggested numerical procedure allows to parallelize the integration in time by the cost of additional computations. Multiple realizations of the same turbulent flow are performed simultaneously. This allows to use more efficient implementations of numerical methods for solving systems of linear algebraic equations with multiple right-hand sides, operating with blocks of vectors. The simple theoretical estimate for the expected simulation speedup, accounting the penalty of additional computations and the linear solver performance improvement, is presented. The two problems of modelling turbulent flows in a plain channel and in a channel with a matrix of wall-mounted cubes are used to demonstrate the correctness of the proposed estimates and efficiency of the suggested approach as a whole. The simulation speedup by a factor of 2 is shown.

1. Introduction

The modelling of turbulent flows is one of the typical applications for high performance computing (HPC) systems. The accurate eddy-resolving simulations of turbulent flows are characterized by huge computational grids and long time integration intervals, making them extremely time-consuming to solve. Despite the constant growth of the computational power of HPC systems, the use of high-fidelity methods is still unattainable for lots of real-life applications. This motivates the researchers to develop new computational algorithms and adapt the known algorithms to the modern HPC systems.

The high-order time integration schemes are typically used to increase the accuracy of turbulent flow simulations with eddy-resolving methods. The 3th or 4th order explicit or semi-implicit Runge–Kutta schemes (e.g., [1–3]) or 2nd order Adams–Bashforth/Crank–Nicolson schemes [4–7] are among the widely used ones for time integration of incompressible flows. In these schemes, the preliminary velocity distributions are obtained from the Navier–Stokes equations, and the continuity equation, transformed to the elliptic pressure Poisson equation, is used to enforce a divergence-free velocity field. Commonly, the solution of elliptic equations and corresponding systems of linear algebraic equations (SLAEs) is a complicated and challenging problem. The time to compute this stage can take up to 95% of the overall simulation time.

For a limited number of problems with regular computational domains, solved on structured grids, the direct methods for solving SLAEs can be used (e.g., [8,9]). Otherwise, the iterative methods are the good candidates to solve SLAEs with matrices of the general form. The multigrid methods [10] or Krylov subspace methods (e.g., BiCGStab [11–13], GMRES [14]) with multigrid preconditioners are the popular ones to solve the corresponding systems. The advantages of these methods are related to their robustness and excellent scalability potential [15]. Mathematically, these methods consist of a combination of linear operations with dense vectors, $\mathbf{z} = a\mathbf{x} + b\mathbf{y}$, scalar products, $a = (\mathbf{x}, \mathbf{y})$, and sparse matrix–dense vector multiplications (SpMV), $\mathbf{y} = \mathbf{A}\mathbf{x}$. While the linear operations with dense vectors and scalar products are easily vectorized by compilers and acceptable performance is achieved, the performance of SpMV operations is dramatically lower. The sparse matrix–vector multiplication is a memory-bound operation with extremely low arithmetic intensity. The real performance of linear algebra algorithms with sparse matrices of the general form does not exceed several percent of the peak performance [16–18].

E-mail address: krasnopolsky@imec.msu.ru.

https://doi.org/10.1016/j.cpc.2018.03.023

The optimizations of operations with sparse matrices, which are related to both optimization of matrix storage formats and implementation aspects are a topic of continuous research for many years (e.g., [19–22]).

The performance of SpMV-like operations can be significantly improved if applied to a block of dense vectors simultaneously (generalized SpMV, GSpMV), $\mathbf{Y} = \mathbf{AX}$, where $\mathbf{X}$ and $\mathbf{Y}$ are dense matrices [16,23–25]. Generally, the performance gain of GSpMV operation with m vectors compared to m successive SpMV operations is achieved due to two main factors: (i) the reduction of the memory traffic to load the matrix $\mathbf{A}$ from the memory (the matrix is read only once) and (ii) vectorization improvement for GSpMV operation.

Despite the significant performance advantage of matrix–vector operations with blocks of vectors over the single-vector operations, the GSpMV-like operations are rarely used in real computations. This fact is a consequence of the numerical algorithms design: most of them operate with single vectors only. Among the several exceptions allowing to exploit the GSpMV-like operations are the applications with natural parallelism over the right-hand sides (RHS) when solving SLAEs. For example, in structural analysis applications, the solution of SLAE with multiple RHS arise for multiple load vectors [26]. In computational fluid dynamics, the operations with groups of vectors can be used for solving Navier–Stokes equations (e.g., for computational algorithms operating with collocated grids and explicit discretization of nonlinear terms). For multiphase flows, the concentration transport equations for the phases forming the carrier fluid can also be solved in a single run.

In addition, several articles are focused on the attempts to modify the computational procedure in order to organize the computations with groups of vectors. For example, the modified Stokesian dynamics method for the simulation of the motion of macromolecules in the cell, exploiting the advantages of operations with groups of vectors, is presented in [23]. The benefits of ensemble computing for current HPC systems are outlined in [25]. The authors suggested to perform together several incompressible flow simulations (e.g. applications with varying initial or boundary conditions), which have a common sparse matrix derived from the pressure Poisson equation. This modification allows to combine multiple solutions of the pressure Poisson equation in a single operation with multiple right-hand sides. The 2.4 and 7.6 times speedup for the successive over-relaxation method used to solve the corresponding SLAEs with up to 128 RHS vectors on Intel and Sparc processors are reported by the authors.

The problem of long time integration for high-fidelity turbulent flow simulations is discussed in [27]. The authors suggested to combine the conventional time averaging approach for the statistically steady turbulent flows with the ensemble averaging. Several turbulent flow realizations are performed independently with a shorter time integration interval, and the obtained results are averaged at the end of the simulation. Scheduling additional resources to perform each of the flow realizations, the proposed approach allows to speedup the simulations beyond the strong scaling limit by the extra computational costs.

The current paper discusses an idea of combining the ensemble averaging for statistically steady turbulent flow simulations [27] with the simultaneous modelling of multiple flow realizations [25] to speedup the high-fidelity simulations. The focus is on the reduction of the overall computational costs for the corresponding simulations. The paper provides the modified computational procedure to model incompressible turbulent flows, allowing to utilize the operations with groups of vectors. For the sake of simplicity, the further narration is focused on the direct numerical simulation (DNS) aspects; however, the proposed methodology can be applied "as is" for the large eddy simulation (LES) computations.

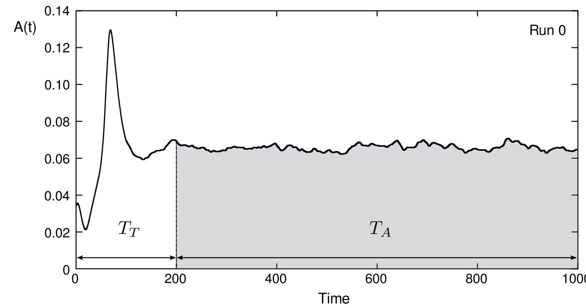

Fig. 1. Integral velocity perturbations amplitude [32] for DNS of turbulent flow in a straight pipe, $Re_D = 6000$; averaging over the single flow.

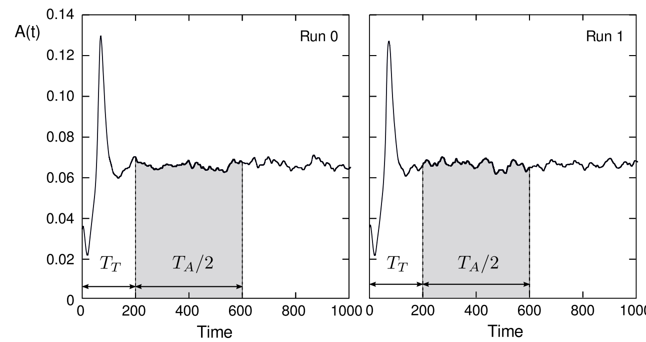

Fig. 2. Integral velocity perturbations amplitude [32] for DNS of turbulent flow in a straight pipe, $Re_D = 6000$; averaging over two flows.

The paper is organized as follows. The motivating observations and theoretical prerequisites of the proposed computational procedure are stated in the second section. The third section contains description of the numerical methods and computational codes used to simulate turbulent flows. Numerical results validating the theoretical estimates and demonstrating the advantages of the proposed algorithm are presented in the fourth section. The fifth section discusses the applicability of the proposed approach to other mathematical models and techniques that can be used for further efficiency improvements.

2. Preliminary observations and theoretical estimates

2.1. Time-averaged and ensemble-averaged statistics

The DNS/LES of turbulent flow comprises integration in time. For the statistically steady turbulent flow the time integration process typically consists of two stages (Fig. 1). The first stage, T_T, reflects transition from the initial state to the statistically steady regime. The second stage, T_A, includes the averaging of instantaneous velocity fields and collecting the turbulent statistics. The ratio of these intervals may vary significantly depending on the specific application. For example, following the ergodicity hypothesis [28,29], the flows with homogeneous directions can be averaged along these directions thus significantly reducing the time averaging interval. Opposite, the applications with complex geometries often need to perform much larger time averaging intervals to obtain reliable statistics compared to the time to reach the statistical equilibrium.

The time averaging to collect turbulent statistics is applied to statistically steady turbulent flows. In case the flow considered is not statistically steady, the ensemble averaging must be performed. The time averaging of statistically steady flows can be combined with the ensemble averaging [27]. Instead of averaging in time over the single flow with time interval T_A, m flows can be averaged over the interval T_A/m (Fig. 2). The flow realizations