AIAA Aviation Forum and ASCEND co-located Conference Proceedings
29 July - 2 August 2024, Las Vegas, Nevada
AIAA AVIATION FORUM AND ASCEND 2024
10.2514/6.2024-4444

Store Trajectory Prediction Using Domain-Decomposed Reduced-Order Modelling

Navdeep Pandey*[1], Jaysinh Jagdishchandra Patel†[2], and Aniruddha Sinha‡[1]
[1]*Dept. of Aerospace Engineering, Indian Institute of Technology Bombay, Mumbai 400076, INDIA*
[2]*Defence Research and Development Laboratory, Hyderabad 500058, INDIA*

Multi-body aerodynamic analysis is crucial whenever a store is designed to separate from its parent aircraft, so as to predict the safe flight envelope. This involves a meticulous analysis of the store trajectory across a large range of operating parameters (i.e., freestream conditions). The present work demonstrates an efficient albeit approximate technique that marries a well-known technique – viz. reduced-order model (ROM) based on proper orthogonal decomposition (POD) – with a domain-decomposed methodology, for predicting the flow field around a store-aircraft dyad. The domain-decomposition approach allows POD-based ROM to be used for the majority of the flow domain, keeping only a small region that must be addressed by full-order simulations, that too Euler calculations. We had exemplified the approach earlier with a toy two-dimensional case; here we extend it to a much more realistic three-dimensional benchmark problem. As it stands now, the final result from our foray is actually subpar vis-à-vis the much simpler linear interpolation. However, we have identified several components of the approach that when modified should deliver a much more satisfactory performance.

I. Introduction

Whenever a new aircraft and/or its stores (missiles, drop tanks, etc.) are developed, or an existing aircraft/store combination undergoes some modifications, the dyad has to undergo an extensive investigation to certify the safe-separation flight envelope. During the analysis, trajectories followed by the store under various operating conditions are predicted to ensure that it does not strike back at the aircraft itself. There are mainly three conventional approaches to conduct such investigations – flight testing, experiments in a wind tunnel, and simulation using computational fluid dynamics (CFD). All of them pose some challenges related to economic and practical viability; moreover, they are all time-consuming.

There are mainly two types of computational approaches to the store-trajectory separation problem, mimicking the corresponding experimental techniques. The first one assumes the instantaneous problem to be quasi-steady (see fig. 1). That is, when the store is at a particular position from the aircraft, then this may be considered to be a steady problem. The aerodynamic forces and moments on the store may be calculated from the corresponding steady flow solution (typically steady Reynolds-averaged Navier Stokes – RANS – is used). The resulting linear and angular accelerations of the store may then be integrated over a short time horizon (Δt) to obtain the new position and orientation of the store. Then these steps may be iterated to obtain the overall store separation trajectory. The overall strategy mimics the captive trajectory system (CTS) experimental approach to the problem. The alternative experimental approach is flight testing. Its computational counterpart replaces steady RANS with the unsteady variant to account for the small effect this may have on fidelity. Moreover, unlike in the first approach, the instantaneous linear and angular velocities of the store are applied as boundary conditions in the simulation instead of being neglected. It turns out that the second approach, although appearing to be more complex, is actually more efficient since the flow solution only changes by small increments at each time step.

Here, we carry forward a store trajectory estimation approach based on empirical reduced-order modelling (ROM) integrated with domain-decomposition that we proposed recently [1, 2]. It is an efficient, albeit approximate, alternative to the steady RANS-based approach described above. The empiricism in the ROM implies the existence of a prior 'learning' database, which typically comes from a small set of preliminary CFD calculations that sparsely sample the parameter space. Such computations are referred to as full-order model (FOM) results in this work, to distinguish them from the output of the proposed ROM. Subsequently, the ROM predicts the flow and/or derived quantities at

*Graduate student; AIAA Student Member
†Scientist E
‡Associate Professor; AIAA Member; Corresponding author: as@aero.iitb.ac.in

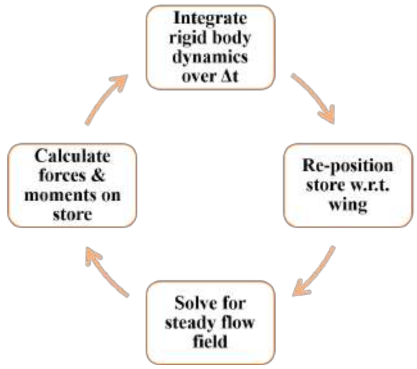


Fig. 1 Flow chart representing the iterative steps involved in store trajectory prediction with the FOM approach, using time integration with a step size of Δt.

unsampled parameter values. ROM is a well-known tool to predict the characteristics of the flow field very efficiently. Its ability to predict flows with a small turnaround time requiring minimal computational resources makes it suitable for applications where rapid design decisions have to be made – e.g., multi-disciplinary analysis and optimization. Evidently, the prediction of store-separation trajectory is a candidate for the employment of ROMs.

The proposed ROM uses proper orthogonal decomposition (POD) to identify the underlying reduced-order topology of the flow. Most of such spatial modal analyses, including POD, necessitate that all the empirical flow data supplied are for the same geometry (and mesh). In the case of store trajectory prediction, the store changes its relative position with respect to the aircraft continuously after its release. The corresponding empirical database thus has arbitrary alteration of the flow geometry. Consequently, the usual POD-ROM approach for single-body aerodynamics cannot be employed for the full-flow domain here. Considering this limitation, the proposed approach decomposes the full-domain into multiple subdomains such that POD-ROM can be employed for the maximal portion of the flow domain; the remaining subdomain that involves changing geometry/mesh is solved using the full-order model (FOM). The flow has to be matched at the interfaces of the subdomains through a few iterations of the ROM(s) and FOM calculations. The overall approach is termed domain-decomposed ROM (DDROM). Our previous work [1, 2] in this regard were on a toy two-dimensional problem, where we did not actually calculate the store separation trajectory but demonstrated the feasibility of the approach. Here, we extend the work by addressing a much more complicated and realistic three-dimensional problem.

The paper is organized as follows. The background of the work is established in section II, and the theory underlying the approach is described in section III. This is followed by a presentation of the results and associated discussions in section IV, and is concluded in section V.

II. Background

We start with a brief background of existing methods for trajectory prediction in Section II.A; subsequently, we discuss the previous applications of ROMs to similar problems in Section II.B.

A. Existing methods for trajectory prediction

Store trajectories have historically been predicted using flight tests, wind tunnel experiments and/or computational fluid dynamics (CFD) simulations. We briefly summarize these approaches; see Ref. [3] for an all-round review.

The very first attempts to predict the trajectory of stores were pursued using flight testing. The store was repeatedly dropped from the aircraft flying at gradually increasing speeds, until the store was observed to come back closer to the aircraft after its release instead of separating monotonically. This method was based on trial-error, and in some cases, this led to the store hitting the aircraft itself. Although flight testing continues to be the ultimate evaluation, it is used very sparingly due to the costs and risks involved. In the early 1960's, wind tunnel model testing for trajectory prediction was developed using the Captive Trajectory System (CTS) [4, 5]; its flowchart is shown in fig. 1. The store model was fixed below the aircraft model on a sting with a 6-axis force/moment balance. The 'Solve for steady flow field' step was performed by the wind tunnel flow itself. The forces and moments determined in this configuration were used for integrating the 6-DOF (six degrees of freedom) rigid body equations over a small time horizon using an online computer, and the changes in position and orientation of the store thus calculated were implemented for the sting-mounted store model. This procedure was repeated until the store was deemed to have moved out of the influence region of the aircraft. Essentially, this was a quasi-steady simulation of the separation trajectory with the online computer affording much faster evaluation than could have been achieved with human intervention. Since wind tunnel tests typically use small-scale models, large deviations were observed sometimes in comparison with the more reliable reference flight test data. Moreover, the quasi-steady nature of the simulation as opposed to a time-accurate one could also have been a source of error. Thus, the wind tunnel tests were employed to identify potentially critical flight parameters that then were used to design a more pared down flight test matrix.

In the 1980's, the Influence Function Method (IFM) [6] was developed to reduce the wind tunnel test matrix. Detailed experiments needed to be conducted for a reference store geometry in the influence of a parent aircraft, as well as any new store geometry in isolation. The IFM used these as inputs to predict the trajectory of the new store in the vicinity of the aircraft. It was assumed that while the aircraft influences the stores' aerodynamics, the stores had no effect on the aircraft. Over time, as CFD capabilities were growing, it was used to predict the flow field around the aircraft, whereas wind tunnel tests continued to be used for accurate characterization of the isolated store's aerodynamics [7].

Today CFD capabilities have matured sufficiently so as to minimize the dependence on flight and wind tunnel tests. However, it is these diligent tests that have provided indispensable validation to the CFD approaches so that they could arrive at their present state. Of particular note are three programs sponsored by U.S. Air Force and Navy research organizations in the 1990's that made available detailed test data of well-documented cases and challenged the CFD community to submit their attempts at replicating the same [3]. Over the years, CFD models went through linear potential flow models [8], nonlinear full potential methods coupled with thin viscous layers [9], (inviscid) Euler calculations [10], and (viscous) Reynolds Averaged Navier Stokes (RANS) methods [11]. These advances in fidelity have been made possible by the exponential growth of computational power, as well as rapid developments in algorithms, particularly regarding overset or Chimera grids [12]. At present, most production simulations of store separation trajectories are pursued with RANS [e.g. 13].

The CFD approaches to this kind of problems can be categorized as quasi-steady or time-accurate. The former mimics the CTS method of wind tunnel testing whose flowchart is given in fig. 1. In particular, the fluid solver assumes a stationary store (relative to the parent aircraft), and solves for the steady flow field. The time-accurate CFD method refers to unsteady RANS (URANS) simulation of the flow field, with the time-varying flow field and moving store accounted for in the fluid dynamic equations. An early comparison of the two approaches [14] showed that the quasi-steady method indeed replicated CTS data, and differed slightly in its results from the time-accurate approach. Owing to the improved availability of computational resources, most current production simulations are carried out in the time-accurate mode [e.g. 15–18]. However, they continue to be validated against CTS data [19], which indicates their minimal differences from the quasi-steady approach.

In our work, we will develop a ROM for the quasi-steady viscous store-separation trajectory prediction problem in a three-dimensional setting. Extension to the time-accurate approach will be pursued in the future.

B. Application of ROM to similar problems

Applications of reduced-order modelling in flow problems can be broadly divided into three categories – steady (parameter-varying) ROM, unsteady (time-varying) ROM for a single parameter set, and a combination of the two. In problems dealing with time-varying flow characteristics, unsteady ROM is employed to predict the flow behaviour at different time instants. Steady ROM finds its applications in the problems where steady (or time-averaged) flow features (for a particular parameter in a set) are sufficient for the engineering purpose at hand. In the present work, we are interested in steady ROMs for the reason described in section II.A.

The application of parameter-varying steady ROM to single-body aerodynamics starts with the generation of the 'learning database' comprising of 'snapshots' of the steady flow field for a sufficiently rich, yet sparsely-sampled, set of operating parameters (e.g., Mach number, angle of attack, side slip angle, etc.). This is typically obtained by solving the FOM – i.e., doing Euler or RANS simulations; rarely do we get this empirical data from experiments. This is the one-time cost of the ROM, albeit a major one. The next step involves the reduction of the order of the problem by identifying the underlying simplicity in the flow features present in the empirical database. Mathematically, we compute a minimal set of linear basis functions or modes of the data such that maximal information is captured. The most common tool used for this is proper orthogonal decomposition (POD) [20–22]. The last step is the prediction of flow field behaviour for a new set of parameters, typically as a linear combination of the basis modes using appropriate weighting coefficients. This step can be performed in two ways – (i) interpolation of the coefficients from the learning database [e.g. 23–25], or (ii) a physics-informed approach called steady ROM wherein the coefficients are determined by minimizing the residual of the steady governing equations constrained by the boundary conditions [22, 26–30]. In the literature, they are also termed non-intrusive vs. intrusive ROM, respectively. The second method is more robust but computationally expensive compared to the simpler interpolation approach. In the present work, we are interested in the application of steady ROM in the more complicated 2-body problem.

The present approach builds on top of our previous works. An initial attempt was made at solving the 2D 2-body problem by decomposing the flow domain into three subdomains [2]; this was subsequently augmented to a five-subdomain strategy [2]. This Domain-Decomposed ROM (DDROM) combines the efficiency of the ROM with the flexibility of the FOM (i.e., standard CFD). In parallel, we refined the single-body single-domain POD-ROM approach to efficiently develop the aerodynamic database of a (3D) missile [22]. The first two efforts established the basic philosophy of the DDROM approach used here, however, it was employed on a simpler 2D problem. The latter paper, although concerning a simpler problem, presented some valuable improvements in the POD-ROM approach itself that are of relevance here. Lately, we have used the DDROM approach to solve for the transonic flow around a single 2D body too [31].

III. Theory and approach

A. POD

Let $\boldsymbol{q}(\boldsymbol{x};\boldsymbol{\mu})$ represent the vector field of relevant flow variables in a steady problem, with $\boldsymbol{x} := (x, y, z)$ in 3D Cartesian coordinates, and $\boldsymbol{\mu}$ being the vector of operating parameters (e.g., freestream Mach number M_∞, angle of attack α and side-slip angle β of the wing). In case of a 3D problem governed by Euler's equations, the vector field $\boldsymbol{q}$ may be $[\rho, \rho u, \rho v, \rho w, p]^T$, where ρ is the density, u, v & w are the x-,y-, & z-components of velocity respectively, and p is the pressure.

We assume that the flow fields available in the learning database may be subjected to an efficient linear modal decomposition such that

$$\boldsymbol{q}(\boldsymbol{x};\boldsymbol{\mu}) \approx \bar{\boldsymbol{q}}(\boldsymbol{x}) + \sum_{n=1}^{N} \eta_n(\boldsymbol{\mu})\boldsymbol{\phi}_n(\boldsymbol{x}). \tag{1}$$

Here, $\bar{\boldsymbol{q}}$ represents the base flow variable vector field, which is typically obtained by averaging across all the snapshots of the learning database. The deviation of each solution (snapshot) from the base flow is assumed to be well approximated by the linear combination of N spatial 'modes' (or basis flow fields) $\{\boldsymbol{\phi}_n(\boldsymbol{x})\}_{n=1}^N$ weighted by the coefficients $\{\eta_n(\boldsymbol{\mu})\}_{n=1}^N$. For an efficient order reduction, N is typically much smaller than the number of grid points needed to represent the flow domain for a converged CFD simulation. For convenience, we will write the vector of weight coefficients as $\boldsymbol{\eta}(\boldsymbol{\mu}) := (\eta_1(\boldsymbol{\mu}), \eta_2(\boldsymbol{\mu}), \ldots, \eta_N(\boldsymbol{\mu}))^T$.

In this work, the above basis flow fields are obtained using the very well-established approach of POD [20, 21]. For brevity, we omit the details here; further discussion of the variant of POD that is most relevant for the present work can be found in Sinha et al. [22].

B. ROM

The (approximate) prediction of the flow field for a new parameter set outside the learning database, say $\boldsymbol{\mu}_0$, is pursued using the ROM. Rather than simply interpolating the parameter space, a more robust and accurate approach is to invoke the underlying governing equations of the flow, or a simplification thereof. It is based on the reduced order modal decomposition of the flow field, i.e., the POD. Equation (1) reveals that this comes down to determining the new set of (POD) coefficients $\boldsymbol{\eta}(\boldsymbol{\mu}_0)$. The basic methodology of POD-based ROM for single-body analysis was proposed by LeGresley and Alonso [26], which was further refined over the subsequent years [22, 29, 30].

Let the vector of governing *unsteady* conservation equations be represented as

$$\frac{d}{dt}(\mathscr{C}(\boldsymbol{q})) = \mathscr{R}(\boldsymbol{q}), \quad \forall \boldsymbol{x} \in \Omega. \tag{2}$$

Here, $\mathscr{C}$ is the operator that maps the chosen set of flow variables $\boldsymbol{q}$ to the conserved flow variables, $\mathscr{R}(\boldsymbol{q})$ is a shorthand notation for the terms other than the local time derivative, and Ω represents the relevant flow domain. For example, if the mass conservation equation is coded as the first row of the vector equation, then the corresponding entry of $\mathscr{C}(\boldsymbol{q})$ is ρ and that of $\mathscr{R}(\boldsymbol{q})$ is $-\nabla \cdot (\rho \boldsymbol{u})$. These equations are supplemented by the vector of boundary conditions, formally represented as

$$\mathscr{B}(\boldsymbol{q}) = 0, \quad \forall \boldsymbol{x} \in \delta, \tag{3}$$

where δ denotes the boundary of the flow domain. Since the *steady* solution is desired for $\boldsymbol{\mu}_0$, we would ideally have $\boldsymbol{q}(\boldsymbol{x}, \boldsymbol{\mu_0})$ such that $\boldsymbol{\mathscr{R}}(\boldsymbol{q}(\boldsymbol{x}, \boldsymbol{\mu_0})) = 0$ along with $\mathscr{B}(\boldsymbol{q}(\boldsymbol{x}, \boldsymbol{\mu_0})) = 0$. Indeed, the FOM attempts to satisfy these constraints at a multitude of grid points/cells over the flow domain and boundary.

In the POD-based ROM, we substitute the approximate expansion of eqn. (1) in the above governing equations. Since the base flow field and the POD modes are known from the learning database, the residual and boundary conditions are now approximated as

$$\boldsymbol{\mathscr{R}}(\boldsymbol{q}(\boldsymbol{x}, \boldsymbol{\mu}_0)) \approx \tilde{\boldsymbol{\mathscr{R}}}(\boldsymbol{x}; \boldsymbol{\eta}(\boldsymbol{\mu}_0)), \qquad \mathscr{B}(\boldsymbol{q}(\boldsymbol{x}, \boldsymbol{\mu}_0)) \approx \tilde{\mathscr{B}}(\boldsymbol{x}; \boldsymbol{\eta}(\boldsymbol{\mu}_0)). \tag{4}$$

Moreover, due to the preceding approximation, one cannot expect the above vector fields to vanish exactly on their respective domains. Instead, we recast the given problem as the following optimization problem:

$$\min_{\boldsymbol{\eta}} J(\boldsymbol{\eta}(\boldsymbol{\mu}_0)), \quad J(\boldsymbol{\eta}(\boldsymbol{\mu}_0)) := \left(\left\|\tilde{\boldsymbol{\mathscr{R}}}(\cdot; \boldsymbol{\eta}(\boldsymbol{\mu}_0))\right\|_p\right)_\Omega \quad \text{subject to } \left(\left\|\tilde{\mathscr{B}}(\cdot; \boldsymbol{\eta}(\boldsymbol{\mu}_0))\right\|_p\right)_\delta < \epsilon. \tag{5}$$

Here, $||\cdot||_p$ denotes the $\mathscr{L}^p$ norm of a vector field, and ϵ denotes a suitable tolerance specified for approximately satisfying the boundary conditions. In the present work, we have used the $\mathscr{L}^1$ norm of the residue, and it is computed over the entire domain Ω. Note that for a reasonably small number of retained POD modes, the above represents a very small optimization problem, that constitutes a significant saving over the corresponding FOM. In this preliminary work, we do not pursue the very useful strategy of 'hyper-reduction' [32] that promises even greater efficiency.

One of the salient conclusions from the POD-ROM modelling of parametric systems is that minimizing the residual of the Euler equations is sufficient even though the data comes from RANS simulations [22, 28–30]. There are two reasons why this approximation is successful. The boundary layer close to the wall (the essentially viscous phenomena) is captured faithfully in all the flow snapshots, and thus is implicitly accounted for in the mean field and POD modes already. Moreover, the POD modes, by design, only model the large-scale coherent structures, and thus the POD-ROM is unable to deliver any significant further improvement to the reconstruction of the fine-scale viscous flow structures anyway.

C. Domain decomposition

Reconsider the quasi-steady two-body aerodynamics problem with the store's position and orientation relative to the aircraft being allowed to be arbitrary, albeit within certain bounds. Evidently, the problem geometry as well as the mesh must change as the store is traversing its separation trajectory. As mentioned before, and substantiated subsequently in section III.A, standard modal decomposition approaches like proper orthogonal decomposition (POD) cannot represent the flow field for such cases involving large changes in the geometry and mesh. We adopt the following domain decomposition strategy so as to deploy POD-based reduced-order model (ROM) for solving the flow problem over the largest possible portion of the flow domain, along with a full-order model (FOM) solution restricted to the remaining small subdomain [1, 2, 31].

As shown in fig. 2, the full domain Ω is decomposed into three (overlapping) subdomains as follows:

- *Capsule*, Ω_C – consists of the region immediately surrounding the store, which may comprise a grid that does not change as the store moves. Evidently, its extent is limited by the configuration of nearest approach between the store and aircraft.
- *Dropbox*, Ω_D – consists of the entire region enveloping the capsule where the capsule is expected to go in the course of the store-separation trajectory, while still remaining under the influence of the aircraft. Basically, this is the only subdomain of the flow where the mesh needs to change in the problem.
- *Aircraftbox*, Ω_A – consists of the rest of the flow domain that remains unchanged in the problem. It envelopes the aircraft as well as the dropbox, and stretches all the way to the far-field boundary.

We also provide for two narrow overlap regions: the one between the capsule and the dropbox is denoted Ω_{CD} ($:= \Omega_C \cap \Omega_D$), and the other between the aircraftbox and the dropbox is denoted Ω_{AD} ($:= \Omega_A \cap \Omega_D$). In fact, the 'basic' aircraftbox domain sans Ω_{AD} is denoted Ω_a ($:= \Omega_A \setminus \Omega_{AD}$). Similarly, the 'basic' capsule domain sans Ω_{CD} is denoted Ω_c ($:= \Omega_C \setminus \Omega_{CD}$). Furthermore, the 'basic' dropbox domain sans Ω_{CD} and Ω_{AD} is denoted Ω_d ($:= \Omega_D \setminus (\Omega_{AD} \cup \Omega_{CD})$). Indeed, these 'basic' subdomains are the ones depicted in fig. 2. One thus has the following five non-overlapping subdomains of Ω: $\Omega = \Omega_c \cup \Omega_{CD} \cup \Omega_d \cup \Omega_{AD} \cup \Omega_A$.

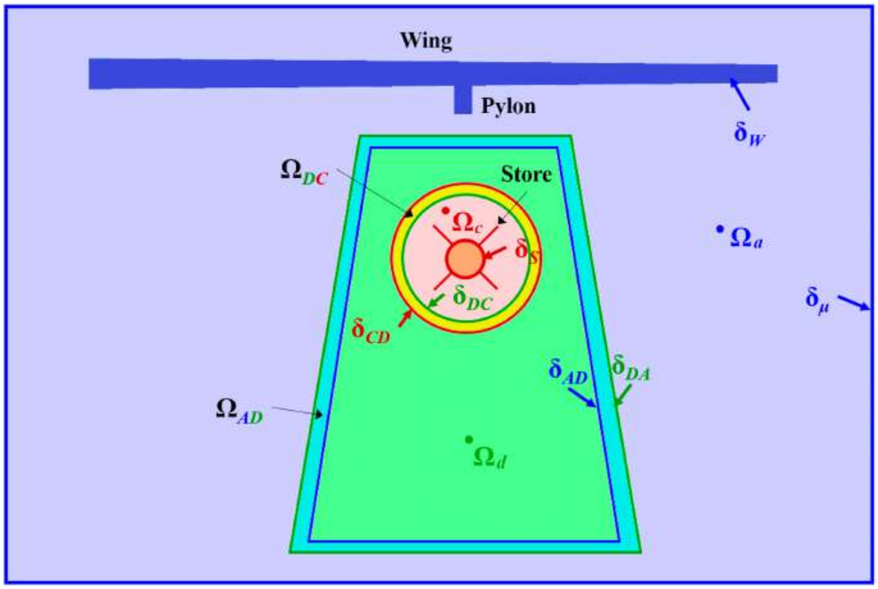


Fig. 2 Three domain-decomposed ROM approach for the two-body problem.

The capsule and aircraftbox are subdomains where the geometry (and even the mesh) can remain same across all possible configurations of the store, as well as all choices of the operating parameters (viz. freestream conditions). Thus, we will pursue POD-based ROM in these two subdomains. On the other hand, the dropbox is the only subdomain whose geometry (and hence the mesh) must change across various possible positions of the store. Hence, POD-ROM is inapplicable here, and we have to revert to FOM calculations instead. The flow solution must be matched at the two interfaces.

The iterative algorithm of DDROM is presented in the form of a self-explanatory flowchart in fig. 3.

D. Overall trajectory computation

We employ the built-in overset mesh method and 6-DOF rigid body dynamics solver of ANSYS Fluent [15] to calculate the trajectory. In the FOM, all the calculations are performed in Fluent. In particular, the procedure starts with a steady RANS calculation corresponding to the initial position of the store relative to the aircraft. The predicted forces and moments on the store are then supplied to the 6-DOF solver to compute the incremental linear and angular displacements and velocities of the store after $\Delta t = 0.005$ s. The displacements are used to move the capsule subdomain appropriately using the overset mesh method. The store velocities are used in its no-slip boundary condition. Then the transient or unsteady RANS solver is invoked to compute the change in the flow (with the changed geometry and grid) over the subsequent Δt, and the cycle repeats. The calculations proceed for about 0.3 s. In the reference wind-tunnel tests [19], the store was initially subjected to ejector forces at two locations for 0.006 s. This is not modelled in our preliminary work here.

In the DDROM approach, we replace only the flow solver, keeping all other components intact. Owing to the greater ease of code manipulation, the FOM solver for the dropbox subdomain is chosen to be SU2 [33]. This allows ready specification of ‘custom’ boundary condition on the subdomain interfaces; unlike usual boundary conditions, these are not constant throughout the surface. The mesh modification is computed in Fluent, and then passed on to SU2. The DDROM-predicted force and moment coefficients of the store are passed back to Fluent’s 6-DOF solver that calculates the new position and velocity of the store. In our preliminary DDROM implementation here, the velocity information is ignored when formulating the no-slip boundary condition of the store; it may be incorporated in the future.

IV. Results

We evaluate the proposed domain-decomposed POD-based ROM methodology on a three-dimensional problem involving a store-like body in the influence region of a wing, as depicted in fig. 2. Here we present preliminary results from this endeavour.

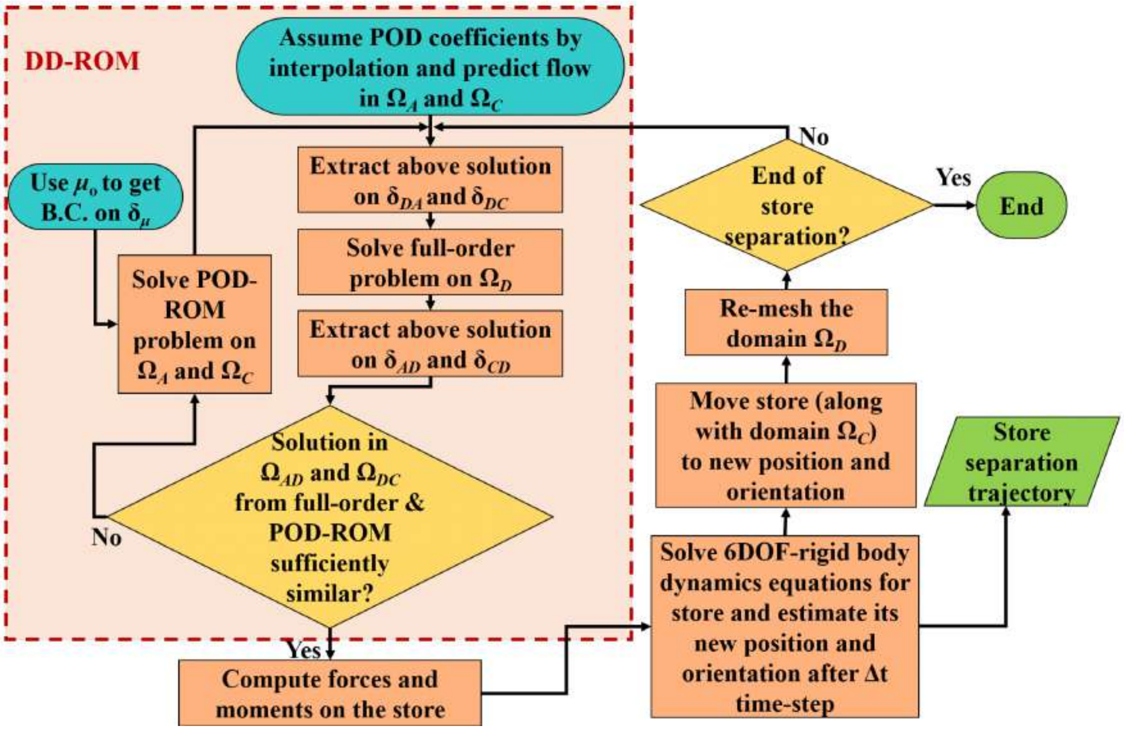


Fig. 3 Flowchart for 5-subdomain DDROM solution procedure for store separation analysis.

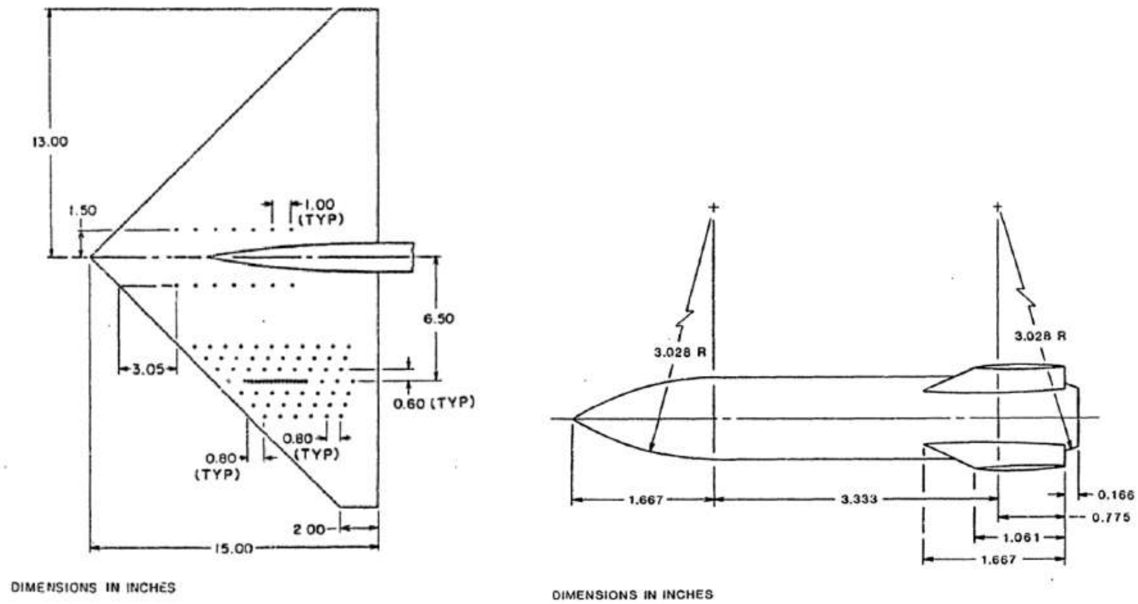
A. Geometry and mesh

The geometries of the aircraft and the store shown in fig. 4 are borrowed from a standard test case performed by [19]. The 3D model as well as coordinates systems employed for the present store separation problem are shown in the fig. 5. The wing is a clipped delta wing (NACA 64A010 airfoil section) with a detachable pylon. Its root chord is $c_0 = 15$ inches, and this will be used for normalizing all length quantities subsequently. The store is a generic finned, but otherwise axisymmetric, body with tangent-ogive forebody and truncated tangent-ogive afterbody. The diameter of the cylindrical middle body is $d = 1$ inch (i.e., $d/c_0 = 0.067$); the overall length is $\ell = 5.941$ inches. The four fins have NACA0008 section and 60° leading edge sweep. The CG of the store is on its longitudinal axis and 73 cm behind its nose. Other parameters of the geometry, including those of the pylon, may be found in Ref. [19].

The mesh is generated using the commercial software Pointwise (v18.2). As the position (X, Y, Z) and orientation (pitch-yaw-roll Euler angles (θ, ψ, ϕ)) of the store are changing throughout its travel, a different mesh needs to be generated corresponding to each specific position and orientation of the store. The mesh topology for the present 3D problem is presented in the fig. 6. Firstly, the surface mesh on the store and wing are generated ensuring that the underlying geometry is properly captured. Then, subdomain-surfaces for all the subdomains are created around these geometries with appropriate dimensions. The diameter of the capsule was chosen to be two times the diameter of the store (d), such that it encompasses the refined mesh around the store while avoiding interference with the dropbox over the trajectory. The dropbox was created with its top surface as close to the pylon as possible from the meshing point of view. Thus, the initial position of the store was decided after addressing these meshing constraints. The dropbox enclosed a region to cover approximately $10d$ of vertical drop and sufficient axial displacement of the store. The far-field of the domain is created at a distance of 10 times the root chord of the wing in all the direction. Note that all these surfaces have triangular face-cells.

Two different meshes were created: the coarser mesh was used for rapid assessment of the proposed DDROM approach here, and the finer mesh was intended for subsequent 'production' calculations. The two had 0.2 million and 3.54 million cells respectively. Maintaining the mesh topology consistent, the coarser mesh thus had individual cells that were about 2.5 larger on all sides. The finer mesh is described and depicted here; the above scale factor should be kept in mind when interpreting the subsequent results.

The total cell count of the finer grid is 3.54 million. To achieve the y^+ constraint of unity at any wall for the $k - \omega$ (SST) turbulence model, prism layers have been generated on the store with the first cell layer height of 1 micron, and



(a) **(b)**

Fig. 4 Models used in reference [19] wind-tunnel tests of the (a) wing, and (b) store.

stretching factor of 1.2. We do not aim to capture the boundary layer on the wing, because we need accurate force and moment predictions only on the store surface. Therefore, a few prism layers were generated on the wing with the sole aim of capturing the flow gradient around it. Both the overlapping regions – i.e., the capsule-dropbox Ω_{CD} and the aircraftbox-dropbox Ω_{AD} – have at least two layers of cell (preferably prism layers). To generate the grid for a given store position and orientation, the store and the capsule sub-domain was translated and rotated as a whole, making sure that the grid inside the capsule remains unchanged. Similarly, the grid for the aircraftbox sub-domain also remained unchanged across snapshots. This is a crucial aspect for POD to work in these sub-domains. It is only the grid inside the dropbox that changes across snapshots with changing store positions and orientations.

B. Learning database generation and POD

If the store is released from the aircraft in the same configuration always, then each store trajectory is completely specified by the particular freestream condition experienced by the aircraft. For 3D problems, these are the freestream Mach number M_∞, angle of attack of the wing α and the side slip angle of the wing β. The learning database comprises of the store trajectories (and their corresponding sets of quasi-steady flow solutions) for a number of $M_\infty - \alpha$ combinations (side slip angle is maintained at zero across all the trajectories). For this preliminary work we have generated the learning database for the combinations of parameters presented in fig. 7.

A point on the trajectory of the store is completely specified by its six degrees of freedom (DOF). These are the three position coordinates of the CG of the store relative to the wing's leading edge at mid-span (X, Y, Z), and three angles namely pitch, yaw and roll angle (θ, ψ, ϕ) of the store relative to the wing (see fig. 5). In our simulations, all trajectories were initiated with zero yaw, pitch and roll angles, and the CG of the store was located at $X = 21.52$ cm, $Y = -16.51$ cm and $Z = -8.13$ cm. Whereas the first two coordinates track the reference, the last one does not. This is because our domain-decomposed meshing strategy requires a finite thickness of the aircraftbox, dropbox and capsule subdomains to intervene between the wing's pylon and the store. In the original reference, the store was initiated such that its gap with the pylon was only 1.8 mm; in our case, the initial gap itself is 38.1 mm. While the domains can certainly redesigned be to allow for a smaller initial gap, our DDROM strategy will never be able to accommodate the kind of small initial gaps found in the reference. This is certainly a disadvantage of the proposed approach. A workaround may be to pursue a FOM solution approach till the store is far enough away from the pylon, and then to switch to the DDROM strategy.

The learning CFD database has been generated using compressible RANS solver with $k - \omega$ (SST) turbulence model on ANSYS Fluent, for the operating and store position parameter combination as described in fig. 7. While

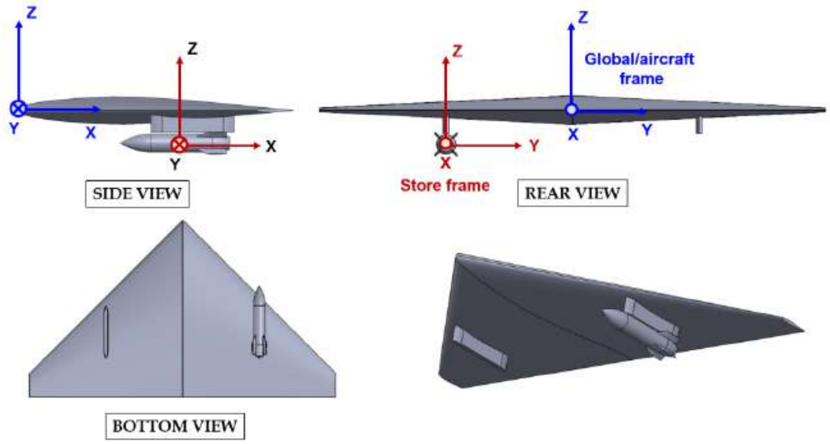
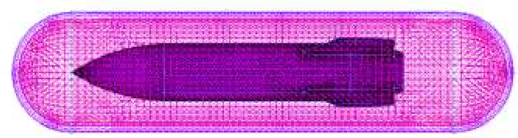
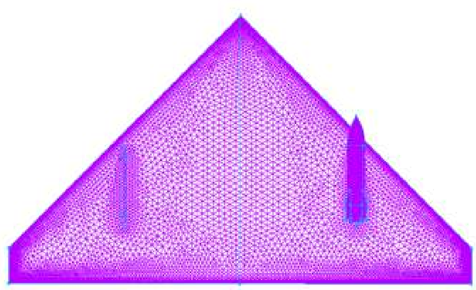


Fig. 5 Model and coordinate setup for the 3D store separation problem. The store frame is not used in this work.

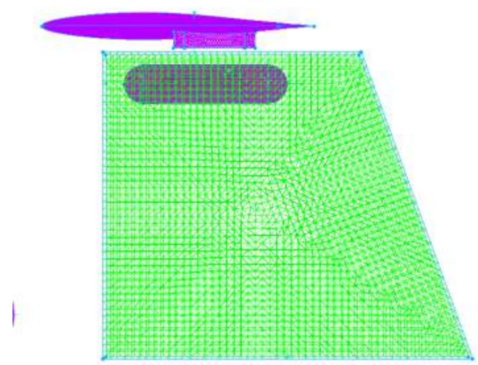
(a) Surface grid: store



(b) Domain: Capsule



(c) Surface grid: wing+store



(d) Domain: Dropbox

Fig. 6 Topology of 'finer' mesh; calculations presented here were on a coarser mesh with individual cells that were about 2.5 larger on all sides.

generating snapshots, the far-field conditions apart from velocities (defied by wing Mach number, wing angle of attack and wing side-slip angle) were pressure and temperature, which were set to constant values – 100500 Pa and 300K respectively – across all the snapshots.

For each of the 'learning' trajectories, the entire flow solution is saved at each time step (about 60 in number, with $\Delta t = 0.005$ s, as mentioned before). Subsequently, when it comes time for predicting the trajectory for a new (say, 'test') parameter case (see fig. 7), then start from the same initial condition of the store. For this aircraft-store configuration, the flow solutions from the 'learning' trajectories are used to create the DDROM model. In particular, we extract the data in two subdomains, viz., aircraftbox (Ω_A) and capsule (Ω_C), for performing the POD calculations separately for each subdomain. The conserved energy variable in the solution is converted to pressure; the remaining

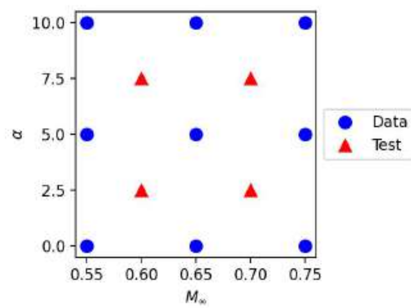


Fig. 7 Freestream Mach number M_∞ and angle of attack α combinations of the wing for the data (i.e., learning) and test (i.e., verification) trajectories.

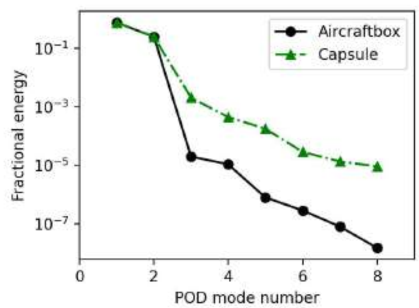


Fig. 8 Vector POD eigenspectra for the two subdomains in terms of the fractional energy content – viz. $\lambda_i / \sum_j \lambda_j$, where λ_i is i^{th} POD eigenvalue.

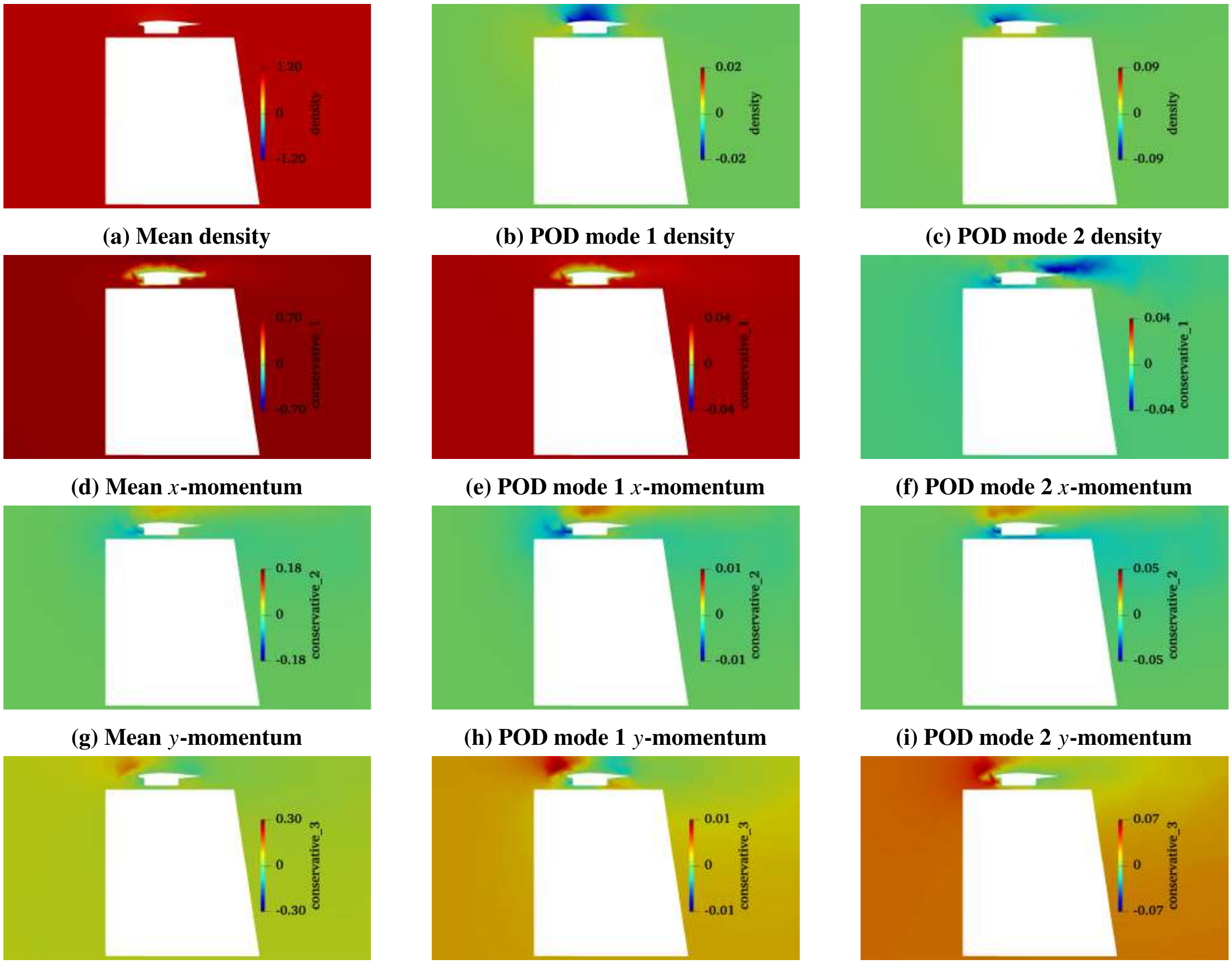
four conserved flow variables (viz. density, and the three Cartesian components of momentum) are retained as they are. Density is normalized by the freestream density ρ_∞, velocity by the freestream sound speed a_∞, and pressure by $\rho_\infty a_\infty^2$. Snapshot-based vector POD is performed on these two subdomains separately. One may remark that the turbulence variables are neglected in extracting the solution vector field; this is because, as mentioned in section III.B, we will use the Euler equation in the POD-ROM rather than the RANS equations. With these empirical inputs, the DDROM can be applied to the initial configuration of the new test trajectory, and the flow field solution can be calculated therefrom.

As described in section III.D, this flow solution is then transferred to Fluent for (a) calculation of forces and moments on the store, (b) application of the 6-DOF solver to calculate the state of the store after a time step of Δt, and (c) modification of the mesh to accommodate this new state of the store. Subsequently, we wish to apply DDROM to this new aircraft-store configuration, but the POD modes calculated for the previous step may be outdated. Thus, we propose to sample the 'learning' trajectories for the nearest Z-location of the store to the one being considered here, and to develop a new POD basis, before iterating the above-described procedure.

The results presented in this paper are for the initial step only. In the future, we intend to execute the iterative trajectory calculation strategy outlined above.

C. POD results

The fractional eigenspectra of POD for the aircraftbox and capsule subdomains are presented in fig. 8. Since there are 9 learning trajectories in our setup and the mean is subtracted, there are only 8 non-trivial POD modes for either subdomain. In both cases, we observe that the first two modes have significantly more 'energy' or salience in the data, compared to the remainder; this disparity is even more pronounced in the aircraftbox. This will be explained below with reference to the corresponding POD modes. Even amongst the remaining 6 POD modes, we observe a rapid decline of energy, with the final mode having two order of magnitude less energy than the third one. However, given the very few POD modes that we have to work with due to the scarcity of learning trajectories chosen for this work, we do not pursue any truncation, and instead work with all 8 non-trivial POD modes in both subdomains.



(a) Mean density **(b) POD mode 1 density** **(c) POD mode 2 density**

(d) Mean x-momentum **(e) POD mode 1 x-momentum** **(f) POD mode 2 x-momentum**

(g) Mean y-momentum **(h) POD mode 1 y-momentum** **(i) POD mode 2 y-momentum**

(j) Mean z-momentum **(k) POD mode 1 z-momentum** **(l) POD mode 2 z-momentum**

Fig. 9 Mean field and first two POD modes of aircraft box.

Figure 9 shows the mean values, and the first two POD modes for four flow components – ρ, ρu, ρv and ρw – corresponding to the aircraftbox subdomain. For the subsonic speeds under consideration, the pressure modes resemble the density modes closely, and hence are omitted for brevity of presentation. The mean flow fields are essentially uniform far from the aircraft and dropbox, since the far-field boundary conditions in any given snapshot is uniform also. In fact, since side-slip angle is zero across the database, the mean y-momentum vanishes in the far field. The POD modes of density and y-momentum vanish in the far-field since all snapshots in the database have the same density in the far field (unity) and same y-momentum thereat (zero). However, there is a variation in the far-field values of the x- and z-momenta due to changes in M_∞ and α in the database, and these manifest in non-trivial values of the lower-order POD modes for these to components. Since such non-zero values prevail over large regions of the domain, the 'energy' of the lower-order POD modes is high, as seen in fig. 8. Since these two major variations in the database – viz. M_∞ and α – are accounted for in the first two POD modes, the higher-order POD modes tend to have vanishing values over most of the domain, explaining the sudden drop of energy in the eigenspectrum after mode 2. It is noted that the absolute values of the POD modes are not particularly meaningful, as they are a function of the normalization chosen for the essentially scalable eigenfunctions; it is their shapes that convey the most meaning and insight as discussed above.

Figure 10 presents the mean values and first and second POD modes for the same four flow variables corresponding to the capsule subdomain. The present results are not difficult to understand with the explanation given above for the aircraftbox counterparts. The reason for the sharp fall-off of POD energy after the second mode is the same; the fall-off is less steep because the region affected by near-body flow features is now a significant fraction of the far-field region of

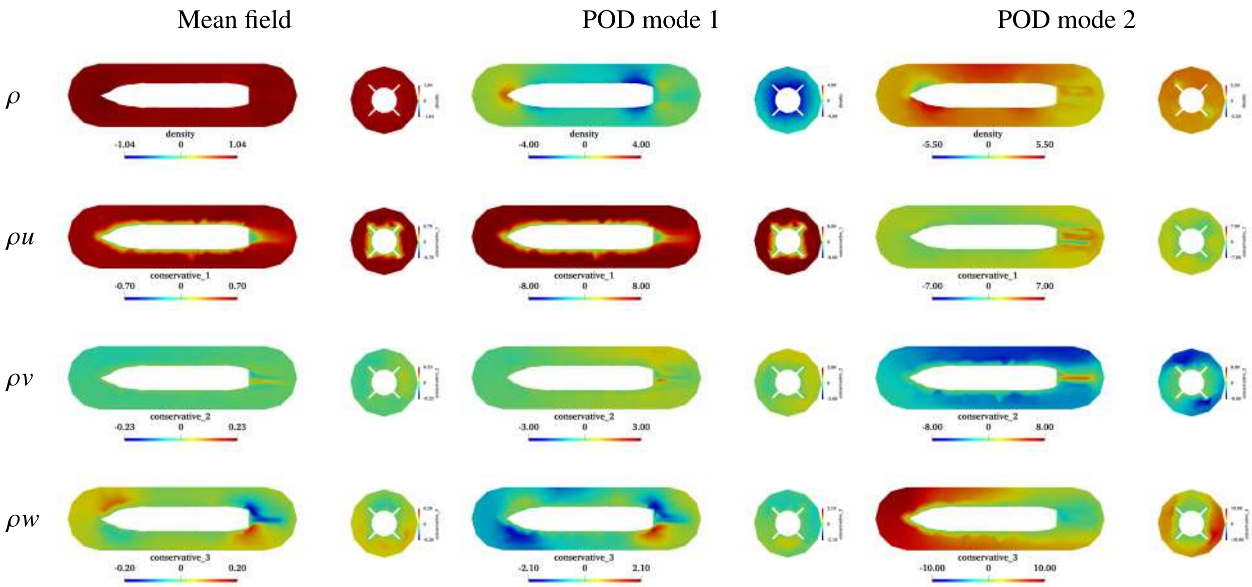


Fig. 10 Mean field and first two POD modes of capsule, following same scheme as fig. 9. Each case now shows two views: $x - z$ view and $y - z$ view.

the limited capsule subdomain.

D. Performance of DDROM

The ultimate assessment of DDROM is its capability to predict the trajectory of the store for a new parameter set. Here, we take the first step along this path by looking at the accuracy of predictions of the force and moment coefficients of the store at its initial configuration (i.e., nearest to the aircraft). The performance is evaluated at the 'test' parameter sets indicated in fig. 7, and the results are presented in table 1. All coefficients are multiplied by 100 to obtain numbers that are easier to present in the table. The 'truth' values of the coefficients come from a FOM trajectory calculation for the test case. The absolute errors of the DDROM predictions are not insignificant as a fraction of the corresponding truth values; indeed, there is 180% relative error in roll moment prediction for the case of $M_\infty = 0.6$ and $\alpha = 7.5°$. However, this is somewhat misleading as the truth value of the same happens to be very close to zero. A more representative metric is the maximum absolute error as a percentage of the range of the respective coefficient observed in the learning database, as presented in the final row. Even in this perspective, the DDROM performance may only be deemed as adequate, but it leaves much ground for improvement. The prediction of roll is always difficult for such slender bodies, and we too encounter this challenge here.

A legitimate question to ask in this context is: what if simple parametric interpolation were deemed acceptable so that the entire complication of the DDROM could be avoided. To answer this, we have used the `griddata` function of the `scipy.interpolate` library of Python. This constructs an interpolant by dividing the convex hull of the (parameter) grid into triangular tiles using Delaunay triangulation and then performing piecewise linear barycentric interpolation on each triangle. The result is presented in table 2 following the schematic of the last row of table 1. It appears that simple linear interpolation is almost twice as accurate as the hugely more complicated DDROM strategy. Although this is discouraging, we think that significant improvements may be obtained with several tactics in the future, as elucidated in section V next.

V. Conclusions

We extend our previous work on efficient albeit approximate computation of store-separation trajectories from aircraft. The adopted strategy has at heart an empirical reduced-order model (ROM) based on proper orthogonal decomposition (POD) that is applied to the majority of the flow domain. To overcome the issue of the changing grid and geometry inherent in the store-separation problem, a domain decomposition strategy has been adopted wherein the remainder of the domain is solved with the full-order model (i.e., Euler equations), and the subdomains' solutions are

Table 1 Performance of DDROM in terms of absolute prediction error of force and moment coefficients at the initial points of the 4 test trajectories. All coefficients and errors are multiplied by 100 before presenting, to highlight the actual variations in the small values. The penultimate row gives the ranges of these coefficients encountered in the learning database, and the last row gives the maximum error encountered in tests as a percentage of the respective ranges.

M_∞	α [deg]	Quantity	C_x	C_y	C_z	$C_{m,x}$	$C_{m,y}$	$C_{m,z}$
0.6	2.5	Truth	28.58	3.92	30.47	-0.28	-56.77	14.47
		Error	1.02	1.92	0.36	0.05	9.23	2.43
0.6	7.5	Truth	27.55	-21.66	45.74	0.10	-39.42	6.54
		Error	0.56	3.84	4.78	0.18	0.24	1.89
0.7	2.5	Truth	42.31	8.23	42.55	-0.52	-83.62	24.25
		Error	1.59	4.97	0.79	0.38	1.57	0.49
0.7	7.5	Truth	40.66	-28.78	62.40	0.19	-57.59	12.56
		Error	2.34	5.88	0.30	0.08	7.71	2.20
Range of truth values			30.56	85.32	65.62	1.09	102.16	37.44
Max error as % of range			7.66	6.89	7.28	35.23	9.03	6.50

Table 2 Performance of linear interpolation as in the last row of table 1.

C_x	C_y	C_z	$C_{m,x}$	$C_{m,y}$	$C_{m,z}$
3.03	4.64	4.36	30.55	5.22	3.80

iteratively matched at the interfaces. This approach may then replicate the result from a quasi-steady RANS-based approach that is considered the truth solution.

Our earlier work in this regard demonstrated the feasibility of this approach in the context of a two-dimensional toy problem. Here we have extended this to a much more realistic three-dimensional problem where we predict the force and moment coefficients on the stores for various test cases. The store trajectories are generated using a 6-DOF solver in ANSYS Fluent, which also saves the flow solution at each trajectory step. To start predicting the trajectory for a new parameter case, we extract this flow solution for the 9 nearest neighboring trajectories, and solve the consequent DDROM problem.

Results demonstrate that the DDROM approach needs to be improved substantially before it can compete with the very simple linear interpolation. We think that the following steps may achieve this.

- A very coarse grid was use for rapid evaluation of the approach, but this may also have been the primary source of the prediction inaccuracy. A refinement of the grid is essential.
- Following Ref. [22], a very simple tactic was used for calculating the force and moment coefficients of the store from the DDROM solution, as a linear combination of the counterparts in the POD database. Although this is guaranteed to be acceptable for the pressure component (since pressure is one of the POD variables), inaccuracies are expected to a lesser or greater extent in the shear stress component. This source of error can be obviated by employing Fluent to directly calculate the force and moment coefficients from the POD-reconstructed flow solution in the capsule subdomain.

Another shortcoming of the present work is that it stops after calculating the flow solution for the initial step of the trajectory only. Some challenges were encountered in accessing and utilizing the grid from the Fluent solution at the subsequent steps of the trajectory. We foresee that these will be overcome in the near future, thus delivering a viable efficient alternative to CFD-intensive store separation trajectory calculation.

References

[1] Sinha, A., and Garg, S., "Reduced-order modeling of steady aerodynamics for 2D store separation analysis," *2018 Applied Aerodynamics Conference, Paper 3168*, 2018. https://doi.org/10.2514/6.2018-3168.

[2] Pandey, N., Tembhare, R., Singh, N., and Sinha, A., "Domain-Decomposed Reduced-Order Modelling of Steady Aerodynamics for 2D Store Separation Analysis," *AIAA Scitech Forum, Paper 1004*, 2023. https://doi.org/10.2514/6.2023-1004.

[3] Cenko, A., "Store separation lessons learned during the last 30 years," *27th Congress of the International Council of the Aeronautical Sciences, Paper 2.11ST1*, 2010.

[4] Bamber, M. J., "Two methods of obtaining aircraft trajectories from wind tunnel investigations," Tech. Rep. 970 (AD 233198), David Taylor Model Basin, Washington DC, 1960.

[5] Christopher, J. P., and Carleton, W. E., "Captive-trajectory store-separation system of the AEDC-PWT 4-foot transonic tunnel," Tech. Rep. AEDC-TR-68-200 (AD839743), Arnold Engineering Development Center, 1968.

[6] Meyer, R., Cenko, A., and Yaros, S., "An influence function method for predicting store aerodynamic characteristics during weapon separation," Tech. rep., Grumman Aerospace Corp, Bethpage, NY, 1981.

[7] Cenko, A., Meyer, R., and Tessitore, F., "Further development of the influence function method for store aerodynamic analysis," *Journal of Aircraft*, Vol. 23, No. 8, 1986, pp. 656–661. https://doi.org/10.2514/3.45358.

[8] Cenko, A., Tinoco, E. N., Dyer, R. D., and DeJongh, J., "PAN AIR applications to weapons carriage and separation," *Journal of Aircraft*, Vol. 18, No. 2, 1981, pp. 128–134. https://doi.org/10.2514/3.57473.

[9] Madson, M., Moyer, S., and Cenko, A., "TranAir computations of the flow about a generic wing/pylon/finned-store configuration," *32nd AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 155*, 1994. https://doi.org/10.2514/6.1994-155.

[10] Meakin, R., and Suhs, N., "Unsteady aerodynamic simulation of multiple bodies in relative motion," *9th Computational Fluid Dynamics Conference, AIAA Paper 1996*, 1989. https://doi.org/10.2514/6.1989-1996.

[11] Sickles, W., Denny, A., and Nichols, R., "Time-accurate CFD predictions for the JDAM separation from an F-18C aircraft," *38th Aerospace Sciences Meeting and Exhibit, AIAA Paper 796*, 2000. https://doi.org/10.2514/6.2000-796.

[12] Steger, J. L., Dougherty, F. C., and Benek, J. A., "A Chimera grid scheme," *Advances in Grid Generation*, edited by K. N. Ghia and U. Ghia, ASME FED Vol. 5, 1983, pp. 59–69.

[13] Anandhanarayanan, K., "Development and validation of a grid-free viscous solver," *AIAA Journal*, Vol. 54, No. 10, 2016, pp. 3310–3313. https://doi.org/10.2514/1.j054863.

[14] Berglind, T., and Tysell, L., "Time-accurate CFD approach to numerical simulation of store separation trajectory prediction," *29th AIAA Applied Aerodynamics Conference, AIAA Paper 3958*, 2011. https://doi.org/10.2514/6.2011-3958.

[15] Khaware, A., Shivanandham, A., Gupta, V. K., Ramakrishnan, S., Azab, M., and Scheidegger, T., "Numerical simulation of store separation trajectories for EGLIN test case using overset mesh," *AIAA Aerospace Sciences Meeting, Paper 1272*, 2018. https://doi.org/10.2514/6.2018-1272.

[16] MacLucas, D., and Gledhill, I. M. A., "Time-accurate transonic CFD simulation of a generic store release case," *R&D Journal of the South African Institution of Mechanical Engineering*, Vol. 34, 2018, pp. 9–16.

[17] Abuhanieh, S., Akay, H. U., and Bicer, B., "A new strategy for solving store separation problems using OpenFOAM," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 236, No. 15, 2022, pp. 3152–3166. https://doi.org/10.1177/09544100221080771.

[18] Tian, S.-L., Xu, K., Xia, J., and Chen, L., "Efficient parallel dynamic overset unstructured grid for flow simulation with moving bodies," *AIAA Journal*, Vol. 61, No. 5, 2023, pp. 2135–2166. https://doi.org/10.2514/1.j062579.

[19] Heim, E. R., "CFD wing/pylon/finned store mutual interference wind tunnel experiment," Tech. Rep. AEDC-TSR-91-P4, Defense Technical Information Center, 1991.

[20] Lumley, J. L., "The structure of inhomogeneous turbulent flows," *Atm. Turb. and Radio Wave Prop.*, edited by A. M. Yaglom and V. I. Tatarsky, Nauka, Moscow, 1967, pp. 166–178.

[21] Holmes, P., Lumley, J. L., Berkooz, G., and Rowley, C. W., *Turbulence, coherent structures, dynamical systems and symmetry*, Cambridge University Press, 2012. https://doi.org/10.1017/cbo9780511919701.

[22] Sinha, A., Kumar, R., and Umakant, J., "Reduced-order model for efficient generation of a subsonic missile's aerodynamic database," *The Aeronautical Journal*, Vol. 126, 2022, pp. 1546–1567. https://doi.org/10.1017/aer.2022.4.

[23] Bui-Thanh, T., Damodaran, M., and Willcox, K. E., "Proper orthogonal decomposition extensions for parametric applications in compressible aerodynamics," *21st Applied Aerodynamics Conference, AIAA Paper 4213*, 2003. https://doi.org/10.2514/6.2003-4213.

[24] Peters, N. J., Wissink, A., and Ekaterinaris, J., "On the construction of a mode based reduced order model for a moving store," *Aerospace Science and Technology*, Vol. 123, 2022, p. 107484. https://doi.org/10.1016/j.ast.2022.107484.

[25] Peters, N., Wissink, A., and Ekaterinaris, J., "Machine learning-based surrogate modeling approaches for fixed-wing store separation," *Aerospace Science and Technology*, Vol. 133, 2023, p. 108150. https://doi.org/10.1016/j.ast.2023.108150.

[26] LeGresley, P. A., and Alonso, J. J., "Investigation of non-linear projection for pod based reduced order models for aerodynamics," *39th Aerospace Sciences Meeting, Paper 926*, 2001. https://doi.org/10.2514/6.2001-926.

[27] Alonso, D., Velazquez, A., and Vega, J. M., "A method to generate computationally efficient reduced order models," *Computer Methods in Applied Mechanics and Engineering*, Vol. 198, 2009, pp. 2683–2691. https://doi.org/10.1016/j.cma.2009.03.012.

[28] Alonso, D., Vega, J. M., and Velazquez, A., "Reduced-order model for viscous aerodynamic flow past an airfoil," *AIAA Journal*, Vol. 48, No. 9, 2010, pp. 1946–1958. https://doi.org/10.2514/1.j050153.

[29] Zimmermann, R., and Görtz, S., "Non-linear reduced order models for steady aerodynamics," *Procedia Computer Science*, Vol. 1, No. 1, 2010, pp. 165–174. https://doi.org/10.1016/j.procs.2010.04.019.

[30] Alonso, D., Vega, J. M., Velázquez, A., and de Pablo, V., "Reduced-order modeling of three-dimensional external aerodynamic flows," *Journal of Aerospace Engineering*, Vol. 25, No. 4, 2012, pp. 588–599. https://doi.org/10.1061/(asce)as.1943-5525.0000148.

[31] Pandey, N., and Sinha, A., "Domain-decomposed reduced order modelling of transonic airfoil flow with transported snapshot approach," *AIAA Scitech Forum, Paper 0249*, 2024. https://doi.org/10.2514/6.2024-0249.

[32] Ryckelynck, D., "A priori hyperreduction method: an adaptive approach," *Journal of Computational Physics*, Vol. 202, No. 1, 2005, pp. 346–366.

[33] Palacios, F., Colonno, M. R., Aranake, A. C., Campos, A., Copeland, S. R., Economon, T. D., Lonkar, A. K., Lukaczyk, T. W., Taylor, T. W. R., and Alonso, J. J., "Stanford University Unstructured (SU2): An open-source integrated computational environment for multi-physics simulation and design," *51st Aerospace Sciences Meeting, AIAA Paper 0287*, 2013. https://doi.org/10.2514/6.2013-287.