DOI: 10.14529/jsfi210401

Technology for Supercomputer Simulation of Turbulent Flows in the Good New Days of Exascale Computing

Andrey V. Gorobets[1], *Alexey P. Duben*[1]

A technology for scale-resolving simulations of turbulent flows in the problems of aerodynamics and aeroacoustics is presented. It is based on the higher accuracy numerical schemes on unstructured mixed-element meshes and latest non-zonal hybrid approaches combining Reynolds-averaged Navier – Stokes (RANS) and Large eddy simulation (LES) methods for turbulence modeling. It targets a wide range of high performance computing (HPC) systems, from a compute server or small cluster to an exascale supercomputer. The advantages of the key components of the technology are summarized. These key components are a hybrid RANS-LES turbulence modeling method, a numerical scheme for discretization in space, a parallel algorithm, and a portable software implementation for modern hybrid systems with extra massive parallelism. Examples of our simulations are given and parallel performance on various HPC systems is presented.

Keywords: computational fluid dynamics, turbulent flows, scale-resolving simulation, hybrid RANS-LES approach, CPU+GPU, MPI+OpenMP+OpenCL.

Introduction

Hyrbid RANS-LES methods are widely recognized as the most efficient ones in terms of cost/accuracy ratio in many computational aerodynamics and aeroacoustics applications [6, 11]. Such methods combine Reynolds-averaged Navier – Stokes (RANS) and Large Eddy Simulation (LES) turbulence models. However, scale-resolving simulation of complex configurations such as an entire aircraft is still too computationally expensive for widespread use in practice. The growing computing power and the emergence of exascale supercomputers are expanding the applicability of such resource-intensive applications. The evolution of high-accuracy schemes and turbulence models is aimed at reducing requirements for mesh resolution, which leads to a significant reduction in computational costs. The development of parallel algorithms and heterogeneous software implementations ensures efficient use of modern hybrid supercomputers. The present work is devoted to the successful choice of these main components of the simulation technology: hybrid turbulence modeling approaches, high-accuracy numerical schemes, parallel algorithms and portable software implementation for hybrid supercomputers. In the following sections, a combination of these key components is proposed and the advantages of the selected state-of-the-art methods are outlined.

1. High-accuracy Schemes

To describe a turbulent flow, the Navier – Stokes equations for a viscous compressible gas are discretized in space using unstructured mixed-element meshes. The following is required from numerical schemes: high accuracy to provide sufficiently accurate solutions on much coarser meshes than are needed for low order schemes, low computational cost, low memory requirements to fit in scarce GPU memory, applicability to flows with discontinuities to simulate supersonic flows, implicit time integration to overcome the time step constraints, which make it by far unacceptably small due to the mesh step size in boundary layers. For spatial discretization, we use vertex-centered edge-based reconstruction schemes (EBR) for smooth flows [1] (subsonic)

[1]Keldysh Institute of Applied Mathematics, RAS Moscow, Russian Federation

and flows with discontinuities [2] (supersonic). Quasi one-dimensional reconstruction of variables with simple interpolation constructs significantly increases accuracy while keeping computational cost almost equal to a basic compact low-order scheme. The EBR schemes have "superpowers" on translationally-invariant meshes (such as structured Cartesian mesh zones), which allow reaching the fifth order of accuracy. On arbitrary unstructured meshes, they can compete in terms of accuracy with higher-order schemes, which are far more expensive. In the case of implicit time integration, another advantage of EBR schemes consists in using a simplified Jacobian with the same sparse matrix portrait as for a compact scheme with only direct nodal adjacency by edges. This ability to discard additional nodal couplings of rather wide interpolation constructs dramatically reduces memory consumption, which is critical on GPUs. In terms of accuracy, the effect of using EBR schemes on the numerical solution is shown in Fig. 1. It is important to note that the difference in computing cost with a low order scheme is within 15%. As a drawback, such vertex-centered schemes require careful use in terms of mesh quality, especially in the transition between structured and unstructured mesh areas. Certain stability or monotonicity problems still remain to be solved.

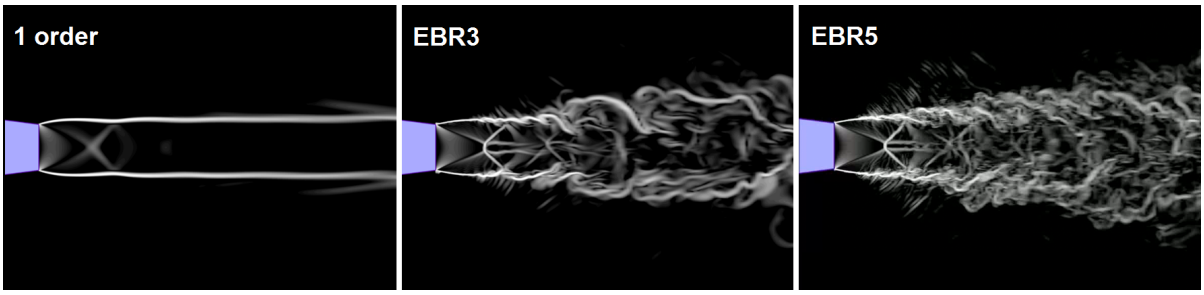


Figure 1. The effect of using EBR schemes compared to a basic first order scheme (4.5 million nodes mesh, round underexpanded jet, density gradient in the mid-span section)

2. Turbulence Modeling

For turbulence modeling, non-zonal hybrid approaches are used, which combine LES and RANS. The former needs fine enough spatial and temporal resolution to capture accurately relevant turbulent structures. The later requires much lower computational costs due to the possibility of using coarser and anisotropic meshes, since only average flow gradients need to be resolved. It is more widely, used but in many applications it can be inaccurate, especially in flows with strong separation, or inapplicable if unsteady characteristics such as noise are needed. Combining RANS in the near-wall regions and LES elsewhere allows taking advantages of both methods: RANS significantly reduces resolution requirements in wall-tangential directions in boundary layers, while LES efficiently reproduce unsteady flow features.

The following is required from modern turbulence modeling approaches: adaptive switching between different modes depending on the mesh resolution and flow features; fast transition from RANS to LES in shear layers; a minimum level of empiricism and user involvement; simplicity of parallel implementation. In accordance with the above requirements, we suggest using the following combination of methods. For faster RANS-LES transition (see Fig. 2), we use the most recent formulations of the detached eddy simulation (DES) approach, the Delayed DES (DDES) [14] and Improved DDES (IDDES) [10], but with alternative LES models and subgrid length scales.

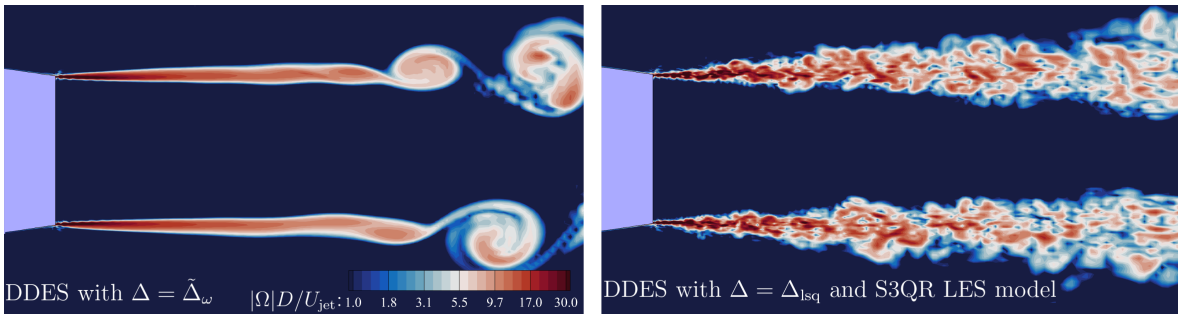


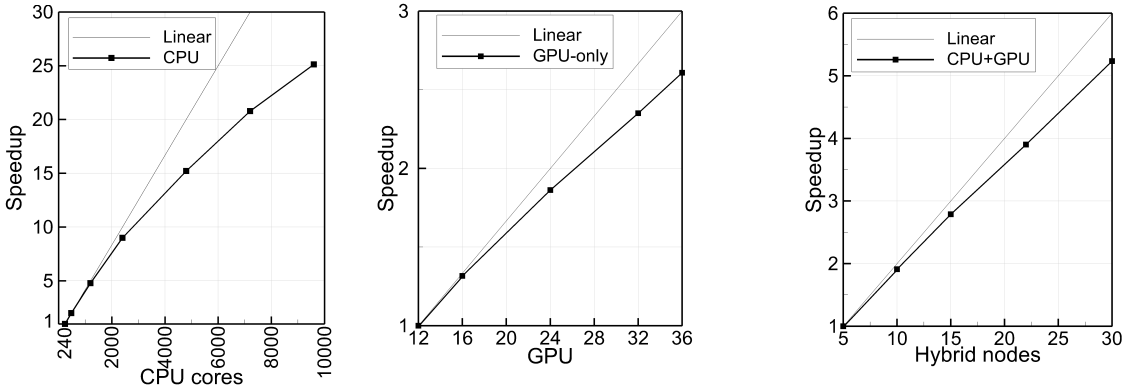
Figure 2. Faster transition to turbulence in shear layers with the Δ_{lsq} scale and S3QR LES model (9 million nodes mesh, round subsonic jet, vorticity magnitude in the mid-span section)

The latest empiricism-free adaptive subgrid length scale Δ_{lsq} [13, 16] is used because it works just as well as the Δ_{SLA} scale used in [10, 14], but it is much easier to implement for unstructured meshes, simply on the basis of a gradient operator with minor modifications. The S3PQR family models [15], used as an alternative LES model, self-adapt to the presence of walls and are sensitive to quasi-2D flow structures, in contrast to Smagorinsky model in [14].

However, despite the fact that new advanced models and length scales have been recently developed, the so-called gray area problem [12] (transition between RANS and LES) still remains to a certain extent. The main efforts are now aimed at its further mitigation.

3. Parallel Computing

From the parallel computing perspective, the following is required: no scalability limitations, distributed-memory parallelization with hiding of communication overhead, efficient shared-memory parallelization for manycore CPUs, full compatibility with stream processing on GPUs, heterogeneous co-execution on both CPU and GPU. These properties enable the use of numerous computing devices, CPUs and GPUs, and open the way to the exascale level. We use hierarchical parallelization based on multilevel mesh decomposition. The Message-passing interface (MPI) is used at the upper level to couple hybrid cluster nodes and devices inside nodes. To reduce network traffic, the mesh is decomposed first among hybrid nodes, then among computing devices, manycore CPUs and GPUs. To hide the data transfer overhead, overlapping communications and computations is used. At the lower level, OpenMP shared memory decomposition-based parallelization is used for manycore CPUs and accelerators. The mesh subdomains of CPUs are further decomposed among parallel threads of MPI processes (instead of using loop-based parallelism, which is less efficient on NUMA systems). Finally, to compute on GPUs, the OpenCL standard is used. In contrast to the NVIDIA CUDA framework, it can engage GPUs from different vendors, including NVIDIA, AMD, Intel. To use CPUs and GPUs concurrently, their mesh subdomains are balanced according to the actual performance ratio. The heterogeneous parallel algorithm is implemented in the NOISEtte code [8]. Further details on parallel algorithm, adaptation of the numerical algorithm and software implementation to GPU computing can be found in [7–9]. Examples of parallel speedups are shown in Fig. 3 for CPUs, GPUs, and CPU+GPU co-execution (numerical configuration: EBR5 scheme, implicit BDF2 scheme, hybrid RANS-LES approach). Relatively coarse meshes of up to 80 million nodes are used in tests in order to observe the degradation of the parallel efficiency of the available rather modest computational resources (on finer meshes, the parallel efficiency is too high to evaluate the limits of parallelism).



(a) Intel Xeon 8160 (b) NVIDIA V100 (c) Intel Xeon E5-2697v3 + NVIDIA K40

Figure 3. Parallel speedups on supercomputers in CPU, GPU, and CPU+GPU computing modes

The summary of performance tests: execution on 36 NVIDIA V100 GPUs is as fast as on 10,000 CPU cores, about 0.27 s per implicit time step; co-execution on CPUs and GPUs on Lomonosov-2 supercomputer (14-core CPU and NVIDIA K40 GPU per node) gives 25–30% speedup compared to GPU-only execution; high parallel efficiency is observed when payload per CPU core is above 10 thousand mesh nodes, or above 1 million nodes per GPU NVIDIA V100 (this load is more than enough to hide most of the exchanges behind computations). On meshes of few billion nodes, several hundred thousand CPU cores or several thousand GPUs can be engaged, which corresponds to computing resources of tens of PFLOPS. When performing a series of simulations for multiple variants of geometry or flow conditions, an entire exascale supercomputer can be fully occupied with high efficiency (if we live to see those bright days when such systems will be available to us).

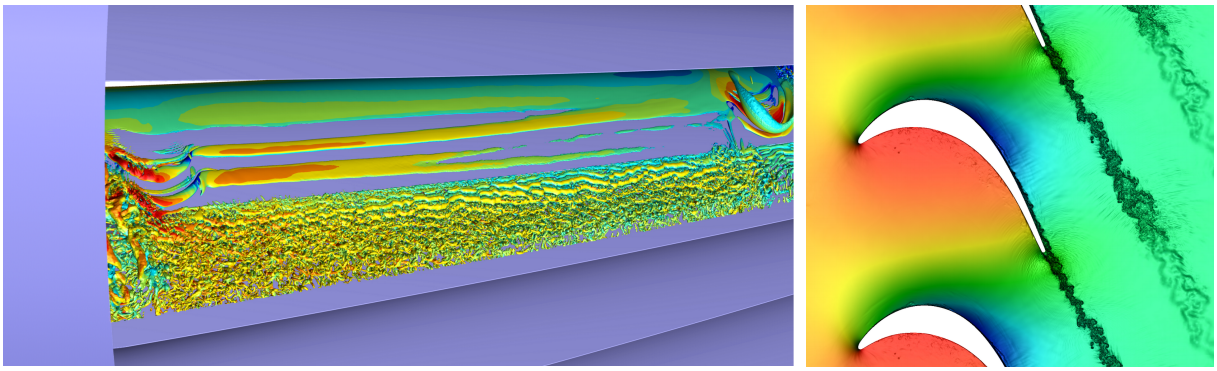
4. Applications

Below are typical examples of our scale-resolving simulations of problems in which stationary RANS methods are either inapplicable or too problematic and inaccurate.

Modeling low-pressure turbine blades of turbofan engines is shown in Fig. 4. The presence of laminar-turbulent (LT) transition and flow separation on the suction side of the blades makes this configuration very problematic for RANS methods, even using LT models, since accurate capturing of the transition location is critical for predicting integral characteristics. For instance, RANS was unable to correctly reproduce the effect of total pressure loss observed in the experiment, so scale-resolving simulations had to be performed, in which sufficiently accurate results were obtained. Details on these simulations can be found in [5].

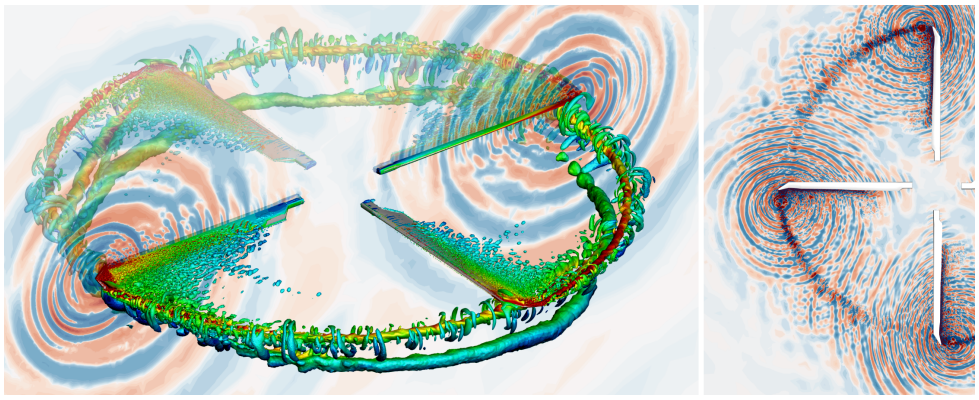
Due to the lack of computational resources, we usually consider only a section of a blade in a linear cascade with periodic conditions when performing a series of simulations. An entire blade has been simulated on meshes up to 150 million nodes so far, which is rather coarse. For accurate simulations of entire blades, meshes with more than a billion of nodes are required.

Modeling aerodynamics and aeroacoustics of helicopter rotors and drone propellers is shown in Fig. 5. In this case, RANS approaches are inapplicable for predicting broad-band aerodynamic noise. This requires high-fidelity scale-resolving simulations. Meshes of about 100 million nodes per blade are needed for resolving relevant flow structures. In our simulations, meshes of up to 400 million nodes have been used so far. More information, including validation and comparison of RANS and DES results, can be found in [3, 4].

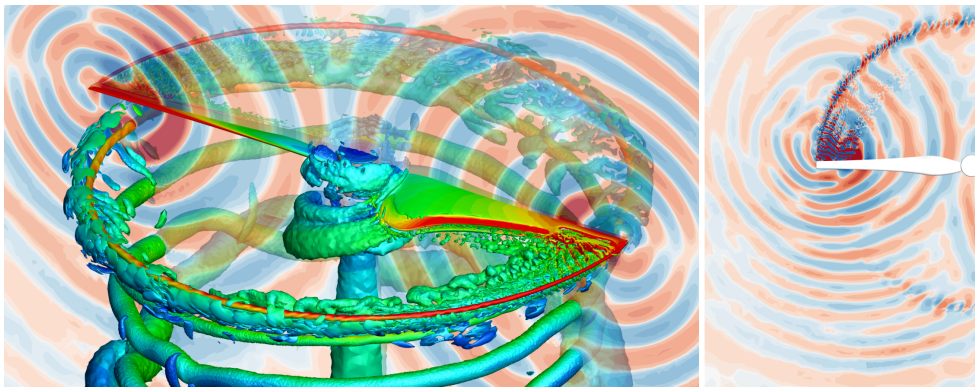


(a) Q-criterion isosurfaces[1] (b) density field in middle section

Figure 4. Flow in a low pressure turbine with LT transition present on the suction side



(a) Helicopter rotor



(b) Drone propeller

Figure 5. Simulation of rotors: turbulence (Q-criterion) and acoustics (time derivative of pressure)

Currently available supercomputer resources allow modeling only separate fragments of aircrafts. We typically use meshes of several hundred million nodes for industrial applications yet (for meshes of the order of a billion nodes, only test runs were performed to demonstrate operability and robustness). The demonstrated parallel efficiency, as well as the inherent potential of multilevel parallelism and the absence of scalability constraints, suggest that future exaflop supercomputers will allow us to use meshes dozens of times more detailed and simulate such complex configurations as an entire aircraft.

[1] $Q = 0.5(\|\Omega\|^2 - \|S\|^2)$, where Ω and S are the vorticity and strain rate tensors, respectively.

Acknowledgements

This work was supported by Moscow Center of Fundamental and Applied Mathematics, Agreement with the Ministry of Science and Higher Education of the Russian Federation, No. 075-15-2019-1623. Results in Section 3 were obtained within the RSF project 19-11-00299. The research is carried out using the equipment of the shared research facilities of HPC computing resources at Lomonosov Moscow State University [17], the equipment of Shared Resource Center of KIAM RAS (http://ckp.kiam.ru). The authors thankfully acknowledge these institutions.

References

1. Abalakin I., Bakhvalov P., Kozubskaya, T.: Edge-based reconstruction schemes for unstructured tetrahedral meshes. International Journal for Numerical Methods in Fluids 81(6), 331–356 (2016), https://doi.org/10.1002/fld.4187

2. Bakhvalov, P.A., Kozubskaya, T.K.: EBR-WENO scheme for solving gas dynamics problems with discontinuities on unstructured meshes. Computers & Fluids 157, 312–324 (2017), https://doi.org/10.1016/j.compfluid.2017.09.004

3. Bobkov, V., Abalikin, I., Kozubskaya, T.: Simulation of Helicopter Rotors On Unstructured Mixed Meshes Using Edge-Based Reconstruction Schemes. WCCM-ECCOMAS2020 (2020). https://doi.org/10.23967/wccm-eccomas.2020.308

4. Bobkov, V., Gorobets, A., Kozubskaya, T., et al.: Supercomputer Simulation of Turbulent Flow Around Isolated UAV Rotor and Associated Acoustic Fields. RuSCDays 2021, CCIS 1510 (2021). https://doi.org/10.1007/978-3-030-92864-3_20

5. Duben, A.P., Kozubskaya, T.K., Marakueva, O.V., et al.: Simulation of flow over high-lifted turbine cascade at low Reynolds numbers. Journal of Physics: Conference Series 1891, 012018 (2021). https://doi.org/10.1088/1742-6596/1891/1/012018

6. Fröhlich, J., von Terzi, D.: Hybrid LES/RANS methods for the simulation of turbulent flows. Progress in Aerospace Sciences 44(5), 349–377 (2008). https://doi.org/10.1016/j.paerosci.2008.05.001

7. Gorobets, A.V., Bakhvalov, P.A., Duben, A.P., et al.: Acceleration of NOISEtte Code for Scale-Resolving Supercomputer Simulations of Turbulent Flows. Lobachevskii Journal of Mathematics 41(8), 1463–1474 (2020), https://doi.org/10.1134/S1995080220080077

8. Gorobets, A.: Parallel Algorithm of the NOISEtte Code for CFD and CAA Simulations. Lobachevskii Journal of Mathematics 39(4), 524–532 (2015), https://doi.org/10.1134/S1995080218040078

9. Gorobets, A., Bakhvalov, P.: Improving Reliability of Supercomputer CFD Codes on Unstructured Meshes. Supercomputing Frontiers and Innovations 6(4), 44–56 (2020). https://doi.org/10.14529/jsfi190403

10. Guseva, E.K., Garbaruk, A.V., Strelets, M.K.: Assessment of Delayed DES and Improved Delayed DES Combined with a Shear-Layer-Adapted Subgrid Length-Scale in Separated Flows. Flow, Turbulence and Combustion 98, 481–502 (2017), https://doi.org/10.1007/s10494-016-9769-7

11. Heinz, S.: A review of hybrid RANS-LES methods for turbulent flows: Concepts and applications. Progress in Aerospace Sciences 114, 100597 (2020). https://doi.org/10.1016/j.paerosci.2019.100597

12. Mockett, C., Haase, W., Schwamborn, D. (eds.): Go4Hybrid: Grey Area Mitigation for Hybrid RANS-LES Methods. Springer International Publishing (2018). https://doi.org/10.1007/978-3-319-52995-0

13. Pont-Vílchez, A., Duben, A., Gorobets, A., et al.: New strategies for mitigating the gray area in delayed-detached eddy simulation models. AIAA Journal pp. 1–15 (2021). https://doi.org/10.2514/1.j059666

14. Shur, M.L., Spalart, P.R., Strelets, M.K., et al.: An enhanced version of DES with rapid transition from RANS to LES in separated flows. Flow, Turbulence and Combustion 95(4), 709–737 (2015). https://doi.org/10.1007/s10494-015-9618-0

15. Trias, F.X., Folch, D., Gorobets, A., et al.: Building proper invariants for eddy-viscosity subgrid-scale models. Physics of Fluids 27, 065103 (2015), https://doi.org/10.1007/s10494-016-9769-7

16. Trias, F.X., Gorobets, A., Silvis, M.H., et al.: A new subgrid characteristic length for turbulence simulations on anisotropic grids. Physics of Fluids 29(11), 115109 (2017). https://doi.org/10.1063/1.5012546

17. Voevodin, Vl., Antonov, A., Nikitenko, D., et al.: Supercomputer Lomonosov-2: Large Scale, Deep Monitoring and Fine Analytics for the User Community. Supercomputing Frontiers and Innovations 6(2), 4–11 (2019), https://doi.org/10.14529/jsfi190201