





# A Further Development of the Asynchronous GPU CABARET Method for Jet-Noise Modelling

Igor A. Solntsev<sup>1</sup> , Sergey A. Karabasov<sup>1</sup> , Georgy A. Faranosov<sup>1</sup> ,  
Oleg P. Bychkov<sup>1</sup> 

© The Authors 2024. This paper is published with open access at SuperFri.org

A new asynchronous modification of the CABARET method is proposed for the solution of Navier–Stokes equations in the Large Eddy Simulation regime. The modification is based on improvement of the asynchronous extrapolation step both for Euler and Navier–Stokes CABARET solver. The algorithm is implemented as a parallel code for NVIDIA GPU using multiple CUDA-cores with MPI multi-CPU support. The algorithm accelerated on Graphics Processing Units (GPUs) is applied for the jet flow simulations in the Wall Model Large Eddy Simulation framework. The efficiency of code parallelization is discussed. The suggested asynchronous CABARET algorithm provides an almost 5000 times acceleration of calculations compared to a single CPU core, and allows us to calculate 300 convective times of jet development per day on a grid of 16 million cells. The flow solutions are analysed and compared with TsAGI anechoic chamber experimental data. It is shown that the structure of the jet flow is reproduced correctly, capturing low-amplitude instability waves in the jet potential core and fine-scale turbulent fluctuations in the near-field. Far-field noise predictions in the Ffowcs Williams–Hawking formulation with the azimuthal decomposition of the far-field radiation reproduce the nontrivial spectra and directivities of individual far-field acoustic modes.

*Keywords:* CABARET, asynchronous, GPU, parallel computing, jet noise.

## Introduction

Unsteady computational methods play an increasingly important role in jet noise modelling and design optimization studies of current propulsion system [2–4, 14, 17, 23, 24]. For simulations of noise generated by aerodynamic flows, high-resolution numerical methods are typically used. Specifically, these methods should not only have small numerical dissipation, but also accurately preserve dispersion properties of the governing gas dynamics equations. One example of such high-resolution methods is CABARET (Compact Accurately Boundary-Adjusting high-REsolution Technique) [10, 13] which was previously applied for modelling of high-speed jets [7, 11, 18, 21] and aerofoil flows [1, 19] using the Monotonically Integrated Large Eddy Simulation (MILES) framework for turbulence modelling. CABARET Navier–Stokes solver method was extended in previous work [27] to multi-zonal meshes including rotating meshes typical of propeller applications.

While the linear wave properties of the CABARET scheme are exceptionally good for Courant–Friedrichs–Lewy ( $CFL$ ) numbers around 0.5, for small  $CFL$  numbers its accuracy deteriorates [10]. However, small  $CFL$  numbers are unavoidable in some cases, e.g. during time marching at a constant global time step on non-uniform meshes. The asynchronous time-stepping technique with an optimal cell-local  $CFL$  to avoid accuracy deterioration was implemented for the CABARET Euler solver in [25] and extended to Navier–Stokes equations [20]. This technique is improved in the current work by modification of the asynchronous extrapolation step making it even more computationally efficient.

The article is organised as follows: In Section 1, the governing equations in the form of hyperbolic conservation laws with a heterogeneous right-hand-side are outlined. In Section 2,

<sup>1</sup>Central Aerohydrodynamic Institute, Zhukovskiy, Russia

the baseline CABARET algorithm for gas dynamics is briefly introduced. In Section 3, the new asynchronous time-stepping technique is presented. In Section 4, the efficiency of parallel asynchronous GPU code is discussed. Numerical results and comparison with TsAGI anechoic chamber experimental data are presented in Section 5.

## 1. Hyperbolic Form of Governing Equations

Following [27], the governing Navier–Stokes equations can be formulated using hyperbolic equations with an inhomogeneous right-hand side, which in Cartesian coordinates have the form:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z} = \mathbf{Q}, \quad (1)$$

where the conservative variables  $\mathbf{U}$  and the conservation fluxes  $\mathbf{F}$ ,  $\mathbf{G}$  and  $\mathbf{H}$  are

$$\begin{aligned} \mathbf{U} &= (\rho, \rho u, \rho v, \rho w, \rho e)^\top, \\ \mathbf{F} &= (\rho u, \rho u^2 + p, \rho uv, \rho uw, \rho ue + up)^\top, \\ \mathbf{G} &= (\rho v, \rho vu, \rho v^2 + p, \rho vw, \rho ve + vp)^\top, \\ \mathbf{H} &= (\rho w, \rho wu, \rho wv, \rho ww + p, \rho we + wp)^\top. \end{aligned}$$

Here  $u, v, w$  are the velocity components in the  $x$ -,  $y$ -, and  $z$ -direction in the relative frame,  $\rho, e, p$  are density, total energy, and pressure. The ideal gas equation of state is assumed, where pressure and density are related to internal energy via the specific heat ratio,  $\gamma$ .

The right-hand-side source in (1) includes contributions of viscous terms defined by components of the viscous stress tensor  $\sigma_{ij}$ , where  $i, j = x, y, z$

$$\mathbf{Q} = (0, \frac{\partial}{\partial x}\sigma_{xx} + \frac{\partial}{\partial y}\sigma_{xy} + \frac{\partial}{\partial z}\sigma_{xz}, \frac{\partial}{\partial x}\sigma_{yx} + \frac{\partial}{\partial y}\sigma_{yy} + \frac{\partial}{\partial z}\sigma_{yz}, \frac{\partial}{\partial x}\sigma_{zx} + \frac{\partial}{\partial y}\sigma_{zy} + \frac{\partial}{\partial z}\sigma_{zz}, \left[ \frac{\partial}{\partial x}(u\sigma_{xx} + v\sigma_{xy} + w\sigma_{xz}) + \frac{\partial}{\partial y}(u\sigma_{yx} + v\sigma_{yy} + w\sigma_{yz}) + \frac{\partial}{\partial z}(u\sigma_{zx} + v\sigma_{zy} + w\sigma_{zz}) \right]^\top). \quad (2)$$

The hyperbolic part of the governing equations (1) comprises 5 characteristics corresponding to the local Riemann invariants – one entropy wave, two pressure waves, and two contact waves in each spatial coordinate direction. The entropy and contact waves propagate at the characteristic speed  $\lambda_1 = uk_x + vk_y + wk_z$ , and the pressure waves propagate at  $\lambda_{2,3} = uk_x + vk_y + wk_z \pm a$ , where  $a = \sqrt{\gamma p / \rho}$  is the local sound speed and  $(k_x, k_y, k_z)^\top$  is a propagation direction.

## 2. 3-steps CABARET Algorithm

Following [7, 13], the Compact Accurately Boundary-Adjusting high-REsolution Technique (CABARET) algorithm for solving hyperbolic equations with an inhomogeneous right-hand-side (1) is presented as a three step procedure. Let us consider a hexagonal spatial grid, where each centre of the control volume (cell-centre point),  $c$ , is surrounded by three pairs of opposite grid faces (face-centre points),  $c-$  and  $c+$ , corresponding to a certain grid direction within the cell. By applying the Gauss–Ostrogradski theorem to integrate the governing system of conservation laws (1) in space and time, the conservative predictor and corrector steps of the CABARET algorithm are given by

$$\frac{\mathbf{U}_c^{n+\frac{1}{2}} - \mathbf{U}_c^n}{\tau^n/2} + \frac{(s_x \mathbf{F}_{c+}^n - s_x \mathbf{F}_{c-}^n) + (s_y \mathbf{G}_{c+}^n - s_y \mathbf{G}_{c-}^n) + (s_z \mathbf{H}_{c+}^n - s_z \mathbf{H}_{c-}^n)}{\Omega_c} = \mathbf{Q}(\mathbf{U}_c^n), \quad (3)$$

$$\frac{\mathbf{U}_c^{n+1} - \mathbf{U}_c^{n+\frac{1}{2}}}{\tau^n/2} + \frac{(s_x \mathbf{F}_{c+}^{n+1} - s_x \mathbf{F}_{c-}^{n+1}) + (s_y \mathbf{G}_{c+}^{n+1} - s_y \mathbf{G}_{c-}^{n+1}) + (s_z \mathbf{H}_{c+}^{n+1} - s_z \mathbf{H}_{c-}^{n+1})}{\Omega_c} = \mathbf{Q}(\mathbf{U}_c^{n+\frac{1}{2}}), \quad (4)$$

where  $\mathbf{U}_c$  is the vector of conservation variables referred to the cell centre,  $\mathbf{F}_{c+}, \mathbf{F}_{c-}, \mathbf{G}_{c+}, \mathbf{G}_{c-}$  and  $\mathbf{H}_{c+}, \mathbf{H}_{c-}$  denote the pairs of conservation fluxes in the  $x$ -,  $y$ - and  $z$ - directions computed at the face centres,  $\Omega_c$  and  $s_x, s_y, s_z$  are the cell volume and the face normal areas respectively,  $\tau^n$  is a time step, and the sub-index and super-index  $n, n + 1/2, n + 1$  denote the space and time discretisation. For both the predictor and corrector steps, the right-hand-side terms  $\mathbf{Q}$  are computed by substituting the most recently updated conservation variable in the source equation (2).

It can be noted that the corrector step (4) involves the unknown flux functions from the new time step,  $n + 1$ . These functions are updated at the intermediate characteristic decomposition and extrapolation step, which follows right after the predictor step (3). In the characteristic decomposition and extrapolation step, we determine flux variables in the centers of faces on a new layer  $n + 1$  so that the characteristic invariants are computed at the center of each cell  $\mathbf{R}_c^{n+1/2}$  and at the centers of the faces

$$\mathbf{R} = \begin{pmatrix} r \\ q \end{pmatrix}, \quad q = \varpi - Gp^\mu, \quad r = \varpi + Gp^\mu, \quad s = \ln(p/\rho^\gamma),$$

where  $\mu = \frac{\gamma-1}{2\gamma}$ ,  $G = \frac{2\gamma}{\gamma-1} \exp(\frac{s}{2\gamma})$ , and  $\varpi$  corresponds to the speed  $u, v$  or  $w$  for each direction  $x, y$  and  $z$ .

The computation of the local Riemann invariants at the current and mid-time level,  $n, n + 1/2$ , is followed by extrapolation of the cell-face variables to the new time step  $n + 1$  in each grid cell,  $c$ ,

$$\tilde{\mathbf{R}}_{c+}^{n+1} = 2\mathbf{R}_c^{n+1/2} - \mathbf{R}_{c-}^n \quad \text{and} \quad \tilde{\mathbf{R}}_{c-}^{n+1} = 2\mathbf{R}_c^{n+1/2} - \mathbf{R}_{c+}^n.$$

For each face that is an interface of two adjacent cells  $L$  and  $R$  the extrapolated values from each side of the contact  $\tilde{\mathbf{R}}_{L+}^{n+1}, \tilde{\mathbf{R}}_{R-}^{n+1}$  are adjusted using the maximum principle. Two sets of corrected invariants on the cell contact surface are then used and the signs of the characteristic wave speeds are accounted for to select a unique set of invariants for each cell face. Having defined the complete set of characteristic variables, the  $(5 \times 5)$  linear system solved to obtain a solution of the Riemann problem, and the values of the variables and fluxes  $\mathbf{F}_{c+}^{n+1}, \mathbf{F}_{c-}^{n+1}, \mathbf{G}_{c+}^{n+1}, \mathbf{G}_{c-}^{n+1}, \mathbf{H}_{c+}^{n+1}, \mathbf{H}_{c-}^{n+1}$  are computed at the new time level  $n + 1$  for each cell face  $c+, c-$ . The results are then substituted in the conservation flux functions for the conservative corrector step (4).

### 3. Asynchronous Time-stepping

Asynchronous time stepping, i.e., when the solution in cells of different sizes is updated at different rates and adjusted to the cell-local  $CFL$  number rather than to a global one, is an effective way of improving the efficiency of explicit methods with highly non-uniform grids without any loss of accurate linear waves propagation properties. In particular, the asynchronous time-stepping technique was implemented for the CABARET Euler [25] and the Navier–Stokes solver [20] keeping the following important properties: (i) simplicity and compactness of the original computational stencil, (ii) strict conservation property and (iii) a built-in recipe for the

treatment of inactive flow regions. This technique is improved in the current work by modification of the asynchronous extrapolation step both for the Euler and for the Navier–Stokes solver.

To illustrate the main concepts, let us consider a one-dimensional flow on a non-uniform grid along the  $X$  axis with a variable cell size  $h_c$  at time  $t^n$ . We will assume that the upper and lower indices correspond to evolving space and time. Let us determine the value of the optimal time step in each of the cells from the relation  $\tau_c^n = CFL \cdot h_c / (|u_c^n| + a)$ , here  $u$  is the local flow speed, and  $a$  is the local speed of sound, assuming  $CFL = 0.5$ . The next active time step of the asynchronous algorithm,  $\tau^n$  is determined by the step from the current moment  $t^n$  to the next state  $t^{n+1}$ , determined by the minimum value for all grid cells.

To increase computational efficiency, the optimal local time step in each of the cells are replaced by closest values from a pre-defined set of acceptable values with a certain minimum step. Following [20, 25], the update is not performed for all cells, but only for cells with a minimum next temporal state  $t^{n+1}$ . We will denote such cells as active cells. The calculation of conservative variables at asynchronous predictor and corrector steps is performed according to the standard algorithm, but for different moments in time  $t_c^n + \bar{\tau}_c^n/2$  and  $t_c^n + \bar{\tau}_c^n$ , determined by the local step in each of the cells. Equations (3),(4) for 1D take the form

$$\frac{\mathbf{U}_c^{n+1/2} - \mathbf{U}_c^n}{\bar{\tau}_c^n/2} + \frac{(\mathbf{F}_{c+}^n - \mathbf{F}_{c-}^n)}{h_c} = \mathbf{Q}(\mathbf{U}_c^n),$$

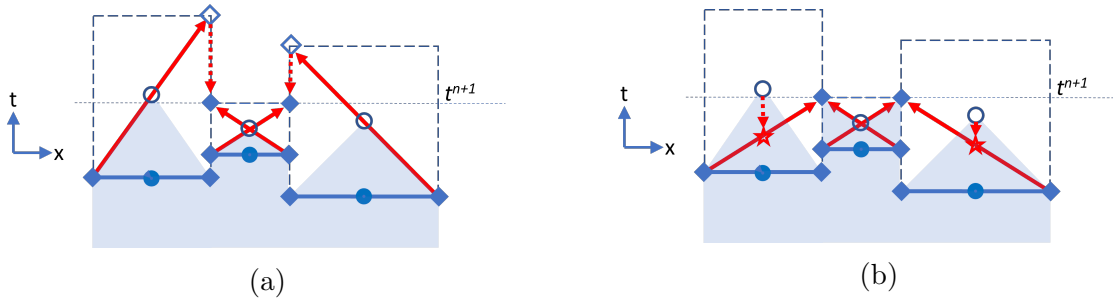
$$\frac{\mathbf{U}_c^{n+1} - \mathbf{U}_c^{n+1/2}}{\bar{\tau}_c^n/2} + \frac{(\mathbf{F}_{c+}^{n+1} - \mathbf{F}_{c-}^{n+1})}{h_c} = \mathbf{Q}(\mathbf{U}_c^{n+1/2}).$$

The main novelty of the suggested algorithm in the current work is due to the change made at the characteristic extrapolation step. Here, to calculate the fluxes through faces of an active cell, it is necessary to perform extrapolation inside the cell while obtaining external values extrapolated to faces from neighboring cells. Extrapolation within a cell is performed in a standard manner. However, the standard extrapolation in neighboring cells with large time steps leads to extrapolated values not for the required time point  $t^{n+1}$ , but for some future time points.

To obtain the required values, [20, 25] use inverse interpolation of extrapolated values (Fig. 1a). Instead, here we use a characteristic extrapolation algorithm based on the modified values of conservative variables in neighboring cells. Rather than using the intermediate values obtained in neighboring cells at the predictor step, the values obtained by their linear interpolation in time to the required point in time  $\tilde{t}_c^{n+1}$  are used, ensuring the construction of the correct extrapolation stencil (Fig. 1b). For example, for extrapolation to the edge of an active cell  $c$ , this moment in time for the neighboring cell  $cN$  is defined as  $\tilde{t}_{cN}^{n+1/2} = (t_c^{n+1} - t_{cN}^n)/2$ , and the intermediate values of conservative variables in the neighboring cell are interpolated to the desired time point as

$$\tilde{\mathbf{U}}_{cN}^{n+1/2} = \mathbf{U}_{cN}^n + (\mathbf{U}_{cN}^{n+1/2} - \mathbf{U}_{cN}^n)(\tilde{t}_{cN}^{n+1/2} - t_{cN}^n)/\bar{\tau}_{cN}^n.$$

This modification allows us to avoid the additional calculation and storage of large arrays of flow invariants at the centers of cells and the centers of faces, which increases the computational efficiency of the final algorithm. The extrapolation step is now performed in asynchronous mode according to the basic algorithm described in Section 2, replacing intermediate values of conservative variables in neighboring cells  $\mathbf{U}_{cN}^{n+1/2}$  with their interpolated values  $\tilde{\mathbf{U}}_{cN}^{n+1/2}$ .



**Figure 1.** The baseline characteristic extrapolation (a) and extrapolation with the new suggested pre-interpolated conservative variables at mid-time levels (b)

In Fig. 1, solid horizontal lines correspond to different points in time, depending on the update history of each cell. The dashed horizontal lines indicate the optimal next time steps for each of the cells. Arrows indicate extrapolation (solid arrows) and interpolation (dotted arrows), stars indicate interpolated values of the new algorithm.

The discrepancy between the time steps of contacting cells leads to the fact that one large time step in a certain cell may correspond to several small steps in a neighboring cell. To keep fluxes conservative in an asynchronous scheme, we apply fluxes synchronization proposed in the previous works [20, 25]: fluxes in a cell with a large step are adjusted according to the total values in cells with a smaller step. Algorithm 1 depicts a pseudocode of the resulting asynchronous algorithm.

## 4. Parallelization of Asynchronous Code

The asynchronous CABARET algorithm is implemented as a parallel code using the CUDA parallel computing architecture for NVIDIA GPU devices and the MPI message passing interface between parallel CPU host processes. Using these tools allows you to use two different parallelization approaches to speed up calculations on grids of several million cells and more.

To speed up calculations of the CABARET algorithm on multi-core CPUs, the MPI interface allows parallel calculations for individual parts of the decomposed mesh. In this case, the grid cells in each decomposition zone are processed by the corresponding CPU core sequentially one after another. The maximum ideal value for speeding up calculations is determined by the number of CPU cores. However, the need to exchange data between cells at the boundaries of decomposition zones by calling MPI-synchronization routines reduces the efficiency of MPI-parallelization.

The CUDA parallel computing architecture and modern NVIDIA GPU graphics accelerators allow a more efficient approach for grids with a moderate number of grid cells. If the mesh fits completely into the memory of the graphics accelerator, then mesh decomposition is not required. Grid cells are processed in parallel in groups which size is determined by the number of CUDA cores. Modern GPU card such as A100 and H100 have about 7000 and 15000 CUDA cores respectively and 80GB onboard memory. The current code implementation with modified asynchronous step requires 1GB of memory for 1 million grid cells in single precision mode. This allows maximum computational efficiency of the code for grids of up to 80 million cells. Combination of CUDA and MPI approaches will increase this limit for multi-GPU computers and clusters.

To assess the actual efficiency of MPI code parallelization for a CPU, a series of calculations of the jet flow from a test nozzle was performed on grids with different numbers of nodes, 2 million

---

**Algorithm 1** ASYNCHRONOUS TIME-STEPPING

---

```

// Initialize the initial states at  $t = t^0$  for conservative variables in the centers of cells  $\mathbf{U}_c^0$ 
// and for primitive variables and fluxes  $\mathbf{F}_{c-}^0, \mathbf{F}_{c+}^0$ , initialize the initial values of the list of
// active/updated cells  $Active_c^n = .false$ .
1: Set  $n = 0, t = t^0$ 
2: For all cells: initialize  $\mathbf{U}_c^0$ 
3: For all faces: initialize  $\mathbf{F}_{c-}^0, \mathbf{F}_{c+}^0$ 
4: For all cells: initialize  $Active_c^n = .false$ .
// Initialize the optimal and quantized values of time steps  $\tau_c^n$  and  $\bar{\tau}_c^n$ 
5: For all cells: initialize  $\tau_c^n$ , initialize  $\bar{\tau}_c^n$ 
// Initialize the initial values of conservative variables on the local intermediate time layer
 $\mathbf{U}_c^{0+1/2}$ 
6: For all cells: initialize  $\mathbf{U}_c^{0+1/2}$ 
// Start of cycle by time
7: do
// Determine the next active step
8: Define  $\bar{\tau}^n$ 
// Predictor for active cells – define conservative variables in the intermediate time
// layer
9: For all Active cells: define  $\mathbf{U}_c^{n+1/2}$ 
// Updating the list of active cells
10: For all cells with  $\bar{\tau}_c^n = \bar{\tau}^n$  define  $Active_c^n = .true$ .
// Extrapolation for active cells: interpolation of conservative variables on the interme-
// diate layer in neighboring cells  $\tilde{\mathbf{U}}_c^{n+1/2}$  and calculation of fluxes on the faces and with
// asynchronous correction  $\mathbf{F}_{c-}^{n+1}$  and  $\mathbf{F}_{c+}^{n+1}$ 
11: For all Active cells: interpolate neighbour cells  $\tilde{\mathbf{U}}_c^{n+1/2}$ 
12: For all Active cells: define face fluxes with asynchronous correction
 $\mathbf{F}_{c-}^{n+1}$  and  $\mathbf{F}_{c+}^{n+1}$ 
// Corrector for active cells – define conservative variables on a new layer  $\mathbf{U}_c^{n+1}$ 
13: For all Active cells: define  $\mathbf{U}_c^{n+1}$ 
// Check the current time
14: If  $t^n \geq t_{END}$  then exit
15: Set  $t^{n+1} = t^n + \bar{\tau}^n$ , set  $n = n + 1$ 
// Determining new optimal and quantized values of time steps  $\tau_c^n$  and  $\bar{\tau}_c^n$ 
16: For all Active cells: define  $\tau_c^n$ , define  $\bar{\tau}_c^n$ 
17: end do

```

---

and 16 million, using from 1 to 64 cores of an AMD EPYC 7742 CPU. Figure 2 shows MPI efficiency parameters for the solver with the suggested modification of the asynchronous step:  $Speedup = T_1/T_{Ncpu}$ , which correspond to the acceleration compared to calculations on a single CPU core. When using the maximum number of cores, the efficiency of parallelization decreases due to an increase in the share of losses for synchronizing the cores with each other. However, when the grid size increases (weak stability test) the developed code provides good scalability with increasing the number of MPI processes. In this case, the acceleration of calculations with increasing grid sizes approaches maximum theoretical values.

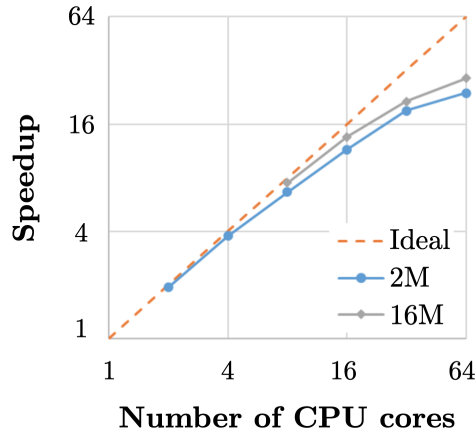


Figure 2. MPI CPU parallelization speedup for 2M and 16M grids

The efficiency of the modified asynchronous algorithm and its parallelization using graphics accelerators was assessed in test calculations of the jet flow from a test nozzle on a non-uniform grid of 16 million cells. For reliable modeling of noise generation by an outflowing jet, the mesh was significantly refined in the area of the nozzle edge (see Section 5). The ratio of the linear size of the maximum cell to the size of the minimum cell was 20000. The use of the asynchronous step ensured acceleration of asynchronous calculations by 8–12 times compared to the global time step, both when using the CPU and when calculating on the GPU. Notably, the modified asynchronous step reduces the required memory by 25% and increases performance by approximately 1.2 times. In turn, parallelization using the CUDA architecture on the NVIDIA A100 graphics card provides a 600 times acceleration of calculations compared to calculations on a single CPU core. In total, this provides almost a 5000 times acceleration of calculations compared to a single CPU core, allowing to calculate 300 convective times of jet development per day on a grid of 16 million cells on a single GPU card.

Figure 3 compares calculation times for 500 convective times using CPU and GPU, and the achieved acceleration of the modified asynchronous mode compared to a single CPU in a global mode.

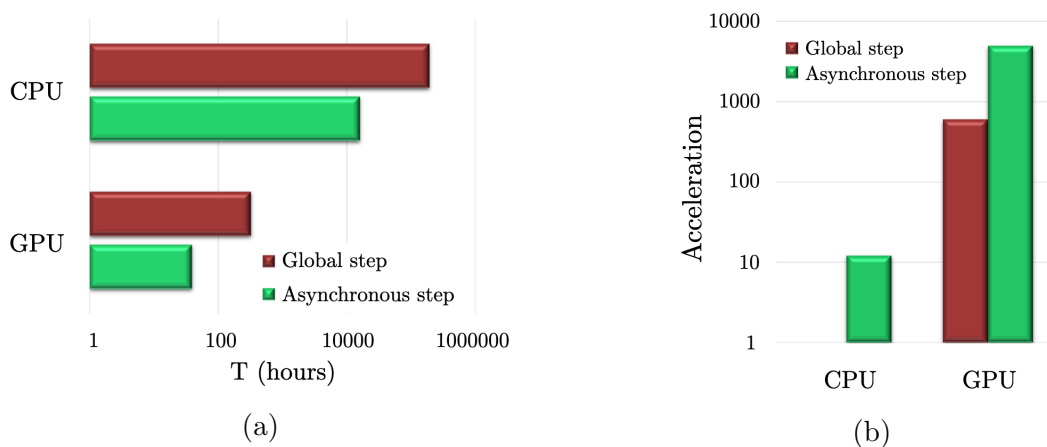
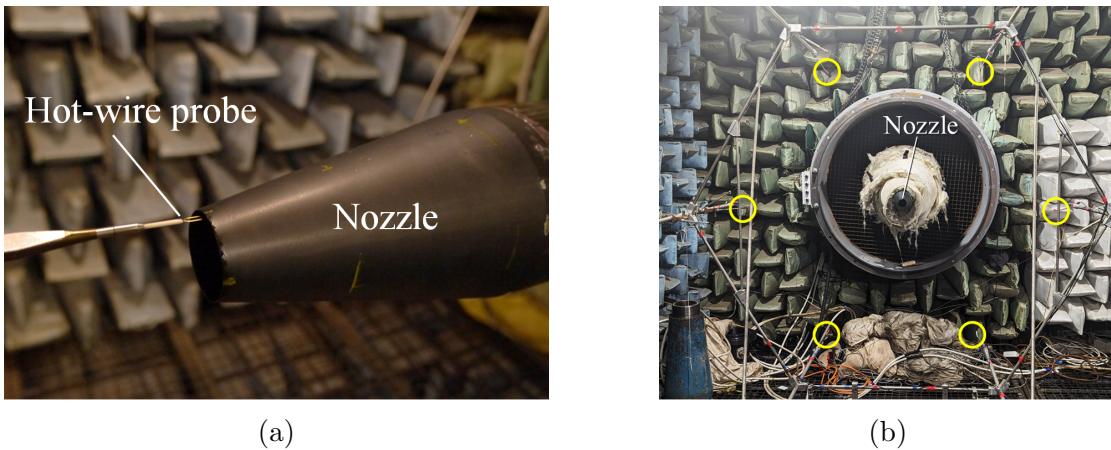


Figure 3. Efficiency of the modified asynchronous algorithm on a heterogeneous grid of 16 million cells for a CPU (AMD EPYC 7742, 1 core) and GPU (NVIDIA A100): calculation time (a) and calculation acceleration (b) on a logarithmic scale

## 5. Numerical Results and Validation

To verify the accuracy of the developed CABARET method, a single-stream axisymmetric static jet was simulated and its results were compared with experimental data. Experiments were carried out in the TsAGI anechoic chamber AC-2 designed for acoustic and aerodynamic measurements of isolated subsonic and supersonic jets. A round profiled nozzle of diameter  $D = 40$  mm was used. An unheated jet with acoustic jet Mach number  $M_j = 0.7$  was considered (jet exit velocity  $U_j \approx 240$  m/s). The two types of experimental data were gathered. First, jet flow parameters were measured by a hot wire. Second, far-field acoustic pressure was measured by an azimuthal microphone array. The detailed description of the corresponding measurement procedures in AC-2 can be found in [5, 6, 16].

For the jet velocity measurements, Dantec probe 55R01 was used. The sensor of the probe had a fiber of diameter  $70 \mu\text{m}$  and length of  $1.25$  mm. The probe was mounted on a Dantec 3D hot-wire traverse mechanism (model 41T33) so that the sensor was perpendicular to the mean flow (Fig. 4a). The traverse system was used to obtain the distribution of the mean velocity and velocity fluctuation spectra along the jet axis and along the lipline. The measurements on each of the lines were conducted in 31 points uniformly distributed along the jet axis from  $x/D = 0$  to  $x/D = 15$  with  $0.5D$  increment.

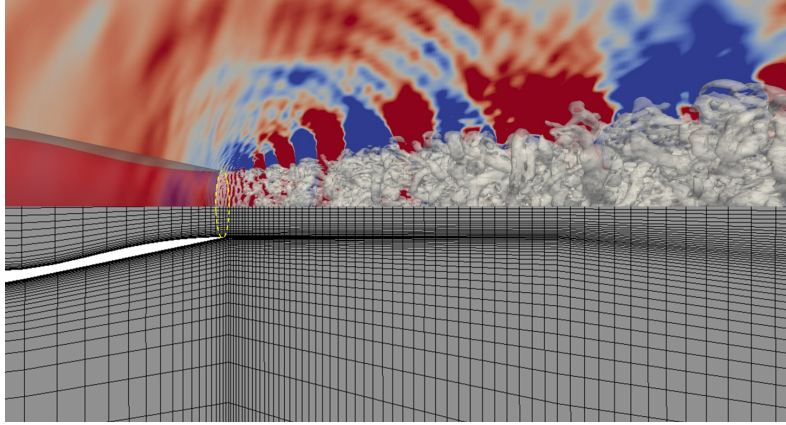


**Figure 4.** Experimental setup: (a) hot-wire measurements; (b) far-field noise measurements by the azimuthal array (microphones are highlighted by the circles)

Far-field measurements were carried out by the standard microphone array of radius  $R = 0.8$  m typically used in AC-2 for jet noise measurements. The array is equipped by six  $\frac{1}{2}$  Bruel&Kjaer microphones (type 4189) with B&K preamplifiers (type 2669, frequency range 40–25600 Hz, sensitivity 50 mV/Pa). The microphones are uniformly distributed over the azimuthal angle  $\theta$  around the jet axis [6, 16] (Fig. 4b). The microphone array can be moved along its axis in the range  $-31.25 \leq x/D \leq 62.5$ , the origin  $x = 0$  corresponding to the nozzle exit plane. The array is capable of measuring the first three azimuthal modes of jet noise at each jet cross-section  $x$  and giving an assessment of the fourth cosine mode. Thus, if acoustic pressure fluctuations  $p(x, R, \theta, t)$  can be represented with acceptable accuracy by a superposition of the first three azimuthal modes

$$p(x, R, \theta, t) \approx A_0(x, R, t) + \sum_{n=1}^2 (A_n(x, R, t) \cos n\theta + B_n(x, R, t) \sin n\theta), \quad (5)$$



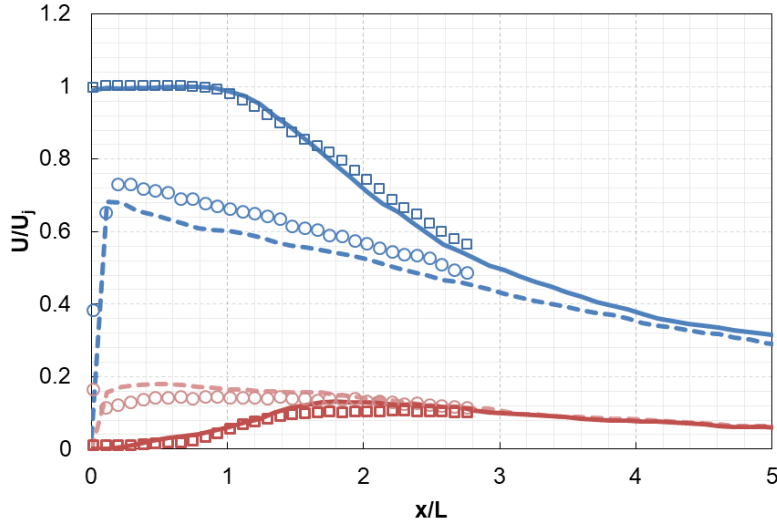


**Figure 5.** Mesh structure (every 4<sup>th</sup> node is shown) and instantaneous pressure field with Q-criteria iso-surfaces ( $Q = 25000$ ) in the jet symmetry plane

then the amplitudes  $A_n(x, R, t)$  and  $B_n(x, R, t)$  can be measured by the array. It is well-known [6, 12, 16] that for a subsonic jet issuing from a round nozzle  $\overline{A_n^2} = \overline{B_n^2}$  ( $n \geq 1$ ), where the overbars stand for the time averaging, due to the axial symmetry of the statistical characteristics of the jet noise. Moreover, each azimuthal mode  $n$  has specific directivity, and higher order modes  $n \geq 3$  are negligibly small for low and moderate frequencies. Thus, the azimuthal content of jet noise serves as a good basis for validation of numerical simulation results [8, 15].

For the numerical simulation with turbulence modelling, the Wall Modelled Large Eddy Simulation framework was utilised, where the wall shear stress is computed from an iterative procedure using an algebraic wall function of the velocity profile and the streamwise velocity solution computed in the first-off-the-wall cell face [18]. For the upstream nozzle inlet boundary, the standard total pressure and total temperature boundary conditions are imposed. For the open lateral and outlet boundaries, non-reflecting characteristic boundary conditions are used. A cylindrical-conical computational solution domain is considered, which is aligned with the evolving jet flow downstream of the nozzle exit. The domain size in the streamwise flow direction is  $227D$ , where  $32.5D$  corresponds to the cylindrical domain part upstream of the nozzle exit. The radius of the computational domain upstream of the nozzle exit is  $50D$ , linearly expanding to  $125D$  in the outlet section.

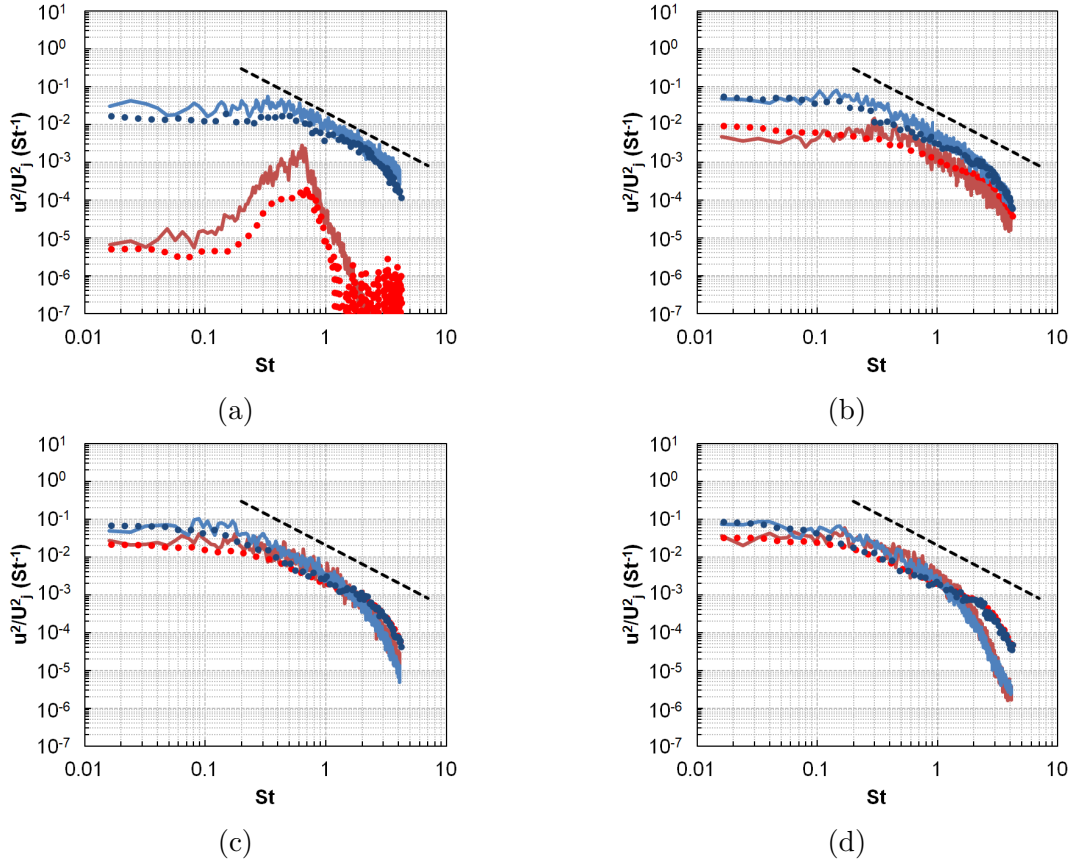
A zoom in to the mesh structure (every 4<sup>th</sup> node) and a snapshot of the instantaneous pressure field with Q-criteria iso-surfaces ( $Q = 25000$ ) in the jet symmetry plane in the near nozzle region are shown in Fig. 5. Near the nozzle exit (at  $x/D \sim 0.5$ ), somewhat delayed numerical transition to turbulent shear layer can be observed: quasi-regular vortical structures transform to stochastic ones. This is typical effect for LES solutions [7, 26], which correspond to the initially laminar inflow condition, like in the present case. In the future, it can be improved by further refinement of the initial shear region as well as using a synthetic turbulent inflow condition upstream of the nozzle exit. Pressure fluctuation pattern reveals near-field wave-packet-like structures, acoustic waves generated by the shear layer and spurious high-frequency noise related to the mesh stretching influence on the CABARET solution. Spurious noise can be reduced by utilizing a better quality Cartesian mesh in the shear region based on octree-type grid generation strategy [25]. Nevertheless, since the main goal of the current paper is to demonstrate adequate operation of the developed parallel code, mesh sensitivity study analogous to [25] was not performed.



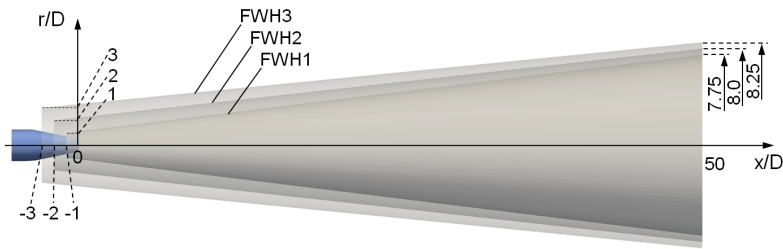
**Figure 6.** Mean (blue) and RMS (red) of the axial velocity fluctuations at the jet axis and at the lipline. Symbols – experiment ( $\square$  – axis,  $\circ$  – lipline), lines – simulation (solid – axis, dashed – lipline)

The main characteristics demonstrating general quality of physical modeling are related to the velocity field in the jet plume. Mean and RMS velocity distributions give general information on the jet development, while power density spectra of the velocity fluctuations indicate what turbulent scales are resolved in the simulation. To compare the results of the simulation with the experimental data, axial velocity component time histories were stored at the jet axis and the lipline during  $\sim 1500$  convective time units  $D/U_j$ . The results of the comparison are shown in Fig. 6. Velocity is scaled by the jet exit velocity  $U_j$  and axial coordinate is scaled by the length of the jet potential core  $L$ . Absolute values of  $L$  was  $4.5D$  and  $5.5D$  for the simulation and experiment, respectively, so that the simulated jet is a bit shorter compared to the physical one, which is typical for relatively coarse meshes [7, 25]. In dimensionless coordinates  $(x/L, U/U_j)$ , the results of the new parallel code are in good agreement with the experimental data.

A more subtle characteristic of the turbulent jet flow is its velocity spectra. It is known that, for a well developed mixing layer, high-frequency spectral tail should roll-off in accordance with the  $-5/3$  slope of the Kolmogorov law for isotropic turbulence. Typical velocity spectra for several distances from the nozzle exit are shown in Fig. 7. Spectral density is scaled by the jet velocity  $U_j$ , and Strouhal number  $St = fD/U_j$  represents dimensionless frequency. One can see that, for the points located in the turbulent region (all points at the jet lipline and axial points beyond the potential core of the jet), simulated and measured spectra are in good agreement with a slight overprediction in the numerical solution. A distinct  $-5/3$  slope region can be observed with the highest resolved frequencies up to  $St = 1...3$  depending on the point location. The resolution of the fine-scale fluctuations degrades for larger distances from the nozzle exit, because of the grid coarsening in the downstream direction. It is interesting to note that for a point located on the jet axis inside the potential core (Fig. 7a), the solution captures rather weak fluctuations related to the instability waves [5] (spectral hump around  $St = 0.1...1$ ). Visible overestimation of the numerical solution in this point is partially related to the fact that the measured signal was masked by the self-noise of the probe and thus the resultant levels of the denoised experimental spectrum may be not precise enough.



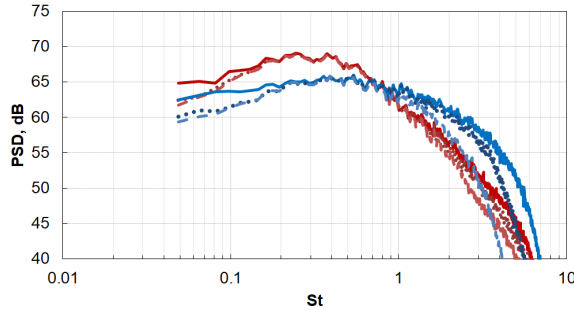
**Figure 7.** Centerline (red) and lipline (blue) velocity spectra. Symbols – experiment, lines – simulation. (a)  $x/L = 0.5$ , (b)  $x/L = 1.1$ , (c)  $x/L = 1.6$ , (d)  $x/L = 2.0$ . Black dashed lines correspond to the  $-5/3$  slope



**Figure 8.** FWH surfaces for far-field noise calculation

For the far-field noise calculation, the Ffowcs Williams–Hawkings (FWH) integral method [9] in the permeable surface formulation given in [22] was used. It is known that closed FWH-surfaces with multiple closing discs [7, 26] may be preferable to improve low-frequency noise components simulation. However, for the purpose of the current study, we used a simplified approach with open control FWH-surfaces of an increased length to capture the main portion of jet noise without the disc-averaging technique. This allowed us to obtain adequate spectra in the frequency range of interest. Three conical FWH-surfaces of different radii were considered (Fig. 8).

The far-field observer points were distributed in axial ( $-62.5 \leq x/D \leq 62.5$ ) and azimuthal directions over the cylindrical surface, of radius  $R = 20D$ , surrounding the jet. This allowed us



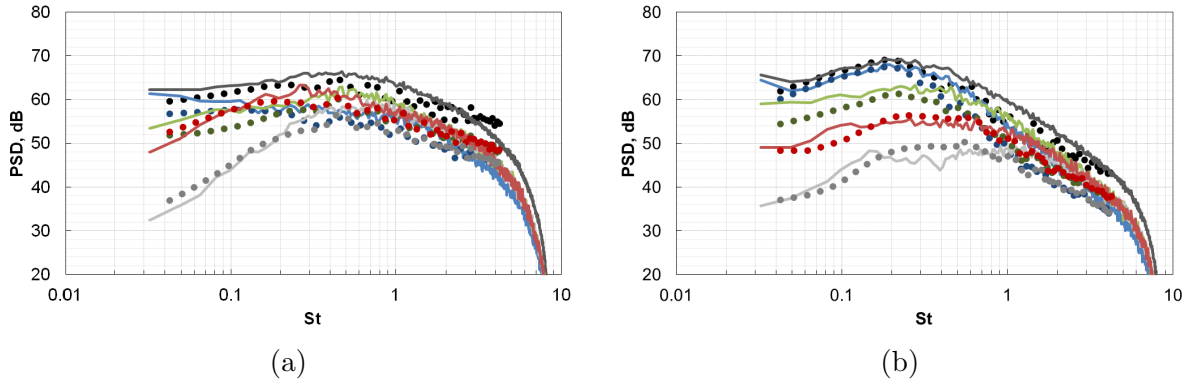
**Figure 9.** Sensitivity of the far-field spectra to the FWH surfaces.

Red – observation angle  $32^\circ$  ( $x/D \approx 38$ ), blue – observation angle  $90^\circ$  ( $x/D \approx 6$ ).

Solid lines – FWH1, dotted lines – FWH2, dashed lines – FWH3

to compare the simulation results with the experimental data obtained by the azimuthal array (Fig. 4b). Sensitivity of the results to the location of FWH-surface is shown in Fig. 9 for the sideline direction and for the direction of most intense radiation (observation angle is counted from the end of the potential core). It is seen that using FWH-surface of larger radius leads to reducing the resolution at high frequencies, but the spectral peak is reproduced identically by all FWH-surfaces. Further results are given for the acoustic control surface denoted as FWH1.

Figures 10 and 11 demonstrate the comparison of the simulation results with the experimental data, both in terms of the total noise and its azimuthal components. From Fig. 10, one can see that the simulation adequately reproduces the total jet noise spectra and the subtle structure of its modal content. There is some overprediction of absolute levels in the sideline direction, which can be improved by increasing the grid resolution of the initial shear layers, as mentioned previously.



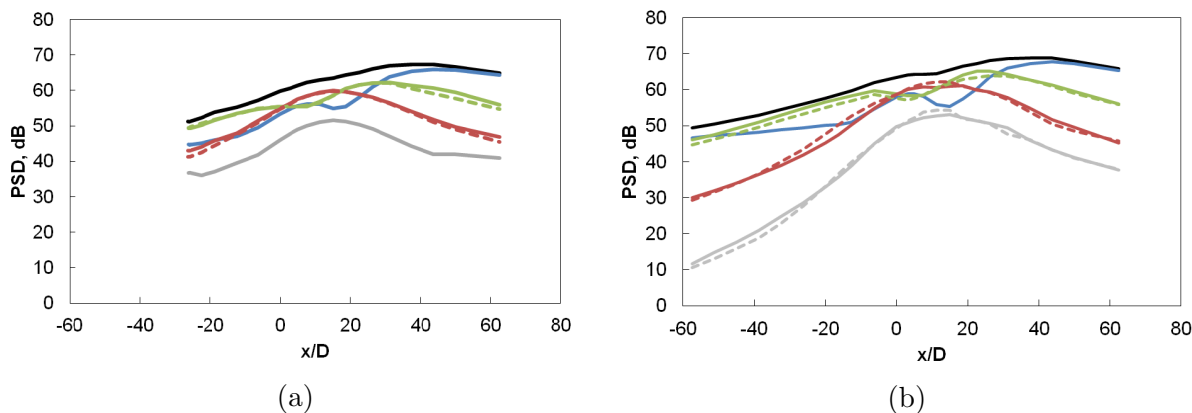
**Figure 10.** Far-field jet noise total and modal spectra. (a) sideline direction ( $x/D \approx 6$ ),

(b) direction of maximum radiation ( $x/D \approx 38$ ). Symbols – experiment, lines – simulation.

Black – total noise, blue – axisymmetric mode  $n = 0$ , green – mode  $n = 1$ , red – mode  $n = 2$ , grey – mode  $n = 3$

Figure 11 presents the distribution of the total jet noise and its azimuthal modes on the cylindrical surface surrounding the jet for the spectral peak  $St = 0.2$ . Again, accurate simulation of the main features of jet noise is observed: the total directivity, the directivities of individual harmonics, the equivalence of the corresponding sine and cosine modes, and the domination of low-order modes  $n = 0, 1, 2$  are all correctly captured. Thus, we have demonstrated acceptable

accuracy of the developed parallel solver on the acoustically sensitive calculations. Even on the rather coarse mesh, encouraging simulation quality, capturing the main physical features of real turbulent jet, was obtained, both in terms of turbulent properties of the jet flow and its acoustic radiation, including rather subtle characteristics of individual azimuthal modes.



**Figure 11.** Far-field distribution of jet noise and its azimuthal modes on the cylindrical surface surrounding the jet for the peak frequency  $St = 0.2$ . (a) experiment, (b) simulation. Black – total noise, blue – axisymmetric mode  $n = 0$ , green – mode  $n = 1$ , red – mode  $n = 2$ , grey – mode  $n = 3$ . Solid lines – cosine modes, dashed lines – sine modes

## Conclusion

A new asynchronous modification of the CABARET method is proposed for the solution of Navier–Stokes equations in the Large Eddy Simulation regime. A new asynchronous characteristic-extrapolation step is implemented increasing computational efficiency of the previously suggested asynchronous algorithm. It is shown that GPU parallelization with an asynchronous time step is an extremely effective tool for increasing the computational performance of the Navier–Stokes CABARET solver for performing acoustically sensitive calculations. The computational efficiency and encouraging accuracy of the developed parallel solver was demonstrated on the acoustics sensitive simulation of a single-stream Mach 0.7 round turbulent jet for a moderate grid resolution. Modified asynchronous CABARET algorithm provides an almost 5000 times acceleration of calculations compared to a single CPU core, thereby enabling the calculation of 300 convective times of jet development on a grid of 16 million cells per day. Comparison with the TsAGI experimental data is performed the mean velocity, turbulence intensity, and turbulent velocity spectra. It is shown that the structure of the jet flow is reproduced correctly, including reasonable resolution of fine-scale turbulent fluctuations in the inertial subrange of the energy spectra and capturing low-amplitude instability waves footprint in the jet potential core. Far-field noise predictions are performed using the permeable surface Ffowcs Williams–Hawking formulation. The conventional aeroacoustic postprocessing is supplemented by the azimuthal decomposition of the far-field noise. It is shown that the method correctly reproduces the nontrivial spectra and directivities of individual far-field acoustic modes. Thus, the developed parallel GPU CABARET solver is demonstrated to correctly capture the main physical features of a natural turbulent jet even on rather coarse meshes. Future work will include increasing the jet LES grid resolution utilising the significant acceleration offered by the developed in-house GPU CABARET solver.

## Acknowledgements

The part of the work related to the development of the numerical method was supported by Russian Science Foundation (grant No. 21–71–30016). Experimental part of the work was supported by Russian Science Foundation (grant No. 19–71–10064).

*This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.*

## References

1. Abid, H.A., Markesteijn, A.P., Karabasov, S.A.: Trailing Edge Noise Modelling of Flow over NACA Airfoils Informed by LES. AIAA 2021-2233 (2021). <https://doi.org/10.2514/6.2021-2233>
2. Advisory Council for Aviation Research and Innovation in Europe (ACARE): Realising Europe's vision for aviation, in: Strategic research and innovation agenda, Vol. 1, Advisory Council for Aviation Research and Innovation in Europe, 2012; Strategic research and innovation agenda: the proposed European partnership for clean aviation., <http://www.clean-aviation.eu/>, July 2020
3. Block, P.J.W., Gentry, G.L.J.: Directivity and trends of noise generated by a propeller in a wake. NASA Technical Paper 2609 (1986)
4. Brelje, B.J., Martins, J.R.R.A.: Electric, hybrid, and turboelectric fixed-wing aircraft: a review of concepts, models, and design approaches. Prog. Aerosp. Sci. 104, 1–19 (2019), <https://doi.org/10.1016/j.paerosci.2018.06.004>
5. Bychkov, O., Faranosov, G., Kopiev, V., *et al.*: Jet installation noise modeling in static and flight conditions using centerline fluctuations. AIAA Journal 60(6), 3620–3634 (2022). <https://doi.org/10.2514/1.J060879>
6. Faranosov, G., Belyaev, I., Kopiev, V., Bychkov, O.: Azimuthal structure of low-frequency noise of installed jet. AIAA Journal 57(5), 1885–1898 (2019). <https://doi.org/10.2514/1.J057476>
7. Faranosov, G.A., Goloviznin, V.M., Karabasov, S.A., *et al.*: CABARET method on unstructured hexahedral grids for jet noise computation. Computers and Fluids 88, 165–179 (2013). <https://doi.org/10.1016/j.compfluid.2013.08.011>
8. Faranosov, G.A., Kopiev, V.F., Karabasov, S.A.: Application of Azimuthal Decomposition Technique for Validation of CAA Methods. AIAA-2013-2238, 19th AIAA/CEAS Aeroacoustics Conference (2013). <https://doi.org/10.2514/6.2013-2238>
9. Ffowcs Williams, J.E., Hawkins, D.L.: Sound generation by turbulence and surfaces in arbitrary motion. Phil. Trans. R. Soc. A 264, 321–343 (1969). <https://doi.org/10.1098/rsta.1969.0031>
10. Goloviznin, V.M., Samarskii, A.A.: Finite difference approximation of convective transport equation with space splitting time derivative. Jour Matem. Mod. 10(1), 86–100 (1998)

11. Gryazev, V., Kalyan, A., Markesteijn, A., Karabasov, S.: Broad band shock associated noise modelling for high-area-ratio under-expanded jets. *Nature Research Journal of The Acoustical Society of America* 150(2), 1534–1547 (2021). <https://doi.org/10.1121/10.0005976>
12. Juve, D., Sunyach, M., Comte-Bellot, G.: Filtered azimuthal correlations in the acoustic far field of a subsonic jet. *AIAA Journal* 17(1), 112–114 (1979). <https://doi.org/10.2514/3.61076>
13. Karabasov, S.A., Goloviznin, V.M.: Compact accurately boundary-adjusting high-resolution technique for fluid dynamics. *Journal of Computational Physics* 228(19), 7426–7451 (2009). <https://doi.org/10.1016/j.jcp.2009.06.037>
14. Kim, H.D.: Distributed propulsion vehicles. 27th Congress of the International Council of the Aeronautical Sciences, Nice, France, September 19-24, 2010 (2010)
15. Kopiev, V.F., Shur, M.L.: Azimuthal components of turbulent jet sound field: measurement results and their implementation for validation of modern noise computation techniques. *TsAGI Science Journal* 41(1), 3–13 (2010)
16. Kopiev, V.F., Zaitsev, M.Y., Chernyshev, S.A., Kotova, A.N.: The role of large-scale vortex in a turbulent jet noise. *AIAA paper* 99-1839 (1999). <https://doi.org/10.2514/6.1999-1839>
17. Magliozzi, B., Hanson, D.B., Amiet, R.K.: Propeller and propfan noise. *NASA Langley Research Center, Aeroacoustics of Flight Vehicles: Theory and Practice, Vol. 1: Noise Sources* pp. 1–64 (1991)
18. Markesteijn, A.P., Gryazev, V., Karabasov, S.A., *et al.*: Flow and noise predictions of coaxial jets. *American Institute of Aeronautics and Astronautics AIAA Journal: Devoted to Aerospace Research and Development* 58(12) (2020). <https://doi.org/10.2514/1.J058881>
19. Markesteijn, A.P., Jawahar, H.K., Karabasov, S.A., Azarpeyvand, M.: GPU CABARET Solutions for 30P30N Three-element High-lift Airfoil with Slat Modifications. *AIAA 2021-2115, Airframe/High-Lift Noise I* (2021)
20. Markesteijn, A.P., Karabasov, S.A.: CABARET solutions on graphics processing units for NASA jets: Grid sensitivity and unsteady inflow condition effect. *Comptes Rendus – Mécanique* 346(10), 948–963 (2018). <https://doi.org/10.1016/j.crme.2018.07.004>
21. Markesteijn, A.P., Karabasov, S.A.: Simulations of co-axial jet flows on graphics processing units: the flow and noise analysis. *Philosophical Transactions of The Royal Society A: Mathematical, Physical and Engineering Sciences* 377(2159) (2019). <https://doi.org/10.1098/rsta.2019.0083>
22. Najafi-Yazdi, A., Bres, G.A., Mongeau, L.: An acoustic analogy formulation for moving sources in uniformly moving media. *Proc. Royal Soc. A* 467(2125), 144–165 (2011). <https://doi.org/10.1098/rspa.2010.0172>

23. National Academies of Sciences Engineering and Medicine: Commercial Aircraft Propulsion and Energy Systems Research: Reducing Global Carbon Emissions. The National Academies Press, Washington, DC (2016). <https://doi.org/10.17226/23490>
24. Sahoo, S., Zhao, X., Kyprianidis, K.: A review of concepts, benefits, and challenges for future electrical propulsion-based aircraft. *Aerospace* 7(44) (2020), <https://doi.org/10.3390/aerospace7040044>
25. Semiletov, V.A., Karabasov, S.A.: CABARET scheme with conservation-flux asynchronous time-stepping for nonlinear aeroacoustics problems. *Journal of Computational Physics* 253, 157–165 (2013). <https://doi.org/10.1016/j.jcp.2013.07.008>
26. Shur, M.L., Spalart, P.R., Strelets, M.K.: Noise Prediction for Increasingly Complex Jets. Part I: Methods and Tests. Part II: Applications. *Int. J. Aeroacoustics* 4(3+4), 213–266 (2005)
27. Solntsev, I.A., Chintagunta, A., Markesteijn, A.P., Karabasov, S.A.: CABARET on rotating meshes. *Applied Mathematics and Computation* 446(127871) (2023). <https://doi.org/10.1016/j.amc.2023.127871>