Implementation of a highly scalable aeroacoustic module based on the Ffowcs-Williams and Hawkings analogy within the open-source CFD software *Code_Saturne*

Stefano Rolfo, Charles Moulinec and David R. Emerson

Scientific Computing Department, STFC Daresbury Laboratory, Warrington

24th October 2016

Abstract

This work presents the implementation of a Ffowcs-Williams and Hawkings acoustic analogy within *Code_Saturne*. The implementation takes into account surface (loading and thickness) and volume (quadrupole) integrals. It relies on an advance-in-time formulation to avoid disk storage of large amount of data for the acoustic propagation. The formulation is verified against an analytical test case using 360 receivers and the scalability of the module is assessed for configurations made of up to 1.2 billion cells and 100 receivers, using up to 49,152 cores on ARCHER.

1. Introduction

Predicting noise is important for many engineering applications ranging from aerospace (rotor noise generated by helicopters and turboprops, for instance) to combustion, to car manufacture industry and to electrical engineering as for electrical appliances. The noise evolution can be directly computed by the compressible Navier-Stokes equations through Direct Numerical Simulation (DNS). However, DNS is a very computationally demanding approach where all length and time scales of the turbulent spectrum have to be resolved. For this reason DNS is still limited, industry-wise, to moderate Reynolds numbers and relatively simple geometries. Noise predictions further increase the computational requirements since noise levels might be required very far away from the noise source locations. As a consequence, very large computational domains, which might not be necessary from a pure hydrodynamic point of view, are mandatory. A very well established technique to reduce computational demand consists of separating the computation of the noise sources, which can be performed using different computational fluid dynamics (CFD) methodologies, from the propagation of the sound itself, assuming there is no feedback from the noise generation. This assumption is well verified in the case of subsonic flows (i.e. Mach number (M) less than 1) and when the noise sources are compact with the sound propagating into a fluid at rest (a compact source is a source region which is small compared to the acoustic wave length). This strategy for modelling noise is referred to as hybrid computational aeroacoustics (CAA) and the theory is known as the

aeroacoustic analogy.

2. Module description

2.1. Equations

The Ffowcs-Williams and Hawkings (FWH) equation [3] is the most general form of the Lighthill's acoustic analogy [4] and can be derived as an exact rearrangement of the Navier-Stokes equations. The method is based on the definition of an arbitrary control surface $f = 0$ which encloses the fluid (this corresponds to $f < 0$). The configuration can be represented by the fluid at rest plus a distribution of sources at $f = 0$ (see figure 1).

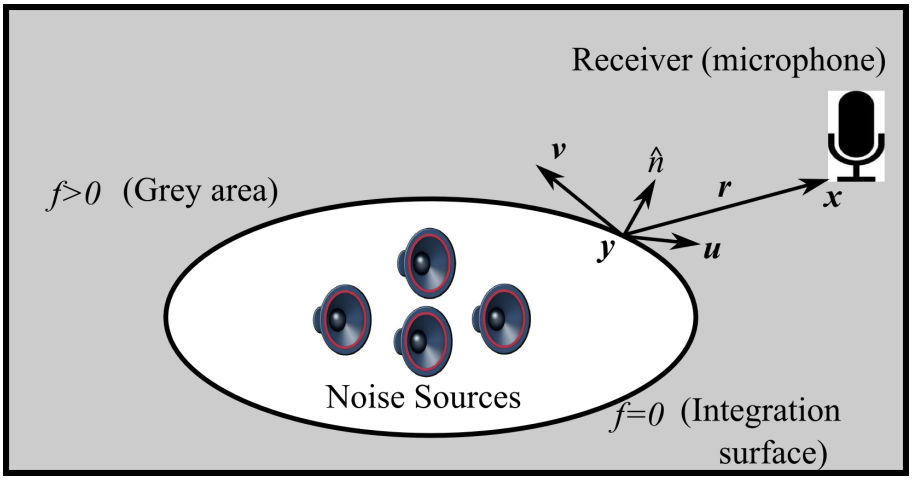


Figure 1: Sketch of the FWH analogy.

The FWH continuous equation takes the form of an inhomogeneous wave equation (see Appendix A for the full nomenclature) and reads:

$$\frac{1}{c^2}\frac{\partial^2 p'}{\partial t} - \nabla^2 p' = \Box^2 p' = \frac{\partial^2 T_{ij} H(f)}{\partial x_i \partial x_j} - \frac{\partial L_i \delta(f)}{\partial x_i} + \frac{\partial \rho_0 U_i \hat{n}_i \delta(f)}{\partial t} \tag{1}$$

with H being the Heaviside function $H(f) = \begin{cases} 1 \text{ if } f > 0 \\ 0 \text{ if } f \leq 0 \end{cases}$ and $\delta(f) = \begin{cases} 1 \text{ if } f = 0 \\ 0 \text{ if } f \neq 0 \end{cases}$.

T_{ij} is the Lighthill's stress tensor

$$T_{ij} = \rho u_i u_j - \tau_{ij} + \left(p' - c^2\rho'\right)\delta_{ij} \tag{2}$$

L_i the linear momentum

$$L_i = P_{ij}\hat{n}_j + \rho u_i (u_n - v_n)$$

and

$$U_i = \left(1 - \frac{\rho}{\rho_o}\right) v_i + \frac{\rho u_i}{\rho_0}$$

An integral formulation of the solution has been proposed by Farassat and Succi [2], where the acoustic pressure fluctuations p' can be expressed as the summation of three different contributions:

$$p' = p'_L + p'_T + p'_Q \tag{3}$$

where p'_L is the loading noise, p'_T the thickness noise and p'_Q the quadrupole noise. The analytical definition of the different terms reads:

$$4\pi p'_L(\mathbf{x},t) = \frac{1}{c}\int_{f=0}\left[\frac{\dot{L}_r}{r(1-M_r)^2}\right]_{ret} dS + \int_{f=0}\left[\frac{L_r - L_m}{r^2(1-M_r)^2}\right]_{ret} dS + \frac{1}{c}\int_{f=0}\left[\frac{L_r\left(r\dot{M}_r + c\left(M_r - M^2\right)\right)}{r^2(1-M_r)^3}\right]_{ret} dS \tag{4}$$

$$4\pi p'_T(\mathbf{x},t) = \int_{f=0}\left[\frac{\rho_0\left(\dot{U}_n + U_{\dot{n}}\right)}{r(1-M_r)^2}\right]_{ret} dS + \int_{f=0}\left[\frac{\rho_0 U_n\left(r\dot{M}_r + c\left(M_r - M^2\right)\right)}{r^2(1-M_r)^3}\right]_{ret} dS \tag{5}$$

$$4\pi p'_Q(\mathbf{x},t) = \int_{f>0}\left[\frac{K_1}{c^2 r} + \frac{K_2}{cr^2} + \frac{K_3}{r^3}\right]_{ret} dV \tag{6}$$

with

$$K_1 = \frac{\ddot{T}_{rr}}{(1-M_r)^3} + \frac{\ddot{M}_r T_{rr} + 3\dot{M}_r \dot{T}_{rr}}{(1-M_r)^4} + \frac{3\dot{M}_r^2 T_{rr}}{(1-M_r)^5}$$

$$K_2 = \frac{-\dot{T}_{ii}}{(1-M_r)^2} - \frac{4\dot{T}_{Mr} + 2T_{\dot{M}r} + \dot{M}_r^2 T_{rr}}{(1-M_r)^3} + \frac{3\left[\left(1-M^2\right)\dot{T}_{rr} - 2\dot{M}_r T_{Mr} - M_i\dot{M}_i T_{rr}\right]}{(1-M_r)^4} + \frac{6\dot{M}_r\left(1-M^2\right)T_{rr}}{(1-M_r)^5}$$

$$K_3 = \frac{2T_{MM} - \left(1 - M^2\right) T_{ii}}{\left(1 - M_r\right)^3} - \frac{6\left(1 - M^2\right) T_{Mr}}{\left(1 - M_r\right)^4} + \frac{3\left(1 - M^2\right) T_{rr}}{\left(1 - M_r\right)^5}$$

The $\dot{}$ stands for the first time derivative (i.e. $\dot{L} = \partial L / \partial t$) and the subscript $_r$ indicates the projection of a vector in the $\hat{r}_i$ direction (i.e $L_r = L_i \cdot \hat{r}_i$).

2.2. Advance-in-time formulation

The retarded time formulation presented above implies that the formulation is expressed in terms of reception time at the receiver. This means that for a given receiver time t the disturbances are emitted at different times t_{ret} depending on the relative position between the receiver and the source. t_{ret} is expressed as:

$$t_{ret} = t - \frac{\|\mathbf{x}(t) - \mathbf{y}(t_{ret})\|}{c} = \frac{r(t_{ret})}{c}$$

where c is the speed of sound, $\mathbf{x}(t)$ is the receiver position and $\mathbf{y}(t)$ is the source position (see figure 1 for a graphical definition of the parameters). To circumvent this problem an advance-in-time formulation is adopted, following the one proposed by Casalino [1]. In this case the time used to compute the acoustics is the emission time and the time at which the disturbance reaches the receiver is computed using:

$$t_{adv} = t + \mathcal{T} = t + \frac{r(t)}{c} \tag{7}$$

With this formulation, only one instance of the CFD field is required and the memory requirement is transferred to the receiver, the number of receivers (about a few hundreds at most) being in general much smaller than the size of the CFD mesh.

2.3. Module implementation

Two different implementations of the module have been tested, the main difference between them being the memory allocation for the time varying acoustic variables at the receiver point (i.e. $p'_T(\mathbf{x}, t)$). For every acoustic variable an uneven bidimensional array (i.e. a non-square matrix) is allocated where the number of columns is the number of receivers,

and the length of each column m_i is computed as:

$$m_i = \left\lfloor \frac{1}{\delta t} \frac{r_{i,max}}{c} \right\rfloor \text{with } i = 1, ..N_{rec} \tag{8}$$

with δt being the acoustic time step, $r_{i,max}$ the maximum distance between the i^{th} receiver and the sources, c the speed of sound and $\lfloor\ \rfloor$ the floor function. The length of each columns m_i depends on the maximum distance between the noise source distribution and the i^{th} receiver position and can be different for each receiver. In the first implementation (see Algorithm 1) the same memory is allocated to each MPI rank. This implementation has the advantage that parallel summation to compute the integrals (4), (5) and (6) is only performed when the results are written to the disk. However, the main drawback of this algorithm is that it keeps in memory a large amount of data. With Algorithm 2 the memory requirements are alleviated since the receivers are evenly distributed across the MPI ranks, but with the disadvantage of requiring more frequent MPI operations. When computing loading and thickness noise only, the implementation without distributing the receivers across the MPI tasks is still acceptable even for a few hundreds receivers, whereas when the quadrupole contribution is also considered it is mandatory to evenly distribute them across the MPI tasks. A more detailed description of the two implementations is given in Appendix B.

3. Results

This section reports results of the module implementation as well as a comparison of the scalability of the code without and with the FWH module. The validation is conducted using the analytical test case of the scattering of a wave plane from a rigid cylinder (see section §3.1.) and the numerical test case of a flow around a circular cylinder at $Re = 150$ (see section §3.2.). The number of receivers is set to 360 for these cases. For the scalability tests the Reynolds number has been increased to $3,900$ to allow the use of Large-Eddy Simulation (LES) with meshes having up to 1.2 billion cells. The number of receivers is set to 100. All the simulations involving the new acoustic module have been

performed using the compressible module of *Code_Saturne*.

3.1. Scattering of a plane wave by a rigid cylinder

The first test case to validate the method deals with the scattering of a plane wave impinging on a rigid cylinder. Figure 2 shows a sketch of the problem. This test case has an analytical solution. The resulting pressure is a function of the time, the distance from the centre of the cylinder and the angle measured from the direction of the incoming wave. Its expression reads:

$$p'(r,t,\theta) \simeq -P\sqrt{\frac{D}{\pi r}}\psi_s(\theta)\exp(ik(r-ct)) \tag{9}$$

with $\psi_s(\theta) = \sqrt{\frac{2}{kD}}\sum_{m=0}^{\infty}\varepsilon_m \sin(\gamma_m)\exp(-i\gamma_m)\cos(m\theta)$. For $m = 0$, $\varepsilon_0 = 1$, whereas for $m > 0$, $\varepsilon_m = 2$. The pressure on the solid cylinder surface has also an analytical definition which reads:

$$p_w(\theta,t) = \frac{8P}{\pi kD}\exp(ickt)\sum_{m=0}^{\infty}\frac{\cos(m\theta)}{E_m}\exp\left(i\left(\frac{\pi m}{2}-\gamma_m\right)\right) \tag{10}$$

where P is the wave amplitude, k the wave number and D the cylinder diameter. The amplitude E_m and the phase angle γ_m are combinations of Bessel functions and their value for a case of $kD = 6$ is given in table 1. This table reports only the first nine elements of the series that have been used for the definition of the analytical pressure at the wall.

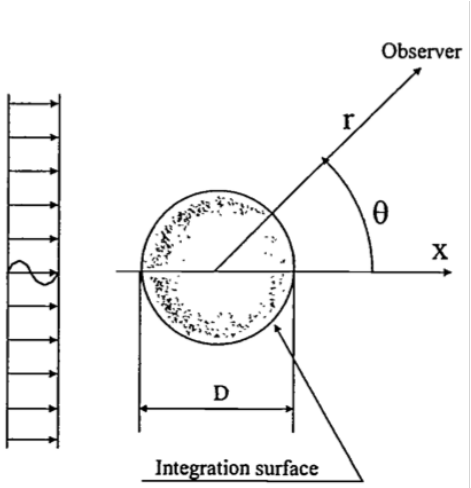
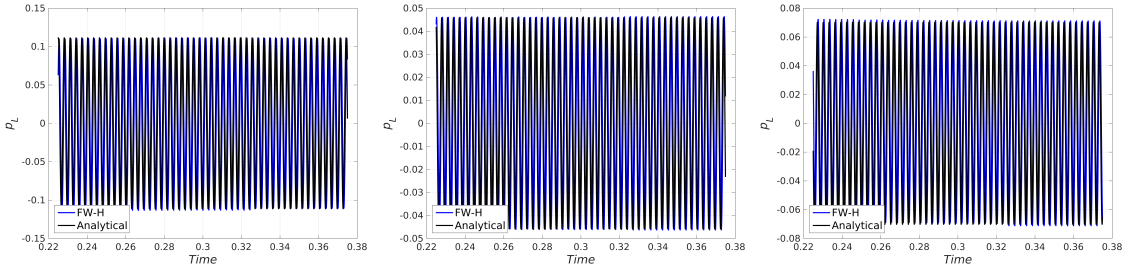


Figure 2: Sketch of the scattering of a plane wave by a rigid cylinder.

m	$\gamma\,[^o]$	E	m	$\gamma\,[^o]$	E
0	133.76	0.9389	5	-1.53	2.2610
1	54.24	0.4597	6	-0.13	8.967
2	-1.99	0.4319	7	0.00	40.86
3	-25.09	0.4175	8	0.00	212.6
4	-11.01	0.6965	9	0.00	1249

Table 1: Phase angle and amplitude for $kD = 6$.

This specific case is a linear problem, therefore only surface integrals of equations (4) and (5) are considered[1] and the integration surface is coincident with the cylinder wall. Figure 3 reports the time series of the scattered pressure at three different locations (i.e. $\theta = 0^0,\ 90^0$ and 180^0), showing a very good agreement between the FWH analogy and the analytical solution. This very good agreement is also confirmed by the directivity pattern presented in figure 4.



(a) $\theta = 0^o$ (b) $\theta = 90^o$ (c) $\theta = 180^o$

Figure 3: Schematic of the FWH analogy.

[1]The loading noise is actually the only contribution since the velocity is 0 which implies that the thickness noise is also 0.

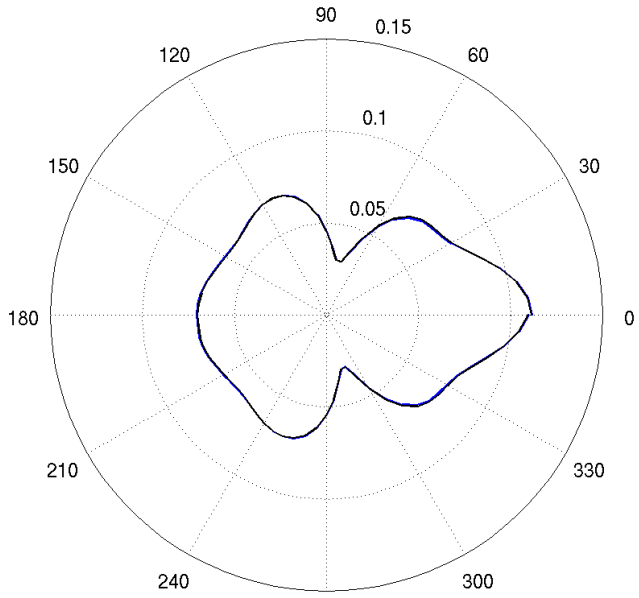


Figure 4: Directivity pattern for the scattered pressure from a rigid cylinder.

3.2. Flow around a circular cylinder

The second test case is the flow around a circular cylinder. The flow is considered as laminar, unsteady and bidimensional (i.e. $Re = 150$) to test the linear part of the propagation. Three Mach numbers have been used, i.e. $M = 0.1$, 0.2 and 0.3 respectively. Two 2D meshes have been generated, M01 which has $27,000$ cells in the cross plane and 176 cells around the cylinder and, M02 which has $105,000$ cells in the cross plane and 384 cells around the cylinder. Firstly for validation purposes, values of the averaged drag coefficients C_D and Strouhal number St computed using the *Code_Saturne*'s compressible module, are compared with the results of a *Code_Saturne*'s incompressible simulation in the same conditions and with the data from literature. Table 2 shows the values of C_D and the Strouhal number for different calculations using *Code_Saturne*'s incompressible

and compressible algorithms for several Mach numbers. A good agreement is observed between both compressible and incompresible approaches and with the reference data. The power spectral density of the time varying lift coefficient is also computed and the comparison between the different calculations is shown in figure 5.

	Ref	M01 $\rho = const$	M01 $M = 0.1$	M02 $M = 0.1$	M02 $M = 0.2$	M02 $M = 0.3$
C_D	1.33	1.326	1.300	1.330	1.363	1.35
St	0.182	0.183	0.181	0.186	0.183	0.19

Table 2: Comparison of the drag coefficient C_D and Strouhal number St for the flow around a cylinder at $Re = 150$. The reference value of the Strouhal number is computed by Roshko's formula [5].

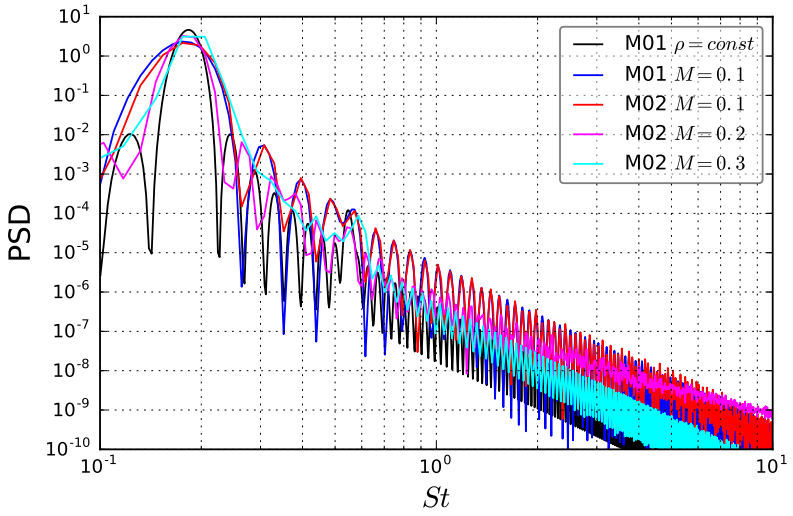
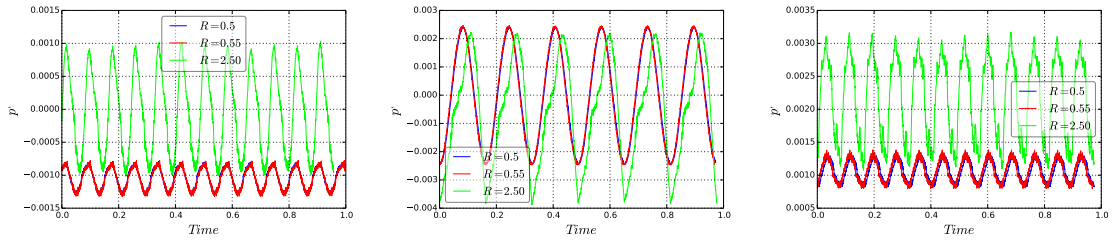


Figure 5: Comparison of the power spectral density of the lift coefficients C_L for different cases of the flow around a laminar cylinder at $Re = 150$.

One of the advantages of the FWH formulation over any other acoustic analogy is the possibility to locate the control surface in any part of the flow domain and not only in the linear region of the acoustic wave propagation. Figure 6 shows the time history of

the acoustic pressure p' at three different angular locations and at a distance $R_x = 50D$ from the cylinder centre. Three different integration surfaces are located at $R_1 = 0.5D$ (solid wall), $R_2 = 0.55D$ and $R = 2.5D$ (see figures 6a, 6b and 6c). Results obtained from the first two integration surfaces are in very good agreement between each other showing the good implementation of the permeable formulation. This is also confirmed by the directivity pattern of figure 7a. Results from further away from the wall integration surface are instead characterised by higher fluctuations. This could be caused by the fact that the CFD schemes are second order in space and time and the mesh resolution is not fine enough. This deficiency of the method has also been highlighted by Casalino [1]. To test 3D meshes and verify the influence of the spanwise width and its resolution, the 2D mesh M01 has been extruded in the third direction with two different lengths and number of cells. Directivity patterns for the two cases are shown in figures 7b and 7c presenting a good agreement between the 2D and 3D formulations.

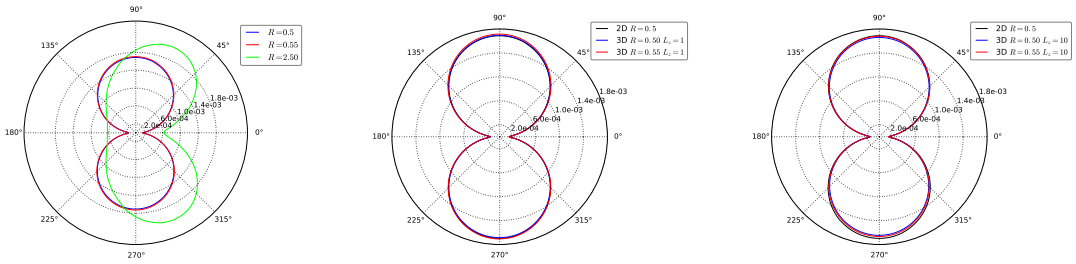


(a) $\theta = 0^o$ (b) $\theta = 90^o$ (c) $\theta = 180^o$

Figure 6: Time history of the acoustic pressure $p' = p'_L + p'_T$ for the flow around a cylinder at $Re = 150$ and $M = 0.1$. The receivers are located at a distance $R = 50D$ from the cylinder centre and the results are obtained using M01.

3.3. Scalability of the code

The scalability of *Code_Saturne* with the new module is tested using different meshes of different sizes. The number of receivers is set to 100 for all the simulations. Except for the largest tests using a 1.2 billion cell mesh, all the other tests are carried out using 3 configurations, which performance are compared against each other: the first one using



(a) Comparison between different integration surfaces.

(b) Comparison between 2D and 3D meshes. The 3D one is obtained by extruding M01 with 5 layers to get a total span length of $L_z = 1D$.

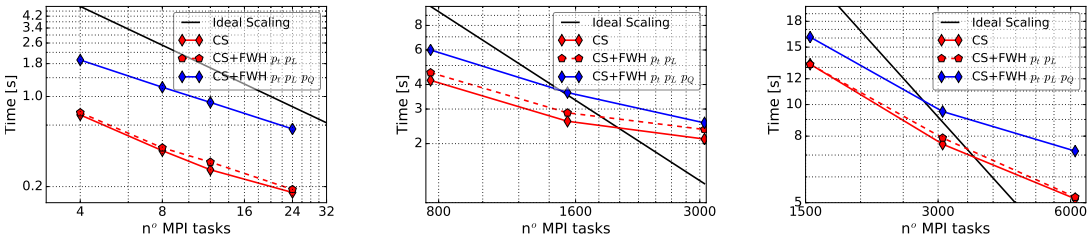
(c) Comparison between 2D and 3D meshes. The 3D one is obtained by extruding M01 with 51 layers to get a total span length of $L_z = 10D$.

Figure 7: Directivity pattern for the flow around a cylinder at $Re = 150$ and $M = 0.1$. The receivers are located at a distance of $R = 50D$ from the cylinder centre. The results are obtained using M01 and different extrusions in the spanwise direction.

Code_Saturne alone, the second one with the addition of the FWH module with only loading and thickness noise and, the third one using also the quadrupole noise. Using the FWH module produces very similar scaling profiles as the ones of *Code_Saturne* alone as can be seen in figure 8. Figure 8a shows the scalability for the flow around a circular cylinder at $Re = 150$ and $M = 0.2$ using a bidimensional mesh made of 105,000 cells (M02). The addition of the computation of the loading and thickness noise increases the CPU time by 3% to 10%. The addition of the quadrupole integral increases the CPU time by 10% to 30%, depending on the number of cores being used.

Similar increases in CPU times are also observed for larger meshes of 38 millions cells (see figure 8b) and 150 million cells (see figure 8c), respectively. For these cases the Reynolds number has been increased to 3,900 while the Mach number has been kept to 0.2. The 2D cross plane mesh is made of about 146,000 cells and several extrusion layers are used to build the 3D meshes. The cases with the 38 million (see figure 9a) and 150 million cell meshes (see figure 9b) have also been tested on an IBM Blue Gene/Q up to 32,768 MPI tasks to demonstrate the portability of the code with the added module. Similar performance as the ones obtained on ARCHER are observed. Finally the full module has been tested on a very large mesh of 1.2 billion cells up to 49,152 MPI tasks,

which corresponds to 44% of the full system (see figure 10). Very good scalability has been observed in this latter case with an efficiency of 87% going from 24,576 to 49,152 MPI tasks.

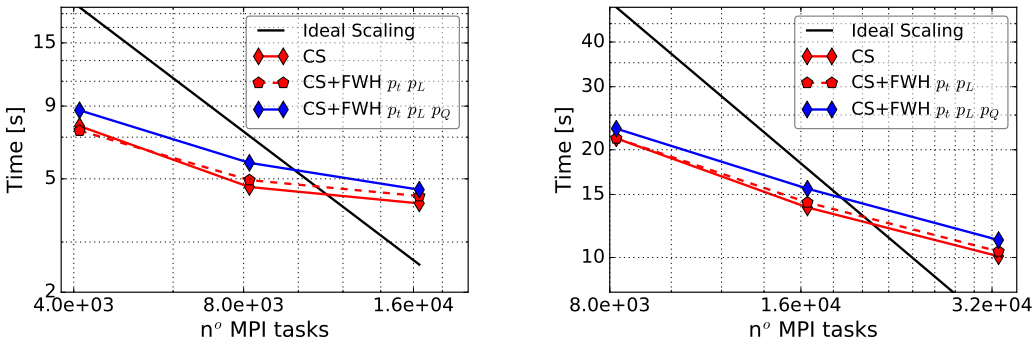


(a) Scalability test for the 2D mesh made of 105,000 cells.

(b) Scalability test for the 3D mesh made of 38M cells.

(c) Scalability test for the 3D mesh made of 150M cells.

Figure 8: Scalability tests for several types of meshes. They are run on ARCHER using up to 6,144 MPI tasks.



(a) Scalability test for the 3D mesh made of 38M cells.

(b) Scalability test for the 3D mesh made of 150M cells.

Figure 9: Scalability tests for several 3D meshes. They are run on a Blue Gene/Q using up to 32,768 MPI tasks.

4. Conclusions

In this work an aeroacoustic module based on the Ffowcs-Williams and Hawkings analogy has been implemented in *Code_Saturne*. The module has been tested against both

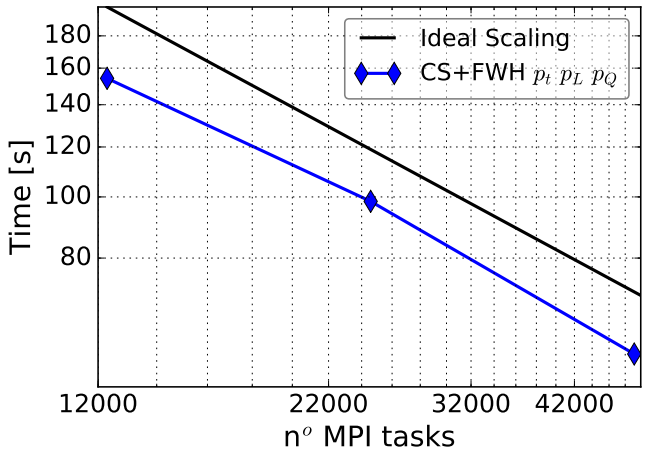


Figure 10: Scalability test for the 1.2 billion cell mesh on ARCHER.

analytical and numerical solutions, showing the correct implementation of the formulation that can be used with both solid and permeable surfaces. The scalability of the code has been accessed with meshes having up to 1.2 billion cells. The addition of the module introduces a marginal increase in CPU time, for the surface integrals only. A more substantial addition of computing time is observed when the quadrupole sources are also considered and this could amount for a 30% increase in CPU time. However, the slope of the scalability profiles is not affected by the addition of the FWH module, showing similar trends of reduction of the CPU time with the increase of the number of MPI tasks used. It has also been shown that the module is able to be run on a large number of MPI tasks, i.e. up to 44% of ARCHER, exhibiting an efficiency of 87% going from 24,576 to 49,152 MPI tasks.

5. Acknowledgements

This work was funded under the embedded CSE programme of the ARCHER UK National Supercomputing Service http://www.archer.ac.uk. The authors would also like to thank the Hartree Centre for testing the scalability of the new module on their Blue Gene/Qs.

References

[1] D. Casalino. *Analytical and numerical methods in vortex-body aeroacoustics*. PhD thesis, PhD thesis, Politcnico di Torino et Ecole Centrale de Lyon, 2002.

[2] F. Farassat and G. P. Succi. The prediction of helicopter rotor discrete frequency noise. In *In: American Helicopter Society, Annual Forum, 1982, DC, American Helicopter Society, p. 497-507.*, volume 1, pages 497–507, 1982.

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

[4] M. J. Lighthill. On sound generated aerodynamically. i. general theory. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 211(1107):564–587, 1952.

[5] A. Roshko. Experiments on the flow past a circular cylinder at very high reynolds number. *Journal of Fluid Mechanics*, 10(3):345–356, 1961.

A Full nomenclature used in the FWH formulation

The list below gives the definition of all the parameters introduced in the formulation of the FWH analogy

- ρ density, ρ_0 reference density, c speed of sound
- n_i normal to the surface, $r_i = x_i - y_i$ distance between source and receiver, $r = \|x_i - y_i\|$ module of the distance between source and receiver
- u_i fluid velocity, v_i source velocity, $M_i = \dfrac{v_i}{c}$ vector Mach number of a source
- $T_{ij} = \rho u_i u_j - \tau_{ij} + \left(p' - c^2\rho'\right)\delta_{ij}$ Lighthill's stress tensor, $L_{ii} = L_{11} + L_{22} + L_{33}$ trace of L_{ij}
- $L_i = P_{ij}\hat{n}_j + \rho u_i\,(u_n - v_n)$ Linear momentum
- $U_i = \left(1 - \dfrac{\rho}{\rho_o}\right)v_i + \dfrac{\rho u_i}{\rho_0}$
- $U_n = U_i\hat{n}_i, \quad \dot{U}_n = \dot{U}_i\hat{n}_i, \quad U_{\dot{n}} = U_i\hat{n}_i$
- $M_r = M_i\hat{r}_i, \;\; \dot{M}_n = \dot{M}_i\hat{n}_i$
- $L_r = L_i\hat{r}_i, \;\; \dot{L}_r = \dot{L}_i\hat{r}_i, \;\; L_M = L_iM_i$
- $T_{MM} = T_{ij}M_iM_j, \;\; T_{Mr} = T_{ij}M_i\hat{r}_j, \;\; T_{\dot{M}r} = T_{ij}\dot{M}_i\hat{r}_j, \;\; \dot{T}_{Mr} = \dot{T}_{ij}M_i\hat{r}_j, \;\; \dot{T}_{rr} = \dot{T}_{ij}\hat{r}_i\hat{r}_j, \;\; \ddot{T}_{rr} = \ddot{T}_{ij}\hat{r}_i\hat{r}_j,$
- p'_L loading noise
- p'_T thickness noise
- p'_Q quadrupole noise

B FWH module algorithms in *Code_Saturne*

In this section the algorithms for the implementation of the FWH module are presented. For the definition of the variables such as L_i, U_i, T_{ij} please refer to Appendix A.

Algorithm 1 FWH algorithm without distribution of the receivers across the MPI tasks

```
for 1 to total_number_of_time_steps do
  Resolution of the compressible Navier-Stokes equations
  if first_iteration then
    User definition of the number of receivers and their position
    Creation and memory allocation of the acoustic variables
  end if
  Computation of n̂_i, L_i and U_i T_ij and their relative time derivatives
  for i = 1 to total_number_of_receivers do
    Computation of r_i, r̂_i and r
    Projection (dot product) of several quantities (i.e. L_i) on r_i
    Computation of p'_L, p'_T and p'_Q
  end for
  if iteration_to_damp_the_acoustic_results then
    Creation of a buffer memory to be used to write the data
    Parallel summation of all the contributions
    Write of the buffer memory on disk
    Update of the FWH memory
  end if
  if last_iteration then
    Parallel summation of all the contributions
    Write of the full results on disk
    Deallocation of the FWH memory
  end if
end for
```

Algorithm 2 FWH algorithm with distribution of the receivers across the MPI tasks

for 1 to *total_number_of_time_steps* **do**
  Resolution of the compressible Navier-Stokes equations
  if *first_iteration* **then**
    User definition of the number of receivers and their position
    Distribution of the receivers across the MPI ranks
    Creation and local memory allocation of the acoustic variables depending on the local number of receivers
  end if
  Computation of $\hat{n}_i$, L_i and $U_i\ T_{ij}$ and their relative time derivatives
  for $i = 1$ to *local_number_of_receivers* **do**
    Retrieving in which MPI rank the i^{th} receiver is located
    Allocation of the support variables to compute the acoustic pressures
    Computation of r_i, $\hat{r}_i$ and r
    Projection (dot product) of several quantities (i.e. L_i) on r_i
    Computation of p'_L, p'_T and p'_Q
    Parallel summation of the contributions and broadcast of the results to the MPI ranks where the i^{th} receiver is allocated
    Deallocation of the support memory
  end for
  if *iteration_to_damp_the_acoustic_results* **then**
    Creation of a buffer memory to be used for data writing
    Write of the buffer memory on disk
    Update of the FWH memory
  end if
  if *last_iteration* **then**
    Write of the full results on disk
    Deallocation of the FWH memory
  end if
end for