# Parametric reduced-order modelling of the unsteady vortex-lattice method

Salvatore Maraniello* and Rafael Palacios†

*Imperial College, London, SW7 2AZ, United Kingdom*

**A method for frequency-limited balancing of the unsteady vortex-lattice equations is introduced that results in compact models suitable for computational-intensive applications in load analysis, aeroelastic optimisation, and control synthesis. The balancing algorithm relies on a frequency-domain solution of the vortex-lattice equations that effectively eliminates the cost associated to the wake states. It is obtained from a Z-transform of the underlying discrete-time equations, and requires no additional geometrical or kinematic assumptions for the lifting surfaces. Parametric reduced-order modelling is demonstrated through interpolation over (a) projection matrices, (b) state-space realisations and (c) transfer functions, which trade accuracy, robustness and cost. Methods are finally exemplified in the dynamic stability of a T-tail configuration with varying incidence. Numerical studies show that a very small number of balanced realisations is sufficient to accurately capture the unconventional aeroelastic response of this system, which includes in-plane kinematics and steady loads, over a wide range of operation conditions.**

## Nomenclature

| | |
|---|---|
| $\gamma_i$ | interpolation weight for design parameter $p_i$ |
| $\mathbf{\Gamma}$ | vector of circulation strengths |
| $\kappa_i$ | integration weight at frequency $k_i$ |
| $\mathcal{K}$ | number of integration points, $k_i$ |

*Research Associate, Department of Aeronautics, Room CAGB 308

†Professor of Computational Aeroelasticity, Department of Aeronautics, Room CAGB 338; r.palacios@imperial.ac.uk. Associate Fellow AIAA

| | |
|---|---|
| $U$ | reference velocity |
| $\boldsymbol{\mathcal{A}}$ | aerodynamic influence coefficient matrix |
| $b$ | reference semi-chord |
| $\boldsymbol{G}$ | state-space model transfer function |
| $\boldsymbol{I}_q$ | identity matrix of size $q \times q$ |
| $k$ | reduced frequency |
| $\bar{k}$ | limit reduced frequency for frequency limited balancing |
| $n$ | discrete-time step index |
| $\boldsymbol{p}$ | design parameters |
| $t$ | reduced time |
| $\boldsymbol{T}$ | right projection matrix |
| $\boldsymbol{W}^\top$ | left projection matrix |

**Subscripts**

| | |
|---|---|
| $\boldsymbol{a}$ | full aerodynamic model |
| $\boldsymbol{b}$ | frequency-limited balanced model |
| $\boldsymbol{r}$ | frequency-limited balanced truncated model |

**Symbols and Mathematical Operators**

| | |
|---|---|
| col[•] | column-wise block-matrix concatenation |
| row[•] | row-wise block-matrix concatenation |
| $\bar{\bullet}$ | $z$-transform of a discrete-time variable |
| $\widetilde{\bullet}$ | quantities associated to an interpolated state-space model |

# I. Introduction

The unsteady vortex lattice method (UVLM) has established itself as a valuable analysis tool for low-speed aeroelasticity of very flexible aircraft, horizontal-axis wind turbines, and flapping fliers [1–3]. Its standard formulation is written in time domain and captures unsteady effects on the circulatory lift through explicit modelling of the wake [4]. It generalises the doublet-lattice method [5] in the incompressible regime by considering arbitrary kinematics of the lifting surfaces, capturing (inviscid) wake dynamics and steady aerodynamic effects, and by computing induced drag [6]. Linearisation around an arbitrary reference finally results in state-space descriptions suitable for control synthesis and design [7–9].

UVLM models of complex configurations typically require tens or even hundreds of thousands of states to define a converged wake [10–12]. While continuous-time formulations exist that allow for sub-sampling of the far wake, a flat-wake assumption has been so far implicitly required [13], which effectively restricts

the applicability of that approach. Alternatively, fast multipole methods [14] have also been successfully implemented for efficient evaluation of the aerodynamic influence coefficients in time-marching solutions, but they are not directly applicable to the linearised formulations needed for stability and control analysis. In such cases, standard methods for model-order reduction, such as internal balancing, can be directly employed. Balanced reduced-order models (ROMs) can have comparable order to the input/output dimensionality of the problem [8, 15–17]. They not only retain the modelling flexibility of the full-state model (e.g., to model arbitrary kinematics), but also are highly accurate and, importantly, preserve stability [8, 17].

Direct methods for internal balancing come however with the high computational cost associated to the evaluation of the system Gramians [18]. Efficient low-rank algorithms have been developed to overcome this, but they typically assume a continuous-time formulation [19], which is not available for a general UVLM. In discrete time, low-rank balancing relies either on projection techniques [20] or Smith's iteration [21, 22], with the latter been recently used by the authors to construct ROMs for the UVLM [17]. However, while this approach guarantees stability and accuracy and is more efficient than direct balancing algorithms [23–25], it still requires relatively large computational resources. Such approach is therefore impractical for situations that consider multiple reference conditions around which linearised aerodynamics may be needed, such as for application in design optimisation or control design [26–30]. In those problems, the dependency of the aerodynamics on the underlying system design parameters and/or operative conditions needs to be captured, which requires a large number of system realisations.

That problem is addressed in this work, which aims to construct an efficient framework for parametric model reduction of the linearised unsteady vortex-lattice (LUVLM) equations around an arbitrary reference. The starting-point is a discrete-time state-space description of the LUVLM in modal coordinates [17]. The most general implementation is considered: it accounts for arbitrary kinematics of the lifting surfaces and arbitrary inflow conditions, resolves all components of the aerodynamic forces [6], and considers arbitrary wake shapes (that is, "rolled-up" wakes) at the reference condition. That formulation will be here extended to frequency domain. No additional assumption is required for that purpose, as it arises naturally from a $z$-transform to the discrete-time equations [31]. As it will be seen, the $z$-transform allows a closed-form solution for the wake states, which results in an efficient algorithm for frequency-domain analysis. A similar approach has been recently developed by Dimitriadis *et al.* [32] for flat surfaces, who has also employed it to generate ROMs using rational-function approximations. Here, instead, the frequency-domain formulation will be exploited to construct a tailored low-rank balancing algorithm for the LUVLM.

Balanced realisations of the LUVLM are obtained from direct numerical integration, in frequency domain, of the observability and controllability Gramians [18, 33, 34]. In order to guarantee computational efficiency, these are resolved in low-rank factorised expressions and with closed-form solutions for the wake variables.

Overall, the approach has strong similarities with balanced proper orthogonal decomposition [35], though the square-root method is here employed in place of an eigen-decomposition [36, 37]. Importantly, numerical integration allows resolving the LUVLM equations directly at the low end of the reduced frequency spectrum, where the potential-flow assumption holds, i.e. resulting in a frequency-limited balancing algorithm [37–41]. Frequency-limited balanced truncation (FLBT) not only provides substantial computational savings, but also results in faster convergence rates than internally-balanced truncated ROMs [37], which unnecessary resolve the aerodynamics over the full Nyquist range [17].

A framework for parametric reduced-order modelling is finally achieved by integrating FLBT with ROM interpolation methods [42]. This extends the portability of the methodology to enable design optimisation or real-time aeroelastic control [27, 29, 43]. Several strategies are investigated. First, interpolation amongst projection matrices is considered [28] and a novel solution based on FLBT is proposed. While this is very accurate, it has only minor computational savings due to the need to re-assemble the full system matrices. Next, methodologies for direct ROM interpolation, which typically assume reduced models in orthonormal bases [43–45], are here adapted to FLBT. These allow real-time execution at the expense of accuracy, due to the need of projecting the ROMs over a common set of generalised coordinates — with unavoidable loss of information. Finally, a new strategy based on the transfer-function interpolation method is also developed [42, 46]. By exploiting the properties of FLBT, it directly produces realisations of the interpolated ROM, hence preventing the growth of the interpolated ROM number of states with the interpolation order.

The rest of this paper is structured as follows. The LUVLM time- and frequency-domain descriptions are introduced in Sec. II, where aeroelastic integration is also briefly discussed. The frequency-domain solution is then used to obtain a FLBT of the system in Sec. III and interpolation methods are subsequently considered in Sec. IV. Numerical investigations will finally assess the accuracy and robustness of both the FLBT (Sec. V.A) and the interpolation strategies (Sec. V.B). In both cases, ROMs are employed to analyse the aeroelastic stability of a T-tail configuration at different attitudes. The dynamic aeroelastic characteristics of this system are highly dependent on the steady aerodynamic loads acting on the tail at the linearisation point [9]. Furthermore, the model also needs to capture the in-plane kinematics associated to the fore-aft motion of the tail plane. Numerical studies will show that these features, which are naturally captured by the full-state LUVLM equations, are retained throughout both the model-order reduction and interpolation processes.

## II. Linear UVLM description in generalised coordinates

Incompressible potential-flow aerodynamics will be modelled by a discrete-time state-space UVLM formulation. It linearises the governing equations about an arbitrary reference, thus capturing the effect of steady deflections of the lifting surfaces, non-zero aerodynamic forces at the linearisation point, as well as arbitrary

wake shapes and background in-flow conditions. It also considers arbitrary kinematics of the lifting surfaces to include, e.g., in-plane displacements, and assumes the wake to be frozen in the subsequent perturbation dynamics. The description is based on linearisation of the three-dimensional form of the Joukowski theorem [6], which resolves all components of the aerodynamic forces, including (unsteady) induced drag, and is scaled with the dynamic pressure, such that solutions depend on the reference geometry and steady loading, but not on the airspeed. We assume here that structural inputs (perturbation of displacements and velocity of the lifting surfaces) and aerodynamic outputs are both given in generalised coordinates. Non-stationary background flows can be easily added as inputs [17], but they are not included here to simplify the description. Sec. II.A introduces the governing equations. Only features relevant to this work are outlined and the reader is referred to Ref. [17] for further details. In Sec. II.B a frequency-domain solution is introduced, including an algorithm for fast computation.

### II.A. Time-domain description

Normalised UVLM equations are used. A reference forward flight speed, $U$, and a reference semi-chord, $b$, are used to scale, respectively, velocities and displacements. This results in the usual scaling of time $b/U$ used in aeroelastic applications. Finally, the resulting aerodynamic forces are normalised with the reference dynamic pressure and the semi-chord, as $\frac{1}{2}\rho U^2 b^2$. An arbitrary number of (interfering) lifting surfaces, and their wakes, are considered, although for simplicity the description here is restricted to a single one, discretised with $K$ and $K_w$ quadrilateral vortex rings in wing and wake, respectively. We indicate with $S$ the total number of panels in the spanwise direction, while $R$ and $R_w$ are the bound and wake panels in chordwise direction, such that $K = RS$ and $K_w = R_w S$.

In the linearised model [17], the (incremental) circulation strengths of each panel in the bound and wake lattices are normalised with respect to $bU$ and collected into the vectors $\boldsymbol{\Gamma} \in \mathbb{R}^K$ and $\boldsymbol{\Gamma_w} \in \mathbb{R}^{K_w}$. In particular, if $r$ and $s$ are indices along the chordwise and spanwise directions, we define $j = J(r, s)$, the mapping used to organise $\boldsymbol{\Gamma}$ and $\boldsymbol{\Gamma_w}$. Furthermore, a frozen, but not necessarily flat, wake geometry is assumed. The kinematics of the lifting surface is arbitrary and described in terms of the perturbations to the displacements and translational velocities at the corner points of the lattice. They are obtained from a set of generalised coordinates and their velocities, $\boldsymbol{\eta} \in \mathbb{R}^{N_\eta}$ and $\boldsymbol{\eta}' \in \mathbb{R}^{N_\eta}$, respectively, where $(\bullet)'$ indicates a time derivative in the normalised time, $t$. $N_\eta$ is the number of generalized coordinates used to describe the kinematics, which is typically much smaller than the number of corner points in the discretisation of the lifting surfaces. The system inputs are thus $\boldsymbol{u_a}^\top = \left[\boldsymbol{\eta}^\top, \boldsymbol{\eta}'^\top\right] \in \mathbb{R}^{N_u}$ with $N_u = 2N_\eta$, while $\boldsymbol{y_a} \in \mathbb{R}^{N_y}$ with $N_y = N_\eta$ will define the outputs as the normalised generalised aerodynamic forces. The linearized

non-penetration boundary condition at the bound lattice is [17]

$$\boldsymbol{\mathcal{A}}\,\boldsymbol{\Gamma} + \boldsymbol{\mathcal{A}}_{\boldsymbol{w}}\,\boldsymbol{\Gamma}_{\boldsymbol{w}} = \boldsymbol{\mathcal{B}}\,\boldsymbol{u}_{\boldsymbol{a}} \tag{1}$$

where the aerodynamic influence coefficients matrices, $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{K\times K}$ and $\boldsymbol{\mathcal{A}}_{\boldsymbol{w}} \in \mathbb{R}^{K\times K_w}$, which define the induced velocities generated by bound and wake vortex rings, are obtained from the Biot-Savart law [4]. Matrix $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{K\times 2N_\eta}$ gives the (incremental) displacement and velocities on the lattice due to the structural dynamics. In order to predict the unsteady component of the aerodynamic force, the rate of change of the bound vorticity is also required. Here, a second-order finite-difference scheme is used

$$\Delta t\,\boldsymbol{\Gamma}'^{n+1} = \beta_i\,\boldsymbol{\Gamma}^{n+i}\,, \quad i = -1, 0, 1 \tag{2}$$

with $\beta_i$ are $\beta_1 = 3/2$, $\beta_0 = -2$ and $\beta_{-1} = 1/2$ and $n$ the current time step. This scheme was shown in Ref. [17] to result in smaller overall system size for a given tolerance than a first-order scheme. Unsteadiness in the fluid response is introduced by propagating the circulation strength from the trailing-edge into the wake using a first-order, explicit, time-stepping scheme [4], i.e. shifting the wake circulation downstream by one chordwise element at each time step. This results into $K_w$ lag operators

$$\Gamma^{n+1}_{w(r,s)} = \Gamma^{n}_{(R,s)}\,, \quad r = 1 \quad \text{and} \tag{3a}$$

$$\Gamma^{n+1}_{w(r,s)} = \Gamma^{n}_{w(r-1,s)}\,, \quad 1 < r \leq R_w, \tag{3b}$$

which are decoupled in the index $s$. This scheme links the normalised time step to the streamwise length of the wake panels [10]. In matrix form, Eq. (3) results into $\boldsymbol{\Gamma}^{n+1}_{\boldsymbol{w}} = \boldsymbol{\mathcal{C}}_{\boldsymbol{\Gamma}}\,\boldsymbol{\Gamma}^{n} + \boldsymbol{\mathcal{C}}_{\boldsymbol{\Gamma}_{\boldsymbol{w}}}\,\boldsymbol{\Gamma}^{n}_{\boldsymbol{w}}$, where $\boldsymbol{\mathcal{C}}_{\boldsymbol{\Gamma}} \in \mathbb{N}^{K_w\times K}$ and $\boldsymbol{\mathcal{C}}_{\boldsymbol{\Gamma}_{\boldsymbol{w}}} \in \mathbb{N}^{K_w\times K_w}$ are highly sparse matrices.

Equations (1)-(3) describe the LUVLM in terms of the state $\boldsymbol{x_a} = \mathrm{col}\big[\boldsymbol{\Gamma}^n, \boldsymbol{\Gamma}^n_{\boldsymbol{w}}, \Delta t\boldsymbol{\Gamma}'^n, \boldsymbol{\Gamma}^{n-1}\big] \in \mathbb{R}^{N_x}$, where col[•] indicates a column-wise concatenation and $N_x = 3K + K_w$. The description above naturally allows for the *staggered solution* commonly found in UVLM implementations [4, 10]. Namely the wake circulation at time step $n+1$ is found first through Eq. (3), while subsequent enforcement of Eq. (1) and (2) provides $\boldsymbol{\Gamma}^{n+1}$ and $\Delta\boldsymbol{\Gamma}^{n+1}$. The LUVLM equations can also be cast in discrete-time state-space form as

$$\boldsymbol{x}^{n+1}_{\boldsymbol{a}} = \boldsymbol{A_a}\,\boldsymbol{x}^{n}_{\boldsymbol{a}} + \boldsymbol{B_a}\boldsymbol{u}^{n+1}_{\boldsymbol{a}} \tag{4}$$

where

$$\boldsymbol{A_a} = \begin{bmatrix} \boldsymbol{\mathcal{P}} & \boldsymbol{\mathcal{P}_w} & 0 & 0 \\ \boldsymbol{\mathcal{C}_\Gamma} & \boldsymbol{\mathcal{C}_{\Gamma_w}} & 0 & 0 \\ \beta_1 \boldsymbol{\mathcal{P}} + \beta_0 \boldsymbol{I}_K & \beta_1 \boldsymbol{\mathcal{P}_w} & 0 & \beta_{-1} \boldsymbol{I}_K \\ \boldsymbol{I}_K & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{B_a} = \begin{bmatrix} \boldsymbol{\mathcal{A}}^{-1}\boldsymbol{\mathcal{B}} \\ 0 \\ \beta_1 \boldsymbol{\mathcal{A}}^{-1}\boldsymbol{\mathcal{B}} \\ 0 \end{bmatrix} \tag{5}$$

with

$$\boldsymbol{\mathcal{P}} = -\boldsymbol{\mathcal{A}}^{-1}\boldsymbol{\mathcal{A}_w}\boldsymbol{\mathcal{C}_\Gamma} \quad \text{and} \quad \boldsymbol{\mathcal{P}_w} = -\boldsymbol{\mathcal{A}}^{-1}\boldsymbol{\mathcal{A}_w}\boldsymbol{\mathcal{C}_{\Gamma_w}}. \tag{6}$$

This form allows convenient manipulation of the LUVLM equations in feedback interconnection with structural models or to employ black-box ROM strategies [16, 17, 47]. However, it is also less efficient for time-marching solutions, as evaluation of $\Delta\boldsymbol{\Gamma}'$ requires large matrix-vector multiplications. In Sec. II.B and III this consideration will be extended to frequency-domain analysis and model-order reduction to build efficient algorithms that exploit the underlying structure of the discretised system.

Perturbation of the aerodynamic forces on the lifting surfaces is finally obtained using the Joukowski method [6]. Those are finally projected on the generalised forces, $\boldsymbol{y_a}$, associated to the generalised coordinates $\boldsymbol{\eta}$ used to describe the structural dynamics. This results into the output equation [17]

$$\boldsymbol{y_a} = \boldsymbol{C_a}\,\boldsymbol{x_a} + \boldsymbol{D_a}\,\boldsymbol{u_a} \tag{7}$$

where the output matrix will be written as $\boldsymbol{C_a} = \begin{bmatrix} \boldsymbol{C_{ab}} & \boldsymbol{C_{aw}} & \boldsymbol{C_{a\Delta}} & 0 \end{bmatrix}$, with quasi-steady contributions in $\boldsymbol{C_{ab}}, \boldsymbol{C_{aw}}$ and $\boldsymbol{D_a}$ arising from applying Joukowski theorem to each of the segments of the lattice. These capture induced-drag and leading-edge suction effects. The unsteady components, in $\boldsymbol{C_{a\Delta}}$ and $\boldsymbol{D_a}$, are calculated applying Bernoulli's equation across the surface.

For consistency with the previous normalisation, the structural dynamics is also written in reduced time $t$. We assume a linear structure about a non-zero equilibrium, with dynamics given as

$$\boldsymbol{M}\frac{\mathrm{d}^2\boldsymbol{\eta}}{\mathrm{dt}^2} + \frac{b^2}{U^2}\left(\boldsymbol{K} + \tfrac{1}{2}\rho U^2 b^2 \boldsymbol{K_a}\right)\boldsymbol{\eta} = \frac{b^2}{U^2}\,\boldsymbol{y_a} \tag{8}$$

where $\boldsymbol{M}$ and $\boldsymbol{K}$ are generalised mass and stiffness matrices, and $\boldsymbol{K_a}$ is a stiffening term arising from linearisation about non-zero reference, that is, a follower-force effect of the steady aerodynamics. Note finally that in Eq. (8) the dimensional Nyquist frequency goes as $\omega_N \approx U/b$, that is, aeroelastic coupling implies a maximum chordwise size in the LUVLM lattice discretisation (which defines the normalised time-step) to capture a required vibration spectrum. This condition is also present in DLM-based aeroelasticity.

### II.B. Frequency-domain description

State equations (1), (3) and (2), can be conveniently transformed through the $z$-transform as

$$\boldsymbol{\mathcal{A}}\,\overline{\boldsymbol{\Gamma}} + \boldsymbol{\mathcal{A}}_{\boldsymbol{w}}\,\overline{\boldsymbol{\Gamma}}_{\boldsymbol{w}} = \boldsymbol{\mathcal{B}}\,\overline{\boldsymbol{u}}_{\boldsymbol{a}} \tag{9a}$$

$$z\,\overline{\boldsymbol{\Gamma}}_{\boldsymbol{w}} = \boldsymbol{\mathcal{C}}_{\boldsymbol{\Gamma}}\,\overline{\boldsymbol{\Gamma}} + \boldsymbol{\mathcal{C}}_{\boldsymbol{\Gamma}_{\boldsymbol{w}}}\overline{\boldsymbol{\Gamma}}_{\boldsymbol{w}} \tag{9b}$$

$$z\,\Delta\overline{\boldsymbol{\Gamma}}' = \sum_{i=-1}^{1} \beta_i\, z^i\,\overline{\boldsymbol{\Gamma}} \tag{9c}$$

where the symbol $(\bar{\bullet})$ indicates the $z$-transform of a discrete-time variable. From Eq. (9) a description in frequency domain is generated by simply setting $z = e^{jk\Delta t}$, where $k$ is a reduced frequency associated to the scaling factor $U/b$. In order to solve system (9), we could first compute $\overline{\boldsymbol{\Gamma}}_{\boldsymbol{w}}$ from (9b), but this leads to the inversion of a large $K_w \times K_w$ complex matrix, $(z\boldsymbol{I}_{Kw} - \boldsymbol{\mathcal{C}}_{\boldsymbol{\Gamma}_{\boldsymbol{w}}})^{-1}$. Instead, a closed-form solution is found in a similar manner to Eq. (3). For that, consider that at each row $s$ of the wake the solution only depends on the trailing-edge circulation, $\bar{\Gamma}_{J(R,s)}$, as

$$\bar{\Gamma}_{wJ(r,s)} = z^{-r}\,\bar{\Gamma}_{J(R,s)}\,, \quad r \leq R_w. \tag{10}$$

Eq. (10) is expressed in matrix form as $\overline{\boldsymbol{\Gamma}}_{\boldsymbol{w}} = \overline{\boldsymbol{\mathcal{C}}}_{\boldsymbol{w}}(z)\,\overline{\boldsymbol{\Gamma}}$ where $\overline{\boldsymbol{\mathcal{C}}}_{\boldsymbol{w}}(z) \in \mathbb{C}^{K_w \times K}$ is a very sparse matrix. Substitution into (9a), finally leads to

$$\overline{\boldsymbol{\Gamma}} = \left[\boldsymbol{\mathcal{A}} + \boldsymbol{\mathcal{A}}_{\boldsymbol{w}}\,\overline{\boldsymbol{\mathcal{C}}}_{\boldsymbol{w}}(z)\right]^{-1}\,\boldsymbol{\mathcal{B}}\,\overline{\boldsymbol{u}}_{\boldsymbol{a}} \tag{11}$$

which only requires a $K \times K$ matrix inversion to compute the bound circulation. Through Eq. (9c) and (10) this allows retrieving the aerodynamic state $z$-transform as

$$\overline{\boldsymbol{x}}_{\boldsymbol{a}} = \mathrm{col}\Big[\,\overline{\boldsymbol{\Gamma}},\, \overline{\boldsymbol{\Gamma}}_{\boldsymbol{w}},\, \Delta\overline{\boldsymbol{\Gamma}}',\, z^{-1}\overline{\boldsymbol{\Gamma}}\,\Big]. \tag{12}$$

With reference to the balancing algorithm that will be discussed in Sec. III, it is important to notice that an equivalent expression for $\overline{\boldsymbol{\Gamma}}$ can also be retrieved starting from the state-space form in Eq. (4) as

$$\overline{\boldsymbol{\Gamma}} = z\,\left[z\,\boldsymbol{I}_K - \boldsymbol{\mathcal{P}} - \boldsymbol{\mathcal{P}}_{\boldsymbol{w}}\,\overline{\boldsymbol{\mathcal{C}}}_{\boldsymbol{w}}(z)\right]^{-1}\boldsymbol{\mathcal{A}}^{-1}\boldsymbol{\mathcal{B}}\,\overline{\boldsymbol{u}}_{\boldsymbol{a}} = z\,\boldsymbol{\mathcal{K}}^{-1}(z)\,\boldsymbol{\mathcal{A}}^{-1}\boldsymbol{\mathcal{B}}\,\overline{\boldsymbol{u}}_{\boldsymbol{a}}. \tag{13}$$

Eq. (13) has comparable computational cost than Eq. (11), as $\boldsymbol{\mathcal{A}}^{-1}\boldsymbol{\mathcal{B}}$ is precomputed during the assembly of $\boldsymbol{B}_{\boldsymbol{a}}$ in Eq. (4). Matrix $\boldsymbol{\mathcal{K}}^{-1}(z) \in \mathbb{C}^{K \times K}$ will later appear in the integration of the Gramians associated to the system defined above. From Eq. (12) the generalised force response is simply obtained as $\overline{\boldsymbol{y}}_{\boldsymbol{a}} = \boldsymbol{C}_{\boldsymbol{a}}\overline{\boldsymbol{x}}_{\boldsymbol{a}}$.

Setting $\overline{\boldsymbol{u}}_{\boldsymbol{a}} = \boldsymbol{I}_{2N_\eta}$ in Eq. (13) results in algorithm 1 for evaluation of the LUVLM frequency response. Compared to methods based on the direct manipulation of the (large) $\boldsymbol{A}_{\boldsymbol{a}}$ matrix in Eq. (4), this approach is computational and memory efficient. Firstly, at each reduced frequency, $k$, only a small $K \times K$ inversion is required. Secondly, only the influence coefficient matrices $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{A}}_{\boldsymbol{w}}$ are allocated and manipulated. Finally, Eq. (9c) can be dropped if the derivatives are written as $(jk\Delta t)\overline{\boldsymbol{\Gamma}}$.

**Algorithm 1** Fast UVLM frequency response

1: define an array of solution frequencies, $\boldsymbol{k}$.
2: **for** $k$ in $\boldsymbol{k}$
3: $z = e^{jk\Delta t}$.
4: build $\overline{\boldsymbol{\mathcal{C}}}_{\boldsymbol{w}}(z)$, Eq. (10) .
5: solve Eq. (11) or (13) for the bound circulation, $\overline{\boldsymbol{\Gamma}}(z)$, with $\overline{\boldsymbol{u}}_{\boldsymbol{a}} = \boldsymbol{I}_{2N_\eta}$.
6: build the state response, $\overline{\boldsymbol{x}}_{\boldsymbol{a}}(z)$, through Eq. (9c), (10) and (12).
7: retrieve generalised forces response $\overline{\boldsymbol{y}}_{\boldsymbol{a}}(z) = \boldsymbol{C}_{\boldsymbol{a}}\overline{\boldsymbol{x}}_{\boldsymbol{a}}(z)$.

## III. Frequency-limited balancing

The formulation of Sec. II.B sets the basis for frequency-limited balancing of the LUVLM. For generic dynamic systems, this is often done by seeking low-rank solutions to a set of Lyapunov equations [34, 37, 39, 40]. However, the mathematical structure of the LUVLM facilitates a computational-efficient direct integration of the system Gramians, as it has been done in other contexts (e.g., balanced proper orthogonal decomposition) [33–35]. The frequency-limited Gramians associated to the discrete-time state-space model of Eq. (4) and (7) are defined as

$$\boldsymbol{W}_{\boldsymbol{c}} = \frac{1}{2k_N}\int_{-\bar{k}_c}^{\bar{k}_c} \boldsymbol{H}(k)\,\boldsymbol{B}_{\boldsymbol{a}}\boldsymbol{B}_{\boldsymbol{a}}^{\top}\,\boldsymbol{H}^{\star}(k)\,dk = \frac{1}{2k_N}\int_{-\bar{k}_c}^{\bar{k}_c} \boldsymbol{\Phi}_{\boldsymbol{c}}(k)\boldsymbol{\Phi}_{\boldsymbol{c}}^{\star}(k)\,dk \tag{14a}$$

$$\boldsymbol{W}_{\boldsymbol{o}} = \frac{1}{2k_N}\int_{-\bar{k}_o}^{\bar{k}_o} \boldsymbol{H}^{\star}(k)\,\boldsymbol{C}_{\boldsymbol{a}}^{\top}\boldsymbol{C}_{\boldsymbol{a}}\,\boldsymbol{H}(k)\,dk = \frac{1}{2k_N}\int_{-\bar{k}_o}^{\bar{k}_o} \boldsymbol{\Phi}_{\boldsymbol{o}}(k)\boldsymbol{\Phi}_{\boldsymbol{o}}^{\star}(k)\,dk \tag{14b}$$

where: $k_N = \pi/\Delta t$ is the reduced Nyquist frequency; $0 < \bar{k}_c \leq k_N$ and $0 < \bar{k}_o \leq k_N$ define the controllability and observability frequencies of interest, respectively; $\boldsymbol{H}(k) = \left(e^{jk\Delta t}\boldsymbol{I}_{Nx} - \boldsymbol{A}_{\boldsymbol{a}}\right)^{-1}$ and $\boldsymbol{H}^{\star}$ is its complex conjugate. Note that if $\bar{k}_c = \bar{k}_o = k_N$, that is the Gramians are resolved over the full Nyquist range, then internal balancing is retrieved. Integrals in Eq. (14) are approximated by the finite sums

$$\boldsymbol{W}_{\boldsymbol{c,o}} \approx \frac{1}{2k_N}\sum_{i=1}^{\mathcal{K}_{c,o}} \kappa_i \left[\boldsymbol{\Phi}_{\boldsymbol{c,o}}(k_i)\boldsymbol{\Phi}_{\boldsymbol{c,o}}^{\star}(k_i) + \boldsymbol{\Phi}_{\boldsymbol{c,o}}(-k_i)\boldsymbol{\Phi}_{\boldsymbol{c,o}}^{\star}(-k_i)\right] \tag{15}$$

where $\kappa_i$ are quadrature weights associated to $\mathcal{K}_{c,o}$ frequency points $k_i \in [0, \bar{k}_{c,o}]$ — the subindex $\bullet_{c,o}$ is used throughout this section to indicate expressions that apply to both the controllability and observability gramians. Both trapezoidal, over uniformly spaced sampling frequencies, and Gauss-Lobotto quadratures

are considered. Due to the symmetry of transfer functions, $\boldsymbol{\Phi_{c,o}}(k) = \boldsymbol{\Phi}^{\star}_{\boldsymbol{c,o}}(-k)$, each term of the summation in Eq. (15) is real. Hence, Eq. (15) can be equivalently expressed as [18]

$$\boldsymbol{W_{c,o}} \approx \boldsymbol{Z_{c,o}} \boldsymbol{Z}^{\top}_{\boldsymbol{c,o}} \quad \text{with} \quad \boldsymbol{Z_{c,o}} = \text{row}\left[\sqrt{\frac{\kappa_i}{k_N}}\left[\Re\left\{\boldsymbol{\Phi_{c,o}}(k_i)\right\}, \Im\left\{\boldsymbol{\Phi_{c,o}}(k_i)\right\}\right]\right] \tag{16}$$

where row[$\bullet$] defines a row-wise concatenation. Eq. (16) approximates the full-state LUVLM Gramians through factors $\boldsymbol{Z_c} \in \Re^{N_x \times (2N_u \mathcal{K}_c)}$ and $\boldsymbol{Z_o} \in \Re^{N_x \times (2N_y \mathcal{K}_o)}$. In practice, only a few points are necessary to approximate Eq. (15), that is, $\mathcal{K}_c, \mathcal{K}_o \ll N_x$, which results in $\boldsymbol{Z_{c,o}}$ being low rank. A projection over the frequency-limited balanced states, $\boldsymbol{x} = \boldsymbol{T}\boldsymbol{x_b}$, is finally obtained through the square-root method [48], as

$$\boldsymbol{T} = \boldsymbol{Z_c} \boldsymbol{V} \boldsymbol{\Sigma}^{-1/2} \in \Re^{N_x \times N_b} \quad \text{and} \quad \boldsymbol{W}^{\top} = \boldsymbol{\Sigma}^{-1/2} \boldsymbol{U}^{\top} \boldsymbol{Z}^{\top}_{\boldsymbol{o}} \in \Re^{N_b \times N_x} \tag{17}$$

where $N_b = 2\min\left(N_u \mathcal{K}_c,\, N_y \mathcal{K}_o\right)$ defines the number of balanced states, $\boldsymbol{W}^{\top}\boldsymbol{T} = \boldsymbol{I}_{Nb}$, and $\boldsymbol{U}$, $\boldsymbol{\Sigma} = \text{diag}(\boldsymbol{\sigma})$ and $\boldsymbol{V}$ derive from the thin SVD $\boldsymbol{Z}^{\top}_{\boldsymbol{o}} \boldsymbol{Z_c} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\top}$. The resulting state-space limited-frequency balanced model, $\boldsymbol{G_b}$, has system matrices $\boldsymbol{A_b} = \boldsymbol{W}^{\top}\boldsymbol{A_a}\boldsymbol{T}$, $\boldsymbol{B_b} = \boldsymbol{W}^{\top}\boldsymbol{B_a}$, $\boldsymbol{C_b} = \boldsymbol{A_a}\boldsymbol{T}$, and $\boldsymbol{D_b} = \boldsymbol{D_a}$. Reduced-order models are obtained by truncation of $\boldsymbol{G_b}$ and the resulting FLBT ROMs of dimension $N_r$ will be referred to as $\boldsymbol{G_r}$. FLBT achieves much faster convergence rates than internal balancing in the low-frequency range and therefore residualisation, which would require the inversion of a $N_b - N_r$ matrix, can often be avoided. While the FLBT does not provide a bounding criterion, the accuracy of the ROM can be directly assessed against the full system $\boldsymbol{G_a}$ via frequency-response comparisons at a minimal computational cost.

Contrary to internal balancing, approximation of the Gramians on a partial spectrum does not guarantee stability preservation [38]. While stability-preservation algorithms in FBLT ROMs are available [37, 39–41], they are based on manipulation of the Lyapunov equations associated to $\boldsymbol{W_{c,o}}$, and hence not applicable to our problem. An alternative solution has been found here building upon the methods for frequency-weighted balancing (FWB) [49]. FWB gives balanced realisations from serial connection of $\boldsymbol{G_a}$ with two finite-dimensional filters, one in input and one in output. Therefore, limited-frequency balacing is equivalent to FWB with ideal (infinite-dimensional) filters [18], while Eq. (14) is numerically equivalent to FWB with highly-resolved low-pass filters with cut-off frequencies $\bar{k}_c$ (in input) and $\bar{k}_o$ (in output). Importantly, FWB can be shown to preserve stability if one of the filters is removed [39, 49]. Therefore, with the purpose of improving the stability properties after reduction, the controllability Gramian, Eq. (14a), is integrated here in the full Nyquist range. As it will be discussed next, this choice is purely algorithmic, since manipulating the observability Grammian is more costly.

The complex factor $\boldsymbol{\Phi_c}(z) = \overline{\boldsymbol{x}}_{\boldsymbol{a}}(z)$ is efficiently computed using algorithm 1. In particular, its partition associated to the bound circulation is computed as per Eq. (13) through the small-matrix inversion

$\boldsymbol{\mathcal{K}}^{-1}(z)$. Next, $\boldsymbol{\Phi_o}(z)$ is obtained from solving $\boldsymbol{\Phi_o^\star}(z) = \boldsymbol{C_a}(z\boldsymbol{I}_{Nx} - \boldsymbol{A_a})^{-1}$ through block matrix operations. Introducing the partition $\boldsymbol{\Phi_o^\star} = \text{row}[\boldsymbol{\phi_b^\star}, \boldsymbol{\phi_w^\star}, \boldsymbol{\phi_\Delta^\star}, \boldsymbol{\phi_v^\star}]$, this results in the sequential solution process

$$\boldsymbol{\phi_\Delta^\star} = z^{-1}\,\boldsymbol{C_{a\Delta}} \tag{18a}$$

$$\boldsymbol{\phi_v^\star} = \beta_{-1}\,z^{-1}\,\boldsymbol{\phi_\Delta^\star} \tag{18b}$$

$$\boldsymbol{\phi_b^\star} = \boldsymbol{R}^\star(z)\,\boldsymbol{\mathcal{K}}^{-1}(z) \tag{18c}$$

$$\boldsymbol{\phi_w^\star} = \boldsymbol{S}^\star(z)\,\left(z\,\boldsymbol{I}_{Kw} - \boldsymbol{\mathcal{C}}_{\boldsymbol{\Gamma_w}}\right)^{-1} \tag{18d}$$

where $\boldsymbol{R}^\star \in \mathbb{C}^{N_\eta \times K}$ and $\boldsymbol{S}^\star \in \mathbb{C}^{N_\eta \times K_w}$ are defined as

$$\boldsymbol{R}^\star(z) = \boldsymbol{\phi_\Delta^\star}\,\left[\left(z^{-1}\beta_{-1} + \beta_0\right)\,\boldsymbol{I}_K + \beta_1\,\left(\boldsymbol{\mathcal{P}} + \boldsymbol{\mathcal{P}_w}\overline{\boldsymbol{\mathcal{C}}}_{\boldsymbol{w}}(z)\right)\right] + \boldsymbol{C_{aw}}\overline{\boldsymbol{\mathcal{C}}}_{\boldsymbol{w}}(z) + \boldsymbol{C_{ab}} \tag{19a}$$

$$\boldsymbol{S}^\star(z) = \left(\boldsymbol{\phi_b^\star} + \beta_1\,\boldsymbol{\phi_\Delta^\star}\right)\,\boldsymbol{\mathcal{P}_w} + \boldsymbol{C_{aw}} \tag{19b}$$

From a physical point of view, Eq. (18) isolates the generalized forces response due to perturbations on individual state variables. Importantly, inversion of the large matrix in Eq. (18d) can be avoided. Recalling the $J(r,s)$ mapping that links the chord and spanwise indices of a panel in the wake, $r$ and $s$ respectively, to an element in $\boldsymbol{\Gamma_w}$ (Sec. II.A), and being $i$ the index associated to the $i$th output force, Eq. (18d) is rewritten as

$$z\,\phi^\star_{w(i,J(r,s))} = S^\star_{(i,J(r,s))} + \phi^\star_{w(i,J(r+1,s))}\,, \quad r < R_w \tag{20a}$$

$$z\,\phi^\star_{w(i,J(r,s))} = S^\star_{(i,J(r,s))}\,, \quad r = R_w. \tag{20b}$$

As in Eq. (3) and (10), these equations are decoupled in the spanwise index, $s$, and can be solved recursively starting from the last row of the wake $r = R_w$. This results in the low-rank frequency-limited balancing algorithm 2. Firstly, a maximum frequency $\bar{k}$ is chosen to set the integration bounds in Eq. (14) as $k_o = \bar{k}$ and $k_c = k_N$. Note that in aeroelasticity it is $\bar{k} = \mathcal{O}(1)$ (and therefore $\bar{k} \ll k_N$). This defines low ($0 \leq k \leq \bar{k}$) and high-frequency ($\bar{k} < k \leq k_N$) regions. Quadrature schemes with $\mathcal{K}_{low}$ and $\mathcal{K}_{high}$ points are chosen for both regions, such that $\mathcal{K}_o = \mathcal{K}_{low}$ and $\mathcal{K}_c = \mathcal{K}_{low} + \mathcal{K}_{high}$ in Eq. (15). In the low-frequency range, $k \leq \bar{k}$, both Gramians are calculated at the same integration points such that a single matrix inversion, $\boldsymbol{\mathcal{K}}^{-1}$ in Eq. (13), is required per frequency. It is finally worth remarking that, in order to correctly obtain a realisation without

lag, the state-space equations (4) and (7) are recast as

$$\boldsymbol{h}_{\boldsymbol{a}}^{n+1} = \boldsymbol{A}_{\boldsymbol{a}}\, \boldsymbol{h}_{\boldsymbol{a}}^{n} + \boldsymbol{A}_{\boldsymbol{a}} \boldsymbol{B}_{\boldsymbol{a}}\, \boldsymbol{u}_{\boldsymbol{a}}^{n} \tag{21a}$$

$$\boldsymbol{y}_{\boldsymbol{a}} = \boldsymbol{C}_{\boldsymbol{a}}\, \boldsymbol{h}_{\boldsymbol{a}} + (\boldsymbol{C}_{\boldsymbol{a}} \boldsymbol{B}_{\boldsymbol{a}} + \boldsymbol{D}_{\boldsymbol{a}})\, \boldsymbol{u}_{\boldsymbol{a}} \tag{21b}$$

with $\boldsymbol{h}_{\boldsymbol{a}} = \boldsymbol{x}_{\boldsymbol{a}} - \boldsymbol{B}_{\boldsymbol{a}} \boldsymbol{u}_{\boldsymbol{a}}$ before projection. Contrary to other low-rank ROM methods, algorithm 2 does

**Algorithm 2** Low-rank frequency-limited balancing

1: set $\bar{k}$.
2: set quadrature scheme and order ($\mathcal{K}_{low}$) in $[0, \bar{k}]$.
3: set quadrature scheme and order ($\mathcal{K}_{high}$) in $[\bar{k}, k_N]$.
4: retrieve frequency array, $\boldsymbol{k} \in \Re^{\mathcal{K}_{low}+\mathcal{K}_{high}}$.
5: **for** $k$ in $\boldsymbol{k}$
6: calculate $\boldsymbol{\mathcal{K}}^{-1} = \left[e^{jk\Delta t}\, \boldsymbol{I}_K - \boldsymbol{\mathcal{P}} - \boldsymbol{\mathcal{P}}_{\boldsymbol{w}}\, \overline{\boldsymbol{\mathcal{C}}}_{\boldsymbol{w}}(e^{jk\Delta t})\right]^{-1}$.
7: solve for $\boldsymbol{\Phi}_{\boldsymbol{c}}(k)$ as per algorithm 1.
8: **if** $k \leq \bar{k}$
9: solve for $\boldsymbol{\Phi}_{\boldsymbol{o}}(k)$ using Eq. (18).
10: produce Gramians factors $\boldsymbol{Z}_{\boldsymbol{c},\boldsymbol{o}}$ as per Eq. (16).
11: calculate $\boldsymbol{T}$, $\boldsymbol{W}^{\top}$ through Eq. (17).
12: project state-space model defined by either Eq. (21) or Eq. (4) and (7).

not require manipulating the full $\boldsymbol{A}_{\boldsymbol{a}}$ matrix in Eq. (4). Instead, only the aerodynamic influence coefficient matrices, $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{A}}_{\boldsymbol{w}}$, need to be assembled, thus minimising memory requirements. As with Algorithm 1, only thin matrix multiplications are required and the only matrix inversions ($\boldsymbol{\mathcal{K}}^{-1}$) are of dimension $K$ (the number of bound panels). Furthermore, since each frequency evaluation is independent, the algorithm is easily parallelised. Finally, as $\boldsymbol{\Phi}_{\boldsymbol{c},\boldsymbol{o}}(k)$ in unsteady aerodynamics are typically smooth functions, very few integration points yield accurate and stable realisations.

## IV. ROM interpolation

The FLBT presented in Sec. III for the LUVLM requires low computational and memory resources, but assembling the full system for reduction can still be rather costly. For situations with a large design space or flight envelope, substantial gains in computational performance can be achieved through ROM interpolation. The full-state LUVLM is thus defined over a parameter space, $\boldsymbol{p}$, such that $\boldsymbol{G}_{\boldsymbol{a}} = \boldsymbol{G}_{\boldsymbol{a}}(\boldsymbol{p})$. This, in turn, sets a dependency on $\boldsymbol{p}$ of the balanced models, as $\boldsymbol{G}_{\boldsymbol{b}}(\boldsymbol{p}) = \boldsymbol{G}_{\boldsymbol{b}}\left(\boldsymbol{G}_{\boldsymbol{a}}(\boldsymbol{p}), \boldsymbol{T}(\boldsymbol{p}), \boldsymbol{W}^{\top}(\boldsymbol{p})\right)$, and their truncations, $\boldsymbol{G}_{\boldsymbol{r}}(\boldsymbol{p})$. We assume that LUVLM realisations are available at a number of tabulated points, $\boldsymbol{p}_{(i)}$, from which ROMs at an interpolation point $\widetilde{\boldsymbol{p}}$ are sought. Tabulated and interpolated quantities are referred to as $(\bullet)_{(i)}$ and $(\widetilde{\bullet})$ respectively, while $\gamma_i$ indicate the interpolation weights. The discussion is general and does not depend neither on the dimensionality of $\boldsymbol{p}$ nor the interpolation method adopted.

We first consider (Sec. IV.A) the problem of interpolating amongst the projection matrices $\boldsymbol{T}(\boldsymbol{p})$ and

$\boldsymbol{W}^\top(\boldsymbol{p})$. In this case, interpolated ROMs are retrieved upon reassembly of the full LUVLM description $\boldsymbol{G_a}(\widetilde{\boldsymbol{p}})$ and projection over the interpolated matrices $(\widetilde{\boldsymbol{T}}, \widetilde{\boldsymbol{W}}^\top)$. Since this class of solutions is not suitable for real-time applications it will be referred to as *offline*. Faster interpolation strategies, but with less robustness (regarding stability preservation) and accuracy, (i.e., *online* methods) are also sought. Interpolation between discrete-time ROMs, $\boldsymbol{G_{r_{(i)}}}$, are first discussed in Sec. IV.B. Most applications in the literature focus on continuous-time systems and assume orthonormal basis, e.g. as deriving from proper orthogonal decomposition [42–45]. None of these conditions apply to this work, and we have adapted those solution to ROMs deriving from FLBT. Finally, a new approach that interpolates on the transfer functions of the balanced models, is introduced in Sec. IV.C. It increases robustness but at the expense of interpolated ROMs with a larger number of states.

### IV.A. Offline strategy: interpolation amongst Gramians factors

ROMs at the interpolation point $\widetilde{\boldsymbol{p}}$ can be obtained upon reassembly of the full LUVLM description $\boldsymbol{G_a}(\widetilde{\boldsymbol{p}})$ and projection through the interpolated matrices $(\widetilde{\boldsymbol{T}}, \widetilde{\boldsymbol{W}}^\top)$. Specifically, we seek realisations of the form $\widetilde{\boldsymbol{G_b}} = \boldsymbol{G_b}(\boldsymbol{G_a}(\widetilde{\boldsymbol{p}}), \widetilde{\boldsymbol{T}}, \widetilde{\boldsymbol{W}}^\top)$ with $(\widetilde{\boldsymbol{T}}, \widetilde{\boldsymbol{W}}^\top)$ obtained from tabulated data at sample points $\boldsymbol{p}_{(i)}$. It is well known that element-wise interpolation amongst basis $(\boldsymbol{T}_{(i)}, \boldsymbol{W}_{(i)}^\top)$ is rarely accurate and does not preserve stability [28, 43, 50]. The problem is often tackled introducing the notion of Grassmann manifold, $\mathcal{G}(N_b, N_x)$, which defines the set of all $N_b$-dimensional subspaces of $\mathbb{R}^{N_x}$ [51]. As each basis $\boldsymbol{T}_{(i)}$ defines a point $\mathcal{Y}_{(i)} \in \mathcal{G}(N_b, N_x)$, interpolation is carried over a space tangent to the Grassmann manifold at a reference point $\mathcal{Y}_{(0)}$ — e.g. associated to a parameter vector $\boldsymbol{p}_0$ in the neighbourhood of $\widetilde{\boldsymbol{p}}$ [28, 42]. Logarithmic and exponential mapping procedures to and from that tangent space are used, but they require orthonormality in the underlying basis [28, 42, 45], which does not occur in Eq. (17).

We propose instead direct interpolation of the transfer functions $\boldsymbol{\Phi}_{\boldsymbol{c},\boldsymbol{o}_{(i)}}(k)$, Eq. (14), at each reduced frequency, $k$. Each element $(m, n)$ of these arrays, in fact, always represents the $m$th state response to the $n$th input/output at frequency $k$. As a result, if the Gramians factors $\boldsymbol{Z}_{\boldsymbol{c}_{(i)}}$ and $\boldsymbol{Z}_{\boldsymbol{o}_{(i)}}$ in Eq. (16) are obtained from quadrature at the same sampling frequencies $\boldsymbol{k}$, element-wise interpolation becomes meaningful and the interpolated Gramians at $\widetilde{\boldsymbol{p}}$ are

$$\widetilde{\boldsymbol{Z}_{\boldsymbol{c},\boldsymbol{o}}} = \sum_{\gamma_i=1}^{N_\gamma} \gamma_i \, \boldsymbol{Z}_{\boldsymbol{c},\boldsymbol{o}_{(i)}}. \tag{22}$$

The projection matrices $(\widetilde{\boldsymbol{T}}, \widetilde{\boldsymbol{W}}^\top)$ at $\widetilde{\boldsymbol{p}}$ can be retrieved through the small-size SVD

$$\widetilde{\boldsymbol{Z}_{\boldsymbol{o}}}^\top \widetilde{\boldsymbol{Z}_{\boldsymbol{c}}} = \sum_{i,j=1}^{N_\gamma} (\gamma_i \gamma_j) \, \boldsymbol{Z}_{\boldsymbol{o}(i)}^\top \boldsymbol{Z}_{\boldsymbol{c}(j)} = \widetilde{\boldsymbol{U}} \widetilde{\boldsymbol{\Sigma}} \widetilde{\boldsymbol{V}}^\top \tag{23}$$

and Eq. (17) as $\widetilde{\boldsymbol{T}} = \widetilde{\boldsymbol{Z}_{\boldsymbol{c}}}\widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{\Sigma}}^{-1/2}$ and $\widetilde{\boldsymbol{W}}^\top = \widetilde{\boldsymbol{\Sigma}}^{-1/2}\widetilde{\boldsymbol{U}}^\top\widetilde{\boldsymbol{Z}_{\boldsymbol{o}}}^\top$. While all $N_b$ balanced states are retained, the projection basis can still be truncated and, importantly, by construction the orthonormality relation $\widetilde{\boldsymbol{W}}^\top\widetilde{\boldsymbol{T}} = \boldsymbol{I}_{Nb}$ is preserved. Note finally that Eq. (22) simplifies considerably the interpolation process, as it does not require mapping of $\boldsymbol{T}_{(i)}$ and $\boldsymbol{W}_{(i)}^\top$. Instead, at each point of the design space $\boldsymbol{p}_{(i)}$, the Gramians factors $\boldsymbol{Z}_{\boldsymbol{c},\boldsymbol{o}_{(i)}}$, which are only marginally larger than $\boldsymbol{T}_{(i)}$ and $\boldsymbol{W}_{(i)}^\top$, need to be stored. The only requirements are that interpolation is carried amongst balanced systems of the same dimension, $N_b$, and that the Gramians at each parameter point $\boldsymbol{p}_{(i)}$ are evaluated at the same reduced frequencies. Thanks to the very fast convergence properties of algorithm 2, we will show in the numerical studies in Sec. V.B.1 that none of these requirements is particularly restrictive.

### IV.B. Online strategy: Interpolation on reduced-order models

Next, interpolation amongst balanced truncated models, $\boldsymbol{G}_{\boldsymbol{r}(i)}$, of equal order, $N_r$, is considered. Similarly to Sec. IV.A, element-wise interpolation in the ROM matrices is not a reasonable choice and ROMs are first projected over a congruent set, $\widehat{\boldsymbol{W}}^\top$ and $\widehat{\boldsymbol{T}}$ [42]. Introducing $(\hat{\bullet})$ to refer to variables expressed in the resulting generalised coordinates, a transformation $\boldsymbol{Q}_{(i)} \in \mathbb{R}^{N_r \times N_r}$ such that $\boldsymbol{x}_{\boldsymbol{r}(i)} = \boldsymbol{Q}_{(i)}\hat{\boldsymbol{x}}_{\boldsymbol{r}}$ is introduced such that the subspace spanned by $\boldsymbol{T}_{(i)}\boldsymbol{Q}_{(i)}$ is as close as possible to the one given by $\widehat{\boldsymbol{T}}$ [42, 43]. Similarly, a transformation $\boldsymbol{P}_{(i)}^\top \in \mathbb{R}^{N_r \times N_r}$ such that $\boldsymbol{P}_{(i)}^\top\boldsymbol{W}_{(i)}^\top \approx \widehat{\boldsymbol{W}}^\top$ is also required since $\boldsymbol{Q}_{(i)}$ is in general not invertible [51]. Using $\boldsymbol{P}_{(i)}^\top$ and $\boldsymbol{Q}_{(i)}$, $\boldsymbol{G}_{\boldsymbol{r}(i)}$ can be projected as

$$\hat{\boldsymbol{A}}_{\boldsymbol{r}(i)} = \boldsymbol{P}_{(i)}^\top\boldsymbol{A}_{\boldsymbol{r}(i)}\boldsymbol{Q}_{(i)}\,, \quad \hat{\boldsymbol{B}}_{\boldsymbol{r}(i)} = \boldsymbol{P}_{(i)}^\top\boldsymbol{B}_{\boldsymbol{r}(i)} \quad \text{and} \quad \hat{\boldsymbol{C}}_{\boldsymbol{r}(i)} = \boldsymbol{B}_{\boldsymbol{r}(i)}\boldsymbol{Q}_{(i)} \tag{24}$$

The choice of reference basis has a big impact on the process accuracy [42] and two options are considered in this work. The first one is to define the reference basis as those associated to the closest sampled design point to $\widetilde{\boldsymbol{p}}$, namely $\hat{\boldsymbol{p}} : \|\hat{\boldsymbol{p}} - \widetilde{\boldsymbol{p}}\| = \min\left(\left\|\boldsymbol{p}_{(i)} - \widetilde{\boldsymbol{p}}\right\|\right)$ [43]. This inherently assumes that, if $\hat{\boldsymbol{p}}$ and $\widetilde{\boldsymbol{p}}$ are close enough, the basis at $\hat{\boldsymbol{p}}$ contain all the fundamental modes describing the aerodynamics at $\widetilde{\boldsymbol{p}}$. Importantly, in this case the projection matrices can be precomputed offline, which makes the interpolation computationally cheap and appealing for online applications. The second option is to consider again the interpolated basis $\widetilde{\boldsymbol{T}}$ and $\widetilde{\boldsymbol{W}}^\top$ derived in Sec. IV.A, with the aim of improving the accuracy of the process. This solution has a larger computational cost, but would still be suitable in the context of loads and design analyses.

Next, the projection of the tabulated ROMs as per Eq. (24) is addressed. Regarding the right projection matrices, $\boldsymbol{T}_{(i)}$, the problem of finding $\boldsymbol{Q}_{(i)}$ such that $\boldsymbol{T}_{(i)}\boldsymbol{Q}_{(i)} \approx \widehat{\boldsymbol{T}}$ can be recast as the minimization of the matrix residual $\boldsymbol{\mathcal{R}} = \boldsymbol{T}_{(i)}\boldsymbol{Q}_{(i)} - \widehat{\boldsymbol{T}}$ according to some norm [42]. Since the literature typically considers the case in which the basis $\boldsymbol{T}_{(i)}$ is an orthonormal set [43, 44, 52], three alternative approaches suitable for FLBT are investigated here. A first solution is to solve the residual after projection on the subspace defined by the

reference left projection matrix $\widehat{\boldsymbol{W}}^\top$. Exploiting that $\boldsymbol{W}^\top\boldsymbol{T} = \boldsymbol{I}_{N_r}$, this leads to $\boldsymbol{Q}_{(i)} = (\widehat{\boldsymbol{W}}^\top\boldsymbol{T}_{(i)})^{-1}$ and $\boldsymbol{P}_{(i)}^\top = \boldsymbol{Q}_{(i)}^{-1} = \widehat{\boldsymbol{W}}^\top\boldsymbol{T}_{(i)}$. Note that this solution corresponds to projecting exactly the state equation onto the balanced modes $\widehat{\boldsymbol{W}}^\top$ if $N_r = N_x$. As it will be shown in Sec. V.B.2, this approach lacks robustness when either the distance between $\hat{\boldsymbol{p}}$ and $\boldsymbol{p}_{(i)}$ or $N_r$ are large, resulting in ill-conditioned $(\widehat{\boldsymbol{W}}^\top\boldsymbol{T}_{(i)})$.

A more robust process is achieved by projecting the residual onto the local left projection matrix $\boldsymbol{W}_{(i)}^\top$, which simply leads to $\boldsymbol{Q}_{(i)} = \boldsymbol{W}_{(i)}^\top\widehat{\boldsymbol{T}}$. In a similar way, solution of $\boldsymbol{W}_{(i)}\boldsymbol{P}_{(i)} = \widehat{\boldsymbol{W}}$ upon projection over $\boldsymbol{T}_{(i)}^\top$ gives $\boldsymbol{P}_{(i)}^\top = \widehat{\boldsymbol{W}}^\top\boldsymbol{T}_{(i)}$. As a result, ROM modes that are not shared by the bases may be lost, with loss of accuracy. However, as no matrix inversion is required, the process is computationally robust even when the basis $\boldsymbol{T}_{(i)}$ and $\boldsymbol{W}_{(i)}^\top$ are known in a coarse sampling.

Finally, projection through the weak Modal Assurance Criterion [42, 43, 52] is also investigated . This approach requires that the $\boldsymbol{T}_{(i)}$ matrix of each ROM is first orthonormalised through an SVD as

$$\boldsymbol{T}_{(i)} = \boldsymbol{L}_{(i)}\boldsymbol{M}_{(i)}\boldsymbol{N}_{(i)}^\top = \boldsymbol{L}_{(i)}\boldsymbol{H}_{(i)} \tag{25}$$

where $\boldsymbol{L}_{(i)} \in \mathbb{R}^{N_x \times N_r}$ is orthonormal and $\boldsymbol{H}_{(i)} = \boldsymbol{M}_{(i)}\boldsymbol{N}_{(i)}^\top \in \mathbb{R}^{N_x \times N_x}$. Importantly, as a result of the SVD properties, $\boldsymbol{H}_{(i)}$ admits inverse $\boldsymbol{H}_{(i)}^{-1} = \boldsymbol{N}_{(i)}\boldsymbol{M}_{(i)}^{-1}$, i.e. $\boldsymbol{T}_{(i)}$ and $\boldsymbol{L}_{(i)}$ span the same $N_r$-dimensional subspace of $\mathbb{R}^{N_x}$ [51]. Employing the change of variables $\boldsymbol{H}_{(i)}\boldsymbol{x}_{\boldsymbol{r}(i)} = \boldsymbol{v}_{\boldsymbol{r}(i)}$, the generic ROM $\boldsymbol{G}_{\boldsymbol{r}(i)}$ can be written as

$$\boldsymbol{v}_{\boldsymbol{r}(i)}^{n+1} = \left(\boldsymbol{H}_{(i)}\boldsymbol{A}_{\boldsymbol{r}(i)}\boldsymbol{H}_{(i)}^{-1}\right)\boldsymbol{v}_{\boldsymbol{r}(i)}^{n} + \left(\boldsymbol{H}_{(i)}\boldsymbol{B}_{\boldsymbol{r}(i)}\right)\boldsymbol{u}_{\boldsymbol{a}}^{n} \quad \text{and} \quad \boldsymbol{y}_{\boldsymbol{a}} = \left(\boldsymbol{C}_{\boldsymbol{r}(i)}\boldsymbol{H}_{(i)}^{-1}\right)\boldsymbol{v}_{\boldsymbol{r}(i)}^{n} + \boldsymbol{D}_{\boldsymbol{a}(i)}\boldsymbol{u}_{\boldsymbol{a}}^{n}, \tag{26}$$

which corresponds to a reduction of $\boldsymbol{G}_{\boldsymbol{a}(i)}$ through the basis $\boldsymbol{H}_{(i)}\boldsymbol{W}_{(i)}^\top$ and $\boldsymbol{L}_{(i)}$. The residual minimization problem used so far is now recast in terms of the orthonormal basis $\boldsymbol{L}_{(i)}$ and $\widehat{\boldsymbol{L}}$ arising from the decomposition of $\widehat{\boldsymbol{T}}$ through Eq. (25). This leads to the orthogonal Procrustes problem

$$\min_{\boldsymbol{R}_{(i)}} \left\| \boldsymbol{L}_{(i)}\boldsymbol{R}_{(i)} - \widehat{\boldsymbol{L}} \right\|_F \quad \text{s.t.} \quad \boldsymbol{R}_{(i)}^\top\boldsymbol{R}_{(i)} = \boldsymbol{I}_{N_r} \tag{27}$$

where $\| \bullet \|_F$ indicated the Frobenius norm and the constraint $\boldsymbol{R}_{(i)}^\top\boldsymbol{R}_{(i)} = \boldsymbol{I}_{N_r}$ guarantees regularity in the transformation [42, 43, 53]. Note that an analogous solution is not sought for the left projection terms of Eq. (26), $\boldsymbol{H}_{(i)}\boldsymbol{W}_{(i)}^\top$, as these are not generally orthonormal. With reference to Eq. (24), substituting $\boldsymbol{v}_{\boldsymbol{r}(i)} = \boldsymbol{R}_{(i)}\hat{\boldsymbol{v}}_{\boldsymbol{r}}$ in Eq. (26) allows to identify the projection matrices

$$\boldsymbol{Q}_{(i)} = \boldsymbol{H}_{(i)}^{-1}\boldsymbol{R}_{(i)} \quad \text{and} \quad \boldsymbol{P}_{(i)}^\top = \boldsymbol{R}_{(i)}^\top\boldsymbol{H}_{(i)} \tag{28}$$

associated to the change of coordinates $\boldsymbol{x}_{\boldsymbol{r}(i)}$ to $\hat{\boldsymbol{v}}_{\boldsymbol{r}}$, where interpolation is performed. Upon projection of the ROMs on common generalised coordinates, $\widetilde{\boldsymbol{G}_{\boldsymbol{r}}}$ is obtained by element-wise interpolation of the matrices

$\boldsymbol{A}_{\boldsymbol{r}(i)}$, $\boldsymbol{B}_{\boldsymbol{r}(i)}$, $\boldsymbol{C}_{\boldsymbol{r}(i)}$ and $\boldsymbol{D}_{\boldsymbol{a}(i)}$. Since $\boldsymbol{G}_{\boldsymbol{r}(i)}$ are given in discrete time, the state matrices $\boldsymbol{A}_{\boldsymbol{r}(i)}$ do not possess special regularities and may have zero eigenvalues, as the full-state matrix in Eq. (4) does. Therefore, interpolation on restricted manifolds, as commonly employed for continuous-time systems [43, 44, 52], is not possible, and stability of the interpolated ROM can not be guaranteed, although it has been observed in practice in our numerical investigations. Note finally that direct ROM interpolation depends on the choice of generalised coordinates and on the projection method, and, as a result, may produce poor models with interpolation in a coarse grid. However, as it will be shown numerically in Sec. V.B.2 an optimal number of reduced states can be (cheaply) found from an estimate of the frequency response at the interpolation point, $\widetilde{\boldsymbol{p}}$.

### IV.C. Transfer function interpolation for frequency-limited balanced models

Transfer-function interpolation has some similarities to the Gramian factors interpolation discussed in Sec. IV.A. Whereas the latter is based on the assumption that $\widetilde{\boldsymbol{\Phi}_{\boldsymbol{c},\boldsymbol{o}}} = \sum_i \gamma_i \boldsymbol{\Phi}_{\boldsymbol{c},\boldsymbol{o}_{(i)}}$, here the transfer function of the interpolated model is expressed as

$$\widetilde{\boldsymbol{G}}(k) = \sum_{i=1}^{N_\gamma} \gamma_i \, \boldsymbol{G}_{(i)}(k). \tag{29}$$

A realisation of Eq. (29) is provided by the set [42]

$$\widetilde{\boldsymbol{A}} = \mathrm{diag}(\boldsymbol{A}_{(i)}), \quad \widetilde{\boldsymbol{B}} = \mathrm{col}\big[\boldsymbol{B}_{(i)}\big], \quad \widetilde{\boldsymbol{C}} = \mathrm{row}\big[\gamma_i \, \boldsymbol{C}_{(i)}\big], \quad \widetilde{\boldsymbol{D}} = \sum_{i=1}^{N_\gamma} \gamma_i \, \boldsymbol{D}_{(i)} \quad \text{and} \quad \widetilde{\boldsymbol{x}} = \mathrm{col}\big[\boldsymbol{x}_{(i)}\big] \tag{30}$$

where row[•] col[•] indicate, as before, row and column-wise concatenations. This realisation preserves stability and does not require special manipulation of the tabulated ROMs, such as projection over a congruent basis. In fact, the tabulated models $\boldsymbol{G}_{(i)}$ may be of different dimensions. The number of states of the interpolated system defined by Eq. (30) is however larger than the order of the underlying ROMs. While this is not always a strong penalty — e.g., for time-domain aeroelasticity, the computational overhead can be minimized by independently coupling each $\boldsymbol{G}_{(i)}(k)$ with a structural description and using the superposition principle — we overcome this by directly retrieving a frequency-limited balanced realisation of the interpolated state-space model, $\widetilde{\boldsymbol{G}}$. For that purpose, we assume that (a) the tabulated models are all balanced but not truncated, i.e. $\boldsymbol{G}_{(i)} = \boldsymbol{G}_{\boldsymbol{b}(i)}$, and (b) they have been obtained by integration of the Gramians over the same frequency grid. The latter condition, in particular, implies that all $\boldsymbol{G}_{\boldsymbol{b}(i)}$ have equal number of states, $N_b$. Under these hypotheses, Eq. (14a) results in the interpolated input-to-state transfer function

$$\widetilde{\boldsymbol{\Phi}_{\boldsymbol{c}}}(z) = \Big(z\boldsymbol{I}_{(N_\gamma N_b)} - \widetilde{\boldsymbol{A}}\Big)^{-1} \widetilde{\boldsymbol{B}} = \mathrm{col}\Big[\big(z\boldsymbol{I}_{N_b} - \boldsymbol{A}_{\boldsymbol{b}(i)}\big)^{-1} \boldsymbol{B}_{\boldsymbol{b}(i)}\Big]. \tag{31}$$

Each term of $\widetilde{\boldsymbol{\Phi}_{\boldsymbol{c}}}$ represents the state response of the $i$th balanced state vector $\boldsymbol{x}_{\boldsymbol{b}(i)}$ which, recalling Eq. (17), can be written as $\boldsymbol{x}_{\boldsymbol{b}(i)} = \boldsymbol{W}_{(i)}^{\top}\boldsymbol{x}_{\boldsymbol{a}(i)}$. Therefore, $\widetilde{\boldsymbol{\Phi}_{\boldsymbol{c}}}$ can be expressed in terms of the full-system frequency response as $\widetilde{\boldsymbol{\Phi}_{\boldsymbol{c}}}(z) = \mathrm{col}\big[{\boldsymbol{W}^{\top}}_{(i)}\boldsymbol{\Phi}_{\boldsymbol{c}(i)}(z)\big]$, which, from Eq. (16) and the SVD of (17), allows finding the square-root factor of the controllability and observability Gramians of $\widetilde{\boldsymbol{G}}$ as

$$\widetilde{\boldsymbol{Z}_{\boldsymbol{c}}}^{\top} = \mathrm{col}\Big[\boldsymbol{W}_{(i)}^{\top}\boldsymbol{Z}_{\boldsymbol{c}(i)}\Big] = \mathrm{col}\Big[\boldsymbol{\Sigma}_{(i)}^{1/2}\boldsymbol{V}_{(i)}^{\top}\Big] \quad \text{and} \tag{32a}$$

$$\widetilde{\boldsymbol{Z}_{\boldsymbol{o}}}^{\top} = \mathrm{row}\Big[\gamma_i\,{\boldsymbol{Z}_{\boldsymbol{o}}^{\top}}_{(i)}\boldsymbol{T}_{(i)}\Big] = \mathrm{row}\Big[\boldsymbol{U}_{(i)}\boldsymbol{\Sigma}_{(i)}^{1/2}\Big]. \tag{32b}$$

Finally, using Eq. (32), projection matrices for the interpolated state-space model $\widetilde{\boldsymbol{G}}$ are obtained from the SVD

$$\widetilde{\boldsymbol{Z}_{\boldsymbol{o}}}^{\top}\widetilde{\boldsymbol{Z}_{\boldsymbol{c}}} = \sum_{i=1}^{N_\gamma}\gamma_i\,\boldsymbol{U}_{(i)}\boldsymbol{\Sigma}_{(i)}\boldsymbol{V}_{(i)}^{\top} = \widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Sigma}}\widetilde{\boldsymbol{V}}^{\top} \tag{33}$$

as

$$\widetilde{\boldsymbol{T}} = \mathrm{col}\Big[\boldsymbol{\Sigma}_{(i)}^{1/2}\boldsymbol{V}_{(i)}^{\top}\,\widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{\Sigma}}^{-1/2}\Big] \quad \text{and} \quad \widetilde{\boldsymbol{W}}^{\top} = \mathrm{row}\Big[\gamma_i\,\widetilde{\boldsymbol{\Sigma}}^{-1/2}\widetilde{\boldsymbol{U}}^{\top}\,\boldsymbol{U}_{(i)}\boldsymbol{\Sigma}_{(i)}^{1/2}\Big]. \tag{34}$$

Importantly, $\widetilde{\boldsymbol{T}}$ and $\widetilde{\boldsymbol{W}}^{\top}$ in Eq. (34) never need to be assembled. Projection of $\widetilde{\boldsymbol{G}}$ in Eq. (30) through $\widetilde{\boldsymbol{T}}$ and $\widetilde{\boldsymbol{W}}^{\top}$ allows writing its frequency-limited balanced realisation, $\widetilde{\boldsymbol{G}_{\boldsymbol{b}}}$, in the $N_b$-dimensional state $\widetilde{\boldsymbol{x}_{\boldsymbol{b}}}$ as

$$\begin{aligned}
\widetilde{\boldsymbol{A}_{\boldsymbol{b}}} &= \sum_{i=1}^{N_\gamma}\gamma_i\,\boldsymbol{P}_{(i)}^{\top}\,\boldsymbol{A}_{\boldsymbol{b}(i)}\,\boldsymbol{Q}_{(i)} & \widetilde{\boldsymbol{C}_{\boldsymbol{b}}} &= \sum_{i=1}^{N_\gamma}\gamma_i\,\boldsymbol{C}_{\boldsymbol{b}(i)}\,\boldsymbol{Q}_{(i)} \\
\widetilde{\boldsymbol{B}_{\boldsymbol{b}}} &= \sum_{i=1}^{N_\gamma}\gamma_i\,\boldsymbol{P}_{(i)}^{\top}\,\boldsymbol{B}_{\boldsymbol{b}(i)} & \widetilde{\boldsymbol{D}_{\boldsymbol{b}}} &= \sum_{i=1}^{N_\gamma}\gamma_i\,\boldsymbol{D}_{\boldsymbol{b}(i)}
\end{aligned} \tag{35}$$

with

$$\boldsymbol{Q}_{(i)} = \Big({\boldsymbol{\Sigma}^{1/2}}_{(i)}\boldsymbol{V}_{(i)}^{\top}\Big)\,\Big(\widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{\Sigma}}^{-1/2}\Big) \quad \text{and} \quad \boldsymbol{P}_{(i)}^{\top} = \Big(\widetilde{\boldsymbol{\Sigma}}^{-1/2}\widetilde{\boldsymbol{U}}^{\top}\Big)\,\Big(\boldsymbol{U}_{(i)}{\boldsymbol{\Sigma}^{1/2}}_{(i)}\Big). \tag{36}$$

There are clear similarities between Eq. (35)-(36) and the direct ROM interpolation in Sec. IV.B, since the tabulated state-space models are also projected here over a common set of generalised coordinates before element-wise interpolation. However, this new interpolation is amongst frequency-limited balanced, non-truncated, discrete linear time-invariant systems. Note that due to the fast convergence properties of FLBT this is not an important penalty, as very accurate representation can be obtained with just a few integration points, $\mathcal{K}_{low}$ and $\mathcal{K}_{high}$ in algorithm 2. Moreover, truncation can still be employed to further reduce the size of $\widetilde{\boldsymbol{G}_{\boldsymbol{b}}}$. Another difference between the methods is that the projection terms $\boldsymbol{Q}_{(i)}$ and $\boldsymbol{P}_{(i)}^{\top}$ always depend on the interpolation point through the SVD in Eq. (33). Contrarily to the methods described in Sec. IV.B, which require manipulating the (larger) $N_x \times N_r$ basis $\boldsymbol{T}_{(i)}$ and $\boldsymbol{W}_{(i)}$, here only the small matrices $\boldsymbol{U}_{(i)}{\boldsymbol{\Sigma}^{1/2}}_{(i)}$ and ${\boldsymbol{\Sigma}^{1/2}}_{(i)}{\boldsymbol{V}^{\top}}_{(i)}$ are required. Overall, the online cost of the method is dominated by the small SVD in

Eq. (33). Despite $\mathcal{K}_{low}$ and $\mathcal{K}_{high}$ being typically small numbers, the approach may still be too costly for real-time evaluation, in which case either the unbalanced transfer function interpolation in Eq. (30) or the direct interpolation of Sec. IV.B may be more suitable. However, as it will be shown in Sec. V.B.3, it provides a generic and computationally-robust low-cost solution to the interpolation problem. Finally, it is worth remarking that Eq. (36) does not automatically extend to other balancing methods and/or projection over arbitrary bases.

## V. Numerical studies

The previous methods will be exercised on the aeroelastic stability of the T-tail configuration introduced by Murua *et al.* [9]. Both vertical and horizontal tailplanes (VTP and HTP, respectively) have prismatic shape, with the stiffness and inertial properties of Tab. 1. Their structural dynamics is described through beams models. Aeroelastic coupling is obtained upon projection of the structural equations in modal coordinates as per Eq. (8). The stability of the T-tail is dominated by the coupling between the VTP first bending and torsional modes (Fig. 1). It also strongly depends on the steady aerodynamic loads on the HTP. Therefore, this test case offers a rich physical context to exercise the ROM interpolation strategies introduced in Sec. IV.

This configuration will be used, first (Sec. V.A), to exercise the FLBT approach introduced in Sec. III, and later (Sec. V.B) to benchmark the interpolation techniques proposed in Sec. IV, using the angle of attack as the system parameter. A convergence study indicated that projection of the aerodynamic inputs/outputs over the first eight structural modes is required to accurately capture the flutter features, and all results in this section use that structural description.

| component | chord ($c$) | span | elastic axis (from L.E.) | inertial axis (from L.E.) | mass (per unit length) | sectional inertia | torsional stiffness | bending stiffness |
|---|---|---|---|---|---|---|---|---|
| VTP | 2 m | 6 m | 0.25 c | 0.35 c | $35\,\mathrm{kg\,m^{-1}}$ | 8 kg m | $10^7\,\mathrm{N\,m^2}$ | $10^7\,\mathrm{N\,m^2}$ |
| HTP | | 8 m | | | | | $10^8\,\mathrm{N\,m^2}$ | $10^8\,\mathrm{N\,m^2}$ |

Table 1: T-tail properties.

### V.A. Frequency limited balancing

Firstly, we assess the convergence and stability features of FLBT with respect to the number of integration points in Eq. (15). To this purpose, a converged UVLM model of the T-tail at zero incidence is considered. The VTP and HTP aerodynamic lattices are discretized with 24 panels in both the chord and spanwise directions. A 25-chord-long wake is added, resulting in a LUVLM state-space model of dimension 32 256. FLBT is used to generate reduced-order realisations in the reduced frequencies range $[0, \bar{k}]$, where $\bar{k} = 0.5$.

(a) Mode 1 (VTP bending, 2.86 Hz) (b) Mode 2 (VTP torsion, 5.21 Hz)

Figure 1: Projection of the first two structural modes over the UVLM lattice.

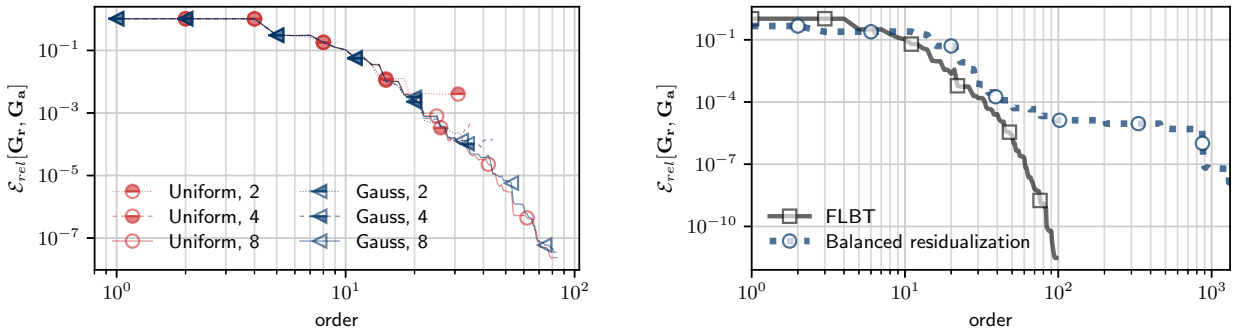
Throughout the rest of this work, the accuracy of the ROMs so generated is measured according to the metric

$$\mathcal{E}_{rel}[\boldsymbol{G_r}, \boldsymbol{G_a}] = \frac{\max_{i,j}\left(\sup_{k\in[0,\bar{k}]}\left[\boldsymbol{G_r}(k) - \boldsymbol{G_a}(k)\right]_{ij}\right)}{\max_{i,j}\left(\sup_{k\in[0,\bar{k}]}\left[\boldsymbol{G_a}(k)\right]_{ij}\right)}. \tag{37}$$

which quantifies the frequency response error of the ROM, $\boldsymbol{G_r}$, with respect to the full LUVLM description, $\boldsymbol{G_a}$, in the frequency range of interest. Note that $\boldsymbol{G_a}$ is cheaply evaluated using algorithm 1.

The number of integration points, $\mathcal{K}_{low}$, used in the low-frequency range, $[0, \bar{k}]$, determines the overall accuracy of the balanced system. To show this, algorithm 2 is employed to obtained six realisations of the LUVLM using $\mathcal{K}_{low} = 2, 4$ and 8 points respectively. For each case, both trapezoidal rule and Gauss-Lobotto quadrature are employed. Integration of the controllability Gramian in the high frequencies is always performed using $\mathcal{K}_{high} = 16$ equally-spaced points and Gaussian quadrature. Fig. 2a, reports the accuracy, $\mathcal{E}_{rel}[\boldsymbol{G_r}, \boldsymbol{G_a}]$, of each truncation of the balanced systems so obtained. As the number of integration points is increased, and regardless of the integration scheme employed, ROMs provide higher accuracy, up to $\mathcal{E}_{rel} < 10^{-7}$ when $\mathcal{K}_{low} = 8$. While the size of the balanced model also increases as more integration points are employed, as little as two integration points are sufficient to provide realisations with $\mathcal{E}_{rel} < 10^{-2}$. Most importantly, neither the number of integration points nor the quadrature scheme significantly affects the convergence with respect to the ROM order. This is further underlined in Fig. 2b. Here the convergence rate of FLBT with $\mathcal{K}_{low} = 12$ is compared against internal balancing with residualisation [17]. As FLBT approximates the original state-space model only in the low-frequency range, a much faster convergence rate than internal balancing rate is achieved on those frequencies. Also, ROMs are obtained by simple truncation rather than through a costly residualisation process.

As it was discussed in Sec. III, integration of the Gramians in $k \in [0, \bar{k}]$ does not guarantee the stability of the balanced model, but stability can obtained by integrating the controllability Gramian over the full Nyquist range. For a numerical demonstration of this, we consider a problem with a fixed number of integration points in the low-frequency range, $\mathcal{K}_{low} = 8$, and a variable number of sample points in the

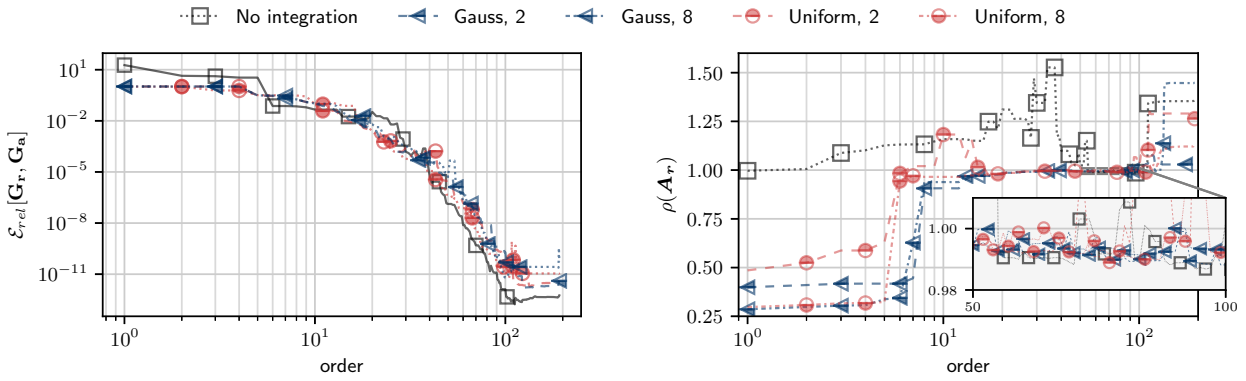



(a) FLBT with different integration points

(b) FLBT vs. internal balancing with residualisation

Figure 2: Impact of the integration of the Gramians over the low-frequency range $[0, \bar{k}]$ on FLBT accuracy.

high-frequency contribution of the controllability Gramian, $\mathcal{K}_{high} = 0, 2$ and 8. The convergence rate with respect to the order of the resulting FLBT ROMs is examined in Fig. 3a. While integration of the Gramians in the high-frequency region does not impact the (low-frequency) accuracy of the ROMs, it is necessary to ensure stability. In this regard, Fig. 3b shows the spectral radius, $\rho(\boldsymbol{A_r})$, of the ROMs considered in Fig. 3a. When $\mathcal{K}_{high} = 0$, balanced-truncation results in unstable state-space models $[\rho(\boldsymbol{A_r}) > 1]$ if, approximately, less than 50 states are retained. Stable and highly accurate ($\mathcal{E}_{rel} < 10^{-6}$) ROMs can still be obtained with more states, but one does not have the trade-off between order and accuracy typical of conventional balancing methods. Adding integration points over $[\bar{k}, k_N]$, however, drastically increases the region of stability of FLBT. Furthermore, Gaussian quadrature provides better convergence properties than trapezoidal rule. As shown in Fig. 3b, stable balanced truncations of order $N_r \in [1, 90]$ are achieved with as little as two integration points in $[\bar{k}, k_N]$. In summary, highly accurate and stable realisations, whose truncations preserve stability, can be obtained with little computational overhead.




(a) balanced truncation accuracy

(b) FLBT models spectral radius

Figure 3: Impact of the integration of the controllability Gramians over the frequencies range $[\bar{k}, k_N]$ on accuracy and stability of FLBT.

The computational performance of our FLBT algorithm is summarised in Tab. 2. The CPU time of

the balancing process, in particular, is here scaled with respect to the time required to balance a reference 23 552-states LUVLM — about 5 s on a four 2.3 GHz CPUs workstation. As the computational cost of FLBT scales with the number of bound panels, $K$, this model-order reduction is two orders of magnitude faster than our previously developed low-rank square-root method (LRSRM) for internal balancing [17]. While algorithm 2 requires $\mathcal{O}(K^3)$ operations, these are automatically parallelized in our high-level implementation, which achieves almost quadratic scaling with $K$. Most importantly, algorithm 2 scales very efficiently with the wake size. As it can be seen in Tab. 2, increasing the wake size by a factor of 5 (which results in $\approx 10^5$ states), only increases the model-reduction time by a factor of 3.56. Finally, the scalability with the number of integration points used to solve Eq. (14) is exemplified. Doubling the refinement in the low-frequency region, $\mathcal{K}_{low}$, roughly doubles the accuracy of the balanced model, but has only a 63% computational penalty. Increasing the number of integration points in the high-frequency region, $\mathcal{K}_{high}$, is less expensive, since only the controllability Gramian is solved for.

All results thus far have been on the unsteady aerodynamic model. FLBT is employed next to analyse the dynamic stability of the T-tail at angles of attack $\alpha$ between $-10$ and 20 deg and sea-level conditions. At each attitude $\alpha$, a normalised realisation limited to the reduced frequencies $k \leq \bar{k} = 0.5$ is produced using $\mathcal{K}_{low} = 12$ and $\mathcal{K}_{high} = 16$. This is coupled with a modal description of the T-tail structural dynamics in non-dimensional time, Eq. (8). Stability is finally determined for speeds between 50 and 350 $\mathrm{m\,s^{-1}}$. This is achieved through an iterative procedure which identifies an upper and lower bound for the flutter speed based on spectral radius of the system. A relative tolerance $10^{-8}$ is used to terminate the process. Finally, it is worth remarking that the lower speed bound of 50 $\mathrm{m\,s^{-1}}$ is chosen to guarantee an accurate representation of structural frequencies up to 15 Hz.

Two LUVLM models of different refinement are employed, as indicated in Tab. 3. Fig. 4 shows that flutter speed and reduced frequency obtained using a relatively course lattice, similar to the one employed by Murua et al. [9], compare well both against reference data and against the stability features as predicted using internal balancing [17]. Non-dimensionalization of the LUVLM equations and FLBT, however, has

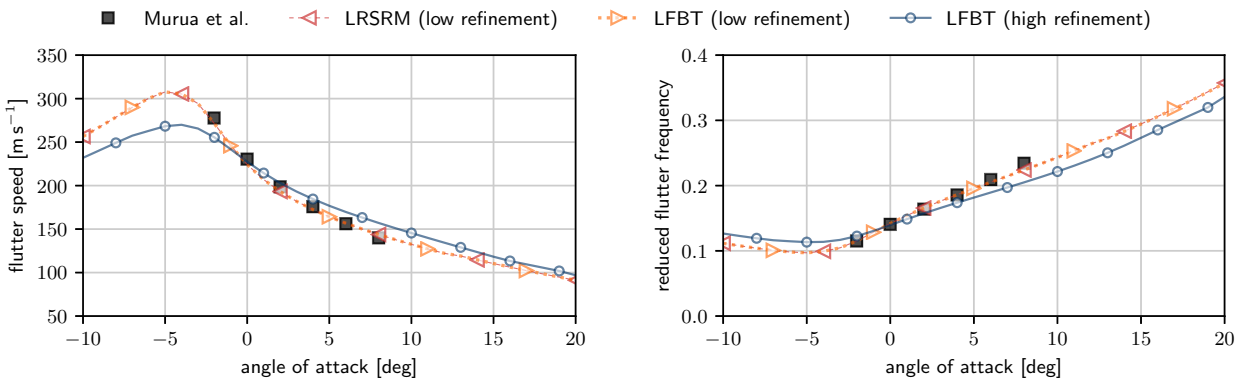
| Case | Discretization | | | Integration points | | Performance | |
|---|---|---|---|---|---|---|---|
| | bound, $K$ | wake, $K_w/K$ | states | $\mathcal{K}_{low}$ | $\mathcal{K}_{high}$ | CPU time | $\mathcal{E}_{rel}[\boldsymbol{G_b}, \boldsymbol{G_a}]$ |
| reference | 1024 | 20 | 23 552 | 4 | 4 | 4.7 s | $2.3 \times 10^{-5}$ |
| LRSRM [17] | 1024 | 20 | 23 552 | 4 | 4 | 841.5× | $6.7 \times 10^{-9}$ |
| scaling with $K$ | 2048 | 20 | 47 104 | 4 | 4 | 4.69× | $2.6 \times 10^{-5}$ |
| scaling with $K_w/K$ | 1024 | 100 | 105 472 | 4 | 4 | 3.56× | $1.2 \times 10^{-4}$ |
| scaling with $\mathcal{K}_{low}$ | 1024 | 20 | 23 552 | 8 | 4 | 1.63× | $8.1 \times 10^{-11}$ |
| scaling with $\mathcal{K}_{high}$ | 1024 | 20 | 23 552 | 4 | 8 | 1.41× | $2.5 \times 10^{-5}$ |

Table 2: Computational performance of the FLBT algorithm.

allowed us to carry on the stability analysis using a highly refined 36 865-states LUVLM, since only one system realisation is required for each angle of incidence. This refined model predicts a smoother trend for the flutter speed. Flutter speed and reduced frequency reach, respectively, a maximum and a minimum around $\alpha = 5\,\text{deg}$. This highlights a change of the VTP torsional-bending flutter mode. To verify this, the flutter mode at $-8\,\text{deg}$ and $-1.5\,\text{deg}$ have been compared. At these attitudes, the flutter speed is similar ($250.3\,\text{m}\,\text{s}^{-1}$ vs. $249.8\,\text{m}\,\text{s}^{-1}$ respectively). However, while at $-8\,\text{deg}$ the magnitude of the VTP torsional displacement mode equals 34 % that of the VTP bending, at $-1.5\,\text{deg}$ this becomes as large as 83.6 %. It is finally worth remarking that for some attitudes flutter occurs at transonic speed. Those results become therefore indicative of stability margins, as the potential flow assumption is no longer valid.

| Refinement | chord-wise panels | | span-wise panels | | States |
|---|---|---|---|---|---|
| | bound | wake | VTP | HTP | |
| low | 22 | 220 | 28 | 14 | 11 440 |
| high | 32 | 480 | 32 | 32 | 36 865 |

Table 3: Detail of UVLM discretizations used for the T-tail stability analysis.




(a) flutter speed

(b) reduced flutter frequency

Figure 4: T-tail flutter stability features as a function of the angle of attack.

### V.B. Studies on parametric reduced-order modelling

The LUVLM is now parametrized with respect to T-tail angle of attack, $\alpha$. Tabulated data are assumed to be available at a discrete set of equally-spaced attitudes $\boldsymbol{a} = [\alpha_i]$. Third-order splines will be used to interpolate amongst quantities. Unless otherwise stated, we seek to construct FLBT ROMs of the T-tail aerodynamics at an attitude $\widetilde{\alpha}$ =5 deg.

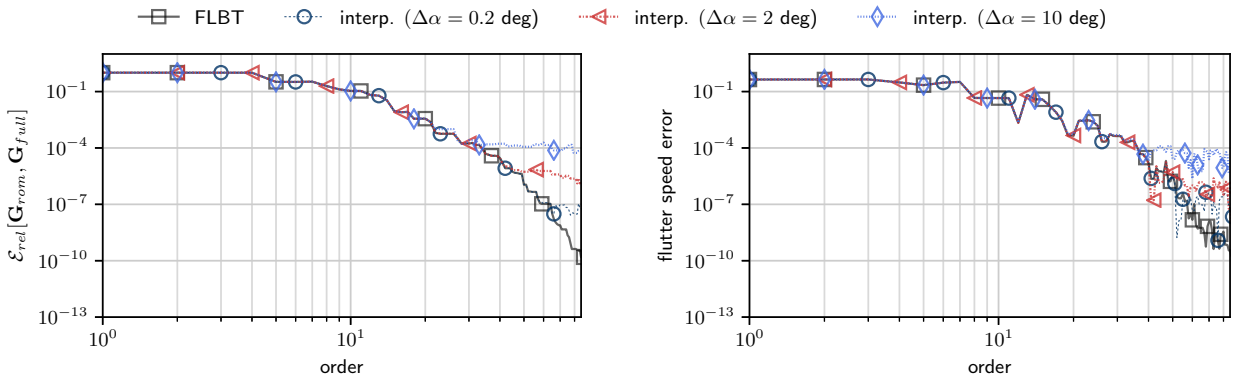
### V.B.1. *Offline interpolation*

The interpolation strategy of Sec. IV.A is considered first. It is assumed that the Gramians factors are available over uniform grids centred in $\widetilde{\alpha}$

$$\boldsymbol{a}(\Delta\alpha) = [\widetilde{\alpha} + (n - 1.5)\Delta\alpha], \quad \text{with} \quad n = 0 \dots 3 \tag{38}$$

where spacings $\Delta\alpha = 0.2,\ 2$, and 10 deg are considered. These have been computed through Eq. (15) using $\mathcal{K}_{low} = 12$ and $\mathcal{K}_{high} = 16$ integration points. FLBT ROMs ranging from 1 to 90 states are finally generated through Eq. (22) and (23) upon re-assembly of the full-states matrices, $\boldsymbol{A_a}$, $\boldsymbol{B_a}$, $\boldsymbol{C_a}$ and $\boldsymbol{D_a}$.

The frequency-response errors $\mathcal{E}_{rel}[\widetilde{\boldsymbol{G_r}}, \boldsymbol{G_a}]$ of the interpolated ROMs are compared in Fig. 5a with those obtained through direct application of FLBT. Interpolated models retain a high level of accuracy, with $\mathcal{E}_{rel}[\widetilde{\boldsymbol{G_b}}, \boldsymbol{G_a}] \approx 10^{-4}$ even when data are provided on the coarser ($\Delta\alpha$ =10 deg) grid. Importantly, while the maximum accuracy achievable depends on the spacing between tabulated data (i.e. on the interpolation error), the interpolated ROMs, $\widetilde{\boldsymbol{G_r}}$, retain the same rate of convergence as those produced through FLBT at $\widetilde{\alpha}$ =5 deg. Furthermore, the stability features of FLBT discussed in Sec. V.A are also preserved. Finally, the fast converge rate of FLBT is also shown when interpolated ROMs are employed to predict the T-tail flutter speed, as shown in Fig. 5b.




(a) Frequency response

(b) Flutter speed

Figure 5: Convergence of interpolation over FLBT Gramians factors of T-tail at 5 deg angle of attack.

A further insight on this study is provided by Fig. 6. This shows the phase responses of the aerodynamic influence coefficients corresponding to the first three modes of the T-tail for different incidence angles. The structural modes are the VTP bending and torsional modes shown in Fig. 1, respectively mode 1 and 2 in Fig. 6, plus the second VTP bending mode (mode 3, natural frequency 11.21 Hz). As the LUVLM description treats modal displacements and velocities as different inputs, the response of the $p$th modal force, $\overline{N}_p$, is

expressed in terms of the $q$th modal displacement input, $\overline{\eta}_q$ as

$$\overline{N}_p = \left[G_{p,q}(k) + jk\, G_{p,r}(k)\right] \overline{\eta}_q \tag{39}$$

where $r$ is the index associated to the $q$th modal velocity input, $\overline{\eta}'_q(k)$. In Fig. 6 these quantities are evaluated using the full LUVLM at the incidence angles associated to a coarse interpolation $\boldsymbol{a}(\Delta\alpha = 10\,\text{deg})$ [Eq. (38)] and at $\widetilde{\alpha} = 5\,\text{deg}$. At this attitude the phase response has also been computed using a 27-states interpolated ROM, with $\mathcal{E}_{rel} \approx 10^{-4}$ as per Fig. 5a. Fig. 6 underlines that the aerodynamic features of the T-tail change considerably within the incidence angles range considered. Nonetheless, and despite the differences in relative magnitude amongst modal responses, interpolation tracks very accurately the reference, full-states, response at $\widetilde{\alpha} = 5\,\text{deg}$.

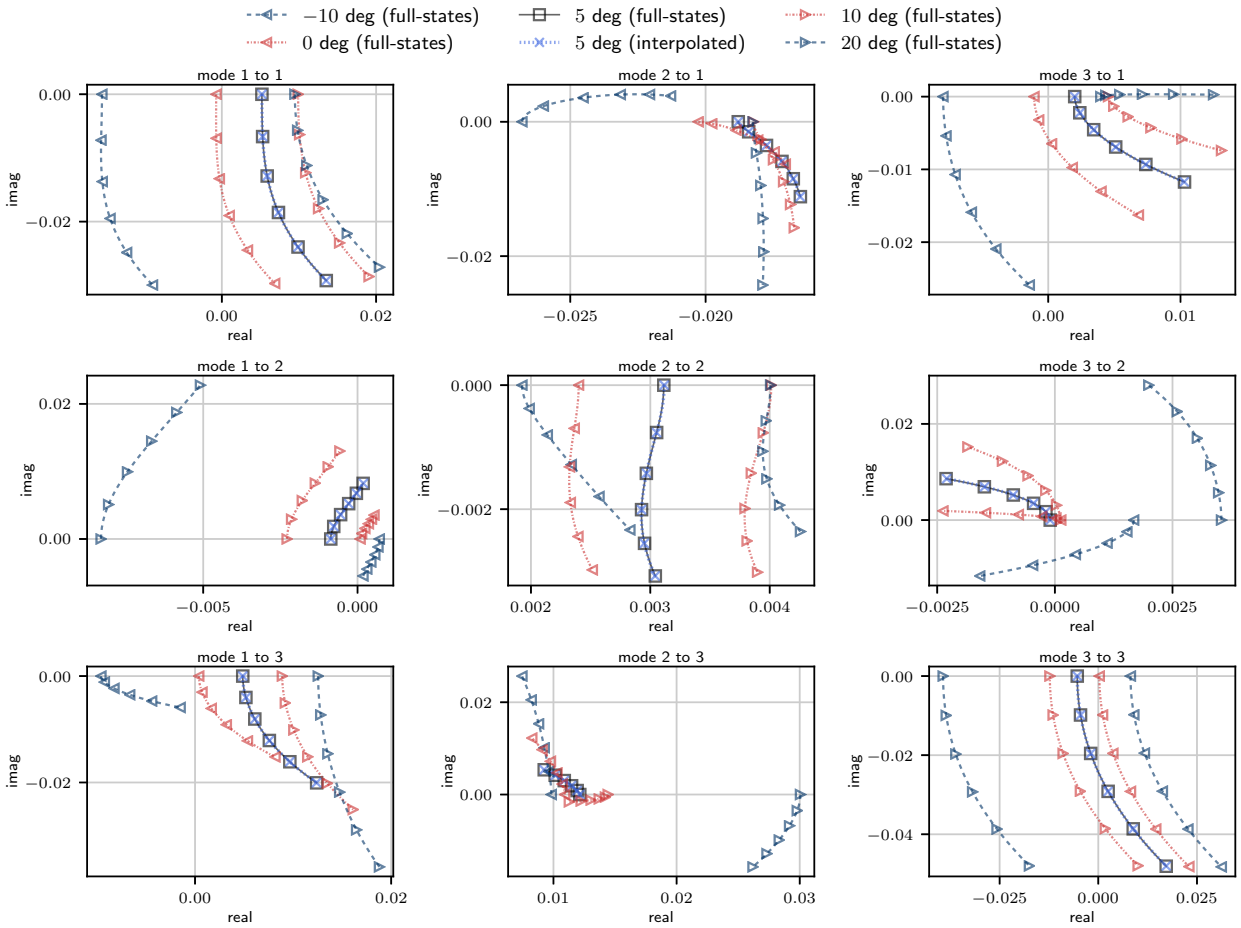


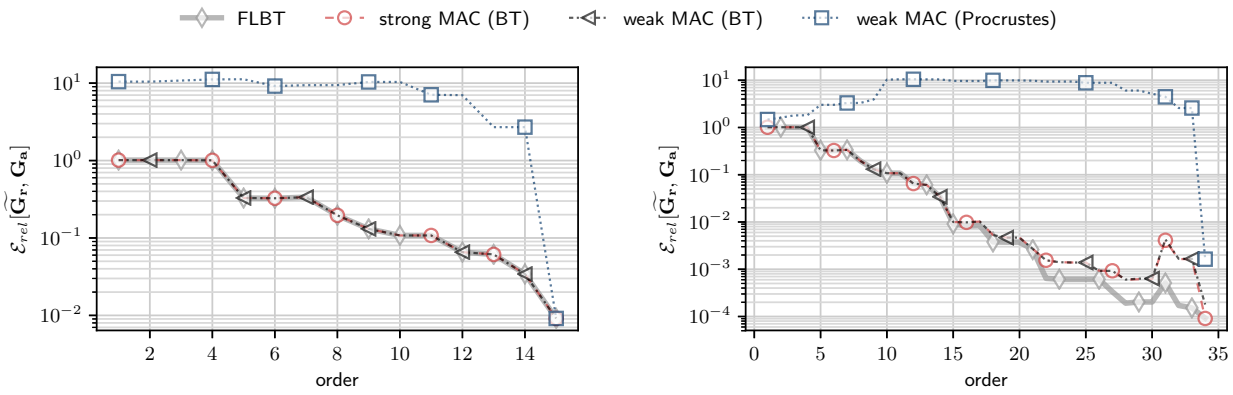

Figure 6: Phase response in the reduced frequency range $k \in [0, 0.5]$ of the first three modes at tabulated and interpolation points. Markers indicate a frequency spacing $\Delta k = 10^{-1}$.

Overall, direct interpolation of the Gramians factors in Eq. (16) does not only facilitates the interpolation process — there is no need to project the basis $\boldsymbol{T}$ and $\boldsymbol{W}^\top$ on a common space before interpolation [28] — but also guarantees accuracy and stability. However, given the efficiency of algorithm 2 ( Tab. 2), the

computational savings with respect to direct model-order reduction is limited, as re-assembly of the full LUVLM matrices is still required.

### *V.B.2. Direct ROM interpolation*

We consider first the case in which tabulated ROMs are provided on a fine grid $\boldsymbol{a}(\Delta\alpha = 1\,\text{deg})$ as per Eq. (38), centered around $\widetilde{\alpha} = 5\,\text{deg}$ as before. Tabulated ROMs are truncated to 15-states (such that $\mathcal{E}_{rel}[\boldsymbol{G}_{\boldsymbol{r}(i)}, \boldsymbol{G}_{\boldsymbol{a}(i)}] < 10^{-2}$) and are projected over the closest basis ($\alpha$ =4.5 deg) using the three methods discussed in Sec. IV.B. Fig. 7a compares the frequency response rate of convergence of all truncations of the ROMs so obtained against direct model-order reduction (FLBT). In all cases, ROM interpolation provides the same accuracy of FLBT when all states are included. However, while the strong and weak MAC preserve the same convergence rate of FLBT, the weak projection in the Procrustes problem solution requires including all reduced states. As the size of the tabulated ROMs is increased to 34 states ($\mathcal{E}_{rel} < 10^{-4}$), Fig. 7b, the latter method also loses accuracy. Balancing methods, instead, still retrieve the same fidelity of direct interpolation. However, their convergence rate departs from the reference one. This highlights a first important feature of direct ROM interpolation, namely, that the projection methods in Sec. IV.B approximate well the first (most important) reduced-states (Fig. 7a). However, even when the tabulated ROMs are close to each other, higher states will generally be less and less correlated, as they describe different aerodynamic features. This information may be lost upon projection, which sets a bound to their accuracy (see Fig. 7b).



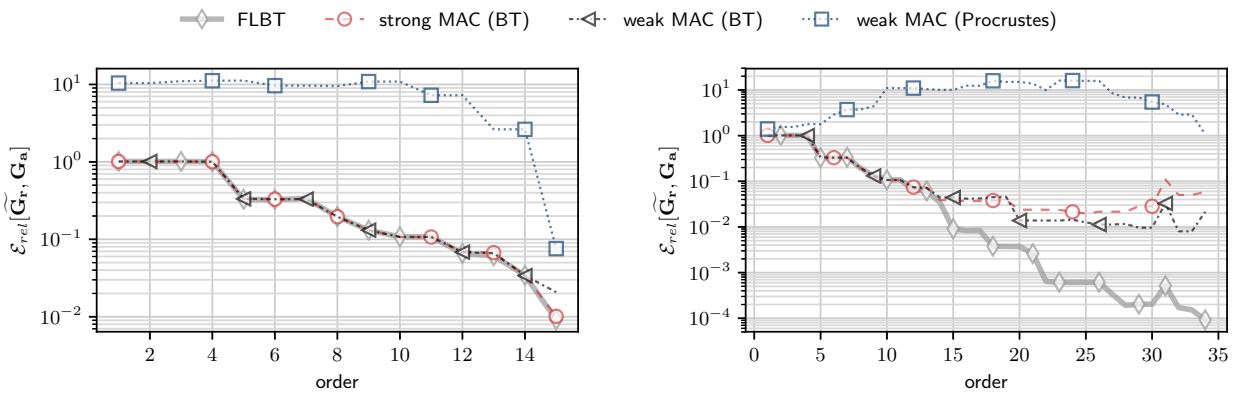

(a) interpolation between $\mathcal{E}_{rel} < 10^{-2}$ accurate ROMs

(b) interpolation between $\mathcal{E}_{rel} < 10^{-4}$ accurate ROMs

Figure 7: Convergence rate of ROM at 5 deg obtained through direct interpolation. Tabulated ROMs of different accuracy ($\mathcal{E}_{rel}$) on a grid with spacing $\Delta\alpha$ =1 deg are considered.

This unfavourable phenomenon is amplified as the spacing between tabulated data increases. To show this, the study in Fig. 7 is repeated using tabulated data on a course grid $\boldsymbol{a}(\Delta\alpha = 10\,\text{deg})$, Eq. (38). As seen in Fig. 8, interpolation upon weak projection and Procrustes solution never produces accurate results. In fact, this methods results in very larger errors when increasing the ROMs order. This further highlights

that the leading factor to detriment the process accuracy is the projection, rather than the interpolation itself. Strong and weak projection exploiting the orthogonality of the balancing basis, instead, achieve greater accuracy. Even in this case, however, the projection method loses robustness when larger basis are considered. Interpolation between small size ROMs ($\mathcal{E}_{rel} < 10^{-2}$) follows closely the convergence rate of direct model-reduction and achieves satisfactory accuracy ($\mathcal{E}_{rel} < 2 \times 10^{-2}$ in both cases). Interpolation amongst 34 states ($\mathcal{E}_{rel} < 10^{-2}$) ROMs, instead, never achieves $\mathcal{E}_{rel}$ below $8 \times 10^{-3}$, against $\mathcal{E}_{rel} \approx 10^{-4}$ of FLBT. Importantly, the convergence rate is not monotonic and the most accurate realizations are obtained when not all reduced states are included. The strong MAC, for example, achieves minimal error, $\mathcal{E}_{rel} \approx 2 \times 10^{-2}$, when only 25 of the 32 reduced-states are retained. Overall, this study highlights that when tabulated data are provided on a course grid, the size of the tabulate ROMs should be carefully selected during the offline phase, so as to verify that direct ROM interpolation is robustness.




(a) interpolation between $\mathcal{E}_{rel} < 10^{-2}$ accurate ROMs

(b) interpolation between $\mathcal{E}_{rel} < 10^{-4}$ accurate ROMs

Figure 8: Convergence rate of ROM at 5 deg obtained through direct interpolation. Tabulated ROMs of different accuracy ($\mathcal{E}_{rel}$) on a grid with spacing $\Delta\alpha$ =10 deg are considered.

Weak and strong projection MAC for BT basis are further analysed in Fig. 9. Here, the accuracy of the interpolated ROMs $\widetilde{\alpha}$ =5 deg is shown as a function of the spacing, $\Delta\alpha$, used to define the tabulated data grid, $\boldsymbol{a}(\Delta\alpha)$ — Eq. (38). Interpolation is carried through ROMs of different size, and considers both projection over the closest basis, as well as over the interpolated projection basis provided by Eq. (22) and (23). Moreover, to overcome the fact that the ROMs rate of convergence does not always decrease with their order (Fig. 7b and 8b), the number of reduced-states retained upon interpolation is here chosen so as to minimize the norm $\mathcal{E}_{rel}[\widetilde{\boldsymbol{G}_\boldsymbol{r}}, \gamma_i \boldsymbol{G}_{\boldsymbol{r}(i)}]$, where $\gamma_i \boldsymbol{G}_{\boldsymbol{r}(i)}$ estimates of the frequency response at $\widetilde{\alpha}$ through transfer function interpolation, Eq. (29). In all cases, the error expectedly increases with the grid spacing $\Delta\alpha$, due to both interpolation and projection error. The strong MAC allows producing more accurate ROMs than weak MAC. However, as shown by interpolation amongst 20 order ROMs, this can fail catastrophically when the reduced states are not well correlated. Weak projection, instead, shows robustness throughout

the whole $\Delta\alpha$ and ROM order range. Finally, projection over the interpolated basis allows in most cases to marginally reduce the interpolation error, but at a higher computational cost.

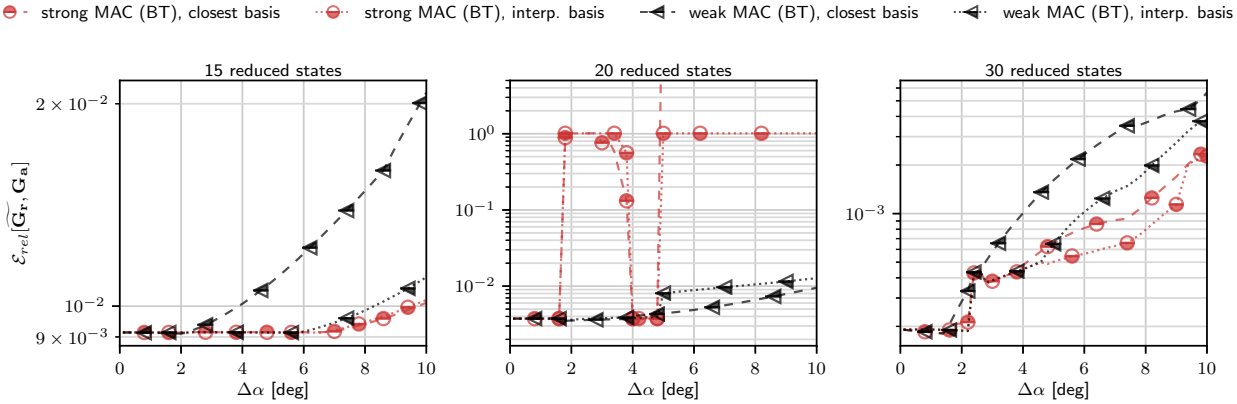



Figure 9: Direct interpolation accuracy as a function of the grid spacing, $\Delta\alpha$, around 5 deg.

In conclusion, direct ROM interpolation with weak MAC projection over the closest basis is the faster and more reliable method, with minimal loss of accuracy with respect to an interpolated basis (Fig. 9). However, due to the discrete-time nature of the UVLM, interpolation is carried on over the manifold of real matrices. In order to bypass this limitation, therefore, transfer function interpolation is investigated next.

### *V.B.3. Transfer function (TF) interpolation amongst balanced models*

As introduced in Sec. IV.C, TF interpolation bypasses the need to project the tabulated state-space models upon a common basis and offers an appealing alternative to increase robustness and accuracy. Interpolated ROMs can be obtained at virtually no cost through the realization in Eq. (29), although the size of the interpolated ROM grows with the interpolation order. Alternatively, a frequency-limited balanced expression of Eq. (29) can be derived directly (FLB-TF), which results in interpolation amongst balanced, non-truncated, models and also requires an SVD, Eq. (33). Those trade-offs are assessed here.

TF interpolation amongst FLBT ROMs, Eq. (29), and the (non-truncated) FLB-TF, Eq. (35), are first compared. Interpolation is carried out again at $\widetilde{\alpha} = 5\,\text{deg}$ and from 96-state frequency-limited balanced models (obtained through quadrature with $\mathcal{K}_{low} = 6$ and $\mathcal{K}_{high} = 12$) defined over a fine grid, $\boldsymbol{a}(\Delta\alpha = 1\,\text{deg})$. Fig. 10 compares the transfer-function error with the FLBT ROM at $\widetilde{\alpha} = 5\,\text{deg}$ (FLBT) of (a) TF interpolation amongst FLBT ROMs of the same order and (b) ROMs obtained by FLB-TF interpolation and truncation. Note that in Fig. 10 the horizontal axis refers to the order of the interpolated ROM. TF interpolation amongst ROMs shows high accuracy (up to $\mathcal{E}_{rel} \approx 6 \times 10^{-5}$) but, as a result of Eq. (29), also a slow convergence rate. FLB-TF, however, allows retrieving the same fast convergence rate of direct FLBT. Overall, Fig. 10 shows that both approaches for TF interpolation have similar fidelity. The choice between one method or the other, therefore, depend on one desired to minimize the interpolated ROM size

or computational cost.

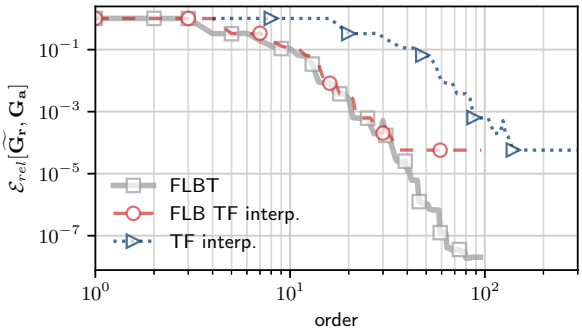



Figure 10: Balanced vs unbalanced TF interpolation convergence rate on a $\boldsymbol{a}(\Delta\alpha = 1\,\text{deg})$ grid.

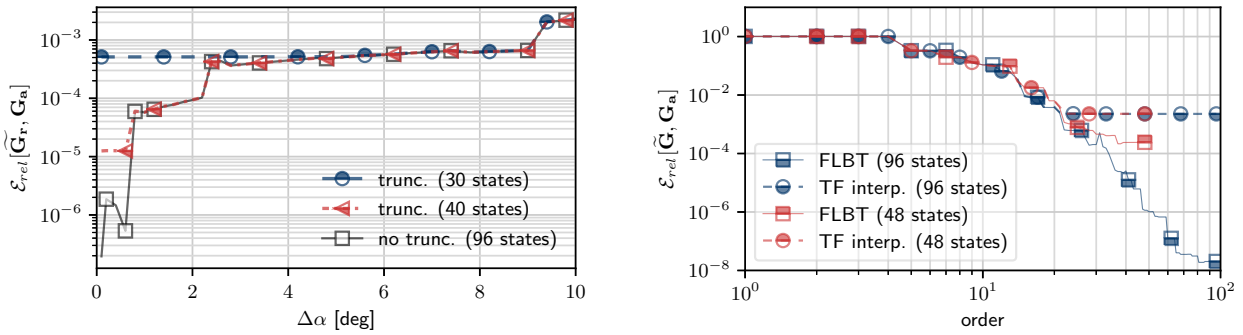
The accuracy of FLB-TF interpolation is analysed in more detail next in Fig. 11. Relatively large frequency-limited balanced models with 96 states ($\mathcal{K}_{low} = 6, \mathcal{K}_{high} = 12$) are produced over uniform grids of different spacing $\boldsymbol{a}(\Delta\alpha)$ around the interpolation point $\widetilde{\alpha}$ =5 deg as per Eq. (38). As shown in Fig. 11a, FLB-TF provides great accuracy even with large $\Delta\alpha$. With reference to the offline interpolation (Fig. 5a), this is only an order of magnitude less accurate. However, no re-assembly of the state-space model at $\widetilde{\alpha}$ is required, but rather an SVD of size $(96 \times 576)$ needs to be solved in Eq. (33). Fig. 11a also includes the error in ROMs of order 30 and 40 obtained by further truncation of the FLB-TF interpolated state-space model. With reference to Fig. 9, FLB-TF interpolation achieves higher or equal accuracy than direct ROM interpolation. However, this process is robust and there is no need to down-select the optimal ROM order. As $\Delta\alpha$ is increased, the error introduced by the interpolation scheme becomes dominant, and all curves in Fig. 11a collapse over each other. This suggests that, once the accuracy of the sought ROMs is fixed, the size of the tabulated FLB models can be reduced, so as to minimize the online computational cost.

This concept is further exemplified in Fig. 11b for interpolation over a $\boldsymbol{a}(\Delta\alpha = 10\,\text{deg})$ grid. Here, FLB-TF interpolation is also performed amongst smaller size FLB models obtained upon Gramians integration over a $(\mathcal{K}_{low}, \mathcal{K}_{high}) = (3, 2)$. The convergence rate of the truncated, FLB-TF interpolated, ROMs is compared to that of FLBT. As the $\boldsymbol{a}$ grid is course, the maximum accuracy of the interpolated ROMs is in all cases bounded by the interpolation error at $\mathcal{E}_{rel} \approx 2 \times 10^{-3}$. Therefore, $\mathcal{E}_{rel} < 10^{-2}$ accurate ROMs can be achieved through interpolation of small-size, 48-states, FLB models, at the cost of a $48 \times 160$ SVD.

Overall, TF interpolation provides more robustness than direct ROM interpolation with minimal penalty on the online computational time required to interpolate.

### *V.B.4. Application to aeroelastic analysis*

ROM interpolation is here finally employed for the parametric flutter analysis already presented in Sec. V.A (Fig. 4). To this aim, FLB realizations of the T-tail aerodynamics are produced at equally-spaced incidence




(a) Dependency on interpolation grid spacing, $\Delta\alpha$ (b) Convergence rate for interpolation over a 10 deg grid

Figure 11: Relative error and convergence rate of UVLM ROMs obtained through TF interpolation at 5 deg.

angles between $-10$ and 20 deg, with sampling steps $\Delta\alpha = 5$ and 10 deg. That means that for the coarser grid, only four realisations of the LUVLM are used. Only the two best performing strategies (direct ROM interpolation with weak MAC projection and FLB-TF intertpolation) are considered. The interpolated models are finally coupled with the structural description in the normalized time and the flutter features are evaluated with a tolerance $10^{-8}$ as explained in Sec. V.A.

Fig. 12 shows the flutter speed relative error of FLB-TF interpolation with respect to a high-fidelity direct model-order reduction with $(\mathcal{K}_{low}, \mathcal{K}_{high}) = (6, 12)$ at each $\alpha$. At first, the case in which the interpolated 96-states models are not further truncated is discussed. In this situation, the flutter speed error is dominated by the interpolation accuracy. With this respect, it is worth noticing that, as this tends to zero at the tabulated points, a lower bound has been set to the $y$ axis of Fig. 12. Interpolation on the course grid with sampling $\Delta\alpha$ =10 deg, leads to a maximum relative error of 6 % at $\alpha \approx -5$ deg. In particular, this is considerably larger than what would be expected from the frequency response analysis at the same point, where the interpolated model features $\mathcal{E}_{rel}[\widetilde{\boldsymbol{G}_\boldsymbol{b}}, \boldsymbol{G}_\boldsymbol{a}] < 5 \times 10^{-3}$. The loss of accuracy is due to the slow rate at which the flutter eigenvalue moves towards the instability region as the flow speed, $U$, is increased. Therefore, for design applications the sampling period needs to be carefully monitored so as to ensure that the interpolated ROMs have precision several order of magnitude higher than the desired accuracy in the flutter analysis. This is demonstrated by the $\Delta\alpha = 5$ deg curves in Fig. 12: FLB-TF interpolation is highly accurate ($\mathcal{E}_{rel}[\widetilde{\boldsymbol{G}_\boldsymbol{b}}, \boldsymbol{G}_\boldsymbol{a}] < 8 \times 10^{-4}$) throughout the full $\alpha$ range, while flutter errors are just below 3 %. As already discussed in Sec. V.B.3, when the interpolation accuracy is set by the sampling period, $\Delta\alpha$, smaller size FLB models can be used with no loss of accuracy. Furthermore, as interpolated ROMs preserve the same fast rate of convergence of FLBT, truncation can be employed to further reduce the size of the ROMs. This is demonstrated here for flutter analysis. To this purpose, ROMs have been produced by interpolation on the $\Delta\alpha = 5$ deg grid amongst 48-states FLB models and further truncation to 20 states. As shown in

Fig. 12, no relevant loss of accuracy is observed. However, as per Sec. V.B.3, the final ROM size and the interpolation cost are now considerably lower.

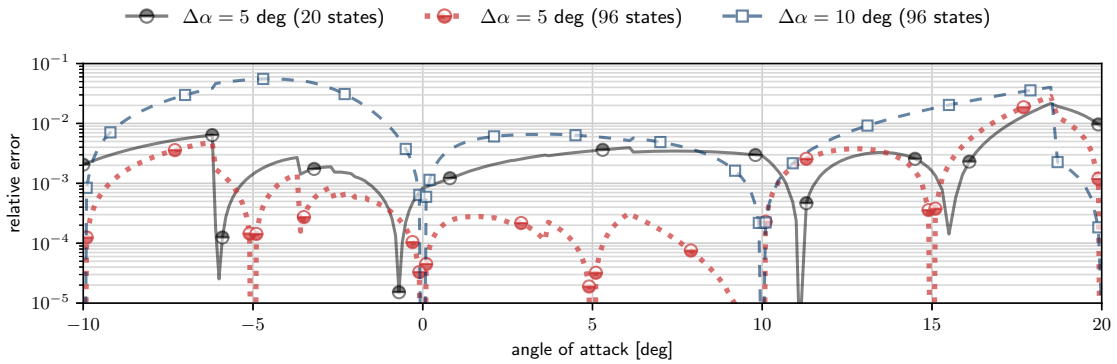



Figure 12: Error in the flutter speed computed via TF interpolation between balanced state-space models.

## VI. Conclusions

This work has investigated methods for parametric model reduction of the linearised unsteady-vortex lattice equations. First, it has introduced a highly-efficient solution to the frequency-domain equations, which naturally arises from a $z$-transform of its discrete-time state-space realisation and thus retains the full modelling flexibility of the underlying formulation. Closed-form solutions for the (frozen) wake states have resulted in algorithms that are highly efficient in both computational and memory requirements for both frequency-response analysis and balanced-truncation on limited frequencies. In particular, the excellent numerical performance of that approach allows for FLBT to be obtained through direct numerical integration of frequency-based low-rank expressions for the Gramians of the LUVLM. Stability-preserving ROMs have been derived by integrating (on a very coarse grid about the maximum frequency of interest) the controllability Gramian to the Nyquist frequency. An investigation into techniques for ROM interpolation has finally extended the methods to general parametrizations of the LUVLM. Three classes of interpolation strategies, associated to different trade-off levels between accuracy, robustness and computational cost, have been here considered, therefore providing a wide spectrum of methods for applications in loads analysis, design optimisation and model-based control.

Numerical studies have considered the aeroelastic stability of a T-tail configuration with varying incidence angle. Our FLBT algorithm has been shown to scale linearly with the number of wake states. This has resulted in realisations of large UVLM lattices, with tens of thousands of states, for which ROMs are generated in a few seconds and with minimal memory overhead. The resulting ROMs are highly-accurate, preserve stability, and present faster convergence rates with the number of states than those obtained by (internally) balanced residualisation.

Finally, studies on parametric ROMs have focused on the trade-off between number of sampling points and interpolation accuracy. ROMs obtained through interpolation of the Gramians factors and re-assembly of the full-states system matrices have been shown to be extremely accurate. However, interpolating ROMs in this manner provides minimal computational saving due to the cost of re-assembling large LUVLM state-space models. Direct interpolation amongst balanced truncated ROMs has resulted in a computationally fast process suitable for real-time applications. However, it was also less accurate due to the need of projecting the ROMs over a common basis before interpolation, with unavoidable loss of information. The robustness of the approach has also been shown to depend strongly on the projection method employed, which therefore requires carefully assessment of the quality of the sampled data for online interpolations in real-time applications. Nonetheless, using a weak projection method, direct ROM interpolation has successfully predicted the T-tail flutter speed over a wide range of angles with 3 % accuracy and using a relatively course grid of sampled data.

The transfer-function interpolation approach has been finally shown to provide an equally accurate, but more robust and easy to use, alternative to direct interpolation. Real-time application of this technique is possible but limited to time marching solution due to the growth in size of the interpolated ROM. For application in design optimisation, instead, a novel solution, which directly produces a FLB realisation of the TF interpolated model, has also been proposed. This provides equal accuracy with minimal computational overhead.

## Acknowledgements

This work was made possible through funding provided by the UK Engineering and Physical Sciences Research Council grant EP/N006224/1. All the source code written for this paper is available within the open-source aeroelastic simulation framework `SHARPy` at `www.imperial.ac.uk/aeroelastics/sharpy`.

## References

[1] Voutsinas, S. G., "Vortex methods in aeronautics: how to make things work," *International Journal of Computational Fluid Dynamics*, Vol. 20, No. 1, 2006, pp. 3–18. doi: 10.1080/10618560600566059.

[2] Stanford, B. K. and Beran, P. S., "Analytical Sensitivity Analysis of an Unsteady Vortex Lattice Method for Flapping Wing Optimization," *Journal of Aircraft*, Vol. 47, No. 2, mar 2010, pp. 647–662. doi: 10.2514/1.46259.

[3] Wang, Z., Chen, P. C., Liu, D. D., and Mook, D. T., "Nonlinear-Aerodynamics/Nonlinear-Structure Interaction Methodology for a High-Altitude Long-Endurance Wing," *Journal of Aircraft*, Vol. 47, No. 2, 2010, pp. 556–566. doi: 10.2514/1.45694.

[4] Katz, J. and Plotkin, A., *Low-Speed Aerodynamics*, Cambridge University Press, New York, 2001. doi: 10.1017/CBO9780511810329.

[5] Albano, E. and Rodden, W. P., "A doublet lattice method for calculating lift distributions on oscillating surfaces in

subsonic flows," *AIAA Journal*, Vol. 7, No. 2, 1969, pp. 279–285. doi: 10.2514/3.5086.

[6] Simpson, R. J. S., Palacios, R., and Murua, J., "Induced-Drag Calculations in the Unsteady Vortex Lattice Method," *AIAA Journal*, Vol. 51, No. 7, 2013, pp. 1775–1779. doi: 10.2514/1.J052136.

[7] Ng, B., Palacios, R., Kerrigan, E. C., Graham, J. M. R., and Hesse, H., "Aerodynamic load control in horizontal axis wind turbines with combined aeroelastic tailoring and trailing-edge flaps," *Wind Energy*, Vol. 19, No. 2, 2016, pp. 243–263. doi: 10.1002/we.1830.

[8] Hesse, H. and Palacios, R., "Dynamic Load Alleviation in Wake Vortex Encounters," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 4, 2016, pp. 801–813. doi: 10.2514/1.G000715.

[9] Murua, J., Martinez, P., Climent, H., van Zyl, L., and Palacios, R., "T-tail flutter: Potential-flow modelling, experimental validation and flight tests," *Progress in Aerospace Sciences*, Vol. 71, 2014, pp. 54–84. doi: 10.1016/j.paerosci.2014.07.002.

[10] Murua, J., Palacios, R., and Graham, J. M. R., "Applications of the unsteady vortex-lattice method in aircraft aeroelasticity and flight dynamics," *Progress in Aerospace Sciences*, Vol. 55, 2012, pp. 46–72. doi: 10.1016/j.paerosci.2012.06.001.

[11] Ng, B., Hesse, H., Palacios, R., Graham, J., and Kerrigan, E., "Aeroservoelastic state-space vortex lattice modeling and load alleviation of wind turbine blades," *Wind Energy*, Vol. 18, No. 7, 2015, pp. 1317–1331. doi: 10.1002/we.1752.

[12] Roesler, B. T. and Epps, B. P., "Discretization Requirements for Vortex Lattice Methods to Match Unsteady Aerodynamics Theory," *AIAA Journal*, Vol. 56, No. 6, 2018, pp. 2478–2483. doi: 10.2514/1.j056400.

[13] Werter, N. P. M., De Breuker, R., and Abdalla, M., "Continuous-Time State-Space Unsteady Aerodynamic Modeling for Efficient Loads Analysis," *AIAA Journal*, Vol. 56, No. 3, 2018, pp. 905–916. doi: 10.2514/1.J056068.

[14] Willis, D. J., Peraire, J., and White, J. K., "A Combined pFFT-Multipole Tree Code, Unsteady Panel Method with Vortex Particle Wakes," *International Journal for Numerical Methods in Fluids*, Vol. 53, 2007, pp. 1399–1422. doi: 10.1002/fld.1240.

[15] Baker, M., Mingori, D., and Goggin, P., "Approximate Subspace Iteration for constructing internally balanced reduced order models of unsteady aerodynamic systems," *37th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference and Exhibit*, Salt Lake City, Utah, USA, April 1996, pp. 1070–1085. doi: 10.2514/6.1996-1441.

[16] Rule, J., Cox, D., and Clark, R., "Aerodynamic Model Reduction Through Balanced Realization," *AIAA Journal*, Vol. 42, No. 5, 2004, pp. 1045–1048. doi: 10.2514/1.9596.

[17] Maraniello, S. and Palacios, R., "State-Space Realizations and Internal Balancing in Potential-Flow Aerodynamics with Arbitrary Kinematics," *AIAA Journal*, 2019, pp. 1–14. doi: 10.2514/1.J058153.

[18] Antoulas, A. C., *Approximation of Large-Scale Dynamical Systems*, Advances in Design and Control, SIAM, Philadelphia, 2005. doi: 10.1137/1.9780898718713.

[19] Simoncini, V., "Computational Methods for Linear Matrix Equations," *SIAM Review*, Vol. 58, No. 3, 2016, pp. 377–441. doi: 10.1137/130912839.

[20] Jbilou, K., "Low rank approximate solutions to large Sylvester matrix equations," *Applied Mathematics and Computation*, Vol. 177, No. 1, 2006, pp. 365–376. doi: 10.1016/j.amc.2005.11.014.

[21] Sadkane, M., "A low-rank Krylov squared Smith method for large-scale discrete-time Lyapunov equations," *Linear Algebra and Its Applications*, Vol. 436, No. 8, 2012, pp. 2807–2827. doi: 10.1016/j.laa.2011.07.021.

[22] Li, T., Weng, P. C. Y., wah Chu, E. K., and Lin, W. W., "Large-scale Stein and Lyapunov equations, Smith method, and applications," *Numerical Algorithms*, Vol. 63, No. 4, 2013, pp. 727–752. doi: 10.1007/s11075-012-9650-2.

[23] Bartels, R. H. and Stewart, G. W., "Solution of the Matrix Equation AX + XB = C [F4]," *Communications of the ACM*, Vol. 15, No. 9, 1972, pp. 820–826. doi: 10.1145/361573.361582.

[24] Laub, A. J., Heath, M. T., Paige, C. C., and Ward, R. C., "Computation of System Balancing Transformations and Other Applications of Simultaneous Diagonalization Algorithms," *IEEE Transactions on Automatic Control*, Vol. 32, No. 2, 1987, pp. 115–122. doi: 10.1109/TAC.1987.1104549.

[25] Hammarling, S., "Numerical solution of the discrete-time, convergent, non-negative definite Lyapunov equation," *Systems and Control Letters*, Vol. 17, No. 2, 1991, pp. 137–139. doi: 10.1016/0167-6911(91)90039-H.

[26] Lieu, T., Farhat, C., and Lesoinne, M., "Reduced-order fluid/structure modeling of a complete aircraft configuration," *Computer Methods in Applied Mechanics and Engineering*, Vol. 195, No. 41-43, 2006, pp. 5730–5742. doi: 10.1016/j.cma.2005.08.026.

[27] Lieu, T. and Farhat, C., "Adaptation of Aeroelastic Reduced-Order Models and Application to an F-16 Configuration," *AIAA Journal*, Vol. 45, No. 6, 2007, pp. 1244–1257. doi: 10.2514/1.24512.

[28] Amsallem, D. and Farhat, C., "Interpolation Method for Adapting Reduced-Order Models and Application to Aeroelasticity," *AIAA Journal*, Vol. 46, No. 7, 2008, pp. 1803–1813. doi: 10.2514/1.35374.

[29] Amsallem, D., Cortial, J., and Farhat, C., "Towards Real-Time Computational-Fluid-Dynamics-Based Aeroelastic Computations Using a Database of Reduced-Order Information," *AIAA Journal*, Vol. 48, No. 9, 2010, pp. 2029–2037. doi: 10.2514/1.J050233.

[30] Allison, J. T. and Herber, D. R., "Multidisciplinary Design Optimization of Dynamic Engineering Systems," *AIAA Journal*, Vol. 52, No. 4, 2014, pp. 691–710. doi: 10.2514/1.J052182.

[31] Franklin, G. and Powell, D., *Digital Control of Dynamic Systems*, Addison-Wesley Publishing Company, 1980.

[32] Dimitriadis, G., Giannelis, N. F., and Vio, G. A., "A modal frequency-domain generalised force matrix for the unsteady vortex lattice method," *Journal of Fluids and Structures*, Vol. 76, 2018, pp. 216–228. doi: 10.1016/j.jfluidstructs.2017.10.010.

[33] Phillips, J. R. and Silveira, L. M., "Poor man's TBR: a simple model reduction scheme," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 24, No. 1, jan 2005, pp. 43–55. doi: 10.1109/TCAD.2004.839472.

[34] Baur, U., Benner, P., and Feng, L., "Model Order Reduction for Linear and Nonlinear Systems: A System-Theoretic Perspective," *Archives of Computational Methods in Engineering*, Vol. 21, No. 4, 2014, pp. 331–358. doi: 10.1007/s11831-014-9111-2.

[35] Willcox, K. and Peraire, J., "Balanced Model Reduction via the Proper Orthogonal Decomposition," *AIAA Journal*, Vol. 40, No. 11, 2002, pp. 2323–2330. doi: 10.2514/2.1570.

[36] Gugercin, S. and Antoulas, A. C., "Model reduction of large-scale systems by least squares," *Linear Algebra and Its Applications*, Vol. 415, No. 2-3, 2006, pp. 290–321. doi: 10.1016/j.laa.2004.12.022.

[37] Benner, P., Kürschner, P., and Saak, J., "Frequency-Limited Balanced Truncation with Low-Rank Approximations," *SIAM Journal on Numerical Analysis*, Vol. 38, No. 1, 2016, pp. A471–A499. doi: 10.1137/15M1030911.

[38] Gawronski, W. and Juang, J. N., "Model reduction in limited time and frequency intervals," *International Journal of Systems Science*, Vol. 21, No. 2, 1990, pp. 349–376. doi: 10.1080/00207729008910366.

[39] Gugercin, S. and Antoulas, A. C., "A survey of model reduction by balanced truncation and some new results," *International Journal of Control*, Vol. 77, No. 8, 2004, pp. 748–766. doi: 10.1080/00207170410001713448.

[40] Ghafoor, A. and Sreeram, V., "A Survey/Review of Frequency-Weighted Balanced Model Reduction Techniques," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 130, No. 6, 2008, pp. 061004–061004–16. doi: 10.1115/1.2977468.

[41] Imran, M. and Ghafoor, A., "Model reduction of descriptor systems using frequency limited Gramians," *Journal of the Franklin Institute*, Vol. 352, No. 1, 2015, pp. 33–51. doi: 10.1016/j.jfranklin.2014.10.013.

[42] Benner, P., Gugercin, S., and Willcox, K., "A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems," *SIAM Review*, Vol. 57, No. 4, 2015, pp. 483–531. doi: 10.1137/130932715.

[43] Amsallem, D. and Farhat, C., "An Online Method for Interpolating Linear Parametric Reduced-Order Models," *SIAM Journal on Scientific Computing*, Vol. 33, No. 5, 2011, pp. 2169–2198. doi: 10.1137/100813051.

[44] Panzer, H., Mohring, J., Eid, R., and Lohmann, B., "Parametric Model Order Reduction by Matrix Interpolation," *At-Automatisierungstechnik*, Vol. 58, No. 8, 2010, pp. 475–484. doi: 10.1524/auto.2010.0863.

[45] Son, N. T., "A real time procedure for affinely dependent parametric model order reduction using interpolation on Grassmann manifolds," *International Journal for Numerical Methods in Engineering*, Vol. 93, 2013, pp. 818–833. doi: 10.1002/nme.4408.

[46] Baur, U. and Benner, P., "Model Reduction for Parametric Systems Using Balanced Truncation and Interpolation," *At-Automatisierungstechnik*, Vol. 57, No. 8, 2009, pp. 411–419. doi: 10.1524/auto.2009.0787.

[47] Hesse, H. and Palacios, R., "Reduced-Order Aeroelastic Models for Dynamics of Maneuvering Flexible Aircraft," *AIAA Journal*, Vol. 52, No. 8, 2014, pp. 1–16. doi: 10.2514/1.J052684.

[48] Penzl, T., "Algorithms for model reduction of large dynamical systems," *Linear Algebra and Its Applications*, Vol. 415, No. 2-3, 2006, pp. 322–343. doi: 10.1016/j.laa.2006.01.007.

[49] Enns, D. F., "Model Reduction with Balanced Realization: an Error Bound and a Frequency Weighted Generalization," *Proceedings of the 23rd Conference on Decision and control*, Las Vegas, Nevada, USA, Dec 1984, pp. 127–132. doi: 10.1109/CDC.1984.272286.

[50] Lieu, T. and Lesoinne, M., "Parameter Adaptation of Reduced Order Models for Three-Dimensional Flutter Analysis," *42nd AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, USA, Jan 2004, pp. 1–9. doi: 10.2514/6.2004-888.

[51] Mahony, R., Sepulchre, R., and Absil, P.-A., "Riemannian geometry of Grassmann manifolds with a view on algorithmic computation," *Acta Applicandae Mathematicae*, Vol. 80, No. 2, 2004, pp. 199–220. doi: 10.1023/B:ACAP.0000013855.14971.91.

[52] Geuss, M., Panzer, H., and Lohmann, B., "On parametric model order reduction by matrix interpolation," *2013 European Control Conference*, July 2013, pp. 3433–3438. doi: 10.23919/ECC.2013.6669829.

[53] Schönemann, P. H., "A Generalized Solution of the Orthogonal Procrustes Problem," *Psychometrika*, Vol. 31, No. 1, 1966, pp. 1–10.