Toward the use of LES for industrial complex geometries. Part II: Reduce the time-to-solution by using a linearised implicit time advancement

Thomas Berthelon, Guillaume Sahut, Julien Leparoux, Guillaume Balarac, Ghislain Lartigue, Manuel Bernard, Vincent Moureau, Olivier Métais

ARTICLE TEMPLATE

Toward the use of LES for industrial complex geometries. Part II: Reduce the time-to-solution by using a linearized implicit time advancement.

T. Berthelon[a], G. Sahut[a], J. Leparoux[b], G. Balarac[a,c], G. Lartigue[d], M. Bernard[a], V. Moureau[d], and O. Métais[a]

[a]Université Grenoble Alpes, CNRS, Grenoble-INP, LEGI, 38000 Grenoble, France; [b]Safran Tech, Rue des Jeunes Bois, Châteaufort, Magny-Les-Hameaux, 78772, France; [c]Institut Universitaire de France (IUF), 75000 Paris, France; [d]CORIA, CNRS UMR6614, Normandie Université, UNIROUEN, INSA of Rouen, France

ARTICLE HISTORY
Compiled November 26, 2023

ABSTRACT
The strong increase in computational power observed during the last few years has allowed to use Large Eddy Simulation (LES) for industrial configurations. Nevertheless, the time-to-solution is still too large for a daily use in the design phases. The objective of this work is to develop a new time integration method to reduce the time-to-solution of LES of incompressible flows by allowing the use of larger time-step. The projection method, probably the most commonly used method in the context of LES of incompressible flow, is generally applied using explicit time advancement which constrains the time-step value for stability reasons (CFL and Fourier constraints). The time-step can then be small with respect to the physical characteristic times of the studied flow. In this case, an implicit time advancement method, which is unconditionally stable, can be used. However, this leads to non-linear resolution of momentum equation which can strongly increase time-to-solution because of non-linear iterations inside a physical iteration. In order to relax the stability constraints while minimizing the computational cost of an iteration, a linearized implicit time advancement based on Backward Differentiation Formula (BDF) scheme is proposed in this work. The linearization is performed using an extrapolated velocity field based on the previous fields. This time integration is first evaluated on a turbulent pipe test case. It is observed a time-to-solution up to five times lower than the explicit time integration while keeping the same accuracy in terms of mean and fluctuating velocity fields. In order to incorporate this new time advancement method in the automatic mesh convergence developed in Part I, a time-step control method based on the local truncation error is used. The resulting automatic time-step and mesh procedure is evaluated on a turbulent round jet case and on PRECCINSTA configuration, a swirl burner which is a representative case of an industrial aeronautical injection system. This new procedure leads to a time-to-solution up to three times lower than the previous procedure, presented in Part I.

KEYWORDS
Large-Eddy Simulation, Time integration, Automatic convergence.

1. Introduction

High-Fidelity simulations are now possible in various complex geometry configurations with the growth of computational resources. In this context, Large-Eddy Simulation (LES) which consists in explicitly solving the largest scales of the flow and modeling the effect of the smallest ones introducing a Subgrid-Scale (SGS) model, appears as a compromise between Direct Numerical Simulation (DNS) and statistical approaches such as Reynolds-Averaged Navier-Stokes equations (RANS). In a previous work [1], a user-independent mesh adaptation strategy has been proposed to guarantee reliable LES in any configuration, without a priori knowledge of the flow dynamics. This paves the way to use LES as a decision support tool for various applications. However, to achieve this goal, the objective is now to be able to reduce as much as possible the time-to-solution of LES to be compatible with project time scales in various industrial or geophysical application fields. The objective of this work is then to revisit the time advancement scheme for LES of viscous incompressible flow to be able to overcome the stability constraint and use larger time-steps to reduce the time-to-solution of simulations.

LES of viscous incompressible flow are most often based on the projection method for pressure-velocity coupling initially introduced by Chorin [2] and Temam [3]. This method can be qualified as prediction-correction method. Indeed, in a first step the velocity is predicted using the momentum equation. While this step advances accurately the solenoidal part of the velocity field, errors in the continuity equation are introduced. Then, this velocity prediction is corrected with a pressure term in order to ensure the continuity equation. This correction step requires the solution of a Poisson equation. The common practice is to perform the prediction using explicit or semi-implicit schemes, which consists in using an explicit scheme for the non-linear convective term and an implicit scheme for the viscous term. In both cases, the time-step has to satisfy the Courant-Friedrichs-Lewy (CFL) constraint to ensure the stability of the scheme. It is then possible that this stability constraint leads to time-steps smaller than the smallest physical characteristic times given by the resolved spatial scales in LES. More precisely, this criterion can lead to very small time-steps in high-velocity region with small cells. However, if the gradients are small in these regions, this limitation is not physical but only numerical. In this case, it should be preferable to use a fully implicit scheme in order to get rid of this constraint. Choi and Moin [4,5] have for example shown that it was possible to gain a factor of 5 on the required CPU time on turbulent flow over riblets. To this purpose, they used the implicit Crank-Nicolson scheme. The non-linearity of the convective term forces them to use a Newton-like method which can be computationally expensive. In order to avoid the non-linear iterations of these methods, Kim *et al.* [6] use the Beam and Warming procedure [7] to linearize the convective term of the Crank-Nicolson scheme. The same idea is present in the work of Simo and Armero [8] who identify a linear implicit scheme of order 2 as well as in that of Olshanskii *et al.* [9] who use a linearized Backward Difference Formulas (BDF) scheme. These different studies show that it is possible to use an implicit scheme to solve incompressible flows, which allows to use time-steps larger than those satisfying the CFL constraint. Nevertheless, a too large time-step can not allow to simulate adequately the studied physical phenomenon. For example, Choi and Moin [5] have shown that for a DNS of a turbulent channel, the turbulent fluctuations can not sustain if the time-step is larger than the characteristic time associated with the Kolmogorov scale. In an LES context, it is difficult to know a priori the relevant time-step. It is however possible to choose the time-step according to the accuracy of

the time integration scheme. This kind of methods is called adaptive time-step methods. In the fluid mechanics literature, these methods have been designed for a large number of time integration schemes [10–12]. They have in common the use of an estimate of the temporal integration error to determine the next time-step in order to reach a specified user-defined error value.

In this work, a linearized implicit time advancement is proposed. The accuracy and the performance in terms of time-to-solution are compared with classic explicit time advancement. Moreover, a time-step adaptation procedure is used, based on the error estimation of the BDF time integration [13,14]. However, there is no universal temporal error value to ensure that the LES computation is accurate. This is why an automatic and a user-independent procedure is proposed to determine the target value of this temporal error. Finally, this procedure is combined with the mesh adaptation strategy proposed in Part I [1] to obtain a user-independent procedure by combining time-step and mesh convergence for any LES configuration at reduced time-to-solution. The paper is organized as follows. Section 2 presents the explicit solver used in this study as a reference by introducing the key concepts to understand the development of the linearized implicit time advancement. Section 3 details this implicit time advancement by introducing the linearization of the convective term using an extrapolated velocity (section 3.1). The proposed method is then evaluated on the academic test case of a turbulent flow pipe, and compared with a classic explicit time advancement in terms of time-to-solution (section 3.2). The time-step adaptation procedure is then detailed in section 4.1 and is embedded in the mesh adaptation strategy by adding a method to determine the target value of the temporal error (section 4.2). The overall procedure is first applied to a turbulent round jet configuration (section 4.3). Finally, it is applied to the industrial case of the PRECCINSTA swirl burner (section 4.4). The results are discussed in terms of accuracy and reduction of time-to-solution in comparison with the use of explicit time advancement.

2. Numerical tool: LES solver based on unstructured body-fitted meshes

As in the previous paper [1], LES were performed using the YALES2 flow solver [15]. This code solves the incompressible and low-Mach-number Navier-Stokes equations for turbulent flows on unstructured meshes using a projection method for pressure-velocity coupling [2,3]. It relies on central finite-volume schemes and on highly efficient linear solvers [16]. YALES2 solver is able to deal with unstructured grids composed of tetrahedral and hexahedral elements allowing to perform LES or DNS of complex geometries with dynamic mesh adaptation [17,18] in the context of massively parallel computations. For all presented cases, the dynamic Smagorinsky subgrid-scale model is used [19]. In this work, the method mentioned as explicit corresponds to the projection method in which the prediction step is performed using an explicit scheme for the non-linear convective term and an implicit scheme for the viscous term. For the sake of clarity, the explicit method is detailed using the forward Euler method, but in practice a fourth-order modified Runge-Kutta scheme [20] is used with a CFL number set to 0.9. The first step of the projection method consists in making a prediction of velocity at time $n+1$, named $u^\star$, by using the previous pressure field,

$$\frac{u^\star - u^n}{\Delta t} + \nabla \cdot (u^n \otimes u^n) = \nabla \cdot ((\nu + \nu_{t,u^n})\nabla u^n) - \frac{1}{\rho}\nabla P^{n-\frac{1}{2}}, \tag{1}$$

where ν_{t,u^n} represents the turbulent viscosity of the LES model. Then, the previous pressure gradient is removed[1],

$$u^{\star\star} = u^{\star} + \Delta t \frac{1}{\rho} \nabla P^{n-\frac{1}{2}}. \tag{2}$$

In order to find the current pressure $P^{n+\frac{1}{2}}$ which leads to divergence-free velocity, the following Poisson equation is solved:

$$\nabla^2 P^{n+\frac{1}{2}} = \frac{\rho}{\Delta t} \nabla \cdot u^{\star\star}. \tag{3}$$

Finally, the velocity field at time $n+1$ is obtained with the correction step,

$$u^{n+1} = u^{\star\star} - \frac{\Delta t}{\rho} \nabla P^{n+\frac{1}{2}}. \tag{4}$$

This method is widely used in the context of LES of incompressible flows. In particular, this is the method employed during the automatic mesh convergence procedure proposed in the previous paper [1]. Nevertheless, the time-step Δt is constrained by the stability of the prediction step, Eq. (1), typically CFL$\leq$1 for forward Euler method. This can lead to a prohibitive time-to-solution for aplications.

3. Implicit time advancement

3.1. Linearized implicit time advancement using extrapolated velocity

One way of reducing the time-to-solution of LES is to increase the time-step. For this purpose, it is necessary to use implicit time integration to overcome the CFL and Fourier constraints, due to the advection and diffusion terms, respectively. The prediction step can then be performed using the BDF time integration scheme at order p,

$$\alpha_0 u^{\star} + \sum_{k=1}^{p} \alpha_k u_i^{n+1-k} + \nabla \cdot (u^{\star} \otimes u^{\star}) = \nabla \cdot ((\nu + \nu_{t,u^{\star}}) \nabla u^{\star}) - \frac{1}{\rho} \nabla P^{n-\frac{1}{2}}. \tag{5}$$

The coefficients α_k depend only on the successive time-steps. These coefficients are computed thanks to a recursive function detailed in [12]. This equation is non-linear and requires Newton-like methods, involving evaluation of Jacobian matrices, that can be complex and/or costly. The idea of the implicit method developed in this work is to linearize this equation by introducing an extrapolation of order l of the velocity field $\tilde{u}^{n+1}$:

$$\tilde{u}^{n+1} = \sum_{k=1}^{l+1} \gamma_k u^{n+1-k}. \tag{6}$$

[1]This could directly be done in the first step for the Euler method, but not for Runge-Kutta methods of order greater than one.

The coefficients γ_k correspond to the Lagrange polynomial evaluated at time t^{n+1}:

$$\gamma_k = \prod_{m=1, m\neq k}^{l+1} \frac{t^{n+1} - t^{n+1-m}}{t^{n+1-m} - t^{n+1-k}}. \tag{7}$$

This allows to write the prediction step, by using an implicit approach, as a linear equation with $u^\star$ as unknown,

$$\alpha_0 u^\star + \sum_{k=1}^{p} \alpha_k u_i^{n+1-k} + \nabla \cdot (u^\star \otimes \tilde{u}^{n+1}) = \nabla \cdot ((\nu + \nu_{t,\tilde{u}^{n+1}}) \nabla u^\star) - \frac{1}{\rho} \nabla P^{n-\frac{1}{2}}. \tag{8}$$

The next steps of the method are similar to the ones of the explicit time advancement. The previous pressure is removed,

$$u^{\star\star} = u^\star + \frac{1}{\alpha_0 \rho} \nabla P^{n-\frac{1}{2}}. \tag{9}$$

The Poisson equation is solved to find the current pressure,

$$\nabla^2 P^{n+\frac{1}{2}} = \alpha_0 \rho \nabla \cdot u^{\star\star}. \tag{10}$$

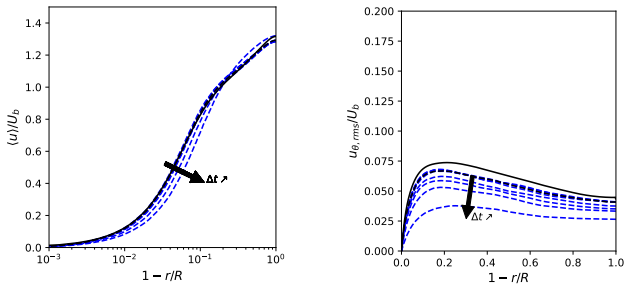
Finally, the velocity is corrected to guarantee the free divergence condition,

$$u^{n+1} = u^{\star\star} - \frac{1}{\alpha_0 \rho} \nabla P^{n+\frac{1}{2}}. \tag{11}$$

This method is a generalization with arbitrary orders of the *linearized BDF2 projection scheme* proposed by Olshanskii *et al.* [9]. The appendix A illustrates the theoretical convergence order in a pure convection test case, showing the benefit of the linearization. The implementation of this method in YALES2 flow solver is relatively straightforward and can easily benefit from development on resolution of linear systems in HPC context [16,21]. Nevertheless, special care is required for the implicitation of boundary conditions. Indeed, depending on the type of a given boundary condition (inlet, outlet, wall, ...) and the presence of backflow, the value of the velocity predictor on the boundary can be known (*e.g.*, previous or reference value, if a Dirichlet condition is used) or unknown (i.e., equal to the one being computed), resulting in a term located in the right- or left-hand side of the linear system of Eq. (8), respectively. In order to take backflow into account in Eq. (8), the volumetric flow rates across all domain boundaries are computed using u^n. Backflow is detected depending on the sign of the scalar product of the volumetric flow rate and the normal vector to the boundary, with the convention that the latter always points to the outside of the domain. Table 1 summarizes the treatment of boundary conditions for inlets and outlets depending on the presence of backflow. Once the left- and right-hand sides are constructed, the linear system, Eq. (8), is solved using the BiCGStab(2) linear solver [21] to deal with the anti-symmetric part of the matrix. The BDF integration order and the extrapolation order are chosen such as $p = l = 2$ because it seems a good compromise between stability and accuracy of the linearized implicit scheme.

Table 1. Boundary conditions for Eq. (8) depending on the presence of backflow: u^{ref} is a known reference condition, *e.g.*, a Dirichlet condition; lhs and rhs stand for left- and right-hand sides, respectively.

Boundary type	Without backflow	With backflow
inlet	u^{ref}, rhs	u^*, lhs
outlet	u^*, lhs	u^n, rhs



(a) Mean axial velocity

(b) Azimuthal velocity rms

(c) Radial velocity rms

(d) Axial velocity rms

Figure 1. Comparison of turbulent pipe flow statistics: DNS (black line), explicit (black dashed line) versus implicit method for different time-steps (blue dashed lines). The different time-steps are kept constant during each simulation, and they roughly correspond to a range of CFL numbers between 0.9 and 100.

3.2. Validation and performance on turbulent pipe flow

To assess the performance and the precision of the linearized implicit time integration method, the case of a turbulent pipe flow with a Reynolds number equal to $5,300$, based on the bulk velocity and the pipe diameter, is considered. The flow configuration is similar to the DNS performed by Wu and Moin [22] on a structured mesh composed of 67.7 million cells and is identical to the one treated in the previous paper [1]. The mesh is the one obtained by the automatic mesh convergence detailed in section 4.4.2 of the previous paper [1] and contains 58,556,190 elements. The computations presented in this section are performed with constant time-steps. The linearized implicit time advancement is used with 8 values of time-steps. The smallest one corresponds to the average value obtained using the explicit time integration performed at a CFL number equal to 0.9, and is mentioned as Δt_{ref}. The largest time-step corresponds approximately to 100 Δt_{ref}. Figure 1 shows statistical quantities comparison between explicit time integration, performed with Δt_{ref}, and linearized implicit time integration. It can be seen that the larger the time-step, the greater the discrepancies.

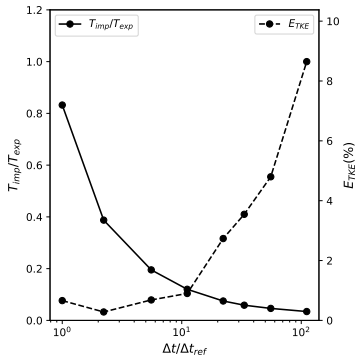


Figure 2. Evolution of time-to-solution scaled by time-to-solution of the explicit case, T_{imp}/T_{exp}, and error on turbulent kinetic energy, E_{TKE}, as a function of the time-step.

The error on the turbulent kinetic energy (TKE), E_{TKE}, is defined as the L^2-norm of the TKE difference between explicit and implicit time integration normalized by the maximum of the TKE in explicit case. The variation of E_{TKE} as a function of time-steps is shown in Fig. 2. For $\Delta t/\Delta t_{ref} < 10$ the error is less than 1%, then increases rapidly to more than 8%. Figure 2 also shows the evolution of time-to-solution of the linearized implicit time advancement, T_{imp}, normalized by time-to-solution of the explicit time advancement T_{exp}. It can be seen that the larger the time-step, the lower the time-to-solution. For $\Delta t/\Delta t_{ref} \approx 10$, the time-to-solution for linearized implicit time advancement is more than 5 times lower than in the explicit case. Note that due to very small elements close to the wall, the semi-implicit diffusion used for the explicit case needs a significant number of sub-steps. In this particular case, the linearized implicit approach has then a smaller time-to-solution, even at small time-step. As a result, the use of implicit time integration allows a significant reduction in time-to-solution while maintaining the accuracy of the statistics. Nevertheless, it is difficult to know a priori the maximum time-step that allows to keep a given level of accuracy. This is why an automatic convergence procedure based on temporal error control is proposed in the next section.

4. Automatic time-step and mesh determination

4.1. Time-step determination

A strategy is required to determine the time-steps of the simulation. The choice has been made to base this strategy on the local truncation error. The time-step is chosen at each iteration in order to stay at a user-defined error level. The idea behind the choice to control time-step with temporal error is to draw a parallel with the automatic meshing approach presented in the previous paper [1], which used spatial discretization error in order to adapt mesh size. The global strategy will be presented in section 4.2.

The determination of time-step is based on the local truncation error E_p^{n+1}, which

can be expressed in the framework of a p-order BDF time integration by [13],

$$E_p^{n+1} = \frac{t^{n+1} - t^n}{t^{n+1} - t^{n-p}} |\boldsymbol{u}^{n+1} - \tilde{\boldsymbol{u}}^{n+1}|_\infty. \tag{12}$$

In this equation, $\boldsymbol{u}^{n+1}$ represents the vector of all the nodes and the three directions of numerical evaluation of u at time t^{n+1}. Moreover, this definition implies the velocity extrapolation:

$$\tilde{\boldsymbol{u}}^{n+1} = \sum_{k=1}^{p+1} \gamma_k \boldsymbol{u}^{n+1-k}. \tag{13}$$

Contrary to Eq. (6), the order of extrapolation is imposed by the order of the BDF scheme. The adaptive time-step method is based on this evaluation of the local truncation error at time t^{n+1} in order to estimate the next time-step, $\Delta t_{n+1} = t^{n+2} - t^{n+1}$, which is defined relatively to the current time-step Δt_n with the ratio σ such that $\Delta t_{n+1} = \sigma \Delta t_n$. The procedure is based on the expected behavior of the error with the time-step within the so-called asymptotic range [12],

$$E_p^{n+1} \approx C \Delta t_n^{p+1}, \tag{14}$$

where C is assumed independent of the time-steps. If E_p^{n+1} is evaluated with Eq. (12) and E_p^{n+2} is targeted to a user-defined error ϵ_T, Eq. (14) leads to:

$$\sigma = \left(\frac{\epsilon_T}{E_p^{n+1}} \right)^{\frac{1}{p+1}}. \tag{15}$$

The evaluation of the error at iteration n is used to compute the time-step at iteration $n+1$. Although the strategy of time-step determination is independent of the linearized implicit method with extrapolated velocity, there is a strong link between these two aspects of the developed approach. Indeed, the error made in the linearization of the convective term in the implicit approach is directly linked to the quality of the extrapolation of the advecting velocity. It appears then natural to determine the time-step to control this extrapolation error. The benefit, in term of computational cost, of the adaptive method is illustrated on a pure convection test case in appendix A. Note that for all the linearized implicit cases presented in this paper, the BDF order as the extrapolation order for the linearization is fixed to 2.

4.2. Methodology to determine the target value of the temporal error

The automatic time-step and mesh convergence is an extension of the mesh convergence presented in the previous paper [1], where the mesh is adapted until the global turbulent kinetic energy (TKE) production, $P_{\text{TKE}}^{\mathcal{V}}$, and the global molecular dissipation, $D_\nu^{\mathcal{V}}$, are independent of the mesh. This procedure is based on the criterion $QC_1(\vec{x})$ allowing to control the spatial discretization error and on the criterion $QC_2(\vec{x})$ allowing to guarantee that enough turbulent scales are resolved [18]. The mesh convergence procedure proposes to sequentially determine the target values for $QC_{1,T}$ and $QC_{2,T}$. The extension proposed in this work is to use linearized implicit time integration while

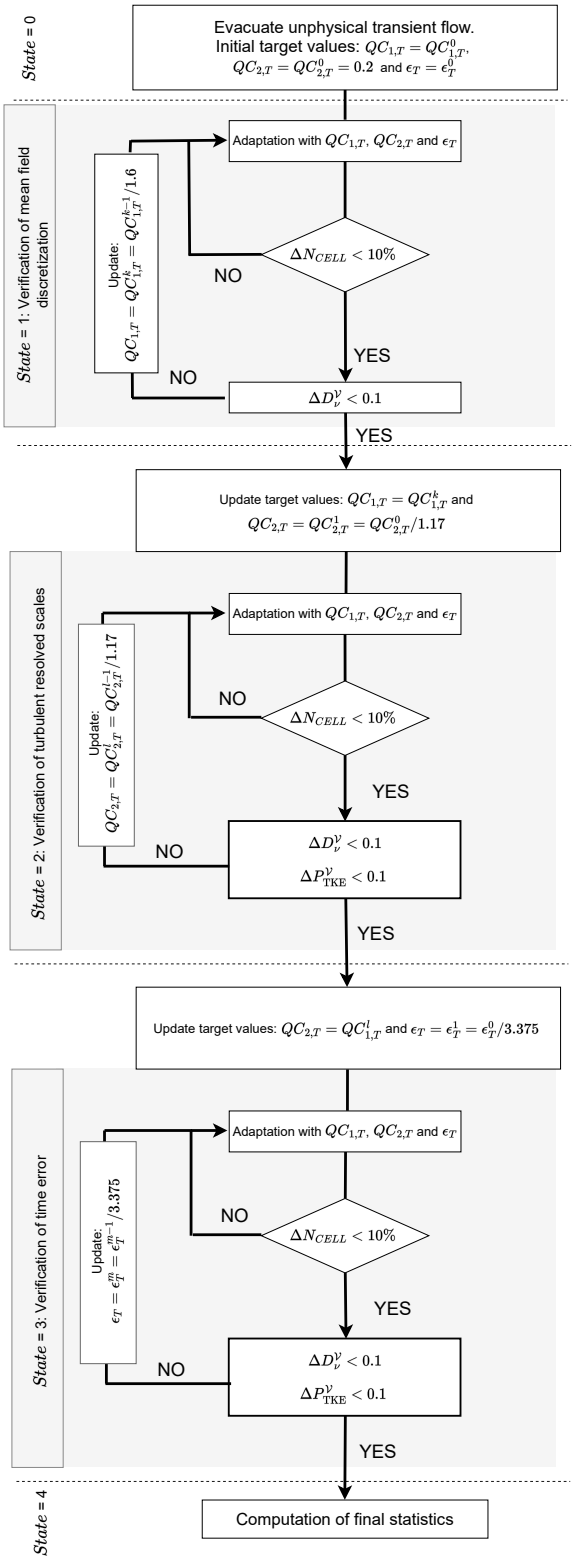


Figure 3. Overview of the automatic time-step and mesh convergence procedure. ΔN_{CELL} is the relative variation of the number of mesh cells before and after an adaptation.

Table 2. Comparison of time-to-solution given in hours for the explicit reference automatic mesh convergence (T_{exp}) and linearized implicit automatic time-step and mesh convergence (T_{imp}) for the turbulent round jet case.

State	**0**	**1**	**2**	**3**	**4**	**Total**
T_{exp}	0.08	7.78	9.39	-	10.05	**27.30**
T_{imp}	0.04	2.24	1.80	1.22	2.69	**7.99**
T_{exp}/T_{imp}	2.00	3.47	5.21	-	3.73	**3.41**

determining automatically the target time error value, ϵ_T. Therefore, the new procedure still does not require any a priori flow knowledge or any comparison to reference results, which is mandatory to be able to use LES in complex flows as a predictive tool. Note also that the procedure has been fully automated and it does not require any action from the user.

A sketch of the overall algorithm is shown in Fig. 3. The first computation is performed with an initial and arbitrary CFL number which allows to determine the initial value of $\epsilon_T = \epsilon_T^0 = \langle |E| \rangle$ ($State = 0$). Then, the values of $QC_{1,T}$ ($State = 1$) and $QC_{2,T}$ ($State = 2$) are obtained with iterative process involving the global molecular dissipation, $D_\nu^{\mathcal{V}}$, and the global TKE production, $P_{\text{TKE}}^{\mathcal{V}}$. This procedure is detailed in [1]. Keeping the final values for $QC_{1,T}$ and $QC_{2,T}$, a reduced value $\epsilon_T = \epsilon_T^1 = \epsilon_T^0/\beta$ is tested. The coefficient β is chosen in order to reduce the time-step by a factor of 1.5. Using temporal mean of Eq. (14) in the case of $p = 2$, the coefficient leading to a reduction of a factor 1.5 of the time-step is $\beta = 1.5^3 = 3.375$. The procedure is repeated m times, until the deviation of the global molecular dissipation, $\Delta D_\nu^{\mathcal{V}}$, and the deviation of the global transfer to TKE, $\Delta P_{\text{TKE}}^{\mathcal{V}}$ are less than 10%. Then, the final value for ϵ_T is fixed equal to ϵ_T^m. This part is identified as $State = 3$ in Fig. 3. Lastly, the final mesh is then obtained and final statistics are computed ($State = 4$).

4.3. Application to turbulent round jet

The proposed automatic time-step and mesh convergence procedure is applied to the turbulent round jet case treated in the previous paper [1]. The numerical set-up including boundary conditions and domain size is detailed in section 4.4.1 of the previous paper [1]. The results obtained with automatic mesh convergence with explicit time integration are considered as reference. The procedure starts from the same initial tetrahedral mesh, composed of 307,806 elements, but the simulation is started with a constant CFL number of 10. The evolution of the main quantities characterizing the procedure is shown in Fig. 4. Based on the statistics of the initial mesh ($State = 0$), the sequential determination of the target values for $QC_{1,T}$ ($State = 1$), $QC_{2,T}$ ($State = 2$) and ϵ_T ($State = 3$) starts. The number of elements has a strong increase at the first steps, leading to a final mesh composed of $8,573,945$ elements which is very close to the $8,125,037$ elements of the reference case. The evolution of the global molecular dissipation and the global transfer to TKE production are also comparable to the reference case. The temporal mean time-step decreases strongly during the first adaptation before converging to value until $State = 3$ where the change of ϵ_T leads to a reduction of the time-step as expected. The statistics prediction obtained by automatic time-step and mesh convergence is compared to the reference case in Fig. 5. There is a good agreement for both mean and root mean square (rms) velocity profiles at the different stages of the round jet transition, which validates the precision of the linearized implicit time advancement. Table 2 shows the comparison of time-to-solution

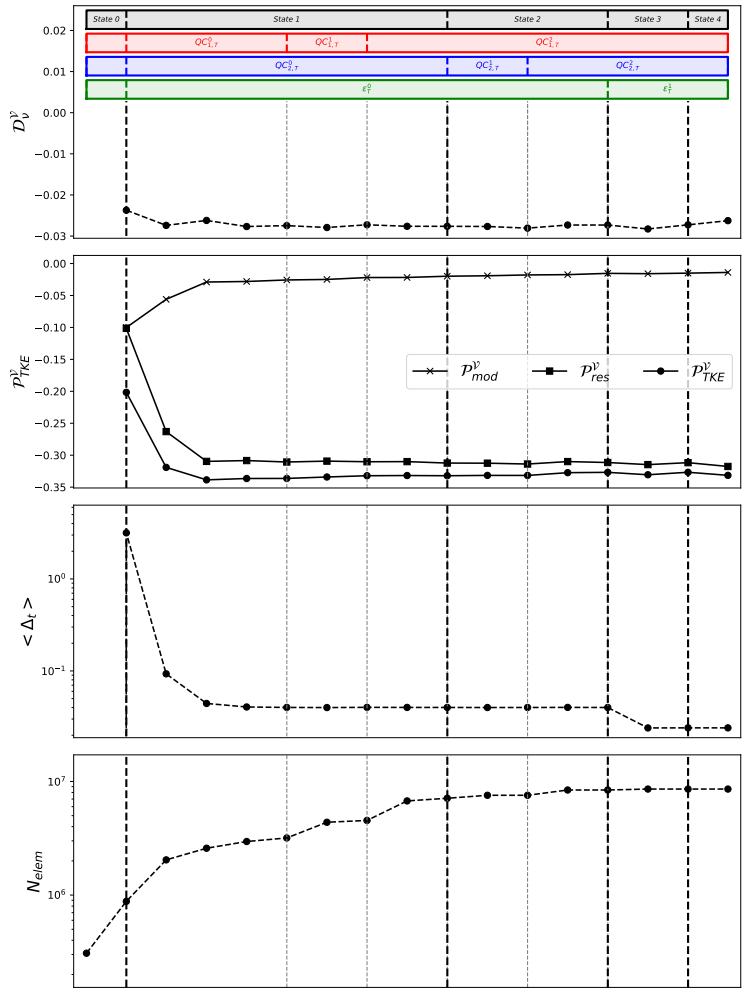


Figure 4. Evolution of the key parameters during the different states of the automatic time-step and mesh convergence procedure in the turbulent round jet configuration. From top to bottom: the global molecular dissipation, $D_\nu^{\mathcal{V}}$, the global TKE production, $P_{\mathrm{TKE}}^{\mathcal{V}}$, the mean time-step $\langle \Delta t \rangle$ and the number of elements, N_{elem}. Note that the resolved, $P_{\mathrm{res}}^{\mathcal{V}}$, and the modeled $P_{\mathrm{mod}}^{\mathcal{V}}$, parts of the TKE production are also shown.

for the different states. Even if an additional step is needed to guarantee the temporal accuracy, the improvement of performance due to linearized implicit time advancement allows to reduce significantly the time-to-solution of the other states. At the end, the automatic time-step and mesh convergence is more than 3 times faster than the reference case. This confirms that the proposed linearized implicit time method is able to correctly predict shear layer transition in addition of wall-bounded flows, as shown in Section 3.2.

4.4. Application to the PRECCINSTA burner

The proposed automatic time-step and mesh convergence procedure is finally applied to the PRECCINSTA burner case treated in the previous paper [1]. The numerical set-up including boundary conditions and domain size is detailed on section 5 of the previous paper [1]. The procedure starts from the same initial tetrahedral mesh, com-

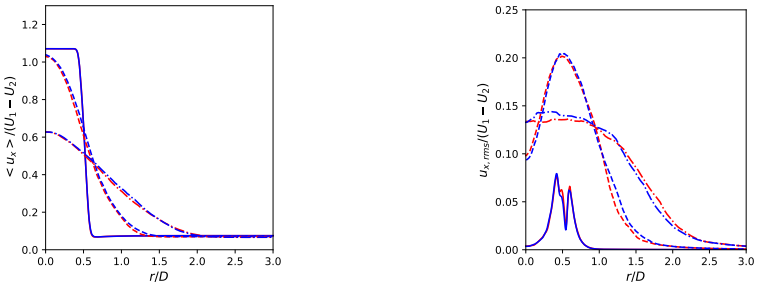


Figure 5. Comparison of turbulent round jet statistics: explicit (red), linearized implicit (blue). Mean axial velocity profile (left) and rms axial velocity profile (right) at three sections: $x/D = 1$ (solide line), 5 (dashed line), and 10 (dotted line).

Table 3. Comparison of time-to-solution given in hours for the explicit reference automatic mesh convergence (T_{exp}) and linearized implicit automatic time-step and mesh convergence (T_{imp}) for the PRECCINSTA burner case.

State	**0**	**1**	**2**	**3**	**4**	**Total**
T_{exp}	0.21	12.19	7.06	-	15.23	**34.69**
T_{imp}	0.14	6.62	3.51	2.93	10.78	**23.98**
T_{exp}/T_{imp}	1.50	1.84	2.01	-	1.41	**1.45**

posed of 446,661 elements, but the simulation is started with a constant CFL number of 10. The evolution of the main quantities characterizing the procedure is shown in Fig. 6. Based on the statistics of the initial mesh ($State = 0$), the sequential determination of the target values for $QC_{1,T}$ ($State = 1$), $QC_{2,T}$ ($State = 2$), and ϵ_T ($State = 3$) starts. The evolution of the quantities is similar to that of the reference case presented in the pevious paper. The final mesh is composed of 185,212,606 elements, which is close to the 178,586,310 elements of the reference case. The statistics prediction obtained by automatic time-step and mesh convergence is compared to the automatic mesh convergence, presented in the previous paper [1], and experimental results [23,24] in Fig. 7. There is a very good agreement for both mean and rms velocity profiles at the different stages between the two automatic procedures. Table 3 shows the comparison of time-to-solution for the different states. The improvement is less pronounced than in the case of the turbulent round jet (see Table 2), but high enough to compensate for the extra step (State 3). As a result, the whole procedure is 1.45 times faster.

5. Conclusion

In this paper, a linearized implicit time advancement is proposed in order to reduce the time-to-solution of LES of incompressible flows. The linearization is carried out using extrapolation of the velocity field in the prediction step of the projection method. This allows the use of larger time-steps than those constrained by the stability of explicit schemes. For a turbulent pipe flow, the use of linearized implicit time advancement allows to reduce the time-to-solution by a factor of 5 while keeping the same accuracy in terms of mean and fluctuating velocity fields. Nevertheless, it is difficult to know a priori the maximum time-step that allows to correctly capture the time scales involved. This is why an automatic convergence procedure based on temporal error control is proposed. This procedure is based on a time-step control strategy which

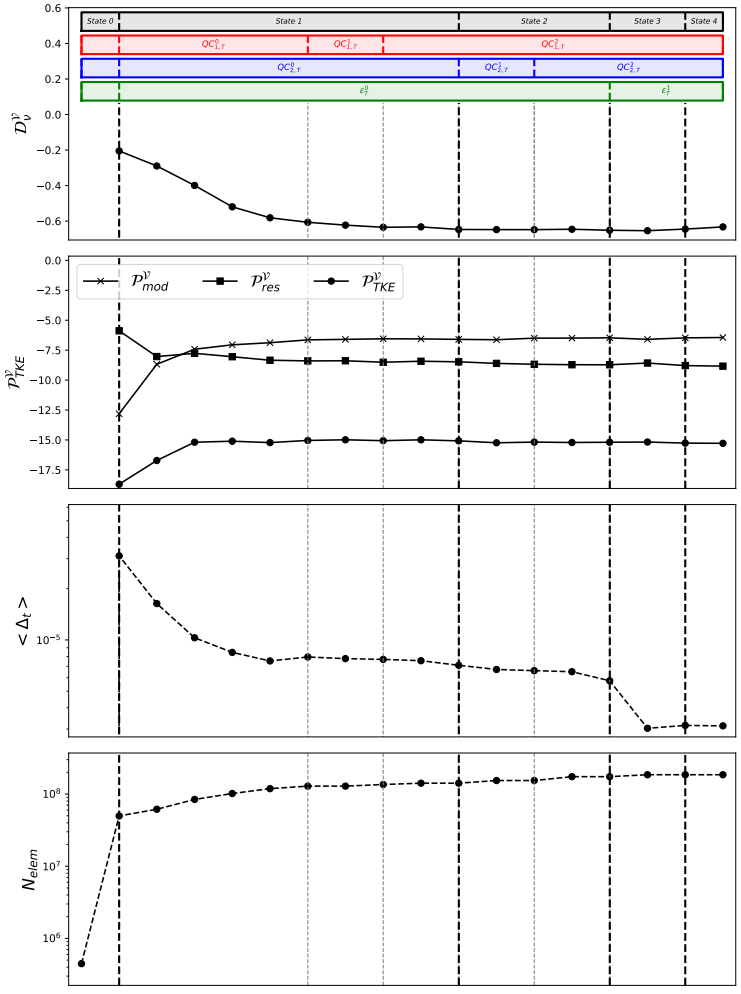


Figure 6. Evolution of the key parameters during the different states of the automatic time-step and mesh convergence procedure in the PRECCINSTA burner case. From top to bottom: the global molecular dissipation, $D_\nu^{\mathcal{V}}$, the global TKE production, $P_{\mathrm{TKE}}^{\mathcal{V}}$, the mean time-step $\langle \Delta t \rangle$, and the number of elements, N_{elem}. Note that the resolved, $P_{\mathrm{res}}^{\mathcal{V}}$, and the modeled $P_{\mathrm{mod}}^{\mathcal{V}}$, parts of the TKE production are also shown.

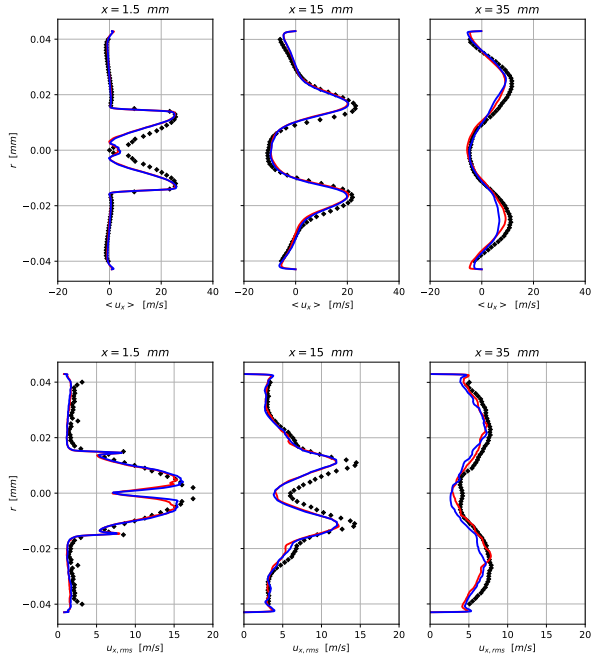


Figure 7. Mean axial velocity profiles (top) and rms axial velocity profiles (bottom) at three sections for the PRECCINSTA burner: experiment (black symbol), explicit reference case (red line), and linearized implicit case (blue line).

aims to determine at each iteration the time-step that keeps the local truncation time error at a certain level. Moreover, in order to be user-independent, this level of error needs to be determined automatically. This is why the automatic mesh convergence, proposed in the previous paper, is adapted. A criterion on the temporal error is then added to the criterion on the correct discretization of the mean field and to the criterion on the determination of the cut-off of the resolved scales. These three criteria are sequentially constrained until a mesh-independent mean kinetic energy field is obtained. The resulting automatic time-step and mesh procedure has been successfully applied to a turbulent round jet and to an industrial swirl burner, named PRECCINSTA. The use of the linearized implicit time advancement significantly reduces the time-to-solution of the whole procedure, despite the extra cost of adding the step of time-error level determination. The automatic time-step and mesh procedure is 3 times faster for the turbulent round jet and 1.45 times faster for the swirl burner than the previous procedure using explicit time integration. Future work will focus on the reduction of the time-to-solution of the approach by optimizing the time windows of the statistical accumulation, as well as by using anisotropic cells to reduce the number of mesh elements.

Acknowledgements

The authors gratefully acknowledge support from NETHUNS project under grant ANR-21-CHIN-0001-01 and from the Hydro'Like industrial chair. This work was granted access to the HPC resources of CINES/TGCC/IDRIS under the projects 2B06880 and 2A00611 made by GENCI. Part of this work has been initiated during

the Extreme CFD Workshop & Hackathon (https://ecfd.coria-cfd.fr).

References

[1] Grenouilloux A, Leparoux J, Moureau V, et al. Toward the use of LES for industrial complex geometries. Part I: automatic mesh definition. J Turbul. 2022;submitted.

[2] Chorin AJ. Numerical solution of the Navier–Stokes equations. Mathematics of computation. 1968;22(104):745–762.

[3] Temam R. Une méthode d'approximation de la solution des équations de Navier–Stokes. Bulletin de la Société Mathématique de France. 1968;96:115–152.

[4] Choi H, Moin P, Kim J. Direct Numerical Simulation of turbulent flow over riblets. J Fluid Mech. 1993;255:503–539.

[5] Choi H, Moin P. Effects of the computational time step on numerical solutions of turbulent flow. J Comp Phys. 1994;113(1):1–4.

[6] Kim D, Choi H. A second-order time-accurate finite volume method for unsteady incompressible flow on hybrid unstructured grids. J Comp Phys. 2000;162(2):411–428.

[7] Beam RM, Warming R. An implicit factored scheme for the compressible Navier–Stokes equations. AIAA journal. 1978;16(4):393–402.

[8] Simo J, Armero F. Unconditional stability and long-term behavior of transient algorithms for the incompressible Navier–Stokes and Euler equations. Comp Meth Appl Mech Eng. 1994;111(1-2):111–154.

[9] Olshanskii MA, Terekhov KM, Vassilevski YV. An octree-based solver for the incompressible Navier–Stokes equations with enhanced stability and low dissipation. Comp Fluids. 2013;84:231–246.

[10] Kay DA, Gresho PM, Griffiths DF, et al. Adaptive time-stepping for incompressible flow part ii: Navier–Stokes equations. SIAM Journal on Scientific Computing. 2010;32(1):111–128.

[11] John V, Rang J. Adaptive time step control for the incompressible Navier–Stokes equations. Comp Meth Appl Mech Eng. 2010;199(9-12):514–524.

[12] Hay A, Etienne S, Pelletier D, et al. hp-Adaptive time integration based on the BDF for viscous flows. J Comp Phys. 2015;291:151–176.

[13] Brayton RK, Gustavson FG, Hachtel GD. A new efficient algorithm for solving differential-algebraic systems using implicit backward differentiation formulas. Proceedings of the IEEE. 1972;60(1):98–108.

[14] Kalkote N, Assam A, Eswaran V. Acceleration of later convergence in a density-based solver using adaptive time stepping. AIAA Journal. 2019;57(1):352–364.

[15] Moureau V, Domingo P, Vervisch L. Design of a massively parallel CFD code for complex geometries. Comptes Rendus Mécanique. 2011;339(2):141–148.

[16] Malandain M, Maheu N, Moureau V. Optimization of the deflated conjugate gradient algorithm for the solving of elliptic equations on massively parallel machines. J Comp Phys. 2013;238:32–47.

[17] Dobrzynski C, Frey P. Anisotropic Delaunay mesh adaptation for unsteady simulations. Springer Berlin Heidelberg; 2008. p. 177–194.

[18] Benard P, Balarac G, Moureau V, et al. Mesh adaptation for Large-Eddy Simulations in complex geometries. International Journal for Numerical Methods in Fluids. 2016; 81(12):719–740.

[19] Germano M, Piomelli U, Moin P, et al. A dynamic subgrid-scale eddy viscosity model. Phys Fluids A: Fluid Dynamics (1989-1993). 1991;3(7):1760–1765.

[20] Kraushaar M. Application of the compressible and low-Mach number approaches to Large–Eddy Simulation of turbulent flows in aero-engines [dissertation]. Institut National Polytechnique de Toulouse-INPT; 2011.

[21] Sleijpen GLG, Fokkema DR. BiCGStab(ℓ) for linear equations involving unsymmetric matrices with complex spectrum. Electron Trans Numer Anal. 1993;1:11–32.

[22] Wu X, Moin P. A Direct Numerical Simulation study on the mean velocity characteristics in turbulent pipe flow. J Fluid Mech. 2008;608.

[23] Meier W, Weigand P, Duan X, et al. Detailed characterization of the dynamics of thermoacoustic pulsations in a lean premixed swirl flame. Combust Flame. 2007;150(1):2–26.

[24] Lartigue G, Meier U, Bérat C. Experimental and numerical investigation of self-excited combustion oscillations in a scaled gas turbine combustor. Applied Thermal Engineering. 2004;24(11):1583–1592. Industrial Gas Turbine Technologies.

Appendix A. Assessment of theoretical errors

The objective of this appendix is to illustrate the behavior of the linearized implicit scheme in an idealized case of pure convection. First, the impact of linearization on the temporal convergence orders of the BDF is evaluated. Then, the benefit of the adaptive time step is demonstrated. The test case corresponds to the convection of a signal on a 2D periodic domain of size $L \times L$,

$$\frac{\partial \phi}{\partial t} + u(t)\nabla\phi = 0, \quad \phi(0) = e^{\frac{1}{r_0^2}} e^{\frac{1}{r-r_0}}. \tag{A1}$$

where r represents the distance to the center of the domain and $r_0 = 0.4L$. The convective velocity u follows a quadratic law in time such that the signal returns to its initial position after a period of T,

$$u(t) = 3\left(\frac{t}{T}\right)^2. \tag{A2}$$

Using the linearized implicit time advancement, the convection equation becomes,

$$\sum_{k=0}^{p} \alpha_k \phi^{n+1-k} + \tilde{u}^{n+1}\nabla\phi = 0, \tag{A3}$$

where p represents the BDF order and $\tilde{u}^{n+1}$ the extrapolation of the convective velocity performed at order l, Eq. (6).

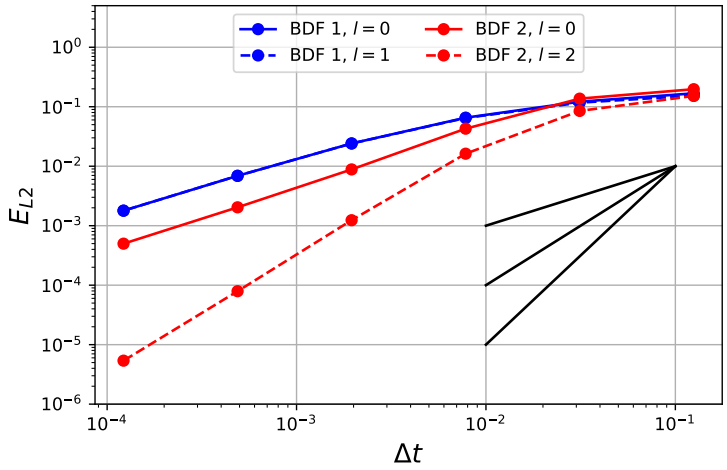


Figure A1. Errors evolution in function of constant time-step. Different combination of BDF order and extrapolation order of the convective velocity (l) are illustrated. Black lines correspond to convergence order 1, 2 and 3.

The equation is solved using an uniform regular triangular mesh composed by 512×512 grid points, to minimize the spatial error. The error is then evaluated with the L_2 norm between the initial scalar field $\phi(0)$ and the final scalar field obtained after one period $\phi(T)$. The evolution of this error as a function of a (constant) time-step size is illustrated on Fig. A1. Concerning the BDF scheme at order 1, noted BDF 1, the

both values of extrapolation order $l = 0$ and $l = 1$ lead to same theoretical order of convergence of 1. The benefit of the extrapolation is clearly shown for the BDF scheme at order 2, noted BDF 2. Indeed, the theoretical order of the scheme is recovered only when the second order extrapolation $l = 2$ is used. With an extrapolation at order 0, $l = 0$, an order of convergence close to 1 is obtained.

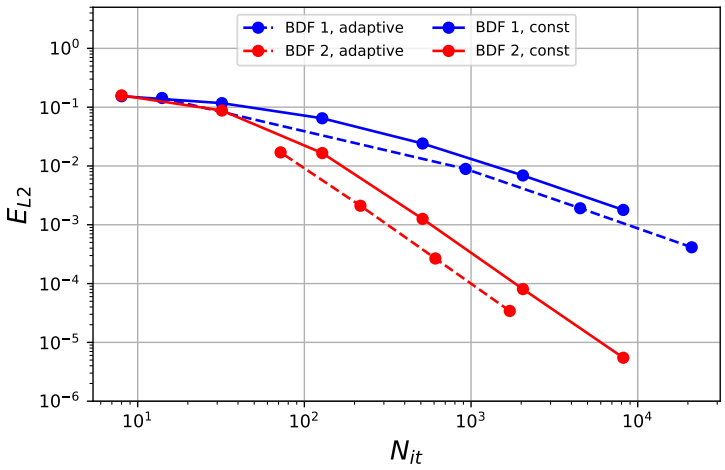


Figure A2. Errors evolution in function of number of iteration. The adaptive time-step is compared to constant time-step in case of BDF 1 ($l = 1$) and BDF 2 ($l = 2$).

The adaptive time-step is now evaluated for BDF schemes at order 1 and 2 with four values of ϵ_T, the targeted value of local truncation error presented in section 4.1. The resulting errors are illustrated on Fig. A2 as a function of the total number of iterations N_{it}, which is an indicator of the computational cost. The adaptive time-step is compared to the constant time-step, already presented on Fig. A1. It can be noticed that the adaptive time-step maintains the theoretical convergence order of the BDF schemes. Moreover, it can also be noticed that to achieve the same level of error, less iterations are needed for the adaptative time-step cases in comparison with the constant time-step cases for both BDF schemes, showing the benefit of the adaptive time-step approach. For example, the number of iterations is roughly divided by 2 for the second order BDF scheme.