ME489-HW1

Batuhan Ülkütaşır

19 March 2024

1 Introduction

This paper have been written to demonstrate proficiency in working at LATEX environment as a part of an undergraduate course. All the written content is based on the article ”HORSES3D: A high-order discontinuous Galerkin solver for flow simulations and multi-physics applications.” [1] Published at Computer Physics Communications in 24 February 2023.

2 Abstract and Overview

This paper presents a summary of the latest developments of High-Order Spectral Element Solver (HORSES3D), an open source high-order discontinuous Galerkin framework, capable of solving a variety of flow applications, including compressible flows with or without shocks (compressible Navier-Stokes equations), incompressible flows (incompressible Navier–Stokes equations), various RANS and LES turbulence models, particle dynamics, multiphase flows, aeroacoustics and the Cahn–Hilliard equation and entropy–stable variants are solved. A comparison between high-order solvers widely adopted by industry and academia to simulate fluid flows having comparable capacity with HORSES3D is also included. Regarding the overflow points in the paper, a very detailed documentation and the test cases included in the GitHub repository are discussed. Its innovative approach to solving complex fluid dynamics problems, integrating multiple flow applications into a single framework, sets it apart from other solvers in the field. HORSES3D is a Fortran 2008 object-oriented solver, originally developed to solve the compressible Navier-Stokes equations using DG (high order discontinuous Galerkin) discretisations and explicit time marching.

2.1 Performance Comparison

A range of high-order academic open-source solvers exist and are briefly summarized in Table (1), where main capabilities are presented. Considering discretisation type, memory versatility and wide range of unique capabilities HORSES3D performs excellently compared to other open-source solvers. It is important to state that commercial products are excluded from the list.

Table 1: Summary of high-order solvers and capabilities.

	HORSES3D	Nek5000	Nektar++	Semtex	deal.II	Flexi/Fluxo	Trixi.jl	PyFR
References		[11]	[12,13]	[14]	[15]	[16,17]	[18,19]	[20]
Language	Fortran	Fortran	C++	C++	C++	Fortran	Julia	Python
Spatial discretisation	DG	CG	CG/DG	CG	DG	DG	DG	FR
Incompressible	✓	✓	✓	✓	✓	✗	✗	✓
Compressible	✓	✗	✓	✗	✓	✓	✓	✓
Energy/Entropy stable	✓	✗	✗	✗	✗	✓	✓	✗
p-adaptation	✓	✗	✓	✗	✓	✓	✗	✓
Steady/Unsteady	S/U	U	S/U	U	S/U	U	U	S/U
LES	✓	✓	✗	✓	✗	✓	✗	✓
RANS	✓	✓	✗	✗	✗	✓	✗	✓
Multiphase	✓	✗	✗	✗	✓	✗	✗	✓
Shocks	✓	✗	✓	✗	✓	✓	✓	✓
Particles	✓	✗	✓	✗	✓	✓	✗	✓
Aeroacoustics	✓	✓	✗	✗	✓	✓	✗	✗
Immersed Boundaries	✓	✗	✓	✗	✓	✓	✓	✗
Parallelisation	✓	✗	✓	✗	✓	✗	✗	✗
Shared/Distributed-Memory	S/D	D	D	D	S/D	D	S/D+GPU	S/D+GPU

2.2 Brief Explanation of Numerical Method

The nodal elements in physical domain is tessellated with non-overlapping curvilinear hexahedral elements, e, which are geometrically transformed to a reference element, $el.$ This transformation is performed using a polynomial transfinite mapping that relates the physical coordinates $\overrightarrow{x}$ the local reference coordinates ξ. The transformation is applied to (1) resulting in the following:

$$Ju_t + \nabla_\xi \cdot \overset{\leftrightarrow}{\tilde{F}}_e = \nabla_\xi \cdot \overset{\leftrightarrow}{\tilde{F}}_e \tag{1}$$

where J is the Jacobian of the transfinite mapping, ∇_ξ is the differential operator in the reference space and $\overset{\leftrightarrow}{\tilde{F}}$ are the contravariant fluxes for each hexahedral element. After integrating the equation in (1) to obtain weak form, first by integration parts then coupling with Riemann fluxes $\overset{\leftrightarrow}{\tilde{F}}{}^*_e$ a simplified volume form obtained in equation (2):

$$\int_{el} Ju_t \cdot \phi_j + \int_{\delta el} \overset{\leftrightarrow}{\tilde{F}}{}^*_e \cdot \overset{\leftrightarrow}{\tilde{F}}_e \phi_j - \int_{el} \overset{\leftrightarrow}{\tilde{F}}_e \cdot \nabla_\xi \phi_j = \int_{el} \nabla_\xi \cdot (\overset{\leftrightarrow}{\tilde{F}}_v) \cdot \phi_j \tag{2}$$

As briefly mentioned in the overview part, HORSES3D can operate on aeronautics by using a direct approach and acoustic analogies. In particular, the Ffowcs-Williams and Hawkings (FWH) aeroacoustic analogy is implemented. As a part of this analogy a brief set of equations regarding the acoustic pressure $p = p - p_0$ can be obtained from the density fluctuation, as $p = c_0^2 \rho$ in the far field (where $\frac{\rho}{\rho_0} << 1$).

$$4\pi p_T = \int_S \left[\frac{(1 - M_0 i R_i)\dot{Q}_i n_i}{R^*} \right]_{\tau_e} dS - \int_S \left[\frac{U_{0i} n_i \hat{R}_i^* Q_i}{R^{*2}} \right]_{\tau_e} dS, \tag{3}$$

$$4\pi p_L = \int_S \left[\frac{\dot{L}_{ij} n_j \hat{R}_i}{c_0 R^*} \right]_{\tau_e} dS + \int_S \left[\frac{\dot{L}_{ij} n_j \hat{R}_i^*}{R^{*2}} \right]_{\tau_e} dS, \tag{4}$$

In the equations (3) & (4) where p_T is the thickness pressure fluctuation, p_L is the loading pressure fluctuation, c_0 is the speed of sound, M_0 is the free-stream Mach number, U_0 is the free-stream velocity, R is the phase radius, R^* the amplitude radius, and the hat $\hat{()}$ of both R and R^* represents the partial derivatives of the radius quantities. The subscript τ_e indicates that the integrals are calculated at the emission retarded time.

2.3 Adaptation

The truncation error adaptation algorithm performs mesh p-adaptation for steady and unsteady simulations. Regarding in this topic, In Fig.(1) an example of a p-adapted mesh based on truncation errors is shown. There are more examples of truncation error-based adaptation for the compressible Navier-Stokes equations where the contours indicate an average polynomial order.

Moreover, the adaptation process inherently involves a critical decision-making step where the user must select an appropriate target truncation error threshold for the adapted mesh. This selection is pivotal as it directly influences the quality and reliability of the simulation outcomes. Historically, the choice of this threshold has been somewhat contentious, primarily because it lacks a straightforward physical or engineering interpretation, making it a notable drawback of the method.

2.4 Solver Efficiency

To reduce the computational cost, all techniques are parallelized for hybrid systems to run on multiprocessor CPUs combining MPI and OpenMP (Shared Memory). To run explicit test on numerical methods the TGV problem (Taylor-Green Vortex) has been widely used to report the subgridscale modeling capabilities of ILES approaches and discretisations. In the Fig.(2) and Fig.(3) the TGV case is simulated in a 64^3 mesh with polynomial or-ders of three to six, giving approximately 16 to 90 million of degrees of freedom. They use Gauss–Lobatto interpolation points and Pirozzoli's split–form, with Roe (convective) and BR1 (diffusive) fluxes. The number of total cores ranges from one to 700, and the number of OpenMP threads when enabled is 48. Simulations were handled in the MareNostrum-3 supercomputer.

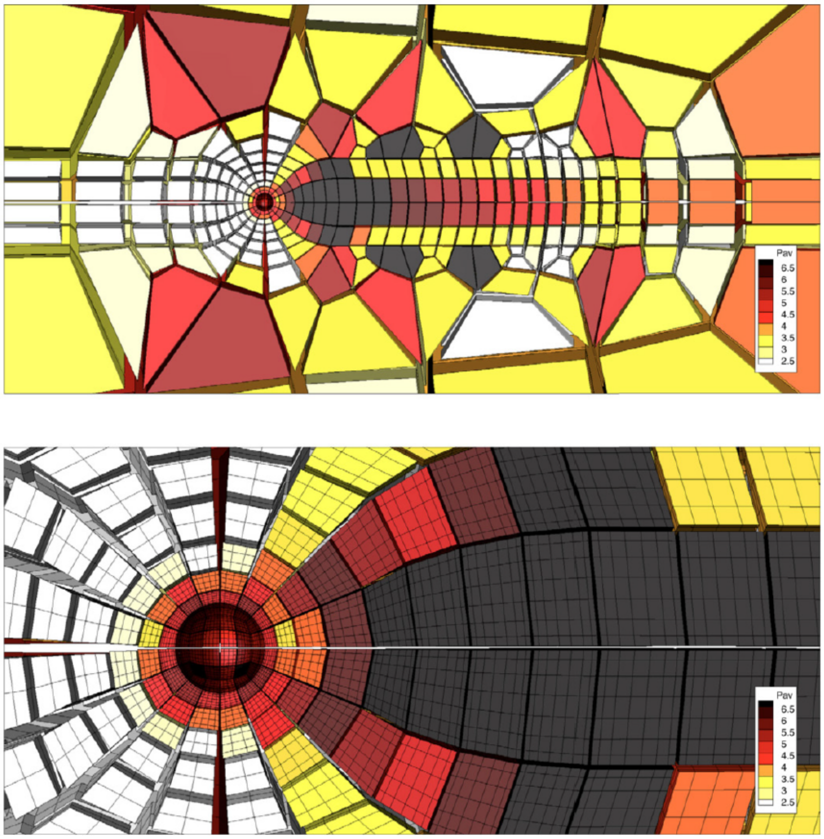
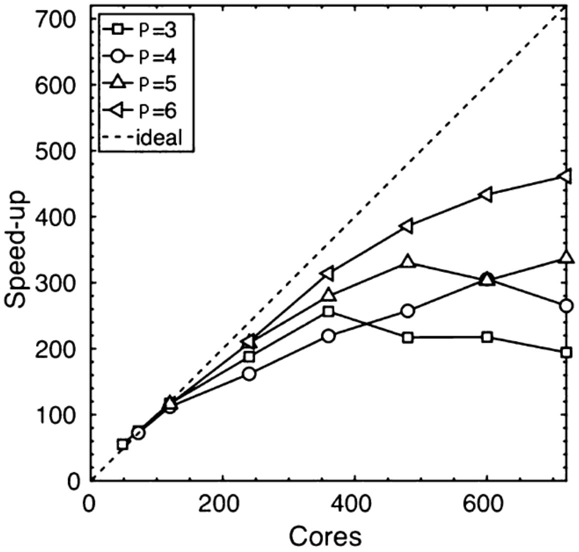
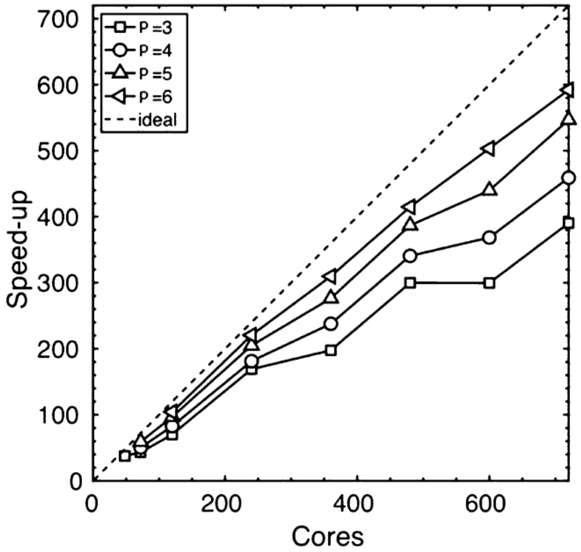


Figure 1: Example of a p-adapted mesh for the flow around a sphere at Reynolds Number 200. The contours indicate the average polynomial order ($P_{av} = (P_x + P_y + P_z)/3$). Figure at the bottom shows zoomed view near the sphere wake region. Retrieved from [1].



(a) MPI parallelization

Figure 2: HORSES3D scalability solving the TGV problem with a 64^3mesh using MPIn. Retrieved from [1].



(b) Hybrid MPI+OpenMP parallelization

Figure 3: HORSES3D scalability solvingthe TGV problem with a 64^3 mesh using MPI/OpenMP. Retrieved from [1].

3 Conclusions and Future Plans

HORSES3D is an open-source parallel framework for the solu-tion of non-linear partial differential equations. The solver allows the simulation of compressible flows (with and without shocks), incompressible flows, RANS and LES turbulence models, particle dynamics, multiphase flows, and aeroacoustics. The numeri-cal schemes provide fast computations, especially when selecting high-order polynomials and MPI/OpenMP combinations for a large number of processors. Entropy-stable schemes, local anisotropic p-adaptation, and efficient dual time-stepping and multigrid allow the solver to tackle problems of industrial relevance, such as air-crafts and wind turbines.

HORSES3D is being constantly updated, with the incorporation of neural networks to accelerate simulations and the enabling of connections to external libraries through Python interfaces. Efficiency improvements and expanded simulation capabilities, such as fluid-structure interactions, are being pursued through integration with solvers like MFEM, enhancing both the speed and breadth of applications.

References

[1] Esteban Ferrer, Gonzalo Rubio, Gerasimos Ntoukas, Wojciech Laskowski, OA Mariño, S Colombo, Andrés Mateo-Gabín, H Marbona, F Manrique de Lara, David Huergo, et al. : A high-order discontinuous galerkin solver for flow simulations and multi-physics applications. *Computer Physics Communications*, 287:108700, 2023.