CABARET scheme with conservation-flux asynchronous time-stepping for computational aero-acoustics

Vasily Semiletov, Sergey Karabasov

CABARET scheme with conservation-flux asynchronous time-stepping for computational aero-acoustics

V. A. Semiletov and S. A. Karabasov

Whittle Laboratory Cambridge University Engineering Department, 1 JJ Thompson Avenue, CB3 0DY Cambridge, UK
vs346@cam.ac.uk

Explicit time stepping renders high-resolution computational schemes to become less efficient when dealing with non-uniform meshes. The non-uniform meshes are, however, almost unavoidable for capturing strong solution gradients, e.g., for an airfoil boundary layer or a high-Reynolds number jet mixing layer. The problem is that for numerical stability with explicit time stepping, the Courant stability condition forces one to march the solution in time with a global time step that can be very small. Asynchronous time stepping, i.e., updating the solution in different cell sizes according to their local rates, is a promising way for improving the efficiency of explicit methods with highly non-uniform grids. The improvement comes by effectively boosting the local grid CFL number without any compromise in accuracy. In the present paper, a new asynchronous time-stepping technique is implemented for the Compact Accurately Boundary-Adjusting high-REsolution Technique (CABARET) Euler method. Numerical examples for 1D, 2D and 3D flow problems are considered and comparisons with the single-time-step method are made.

1 Introduction

Large disparity of flow scales is a typical feature of aeroacoustic calculations. This requires high-resolution numerical schemes that are able to efficiently propagate acoustic waves without significant dissipation and dispersion errors on computational grids at affordable cost. The latter requirement is especially difficult to maintain with the high-resolution non-uniform grids that are essential in multi-space-time-scale problems. Such challenging applications include airfoil or jet flows, for instance, where the grid nodes typically need to be clustered in the vicinity of a viscous boundary layer or shear layer.

Because of the low-dispersion and low-dissipation requirement of aeroacoustics schemes, most of the numerical schemes used for this kind of applications are based on explicit time stepping. The largest time step with such methods is restricted by the smallest grid size in accordance with the Courant-Friedrichs-Lewy (CFL) stability criterion. With non-uniform grids, the numerical efficiency in case of the single/synchronous time stepping, drops down because all but the smallest grid cells are forced to march in time with a very small time step. The CFL restriction can be relaxed by using semi-implicit or fully implicit schemes, e.g., as done in classical dual time-stepping algorithms where the solution at each sub-iteration is treated as quasi-steady. Such implicit algorithms, however, are generally less accurate for unsteady problems in comparison with the fully explicit schemes.

Asynchronous time stepping, i.e., when the solution in different cell sizes is updated at different rates and adjusted to the cell-local CFL number rather than to a global one, is one possible way of improving the efficiency of explicit methods with highly non-uniform grids without any loss of the original accuracy of explicit algorithms. Typical examples of asynchronous time-stepping algorithms include: (i) adaptive mesh refinement that is based on a hierarchy of nested levels of logically rectangular patches and (ii) adaptive time refinement that allows solution values in different elements to be adapted with different time increments.

For computational aeroacoustics, examples of implementations of asynchronous time stepping include the Multi-Size-Mesh Multi-Time-Step DRP Scheme of Tam and Kurbatskii, 2003 (based around approach (i)) and the Solution-Element Conservation-Element with local time stepping of Chang et al, 2005 (based around approach (ii)).

In this paper, a new asynchronous time-stepping technique, along the same line of thought as in (Dawson and Kirby, 2001) and (Omelechenko and Karimabadi, 2006, 2007) is implemented for the Compact Accurately Boundary-Adjusting high-REsolution Technique (CABARET) scheme (Goloviznin and Samarski, 1998; Karabasov and Goloviznin, 2009). CABARET can be viewed as a generalisation of the Upwind Leapfrog Scheme of Iserlis (1986) to nonlinear conservation laws. It is based on a conservative, low-dissipative and low-dispersive explicit advection scheme with very compact stencil that for linear advection takes only one cell in space and time.

With standard synchronous/single-time stepping, CABARET has been successfully used for computational aeroacoustics and hydrodynamics problems before (e.g., Karabasov and Goloviznin, 2007). The current work is devoted to introducing the asynchronous time stepping in the CABARET scheme with keeping the following important properties: (i) simplicity and compactness of the original CABARET stencil, (ii) strict conservation property and (iii) a built-in recipe for the treatment of inactive flow regions.

2 1D Example

To illustrate the idea of asynchronous time-stepping, a one-dimensional scalar conservation law

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = 0, \quad (1)$$

is considered in the solution domain $x = [0, L],\ t = [0, T]$. The domain is covered by a non-uniform grid of cell volumes with spacings h_i. It is assumed that the positive x-direction corresponds to the increase of mesh index *i*. Each cell is allowed to march in time according to its own time t_i and with its own time step τ_i.

2.1 Basic CABARET scheme

Error! Reference source not found. shows the associated data structure for computational cell in space and time: the solid circles refer to the location of conservation variable U and the open circles stand for the locations of flux variables F. The conservation variables (U) that correspond to the cell centres are labeled with i-indices and the cell faces that correspond to the fluxes (F) are labeled with i, R and i, L where indices i, R and $i + 1, L$ denote different sides of the same face.

Starting from the known conditions at the previous time step, U_i, $F_{i,R}$ and $F_{i,L}$ the CABARET algorithm first advances the solution a half step in time, i.e., at the predictor stage:

$$\frac{\widehat{U}_i - U_i}{\tau_i/2} + \frac{F_{i,R} - F_{i,L}}{h_i} = 0 \quad (2)$$

The solution at the new time step is computed at the corrector stage:

$$\frac{\widehat{U}_i - \widetilde{U}_i}{\tau_i/2} + \frac{\widehat{F}_{i,R} - \widehat{F}_{i,L}}{h_i} = 0, \quad (3)$$

where $\hat{F}_{i,R}$, $\hat{F}_{i,L}$ are the fluxes at the new time level $\hat{t}_i = t_i + \tau_i$. For their calculation, the simple upwind extrapolation is used which amounts to a second-order approximation in space and in time.

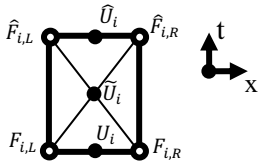


Figure 1: Computational stencil of the CABARET scheme.

2.2 Nonlinear flux reconstruction

Suppose the two adjacent cells have reachd the same local time at the new time step, $\hat{t}_i = \hat{t}_j$, (Figure 2).

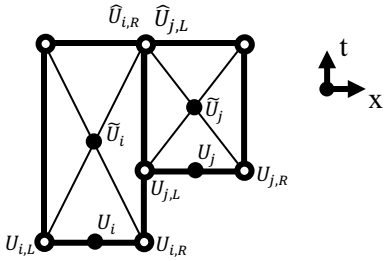


Figure 2: Case of two adjacent cell volumes corresponding to the same local time at the new time level.

Then the computational algorithm for updating the interface flux value $\hat{F}_{i,R}$ is the same as for the homogeneous/single time-stepping. In this case the following algorithm of the flux variable extrapolation and its correction based on the direct application of the solution maximum principle is used:

$$\begin{array}{ll} \hat{U}_{i,R} = 2\widetilde{U}_i - U_{i,L}, & \hat{U}_{i,L} = 2\widetilde{U}_j - U_{j,R}, \\ M_i = \max\left(U_{i,L}, U_i, U_{i,R}\right), & M_j = \max\left(U_{j,L}, U_j, U_{j,R}\right), \\ m_i = \min\left(U_{i,L}, U_i, U_{i,R}\right), & m_j = \min\left(U_{j,L}, U_j, U_{j,R}\right) \\ \hat{U}_{i,R} = \max\left(m_i, \min\left(M_i, \hat{U}_{i,R}\right)\right); & \hat{U}_{i,L} = \max\left(m_i, \min\left(M_i, \hat{U}_{i,L}\right)\right). \end{array} \quad (4)$$

From the two cell-face values, the choice is made based on solving the corresponding Riemann problem with a characteristic decomposition method, which for the case of linear advection equation amounts to the standard upwinding procedure. Suppose the face fluxes are defined in direction of the external normal to the cell face and the positive normal direction is defined according to the direction from cell i to cell j. Then $\hat{F}_{i,R} = -\hat{F}_{j,L} = F\left(\hat{U}_f\right)$, where $\hat{U}_f$ is determined according to the following algorithm

$$\hat{U}_f = \begin{cases} \hat{U}_{i,R}, & \frac{dF(U_i)}{dU} - \frac{dF(U_j)}{dU} > 0, \\ \hat{U}_{j,L}, & \frac{dF(U_i)}{dU} - \frac{dF(U_j)}{dU} < 0. \end{cases} \quad (5)$$

For asynchronous time-stepping with different local time steps, the flux variables that correspond to the left and right side of the same grid face may not always perfectly match in time, as shown in Figure 3. In case of the mismatch, instead of single $\hat{U}_f$ we need to introduce 2 values: $\hat{U}_{f,i}$ and $\hat{U}_{f,j}$. These denote variables on the interface f between cells i and j at the new local times, $\hat{t}_i$, and $\hat{t}_j$, respectively (Figure 3).

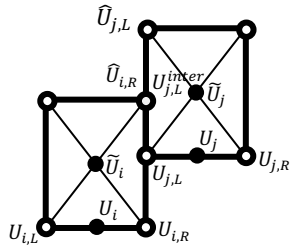


Figure 3: Case of two adjacent cell volumes corresponding to different local times at the new time level.

Let's now assume that $\hat{t}_i < \hat{t}_j$ and for the flux reconstruction at the same space-time location the face variables are linearly interpolated with keeping the second-order approximation of the scheme:

$$U_{j,L}^{inter} = \hat{U}_{i,L} + \left(\hat{U}_{i,L} - U_{j,L}\right)\frac{\hat{t}_i - t_j}{\hat{t}_j - t_j}. \quad (6)$$

Once both flux values, $\hat{U}_{i,R}$ and $U_{j,L}^{inter}$ are known, the flux reconstruction problem for computing $\hat{U}_{f,i}$ reduces to the same one as for the synchronous time stepping algorithm:

$$\hat{U}_{f,i} = \begin{cases} \hat{U}_{i,R}, & \frac{dF(U_i)}{dU} - \frac{dF(U_j)}{dU} > 0, \\ U_{j,L}^{inter}, & \frac{dF(U_i)}{dU} - \frac{dF(U_j)}{dU} < 0. \end{cases} \quad (7)$$

After the new flux variable is computed the corresponding flux f for cell i as defined in a usual manner:

$$\hat{F}_{i,R} = F\left(\hat{U}_{f,i}\right). \quad (8)$$

2.3 Time Step Definition

Local time step is defined from the standard CFL condition by considering the cell-centre and cell-flux values available from the CABARET stencil:

$$\tau_i = \frac{h_i}{\frac{dF(U_i)}{dU}}, \quad \tau_{i,L} = \frac{h_{i,L}}{\frac{dF(U_{i,L})}{dU}}, \quad \tau_{i,R} = \frac{h_{i,R}}{\frac{dF(U_{i,R})}{dU}}, \quad (9)$$

$$\tau_i = \min\left(\tau_i, \tau_{i,L}, \tau_{i,R}, \tau_{max}\right),$$

where τ_{max} is some adjustable large-time-step parameter.

Notably, in order to avoid the interpolation procedure that could become inaccurate and computationally expensive when the difference in the local time between the two adjacent cells tends to the round-off error, the calculation rule for getting time step is modified by introducing a small parameter, τ_{min}:

$$\tau_i = \left[\frac{\tau_i}{\tau_{min}}\right]\tau_{min}. \quad (10)$$

In comparison with the previous works (e.g., Yen, 2011), where time step size was modified according to some multiple of dyadic integer and the minimal local time step, the synchronization in our approach is directly linked to the global output time T, i.e. $\tau_i = \min(\tau_i, T - t_i)$ and always remains local.

2.4 Event Synchronization

It is useful to recall that the CABARET scheme has the following stages: (i) predictor step, (ii) updating of the flux variables from next time level, and (iii) corrector step. For asynchronous time stepping where every cell is allowed to have its own local time t_i and time step τ_i, the major question is how to synchronise all 3 stages of the scheme. First, the predictor/corrector steps only use the local cell information, hence, no special event synchronisation is needed for these stages so we only have to deal with the upwind flux extrapolation stage.

For flux synchronization, let's introduce the cell indicator flags which equal $'true'$ or $'false'$ as the following:

$IU_i = true \Leftrightarrow$ if both $\widehat{U}_{i,R}$, $\widehat{U}_{i,L}$ are updated/known from the previous time step;

$IF_{i,R}, IF_{i,L} = true \Leftrightarrow$ known $\hat{F}_{i,R}$, $\hat{F}_{j,L}$, accordingly;

$IC_i = true \Leftrightarrow$ both fluxes $\hat{F}_{i,R}$, $\hat{F}_{i,L}$ are updated/known;

$ID_i = true \Leftrightarrow t_i = T$.

Initially all these indicators are equal to $'false'$ except for the case $IC_i = true$. The use of indicator flags saves one from doing expensive global-time-data operations to make decisions in each particular cell, e.g., whether it is going to run away in time or not. In our algorithm, such decision is made locally for each cell, i.e., independently on the other cell times. This is quite different in comparison with the currently existing asynchronous time-stepping algorithms, e.g., of Dawson and Kirby (2001) and Omelechenko and Karimabadi (2007), where one needs to track down all sets of cells which correspond to the same local time and which may cause significant computational overhead costs, especially for problems in multiple dimensions.

Let's now suppose we need the solution in all cells to be defined at time T. Then the block-scheme for asynchrony method is written as the following:

1. **while** $(\min(t_i) < T)$**:**
2. *Predictor step* (Figure 4)
   for $\forall i$: **if** $(IC_i == true)$:
   a. compute $\widetilde{U}_i$ // of Eq. (1)
   b. $IC_i = false$

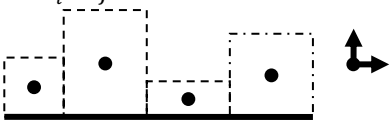


Figure 4: First predictor step.

3. *Flux variable definition*
   for $\forall i$: **if** $(IU_i == false)$:
   a. compute $\widehat{U}_{i,R}$, $\widehat{U}_{i,L}$
   b. $IU_i = true$

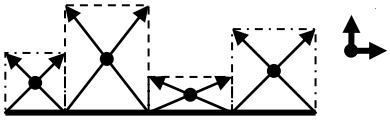


Figure 5: Definition of flux variables on the new time level.

4. *Flux calculation*
   for $\forall f$: // f denotes the face between cells i and j
   if ($IF_{i,R} == false$ && $IF_{j,L} == false$):
   if $(\hat{t}_i \le \hat{t}_j)$:
   a. compute $\widehat{U}_{f,i}, \hat{F}_{i,R}$
   b. $IF_{j,R} = true$
   if $(\hat{t}_i \ge \hat{t}_j)$:
   a. compute $\widehat{U}_{f,j}, \hat{F}_{j,L}$
   b. $IF_{j,L} = true$

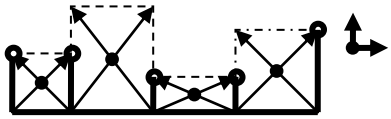


Figure 6: Flux calculation.

5. *Corrector step*
   for $\forall i$: **if** $(ID_i == false)$:
   if $\left(IF_{i,R} = true \;\&\&\; IF_{i,L} = true\right)$:
   a. compute $\widehat{U}_i$
   b. $IC_i = true$
   c. $IU_i = false$
   d. $IF_{i,R} = IF_{i,L} = false$
   e. $t_i = t_i + \tau_i$
   f. compute new τ_i
   g. $U_i = \widehat{U}_i, U_{i,R} = \widehat{U}_{i,R}, U_{i,L} = \widehat{U}_{i,L}$
   h. **if** $(t_i + t_\varepsilon \le T)$: $ID_i = true$, $IC_i = false$

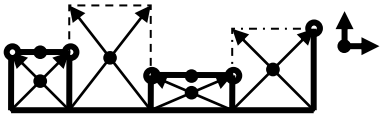


Figure 7: Corrector step.

For solution synchronisation, the conservative corrector-step of the scheme is postponed for each cell where the fluxes have not reached the same local time at the new time level. This is the situation when one of the two adjacent cells has the next local time level that lies in between the current time level and the next time level of the adjacent cell. Because of this, the new fluxes of the cell that corresponds to small time steps are always updated first and do so more frequently in comparison with the grid locations which correspond to the large time steps. This condition is a built-in recipe for saving the computational time in the domain regions where the solution is less inactive and which correspond to large computational steps. This useful feature of the current algorithm is lacking in some of the modern multi-step algorithms in the literature.

2.5 Conservative Flux-Correction

Because of the time-stepping differences, the fluxes across the cell interface may not be evaluated consistently for the small and large cells unless some additional flux correction is performed to restore the conservation property.

Figure 8 shows two adjacent cells that have different time-stepping. According to the asynchronous time-stepping algorithm for time interval $\left[t_i, t_j^n\right]$ (bold line in Figure 8) the two different flux contributions that correspond to the right and left sides of the same cell face $f(i,R;j,L)$ are:

$$\begin{aligned}
\bar{F}_{j,L}^{\left[t_i,t_j^n\right]} &= \sum_{n_j=1}^{n}\left(\bar{\tau}_j^{n_j} F_{j,L}^{n_j} + \bar{\hat{\tau}}_j^{n_j} \hat{F}_{j,L}^{n_j}\right), \\
\bar{\tau}_j^{n_j} &= min\left(\frac{\tau_j^{n_j}}{2}, max\left(0, t_j^{n_j} + \frac{\tau_j^{n_j}}{2} - t_i\right)\right), \\
\bar{\hat{\tau}}_j^{n_j} &= min\left(\frac{\tau_j^{n_j}}{2}, \hat{t}_j^{n_j} - t_i\right); \qquad (11) \\
\bar{F}_{i,R}^{\left[t_i,t_j^n\right]} &= \left(\bar{\tau}_i F_{i,R} + \bar{\hat{\tau}}_i \hat{F}_{i,R}\right), \\
\bar{\tau}_i &= min\left(\frac{\tau_i}{2}, \hat{t}_j^n - t_i\right), \\
\bar{\hat{\tau}}_i &= min\left(\frac{\tau_i}{2}, max\left(0, \hat{t}_j^n - t_i - \frac{\tau_i}{2}\right)\right).
\end{aligned}$$

It is easy to see that the sum of the left and right fluxes is not identically zero, $\bar{F}_{j,L}^{\left[t_i,t_j^n\right]} + \bar{F}_{i,R}^{\left[t_i,t_j^n\right]} \neq 0$. Hence, the following correcting flux needs to be added to one of the face sides, is used:

$$\hat{F}_{i,R}^{corr} = \bar{F}_{j,L}^{\left[t_i,t_j^n\right]} + \bar{F}_{i,R}^{\left[t_i,t_j^n\right]}. \qquad (12)$$

Indeed, by adding the correcting flux of equation (12) to the new flux on the right-hand-side of eq. (8) one obtains a conservative asynchronous time-stepping scheme. Also, in addition to the conservation property of the original single-time-step, the new conservative asynchronous scheme also preserves the CABARET non-oscillatory behaviour. This is because the fluxes for cell with large time-stepping are always defined from the fast-rate updated cells with the fine time-stepping. In these fast cells the flux update is always performed using the CABARET flux correction based on the maximum principle.

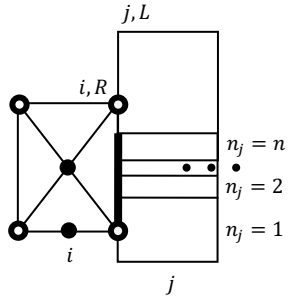


Figure 8: Solution update with different local times.

3 Numerical Examples1D wave propagation

Consider a one-dimensional acoustic wave:

$$\begin{aligned} &p_0 = P(x,0), \quad u_0 = U(x,0), \quad \rho_0 = R(x,0), \\ &p|_{x=0} = P(0,t), \quad u|_{x=0} = U(0,t), \quad \rho|_{x=0} = R(0,t), \\ &p|_{x=L} = P(L,t), \quad u|_{x=L} = U(L,t), \quad \rho|_{x=L} = R(L,t); \end{aligned} \qquad (13)$$

where

$$\begin{aligned} P(x,t) &= e\pi \cdot cos[2a\pi(t - x/c)] + p_{fon}, \\ U(x,t) &= e\pi \cdot \frac{cos[2a\pi(t-x/c)]}{c}, \\ R(x,t) &= \rho_{fon} + P(x,t)/c^2, \end{aligned} \qquad (14)$$

$e = 20, \quad a = 7, \quad p_{fon} = 10^5, \quad \rho_{fon} = 1, \quad c = 340, \quad \pi = 3.14\ldots, \quad L = 100.$

The problem is solved with a non-uniform grid that has the ratio of the largest to the smallest cell size $\frac{h_{max}}{h_{min}} = 36$. For numerical solution, the asynchronous time-stepping has been implemented in a 3D CABARET Euler code on the N x 1 x 1 grid (x,y,z) as shown in Figure 9 where periodic boundary conditions in the y- znd z –directions are imposed.

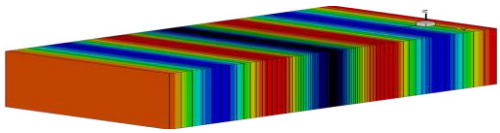


Figure 9: Non-uniform mesh.

Fig.10 shows the snapshots of the instantaneous solution with two different grid resolutions where the analytical solution is also shown for comparsion.

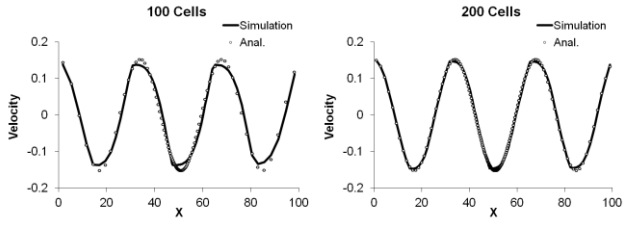


Figure 10: Solution of the 1D acoustic wave problem: comparison of results of the new asynchronous algorithm with the analytical solution for different grid resolutions.

The numerical results for the standard time stepping are visually undetectable from the asynchronous time-stepping solution. The difference between the synchronous single-time-step solution and the asynchronous one is more notable for the grid convergence. Table 1 shows the errors of the two algorithms in several different norms. The single time-stepping and the asynchronous time-stepping both show approximately 1.5-2 order of convergence. The absolute errors of the asynchronous algorithm are smaller in comparison with the single-step method. The latter is because the asynchronous algorithm pushes the CABARET algorithm to march in time with a bigger local CFL number in the coarse regions of the grid. On the other hand, the accuracy of CABARET improves for high CFL numbers, as is the case with many explicit time schemes, e.g., Central Leapfrog or Lax-Wendroff.

	Homogenous time-stepping				Asynchrony			
Cells	P, C	U, C	P, L2	U, L2	P, C	U, C	P, L2	U, L2
26	0.042447	0.0001	0.020344	4.94E-05	0.041279	0.0001	0.019609	4.76E-05
50	0.013108	3.2E-05	0.005574	1.36E-05	0.012253	2.98E-05	0.005067	1.24E-05
100	0.003447	8.5E-06	0.001366	3.39E-06	0.002972	7.31E-06	0.001095	2.72E-06
200	0.000867	2.2E-06	0.000337	9.14E-07	0.000636	1.65E-06	0.000205	5.54E-07

Table 1: The error convergence of the homogeneous time-stepping and the asynchronous time-stepping algorithm: P is pressure, U is velocity, C and L2 are the standard uniform/maximum norm and integral norms, respectively.

The biggest driver for implementing the asynchronous algorithm in CABARET was the code acceleration. Hence, Table 2 demonstrates the gain in the algorithm speed-up due to the asynchronous algorithm, Q_A in the comparison with the single-time-stepping and also with the maximum estimated speed-up $Q_E = \frac{\sum_i h_i/c}{\min_i(h_i/c)}$.

As expected, the theoretical maximum Q_A is less than Q_E because the asynchronous time-stepping has some overheads for performing additional cell cycles, interpolation and the flux-correction procedure. However, this overhead remains small and the asynchronous time-stepping accelerates the solution by more than a factor of 3 for this nonuniform grid configuration.

Cells	Q_E	Q_A
400	3.66	3.125
800	3.66	3.18
1600	3.66	3.14

Table 2: Speed comparison, one-dimension case.

3.2 2D Acoustic wave scattering by a cylinder in a subsonic free-stream flow

The 2D subsonic flow around a cylinder is considered next. A point monopole acoustic source is specified directly below the cylinder center at distance 10 cylinder diameters from it. A uniform subsonic flow of Mach number 0.15 in the x-direction is imposed in the domain through the characteristic boundary conditions. On the cylinder surface a slip boundary condition is assumed. Figure 11 shows the computational grid in the solution domain that consists of the O-type mesh around cylinder that is embedded into an H-type grid mesh close to the external boundaries.

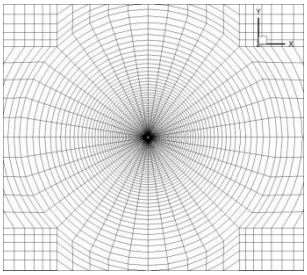
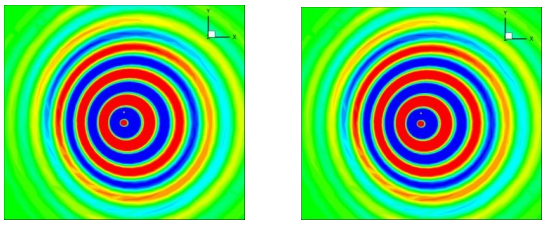


Figure 11: Grid configuration (coarse) around the cylinder.



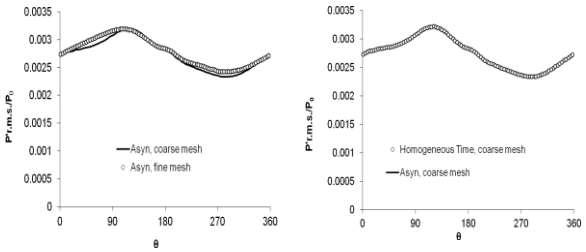
(a) (b)

Figure 12: Instantaneous pressure fluctuation field solution for (a) homogeneous time-stepping, (b) asynchronous time-stepping algorithm.

The ratio of the maximum and the minimum grid size for this problem is $\frac{h_{max}}{h_{min}} = 10.03$. For the numerical solution, the same 3D CABARET Euler code used which now has only 1 periodic boundary in the z-direction. Accordingly, the grid in the z-direction is made of just 1 cell to keep the problem fully 2D.

Figure 12 shows the instantaneous pressure fluctuation fields obtained with and without the asynchronous algorithm used. The two fields virtually coincide.

Figs.13 show the root-mean-square pressure fluctuation directivity (p r.m.s.) normalised by the static pressure at infinity P_0. The directivity is plotted for the circle centred at the acoustic source location with radius of 20 cylinder diameters. Results are obtained for two mesh resolutions: the coarse and the fine grid. The linear scale of the fine grid is a factor of 4 smaller in comparison with the coarse grid. Fig.13b shows the coarse grid solutions for the asynchronous and the single time-stepping method that perfectly collapse to a single curve. Also, the solution with asynchronous time-stepping at two different grid resolution are in a good agreement. Fig.14 shows the corresponding 2D p r.m.s. field. There is some noticeable asymmetry because of the effect of the scattering from the cylinder.



(a) (b)

Figure 13: R.m.s. pressure fluctuations for the 2D acoustic scattering problem: (a) coarse grid and fine grid solution comparison for the asynchronous time stepping, (b) asynchronous time vs single time step solution for same coarse grid.

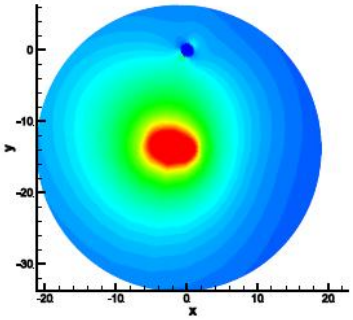


Figure 14: 2D field of r.m.s. pressure fluctuations from the coarse grid solution.

For the 2D problem, the speed-up of the Euler solution due to the asynchronous time-stepping algorithm is a factor of 3, as shown in Table 3. The speed-up is almost the same as it was for the 1D acoustic wave problem, however, in the 2D case the grid-size ratio is more than 3 times smaller in comparison with the 1D problem. This indicates that the relative benefits of using the new asynchronous algorithm should grow with the problem size.

Mesh	Q_E	Q_A
2D, 4 396cells	4.66	3.0

Table 3: Speedup comparison for the 2D acoustic wave scattering problem.

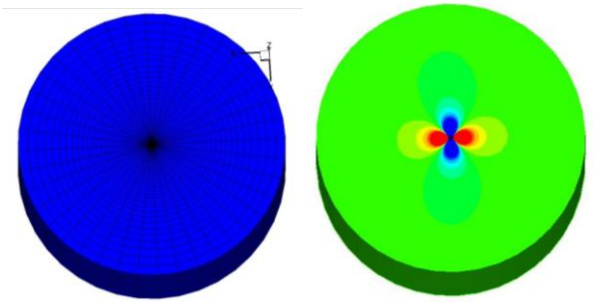
On the other hand, as comparison with the theoretical speed-up shows, there is also some computational overhead increase in the 2D case versus the 1D test case. This is likely to be associated with the increase of the cell-flux communications in multiple dimensions. Hence, the next step is to investigate how strongly the communication costs of the asynchronous time-stepping grow with increase in the number of cell fluxes typical of 3D calculations.

3.3 Dependency of the speed-up gain on the cell-face-flux communications

The following test problem is considered. A cylinder with a slip boundary condition is put in a steady free stream of M=0.15. In the x-y cylinder plane, the computational domain is covered by an O-type grid with the refinement in the vicinity of the solid boundary. The external boundary of the circular open computational domain is located at 50 cylinder diameters from the cylinder centre. To check the solution speed-ups, the same 3D CABARET Euler code is used in two configurations. One is for the 2D problem which corresponds to 1 cell in the span-wise z-direction. The other one is the 3D problem that corresponds to six 2D grids which are uniformly stacked in the z-direction. For both cases, periodic boundary condition in the z-direction and characteristic non-reflecting conditions in the x-y cylinder plane are used. For investigation, 3 computational grids are considered: 2 grids are used for the 2D problem and 1 is used for the 3D problem. The 2D section of the 3D grid is similar to the 2D grid with a bigger disparity of the grid sizes/bigger stretching (2000 cells). The grid details are summarised in Table 4.

Figure 1215 shows the 3D mesh and a snapshot of the corresponding pressure solution for the asynchronous time-stepping algorithm. Very similar solutions are obtained for

the 2D grid configurations. The results of the speed-up gains due to the asynchronous algorithm in comparison with the single time-step method for each grid are summarised in Table 4**.** It can be noticed that the actual code acceleration due to the growing number of cell faces slows down with increase of the problem dimension. However, in comparison with increase of the number of cell-face communications this relative efficiency decrease is very moderate. Indeed, while the ratio of the number of the large cells to the small cells was kept the same the same the increase in the number of cell-face communications by a factor of 5 leads to a relative drop of speed-up efficiency of the asynchronous algorithm by some 30%.



(a) (b)

Figure 155: (a) 3D mesh around the cylinder, (b) pressure distribution

Mesh	N_R	N_φ	N_Z	Q_E	Q_A
2D, 2 000cells	50	40	1	3.6	2.3
2D, 3 000 cells	50	60	1	3.6	2.08
3D, 10 000 cells	50	40	5	3.6	1.67

Table 4: Speed-up gain comparisons for the 2D and 3D cases.

4. Conclusion

A new asynchronous time stepping algorithm is proposed for computational aeroacoustic problems. The efficiency of the new method is demonstrated for the CABARET Euler scheme in 1, 2 and 3 spatial dimensions. In particular, it has been shown that the new asynchronous method not only maintains the same convergence rate of the original second-order CABARET method but also decreases the absolute error because of the local time-stepping performed closer to the CABARET's optimal CFL condition in case of the asynchronous algorithm. The speed-up gain of the asynchronous time-stepping method generally increases with the problem size, i.e., the ratio of the large-size cells to the small ones, and only weakly depends on the increase in cell-flux communications for multidimensional problems. For a 2D test problem of acoustic wave interaction with a cylinder in a subsonic free stream, the new asynchronous method shows a 3-fold acceleration in comparison with the original single-time stepping for the maximum to minimum grid size ratio of about 10.

It can be further expected that the real benefits of the new asynchronous time-stepping method are expected for calculations of unsteady viscous flow problems, which typically require very non-uniform grids to capture fine-viscous-scale solution details. In such problems, the ratio of the maximum to the minimum grid size can be as large as 10000.

Acknowledgments

The work has been supported by UK Engineering and Physical Sciences Research Council (EPSRC Grant EP/I017747/1). One of the authors (SK) gratefully acknowledges the support of the Royal Society of London.

5 References

[1] S.C. Chang, Y. Wu, V. Yang and X.Y. Wang, "Local Time-Stepping Procedures for the Space-Time Conservation Element and Solution Element Method", *International Journal of Computational Fluid Dynamics,* Vol. 19, No 5, 359-380 (July 2005)

[2] C. Dawson and R. Kirby, "High resolution schemes for conservation laws with locally varying time steps", *SIAM J. Sci. Comput*. 22(6), 2256 (2001)

[3] V.M. Goloviznin, and A.A. Samarskii, "Difference Approximation of Convective Transport with Spatial Splitting of Time Derivative", *Mathematical Modelling*, Vol. 10, No. 1, 86-100 (1998)

[4] A. Iserles, "Generalized Leapfrog Methods", IMA Journal of Numerical Analysis, 6, 3 (1986)

[5] S.A. Karabasov, V.M. Goloviznin, "New Efficient High-Resolution Method for Nonlinear Problems in Aeroacoustics", *AIAA Journal*, Vol. 45, No 12 (2007)

[6] S.A. Karabasov, V.M. Golovizni, "Compact Accurately Boundary Adjusting high-REsolution Technique for Fluid Dynamics", J. Comput.Phys., 228(2009)

[7] Y.A. Omelechenko, H. Karimabadi, "Self-adaptive time integration of flux-conservative equations with sources", *Journal of Computational Physics* 216, 179-194 (2006)

[8] Y.A. Omelechenko, H. Karimabadi, "A time-accurate explicit multi-scale technique for gas dynamics", *Journal of Computational Physics* 226, 282-300 (2007)

[9] C.K.W. Tam and K.A. Kurbatskii, , "Multi-Size-Mesh Multi-Time-Step Dispersion-Relatiion-Preserving Scheme for Multiple-Scales Aeroacoustics Problems", *International Journal of Computational Fluid Dynamics*, 17(2), 119-132 (2003).

[10] J.C. Yen, "Demonstration of a Multi-Dimansional Time-Accurate Local Time Stepping CESE Method", *17th AIAA/CEAS Aeroacoustics Conference (32nd AIAA Aeroacoustics Conference)*, Portland, Oregon (05-08 June 2011)