



## **Convergence Acceleration of the Harmonic Balance Method using a Time-Level Preconditioner**

Downloaded from: <https://research.chalmers.se>, 2026-04-05 19:50 UTC

Citation for the original published paper (version of record):

Lindblad, D., Andersson, N. (2020). Convergence Acceleration of the Harmonic Balance Method using a Time-Level Preconditioner. AIAA Scitech 2020 Forum, 3.  
<http://dx.doi.org/10.2514/6.2020-0563>

N.B. When citing this work, cite the original published paper.

# Convergence Acceleration of the Harmonic Balance Method using a Time-Level Preconditioner

Daniel Lindblad <sup>\*</sup>, and Niklas Andersson <sup>†</sup>  
Chalmers University of Technology, Gothenburg, SE-412 96, Sweden

The Harmonic Balance method is nowadays widely applied for numerically solving problems that are known to possess time-periodic solutions. Key reasons for its success are its wide range of applicability, relative ease of implementation, and computational efficiency compared to time-accurate approaches. The computational efficiency of the Harmonic Balance method is partly derived from the fact that it searches directly for a periodic solution, instead of integrating the governing equations in time until a periodic solution is reached. Convergence acceleration techniques such as multigrid, implicit residual smoothing and local time stepping may also be used to improve the efficiency of the Harmonic Balance method. This paper considers another option for accelerating convergence, namely a novel time-level preconditioner that can be applied to the Harmonic Balance residual locally in each computational cell. This preconditioner is derived from a rigorous stability analysis of the Harmonic Balance equations and is shown to give a speed-up factor of 2 when applied to simulations of laminar vortex shedding behind a circular cylinder.

## I. Nomenclature

Roman/Greek

$c$	= speed of sound, $\text{m s}^{-1}$
$C_L, C_D$	= lift, drag coefficient
$D$	= time spectral derivative matrix, $\text{s}^{-1}$
$E_K(\mathbf{t}), E_K^{-1}(\mathbf{t})$	= discrete Fourier transform matrix pair
$F$	= discrete flux operator
$\mathcal{F}$	= vector containing convective flux of each conserved variable
$i$	= imaginary unit
$k$	= axial wavenumber, $\text{rad m}^{-1}$
$K$	= number of frequencies resolved in Harmonic Balance method
$N$	= number of axial wavenumbers resolved on mesh
$N_{\text{var}}$	= number of conserved variables
$M$	= number of time instances used in Harmonic Balance method
$\mathbf{P}$	= time-level preconditioning matrix
$Q$	= vector containing the conserved variables
$\mathcal{R}$	= Residual operator, $\text{s}^{-1}$
$S$	= number of Runge-Kutta stages
$t$	= physical time, $\text{s}$
$\Delta t$	= physical time step, $\text{s}$
$\mathbf{t}$	= vector containing time instances used in Harmonic Balance method, $\text{s}$
$T$	= time period, $\text{s}$
$\mathcal{T}, \mathcal{T}^{-1}$	= matrices containing right, left eigenvectors to flux Jacobian
$u$	= velocity, $\text{m s}^{-1}$
$U_\infty$	= freestream velocity, $\text{m s}^{-1}$
$w$	= characteristic variable
$\mathcal{W}$	= vector containing the characteristic variables
$x$	= axial coordinate, $\text{m}$

<sup>\*</sup>Ph.D. Student, Department of Mechanics and Maritime Sciences, Hörsalsvägen 7A, SE-412 96 Gothenburg.

<sup>†</sup>Associate Professor, Department of Mechanics and Maritime Sciences, Hörsalsvägen 7A, SE-412 96 Gothenburg.

$\Delta x$	=	length of computational cell, m
$\alpha$	=	Runge-Kutta coefficient
$\beta$	=	Runge-Kutta coefficient
$\lambda$	=	eigenvalue to flux Jacobian, m s <sup>-1</sup>
$\Lambda$	=	matrix containing eigenvalues to flux Jacobian along its diagonal, m s <sup>-1</sup>
$\sigma$	=	Courant–Friedrichs–Lewy number
$\tau$	=	pseudo-time, s
$\Delta\tau$	=	pseudo-time step, s
$\phi$	=	nondimensional axial wavenumber
$\omega$	=	frequency, rad s <sup>-1</sup>
$\Omega_K$	=	matrix containing frequencies resolved in Harmonic Balance method along its diagonal, rad s <sup>-1</sup>
$\gamma$	=	constant in approximate scaling function

### *Subscripts/Superscripts*

$\square_j$	=	cell index
$\square_k$	=	frequency index
$\square_l$	=	physical time index
$\square_m$	=	pseudo-time index
$\square_n$	=	spatial wavenumber index
$\square_p$	=	characteristic variable index
$\overline{\square}$	=	spatial+temporal average
$\hat{\square}$	=	temporal Fourier transform
$\hat{\hat{\square}}$	=	spatial + temporal Fourier transform

## II. Introduction

NUMERICAL simulations are today important tools in several engineering fields. In particular, the field of turbomachinery relies heavily on using numerical methods to analyze and design robust and high performance machines. A characteristic feature of turbomachines is that fluid flows inside them are inherently unsteady and time-periodic. This feature has promoted a lot of research within the turbomachinery community on numerical methods that are tailored for solving time periodic problems, both on the fluid and the structural side [1, 2].

Early research on time-periodic flows inside turbomachines focused on solving a linearized set of equations. The advantage of this approach is that it decouples each frequency in the solution and that it often is a good model of the flow. One of the first approaches that incorporated nonlinear terms into the governing equations was the Nonlinear Harmonic method [3]. Later, the Harmonic Balance (HB) method was introduced in the Computational Fluid Dynamics (CFD) community by Hall et al. [4]. Hall et al. formulated their HB method in the time domain. An equivalent formulation in the frequency domain is the Nonlinear Frequency Domain (NLFD) method developed by McMullen [5]. In both these methods, the solution is assumed to take on the form of a truncated Fourier series in time, with spatially varying coefficients. This allows the time derivative to be calculated exactly and then be included in the governing equations as a source term. This effectively transforms the problem from finding the time evolution of the governing equations, into a steady state problem for finding the Fourier coefficients. An appealing feature of the HB and NLFD methods is that they can be readily incorporated into an existing CFD code. In addition to this, the HB/NLFD methods can account for nonlinear interaction among the resolved frequencies. Since their introduction into the CFD community, these methods have therefore been adopted by many researchers and have proven to be both accurate and computationally efficient [6–10].

Early applications of the HB method showed that it can give rise to numerical instabilities when a large number of frequencies is included in the simulation and the equations are solved using explicit pseudo-time integration [4]. McMullen et al. [11] analyzed the stability of the NLFD method and showed that the time derivative source term effectively adds an imaginary part to the eigenvalue spectrum of the spatial discretization. They further concluded that this will lead to stability issues if the time derivative source term is large enough to push the eigenvalue spectrum outside the limited stability region of an explicit pseudo-time integration scheme. These findings are indeed consistent with the stability issues reported by Hall et al. [4]. McMullen et al. [11] initially proposed to overcome the stability issue

associated with the time derivative source term by treating it in a point implicit manner inside an explicit Runge-Kutta scheme. Later, it was shown by the same authors that this approach rotates the stability region of the Runge-Kutta scheme in the complex plane such that it no longer covers parts of the imaginary axis [5, 12]. From this, it was concluded that the point implicit treatment of the time derivative source term was unsuitable, at least when a central discretization of convection terms is used since the eigenvalue spectrum then is purely imaginary. Other authors have however reported that the stability of the HB method can be improved with a point implicit treatment of the time derivative source term [9]. This suggests that the stability of the point implicit treatment depends on which pseudo-time integration and spatial discretization scheme it is applied to.

Instead of applying the point implicit treatment, McMullen [5] scales the pseudo-time step used in his NLFD solver so that each frequency component of the solution is updated with the largest possible pseudo-time step allowed by his explicit Runge-Kutta solver. Later, van der Weide et al. [13] derived a similar restriction on the pseudo-time step. They also noted that it is necessary to perform a matrix-vector multiplication in each cell and iteration when this restriction is implemented in the time domain HB method. In order to avoid this overhead, van der Weide et al. [13] proposed to use the smallest pseudo-time step, corresponding to the highest frequency, when the time domain HB method is employed. Similar pseudo-time step restrictions have since then been adopted by several other authors in their respective time domain HB implementations [10, 14, 15]. An extensive stability analysis of the HB method has also been performed by Gentilli [16], who showed that the stability of an explicit HB solver depends both on the classical CFL number, as well as an additional parameter referred to as the grid reduced frequency. In his work, Gentilli also derives exact pseudo-time step restrictions for a four stage Runge-Kutta solver and a set of spatial discretization schemes. These restrictions can in fact be seen as tailored versions of the pseudo-time step restriction developed earlier by van der Weide et al. [13].

The restrictions on the pseudo-time step associated with explicit schemes can in principle be removed by employing an A-stable pseudo-time integration scheme, e.g. backward Euler, to integrate the equations. When this approach is followed for the HB method, the Jacobian matrix will be  $M$  times larger than for a steady state computation, where  $M$  denotes the number of time instances used in the HB method [17, 18]. The structure of the Jacobian will further be such that it consists of  $M$  diagonal sub blocks that contain the normal flux Jacobian, plus additional off diagonal terms that correspond to the linearization of the time-derivative source term. This structure reduces the diagonal dominance of the linear system when the time derivative source term is large [14], and was found to impact convergence rate of the linear solver in [18]. It is interesting to note that the ratio derived in [14] to quantify the diagonal dominance of the linear system is equivalent to the grid reduced frequency introduced by Gentilli [16] to quantify the stability of explicit HB solvers. One way to strengthen the diagonal dominance of the linear system is to restrict the pseudo-time step used in the backward Euler method. Su and Yuan [14] do this based on the formula originally proposed by van der Weide et al. [13]. Another approach, adopted by Antheaume and Corre [19], is to add the norm of the time derivative source term to the diagonal of the linear system. In either of these cases, the linear system is better conditioned at the cost of slower convergence in pseudo-time.

If the time derivative source term on the other hand is treated explicitly in the backward Euler scheme, the resulting linear system decouples across all time instances. This approach facilitates the implementation of the HB method into an existing CFD code but has unfortunately been proven to be unconditionally unstable [20]. Woodgate and Badcock [18] and Sicot et al. [21] have also verified this stability issue numerically. Several schemes have therefore been proposed to circumvent the problem of having to solve a large linear system, while at the same time retaining numerical stability. Sicot et al. [21] use a block-Jacobi symmetric successive over-relaxation (BJ-SSOR) technique to solve the linear system. Another approach is the two step approximate factorization method developed by Thomas et al. [17]. Implicit residual smoothing using a LU-SGS method combined with the block Jacobi method to treat the time spectral derivative source term implicitly has also been shown to greatly improve the stability of an explicit Runge-Kutta solver in [22].

In summary, a wealth of research has been conducted on numerical methods for solving the Harmonic Balance form of the Navier-Stokes equations in fluid dynamics. Most of these methods are based on employing either explicit or implicit pseudo-time integration to converge the discrete residual to zero. In both these cases, it has been shown that the time derivative source term introduces new challenges. For an explicit scheme, the stability of the pseudo-time integration may be compromised, whereas, for an implicit scheme, the linear system becomes harder to solve. This paper starts by deriving the eigenvalue spectra obtained from using the HB method and finite volume method to respectively discretize the Euler equations in time and space. Based on these results, three different methods available in the literature that potentially could be used to stabilize explicit pseudo-time integration of the discrete equations are analyzed. These include the point implicit Runge-Kutta method proposed by McMullen et al. [11] and Campobasso and Baba-Ahmadi [9], the HB source term stabilization matrix used by Thomas et al. [17] in their two step approximate factorization approach, and the pseudo-time step restriction proposed by van der Weide et al. [13]. Based on the knowledge gained from this

analysis, a novel time-level preconditioner is then derived for the time domain HB method. This preconditioner scales the pseudo-time step with which each frequency component is being updated, without having to explicitly convert the time domain HB solution into the frequency domain. The benefit of this time-level preconditioner is also demonstrated by simulating laminar vortex shedding behind a cylinder. Although the time-level preconditioner presented in this paper has been developed for an explicit solver, it is believed that it also can enhance the convergence of an implicit solver by increasing the diagonal dominance of the linear system.

### III. Method

#### A. Harmonic Balance Temporal Discretization

The Harmonic Balance method proposed by Hall et al. [4] can be used to compute time-periodic solutions to a wide variety of differential equations. In this work, the HB method will be applied to governing equations in fluid dynamics. For the stability analysis presented later, the one dimensional Euler equations for a perfect, ideal gas will be considered. These equations may be written in compact form as

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathcal{F}(\mathbf{Q})}{\partial x} = 0 \quad (1)$$

In this equation,  $\mathbf{Q} \in \mathbb{R}^{N_{\text{var}}}$  is a vector that contains the conserved variables and  $\mathcal{F} : \mathbb{R}^{N_{\text{var}}} \rightarrow \mathbb{R}^{N_{\text{var}}}$  a function that computes the convective flux of each conserved variable. The convective flux is a nonlinear function of the conserved variables. This implies that periodic solutions to Eq. (1) can contain an infinite number of sinusoids with different frequencies. Computing the amplitude of each of these sinusoids is however not feasible from a computational point of view. Therefore, the HB method considers approximate solutions to Eq. (1) instead

$$\mathbf{Q}(t) = \sum_{k=-K}^K \hat{\mathbf{Q}}_k e^{i\omega_k t} \quad (2)$$

Here,  $\omega_k = k\omega$  is the frequency of the  $k^{\text{th}}$  harmonic and  $\omega$  the fundamental frequency of the problem.

The first step in deriving the HB form of Eq. (1) is to substitute Eq. (2) into Eq. (1). This yields the following equation

$$\mathbf{E}_K^{-1}(t) \mathbf{\Omega}_K \hat{\mathbf{Q}}_K + \frac{\partial \mathcal{F}(\mathbf{E}_K^{-1}(t) \hat{\mathbf{Q}}_K)}{\partial x} = 0 \quad (3)$$

In this equation,  $\mathbf{E}_K^{-1}(t)$  and  $\mathbf{\Omega}_K$  are two block diagonal matrices with  $N_{\text{var}} \times N_{\text{var}}$  blocks. The diagonal blocks in these two matrices are respectively defined as

$$\mathbf{E}_K^{-1}(t) = [1 \ e^{i\omega_1 t} \ e^{i\omega_2 t} \ \dots \ e^{i\omega_{-1} t}] \quad \mathbf{\Omega}_K = \begin{bmatrix} 0 & 0 & \dots & & 0 \\ 0 & i\omega_1 & 0 & \dots & 0 \\ \vdots & 0 & i\omega_2 & 0 & \dots & 0 \\ & \vdots & 0 & \ddots & & 0 \\ & & \vdots & & & 0 \\ 0 & 0 & 0 & 0 & 0 & i\omega_{-1} \end{bmatrix} \quad (4)$$

The vector  $\hat{\mathbf{Q}}_K \in \mathbb{R}^{N_{\text{var}} \times 2K+1}$  in Eq. (3) is further structured into  $N_{\text{var}}$  sub-vectors, in which the  $n^{\text{th}}$  sub-vector contains all Fourier coefficients of the  $n^{\text{th}}$  conserved variable.

The derivation of the HB form of Eq. (1) continues by imposing the requirement that Eq (3) should be satisfied at  $M = 2K + 1$  time instances that are evenly distributed over the time period  $T = 2\pi/\omega$  [4]. This requirement may be stated as

$$\mathbf{E}_K^{-1}(\mathbf{t}) \mathbf{\Omega}_K \hat{\mathbf{Q}}_K + \frac{\partial \mathcal{F}^*(\mathbf{E}_K^{-1}(\mathbf{t}) \hat{\mathbf{Q}}_K)}{\partial x} = 0 \quad (5)$$

where

$$\mathbf{t} = [t_0, t_1, \dots, t_{M-1}]^T \quad (6)$$

is a vector that contains the time instances  $t_l = lT/M$  at which Eq. (3) is required to be satisfied. Similar to  $\mathbf{E}_K^{-1}(t)$  in Eq. (3), the matrix  $\mathbf{E}_K^{-1}(\mathbf{t})$  in Eq. (5) also has a block diagonal structure with  $N_{\text{var}} \times N_{\text{var}}$  blocks. Each diagonal block in  $\mathbf{E}_K^{-1}(\mathbf{t})$  contains the following inverse discrete Fourier transform matrix

$$E_K^{-1}(\mathbf{t}) = \begin{bmatrix} 1 & e^{i\omega_1 t_0} & e^{i\omega_2 t_0} & \dots & e^{i\omega_{-1} t_0} \\ 1 & e^{i\omega_1 t_1} & e^{i\omega_2 t_1} & \dots & e^{i\omega_{-1} t_1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & e^{i\omega_1 t_{M-1}} & e^{i\omega_2 t_{M-1}} & \dots & e^{i\omega_{-1} t_{M-1}} \end{bmatrix} \quad (7)$$

The function  $\mathcal{F}^* : \mathbb{R}^{N_{\text{var}} \times M} \rightarrow \mathbb{R}^{N_{\text{var}} \times M}$  in Eq. (5) further computes the convective flux of each conserved variable at each time instant. The argument to this function is the realization of the Fourier coefficients in  $\hat{\mathbf{Q}}_K$  at the time instances in  $\mathbf{t}$

$$\mathbf{Q}^* = \mathbf{E}_K^{-1}(\mathbf{t}) \hat{\mathbf{Q}}_K \quad (8)$$

The inverse discrete Fourier transform matrix defined in Eq. (7) is always invertible for the given choice of frequencies and time instances. Due to the block diagonal structure of  $\mathbf{E}_K^{-1}(\mathbf{t})$ , it follows that  $\mathbf{E}_K^{-1}(\mathbf{t})$  is also invertible. The inverse of  $\mathbf{E}_K^{-1}(\mathbf{t})$ , here denoted  $\mathbf{E}_K(\mathbf{t})$ , is also a block diagonal matrix, whose diagonal blocks contain the following discrete Fourier transform matrix

$$E_K(\mathbf{t}) = \frac{1}{M} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ e^{-i\omega_1 t_0} & e^{-i\omega_1 t_1} & e^{-i\omega_1 t_2} & \dots & e^{-i\omega_1 t_{M-1}} \\ \vdots & \vdots & \vdots & & \vdots \\ e^{-i\omega_{-1} t_0} & e^{-i\omega_{-1} t_1} & e^{-i\omega_{-1} t_2} & \dots & e^{-i\omega_{-1} t_{M-1}} \end{bmatrix} \quad (9)$$

The matrix  $\mathbf{E}_K(\mathbf{t})$  may be used to compute the Fourier coefficients of the conserved variables from the vector  $\mathbf{Q}^*$  as follows

$$\hat{\mathbf{Q}}_K = \mathbf{E}_K(\mathbf{t}) \mathbf{Q}^* \quad (10)$$

This relation together with Eq. (8) may now be inserted in Eq. (5) to obtain the time domain formulation of the HB method

$$\underbrace{\mathbf{E}_K^{-1}(\mathbf{t}) \boldsymbol{\Omega}_K \mathbf{E}_K(\mathbf{t})}_{\mathbf{D}} \mathbf{Q}^* + \frac{\partial \mathcal{F}^*(\mathbf{Q}^*)}{\partial x} = 0 \quad (11)$$

The operator denoted  $\mathbf{D}$  in this equation is often referred to as the time spectral derivative operator [4]. It represents a finite difference operator of spectral order, that returns the exact time derivative of  $\mathbf{Q}$  at the time instances in  $\mathbf{t}$  as long as the solution contains at most  $K$  harmonics.

The HB method may also be formulated in the frequency domain, in which case it often is referred to as the Nonlinear Frequency Domain (NLFD) method [12]. The NLFD formulation of Eq. (1) is obtained by multiplying Eq. (5) by  $\mathbf{E}_K(\mathbf{t})$ . This yields the following result

$$\boldsymbol{\Omega}_K \hat{\mathbf{Q}}_K + \mathbf{E}_K(\mathbf{t}) \frac{\partial \mathcal{F}^*(\mathbf{E}_K^{-1}(\mathbf{t}) \hat{\mathbf{Q}}_K)}{\partial x} = 0 \quad (12)$$

The frequency domain formulation of the HB method presented in the above equation will be used for the upcoming stability analysis. The results presented will however be equally valid for the time domain HB method. This follows from the fact that Eqs (11) and (12) are equivalent.

## B. Finite Volume Spatial Discretization

Before the stability analysis is performed in the next section, Eq. (12) will also be discretized in space using the finite volume method. The first step in this process is to rewrite Eq. (12) as follows

$$\mathbf{\Omega}_K \hat{\mathbf{Q}}_K + \mathbf{E}_K(\mathbf{t}) \frac{\partial \mathcal{F}^*(\mathbf{Q}^*)}{\partial \mathbf{Q}^*} \frac{\partial \mathbf{Q}^*}{\partial x} = 0 \quad (13)$$

In this equation, the flux Jacobian  $\frac{\partial \mathcal{F}^*(\mathbf{Q}^*)}{\partial \mathbf{Q}^*}$  was introduced together with the realization of the Fourier coefficients in the time domain given by Eq. (8). This flux Jacobian is structured into  $N_{\text{var}} \times N_{\text{var}}$  blocks, where each block contains a diagonal matrix of size  $M \times M$ . The diagonal elements in block  $(m, l)$  are obtained by evaluating element  $(m, l)$  in the flux Jacobian of the 1D Euler equations ( $\frac{\partial \mathcal{F}(\mathbf{Q})}{\partial \mathbf{Q}}$ ) at each of the time instances in  $\mathbf{t}$ . If all flow perturbations are small in amplitude, however, the flux Jacobian may be approximated by evaluating it from a spatially and temporally averaged mean state. This yields the following linearized form of Eq. (13)

$$\mathbf{\Omega}_K \hat{\mathbf{Q}}_K + \mathbf{E}_K(\mathbf{t}) \frac{\partial \mathcal{F}^*(\bar{\mathbf{Q}})}{\partial \mathbf{Q}^*} \frac{\partial \mathbf{Q}^*}{\partial x} = 0 \quad (14)$$

Since the elements in the flux Jacobian have been assumed to be the same for all time instances, it commutes with  $\mathbf{E}_K(\mathbf{t})$  [16]. Equation (14) may therefore be rewritten as

$$\mathbf{\Omega}_K \hat{\mathbf{Q}}_K + \frac{\partial \mathcal{F}^*(\bar{\mathbf{Q}})}{\partial \mathbf{Q}^*} \frac{\partial \hat{\mathbf{Q}}_K}{\partial x} = 0 \quad (15)$$

The next step in computing the spatial discretization of Eq. (12) is to introduce the diagonalization of the flux Jacobian

$$\frac{\partial \mathcal{F}^*(\bar{\mathbf{Q}})}{\partial \mathbf{Q}^*} = \mathcal{T}^* \mathbf{\Lambda}^* \mathcal{T}^{*-1} \quad (16)$$

In this equation,  $\mathcal{T}^*$ ,  $\mathbf{\Lambda}^*$  and  $\mathcal{T}^{*-1}$  are all block structured matrices that respectively contain the right eigenvectors, eigenvalues and left eigenvectors of  $\frac{\partial \mathcal{F}^*(\bar{\mathbf{Q}})}{\partial \mathbf{Q}^*}$ . For the one dimensional Euler equations, the eigenvalues are given by  $\bar{u}$ ,  $\bar{u} - \bar{c}$  and  $\bar{u} + \bar{c}$ , where  $\bar{u}$  and  $\bar{c}$  respectively denote the mean axial velocity and mean speed of sound. By substituting Eq. (16) into Eq. (15) and use the fact that the matrices  $\mathcal{T}^*$  and  $\mathcal{T}^{*-1}$  are constant in time and space, the following relation may be derived

$$\mathbf{\Omega}_K \hat{\mathbf{W}}_K + \mathbf{\Lambda}^* \frac{\partial \hat{\mathbf{W}}_K}{\partial x} = 0 \quad (17)$$

In this equation, the Fourier coefficients of the characteristic variables were introduced. They are defined as

$$\hat{\mathbf{W}}_K = \mathcal{T}^{*-1} \hat{\mathbf{Q}}_K \quad (18)$$

Equation (17) represents a set of uncoupled transport equations for the Fourier coefficients of the characteristic variables. The fact that these equations are uncoupled greatly facilitates the stability analysis, since it allows the propagation of a single Fourier coefficient of a single characteristic variable to be considered at a time. Let the  $k^{\text{th}}$  Fourier coefficient of the  $p^{\text{th}}$  characteristic variable be denoted  $\hat{w}_k^p$ . From Eq. (17), it follows that the transport equation for this variable may be written as

$$i\omega_k \hat{w}_k^p + \lambda_p \frac{\partial \hat{w}_k^p}{\partial x} = 0 \quad (19)$$

Here,  $\lambda_p$  denotes the eigenvalue corresponding to the  $p^{\text{th}}$  characteristic variable. As can be seen from the above relation, this eigenvalue represents the speed with which the corresponding characteristic variable propagates.

The spatial discretization is performed on a uniform grid with  $2N + 1$  equally sized cells of size  $\Delta x$ . By integrating Eq. (19) over one cell in this grid and applying Gauss' divergence theorem to the convection term, one obtains

$$i\omega_k \Delta x \hat{w}_{k,j}^p + \lambda_p (\hat{w}_{k,e} - \hat{w}_{k,w}) = 0 \quad (20)$$

In this equation,  $\hat{w}_{k,j}^p$ ,  $\hat{w}_{k,w}^p$  and  $\hat{w}_{k,e}^p$  respectively denote the average value, left face value and right face value of the  $j^{\text{th}}$  cell. The face values must be estimated using some appropriate flux scheme. In the G3D::Flow code used by the present

authors, a third-order accurate upwind scheme that utilises two cells on each side of the face is employed [23]. The direction of upwinding is further determined based on the sign of  $\lambda_p$ . Since two cells on each side of a cell face are used, the flux term in Eq. (20) can be written in functional form as

$$i\omega_k \hat{w}_{k,j}^p + \frac{\lambda_p}{\Delta x} F(\hat{w}_{k,j-2}^p, \dots, \hat{w}_{k,j+2}^p) = 0 \quad (21)$$

Note that  $F$  is a linear operator that simply computes a weighed sum of the average cell values surrounding the  $j^{\text{th}}$  cell. The above equation represents the spatially and temporally discretized one dimensional Euler equations in the frequency+characteristic variable domain.

As noted in the introduction, many codes today use local time stepping to converge the spatially+temporally discretized residual to zero. This implies that the following set of equations are integrated until a steady state is reached

$$\frac{\partial \hat{w}_{k,j}^p}{\partial \tau} + i\omega_k \hat{w}_{k,j}^p + \frac{\lambda_p}{\Delta x} F(\hat{w}_{k,j-2}^p, \dots, \hat{w}_{k,j+2}^p) = 0 \quad (22)$$

In the next section, the stability of this pseudo-time integration will be analyzed when it is performed with explicit Runge-Kutta schemes.

### C. Stability Analysis

The first step in the stability analysis is to introduce eigenvectors to the spatial operator in Eq. (22). In this work, it is assumed that the solution is periodic in space. Under this assumption, the following spatial Fourier mode represents an eigenvector to  $F$  [24]

$$\begin{aligned} \hat{w}_{k,j}^p &= \hat{w}_{k,n}^p e^{ik_n x_j} \\ &= \hat{w}_{k,n}^p e^{ij\phi} \end{aligned} \quad (23)$$

Here,  $\hat{w}_{k,n}^p$  is the spatio-temporal Fourier coefficient for the  $n^{\text{th}}$  axial wavenumber ( $k_n$ ) and  $k^{\text{th}}$  frequency ( $\omega_k$ ),  $x_j = j\Delta x$  is the axial coordinate of the  $j^{\text{th}}$  cell, and the quantity  $\phi$  is defined as

$$\phi = k_n \Delta x \quad (24)$$

The highest axial wavenumber that can be resolved on the mesh is  $\pi/\Delta x$  by the Whittaker-Kotel'nikov-Shannon sampling theorem. This implies that  $\phi$  can take on all values between  $-\pi$  and  $\pi$ . By now inserting Eq. (23) into Eq. (22), one obtains the following relation

$$\frac{\partial \hat{w}_{k,n}^p}{\partial \tau} + i\omega_k \hat{w}_{k,n}^p + \frac{\lambda_p}{\Delta x} \xi(\phi) \hat{w}_{k,n}^p = 0 \quad (25)$$

In this relation,  $\xi(\phi)$  represents the eigenvalues of the spatial operator  $F$ . An analytical expression for these eigenvalues may be obtained by applying  $F$  to the Fourier mode in Eq. (23). Without loss of generality one may assume that  $\lambda_p > 0$ . For the upwind schemes used in the G3D::Flow code,  $\xi(\phi)$  then becomes as follows

$$\xi(\phi) = -a_1 e^{-i2\phi} + (a_1 - a_2) e^{-i\phi} + (a_2 - a_3) + (a_3 - a_4) e^{i\phi} + a_4 e^{i2\phi} \quad (26)$$

Here,  $a_i$  are the weights used in the upwind scheme.

For the purpose of the stability analysis, Eq. (25) is now rewritten as follows

$$\frac{\partial \hat{w}_{k,n}^p}{\partial \tau} = - \underbrace{\frac{\lambda_p}{\Delta x} (i\bar{\omega}_{k,p} + \xi(\phi))}_{\Omega} \hat{w}_{k,n}^p \quad (27)$$

where

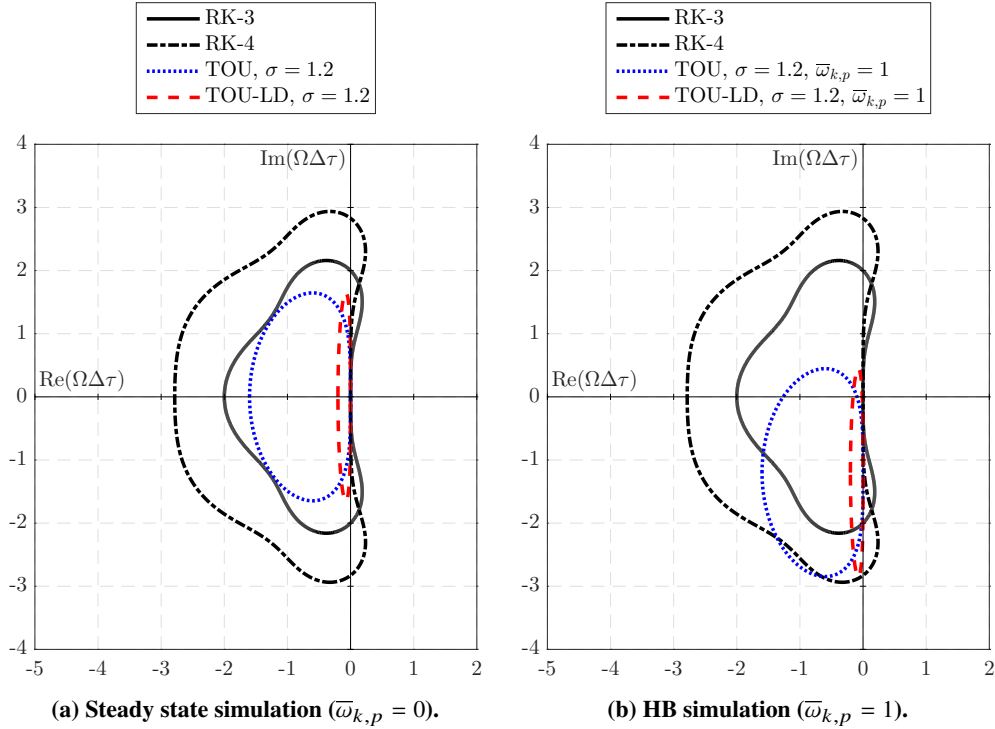
$$\bar{\omega}_{k,p} = \frac{\omega_k \Delta x}{\lambda_p} \quad (28)$$

is the grid reduced frequency introduced by Gentili [16]. The linear assumption introduced in Eq. (14) implies that pseudo-time integration of the original governing equations will be stable if the pseudo-time integration of Eq. (27) is stable for all values of  $\Omega$ . A necessary condition for this to hold is that the contour of  $\Omega\Delta\tau$ , where  $\Delta\tau$  is the pseudo-time step, lies within the stability boundary of the pseudo-time integration scheme for all values of  $\phi$  and  $\bar{\omega}_{k,p}$  [24]. The stability boundary of the pseudo-time integration scheme is further defined as the contour in the complex plane where the norm of its eigenvalue is unity [24]. Before this stability condition is analyzed, however, the product of  $\Omega\Delta\tau$  is first expressed as

$$\begin{aligned}\Omega\Delta\tau &= -\frac{\lambda_p\Delta\tau}{\Delta x} \left( i\bar{\omega}_{k,p} + \xi(\phi) \right) \\ &= -\sigma \left( i\bar{\omega}_{k,p} + \xi(\phi) \right)\end{aligned}\quad (29)$$

In this equation, the well known Courant–Friedrichs–Lewy (CFL) number  $\sigma$  was introduced. This number can be seen to scale the eigenvalues of the spatial+temporal discretization.

From Eq. (29) it can be noted that the eigenvalues of the temporal discretization are purely imaginary. The presence of the temporal discretization will thus move the eigenvalue spectra along the imaginary axis. If an explicit pseudo-time integration scheme is used, this can cause instabilities due to the limited stability region of these integration schemes. In order to illustrate this more clearly, contours of  $\Omega\Delta\tau$  have been plotted in Fig. 1 together with stability boundaries of two different Runge-Kutta pseudo-time integration schemes. The contours of  $\Omega\Delta\tau$  shown in this figure were obtained by varying  $\phi$  in Eq. (29) between  $-\pi$  and  $\pi$  for different combinations of the CFL number, grid-reduced frequency, and spatial discretization scheme. The eigenvalue contours denoted TOU stem from a standard, third order accurate upwind scheme, whereas the contours denoted TOU-LD stem from a third order accurate upwind scheme with lower dissipation than the standard one [23]. In Fig. 1a, the grid reduced frequency is zero and the CFL number is  $\sigma = 1.2$ . For this case, the stability condition can be seen to be satisfied for all combinations of pseudo-time integration and spatial discretization scheme. As shown in Fig. 1b, this is no longer the case when the CFL number is kept constant and the grid reduced frequency is increased to  $\bar{\omega}_{k,p} = 1$ . It is thus not possible to retain the same settings as is used for steady state simulations when the HB equations are being solved.



**Fig. 1** Stability boundaries for two Runge-Kutta schemes together with eigenvalue spectra for different combinations of spatial discretization scheme, CFL number, and grid reduced frequency.

Based on the previous findings, one can think of two ways to restore stability when the HB equations are solved: select a pseudo-time integration scheme with a wider stability margin, or scale the residual using a preconditioner. A simple way to achieve the first option could be to switch from, e.g., a three stage to a four stage Runge-Kutta scheme when the HB equations are being solved. This would allow a similar CFL number as was used in the steady state simulation to be used for the HB simulation. A simple scaling of the residual could on the other hand be achieved by lowering the CFL number. Although this latter approach would suffer from a worse convergence rate, the total CPU time to reach convergence may be the same as the previous approach since a Runge-Kutta scheme with more stages is computationally more expensive per iteration. Given that these two approaches both have strengths and weaknesses, it is likely that the optimal setup is a combination of a tailored Runge-Kutta scheme and a good preconditioner. In the upcoming sections, one tailored Runge-Kutta scheme and two preconditioners that have been proposed in the literature will be investigated. Based on the findings obtained from this analysis, a new preconditioner for the time domain HB method will be presented. Alternative tailored pseudo-time integration schemes will on the other hand not be covered in this work.

### 1. Point Implicit Runge-Kutta

The point implicit Runge-Kutta method was originally developed by Melson et al. [25] for the dual time stepping method. Later, the application of this method to solve the HB equations was considered by McMullen [5] and Campobasso and Baba-Ahmadi [9]. The point implicit Runge-Kutta method is based on treating the term associated with the time derivative implicitly within each Runge-Kutta stage. In order to illustrate this, a general  $S$  stage Runge-Kutta algorithm will first be considered. It may be written as follows [24]

$$\begin{aligned}
q^{(1)} &= q^m \\
q^{(2)} &= q^m + \Delta\tau\alpha_{2,1}\mathcal{R}q^{(1)} \\
&\vdots \\
q^{(s)} &= q^m + \Delta\tau \sum_{r=1}^{s-1} \alpha_{s,r}\mathcal{R}q^{(r)} \\
&\vdots \\
q^{(S)} &= q^m + \Delta\tau \sum_{r=1}^{S-1} \alpha_{S,r}\mathcal{R}q^{(r)} \\
q^{(S+1)} &= q^m + \Delta\tau \sum_{r=1}^S \beta_r\mathcal{R}q^{(r)} \\
q^{m+1} &= q^{(S+1)}
\end{aligned} \tag{30}$$

In this algorithm,  $q$  is the solution,  $\mathcal{R}$  is the residual operator,  $m$  is the pseudo-time time step index, and  $\alpha_{s,r}, \beta_r$  are the Runge-Kutta coefficients. These coefficients are normally summarized in a Butcher tableau as shown in Table 1.

$\alpha_{2,1}$				
$\vdots$	$\ddots$			
$\alpha_{S,1}$	$\dots$	$\alpha_{S,S-1}$		
$\beta_1$	$\dots$	$\beta_{S-1}$	$\beta_S$	

**Table 1** Butcher tableau for Runge-Kutta coefficients.

The point implicit Runge-Kutta method is derived from Eq. (30) by first splitting the residual operator in two parts, one representing the spatial discretization and one representing the temporal discretization. After this, the residual term corresponding to the temporal discretization is treated implicitly within each Runge-Kutta stage. This is achieved by evaluating the temporal residual as  $\mathcal{R}_{\text{time}}q^{(r+1)}$ , rather than as  $\mathcal{R}_{\text{time}}q^{(r)}$ , when the summation is performed in each Runge-Kutta stage. Due to the construction of the Runge-Kutta algorithm, all terms in this summation can be directly

evaluated except the last one, which instead is moved to the left hand side. When these steps are followed, the following general form of the point implicit Runge-Kutta method is obtained

$$\begin{aligned}
q^{(1)} &= q^m \\
(I - \Delta\tau\alpha_{2,1}\mathcal{R}_{\text{time}})q^{(2)} &= q^m + \Delta\tau\alpha_{2,1}\mathcal{R}_{\text{space}}q^{(1)} \\
&\vdots \\
(I - \Delta\tau\alpha_{s,s-1}\mathcal{R}_{\text{time}})q^{(s)} &= q^m + \Delta\tau\sum_{r=1}^{s-1}\alpha_{s,r}\mathcal{R}_{\text{space}}q^{(r)} + \Delta\tau\sum_{r=1}^{s-2}\alpha_{s,r}\mathcal{R}_{\text{time}}q^{(r+1)} \\
&\vdots \\
(I - \Delta\tau\alpha_{S,S-1}\mathcal{R}_{\text{time}})q^{(S)} &= q^m + \Delta\tau\sum_{r=1}^{S-1}\alpha_{S,r}\mathcal{R}_{\text{space}}q^{(r)} + \Delta\tau\sum_{r=1}^{S-2}\alpha_{S,r}\mathcal{R}_{\text{time}}q^{(r+1)} \\
(I - \Delta\tau\beta_S\mathcal{R}_{\text{time}})q^{(S+1)} &= q^m + \Delta\tau\sum_{r=1}^S\beta_r\mathcal{R}_{\text{space}}q^{(r)} + \Delta\tau\sum_{r=1}^{S-1}\beta_r\mathcal{R}_{\text{time}}q^{(r+1)} \\
q^{m+1} &= q^{(S+1)}
\end{aligned} \tag{31}$$

It should be noted that the operators on the left hand side in the above relation needs to be inverted before the solution can be updated in the corresponding Runge-Kutta stage. When the point implicit Runge-Kutta method is applied to Eq. (27), this amounts to a simple division since the residual operator in this equation is represented by a scalar. More precisely, the residual operator in Eq. (27) is defined as

$$\Delta\tau\mathcal{R} = \underbrace{-i\sigma\bar{\omega}_{k,p}}_{\Delta\tau\mathcal{R}_{\text{time}}} - \underbrace{\sigma\xi(\phi)}_{\Delta\tau\mathcal{R}_{\text{space}}} \tag{32}$$

In order to assess the applicability of the point-implicit Runge-Kutta method for solving the HB form of the governing equations, the stability boundary of this algorithm was calculated for two sets of Runge-Kutta coefficients and three different grid reduced frequencies using the residual operator defined in the above equation. The Runge-Kutta coefficients that were employed for this analysis are summarized in Table 2, and the resulting stability boundaries are shown in Fig. 2. The eigenvalue spectra for the two spatial discretization schemes described earlier have also been included in this figure. The reason for not including the contribution of the temporal discretization in these eigenvalue spectra is that this contribution now is taken care of by the point implicit Runge-Kutta method. From Fig. 2, it can be seen that the stability boundary for the point implicit Runge-Kutta method is rotated and expanded compared to that of the original Runge-Kutta algorithm. The expansion of the stability boundary is clearly beneficial from a stability point of view. Unfortunately, the corresponding rotation of the boundary has the opposite effect since it can uncover some eigenvalues of the spatial discretization. Figure 2 shows that this primarily happens for small values of the grid reduced frequency. This drawback of the point implicit Runge-Kutta method in the context of the HB method was originally discovered by McMullen [5]. The fact that the point implicit Runge-Kutta method can have a stabilizing effect for cases with high frequencies, as shown in Fig. 2c, was further reported by Campobasso and Baba-Ahmadi [9]. The stability of the point implicit Runge-Kutta method is thus dependent on the grid reduced frequency, as well as the spatial discretization scheme used. Given that the grid reduced frequency can vary a lot in the computational domain due to mesh size variations, it must however be concluded that this method can not be used as a generic approach for stabilizing explicit time integration of the HB equations.

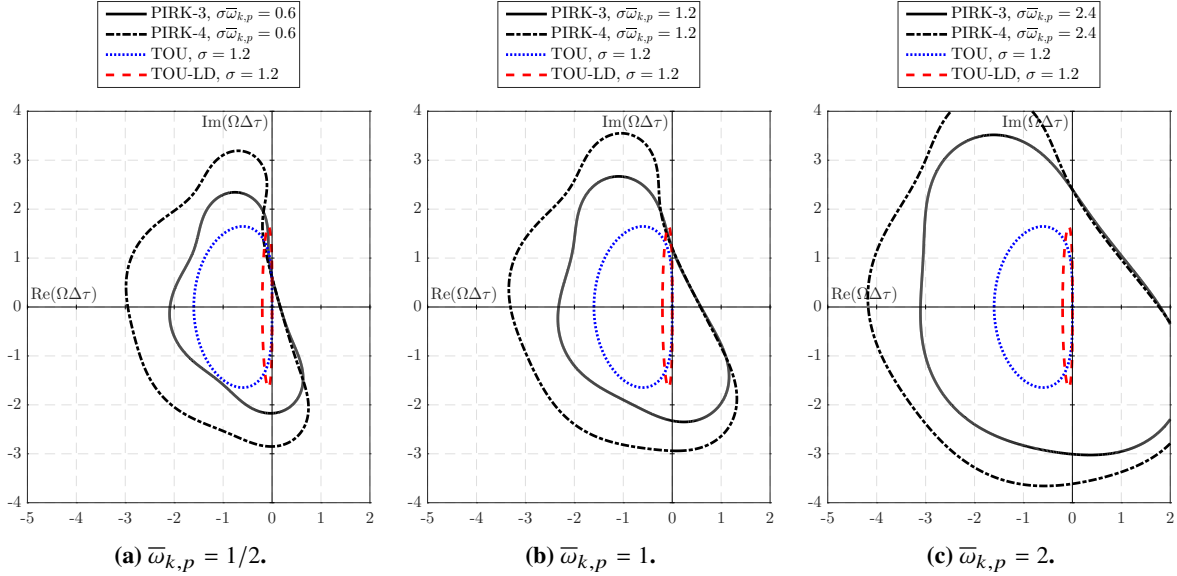
1		
1/2	1/2	
1/2	0	1/2

(a) RK-3

		1/4	
	0	1/3	
	0	0	1/2
	0	0	0
			1

(b) RK-4

**Table 2** Butcher tableau for a three stage and a four stage Runge-Kutta method.



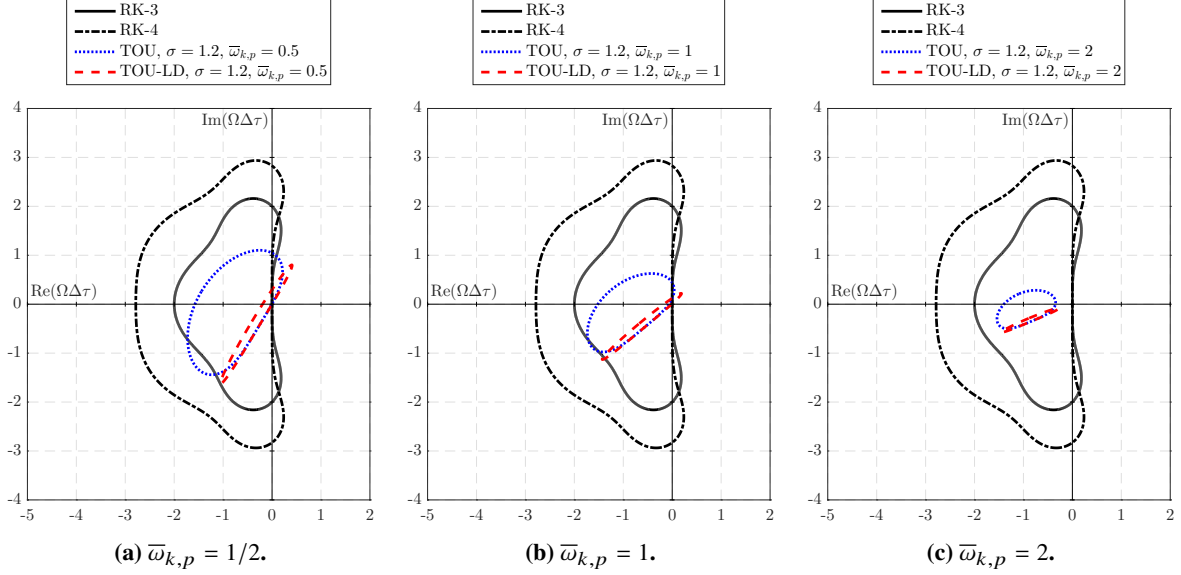
**Fig. 2** Stability boundaries for two point implicit Runge-Kutta schemes and three different grid reduced frequencies together with eigenvalue spectra for two different spatial discretization schemes.

## 2. HB Source Term Stabilization Matrix

The next stabilization method that has been considered is the HB source term stabilization matrix used by Thomas et al. [17] in their two step approximate factorization approach. In this method, the time domain HB residual is multiplied by the matrix  $(I + \Delta\tau\mathbf{D})^{-1}$ . If this technique is applied to explicit pseudo-time integration of Eq. (27), it yields the following scaled residual

$$\Delta\tau\mathcal{R}_{\text{stab}} = \frac{1}{1 + i\sigma\bar{\omega}_{k,p}} (-\sigma (i\bar{\omega}_{k,p} + \xi(\phi))) \quad (33)$$

The eigenvalue spectra defined by this residual operator is shown for three different values of the grid reduced frequency and the two spatial discretization schemes presented earlier in Fig. 3. In this figure, the stability boundaries of the two Runge-Kutta schemes presented in Table 2 have also been included. From Fig. 3, it can be seen that the stabilization matrix scales and rotates the eigenvalue spectra. The scaling of the eigenvalue spectra is clearly an advantage from a stability point of view. The corresponding rotation can however be seen to cause stability issues for small values of the grid reduced frequency. This implies that, although the HB source term stabilization matrix ensures that the implicit pseudo-time integration used in the two step approximate factorization approach remains stable [17], this matrix unfortunately does not stabilize the explicit pseudo-time integration approach used in this work. It should however be noted that the HB source term stabilization matrix can stabilize solvers using e.g. a first order upwind spatial discretization together with an Euler explicit pseudo-time integration scheme [20].



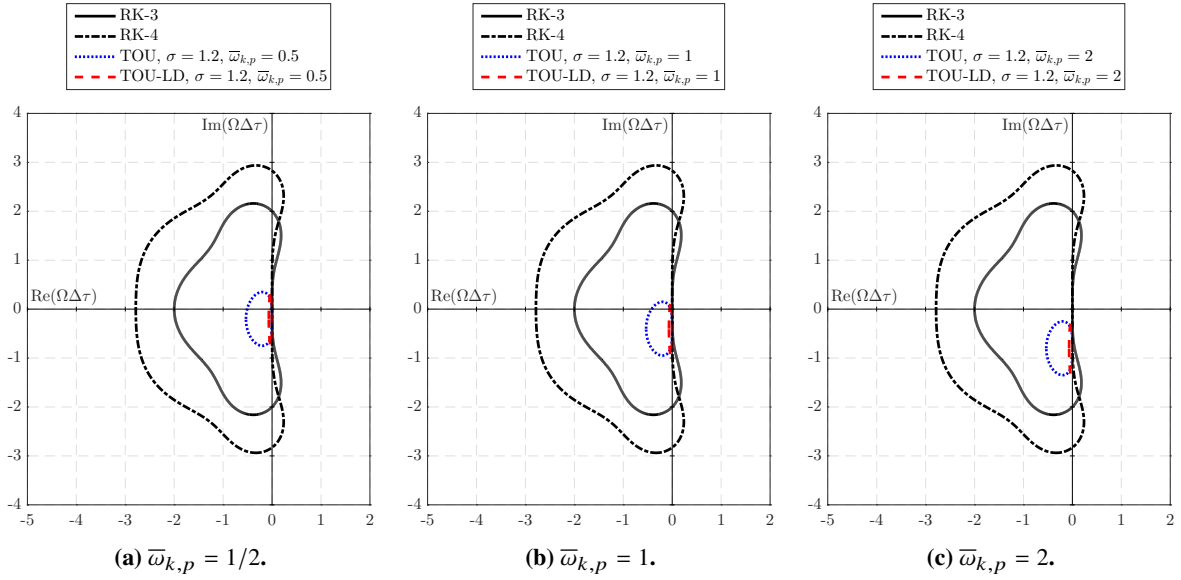
**Fig. 3** Stability boundaries for two different Runge-Kutta schemes together with eigenvalue spectra obtained with the HB source term stabilization matrix for different combinations of spatial discretization scheme, CFL number, and grid reduced frequency.

### 3. Restriction of Pseudo-Time Step

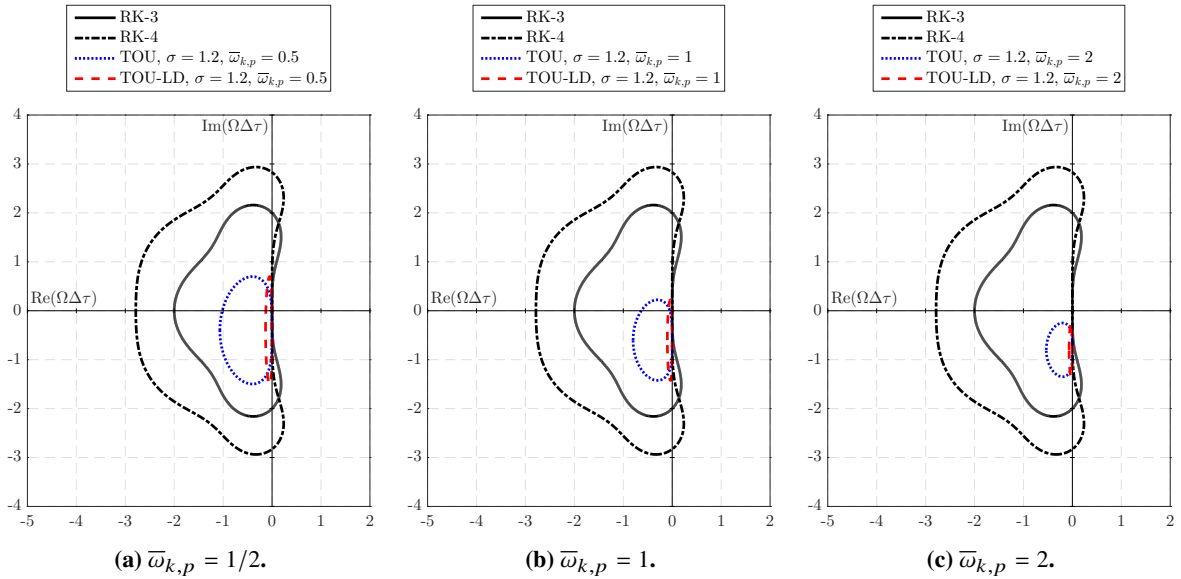
Several authors have proposed to restrict the pseudo-time step in order to stabilize explicit integration of the HB equations [5, 13, 16]. Among these, the pseudo-time step restriction proposed by van der Weide et al. [13] appears to be the most widely adopted one for the time domain HB method. When this approach is applied to Eq. (27), the residual reads as follows

$$\Delta\tau\mathcal{R}_{\text{restrict}} = \frac{1}{1 + \bar{\omega}_{K,p}} (-\sigma (i\bar{\omega}_{k,p} + \xi(\phi))) \quad (34)$$

In this equation, the residual can be seen to be scaled by a term that includes the grid reduced frequency for the largest harmonic present in the HB computation ( $K$ ). The reason for choosing this scaling factor is that it ensures that the corresponding pseudo-time integration of the governing equations in the time domain, where all frequencies are updated simultaneously with the same time step, remains stable [13]. The eigenvalue spectra defined by the residual operator in Eq. (34) is shown for three different values of the grid reduced frequency and the two spatial discretization schemes presented earlier in Fig. 4. In this figure, the stability boundaries of the two Runge-Kutta schemes presented in Table 2 have also been included. Figure 4 shows that the scaled pseudo-time step proposed by van der Weide et al. [13] successfully stabilizes explicit pseudo-time integration of the HB equations for all values of the grid reduced frequency. From Fig. 4a, it can also be seen that the pseudo-time step restriction is rather conservative for smaller values of the grid-reduced frequency. This is because the residual in Eq. (34) is scaled with a factor that only depends on the largest grid reduced frequency ( $\bar{\omega}_{K,p} = 2$  in Fig. 4). If on the other hand Eq. (34) is adapted so that the scaling factor depends on  $|\bar{\omega}_{k,p}|$ , the eigenvalue spectra shown in Fig. 5 are obtained instead. From this figure, it can be seen that the adaptive scaling factor ensures that the eigenvalue spectra for the lower frequencies are scaled less than when a global scaling factor is employed. This is positive from a computational point of view since a larger scaling factor implies faster convergence. However, as noted by van der Weide et al. [13], an adaptive time step for each frequency translates into a matrix multiplication of the residual in the time domain HB method. van der Weide et al. [13] avoid this matrix multiplication by choosing the most restrictive pseudo-time step that corresponds to the highest frequency in the problem for the time domain HB method. In this work, the matrix multiplication (preconditioning) approach will instead be followed. In addition to this, the scaling factor that is applied to each frequency will be tailored based on the chosen combination of spatial discretization and pseudo-time integration scheme, similar to what has previously been done by Gentili [16]. The derivation of this tailored preconditioner for the time domain HB method will be outlined next.



**Fig. 4** Stability boundaries for two different Runge-Kutta schemes together with eigenvalue spectra obtained with the pseudo-time step restriction defined in Eq. (34) for different combinations of spatial discretization scheme, CFL number, and grid reduced frequency.



**Fig. 5** Stability boundaries for two different Runge-Kutta schemes together with eigenvalue spectra obtained with an adapted scaling of the residual for different combinations of spatial discretization scheme, CFL number, and grid reduced frequency.

#### D. Derivation of Time Level Preconditioner

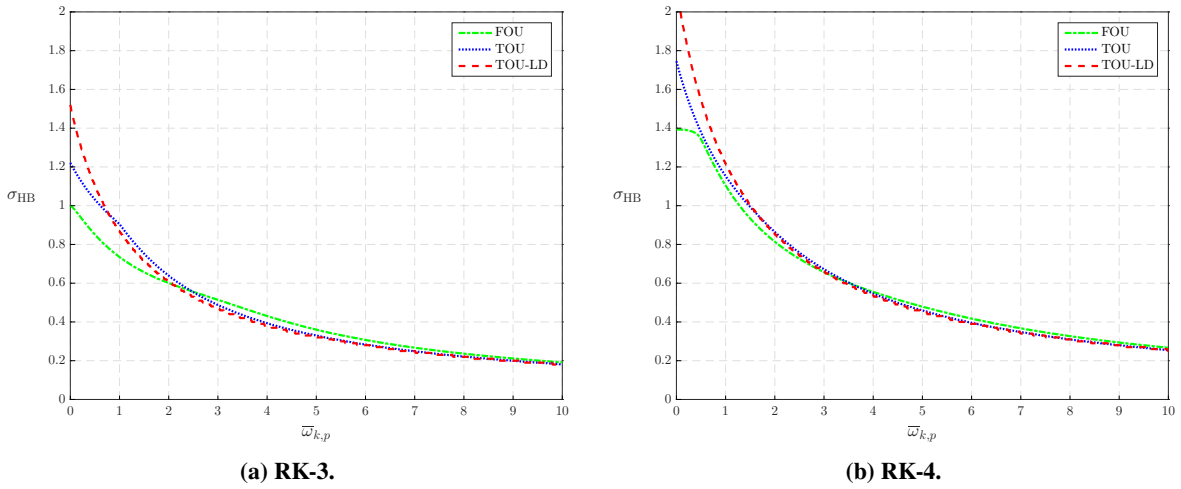
The time-level preconditioner that will be derived in this section is based on applying a tailored scaling factor to each frequency component of the residual as follows

$$\Delta\tau\mathcal{R}_{\text{prec}} = g(\bar{\omega}_{p,k}) (-\sigma (i\bar{\omega}_{k,p} + \xi(\phi))) \quad (35)$$

The scaling function  $g(\bar{\omega}_{p,k})$  should be determined to ensure that the pseudo-time integration remains stable. In order to do this, start by rewriting Eq. (35) as

$$\Delta\tau\mathcal{R}_{\text{prec}} = -\sigma_{\text{HB}} (i\bar{\omega}_{k,p} + \xi(\phi)) \quad (36)$$

Here,  $\sigma_{\text{HB}} = g(\bar{\omega}_{p,k})\sigma$  is a scaled CFL number. The largest value of this scaled CFL number that ensures stability may be determined for each grid reduced frequency by verifying that the eigenvalue spectra defined by the residual operator in the above equation lies within the stability boundary of the chosen pseudo-time integration scheme. The results obtained from this analysis are presented in Fig. 6.



**Fig. 6** Variation of largest allowable CFL number that ensures stability with grid-reduced frequency for different combinations of spatial discretization and pseudo-time integration schemes.

Figure 6 shows that the largest allowable CFL number that ensures stability decreases as the grid reduced frequency increases. As noted previously, this follows from the fact that a non-zero grid-reduced frequency moves the eigenvalue spectra along the imaginary axis, which in turn requires the CFL number to be reduced in order for the spectra to remain inside the stability boundary. From Fig. 6, it can also be seen that the largest allowable CFL number for a steady state computation ( $\bar{\omega}_{k,p} = 0$ ) depends on the chosen combination of spatial discretization and pseudo-time integration scheme, as should be expected. Suppose that the CFL number in Eq. (35) is chosen to be less than or equal to this value. It then follows that the HB computation remains stable if the scaling function  $g(\bar{\omega}_{k,p})$  in Eq. (35) is chosen as

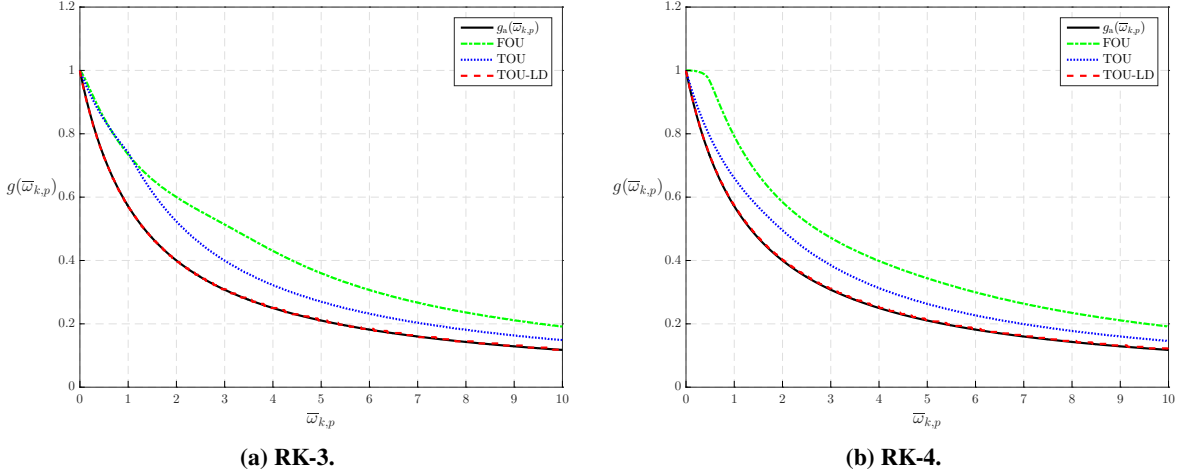
$$g(\bar{\omega}_{k,p}) = \frac{\sigma_{\text{HB}}}{\sigma_{\text{HB}}(\bar{\omega}_{k,p} = 0)} \quad (37)$$

In Fig. 7, the scaling function defined by the above relation is plotted for the same combinations of spatial discretization and pseudo-time integration schemes as was shown in Fig. 6. An approximate scaling function defined as

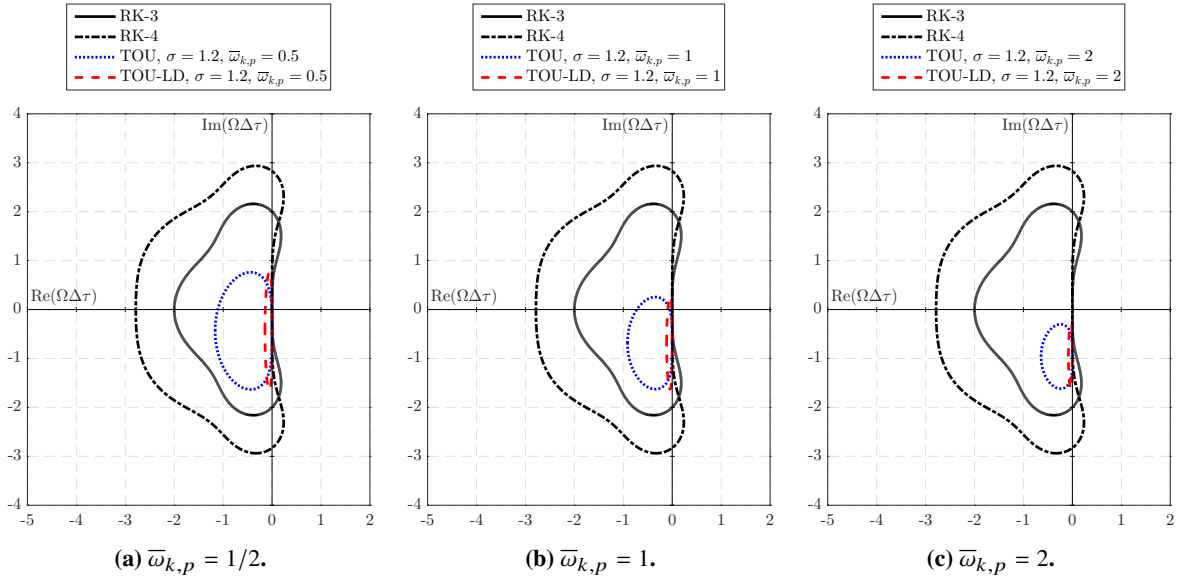
$$g_a(\bar{\omega}_{k,p}) = \frac{1}{1 + \gamma|\bar{\omega}_{k,p}|}, \quad \gamma = 0.75 \quad (38)$$

has also been included in Fig. 7. As can be seen from this figure, the approximate scaling function defined by the above equation satisfies  $g_a(\bar{\omega}_{k,p}) \leq g(\bar{\omega}_{k,p})$  for all combinations of spatial discretization, pseudo-time integration scheme and grid-reduced frequency. This function can thus be used to stabilize HB computations that use any of the numerical settings considered, although it will only be optimal for the TOU-LD spatial discretization as seen from Fig. 7. A plot of the eigenvalue spectra obtained when the approximate scaling function is used in Eq. (35) is shown in Fig. 8. By

comparing this figure to Fig. 5, it can be seen that the proposed scaling function gives a slightly less conservative scaling of the residual than the pseudo-time step restriction proposed by van der Weide et al. [13]. This small difference can be explained from the fact that the approximate scaling function is equivalent to the scaling used in Fig. 5 when  $\gamma = 1$ .



**Fig. 7** Plots of scaling function  $g(\bar{\omega}_{k,p})$  that ensures stability for different combinations of spatial discretization and pseudo-time integration schemes.



**Fig. 8** Stability boundaries for two different Runge-Kutta schemes together with eigenvalue spectra obtained with the approximate scaling function defined in Eq. (38) for different combinations of spatial discretization scheme, CFL number, and grid reduced frequency.

The integration of the proposed approximate scaling function into a preconditioner for the time domain HB method is outlined next. The first step in this process is to rewrite Eq. (38) as follows

$$g_a(\omega_k) = \frac{\sigma}{\sigma + \gamma|\omega_k|\Delta\tau}, \quad \gamma = 0.75 \quad (39)$$

The benefit of this new expression for the approximate scaling function is that it only contains properties that are readily available in a CFD code. In particular, the CFL number in the above relation is the CFL number that ensures

stability of the corresponding steady state computation. The scaling defined in Eq. (35) may now be implemented by preconditioning the spatially+temporally discretized residual of the governing equations in the time domain as follows

$$(\Delta\tau\mathcal{R}^*)_{\text{prec}} = \underbrace{\mathbf{E}_K^{-1}(\mathbf{t})\mathbf{G}_K\mathbf{E}_K(\mathbf{t})}_{\mathbf{P}}(\Delta\tau\mathcal{R}^*) \quad (40)$$

In this equation,  $\mathbf{G}_K$  is a block diagonal matrix with  $N_{\text{var}} \times N_{\text{var}}$  blocks. The diagonal blocks in this matrix are further defined as

$$G_K = \begin{bmatrix} 1 & 0 & \dots & & 0 \\ 0 & g_a(\omega_1) & 0 & \dots & 0 \\ \vdots & 0 & g_a(\omega_2) & 0 & \dots & 0 \\ & \vdots & 0 & \ddots & & 0 \\ & & \vdots & & & 0 \\ 0 & 0 & 0 & 0 & 0 & g_a(\omega_{-1}) \end{bmatrix} \quad (41)$$

The preconditioning matrix  $\mathbf{P}$  defined in Eq. (40) consists of three steps. First, the time domain residual of the HB equations is Fourier transformed to obtain the residual in the frequency domain. Next, each frequency component of the residual is scaled using the approximate scaling function in Eq. (39). Finally, the scaled residual in the frequency domain is converted back to the time domain using an inverse discrete Fourier transform. All these three operations can be performed with one single matrix multiplication by analytically calculating the elements in  $\mathbf{P}$ . Since its constituent matrices all are block diagonal,  $\mathbf{P}$  is block diagonal as well. The matrix elements in one of the blocks are further obtained as

$$P_{a,b} = \frac{1}{M} + \frac{2}{M} \sum_{k=1}^K g_a(\omega_k) \cos\left(\frac{2\pi k(a-b)}{M}\right) \quad (42)$$

Note that it is assumed that the pseudo-time step in Eqs. (39) and (40) has been chosen based on a CFL number that ensures that the corresponding steady state computation remains stable.

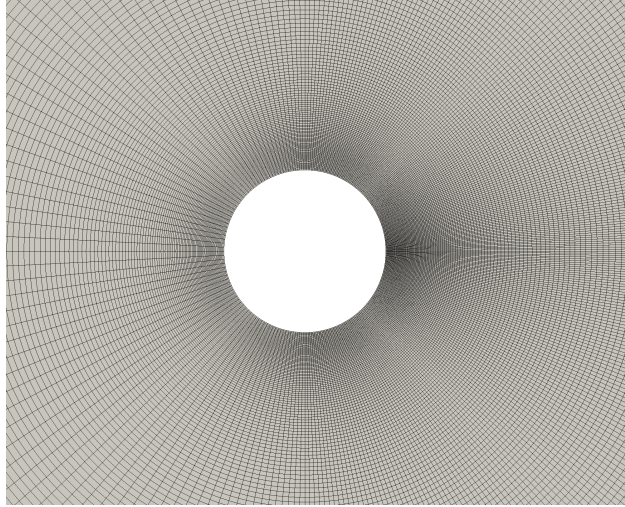
## IV. Results

### A. Laminar Vortex Shedding

Two dimensional simulations of laminar vortex shedding behind a circular cylinder at  $\text{Re} = \frac{\rho U_\infty d}{\mu} = 185$  were performed to test the time-level preconditioner presented in this work. The computational domain used for all simulations was constructed using a cylindrical o-grid with an outer diameter of  $100d$ , 280 cells in the radial direction and 320 cells in the circumferential direction. As shown in Fig. 9, these cells were distributed to ensure that the majority of them were situated in the wake region and close to the cylinder wall. The size of the cells in the wall normal direction at the cylinder surface was further set to  $0.0075d$ .

The vortex shedding simulations were performed by solving the compressible Navier-Stokes equations in conservative form using the G3D::Flow code developed by the present authors. G3D::Flow uses a finite volume discretization method, in which convective and viscous fluxes at cell faces are reconstructed with different flux schemes. The third order upwind scheme with lower dissipation (TOU-LD) presented earlier was selected for the convective fluxes and a centered difference scheme of second order accuracy was used for the viscous fluxes. The cylinder was modeled as a no-slip wall. At the far field boundary, a wave-transmissive freestream condition based on Riemann-characteristics was imposed. The freestream state was set to give the desired Reynolds number and a Mach number of 0.2. This Mach number was selected to give a virtually incompressible flow, while at the same time avoiding bad convergence due to a large discrepancy between acoustic and convective propagation speeds.

Before the HB simulations were run, a reference time-accurate simulation was performed with the three stage Runge-Kutta scheme presented in Table 2a. This simulation was performed with a nondimensional time step of  $\frac{U_\infty \Delta t}{d} = 0.00026$  and gave a Strouhal number of  $\text{St} = 0.192$ . This Strouhal number is compared to results published by other authors in Table 3. The frequency of the vortex shedding obtained from the time-accurate simulation was



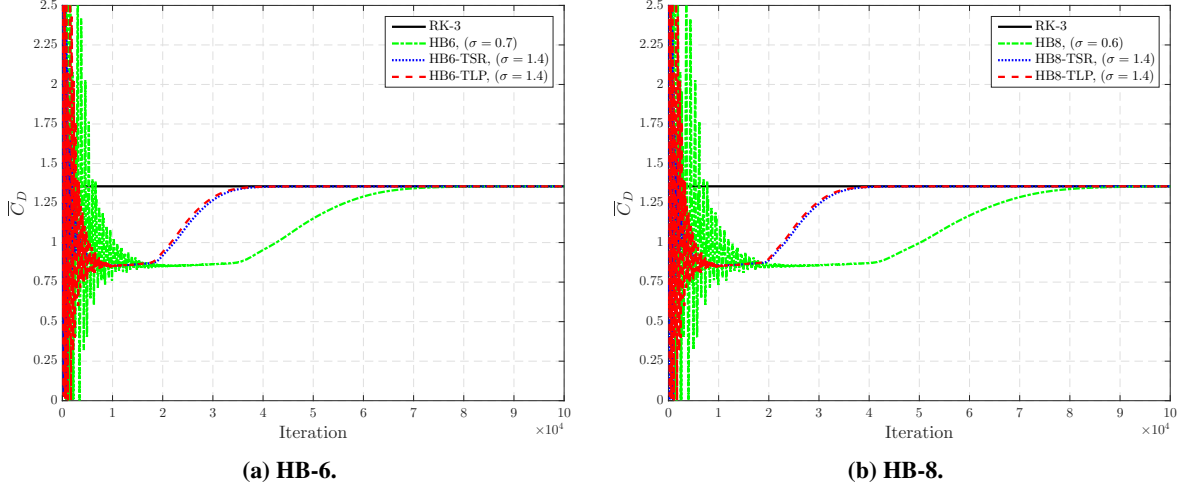
**Fig. 9 Close up view of the computational mesh.**

used as the fundamental frequency in the HB computations. Two sets of HB computations were performed, one with  $K = 6$  harmonics and one with  $K = 8$  harmonics. For each of these sets, three different setups were used, one without any preconditioning, one with the pseudo-time step restriction proposed by van der Weide et al. [13] and one with the time-level preconditioner presented in this work. In all simulations, the three stage Runge-Kutta scheme presented in Table 2a was used for pseudo-time integration using the largest CFL number that ensured convergence. Apart from local time stepping, no other acceleration methods such as multi-grid or implicit residual smoothing were used to ensure that the effects of the pseudo-time step restriction/time level preconditioner could be isolated. A summary of the results obtained, including results reported by other authors, is provided in Table 3. From this table, it can be seen that the results obtained with the G3D::Flow code agree well with previously published results. It is also interesting to note that the drag and lift coefficients obtained from the HB simulations approach those obtained with the time-accurate simulation as the number of harmonics increases. The results obtained for a fixed number of harmonics do also not depend on whether the preconditioner or pseudo-time step restriction is used, as should be expected.

	St	$\overline{C}_D$	$C_{L,rms}$
RK-3	0.192	1.356	0.456
HB-6, ( $\sigma = 0.7$ )	–	1.355	0.454
HB-6-TSR, ( $\sigma = 1.4$ )	–	1.355	0.454
HB-6-TLP, ( $\sigma = 1.4$ )	–	1.355	0.454
HB-8, ( $\sigma = 0.6$ )	–	1.356	0.456
HB-8-TSR, ( $\sigma = 1.4$ )	–	1.356	0.456
HB-8-TLP, ( $\sigma = 1.4$ )	–	1.356	0.456
Lu and Dalton [26]	0.195	1.31	0.422
Guilmineau and Queutey [27]	0.195	1.287	0.443

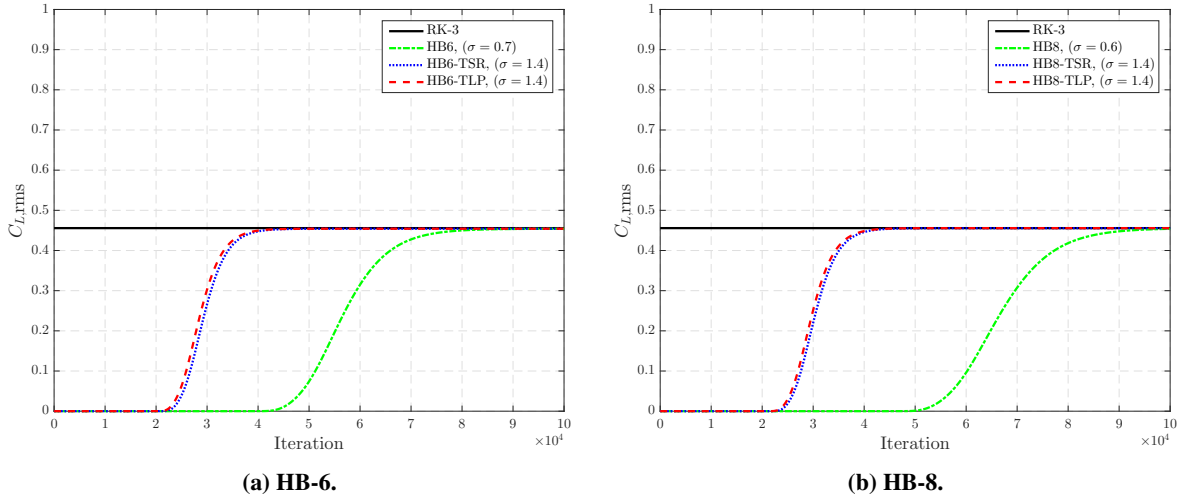
**Table 3 Summary of results obtained with time-accurate solver, with HB solver and by other authors. TSR = pseudo-time step restriction proposed by van der Weide et al. [13], TLP = time-level preconditioner presented in this work.**

The convergence history of the drag coefficient is presented for all the HB simulations in Fig. 10. In this figure, the drag coefficient obtained from the time accurate simulation has also been included for reference. From Fig. 10, it can be seen that both the pseudo-time step restriction proposed by van der Weide et al. [13] and the time-level preconditioner presented in this work give faster convergence than an HB simulation that only uses the CFL number to control the pseudo-time step.



**Fig. 10** Convergence history of  $\overline{C}_D$  for different numerical setups. TSR = pseudo-time step restriction proposed by van der Weide et al. [13], TLP = time-level preconditioner presented in this work.

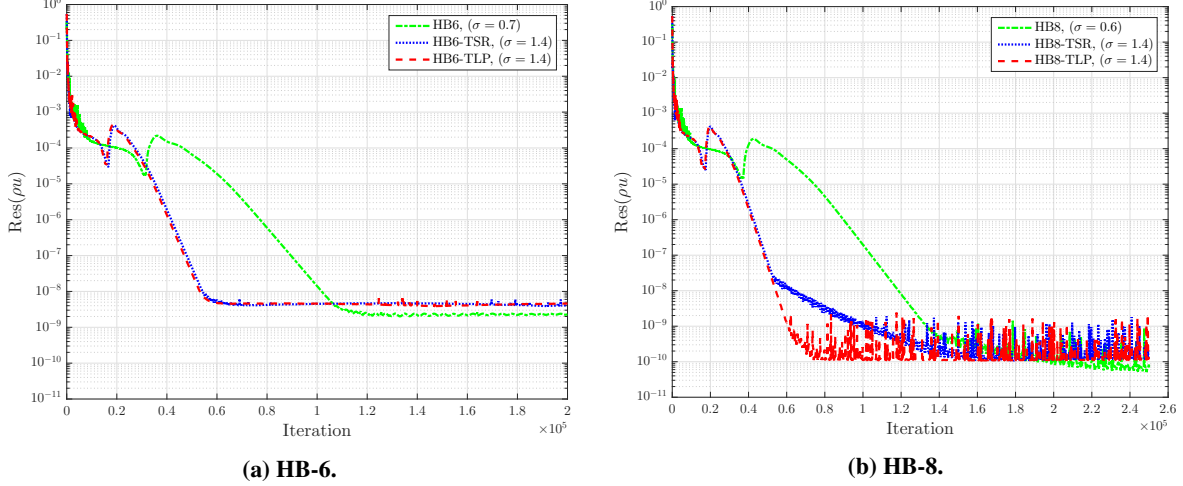
The convergence history of the lift coefficient is presented for all the different HB simulations in Fig. 11. In this figure, the lift coefficient obtained from the time accurate simulation has also been included for reference. The trends in Fig. 11 very much resemble those shown previously for the drag coefficient. This shows that both the pseudo-time step restriction proposed by van der Weide et al. [13] and the time-level preconditioner presented in this work can give better convergence than when the CFL number simply is used to determine the local pseudo-time step.



**Fig. 11** Convergence history of  $C_{L,rms}$  for different numerical setups. TSR = pseudo-time step restriction proposed by van der Weide et al. [13], TLP = time-level preconditioner presented in this work.

The convergence histories of the momentum residual for all HB simulations performed are finally shown in Fig. 12. This figure shows that the convergence of the HB method greatly benefits from employing either the pseudo-time step restriction or the time-level preconditioner. The oscillations in the residual seen in Fig. 12b are further thought to originate from some small amplitude oscillations in pseudo-time that is not efficiently damped by the low-dissipation TOU-LD scheme.

The reason why the convergence suffers when the CFL numbers is used to determine the pseudo-time step is that it must be selected to ensure that the integration of the governing equations remains stable in all computational cells. This implies that cells in the far field, which typically are large and thus have a large grid reduced frequency, must be accounted for when the CFL number is selected. If on the other hand the pseudo-time step restriction or



**Fig. 12** Convergence history of  $\text{Res}(\rho u)$  for different numerical setups. TSR = pseudo-time step restriction proposed by van der Weide et al. [13], TLP = time-level preconditioner presented in this work.

time-level preconditioner is employed, a local adjustment of the pseudo-time step will be done in each cell based on the grid-reduced frequency in that cell. This means that close to the cylinder, where the grid reduced frequency is significantly smaller, these two methods give a less restrictive scaling of the residual. Given that most flow variations also occur in the well resolved region close to the cylinder, it follows that the time-level preconditioner and pseudo-time step restriction should give better convergence as well.

The assumption that the grid-reduced frequency is small in the cells close to the cylinder can further be proven as follows: Assume that the flow only contains small amplitude perturbations that propagate through convection. Assume further that the mesh has been designed to give  $n_p$  points per wavelength for the highest frequency perturbation in the flow. Given that this perturbation propagates at speed  $\lambda_p$ , it then follows that the largest grid reduced frequency becomes

$$\frac{\omega_K \Delta x}{\lambda_p} = \frac{2\pi}{n_p} \quad (43)$$

This equation shows that the grid-reduced frequency will be small in well resolved regions ( $n_p$  large).

## V. Conclusion

In this work, a thorough stability analysis of the Harmonic Balance method has been performed. This stability analysis showed that the time spectral derivative source term employed in the Harmonic Balance method adds an imaginary part to the eigenvalue spectra of the spatial discretization, as was previously shown by McMullen [5]. The imaginary part added by the time spectral derivative was also shown to be proportional to the grid reduced frequency introduced by Gentili [16]. Given that explicit pseudo-time integration schemes such as, e.g., the Runge-Kutta method, only are stable when the eigenvalue spectra lies in a limited region in the complex plane, the present stability analysis helps to explain why explicit Harmonic Balance solvers can encounter stability issues when the grid reduced frequency is large, as was previously shown by Gentili [16].

Three approaches for stabilizing the Harmonic Balance method, namely the point implicit Runge-Kutta method [9, 12], the Harmonic Balance source term stabilization matrix [28], and the pseudo-time step restriction proposed in [13], have also been investigated using stability analysis. Although these methods have been proven valuable in the particular solvers used in the cited references, only the pseudo-time step restriction proposed in [13] provided a generic stabilization approach for the numerical methods tested in this work (explicit Runge-Kutta pseudo-time integration and third order upwind finite volume schemes).

Based on knowledge gained from the stability analysis, a time-level preconditioner was also developed. This preconditioner can be seen as a tailored version of the pseudo-time step restriction proposed by van der Weide et al. [13] in the sense that the preconditioner updates each frequency in the solution with a different pseudo-time step, and that the pseudo-time step is chosen to fit the spatial discretization scheme used. The advantage of employing

either the pseudo-time step restriction developed by van der Weide et al. [13] or the time-level preconditioner is also demonstrated by simulating two dimensional laminar vortex shedding behind a circular cylinder. For this case, both aforementioned methods give a speed up factor of 2 compared to when the CFL number simply is used to control the size of the pseudo-time step. The reason for this speed up is that the pseudo-time step restriction and the time-level preconditioner can use a larger local-time step in each cell by accounting for the local grid-reduced frequency. Although the time-level preconditioner has a theoretical advantage over the pseudo-time step restriction, no significant advantage is observed for the laminar vortex shedding test-case. The reason for this is thought to originate from the fact that most variations in the flow occur in regions where the grid is well resolved, and consequently, the grid-reduced frequency is small. If on the other hand multi-grid and/or a less well resolved grid is employed, the time-level preconditioner is expected to give a larger advantage. Although the time-level preconditioner was developed and demonstrated for explicit pseudo-time integration schemes, it could also be used to improve convergence of implicit schemes by increasing the diagonal dominance of the linear system.

### Acknowledgments

The authors would like to acknowledge the Swedish National Infrastructure for Computing (SNIC) for providing computer resources at the Chalmers Centre for Computational Science and Engineering (C<sup>3</sup>SE) in Gothenburg, Sweden

### References

- [1] He, L., "Fourier Methods for Turbomachinery Applications," *Progress in Aerospace Sciences*, Vol. 46, No. 8, 2010, pp. 329–341. doi:<https://doi.org/10.1016/j.paerosci.2010.04.001>.
- [2] Krack, M., and Groß, J., *Harmonic Balance for Nonlinear Vibration Problems*, Mathematical Engineering, Springer International Publishing, Gewerbestrasse 11, 6330, Cham, Switzerland, 2019. doi:10.1007/978-3-030-14023-6.
- [3] He, L., and Ning, W., "Efficient Approach for Analysis of Unsteady Viscous Flows in Turbomachines," *AIAA Journal*, Vol. 36, No. 11, 1998, pp. 2005–2012. doi:10.2514/2.328.
- [4] Hall, K. C., Jeffrey, P. T., and Clark, W. S., "Computation of Unsteady Nonlinear Flows in Cascades Using a Harmonic Balance Technique," *AIAA Journal*, Vol. 40, No. 5, 2002, pp. 879–886. doi:10.2514/2.1754.
- [5] McMullen, M. S., "The Application of Nonlinear Frequency Domain Methods to the Euler and Navier-Stokes Equations," Ph.D. thesis, Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, 2003.
- [6] Hall, K. C., Ekici, K., Thomas, J. P., and Dowell, E. H., "Harmonic Balance Methods Applied to Computational Fluid Dynamics Problems," *International Journal of Computational Fluid Dynamics*, Vol. 27, No. 2, 2013, pp. 52–67. doi:10.1080/10618562.2012.742512.
- [7] Frey, C., Kersken, H.-P., Ashcroft, G., and Voigt, C., "A Harmonic Balance Technique for Multistage Turbomachinery Applications," *ASME Turbo Expo 2014: Turbine Technical Conference and Exposition*, GT2014-25230, 2014. doi:10.1115/GT2014-25230.
- [8] Sicot, F., Dufour, G., and Gourdain, N., "A Time-Domain Harmonic Balance Method for Rotor/Stator Interactions," *Journal of Turbomachinery*, Vol. 134, No. 1, 2012, pp. 011001–1–011001–13. doi:10.1115/1.4003210.
- [9] Campobasso, M. S., and Baba-Ahmadi, M. H., "Analysis of Unsteady Flows Past Horizontal Axis Wind Turbine Airfoils Based on Harmonic Balance Compressible Navier-Stokes Equations With Low-Speed Preconditioning," *Journal of Turbomachinery*, Vol. 134, No. 6, 2012, pp. 061020–1–011001–13. doi:10.1115/1.4006293.
- [10] Huang, H., and Ekici, K., "An Efficient Harmonic Balance Method for Unsteady Flows in Cascades," *Aerospace Science and Technology*, Vol. 29, No. 1, 2024, pp. 144–154. doi:<https://doi.org/10.1016/j.ast.2013.02.004>.
- [11] McMullen, M. S., Jameson, A., and Alonso, J. J., "Acceleration of Convergence to a Periodic Steady State in Turbomachinery Flows," *39th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2001–0152, 2001. doi:10.2514/6.2001-152.
- [12] McMullen, M. S., and Jameson, A., "The Computational Efficiency of Non-Linear Frequency Domain Methods," *Journal of Computational Physics*, Vol. 212, No. 2, 2006, pp. 637–661. doi:<https://doi.org/10.1016/j.jcp.2005.07.021>.
- [13] van der Weide, E., Gopinath, A. K., and Jameson, A., "Turbomachinery Applications with the Time Spectral Method," *35th AIAA Fluid Dynamics Conference and Exhibit*, AIAA Paper 2005-4905, 2005. doi:10.2514/6.2005-4905.

- [14] Su, X., and Yuan, X., “Implicit solution of time spectral method for periodic unsteady flows,” *International Journal for Numerical Methods in Fluids*, Vol. 63, No. 7, 2010, pp. 860–876. doi:10.1002/fld.2111.
- [15] Guédeney, T., Gomar, A., Gallard, F., Sicot, F., Dufour, G., and Puigt, G., “Non-uniform time sampling for multiple-frequency harmonic balance computations,” *Journal of Computational Physics*, Vol. 236, 2013, pp. 317–345. doi:https://doi.org/10.1016/j.jcp.2012.11.010.
- [16] Gentilli, N. C., “Efficient Solution of Unsteady Nonlinear Flows Using a Multiple Zone Harmonic Balance Technique,” Master’s thesis, Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC, December 2010.
- [17] Thomas, J. P., Custer, C. H., Dowell, E. H., Hall, K. C., and Corre, C., “Compact Implementation Strategy for a Harmonic Balance Method Within Implicit Flow Solvers,” *AIAA Journal*, Vol. 51, No. 6, 2013, pp. 1374–1381. doi:10.2514/1.J051823.
- [18] Woodgate, M. A., and Badcock, K. J., “Implicit Harmonic Balance Solver for Transonic Flow with Forced Motions,” *AIAA Journal*, Vol. 47, No. 4, 2009, pp. 893–901. doi:10.2514/1.36311.
- [19] Antheaume, S., and Corre, C., “Implicit Time Spectral Method for Periodic Incompressible Flows,” *AIAA Journal*, Vol. 49, No. 4, 2011, pp. 791–805. doi:10.2514/1.J050785.
- [20] Custer, C. H., “A Nonlinear Harmonic Balance Solver for an Implicit CFD Code: OVERFLOW 2,” Ph.D. thesis, Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC, 2009.
- [21] Sicot, F., Puigt, G., and Montagnac, M., “Block-Jacobi Implicit Algorithms for the Time-Spectral Method,” *AIAA Journal*, Vol. 46, No. 13, 2008, pp. 3080–3089. doi:10.2514/1.36792.
- [22] Wang, D. X., and Huang, X., “Solution Stabilization and Convergence Acceleration for the Harmonic Balance Equation System,” *Journal of Engineering for Gas Turbines and Power*, Vol. 139, No. 9, 2017, pp. 092503–2–092503–9. doi:10.1115/1.4035912.
- [23] Andersson, N., “A Study of Subsonic Turbulent Jets and Their Radiated Sound Using Large-Eddy Simulation,” Ph.D. thesis, Division of Fluid Dynamics, Department of Applied Mechanics, Chalmers University of Technology, Gothenburg, Sweden, 2005.
- [24] Hirsch, C., *Numerical Computation of Internal and External Flows*, 2<sup>nd</sup> ed., Vol. 1, Butterworth-Heinemann, 2007. doi:https://doi.org/10.1016/B978-0-7506-6594-0.X5037-1.
- [25] Melson, N. D., Atkins, H. L., and Sanetrik, M. D., “Time-accurate Navier Stokes calculations with multigrid acceleration,” *The 6<sup>th</sup> Copper Mountain Conference on Multigrid Methods*, Part 2: pp. 423–437, 1993.
- [26] Lu, X.-Y., and Dalton, C., “Calculation of the Timing of Vortex Formation from an Oscillating Cylinder,” *Journal of Fluids and Structures*, Vol. 10, No. 5, 1996, pp. 527–541. doi:https://doi.org/10.1006/jfls.1996.0035.
- [27] Guilmineau, E., and Queutey, P., “A Numerical Simulation of Vortex Shedding from an Oscillating Circular Cylinder,” *Journal of Fluids and Structures*, Vol. 16, No. 6, 2002, pp. 773–794. doi:https://doi.org/10.1006/jfls.2002.0449.
- [28] Thomas, J. P., Custer, C. H., Dowell, E. H., and Hall, K. C., “Unsteady Flow Computation Using a Harmonic Balance Approach Implemented about the OVERFLOW 2 Flow Solver,” *19th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2009-4270, 2009. doi:10.2514/6.2009-4270.