COB-2021-0686
IMPLEMENTATION OF THE STEPWISE IMPLICIT-EXPLICIT TIME-STEP METHOD ON A DNS CODE

Giovanni Belloni Fernandes Braga
Marcello Augusto Faraco de Medeiros
University of São Paulo - USP
gbelloni@usp.br
marcello@sc.usp.br

Sávio Brochini Rodrigues
Federal University of São Carlos - UFSCar
savio.rodrigues@ufscar.br

Abstract. *This study is dedicated to the implementation, verification via the method of manufactured solutions, and measurement of execution speed of a new temporal integration method of the implicit-explicit kind, the Stepwise-IMEX (SIMEX), in the Aeroacoustics, Transition and Turbulence Group's (GATT) Direct Numerical Simulation (DNS) code. The SIMEX method can extend the maximum stable step size when compared with explicit integration schemes because explicit methods are severely constrained by the CFL condition especially at low Mach number, $Ma \approx 0.1$ or less. The SIMEX method can attain a larger step-size by decomposing the compressible Navier-Stokes equation in two parts, one implicit and one explicit, so that only the stiff terms are integrated implicitly. This method's aim is to reduce the execution time demanded by the explicit 4^{th}order Runge-Kutta (RK4) which is currently being used by the research group. SIMEX was implementated on the DNS code and three test cases were designed by varying the boundary conditions, then different ESDIRK-ERK tableaus pairs ranging from 2^{nd} to 5^{th} order were tested. The iterative method employed for linear systems was the GMRES and for non-linear systems was the Jacobian-free Newton-Krylov method. Both the choice of tableau pair and the presence, or not, of boundary conditions, as well as the decomposition, showed an enormous influence on SIMEX's capability to extend the maximum step-size. Regarding the computational efficiency, SIMEX allows one to explore the trade off between the maximum stable step-size and the amount of iterations per step; thus, allowing for an optimal balance between the computational effort placed in each step and the increase in step-size.*

Keywords: *IMEX, SIMEX, Runge-Kutta, Temporal integration, Navier-Stokes.*

1. INTRODUCTION

The current group's (Turbulence, Transition and Aeroacoustic Group - GATT) Direct Numerical Simulation (DNS) code is a reorganization of the previous versions, (Mathias, 2017). Details about its use and validation can be found at (Silva *et al.*, 2010; Bergamo, 2014; Bergamo *et al.*, 2015; Martinez and Medeiros, 2016; Martinez, 2016; Mathias, 2017). In this code the acoustic phenomena are always resolved which allows aeroacoustic analysis for every subsonic Mach number. At low Mach numbers, say $0 < Ma \approx 0.1$, the velocity fields are much slower then the acoustic waves, which in pratice makes the system stiff, thus imposing time-step restrictions for the simulations with explicit integration methods.

Fully implicit methods build up complexity and CPU cost proportionally to the integrated equation's complexity. So it is quite clear that Navier-Stokes equations are not very well suited for these kind of methods at first glance. But, because different terms contribute distinctly to the stiffness, implicit-explicit methods can offer some advantage. This class of methods decomposes the equation in two parts, the part with the stiff terms are integrated implicitly and the remaining part is integrated explicitly. Bigger stable time-steps may reduce execution time as consequence of this approach. The trade-off to be managed is the necessity of keeping the implicit equation simple to solve against the possibility of using bigger time-steps.

The use of implicit and explicit methods in an ODE system dates back to the 1970s. Early studies are (Hofer, 1976) and (Cooper and Sayfy, 1980). A comprehensive review can be found in (Christopher A. Kennedy, 2001; Kennedy and Carpenter, 2003). IMEX methods have been studied with a variety of time-stepping schemes. Here we use singly-diagonally implicit Runge-Kutta with explicit first stage (ESDIRK) schemes (Ascher *et al.*, 1997; Christopher A. Kennedy, 2018; Wang *et al.*, 2015). A comprehensive review of ESDIRK schemes can be found in (Christopher A. Kennedy, 2016).

Recently, IMEX methods have been used in compressible fluids simulations such as in atmospheric studies, where the stiffness is associated with terms that generates acoustic waves (Bispen *et al.*, 2017; Zhang *et al.*, 2016b; Colavolpe *et al.*, 2017; David J. Gardner *et al.*, 2018; Ghosh and Constantinescu, 2016; Restelli and Giraldo, 2009; Christopher J. Vogl *et al.*, 2019; Weller *et al.*, 2013), which imposes strong time-step restrictions and are not relevant for climate phenomena for example.

Within this scenario, SIMEX is a modification of the IMEX method. The advantage in this new class of methods is the absence of the need to iterate the implicit equation until a really low residue level. The residue that remains from the implicit equation is integrated along with the explicit part of the decomposition, thus keeping the method's precision. Due to the possibility of using less iterations, SIMEX has the potential to obtain precise solutions, as in IMEX, but faster (Rodrigues, 2017).

This paper will focus on the work's cut where the SIMEX method SIMEX is finally implemented on the GATT's DNS code, which is mainly written in FORTRAN 90. Now the system of implicit equations is solved using Krylov methods, GRMES for the linear systems and Newton-Krylov for the non-linear ones. In this stage it's necessary to accommodate the writting to the already existing structure of the code, which may impose some limitations since there's no intention to make profound changes in the code.

2. METHODOLOGY

2.1 Navier-Stokes Equantions and Thermodynamics Relations

A more detailed explanation on how the DNS code works can be found in (Mathias, 2017). The compressible Navier-Stokes equations are used to model the flow, along with mass and energy conservation equations. A compressible flow can be entirely defined by five variables: density (ρ), internal energy (e) and the three velocity components (u,v,w), in this case. Each variable depends on location (x,y,z) and time (t).

Equations are written in the non-conservative form, which solves for each individual variable mentioned above, simplifying the boundary conditions.

$$\frac{\partial \rho}{\partial t} = -\rho \frac{\partial u_i}{\partial x_i} - \frac{\partial \rho}{\partial x_i} u_i, \tag{1}$$

$$\frac{\partial u_j}{\partial t} = -\frac{\partial u_j}{\partial x_i} u_i - \frac{1}{\rho}\frac{\partial p}{\partial x_j} + \frac{1}{\rho}\frac{\partial \tau_{ij}}{\partial x_i}, \tag{2}$$

$$\frac{\partial e}{\partial t} = -\frac{\partial e}{\partial x_i} u_i - \frac{p}{\rho}\frac{\partial u_i}{\partial x_i} + \frac{1}{\rho}\tau_{ij}\frac{\partial u_j}{\partial x_i} - \frac{1}{\rho}\frac{\partial q_i}{\partial x_i}. \tag{3}$$

Where the viscous tensor and the heat flux term are:

$$\tau_{ij} = \frac{\mu(T)}{Re}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right) \tag{4}$$

and,

$$q_i = -\frac{\mu}{(\gamma-1)\, Re\, Pr\, M_\infty^2}\frac{\partial T}{\partial x_i} \tag{5}$$

Temperature and pressure, assuming an ideal gas, are given by:

$$T = e\gamma\,(\gamma-1)\,M_\infty^2, \quad p = (\gamma-1)\,\rho\, e \tag{6}$$

Viscosity is modeled by Sutherland's law:

$$\frac{\mu^*}{\mu_\infty} = \mu(T) = \frac{1+C}{T+C}T^{\frac{3}{2}} \tag{7}$$

Defining $C = \frac{110K}{T_\infty^*}$ and $T_\infty^* = 300K$.

All values are non-dimensional, being normalized by the free flow speed, cavity depth and initial density. Reynolds, Prandtl and Mach numbers are given by:

$$Re = \frac{\rho_\infty^*\, U_\infty^*\, L_0}{\mu_\infty^*}, \quad Pr = \frac{\mu_\infty^* c_p^*}{k^*}, \quad M_\infty = \frac{U_\infty^*}{\sqrt{\gamma\frac{p_\infty^*}{\rho_\infty^*}}} \tag{8}$$

Dimensional (denoted with a *) and non-dimensional values are related by the following equations.

$$\rho = \frac{\rho^*}{\rho_\infty^*}, \quad p = \frac{p^*}{p_\infty^* U_\infty^{*2}}, \quad x_i = \frac{x_i^*}{L_0}, \quad T = \frac{T^*}{T_\infty^*}, \quad u_i = \frac{u_i^*}{U_\infty^*}, \quad t = \frac{t^* U_\infty^*}{L_0}, \quad e = \frac{e^*}{U_\infty^{*2}} \tag{9}$$

2.2 Compact Finite Differences

Spatial derivatives are calculated with a spectral-like compact finite differences method described by (Lele, 1992), where centered stencil is defined as follows:

$$\beta(u'_{i+2} + u'_{i-2}) + \alpha(u'_{i+1} + u'_{i-1}) + u'_i = a\frac{u_{i+1} - u_{i-1}}{2h} + b\frac{u_{i+2} - u_{i-2}}{4h} + c\frac{u_{i+3} - u_{i-3}}{6h}, \tag{10}$$

where h is the mesh spacing. The correspondent modified wavenumber w' is

$$\omega'(\omega) = \frac{a\sin(\omega) + (b/2)\sin(2\omega) + (c/3)\sin(3\omega)}{1 + 2\alpha\cos(\omega) + 2\beta\cos(2\omega)}. \tag{11}$$

Close to the boundaries, the stencil is shifted and coefficients are adjusted accordingly, as an example, the first derivative at the boundary can be calculated from the following relation:

$$u'_1 + \alpha u'_2 = \frac{1}{h}(au_1 + bu_2 + cu_3 + du_4) \tag{12}$$

This system may be represented as

$$Au' = Bu, \tag{13}$$

where A and B are $N \times N$ square matrices with N being the number of grid points. The entries of B are proportional to h^{-1}.

Its worth mentioning that matrices A and B are narrow band-diagonal matrices. By choosing $\beta = c = 0$ in Eq. 10 it is possible to obtain up to a sixth-order tridiagonal scheme whose coefficients are determined by matching Taylor series. This work uses a fourth-order tridiagonal scheme with $w = 1.8$. When necessary, we will formally write

$$u' = A^{-1}Bu, \tag{14}$$

although Eq. 13 is used for computational purposes.

2.3 ESDIRK-IMEX

IMEX is a class of methods that combines an explicit with an implicit integrator and to do so it decomposes the right-hand-side of the ODE as the sum of two functions

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(t, \mathbf{y}) + \mathbf{g}(t, \mathbf{y}), \tag{15}$$

where f is the explicit part of the decomposition and g the implicit part. This allows stiff terms to be handled implicitly while non-stiff terms are placed in the explicit part. For IMEX methods, it is imperative to solve the implicit equation up to a given precision level in order to avoid introducing errors. This task is done by an iterative method that will iterate until de precision requirement is fulfilled or the maximum iterations number is reached, here we call this a Solver.

Explicit-first-stage singly diagonally implicit Runge-Kutta (ESDIRK) is a particular case of the DIRK class of implicit methods, for which an extensive review can be found in (Christopher A. Kennedy, 2016). The ESDIRK-IMEX implicit equation for the ith stage has the following form

$$\tilde{\mathbf{y}}_i - h\gamma\mathbf{g}(t_n + c_i h,\ \tilde{\mathbf{y}}_i) = \mathbf{y}_n + h\sum_{j=1}^{i-1}(a_{i,j}\mathbf{k}_j + \tilde{a}_{i,j}\tilde{\mathbf{k}}_j), \qquad i = 2, \ldots, \nu. \tag{16}$$

Where $\mathbf{y}_n$ is the current approximation for $\mathbf{y}(t_n)$, h is the step size and γ is the *tableau* diagonal's value. $\tilde{\mathbf{y}}_i$ is the current stage approximation, which is obtained by a Solver S within some predefined residue tolerance. The implicit equation above can be rewritten in a way it is less affected bay rounding errors,

$$\eta_i - h\gamma\left(\mathbf{g}(t_n + c_i h,\ \tilde{\mathbf{y}}_i) - \mathbf{g}(t_n, \mathbf{y}_n)\right) = h\sum_{j=1}^{i-1}(a_{i,j}\mathbf{k}_j + \tilde{a}_{i,j}\tilde{\mathbf{k}}_j) + \mathbf{g}(t_n, \mathbf{y}_n), \qquad i = 2, \ldots, \nu. \tag{17}$$

Where $\eta = \tilde{\mathbf{y}}_i - \mathbf{y}_n$. Finally, the next step approximation $\mathbf{y}_{n+1}$ is computed as follows:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\sum_{j=1}^{\nu} b_j(\mathbf{k}_j + \tilde{\mathbf{k}}_j). \tag{18}$$

2.4 IMEX Decompositions

The choice of the EDO decomposition é cruscial for the implict-explicit method's efficiency, always seeking a better relation between how much is possible to extend the time-step and how the resulting implicit system complexity grows.

Having in mind that the advective term is the stiffness source to this problem with Burgers' equation, it should now be clear that this is the one to be placed in the implicit part of the decomposition. To avoid solving a non-linear system of equations, a linearisation of the advective term can be made.

When dealing with the DNS code, the propagation of acoustic waves is what gives stiffness to the problem, specially in the y direction, considering that the y-axis is around 1 order of magnitude more refined than the x-axis. So, the challenge lies on identifying which of theses terms are responsible for that propagation.

That said, decomposition 1 ("L" - Linear) is proposed, which has in its implicit part $\mathbf{g_1}$ a linear version of the chosen terms. It is possible to show that a wave equation can be derived from these terms and its velocity matches the adiabatic sound speed. Decomposition 2 ("NL" - Non-Linear), with $\mathbf{g_2}$, groups the terms of $\mathbf{g_1}$ in its original non-linear forms.

To avoid ambiguities regarding variables representation, inside one integration step, define:

$$\mathbf{y} = \mathbf{y_0} + \eta = \begin{pmatrix} \rho_0 \\ U_0 \\ V_0 \\ E_0 \end{pmatrix} + \begin{pmatrix} \phi(y,t) \\ u(y,t) \\ v(y,t) \\ e(y,t). \end{pmatrix} \tag{19}$$

So, the decompositions are as follows:

$$\mathbf{g_1}(\mathbf{y}) = \begin{pmatrix} -\rho_0 V_y \\ 0 \\ -\frac{\gamma-1}{\rho_0}(\rho_0 E_y + \rho_y E_0) \\ -\frac{p_0}{\rho_0} V_y \end{pmatrix} \quad \mathbf{g_2}(\mathbf{y}) = \begin{pmatrix} -\rho V_y \\ 0 \\ -\frac{\gamma-1}{\rho}(\rho E_y + E\rho_y) \\ -\frac{p}{\rho} V_y \end{pmatrix} \tag{20}$$

The explicit part for each decomposition can be easily obtained through the relation below:

$$\mathbf{f_i}(\mathbf{y}) = \frac{\mathbf{dy}}{\mathbf{dt}} - \mathbf{g_i}(\mathbf{y}), \qquad \mathbf{i} = \mathbf{1} \text{ and } \mathbf{2}. \tag{21}$$

Onde

$$\mathbf{y} = \begin{pmatrix} \rho(x,y,t) \\ U(x,y,t) \\ V(x,y,t) \\ E(x,y,t) \end{pmatrix}. \tag{22}$$

2.5 ESDIRK-Stepwise-IMEX

SIMEX (Rodrigues, 2017) is a new implicit-explicit integration method that can be seen as an improved IMEX, where a within-step decomposition adjustment is proposed with the intent of removing the need of getting precise solutions for the implicit equations. This new decomposition is named Residual Balanced Decomposition (RBD) and it is directly related to the traditional IMEX decomposition, Eq. 23. The job once done by a Solver is now done by a *implicit step filter*, or *Filter* for short, which is defined as a predefined successive iterations of one or more iterative methods. For example, one can define a Filter as 2 Newton iterations. Thus, solvers and filters are distinguished because neither precision nor convergence are required from a filter.

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(t,\mathbf{y}) + \mathbf{g}(t,\mathbf{y}) = \mathbf{f}^{rbd}(t,\mathbf{y}) + \mathbf{g}^{rbd}(t,\mathbf{y}). \tag{23}$$

The implicit equation that has to be solved at each SIMEX implicit step is exactly the same as the one for IMEX, Eq. 16. After that, the remaining residue has to be managed properly and that's what motivates the following definition for the RBD decomposition:

$$\mathbf{g}^{rbd}(t,\mathbf{y}) = \frac{\mathbf{y} - \mathbf{y}_n - \mathcal{F}^{-1}(\mathbf{y} - \mathbf{y}_n, t)}{h\gamma}. \tag{24}$$

Where $\mathcal{F}^{-1}$ is the inverse filter $\mathcal{F}$ and,

$$\mathbf{f}^{rbd}(t,\mathbf{y}) = \mathbf{f}(t,\mathbf{y}) + \mathbf{g}(t,\mathbf{y}) - \mathbf{g}^{rbd}(t,\mathbf{y}). \tag{25}$$

Now the question that arises is how to calculate the inverse filter, but even when possible it is certainly not a quick job. Altough its existence being important for the theoretical justification of the method, fortunately, there's a practical way to evaluate $\mathbf{g}^{rbd}$ and $\mathbf{f}^{rbd}$ without the need of computing $\mathcal{F}^{-1}$. So, let the value returned by the filter, Eq. 26, be the exact answer for the implicit equation, Eq. 16, when the function $\mathbf{g}$ is replaced by $\mathbf{g}^{rbd}$, Eq. 27. Mathematically,

$$\tilde{\mathbf{y}}_{i,rbd} = \mathcal{F}(\mathbf{rhs}, t) \tag{26}$$

such that

$$\tilde{\mathbf{y}}_{i,rbd} - h\gamma\mathbf{g}^{rbd}(t_n + c_i h,\ \tilde{\mathbf{y}}_{i,rbd}) = \mathbf{rhs}, \tag{27}$$

where $\mathbf{rhs} = \mathbf{y}_n + h\sum_{j=1}^{i-1}(a_{i,j}\mathbf{k}_j + \tilde{a}_{i,j}\tilde{\mathbf{k}}_j)$. Hence, solving for $\mathbf{g}^{rbd}$,

$$\mathbf{g}^{rbd}(t_n + c_i h,\ \tilde{\mathbf{y}}_{i,rbd}) = \frac{\tilde{\mathbf{y}}_{i,rbd} - \mathbf{rhs}}{h\gamma}. \tag{28}$$

And now $\mathbf{f}^{rbd}$ is completely determined by Eq. 25. After the calculations for the RBD decomposition the only thing missing is the variable update which is done exactly in the same way as in IMEX, Eq. 18.

The ESDIRK-SIMEX algorithm for a single step of size h starting from $t = t_n$ with step approximation $\mathbf{y}_n$ is presented bellow:

Algorithm 1 ESDIRK-SIMEX algorithm

```
$\mathbf{k}_1 = \mathbf{g}(t_n, \mathbf{y}_n)$
$\tilde{\mathbf{k}}_1 = \mathbf{f}(t_n, \mathbf{y}_n)$
for $i = 2;\quad i <= \nu;\quad i++$ do
    $\mathbf{rhs} = \mathbf{y}_n + h\sum_{j=1}^{i-1}(a_{i,j}\mathbf{k}_j + \tilde{a}_{i,j}\tilde{\mathbf{k}}_j)$
    $\tilde{\mathbf{y}}_{i,rbd} = \mathcal{F}$
    $\mathbf{k}_i = (\ \tilde{\mathbf{y}}_{i,rbd} - \mathbf{rhs})/h\gamma$
    $\tilde{\mathbf{k}}_i = \mathbf{f}(t_n + c_i h,\ \tilde{\mathbf{y}}_{i,rbd}) + \mathbf{g}(t_n + c_i h,\ \tilde{\mathbf{y}}_{i,rbd}) - \mathbf{k}_i$
end for
$\mathbf{y}_{n+1} = \mathbf{y}_n + h\sum_{j=1}^{\nu} b_j(\mathbf{k}_j + \tilde{\mathbf{k}}_j)$
```

An important but not so obvious observation that has to be made is about how to interpret this stepwise adjustment of the decomposition. Taking the filter's answer, Eq. 26, and putting it inside Eq. 16, it yields the following result:

$$\mathbf{g}(t_n + c_i h,\ \tilde{\mathbf{y}}_{i,rbd}) = \frac{\tilde{\mathbf{y}}_{i,rbd} - \mathbf{rhs} + \mathbf{res}}{h\gamma}, \tag{29}$$

Where $\mathbf{res}$ is the implicit equation residue. So,

$$\mathbf{g} - \mathbf{g}^{rbd} = \frac{\tilde{\mathbf{y}}_{i,rbd} - \mathbf{rhs} + \mathbf{res}}{h\gamma} - \frac{\tilde{\mathbf{y}}_{i,rbd} - \mathbf{rhs}}{h\gamma} = \mathbf{res}/h\gamma. \tag{30}$$

And Hence,

$$\mathbf{f}^{rbd} = \mathbf{f} + \mathbf{res}/h\gamma. \tag{31}$$

Therefore it is possible to say that the implicit equations residue is integrated along with the explicit part.

2.6 Iterative Methods for Implicit Equations

Iterative methods for implicit equations plays a big role in fully, or partially, implicit integration methods by affecting directly its computational efficiency. Thus, equating robustness and memory and processing costs is fundamental.

For the DNS tests, we use Krylov methods which are matrix-free. When using linear decompositions, the resulting implicit system is linear and here it is solved with Generalized Minimal Residual (GMRES) Method. An extensive review about projection and Arnoldi Methods can be found in chapters 5 and 6 of SAAD's book, (Saad, 2003). So, starting from 17 and applying $\mathbf{g_1}$, it yelds

$$\begin{pmatrix} I & \theta[\rho_0]A^{-1}B & 0 \\ \theta(\gamma-1)\left[\frac{E_0}{\rho_0}\right]A^{-1}B & I & \theta(\gamma-1)A^{-1}B \\ 0 & \theta\left[\frac{p_0}{\rho_0}\right]A^{-1}B & I \end{pmatrix} \begin{pmatrix} \phi \\ v \\ e \end{pmatrix} = \mathbf{r}. \tag{32}$$

And at last, but not least, non-linear decompositions as $\mathbf{g_2}$ requires different strategies and here we use Newton-Krylov method. The implicit equation can be condensed in

$$\mathbf{G}(\eta) = \mathbf{r}. \tag{33}$$

where $\mathbf{r} =$. And the Newton step is than defined as

$$\mathbf{J}(\eta_\mathbf{0})\mathbf{d} = \mathbf{r} - \mathbf{G}(\eta_\mathbf{0}). \tag{34}$$

To solve this linear system for the Newton step a Krylov method is applied, where a first order approximation is made for the matrix-vector product $\mathbf{J}(\eta_\mathbf{0})\mathbf{v}$.

2.7 Butcher's Tableau for the used Runge-Kuttas

In this work, some tests are performed with explicit and implicit-explicit Runge-Kutta methods. The explicit one that is took as a reference of precision and execution time is the classical 4^{th} order 4 stages ERK, also called as "RK4". The *tableau* representation of RK4 is given in Eq. 35. Five ESDIRK-SIMEX pairs are used and they're listed bellow.

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6} \end{array} \tag{35}$$

The pair "HCN" is a combination of Heun and Crank-Nicolson methods, which has 2 stages and a formal 2^{nd} order convergence rate. Also with 2^{nd} order, but 3 stages, there is the pair "ARK32", proposed by (Ascher *et al.*, 1997) under the name "ARS222", using the choice of parameters $\gamma = 1 - \sqrt{2}/2$ and $\delta = 1 - 1/2\gamma$. The last three, which are proposed by (Christopher A. Kennedy, 2001) under the names "ARK3(2)4L[2]SA-(ERK and ESDIRK)", "ARK4(3)6L[2]SA-(ERK and ESDIRK)" and "ARK5(4)8L[2]SA-(ERK and ESDIRK)", pairs are respectively: "ARK423", 3^{rd} order 3 stages; "ARK634", 4^{th} order 6 stages and "ARK845", 5^{th} order 8 stages. The following table summarises theses informations.

Table 1. Summary of the *tableaus* characteristics.

Tableau	Steps	Order	Order of Err Ctrl
RK4	4	4	–
ARK32	3	2	–
ARKHCN	2	2	–
ARK423	4	3	2
ARK634	6	4	3
ARK845	8	5	4

3. Tested Cases

Three cases were created to investigate the impact of the presence, or not, of boundary conditions on SIMEX CFL stability, they are: "Boundary Layer"; "Free Slip" and "Periodic Y". The first one refers to the formation of boundary layer in a flat plate, the second is somewhat like removing the plate, the fluid can slip freely everywhere. The last one refers to a periodic domain in the y direction. All of them has the same mesh in the x direction which is uniform, differing only in the y direction. More specificaly, the "Boundary Layer" case requires a refinement near the plate to accurately capture the boundary layer. The following table contains some additional relevant information about these cases.

Table 2. Additional information about the tested cases.

Caso	x_i;x_f	n_x	y_i;y_f	n_y	$\Delta x_{min}/\Delta y_{min}$	Initial Flow
Boundary Layer	-10;300	392	0;10	212	100.03	Blasius
Free Slip	-10;300	392	0;10	147	12.54	Blasius
Periodic Y	-10;300	392	0;10	127	12.54	Uniform

3.1 Fluid Properties

Fluid properties are the same for every case and they are:

$$\begin{cases} Re &= 1000 \\ Pr &= 0.71 \\ \gamma &= 1.4 \\ Ma &= 0.1 \end{cases}, \tag{36}$$

where Re, Pr and Ma are Reynolds, Prandtl and Mach numbers respectively and γ is the thermodynamics relation c_p/c_v.

4. DNS RESULTS

4.1 Maximum Stable CFL

In this section tables containing data about maximum stable CFL from each ERK and ESDIRK-SIMEX *tableau* pair listed in section 2.7and from each case are presented. Table 3 refers to decomposition 1, where a linear version of the acoustic terms is used, and Table 4 refers to Decomposition 2. Here the maximum CFL is defined as the highest CFL number that is possible to run the simulation without some kind of floating point error or with a variable change between steps above 10^{-1}. The uncertainty of these CFL numbers is smaller then 0.1. The reference method, RK4, has a maximun stable CFL of 1.3. A great number of theses runs were executed for a suficiently long time to be considered “infinite”, therefore some of them did become steady and others didn’t, but they’re all numerically stable. Since baseflows calculations and method’s precision is beyond the scope of this work, it will not be addressed in any further detail.

In Table 3 it is possible to see that SIMEX has allowed a bigger stable time-step if compared with its respective ERK. But, in a closer look, the results were highly influenced by the boundary conditions and *tableau* pair used and were far from the expected. Considering only the explicit integrators, the results were in some degree consistent with their formal orders of convergence, except for the ARK845 which were supposed to deliver the biggest CFL number, since it has the highest order.

For the SIMEX integrationm, starting from the first colunm, which referres to the Boundary Layer case, the CFL number was allowed to increase by a factor of 7 using ARS222 and effectively nothing when using ARK423. When Mach number is raised from 0.1 to 0.5, nothing really changes. The other case involving Dirichlet and Neumann conditions is the Free Slip case, and the results were slightly better. One interesting fact is that the same results from the Free Slip case can be obtained from the Boundary Layer case by adding a mesh refinement in the x direction centered on the plate’s leading edge.

Table 3. Maximum stable CFL for ESDIRK-SIMEX and its associated ERK for each tested case - Decomposition 1.

Method	Maximum CFL – ERK/SIMEX – Decomposition 1			
	BoundLay	BoundLay Ma05	FreeSlip	Periodic Y
ARK32	0.3/2.1	0.3/2.0	0.3/2.3	0.2/3.2
ARKHCN	0.3/1.2	0.4/1.1	0.3/1.4	0.2/1.5
ARK423	1.2/1.3	1.2/1.3	1.2/1.4	1.2/8.6
ARK634	1.8/2.4	1.8/2.3	1.8/2.4	1.9/5.3
ARK845	1.0/2.9	1.1/2.7	0.9/3.1	0.9/8.1

In the last colunm of Table 3 are presented the results from the last tested case, Periodic Y, which consists of imposing periodic boundary conditions in the y direction. This last round of runs showed that the presence, or not, of boundary conditions plays a big role on SIMEX stability. Every single *tableau* pair delivered a much more significant CFL number extension.

Table 4 presents the results for Decomposition 2, using the same structure as in the previous table. The results with this non-linear decomposition are pretty much the same except for a slightly greatter CFL number for SIMEX-ARK634 in the periodic case and small decrease on the SIMEX-ARK845 CFL number. This last one may be explained by an insufficient amount of Newton-Krylov steps. Considering that this *tableau* has 8 stages, CPU time required by the implicit stages increases rapidly.

4.2 CPU Time

In this section, a table containing the results from execution time measurements for the “Periodic Y” case is presented. Only a subset of the *tableau* pairs were selected to be in this round os testing and the selection was guided by the greatter CFL per stage achieved and decomposition used. As could be seen in the previous section, the linear decomposition

Table 4. Maximum stable CFL for ESDIRK-SIMEX and its associated ERK for each tested case - Decomposition 2.

Method	Maximum CFL – ERK/SIMEX – Decomposition 2			
	BoundLay	BoundLay Ma05	FreeSlip	Periodic Y
ARK32	0.3/2.1	0.4/2.0	0.3/2.3	0.2/3.2
ARKHCN	0.3/1.1	0.4/1.1	0.3/1.2	0.2/1.5
ARK423	1.2/1.5	1.2/1.5	1.2/1.6	1.2/9.1
ARK634	1.8/2.4	1.8/2.2	1.9/2.2	1.9/8.0
ARK845	1.0/2.2	1.1/1.7	0.9/2.1	0.9/7.8

was as capable of extending the maximun CFL as the non-linear one. Hence, it is much more efficient to use the linear decomposition.

The execution time is measured by means of a MATLAB's internal function and the reading does not account the preprocessing step and FORTRAN compiling time. The number of domain decompositions was set to 10, which is sufficiently large to confer a 10 times speed up and still represent less than 10% of the machine capacity. The reference time is, of course, set by the RK4 with maximun CFL and it's also shown in this table the execution time for the integration solely with the explicit *tableau* of each pair. Lastly, the non-dimensional simulation time was chosen to 300.

Table 5. CPU time comparison for ESDIRK-SIMEX, its associated ERK and RK4 for the periodic case varing maximum CFL as a function of number of iterations.

				Execution time in seconds		
Case	Method	Iter.	CFL simex	RK4	ERK	SIMEX
Periodic Y	ARK423	1×14	8.6	350.3	451.0	229.5
		1×10	6.6			185.4
		1×8	5.5			176.7
		1×4	3.2			209.7
	ARK32	1×14	3.2		1674.3	472.2
		1×10	3.1			423.4
		1×4	3.0			340.3
		1×2	1.7			432.0
	ARK845	1×12	8.0		969.2	449.1
		1×10	6.6			451.9
		1×8	5.5			373.3
		1×4	3.1			635.0

The first thing to take note is that SIMEX integration greattly reduced the execution time if compared with the integration with the ERK from the *tableau* pair. Second, ARK423 was faster than RK4 in every scenario and ARK32 got one combination of implicit stage iterations and respective maximun CFL that yield a faster execution. As could be seen, more effort placed in the implicit stage usually means more CFL stabillity, but this relation isn't linear. And, in addition to this, the smallest CPU time required appears to be always somewhere in the middle.

5. CONCLUSIONS

In the end, SIMEX was successfully implemented on the research group's DNS code and new ESDIRK-ERK *tableau* pairs can be easely added to the library. The CFL extension obtained with this implicit integration of the acoustic terms were substantial only in the periodic case. Each *tableau* pair also had a very particular behavior, which shows the importance of picking the right one for the application. Another important variable in SIMEX usabillity is the choice of decomposition and here it was clear that the linear decomposition was as good as the non-linear one and faster due to not having to solve a non-linear system of equations. In the second round of testing it could be seen that SIMEX yielded a smaller CPU time than RK4 in some scenarios, depending on the *tableau* and the combination of iterations and respective maximun CFL.

6. ACKNOWLEDGEMENTS

The authors would like to thank the US Air Force Office of Scientific Research (AFOSR) for grant FA9550-18-1-0112, managed by Dr. Geoff Andersen from SOARD. M.A.F.M. is sponsored by CNPq/Brazil (grant no. 307956/2019-9). G.B.F.B. is sponsored by CAPES/Brazil (grant no. 88882.379193/2019-01).

7. REFERENCES

Ascher, U.M., Ruuth, S.J. and Spiteri, R.J., 1997. "Implicit-explicit runge-kutta methods for time-dependent partial differential equations". *Applied Numerical Mathematics*, Vol. 25, No. 2, pp. 151 – 167. ISSN 0168-9274. doi:https://doi.org/10.1016/S0168-9274(97)00056-1. URL http://www.sciencedirect.com/science/article/pii/S0168927497000561. Special Issue on Time Integration.

Bergamo, L.F., Gennaro, E.M., Theofilis, V. and Medeiros, M.A., 2015. "Compressible modes in a square lid-driven cavity". *Aerospace Science and Technology*, Vol. 44, pp. 125–134. ISSN 12709638. doi:10.1016/j.ast.2015.03.010. URL https://linkinghub.elsevier.com/retrieve/pii/S1270963815001030.

Bergamo, L.F., 2014. *Instabilidade hidrodinâmica linear do escoamento compressível em uma cavidade*. Master thesis, Universidade de São Paulo, São Carlos. doi:10.11606/D.18.2014.tde-28052014-164324. URL http://www.teses.usp.br/teses/disponiveis/18/18148/tde-28052014-164324/.

Bispen, G., Lukáčová-Medvid'ová, M. and Yelash, L., 2017. "Asymptotic preserving imex finite volume schemes for low mach number euler equations with gravitation". *Journal of Computational Physics*, Vol. 335, pp. 222 – 248. ISSN 0021-9991. doi:https://doi.org/10.1016/j.jcp.2017.01.020. URL http://www.sciencedirect.com/science/article/pii/S002199911730030X.

Christopher A. Kennedy, M.H.C., 2001. "Additive Runge-Kutta Schemes for Convection-Diffusion-Reaction Equations". Technical report, National Aeronautics And Space Administration, Virginia. URL https://ntrs.nasa.gov/search.jsp?R=20010075154 2019-09-05T22:35:36+00:00Z.

Christopher A. Kennedy, M.H.C., 2016. "Diagonally Implicit Runge-Kutta Methods for Ordinary Differential Equations. A Review". *National Aeronautics And Space Administration*. URL https://ntrs.nasa.gov/search.jsp?R=20160005923 2020-03-22T21:22:55+00:00Z.

Christopher A. Kennedy, M.H.C., 2018. "Higher-order additive Runge–Kutta schemes for ordinary differential equations". *Applied Numerical Mathematics*, p. 183–205. URL https://doi.org/10.1016/j.apnum.2018.10.007.

Christopher J. Vogl, A.S., Reynolds, D.R., Ullrich, P.A. and Woodward, C.S., 2019. "Evaluation of Implicit-Explicit Additive Runge-Kutta Integrators for the HOMME-NH Dynamical Core". *Journal of Advances in Modeling Earth Systems (JAMES)*.

Colavolpe, C., Voitus, F. and Bénard, P., 2017. "Rk-imex hevi schemes for fully compressible atmospheric models with advection: analyses and numerical testing". *Quarterly Journal of the Royal Meteorological Society*, Vol. 143, No. 704, pp. 1336–1350. doi:10.1002/qj.3008. URL https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.3008.

Cooper, G. and Sayfy, A., 1980. "Additive methods for the numerical solution of ordinary differential equations". *Mathematics of Computation*, Vol. 35, No. 152, pp. 1159–1172.

Crouzeix, M., 1980. "Une méthode multipas implicite-explicite pour l'approximation des équations d'évolution paraboliques". *Numerische Mathematik*, Vol. 35, No. 3, pp. 257–276.

David J. Gardner, J.E.G., Hamon, F.P., Reynolds, D.R., Ullrich, P.A. and Woodward, C.S., 2018. "Implicit–explicit (IMEX) Runge–Kutta methods for non-hydrostatic atmospheric models". *Copernicus Publications on behalf of the European Geosciences Union*. URL https://doi.org/10.5194/gmd-11-1497-2018.

Durran, D.R. and Blossey, P.N., 2012. "Implicit–explicit multistep methods for fast-wave–slow-wave problems". *Monthly Weather Review*, Vol. 140, No. 4, pp. 1307–1325.

Ghosh, D. and Constantinescu, E.M., 2016. "Semi-implicit time integration of atmospheric flows with characteristic-based flux partitioning". *SIAM Journal on Scientific Computing*, Vol. 38, No. 3, pp. A1848–A1875. doi: 10.1137/15M1044369. URL https://doi.org/10.1137/15M1044369.

Hofer, E., 1976. "A partially implicit method for large stiff systems of odes with only few equations introducing small time-constants". *SIAM Journal on Numerical Analysis*, Vol. 13, No. 5, pp. 645–663. doi:10.1137/0713054. URL https://doi.org/10.1137/0713054.

Kennedy, C.A. and Carpenter, M.H., 2003. "Additive runge–kutta schemes for convection–diffusion–reaction equations". *Applied Numerical Mathematics*, Vol. 44, No. 1, pp. 139 – 181. ISSN 0168-9274. doi:https://doi.org/10.1016/S0168-9274(02)00138-1. URL http://www.sciencedirect.com/science/article/pii/S0168927402001381.

Lele, S.K., 1992. "Compact finite difference schemes with spectral-like resolution". *Journal of Computational Physics*, Vol. 103, No. 1, pp. 16–42. ISSN 00219991. doi:10.1016/0021-9991(92)90324-R.

Martinez, A., 2016. *Towards natural transition in compressible boundary layers*. Phd. thesis, University of São Paulo.

Martinez, A. and Medeiros, M.F., 2016. "Direct numerical simulation of a wavepacket in a boundary layer at Mach 0.9". In *46th AIAA Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, Reston, Virginia, Vol. 414, pp. 1–33. ISBN 978-1-62410-436-7. ISSN 00221120. doi:10.2514/6.2016-3195. URL http://arc.aiaa.org/doi/10.2514/6.2016-3195.

Mathias, M.S., 2017. *Instability analysis of compressible flows over open cavities by a Jacobian-free numerical method.*

Master thesis, University of São Paulo.

Restelli, M. and Giraldo, F.X., 2009. "A conservative discontinuous galerkin semi-implicit formulation for the navier–stokes equations in nonhydrostatic mesoscale modeling". *SIAM Journal on Scientific Computing*, Vol. 31, No. 3, pp. 2231–2257.

Rodrigues, S.B., 2017. "A shortcut for IMEX methods: integrate the residual explicitly". *eprint arXiv*, Vol. 1705.04870. URL `http://arxiv.org/abs/1705.04870`.

Saad, Y., 2003. *Iterative methods for sparse linear systems*. SIAM.

Silva, H.G., Souza, L.F. and Medeiros, M.A.F., 2010. "Verification of a mixed high-order accurate dns code for laminar turbulent transition by the method of manufactured solutions". *International Journal for Numerical Methods in Fluids*, Vol. 64, No. 3, pp. 336–354. doi:10.1002/fld.2156. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.2156`.

Wang, H., Shu, C.W. and Zhang, Q., 2015. "Stability and error estimates of local discontinuous galerkin methods with implicit-explicit time-marching for advection-diffusion problems". *SIAM Journal on Numerical Analysis*, Vol. 53, No. 1, pp. 206–227. doi:10.1137/140956750. URL `https://doi.org/10.1137/140956750`.

Weller, H., Lock, S.J. and Wood, N., 2013. "Runge–kutta imex schemes for the horizontally explicit/vertically implicit (hevi) solution of wave equations". *Journal of Computational Physics*, Vol. 252, pp. 365 – 381. ISSN 0021-9991. doi:https://doi.org/10.1016/j.jcp.2013.06.025. URL `http://www.sciencedirect.com/science/article/pii/S0021999113004555`.

Zhang, H., Sandu, A. and Blaise, S., 2016a. "High order implicit-explicit general linear methods with optimized stability regions". *SIAM Journal on Scientific Computing*, Vol. 38, No. 3, pp. A1430–A1453.

Zhang, H., Sandu, A. and Blaise, S., 2016b. "High order implicit-explicit general linear methods with optimized stability regions". *SIAM Journal on Scientific Computing*, Vol. 38, No. 3, pp. A1430–A1453. doi:10.1137/15M1018897. URL `https://doi.org/10.1137/15M1018897`.

8. RESPONSIBILITY NOTICE

The following text, properly adapted to the number of authors, must be included in the last section of the paper: The author(s) is (are) solely responsible for the printed material included in this paper.