XLI CILAMCE
XLI Ibero-Latin American Congress on Computational Methods in Engineering
16-19 November, 2020 FOZ DO IGUAÇU PARANÁ - BRAZIL

# On the accuracy of time-stepping methods for flow stability analysis

Marlon Sproesser Mathias[1], Marcello A. F. Medeiros[1]

[1]*São Carlos School of Engineering, University of São Paulo*
*Av. João Dagnone, 1100, São Carlos - SP, 13563-120*
*marlon.mathias@usp.br, marcello@sc.usp.br*

**Abstract.** Computing the linear stability modes of a flow requires obtaining the eigenmodes of the Jacobian matrix of its governing equations. For bi-global and tri-global stability analysis, this matrix is very large, which poses a challenge for solving its eigenproblem. Time-stepping methods are one of the possible solutions for this problem, as they do not require explicitly building this matrix. One advantage of this class of methods is their relative ease of implementation, as it can treat the flow solver as a black box, which is integrated into the rest of the algorithm. The routine calls the flow solver multiple times with different initial conditions to observe the flow behavior. We study the trade-off between two of the parameters that must be chosen when setting up a case: The length of time each call to the flow solver will last and the total number of iterations. Longer call times allow the algorithm to converge in fewer iterations but each one is more costly. The available literature indicates that for a constant product of call time by number of calls, the accuracy should be roughly the same. We seek both upper and lower boundaries for the call time length as well as an optimum value that will generate the most accurate results for a given computational cost.

**Keywords:** Global flow stability, Jacobian-free methods, Frèchet derivative

## 1 Introduction

Methods for computing the global stability of flows have been used for several applications in the past decades [1]. Many of the implementations involve building the Jacobian matrix of the flow, commonly as a sparse matrix, and solving its eigenproblem [2]. Starting in the 1980s and 1990s, Jacobian-free time-stepping methods were developed [3–5], those methods are usually based on the Arnoldi [6] algorithm for solving the eigenproblem, which builds a Krylov vector space from the input matrix and approximates the most unstable eigenmodes from there.

The Jacobian-free approach relies on the fact that computing eigenvalues and eigenvectors via the Arnoldi algorithm does not require the matrix explicitly, instead, it only needs the ability to multiply such matrix by an arbitrary vector. The idea behind this method is to arrange the equations in a such way that the result of this multiplication can be obtained by a call to the flow solver. This method has proven itself to be very flexible and allows the flow solver to be treated as a black box, which can dramatically reduce its implementation effort [7] and make use of an already validated flow solver. In summary, the algorithm adds a series of small disturbances to a base flow and observes their temporal evolution for a short period of time.

In this work, we analyze the parameters which influence the accuracy of the resulting modes of this method and seek to establish some guidelines for obtaining the best result for a given amount of computational time. Naturally, the specific parameters one should use when setting up an analysis will vary according to their case and computational framework, but we seek to understand the effect of each variable in the overall performance of the method. We study the trade-off between two parameters: The length of time each call to the flow solver will last and the total number of Arnoldi iterations. Longer call times improve the separation between different modes and allow the algorithm to converge in fewer iterations but each one is more costly. On the other hand, more iterations allow more modes to be accurately solved, but also increase the total cost. The available literature indicates that accuracy is dictated by the product of number of calls times the length of each call [8].

## 2 Methods

The global stability routine used in this work is based on the work of Eriksson and Rizzi [3], Chiba [4]. It is of the time-stepping type and does not build the Jacobian matrix of the governing flow equations.

The temporal evolution of a flow can be described by

$$\frac{\partial U}{\partial t} = f(U), \tag{1}$$

where $f$ represents the Navier-Stokes equations and all boundary conditions. $U$ is the current state of the flow, which is given by a sum of a base flow with a disturbance: $U = U_0 + \tilde{u}$.

The function $f$ can be linearized around the base flow $U_0$, hence

$$\frac{\partial \tilde{u}}{\partial t} = A\tilde{u} \tag{2}$$

By solving the eigenproblem of $A$, one can have an insight on the linear stability of the flow. Each eigenvector describes the shape of a mode, and its associated eigenvalue describes its stability and temporal frequency. However, $A$ is a very large matrix, of dimension $N \times N$, where $N$ is the total number of variables in the system, given by the amount of nodes in the mesh times the number of variables in each node.

The formal solution of eq. (2) is

$$\tilde{u} = e^{tA}\tilde{u}_0, \tag{3}$$

where $\tilde{u}_0$ is the initial disturbance given to the flow. In this method, we replace $e^{tA}$ by a matrix $B$, and the initial disturbance $\tilde{u}_0$ by a normalized vector $\zeta$ multiplied by a small scalar $\epsilon$ so that $\tilde{u}_0 = \epsilon\zeta$. To simplify the notation, we define

$$F(U) = \int_0^\tau f(U)dt, \tag{4}$$

$\tau$ is a predefined length of time. The value of $F(U)$ can be approximated by calling a flow solver with initial condition $U$ and simulation the flow for a time $\tau$.

The Arnoldi iteration method is then used to compute the eigenvalues and eigenvectors of $B$, which can be easily related to those of $A$. The multiplication of the matrix $B$ by an arbitrary vector $\zeta$, can be approximated by

$$B\zeta = \frac{1}{\epsilon}\left[F\left(U_0 + \epsilon\zeta\right) - F\left(U_0\right)\right] + O(\epsilon) \tag{5}$$

with accuracy. Or by

$$B\zeta = \frac{1}{2\epsilon}\left[F\left(U_0 + \epsilon\zeta\right) - F\left(U_0 - \epsilon\zeta\right)\right] + O(\epsilon^2) \tag{6}$$

with second-order accuracy. And by

$$B\zeta = \frac{1}{12\epsilon}\left[-F\left(U_0 + 2\epsilon\zeta\right) + 8F\left(U_0 + \epsilon\zeta\right) - 8F\left(U_0 - \epsilon\zeta\right) + F\left(U_0 - 2\epsilon\zeta\right)\right] + O(\epsilon^4) \tag{7}$$

with fourth-order accuracy. These equations are known as the Frèchet derivative of $F$ with respect to $\zeta$. Most of the literature has chosen the second-order variant of the approximation [4, 5, 7].

In this paper, we compare the accuracy of the first and second-order approximations, as well as the choice of $\tau$. Our goal is to understand the effect of choosing each parameter on both computational cost and accuracy.

Mode details on the method, the implementation and the validation can be found in [9], which also brings information on the DNS used as flow solver for this work.

### 2.1 Computational cost

The computational cost of this algorithm is largely due to the use of the flow solver, which scales linearly with $n\tau M$, where $n$ is the order of the Frèchet derivative chosen, which dictates how many calls to the flow solver will happen at each iteration of the Arnoldi algorithm. $\tau$ is the length of each call, as per eq. (4), and $M$ is the total number of Arnoldi iterations.

Nevertheless, other types of costs must be considered, most notably the overhead of starting up the flow solver at each call and the cost of the Gram-Schmidt orthonormalization step in the Arnoldi algorithm. This overhead adds a constant cost at each call of the solver, while the Gram-Schmidt orthonormalization has its cost increase proportional to $M^2$, therefore it might become the leading cost for large values of $M$.

Another cost that scales superlinearly with $M$ is that of solving the eigenproblem of the reduced order matrix created by the Arnoldi method, which is an $M \times M$ matrix. However, despite this cost scaling roughly with $M^3$, for the range of values used, it is still comparatively small.

### 2.2 Reference case

The test scenario is a two-dimensional compressible open cavity with Reynold number of $Re_D = 1000$, based on cavity depth; free-flow Mach number of $Ma = 0.5$; boundary layer momentum thickness of $\theta/D = 1/100$ at the cavity leading edge; the cavity aspect ratio is $L/D = 2$. This flow is known to have three unstable modes, namely Rossiter modes 2, 3 and 1, in order of decreasing instability.

Figure 1 illustrates the flow and its main parameters. There is a recirculating flow inside the cavity and a mixing layer between it and the outer flow. In this case, the flow is unstable, which causes an acoustic emission when the disturbances in the mixing layer reach the trailing edge of the cavity. Due to the instability of the flow, it is impossible to generate a base flow for stability analysis by simply running the simulation for long enough, as these instabilities drive the flow away of its equilibrium point and towards a limit cycle. A Selective Frequency Damping (SFD) filter is used in the flow solver to damp these modes and allow a steady base flow to be reached [10]. This filter is turned off for the stability analysis. The base flow residual is in the order of $10^{-13}$ of the free-flow, which is caused by the numerical rounding error of the computer.

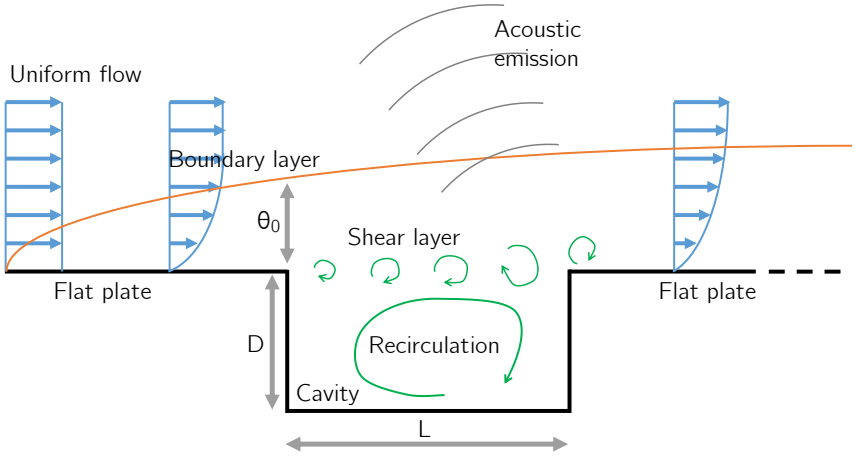


Figure 1. Illustration of the open cavity flow.

Figure 2 shows the reference results for our stability analysis. The left hand side shows each eigenvalue in the complex plane, positive real parts indicate unstable modes. Rossiter modes 1 to 3 (R1 to R3) are highlighted, as well as mode S1, which is the least stable stationary mode. The right hand side shows contours of density fluctuation for those modes at arbitrary phases.

## 3 Results

The results shown in sec. 2.2 and in fig. 2 were obtained by using the fourth-order accurate Frèchet derivative, eq. (7), and an integration time of $\tau = 1$. The total number of Arnoldi iterations was $M = 2500$, enough for the

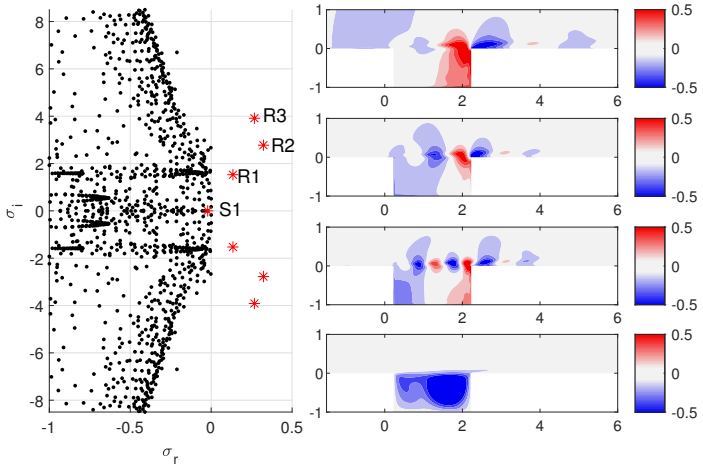


Figure 2. (left) Complex plane with all the modes retrieved by the algorithm. (right) Isocontours of density for modes R1 to R3 and S1.

residual to be limited by the computer rounding error. In all cases shown here, the disturbance magnitude was given by $\epsilon = 5 \times 10^{-4}$.

We compare these reference results to those we have obtained with various values of $\tau$ and $M$, for first and second-order accurate Frèchet derivatives. Modes R2 and S1 will be taken as reference as they are, respectively, the most unstable mode and the least stable stationary mode.

Figures 3 and 4 bring contours of the accuracy obtained by each combination of parameters. The left hand side plots are for accurate Frèchet and the right hand side, for second-order accurate. In the white areas for small $\tau$ or small $M$, the algorithm did not find the sought mode, while the white areas for large $\tau$ and $M$ were excluded from our parametric space as the convergence had already stalled at smaller values of $M$. The solid light gray lines are isocontours of computational time in our system. From left to right, they represent 5 minutes, 10 minutes, 30 minutes and 1 hour of wall time, excluding compiling and start-up time.

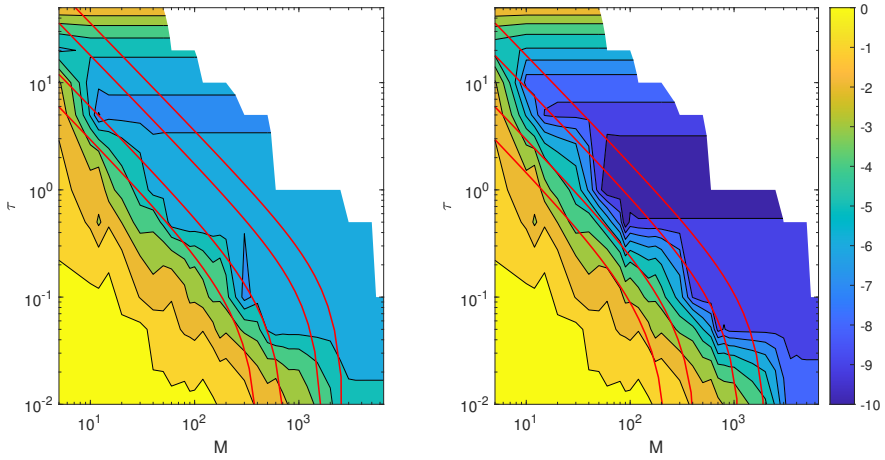


Figure 3. Isocontours of $\log_{10}$(Error) of the real part of the R2 mode. The red lines represent a constant computational cost of 5 minutes, 10 minutes, 30 minutes and 1 hour. (left) First order accurate Frèchet derivative. (right) Second order accurate Frèchet derivative.

The bottom right quadrant of both first and second-order accuracy plots are very similar, therefore for a set of $\tau$ and $M$ values, both types of approximation yield similar accuracies, which means that in this region of the plot, the overall accuracy is not dictated by the Frèchet derivative's truncation error, but by other factors, such as the product of $\tau M$[8].

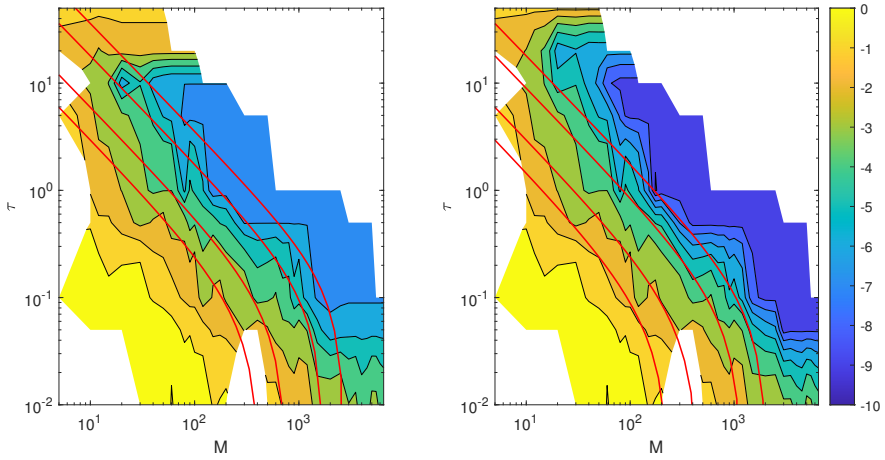


Figure 4. Isocontours of $\log_{10}$(Error) of the real part of the S1 mode. The red lines represent a constant computational cost of 5 minutes, 10 minutes, 30 minutes and 1 hour. (left) First order accurate Frèchet derivative. (right) Second order accurate Frèchet derivative.

However, as the product $\tau M$ becomes larger and the accuracy improves, the second-order approximation is able to reach a lower plateau, which is likely dictated by the truncation errors in the Frèchet derivative.

On the other hand, despite the fact that, before their plateaus, both methods reach similar accuracies for the same combination of $\tau$ and $M$, the accurate method is able to do so with a considerably lower computational cost, as it only requires a single call to the flow solver per Arnoldi iteration.

Another feature that can be observed in the figures is the minimum and maximum values of $\tau$ that still produce accurate results and that, away from these boundaries, the final accuracy is roughly constant for a constant product $\tau M$, but the plateaus of accuracy may differ.

The computational cost lines are mostly parallel to the constant product $\tau M$, but they become more parallel to the vertical ($\tau$) axis at its lower range. This is caused by the superlinear cost of the Gram-Schmidt orthonormalization step of the routine becoming the leading cost, which is independent of $\tau$ and scales with $M^2$ as discussed in sec. 2.1.

## 4 Conclusions

The majority of the literature available on the time-stepping method for computing the global stability of a flow uses a second-order approximation for the Frèchet derivative present in the algorithm. We have analyzed this choice and concluded that higher accuracies in this derivative do result in better accuracy, however this is only noticeable if enough Arnoldi iterations are taken. A approximation for this derivative is much less costly and may still produce accurate results. Therefore, the optimal choice of the derivative approximation order depends on the final accuracy desired for the results.

In our case, if up to five decimal places are required, the approximation is able to produce the same results in just over half of the computational time. If more decimal places are needed, then higher accuracy order approximations are needed for the Frèchet derivative.

The choice of $\tau$ for the method should be such that it is not long enough as to trigger non-linear effects nor short enough as to be affected by rounding errors when evaluating eq. (4).

**Acknowledgements.** The authors would like to thank the São Paulo Research Foundation (FAPESP/Brazil), for grants 2018/04584-0 and 2017/23622-8; the National Council for Scientific and Technological Development (CNPq/Brazil) for grants 134722/2016-7 and 307956/2019-9; the US Air Force Office of Scientific Research (AFOSR) for grant FA9550-18-1-0112, managed by Dr. Geoff Andersen from SOARD; the University of Liverpool for the access to the Barkla cluster, provided by Prof. Vassilios Theofilis; and the Center for Mathematical Sciences Applied to Industry (CeMEAI) funded by São Paulo Research Foundation (FAPESP/Brazil), grant #2013/07375-0, for access to the Euler cluster, provided by Prof. José Alberto Cuminato.

**Authorship statement.** The authors hereby confirm that they are the sole liable persons responsible for the au-

thorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

## References

[1] Theofilis, V., 2011. Global Linear Instability. *Annual Review of Fluid Mechanics*, vol. 43, n. 1, pp. 319–352.
[2] Gennaro, E. M., Rodríguez, D., Medeiros, M. A., & Theofilis, V., 2013. Sparse techniques in global flow instability with application to compressible leading-edge flow. *AIAA Journal*, vol. 51, n. 9, pp. 2295–2303.
[3] Eriksson, L. E. & Rizzi, A., 1985. Computer-aided analysis of the convergence to steady state of discrete approximations to the euler equations. *Journal of Computational Physics*, vol. 57, n. 1, pp. 90–128.
[4] Chiba, S., 1998. Global Stability Analysis of Incompressible Viscous Flow. *Journal of Japan Society of Computational Fluid Dynamics*, vol. 7, n. 1, pp. 20–48.
[5] Tezuka, A. & Suzuki, K., 2006. Three-dimensional global linear stability analysis of flow around a spheroid. *AIAA journal*, vol. 44, n. 8, pp. 1697–1708.
[6] Arnoldi, W. E., 1951. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, vol. 9, n. 1, pp. 17–29.
[7] Gómez, F., Gómez, R., & Theofilis, V. V., 2014. On three-dimensional global linear instability analysis of flows with standard aerodynamics codes. *Aerospace Science and Technology*, vol. 32, n. 1, pp. 223–234.
[8] Goldhirsch, L., Orszag, S. A., & Maulik, B. K., 1987. An efficient method for computing leading eigenvalues and eigenvectors of large asymmetric matrices. *Journal of Scientific Computing*, vol. 2, n. 1, pp. 33–58.
[9] Mathias, M. S. & Medeiros, M., 2018. Direct Numerical Simulation of a Compressible Flow and Matrix-Free Analysis of its Instabilities over an Open Cavity. *Journal of Aerospace Technology and Management*, vol. 10, pp. 1–13.
[10] Åkervik, E., Brandt, L., Henningson, D. S., Hœpffner, J., Marxen, O., & Schlatter, P., 2006. Steady solutions of the Navier-Stokes equations by selective frequency damping. *Physics of Fluids*, vol. 18, n. 6, pp. 68–102.