

Generation of Multiple Turbulent Flow States for the Simulations with Ensemble Averaging

B. Krasnopolsky

Institute of Mechanics, Lomonosov Moscow State University

Abstract. The paper deals with the problem of improving the performance of high fidelity incompressible turbulent flow simulations on high performance computing systems. The ensemble averaging approach, combining averaging in time together with averaging over multiple ensembles, allows to speedup the corresponding simulations by increasing the computing intensity of the numerical method (flops per byte ratio). The current paper focuses on further improvement of the proposed computational methodology, and particularly, on the optimization of procedure to generate multiple independent turbulent flow states.

Keywords: Incompressible turbulent flow · Generalized sparse matrix vector multiplication · Multiple right-hand sides · Ensemble averaging

1 Introduction

The high-fidelity simulations of turbulent flows for the complex geometries are among the typical applications for the high performance computing (HPC) systems. These applications are characterized by the high computational costs, long simulation times, and low computational efficiency, not exceeding several percent of the peak performance of HPC system. This leads to the situation when the simulations may take months to complete even using modern compute systems.

The present paper focuses on the problem of improving the performance and efficiency of the high-fidelity turbulent flow simulations. An approach to model incompressible turbulent flows, combining conventional time averaging with the ensemble averaging has been proposed in [?,?]. This approach allows to parallelize the simulations in time and replace the single simulation with long time averaging by multiple simulations with much shorter time averaging intervals. The paper [?] suggests to perform each of these simulations independently and to obtain the simulation speedup by increasing the amount of utilized compute resources. This approach increases the corresponding computational costs for the simulation, but allows to speedup the calculations beyond the strong scalability limit. Alternatively, the paper [?] suggests to perform the simulations in a single run and rearrange the computations in order to utilize the operations with blocks of vectors. The multiple solutions of system of linear algebraic equations (SLAE) for the pressure Poisson equation can be combined in a single solution of SLAE with multiple right-hand sides (RHS). The memory traffic reduction

when solving the SLAE with multiple RHS compared to multiple solutions of SLAE with single RHS vectors allows to increase the computational intensity of the algorithm (flops per byte ratio), and as a result, to speedup the turbulent flow simulations by a factor of 1.5-2.

Both simulation scenarios mentioned above are the limiting cases of the generalized algorithm, combining several independent runs and several flow states inside each run. The corresponding estimates and recommendations to choose the optimal simulation scenario have been proposed in [?].

While the results presented in [?,?] demonstrate the proof of concept for the suggested approach to parallelize the simulations in time, several questions affecting overall simulation speedup are not discussed in detail in these publications. Among them is the problem of generation of multiple uncorrelated turbulent flow states to start averaging in time. The simplest approach used in the mentioned publications assumes the simulation of flow transition for multiple different initial flow states. These calculations, however, produce significant computational overhead thus reducing the efficiency of the suggested methodology. The current paper focuses on the problem of improving the efficiency of the proposed ensemble averaging methodology, and particularly, on the aspects of efficient generation of multiple initial turbulent flow states. An alternative strategy to generate initial turbulent flow states based on introduction of random perturbations to the single flow state at different moments in time is examined. The efficiency of the methodology is analyzed in terms of the total time to compute m uncorrelated flow states, that can be used as initial flow states for time averaging.

The paper is organized as follows. The second section contains theoretical estimates reflecting the influence of generation of multiple turbulent flow states on the overall efficiency of the ensemble averaging methodology. A brief description of the computational codes used for the experimental evaluation is presented in the third section. The fourth section is focused on the detailed description of the procedure used to generate multiple turbulent flow states and its experimental comparison with the basic one, including the full simulation of the transition interval for each of flow states.

2 Theoretical estimates

The simplest, but computationally expensive way to generate multiple independent turbulent flow states comprises of modelling of transition to the turbulence for the flows with different initial states (e.g. laminar flows with different perturbations). The choice of different initial flow states guarantees the turbulent flow states finally to become uncorrelated. This statement is a result of exponential divergence of the turbulent flow trajectories in the phase state, and the corresponding divergence growth rate is determined by the Lyapunov exponent [?,?].

The corresponding simulation of transition to the turbulence for multiple flow states includes two characteristic time scales: (i) the time scale to perform transition to the statistically steady turbulent flow state, T_{turb} , and (ii) the

decorrelation time scale when two different turbulent flow states become uncorrelated, T_{corr} . The basic strategy to generate multiple turbulent flow states can be optimized depending on the ratio of these time scales. In case the transition time scale exceeds the decorrelation time scale, $T_{turb} \gg T_{corr}$, the procedure of generation multiple turbulent flow states can be obviously improved. For example, the simulation of the transition interval can be started for the single flow state. Then, the small perturbations are introduced at the bifurcation point, T_B , to generate multiple uncorrelated flow states. The bifurcation point is reasonable to be chosen close to the end of the transition interval, T_T , with the limitation $T_T - T_B \gtrsim T_{corr}$ (see, Fig. 1).

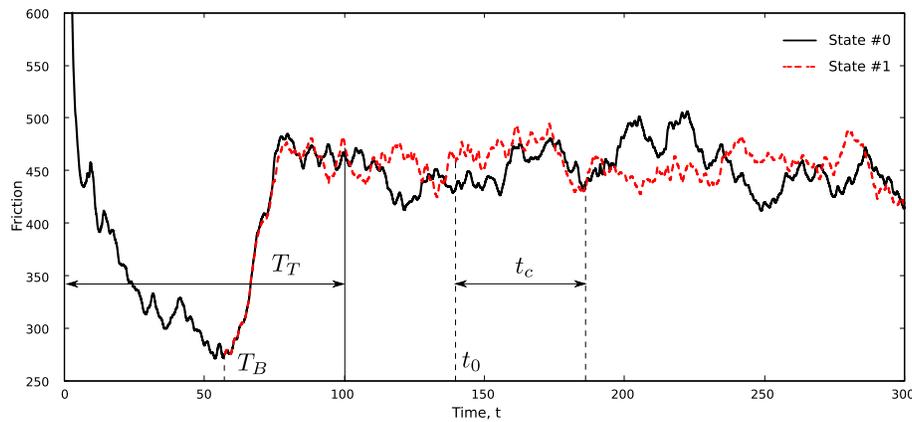


Fig. 1. Evolution of the friction at the channel walls for the problem of modelling turbulent flow in a channel with a matrix of wall-mounted cubes. Results are presented for two flow states; the second state is produced as a result of evolution of random perturbations introduced in T_B moment in time.

The contribution of the optimized procedure to generate multiple turbulent flow states on the overall simulation speedup can be expressed with simple theoretical estimates. Let us consider the expected simulation speedup for the ensemble averaging approach with the suggested procedure to generate multiple initial flow states. Following the methodology suggested in [?], the simulation speedup P_m can be expressed as:

$$P_m = \frac{\mathcal{T}_1}{\mathcal{T}_m}, \quad (1)$$

where \mathcal{T}_1 and \mathcal{T}_m are the times to simulate the problem using averaging over single and m flow states respectively. The time to solve the problem with averaging over single flow state is expressed as:

$$\mathcal{T}_1 = \frac{T_T + T_A}{\tau} t_1, \quad (2)$$

where τ is the integration step, t_1 is the time to simulate single integration step, and T_A is the overall time averaging interval. The simulation time with multiple flow states includes the simulation with single flow state for the initial interval T_B and the subsequent simulation of interval $T_T - T_B + T_A/m$ with m flow states:

$$\mathcal{T}_m = \frac{T_B}{\tau} t_1 + \frac{(T_T - T_B + T_A/m)}{\tau} t_m. \quad (3)$$

Here, t_m is the time to simulate single integration step with m flow states. Substituting these expressions into (1) and introducing the parameters $\beta = T_A/T_T$ and $\gamma = (T_T - T_B)/T_T$, one can obtain the following relation:

$$P_m = \frac{1 + \beta}{(1 - \gamma) + (\beta + m\gamma) \left(\frac{mt_1}{t_m}\right)^{-1}}. \quad (4)$$

Using the estimate for the simulation times ratio based on the memory traffic reduction, proposed in [?], the final form of the estimate is obtained:

$$P_m = \frac{5m(1 + \beta)}{5m(1 - \gamma) + (5m - 3\theta(m - 1))(\beta + m\gamma)}. \quad (5)$$

The expression (5) is the generalization of the basic estimate in [?] on the case $\gamma \neq 1$.

The suggested estimate allows to analyse the influence of the computational costs reduction to perform transition for multiple flow states on the overall performance gain when using the method combining averaging in time with the ensemble averaging to model turbulent flows. The corresponding performance gain distributions presented in Fig. 2, show that the parameter γ strongly affects on the simulation speedup. The reduction of computational overhead to generate multiple turbulent flow states leads to three major improvements: (i) increase of the expected overall simulation speedup, (ii) increase of the optimal number of modelled turbulent flow states, and (iii) extension of the range of applicability for the proposed ensemble averaging methodology towards $\beta \sim 1$.

3 Validation

The estimate (5) vividly demonstrate that reduction of computational costs to construct multiple independent flow states allows to significantly improve the range of applicability of the ensemble averaging methodology and the overall simulation speedup. The suggested approach to generate multiple turbulent flow states is based on fast decorrelation of different flow states compared to the transition interval. In practice, however, the length of both transition and decorrelation intervals is problem specific, and the reliable estimates can be obtained only from some preliminary simulations.

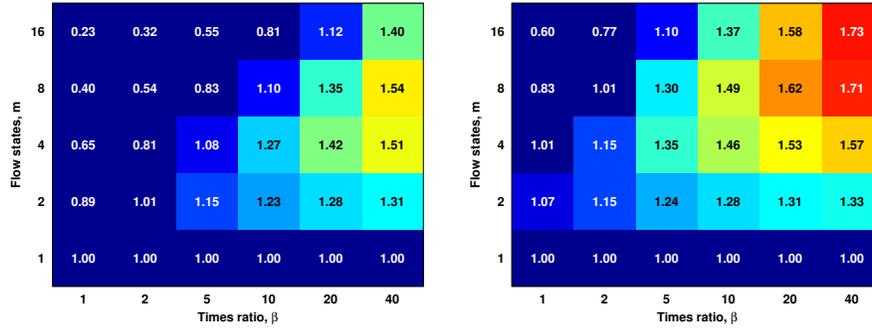


Fig. 2. Expected overall simulation speedup for $\gamma = 1$ (left) and $\gamma = 0.25$ (right).

3.1 Computational codes

The corresponding computational codes to perform the simulations with multiple turbulent flow states, including the SLAE solver and “in-house” application for direct numerical simulation of turbulent flows have been developed. The SLAE solver implements the Krylov-subspace iterative methods, accelerated by the algebraic multigrid preconditioner. The “in-house” DNS code is based on finite difference scheme for structured grids [?] providing second-order spatial discretization together with third-order Runge-Kutta scheme for time integration. The further details on these codes can be found in [?,?].

3.2 Test problem

The current paper uses the problem of modelling turbulent flow in a channel with a matrix of wall-mounted cubes [?] as a validation example. The problem statement assumes the isothermal flow of incompressible viscous fluid, governed by Navier-Stokes and continuity equations, in a rectangular domain containing single dedicated cube (Fig. 3). The corresponding equations are supplemented with periodic boundary conditions on streamwise and spanwise directions, and no-slip conditions on the rigid walls. The constant flow rate is preserved with the Reynolds number $Re_b = U_b h / \nu = 3854$, where U_b is the bulk velocity, h is the cube height, and ν is the kinematic viscosity of the fluid. The integration interval for this problem is set to $T = 2100$ with transition interval $T_T = 100$ and averaging interval $T_A = 2000$ time units, i.e. $\beta = T_A / T_T = 20$. The simulations presented in this paper are performed on structured orthogonal non-uniform computational grid consisting of 2.32 mln. cells. All the simulations are calculated on “Lomonosov-2” supercomputer and use 32 compute nodes per each run.

3.3 Validation procedure

The several simulations are performed to validate the procedure of generation multiple independent turbulent flow states with the introduction of random per-

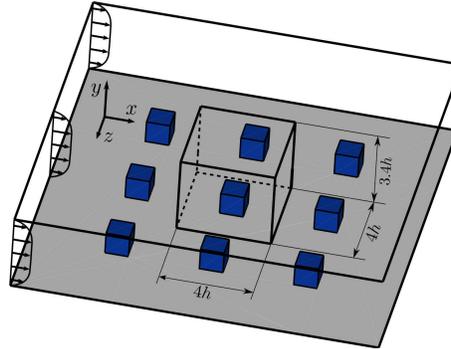


Fig. 3. Sketch of the computational domain.

turbations during the modelling of transition interval. The simulations differ in the choice of the bifurcation point. The cross-correlation functions are used to ensure the decorrelation of the corresponding flow states. The corresponding velocity components time series are collected in the control points around the cube. The cross-correlation function for two time series X and Y is defined as:

$$\rho_{XY} = \frac{cov(X, Y)}{var(X) var(Y)}, \quad (6)$$

where

$$\begin{aligned} cov(X, Y) &= \frac{1}{t_c} \int_{t_0}^{t_0+t_c} (x(t) - \bar{x})(y(t) - \bar{y}) dt, \\ \bar{x} &= \frac{1}{t_c} \int_{t_0}^{t_0+t_c} x(t) dt, \\ var(X) &= cov(X, X). \end{aligned}$$

The time series X and Y are the corresponding subsets of the velocity components profiles, defined by the starting point t_0 and the length of the interval t_c (see, Fig. 1).

3.4 Numerical validation

A series of simulations has been performed to validate the proposed optimized procedure of generation multiple turbulent flow states. The first series comprised of modelling transition interval to generate decorrelated initial turbulent flow states at the end of transition interval. Initially, the simulation is performed with single flow state. Using the obtained intermediate data sets the simulation is restarted at $T_B = 77$ to generate 4 decorrelated flow states. The corresponding simulation time, including the time to simulate single flow state over the interval T_B and 4 flow states from the bifurcation point to the end of transition interval is equal to 71 minutes (see, Tab. 1). This result indicates about

1.9 times speedup for the generation of multiple turbulent flow states compared to basic methodology where the perturbations are introduced at the beginning of transition interval.

Table 1. The time to generate multiple turbulent flow states with various simulation scenarios, results in min.

Flow states, m	T_B	1 flow state	m flow states	Total
1	–	53	–	53
4	0	–	132	132
4	77	41	30	71

The second test series includes simulation of several full scale DNS runs to model flow over a matrix of wall mounted cubes. This includes the run with single flow state, the run with 4 flow states and two runs with restarts from the bifurcation point $T_B = 77$ for 4 and 8 flow states. The corresponding compute times are summarized in tab. 2. Using the expression (5) one can obtain the expected simulation speedup with the optimized procedure of generation multiple turbulent flow states compared to conventional DNS run. With basic procedure to generate initial turbulent flow states, the optimal number of flow states is equal to $m = 4$, and the speedup by a factor of 1.42 is expected. The obtained simulation speedup is 1.38, which is close to predicted one¹. For the simulations with bifurcation point $T_B = 77$, corresponding to $\gamma = 0.23$, performance gain for 4 flow states of about 1.53 is expected, and the optimal number of states is equal to 8 with the speedup by a factor of 1.63. These results correlate with the full scale simulation results, which are equal to 1.49 and 1.57 correspondingly.

Table 2. The computational times to simulate the turbulent flow with various number of simultaneously modelled flow states, results in min.

Flow states, m	1 flow state	m flow states	Total
1	1088	–	1088
4	–	790	790
4	41	688	729
8	41	654	695

The time series for the velocity components monitored in 8 control points around the cube are analyzed to verify the obtained initial turbulent flow states are uncorrelated. The corresponding cross-correlation function distributions are evaluated for the time length $t_c = 100$. The distributions presented in Fig. 4,

¹ The presented simulation times obtained in the current test session are systematically 10-15% higher than the ones published in [?]. The reason for the observed difference will be further investigated.

demonstrate that the turbulent flows become uncorrelated before the end of transition interval, $T_T = 100$.

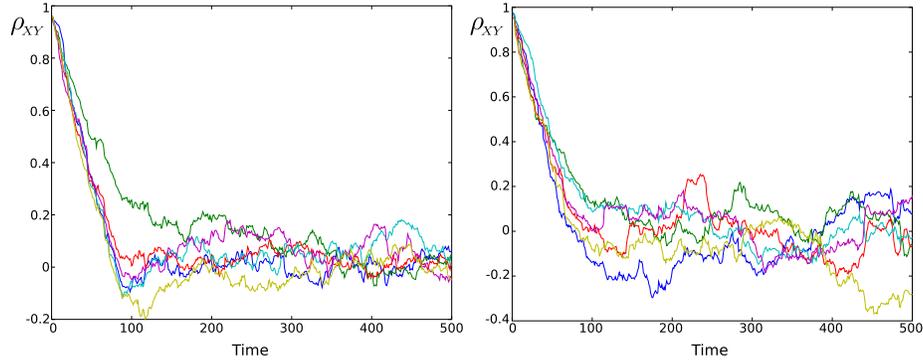


Fig. 4. Cross-correlation distributions for the streamwise velocity in different control points. Results for the simulation with 4 flow states.

4 Conclusion

The current paper focuses on further improvement of the algorithm to model turbulent flows based on ensemble averaging, proposed earlier in [?]. The problem of reducing the computational overhead when generating multiple initial turbulent flow states is investigated. The new procedure to generate uncorrelated flow states based on introduction of random perturbations during the modelling transition to the statistically steady turbulent flow state is proposed. The corresponding theoretical estimates demonstrate that the optimized procedure for generation multiple turbulent flows extends the range of applicability and improves the overall performance gain for the ensemble averaging approach. The corresponding numerical experiments performed for the problem of modelling the turbulent flow in a channel with a matrix of wall-mounted cubes are in agreement with suggested theoretical estimates and demonstrate additional 15% overall speedup for the ensemble averaging methodology.

Acknowledgements

The presented work is supported by the RSF grant No. 18-71-10075. The research is carried out using the equipment of the shared research facilities of HPC computing resources at Lomonosov Moscow State University.