A hybrid parallel open source Ffowcs Williams-Hawkings code for aeroacoustics calculation

Keli Zhang*, Xinliang Li** and Peiqing Liu*
Corresponding author: lpq@buaa.edu.cn
* School of Aeronautic Science and Engineering, Beihang University, Beijing, 100191, PR China.
** LHD, Institute of Mechanics, Chinese Academy of Sciences, Beijing, 100190, PR China.

Abstract: This paper introduces OpenCFD-FWH, an open-source implementation of the Ffowcs Williams-Hawkings (FW-H) acoustic analogy for permeable surfaces under incident mean flow. The method is developed for coupling with the OpenCFD-EC solver. Convective effects are accounted for by transforming the problem into a moving reference frame using the Garrick triangle approach. A rotational transformation further corrects angle-of-attack (AoA) effects. This formulation simplifies the FW-H surface integral and improves computational efficiency. Validation is performed on three benchmark cases: a stationary monopole, a stationary dipole (both in uniform flow with angle of incidence), and the 30P30N high-lift airfoil. For the monopole and dipole, numerical results match analytical solutions closely. For the 30P30N airfoil at $Re = 1.71 \times 10^6$ and 5.5° AoA, far-field noise is predicted. The flow field is computed using improved delayed detached-eddy simulation (IDDES) in OpenCFD-EC. Acoustic predictions agree well with experimental data. To handle large-scale simulations, a hybrid MPI/OpenMP parallelization is adopted. Speedup reaches 538.5 times in the 30P30N case and overcomes memory constraints. The code is released publicly at `https://github.com/Z-K-L/OpenCFD-FWH`.

Keywords: Aeroacoustics, Parallel algorithms, Computational Fluid Dynamics, FW-H.

1. Introduction

Due to the growing focus on environmental conservation, aeroacoustic noise has garnered significant interest from both industry and academia, particularly within the aviation sector. The impact of aircraft noise on airport development is substantial, underscoring the importance of conducting far-field noise assessments during aircraft design and development phases. The hybrid computational aeroacoustics (CAA) approach stands out as the most practical method for accurately evaluating the far-field noise of an aircraft or airframe components, given the impracticality of resolving acoustic wave propagation for far-field noise evaluation in engineering applications.

The hybrid CAA method involves obtaining the unsteady flow field through computational fluid dynamics (CFD) solvers and then applying acoustic analogy to calculate far-field noise. This method is widely used because it substantially reduces computational complexity. For instance, Molina et al. [1] studied tandem cylinder noise using delayed detached-eddy simulation (DDES) and Ffowcs Williams-Hawkings (FW-H) acoustic analogy. Ma et al. [2] investigated the aeroacoustic characteristics of the swept constant-chord half (SCCH) model with four different types of high-lift devices using improved DDES (IDDES) and FW-H acoustic analogy. Hu et al. [3] utilized implicit wall-resolved large eddy simulation (WRLES) and FW-H acoustic analogy to explore the noise reduction mechanisms of trailing-edge serrations. Chen et al. [4] also employed the hybrid method of LES and FW-H acoustic analogy to study noise generated by flow across a cylinder with different spanwise lengths. Souza et al. [5] conducted lattice Boltzmann method (LBM) simulations on the 30P30N high-lift configuration and applied FW-H acoustic analogy to calculate the corresponding far-field signals. Teruna et al. [6] analyzed the noise reduction effect of a fully resolved 3D printed porous trailing-edge also using a LBM solver combined with FW-H acoustic analogy. DNS and FW-H acoustic analogy were used by Turner and Kim [7] to evaluate the significance of quadrupole noise in airfoil flow separation or stall conditions. They obtained the quadrupole noise by comparing the FW-H results for solid and permeable surfaces.

At present, only a few open source FW-H implementations are available. Existing options include libAcoustics developed by Epikhin et al. [8] for OpenFOAM in C++, SU2PY_FWH developed by Zhou et al. [9] for SU2 in Python, and a Farassat 1A solver developed by Shen et al. [10] for HiFiLES in C++. These codes, however, possess notable deficiencies:

1. **Lack of MPI support:** none of the codes support MPI parallelism, which is essential for efficient computation and reduced memory usage when dealing with large datasets;
2. **Limited to solid-surface FW-H implementation:** some of them only support the solid-surface implementation of the FW-H method, which excludes quadrupole sources;
3. **Dependency on specific CFD platforms:** libAcoustics requires OpenFOAM, and SU2PY_FWH requires SU2, limiting their general applicability.
4. **Does not account for inflow with incidence angles:** inflow conditions with angles-of-attack (AoAs) are not considered.
5. **Insufficient documentation:** comprehensive tutorials are lacking, hindering integration with other CFD solvers.

Hence, an open CFD code for FW-H acoustic analogy (OpenCFD-FWH) has been developed for our compressible finite volume solver, OpenCFD-EC [11]. Moreover, it is readily applicable to any solver equipped with suitable data structures. Users may also adapt the data reading module as needed. For instance, Zhang et al. [12] have successfully integrated OpenCFD-FWH into a high-order commercial solver, Dimaxer.

The code implements a permeable surface FW-H integration scheme that incorporates the Garrick triangle [13], thereby simplifying the governing equations. Inflow with an incidence is also accounted for. OpenCFD-FWH is developed in a hybrid parallel framework to accelerate the computing processes and reduce memory requirements for a single node or computer. In addition, MATLAB scripts for monopole and dipole benchmarks are provided to produce the data needed for illustrative examples.

The rest of the paper is organized as follows. In § 2, the FW-H formulation for permeable surface is derived, with convection and AoA effects accounted for. § 3 outlines the structure and parallel implementation of OpenCFD-FWH. The results of OpenCFD-FWH for three validation benchmarks are reported in § 4. Finally, the conclusions are summarised in § 5.

2. FW-H acoustic analogy

2.1 Permeable surface FW-H equation

The FW-H equation [14] for permeable surface has the following form:

$$\Box^2 c^2(\rho-\rho_0)=\frac{\partial}{\partial t}\left[Q_n\delta(f)\right]-\frac{\partial}{\partial x_i}\left[L_i\delta(f)\right]+\frac{\partial^2}{\partial x_i\partial x_j}\left[T_{ij}H(f)\right], \tag{1}$$

where $\Box^2=1/c^2\partial^2/\partial t^2-\nabla^2$ is the D'Alembert operator, c is the speed of sound, ρ is the density, and ρ_0 denotes the density of the undisturbed medium. δ and H are the Dirac delta and Heaviside function, respectively. The moving surface is described by $f(\boldsymbol{x},t)=0$ such that $\hat{\boldsymbol{n}}=\nabla f$ represents the unit outward normal of the surface [15]. Q_n, L_i, and Lighthill stress tensor T_{ij} are defined as follows:

$$Q_n = Q_i\hat{n}_i = [\rho_0 v_i + \rho(u_i - v_i)]\,\hat{n}_i, \tag{2}$$
$$L_i = L_{ij}\hat{n}_j = \left[P_{ij} + \rho u_i(u_j - v_j)\right]\hat{n}_j, \tag{3}$$
$$T_{ij} = \rho u_i u_j + P_{ij} + c^2(\rho - \rho_0)\delta_{ij}, \tag{4}$$

where v_i is the velocity component of the moving surface, u_i is the velocity component of the fluid, δ_{ij} is the Kronecker delta, and P_{ij} represents the stress tensor:

$$P_{ij} = (p - p_0)\delta_{ij} - \sigma_{ij}, \tag{5}$$

where p_0 is the ambient pressure, σ_{ij} is the viscous stress tensor. Usually, σ_{ij} is considered a negligible source of sound and is neglected by most other FW-H implementations [8, 9, 10, 15]. Therefore, $P_{ij} = (p - p_0)\delta_{ij}$ is used in this paper.

2.2 Integration solution for general cases

Omitting the quadrupole term in Eq. (1), and following the derivation procedure of Farassat 1A formulation [15], the integral solution of the permeable surface FW-H equation can be obtained:

$$\begin{aligned} 4\pi p'_T(\boldsymbol{x},t) = &\int_{f=0}\left[\frac{\dot{Q}_i\hat{n}_i + Q_i\dot{\hat{n}}_i}{r(1-M_r)^2}\right]_{ret} dS \\ &+ \int_{f=0}\left[\frac{Q_n(r\dot{M}_r + c_0(M_r - M^2))}{r^2(1-M_r)^3}\right]_{ret} dS, \end{aligned} \tag{6}$$

$$\begin{aligned} 4\pi p'_L(\boldsymbol{x},t) = &\frac{1}{c_0}\int_{f=0}\left[\frac{\dot{L}_r}{r(1-M_r)^2}\right]_{ret} dS \\ &+ \int_{f=0}\left[\frac{L_r - L_M}{r^2(1-M_r)^2}\right]_{ret} dS \\ &+ \frac{1}{c_0}\int_{f=0}\left[\frac{L_r(r\dot{M}_r + c_0(M_r - M^2))}{r^2(1-M_r)^3}\right]_{ret} dS, \end{aligned} \tag{7}$$

$$p'(\boldsymbol{x},t) = p'_T(\boldsymbol{x},t) + p'_L(\boldsymbol{x},t), \tag{8}$$

where $\boldsymbol{x}$ is the coordinate vector of the observer, t is the observer time, r is the distance between the observer and the source, and c_0 denotes the speed of sound in the undisturbed medium. The superscript "·" indicates differentiation with respect to the source time τ, and the subscripts T and L denote the thickness and loading components, respectively. M represents the Mach number of the moving surface, with its component given by $M_i = v_i/c_0$. The terms M_r, L_r, and L_M are defined as:

$$M_r = M_i\hat{r}_i, \tag{9}$$
$$L_r = L_i\hat{r}_i, \tag{10}$$
$$L_M = L_i M_i, \tag{11}$$

where $\hat{r}_i$ is the unit radiation vector component.

The subscript ret in Eqs. (6) and (7) indicates the quantities inside the square brackets are determined at the retarded time:

$$\tau_{ret} = t - r_{ret}/c \ . \tag{12}$$

It is worth noting that although the quadrupole term is omitted, contributions from quadrupole sources inside the permeable FW-H surface are still retained by Eq. (8) according to Brentner and Farassat [16].

2.3 Integration solution for inflow with an incidence

Eq. (8) is derived in a frame where the source moves through a quiescent medium and the far-field observers are at rest. For the common CFD cases of stationary sources and observers in uniform flow at incidence, the coordinate system can be transformed using the Garrick triangle. In this new coordinate system, the source is now in motion within a stationary medium, while observers remain stationary relative to the source. This transformation results in a significant simplification of the formulation and enhances the computational efficiency of the code.

Initially, let the mean flow have a constant velocity U_0 in the direction of the positive x_1 axis, Eq. (12) will be changed to:

$$\tau_{ret} = t - R/c_0 \ . \tag{13}$$

where R is the effective acoustic distance between the source and the observer [17]:

$$R = \frac{-M_0 d_1 + R_*}{\beta^2}, \tag{14}$$

$$R_* = \sqrt{d_1^2 + \beta^2 [d_2^2 + d_3^2]}, \tag{15}$$

$$\beta = \sqrt{1 - M_0^2}, \tag{16}$$

$$M_0 = U_0/c_0, \tag{17}$$

where $d_i = x_i - y_i$ is the distance vector component between the observer and the source. Thus, the unit radiation vector component is now changed to:

$$\hat{R}_1 = \frac{-M_0 R_* + d_1}{\beta^2 R}, \tag{18}$$

$$\hat{R}_2 = d_2/R, \tag{19}$$

$$\hat{R}_3 = d_3/R \ . \tag{20}$$

Subsequently, consider a mean flow with an incidence in the x-y plane, and its velocity magnitude remains unchanged. By using the 2D plane coordinate transformation:

$$d_1' = d_1 \cos(\text{AoA}) + d_2 \sin(\text{AoA}), \tag{21}$$

$$d_2' = -d_1 \sin(\text{AoA}) + d_2 \cos(\text{AoA}), \tag{22}$$

and substituting them into the d_1 and d_2 of Eqs. (14) and (15) yields:

$$R = \frac{-M_1 d_1 - M_2 d_2 + R_*}{\beta^2}, \tag{23}$$

$$R_* = \sqrt{(M_1 d_1 + M_2 d_2)^2 + \beta^2 [d_1^2 + d_2^2 + d_3^2]}, \tag{24}$$

$$M_1 = M_0 \cos(\text{AoA}), \tag{25}$$

$$M_2 = M_0 \sin(\text{AoA}) \ . \tag{26}$$

The first two unit radiation vector components are now changed to:

$$\hat{R}'_1 = \frac{-M_0 R_* + d'_1}{\beta^2 R}, \tag{27}$$

$$\hat{R}'_2 = d'_2/R, \tag{28}$$

$$\hat{R}_1 = \hat{R}'_1 \cos(\mathrm{AoA}) - \hat{R}'_2 \sin(\mathrm{AoA}), \tag{29}$$

$$\hat{R}_2 = \hat{R}'_1 \sin(\mathrm{AoA}) + \hat{R}'_2 \cos(\mathrm{AoA}) \ . \tag{30}$$

Update all instances of r in Eqs. (6)~(10) by Eq. (23). The velocities of both the moving surface and the fluid are adjusted by subtracting the mean flow velocity, thereby accounting for the reference frame change. Consequently, R is constant for each observer and can be pre-computed. The uniform rectilinear motion of the surface also renders the time derivative of the unit normal vector $\hat{n}_i$ and the quantity M_R zero. Eqs. (6) and (7) are therefore reduced to the following simplified forms:

$$\begin{aligned} 4\pi p'_T(\boldsymbol{x}, t) = &\int_{f=0} \left[\frac{\dot{Q}_i \hat{n}_i}{R(1-M_R)^2}\right]_{ret} dS \\ &+ \int_{f=0} \left[\frac{Q_n c_0 (M_R - M^2)}{R^2(1-M_R)^3}\right]_{ret} dS, \end{aligned} \tag{31}$$

$$\begin{aligned} 4\pi p'_L(\boldsymbol{x}, t) = &\frac{1}{c_0}\int_{f=0} \left[\frac{\dot{L}_R}{R(1-M_R)^2}\right]_{ret} dS \\ &+ \int_{f=0} \left[\frac{L_R - L_M}{R^2(1-M_R)^2}\right]_{ret} dS \\ &+ \int_{f=0} \left[\frac{L_R (M_R - M^2)}{R^2(1-M_R)^3}\right]_{ret} dS, \end{aligned} \tag{32}$$

with

$$Q_n = [-\rho_0 U_{0i} + \rho u_i]\, \hat{n}_i, \tag{33}$$

$$L_i = [P_{ij} + \rho(u_i - U_{0i}) u_j] \hat{n}_j, \tag{34}$$

$$M_R = M_i \hat{R}_i, \tag{35}$$

$$L_R = L_i \hat{R}_i \ . \tag{36}$$

It is worth noting that the quantities enclosed within the square brackets in Eqs. (31) and (32) are now determined at the retarded time computed by Eq. (13). The essential parameters required for the far-field noise calculation from the CFD solver are the coordinate, unit outward normal, and area of the FW-H surfaces, along with the density, velocity, and pressure fluctuation at each sampling frame.

2.4 Non-dimensionalization

Since OpenCFD-EC employs dimensionless Navier-Stokes equations, OpenCFD-FWH is based on a non-dimensional form of Eqs. (31) and (32) to prevent inaccuracies in data conversion and reduce computational costs.

The dimensionless transformation utilizes mean flow quantities as the reference values, except for pressure and the coordinate, which are made non-dimensional by $\rho_0 U_0^2$ and the units of the mesh, respectively.

The resulting expressions are:

$$\rho^* = \rho/\rho_0,\ \ u_i^* = u_i/U_0,\ \ v_i^* = v_i/U_0,\ \ p^* = p/\rho_0 U_0^2, \tag{37}$$

$$c_0^* = c_0/U_0 = 1/M_0,\ \ x_i^* = x_i/L_{ref},\ \ y_i^* = y_i/L_{ref}, \tag{38}$$

$$dS^* = dS/L_{ref}^2,\ \ t^* = tU_0/L_{ref},\ \ \tau^* = \tau U_0/L_{ref}\ . \tag{39}$$

Then the dimensionless FW-H integration solution for inflow with an incidence can be derived by substituting the variables in Eqs. (31) and (32) with their respective non-dimensional equivalents:

$$\begin{aligned} 4\pi {p'_T}^*(\boldsymbol{x}^*, t^*) = & \int_{f=0} \left[\frac{\dot{Q}_i^* \hat{n}_i}{R^*(1-M_R)^2} \right]_{ret} dS^* \\ & + \int_{f=0} \left[\frac{Q_n^*(M_R - M^2)}{M_0 {R^*}^2 (1-M_R)^3} \right]_{ret} dS^*, \end{aligned} \tag{40}$$

$$\begin{aligned} 4\pi {p'_L}^*(\boldsymbol{x}^*, t^*) = & \int_{f=0} \left[\frac{M_0 \dot{L}_R^*}{R^*(1-M_R)^2} \right]_{ret} dS^* \\ & + \int_{f=0} \left[\frac{L_R^* - L_M^*}{{R^*}^2(1-M_R)^2} \right]_{ret} dS^* \\ & + \int_{f=0} \left[\frac{L_R^*(M_R - M^2)}{{R^*}^2(1-M_R)^3} \right]_{ret} dS^*. \end{aligned} \tag{41}$$

with

$$Q_n^* = \left[-U_{0i}^* + \rho^* u_i^* \right] \hat{n}_i, \tag{42}$$

$$L_i^* = [P_{ij}^* + \rho^*(u_i^* - U_{0i}^*) u_j^*] \hat{n}_j, \tag{43}$$

$$U_{01}^* = \cos(\text{AoA}),\ \ U_{02}^* = \sin(\text{AoA}),\ \ U_{03}^* = 0\ . \tag{44}$$

Since $f = 0$, $\hat{R}_i$, and $\hat{n}_i$ are identical in both dimensional and non-dimensional forms, the use of "*" to denote non-dimensional quantities is unnecessary.

3. Implementation of OpenCFD-FWH

OpenCFD-FWH can be segmented into 4 main parts: initialization, calculation of pressure signals, data output and finalization, as illustrated in Fig. 1.

3.1 Initialization

The first step of OpenCFD-FWH is to initialize all MPI processors, which involves reading the control file, acquiring surface geometric data, assigning surfaces to the corresponding MPI processor, allocating memory, and reading the location of the observers as well as the FW-H dataset.

The control.fwh file specifies crucial parameters for OpenCFD-FWH, such as Mach number, AoA, time step, number of observers, number of sampling frames, and number of OpenMP threads. An example of the control.fwh file is provided in Appendix A.1.

The FWH_Surface_Geo.dat file contains the coordinates y_i, unit outward normals $\hat{n}_i$, and area dS of each subface in the FW-H data surface. These quantities are specified at the center of the subfaces and split across different Faces due to OpenCFD-EC being a cell center solver for multiblock structure mesh. A

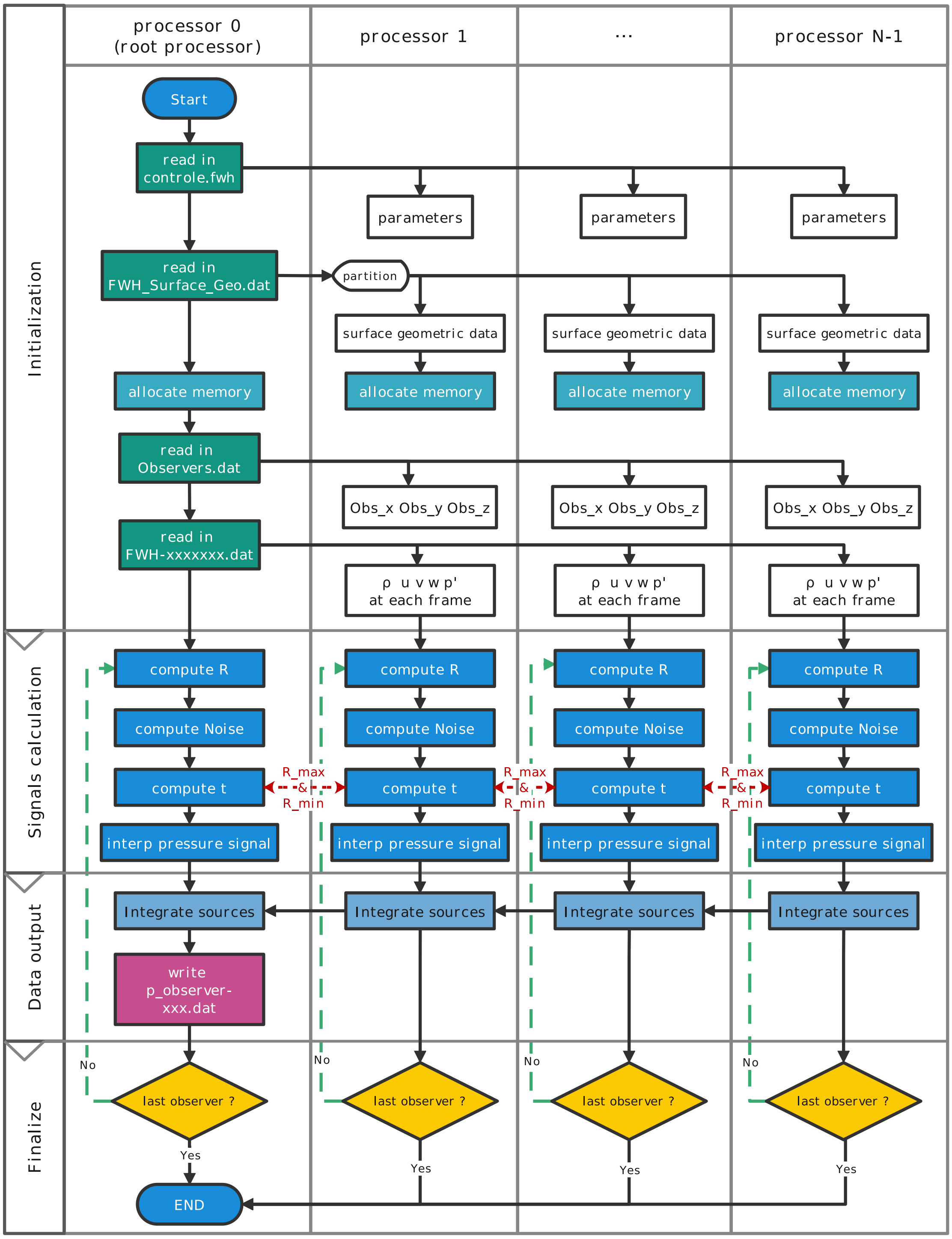


Figure 1: MPI parallel framework of OpenCFD-FWH.

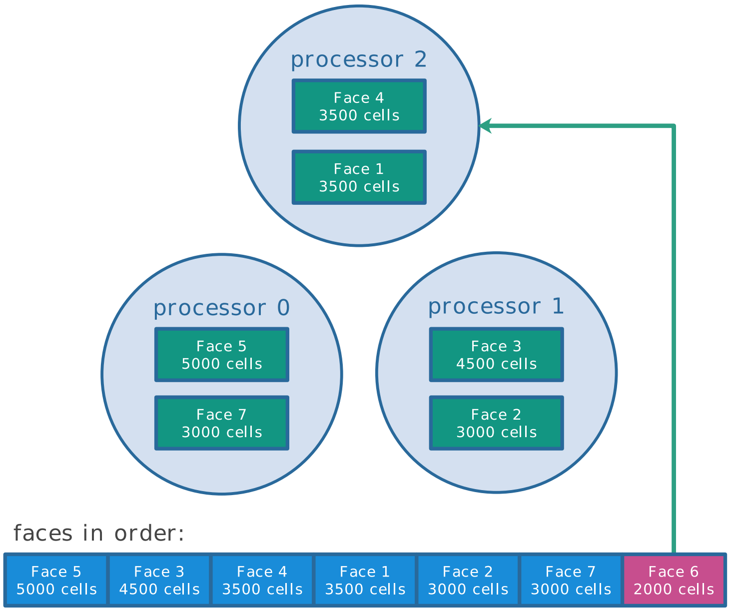


Figure 2: Schematic of the MPI partition method for a case involving 3 processors and 7 Faces.

detailed description of the file can be found in Appendix A.2.

Once the Faces information is obtained, a partition method deployed by OpenCFD-EC for block splitting is applied for load balancing, as illustrated in Fig. 2. This method ranks the Faces by their cell numbers and then assigns the Faces to the processor with the least number of cells in order. Consequently, the upper limit for the utilization of MPI processors by OpenCFD-FWH corresponds to the number of Faces. To achieve faster MPI acceleration, the FW-H surface can be divided as much and as equally as possible during the mesh generation stage. Additionally, when using new Intel CPUs with a big.LITTLE-style heterogeneous core layout or parallelizing an old system with a new one, a partition method that considers the performance of the processors can be adopted to achieve better load balancing.

Following the MPI partition, each processor allocates memory for FW-H data at all the sampling frames according to the assigned Faces. The root processor reads the observers.dat file, in which the coordinates of each observer occupy a single row, and then broadcasts them to all other processors.

Subsequently, the FW-H dataset is read by the root processor and distributed to the corresponding processor. An illustrative description of the FW-H dataset can be found in Appendix A.3. Finally, memory is allocated for interpolated observer time across all processors, along with the final acoustic signal result at the root processor.

3.2 Pressure signals calculation

The second step of OpenCFD-FWH involves the computation of pressure signals for one observer at a time, as opposed to simultaneously calculating at all observers for the sake of conserving memory usage. Furthermore, OpenMP parallelization is deployed within each processor to accelerate the calculation process. Further elaboration on the hybrid parallel implementation of MPI and OpenMP will be provided in Section 3.4.

The "compute R" subroutine depicted in Fig. 1 is tasked with determining the effective acoustic distance using Eqs. (23)~(30).

The "compute Noise" subroutine in Fig. 1 is responsible for computing the pressure signals at each subface during the respective source time, based on Eqs. (40)~(44). It is important to note that Q_n and L_M remain constant across different observers, but are not stored to conserve memory. Additionally, $\dot{Q}_n$ and $\dot{L}_i$ are calculated using a one-sided scheme for the first and last frame, and a central scheme for the

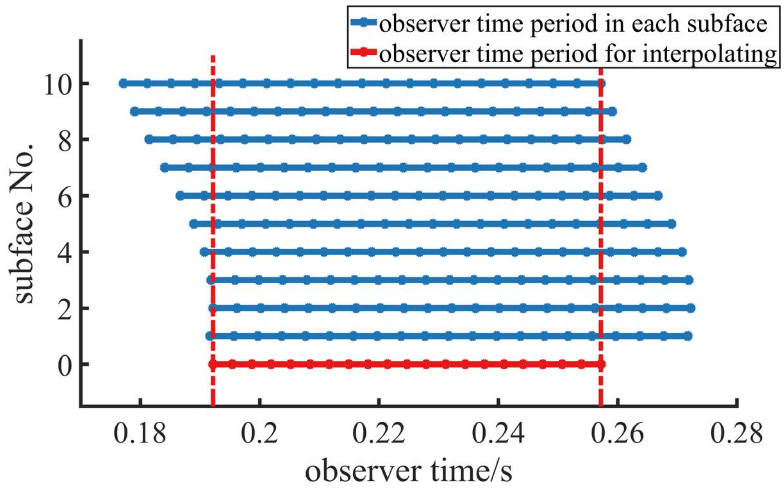


Figure 3: A schematic representation of the observer time period regarding 10 different subfaces.

other frames.

The "compute t" subroutine in Fig. 1 is responsible for determining the observer time based on:

$$t^*_{start} = R^*_{max} M_0, \tag{45}$$

$$t^*_{end} = \tau_{max} + R^*_{min} M_0 \ , \tag{46}$$

where R^*_{max} and R^*_{min} are computed via the MPI_ALLREDUCE function. The observer time period, over which all subfaces contribute collectively to the pressure signal, is then given by $t^*_{end} - t^*_{start}$, as illustrated in Fig. 3.

The "interp pressure signal" subroutine in Fig. 1 performs cubic spline interpolation to map pressure signals from source time to observer time for each subface. After interpolation, the stored pressure data at both source and observer times are deallocated to save memory.

3.3 Data output and finalization

The third step of OpenCFD-FWH involves conducting surface integration across all the subfaces to acquire the pressure signal at an observer and saving it in the p_observer-xxx.log file, where xxx represents the observer number, within the *FWH_result-mpi* folder.

The final step in OpenCFD-FWH checks whether all observers have finished their computations. If not, the code returns to the second step for the next observer. Once all observers are complete, all processors call MPI_Finalize to exit the code.

3.4 Parallelization

OpenMP is a widely used API (Application Programming Interface) designed to support shared-memory parallelization in multi-core and multi-processor systems. It is developed to simplify parallel programming in C, C++, and Fortran, and can be deployed without extensive modifications to existing serial codes. OpenCFD-FWH is implemented in a hybrid parallel way, where the FW-H data surface is distributed to multiple MPI processors, and OpenMP is deployed to divide the loop in the computing stage among each MPI processor. This leads to a significant reduction in computation time, as demonstrated in Table 1. By employing 31 nodes, each with 32 CPU cores for OpenMP parallelization, a notable acceleration of up to 538.5 times is achieved compared to the serial execution. In addition, when the number of MPI processors (only MPI parallel) and OpenMP threads (only OpenMP parallel) is almost equal, their acceleration effect is nearly the same.

Table 1: Wall-clock time taken for various stages of OpenCFD-FWH using different combinations of MPI processors and OpenMP threads on the CAS SunRising platform for the 30P30N benchmark. The platform has 32 cores x86 CPU on each node, with a base clock speed of 2.0GHz. The benchmark consists of 18,252 subfaces, 6,535 sampling frames, and 40 observers.

MPI processors	OMP threads per processor	$t_{I/O}$/s	t_{cal}/s	t_{total}/s	Calculation acceleration ratio	Total acceleration ratio
1	\	753.7	18120.3	18873.9	\	\
31	\	887.2	811.3	1698.5	22.3	11.1
1	32	823.9	774.3	1598.1	23.4	11.8
31	32	717.5	33.7	751.2	538.5	25.1

Table 2: Parameters for the monopole benchmark.

$c_0\ (m/s)$	M_0	$\rho_0\ (kg/m^3)$	$AoA\ (°)$	$A\ (m^2/s)$	$f\ (Hz)$
340	0.6	1	45	1	5

Additionally, Table 1 further reveals that most of the execution time in the hybrid parallel case is consumed by initialization. The primary cause is the lack of parallelization in the present I/O operations.

Given that the 30P30N benchmark consumes up to 62.6 GB of RAM (Random Access Memory), larger FW-H datasets risk exceeding single-node memory limits. The MPI approach enables execution on memory-limited distributed systems and extends feasibility to more memory-demanding cases.

4. Validation

Analytical solutions for a stationary monopole and dipole in an incident mean flow, as well as a 30P30N benchmark case calculated using the OpenCFD-EC solver, are employed to verify the accuracy of OpenCFD-FWH.

4.1 Stationary monopole in a uniform flow with an AoA

The complex velocity potential for a stationary single-point monopole in a uniform flow is given by Najafi et al. [18]:

$$\phi(\boldsymbol{x}, t) = \frac{A}{4\pi R_*}\exp\left[\mathrm{i}\omega\left(t - \frac{R}{c_0}\right)\right] . \tag{47}$$

where A is the amplitude of the monopole, ω is the angular frequency of the monopole, and i is the imaginary unit.

The velocity, pressure, and density pulsations induced by the monopole are:

$$\boldsymbol{u}'(\boldsymbol{x}, t) = \nabla\phi(\boldsymbol{x}, t), \tag{48}$$

$$p'(\boldsymbol{x}, t) = -\rho_0\left[\frac{\partial}{\partial t} + U_{01}\frac{\partial}{\partial x_1} + U_{02}\frac{\partial}{\partial x_2}\right]\phi, \tag{49}$$

$$\rho'(\boldsymbol{x}, t) = \frac{p'}{c_0^2}, \tag{50}$$

$$U_{01} = U_0\cos(\mathrm{AoA}),\ U_{02} = U_0\sin(\mathrm{AoA}) . \tag{51}$$

The parameters of the monopole are given in Table 2 and the FW-H dataset for OpenCFD-FWH is generated by Eqs. (47)~(50).

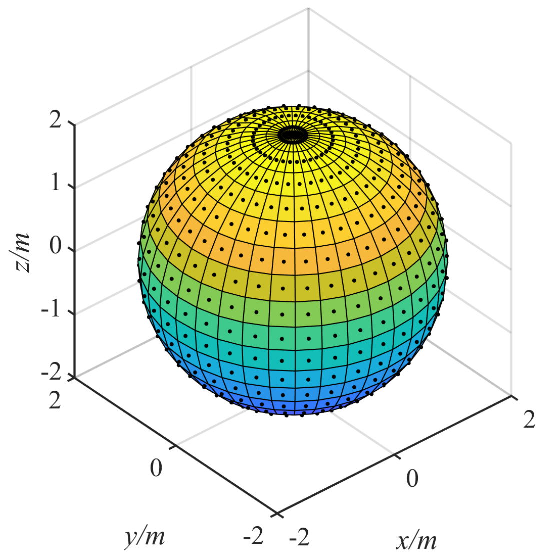


Figure 4: Schematic of the permeable FW-H data surface used in the monopole and dipole benchmarks, with black points denoting subface centroids.

The permeable FW-H data surface is a 2m radius sphere. It is centered on the single-point monopole. The sphere is divided into 18 segments along the polar angle direction and 36 segments along the azimuth angle direction, resulting in a total of 648 subfaces, as shown in Fig. 4.

Observers are placed uniformly on a 340m radius circle in the x-y plane, centered at the monopole. Using a laptop equipped with an Intel i7-13620H CPU, OpenCFD-FWH computes the acoustic signal for 20 observers in less than 2.5s with 12 OpenMP threads on a single MPI processor, since the sphere FW-H surface is generated as a single Face.

Fig. 5 displays the far-field RMS pressure directivity pattern, while Fig. 6 shows the pressure signal at the receiver right below the source. Excellent agreement is observed between the OpenCFD-FWH results and the exact analytical solutions. Moreover, Fig. 5 illustrates that the directivity pattern of the monopole is shifted towards the incoming flow due to convective effects.

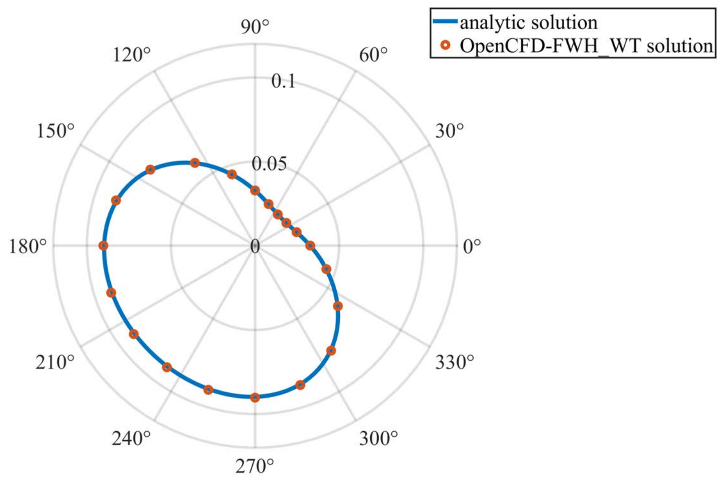


Figure 5: Far-field directivity pattern of the RMS pressure induced by the monopole at $r = 340\ m$.

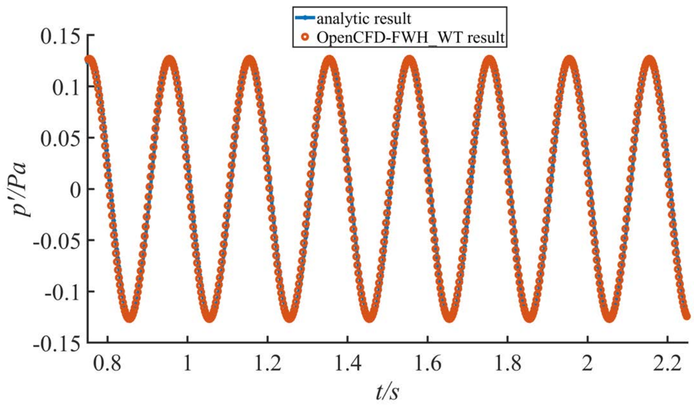


Figure 6: Acoustic pressure signal at $(0\ m, -340\ m, 0\ m)$ induced by the monopole

Table 3: Parameters for the dipole benchmark.

$c_0\ (m/s)$	M_0	$\rho_0\ (kg/m^3)$	$AoA\ (°)$	$A\ (m^2/s)$	$f\ (Hz)$
340	0.5	1	10	1.5	7.5

4.2 Stationary dipole in a uniform flow with an AoA

The complex velocity potential for a stationary single-point dipole, with the polar axis coinciding with the x_2 axis in a uniform flow can be expressed as [18]:

$$\phi(\boldsymbol{x}, t) = \frac{\partial}{\partial x_2} \left\{ \frac{A}{4\pi R_*} \exp\left[i\omega \left(t - \frac{R}{c_0} \right) \right] \right\} \ . \tag{52}$$

The velocity, pressure, and density pulsations induced by a dipole are acquired by Eqs. (48)~(50) as well. The parameters of the dipole are given in Table 3.

The FW-H data surface remains consistent with the monopole benchmark, while the observer locations have been relocated to a radius of 50 meters in the x-y plane. The dipole far-field RMS pressure directivity pattern is presented in Fig. 7. The pressure signal at the downstream observer is shown in Fig. 8. Excellent agreement is observed between the exact solutions and OpenCFD-FWH results.

Similarly, the mean flow causes a reorientation of the maximum RMS pressure, resulting in a larger RMS

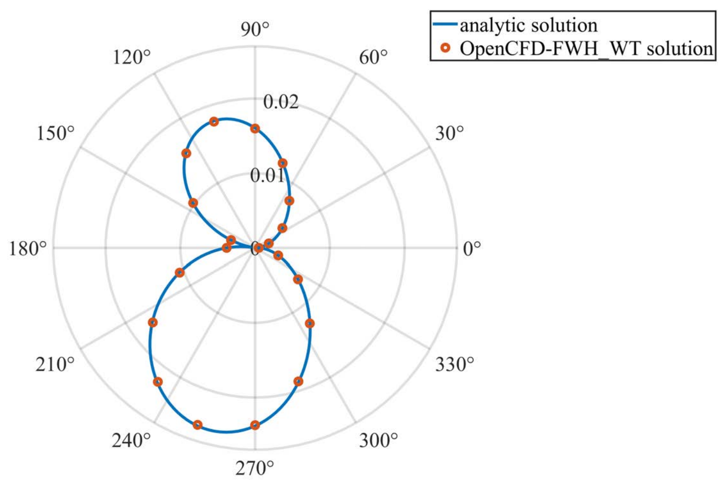


Figure 7: Far-field directivity pattern of the RMS pressure induced by the dipole at $r = 50\ m$.

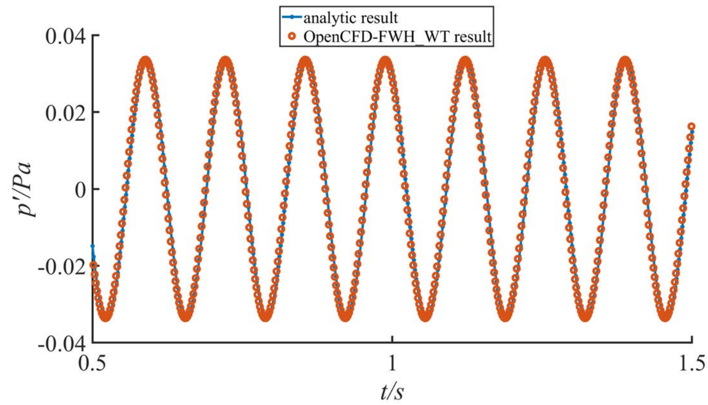


Figure 8: Acoustic pressure signal at $(0\ m, -50\ m, 0\ m)$ induced by the dipole.

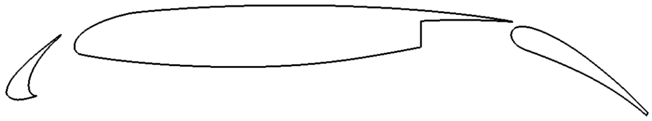


Figure 9: Profile of the JAXA modified 30P30N airfoil.

pressure in the inflow direction, as shown in Fig. 7.

To aid in validation and code comprehension, the MATLAB scripts for the monopole and dipole FW-H benchmarks are available in the Tutorials folder of the OpenCFD-FWH GitHub repository. Users can modify these scripts to support validation of OpenCFD-FWH and to deepen their understanding of the methodology.

4.3 30P30N far-field noise prediction

The 30P30N configuration was developed by McDonnell Douglas (now Boeing) in the early 1990s. It has been widely used in studies of high-lift device aeroacoustics, especially for slat noise [12, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32].

The modified version 30P30N from Japan Aerospace Exploration Agency (JAXA) [23] is employed in this study to validate OpenCFD-FWH. The airfoil profile of the 30P30N configuration is shown in Fig. 9, with a stowed chord length of $c_s = 0.4572m$. Both the slat and flap have deflection angles of 30°, and their chord lengths are $0.15c_s$ and $0.3c_s$, respectively.

IDDES based on Spalart-Allmaras (SA) turbulence model is carried out on the OpenCFD-EC solver. The simulation conditions include an inflow Mach number of 0.17 and an AoA of 5.5°. The Reynolds number, calculated based on the stowed chord length, is 1.71×10^6.

Fig. 10 illustrates the computational domain, which extends $50c_s$ in the forward and vertical directions and $75c_s$ in the rear direction. Its spanwise length is $1/9c_s$, as recommended in the 3^{rd} AIAA Workshop on Benchmark Problems for Airframe Noise Computations (BANC-III) [19], with a periodic boundary condition applied in the spanwise direction. The permeable FW-H data surface, denoted by the blue line in Fig. 10, is located a c_s away from the 30P30N airfoil and extends $5c_s$ in the wake flow direction. To prevent spurious numerical noise from wake flows crossing the permeable FW-H data surface, no endcap is applied. The spanwise length of the FW-H surface matches that of the computational domain. Additionally, reflections from the boundaries are mitigated through the use of a sponge layer. The viscosity in the sponge layer is set to 100 times the physical value, following Himeno et al. [25].

A multiblock structure mesh with C-type topology is generated, resulting in a total cell count exceeding 43 million. Each plane of the 2.5D mesh consists of approximately 0.25 million cells, and the entire mesh

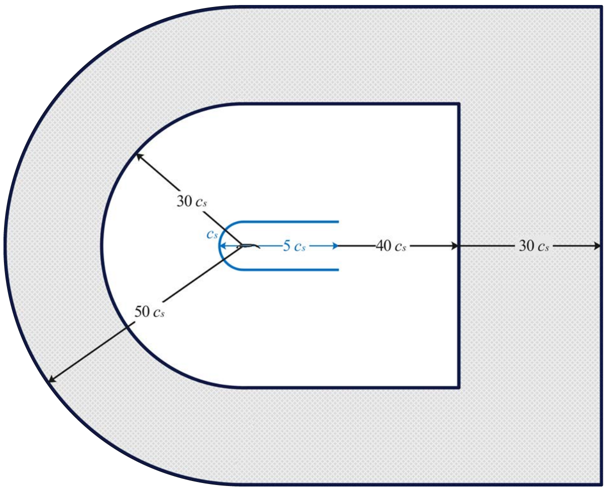
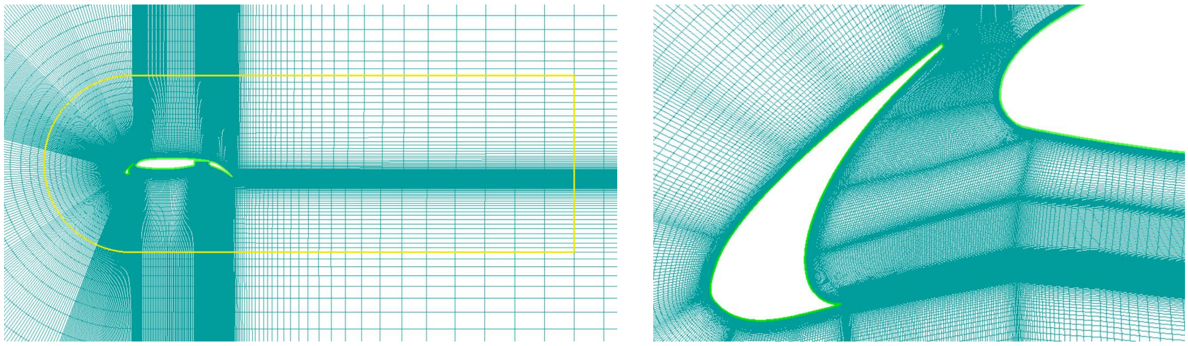


Figure 10: Spanwise cross-section of the computational domain (not in scale). The grey area represents the sponge layer, and the permeable FW-H data surface is indicated by the blue line.



(a) (b)

Figure 11: Mesh details of (a) permeable FW-H data surface, (b) slat.

comprises 175 planes with uniform spacing in the spanwise direction. Close-up views of the mesh around the FW-H surface and slat cove area are presented in Fig. 11. Furthermore, the average dimensionless wall distance y^+ of the mesh is below unity.

The well-established Roe scheme is utilized for the decomposition of the inviscid flux, in conjunction with the third-order MUSCL scheme for variable reconstruction. The implicit dual-time LU-SGS method is implemented for time advancement, employing a time step of $2 \times 10^{-7} s$. To expedite the convergence process, five inner sub-iterations are used, along with a local time-stepping approach. The FW-H data sampling frequency is 10 kHz. The flow field is initialized using a RANS solution with the SA turbulence model. Subsequently, the IDDES-SA simulation is carried out and approximately $0.1s$ of physical time is obtained, with 8 flow pass times (FPTs, based on c_s) available for data processing after the removal of the initial transient.

A time-averaged lift coefficient (C_L) of 2.6214 is obtained, which differs by under 2.3% from the BANC-III average of $C_L = 2.6821$. Fig. 12 illustrates the distribution of the time-averaged C_p in comparison with experimental data [21, 23]. While there is a slight underestimation of negative pressure on the suction side, a reasonably good agreement is observed, particularly in the slat cove region when compared to the JAXA experiment [23].

To address the differences in the spanwise acoustic integration length between the numerical and experi-

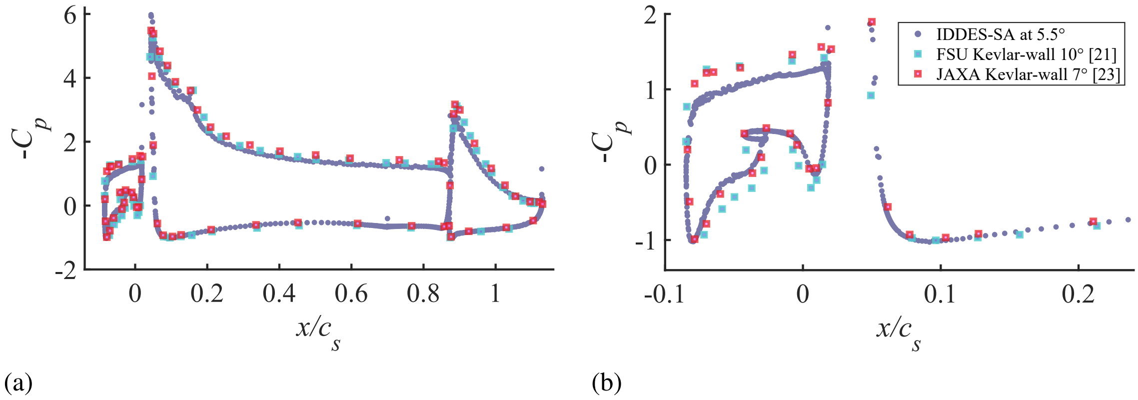


Figure 12: Time-averaged C_p distribution at the airfoil midplane of (a) all three elements, (b) around slat.

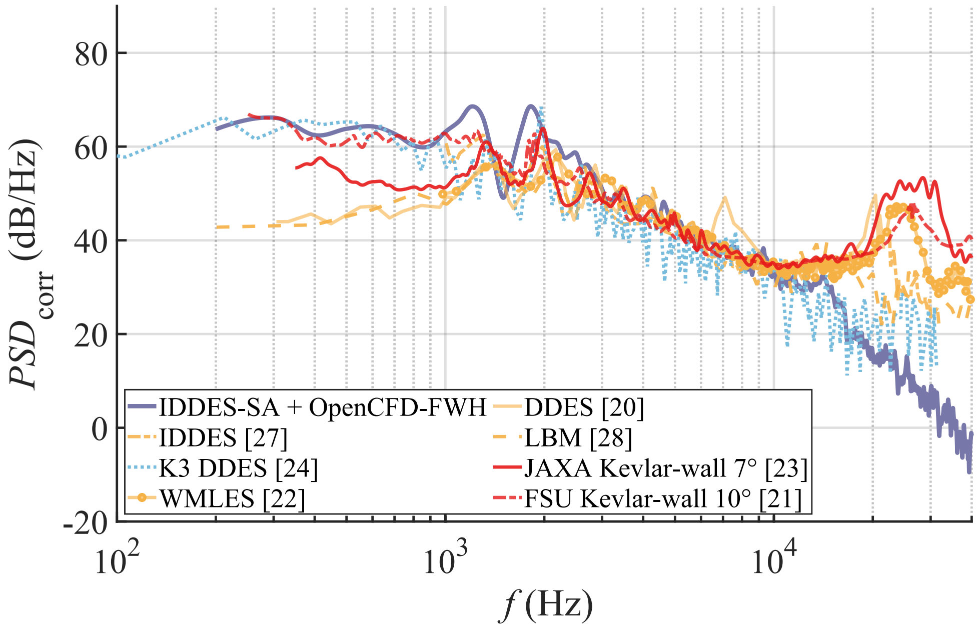


Figure 13: PSD of acoustic signals at the center of the JAXA phased-microphone array [23].

mental configurations, the scaling approach of Avallone et al. [33] is employed:

$$PSD_{corr} = PSD + 10\log_{10}\left(\frac{b_{exp}}{b_m}\right), \tag{53}$$

where b_{exp} and b_m are 650mm and 50.8mm, respectively.

The pressure signal acquired from OpenCFD-FWH is segmented into blocks with a 50% overlap and is multiplied by a Hanning window, yielding a frequency resolution of 100 Hz. A total of 6,535 sampling frames are input to OpenCFD-FWH, and the execution time with various parallel strategies is detailed in Table 1. The Power Spectral Density (PSD) result of OpenCFD-FWH is plotted in Fig. 13. Reasonably good agreement with the experimental data [21, 23] and numerical solutions [20, 22, 24, 27, 28] below 10 kHz is achieved, with the exception of a slightly higher level at the first tonal peak. Additionally, the humping noise originating from the high-frequency vortex shedding from the slat trailing-edge of the reduced-scale wind tunnel model is absent in the OpenCFD-FWH result. This is attributed to the relatively coarse mesh at the slat trailing-edge, which is incapable of capturing the high-frequency vortex shedding. Overall, the result validates that the far-field noise can be accurately calculated by OpenCFD-FWH.

5. Conclusions

This paper presents the methodology, parallel implementation, and validation of a FW-H post-processing code, OpenCFD-FWH. It is specifically developed for predicting far-field noise under incident mean flow, which covers a wide range of scenarios encountered in aircraft CFD simulations. The OpenCFD-FWH code is developed for post-processing OpenCFD-EC flow field results. However, it is compatible with other solvers provided the right FW-H dataset format or the data reading module of the code is modified accordingly. Notably, the deployment of OpenCFD-FWH only necessitates an MPI library and a Fortran 90 compilation environment, without the need of OpenCFD-EC.

OpenCFD-FWH is based on the integration formulation of a non-dimensional FW-H equation for permeable surface, with convective and AoA effects corrected by the Garrick triangle and 2D plane coordinate transformation, respectively. This formulation enhances computational efficiency compared to the original. Furthermore, the non-dimensionalization of the FW-H equation is consistent with that used for the Navier-Stokes equations in the OpenCFD-EC solver.

To speed up post-processing and lower per-node memory usage, OpenCFD-FWH employs MPI/OpenMP hybrid parallelization in distributed computing environments. The support of MPI prevents memory exhaustion when handling large datasets, as often encountered in LES simulations of landing gear and high-lift devices. Performance tests of OpenCFD-FWH are conducted on 31 nodes of the CAS SunRising platform, with 32 OpenMP threads per node. The results demonstrate a 538.5 times acceleration in computing time compared to serial execution.

Three validation benchmarks are presented in this paper. The monopole and dipole benchmarks are compared with exact analytical solutions, showing excellent agreement. The 30P30N benchmark is first simulated using the OpenCFD-EC solver with IDDES-SA. Then the far-field noise is predicted by OpenCFD-FWH. The PSD result for frequencies below 10 kHz demonstrates relatively good agreement with multiple experimental and numerical results reported in previous studies.

Moreover, OpenCFD-FWH is openly accessible on GitHub, along with the MATLAB scripts for the monopole and dipole validation benchmarks, to facilitate its utilization by users.

A. File structures for OpenCFD-FWH

A.1 An example of the control.fwh file

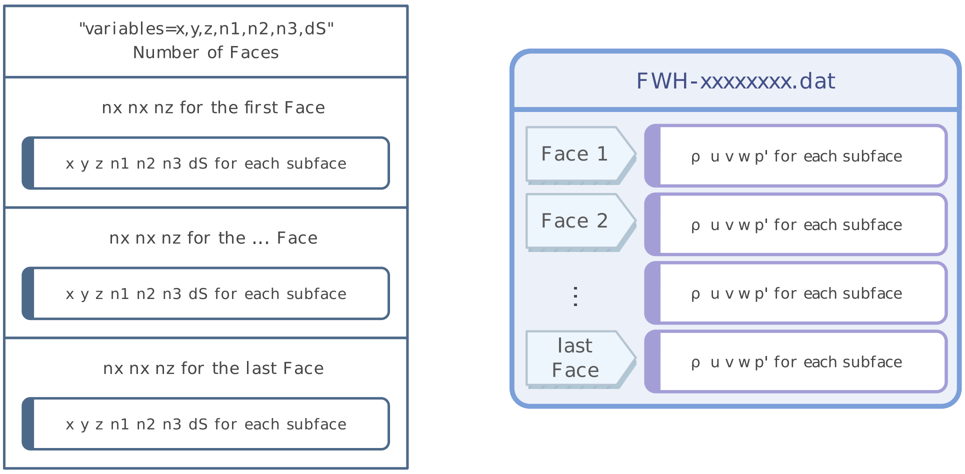
OpenCFD-FWH employs the namelist method to parse the control.fwh file. An example of this file is presented in Fig. A1, showcasing the default settings in OpenCFD-FWH. The parameters Kstep_start and Kstep_end define the initial and final steps for FW-H post-processing calculations, respectively. By utilizing the step interval, delta_step, OpenCFD-FWH can compute the number of sampling frames. The parameter FWH_data_Format specifies the format of the FW-H dataset. A value of 0 denotes binary format, while a value of 1 indicates ASCII format.

A.2 Structure of the FWH_Surface_Geo.dat file

The FWH_Surface_Geo.dat file is in ASCII format, which is convenient for data-checking purposes. Its structure is akin to that of the Generic boundary description .inp file. The file commences with an initial line of text as "variables=x,y,z,n1,n2,n3,dS," as illustrated in Fig. A2a. The subsequent line contains a singular numerical value indicating the total number of Faces. Following this, the nx, ny, nz values for the Face are provided, along with the $x, y, z, n1, n2, n3, dS$ values for its subfaces. This pattern repeats until all Faces are described.

```
$control_FWH
  Ma=0.2d0
  AoA=0.d0
  delta_t=5.8d-1
  NUM_Obs=1
  Kstep_start=1
  Kstep_end=100
  delta_step=1
  NUM_THREADS=5
  FWH_data_Format=0 ! binary file
$end
```

Figure A1: An example of the control.fwh file.



(a) (b)

Figure A2: A schematic of (a) FWH_Surface_Geo.dat file, (b) FWH-xxxxxxxx.dat file.

A.3 Structure of the FW-H dataset

The FW-H dataset used in OpenCFD-FWH contains multiple files named FWH-xxxxxxxx.dat, where "xxxxxxxx" denotes the iteration number. These files are stored either in ASCII or binary format. Binary is preferred due to its significantly smaller size. The file structure is illustrated in Fig. A2b.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) under grant number 12272024.

References

[1] Eduardo Molina, Beckett Yx Zhou, Juan J Alonso, Marcello Righi, and Roberto G Silva. Flow and noise predictions around tandem cylinders using ddes approach with su2. In *AIAA Scitech 2019 Forum*, page 0326, 2019.

[2] Xuyang Ma, Yuejun Shi, and Wenbin Song. Aerodynamic and aeroacoustic analysis of scch models of four high-lift configurations near stall angle of attack. In *AIAA aviation 2020 forum*, page 2508, 2020.

[3] Ya-Sen Hu, Zhen-Hua Wan, Chuang-Chao Ye, De-Jun Sun, and Xi-Yun Lu. Noise reduction mechanisms for insert-type serrations of the naca-0012 airfoil. *Journal of Fluid Mechanics*, 941:A57, 2022.

[4] Guanjiang Chen, Bin Zang, and Mahdi Azarpeyvand. Numerical investigation on aerodynamic noise of flow past a cylinder with different spanwise lengths. *Physics of Fluids*, 35(3), 2023.

[5] Daniel S Souza, Daniel Rodríguez, Fernando HT Himeno, and Marcello AF Medeiros. Dynamics of the large-scale structures and associated noise emission in airfoil slats. *Journal of Fluid Mechanics*, 875:1004–1034, 2019.

[6] C Teruna, F Avallone, D Ragni, A Rubio-Carpio, and D Casalino. Numerical analysis of a 3-d printed porous trailing edge for broadband noise reduction. *Journal of Fluid Mechanics*, 926:A17, 2021.

[7] Jacob M Turner and Jae Wook Kim. Quadrupole noise generated from a low-speed aerofoil in near-and full-stall conditions. *Journal of Fluid Mechanics*, 936:A34, 2022.

[8] Andrey Epikhin. Validation of the developed open source library for far-field noise prediction. In *Advances in Acoustics, Noise and Vibration-2021, Proceedings of the 27th International Congress on Sound and Vibration, ICSV*, 2021.

[9] Beckett Y Zhou, Tim Albring, and Nicolas R Gauger. Aeroacoustic prediction and optimization capabilities in SU2.

[10] Weiqi Shen and Steven AE Miller. Validation of a high-order large eddy simulation solver for acoustic prediction of supersonic jet flow. *Journal of Theoretical and Computational Acoustics*, 28(03):1950023, 2020.

[11] Xinliang Li, Dexun Fu, and Yanwen Ma. Direct numerical simulation of hypersonic boundary layer transition over a blunt cone with a small angle of attack. *Physics of Fluids*, 22(2), 2010.

[12] Keli Zhang, Peiqing Liu, Shizhi Lin, Kai Liu, and Yi Lu. Sweep effects on slat tonal noise. *AIAA Journal*, 64(6):3399–3420, 2026.

[13] IE Garrick and Charles E Watkins. A theoretical study of the effect of forward speed on the free-space sound-pressure field around propellers. Technical report, 1953.

[14] John E Ffowcs Williams and David L Hawkings. Sound generation by turbulence and surfaces in arbitrary motion. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 264(1151):321–342, 1969.

[15] Fereidoun Farassat. Derivation of formulations 1 and 1a of farassat. Technical report, 2007.

[16] Kenneth S Brentner and Feri Farassat. Analytical comparison of the acoustic analogy and kirchhoff formulation for moving surfaces. *AIAA journal*, 36(8):1379–1386, 1998.

[17] Guillaume Brès, Franck Pérot, and David Freed. A ffowcs williams-hawkings solver for lattice-boltzmann based computational aeroacoustics. In *16th AIAA/CEAS aeroacoustics conference*, page 3711, 2010.

[18] Alireza Najafi-Yazdi, Guillaume A Brès, and Luc Mongeau. An acoustic analogy formulation for moving sources in uniformly moving media. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 467(2125):144–165, 2011.

[19] Meelan M Choudhari and David P Lockard. Assessment of slat noise predictions for 30p30n high-lift configuration from banc-iii workshop. In *21st AIAA/CEAS aeroacoustics conference*, page 2844, 2015.

[20] Marc Terracol, Eric Manoha, Mitsuhiro Murayama, and Kazuomi Yamamoto. Aeroacoustic calculations of the 30p30n high-lift airfoil using hybrid rans/les methods: modeling and grid resolution effects. In *21st AIAA/CEAS Aeroacoustics Conference*, page 3132, 2015.

[21] Kyle A Pascioni and Louis N Cattafesta. Aeroacoustic measurements of leading-edge slat noise. In *22nd AIAA/CEAS aeroacoustics conference*, page 2960, 2016.

[22] Yufei Zhang, Haixin Chen, Kan Wang, and Meng Wang. Aeroacoustic prediction of a multi-element airfoil using wall-modeled large-eddy simulation. *AIAA Journal*, 55(12):4219–4233, 2017.

[23] Mitsuhiro Murayama, Yuzuru Yokokawa, Hiroki Ura, Kazuyuki Nakakita, Kazuomi Yamamoto, Yasushi Ito, Takehisa Takaishi, Ryotaro Sakai, Keiji Shimoda, Takayuki Kato, and Tomoyuki Homma. Experimental study of slat noise from 30p30n three-element high-lift airfoil in jaxa kevlar-wall low-speed wind tunnel. In *2018 AIAA/CEAS Aeroacoustics Conference*, page 3460, 2018.

[24] Junhui Gao, Xiaodong Li, and Dakai Lin. Numerical simulation of the 30p30n high-lift airfoil noise with spectral difference method. *AIAA Journal*, 58(6):2517–2532, 2020.

[25] Fernando HT Himeno, Daniel S Souza, Filipe R Amaral, Daniel Rodríguez, and Marcello AF Medeiros. Spod analysis of noise-generating rossiter modes in a slat with and without a bulb seal. *Journal of Fluid Mechanics*, 915:A67, 2021.

[26] Jiang-Sheng Wang and Jin-Jun Wang. Vortex dynamics for flow around the slat cove at low reynolds numbers. *Journal of Fluid Mechanics*, 919:A27, 2021.

[27] Guoyong Chen, Xiaolong Tang, Xiaoquan Yang, Peifen Weng, and Jue Ding. Noise control for high-lift devices by slat wall treatment. *Aerospace Science and Technology*, 115:106820, 2021.

[28] Kazuomi Yamamoto, Mitsuhiro Murayama, Kazuhide Isotani, Yosuke Ueno, Kensuke Hayashi, and Tohru Hirai. Slat noise reduction based on turbulence attenuation downstream of shear-layer reattachment. In *28th AIAA/CEAS Aeroacoustics 2022 Conference*, page 2954, 2022.

[29] Hiroyuki Asada, Seiya Sato, Naoki Miwa, and Soshi Kawai. Aeroacoustics simulations using kinetic-energy and entropy preserving (keep) schemes. In *30th AIAA/CEAS Aeroacoustics Conference (2024)*, page 3244, 2024.

[30] Keli Zhang, Shizhi Lin, Peiqing Liu, Shihao Liu, Yuping Ren, Kai Liu, Yi Lu, and Yu Lv. Aeroacoustic control of slat noise: Dependence of noise reduction on angle-of-attack using a novel staggered serration design. In *32nd AIAA/CEAS Aeroacoustics Conference (2026)*, page 3208, 2026.

[31] Keli Zhang, Shizhi Lin, Peiqing Liu, Zhibo Zhang, Shihao Liu, Can Fang, Kai Liu, Yi Lu, and Yu Lv. Cfd-based aerodynamic and aeroacoustic optimization of the 30p30n high-lift airfoil. In *32nd AIAA/CEAS Aeroacoustics Conference (2026)*, page 3256, 2026.

[32] Keli Zhang, Zhibo Zhang, Shizhi Lin, Peiqing Liu, and Shihao Liu. Multiple deep learning surrogate models for aeroacoustic prediction of the 30p30n high-lift airfoil. In *32nd AIAA/CEAS Aeroacoustics Conference (2026)*, page 3450, 2026.

[33] Francesco Avallone, Wouter CP Van der Velden, and Daniele Ragni. Benefits of curved serrations on broadband trailing-edge noise reduction. *Journal of Sound and Vibration*, 400:167–177, 2017.