Journal of Physics:
Conference Series

PURPOSE-LED PUBLISHING™

PAPER • OPEN ACCESS

Systematic Assessment of DGFEM Accuracy for Supersonic Flow

To cite this article: Mohamed Ezzat Abd ElMohsen *et al* 2025 *J. Phys.: Conf. Ser.* **3070** 012012

View the article online for updates and enhancements.

This content was downloaded by mohamed.mab from IP address 154.186.120.120 on 07/08/2025 at 07:28

Systematic Assessment of DGFEM Accuracy for Supersonic Flow

Mohamed Ezzat Abd ElMohsen[1*], Tamer Heshmat Mohamed Aly Kasem[2,3] and Mohamed Madbouly Abdelrahman[1]

[1] Aerospace Engineering Department, Cairo University, Giza, Egypt
[2] Smart Engineering Systems Research Center (SESC), Nile University, Giza, Egypt
[3] Department of Engineering Mathematics and Physics, Secondment from Cairo University, Giza, Egypt.

[*]E-mail: mohamed.202110817@eng-st.cu.edu.eg

Abstract. The modern High-Order Discontinuous Galerkin Finite Element Method (DGFEM) is considered. The method offers the advantages of the finite volume and the traditional finite element methods within a unified framework. However, the performance of the DGFEM may be affected for supersonic flow because of the limiters. Hence, a systematic assessment of the accuracy of the DGFEM is performed. The improved accuracy of the DGFEM is quantified and the benefits are highlighted.

1. Introduction

Supersonic flow is a significant phenomenon with diverse applications in aerospace, astrophysics, and geophysics [1]. Several closed-form solutions are available for one-dimensional and linearized assumptions [2]. However, numerical simulations offer the advantage of solving the nonlinear multidimensional governing equations and they are an efficient choice for most practical problems [3]. Nevertheless, numerical simulations of supersonic flow problems should be performed carefully, and various issues should be considered.

In numerical simulations, the problem domain is discretized by a numerical grid [4]. Cartesian grids offer simplicity and fast convergence. However, cartesian grids suffer limited capability because complex boundaries are not necessarily aligned to the numerical grid. This limitation could be handled using the immersed boundary technique [4] that was introduced more than five decades ago [5]. However, the immersed boundary results may suffer the presence of spurious oscillations near the boundaries, and this is an active research topic [6], [7].

Generally, unstructured boundary-fitted grids are the widespread choice to model flows in complex geometries [4]. Using the finite-difference method with these grids requires relatively more computational effort and complexity. Generally, the solution is calculated using FV in terms of the solution averaged over polygon-shaped cells. Hence, the geometric flexibility of the FV method is evident [8]. Furthermore, the solution algorithm is based on calculating the cell interface fluxes that satisfy the conservation laws and reproduce the problem physics. However, developing higher-order discretization on unstructured grids using FV is not straightforward [8]. Generally, the solution is obtained using the FE in terms of arbitrary order polynomials defined over polygon-shaped elements. Hence, the FE shares with the FV the advantage of geometric flexibility and offers the benefit of high-order discretization. However, several obstacles are faced

Content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

in case the traditional FE technique is applied to practical problems. Specifically, continuity across elements is imposed within the traditional FE. However, supersonic flow solutions may include high gradients or discontinuities, the FE method usually suffers from nonphysical oscillations [9]. Hence, stabilization methods are needed that include the Galerkin least squares method and shock capturing stabilization. However, the extension of these stabilization methods to practical problems is complicated and other alternatives are needed [9].

Based on the above discussion, it is natural to look for an intelligent combination of FE and FV methods. This combination leads to the Discontinuous Galerkin Finite Element Method (DGFEM) [8]. The DGFEM employs arbitrary order polynomials to obtain the solution. However, the continuity requirement of the FE is relaxed and discontinuity across elements is allowed. The concept of interface fluxes is borrowed from the FV and applied to DGFEM to connect elements.

Several modern studies focused on applying the DGFEM to supersonic flow problems [10], [11], [12] . However, the DGFEM is not widely adopted in commercial codes. Hence, the utilization of DGFEM in practical problems is limited. This may be attributed to the absence of systematic assessment of the benefits of the DGFEM.

Although the DGFEM offers several advantages, the accuracy of simulation of supersonic flow and shock waves needs to be clarified. Specifically, the limiters used to avoid spurious oscillations generally reduce the high-order accuracy near discontinuities and shock waves. Hence, the overall performance should be carefully assessed.

Towards this goal, the opensource MATLAB code “nodal-dg” is used in this study [13]. The code is based on the DGFEM method. The code is used to simulate the forward step problem. This problem is well documented and includes shock waves, reflection, and complex geometry. To assess the accuracy of the DGEFEM, the widely adopted methodology developed by [14] will be used.

2. Governing Equations

The governing equation for solving the Forward step using DG-FEM is Euler equations which describe the conservation of Mass, Momentum, and Energy for inviscid fluid in 2D [15]. The equation is defined $\boldsymbol{q}$, $\boldsymbol{F}_x$, and $\boldsymbol{F}_y$ vectors stand for the conserved variables, horizontal and vertical fluxes, respectively. The vectors are defined as follows:

$$\boldsymbol{q} = [\rho\,, \rho u\,, \rho v, E]^T, \tag{1}$$

$$\boldsymbol{F} = [\rho u\,, \rho u^2 + p\,, \rho uv, u(E+P)]^T, \tag{2}$$

$$\boldsymbol{G} = [\rho v\,, \rho uv\,, \rho v^2 + P, v(E+P)]^T. \tag{3}$$

The vectors in equations 1, 2,and 3 are defined in terms of the fluid density ρ, the horizontal velocity u, the vertical velocity v, the total energy per unit volume E, and the pressure P. The pressure is defined in terms of the conserved variables and the as specific heat γ as:

$$P = (\gamma - 1)\left(E - \frac{1}{2}\rho(u^2 + v^2)\right). \tag{4}$$

To simplify notation, the flux vector $\boldsymbol{f} = (\boldsymbol{F}, \boldsymbol{G})$ is defined. The Euler equations are written in the conservative form as:

$$\frac{\partial \boldsymbol{q}}{\partial t} + \nabla \cdot \boldsymbol{f} = 0. \tag{5}$$

Generally, the solution of the equation (5) is obtained for a domain Ω whose boundaries are denoted by ∂. The boundary conditions should be prescribed according to the problem physics. The wall boundaries should satisfy the no-penetration condition:

$$\boldsymbol{V} \cdot \hat{\boldsymbol{n}} = 0. \tag{6}$$

Here, **V** is the velocity vector and $\hat{\mathbf{n}}$ is a unit vector normal to ∂. For supersonic inlet conditions, the values of the components of $\boldsymbol{q}$ are specified. For supersonic exit conditions the components of $\boldsymbol{q}$ are obtained by extrapolation from the inside the domain.

2.1 Benchmark problem – Forward Facing step

The simulations will be focused on supersonic flow over a forward-facing step. This is a classic benchmark problem that includes complex flow features. These features are strong oblique shock waves, shock wave reflections, and intersections. The computational domain is defined as shown in Figure (1).

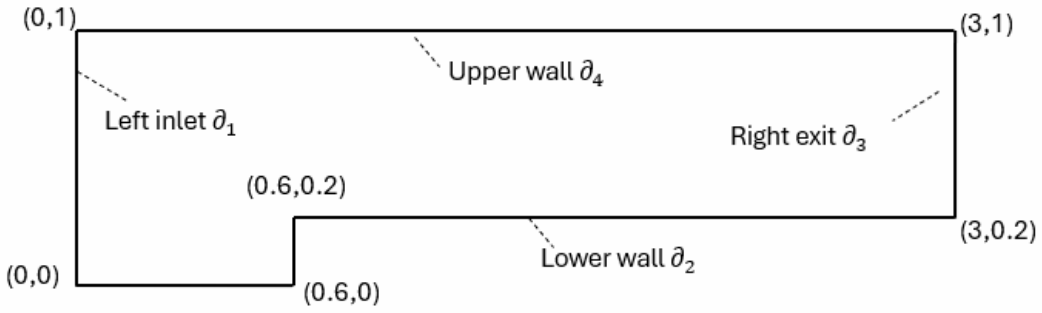

Figure 1. The domain of the benchmark problem, forward facing step.

The initial condition is defined as follows:

$$\begin{bmatrix} \rho = \gamma, \\ \rho u = 3\gamma \\ \rho v = 0 \\ E = \dfrac{1}{\gamma - 1} + \dfrac{9\gamma}{2} \end{bmatrix} \tag{7}$$

The value of γ=1.4 was adopted. Supersonic inflow conditions with a Mach number of 3 were applied at ∂_1. Outflow conditions were applied at ∂_3. Wall boundary conditions were applied at ∂_2 and ∂_4.

3. Methodology

The numerical solution of Eq. (5) will be obtained using the Discontinuous Galerkin Finite Element Method (DGFEM). The advantages of the DGFEM were discussed in the introduction.

These advantages are summarized in three points: High-order approximation, modelling complex boundaries, satisfaction of conservation, and accurate simulation of shock waves.

Each of these advantages is provided based on a specific algorithm within the DGFEM. Specifically, the high-order approximation is achieved through careful selection of local shape functions. The complex boundaries are described using systematic procedures for mesh generation and data structures. The conservation is implemented by employing flux formulas that satisfy the flow physics. Finally, shock waves are accurately modelled avoiding spurious oscillations. A brief description is provided for each of these algorithms. The material is based on the concise theoretical framework presented in [8].

3.1 DGFEM High-order Approximation

The problem domain Ω is divided into a set of non-overlapping triangular elements of total number K. The solution is approximated locally within each element using a high-order polynomial basis function. The approximate solution $u_h^k(\boldsymbol{x}, t)$ is defined locally at the element $k \in \{1, 2, \dots, K\}$ as a function in the space and time variables $\boldsymbol{x} = [x, y]$ and t, respectively.

For a polynomial of order N, the local solution is represented as:

$$\boldsymbol{q}_h^k(\boldsymbol{x}, t) = \sum_{i=1}^{N_p} \boldsymbol{q}_h^k(\boldsymbol{x}_i, t) l_i^k(\boldsymbol{x}). \tag{8}$$

Here, $l_i^k(\boldsymbol{x})$ is the two-dimensional Lagrange polynomial based on the grid points, $\boldsymbol{x}_i$, and N_p is the number of basis for a 2D polynomials of order N. The value of N_p is given by

$$N_p = \frac{(N+1)(N+2)}{2}. \tag{9}$$

The solution is defined in terms of the nodal values $\boldsymbol{q}_h^k(\boldsymbol{x}_i, t)$.

3.2 Mesh Generation

An unstructured mesh is generated to discretize the domain , and triangular elements are used. The mesh is defined and assembled using several matrices. The two parameters that are used to determine the matrices sizes are K (total number of elements) and the total number of nodes N_v. The matrices are listed in Table 1 along with their sizes, types, and description. These matrices are used to implement the solver building blocks (fluxes, numerical integration).

Table 1. Matrices used to define the grid**.** The sizes are determined in terms of the total number of elements K and the total number of nodes N_v.

Matrix title	Size	description	Data type (integer / real)	Storage (dense / sparse)
VX	$1 \times N_v$	x -coordinates of the nodes	real	dense
VY	$1 \times N_v$	y -coordinates of the nodes	real	dense
$ETOV$	$K \times 3$	mapping element to nodes	integer	dense
$ETOE$	$K \times 3$	mapping element to neighbour elements	integer	dense
$ETOF$	$K \times 3$	mapping element to neighbour faces	integer	dense
$FTOV$	$3K \times N_v$	mapping face to nodes	integer	sparse

3.3 Numerical flux

The Numerical flux is computed at the interface to capture the discontinuities. The choice of the numerical flux is essential to capture the discontinuities due to the shock waves and contact discontinuities. Harten-Lax-van-Leer (HLL) is used for solving the fluxes due to its ability to provide a stable approximation of the Riemann problem. The flux is computed by evaluating the fastest and slowest wave speeds. This is a good choice for problems involving strong shocks and discontinuities. The HLL flux does not suffer the disadvantages of other exact and approximate alternatives [16]. The flux f^* at interfaces between two states denoted as $u^-$for the left state and u^+ for the right state and it's defined as:

$$f^*(u^-, u^+) = \frac{S_R f_L - S_L f_R + S_R S_L (u_R - u_L)}{S_R - S_L}. \tag{10}$$

Here, $f(u^-)$ and $f(u^+)$ are the fluxes corresponding to the left and right states, respectively. The minimum and maximum wave speed are defined as

$$S_L = \min(u_L - c_L, u - c), \tag{11}$$

$$S_R = \max(u_R + c_R, u + c). \tag{12}$$

Here, subscripts *L* and *R* are used to denote the left and right states, respectively. The sound speeds in the left and right states are denoted by c_L and c_R, respectively. The Roe-averaged quantities [17] are denoted by the overbar. The sound speed *c* and the entropy *H* are defined as:

$$c = \sqrt{(\gamma - 1)\left(H - \frac{(u)^2 + (v)^2}{2}\right)}, \tag{13}$$

$$H = \frac{E + P}{\rho}. \tag{14}$$

3.4 Limiter

The high-order interpolation may generate spurious oscillations near shocks and discontinuities due to the Gibbs phenomenon. Hence, the limiter is introduced to avoid this type of oscillations. The limiter smooths gradients and ensures physical integrity by restricting excessive variations in conserved variables. It's essential to use a limiter in supersonic flow simulations because of the formation of the shocks and discontinuities can introduce numerical instabilities [18].

The limiter algorithm used in this study operates by first grouping each element with its three neighbours to form a patch, and a ghost element is introduced for the missing neighbours forming a boundary patch. A cell average is computed for the primitive variables (ρ, u, v, P) followed by averaging values across each vertex in each face, this will be calculated for each element and this averaging will be based on the primitive variable at each vertex and its neighbor. The gradients of these variables are then reconstructed using an area-weighted averaging technique to ensure stability and smooth transitions. Then the gradient of the conserved variables $(\rho, \rho u, \rho v, E)$ is reconstructed using the product rule. The first-order accurate-term Taylor Expansion of the conserved variables is computed. Finally, the reconstructed conserved variables are checked for unphysical values. This procedure is applied after each Runge-Kutta time step to prevent numerical instabilities while maintaining accurate shock capturing.

3.5 Numerical integration

The space discretization is obtained by substitution with equation 8 into equation 5. Next, the resulting equation is multiplied by a weight function $l_i^k(\boldsymbol{x})$ and integrated over the element k. The result is integrated by parts and written in the weak form.

$$\int_{D^k}\left(\frac{\partial q_h}{\partial t} l_i^k(\boldsymbol{x}) - \nabla l_i^k(\boldsymbol{x}) \cdot \boldsymbol{f}_h\right) d\boldsymbol{x} + \int_{\partial D^k} \hat{\boldsymbol{n}} \cdot \boldsymbol{f}_h^* \, l_i^k(\boldsymbol{x}) d\boldsymbol{x} \;=\; 0. \tag{15}$$

Here, D^k, ∂D^k, and $\hat{\boldsymbol{n}}$ denote the element region, element boundary, and unit outwards normal, respectively.

The time integration of the semi-discrete DGFEM equations is performed using explicit time-stepping schemes. The second-order Runge-Kutta (SSP–RK2) method is used for the integration of time due to its simplicity and effectiveness in handling stability constraints while maintaining accuracy. The semi-discrete form the governing equations can be written as:

$$\frac{dQ}{dt} = L(Q). \tag{16}$$

Here, Q represents the conserved variables, and $L(Q)$ is the spatial discretization operator, which includes the flux computations and the numerical solutions of the Euler equations. The time integration algorithm is used to advance the solution from $t^{(n)}$ to $t^{(n+1)} = t^{(n)} + \Delta t$. The SSP-RK2 method advances the solution in two stages as follows:

$$Q^{(1)} = Q^{(n)} + \Delta t L\left(Q^{(n)}\right), \tag{17}$$

$$Q^{(n+1)} = \frac{1}{2} Q^{(n)} + \frac{1}{2}\left(Q^{(1)} + \Delta t \, L\left(Q^{(1)}\right)\right). \tag{18}$$

Here, $Q^{(1)}$ is an intermediate value, and $Q^{(n)}$ and $Q^{(n+1)}$ stand for the solution at time steps $t^{(n)}$ and $t^{(n+1)}$, respectively. The time step Δt is determined by computing the maximum allowable time step based on the CFL conditions.

3.6 Accuracy Assessment

The Grid Convergence Index (GCI) is calculated to quantify numerical accuracy. A smaller GCI indciates lower uncertainity and higher accurraccy and vice versa. It is required to identify a flow feature ϕ to calculate the GCI. Three mesh configurations—coarse, intermediate, and fine—are required to evaluate the impact of mesh resolution and polynomial order on solution accuracy.

The procedure is based on the algorithm provided by [14]. Given the average mesh size for the three grids to be $h_1 < h_2 < h_3$, the algorithm proceeds as follows:

$$r_{21} = h_2/h_1, \tag{19}$$

$$r_{32} = h_3/h_2, \tag{20}$$

$$\varepsilon_{21} = \phi_2 - \phi_1, \tag{21}$$

$$\varepsilon_{32} = \phi_3 - \phi_2, \tag{22}$$

The Apparent order p is calculated using the following expression:

$$p = \frac{1}{\ln(r_{21})}\left|\ln\left|\frac{\varepsilon_{32}}{\varepsilon_{21}}\right| + \ln\left(\frac{r_{21}^p - s}{r_{32}^p - s}\right)\right|. \tag{23}$$

Here, $s = 1 \cdot sgn\left(\frac{\varepsilon_{32}}{\varepsilon_{21}}\right)$. Equation 23 should be solved iteratively. Finally, the GCI is calculated as

$$GCI = \frac{1.25\left|\frac{\phi_1 - \phi_2}{\phi_1}\right|}{r_{21}^p - 1} \tag{24}$$

4. Results and Discussion

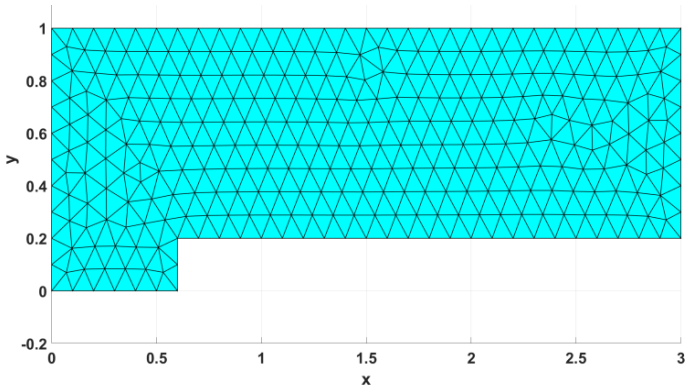

Figure 2. Sample for domain discretization using triangular elements.

The simulations were performed using three grids, coarse K=576, intermediate K=2266, and fine K=14168. The coarse grid is displayed in Figure 2. For each mesh, the simulations were performed for three discretization orders, $N = 1, 2,$ and 3. The simulations were conducted reaching the time $t = 4$. The flow field is displayed in Figures 3 and 4, for the coarse and fine meshes, and first-order discretization ($N = 1$).

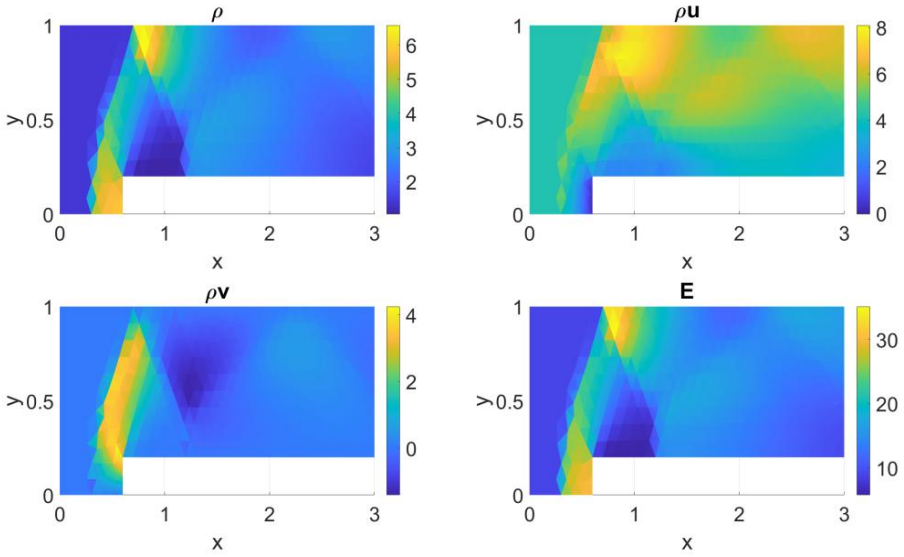

Figure 3. Flow field, coarse mesh, N=1, $t = 4$.

The flow field features agree with the results provided by [19]. These features include a curved shock wave at x~0.3 on ∂_2, an oblique shock reflected at x~1.3 on ∂_2 and x~2.4 on ∂_4. The flow features are relatively blurred smeared in Figure 3, due to the coarse mesh. However, the features are well resolved by the fine mesh in Figure 4.

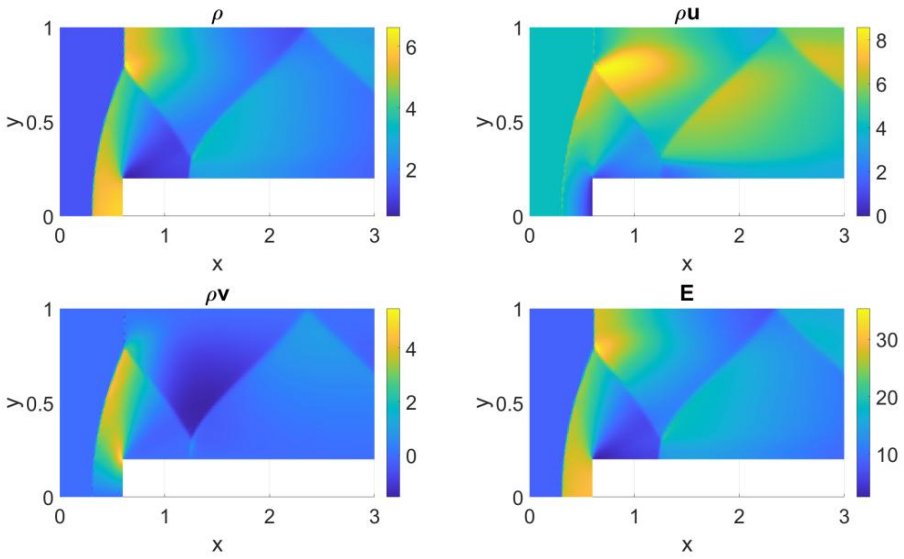

Figure 4. Flow field, fine mesh, N=1, $t = 4$.

4.1 Grid Convergence

The discretization order N was varied to conduct a parametric analysis. In the current study, the GCI will be based on the vertical pressure force $\phi = \mathcal{F}$ acting on ∂_2. The force $\mathcal{F}$ is defined as:

$$\mathcal{F} = \int_{\partial_2} P d\boldsymbol{x}. \tag{25}$$

The pressure distribution along the lower wall is shown in Figure 5. The results are provided for three meshes and first-order discretization ($N = 1$) and the fine mesh and $N = 3$. The results of the fine mesh are almost indistinguishable for $N = 1$ and $N = 3$. Hence, the overall accuracy of the simulations is further verified.

We can observe a sharp increase at $x = 0.3\ m$, corresponding to the formation of the shock wave due to the step. Another pressure rise is observed at $x = 1.3\ m$, resulting from the shock reflections interacting with the lower boundary wall. It is important to note that the results are montone and free from any spurious oscillations.

Table 2 presents the results, including the number of elements, computational time, vertical pressure force, apparent order (p), and GCI for each polynomial order N. The findings show that increasing mesh resolution leads to a lower GCI, indicating improved solution convergence. Furthermore, the apparent order p rises with N, confirming that higher-order polynomial approximations enhance accuracy.

On the other hand, the apparent order p is significantly lower than N. This result is expected since the high order accuracy is expected to degrade near discontinuities. This degradation occurs due to the effect of the limiter. However, the benefit of increasing N is clarified in terms of reducing the GCI. The results indicate a 50% reduction in the GCI for each increase in N. Hence, high-order DGFEM results should be useful for accuracy-demanding applications.

Table 2. Grid convergence Discretization order N, Case №Number of elements K.Computation time (seconds)Vertical pressure force $\mathcal{F}$, Apparent order p, and GCI

N	№	K	Computation time (seconds)	$\mathcal{F}$	p	GCI
	1	576	37	12.01817		
1	2	2266	137	11.65513	1.363	0.54%
	3	14168	3452	11.52225		
	4	576	189	12.01903		
2	5	2266	745	11.6294	1.692	0.32%
	6	14168	13130	11.52357		
	7	576	451	11.93982		
3	8	2266	3518	11.60935	2.044	0.14%
	9	14168	23610	11.54038		

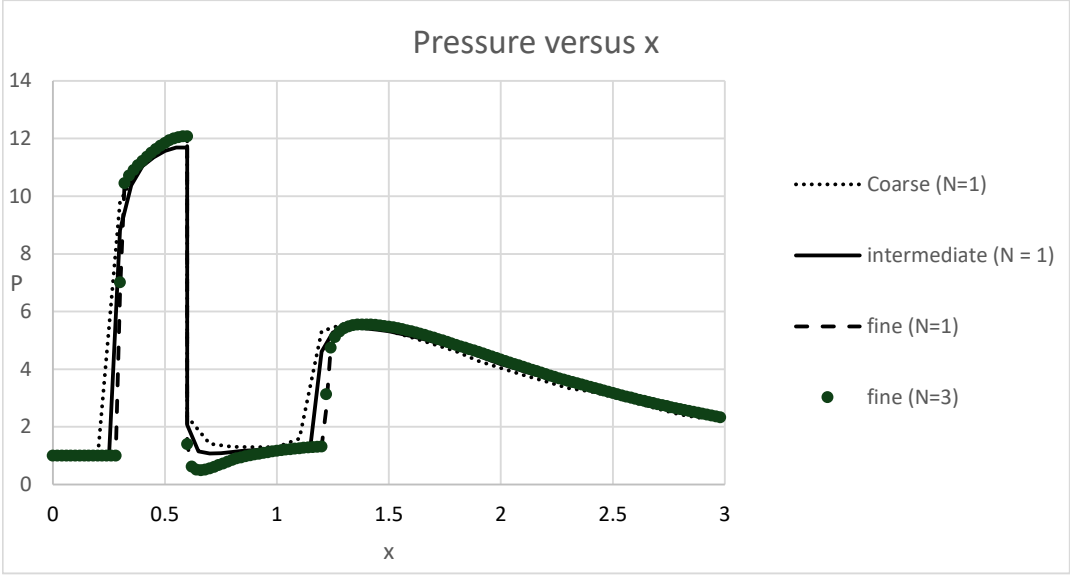

Figure 5. Pressure distribution versus x along ∂_2 at $t = 4$. The effects of grid refinement and discretization order is illustrated.

5. Conclusions

This study demonstrates the effectiveness of using the Discontinuous Galerkin Finite Element Method (DG-FEM) for simulating supersonic flow over a forward-facing step. The numerical study accurately captured the complex flow features such as curved shock waves and oblique shock reflections, while maintaining stability and high-order accuracy. The application of limiters is accurate and stable, especially in high-gradient regions where shock waves were formed.

The mesh is performed using different types of mesh (Coarse, Intermediate, fine). The fine mesh with higher-order polynomial approximation enhances shock resolutions and reduces numerical dissipation. The accuracy is assessed systematically using GCI. The results indicate a

50% reduction in the GCI for each increase in N. Hence, high-order DGFEM results should be useful for accuracy-demanding applications.

Overall, This research highlights the capability of DG-FEM as a powerful tool for solving supersonic flow simulation and it's able to accurately capture shock waves. Future work may explore the simulation of more complex geometries and industry-oriented applications.

References

[1] A. Achterberg, *Gas Dynamics, An Introduction with Examples from Astrophysics and Geophysics*. Paris: Atlantis Press, 2016. doi: 10.2991/978-94-6239-195-6.
[2] J. D. Anderson, *Modern compressible flow: with historical perspective*, Fourth edition. New York, NY: McGraw Hill, 2021.
[3] J. D. Anderson and C. P. Cadou, *Fundamentals of aerodynamics*, Seventh edition, International Student Edition. in McGraw Hill series in aeronautical and aerospace engineering. New York: McGraw Hill, 2024.
[4] J. H. Ferziger, M. Perić, and R. L. Street, *Computational methods for fluid dynamics*, Fourth edition. Switzerland: Springer Nature, 2020.
[5] C. S. Peskin, "Flow patterns around heart valves: A numerical method," *Journal of Computational Physics*, vol. 10, no. 2, pp. 252–271, Oct. 1972, doi: 10.1016/0021-9991(72)90065-4.
[6] A. Chaudhuri, A. Hadjadj, and A. Chinnayya, "On the use of immersed boundary methods for shock/obstacle interactions," *Journal of Computational Physics*, vol. 230, no. 5, pp. 1731–1748, Mar. 2011, doi: 10.1016/j.jcp.2010.11.016.
[7] H. Choung, V. Saravanan, and S. Lee, "Jump-reduced immersed boundary method for compressible flow," *Numerical Methods in Fluids*, vol. 92, no. 9, pp. 1135–1161, Sep. 2020, doi: 10.1002/fld.4821.
[8] J. S. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods*, vol. 54. in Texts in Applied Mathematics, vol. 54. New York, NY: Springer New York, 2008. doi: 10.1007/978-0-387-72067-8.
[9] V. Dolejší and M. Feistauer, *Discontinuous Galerkin Method: Analysis and Applications to Compressible Flow*, vol. 48. in Springer Series in Computational Mathematics, vol. 48. Cham: Springer International Publishing, 2015. doi: 10.1007/978-3-319-19267-3.
[10] D. Wang, Y. Lian, and H. Xiao, "Application of discontinuous Galerkin method in supersonic and hypersonic gas flows," *Computers & Mathematics with Applications*, vol. 80, no. 1, pp. 227–246, Jul. 2020, doi: 10.1016/j.camwa.2020.03.013.
[11] L. Pesch and J. J. W. Van Der Vegt, "A discontinuous Galerkin finite element discretization of the Euler equations for compressible and incompressible fluids," *Journal of Computational Physics*, vol. 227, no. 11, pp. 5426–5446, May 2008, doi: 10.1016/j.jcp.2008.01.046.
[12] L. Wei and Y. Xia, "Steady-state simulation of Euler equations by the discontinuous Galerkin method with the hybrid limiter," *Journal of Computational Physics*, vol. 515, p. 113288, Oct. 2024, doi: 10.1016/j.jcp.2024.113288.
[13] T. Warburton and J. Hesthaven, *nodal-dg*. MATLAB. [Online]. Available: https://github.com/tcew/nodal-dg
[14] "Procedure for Estimation and Reporting of Uncertainty Due to Discretization in CFD Applications," *J. Fluids Eng.*, vol. 130, no. 7, p. 078001, 2008, doi: 10.1115/1.2960953.
[15] E. F. Toro, *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*, 3rd ed. Dordrecht New York: Springer, 2009.
[16] G. Capdeville, "A high-order multi-dimensional HLL-Riemann solver for non-linear Euler equations," *Journal of Computational Physics*, vol. 230, no. 8, pp. 2915–2951, Apr. 2011, doi: 10.1016/j.jcp.2010.12.043.
[17] P. L. Roe, "Approximate Riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, vol. 43, no. 2, pp. 357–372, Oct. 1981, doi: 10.1016/0021-9991(81)90128-5.
[18] D. Kuzmin, "Slope limiting for discontinuous Galerkin approximations with a possibly nonorthogonal Taylor basis," 2012.
[19] A. U. Shirsat, S. G. Nayak, and D. V. Patil, "Supersonic Inviscid Flow Over 2D Forward-Facing Step Using RK3 + WENO and Finite-Difference Lattice Boltzmann Equation Method," in *Fluid Mechanics and Fluid Power (Vol. 1)*, S. Bhattacharyya and H. Chattopadhyay, Eds., in Lecture Notes in Mechanical Engineering. , Singapore: Springer Nature Singapore, 2023, pp. 93–98. doi: 10.1007/978-981-19-7055-9_16.