

POD-Galerkin Based ROM for Fluid Flow with Moving Boundaries and the
Model Adaptation in Parametric Space

by

Haotian Gao

B.S., University of Science and Technology of China, China, 2009

M.S., University of Science and Technology of China, China, 2012

M.S., New Mexico State University, 2016

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Mechanical and Nuclear Engineering
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2018

Abstract

In this study, a global Proper Orthogonal Decomposition (POD)-Galerkin based Reduced Order model (ROM) is proposed. It is extended from usual fixed-domain problems to more general fluid-solid systems with moving boundaries/interfaces. The idea of the extension is similar to the immersed boundary method in numerical simulations which uses embedded forcing terms to represent boundary motions and domain changes. This immersed boundary method allows a globally defined fixed domain including both fluid and solid, where POD-Galerkin projection can be directly applied. However, such a modified approach cannot get away with the unsteadiness of boundary terms which appear as time-dependent coefficients in the new Galerkin model. These coefficients need to be pre-computed for prescribed periodic motion, or worse, to be computed at each time step for non-prescribed (e.g. with fluid-structure interaction) or non-periodic situations. Though computational time for each unsteady coefficient is smaller than the coefficients in a typical Galerkin model, because the associated integration is only in the close neighborhood of moving boundaries. The time cost is still much higher than a typical Galerkin model with constant coefficients. This extra expense for moving-boundary treatment eventually undermines the value of using ROMs. An aggressive approach is to decompose the moving boundary/domain to orthogonal modes and derive another low-order model with fixed coefficients for boundary motion. With this domain decomposition, an approach including two coupled low-order models both with fixed coefficients is proposed. Therefore, the new global ROM with decomposed approach is more efficient. Though the model with the domain decomposition is less accurate at the boundary, it is a fair trade-off for the benefit on saving computational cost. The study further shows, however, that the most time-consuming integration in both approaches, which come from the unsteady motion, has almost negligible impact on the overall dynamics. Dropping these

time-consuming terms reduces the computation cost by at least one order while having no obvious effect on model accuracy.

Based on this global POD-Galerkin based ROM with forcing term, an improved ROM which can handle the parametric variation of body motions in a certain range is also presented. This study shows that these forcing terms not only represent the moving of the boundary, but also decouple the moving parameters from the computation of model coefficients. The decoupling of control parameters provides the convenience to adapt the model for the prediction on states under variation of control parameters. An improved ROM including a shift mode seems promising in model adaptation for typical problems in a fixed domain. However, the benefit from adding a shift mode to model diminishes when the method is applied to moving-boundary problems. Instead, a combined model, which integrates data from a different set of parameters to generate the POD modes, provides a stable and accurate ROM in a certain range of parametric space for moving-boundary problems. By introducing more data from a different set of parameters, the error of the new model can be further reduced. This shows that the combined model can be trained by introducing more and more information. With the idea of the combined model, the improved global ROM with forcing terms shows impressive capability to predict problems with different unknown moving parameters, and can be used in future parametric control and optimization problems.

POD-Galerkin Based ROM for Fluid Flow with Moving Boundaries and the
Model Adaptation in Parametric Space

by

Haotian Gao

B.S., University of Science and Technology of China, China, 2009

M.S., University of Science and Technology of China, China, 2012

M.S., New Mexico State University, 2016

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Mechanical and Nuclear Engineering
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2018

Approved by:

Major Professor
Dr. Mingjun Wei

Copyright

© Haotian Gao 2018.

Abstract

In this study, a global Proper Orthogonal Decomposition (POD)-Galerkin based Reduced Order model (ROM) is proposed. It is extended from usual fixed-domain problems to more general fluid-solid systems with moving boundaries/interfaces. The idea of the extension is similar to the immersed boundary method in numerical simulations which uses embedded forcing terms to represent boundary motions and domain changes. This immersed boundary method allows a globally defined fixed domain including both fluid and solid, where POD-Galerkin projection can be directly applied. However, such a modified approach cannot get away with the unsteadiness of boundary terms which appear as time-dependent coefficients in the new Galerkin model. These coefficients need to be pre-computed for prescribed periodic motion, or worse, to be computed at each time step for non-prescribed (e.g. with fluid-structure interaction) or non-periodic situations. Though computational time for each unsteady coefficient is smaller than the coefficients in a typical Galerkin model, because the associated integration is only in the close neighborhood of moving boundaries. The time cost is still much higher than a typical Galerkin model with constant coefficients. This extra expense for moving-boundary treatment eventually undermines the value of using ROMs. An aggressive approach is to decompose the moving boundary/domain to orthogonal modes and derive another low-order model with fixed coefficients for boundary motion. With this domain decomposition, an approach including two coupled low-order models both with fixed coefficients is proposed. Therefore, the new global ROM with decomposed approach is more efficient. Though the model with the domain decomposition is less accurate at the boundary, it is a fair trade-off for the benefit on saving computational cost. The study further shows, however, that the most time-consuming integration in both approaches, which come from the unsteady motion, has almost negligible impact on the overall dynamics. Dropping these

time-consuming terms reduces the computation cost by at least one order while having no obvious effect on model accuracy.

Based on this global POD-Galerkin based ROM with forcing term, an improved ROM which can handle the parametric variation of body motions in a certain range is also presented. This study shows that these forcing terms not only represent the moving of the boundary, but also decouple the moving parameters from the computation of model coefficients. The decoupling of control parameters provides the convenience to adapt the model for the prediction on states under variation of control parameters. An improved ROM including a shift mode seems promising in model adaptation for typical problems in a fixed domain. However, the benefit from adding a shift mode to model diminishes when the method is applied to moving-boundary problems. Instead, a combined model, which integrates data from a different set of parameters to generate the POD modes, provides a stable and accurate ROM in a certain range of parametric space for moving-boundary problems. By introducing more data from a different set of parameters, the error of the new model can be further reduced. This shows that the combined model can be trained by introducing more and more information. With the idea of the combined model, the improved global ROM with forcing terms shows impressive capability to predict problems with different unknown moving parameters, and can be used in future parametric control and optimization problems.

Table of Contents

List of Figures	ix
List of Tables	xix
Acknowledgements	xix
Dedication	xx
1 Introduction	1
1.1 Global POD-Galerkin Based Reduced Order Models for Flows with Moving Boundaries	1
1.2 Model Adaptation of Parametric Variation for Global POD-Galerkin Models with Forcing Term	3
2 Global POD-Galerkin Based Reduced-Order Models for Flows with Moving Bound- aries: Continuous Solid Domain	6
2.1 Basic Idea of Traditional POD-Galerkin Based Reduced-Order Models	6
2.1.1 Galerkin Projection	7
2.1.2 Inner Product	8
2.1.3 Hilbert Space	8
2.1.4 Proper Orthogonal Decomposition	9
2.1.5 Traditional POD-Galerkin Projection for Fluid Flows	11
2.2 Methodology of Global POD-Galerkin Based Reduced Order Model with Mov- ing Boundary: Continuous Solid Domain	12
2.2.1 Global Proper Orthogonal Decomposition	12

2.2.2	Global POD-Galerkin Based Reduced Order Model with Moving Bound- ary: Continuous Solid Domain	13
2.3	Applications and Results: Continuous Solid Domain	14
2.3.1	Two-dimensional Flow Passing an Oscillatory Cylinder	15
2.3.2	Three-dimensional Flow Passing an Oscillatory Sphere	20
2.3.3	Three-dimensional Flow around a Rigid Flapping Plate	26
2.3.4	Three-dimensional Flow around a Hovering Bird	30
2.3.5	Three-dimensional Flow around a Flexible Flapping Plate	33
3	Global POD-Galerkin Based Reduced-Order Models for Flows with Moving Bound- aries: Decomposed Solid Domain	41
3.1	Methodology of Global POD-Galerkin Projection for Flows with Moving Bound- ary: Decomposed Solid Domain	42
3.2	Applications and Results: Decomposed Solid Domain	45
3.2.1	Two-dimensional Flow Passing an Oscillatory Cylinder	45
3.2.2	Three-dimensional Flow Passing an Oscillatory Sphere	49
3.2.3	Three-dimensional Flow around Flapping Plate	51
4	The Analysis of Computational Time Cost for Different Terms and Different Models	58
4.1	Term Analysis	58
4.2	Term Dropping for the Flow Passing an Oscillatory Sphere	61
5	Model Adaptation of Parametric Variation for Global POD-Galerkin Models with Forcing Term	65
5.1	Parametric Variation by Global ROM with Forcing Term	65
5.1.1	Methodology of Parametric Variation by Global ROM with Forcing Term	66
5.1.2	Applications and Results for Three-dimensional Oscillatory Sphere with Variation of Oscillatory Amplitude	68

5.2	Shift Mode	70
5.2.1	The Application of Shift Mode: Flow Passing a Fixed Cylinder	72
5.2.2	The Application of Shift Mode: Flow Passing an Oscillatory Sphere .	72
5.3	Combined Model	76
5.3.1	The Application of Combined Model on the Global ROM with Forcing Term: Two Sets of Snapshots	77
5.3.2	The Application of the Combined Model on the Global ROM with Forcing Term: Three Sets of Snapshots	86
6	Conclusion	94
	Bibliography	97
A	Decoupling of the Two-dimensional Pitching and Plunging Plate	103

List of Figures

2.1	A schematic of flow passing an oscillatory cylinder.	15
2.2	A typical snapshot for the two-dimensional flow passing an oscillatory cylinder from direct numerical simulation: vorticity is marked by contours.	16
2.3	The energy distribution and most energetic modes (shown by vorticity contours) from POD analysis of flow passing an oscillatory cylinder: (a) the energy spectrum of POD modes; (b) mean flow; (c) mode 1; (d) mode 2; (e) mode 3; (f) mode 4.	17
2.4	The phase portraits of two-dimensional flow passing an oscillatory cylinder: (a) (a_1, a_2) and (b) (a_1, a_3) , where the results computed from global ROM with forcing term and traditional ROM without forcing terms are benchmarked by the result from direct numerical simulation.	18
2.5	The comparison of different ROMs for flow passing an oscillatory cylinder: (a) the amplitude of oscillation $A := \sqrt{a_1^2 + a_2^2}$ and (b) the error of amplitude $E = A_r - A_d /A_d$	18
2.6	Comparison of flow field of two-dimensional flow passing an oscillatory cylinder after four periods of simulation/prediction: (a) DNS flow field; (b) rebuilt result by global POD coefficients computed by DNS; (c) rebuilt result by traditional ROM without forcing term; (d) rebuilt result by global ROM with forcing term.	19
2.7	A typical snapshot from direct numerical simulation: vorticity structure of a flow passing an oscillatory sphere by iso-contour of Q-criterion.	20

2.8	The energy distribution and most energetic modes by iso-contour of Q-criterion from POD analysis of flow passing an oscillatory sphere: (a) the energy spectrum of POD modes; (b) mean flow; (c) mode 1; (d) mode 2; (e) mode 3; (f) mode 4.	22
2.9	The phase portraits of three-dimensional flow passing an oscillatory sphere: (a) (a_1, a_2) and (b) (a_1, a_3) , where the results computed from the global ROM with forcing term and the traditional ROM without forcing terms are benchmarked by the result from direct numerical simulation.	23
2.10	The comparison of different ROMs for flow passing an oscillatory sphere: (a) the amplitude of oscillation $A := \sqrt{a_1^2 + a_2^2}$ and (b) the error of amplitude $E = A_r - A_d /A_d$	23
2.11	Comparison of flow field by iso-contour of Q-criterion of three-dimensional flow passing an oscillatory sphere after four periods of simulation/prediction: (a) DNS flow field; (b) rebuilt result by global POD coefficients computed by DNS; (c) rebuilt result by traditional ROM without forcing term; (d) rebuilt result by global ROM with forcing term.	24
2.12	Comparison of (a) lift and (b) drag of three-dimensional flow passing an oscillatory sphere after two periods of simulation/prediction computed by (—) DNS, rebuilt result by (—) global POD coefficients computed by DNS, (—) traditional ROM without forcing term, and (—) global ROM with forcing term.	25
2.13	The experimental setup of the flapping plate done by J. Tran et al.	26
2.14	A typical snapshot from direct numerical simulation: vorticity structure of flow around a flapping plate by iso-contour of Q-criterion.	27
2.15	The energy distribution and most energetic modes by iso-contour of Q-criterion from POD analysis of flow passing a rigid flapping plate: (a) the energy spectrum of POD modes; (b) mean flow; (c) mode 1; (d) mode 2; (e) mode 3; (f) mode 4.	28

2.16	The phase portraits of: (a) (a_1, a_2) and (b) (a_1, a_3) , where the results computed from continuous global ROM and traditional ROM without forcing terms are benchmarked by the result from direct numerical simulation. . . .	29
2.17	The comparison of different ROMs for flow around a flapping plate: (a) the amplitude of oscillation $A := \sqrt{a_1^2 + a_2^2}$ and (b) the error of amplitude $E = A_r - A_d /A_d$	29
2.18	Comparison of flow field by iso-contour of Q-criterion after four periods of simulation/prediction of (a) DNS, (b) rebuilt result by global POD coefficients computed by DNS, (c) traditional ROM, and (d) global ROM with continuous solid domain.	31
2.19	Comparison of (a) lift (F_z) and (b) drag (F_x) in one period after four periods simulation/prediction, computed by (—) DNS, rebuilt result by (—) global POD coefficients computed by DNS, (—) traditional ROM, and (—) global ROM with continuous solid domain.	32
2.20	Schematic of a hovering bird.	32
2.21	Angles change with time for the hovering bird.	33
2.22	A typical snapshot from direct numerical simulation: vorticity structure of flow around a hovering bird by iso-contour of Q-criterion.	34
2.23	The phase portraits of: (a) (a_1, a_2) and (b) (a_1, a_3) , where the results computed from continuous global ROM are benchmarked by the result from direct numerical simulation.	35
2.24	Comparison of flow field by iso-contour of Q-criterion after four periods of simulation/prediction of DNS, rebuilt result by (a) global POD coefficients computed by DNS and (b) global ROM with continuous solid domain. . . .	35
2.25	Comparison of (a) lift (F_z) and (b) drag (F_x) , computed by (—) DNS, rebuilt result by (—) global POD coefficients computed by DNS, and (—) global ROM with continuous solid domain.	36

2.26	The experimental setup of the flexible flapping plate done by J. Tran et al.	36
2.27	A typical snapshot from direct numerical simulation: vorticity structure of flow around a flexible flapping plate by iso-contour of Q-criterion.	37
2.28	The energy distribution and most energetic modes by iso-contour of Q-criterion from POD analysis of flow passing a flexible flapping plate: (a) the energy spectrum of POD modes; (b) mean flow; (c) mode 1; (d) mode 2; (e) mode 3; (f) mode 4.	38
2.29	The phase portraits of: (a) (a_1, a_2) and (b) (a_1, a_3) , where the results computed from continuous global ROM and traditional ROM without forcing terms are benchmarked by the result from direct numerical simulation. . . .	39
2.30	The comparison of different ROMs for flow around a flexible flapping plate: (a) the amplitude of oscillation $A := \sqrt{a_1^2 + a_2^2}$ and (b) the error of amplitude $E = A_r - A_d /A_d$	39
2.31	Comparison of flow field around a flexible flapping plate by iso-contour of Q-criterion after four periods of simulation/prediction of (a) DNS, (b) rebuilt result by global POD coefficients computed by DNS, (c) traditional ROM, and (d) global ROM with continuous solid domain.	40
3.1	Comparison of the exact shape and location of the solid (black circle) and the shape rebuilt from eight solid modes (color contours) at three time moments in one period: (a) $t = 0$, (b) $t = 1/4T$, (c) $t = 3/4T$	46
3.2	Phase portraits of the coefficients for solid domain modes: (a) (b_1, b_2) and (b) (b_1, b_3)	46
3.3	Comparison of phase portraits of global POD coefficients computed by DNS, solid domain decomposition (i.e. projected domain), and continuous solid domain (i.e. original domain): (a) (a_1, a_2) and (b) (a_1, a_3)	47

3.4	Comparison of flow field by iso-contour of Q-criterion after four periods of simulation/prediction of DNS, rebuilt result by global POD coefficients computed by DNS, traditional ROM, and global ROM with continuous solid domain: (a) DNS flow field; (b) rebuilt result by global POD coefficients computed by DNS; (c) rebuilt result by global ROM with continuous solid domain; (d) rebuilt result by global ROM with decomposed solid domain.	47
3.5	The comparison of different ROMs for flow passing an oscillatory cylinder: (a) the amplitude of oscillation $A := \sqrt{a_1^2 + a_2^2}$ and (b) the error of amplitude $E = A_r - A_d /A_d$	48
3.6	Comparison of (a) lift and (b) drag computed by (—) DNS, rebuilt result by (—) global POD coefficients computed by DNS, (—) solid domain decomposition, and (—) continuous solid domain.	48
3.7	Comparison of the exact shape and location of the solid (black circle) and the shape rebuilt from eight solid modes (iso-surface in green) at three time moments in one period: (a) $t = 0$, (b) $t = 1/4T$, (c) $t = 3/4T$	49
3.8	Phase portraits of the coefficients for solid domain modes: (a) (b_1, b_2) and (b) (b_1, b_3)	50
3.9	Comparison of phase portraits of global POD coefficients computed by DNS, solid domain decomposition (i.e. projected domain), and continuous solid domain (i.e. original domain): (a) (a_1, a_2) and (b) (a_1, a_3)	50
3.10	The comparison of different ROMs for flow passing an oscillatory sphere: (a) the amplitude of oscillation $A := \sqrt{a_1^2 + a_2^2}$ and (b) the error of amplitude $E = A_r - A_d /A_d$	51

3.11	Comparison of flow field by iso-contour of Q-criterion of DNS, rebuilt result by global POD coefficients computed by DNS, solid domain decomposition, and continuous solid domain: (a) DNS flow field; (b) rebuilt result by global POD coefficients computed by DNS; (c) rebuilt result by continuous solid domain approach; (d) rebuilt result by solid domain decomposition approach.	52
3.12	Comparison of (a) lift and (b) drag computed by (—) DNS, rebuilt result by (—) global POD coefficients computed by DNS, (—) solid domain decomposition, and (—) continuous solid domain.	53
3.13	Comparison of phase portraits of global POD coefficients computed by DNS, solid domain decomposition (i.e. projected domain), and continuous solid domain (i.e. original domain): (a) (a_1, a_2) and (b) (a_1, a_3)	54
3.14	The comparison of different ROMs for flow around a flapping plate: (a) the amplitude of oscillation $A := \sqrt{a_1^2 + a_2^2}$ and (b) the error of amplitude $E = A_r - A_d /A_d$	54
3.15	Comparison of flow field by iso-contour of Q-criterion of DNS, rebuilt result by global POD coefficients computed by DNS, solid domain decomposition, and continuous solid domain: (a) DNS flow field; (b) rebuilt result by global POD coefficients computed by DNS; (c) rebuilt result by continuous solid domain approach; (d) rebuilt result by solid domain decomposition approach.	56
3.16	Comparison of (a) lift (F_z) and (b) drag (F_x) in one period computed by (—) DNS, rebuilt result by (—) global POD coefficients computed by DNS, (—) solid domain decomposition, and (—) continuous solid domain.	57
4.1	Different contributions of each terms for (a) a_1 and (b) a_2 by continuous domain method (three-dimensional oscillatory sphere case, $dt=0.1$): (—) $F_i(a_j, a_k)$, (—) $f_i^{(1)}$, (—) $f_i^{(2)}$, (—) $f_i^{(3)}$, and (—) a_i	61

4.2	Different contributions of each terms for (a) a_1 and (b) a_2 by decomposed domain method(three-dimensional oscillatory sphere case, $dt=0.1$): (— — —) $F_i(a_j, a_k)$, (— · —) $f_i^{*(1)}$, (· · ·) $f_i^{*(2)}$, (— · ·) $f_i^{*(3)}$, and (—) a_i	62
4.3	Phase portrait of (a) a_1 and (b) a_2 (three-dimensional oscillatory sphere case, $dt=0.1$): (—) DNS projection, (■) continuous ROM with $f_i^{(2)}$, and (▲) continuous ROM without $f_i^{(2)}$	62
4.4	Phase portrait of (a) a_1 and (b) a_2 (three-dimensional oscillatory sphere case, $dt=0.1$): (—) DNS projection, (●) decomposed ROM with $f_i^{*(2)}$, and (▼) decomposed ROM without $f_i^{*(2)}$	63
4.5	Comparison of flow field by iso-contour of Q-criterion after four periods of simulation/prediction of DNS: (a) rebuilt result by global continuous ROM with $f_i^{(2)}$ terms; (b) rebuilt result by global decomposed ROM with $f_i^{*(2)}$ terms; (c) rebuilt result by global continuous ROM without $f_i^{(2)}$ terms; (d) rebuilt result by global decomposed ROM without $f_i^{*(2)}$ term.	64
5.1	The lift and drag for the three-dimensional oscillatory sphere: (a) lift and (b) drag.	69
5.2	The velocity profile along the y-direction at $x = 2, z = 0$ when $t = nT + 3/4T$. (a) u_x and (b) u_y	70
5.3	The error changing with time for both native ROM and normal $a = 1.0$ ROM for the prediction of the flow with the oscillation amplitude $a = 0.5$	71
5.4	The vorticity field of: (a) mean flow of $\mathbf{Re} = 100$, (b) mean flow of $\mathbf{Re} = 150$, (c) the shift mode built by $\mathbf{Re} = 100$ and $\mathbf{Re} = 150$	73
5.5	Comparison of (a) lift and (b) drag for the two-dimensional flow passing a fixed cylinder, computed by (—) ROM prediction for $\mathbf{Re} = 100$ based on the native $\mathbf{Re} = 100$ database, (—) ROM prediction for $\mathbf{Re} = 150$ based on the $\mathbf{Re} = 100$ database, and (—) ROM prediction for $\mathbf{Re} = 150$ based on the native $\mathbf{Re} = 150$ database.	74

5.6	Comparison of (a) lift and (b) drag for the two-dimensional flow passing a fixed cylinder, computed by (—) ROM prediction for $\mathbf{Re} = 100$ based on the native $\mathbf{Re} = 100$ database, (—) ROM prediction for $\mathbf{Re} = 150$ based on the $\mathbf{Re} = 100$ database and shift mode, and (—) ROM prediction for $\mathbf{Re} = 150$ based on the native $\mathbf{Re} = 150$ database.	75
5.7	The error changing with time for native ROM, normal $a = 1.0$ ROM and $a = 1.0$ ROM with shift mode for the prediction of the flow with the oscillation amplitude $a = 0.5$	76
5.8	The iso-contour of Q-criterion for the first four modes from (left) the combined model, (center) $a = 1.0$ native model, and (right) $a = 0.5$ native model. . . .	78
5.9	The velocity profile along the y-direction at $x = 2, z = 0$ when $t = nT + 3/4T$. (a) u_x and (b) u_y	79
5.10	The phase portraits of: (a) (a_1, a_2) and (b) (a_1, a_3) , (c) (a_1, a_4) and (d) (a_1, a_5) , (e) (a_1, a_6) and (f) (a_1, a_7) for $a = 0.5$ where the results computed from the combined global ROM are benchmarked by the result from direct numerical simulation.	80
5.11	The error changing with time for native ROM, normal $a = 1.0$ ROM, $a = 1.0$ ROM with shift mode, and combined ROM for the prediction of the flow with the oscillation amplitude $a = 0.5$	81
5.12	The phase portraits of: (a) (a_1, a_2) and (b) (a_1, a_3) , (c) (a_1, a_4) and (d) (a_1, a_5) , (e) (a_1, a_6) and (f) (a_1, a_7) for $a = 1.0$ where the results computed from the combined global ROM are benchmarked by the result from direct numerical simulation.	82
5.13	The error changing with time for native ROM, normal $a = 0.5$ ROM, $a = 1.0$ ROM with shift mode, and combined ROM for the prediction of the flow with the oscillation amplitude $a = 1.0$	83

5.14	The phase portraits of: (a) (a_1, a_2) and (b) (a_1, a_3) , (c) (a_1, a_4) and (d) (a_1, a_5) , (e) (a_1, a_6) and (f) (a_1, a_7) for $a = 0.75$ where the results computed from the combined global ROM are benchmarked by the result from direct numerical simulation.	84
5.15	The error changing with time for native ROM, normal $a = 1.0$ ROM, $a = 1.0$ ROM with shift mode, and combined ROM for the prediction of the flow with the oscillation amplitude $a = 0.75$	85
5.16	The mean error of different ROMs to predict the flow of other amplitudes $a \in [0.5, 1]$	86
5.17	The error changing with time for native ROM, normal $a = 1.0$ ROM, the combined ROM with two sets of snapshots, and the combined ROM with three sets of snapshots for the prediction of the flow with the oscillation amplitude $a = 0.75$	87
5.18	Comparison of flow field by iso-contour of Q-criterion at $a = 0.75$: (a) DNS flow field; (b) rebuilt result by native ROM prediction; (c) rebuilt result by the combined model with two sets of snapshots; (d) rebuilt result by the combined model with three sets of snapshots.	88
5.19	The error changing with time for native ROM, normal $a = 1.0$ ROM, the combined ROM with two sets of snapshots, and the combined ROM with three sets of snapshots for the prediction of the flow with the oscillation amplitude $a = 0.875$	89
5.20	Comparison of flow field by iso-contour of Q-criterion at $a = 0.875$: (a) DNS flow field; (b) rebuilt result by native ROM prediction; (c) rebuilt result by the combined model with two sets of snapshots; (d) rebuilt result by the combined model with three sets of snapshots.	90

5.21	The error changing with time for native ROM, normal $a = 1.0$ ROM, the combined ROM with two sets of snapshots, and the combined ROM with three sets of snapshots for the prediction of the flow with the oscillation amplitude $a = 0.625$	91
5.22	Comparison of flow field by iso-contour of Q-criterion at $a = 0.625$: (a) DNS flow field; (b) rebuilt result by native ROM prediction; (c) rebuilt result by the combined model with two sets of snapshots; (d) rebuilt result by the combined model with three sets of snapshots.	92
5.23	The mean error of different ROMs to predict the flow of other amplitudes $a \in [0.5, 1]$	93

List of Tables

4.1	The nomenclature of computational time cost.	60
4.2	The cost analysis of computational time for different moving-boundary terms.	60
4.3	The time cost of four period calculations of different methods for eight modes oscillatory sphere case.	63

Acknowledgments

First and foremost, I would like to gratefully thank to my advisor, Prof. Mingjun Wei, for his continuous support through my Ph.D. studies. He revealed me the fascination of science, encouraged me throughout the tough times in the Ph.D. pursuit, mentored me in both academic and personal development. His enthusiasm and insight in research are contagious and motivational, making every discussion an enjoyable moment for me. His guidance and support are far beyond scientific research but also in my life and future career. It has been an honor for me to be his Ph.D. student and I could not have imagined having a better advisor.

I also want to express my appreciation to Prof. Fangjun Shu, Prof. Young S. Lee, Prof. Chunpei Cai, and Prof. James Chen for their encouragement and thoughtful discussions.

My sincere thanks also go to my labmates and friends in both NMSU and KSU. I will particularly mention Min Xu and Tao Yang, they helped me to start my new life and research when I first arrived U.S., and helped me a lot during my whole Ph.D study. I will also thanks to Chun Zou, Xionghui Huang, Wei Zhang, Mehdi Tabandeh, Kun Jia and Elnaz Rezaianzadeh.

I gratefully acknowledge the financial support from the Army High Performance Computing Research Center (AHPCRC) and Micro Autonomous System and Technology (MAST) CTA.

Lastly, I would like to thank my family for all their love and encouragement.

Dedication

To my parents and family,

Chapter 1

Introduction

1.1 Global POD-Galerkin Based Reduced Order Models for Flows with Moving Boundaries

In the real world, physical phenomena are usually very complex and have infinite orders. During human history, people are always trying to build mathematical models to describe and study these phenomena. For some simple problems, such as the motion of particles or a rigid body, model can be easily built in several orders. However, for complex problems, such as complex fluid dynamics, it is very difficult to build exact models. In the past century, the Finite-Element Method (FEM) and the Finite-Difference Method (FDM) have been widely used to study these kinds of complex problems. In recent decades, the Reduced-Order Model (ROM) has become popular for approximating the results of these problems. However, ROM has never been defined in any dictionary. If it is to be understood literally, the idea of ROM is just to build a model which can reduce the order of the problem. Generally speaking, a lot of methods that have been employed in scientific research can also be considered as ROMs. For example, the above FEM/FDM can reduce the order of the a problem from infinite (the real physical phenomena) to some certain numbers (the number of elements/mesh point). Since the problem becomes more and more complex, and the accuracy of the problem becomes more and more important, high-fidelity large-scale numerical simulation (based on

FEM/FDM) has been one of the most important research tools for the understanding of fluid dynamics, structural dynamics, fluid-structure interaction, and other problems in a broader scientific community. While numerical simulation is getting more capable from advances in computers and computational science, the computational cost (e.g. time, memory) of high-fidelity simulation is still too high to be practical in many situations such as real-time control and fast optimal design. Thus, the “real” ROM, which have been widely studied during recent years, provide a way for fast computation with lower but sufficient accuracy. According to Amsallem and Farhat^{1;2}, in current scientific research, “Reduced-order models (ROMs) are usually thought of as computationally inexpensive mathematical representations that offer the potential for near real-time analysis.” One way to construct ROM is called Model Order Reduction. It is a projection-based approach, which starts with a complete first-principle model (i.e. the one used by high-fidelity simulation) and reduces its cost by mapping the problem to a low-dimensional space. The difference from other phenomenological models is that Model Order Reduction provides a mathematically rigorous way to represent the same high-fidelity physics as those provided by high-fidelity Direct Numerical Simulation (DNS) or even experiments in a low-order space constructed by appropriate base modes. As a data-driven and mathematically well-defined approach, Model Order Reduction can potentially outperform other models when proper projection, base modes, and reconstruction approaches are applied.

Since its introduction to the fluid mechanics community^{3;4}, Proper Orthogonal Decomposition (POD)-Galerkin projection has quickly become a popular tool for Model Order Reduction. POD-Galerkin projection has been successfully applied to solve many fluid problems^{5–15}. However, most earlier studies are limited for fluid flows in a fixed or infinite domain. The implementation of traditional POD-Galerkin projection is not straightforward when there are moving solid boundaries or structures in fluid flow, which is the case in many modern applications, including flapping-wing micro air vehicles, robot fish, energy harvesters, and wind turbines. As it has been reviewed recently¹⁶, actuation modes^{17;18}, Eulerian-Lagrangian dynamic mesh adaptation methods^{19–22} have been proposed to allow transition between the moving/deforming near field to the stationary far field. There is

also a different route taken by Liberge and Hamdouni, in which they consider a fictitious stationary domain including both fluid and solid, and apply POD-Galerkin projection on a modified equation for both fluid and solid²³. Their approach shares some similarity with our efforts^{24–27}, and more complete work presents in this dissertation. The idea is borrowed from the immersed boundary technique used in numerical simulation^{28–30}, where the effect of the moving boundary is represented as extra bodyforce terms added to the original Navier-Stokes equation. Generally speaking, there are two types of numerical method of the immersed boundary method: the weak-coupling approach, such as Peskin’s method^{28;29}, that solves the solid and fluid separately, usually iterative methods are used to get the converged results; and the strong-coupling approach, such as the method in this study and some early efforts^{31;32}, that solves the solid and fluid together. A global POD-Galerkin projection is developed in a fixed domain with moving body-force terms for boundaries. Different from the approach of Liberge and Hamdouni²³, our method is demonstrated robust for the prediction of large and arbitrary movements of boundaries, and also allows the modeling of solid deformations, which cannot be predicted because Liberge and Hamdouni’s method is based on the fictitious domain method which is based on the rigid body assumption. There are two different descriptions of solid motion present in this paper: a continuous domain with prescribed motion^{24;25;27} and a combination of decoupled modes with time-dependent coefficients²⁶. The domain decomposition technique is similar to the one introduced by Liberge and Hamdouni in their different application for small and fluid induced solid motion²³.

1.2 Model Adaptation of Parametric Variation for Global POD-Galerkin Models with Forcing Term

The ability to use a ROM with parameters using one set of native/original values to predict results under a different set of new parametric values is critical for the whole research on model order reduction, and usually the small deviation in parametric space from their “original/native” values could lead to severely inaccurate or even unstable results^{5;33}. The

concept of model adaptation is to allow ROMs to be used for new situations with non-native parameters. Different methods have been applied to improve this adaptation. It is easy to consider using interpolation to build the extended bases^{1;8}. However, the interpolations for complex problems are very hard and require the data to be under different sets of parametric values. Noack et al.⁵ introduce shift mode which incorporates the mean-field correction to the Galerkin based ROM, in which the correction between the steady solution and the time-averaged mean flow from unsteady snapshots is employed for the flow passing a fixed cylinder case. It shows better performance to predict the variation of the Reynolds number. Sensitivity analysis, which is based on the Continuous Sensitivity Equation³⁴ has also been employed to make the model adaptive by building the extended ROM. Similarly, extracting the combined eigenmodes from flow data under a different set of parametric values is also a straightforward idea: Ma et al.⁶ use the combined model (what they call hybrid model) to solve the bifurcation problem of flow passing a three-dimensional cylinder; Galletti et al.³⁵ use the combined model to predict the flow passing a confined square cylinder; Bourguet et al.³⁶ have applied a similar method (what they call mix POD) to the transonic flow around an airfoil submitted to small deformations: they parameterize the small deformations by a simplified model with the constraint that these small deformations will not lead to strong modifications in flow topology; Camphouse et al.³⁷ also present a similar method (what they call split-POD) by applying the POD separately for baseline snapshots and actuation snapshots, and combining them together after orthogonalization. There are also some similar methods, such as two-level POD³⁸ and Bergmann’s hybrid POD ROM³⁹, of which the idea is adding the snapshots under the new parameters to the old POD modes one by one, until the result converges.

However, there is another problem when we consider the parametric variation for the moving boundary problems: how to apply the changes of motion control parameters as input into the ROM. It is not a problem for the variation of the Reynolds number which has been widely studied^{5;6;34;35;38;39}, because the Reynolds number is naturally explicit in traditional POD-Galerkin based ROM; however, the other parameters, such as boundary conditions, body deformations, and body motions, are implicit and coupled with eigenmodes in tradi-

tional POD-Galerkin based ROM. Usually, an ad hoc term, which is explicitly related to the parameters, is employed to present the control parameters. For example, Graham and Bergmann^{40–42} use an ad hoc term to predict the flow passing a rotary cylinder; Camphouse et al.³⁷ use an ad hoc term to present the changing of boundary conditions; and Bourguet et al.³⁶ use a deformation mode as an ad hoc term to describe the airfoil deformations. However, these terms all come from simplified models or approximations, so they are non-physical and can only solve very special cases respectively. For comparison, the forcing term in our global ROM shows strong capability to better represent the changes of motion control parameters. It is directly derived from the immersed boundary Navier-Stokes equation, and is able to simulate any kind of moving boundary problems.

The rest of the thesis is arranged in the following manner. The global ROM with continuous solid domain is presented in chapter 2, and the global ROM with decomposed solid domain is presented in chapter 3. Then the computational time cost for the ROM has been analyzed term by term in chapter 4. In chapter 5, the model adaptation of parametric variation for global POD-Galerkin models with forcing term is discussed. The final conclusions are in chapter 6.

Chapter 2

Global POD-Galerkin Based Reduced-Order Models for Flows with Moving Boundaries: Continuous Solid Domain

2.1 Basic Idea of Traditional POD-Galerkin Based Reduced-Order Models

In this section, the idea of Galerkin method is introduced. It is defined by the the inner product which is based on Hilbert space. Proper Orthogonal Decomposition (POD) is also introduced to build the optimal Hilbert space which can capture most of the energy for the same number of modes. By above methods, traditional POD-Galerkin method is finally introduced.

2.1.1 Galerkin Projection

The idea of Galerkin projection is to project the functions defining the original equation onto a finite-dimensional subspace of the full phase space⁴. To perform Galerkin projection, the phase space $\mathcal{X}$ must be an inner product space spanned by a suitable set of basis functions. The suitable choices for basis functions include *mathematical modes*, such as Fourier modes and Chebyshev polynomials, as well as *empirical modes*, such as POD modes. For demonstration, consider the simple dynamics described by:

$$\dot{\mathbf{u}} = f(\mathbf{u}), \quad (2.1)$$

where f is a general operator linear or nonlinear on $\mathbf{u}$. $\mathbf{u}(\mathbf{x}, t)$ can be expanded in terms of suitable orthogonal basis functions of $\mathcal{X}$ (e.g. POD modes):

$$\mathbf{u}(x, t) = \sum_{k=1}^{\infty} a_k(t) \phi_k(\mathbf{x}), \quad (2.2)$$

where the basis functions ϕ_k are often ordered by certain physical criteria: decreasing captured energy as for POD modes, increasing the wavenumber (decreasing scale) as for Fourier modes, etc. By projecting the equation onto the set of basis functions, the dynamics of time coefficients can be described as:

$$\dot{a}_j = \langle f(\mathbf{u}), \phi_j \rangle, \quad j = 1, \dots, \infty, \quad (2.3)$$

where $\langle, \rangle$ is the inner product defined on the corresponding orthogonal phase space $\mathcal{X}$. A different $\mathcal{X}$ can lead to different definitions of $\langle, \rangle$, which will be further explained in section [2.1.2](#).

Then the finite truncation in both expansion and projection can give a model equation of lower dimension as:

$$\mathbf{u}(\mathbf{x}, t) = \sum_{k=1}^N a_k(t) \phi_k(\mathbf{x}), \quad (2.4)$$

$$\dot{a}_j = \langle f(\mathbf{u}), \phi_j \rangle, \quad j = 1, \dots, N. \quad (2.5)$$

The truncation order will depend on the properties of original equations and the requirements of specific problems.

2.1.2 Inner Product

As described in the section above, the phase space $\mathcal{X}$ must be an inner product space to perform the Galerkin projection. An inner product is a generalization of the dot product, and follows the same basic rules as dot product. And in fluid dynamics, usually the inner product is defined in the following way: in the space $\mathbb{R}$, for f_1 and f_2 defined in $\Omega \in \mathbb{R}$, the definition of inner product is in the equation:

$$\langle f_1, f_2 \rangle = \int_{\Omega} g(f_1, f_2) dA. \quad (2.6)$$

Theoretically, the function g can be anything, usually, it is chosen to represent some physical means, or to make the results better. In this paper, for incompressible flow, while f_1 and f_2 are represented by the velocity field $\mathbf{u}_1$ and $\mathbf{u}_2$, this function is directly defined as the kinetic energy, so the inner product is:

$$\langle \mathbf{u}_1, \mathbf{u}_2 \rangle = \int_{\Omega} \mathbf{u}_1 \cdot \mathbf{u}_2 dA = \int_{\Omega} u_1 u_2 + v_1 v_2 + w_1 w_2 dA. \quad (2.7)$$

If this process results in a complete metric space, it is called a Hilbert space.

2.1.3 Hilbert Space

Hilbert space is a real or complex inner product space that is also a complete metric space with respect to the distance function induced by the inner product. Usually it is described as:

$$\mathbb{H} : \phi_1, \phi_2, \dots, \phi_n, \quad (2.8)$$

where n is the number of dimensions, it can be infinite for continuous analytic problems, or finite (the number of mesh points) for discretized numerical problems. And $\phi_i, i = 1, \dots, n$ are orthogonal and normalized basic functions, which means they are following the basic rule:

$$\langle \phi_i, \phi_j \rangle = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}. \quad (2.9)$$

In fluid dynamics, the fluid velocity is usually described in the real space, and it can also be described in the Hilbert space as:

$$\mathbf{u}(x, y, z, t) = u(x, y, z, t)\mathbf{i} + v(x, y, z, t)\mathbf{j} + w(x, y, z, t)\mathbf{k} \quad (2.10)$$

$$= a_1\phi_1 + a_2\phi_2 + \dots + a_n\phi_n. \quad (2.11)$$

To make the application simple and practical, it is reasonable to find a Hilbert space

$$\mathbb{H} : \phi_1, \phi_2, \dots, \phi_k, \dots, \phi_n, \quad (2.12)$$

where k is a small number. If $\phi_i, i = 1, \dots, k$ can contain most of the information, $\phi_i, i = k, \dots, n$ can be neglected. With this kind of particular low-dimensional space, the order can be reduced significantly.

2.1.4 Proper Orthogonal Decomposition

POD has been proven as the most efficient way of capturing the dominant components of an infinite-dimensional process with finite and usually only a few number of modes⁴. The goal for POD is to find an *optimal* subspace of finite dimension to present an ensemble of data $u(t) \in \mathbb{H}$, where $\mathbb{H}$ is a Hilbert space with inner product $\langle \cdot, \cdot \rangle$. The *optimality* here can be defined by the minimization of the average error $\overline{\|u - u_s\|}$ between the original data $u(t)$ and the reconstruction from the subspace $u_s(t) = \sum_j \langle u, \phi_j \rangle \phi_j$, where $\bar{\cdot}$ denotes a time average

and $\|\cdot\|$ is the induced norm on $\mathbb{H}$. In practice, people use Singular Valuable Decomposition (SVD) to solve the subspace. SVD is a factorization of a real or complex matrix. It is the generalization of the eigen-decomposition of a positive semidefinite normal matrix to any $n \times m$ matrix via an extension of the polar decomposition. By SVD, any $n \times m$ matrix can be decomposed in the following way:

$$\mathbf{M}_{n \times m} = \mathbf{U}_{n \times n} \mathbf{\Sigma}_{n \times m} \mathbf{V}_{m \times m}^T, \quad (2.13)$$

where $\mathbf{\Sigma}$ is a diagonal matrix with non-negative real numbers on diagonal, and $\mathbf{U}, \mathbf{V}$ are unitary matrixes, and if $\mathbf{M}$ is real, they are orthogonal. In this way, if $\mathbf{U}\mathbf{\Sigma}$ is written as $\mathbf{Q}$, it will give us the POD results of a flow field as:

$$\mathbf{M} = \mathbf{Q}\mathbf{V}^T, \quad (2.14)$$

$$[u]_{n \times m} = [a]_{n \times m} [\phi]_{m \times m}. \quad (2.15)$$

From the definition of SVD in equation 2.13, we can easily get:

$$\mathbf{M}^T \mathbf{M} = \mathbf{V} \mathbf{\Sigma}^T \mathbf{U}^T \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T = \mathbf{V} (\mathbf{\Sigma}^T \mathbf{\Sigma}) \mathbf{V}^T, \quad (2.16)$$

$$\mathbf{M} \mathbf{M}^T = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T \mathbf{V} \mathbf{\Sigma}^T \mathbf{U}^T = \mathbf{U} (\mathbf{\Sigma} \mathbf{\Sigma}^T) \mathbf{U}^T. \quad (2.17)$$

So the columns of $\mathbf{V}$ are eigenvectors of $\mathbf{M}^T \mathbf{M}$, the columns of $\mathbf{U}$ are eigenvectors of $\mathbf{M} \mathbf{M}^T$, and the non-zero elements of $\mathbf{\Sigma}$ are the square roots of the non-zero eigenvalues of $\mathbf{M}^T \mathbf{M}$ or $\mathbf{M} \mathbf{M}^T$. By the above approach, the SVD problem has been transferred to the eigenvalue and eigenvector problem, which is more common in numerical studies and easy to solve. In most numerical simulations, the method of snapshots⁴³ is often used when the number of snapshots m and the number of spacial grid points n satisfies: $m \ll n$. The method of snapshots solves an m -dimensional eigenvalue problem (equation 2.16) instead of the original n -dimensional eigenvalue problem (equation 2.17) and achieves the same first m modes, which can largely reduced the dimension of the problem from n to m .

2.1.5 Traditional POD-Galerkin Projection for Fluid Flows

The idea of Galerkin projection is to project the functions defining the original equation onto a finite-dimensional subspace of the full phase space⁴. To perform Galerkin projection, the phase space $\mathcal{X}$ must be an inner product space spanned by a suitable set of basis functions. And with the above discussion, in order to capture more energy with a small number of basis functions, the POD method is employed to build the phase space.

Using POD together with Galerkin projection, we have a systematic procedure to obtain ROMs from simulation or experimental data. For traditional fluid mechanics problems, the approach can be applied to the Navier-Stokes equation^{4;5}:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{\text{Re}} \nabla^2 \mathbf{u}, \quad (2.18)$$

where $\mathbf{u}$ is velocity vector, p is pressure, and Re is the Reynolds number. The velocity can be expanded about its mean value $\mathbf{u}_0$ and on POD modes ϕ_j as:

$$\mathbf{u} \approx \mathbf{u}^N = \mathbf{u}_0 + \sum_{j=1}^N a_j(t) \phi_j(\mathbf{x}), \quad (2.19)$$

where the finite truncation at $j = N$ is performed for low-dimensional modeling. Substitute (2.19) in (2.18) and project the equation to bases ϕ_i to get dynamic equations for time coefficients:

$$\dot{a}_i = \frac{1}{\text{Re}} \sum_{j=0}^N l_{ij} a_j + \sum_{j=0}^N \sum_{k=0}^N q_{ijk} a_j a_k, \quad i = 1, \dots, N, \quad (2.20)$$

where $l_{ij} = \langle \nabla^2 \phi_j, \phi_i \rangle_\Omega$ and $q_{ijk} = \langle \nabla \cdot (\phi_j \phi_k), \phi_i \rangle_\Omega$. $\phi_0 = \mathbf{u}_0$ and $a_0 = 1$ are introduced here for simple notation. In equation (2.20), orthogonality has been applied and the pressure term has been cancelled by appropriate boundary conditions (e.g. periodic condition). This traditional method has been applied to many fluid dynamic problems^{4;5}, and demonstrated to be powerful and efficient to predict fixed boundary problems.

2.2 Methodology of Global POD-Galerkin Based Reduced Order Model with Moving Boundary: Continuous Solid Domain

2.2.1 Global Proper Orthogonal Decomposition

The traditional approach of POD-Galerkin projection requires a fixed fluid domain, and the inner product is defined only in the fluid domain as:

$$\langle \mathbf{u}_1, \mathbf{u}_2 \rangle = \int_{\Omega_f} \mathbf{u}_1 \cdot \mathbf{u}_2 dA = \int_{\Omega_f} u_1 u_2 + v_1 v_2 + w_1 w_2 dA. \quad (2.21)$$

This is not the case for flows with moving solid boundaries and structures. Instead of considering a time-dependent fluid domain, in this study, the combination of fluid and solid is treated as one whole stationary domain. The idea is similar to the immersed boundary technique used in numerical simulations²⁸⁻³⁰, where the combined domain allows simple fixed meshes for discretization. And the effect of moving boundary/structure is represented as extra body-force terms added to the original Navier-Stokes equation on specific “solid” areas. Based on this combined domain, the inner product of the POD has been re-defined in the following way:

$$\langle \mathbf{u}_1, \mathbf{u}_2 \rangle = \int_{\Omega_f \cup \Omega_s} \mathbf{u}_1 \cdot \mathbf{u}_2 dA = \int_{\Omega_f \cup \Omega_s} u_1 u_2 + v_1 v_2 + w_1 w_2 dA. \quad (2.22)$$

In the current study, the weights of the energy contribution from fluid and solid areas in the definition of inner product are kept the same for simplicity, however, it is possible to consider fluid and solid with different weights as in equation (2.23). And the change may improve the global POD modes for better ROM performance.

$$\langle \mathbf{u}_1, \mathbf{u}_2 \rangle = \int_{\Omega_f \cup \Omega_s} \gamma \mathbf{u}_1 \cdot \mathbf{u}_2 dA = \int_{\Omega_f \cup \Omega_s} \gamma u_1 u_2 + \gamma v_1 v_2 + \gamma w_1 w_2 dA. \quad (2.23)$$

The weight function γ is defined as:

$$\gamma = \begin{cases} \alpha_s & \text{in } \Omega_s \\ \alpha_f & \text{otherwise} \end{cases}, \quad (2.24)$$

where α_s and α_f are weight numbers. In this paper, the same weights are used as $\alpha_s = \alpha_f = 1$, and a possible weights are $\alpha_f = 1$ and $\alpha_s = \frac{\rho_s}{\rho_f}$.

2.2.2 Global POD-Galerkin Based Reduced Order Model with Moving Boundary: Continuous Solid Domain

By considering the combination of fluid and solid as one whole stationary domain, as discussed, similar to the immersed boundary technique, the effect of moving boundary/structure is represented as extra body-force terms added to the original Navier-Stokes equation on specific “solid” areas. Here, similar to the immersed boundary method, the global Galerkin projection is based on a modified Navier-Stokes equation for the combined domain [31;32;44–46](#).

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{\text{Re}} \nabla^2 \mathbf{u} + \mathbf{f}, \quad (2.25)$$

where $\mathbf{f}$ is the extra bodyforce term added in solid domain Ω_s to define the trajectory of solid boundary/structure:

$$\mathbf{f} = \begin{cases} (\mathbf{u} \cdot \nabla \mathbf{u} - \frac{1}{\text{Re}} \nabla^2 \mathbf{u})^n + \frac{1}{\Delta t} (\mathbf{V} - \mathbf{u}^n) & \text{in } \Omega_s \\ 0 & \text{otherwise} \end{cases}, \quad (2.26)$$

with $\mathbf{V}$ being the velocity of solid which can be prescribed [32;45;46](#) or calculated by the interaction with fluid [31;44](#).

With the modified fluid-solid equation (2.25), the classical approach for fixed fluid domain

can be directly applied on the whole fluid-solid domain to get:

$$\dot{a}_i = \frac{1}{\text{Re}} \sum_{j=0}^N l_{ij} a_j + \sum_{j=0}^N \sum_{k=0}^N q_{ijk} a_j a_k + \mathcal{T}_i, \quad i = 1, \dots, N, \quad (2.27)$$

where $\mathcal{T}_i = \langle \mathbf{f}, \phi_i \rangle$ is the body forcing term in the low-dimensional space. With the definition of $\mathbf{f}$ in equation 2.26, equation 2.27 can be written as:

$$\dot{a}_i = \sum_{j=0}^N \left(\frac{1}{\text{Re}} l_{ij} - l'_{ij} \right) a_j - \sum_{j=0}^N \sum_{k=0}^N (q_{ijk} - q'_{ijk}) a_j a_k + c'_i, \quad i = 1, \dots, N, \quad (2.28)$$

which is similar to the traditional form in (2.20). However, the new dynamic equation (2.28) has some new parameters defined by inner products with support only in the solid domain:

$$l'_{ij} = \left\langle \left(\frac{1}{\text{Re}} \nabla^2 \phi_j + \frac{1}{\Delta t} \phi_j \right), \phi_i \right\rangle_{\Omega_s(t)}, \quad (2.29)$$

$$q'_{ijk} = \langle \nabla \cdot (\phi_j \phi_k), \phi_i \rangle_{\Omega_s(t)}, \quad (2.30)$$

$$c'_i = \left\langle \frac{1}{\Delta t} \mathbf{V}, \phi_i \right\rangle_{\Omega_s(t)}, \quad (2.31)$$

and old parameters with a similar definition but being extended to support the entire combined domain with both fluid and solid. The resulting model (2.28) describes globally both fluid and solid. Different from the approach by Liberge and Hamdouni²³, this approach allows large and arbitrary movement of boundaries, and the presence of solid velocity $\mathbf{V}$ allows the model to simulate the prescribed motion.

2.3 Applications and Results: Continuous Solid Domain

The proposed method is first applied to model a two-dimensional flow passing an oscillatory cylinder, then a three-dimensional flow passing an oscillatory sphere, a three-dimensional flow around a rigid flapping plate, a three-dimensional flow around a hovering bird, and

finally, a three-dimensional flow around a flexible flapping plate.

2.3.1 Two-dimensional Flow Passing an Oscillatory Cylinder



Figure 2.1: A schematic of flow passing an oscillatory cylinder.

As sketched in figure 2.1, a two-dimensional cylinder oscillates (along y) normal to the incoming flow. The Reynolds number is 100, which is defined by the incoming velocity and cylinder diameter. The computational domain is $l_x \times l_y = (0, 50) \times (0, 30)$ with uniform mesh at $N_x \times N_y = 1000 \times 600$, and the neutral position of the cylinder center is at $(x_0, y_0) = (10, 15)$. The oscillation is defined by $y(t) = y_0 + A \sin(2\pi ft)$ with amplitude $A = 1.5$ and frequency $f = 0.2$. Here, the large amplitude is picked intentionally to challenge the capability of our approach to handle large domain change. The numerical algorithm is the same as the one reported and tested in previous numerical simulations^{31;32}, where the projection method is applied to solve incompressible Navier-Stokes equations⁴⁷, the third-order Runge-Kutta method is used for time advancement, and the immersed boundary method is implemented to describe moving boundaries/structures³⁰.

With the chosen amplitude and frequency, the wake vortex structure from our simulation shows a P+S pattern (figure 2.2), which matches the prediction from the classic parametric study by Williamson⁴⁸.

Global POD modes are computed from 400 snapshots of the above simulation data which covers four whole periods of oscillation. The POD results are shown in figure 2.3: the energy efficiency of POD is clearly indicated by the energy spectrum in figure 2.3 (a); both the energy distribution and the individual POD modes (figure 2.3 c-f) show the pairing between modes 1 & 2 and modes 3 & 4, which is typically seen in POD modes of convection-dominant

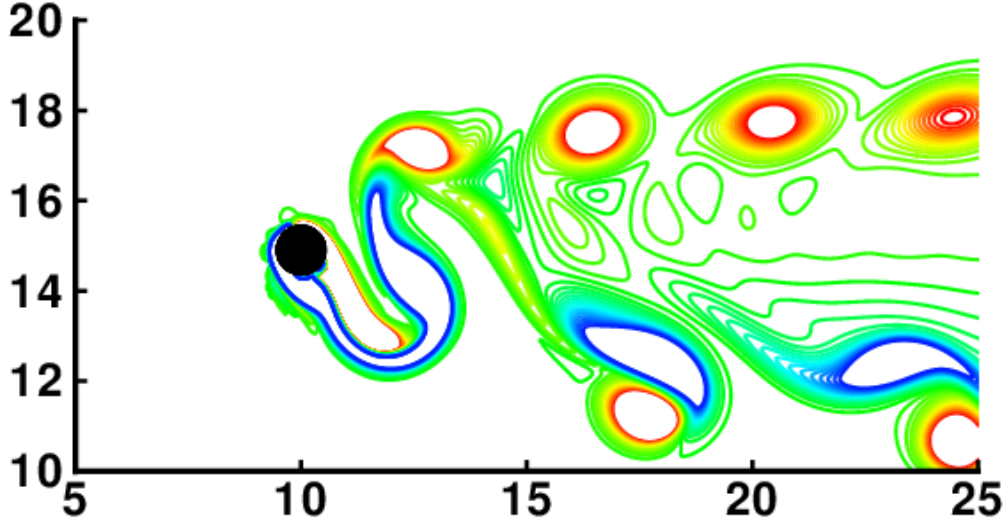


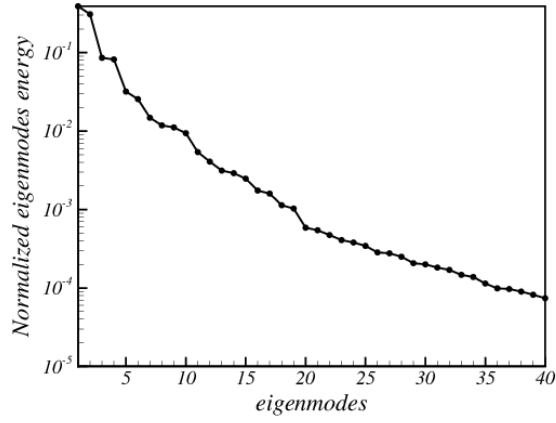
Figure 2.2: A typical snapshot for the two-dimensional flow passing an oscillatory cylinder from direct numerical simulation: vorticity is marked by contours.

flows.

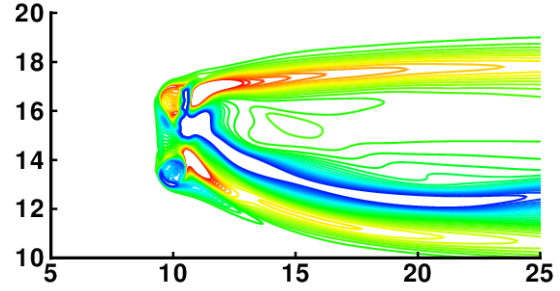
The low-order dynamic system of the most energetic modes (i.e. currently first eight modes) can then be achieved by the global projection onto these modes. In figure 2.4, the phase portraits for coefficients (a_1, a_2) and (a_1, a_3) have been plotted respectively, where, in each phase portrait, different lines and symbols indicates results from direct numerical simulation (considered as a benchmark), our global model reduction (using forcing terms to represent solid motion), and the traditional POD-Galerkin projection (applied naively without forcing terms for solid motion). The global approach shows the capability to capture basic dynamics and sustain the system energy by using the first eight modes, while the one without forcing terms cannot sustain the system energy and the modes are quickly dissipated.

Similar to Noack et al.⁵, the amplitude of oscillation $A := \sqrt{a_1^2 + a_2^2}$ has been plotted to compare the performance of different ROMs. The error defined as $E = |A_r - A_d|/A_d$ is also plotted where A_r is the amplitude calculated from ROM results, while A_d is the amplitude calculated from DNS projection.

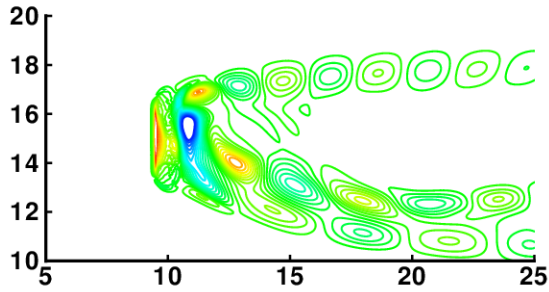
From figure 2.5, it is clear that the global ROM preserves the oscillation of the system accurately and the error is lower than 10%, while the error of traditional ROM increases to



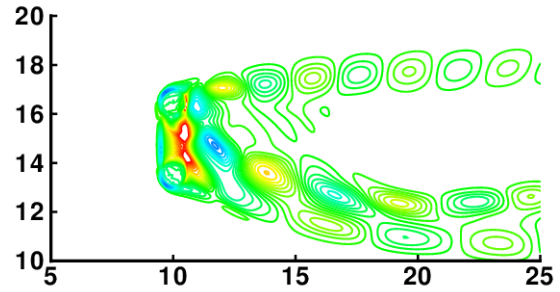
(a)



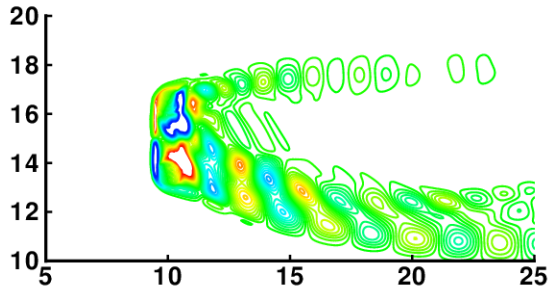
(b)



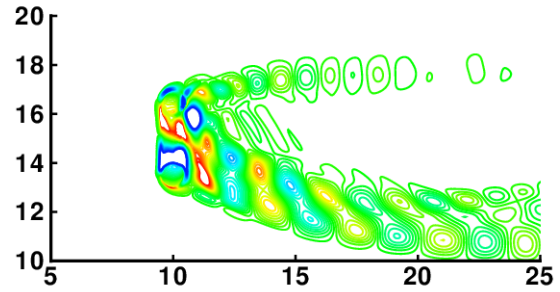
(c)



(d)



(e)



(f)

Figure 2.3: The energy distribution and most energetic modes (shown by vorticity contours) from POD analysis of flow passing an oscillatory cylinder: (a) the energy spectrum of POD modes; (b) mean flow; (c) mode 1; (d) mode 2; (e) mode 3; (f) mode 4.

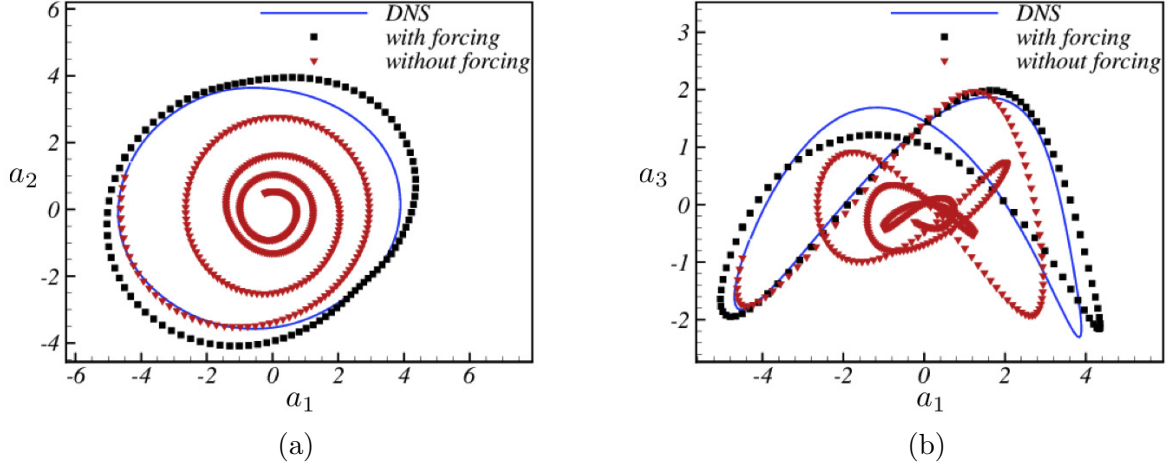


Figure 2.4: The phase portraits of two-dimensional flow passing an oscillatory cylinder: (a) (a_1, a_2) and (b) (a_1, a_3) , where the results computed from global ROM with forcing term and traditional ROM without forcing terms are benchmarked by the result from direct numerical simulation.

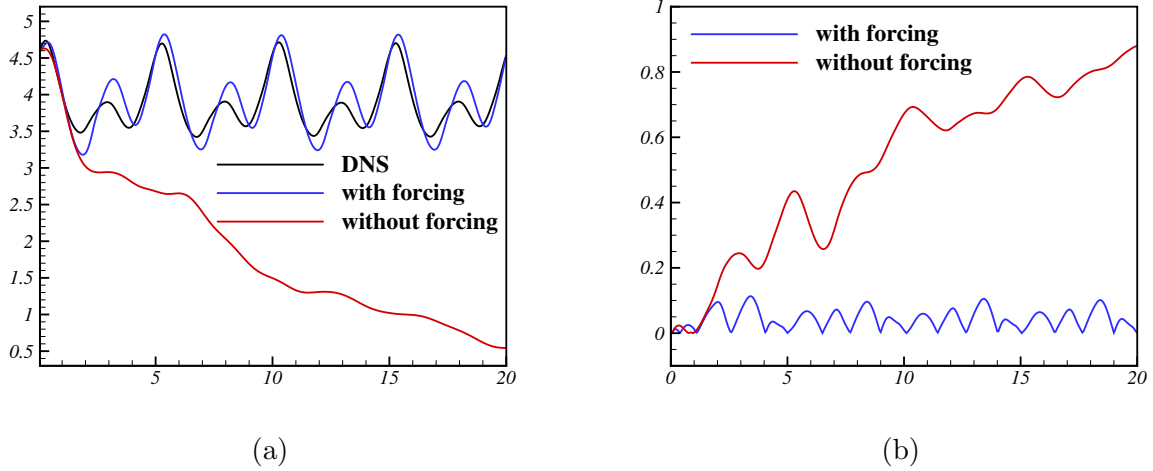


Figure 2.5: The comparison of different ROMs for flow passing an oscillatory cylinder: (a) the amplitude of oscillation $A := \sqrt{a_1^2 + a_2^2}$ and (b) the error of amplitude $E = |A_r - A_d|/A_d$.

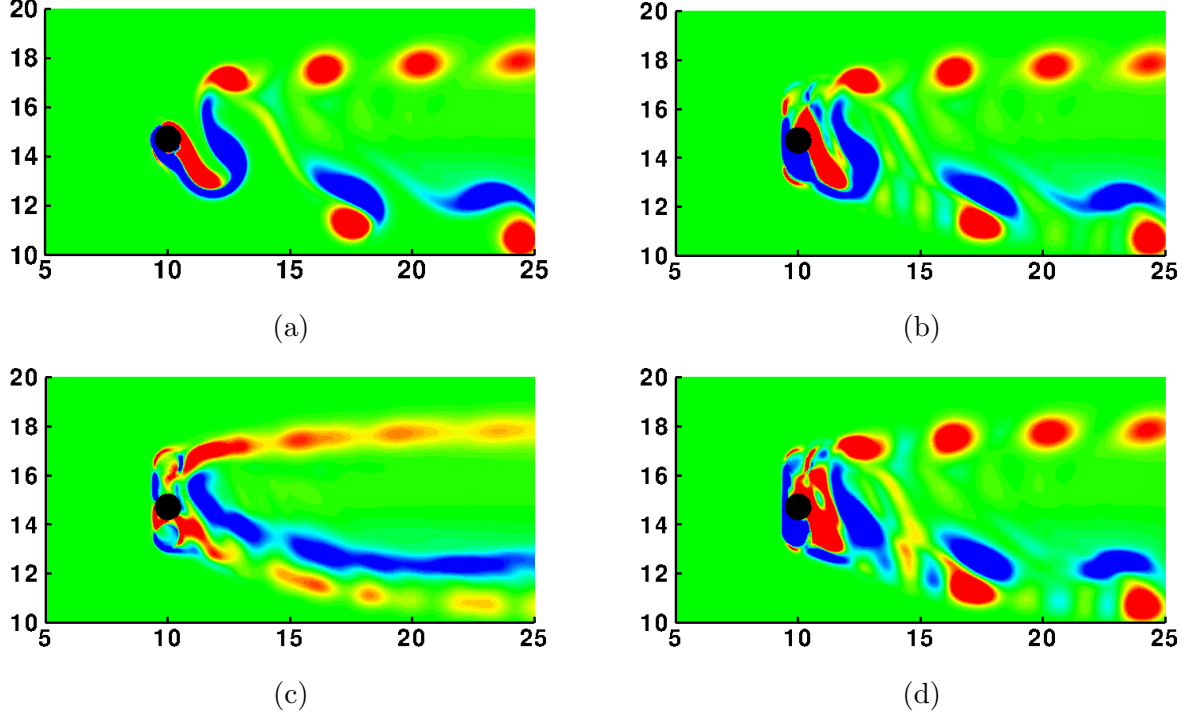


Figure 2.6: Comparison of flow field of two-dimensional flow passing an oscillatory cylinder after four periods of simulation/prediction: (a) DNS flow field; (b) rebuilt result by global POD coefficients computed by DNS; (c) rebuilt result by traditional ROM without forcing term; (d) rebuilt result by global ROM with forcing term.

almost 90%.

To show the capability of our global model more clearly, the flow structures have been plotted in figure 2.6. There are three kinds of rebuilt results plotted here: the flow rebuilt by global POD coefficients computed by DNS, the flow rebuilt by traditional ROM without forcing term, and the flow rebuilt by our global ROM with forcing term, and compared to the DNS flow field. For the traditional ROM without the forcing term, with the dissipation of time coefficients, the vortex structures have disappeared after several periods, and it degenerates to the flow similar to the mean flow. Compared to the DNS flow field, all rebuilt flows are not accurate close to the solid due to the large motion of the cylinder. However, compared to the traditional ROM, our global ROM with forcing term predicted the downstream vortexes accurately.

2.3.2 Three-dimensional Flow Passing an Oscillatory Sphere

Similarly, the global ROM has been applied to a three-dimensional flow passing an oscillatory sphere. The Reynolds number defined by the sphere diameter and incoming velocity is 300. The computational domain for the ROM is $l_x \times l_y \times l_z = (-2, 8) \times (-4, 4) \times (-2, 2)$ with uniform mesh at $N_x \times N_y \times N_z = 200 \times 160 \times 80$, while the larger domain with stretched mesh is used for direct numerical simulation to get an accurate database. The neutral position of the sphere center is at $(x_0, y_0, z_0) = (0, 0, 0)$. The oscillation is defined by $y(t) = y_0 + A \sin(2\pi f t)$ with amplitude $A = 1.0$ and frequency $f = 0.16$. Figure 2.7 shows a typical snapshot from direct numerical simulation where vortex structures are shedding behind the sphere by plotting the iso-contour of Q-criterion, which defines a vortex as a connected fluid region with a positive second invariant of $\Delta \mathbf{u}$.

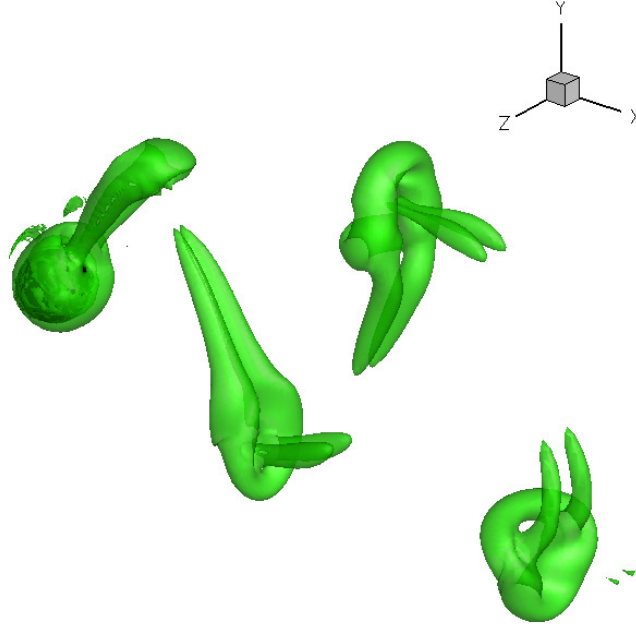


Figure 2.7: A typical snapshot from direct numerical simulation: vorticity structure of a flow passing an oscillatory sphere by iso-contour of Q-criterion.

The POD results are shown in figure 2.8 as the energy distribution, the mean flow, and the first four modes. Same as the two-dimensional case in chapter 2.3.1, the modes are in

pairs because the flow is convection-dominant.

Similarly, the results from traditional ROM without forcing term and the results from our global ROM with forcing term are compared. In figure 2.9, the phase portraits for coefficients (a_1, a_2) and (a_1, a_3) are plotted respectively. Same as in the two-dimensional case, the global ROM shows the capability to capture basic dynamics and sustain the system energy by using the first eight modes, while the one without forcing term cannot sustain the system energy and the modes are quickly dissipated.

The amplitude of oscillation $A := \sqrt{a_1^2 + a_2^2}$ and the error of it $E = |A_r - A_d|/A_d$ are also plotted in figure 2.10 to compare the performance of different ROMs. It shows that the global ROM preserves the oscillation of the system accurately, the error of it is lower than 10%, while the error of traditional ROM increases to almost 90%.

The rebuilt results by both global ROM and traditional ROM are also compared by the global POD modes and the above ROMs' coefficients to the DNS flow field (figure 2.11). For the traditional ROM without forcing term, similar to two-dimensional results, all the vortex structures in the downstream disappeared after several periods of calculation. This means the traditional ROM cannot predict the flow with a moving boundary at all: all the dynamics are lost in the traditional method. However, for our global ROM with forcing term, although the flow near the solid is not as accurate as the DNS results, the main vortexes in the downstream are preserved accurately and the dynamics of the system are preserved. To be more persuasive and practical, the lift and drag were also calculated by control volume approach (figure 2.12). Due to the truncate of the higher number of modes, it is reasonable that the model results, including the rebuilt by global POD coefficients computed by DNS, lose the high frequency information. For the traditional ROM, the amplitudes of both lift and drag decrease quickly and the results become inaccurate very fast. And for our global ROM with forcing term, the main shapes (including amplitudes, frequencies, mean values, etc.) of both lift and drag are preserved accurately, there are only small differences from the DNS results.

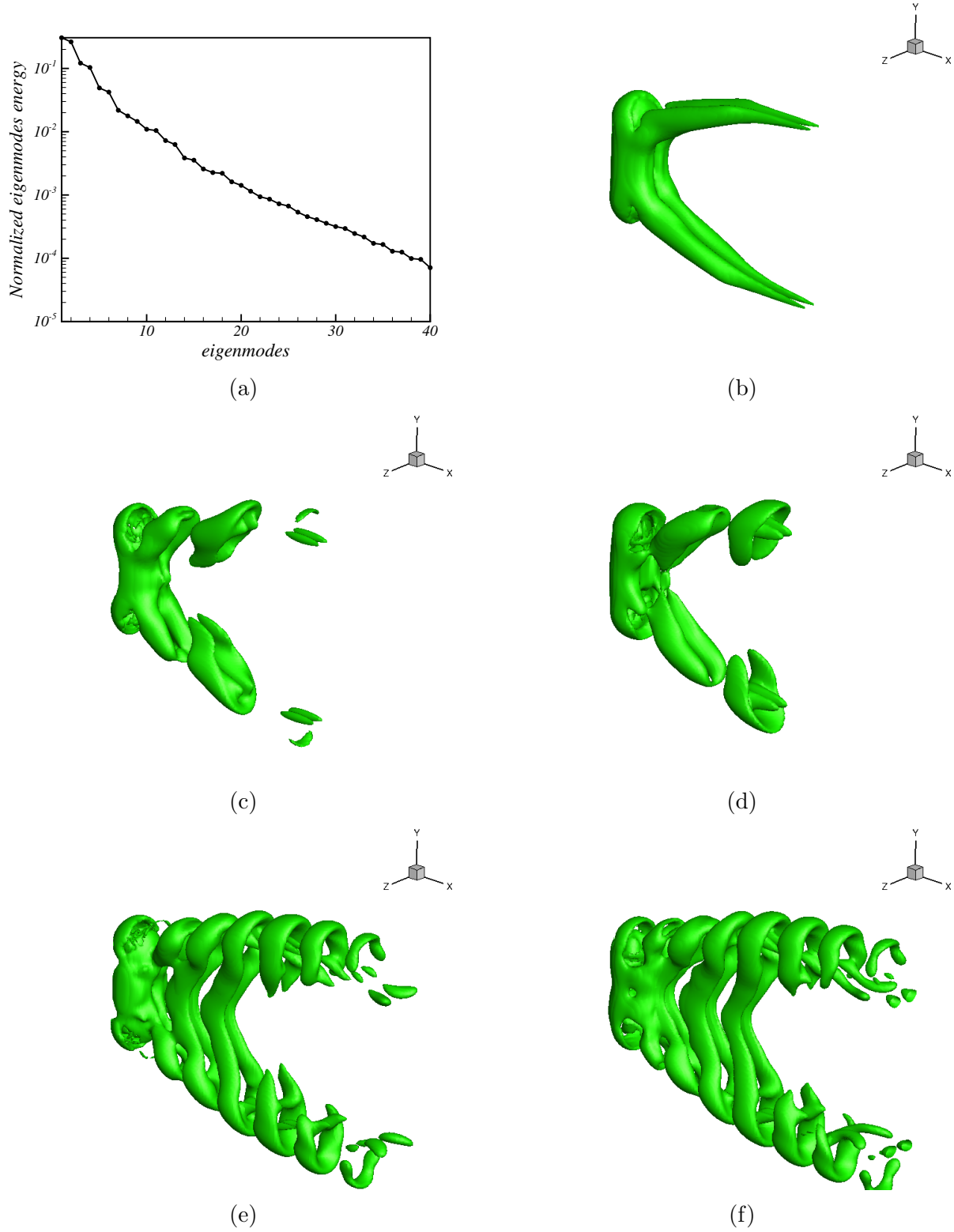


Figure 2.8: The energy distribution and most energetic modes by iso-contour of Q-criterion from POD analysis of flow passing an oscillatory sphere: (a) the energy spectrum of POD modes; (b) mean flow; (c) mode 1; (d) mode 2; (e) mode 3; (f) mode 4.

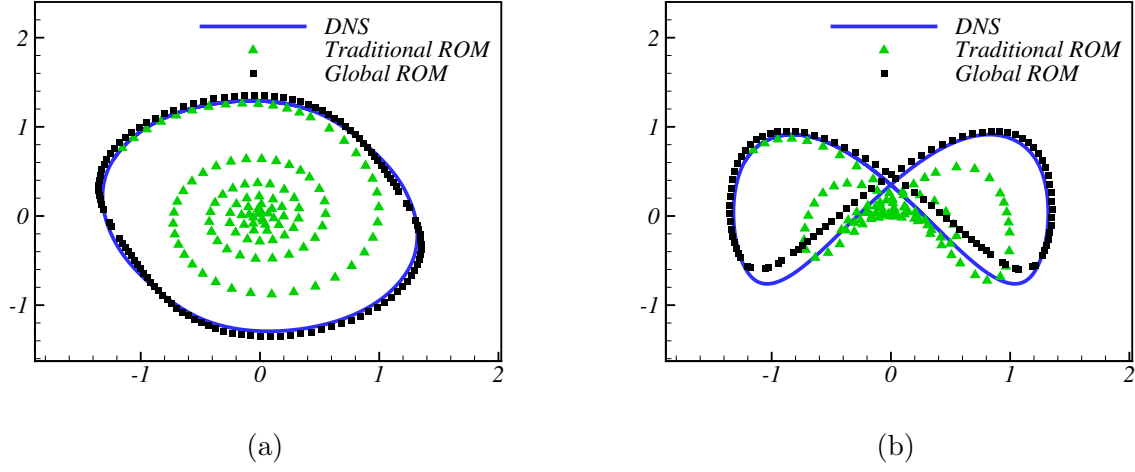


Figure 2.9: The phase portraits of three-dimensional flow passing an oscillatory sphere: (a) (a_1, a_2) and (b) (a_1, a_3) , where the results computed from the global ROM with forcing term and the traditional ROM without forcing terms are benchmarked by the result from direct numerical simulation.

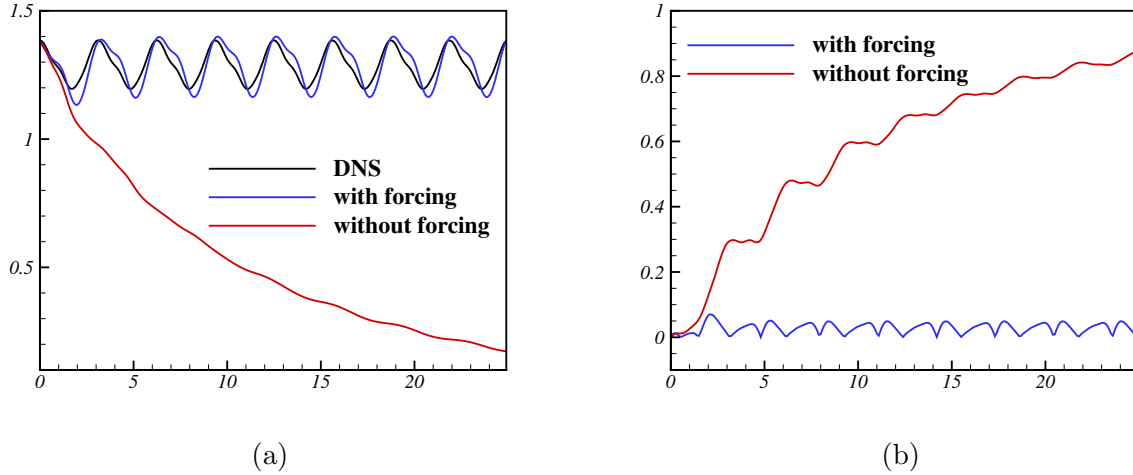


Figure 2.10: The comparison of different ROMs for flow passing an oscillatory sphere: (a) the amplitude of oscillation $A := \sqrt{a_1^2 + a_2^2}$ and (b) the error of amplitude $E = |A_r - A_d|/A_d$.

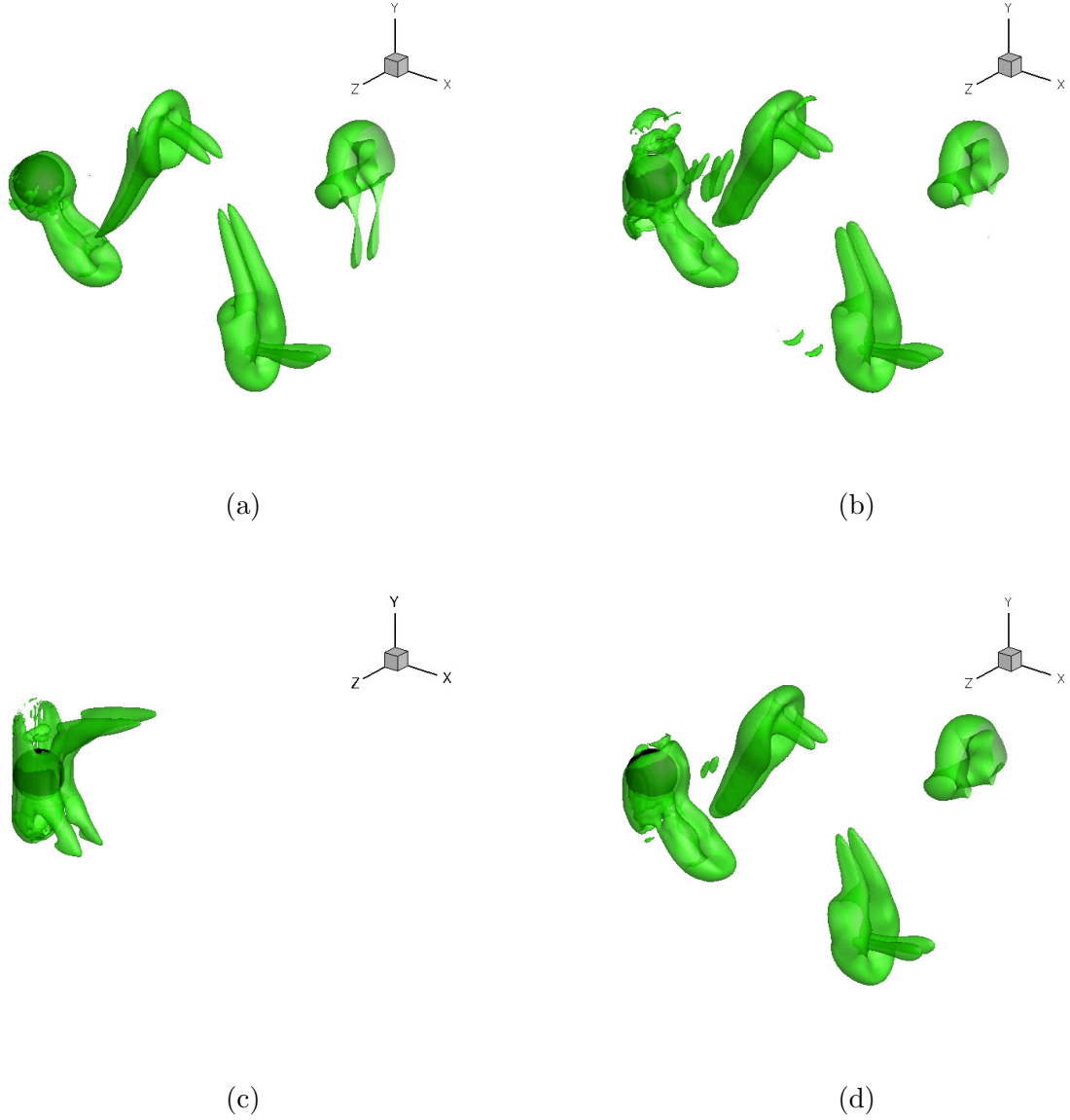
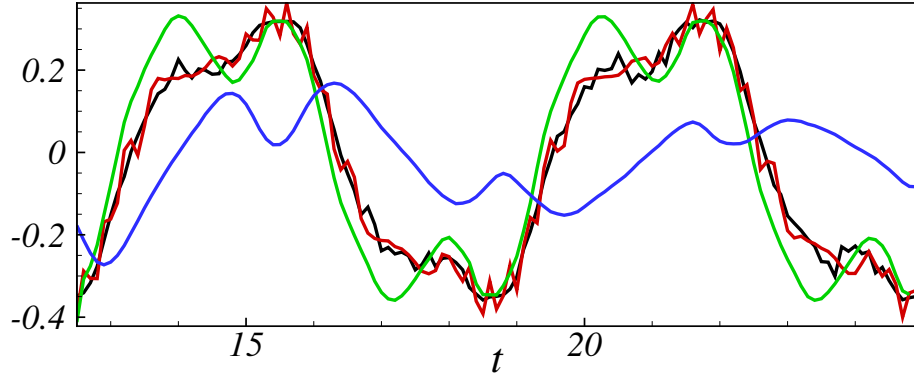
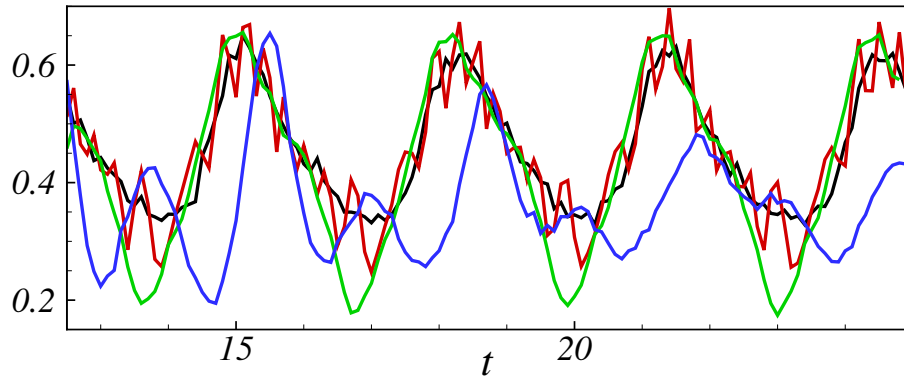


Figure 2.11: Comparison of flow field by iso-contour of Q -criterion of three-dimensional flow passing an oscillatory sphere after four periods of simulation/prediction: (a) DNS flow field; (b) rebuilt result by global POD coefficients computed by DNS; (c) rebuilt result by traditional ROM without forcing term; (d) rebuilt result by global ROM with forcing term.



(a)



(b)

Figure 2.12: Comparison of (a) lift and (b) drag of three-dimensional flow passing an oscillatory sphere after two periods of simulation/prediction computed by (—) DNS, rebuilt result by (—) global POD coefficients computed by DNS, (—) traditional ROM without forcing term, and (—) global ROM with forcing term.

2.3.3 Three-dimensional Flow around a Rigid Flapping Plate

The above two cases are both about translational motions. The new global ROM has also been applied to some rotational motion cases, such as this three-dimensional rigid flapping plate case. This is a validation case compared to the experiments done by J. Tran et al.²⁷ Figure 2.13 shows the experimental setup of the plate.

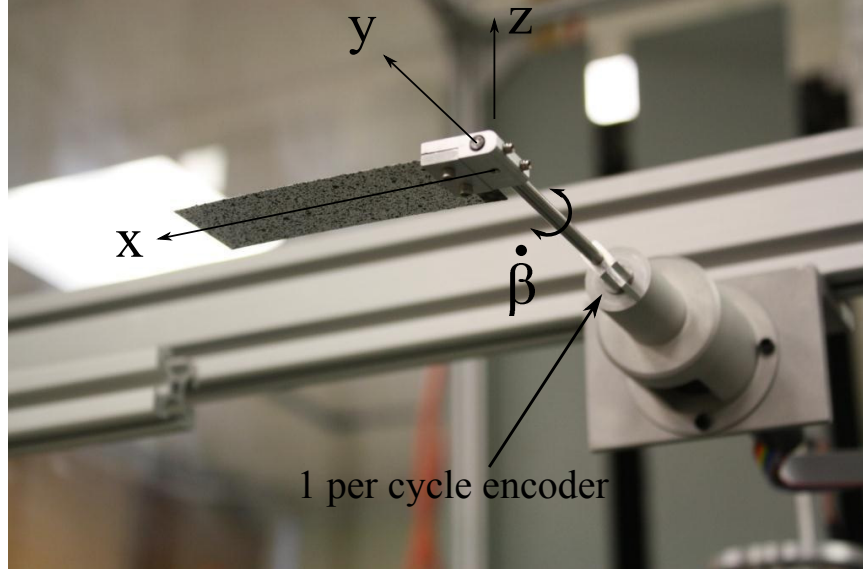


Figure 2.13: The experimental setup of the flapping plate done by J. Tran et al.

For the non-dimensional simulation, the Reynolds number defined by the plate length and maximum tail velocity is 700. The computational domain for the ROM is $l_x \times l_y \times l_z = (-2, 8) \times (-1.5, 1.5) \times (-5, 5)$ with uniform mesh at $N_x \times N_y \times N_z = 200 \times 60 \times 200$, while the larger domain with stretched mesh is used for direct numerical simulation to get an accurate database. The plate with non-dimensional size $(1, 0.3544, 0.006962)$ was placed at the $x-o-y$ plane along the x-axis. The rotation center is at $(x_0, y_0, z_0) = (0, 0, 0)$, while the leading edge of the plate is at $x = 0.076$. The plate is flapping around the y-axis defined by $\theta(t) = \theta_0 \sin(2\pi ft)$ with maximum angle $\theta_0 = \pi/5$ and frequency $f = 0.4$. Figure 2.14 shows a typical snapshot from direct numerical simulation where vortex structures are shedding around the plate.

The global POD modes, mean flow, and energy distribution are shown in figure 2.15.

By applying the same approaches on the flapping plate case, similarly, the fundamental

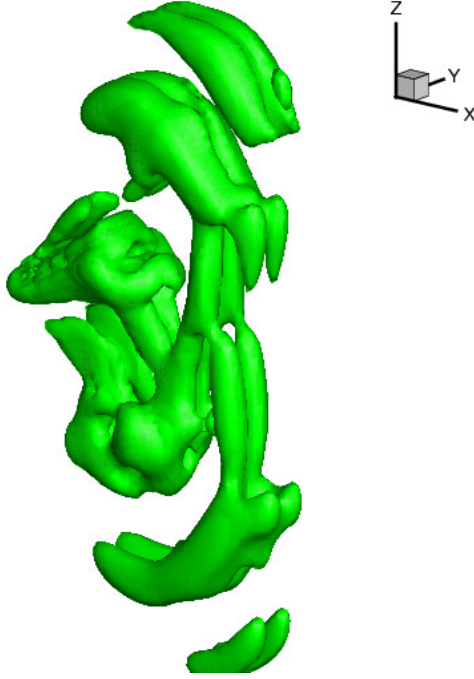


Figure 2.14: A typical snapshot from direct numerical simulation: vorticity structure of flow around a flapping plate by iso-contour of Q-criterion.

dynamics can also be described by the phase portraits of coefficients (figure 2.16). Because rotation is more complex than translation, and the plate is very thin, the global ROM results are not as accurate as the above cases, especially for the amplitude of a_2 . However, compared to the traditional ROM without the forcing term, in which the results have been quickly dissipated, the global ROM still preserves the basic dynamics and makes the prediction stable.

The amplitude of oscillation $A := \sqrt{a_1^2 + a_2^2}$ and the error of it $E = |A_r - A_d|/A_d$ are also plotted in figure 2.17 to compare the performance of different ROMs. It shows that the global ROM still preserves the oscillation of the system accurately. Although the mean error of global ROM is about 15%, it is still much less than the error of traditional ROM which is almost 100%.

Similarly, by comparing the rebuilt results (figure 2.18), our model has been demonstrated robust for the rotation problems, and can still preserve the energy and main fluid structures

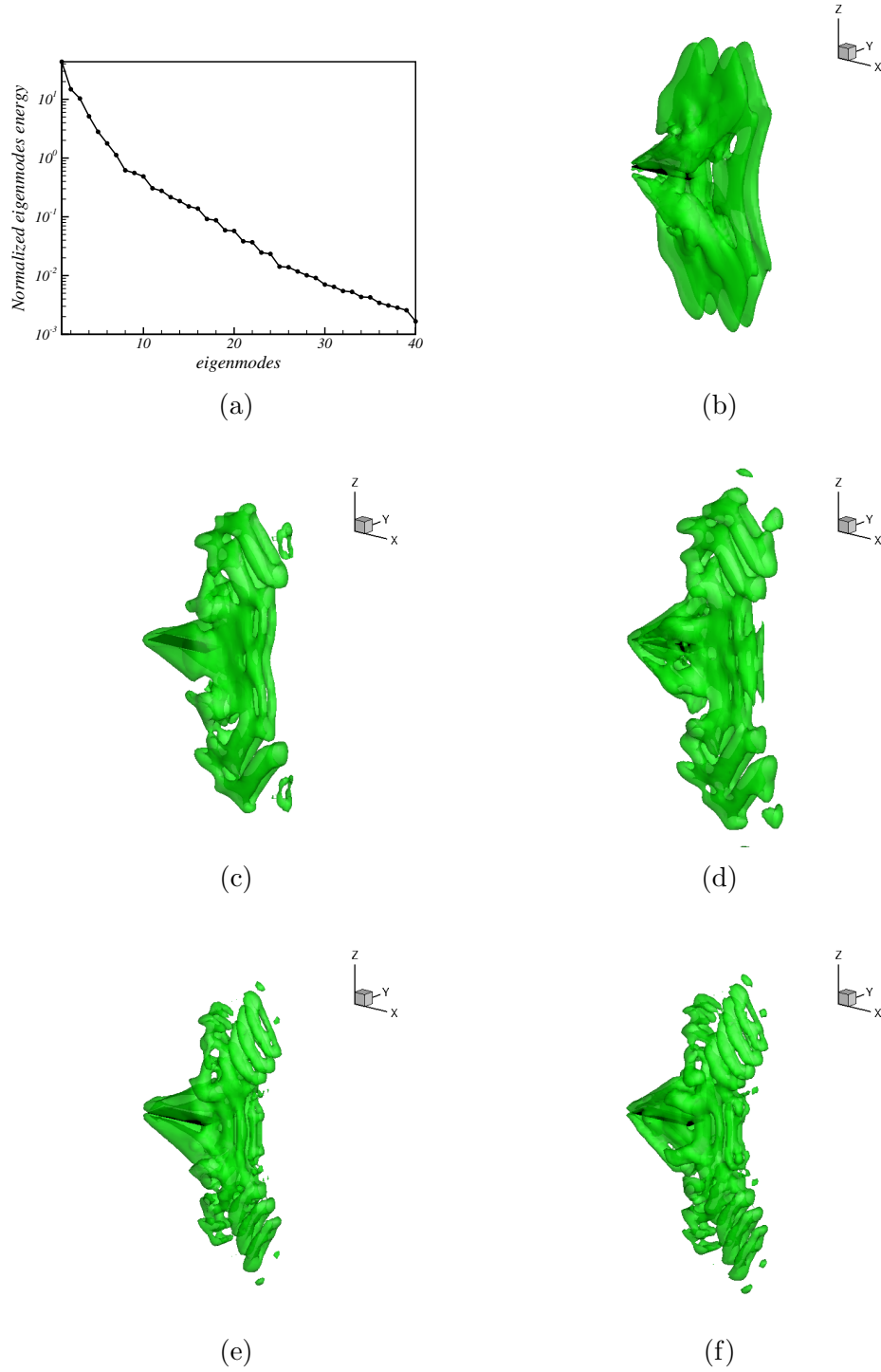


Figure 2.15: The energy distribution and most energetic modes by iso-contour of Q-criterion from POD analysis of flow passing a rigid flapping plate: (a) the energy spectrum of POD modes; (b) mean flow; (c) mode 1; (d) mode 2; (e) mode 3; (f) mode 4.

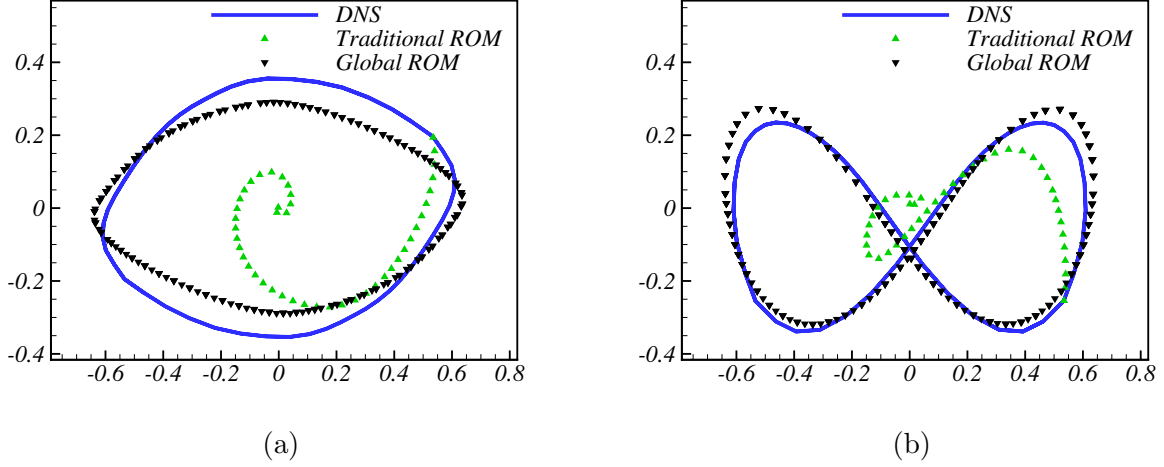


Figure 2.16: The phase portraits of: (a) (a_1, a_2) and (b) (a_1, a_3) , where the results computed from continuous global ROM and traditional ROM without forcing terms are benchmarked by the result from direct numerical simulation.

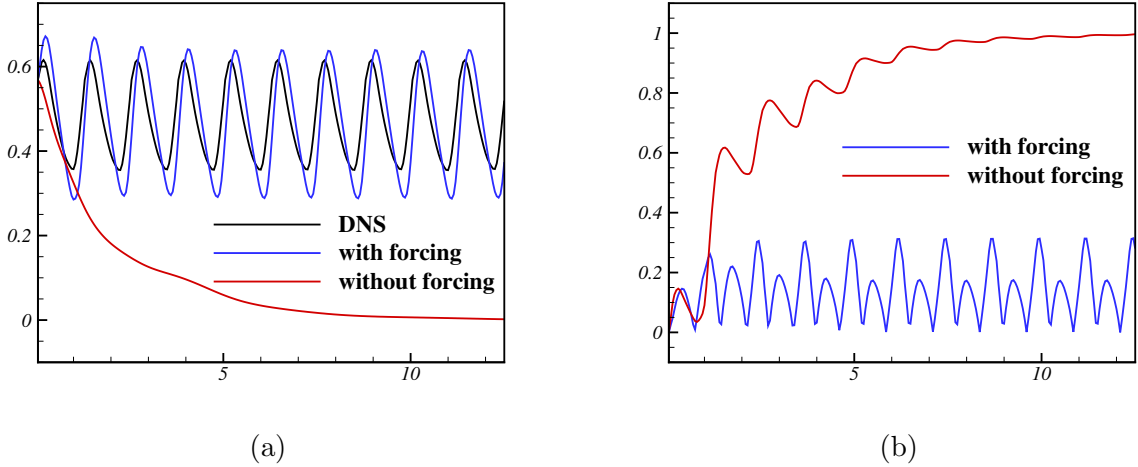


Figure 2.17: The comparison of different ROMs for flow around a flapping plate: (a) the amplitude of oscillation $A := \sqrt{a_1^2 + a_2^2}$ and (b) the error of amplitude $E = |A_r - A_d|/A_d$.

for the dynamic system.

The lift and drag of all methods can also be calculated by the control volume approach, and the results are plotted in figure 2.19. For the traditional ROM, after four periods of prediction, both lift and drag decrease to very small numbers and become almost constant. This means almost all dynamic information of the system are lost in this prediction. And for our global ROM, similar to the translational motion, the information of both lift and drag are predicted relatively accurately.

From both rebuilt flows and force results, it is clear that the traditional ROM cannot predict the flow with a moving boundary at all. Meanwhile, as a powerful tool, our Global ROM with forcing term improves the performance of the prediction significantly. The main vortex structures of the flow and the forces on the solid can be predicted reasonably accurately, and much faster than the DNS.

2.3.4 Three-dimensional Flow around a Hovering Bird

For more general motions, in this section, the hovering bird which has complex rotational motions has been employed. The Reynolds number defined by the plate length and maximum tail velocity is 700. The computational domain for the ROM is $l_x \times l_y \times l_z = (-1, 2) \times (-1, 1.5) \times (-1.5, 1.5)$ with uniform mesh at $N_x \times N_y \times N_z = 120 \times 100 \times 120$, while a larger domain with stretched mesh is used for direct numerical simulation to get an accurate database. The numerical simulation is done by Tao Yang⁴⁹, and figure 2.20 shows the schematic of the flying bird, where $\alpha_c \in [-50^\circ, 100^\circ]$, $\gamma_h = 13.6^\circ$, $\beta = 49.5^\circ$, and $\eta = \alpha_c + \beta + \gamma_h$.

For the details of the flying, figure 2.21 shows how the angles changed (compared to the experimental results done by Tobalske et al.⁵⁰).

Figure 2.22 shows a typical snapshot from direct numerical simulation where vortex structures are shedding around the bird.

By applying the same approach on the hovering bird case, similarly, the phase portraits of coefficients and the rebuilt flow are plotted in figures 2.23 and 3.15. For this complex motion

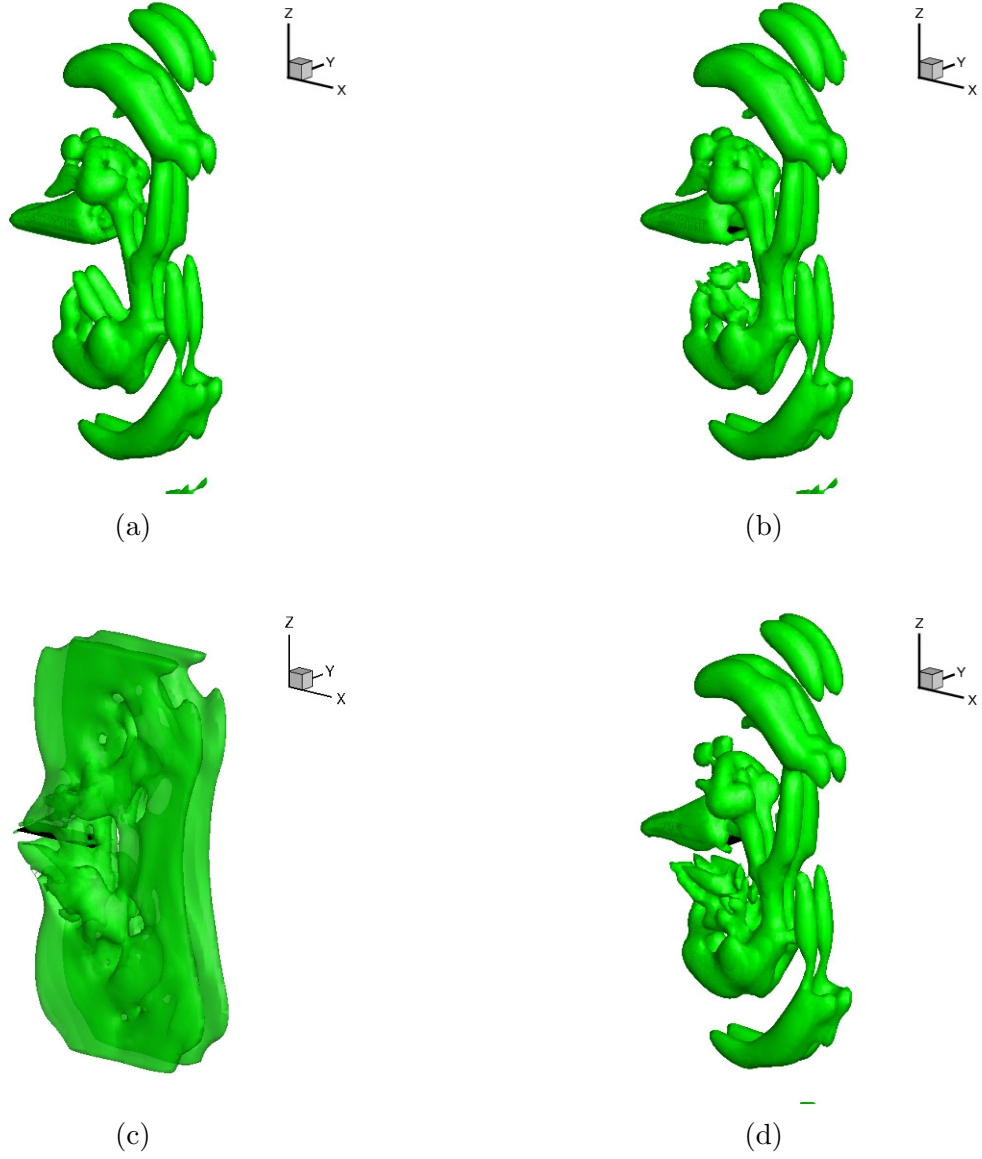


Figure 2.18: Comparison of flow field by iso-contour of Q-criterion after four periods of simulation/prediction of (a) DNS, (b) rebuilt result by global POD coefficients computed by DNS, (c) traditional ROM, and (d) global ROM with continuous solid domain.

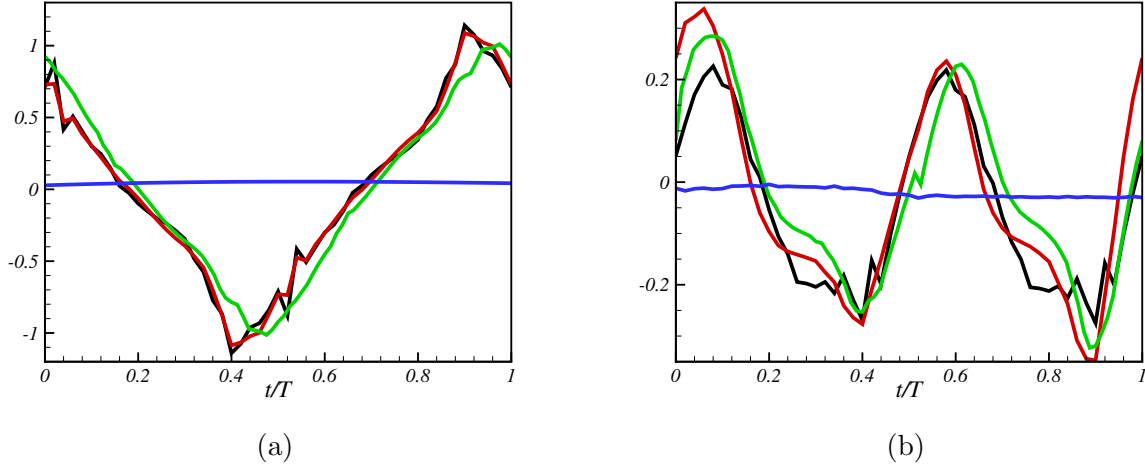


Figure 2.19: Comparison of (a) lift (F_z) and (b) drag (F_x) in one period after four periods simulation/prediction, computed by (—) DNS, rebuilt result by (—) global POD coefficients computed by DNS, (—) traditional ROM, and (—) global ROM with continuous solid domain.

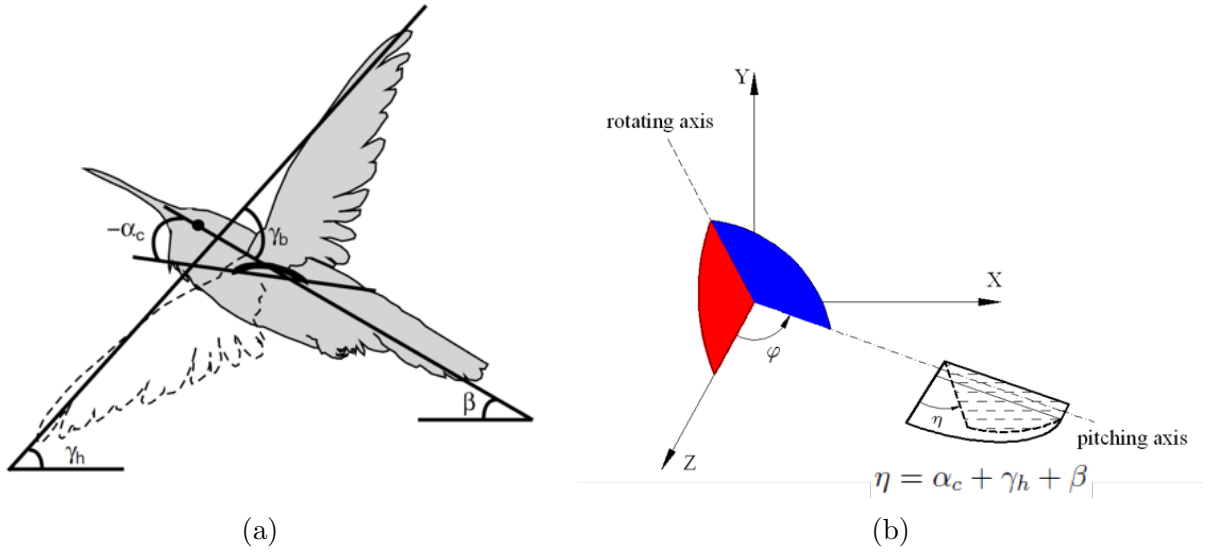


Figure 2.20: Schematic of a hovering bird.

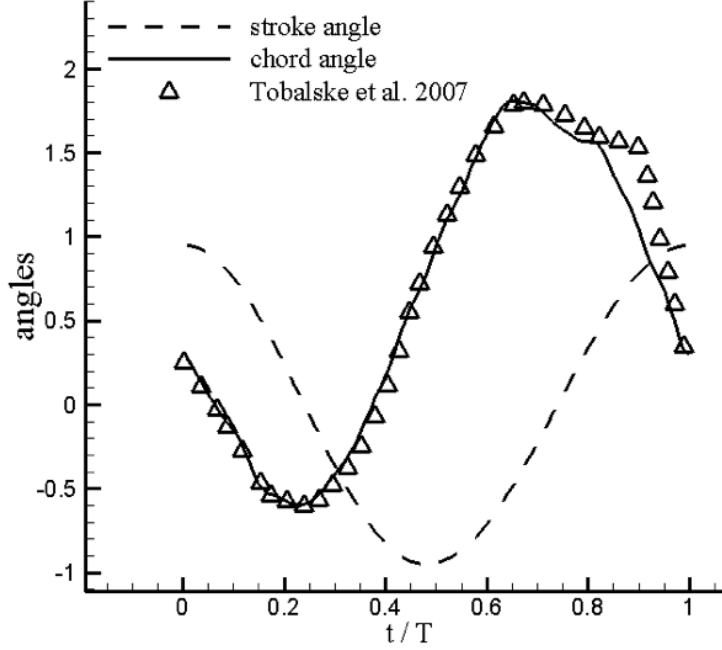


Figure 2.21: Angles change with time for the hovering bird.

problem, the global ROM results look more inaccurate than the previous cases, especially for the near flow around the bird. However, because the main vortex structures generated by the bird have still been predicted, from the force results in figure 2.25, it is demonstrated that our global ROM is still accurate enough to preserve the basic dynamics and make the prediction stable.

2.3.5 Three-dimensional Flow around a Flexible Flapping Plate

After the prediction of several rigid body motions, in this section, the flexible flapping plate, whose motions are directly taken from the latest experiments done by J. Tran et al.²⁷, has been predicted. The experimental setup by J. Tran et al. is shown in figure 2.26, the flexible plate is forced to rotate around the z-axis with the deformation due to the flexibility.

This is a fluid-structure interaction problem, and because of the deformation, it is hard to define the reference velocity. So the maximum tail velocity from a rigid plate with the same setup, as $U = l\theta_0 2\pi f$, has been employed as the reference velocity, and the Reynolds

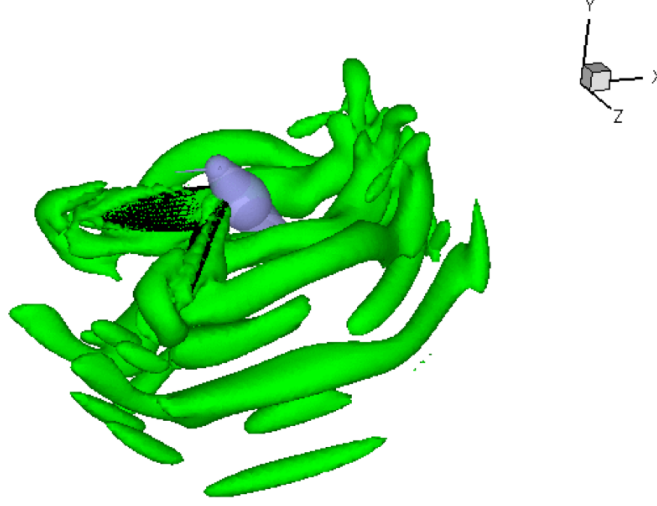


Figure 2.22: A typical snapshot from direct numerical simulation: vorticity structure of flow around a hovering bird by iso-contour of Q-criterion.

number defined by the plate length and this velocity is about 962. The computational domain for the ROM is $l_x \times l_y \times l_z = (-2, 4) \times (-5, 5) \times (-1.6, 1.6)$ with uniform mesh at $N_x \times N_y \times N_z = 150 \times 250 \times 80$, while the larger domain with stretched mesh is used for direct numerical simulation to get an accurate database. The plate with non-dimensional size $(1, 0.002768, 0.122983)$ was placed at the $x-o-z$ plane along the x-axis. The rotation center is at $(x_0, y_0, z_0) = (0, 0, 0)$, while the leading edge of the plate is at $x = 0.114679$. The plate is forced to rotate around the z-axis by $\theta(t) = \theta_0 \sin(2\pi ft)$ with maximum angle $\theta_0 = \frac{63}{180}\pi$ and frequency $f = 0.25$. Figure 2.27 shows a typical snapshot from direct numerical simulation where vortex structures are shedding around the plate.

The global POD results have been plotted in figure 2.28 as the energy distribution, the mean flow, and the first four modes.

By applying the same approach on the flexible flapping plate case, similarly, the phase portraits of coefficients, amplitudes of oscillation, and the rebuilt flow are plotted in figures 2.29, 2.30, and 2.31. From the phase portraits, with enough information of solid motion, the

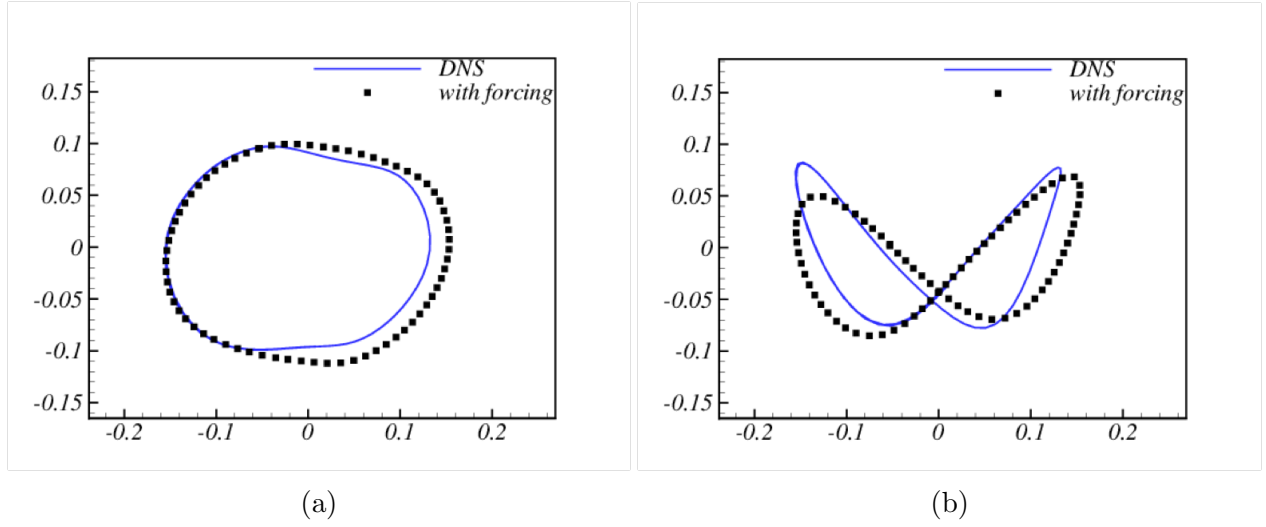


Figure 2.23: The phase portraits of: (a) (a_1, a_2) and (b) (a_1, a_3) , where the results computed from continuous global ROM are benchmarked by the result from direct numerical simulation.



Figure 2.24: Comparison of flow field by iso-contour of Q-criterion after four periods of simulation/prediction of DNS, rebuilt result by (a) global POD coefficients computed by DNS and (b) global ROM with continuous solid domain.

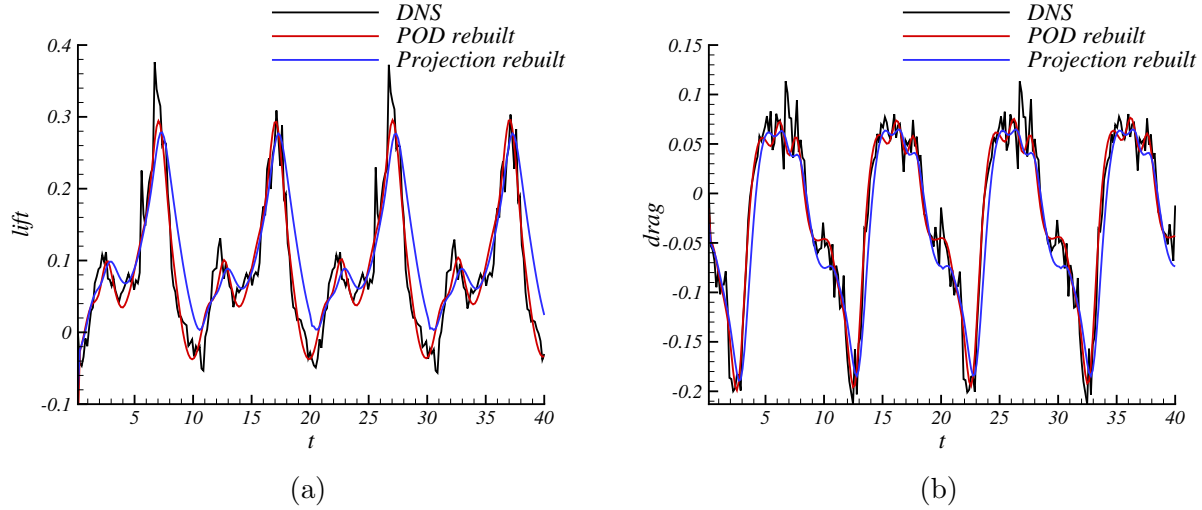


Figure 2.25: Comparison of (a) lift (F_z) and (b) drag (F_x), computed by (—) DNS, rebuilt result by (—) global POD coefficients computed by DNS, and (—) global ROM with continuous solid domain.

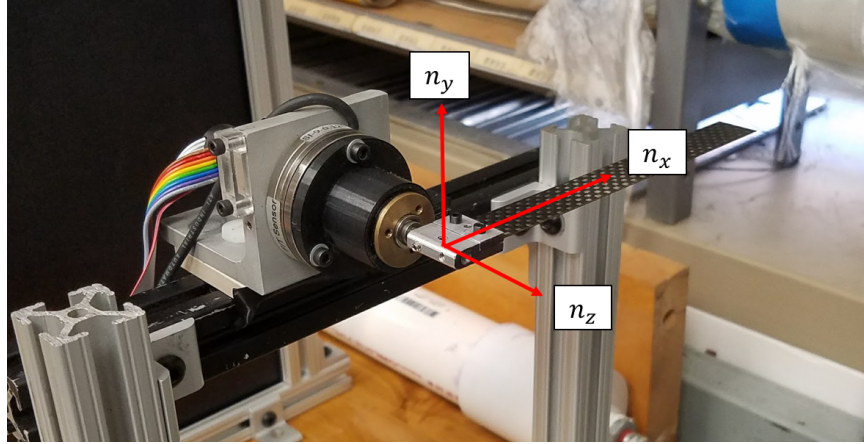


Figure 2.26: The experimental setup of the flexible flapping plate done by J. Tran et al.

flexibility has not ruined the function of our forcing term: our method can still handle this kind of fluid-structure interaction problem. From the amplitudes of oscillation, although the mean error of global ROM is about 15%, it is still much less than the error of traditional ROM which is almost 100%. From the rebuilt results, even for this flexible motion, our global ROM can still predict the main vortex structures accurately, although the near flow close to the plate is not accurate due to the large motion of the plate.

From all the above cases, it is demonstrated that our global POD-Galerkin based ROM

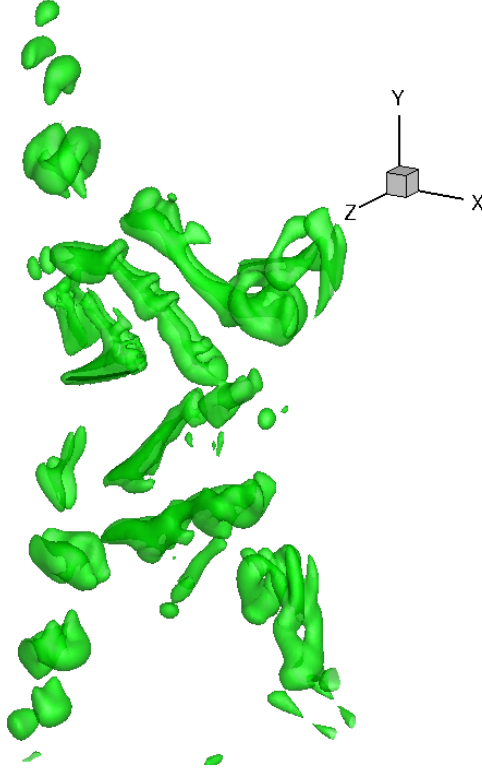


Figure 2.27: A typical snapshot from direct numerical simulation: vorticity structure of flow around a flexible flapping plate by iso-contour of Q -criterion.

with forcing term is powerful and robust for almost all kinds of moving boundary problems, the low-dimensional prediction has been improved significantly from the traditional ROM without the forcing term. The method in this chapter is using a continuous solid domain to describe the solid motion, so-called continuous approach. It is straightforward, easy to apply, and suitable for any kind of motion, either rigid or flexible, either prescribed or fluid-structure interacted. However, because the low-dimensional parameters in the forcing term are still time-dependent, it is hard to pre-calculate them and the method is still relatively time-consuming, which will devalue the benefits of ROM itself. A more powerful method is still required to further reduce the computational cost, and make those parameters pre-calculable.

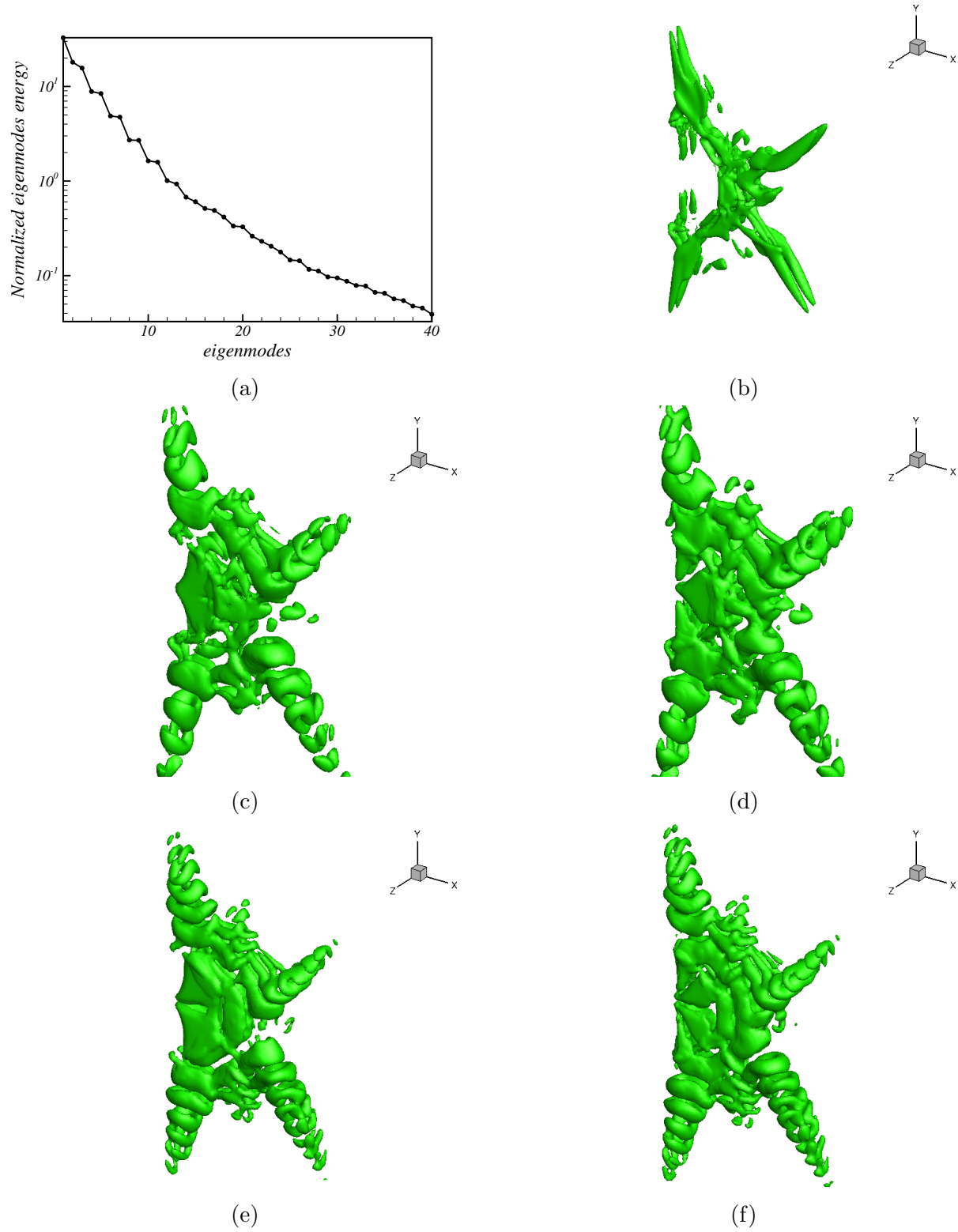


Figure 2.28: The energy distribution and most energetic modes by iso-contour of Q-criterion from POD analysis of flow passing a flexible flapping plate: (a) the energy spectrum of POD modes; (b) mean flow; (c) mode 1; (d) mode 2; (e) mode 3; (f) mode 4.

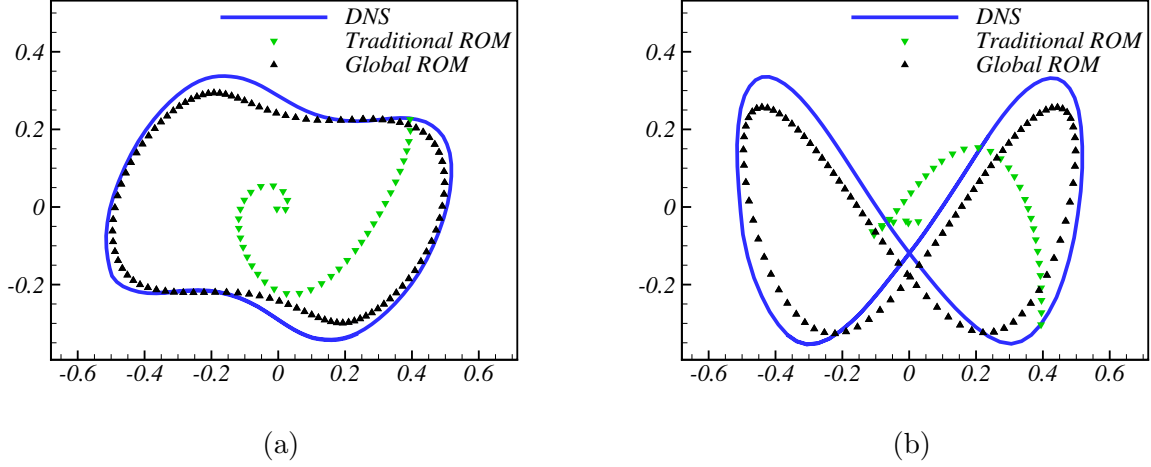


Figure 2.29: The phase portraits of: (a) (a_1, a_2) and (b) (a_1, a_3) , where the results computed from continuous global ROM and traditional ROM without forcing terms are benchmarked by the result from direct numerical simulation.

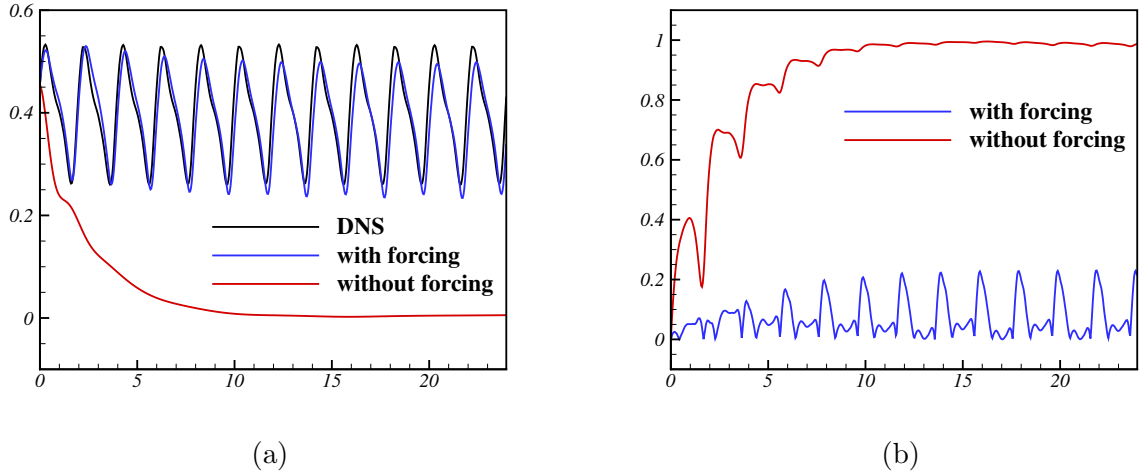


Figure 2.30: The comparison of different ROMs for flow around a flexible flapping plate: (a) the amplitude of oscillation $A := \sqrt{a_1^2 + a_2^2}$ and (b) the error of amplitude $E = |A_r - A_d|/A_d$.

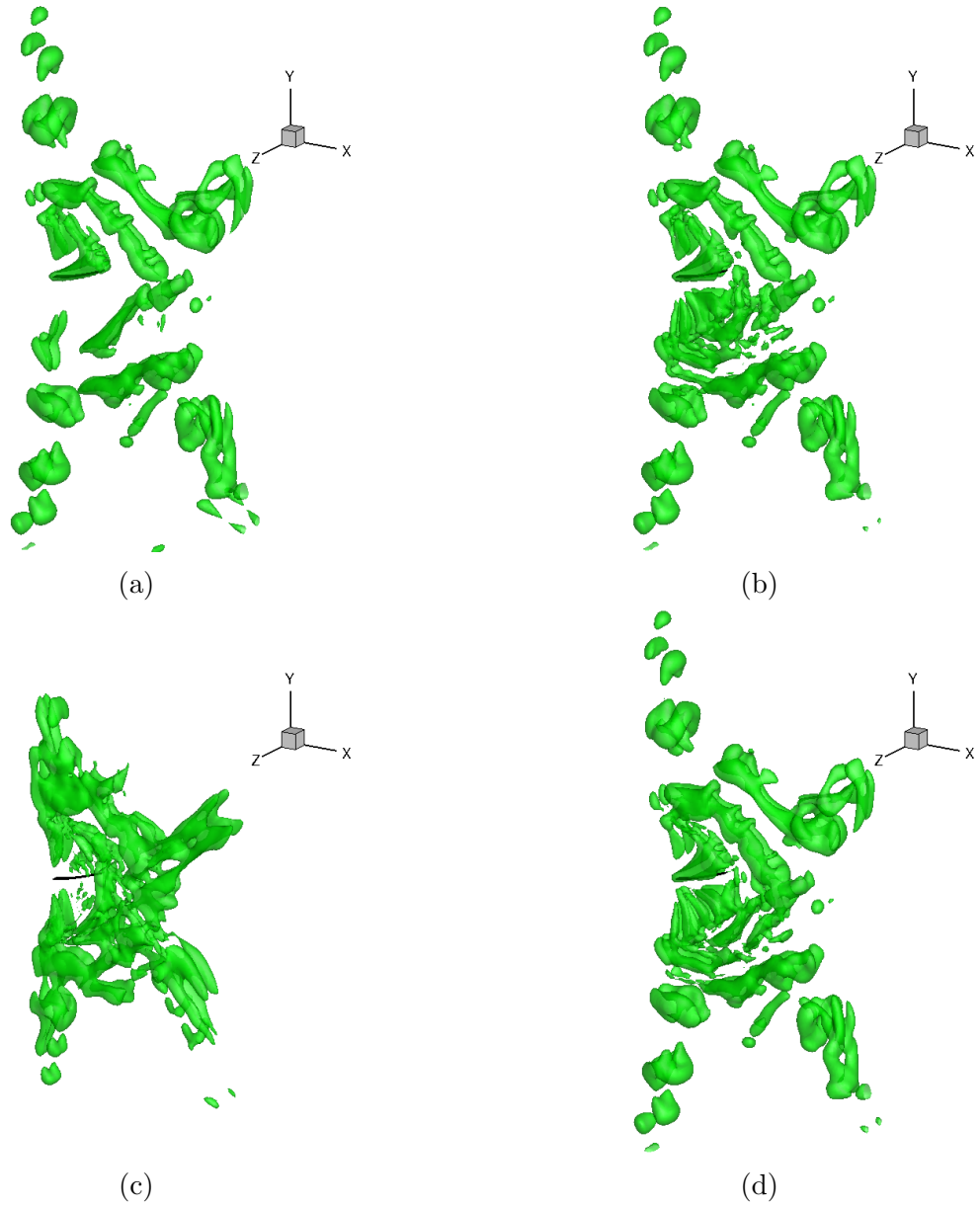


Figure 2.31: Comparison of flow field around a flexible flapping plate by iso-contour of Q -criterion after four periods of simulation/prediction of (a) DNS, (b) rebuilt result by global POD coefficients computed by DNS, (c) traditional ROM, and (d) global ROM with continuous solid domain.

Chapter 3

Global POD-Galerkin Based Reduced-Order Models for Flows with Moving Boundaries: Decomposed Solid Domain

In this chapter, the idea of solid domain decomposition is introduced to improve the performance of our global POD-Galerkin Based Reduced-Order Models for flows with moving boundaries. By this approach, the time-varied low-dimensional coefficients can be transferred to fixed and pre-calculated coefficients. Meanwhile, the computational time cost can also be reduced significantly. In the following sections, first, the methodology of the decomposed approach is introduced, and then, the method has been applied on a two-dimensional flow passing a large amplitude oscillatory cylinder, a three-dimensional flow passing an oscillatory sphere, and a three-dimensional flow around a rigid flapping plate.

3.1 Methodology of Global POD-Galerkin Projection for Flows with Moving Boundary: Decomposed Solid Domain

In section ??, the global ROM with continuous solid domain $\Omega_s(t)$, so-called continuous approach, has been introduced. Since $\Omega_s(t)$ is a time-dependent function, the integration in (2.29–2.31) needs to be computed at every time step. Though the computation process for simple integration is much less complex than the original high-fidelity model, and the solid domain is also much smaller than the fluid domain, the increased computational cost by these terms is much larger than the ROM itself and overshadows the merit of using ROM (this will be discussed detailed in section 4.1). In the current work, solid domain decomposition²³ is implemented to remove this extra cost and make the coefficients pre-calculated.

The characteristic function is introduced to describe the solid motion:

$$\chi_s = \begin{cases} 1 & \text{in } \Omega_s \\ 0 & \text{otherwise} \end{cases}, \quad (3.1)$$

the integration in (2.29–2.31) can be rewritten as:

$$l'_{ij} = \langle (\frac{1}{Re} \nabla^2 \phi_j + \frac{1}{\Delta t} \phi_j), \chi_s \phi_i \rangle, \quad (3.2)$$

$$q'_{ijk} = \langle \nabla \cdot (\phi_j \phi_k), \chi_s \phi_i \rangle, \quad (3.3)$$

$$c'_i = \langle \frac{1}{\Delta t} \mathbf{V}, \chi_s \phi_i \rangle. \quad (3.4)$$

Similar to the fluid field, the solid domain itself, marked now by χ_s , can also be decomposed to individual modes, and the dimension is then reduced from a continuous space (or a space at the order of grid points in numerical simulation) to a low-order space with a truncated

number of modes:

$$\chi_s(x, t) = \sum_{m=0}^{\infty} b_m(t) \psi_m(x) \approx \sum_{m=1}^M b_m(t) \psi_m(x), \quad (3.5)$$

where ψ is the solid modes computed by a similar POD approach but applied on domain function χ_s with the inner product as:

$$\langle \chi_s^i, \chi_s^j \rangle = \int_{\Omega} \chi_s^i \chi_s^j d\Omega. \quad (3.6)$$

The truncation of modes provides a good approximation of the same coefficients:

$$l'_{ij} \approx \sum_{m=0}^M \langle (\frac{1}{Re} \nabla^2 \phi_j + \frac{1}{\Delta t} \phi_j), \psi_m \phi_i \rangle b_m = \sum_{m=0}^M l_{ijm}^* b_m, \quad (3.7)$$

$$q'_{ijk} \approx \sum_{m=0}^M \langle \nabla \cdot (\phi_j \phi_k), \psi_m \phi_i \rangle b_m = \sum_{m=0}^M q_{ijkm}^* b_m, \quad (3.8)$$

$$c'_i \approx \sum_{m=0}^M \langle \frac{1}{\Delta t} \mathbf{V}, \psi_m \phi_i \rangle b_m = \sum_{m=0}^M c_{im}^* b_m, \quad (3.9)$$

where the new integrals, l_{ijm}^* and q_{ijkm}^* , are not time-dependent and need to be computed only once for each model. The time-dependent variables are in b_m which are at much lower order (e.g. number of modes) than the original solid domain (e.g. number of mesh points in the solid domain). Then, by putting these new terms in equation (2.28), the new governing equations can be written as:

$$\dot{a}_i = \sum_{j=0}^N (\frac{1}{Re} l_{ij} - \sum_{m=0}^M l_{ijm}^* b_m) a_j - \sum_{j=0}^N \sum_{k=0}^N (q_{ijk} - \sum_{m=0}^M q_{ijkm}^* b_m) a_j a_k + \sum_{m=0}^M c_{im}^* b_m, \quad i = 1, \dots, N. \quad (3.10)$$

Different from Liberge and Hamdouni's model, our model can be easily closed by introducing the level-set like equation (3.11) to track the domain changes, while their model needs to update the Lagrange multiplier which cannot be modeled in the model order reduction.

$$\dot{b}_n = - \sum_{n=0}^N \langle \mathbf{V}(\mathbf{x}, t) \cdot \nabla \psi_m, \psi_n \rangle b_m, \quad n = 1, \dots, N. \quad (3.11)$$

For some extremely complex flexible deformations (e.g. the flow-structure interactions in section 2.3.5), $\mathbf{V}(\mathbf{x}, t)$ may need be specified for every time step. However, for most of the prescribed motion, the Eulerian solid velocity can be further decoupled as:

$$\mathbf{V}(\mathbf{x}, t) = \sum_p T_p(t) \mathbf{X}_p(\mathbf{x}). \quad (3.12)$$

Currently only the rigid body motions, i.e. translation and rotation, are considered, so the velocity field can be decoupled and largely simplified as:

$$\begin{aligned} \mathbf{V}(\mathbf{x}, t) &= \dot{x}_c \vec{n}_x + \dot{y}_c \vec{n}_y + \dot{z}_c \vec{n}_z + \vec{\omega} \times (\vec{r} - \vec{r}_c) \\ &= \dot{x}_c \vec{n}_x + \dot{y}_c \vec{n}_y + \dot{z}_c \vec{n}_z + (-\omega_z(y - y_c) + \omega_y(z - z_c) + \omega_x(x - x_c) \\ &\quad - \omega_x(z - z_c) - \omega_y(x - x_c) + \omega_z(y - y_c))^T \\ &= (\dot{x}_c - \omega_y z_c + \omega_z y_c) \vec{n}_x + (\dot{y}_c - \omega_z x_c + \omega_x z_c) \vec{n}_y + (\dot{z}_c - \omega_x y_c + \omega_y x_c) \vec{n}_z \\ &\quad + \omega_x(0, -z, y)^T + \omega_y(z, 0, -x)^T + \omega_z(-y, x, 0)^T \\ &= V_x \vec{n}_x + V_y \vec{n}_y + V_z \vec{n}_z + \omega_x \vec{R}_x + \omega_y \vec{R}_y + \omega_z \vec{R}_z \\ &= \sum_{p=1}^6 T_p(t) \mathbf{X}_p(\mathbf{x}). \end{aligned} \quad (3.13)$$

Thus, the velocity field has been written as the combination of the fixed “modes” multiplied by time-varied scalar numbers, which allows the inner products in equations (3.9) and (3.11) to be decoupled from time:

$$c_{im}^* = \left\langle \frac{1}{\Delta t} \mathbf{V}, \psi_m \phi_i \right\rangle = \frac{1}{\Delta t} \sum_{p=1}^6 T_p(t) \langle \mathbf{X}_p, \psi_m \phi_i \rangle, \quad (3.14)$$

$$\langle \mathbf{V}(x, t) \cdot \nabla \psi_m, \psi_n \rangle = \sum_{p=1}^6 T_p(t) \langle \mathbf{X}_p \cdot \nabla \psi_m, \psi_n \rangle. \quad (3.15)$$

For the flexible motion prescribed by the additional deformation basis functions, the appendix A shows a detailed decoupling of a two-dimensional flexible pitching and plunging plate. Other motions can also be applied by a similar approach.

In the above approach, the time-dependent inner products are transferred to the combination of several fixed inner products multiplied by time-varied scaler numbers, and these fixed inner products only need to be computed once, which will largely reduce the computational time cost.

3.2 Applications and Results: Decomposed Solid Domain

In this section, the decomposed approach is first applied to model a two-dimensional flow passing an oscillatory cylinder, then a three-dimensional flow passing an oscillatory sphere, and finally a three-dimensional flow around a rigid flapping plate.

3.2.1 Two-dimensional Flow Passing an Oscillatory Cylinder

First, the decomposed approach is applied on the same two-dimensional flow passing a large amplitude oscillatory cylinder as shown in chapter 2.3.1. All the computational and physical parameters are the same, which lead to the same DNS database (figure 2.2) and flow modes (figure 2.3).

Then, to remove the computational cost burdened by the integration at each time step, the domain decomposition approach, which has a much lower time cost, has been applied. The results have been compared to global ROM with a continuous solid domain in section 2.3.1.

For solid decomposition, eight solid domain modes are used, which can describe the solid motion fairly well as shown in figure 3.1. The time-dependent coefficients in level-set like equation (3.11) for solid domain change also show accurate estimations, as shown in the comparison of the phase portraits of coefficients computed by equation (3.11) and the exact

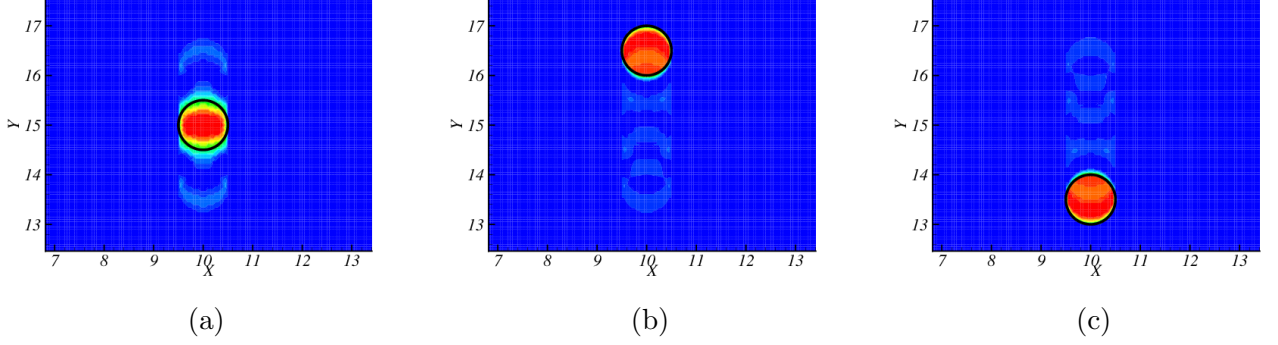


Figure 3.1: Comparison of the exact shape and location of the solid (black circle) and the shape rebuilt from eight solid modes (color contours) at three time moments in one period: (a) $t = 0$, (b) $t = 1/4T$, (c) $t = 3/4T$.

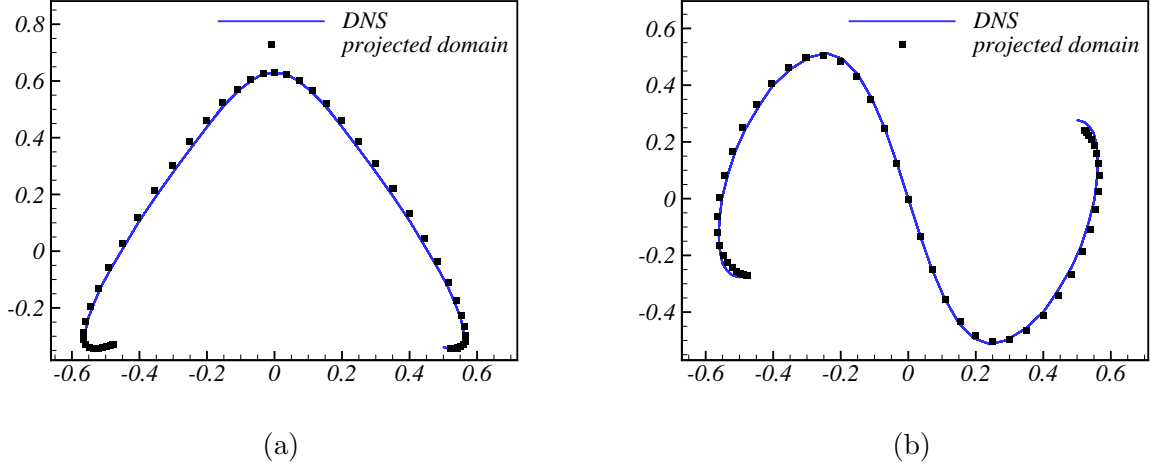
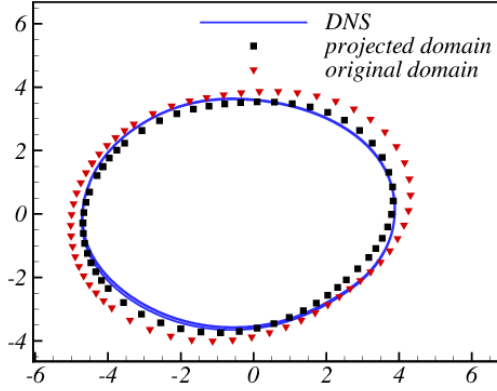


Figure 3.2: Phase portraits of the coefficients for solid domain modes: (a) (b_1, b_2) and (b) (b_1, b_3) .

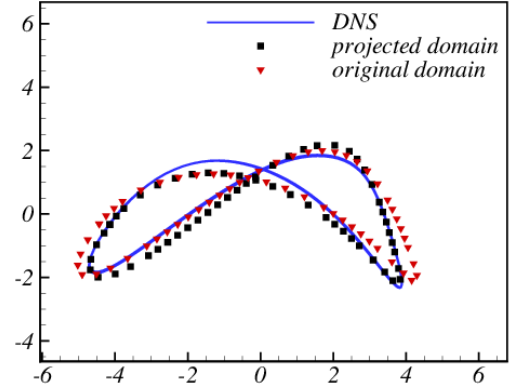
solution directly from the POD of solid domain (figure 3.2).

With this decomposed solid domain, the new decomposed approach can also preserve the energy and dynamics for this prescribed large solid motion problem which has been shown by the phase portraits in figure 3.3. It has also shown that the accuracies of both continuous approach and decomposed approach are preserved almost the same in their comparison to the DNS (figure 3.4).

Similarly, the amplitude of oscillation $A := \sqrt{a_1^2 + a_2^2}$ and the error defined as $E = |A_r - A_d|/A_d$ are also plotted in figure 3.5. The decomposed approach performs almost the

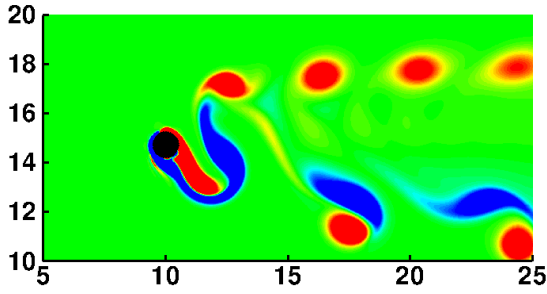


(a)

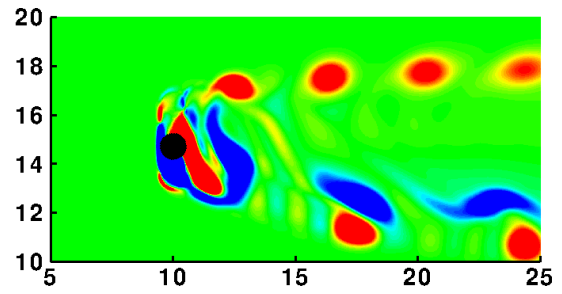


(b)

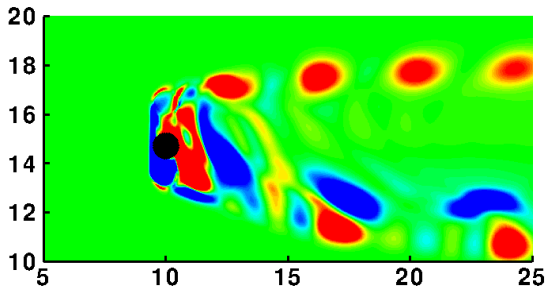
Figure 3.3: Comparison of phase portraits of global POD coefficients computed by DNS, solid domain decomposition (i.e. projected domain), and continuous solid domain (i.e. original domain): (a) (a_1, a_2) and (b) (a_1, a_3)



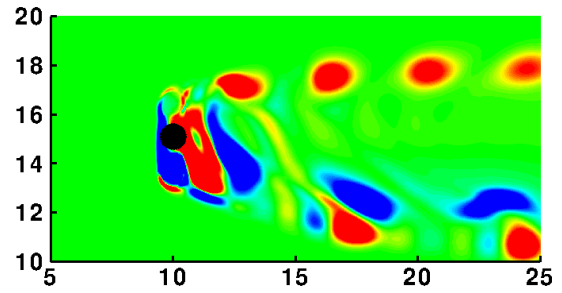
(a)



(b)



(c)



(d)

Figure 3.4: Comparison of flow field by iso-contour of Q-criterion after four periods of simulation/prediction of DNS, rebuilt result by global POD coefficients computed by DNS, traditional ROM, and global ROM with continuous solid domain: (a) DNS flow field; (b) rebuilt result by global POD coefficients computed by DNS; (c) rebuilt result by global ROM with continuous solid domain; (d) rebuilt result by global ROM with decomposed solid domain.

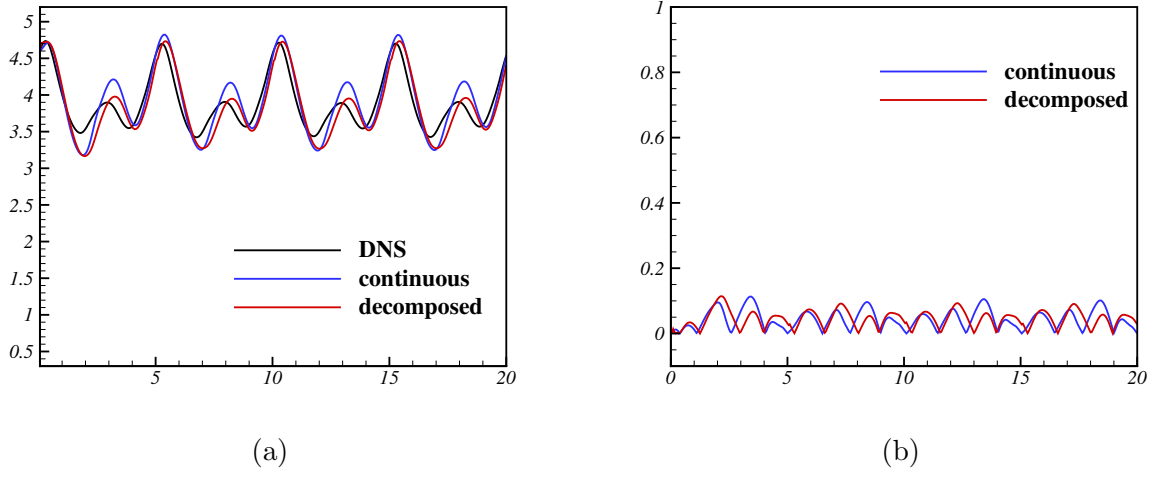


Figure 3.5: The comparison of different ROMs for flow passing an oscillatory cylinder: (a) the amplitude of oscillation $A := \sqrt{a_1^2 + a_2^2}$ and (b) the error of amplitude $E = |A_r - A_d|/A_d$.

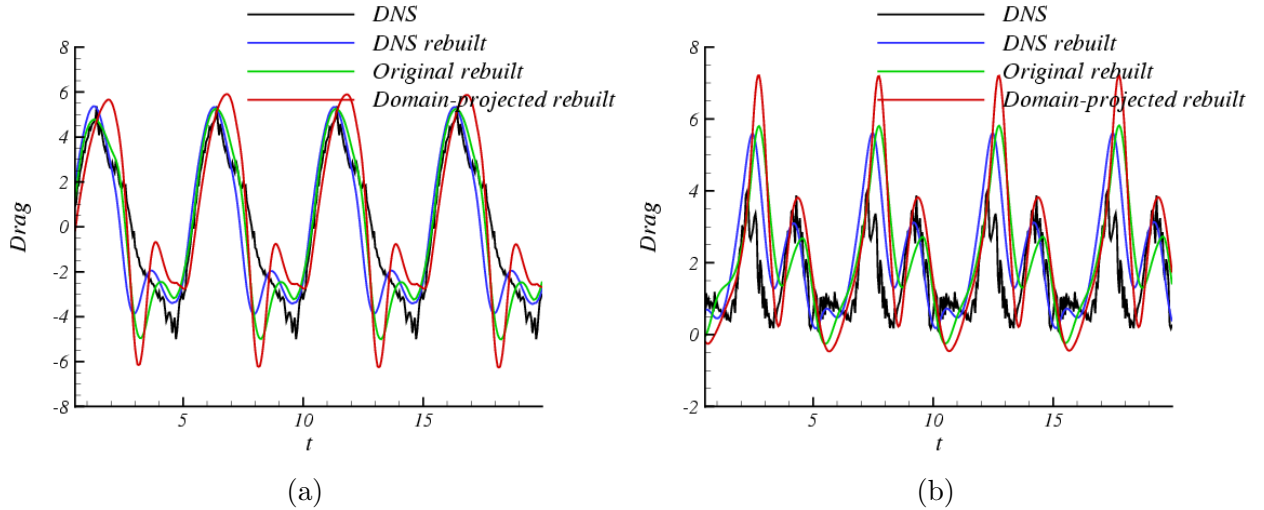


Figure 3.6: Comparison of (a) lift and (b) drag computed by (—) DNS, rebuilt result by (—) global POD coefficients computed by DNS, (—) solid domain decomposition, and (—) continuous solid domain.

same as the continuous approach.

The lift and drag for the cylinder have also been calculated via control volume approach and plotted in figure 3.6.

Compared to the continuous approach, the new decomposed approach predicts almost the same results. Although there are some small differences in the force results, the main

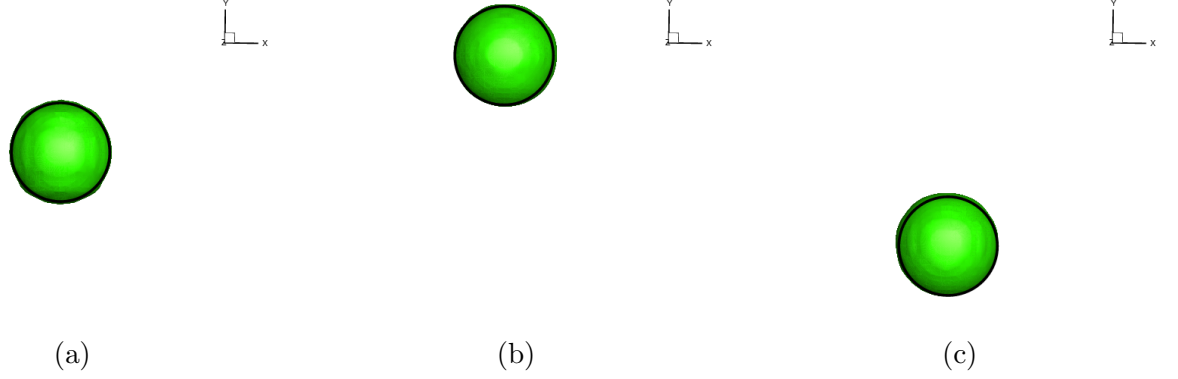


Figure 3.7: Comparison of the exact shape and location of the solid (black circle) and the shape rebuilt from eight solid modes (iso-surface in green) at three time moments in one period: (a) $t = 0$, (b) $t = 1/4T$, (c) $t = 3/4T$.

shapes, frequencies, and amplitudes are still predicted with reasonable accuracy. Compared to the benefits of saving the computational cost and making the coefficients pre-calculated, this sacrifice of accuracy is acceptable.

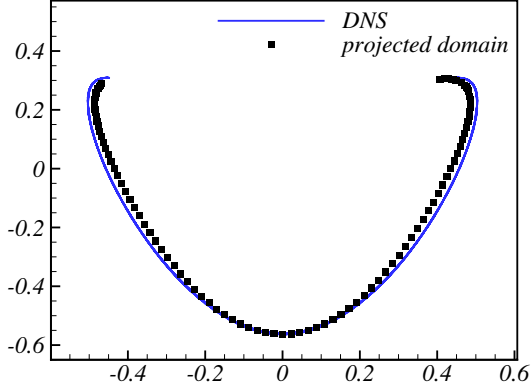
3.2.2 Three-dimensional Flow Passing an Oscillatory Sphere

Similarly, the decomposed approach of our global ROM has been applied to the same three-dimensional flow passing an oscillatory sphere as in chapter 2.3.2. For the decomposed approach, the accuracy of domain decomposition itself is compared first. Shown in figure 3.7, by using eight solid domain modes, the shapes and locations of solid domain (i.e. the sphere) can be captured in a very accurate manner.

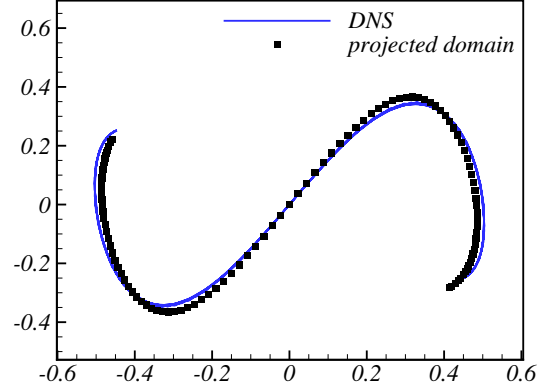
The dynamics of the solid POD modes described by equation (3.11) and the global POD modes described by (3.10) have been captured successfully by the respective ROMs in a similar comparison of phase portraits of coefficients (figures 3.8 and 3.9).

The amplitude of oscillation $A := \sqrt{a_1^2 + a_2^2}$ and the error of it $E = |A_r - A_d|/A_d$ are also plotted in figure 3.10 to compare the performance of different ROMs. Similar to the two-dimensional results, both approaches keep almost the same accuracy.

Also, the rebuilt results by both approaches are compared by the global POD modes and above ROMs' coefficients to the DNS flow field (figure 3.11). Same as the continuous approach, the flow near the solid is not as accurate as the DNS results, but the mainly

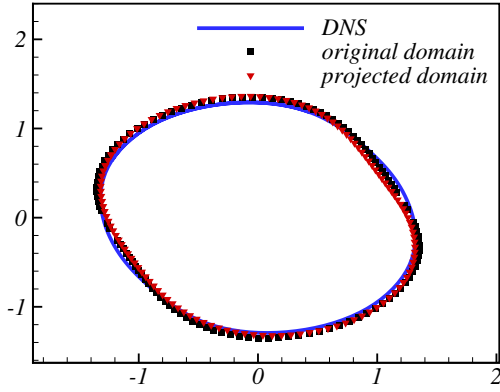


(a)

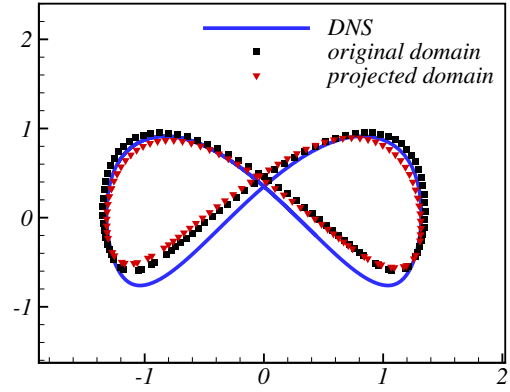


(b)

Figure 3.8: Phase portraits of the coefficients for solid domain modes: (a) (b_1, b_2) and (b) (b_1, b_3) .



(a)



(b)

Figure 3.9: Comparison of phase portraits of global POD coefficients computed by DNS, solid domain decomposition (i.e. projected domain), and continuous solid domain (i.e. original domain): (a) (a_1, a_2) and (b) (a_1, a_3) .

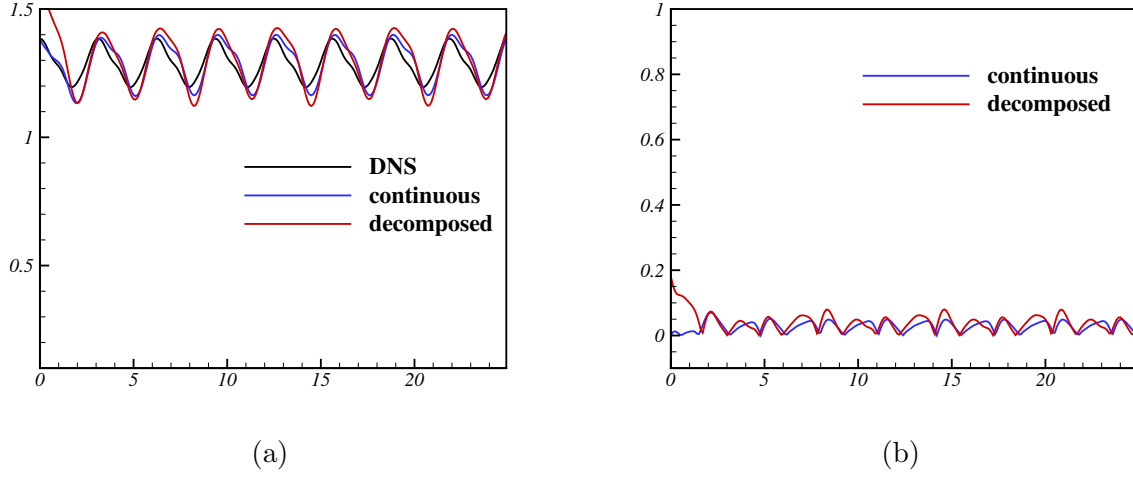


Figure 3.10: The comparison of different ROMs for flow passing an oscillatory sphere: (a) the amplitude of oscillation $A := \sqrt{a_1^2 + a_2^2}$ and (b) the error of amplitude $E = |A_r - A_d|/A_d$.

vortexes predicted by the solid decomposed ROM in the downstream are kept accurately and the dynamics of the system are preserved. The lift and drag are also calculated by control volume approach (figure 3.12). For both the continuous solid domain approach and the decomposed solid domain approach, there are only small differences from the DNS results. This result further explains that the inaccuracy near the solid is acceptable, and our model preserves the energy and main fluid structures for this dynamic system, and makes it stable, accurate enough, and practical.

3.2.3 Three-dimensional Flow around Flapping Plate

Same as the continuous approach, the decomposed approach has also been applied on the translation motion problem. The validation case compared to the experiments done by J. Tran et al.²⁷ has been employed again. By applying the same approach on the flapping plate case, the fundamental dynamics can also be described by the phase portraits of coefficients (figure 3.13). Because rotation is more complex than translation, and the plate is very thin, the results are not as accurate as the above cases, especially for the amplitude of a_2 . However, the basic dynamics are still preserved very well.

The amplitude of oscillation $A := \sqrt{a_1^2 + a_2^2}$ and the error of it $E = |A_r - A_d|/A_d$ are

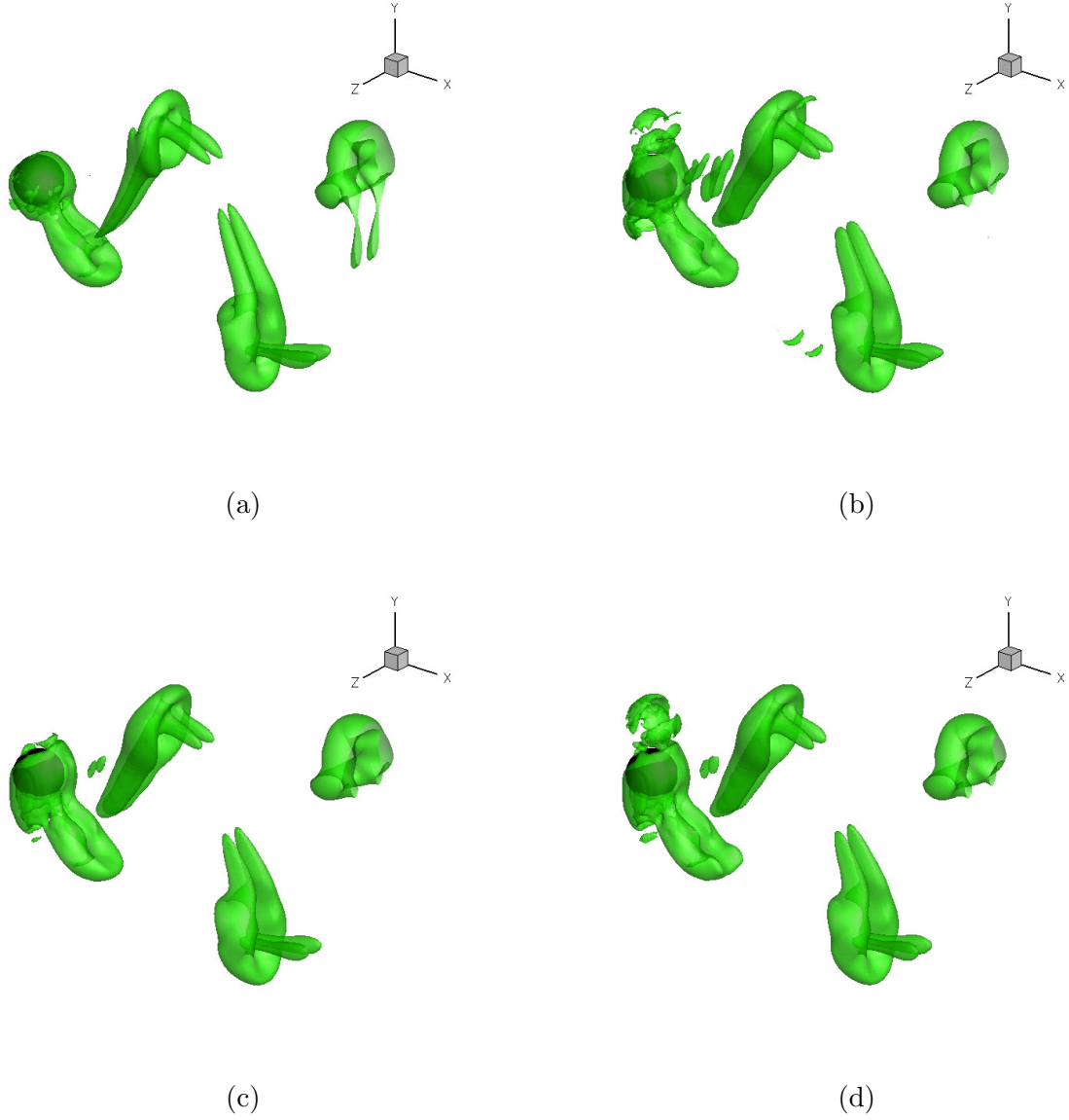
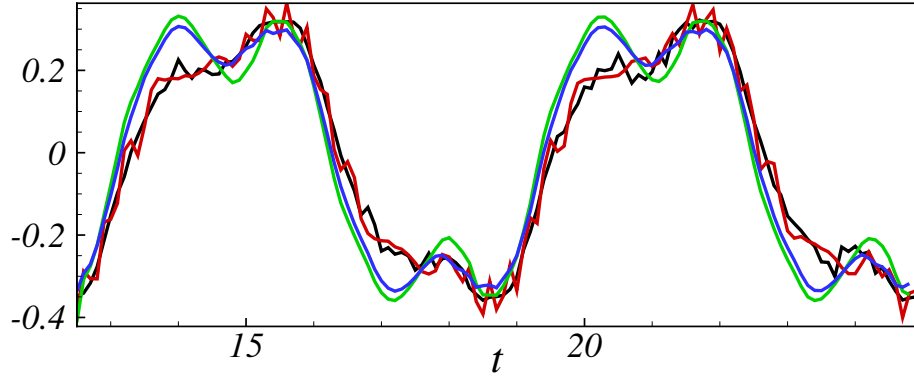
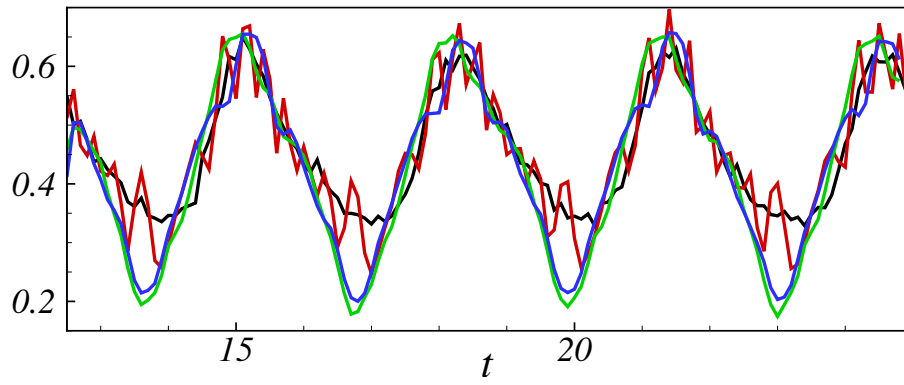


Figure 3.11: Comparison of flow field by iso-contour of Q-criterion of DNS, rebuilt result by global POD coefficients computed by DNS, solid domain decomposition, and continuous solid domain: (a) DNS flow field; (b) rebuilt result by global POD coefficients computed by DNS; (c) rebuilt result by continuous solid domain approach; (d) rebuilt result by solid domain decomposition approach.



(a)



(b)

Figure 3.12: Comparison of (a) lift and (b) drag computed by (—) DNS, rebuilt result by (—) global POD coefficients computed by DNS, (—) solid domain decomposition, and (—) continuous solid domain.

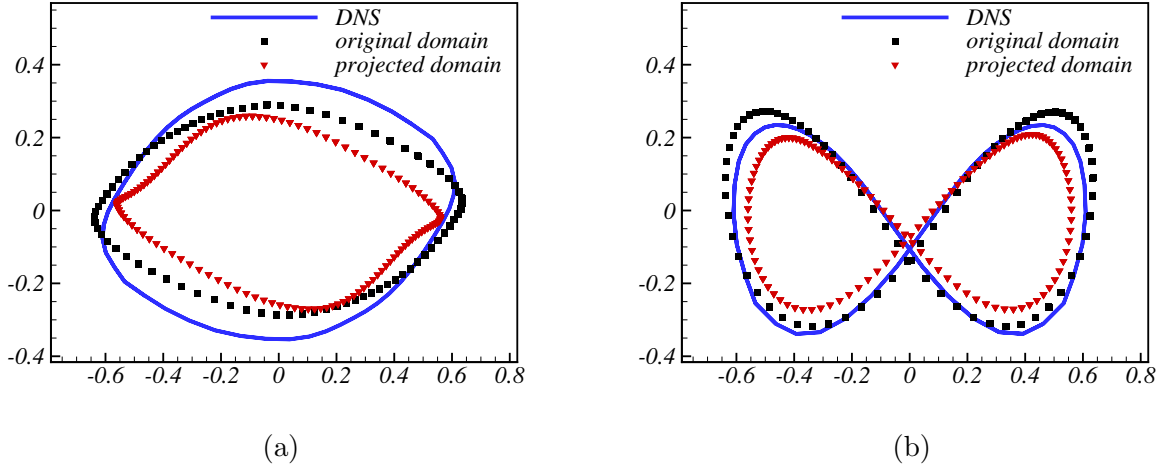


Figure 3.13: Comparison of phase portraits of global POD coefficients computed by DNS, solid domain decomposition (i.e. projected domain), and continuous solid domain (i.e. original domain): (a) (a_1, a_2) and (b) (a_1, a_3) .

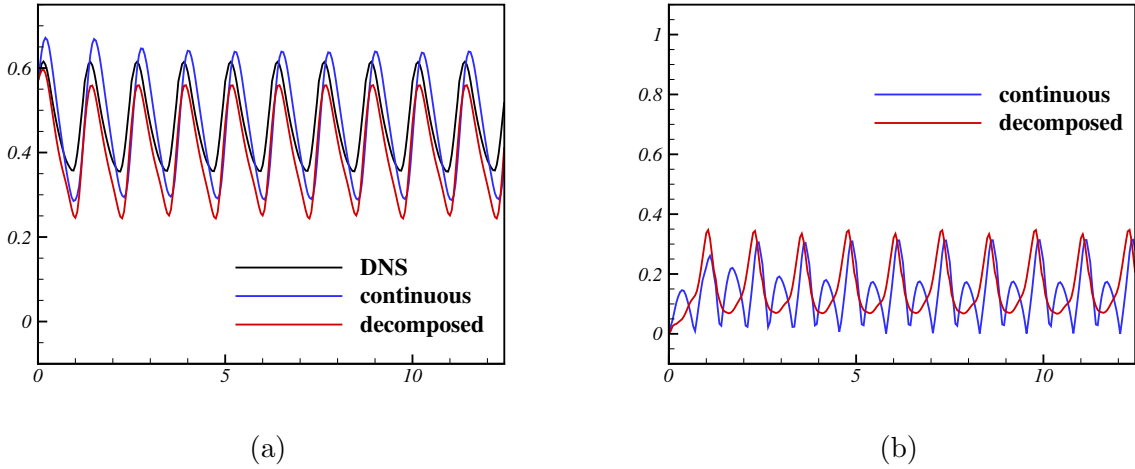


Figure 3.14: The comparison of different ROMs for flow around a flapping plate: (a) the amplitude of oscillation $A := \sqrt{a_1^2 + a_2^2}$ and (b) the error of amplitude $E = |A_r - A_d|/A_d$.

also plotted in figure 2.17 to compare the performance of different ROMs. It shows that the decomposed approach is slightly less accurate than the continuous approach in this case. However, the error is still relatively lower than the traditional ROM without the forcing term, and is still acceptable.

Similarly, by comparing the rebuilt results (figure 3.15) and lift and drag (figure 3.16), the decomposed approach is also demonstrated robust for the rotation problems, and can

still preserve the energy and main fluid structures for the dynamic system. And as shown in the previous work²⁷, the overall forces of global ROMs are smaller than the experimental results, reasonably, in simulation at the lower Reynolds number, but all basic features are closely represented.

From the above cases, our global POD-Galerkin based ROM with decomposed solid domain has been proven to be powerful for the moving boundary problems with prescribed solid motions. It not only saves the computational time cost, but also makes the low-dimensional coefficients pre-calculated, which is a huge benefit for optimization and real time control. However, compared to the continuous domain approach, the decomposed domain approach requires the decoupling of the solid velocity in Eulerian frame, which is not available for some complex flow-structure interaction problems. However, it is still not that clear how much time has been saved by our global ROMs, and we are always seeking a further improved model that is faster.

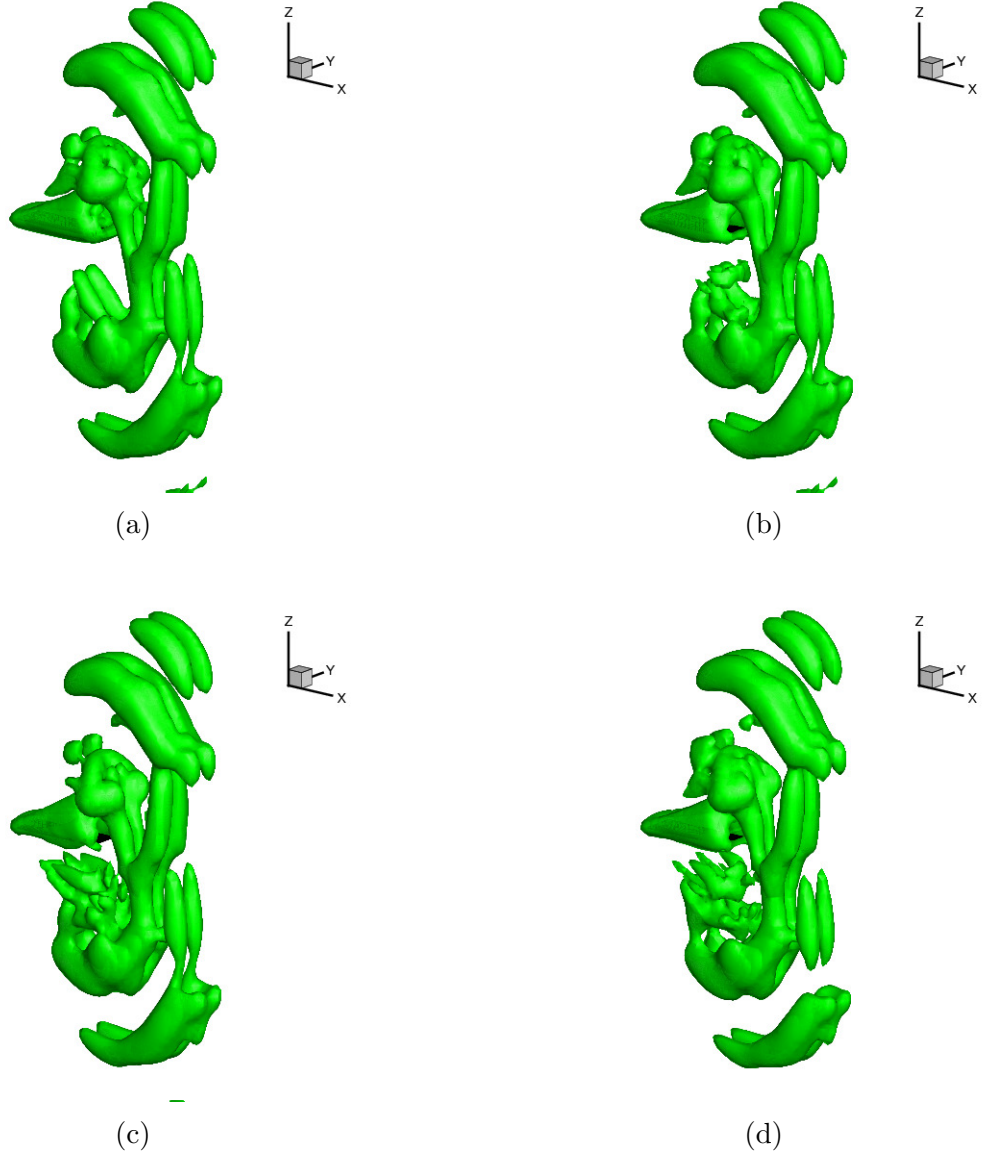


Figure 3.15: Comparison of flow field by iso-contour of Q -criterion of DNS, rebuilt result by global POD coefficients computed by DNS, solid domain decomposition, and continuous solid domain: (a) DNS flow field; (b) rebuilt result by global POD coefficients computed by DNS; (c) rebuilt result by continuous solid domain approach; (d) rebuilt result by solid domain decomposition approach.

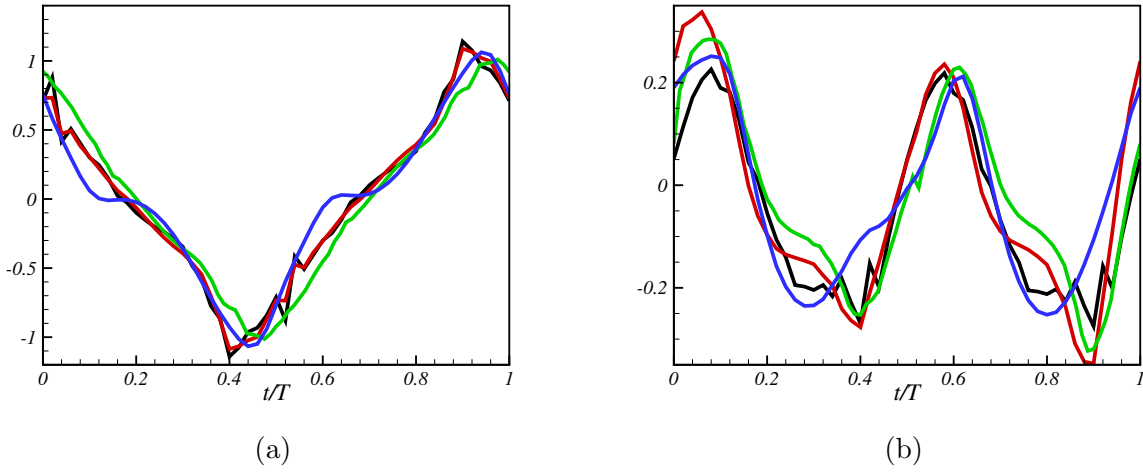


Figure 3.16: Comparison of (a) lift (F_z) and (b) drag (F_x) in one period computed by (—) DNS, rebuilt result by (—) global POD coefficients computed by DNS, (—) solid domain decomposition, and (—) continuous solid domain.

Chapter 4

The Analysis of Computational Time Cost for Different Terms and Different Models

In this section, analysis is conducted to check the time cost of individual terms in model (2.28) and model (3.10) and to seek the possibility of dropping expensive terms without sacrificing accuracy.

4.1 Term Analysis

For simplicity, terms in both the continuous domain model and the decomposed domain model are regrouped and represent the models in a new form:

$$\dot{a}_i = F_i(a_j, a_k) + f_i^{(1)} + f_i^{(2)} + f_i^{(3)}, \quad (4.1)$$

where

$$F_i(a_j, a_k) = \frac{1}{\text{Re}} \sum_{j=0}^N l_{ij} a_j + \sum_{j=0}^N \sum_{k=0}^N q_{ijk} a_j a_k, \quad (4.2)$$

which comes from the traditional POD-Galerkin approach for stationary domain, that, in our case, is the combined fluid and solid domain. It has fixed parameters in both the continuous solid domain approach and the decomposed solid domain approach, which give them the benefit that they can be pre-calculated and saved for future applications. The other three terms, $f_s^i, i = 1, 2, 3$, represent the terms for the description of solid motion. They have different formats for two different approaches. For the continuous solid domain approach, they all have time-varied coefficients and can be written as:

$$f_i^{(1)} = - \sum_{j=0}^N l'_{ij} a_j, \quad (4.3)$$

$$f_i^{(2)} = \sum_{j=0}^N \sum_{k=0}^N q'_{ijk} a_j a_k, \quad (4.4)$$

$$f_i^{(3)} = c'_i. \quad (4.5)$$

Although the cases in chapter 2 are periodic in time, those coefficients are hard to be pre-calculated because different time steps lead to different coefficients. For the decomposed solid domain approach, because time has been decoupled from those time-varied coefficients with the introduction of $b_m(t)$, these three terms can be written as:

$$f_i^{*(1)} = - \sum_{j=0}^N \sum_{m=0}^M l_{ijm}^* b_m a_j, \quad (4.6)$$

$$f_i^{*(2)} = \sum_{j=0}^N \sum_{k=0}^N \sum_{m=0}^M q_{ijkm}^* b_m a_j a_k, \quad (4.7)$$

$$f_i^{*(3)} = \sum_{m=0}^M c_{im}^* b_m. \quad (4.8)$$

Thus, those time-consuming parameters become fixed and pre-calculated. Because all the inner products which relate to solid motion only have values near the solid domain, instead of doing the inner product in the whole domain, those inner products can be calculated in a smaller domain around the solid to achieve the exact same results. Thus, the computational time cost can be estimated with the nomenclature defined in table 4.1, where H is the

computational time cost for one inner product on one mesh points.

Table 4.1: The nomenclature of computational time cost.

Label	Definition	Estimated order of the time cost
N	Number of fluid modes	10^1
M	Number of solid modes	10^1
N_p	Number of snapshots in one period	10^2
I	The inner product cost in the whole domain	$(2D)10^5 H / (3D)10^6 H$
I_s	The inner product cost in the domain near the solid	$(2D)10^4 H / (3D)10^5 H$

It should be noted that I and I_s can be changed for different cases, and here two cases from both two dimensions (a flow passing an oscillatory cylinder) and three dimensions (a flow passing an oscillatory sphere) in chapters 2 and 3 are taken as examples to estimate the computational time cost. For both methods and both cases, different terms have different time costs, as shown in table 4.2.

Table 4.2: The cost analysis of computational time for different moving-boundary terms.

Term	Most time-costly computation	Time cost	Order of time cost
$F_i(a_j, a_k)$	The inner product for q_{ijk}	$N^3 \times I$	$(2D)10^8 H / (3D)10^9 H$
$f_i^{(1)}$	The inner product for l'_{ij}	$N^2 \times I_s \times N_p$	$(2D)10^8 H / (3D)10^9 H$
$f_i^{(2)}$	The inner product for q'_{ijk}	$N^3 \times I_s \times N_p$	$(2D)10^9 H / (3D)10^{10} H$
$f_i^{(3)}$	The inner product for c'_i	$N \times I_s \times N_p$	$(2D)10^7 H / (3D)10^8 H$
$f_i^{*(1)}$	The inner product for l^*_{ijm}	$N^2 \times I_s \times M$	$(2D)10^7 H / (3D)10^8 H$
$f_i^{*(2)}$	The inner product for $q^*_{ijk m}$	$N^3 \times I_s \times M$	$(2D)10^8 H / (3D)10^9 H$
$f_i^{*(3)}$	The inner product for c^*_{im}	$N \times I_s \times M$	$(2D)10^6 H / (3D)10^7 H$

In most of the cases, $M = 8(O(10^1))$ is enough to represent the solid motion. Meanwhile, N_p usually should be more than one hundred($O(10^2)$). Because $M \ll N_p$, the decomposed solid domain method demonstrates that it can reduced the total time cost significantly.

From each methods, it is no doubt that the $f_i^{(2)}(f_i^{*(2)})$ term is the most expensive term in the computations. If this term can be directly dropped without decreasing the accuracy, the computational time cost will be further reduced.

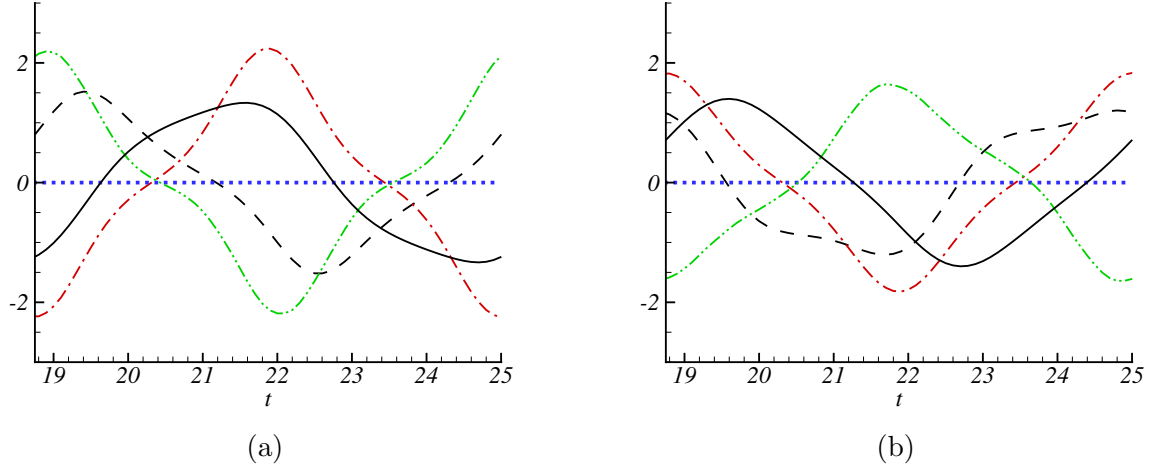


Figure 4.1: Different contributions of each terms for (a) a_1 and (b) a_2 by continuous domain method (three-dimensional oscillatory sphere case, $dt=0.1$): $(---) F_i(a_j, a_k)$, $(-.-) f_i^{(1)}$, $(\cdots) f_i^{(2)}$, $(-\cdots) f_i^{(3)}$, and $(—) a_i$.

4.2 Term Dropping for the Flow Passing an Oscillatory Sphere

In order to check the importance of the expensive $f_i^{(2)}/f_i^{*(2)}$ term, the contributions of each terms should be analyzed, respectively. Using the case of the three-dimensional flow passing an oscillatory sphere as an example, the different contributions of each terms were plotted out in figures 4.1 and 4.2.

In both approaches, the contributions of $f_i^{(2)}/f_i^{*(2)}$ to both a_1 and a_2 (which correspond to the most energetic modes containing about 60% total energy) are very small. To confirm its small impact to actual dynamics, these nonlinear solid terms are directly removed from the models to build the modified models and are compared to the original models with all terms. Figures 4.3 and 4.4 show that almost the same accuracies are achieved by both continuous and decomposed approaches for the three-dimensional flow passing an oscillatory sphere when those expensive terms have been removed. The flow fields of each method have also been plotted in figure 4.5, and the impact of removing the expensive terms is so small that no difference can be detected from the rebuilt flow field.

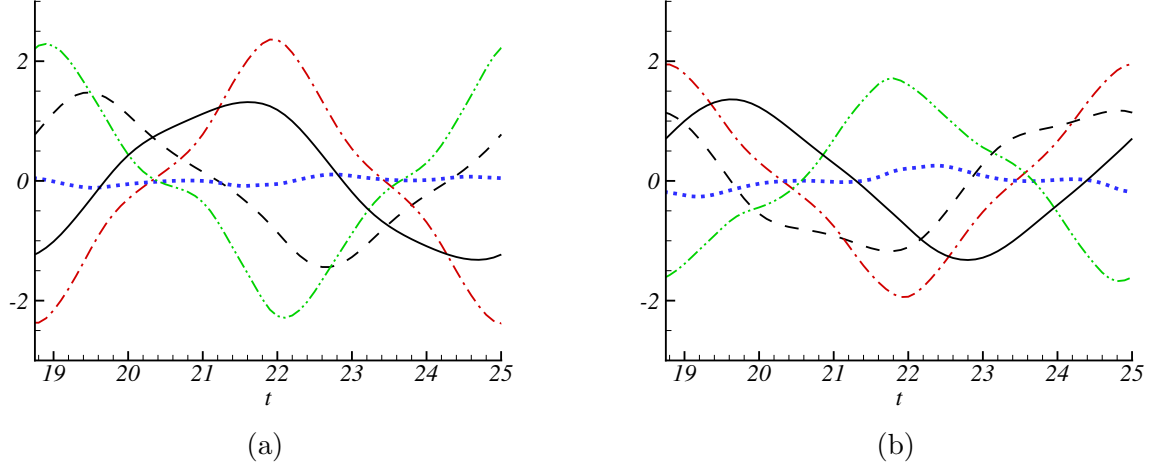


Figure 4.2: Different contributions of each terms for (a) a_1 and (b) a_2 by decomposed domain method(three-dimensional oscillatory sphere case, $dt=0.1$): $(- - -) F_i(a_j, a_k)$, $(- \cdot -) f_i^{*(1)}$, $(\cdot \cdot \cdot) f_i^{*(2)}$, $(- \cdot \cdot) f_i^{*(3)}$, and $(—) a_i$.

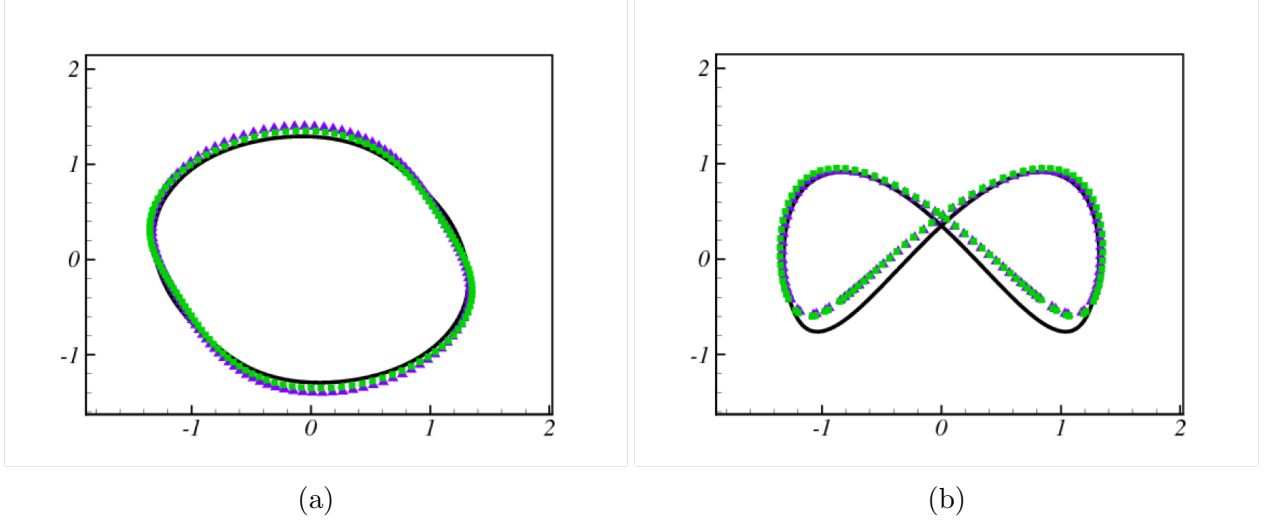


Figure 4.3: Phase portrait of (a) a_1 and (b) a_2 (three-dimensional oscillatory sphere case, $dt=0.1$): $(—)$ DNS projection, $(\blacksquare)$ continuous ROM with $f_i^{(2)}$, and $(\blacktriangle)$ continuous ROM without $f_i^{(2)}$.

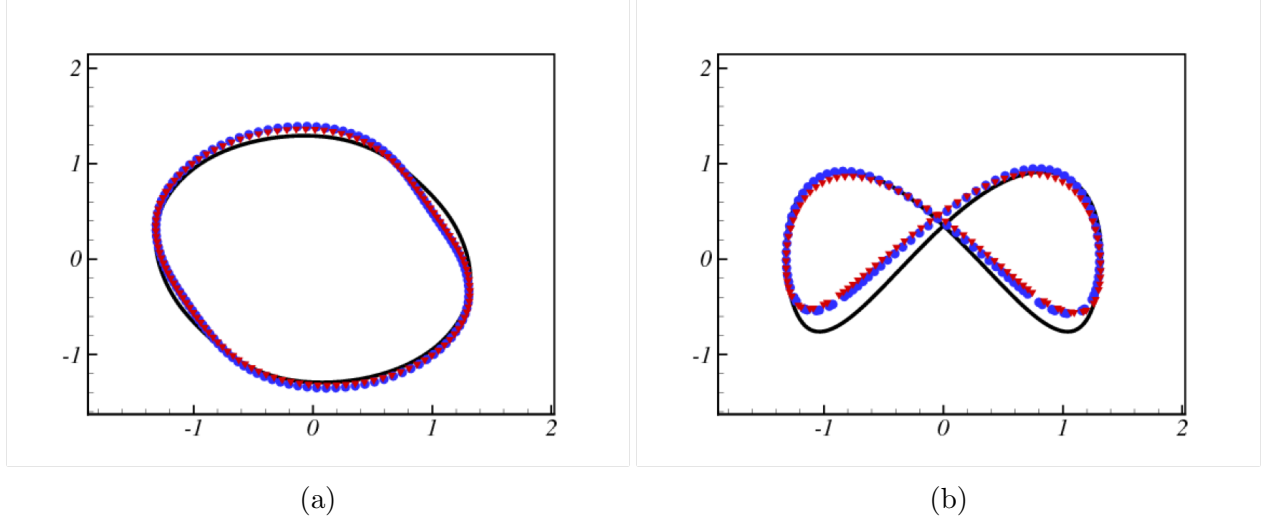


Figure 4.4: Phase portrait of (a) a_1 and (b) a_2 (three-dimensional oscillatory sphere case, $dt=0.1$): (—) DNS projection, ($\bullet$) decomposed ROM with $f_i^{*(2)}$, and ($\blacktriangledown$) decomposed ROM without $f_i^{*(2)}$.

For this particular case, the computational time costs for all the above models are computed and compared to DNS time cost and the “easiest” ROM, which directly rebuild the flow field every step and do the calculation (4.3). As noticed earlier in table 4.2, the modified model has reduced the total cost by one order, while the accuracy of the model is still preserved. Compared to the other methods, the decomposed solid domain model transfers almost all online-calculation to off-line calculation, which means almost everything is pre-calculated and ready for control and optimization.

Table 4.3: The time cost of four period calculations of different methods for eight modes oscillatory sphere case.

Method	Online-time	Offline-time
DNS	7 hours 15 mins	0
“Easiest” method with Q terms	2 hours 8 mins	7 mins
Continuous method with Q terms	1-2 s	1 hour 54 mins
Decomposed method with Q terms	1-2 s	1 hour 1 mins
“Easiest” method without Q terms	52 mins	7 mins
Continuous method without Q terms	1-2 s	22 mins
Decomposed method without Q terms	1-2 s	8.4 mins

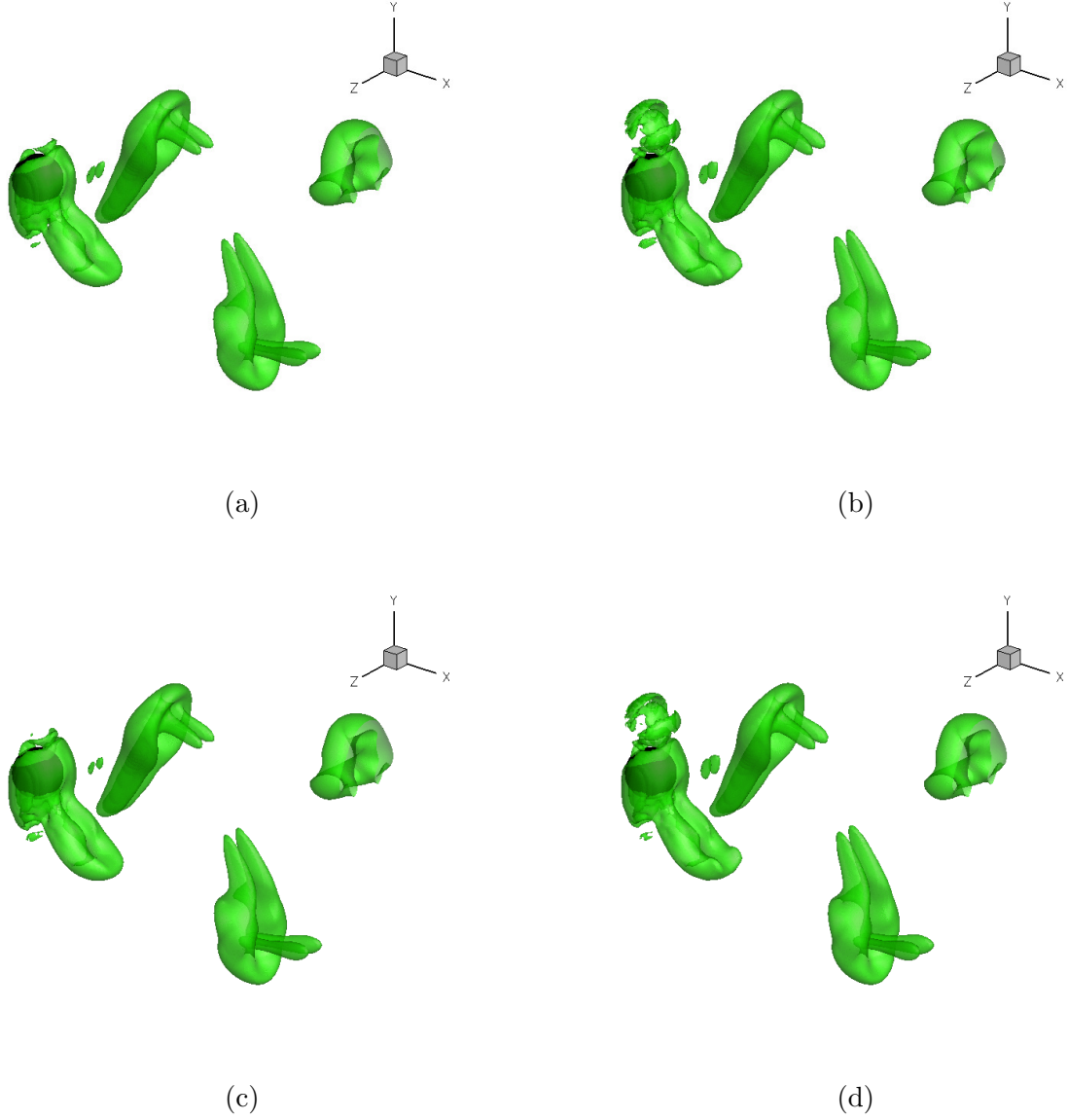


Figure 4.5: Comparison of flow field by iso-contour of Q-criterion after four periods of simulation/prediction of DNS: (a) rebuilt result by global continuous ROM with $f_i^{(2)}$ terms; (b) rebuilt result by global decomposed ROM with $f_i^{*(2)}$ terms; (c) rebuilt result by global continuous ROM without $f_i^{(2)}$ terms; (d) rebuilt result by global decomposed ROM without $f_i^{*(2)}$ term.

Chapter 5

Model Adaptation of Parametric Variation for Global POD-Galerkin Models with Forcing Term

In this chapter, the problem of parameter variation for the ROM is discussed. First, our original global ROM applied for the parametric variation is introduced in section ??, the method is applied to the three-dimensional flow passing an oscillatory sphere with the variation of oscillatory amplitude. Then the shift mode is introduced to improve the model adaptation of the original global ROM in section ?. However, although it works fine for the Reynolds number variation of the fixed cylinder flow, shift mode is not powerful enough to improve the results of moving boundary problems. Finally, the combined model, as a more powerful tool, is introduced in section ?? to improve the model adaptation.

5.1 Parametric Variation by Global ROM with Forcing Term

In this section, how the forcing term works for the parametric variation problem is shown firstly. Then, the method is applied on the three-dimensional flow passing an oscillatory

sphere for the variation of oscillation amplitude.

5.1.1 Methodology of Parametric Variation by Global ROM with Forcing Term

As discussed in chapters 2 and 3, by introducing the characteristic function χ_s , the information of solid domain is decoupled from the fluid modes ϕ_i . The equation (2.28) can be rearranged as:

$$\dot{a}_i = \sum_{j=0}^N \frac{1}{\text{Re}} l_{ij} a_j - \sum_{j=0}^N \sum_{k=0}^N q_{ijk} a_j a_k + \mathbf{F}(\chi_s), \quad i = 1, \dots, N. \quad (5.1)$$

In this way, the moving of the body is explicit in the equation, and for the parametric controlled moving problem, the parametric variation is applicable by changing parameters to control the changing of χ_s .

However, for the above continuous approach, the moving of the boundary is only decoupled from the fluid modes, but still coupled in the low-dimensional forcing term coefficients l'_{ij} , q'_{ijk} and c'_i . Although the parametric variation is applicable, these coefficients need to be re-calculated every time when the moving parameters are changed. It is a significant drawback for the ROM because these parameters in whole parametric space are very time-consuming and hard to be pre-calculated, which devalues and overshadows the benefit of ROM.

In chapter 3, the decomposed approach has been demonstrated to be as accurate as the continuous approach and more efficient. And what's more, this decomposition is much more powerful for the decoupling of moving parameters. Because of the decomposition of solid domain, the new low-dimensional variables l^*_{ijm} , q^*_{ijkm} and c^*_{im} have been decoupled from the time-dependent coefficients b_m . By introducing all the possible χ_s in whole moving parametric space, the solid modes for the whole parametric space can be easily defined. Since the b_m are controlled by the level-set like equation (3.11), the parametric variation is specified by the prescribed velocity $\mathbf{V}(\mathbf{x}, t)$, which applied to both the low-dimensional

equations (3.10) and (3.11). From the equation (3.11), we know that b_m can also be written as the function of $\mathbf{V}(\mathbf{x}, t)$, so the equation (5.1) in the decomposed approach is:

$$\dot{a}_i = \sum_{j=0}^N \frac{1}{\text{Re}} l_{ij} a_j - \sum_{j=0}^N \sum_{k=0}^N q_{ijk} a_j a_k + \mathbf{F}(\mathbf{V}(\mathbf{x}, t)), \quad i = 1, \dots, N. \quad (5.2)$$

For the complex flexible deformation, $\mathbf{V}(\mathbf{x}, t)$ is hard to define by some parameters, and must be specified for every time step. However, for the parametric controlled moving, $\mathbf{V} = \mathbf{V}(c_1, c_2, \dots, c_n, t)$ is the function of moving parameters $c_1, c_2, \dots, c_n$. Especially for the rigid body moving, i.e. translation and rotation, the prescribed velocity field has been further decoupled as equation (3.13).

Thus, the velocity field has been written as the combination of the fixed vectors multiplied by time-varied scaler numbers, which means the inner product in equations (3.9) and (3.11) can also be decoupled from moving parameters as in equations (3.14) and (3.15).

For the parametric controlled moving, such as sinusoidally oscillating, these time-varied scaler numbers can be easily decoupled to the control parameters, such as amplitudes and frequencies (section 5.1.2).

For the prescribed flexible deformation, most of them can still be decoupled. A decoupling of two-dimensional pitching and plunging plate is shown in appendix A.

In the above approach, for most of the prescribed moving problems, every low-dimensional variables can be fully pre-calculated and directly used to predict problems in the whole parametric space. Different from other ad hoc terms used in some other studies, our forcing term is fully physical, accurate, and could be applied to any kind of moving boundary problems, especially for the rigid body moving problems. This method is applied to the three-dimensional flow passing an oscillatory sphere for examination.

5.1.2 Applications and Results for Three-dimensional Oscillatory Sphere with Variation of Oscillatory Amplitude

The proposed method is first applied to model the three-dimensional flow passing an oscillatory sphere in chapters 2.3.2 and 3.2.2 where the sphere is oscillating sinusoidally in the y-direction as:

$$y = a \sin(2\pi ft) \quad (5.3)$$

where a is the oscillatory amplitude as the variable parameter, and $f = 0.16$ is fixed oscillatory frequency for simplicity. Equations (3.14) and (3.15) can be written as:

$$c_{im}^* = \langle \frac{1}{\Delta t} \mathbf{V}, \psi_m \phi_i \rangle = \frac{1}{\Delta t} 2\pi f \cos(2\pi ft) \langle \vec{n}_y, \psi_m \phi_i \rangle \cdot a, \quad (5.4)$$

$$\langle \mathbf{V}(x, t) \cdot \nabla \psi_m, \psi_n \rangle = 2\pi f \cos(2\pi ft) K_{mn} \cdot a, \quad (5.5)$$

where the amplitude a has been easily decoupled from $V(x, t)$.

The native ROM prediction, in which the control parameters of ROM and DNS database are the same, for this problem has been demonstrated accurate in chapters 2 and 3. For the variation of amplitude, it is notable that $\{\chi_s(t)\}^{a \leq 1} \subseteq \{chi_s(t)\}^{a=1.0}$, so that directly using the solid modes from $a = 1.0$ case is reasonable. First, the DNS results from $a = 1.0$ have been directly employed as the database to build the fluid mode, and a is reduced to 0.5 for the parametric variation. Another native ROM, which is built on the DNS database from the same moving parameters as ROM ($a = 0.5$), has also been employed for comparison.

First, the force results are calculated to check the accuracy of the method. The 32 modes results are presented in figure 5.1. It is reasonable that parametric variation prediction requires a larger number of modes than native ROM prediction (eight is enough for this problem), because with the changing of the amplitude, the energy cannot be concentrated

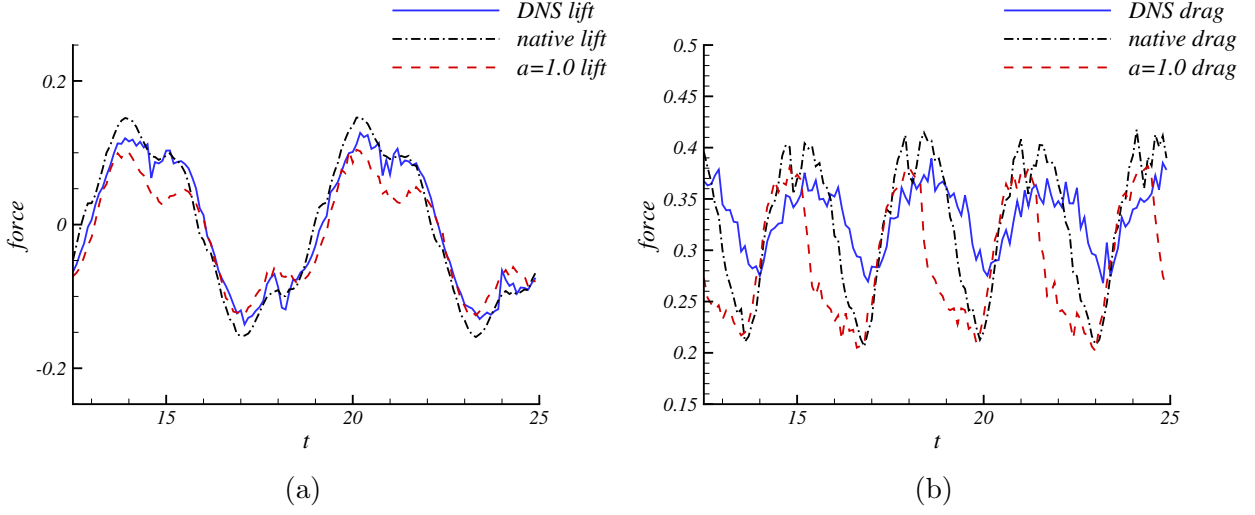


Figure 5.1: The lift and drag for the three-dimensional oscillatory sphere: (a) lift and (b) drag.

to the first several modes as what the native ROM did. The lift and drag from the 32 modes ROM seems reasonable and relatively accurate. However, the force results here are calculated by the control volume approach, which introduces the surface and volume integrals. It is hard to exactly represent the accuracy of the ROM after those integrals. In figure 5.2, the velocity profile along the y-direction at $x = 2, z = 0$ shows that although the maximal velocities and the shapes of the profile seem to be accurate (which lead to the good results of lift and drag), the shifting of the whole shape makes the ROM results not accurate anymore.

In order to get better measurement of the accuracy of our global ROM, the error defined by L^2 norm has been introduced:

$$error(t) = \frac{1}{V_\Omega} \int_\Omega \sqrt{[u(t) - u_d(t)]^2 + [v(t) - v_d(t)]^2 + [w(t) - w_d(t)]^2} dV = \frac{\|\mathbf{u}(t) - \mathbf{u}_d(t)\|}{V_\Omega}, \quad (5.6)$$

where $\mathbf{u}_d(t)$ is the original DNS data and $\mathbf{u}(t)$ is the rebuilt flow field predicted by ROMs. Figure 5.3 shows the errors defined to match the flow with the oscillation amplitude $a = 0.5$ to compare the performance of ROMs in their prediction of the flow velocity. Similarly as the force results, there are two ROMs used in this comparison: the native ROM using database

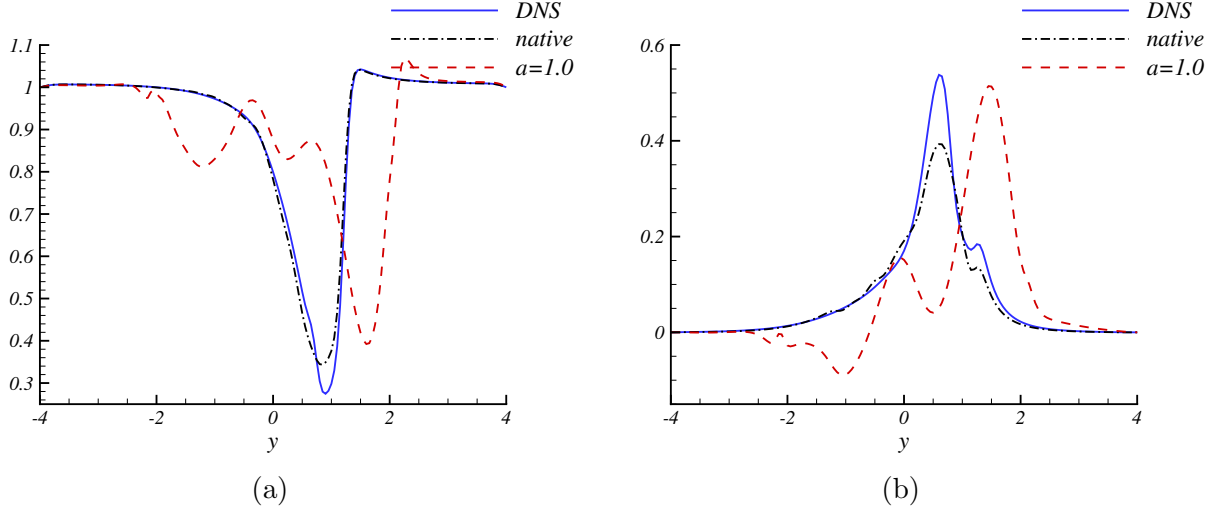


Figure 5.2: The velocity profile along the y -direction at $x = 2, z = 0$ when $t = nT + 3/4T$. (a) u_x and (b) u_y .

at $a = 0.5$ (as the benchmark), and the direct use of the single-parameter ROM at $a = 1.0$ to predict the flow with $a = 0.5$.

Although our forcing term can powerfully represent the parametric variation and make the global ROM results stable, the error of the ROM increased dramatically when the moving parameter (amplitude) deviated from the native parameter which generated the fluid modes. This means the model adaptation of the normal ROM is far from enough, and more techniques are required to make the improvements.

5.2 Shift Mode

In this section, the shift mode is employed to improve the model adaptation for the moving parametric variation. Since the shift mode introduced by Noack et al.⁵ had significantly improved the model adaptation of the Reynolds number for two-dimensional flow passing a fixed cylinder case, it is natural to apply this technique on our global ROM with moving boundaries. Noack's shift mode is defined on the differences between the mean flow and the steady flow. However, for the moving boundary problem, there is no steady flow for the shifting, so the two mean flows from different databases with different sets of parameters α

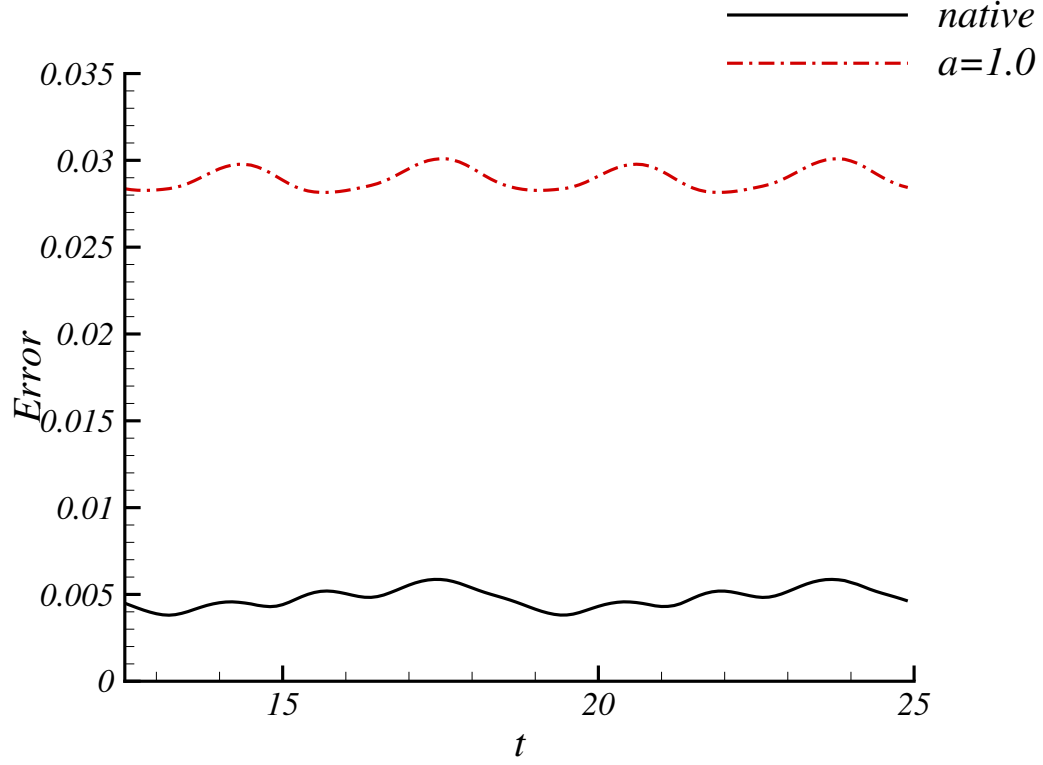


Figure 5.3: The error changing with time for both native ROM and normal $a = 1.0$ ROM for the prediction of the flow with the oscillation amplitude $a = 0.5$.

and β are employed to create the shift mode. Following the same routine as Noack's shift mode, our shift mode is defined as:

$$\mathbf{u}_{\Delta}^a := \mathbf{u}_0^{\alpha} - \mathbf{u}_0^{\beta}, \quad (5.7)$$

$$\mathbf{u}_{\Delta}^b := \mathbf{u}_{\Delta}^a - \sum_{i=1}^N \langle \mathbf{u}_{\Delta}^a, \phi_i \rangle \phi_i, \quad (5.8)$$

$$\mathbf{u}_{\Delta} := \frac{\mathbf{u}_{\Delta}^b}{\|\mathbf{u}_{\Delta}^b\|}. \quad (5.9)$$

5.2.1 The Application of Shift Mode: Flow Passing a Fixed Cylinder

First, the method has been tested by the same fixed cylinder case as Noack's shift mode⁵. For the two sets of parameters, $\mathbf{Re} = 100$ and $\mathbf{Re} = 150$ have been chosen to calculate the shift mode. After the routine in equation (5.9), the shift mode and two mean flows from both cases are plotted in figure 5.4.

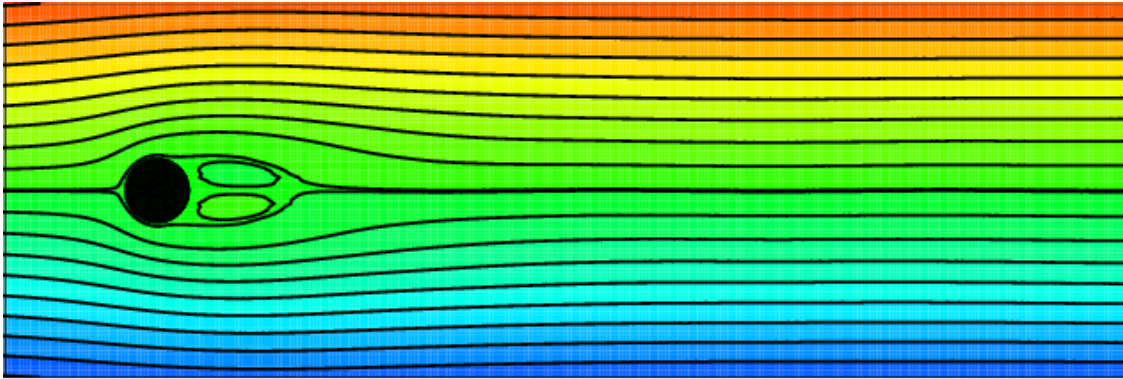
Before applying the shift mode on the ROM prediction, lift and drag have been calculated by the original ROM. From figure 5.5, it is clear that the original ROM cannot handle this parametric variation problem: the results become unstable and inaccurate after several periods of prediction, and the errors will keep growing because of instability and finally blow up the calculation.

However, with the shift mode (figure 5.6), the ROM performs much better than the original one. Although the details are still not very accurate, the amplitudes of both lift and drag become reasonable and the results become stable. This demonstrates that our shift mode based on the mean flows from two different databases with different sets of parameters can also work for the fixed boundary case.

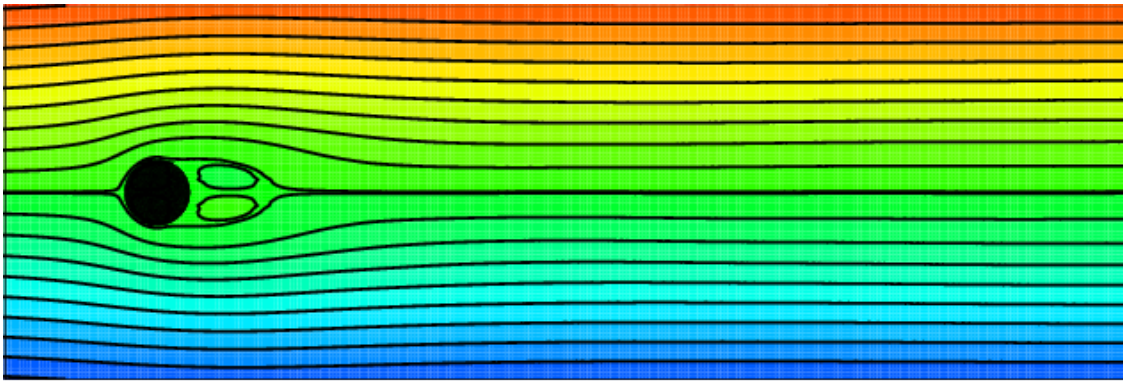
5.2.2 The Application of Shift Mode: Flow Passing an Oscillatory Sphere

For the three-dimensional flow passing an oscillatory sphere in chapter 5.1.2, mean flow $\mathbf{u}_0^{a=1.0}$ from DNS $a = 1.0$ (the amplitude which built the modes) and the mean flow $\mathbf{u}_0^{a=0.5}$ from DNS $a = 0.5$ (the amplitude which we want to predict) are employed, and the same routine in equation (5.9) is applied on these two mean flows to construct the shift mode.

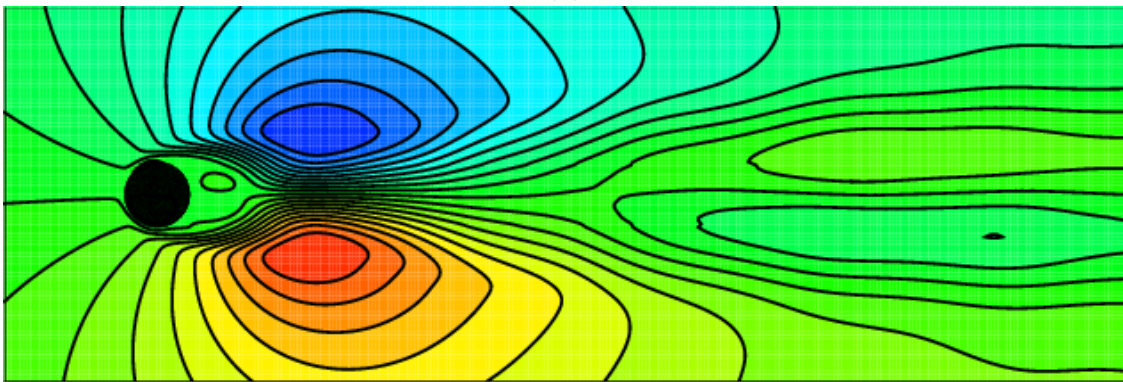
By introducing this shift mode, figure 5.7 also shows the error function defined to match the flow with an oscillation amplitude $a = 0.5$ to compare the performance of ROMs in their prediction of the flow velocity. Compared to figure 5.3, there is an extra ROM used in this comparison: the ROM at $a = 1.0$ but with a shift mode to predict the flow with $a = 0.5$.



(a)

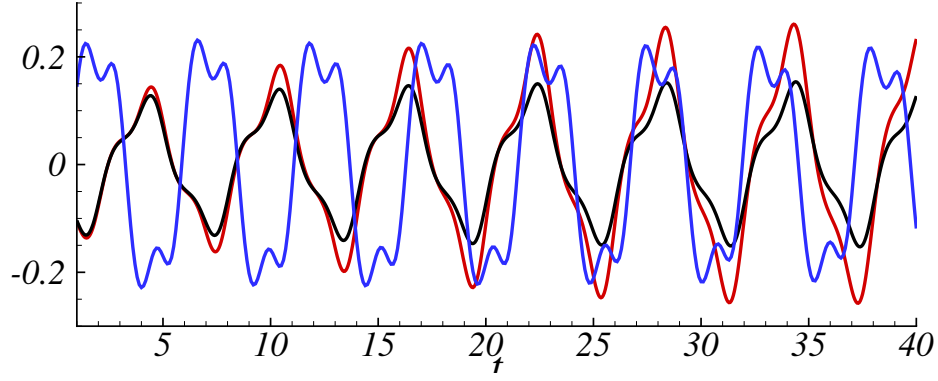


(b)

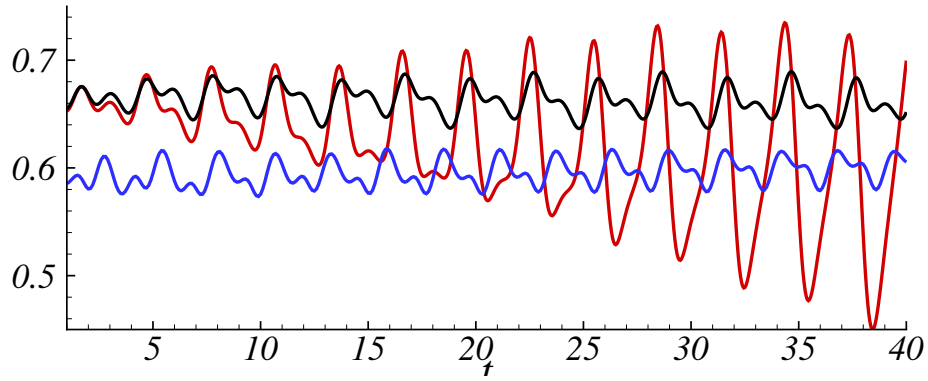


(c)

Figure 5.4: The vorticity field of: (a) mean flow of $\text{Re} = 100$, (b) mean flow of $\text{Re} = 150$, (c) the shift mode built by $\text{Re} = 100$ and $\text{Re} = 150$.

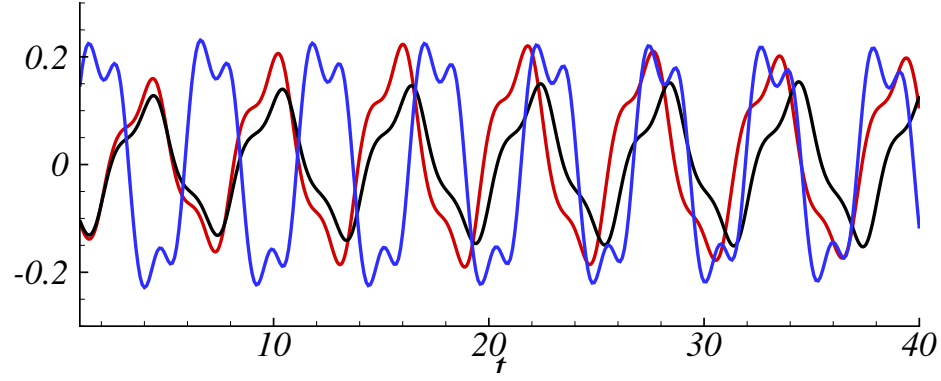


(a)

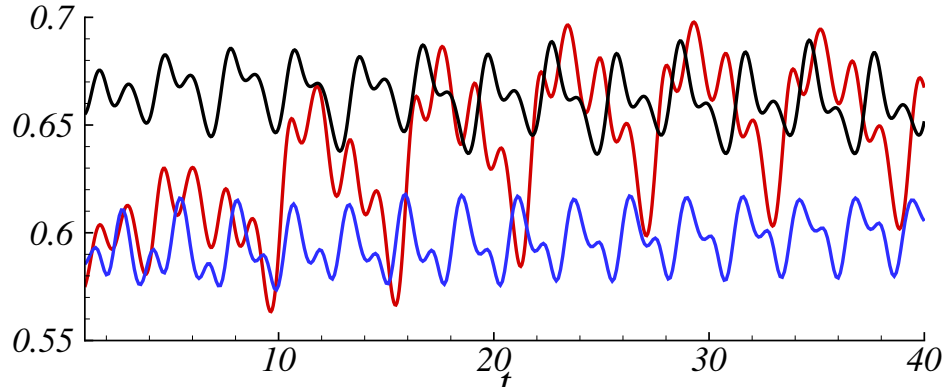


(b)

Figure 5.5: Comparison of (a) lift and (b) drag for the two-dimensional flow passing a fixed cylinder, computed by (—) ROM prediction for $\mathbf{Re} = 100$ based on the native $\mathbf{Re} = 100$ database, (—) ROM prediction for $\mathbf{Re} = 150$ based on the $\mathbf{Re} = 100$ database, and (—) ROM prediction for $\mathbf{Re} = 150$ based on the native $\mathbf{Re} = 150$ database.



(a)



(b)

Figure 5.6: Comparison of (a) lift and (b) drag for the two-dimensional flow passing a fixed cylinder, computed by (—) ROM prediction for $\mathbf{Re} = 100$ based on the native $\mathbf{Re} = 100$ database, (—) ROM prediction for $\mathbf{Re} = 150$ based on the $\mathbf{Re} = 100$ database and shift mode, and (—) ROM prediction for $\mathbf{Re} = 150$ based on the native $\mathbf{Re} = 150$ database.

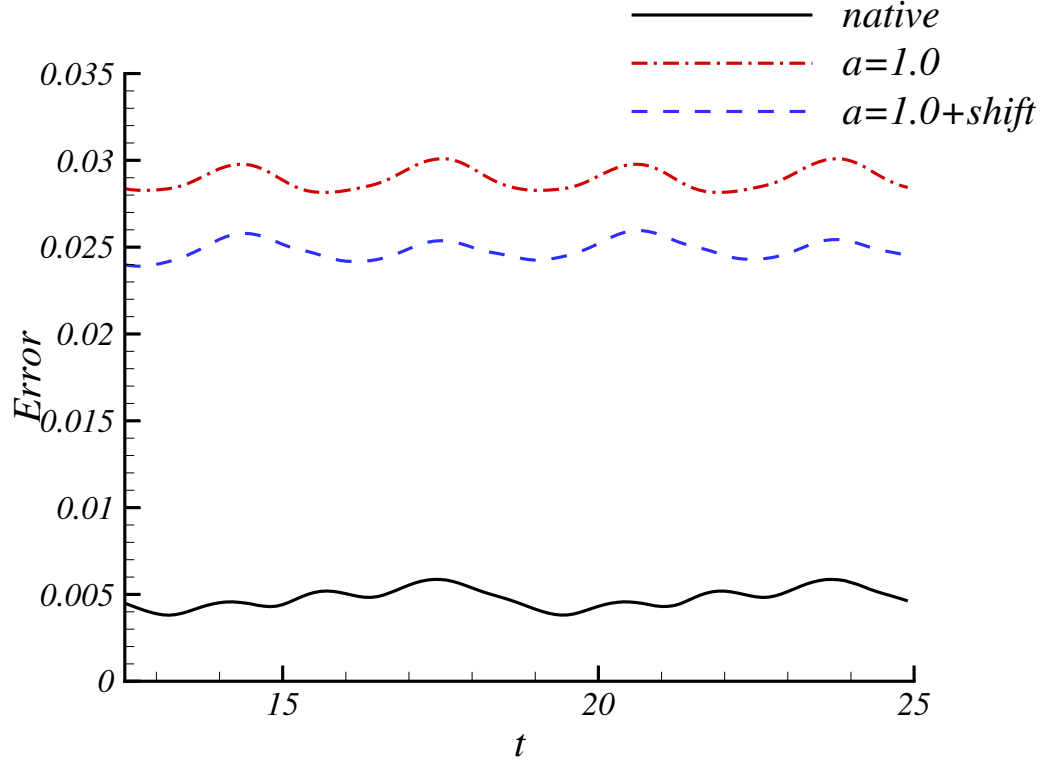


Figure 5.7: The error changing with time for native ROM, normal $a = 1.0$ ROM and $a = 1.0$ ROM with shift mode for the prediction of the flow with the oscillation amplitude $a = 0.5$.

Different from the powerful improvement for the adaptation of the Reynolds number in chapter 5.1.2, even at the same amplitude ($a = 0.5$) where the shift mode is created, the shift mode shows very limited help to improve the normal single-parameter ($a = 1.0$) ROM in the model adaptation for different amplitudes of the oscillations. More techniques are required to do the further improvement.

5.3 Combined Model

In this section, the combined model is employed to improve the model adaptation of the moving parametric variation. Different from the shift mode, the combined (hybrid/split/mixed) model has been considered as a more powerful tool for the model adaptation problem because it introduces more information from different databases with different sets of parameters. The idea of a combined model is to build the ROM based on the combined POD modes (or

other orthogonal basis) which are computed from all snapshots from different parameters. It was not only applied on the Reynolds number variation^{6;35}, but also for the variation of boundary condition³⁷ and shape³⁶. In the study, the combined model is built in the same way as Galletti et al. 2004. The snapshots from different sets of DNS results with different parameters are combined together directly, and eigenmodes are extracted from these combined snapshots.

5.3.1 The Application of Combined Model on the Global ROM with Forcing Term: Two Sets of Snapshots

For the above problem of an oscillatory sphere, two sets of snapshots for the motions with $a = 1.0$ and $a = 0.5$ are included to compute the combined modes, and each of them contain 125 snapshots for two whole limit cycles.

Figure 5.8 shows the differences, as well as similarities between the first four modes of the combined POD modes and the original POD modes at $a = 0.5$ and $a = 1.0$. Because of this combination, it is reasonable that more modes are required to capture enough percentage of total energy. Figure 5.9a plots the energy distribution for each mode, showing that the higher number of modes from the combined model have higher percentages of energy than both native models. The cumulated energy by increasing the number of modes in figure 5.9b is more clear. The two native models reach a certain percentage (e.g. 90%) of total energy much faster than the combined model.

Based on these combined modes, the combined ROM with forcing term has been applied to the same three-dimensional flow passing an oscillatory sphere case. Firstly, similar as the previous method, flow with $a = 0.5$ is predicted, and the phase portraits for time coefficients are plotted in figure 5.10. Although the results are not as good as the native ROM shown in chapters 2 and 3, the dynamic of the system is still predicted with acceptable accuracy.

By calculating the error defined in equation (5.6), the combined ROM has been compared to the native ROM, normal single-parameter ($a = 1.0$) ROM, and ROM with shift mode in figure 5.11.

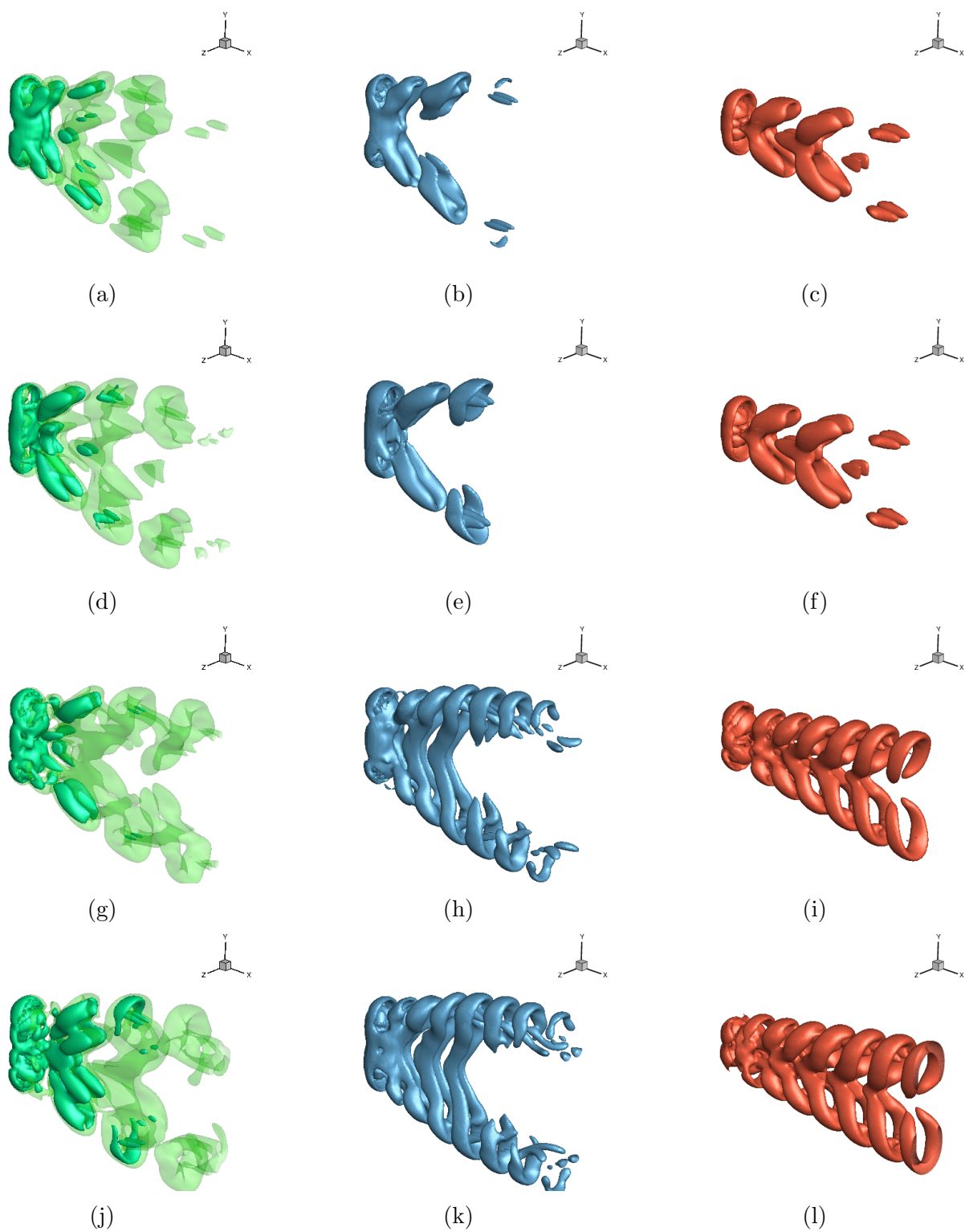


Figure 5.8: The iso-contour of Q-criterion for the first four modes from (left) the combined model, (center) $a = 1.0$ native model, and (right) $a = 0.5$ native model.

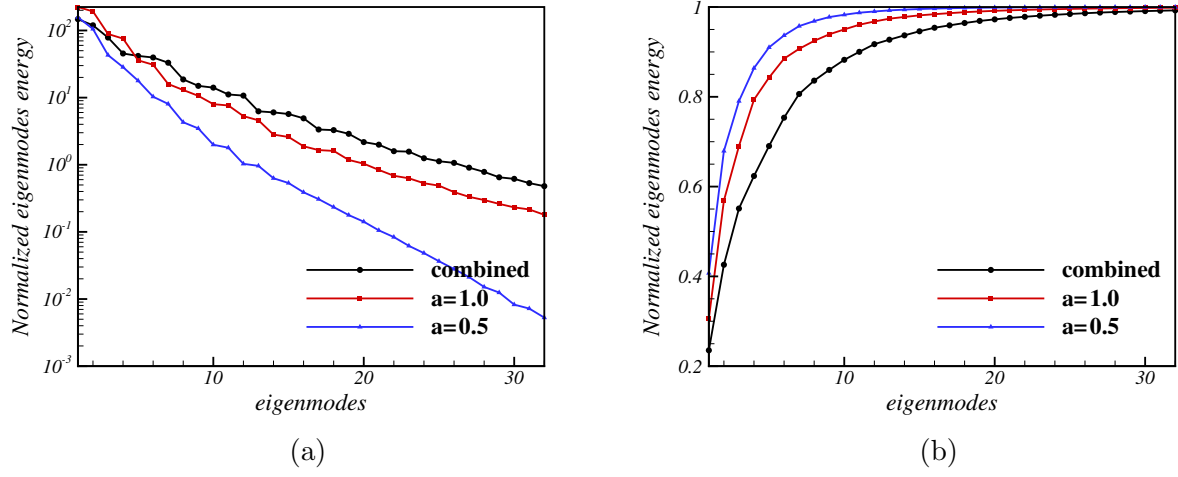


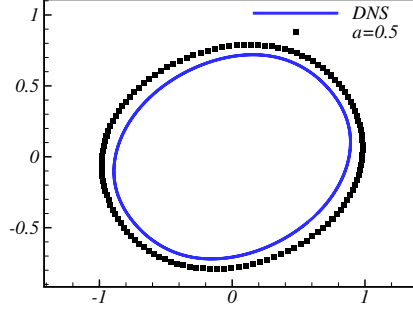
Figure 5.9: The velocity profile along the y-direction at $x = 2, z = 0$ when $t = nT + 3/4T$. (a) u_x and (b) u_y .

Because of introducing DNS snapshots from $a = 0.5$, the $a = 0.5$ can be considered as the “half native” amplitude. From figure 5.11, it is noted that, compared to the single-parameter ROM and the ROM with shift mode, the combined model shows significantly better prediction at $a = 0.5$. However, because of the effects of DNS snapshots from $a = 1.0$, the error of the combined model is still slightly larger than the native ROM. This is reasonable for this “half native” amplitude, and this error is still small enough for the prediction.

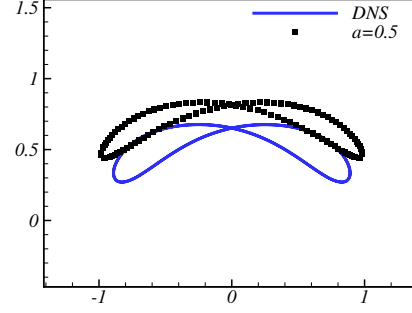
Then, the combined ROM has been used to predict the other “half native” amplitude, i.e. $a = 1.0$. Similarly, the phase portraits are plotted in figure 5.12, while the error is compared to the native ROM, normal single-parameter ($a = 0.5$) ROM, and ROM with shift mode in figure 5.13.

Compared to the prediction of $a = 0.5$, $a = 1.0$ ROM with shift mode performs much better when it predicts $a = 1.0$. It is easy to understand because it is based on the normal $a = 1.0$ ROM. And there is no doubt that our combined model performs more accurately than the normal ROM at both parameters.

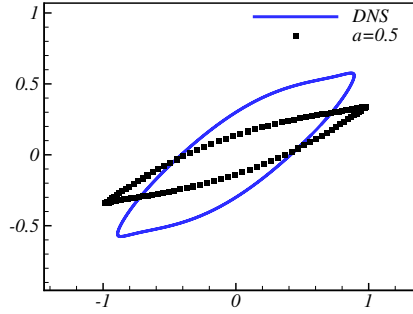
Compared to the prediction of “half native” amplitudes, the predictions of the unknown amplitudes are much more exciting and important. First, the amplitude $a = 0.75$, which is



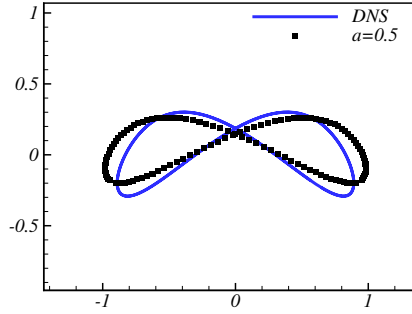
(a)



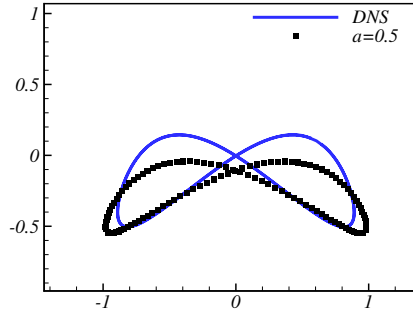
(b)



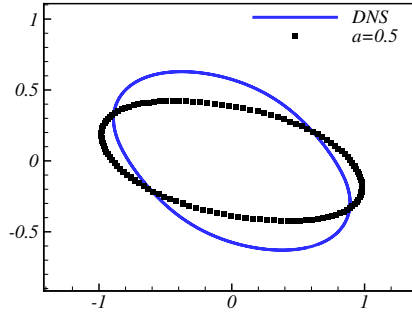
(c)



(d)



(e)



(f)

Figure 5.10: The phase portraits of: (a) (a_1, a_2) and (b) (a_1, a_3) , (c) (a_1, a_4) and (d) (a_1, a_5) , (e) (a_1, a_6) and (f) (a_1, a_7) for $a = 0.5$ where the results computed from the combined global ROM are benchmarked by the result from direct numerical simulation.

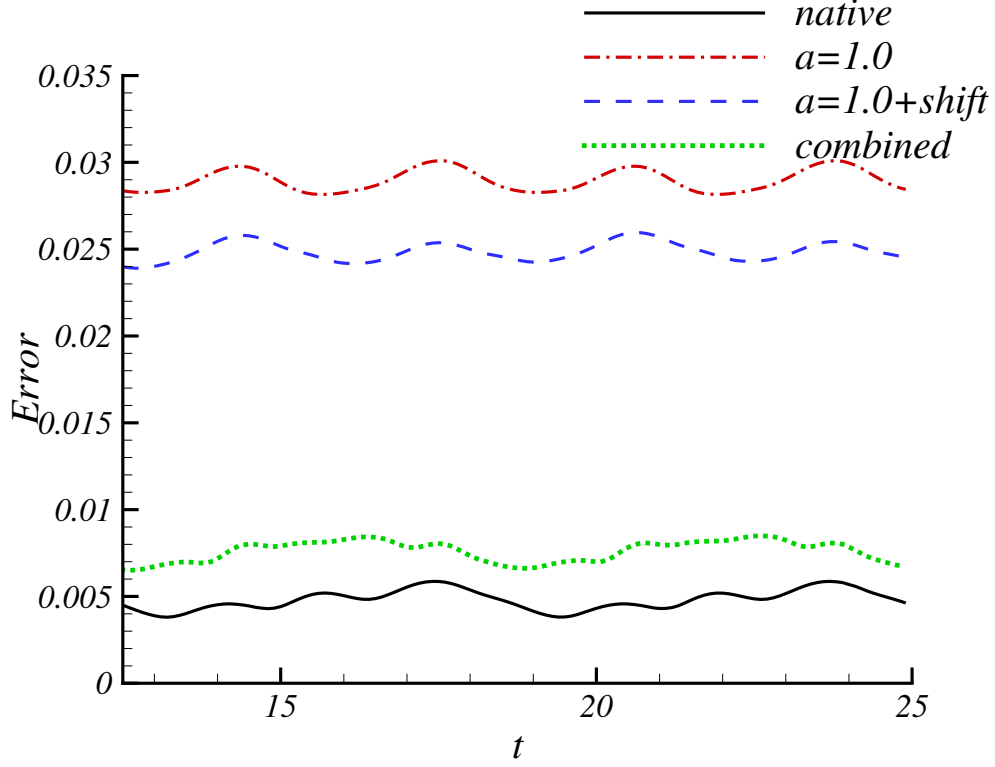


Figure 5.11: The error changing with time for native ROM, normal $a = 1.0$ ROM, $a = 1.0$ ROM with shift mode, and combined ROM for the prediction of the flow with the oscillation amplitude $a = 0.5$.

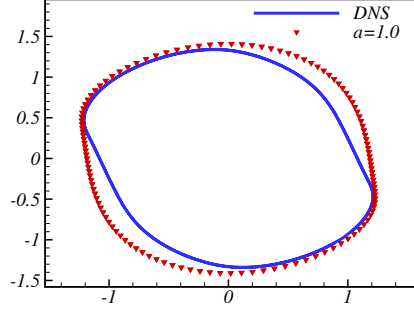
in the middle of the two “half native” amplitudes, has been predicted. The phase portraits are plotted in figure 5.14, while the error is compared to the native ROM, normal single-parameter ($a = 1.0$) ROM, and ROM with shift mode in figure 5.15.

Although the combined ROM cannot predict the flow at $a = 0.75$ as accurately as $a = 0.5$ and $a = 1.0$, the error has still been largely decreased.

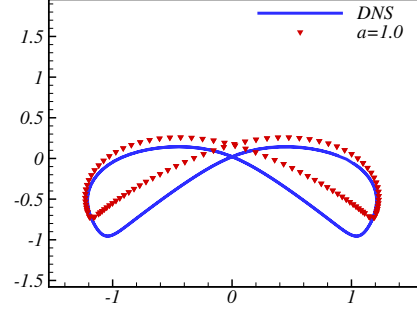
In order to better illustrate the error of different methods at different amplitudes, the mean error which is defined as equation 5.10 has been introduced.

$$error_m = \frac{1}{T} \int_0^T \frac{\|\mathbf{u}(t) - \mathbf{u}_d(t)\|}{V_\Omega} dt. \quad (5.10)$$

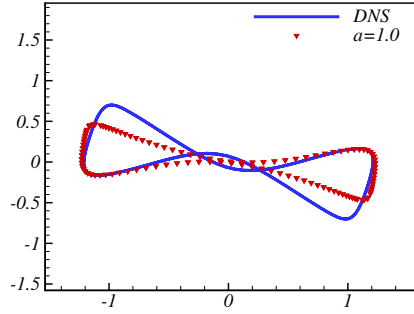
The accuracies of different ROMs for the parametric variation during $a \in [0.5, 1]$ are plotted in figure 5.16.



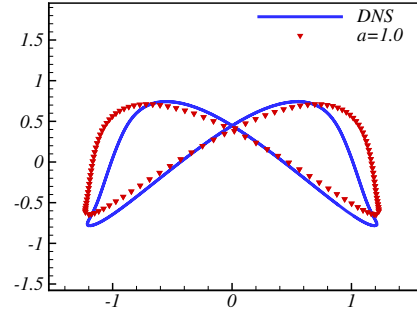
(a)



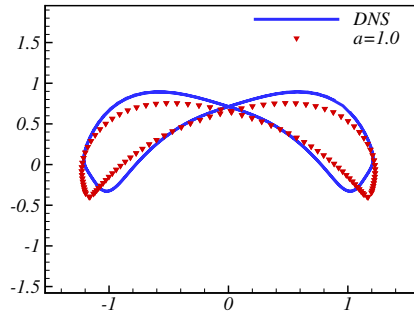
(b)



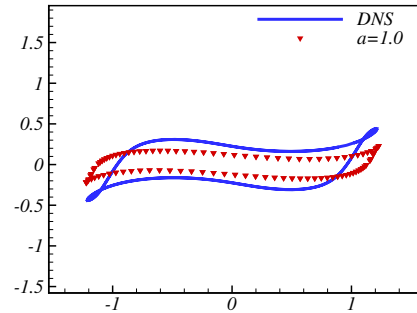
(c)



(d)



(e)



(f)

Figure 5.12: The phase portraits of: (a) (a_1, a_2) and (b) (a_1, a_3) , (c) (a_1, a_4) and (d) (a_1, a_5) , (e) (a_1, a_6) and (f) (a_1, a_7) for $a = 1.0$ where the results computed from the combined global ROM are benchmarked by the result from direct numerical simulation.

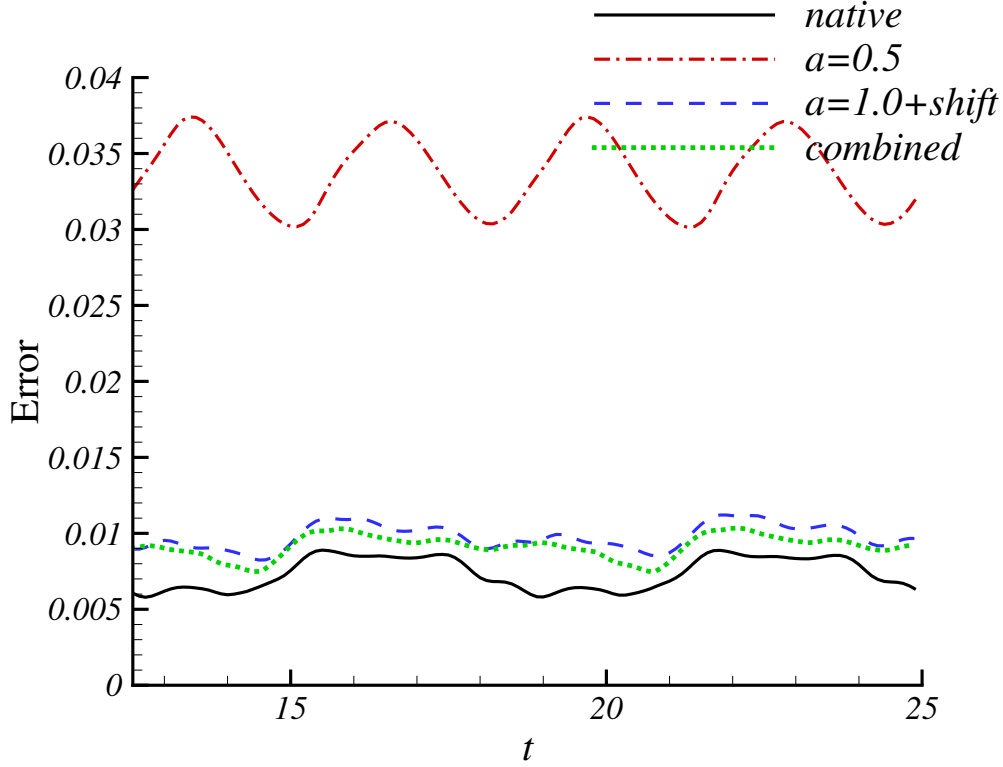
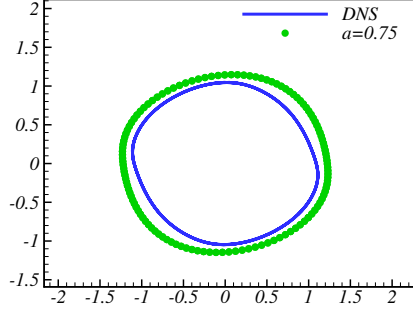
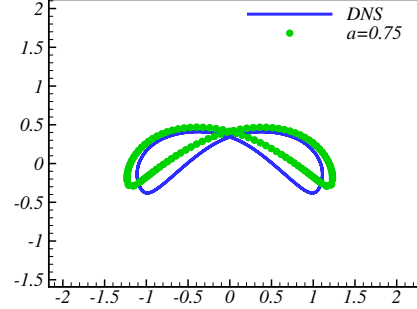


Figure 5.13: The error changing with time for native ROM, normal $a = 0.5$ ROM, $a = 1.0$ ROM with shift mode, and combined ROM for the prediction of the flow with the oscillation amplitude $a = 1.0$.

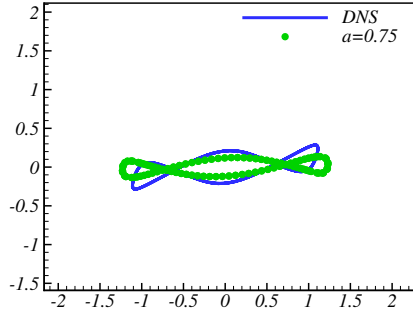
It is reasonable that the native ROM, which is based on the different DNS snapshots for different parameters, is the most accurate ROM. But as described, in practice, it is not possible to get native ROMs for all parameters. This requires computation or experiments for all the parameters and defeats the purpose of ROM. The two normal single-parameter ($a = 0.5$ and $a = 1.0$) ROMs with forcing term are only accurate to predict the flow extremely close to their DNS parameters. The error of them increase quickly as the parameter (amplitude) deviates further from their native value. For the ROM based on $a = 1.0$ and shift mode, because the shift mode is based on the difference between mean flow from $a = 1.0$ and $a = 0.5$, the accuracy at $a = 0.5$ is limited help to improve from the single-parameter $a = 1.0$ ROM. However, for the other parameters, the shift mode does not help much to reduce the derivation, even at the cost of less accuracy at the native parameter. Finally, the combined global ROM with forcing term performs both stably and accurately during the



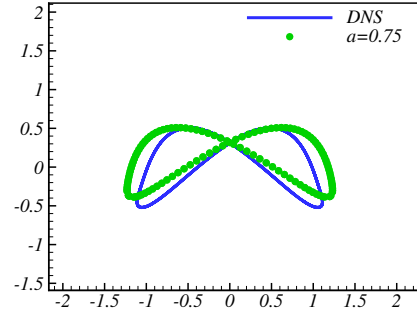
(a)



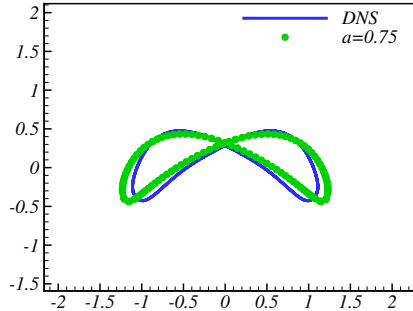
(b)



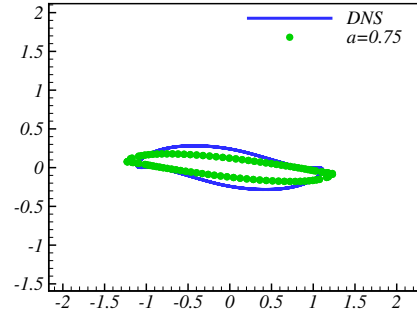
(c)



(d)



(e)



(f)

Figure 5.14: The phase portraits of: (a) (a_1, a_2) and (b) (a_1, a_3) , (c) (a_1, a_4) and (d) (a_1, a_5) , (e) (a_1, a_6) and (f) (a_1, a_7) for $a = 0.75$ where the results computed from the combined global ROM are benchmarked by the result from direct numerical simulation.

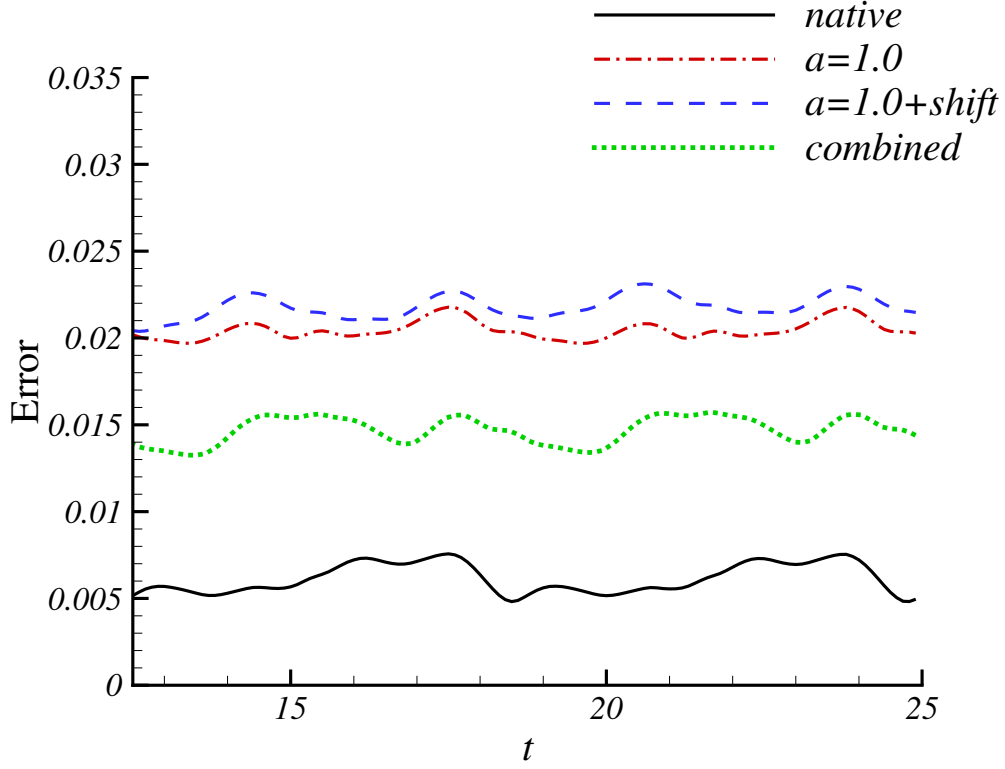


Figure 5.15: The error changing with time for native ROM, normal $a = 1.0$ ROM, $a = 1.0$ ROM with shift mode, and combined ROM for the prediction of the flow with the oscillation amplitude $a = 0.75$.

whole range of parametric changes. At the two ends $a = 0.5$ and $a = 1.0$, the errors of the combined model are only slightly larger than the native ROM. And during the variation of $a \in [0.5, 1]$, the combined model still keeps the error low in its adaptation to any amplitudes between its two native amplitudes ($a = 0.5$ and $a = 1.0$). Above all, for the parametric variation of the whole range $a \in [0.5, 1]$, the combined model, with only two data sets, shows an impressive capability in its prediction of other parameters for the model adaptation purpose to predict unknown states.

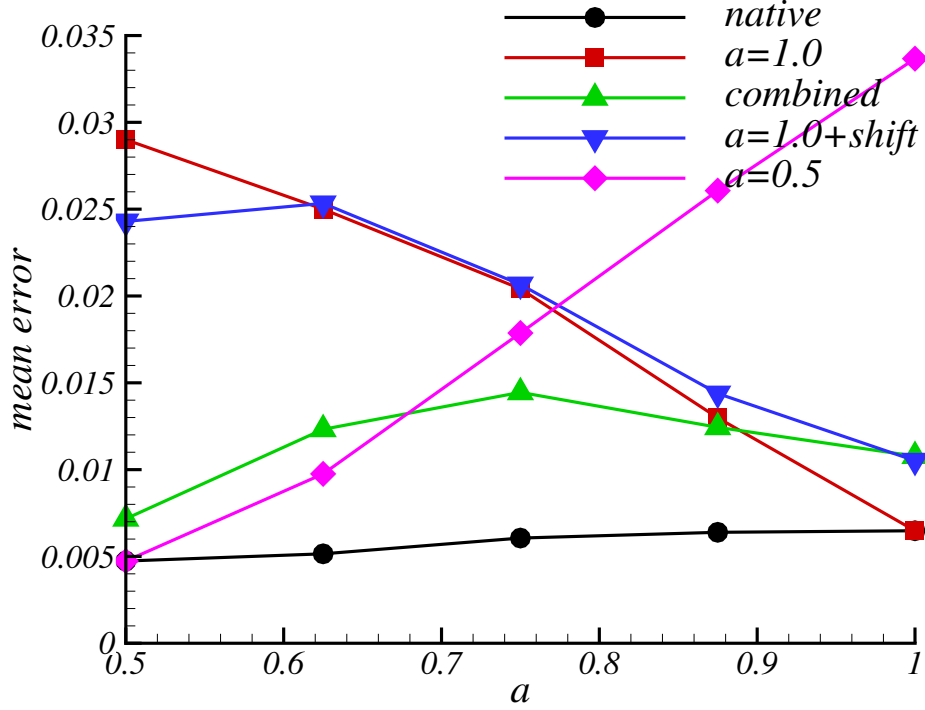


Figure 5.16: The mean error of different ROMs to predict the flow of other amplitudes $a \in [0.5, 1]$.

5.3.2 The Application of the Combined Model on the Global ROM with Forcing Term: Three Sets of Snapshots

The idea of the combined model is adding the information from the different states under different parameters; it has never been restricted by two different states from two different parameters. Thus, because the most inaccurate parameter for the above combined model is $a = 0.75$, it is natural to introduce the snapshots from $a = 0.75$ to build the improved model. Because the idea of it is the same as the combined model with two sets of snapshots, the error has been directly compared to the above method. First, the error at $a = 0.75$ has been compared in figure 5.17.

From the flow structures in figure 5.18, it is more clear that the flow rebuilt by the combined model with three sets of snapshots is more accurate than the above combined model with only the information from two different sets of snapshots.

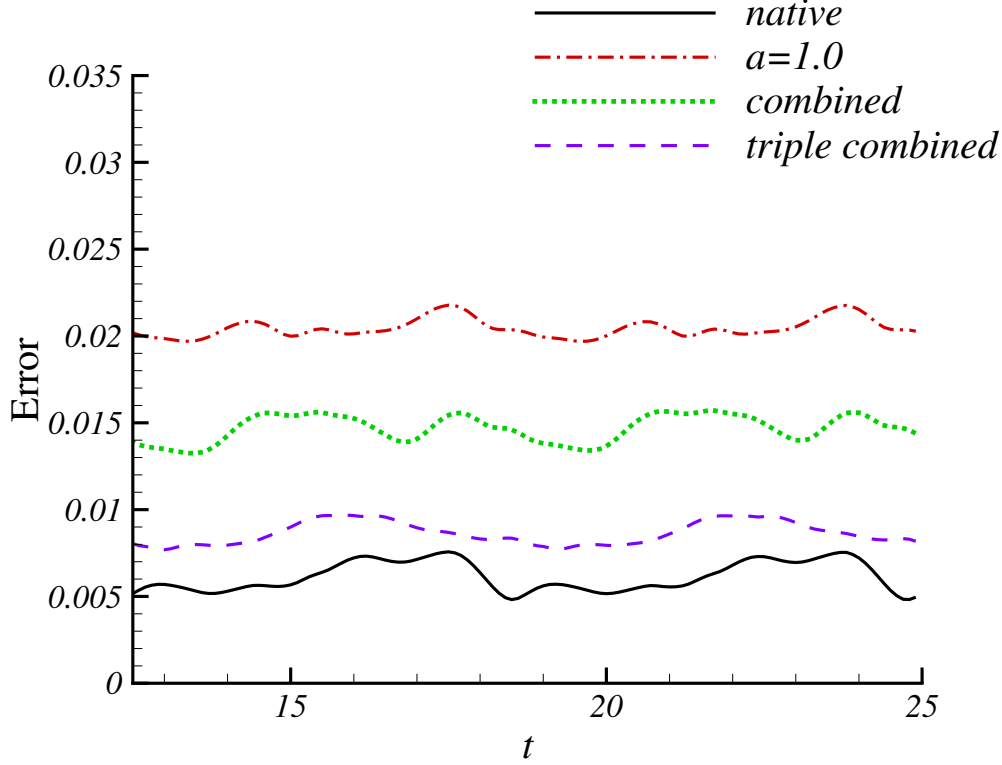


Figure 5.17: The error changing with time for native ROM, normal $a = 1.0$ ROM, the combined ROM with two sets of snapshots, and the combined ROM with three sets of snapshots for the prediction of the flow with the oscillation amplitude $a = 0.75$.

Of course, now $a = 0.75$ also becomes “native” parameters, the prediction of “unknown” parameters $a = 0.625$ and $a = 0.875$ are also plotted. The errors are compared to the native ROM, normal single-parameter ($a = 1.0$) ROM, and combined ROM with two sets of snapshots in figures 5.19 and 5.21, while the flow structures are plotted in figures 5.20 and 5.22.

Similarly, the mean errors of different ROMs for the parametric variation during $a \in [0.5, 1]$ are plotted in figure 5.23.

Compared to the combined ROM with two sets of snapshots at different parameters, the results of the combined ROM with three sets of snapshots has been significant improved during the entire range. At the original two ends of the two sets combined model, the errors of the three sets model are almost the same as the two sets model. The adding of the third set has not increased the error at these amplitudes. At the third “native” point $a = 0.75$, the

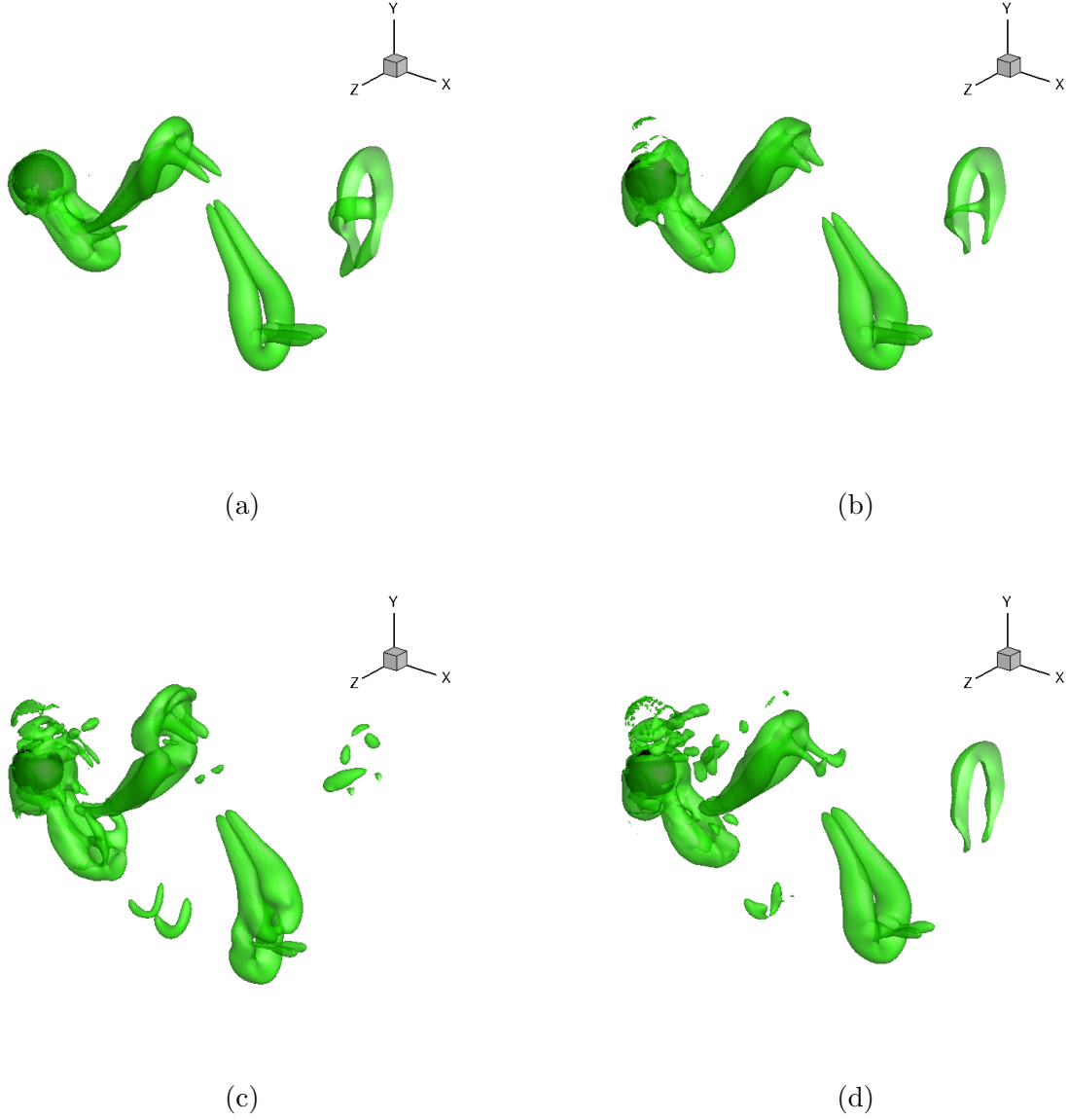


Figure 5.18: Comparison of flow field by iso-contour of Q-criterion at $a = 0.75$: (a) DNS flow field; (b) rebuilt result by native ROM prediction; (c) rebuilt result by the combined model with two sets of snapshots; (d) rebuilt result by the combined model with three sets of snapshots.

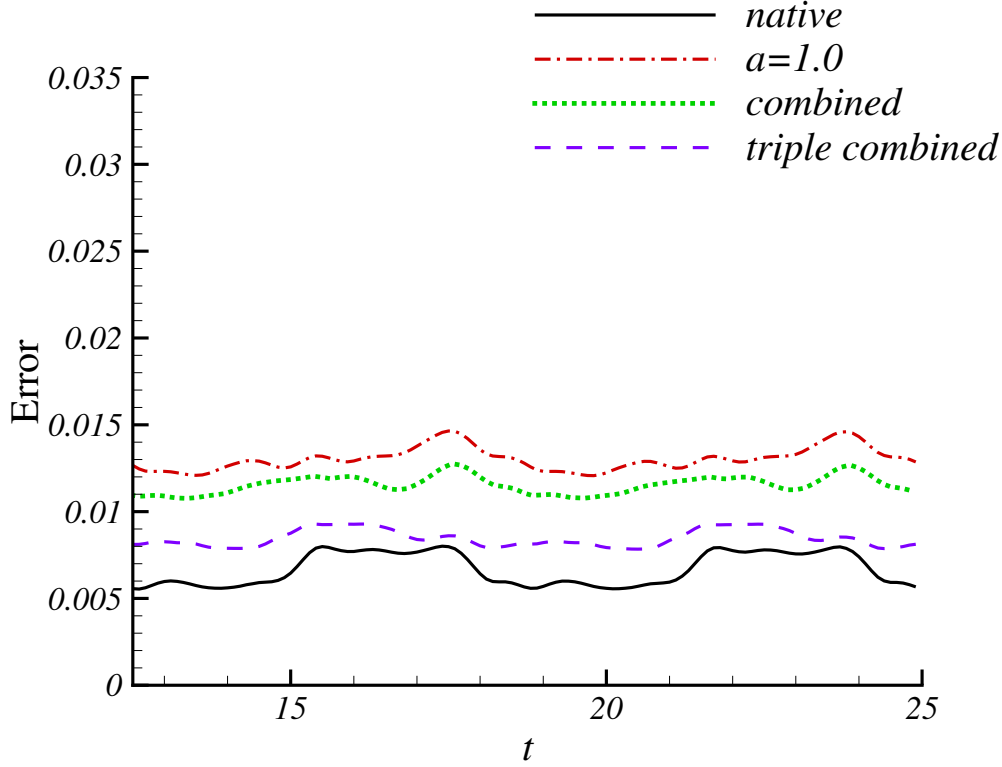


Figure 5.19: The error changing with time for native ROM, normal $a = 1.0$ ROM, the combined ROM with two sets of snapshots, and the combined ROM with three sets of snapshots for the prediction of the flow with the oscillation amplitude $a = 0.875$.

improvement of the error is significant. Even at the “unknown” amplitudes, the three sets model predicts the results almost as accurately as those “native” amplitudes. This means that by adding the third set of snapshots, our combined model has been improved and can keep the same high accuracy during the prediction of the entire range.

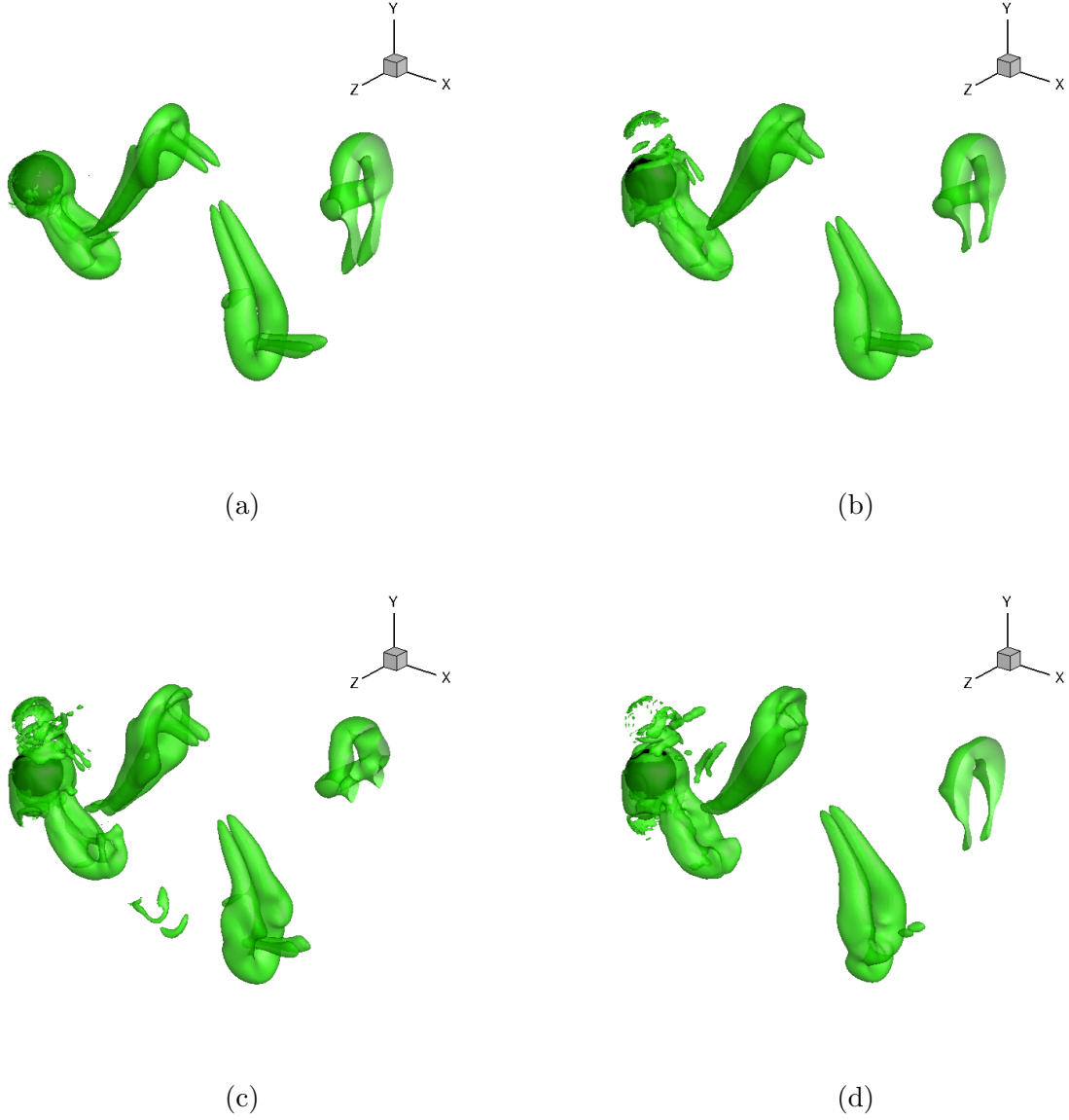


Figure 5.20: Comparison of flow field by iso-contour of Q -criterion at $a = 0.875$: (a) DNS flow field; (b) rebuilt result by native ROM prediction; (c) rebuilt result by the combined model with two sets of snapshots; (d) rebuilt result by the combined model with three sets of snapshots.

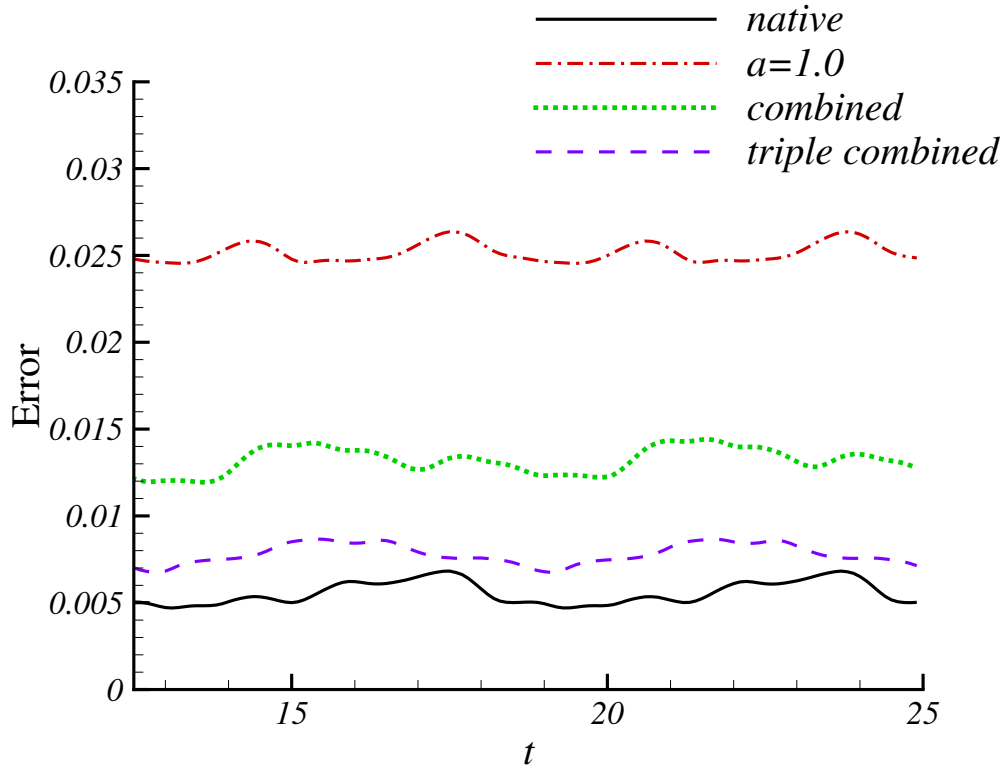


Figure 5.21: The error changing with time for native ROM, normal $a = 1.0$ ROM, the combined ROM with two sets of snapshots, and the combined ROM with three sets of snapshots for the prediction of the flow with the oscillation amplitude $a = 0.625$.

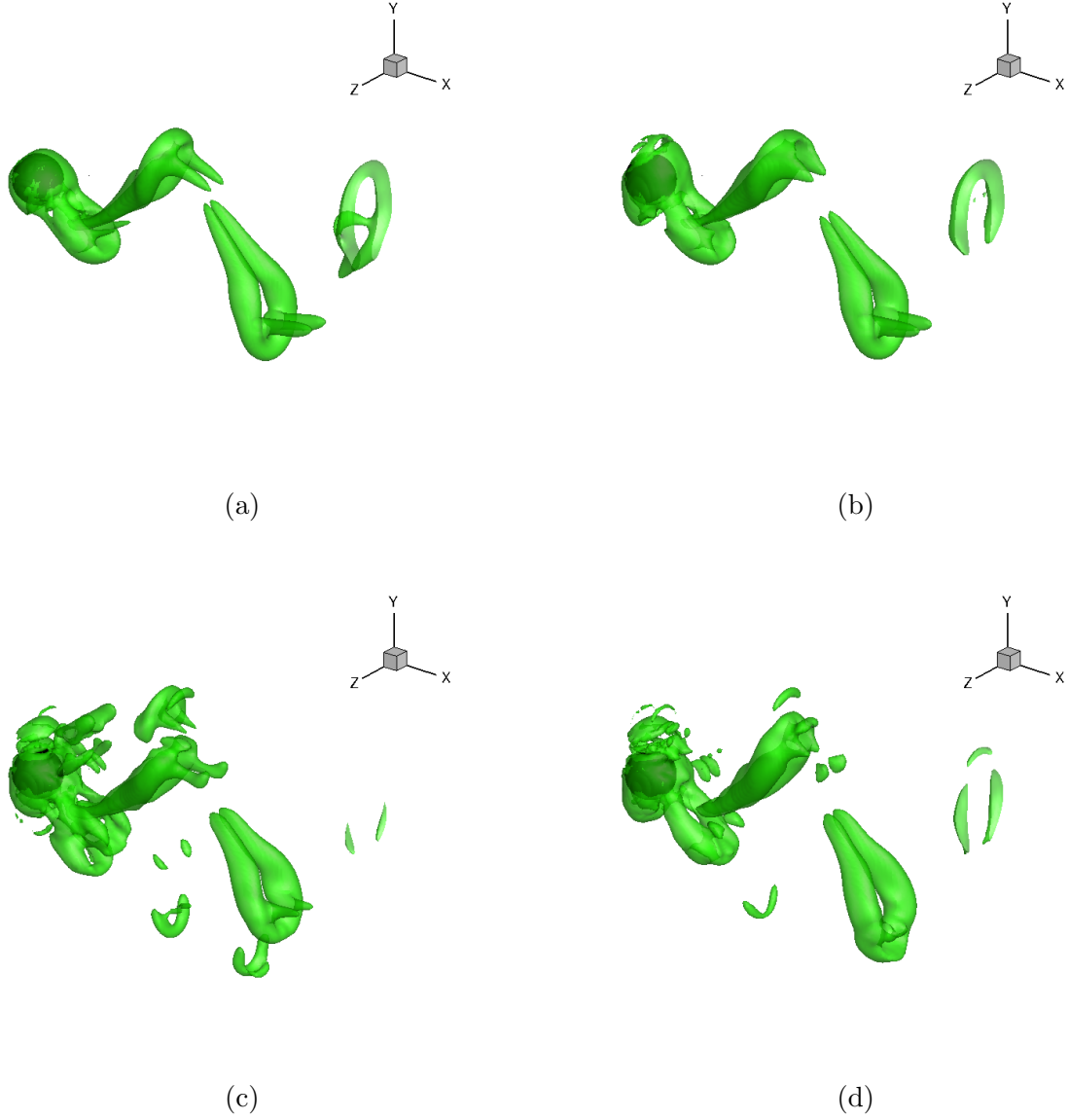


Figure 5.22: Comparison of flow field by iso-contour of Q -criterion at $a = 0.625$: (a) DNS flow field; (b) rebuilt result by native ROM prediction; (c) rebuilt result by the combined model with two sets of snapshots; (d) rebuilt result by the combined model with three sets of snapshots.

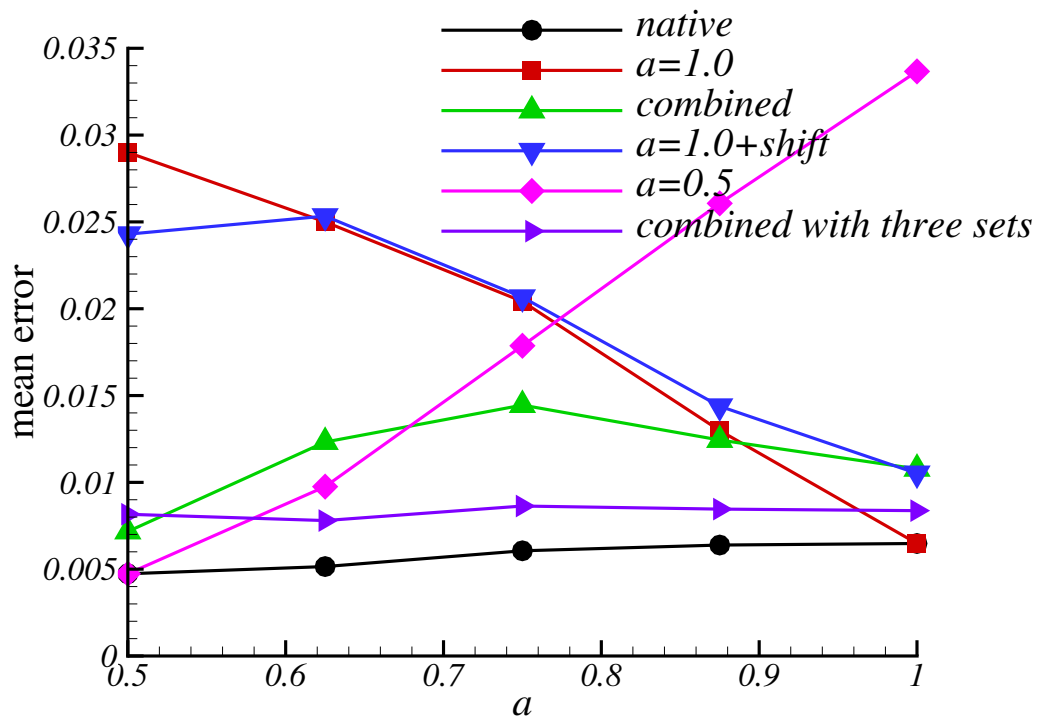


Figure 5.23: The mean error of different ROMs to predict the flow of other amplitudes $a \in [0.5, 1]$.

Chapter 6

Conclusion

In this study, a new global POD-Galerkin based ROM is developed by the author for dynamic problems with both fluid flow and moving solid boundaries/bodies. There are two approaches for the new ROM.

The first approach is the continuous approach, which defines the moving solid boundaries/bodies as a continuous time-varied domain. It is straight-forward, easy to apply, and can predict any kind of moving boundaries, no matter if it is rigid or flexible, flow-induced or prescribed, as long as the solid information is known. A series of flows with moving boundaries have been predicted by this method, including both two-dimensional and three-dimensional, both translational and rotational, and both rigid and flexible cases. The predictions match the simulation results accurately. The model in this dissertation with forcing preserves the dynamics and keeps its capability of capturing most energy of flow, while the traditional ROM without forcing term cannot be stable, and the flow structures will eventually damp out. However, this continuous approach is relatively time-consuming and hard to be pre-calculated, which undermines the benefit of using ROM.

The second approach is the decomposed approach, which is based on the continuous approach. By defining the moving solid boundaries/bodies as a characteristic function on the same combined fluid-solid domain, the POD-Galerkin projection method has also been applied on the moving solid boundaries/bodies. Thus, by decoupling time-dependent vari-

ables and fixed solid modes from time-varied solid domain, those time-varied parameters in the forcing terms of the continuous approach become constant and pre-calculated. This approach has been applied on both two-dimensional and three-dimensional, both translational and rotational cases with moving boundaries. Currently, only rigid body motions have been considered in this study, however, the decomposed approach still has the potential to predict prescribed flexible body motions. Same as the continuous approach, the decomposed approach preserves the dynamics and keeps its capability of capturing most energy of flow. The fluid structures and force results show that the forcing terms perform powerfully to maintain the dynamics of the system for large solid motions, especially the prescribed velocity terms which present as energy input and make the system stable, and allow the modeling of the prescribed solid motions. However, the decomposed approach is not suitable for some extremely complex flexible motions.

By analyzing the computational time cost by terms, the decomposed approach is demonstrated not only pre-calculated, but also much faster than the continuous approach. By combining the results of computational time cost and the contributions of different terms, the most expansive terms in both approaches are proven to be negligible, which can reduce the computational time cost for each approach significantly while the accuracy is still preserved.

This global ROM has also been employed to predict moving parametric variation problems. It can decouple the moving parameters from the fluid modes and make it explicit and applicable for parametric variation in the low-dimensional model, especially for the decomposed approach, which allows that the parametric variation does not affect any low-dimensional coefficients. Thus, based on the decomposed approach, the model in this dissertation is also pre-calculated for parametric variation. Different from other empirical terms which have been introduced by some researchers, the forcing term in this dissertation can represent not only the effects of moving boundaries, but also the moving parametric variation physically, accurately, and universally for all kinds of motions, especially for the rigid body moving problems. Based on this ROM, shift mode and combined models have been employed to improve stability and accuracy in model adaptation. Although the shift mode

has improved results for the Reynolds number variation for the fixed boundary problem, it fails to make improvements for the motion parameter variations. On the other hand, the combined model shows an impressive capability to significantly improve model adaptation to predict unknown states. By combining the information from two sets of snapshots at different parameters, the combined model, based on the solid-decomposed global ROM with forcing term, can predict the variation of moving parameters in a certain range physically and reasonable accurately, without introducing any extra computational time cost. By combining one more set of snapshots, the results can be further improved and become more accurate during the whole range of parametric variation.

Bibliography

- [1] David Amsallem and Charbel Farhat. Interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA Journal*, 46(7):1803, 2008.
- [2] Charbel Farhat and David Amsallem. Recent advances in reduced-order modeling and application to nonlinear computational aeroelasticity. In *46th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV*, 2008.
- [3] G. Berkooz, P. Holmes, and J. L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Ann. Rev. Fluid Mech.*, 25:539–575, 1993.
- [4] P. Holmes, J. L. Lumley, and G. Berkooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge University Press, Cambridge, 1996.
- [5] Bernd R. Noack, Konstantin Afanasiev, Marek Morzyński, Gilead Tadmor, and Frank Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics*, 497(1):335–363, 2003.
- [6] Xia Ma and George Em Karniadakis. A low-dimensional model for simulating three-dimensional cylinder flow. *Journal of Fluid Mechanics*, 458:181–190, 2002.
- [7] Marcelo Buffoni, Simone Camarri, Angelo Iollo, Maria Vittoria Salvetti, et al. Low-dimensional modelling of a confined three-dimensional wake flow. *Journal of Fluid Mechanics*, 569:141–150, 2006.
- [8] Thuan Lieu and Charbel Farhat. Adaptation of aeroelastic reduced-order models and application to an f-16 configuration. *AIAA Journal*, 45(6):1244, 2007.
- [9] Stefan G. Siegel, Jürgen Seidel, Casey Fagley, D.M. Luchtenburg, Kelly Cohen, and Thomas McLaughlin. Low-dimensional modelling of a transient cylinder wake using

- double proper orthogonal decomposition. *Journal of Fluid Mechanics*, 610(1):1–42, 2008.
- [10] Mingjun Wei and Clarence W. Rowley. Low-dimensional models of a temporally evolving free shear layer. *J. Fluid Mech.*, 618:113–134, 2009.
 - [11] Clarence W. Rowley and Jerrold E. Marsden. Reconstruction equations and the Karhunen-Loève expansion for systems with symmetry. *Phys. D*, 142:1–19, 2000.
 - [12] C. W. Rowley, T. Colonius, and R. M. Murray. Model reduction for compressible flow using POD and Galerkin projection. *Phys. D*, 189(1–2):115–129, February 2004.
 - [13] Bernd R. Noack, Michael Schlegel, Marek Morzyński, and Gilead Tadmor. System reduction strategy for galerkin models of fluid flows. *International Journal for Numerical Methods in Fluids*, 63(2):231–248, 2010.
 - [14] Mingjun Wei, Bashar R. Qawasmeh, Matthew Barone, Bart G. van Bloemen Waanders, and Lin Zhou. Low-dimensional model of spatial shear layers. *Physics of Fluids*, 24(1):014108, 2012. doi: 10.1063/1.3678016. URL <http://link.aip.org/link/?PHF/24/014108/1>.
 - [15] Ada Cammilleri, Florimond Guéniat, Johan Carlier, L. Pastur, Etienne Memin, François Lusseyran, and Guillermo Artana. Pod-spectral decomposition for fluid flow analysis and model reduction. *Theoretical and Computational Fluid Dynamics*, pages 1–29, 2013.
 - [16] Gilead Tadmor, Oliver Lehmann, Bernd R. Noack, and Marek Morzyński. Galerkin models enhancements for flow control. In *Reduced-Order Modelling for Flow Control*, pages 151–252. Springer, New Yorker, NY., 2011.
 - [17] Bernd R. Noack, Gilead Tadmor, and Marek Morzynski. Actuation models and dissipative control in empirical galerkin models of fluid flows. In *American Control Conference, 2004. Proceedings of the 2004*, volume 6, pages 5722–5727. IEEE, 2004.

- [18] Gilead Tadmor and Bernd R. Noack. Dynamic estimation for reduced galerkin models of fluid flows. In *American Control Conference, 2004. Proceedings of the 2004*, volume 1, pages 746–751. IEEE, 2004.
- [19] J.S.R. Anttonen, P.I. King, and P.S. Beran. Pod-based reduced-order models with deforming grids. *Mathematical and Computer Modelling*, 38(1):41–62, 2003.
- [20] J.S.R. Anttonen, P.I. King, and P.S. Beran. Applications of multi-pod to a pitching and plunging airfoil. *Mathematical and Computer Modelling*, 42(3):245–259, 2005.
- [21] Zhengkun Feng and Azzeddine Soulaïmani. Nonlinear aeroelasticity modeling using a reduced order model based on proper orthogonal decomposition. In *ASME Pressure Vessels and Piping Division Conference*, number 26006, 2007.
- [22] Erwan Liberge, Mustapha Benaouicha, and Aziz Hamdouni. Proper orthogonal decomposition investigation in fluid structure interaction. *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*, 16(3-4):401–418, 2007.
- [23] Erwan Liberge and Aziz Hamdouni. Reduced order modelling method via proper orthogonal decomposition (pod) for flow around an oscillating cylinder. *Journal of Fluids and Structures*, 26(2):292–311, 2010.
- [24] Haotian Gao and Mingjun Wei. Global model reduction for flows with moving boundary. *AIAA Paper 2014-0222*, 2014.
- [25] Jason Tran, Jayant Sirohi, Haotian Gao, and Mingjun Wei. Reduced-order modeling of loads and deformation of a flexible flapping wing. *Target*, 20:40, 2015.
- [26] Haotian Gao and Mingjun Wei. Domain decomposition in pod-galerkin projection for flows with moving boundary. In *54th AIAA Aerospace Sciences Meeting*, page 1102, 2016.
- [27] Jason Tran, Haotian Gao, Jayant Sirohi, and Mingjun Wei. Reduced-order methodology

- for prediction of loads generated by a flexible flapping wing. *International Journal of Micro Air Vehicles*, page 1756829317708318, 2017.
- [28] C. S. Peskin. Flow patterns around heart valves: a numerical method. *J. Comput. Phys.*, 10(2):252–271, 1972.
- [29] C. S. Peskin. Numerical analysis of blood flow in the heart. *J. Comput. Phys.*, 25(3):220–252, 1977.
- [30] J. Mohd-Yusof. Combined immersed-boundary/B-spline methods for simulations of flow in complex geometries. Center for Turbulence Research, Annual Research Briefs – 1997, 1997.
- [31] T. Yang, M. Wei, and H. Zhao. Numerical study of flexible flapping wing propulsion. 48th AIAA Aerospace Sciences Meeting and Exhibit, Orlando, FL, AIAA paper 2010-0553, 2010.
- [32] M. Xu, M. Wei, T. Yang, Y. S. Lee, and T. D. Burton. Nonlinear structural response in flexible flapping wings with different density ratio. 49th AIAA Aerospace Sciences Meeting and Exhibit, Orlando, FL, AIAA paper 2011-376, 2011.
- [33] A.E. Deane, I.G. Kevrekidis, G. Em Karniadakis, and S.A. Orszag. Low-dimensional models for complex geometry flows: Application to grooved channels and circular cylinders. *Physics of Fluids A: Fluid Dynamics*, 3(10):2337–2354, 1991.
- [34] Alexander Hay, Jeffrey T. Borggaard, and Dominique Pelletier. Local improvements to reduced-order models using sensitivity analysis of the proper orthogonal decomposition. *Journal of Fluid Mechanics*, 629:41–72, 2009.
- [35] B. Galletti, C.H. Bruneau, Luca Zannetti, and Angelo Iollo. Low-order modelling of laminar flow regimes past a confined square cylinder. *Journal of Fluid Mechanics*, 503:161–170, 2004.

- [36] Rémi Bourguet, Marianna Braza, and Alain Dervieux. Reduced-order modeling of transonic flows around an airfoil submitted to small deformations. *Journal of Computational Physics*, 230(1):159–184, 2011.
- [37] Russell Chris Camphouse, J.H. Myatt, Ryan Schmit, Mark Glauser, Julie Ausseur, Marlyn Andino, and Ryan Wallace. A snapshot decomposition method for reduced order modeling and boundary feedback control. *AIAA paper*, 4195:2008, 2008.
- [38] Christophe Audouze, Florian De Vuyst, and Prasanth B. Nair. Nonintrusive reduced-order modeling of parametrized time-dependent partial differential equations. *Numerical Methods for Partial Differential Equations*, 29(5):1587–1628, 2013.
- [39] Michel Bergmann, C.-H. Bruneau, and Angelo Iollo. Enablers for robust pod models. *Journal of Computational Physics*, 228(2):516–538, 2009.
- [40] W.R. Graham, J. Peraire, and K.Y. Tang. Optimal control of vortex shedding using low-order models. part i: Open-loop model development. *International Journal for Numerical Methods in Engineering*, 44(7):945–972, 1999.
- [41] W.R. Graham, J. Peraire, and K.Y. Tang. Optimal control of vortex shedding using low-order models. part ii: model-based control. *International Journal for Numerical Methods in Engineering*, 44(7):973–990, 1999.
- [42] Michel Bergmann, Laurent Cordier, and Jean-Pierre Brancher. Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model. *Physics of Fluids*, 17(9):097101–097101, 2005.
- [43] L. Sirovich. Chaotic dynamics of coherent structures. Parts I–III. *Quarterly of Applied Math.*, XLV(3):561–582, 1987.
- [44] H. Zhao, J. B. Freund, and R. D. Moser. A fixed-mesh method for incompressible flow-structure systems with finite solid deformations. *J. Comput. Phys.*, 227:3114–3140, 2008.

- [45] Min Xu and Mingjun Wei. Using adjoint-based optimization to study kinematics and deformation of flapping wings. *Journal of Fluid Mechanics*, 799:56–99, 2016.
- [46] Min Xu, Mingjun Wei, Tao Yang, and Young S. Lee. An embedded boundary approach for the simulation of a flexible flapping wing at different density ratio. *European Journal of Mechanics-B/Fluids*, 55:146–156, 2016.
- [47] D. L. Brown, R. Cortez, and M. L. Minion. Accurate projection methods for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 168:464–499, 2001.
- [48] C.H.K. Williamson. Vortex dynamics in the cylinder wake. *Annual Review of Fluid Mechanics*, 28(1):477–539, 1996.
- [49] Tao Yang and Mingjun Wei. Effects of wing flexibility on aerodynamic performance in hovering flight. In *APS Meeting Abstracts*, 2012.
- [50] Bret W. Tobalske, Douglas R. Warrick, Christopher J. Clark, Donald R. Powers, Tyson L. Hedrick, Gabriel A. Hyder, and Andrew A. Biewener. Three-dimensional kinematics of hummingbird flight. *Journal of Experimental Biology*, 210(13):2368–2382, 2007.

Appendix A

Decoupling of the Two-dimensional Pitching and Plunging Plate

For the two-dimensional pitching and plunging plate, the motion of the solid can be described as:

$$\mathbf{x} = a(t)\mathbf{I} + \mathbf{R}(\mathbf{X} + \sum_{k=1}^n b_k(t)\psi_k(X)\mathbf{I}), \quad (\text{A.1})$$

where $\mathbf{x} = (x, y)^T$ and $\mathbf{X} = (X, Y)^T$ are the locations of solid points in Eulerian frame and in the undeformed Lagrangian coordinate respectively, and $\mathbf{I} = (0, 1)^T$,

$$\mathbf{R} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}.$$

So the motion in the undeformed Lagrangian coordinate can be written as:

$$\mathbf{X} = \mathbf{R}^{-1}\mathbf{x} - a(t)\mathbf{R}^{-1}\mathbf{I} - \sum_{k=1}^n b_k(t)\psi_k(X)\mathbf{I}. \quad (\text{A.2})$$

From equation (A.1), by applying the time derivative, it can be written as:

$$\mathbf{V} = \dot{\mathbf{x}} = \dot{a}(t)\mathbf{I} + \dot{\mathbf{R}}\mathbf{X} + \dot{\mathbf{R}} \sum_{k=1}^n b_k(t)\psi_k(\mathbf{X})\mathbf{I} + \mathbf{R} \sum_{k=1}^n \dot{b}_k(t)\psi_k(X)\mathbf{I}. \quad (\text{A.3})$$

Put equation (A.2) into equation (A.3), and simplify it:

$$\mathbf{V} = \dot{a}(t)\mathbf{I} + \dot{\mathbf{R}}\mathbf{R}^{-1}\mathbf{x} - a(t)\dot{\mathbf{R}}\mathbf{R}^{-1}\mathbf{I} + \mathbf{R} \sum_{k=1}^n \dot{b}_k(t)\psi_k(X)\mathbf{I}. \quad (\text{A.4})$$

Put the value of $\mathbf{R}$ into the above equation,

$$\mathbf{V} = \dot{a}(t) \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \dot{\theta} \begin{bmatrix} -y \\ x \end{bmatrix} + a(t)\dot{\theta} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \sum_{k=1}^n \dot{b}_k(t) \cos \theta \begin{bmatrix} 0 \\ \psi_k(\mathbf{X}) \end{bmatrix} - \sum_{k=1}^n \dot{b}_k(t) \sin \theta \begin{bmatrix} \psi_k(\mathbf{X}) \\ 0 \end{bmatrix}. \quad (\text{A.5})$$

From equation (A.2), X can be written as:

$$X = \cos \theta x + \sin \theta y - a \sin \theta. \quad (\text{A.6})$$

The version of ψ can be any kind of functions, if it is too complex like $\sinh(X)$ and $\cosh(X)$ which will become $\sinh(x, t)$ and $\cosh(x, t)$, it cannot be separated. However, if ψ are polynomials, it can be written as:

$$\psi(X) = \sum_i c_i(t) X^i, \quad (\text{A.7})$$

and if we put equation (A.7) into it:

$$\psi(x, t) = \sum_m \sum_n p_m(t) q_n(t) x^m y^n, \quad (\text{A.8})$$

and it can be separated now.