

Article

High-Fidelity Sliding Mesh Method for Moving Geometries in Turbomachinery Applications [†]

Junjie Ye ^{1,*}, Guglielmo Vivarelli ¹ , João Isler ¹ , David Moxey ², Spencer Sherwin ¹  and Chris Cantwell ¹ 

¹ Department of Aeronautics, Imperial College London, South Kensington Campus, London SW7 2AZ, UK; guglielmo.vivarelli12@imperial.ac.uk (G.V.); j.isler20@imperial.ac.uk (J.I.); s.sherwin@imperial.ac.uk (S.S.); c.cantwell@imperial.ac.uk (C.C.)

² Department of Engineering, King's College London, Strand Campus, London WC2R 2LS, UK; david.moxey@kcl.ac.uk

* Correspondence: jy21@ic.ac.uk; Tel.: +44-0742-1723-264

[†] This paper is an extended version of our paper published in the Proceedings of the 16th European Turbomachinery Conference (ETC16), Hannover, Germany, 24–28 March 2025, paper no. ETC16-319.

Abstract

The accurate, high-fidelity modeling of moving geometries involving translation and rotation is important in fluid mechanics applications, particularly turbomachinery. These simulations facilitate the analysis of various turbomachinery stages, where complex movements are integral to their functionality. We integrated a sliding mesh capability into the iLES/DNS compressible Navier–Stokes solver within Nektar++, which is a spectral/hp element open-source framework. Static and moving regions of the finite element mesh are coupled across a non-conformal interface using a point-to-point interpolation method (PTPM). The solver is verified on two canonical benchmarks, the isentropic vortex and Couette flow, and validated on the T106A low-pressure turbine cascade with an upstream translating bar that generates inflow wakes, emulating rotor–stator interaction. This paper aims to provide robust validation of the new solver developments and supply a tool to understand how the upstream bar motion influences the downstream blade efficiency, which is crucial for enhancing turbomachinery design and performance.

Keywords: spectral/hp method; point-to-point interpolation; sliding mesh; high-fidelity simulations; Nektar++



Academic Editor: Marcello Manna

Received: 5 February 2026

Revised: 19 June 2026

Accepted: 28 August 2026

Published: 1 October 2026

Copyright: © 2026 by the authors.

Published by MDPI on behalf of the European Turbomachinery Society.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY-NC-ND\)](https://creativecommons.org/licenses/by-nc-nd/4.0/) license.

1. Introduction

The design of machines with rotating or oscillating components often presents substantial engineering challenges, especially due to the high costs and technical complexities involved in conducting large-scale experimental tests in wind tunnels. Consequently, computational fluid dynamics has become an essential tool for supporting the design process, allowing for more efficient and cost-effective design optimization and performance evaluation. However, it is numerically challenging to discretely represent the relative motion of the different geometric components, such as their rotation and translation, to be able to study their transient effects on the fluid flow. A well-established method to address this issue is the sliding mesh approach, which subdivides the computational domain into multiple regions that undergo rigid transformations while maintaining domain connectivity. As shown in Figure 1, the blue region undergoes motion, whereas the black region is stationary.

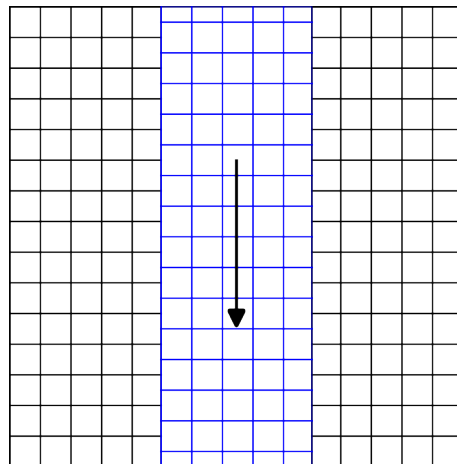


Figure 1. Schematic of the non-conformal translating mesh. The arrow indicates the direction of motion of the translating (blue) region.

The sliding mesh (SM) problem is typically addressed using the Arbitrary Lagrangian–Eulerian (ALE) formulation, which incorporates mesh movement [1]. Various solutions have been developed to handle the movement of the mesh. Zhang et al. conducted a series of studies employing the flux reconstruction and mortar element methods to facilitate sliding mesh simulations [2,3]. Furthermore, they introduced a transfinite mortar element method based on the flux reconstruction method, aiming to improve simulation accuracy and reduce numerical errors [4]. In addition to the flux reconstruction method, the use of the Discontinuous Galerkin (DG) method has also been explored for addressing sliding mesh problems. In 2012, Ferrer and Willden [5] developed an algorithm to integrate hanging node contributions into the computational scheme for non-conformal rotating mesh, which is based on the DG method and Interior Penalty method to perform spatial discretization. In 2014, Kang et al. [6] proposed an alternative sliding mesh technique that ensures mesh continuity through variable-node elements, focusing on fluid–solid interaction problems, particularly large structural motions. Dürrwächter et al. [7] employed the DG method and the mortar element and focused on parallel computation aspects. For representing curved interfaces, Pezzano et al. [8] introduced the use of Non-Uniform Rational B-Spline representation in conjunction with the DG method. Targeting rotor–stator interactions (RSIs), Sandberg et al. [9] implemented an SM solver using a characteristic interface condition for DNS of low-pressure turbine blades, Gao [10] developed a high-order spectral difference mortar-based SM solver for similar applications, and Ying et al. [11] utilized sliding meshes to capture broadband noise in turbofan RSI. Michelassi et al. [12] applied the solver of Sandberg et al. [9] to the effect of incoming wakes on the T106A cascade, the configuration adopted for validation in this paper.

Non-conformal interfaces are required to accommodate the relative motion between sliding-mesh regions. Numerous interpolation techniques have been devised to handle these non-conformal interfaces efficiently. The mortar method, a commonly employed approach for addressing these issues, is extensively detailed in reference [13]. This method creates an intermediary conformal interface on which variables are projected and integrated before reprojection onto the original non-conformal mesh. Re-meshing techniques offer an approach to update the mesh during simulations [14]. Another strategy is to use moving least-squares approximations to formulate shape functions for interface elements [15]. These strategies typically result in increased computational expense. Within the Nektar++ framework, two techniques were compared by Laughton et al. [16]: the mortar element method and the point-to-point direct interpolation method (PTPM). The investigation suggested that the mortar element method is effective in non-conformal two-dimensional

meshes. However, the construction of mortar elements in three-dimensional settings is more challenging due to the complexity of the mesh, the alignment of these elements, and the intricacies involved in flux calculation. In comparison, the PTPM exhibits good robustness in three-dimensional cases although, due to dealiasing requirements, it potentially requires more computational resources than mortar elements. The PTPM bears similarities to the Chimera method described by Galbraith et al. [17], which locates the donor cell by a comparable search and evaluates its local polynomial. The two differ in how the exterior state is then formed. The Chimera method projects the located values onto a modal representation before the flux is evaluated, whereas the PTPM uses the interpolated values directly at the interface quadrature points.

Most of the sliding mesh techniques discussed above rely on explicit time-integration schemes. The implementation of implicit solvers remains relatively limited in this domain, highlighting a significant gap. In computational simulations, explicit time-integration schemes are often favored for their simplicity of implementation. However, explicit schemes are constrained by the Courant–Friedrichs–Lewy (CFL) condition, which imposes stringent limitations on time step sizes, particularly in high-order methods [18]. In contrast, implicit time-integration schemes provide enhanced stability, allowing for larger time steps and reducing the strict constraints imposed by the CFL condition. Consequently, an implicit approach improves simulation efficiency while mitigating the risk of runtime instabilities caused by local nonlinear flow-field evolution [19]. Among the available implicit approaches, the diagonally implicit Runge–Kutta (DIRK) scheme has gained popularity due to its relatively straightforward implementation [20]. Furthermore, the singly DIRK scheme, as illustrated in [21], employs identical diagonal coefficients, enabling the utilization of the same linearization in the preconditioning process for all sub-stages [22]. Building on these advantages, Yan et al. [19] have incorporated the singly DIRK scheme into the Nektar++ framework.

The primary objective of this paper is to validate an implicit compressible sliding mesh (SM) methodology and its implementation within a high-order spectral/hp element framework. Such an approach has significant potential for the accurate simulation of complex turbomachinery configurations [23]. A comparative study is performed against the incompressible immersed boundary method (IBM) solver developed by Cassinelli [24]. This paper extends our earlier conference contribution presented at the 16th European Turbomachinery Conference (ETC16) [25] by incorporating fully three-dimensional simulations and a significantly expanded turbulence analysis. These enhancements provide a more comprehensive validation of the methodology and enable a deeper evaluation of its robustness and capability. Section 2 summarizes the mathematics of the SM solver. Section 3 presents the simulation setup, detailed results, and performance evaluation. The paper concludes by assessing the effectiveness of the proposed SM approach, discussing its limitations, and outlining directions for future research.

2. Methodology

This paper's contribution is an extension of the compressible Navier–Stokes solver framework in Nektar++ to support implicit SM simulations. Implicit SM methods remain relatively limited in the current literature. Previous work by Laughton et al. [26] introduced a 2D explicit rotating SM solver, while Yan et al. [19] developed an implicit compressible solver for fixed mesh simulations. The novelty of this paper is the coupling of the SM treatment with the implicit solver framework and its extension to 3D translational sliding motion. This requires modifications to the advection term and the treatment of non-conformal sliding interface periodicity within the implicit time integration. The following mathematical sections are included for completeness and to clarify the governing equations,

spatial discretization, and implicit algorithm used in the SM solver implementation. The equations are presented in dimensional form. The simulations themselves are performed in nondimensional variables, with the reference quantities for each case given in Section 3.

Nektar++ is an open-source C++ platform and is designed to facilitate the implementation of high-order spectral/hp element methods [27,28], providing a series of libraries and core functionalities for constructing high-fidelity solvers. Here, high-fidelity refers to the use of a high-order spectral/hp element method that is intrinsically capable of resolving a wide range of flow scales with minimal numerical dissipation, making it well suited for under-resolved DNS or implicit LES. When the grid resolution does not suffice to capture all turbulent scales, the low numerical dissipation of the DG method captures dominant flow features while inherently functioning as an implicit subgrid-scale model, thus eliminating the need for explicit turbulence modeling.

2.1. Navier–Stokes Equations in ALE Form

The three-dimensional compressible Navier–Stokes equations in ALE form can be defined as

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{F}(\mathbf{u}) - \mathbf{F}(\mathbf{u}, \nabla \mathbf{u})) - \mathbf{v}_g \cdot \nabla \mathbf{u} = 0, \quad (1)$$

where $\mathbf{u}$ is the conservative variables vector, $\mathbf{F}(\mathbf{u})$ and $\mathbf{F}(\mathbf{u}, \nabla \mathbf{u})$ are the advection and viscous flux vectors, and $\mathbf{v}_g = [v_{g1}, v_{g2}, v_{g3}]^T$ is the grid velocity. These vectors are defined by

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E_t \end{bmatrix}, \quad \mathbf{F}(\mathbf{u})_i = \begin{bmatrix} \rho u_i \\ \rho u_i u_1 + p \delta_{1i} \\ \rho u_i u_2 + p \delta_{2i} \\ \rho u_i u_3 + p \delta_{3i} \\ (\rho E_t + p) u_i \end{bmatrix},$$

$$\mathbf{F}(\mathbf{u}, \nabla \mathbf{u})_i = \begin{bmatrix} 0 \\ \tau_{1i} \\ \tau_{2i} \\ \tau_{3i} \\ u_j \tau_{ij} - q_i \end{bmatrix}.$$

where ρ and p are the density and the pressure, respectively, $E_t = C_v T + u_k u_k / 2$ is the total energy, C_v is the specific heat at constant volume and T is temperature. The stress tensor is

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \mu \delta_{ij} \frac{\partial u_k}{\partial x_k}$$

where δ_{ij} is the Kronecker delta. The symbol μ represents the dynamic viscosity determined by Sutherland's law. The heat flux is $q_i = -k \partial T / \partial x_i$, where k denotes the thermal conductivity.

2.2. Discontinuous Galerkin Method in ALE Form

The fluid domain Ω is decomposed into non-overlapping subdomains, $\Omega = \bigcup_{n=1}^N \Omega_n$, where $\Omega_i \cap \Omega_j = \emptyset$. Considering the advection equation as the governing equation and employing the Galerkin method with test function ϕ , we obtain

$$\int_{\Omega_n} \phi_i \frac{\partial \mathbf{u}}{\partial t} d\Omega_n + \int_{\Omega_n} \phi_i \nabla \cdot \mathbf{F}(\mathbf{u}) d\Omega_n = 0 \quad (2)$$

In ALE form, the grid velocity is introduced and the Reynolds transport theorem can be written as

$$\begin{aligned} \frac{d}{dt} \int_{\Omega_n} \phi_i u dx &= \int_{\Omega_n} \phi_i \frac{\partial u}{\partial t} dx + \int_{\Omega_n} \frac{\partial \phi_i}{\partial t} u dx \\ &+ \oint_{\partial \Omega_n} \phi_i u v_n ds, \end{aligned}$$

where $\partial \Omega_n$ is the elemental boundary interface. $v_n = \mathbf{v}_g \cdot \mathbf{n}$ is the grid velocity in the normal direction. Applying the divergence theorem to Equation (2), the DG method in ALE form can be defined as

$$\begin{aligned} \frac{d}{dt} \int_{\Omega_n} \phi_i u dx - \int_{\Omega_n} \frac{\partial \phi_i}{\partial t} u dx - \int_{\Omega_n} \nabla \phi_i \cdot \mathbf{F}(\mathbf{u}) dx \\ + \oint_{\partial \Omega_n} \phi_i \mathcal{F}(\mathbf{u}^+, \mathbf{u}^-, \mathbf{n}, v_n) ds = 0 \end{aligned} \quad (3)$$

where $\mathcal{F}(\mathbf{u}^+, \mathbf{u}^-, \mathbf{n}, v_n) = \mathcal{F}_n(\mathbf{u}) - \mathbf{u} v_n$ is the numerical flux between elemental interfaces and $\mathcal{F}_n(\mathbf{u})$ is the numerical flux of advection term in the normal direction obtained from the Riemann solver.

Under the fundamental ALE relation,

$$\frac{d\phi_i}{dt} = \frac{\partial \phi_i}{\partial t} + \mathbf{v}_g \cdot \nabla \phi_i, \quad (4)$$

the velocity of the grid causes the base functions to move, which results in the substantial derivative of the base functions being eliminated. Therefore,

$$\frac{\partial \phi_i}{\partial t} = -\mathbf{v}_g \cdot \nabla \phi_i \quad (5)$$

By substituting the ALE relation (5) back into Equation (3), the DG method in ALE form can be rewritten as

$$\begin{aligned} \frac{d}{dt} \int_{\Omega_n} \phi_i u dx &= \int_{\Omega_n} \nabla \phi_i \cdot (\mathbf{F}(\mathbf{u}) - \mathbf{v}_g \mathbf{u}) dx \\ &- \oint_{\partial \Omega_n} \phi_i \mathcal{F}(\mathbf{u}^+, \mathbf{u}^-, \mathbf{n}, v_n) ds \end{aligned} \quad (6)$$

We now employ the quadrature integration rule with the approximate solution $\mathbf{u}^\delta$. In the DG method, the approximate solution is expanded in the basis Φ , and the test functions ϕ are chosen from the same polynomial space. Hence, for the DG method used here, Φ and ϕ denote the same set of basis functions.

$$\mathbf{u}^\delta(\mathbf{x}, t) = \mathbf{u}_0(\mathbf{x}, t) + \sum_{i=1}^{N_{\text{dof}}} \hat{u}_i(t) \Phi_i(\mathbf{x}),$$

In the present Nektar++ implementation, the solution is approximated using the MODIFIED expansion basis, which is a hierarchical modal basis used in the spectral/hp element formulation [29]. Considering a one-dimensional reference coordinate $z \in [-1, 1]$, the basis can be expressed as

$$\varphi_i(z) = \begin{cases} \frac{1-z}{2}, & i = 0, \\ \frac{1+z}{2}, & i = 1, \\ \frac{1-z}{2} \frac{1+z}{2} P_{i-2}^{(1,1)}(z), & 2 \leq i < P, \end{cases} \quad (7)$$

where $P_{i-2}^{(1,1)}(z)$ is the Jacobi polynomial of degree $i - 2$, and P denotes the number of modes in the one-dimensional expansion. For quadrilateral and hexahedral elements, the multidimensional expansion is formed by tensor products of the corresponding one-dimensional bases. For triangular and tetrahedral elements, Nektar++ uses the related collapsed coordinate modified bases.

Then, we substitute the approximate solution $\mathbf{u}^\delta$ in Equation (6) with N_Q quadrature points to obtain

$$\sum_{q=1}^{N_Q} \sum_{i=1}^{N_{\text{dof}}} \phi_i w_q J_q \Phi_i \frac{d\hat{u}_i}{dt} = \sum_{q=1}^{N_Q} w_q J_q \nabla \phi_i \cdot \hat{\mathbf{F}} - \sum_{q=1}^{N_Q} \phi_i^\Gamma w_q^\Gamma J_q^\Gamma \hat{\mathcal{F}}_i$$

The quadrature weights are represented by w_q on the q th quadrature point inside each element, while J_q is the grid metric Jacobian. On the element boundary, these two variables are denoted by w_q^Γ and J_q^Γ .

The discussion above focuses on the discretization of the advection term, where the main modifications for the SM solver are introduced. For the diffusion term, no special treatment is required for the SM solver. The standard interior penalty method is used, following the implementation described in [19].

2.3. Point-to-Point Interpolation

The point-to-point method performs direct interpolation between two sides of a sliding interface, accommodating both translational and rotational motion. In contrast to straight elements, parametric mapping, as seen in Figure 2, is used for curved elements before interpolation. The gradient descent method is used in Nektar++ to determine point positions on the curved interface. This employs a quasi-Newton search direction and a backtracking line search [16]. To improve computational efficiency, an R-tree structure is implemented, constructing bounding boxes around curvilinear edges to efficiently identify potential matching edges. Additionally, the algorithm enhances performance by reusing information from previous exchanges, prioritizing edges or elements successfully matched in the last iteration. Figure 3 illustrates the point-to-point interpolation on interfaces. Once the corresponding point is identified, the solution values at the target interface quadrature points are obtained via barycentric interpolation from the specific element, obviating the need for explicit mortar elements. These interpolated values are then used as external states to calculate the numerical flux at the sliding interface.

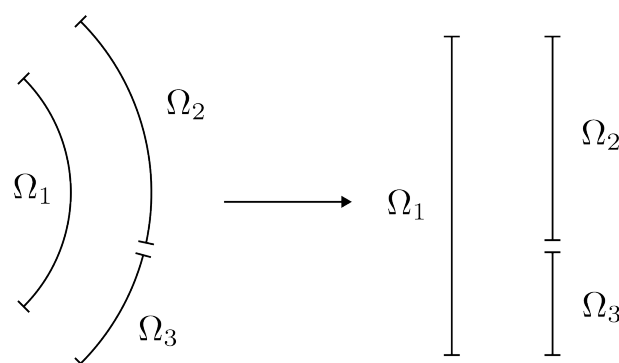


Figure 2. Parametric mapping of curved elements interface.

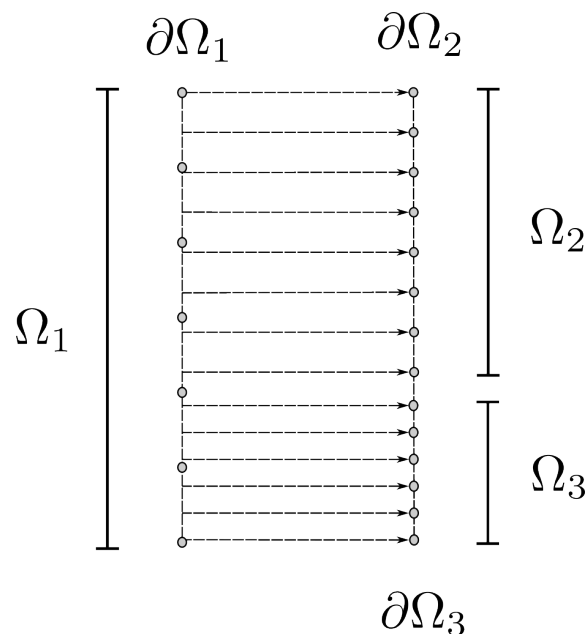


Figure 3. Schematic of point-to-point interpolation on interfaces, adapted from Laughton et al. [16], Figure 3.

2.4. Time Discretization

After applying the spatial discretization and quadrature integration rule, the governing Equation (1) is transformed into the following series of ordinary differential equations,

$$\frac{d\hat{\mathbf{u}}}{dt} = \mathcal{L}(\hat{\mathbf{u}}, t)$$

The time discretization employed in the implicit solver of Nektar++ is the singly DIRK method. Its general form [21] is

$$\begin{aligned}\hat{\mathbf{u}}^m &= a^{mm}\mathbf{E}^m + \sum_{i=0}^{m-1} a^{mi}\mathbf{E}^i + \mathbf{u}^n, \quad m = 0, 2, \dots, M-1 \\ \mathbf{E}^m &= \Delta t \mathcal{L}(\hat{\mathbf{u}}^m, t^n + c^m \Delta t); \quad c^m = \sum_{i=0}^m a^{mi} \\ \mathbf{u}^{n+1} &= \sum_{i=0}^{M-1} b^i \mathbf{E}^i + \mathbf{u}^n\end{aligned}$$

At the n^{th} time step and m^{th} stage, $\mathbf{u}^n$ and $\hat{\mathbf{u}}^m$ are the flow field solutions and approximation solutions. M denotes the total number of stages, and the Runge–Kutta scheme coefficients are given by the Butcher tableaux [21], which are represented by a , b , and c . At each time stage of the implicit scheme, the residual

$$\mathbf{R}(\hat{\mathbf{u}}^m) = \hat{\mathbf{u}}^m - \mathbf{S}_m - \Delta t a^{mm} \mathcal{L}(\hat{\mathbf{u}}^m, t^n + c^m \Delta t) = 0, \quad (8)$$

where

$$\mathbf{S}_m = \sum_{i=0}^{m-1} a^{mi} \mathbf{E}^i + \mathbf{u}^n,$$

is the source term and is defined to attain the m^{th} stage approximation solution $\hat{\mathbf{u}}^m$. Notably, if $a^{mm} = 0$, the time integration scheme is transformed to the explicit form. In Nektar++, singly DIRK schemes of orders 1 through 4 are implemented. However, this paper primarily

employs the second-order scheme. The corresponding Butcher tableaux are presented below [19]

$$\begin{array}{c|cc} \lambda & \lambda & \\ \hline 1.0 & 1.0 - \lambda & \lambda \\ ine & 1.0 - \lambda & \lambda \end{array} \quad \text{Order2: } \lambda = 1 - \frac{\sqrt{2}}{2}$$

The residual in Equation (8) is solved at each stage using the Jacobian-free Newton–Krylov approach of Yan et al. [19], and coupling it to a sliding mesh requires two modifications. First, the mesh is advanced once per DIRK stage, to the stage time $t^n + c^m \Delta t$, and is then held fixed while the Newton and Krylov iterations proceed. Each move applies the rigid-body transformation to the reference positions stored at $t = 0$, so the position of a moving zone depends only on absolute time. Second, the point-to-point interpolation of Section 2.3 is applied within every residual evaluation, immediately after the trace states are assembled. The interpolated exterior state therefore enters the Newton residual, the Jacobian–vector products formed by differencing that residual, and the preconditioner. The interpolation weights depend only on the relative position of the two sides, so they are constant within a stage. The operator differentiated by the Jacobian–vector product is then identical to the one in the residual. The interface coupling is consequently satisfied within the nonlinear solve at each stage rather than lagged between time steps. For a translating interface, a search coordinate may fall outside the extent of the opposite side. It is then wrapped back by whole zone lengths, so the two sides remain matched after an arbitrary distance of travel.

3. Results and Discussion

The verification cases presented in Sections 3.1 and 3.2, together with the 2D blade simulations, were previously reported in [25]. They are included here for completeness and to establish a baseline validation before presenting the extended 3D analyses introduced in this paper.

3.1. Advection of an Isentropic Vortex

This section examines the advection of an isentropic vortex in a computational domain with a translating mesh. The dynamics of the isentropic vortex follow the compressible Euler equations. All quantities in this section are nondimensional. The exact solution for the vortex is

$$u = \left(u_0 - \frac{\varphi e^f (y - y_0)}{2\pi} \right), v = \left(v_0 + \frac{\varphi e^f (x - x_0)}{2\pi} \right)$$

$$\rho = \left(1 - \frac{\varphi^2 (\gamma - 1) e^{2f}}{16\gamma\pi^2} \right)^{\frac{1}{\gamma-1}}, p = \rho^\gamma$$

where $f = 1 - (x - x_0)^2 - (y - y_0)^2$. Initially, the vortex is centered at $(x_0, y_0) = (5, 0)$ with a strength of $\varphi = 5.0$ and $\gamma = 1.4$. The prescribed advection velocity is $(u_0, v_0) = (1, 0.5)$. The vortex is advected without deformation, so at time t the profile above is centered at $(x_0 + u_0 t, y_0 + v_0 t)$. The computational domain, spanning $[0, 10] \times [-5, 5]$ with its initial condition, is illustrated in Figure 4. The computational domain is divided into two regions. The right (blue) domain undergoes translational motion with velocity $(v_{g1}, v_{g2}) = (0, \pi \cos(2\pi t))$, while the left (black) domain remains stationary. All boundaries are imposed with periodic boundary conditions.

To assess the convergence order of accuracy (OoA) of the solver, progressively refined meshes ranging from $[10 \times 10]$ to $[40 \times 40]$ elements are employed. A small time step corresponding to $\text{CFL} = 0.01$ is used to ensure that spatial errors dominate over temporal inaccuracies. Solutions are generated at $T = 0.2$ for polynomial orders $P = 1, \dots, 4$, and the

results are compared against the analytical solution to quantify spatial discretization errors. Figure 5 illustrates the corresponding error distributions as a function of the mesh size h . The observed OoA values are broadly consistent with the theoretical $P + 1$ convergence expected for smooth solutions with $P1$ – $P3$ close to the expected rates and the $P4$ case showing a higher apparent order over the present mesh range. This behavior confirms that the spatial discretization and sliding-mesh coupling recover the expected high-order convergence trend.

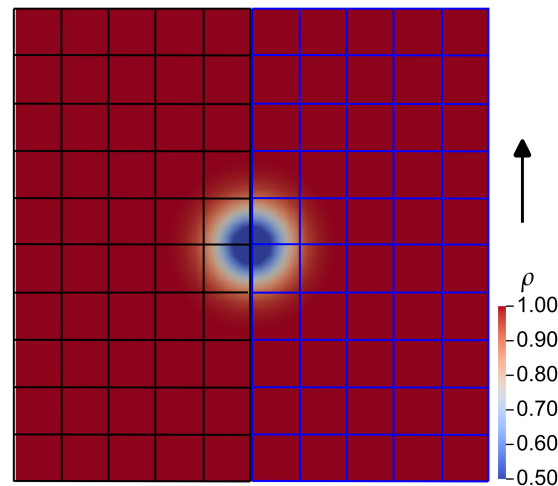


Figure 4. Mesh configuration and initial conditions of the isentropic vortex. The arrow indicates the direction of translation of the moving (blue) region.

3.2. Couette Flow

To evaluate the discrete geometric conservation law (DGCL) error, a low Mach number (Ma) Couette flow is simulated between two parallel plates. Figure 6 illustrates the computational domain. The domain is $[0, 1]^3$. The velocity prescribed at the wall $x = 1$ is $(0, V_\infty, 0)$, while the opposite wall at $x = 0$ is prescribed as $(0, 0, 0)$, as indicated by the green curved arrows. These boundary conditions impose the Couette velocity profile

$$u_2 = \frac{x}{H} V_\infty,$$

where H is the distance between the two walls. All other boundaries are treated as periodic. All quantities in this section are nondimensional. The length scale is the wall separation $H = 1$ and the velocity scale is the wall speed V_∞ . The Mach and Reynolds numbers, $Ma = 0.2$ and $Re = 100$, are formed on these quantities. The prescribed mesh motion is independent of the wall velocity. In the blue region, the mesh undergoes translational motion in both the y and z directions with grid velocity $(0, V_g, -V_g)$, where $V_g = 0, 2, 4, 6$ is used to test different relative sliding speeds. The norm of the ρu_2 error is calculated for polynomial orders $P = 2 \dots 5$ and then compared to the exact solution. The results, summarized in Figure 7, indicate a consistent preservation of DGCL accuracy across all cases. As illustrated, the error tends to decrease with increasing polynomial order, highlighting the benefit of higher-order methods in diminishing numerical inaccuracies. At the same time, the introduction of a nonzero translation velocity leads to a slight yet consistent increase in error compared with the stationary case ($V_g = 0$). However, this effect remains modest even at higher translation speeds. Over the entire simulation period $t \in [0, 1]$, the error remains nearly constant or experiences minimal fluctuations, indicating that any minor deviations introduced by mesh movement do not propagate significantly. Overall, these findings confirm that the implicit SM solver preserves DGCL accuracy under various mesh-translation scenarios.

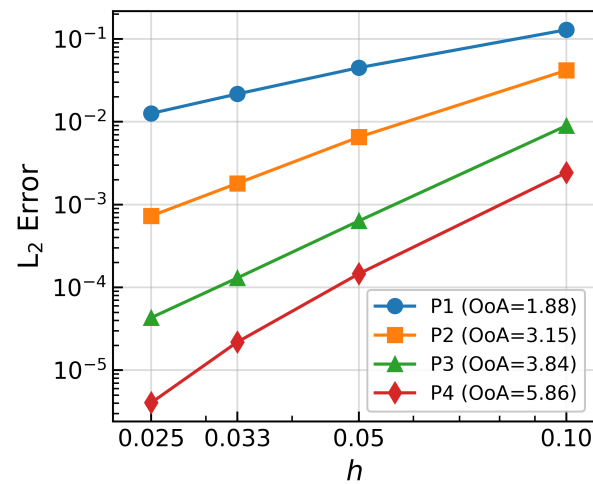


Figure 5. Investigation on convergence order of accuracy for advection of an isentropic vortex. OoA between the two finest mesh levels is indicated in the legend. The plotted error is the L_2 norm of the density error, and the mesh size is $h = 1/N$ for an $N \times N$ element mesh.

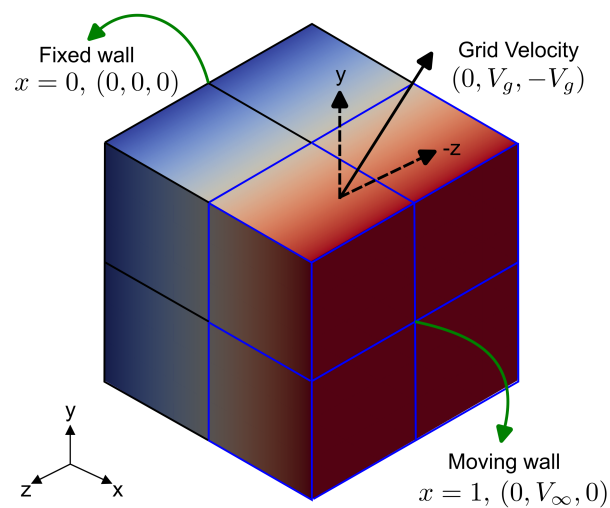


Figure 6. Mesh configuration and initial conditions of Couette flow.

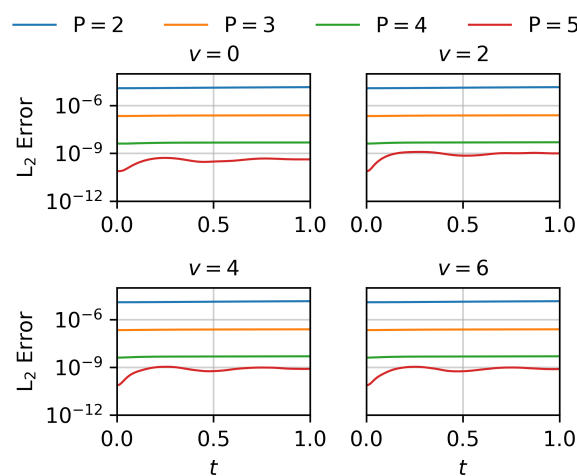


Figure 7. Errors for various translation velocity magnitudes (0,2,4,6) and polynomial orders $P = 2 \dots 5$ on Couette flow.

3.3. T106A Blade Simulation

3.3.1. 2D Simulations: Initial Validation

The solver is validated using the T106A low-pressure-turbine cascade, which is a configuration widely recognized in turbomachinery. Using Nektar++, the T106A blade has been extensively investigated by Cassinelli [24] under both clean inflow conditions and wake inflow generated using an immersed boundary method, demonstrating strong agreement with experimental results. Therefore, a similar setup is employed in this paper to validate the implicit sliding mesh solver. Figure 8 illustrates the setup involving moving bars positioned in front of the T106A blade to generate inflow wakes, representing the equivalent relative motion between rotor blades and fixed stators. The upstream mesh region containing the bar translates in the vertical direction (depicted in blue in Figure 8), while the blade geometry remains static. This configuration facilitates examining the transient flow interactions between the blade and the generated wakes. The 2D validation reported here was set up first. It follows the incompressible immersed boundary configuration of Cassinelli [24], so that the two solvers are compared on the same case. Its moving region is a narrow box around the bar, coupled to the static mesh across two sliding interfaces, with the bar at $x = -0.3535c$ (Figure 10). The 3D case described later uses a single interface to reduce the computational cost, with the moving region extended to the inflow boundary and the bar at $x = -0.7c$. Figure 8 sketches the latter arrangement.

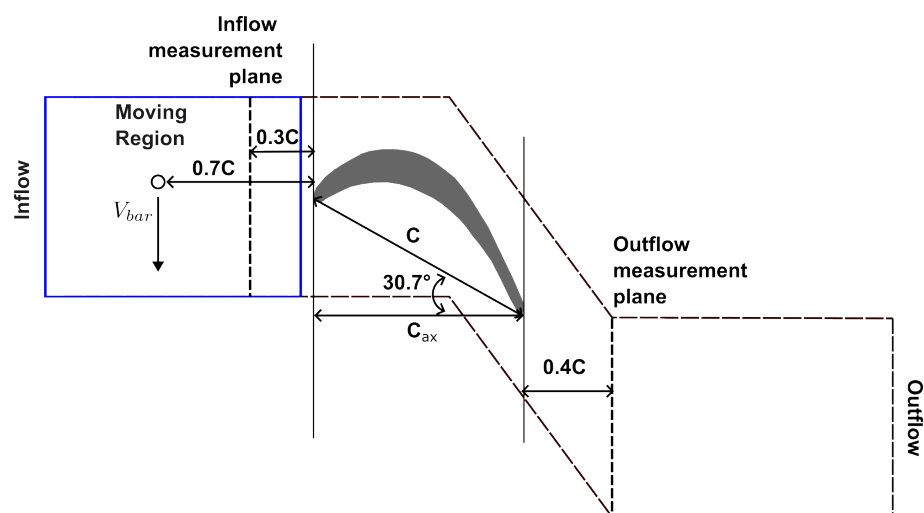


Figure 8. Sketch of moving bars and the blade geometry, drawn for the 3D configuration with a single sliding interface. The area denoted in blue indicates the moving mesh region. The 2D mesh of Figure 10 uses two interfaces and places the bar closer to the leading edge.

Using the actual chord length of $c = 198$ mm as a reference, the flow field is nondimensionalized to $c = 1$. Measurement planes are strategically placed in the flow field: the inflow measurement plane is located at $-0.3c$ upstream from the blade leading edge, and the outflow measurement plane is at $0.4c$ downstream from the trailing edge, enabling detailed flow dynamics to be captured at these critical sections. The boundary conditions for the simulation are configured as follows: the upper and lower edges of the domain are set to periodic. The inflow boundary is defined with far-field conditions, while the outflow is managed through pressure outflow conditions. All wall boundaries are set to be adiabatic, ensuring that no heat transfer occurs through these surfaces, simplifying the thermal dynamics within the simulation.

The initial validation focuses on a comparison between the compressible SM solver and the incompressible IBM solver using a 2D mesh under clean and wake inflow conditions. For the incompressible solver, the inlet conditions are set with a design inflow angle

of 37.7° and a Reynolds number of 50,000 based on the inflow velocity U_∞ and chord length of the blade [24]. The compressible solver maintains these flow parameters but incorporates an inlet Mach number of 0.0114. The mesh configuration used for the clean inflow case and the IBM solver is shown in Figure 9. It consists of 20,659 triangular elements and 3699 quadrilateral elements in 2D with the curved geometry represented using geometric order 3. With polynomial order $P = 4$, this corresponds to approximately 4.02×10^5 DoF. The computational mesh used for the compressible SM solver is shown in Figure 10. The translating region is extended to the blade leading edge with refinement concentrated in the wake-affected region to reduce the computational cost. This mesh consists of 15,467 triangular elements and 3839 quadrilateral elements in 2D, also using the geometric order of 3. With the same polynomial order $P = 4$, this corresponds to approximately 3.28×10^5 DoF.

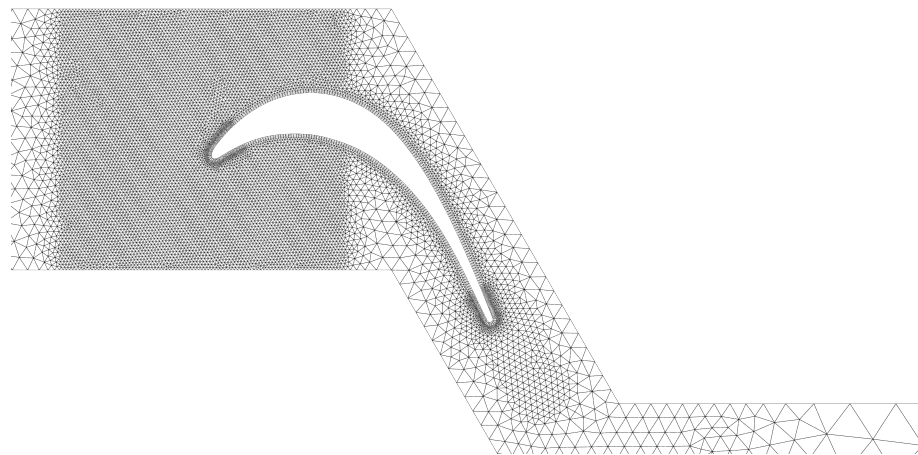


Figure 9. Computational mesh of the 2D clean inflow and the IBM configuration.

Under clean inflow conditions, both solvers are run without an imposed wake. Under wake inflow conditions, the implicit compressible SM solver setup (Figure 10) uses a translating bar to generate the inflow wake. In the incompressible IBM solver, the wake is induced by a shape function, which is defined as

$$\begin{aligned}
 & -\frac{1}{2} \left(\tanh \left(\frac{\text{rad}(x - X_c, y - (-3P_y + U_b \cdot t - 3 \cdot P_y \cdot z)) - r_b}{\xi} \right) \right. \\
 & + \tanh \left(\frac{\text{rad}(x - X_c, y - (-4P_y + U_b \cdot t - 3 \cdot P_y \cdot z)) - r_b}{\xi} \right) \\
 & + \tanh \left(\frac{\text{rad}(x - X_c, y - (-2P_y + U_b \cdot t - 3 \cdot P_y \cdot z)) - r_b}{\xi} \right) \\
 & \left. - 3.0 \right)
 \end{aligned}$$

where $z = \left\lfloor \frac{-3P_y/2 + U_b \cdot t}{3 \cdot P_y} \right\rfloor$, $X_c = -0.3535c$, $P_y = 0.799c$ is the bar pitch, $U_b = -0.95345U_\infty$ is the bar speed, the scale factor is $\xi = 5 \times 10^{-6}$, and $r_b = 0.0031c$ is the bar radius. The function $\tanh(x)$ denotes the hyperbolic tangent, and $\text{rad}(x, y)$ computes the polar radius $\text{rad}(x, y) = \sqrt{x^2 + y^2}$.

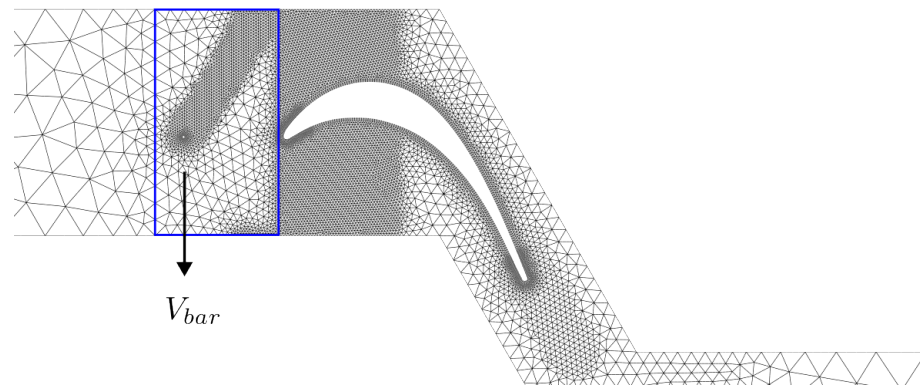


Figure 10. Computational mesh of the 2D wake inflow configuration. The area denoted in blue indicates the moving mesh region.

The pressure coefficient (C_p) distributions are compared with incompressible IBM results to validate the accuracy of the implicit compressible SM solver. The pressure coefficient is calculated based on the mixed-out average

$$C_p = \frac{p_w - p_{2,is}}{p_{t,1,is} - p_{2,is}}$$

Here, p_w denotes the wall static pressure. The quantities $p_{2,is}$ and $p_{t,1,is}$ denote the outlet static pressure and inlet total pressure, respectively, which are obtained by applying the isentropic relations to the mixed-out average quantities. The subscript “is” indicates quantities evaluated under the assumption of an isentropic process.

The isentropic Reynolds number and Mach number are defined as

$$Re_{is} = \frac{\rho_{is} u_{is} c}{\mu_{is}}$$

$$Ma_{is} = \sqrt{\left[\left(\frac{p_{t,1,is}}{p_{2,is}} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right] \frac{2}{\gamma-1}}$$

All statistics reported in this section are obtained by time averaging over eight convective time lengths (CTLs) after the initial transient has been discarded. Figure 11 displays the C_p distributions for the clean and wake inflow configurations. Both solvers produce similar C_p distributions along the axial chord of the blade. The close agreement between the compressible and incompressible solvers validates the numerical implementation and confirms that compressibility effects remain negligible under the given inflow conditions. However, the slight discrepancy near the leading edge and trailing edge in the compressible case indicates a possible influence of insufficient resolution. The matching results from the IBM and the SM solver demonstrate strong agreement, providing evidence for the accuracy of the SM solver.

The data in Table 1 compare the exit parameters obtained at inflow and outflow measurement planes for compressible and incompressible solvers under various inflow conditions applied to the T106A blade. All four computations impose the same design inflow angle of 37.7° at the inlet boundary. The α_1 values listed are mixed-out averages evaluated at the inflow measurement plane, $0.3c$ upstream of the leading edge and inside the blade potential field, not the angle imposed at the inlet. In the wake configurations the plane also sits within the bar wake. Small differences between the two solvers, and between the clean and wake configurations, are therefore to be expected. For clean inflow conditions, a minor variation was observed in the inflow angle (α_1), where the incompressible clean

inflow (ICI) recorded 37.7° compared to 38.1° for the compressible clean inflow (CCI). The outflow exit Re_{is} obtained is 88,353 for the ICI and 86,798 for the CCI, demonstrating a close agreement with a slight discrepancy of only 1.79%. Notably, both solvers yield an identical outflow exit angle (α_2) at 63.4° . In the wake inflow configuration, a comparison between the SM solver and the IBM solver revealed similar $Re_{2,is}$ values. A small difference was also noted in the α_1 , while the α_2 remained consistent across both solvers. This consistent performance in Reynolds number across different cases and solvers highlights the reliability of the implicit compressible SM solver in simulating RSI.

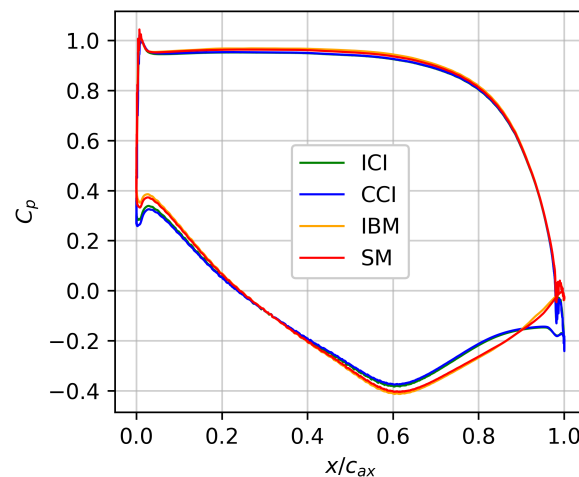


Figure 11. Wall pressure distribution for the 2D configuration under clean and wake inflow conditions.

Table 1. Inflow angle, exit isentropic Reynolds number, exit angle, and exit isentropic Mach number for clean inflow and wake inflow configurations. The exit isentropic Mach number is not defined for the incompressible solver.

	α_1	$Re_{2,is}$	α_2	$Ma_{2,is}$
ICI	37.7	88,353	63.4	-
CCI	38.1	86,798	63.4	0.021
IBM	36.9	88,081	63.3	-
SM	37.4	88,130	63.3	0.021

3.3.2. 3D Configurations

To further validate the solver, a 3D simulation is performed to demonstrate its capability for complex unsteady flow problems. Following Michelassi et al. [12], the simulations target exit $Re_{is} = 60,000$ and exit $Ma_{is} = 0.405$. The inlet flow angle is set to 46.1° , and the simulation yields an exit flow angle of approximately 63.3° consistent with the experimental measurements of Stadtmüller [30]. Periodic wakes are generated by a translating upstream bar with a diameter of $0.02c$, translation speed of $-0.41U_\infty$, pitch equal to the blade pitch ($0.799c$), and center position of $x = -0.7c$. As shown in Figure 12, to reduce the computational cost, a single sliding interface is employed rather than two, which also permits targeted refinement along the wake region. In addition to the wake region, the interface zone and the leading edge and trailing edge regions are refined to ensure that small-scale turbulent structures are resolved. The 3D mesh is obtained by extruding a 2D mesh comprising 22,134 triangular elements and 3512 quadrilateral elements. The curved blade and bar geometries are represented using the geometric order 3. The spanwise extent is 20% of the true chord length. A 20% extrusion produces results indistinguishable from those obtained with a 60% extrusion, and it is therefore sufficient for statistical convergence while reducing computational cost [31]. Using 24 planes in the spanwise extrusion

yields a total of 615,504 elements, with polynomial order $P = 3$; the configuration contains approximately 26.64 million DoF.

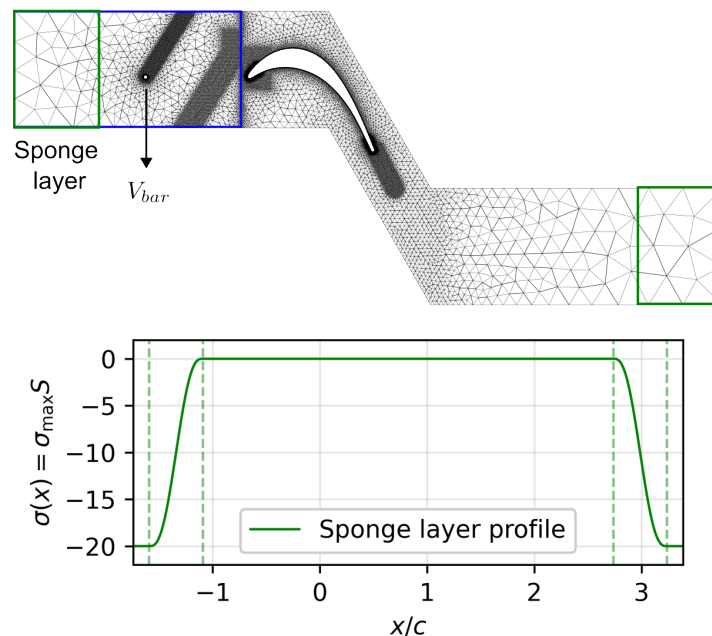


Figure 12. Computational mesh of 3D configurations. The area denoted in blue indicates the moving mesh region, and the green boxes mark the sponge layers. In the lower panel, the dashed lines mark the start and end of the inlet and outlet sponge zones.

Under this mesh configuration, the near wall resolution along the blade is reported in Figure 13. Over the suction side, the mesh resolution satisfies the DNS limit recommendation of Georgiadis et al. [32], namely $\Delta x^+ < 20$, $\Delta y^+ < 1$ and $\Delta z^+ < 10$. On the pressure side, the streamwise and wall-normal resolution also comply with maxima $\Delta x^+ \approx 11.8$ and $\Delta y^+ \approx 0.44$. A localized exceedance occurs at the leading and trailing edges where the spanwise resolution Δz^+ is larger than the nominal $\Delta z^+ < 10$ target. Increasing the number of spanwise planes from 24 to 32 removes this excess, and the predicted results remain unchanged. Overall, the wall-normal resolution lies within DNS limits, indicating that the current mesh provides DNS-quality near-wall resolution at reasonable computational cost.

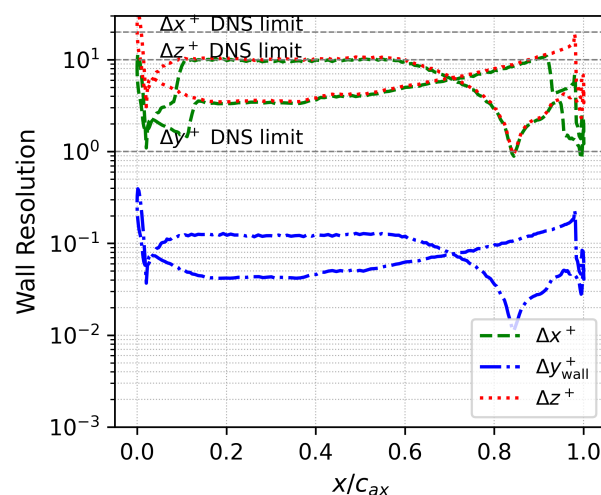


Figure 13. Near wall mesh resolution along the blade.

Considering the boundary condition, at the inlet, an entropy-velocity compatible Riemann boundary condition [33] is imposed. This boundary condition maintains a time-invariant streamwise velocity component at the boundary and prevents spurious variations caused by reflections. Consequently, the inflow angle is fixed at 46.1° . The outlet employs a pressure outflow condition. All walls are treated as adiabatic without heat transfer, and the remaining boundaries are periodic. To mitigate unphysical reflections, sponge layers are applied in $0.5c$ -wide zones adjacent to the inlet and outlet (blue regions in Figure 12). Throughout most of the domain, the damping is inactive and is activated smoothly only in the zones directly downstream of the inlet and directly upstream of the outlet. The sponge profile follows formulations used in jet aeroacoustics [34] and intake engine simulation [35] with maximum damping coefficient $\sigma_{\max} = 10U_\infty/L_{\text{sponge}}$ (where L_{sponge} is the layer width) and a fifth-order smooth-step function $S(x)$ to ensure sufficiently smooth damping. The reference state in the damping term is taken from a long-time-averaged, low-order 2D solution.

The airflow in the simulation is nondimensionalized using a reference temperature of $T_{\text{ref}} = 288 \text{ K}$, a reference density of $\rho_{\text{ref}} = 1.225 \text{ kg m}^{-3}$, and the inlet velocity U_∞ as the velocity scale. The implicit time-integration scheme enables the use of a nondimensional time step of $\Delta t = 5 \times 10^{-4}$, where time is scaled by the reference convective time c/U_∞ , corresponding to $\text{CFL} \approx 68$. The same time step was used for the T106A cascade in [36]. This relaxes the CFL constraint and improves computational efficiency in reaching the target final time. To obtain meaningful statistics, a three-stage procedure is followed. First, a 2D simulation is run for 20 CTLs, and the sponge layer reference state is formed by time averaging the final 10 CTLs. Second, the instantaneous 2D field is interpolated to 3D and advanced for 12 CTLs, which is executed in three segments of four CTLs to bypass start-up transients and establish the initial 3D turbulent state. Finally, the simulation is restarted from this state, and eight CTLs of time-averaged data are collected for analysis. In the present moving bar configuration, one bar pass takes approximately two CTLs. Therefore, the final eight CTLs used for statistical averaging include approximately four bar passes.

3.3.3. 3D Results

Because the inlet Reynolds and Mach numbers are not directly available, 2D tuning runs are performed to approach the target operating point. The 2D solution was iterated to an inlet condition with $Re = 37,000$ and $Ma = 0.238$. From the corresponding time-averaged fields, the resulting exit isentropic values are $Re_{\text{is}} = 61,045$ and $Ma_{\text{is}} = 0.407$, which are close to the target values. Figure 14 displays the C_p distributions for the 3D configuration together with the reference simulation and experimental results of Michelassi et al. [12]. To maintain consistency with Michelassi et al. [12], mass-averaged quantities are used on the measurement plane rather than mixed-out values. Scaling the bar passing frequency by the chord and the mass-averaged exit velocity gives a reduced frequency of $F_{\text{red}} = 0.308$ for the present configuration, against 0.31 for the reference case. Unlike Michelassi's setup, which imposed a 0.5% background turbulence intensity, the present simulations do not include background turbulence due to the computational cost. In this paper's configuration, the moving bar is located relatively far upstream of the blade. Although Nektar++ supports a source term-based synthetic turbulence generator [36], which allows turbulence to be introduced through a user-defined generator box rather than from a far upstream inlet, additional mesh refinement would still be required between the bar and the blade in the current setup. This capability has also been investigated for the T106A blade configuration and is important for more realistic simulations. However, the generator box is presently fixed in space. A possible future extension would be to allow the synthetic turbulence generator to move with the mesh, enabling turbulent inflow to be introduced more flexibly

in sliding mesh simulations. Since the operating condition is not reproduced exactly, discrepancies near the leading edge and trailing edge are expected. Nevertheless, the overall agreement is good, and from $x/c_{ax} = 0.6$ to $x/c_{ax} = 0.9$, the present simulation matches the experiment more closely. The wall shear stress distribution on the suction side (Figure 15) indicates separation at $x/c_{ax} = 0.84$, and reattachment occurs at $x/c_{ax} = 0.996$. This implies a longer bubble than reported by Michelassi et al. [12], where separation and reattachment were observed at $x/c_{ax} = 0.88$ and $x/c_{ax} = 0.94$, respectively. The difference is consistent with the absence of background turbulence in the present case, which tends to delay transition and prolong the separation bubble.

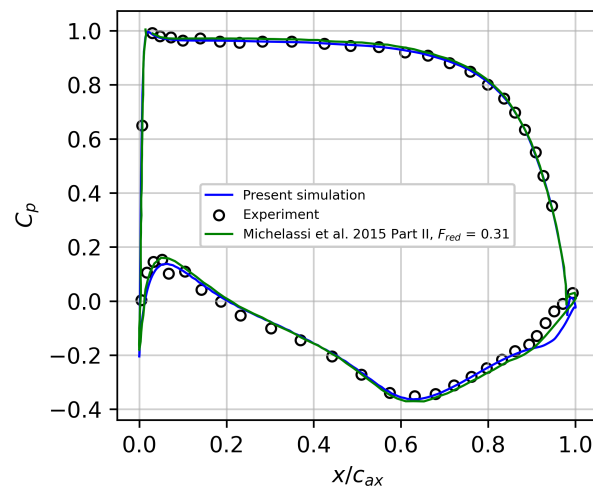


Figure 14. Wall pressure distribution for the 3D configuration, compared with the reference simulation and experimental data of Michelassi et al. [12].

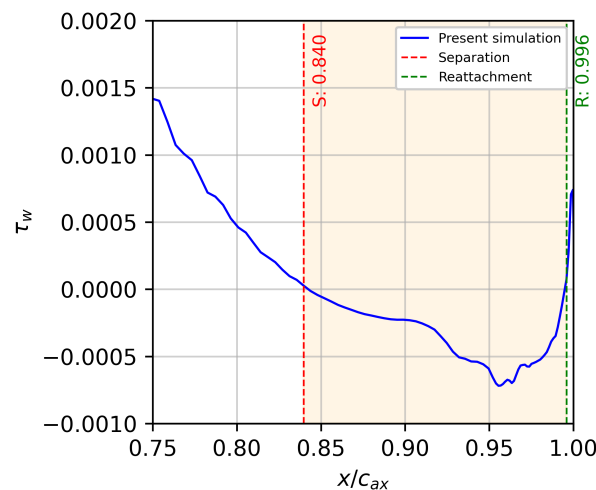


Figure 15. Wall-shear stress along the suction-side separation bubble. The plotted quantity is the surface tangential component of the wall shear stress, nondimensionalized by $\rho_{ref} U_\infty^2$.

To assess whether turbulence is adequately resolved, the streamwise velocity power spectral density (PSD) is analyzed at five probe points in the separation bubble and wake, as shown in Figure 16. The velocity contour in Figure 16 is nondimensionalized by the inlet velocity U_∞ . The PSD is computed using Welch's method: the eight CTLs time record is partitioned into two segments, Hann windows are applied with 50% overlap, and spectra are averaged over 25 spanwise samples at each probe to reduce variance. Frequencies are nondimensionalized as fc/U_∞ . The vertical dotted line in Figure 17 at $fc/U_\infty = 0.5$ marks the estimated wake passing frequency imposed by the upstream moving bars. The

spectra show enhanced low-frequency content near this frequency, suggesting that the separation bubble and wake respond to the imposed periodic incoming wakes. However, no sharply isolated peak is observed at this location, indicating that the wake passing excitation is broadened by its interaction with shear layer roll-up, transition, separation dynamics, and turbulent wake fluctuations. Across all probes in Figure 17, the spectra show a pronounced low-frequency peak associated with wake-passing and shear-layer roll-up, which is followed by broadband decay. At points 2–5, the mid-frequency decay aligns closely with the reference $-5/3$ slope, indicating an initial inertial subrange and well-developed turbulence in the wake. The downstream probes (4–5) retain higher energy over a wider band and roll off at slightly higher fc/U_∞ than the upstream probes (2–3), which is consistent with increased small-scale content and a more complete development of flow structures downstream. By contrast, points 1 and 2, located within the separation bubble, show lower spectral energy and an earlier high-frequency roll-off, reflecting weaker turbulence and stronger local dissipation. At the highest resolved frequencies, all spectra steepen appreciably beyond the $-5/3$ reference, which is attributable to numerical dissipation and finite spatial/temporal resolution. The spectra approach the noise floor well before the Nyquist limit, indicating that aliasing is unlikely. Overall, the $-5/3$ -type decay at the points confirms that the wake turbulence is resolved to a satisfactory degree.

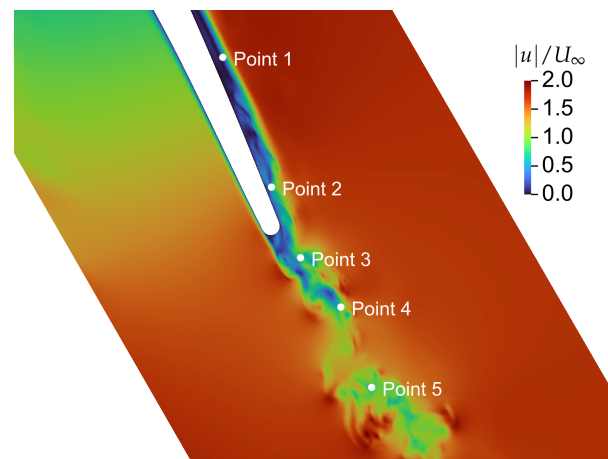


Figure 16. Instantaneous velocity magnitude, with the sampling point locations within the suction side separation bubble and the blade wake.

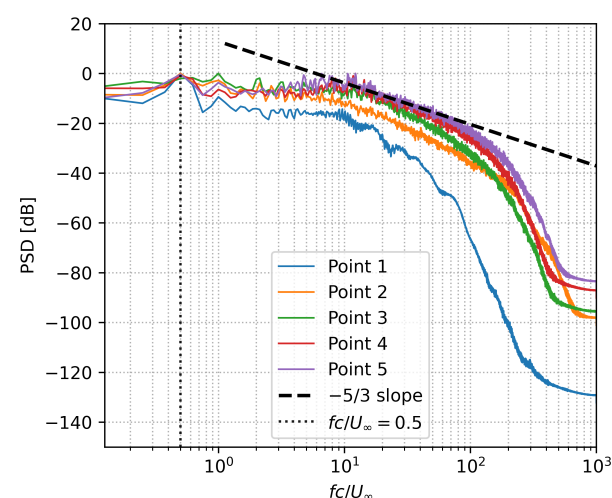


Figure 17. Power spectral density of the sampling points.

Complementary to the spectral analysis, the two-point autocorrelation [24] quantifies how velocity fluctuations remain correlated in the spanwise direction:

$$R_{ii}(\Delta z) = \frac{1}{N_z} \sum_{k=1}^{N_z} \frac{\langle u'_i(t, z_k) u'_i(t, z_k + \Delta z) \rangle_t}{\langle u'^2_i \rangle_{z,t}},$$

where u_i denotes the i -th velocity component and the fluctuation is $u'_i = u_i - \langle u_i \rangle_t$ with $\langle \cdot \rangle_t$ denoting a time average at a fixed spatial location. The coordinate z is the spanwise direction along which correlations are evaluated with discrete sample locations z_k for $k = 1, \dots, N_z$ and $N_z = 25$. Δz represents the separation between two positions along z .

The five probe points exhibit broadly similar behavior, so points 2 and 4 (Figure 18) are representative. The normalized spanwise two-point correlations are strongly anisotropic. At point 2 (nearer the trailing edge), the ordering $R_{22} > R_{11} \gg R_{33}$ holds over most separations, indicating the longest spanwise coherence in the pitch-wise component, intermediate coherence in the streamwise component, and the shortest in the spanwise component. This pattern is consistent with shear-layer dynamics in the suction-side separation bubble, where vortical roll-up produces elongated fluctuations in u_2 while it decorrelates u_3 rapidly. Farther downstream at point 4, R_{33} decays even faster and the negative correlation is weaker, indicating stronger mixing and the breakdown of spanwise flow structures with distance from the trailing edge. The contrast between R_{11} and R_{22} becomes less pronounced at intermediate separations. Taken together, all components drop steeply from unity to roughly $\Delta z/c \approx 0.01 - 0.02$, evidencing intense small-scale shear and limited spanwise coherence. Beyond this initial drop, the spanwise velocity correlation R_{33} crosses zero, doing so at approximately $\Delta z/c \approx 0.03 - 0.04$. The other components, R_{11} and R_{22} , remain positive over the plotted spanwise separation range. Overall, the results indicate a transition from a more organized, shear-dominated near wake to a more three-dimensional, mixed flow downstream.

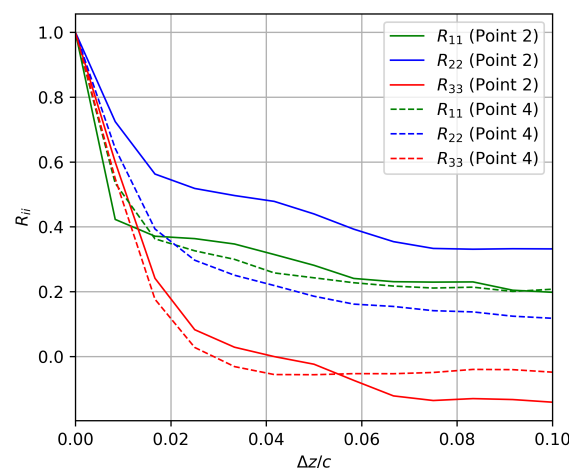


Figure 18. Two-point correlation function for probe point 2 and point 4.

3.4. Computational Performance

The purpose of this section is to assess the additional computational cost introduced by the compressible SM solver. To ensure a fair comparison, the fixed mesh and SM cases are run on the same 2D mesh with the same polynomial order and the same time step. Both cases are advanced for one convective time, $T = c/U_\infty$, using a nondimensional time step of 4×10^{-5} , again scaled by c/U_∞ . The simulations are performed on the Imperial College Research Computing Service using four compute nodes. Each node contains two Intel® Xeon® Platinum 8358 processors, giving a total of 256 CPU cores. The comparison therefore

isolates the overhead associated with the SM treatment, including the moving mesh update and sliding interface communication.

Table 2 demonstrates the computational cost of the SM solver, which consumes 6222 s CPU time in total for one convective time. During the solving process, the diffusion and advection terms take 1665 s and 664 s, respectively, leading to 3156 s for implicit solving, accounting for nearly 50% of the running time. Another significant time consumption is associated with the preconditioning process, which requires 2532 s. For the SM case, the time spent on interface interpolation and mesh movement is minimal—only 8 s.

Table 2. The computational wall-clock times of the SM solver.

	Time/s	Operation Count
Execute	6222	-
Diffusion Terms	1665	600,240
Advection Terms	664	600,240
Implicit Solve	3156	600,240
Preconditioning	2532	550,240
Interface	5	50,001
Mesh Movement	3	50,000

Comparatively, Table 3 presents the computational performance of the compressible solver under identical conditions but with a fixed mesh, contrasting with the SM solver. The total execution time for the compressible solver is 5433 s, so the SM solver requires approximately 14.5% more time. The fixed mesh configuration consistently demands less computation time across all operations. Notably, the advection terms in the fixed case use only 240 s. This significant reduction is primarily due to the absence of flux operations that are necessary for the SM solver to handle the advection terms. Despite fewer operations in the moving mesh configuration, it demands more time due to non-conformal interface interpolations that introduce additional points along the interface compared to the conformal configuration. This reveals the efficiency challenges faced by the SM solver. Consequently, increasing the polynomial order or decreasing the time step to enhance accuracy will lead to higher computational costs. Balancing the accuracy and computational efficiency of the SM solver is essential to optimize performance in practical applications.

Table 3. The computational wall-clock times of the compressible solver.

	Time/s	Operation Count
Execute	5433	-
Diffusion Terms	1523	651,067
Advection Terms	240	651,067
Implicit Solve	2394	651,067
Preconditioning	2467	601,067

4. Conclusions

This paper presents the development and implementation of a compressible implicit SM solver within the Nektar++ framework. The solver is designed to address critical challenges faced in turbomachinery simulations, particularly those involving moving components such as RSI. Using an implicit time integration scheme, the solver eliminates the limitations imposed by the CFL condition, allowing for larger time steps and enhanced stability. The solver was validated through comparison with an IBM, which is another commonly used approach for handling wake generation in computational fluid dynamics. These comparisons show a strong agreement between the two methods, indicating that the compressible implicit SM solver can accurately capture the fluid dynamics of the RSI.

Despite the successful implementation and validation, the compressible implicit SM solver has some limitations. One concern is the computational cost. As noted in the performance section, the solver requires approximately 14.5% more computational resources than fixed configurations. This increase is generally acceptable, but the overall computational cost will increase when considering more complex geometries and higher polynomial orders. The inclusion of inflow turbulence would align simulations more closely with experimental results but would further increase the computational cost, since additional mesh refinement would be required to resolve the incoming turbulent scales. Although implicit schemes offer the advantage of larger time steps and enhanced stability, PTPM requires more computational resources to handle the non-conformal interfaces, particularly in 3D cases. Another limitation of the current implementation is its inability to handle non-integer pitch ratios. In turbomachinery design, it is expected to encounter configurations where the stator and rotor do not have integer pitch ratios. All these restrictions may limit the flexibility and applicability of the SM solver in certain real-world turbine simulations.

Future work on the SM solver should focus on improving computational efficiency, extending inflow turbulence capability, demonstrating the method on annular configurations, and relaxing the current pitch ratio restriction. A first direction is to optimize the computational workflow through detailed profiling. Although the solver supports different element shapes, this flexibility can introduce load imbalance and MPI communication overhead, particularly for complex 3D meshes. Improving load balancing and communication efficiency would make the solver more practical for large-scale applications. A second direction is to include background inflow turbulence, either through additional free stream refinement or by developing a turbulence generator compatible with sliding mesh motion, similar to that presented in [36]. A third direction, now under way, is the extension to an annular stator–rotor configuration. Simulating a single sector rather than the full annulus would limit the computational cost. Finally, the solver should be extended to handle non-integer pitch ratios. Although PTPM with DG can in principle support such configurations, care is required because imposing periodicity between passages with different pitches may introduce non-physical errors. Future developments should therefore account for the relevant flow physics and carefully assess the accuracy and conservation properties of the interface treatment.

In conclusion, the compressible implicit SM solver implemented in Nektar++ provides a robust and accurate tool for simulating complex geometries undergoing relative motion. Validation on the T106A low-pressure turbine cascade confirms the fidelity of the solver and illustrates its suitability for turbomachinery applications. However, to fully realize its potential, further development is required to mitigate its computational cost and enhance its flexibility to handle a wider range of turbine geometries. Addressing these challenges will be crucial for expanding the applicability of the solver in industrial settings.

Author Contributions: Conceptualization, J.Y., D.M., S.S. and C.C.; methodology, J.Y., D.M., S.S. and C.C.; software, J.Y., D.M., S.S. and C.C.; validation, J.Y., G.V. and J.I.; formal analysis, J.Y.; investigation, J.Y.; resources, D.M., S.S. and C.C.; data curation, J.Y.; writing—original draft preparation, J.Y.; writing—review and editing, J.Y., G.V., J.I., D.M., S.S. and C.C.; visualization, J.Y.; supervision, S.S. and C.C.; project administration, S.S. and C.C.; funding acquisition, S.S. and C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Aerospace Institute of Technology through the FANFARE grant 113286.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting the findings of this paper are available from the corresponding author upon reasonable request.

Acknowledgments: The authors would like to acknowledge the computational resources provided by the ARCHER2 UK National Supercomputing Service (DOI: <https://doi.org/10.5281/zenodo.14507040>, <https://www.archer2.ac.uk> (accessed on 28 June 2024)) under the UK Turbulence Consortium (EP/R029326/1) and Imperial College Research Computing Service (DOI: <https://doi.org/10.14469/hpc/2232>).

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this paper:

PTPM	Point-to-point direct interpolation method
ALE	Arbitrary Lagrangian–Eulerian
DG	Discontinuous Galerkin
CFL	Courant–Friedrichs–Lewy
DIRK	Diagonally implicit Runge–Kutta
SM	Sliding mesh
IBM	Immersed boundary method
DoF	Degrees of freedom
ICI	Incompressible clean inflow
CCI	Compressible clean inflow
RSI	Rotor–stator interactions
DGCL	Discrete geometric conservation law
CTLs	Convective time lengths
PSD	Power spectral density
DNS	Direct numerical simulation
LES	Large eddy simulation

Nomenclature

c	Chord length
c_{ax}	Axial chord length
C_p	Pressure coefficient
$\mathbf{F}$	Advection and viscous flux vectors
$\mathcal{F}$	Numerical (trace) flux
F_{red}	Reduced frequency
Ma	Mach number
p	Pressure
P	Number of modes in the one-dimensional expansion
$\mathbf{R}$	Residual
R_{ii}	Spanwise two-point autocorrelation
Re	Reynolds number
T	Temperature
$\mathbf{u}$	Conservative variable vector
u_i	Velocity components
U_∞	Inlet velocity
$\mathbf{v}_g$	Grid velocity
v_n	Grid velocity in the normal direction
α_1, α_2	Inflow and outflow angle
γ	Ratio of specific heats

Δz	Spanwise separation
λ	Diagonal coefficient of the singly DIRK scheme
ρ	Density
τ_w	Wall shear stress
φ	Vortex strength
ϕ	Test function
Φ	Expansion basis, spanning the same space as ϕ
$\hat{\cdot}$	Variables in coefficient space
δ	Approximate (discrete) solution
+	Wall units
'	Fluctuation
1	Inlet
2	Outlet
ax	Axial
b	Bar
g	Grid
is	Isentropic
n	Normal direction quantity
ref	Reference
t	Total
w	Wall

References

1. Donea, J.; Huerta, A. *Finite Element Methods for Flow Problems*; Wiley: Chichester, UK, 2003.
2. Zhang, B.; Ding, C.; Liang, C. High-order implicit large-eddy simulation of flow over a marine propeller. *Comput. Fluids* **2021**, *224*, 104967. [\[CrossRef\]](#)
3. Zhang, B.; Liang, C. A simple, efficient, and high-order accurate curved sliding-mesh interface approach to spectral difference method on coupled rotating and stationary domains. *J. Comput. Phys.* **2015**, *295*, 147–160. [\[CrossRef\]](#)
4. Zhang, B.; Liang, C. A conservative high-order method utilizing dynamic transfinite mortar elements for flow simulations on curved nonconforming sliding meshes. *J. Comput. Phys.* **2021**, *443*, 110522. [\[CrossRef\]](#)
5. Ferrer, E.; Willden, R.H. A high order discontinuous Galerkin–Fourier incompressible 3D Navier–Stokes solver with rotating sliding meshes. *J. Comput. Phys.* **2012**, *231*, 7037–7056. [\[CrossRef\]](#)
6. Kang, Y.S.; Sohn, D.; Kim, J.H.; Kim, H.G.; Im, S. A sliding mesh technique for the finite element simulation of fluid-solid interaction problems by using variable-node elements. *Comput. Struct.* **2014**, *130*, 91–104. [\[CrossRef\]](#)
7. Dürrwächter, J.; Kurz, M.; Kopper, P.; Kempf, D.; Munz, C.D.; Beck, A. An efficient sliding mesh interface method for high-order discontinuous Galerkin schemes. *Comput. Fluids* **2021**, *217*, 104825. [\[CrossRef\]](#)
8. Pezzano, S.; Duvigneau, R. A fully-conservative sliding grid algorithm for compressible flows using an Isogeometric Discontinuous Galerkin scheme. *Comput. Methods Appl. Mech. Eng.* **2022**, *395*, 115000. [\[CrossRef\]](#)
9. Sandberg, R.D.; Michelassi, V.; Pichler, R.; Chen, L.; Johnstone, R. Compressible Direct Numerical Simulation of Low-Pressure Turbines—Part I: Methodology. *J. Turbomach.* **2015**, *137*, 051011. [\[CrossRef\]](#)
10. Gao, J. A sliding-mesh interface method for three dimensional high order spectral difference solver. *J. Comput. Phys.* **2022**, *454*, 110988. [\[CrossRef\]](#)
11. Ying, W.; Fattah, R.; Zhong, S.; Guo, J.; Zhang, X. An efficient and high-order sliding mesh method for computational aeroacoustics. *J. Supercomput.* **2022**, *78*, 9492–9520. [\[CrossRef\]](#)
12. Michelassi, V.; Chen, L.W.; Pichler, R.; Sandberg, R.D. Compressible Direct Numerical Simulation of Low-Pressure Turbines—Part II: Effect of Inflow Disturbances. *J. Turbomach.* **2015**, *137*, 071005. [\[CrossRef\]](#)
13. Stefanica-Nica, D.O. *Domain Decomposition Methods for Mortar Finite Elements*; New York University: New York, NY, USA, 1999.
14. Qu, K.; Xie, F.; Cai, J. Extending slide-slip mesh update method to finite volume method. *Aerosp. Sci. Technol.* **2018**, *77*, 50–57. [\[CrossRef\]](#)
15. Kim, H.G. Interface element method (IEM) for a partitioned system with non-matching interfaces. *Comput. Methods Appl. Mech. Eng.* **2002**, *191*, 3165–3194. [\[CrossRef\]](#)
16. Laughton, E.; Tabor, G.; Moxey, D. A comparison of interpolation techniques for non-conformal high-order discontinuous Galerkin methods. *Comput. Methods Appl. Mech. Eng.* **2021**, *381*, 113820. [\[CrossRef\]](#)

17. Galbraith, M.C.; Benek, J.A.; Orkwis, P.D.; Turner, M.G. A Discontinuous Galerkin Chimera scheme. *Comput. Fluids* **2014**, *98*, 27–53. [\[CrossRef\]](#)
18. Hesthaven, J.S.; Warburton, T. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*; Springer: New York, NY, USA, 2008.
19. Yan, Z.G.; Pan, Y.; Castiglioni, G.; Hillewaert, K.; Peir, J.; Moxey, D.; Sherwin, S.J. Nektar++: Design and implementation of an implicit, spectral/hp element, compressible flow solver using a Jacobian-free Newton Krylov approach. *Comput. Math. Appl.* **2021**, *81*, 351–372. [\[CrossRef\]](#)
20. Yan, Z. Efficient Implicit Spectral/hp Element DG Techniques for Compressible Flows. Ph.D. Thesis, Aeronautics, Imperial College London, London, UK, 2022.
21. Kennedy, C.A.; Carpenter, M.H. *Diagonally Implicit Runge-Kutta Methods for Ordinary Differential Equations. A Review*; Technical Report NASA/TM-2016-219173; NASA Langley Research Center: Hampton, VA, USA, 2016.
22. Benzi, M. Preconditioning Techniques for Large Linear Systems: A Survey. *J. Comput. Phys.* **2002**, *182*, 418–477. [\[CrossRef\]](#)
23. Vivarelli, G.; Isler, J.A.; Montomoli, F.; Cantwell, C.; Sherwin, S.J.; Frey-Marioni, Y.; Vazquez-Diaz, R. Applications and Recent Developments of the Open-Source Computational Fluid Dynamics High-Fidelity Spectral/Hp Element Framework Nektar++ for Turbomachinery Configurations. In Proceedings of the Turbo Expo: Power for Land, Sea, and Air, London, UK, 24–28 June 2024; Paper No. GT2024-125039. [\[CrossRef\]](#)
24. Cassinelli, A. A Spectral/hp Element DNS Study of Flow Past Low-Pressure Turbine Cascades and the Effects of Inflow Conditions. Ph.D. Thesis, Imperial College London, London, UK, 2021.
25. Ye, J.; Vivarelli, G.; Isler, J.; Moxey, D.; Sherwin, S.; Cantwell, C. High-fidelity sliding mesh method for moving geometries in turbomachinery applications. In Proceedings of the 16th European Conference on Turbomachinery Fluid Dynamics & Thermodynamics, Hannover, Germany, 24–28 March 2025.
26. Laughton, E. Enabling Efficient High-Fidelity Computational Fluid Dynamics (CFD) for Moving Geometries. Ph.D. Thesis, University of Exeter, Exeter, UK, 2024.
27. Cantwell, C.D.; Moxey, D.; Comerford, A.; Bolis, A.; Rocco, G.; Mengaldo, G.; De Grazia, D.; Yakovlev, S.; Lombard, J.E.; Ekelschot, D.; et al. Nektar++: An open-source spectral/hp element framework. *Comput. Phys. Commun.* **2015**, *192*, 205–219. [\[CrossRef\]](#)
28. Moxey, D.; Cantwell, C.D.; Bao, Y.; Cassinelli, A.; Castiglioni, G.; Chun, S.; Juda, E.; Kazemi, E.; Lackhove, K.; Marcon, J.; et al. Nektar++: Enhancing the capability and application of high-fidelity spectral/hp element methods. *Comput. Phys. Commun.* **2020**, *249*, 107110. [\[CrossRef\]](#)
29. Karniadakis, G.E.; Sherwin, S.J. *Spectral/hp Element Methods for Computational Fluid Dynamics*; Oxford University Press: Oxford, UK, 2005.
30. Stadtmüller, P. *Investigation of Wake-Induced Transition on the LP Turbine Cascade T106A-EIZ*; DFG-Verbundprojekt Fo 136/11, Version 1.0; Universität der Bundeswehr München: Munich, Germany, 2001.
31. Sandberg, R.D.; Michelassi, V. The Current State of High-Fidelity Simulations for Main Gas Path Turbomachinery Components and Their Industrial Impact. *Flow Turbul. Combust.* **2019**, *102*, 797–848. [\[CrossRef\]](#)
32. Georgiadis, N.J.; Rizzetta, D.P.; Fureby, C. Large-Eddy Simulation: Current Capabilities, Recommended Practices, and Future Research. *AIAA J.* **2010**, *48*, 1772–1784. [\[CrossRef\]](#)
33. Lyu, G.; Chen, C.; Du, X.; Sherwin, S.J. Stable, entropy-pressure compatible subsonic Riemann boundary condition for embedded DG compressible flow simulations. *J. Comput. Phys.* **2023**, *476*, 111896. [\[CrossRef\]](#)
34. Lindblad, D.; Isler, J.; Moragues Ginard, M.; Sherwin, S.J.; Cantwell, C.D. Nektar++: Development of the compressible flow solver for jet aeroacoustics. *Comput. Phys. Commun.* **2024**, *300*, 109203. [\[CrossRef\]](#)
35. Isler, J.; Vivarelli, G.; Montomoli, F.; Sherwin, S.; Adami, P.; Vazquez, R. High Fidelity Compressible and Incompressible Flow Simulations of a Turbo-machinery Geometrical Configuration. In Proceedings of the 15th European Conference on Turbomachinery Fluid Dynamics & Thermodynamics, Budapest, Hungary, 24–28 April 2023.
36. Isler, J.; Vivarelli, G.; Cantwell, C.; Montomoli, F.; Sherwin, S.; Frey, Y.; Meyer, M.; Vazquez, R. Source Term-Based Synthetic Turbulence Generator Applied to Compressible DNS of the T106A Low-Pressure Turbine. *Int. J. Turbomach. Propuls. Power* **2025**, *10*, 13. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.