

IMPLICIT MONOLITHIC PARALLEL SOLUTION ALGORITHM
FOR SEISMIC ANALYSIS OF DAM-RESERVOIR SYSTEMS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SEMİH ÖZMEN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
CIVIL ENGINEERING

FEBRUARY 2016

Approval of the thesis:

**IMPLICIT MONOLITHIC PARALLEL SOLUTION ALGORITHM
FOR SEISMIC ANALYSIS OF DAM-RESERVOIR SYSTEMS**

submitted by **SEMİH ÖZMEN** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Civil Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İsmail Özgür Yaman
Head of Department, **Civil Engineering**

Assoc. Prof. Dr. Özgür Kurç
Supervisor, **Civil Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Barış Binici
Civil Engineering Dept., METU

Assoc. Prof. Dr. Özgür Kurç
Civil Engineering Dept., METU

Assoc. Prof. Dr. M.Tolga Yılmaz
Engineering Sciences Dept., METU

Assoc. Prof. Dr. Ahmet Hakan Argeşo
Manufacturing Engineering Dept., Atılım University

Assist. Prof. Dr. Burcu Güldür
Civil Engineering Dept., Hacettepe University

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Semih Özmen

Signature :

ABSTRACT

IMPLICIT MONOLITHIC PARALLEL SOLUTION ALGORITHM FOR SEISMIC ANALYSIS OF DAM-RESERVOIR SYSTEMS

Özmen, Semih

Ph.D., Department of Civil Engineering

Supervisor: Assoc. Prof. Dr. Özgür Kurç

February 2016, 173 pages

This research mainly focuses on developing a computationally scalable and efficient solution algorithm that can handle linear dynamic analysis of dam-reservoir interaction problem. Lagrangian fluid finite elements are utilized and compressibility and viscosity of the fluid are taken into consideration during the reservoir modeling. In order to provide computational scalability and efficiency, domain decomposition methods implemented with parallel computing approaches such as Finite Element Tearing and Interconnecting (FETI) family solution algorithms are utilized for the coupling of the subdomains and a fully implicit monolithic solution algorithm is developed. Following that, the ways of performance improvements for the algorithm are demonstrated. Re-orthogonalization is utilized to increase the convergence rate of the solution of system equations and Krylov subspaces are utilized in order to decrease the required iterations for the future time integration steps.

Additionally, utilization of deflation methods on Preconditioned Conjugate Gradient (PCG) and Finite Element Tearing and Interconnecting (FETI) family solvers is discussed. Due to the fact that efficiency and behavior of the deflation methods depends on the deflation vectors utilized, different deflation vector generation methods are also investigated. Two of the deflation vector generation methods are from literature, i.e. “Subdomain Deflation Method” and “Recursive Deflation

Method for Heterogeneous Problems”. In addition to them, a novel semi-heuristic deflation vector generation strategy which relies on the pre-selected zero energy modes of finite element formulations is proposed. Requirements, improvements and efficiencies of these methods are demonstrated for the serial solution of water tank with flexible walls problem.

In order to investigate the efficiency and scalability of the presented solution approach on the solutions of more realistic problems by computer clusters, this approach is implemented by utilizing C++ programming language and PETSc library. In this parallel implementation, FETI-DP solution algorithm is utilized with different deflation vector generation algorithms. The efficiency and the scalability of the parallel solution framework are discussed for different types of finite elements, for different partitioning approaches and for different number of processors. Finally, the solution performance is presented for a large actual dam model, Pine Flat Dam in California, USA.

Keywords: Monolithic, Lagrangian, FETI, Dam-Reservoir Interaction, Deflation

ÖZ

BARAJ-REZERVUAR SİSTEMLERİNİN DEPREM ETKİSİ ALTINDA ÖRTÜK YEKPARE VE PARALEL OLARAK ÇÖZÜMLENMESİ

Özmen, Semih

Doktora, İnşaat Mühendisliği Bölümü

Tez Yöneticisi: Doç. Dr. Özgür Kurç

Şubat 2016, 173 sayfa

Bu araştırmada, barajlar için yapı-akışkan etkileşiminin doğrusal dinamik olarak çözümleyebilecek yüksek başarım ve ölçeklenebilir performanslı bir çözüm yöntemi geliştirilmiştir. Rezervuarın modellenmesinde suyun sıkıştırılabilir ve viskozite özellikleri göz önüne alınmıştır. Yüksek başarım ve arttırılabilir performans elde edebilmek için paralel hesaplama uygun alt-yapı tabanlı çözüm yöntemleri seçilmiştir. Lagrange çarpanlarına dayalı alt-yapı tabanlı bir çözüm yöntemi olan SEYY (Sonlu Elemanlar Yırt ve Yapıştır)(ing. FETI) yapı-akışkan etkileşimi problemlerini çözmek için uygulanmış ve sonuç olarak tamamen örtük yekpare bir çözüm yöntemi geliştirilmiştir. Çözüm yönteminin sayısal olarak oluşturulmasından sonra geliştirilen yöntemin nasıl iyileştirilebileceği araştırılmıştır. Sonuca ulaşma hızını arttırabilmek amacıyla farklı iyileştirme yöntemleri uygulanmıştır. Buna ek olarak, dinamik çözümleme sırasında takip eden zaman aralıklarında çözümleme için gereken çözüm yineleme sayısını azaltmak amacıyla daha önceki çözümler sırasında elde edilen Krylov alt uzay vektörleri kullanılmıştır.

Bunların dışında, deflasyon yöntemlerinin İyileştirilmiş Eşlenik Gradyan (ing. Preconditioned Conjugate Gradient) ve SEYY çözümleme yöntemlerine uygulaması incelenmiştir. Deflasyon yöntemlerinin başarımı kullanılan deflasyon vektörlerine bağlı olduğu için literatürde yer alan farklı deflasyon vektörü hesaplama yöntemleri

araştırılmıştır. Bunlara ek olarak, kullanılan sonlu eleman formülasyonlarından elde edilen önceden seçilmiş sıfır enerji şekillerine dayalı tamamen yeni bir yarı-buluşsal deflasyon vektörü hesaplama yöntemi önerilmiştir. Bu yöntemlerin gereksinimleri ve sonuç olarak kullanımlarıyla elde edilen iyileşmeler esnek duvarlı su tankı probleminin çözümlenmesi örneği üzerinde tartışılmıştır.

Baraj sistemlerindeki yapı-akışkan etkileşiminin çözümü için sunulan hesaplama yaklaşımın daha gerçekçi ve büyük örneklerin (bilgisayar kümeleri yardımıyla) çözümündeki başarımını ve ölçeklenebilirliğini incelemek amacıyla, bu yaklaşım C++ programlama dili ve PETSc kütüphanesi kullanılarak uygulanmıştır. Bu paralel uygulamada, FETI-DP çözüm yöntemi farklı deflasyon vektörü hesaplama yöntemleri ile birlikte kullanılmıştır. Geliştirilen paralel çözüm sisteminin başarımı ve arttırılabilirliği farklı sonlu eleman tipleri, farklı bölümlenme yaklaşımları ve farklı işlemci sayıları için araştırılmıştır. Son olarak, geliştirilen bu sistem ABD'de bulunan Pine Flat Barajı'nın çözümlenmesinde kullanılmış ve çözüm sisteminin nihai performansı gösterilmiştir.

Anahtar Kelimeler: Yekpare çözüm, Lagrange gösterimi, SEYY–Sonlu Elemanlar Yırt ve Yapıştır yöntemi, Baraj-Rezervuar Etkileşimi, Deflasyon yöntemi

To My Lovely Family

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor, Assoc. Prof. Dr. Özgür Kurç, because of his great guidance, support, and complete confidence in me during the whole research, publications, and presentations. Without his existence and experience, this research could not be completed.

Two other people that I would like to thank are Tunç Bahçecioglu and Utku Albostan from Panthalassa development team. In any obstacle, I felt comfortable with their support and collaboration.

I would like to thank COMOSYS developers and especially Mr. Syed Ateeq Ahmad for their understanding and support due to the fact that I spent some of my working hours to this study.

Most importantly, none of this would have been possible without the love of my family. I want to exhibit my special thanks to my dearest wife and my best friend, Özlem for her support and patience during this heavy work. Another special thanks to my son, Baran Ege who is the pride and joy of my life. I hope he will also find peace of mind in tough times with playing his children as I do with him. I wish to dedicate this study to them since their existence gave me the strength to accomplish this heavy task.

The hardest thing about such a study is to be continuously standing still against the questions starting with "when" and "why". For me, it is so important since this is the first PhD degree even in my extended families. Therefore, first, I need to apologize to my beloved ones for not spending a lot more time with them. I wish to thank my parents and my brothers that support and encourage me throughout this endeavor. Last but not least, I have to express my gratitude to my friends for their existence, thus, I could find the opportunity to relax with them.

The financial support provided during Ph.D. study by "2211-Scholarship for Ph.D. Students" of Turkish Scientific and Technical Research Council (TÜBİTAK) is highly appreciated.

TABLE OF CONTENTS

ABSTRACT	v
ÖZ	vii
ACKNOWLEDGMENTS	x
TABLE OF CONTENTS	xii
LIST OF TABLES	xvi
LIST OF FIGURES	xvii
LIST OF SYMBOLS	xix
CHAPTERS	
1. INTRODUCTION	1
1.1. Problem Definition	1
1.2. Objectives and Scope	6
1.3. Thesis Outline	8
2. BACKGROUND	9
2.1. Overview on Dam-Reservoir Systems	9
2.1.1. Dam Model	9
2.1.2. Foundation Model	10
2.1.3. Reservoir Model	11
2.1.4. Finite Element Time-History Analysis	12
2.2. Dam-Reservoir System Solutions	13
2.2.1. Analysis Approaches for Fluid-Structure Interaction	13
2.2.2. Discretization of Dam-Reservoir Systems	15
2.2.3. Overview on Domain Decomposition Methods	17
2.2.3.1. Equilibrium Definitions for the Partitioned System	18
2.2.3.1.1. Definition with Primal Variables	19
2.2.3.1.2. Definition with Dual Variables	22
2.2.3.1.3. Hybrid Definition with Primal and Dual Variables	26
2.2.4. Overview on FETI Family Solution Methods	26
2.2.4.1. One-Level FETI Method (FETI-1)	26

2.2.4.2. Two-Level FETI Method (FETI-2).....	30
2.2.4.3. Dual-Primal FETI Method (FETI-DP).....	32
2.2.4.4. Preconditioners.....	35
2.2.4.4.1. Dirichlet Preconditioner:.....	36
2.2.4.4.2. Lumped Preconditioner.....	36
2.2.4.4.3. Super-Lumped Preconditioner.....	37
2.2.4.5. Extension to Dynamic Analysis.....	37
2.2.4.5.1. Dynamic Analysis with FETI-1.....	40
2.2.4.5.2. Dynamic Analysis with FETI-2.....	41
3. SERIAL IMPLEMENTATION OF SOLUTION FRAMEWORK.....	43
3.1. Introduction.....	43
3.2. Implementation.....	43
3.2.1. Finite Element Implementations.....	44
3.2.2. Implementation for Static Analysis.....	46
3.2.3. Implementation for Dynamic Analysis.....	49
3.2.4. Coarse Problem Termination Criteria.....	51
3.3. Methods for Improving Convergence Rate.....	51
3.3.1. Re-orthogonalization of Direction Search Vectors.....	52
3.3.1.1. Mathematical Background.....	52
3.3.1.2. Improvement Achieved by Re-orthogonalization of Direction Vectors.....	53
3.3.2. Improvement of Convergence by Krylov Subspaces.....	56
3.3.2.1. Mathematical Background.....	56
3.3.2.2. Improvement Achieved by Utilization of Krylov Subspaces.....	58
3.3.3. Performance Tests with Different Preconditioners.....	61
3.3.3.1.1. Solid Mechanics Problem.....	61
3.3.3.1.2. Hydrostatic Problem.....	63
3.3.3.1.3. Fluid-Structure Interaction Problem.....	65
4. CONDITIONING THE INSTABILITIES.....	69
4.1. Introduction.....	69
4.2. Theory.....	71

4.2.1. Discretization	71
4.2.2. System Solution	72
4.2.3. Deflation Method	75
4.2.4. Implementation of Deflation	77
4.3. Deflation Vector Generation	79
4.3.1. Subdomain Deflation Method	80
4.3.2. Recursive Deflation Method for Heterogeneous Problems	81
4.3.3. A Novel Strategy: Semi-heuristic Deflation Method.....	82
4.3.4. Further Heuristics; Deflation Vector Assembly.....	84
4.4. Numerical Experiments.....	85
4.4.1. Water Tank Problem	85
4.4.1.1. Model with Pure Displacement Based Fluid Elements.....	87
4.4.1.2. Almost Incompressible Case by CG Variants.....	91
4.4.1.3. Almost Incompressible Case by FETI Variants.....	93
4.4.2. Actual Dam Model.....	95
4.5. Conclusion.....	99
5. PARALLEL IMPLEMENTATION OF SOLUTION FRAMEWORK	101
5.1. Introduction	101
5.2. Implementation	101
5.3. Numerical Experiments.....	107
5.3.1. Effect of Different Partitioning Configurations	108
5.3.2. Effect of Finite Element Formulations.....	114
5.3.3. Effect of Utilizing Different Processor Counts.....	117
5.3.4. Utilization of Krylov Subspaces	125
5.3.5. Solution Performance of a Large Actual Dam Model	128
6. CONCLUSIONS.....	131
6.1. Summary	131
6.2. Concluding Remarks	132
6.3. Recommendations for Future Study.....	133
REFERENCES.....	135

APPENDICES

A. FINITE ELEMENT FORMULATIONS	145
A.1 Pure Displacement Based Formulation	145
A.1.1 Theory	145
A.1.2 Finite Element Discretization.....	147
A.2 Mixed Formulation – u/p (Displacement/Pressure Based)	148
A.2.1 Theory	148
A.2.2 Finite Element Discretization.....	149
A.2.3 2D Pressure Interpolation.....	151
A.2.4 3D Pressure Interpolation.....	152
B. VALIDATIONS	155
B.1 Validations of Finite Element Implementations.....	155
B.1.1 Cantilever Beam Problem	155
B.1.2 Hydrostatic Pressure of Water Tank	158
B.1.3 Explicit Dynamic Analysis of Cantilever Beam Problem.....	160
B.1.4 Explicit Dynamic Analysis of Tall Water Column	161
B.2 Validations of Parallel Implementation of Solution Framework	164
B.2.1 Cantilever Beam	164
B.2.2 Water Tank Problem	166
B.2.3 Pine Flat Dam Problem	169
CURRICULUM VITAE	173

LIST OF TABLES

TABLES

Table 2.1. Pseudocode for FETI-1 PCPG algorithm	29
Table 2.2. Pseudocode for FETI-2 PCPG algorithm	31
Table 2.3. Pseudocode for Extension of FETI Methods to Dynamics.....	38
Table 3.1. List of implemented Finite Elements	46
Table 3.2. Improvement by re-orthogonalization with total iteration counts	56
Table 4.1. Pseudocode for Deflated PCG algorithm.....	78
Table 5.1. Partitioning details for different approaches	110
Table 5.2. 2D vs 3D finite element formulations.....	115
Table 5.3. Solution times (s) of models discretized with Q9 elements.....	118
Table 5.4. Solution times (s) of models discretized with H8 elements.....	119
Table 5.5. Performance of SH-ASM on actual cases.....	128

LIST OF FIGURES

FIGURES

Figure 2.1. Interface transformation.....	20
Figure 2.2. Interface compatibility.....	23
Figure 2.3. Variable types of dual-primal system definition	32
Figure 3.1. fQ9P3u/p and fH27P4u/p elements	45
Figure 3.2. Flowchart of solution framework for static analysis	48
Figure 3.3. Flowchart of solution framework for dynamic analysis.....	50
Figure 3.4. Improvement by re-orthogonalization on cantilever problem.....	54
Figure 3.5. Improvement by re-orthogonalization on tall water column problem.....	54
Figure 3.6. Improvement by re-orthogonalization on water tank problem.....	55
Figure 3.7. Iteration counts without Krylov subspaces improvement	59
Figure 3.8. Iteration counts with Krylov subspaces improvement.....	60
Figure 3.9. Partitioning of cantilever beam problem	62
Figure 3.10. Cantilever beam solved by FETI-1 with different preconditioners.....	62
Figure 3.11. Cantilever beam solved by FETI-2 with different preconditioners.....	63
Figure 3.12. Hydrostatic problem solved by FETI-1 with different preconditioners	64
Figure 3.13. Hydrostatic problem solved by FETI-2 with different preconditioners	65
Figure 3.14. Water tank problem solved by FETI-1 with different preconditioners .	66
Figure 3.15. Water tank problem solved by FETI-2 with different preconditioners .	67
Figure 4.1. Flowchart for deflated system solution.....	76
Figure 4.2. Some basic a priori known zero energy modes	83
Figure 4.18. Water tank with flexible walls.....	85
Figure 4.4. Water tank problem partitioned into (a) 4 and (b) 42 subdomains.....	86
Figure 4.5. Deflation vector counts and iteration counts	88
Figure 4.6. Additional memory requirements (in # of non-zeros).....	89
Figure 4.7. Operation counts for deflation vector generation, deflated solution	90
Figure 4.8. Solution by CG variants w/o $PC = IC$ ($\nu = 0.499$).....	91
Figure 4.9. Solution by CG variants w/o $PC = IC$ ($\nu = 0.49999$)	92

Figure 4.10. Operation counts for different number of subdomains.....	93
Figure 4.11. Additional memory requirements for different number of subdomains	94
Figure 4.12. Pine Flat Dam and its finite element mesh	95
Figure 4.13. Solution with displacement based fluid elements.....	96
Figure 4.14. Solution with mixed formulation fluid elements.....	97
Figure 4.15. Additional memory required (in # of non-zeros) for the solution	98
Figure 5.1. Flowchart for overall solution on Zargana	103
Figure 5.2. Flowchart for solution of equations on Orkinos.....	106
Figure 5.3. Homogeneous and heterogeneous partitioning of water tank model.....	109
Figure 5.4. Number of dual variables for different partitioning approaches	111
Figure 5.5. Interface solution times and iteration counts for both partitioning approaches.....	112
Figure 5.6. The total amount of data transfer and memory consumption for both partitioning approaches	113
Figure 5.7. Scaled solution timings for Q9 elements discretization	120
Figure 5.8. Scaled solution timings for H8 elements discretization	121
Figure 5.9. Total memory allocated for the solutions with Q9 elements.....	122
Figure 5.10. Memory allocated for solutions with H8 elements.....	123
Figure 5.11. The amount of data transfer for the solutions with Q9 elements.....	124
Figure 5.12. Iteration counts for the first 0.5 seconds of the dynamic analysis with step force	126
Figure 5.13. Iteration counts for the first 2 seconds of seismic analysis	127

LIST OF SYMBOLS

H	effective subdomain edge length
h	effective element edge length
Ω	solution domain
$\Omega^{(s)}$	subdomain
N_s	number of non-overlapping subdomains
$K^{(s)}$	stiffness matrix of a subdomain
$\mathbf{u}^{(s)}$	displacement vector of a subdomain
$\mathbf{f}^{(s)}$	applied force vector associated to a subdomain
$\mathbf{g}^{(s)}$	connecting forces on the interface between subdomains
$L^{(s)}$	assembly (global to local) matrix for a subdomain
$B^{(s)}$	compatibility constraints at the interface of a subdomain
K	block diagonal stiffness matrix obtained from all $K^{(s)}$
\mathbf{u}	block diagonal displacement vector obtained from all $\mathbf{u}^{(s)}$
\mathbf{f}	block diagonal applied force vector obtained from all $\mathbf{f}^{(s)}$
\mathbf{g}	block diagonal connecting force vector obtained from all $\mathbf{g}^{(s)}$
L	block diagonal assembly matrix obtained from all $L^{(s)}$
B	block diagonal assembly matrix obtained from all $B^{(s)}$
\mathbf{u}_g	global set of DOFs unique on the interface
<i>subscript i</i>	designates some property related to internal DOFs
<i>subscript b</i>	designates some property related to boundary/interface DOFs
$S^{(s)}$	condensed stiffness contributions of a subdomain to interfaces
$\tilde{\mathbf{f}}_b^{(s)}$	condensed loading contributions of a subdomain to interfaces
\tilde{S}	block diagonal Schur Complement matrix obtained from all $S^{(s)}$
$\tilde{\mathbf{f}}_b$	block diagonal condensed force vector obtained from all $\tilde{\mathbf{f}}_b^{(s)}$
λ	Lagrange multipliers (i.e. interface forces on a dual system)
$R^{(s)}$	zero energy mode vectors of a subdomain

$\alpha^{(s)}$	amplitudes of the zero energy modes
R	block diagonal zero energy mode vectors obtained from all $R^{(s)}$
α	block diagonal amplitudes of $R^{(s)}$ obtained from all $\alpha^{(s)}$
<i>superscript</i> +	designates inverse of a matrix, if not applicable then designates pseudo-inverse of a matrix
N_f	number of floating subdomains
C	admissible constraints matrix
Γ	designates the interface or some property related to interface
B_{p_b}	compatibility constraints at the interface scaled and restricted to boundary/interface
\tilde{F}_I^{-1}	designates a chosen preconditioner
I	designates some property related to internal DOFs
Δ	designates some property related to dual DOFs
Π	designates some property related to primal DOFs
$\bar{F}_I^D^{-1}$	Dirichlet preconditioner
$\bar{F}_I^L^{-1}$	Lumped preconditioner
$\bar{F}_I^{SL^{-1}}$	Super-lumped preconditioner
$M^{(s)}$	mass matrix of a subdomain
$C^{(s)}$	damping matrix of a subdomain
$\dot{\mathbf{u}}^{(s)}$	acceleration vector of a subdomain
$\mathbf{u}^{(s)}$	velocity vector of a subdomain
$t, \Delta t, t_{total}$	current time, time step and total time
$\mathbf{v}^{(s)}$	momentum vector of a subdomain
\mathbf{v}	global momentum vector
$\dot{\mathbf{u}}$	global velocity vector
M	global mass matrix
<i>superscript</i> n	a property related to n^{th} time step
<i>head accent</i> $\hat{}$	designates a property related to dynamics
<i>superscript</i> k	value of a parameter at iteration k

CHAPTER 1

INTRODUCTION

1.1. Problem Definition

Dam analysis is a wide, multi-disciplinary research area due to the fact that it is a multi-physics problem with interacting domains such as dam structure, reservoir water and foundation rock. Not only dam analysis involves the computation of the behavior of domains with different material properties and physics in a spatially large volume, but also accurate modeling of the application of loads, seismic actions, construction sequences, indirect actions such as thermal and shrinkage effects, and more are still being researched for a better representation of the behavior of a dam structure. As being an important structure itself, assessment of the dam's seismic performance is still a current research area. One of the main problems of the seismic dam analysis is the modeling of the interactions among the dam, reservoir, and foundation domains. For a more accurate representation of the interaction among different domains, not only the dynamic domain behavior itself but also the coupling between neighboring domains should be accurately formulated [1].

During the seismic analysis of dams, representation of different domain behaviors is generally simplified. The dynamic behavior of the reservoir water domain is usually represented as an equivalent time dependent, parabolic pressure/force field applied to the surface of the dam. Although such an approximation has several limitations, it usually produces conservative results for modeling the reservoir water [2]. For a more detailed analysis, the reservoir water can be modeled with finite elements. In order to characterize full properties of the fluid flow, in general Navier-Stokes formulation is utilized. On the other hand, when the fluid behaves within the limits of

acoustics, the fluid flow is assumed to be linear and inviscid. This way, a considerable simplification can be utilized for the fluid formulation. However, in this case, dam-reservoir analysis under dynamic loading is significantly affected by the viscous effects [1, 3, 4].

The interaction between dam and reservoir, is mostly assumed to take place in one-way, i.e. effect of structure's deformation on the fluid behavior is neglected since the dam lateral deformations are negligible [2]. Otherwise, a coupled analysis can be performed where dam-reservoir interaction is approximated by modeling both the dam and reservoir by finite elements and analyzing both domains together (monolithic solution [5–12]) or separately (partitioned solution [13–17]). In monolithic analysis approach, all equations of multi-physics system are solved simultaneously with a single method and a time step. In partitioned solution methods, in contrast, equations of different domains can be solved separately which allows using an optimum analysis approach, system equation solution method and different time steps for each domain. Even though, partitioned solution methods are considered as computationally less expensive than the monolithic solutions, accurate formulation and numerical implementation of the coupling between partitioned domains are a significant challenge [16]. In addition to that, since solutions of each partitioned system are carried out one after another, for each time step, there is a time lag in between solutions and it is artificially causing an insertion of additional energy to the overall system [9].

During the monolithic dynamic analysis of dam-reservoir problems, dam structure and foundation rock is generally represented by displacement based (Lagrangian) solid finite elements [18]. On the other hand, reservoir can be represented with fluid elements with displacement field, pressure field, velocity field, or combined field formulations [1]. When the reservoir is discretized with fluid finite elements without a displacement field definition, a coupling equation should be defined in between the displacement field of the solid finite elements and the formulation variable of the fluid finite elements. If the reservoir is discretized with Lagrangian fluid finite

elements, all finite elements in the dam-reservoir system will have only displacement variables as their degrees of freedom (DOFs). This way there is no need for any special dam-reservoir interface coupling elements or methods. Coupling will be automatically handled by the solution of system equations of the whole domain simultaneously [9]. As a result, any approximations and limitations of coupling algorithms will be eliminated. Monolithic solution of such fully Lagrangian system is, however, computationally expensive, because of the utilization of displacement formulation for fluid elements. For example, in case of 3D problems, when displacement formulation is utilized for the modelling of fluid material, three displacement variables at each node of a brick element is required to represent a single pressure variable in pressure field formulation.

Another important topic for the seismic dam-reservoir analysis is the energy absorption by the far-end boundaries. Briefly, this phenomenon originates from the existence of acoustic waves leaving the system fully or partially in time or space sense. One way of handling this problem is to model the foundation and reservoir domain large enough such that the reflected waves from the boundaries have a negligible effect on the results [1, 4, 19]. As expected, this approach requires significant amount of computational resources. Alternatively, truncated models with absorbing boundary condition definitions [20] can be utilized.

The monolithic dynamic solution of dam-reservoir systems requires considerable computational resources because of the simultaneous solution of differential equations of different materials in a single time step. When the problem is intended to be solved with non-linear material behaviors, not only the mesh size increases but also additional computations are required for computing the stresses at every node of a single finite element. Because of this reason, utilization of high performance computing techniques is compulsory for solving such problems at a reasonable time frame, days rather than months.

The requirement of high computational power for the dam-reservoir interaction problems needs highly scalable solution methods. Thus, iterative solution methods such as Finite Element Tearing and Interconnecting (FETI) family of solution methods [21–24] can be a good alternative since they are proven to be highly scalable for problems having a single domain [25]. In FETI family of solution methods, as being a domain decomposition method of solution, domain is partitioned into non-overlapping subdomains and the interface DOFs are replaced with interface equilibrium forces (Lagrange Multipliers). Therefore, each subdomain can be solved independently and a coarse problem is defined by the interface equilibrium forces. As a result, comparatively small sized coarse problem [26] is generally solved for interface equilibrium forces by utilizing an iterative solution method and overall system solution is recovered from the resultant interface equilibrium forces. However, in case of huge stiffness difference in the interface of solid and fluid domains, convergence of the iterative solution is degraded [27]. Moreover, as a result of the domain decomposition, floating subdomains that are insufficiently supported or even unsupported subsystems whose stiffness matrix involves several singularities can be formed. When the floating subdomains contain Lagrangian fluid finite elements, they may have additional spurious zero energy modes (a generalization for rigid body modes in solid mechanics) [28] when compared to solid subdomains. Besides, these may result in slow convergence or complete divergence of iterative solution [29]. Therefore, a special consideration is required for such problems.

Deflation which is one of the problem conditioning methods reduces the condition number of the system by defining a projection matrix which projects the extreme eigenvalues out of the system. Therefore, the convergence rate of the iterative solution is improved or the solution is carried out successfully with lesser computational requirements [30]. Deflation methods have several algebraic connections with multi-grid methods and projection preconditioners [29, 31, 32]. Performance and efficiency of solutions with deflation method are highly depends on the deflation vectors utilized [33–36]. In case of subdomains composed of solid elements, computation of rigid body modes and utilizing them as deflation vectors is

a straight forward task [33, 37, 38]. However, in case of subdomains with Lagrangian fluid finite elements, such methods are not applicable and a fully rank revealing Eigen solution is required [39]. Generation of vectors that define the projection matrix for the deflation method is a challenging task. Carefully selected, sufficient number of deflation vectors would result in a faster convergence rate but generation of the deflation vectors should also be computationally efficient [29].

The seismic analysis of dam structure requires the dynamic analysis of the whole domain. One of the dynamic analysis methods is the time integration method where time derivatives of the equation of motion are discretized by finite difference method [40]. The dynamic solution of a large domain with a FETI like iterative solver has also several challenges and opportunities. Since the coarse problem solution in FETI family methods is generally solved by a Krylov subspace generating iterative solution, implicit time integration becomes attracting. In other words, obtained subspace vectors can be utilized to enhance the solution for the following time steps by assuming the overall system behavior is not changing drastically [41]. Considering the scalability of FETI family methods and improved convergence rate with Krylov subspace vectors, unconditionally stable implicit time integration methods can be considered for dynamic analysis. This way, larger time steps can be utilized when compared to explicit time integration based analysis methods which is also another scalable solution approach [42]. However, direct utilization of Newmark's implicit time integration scheme suffers from spectral drift. Therefore, a variant of this method based on conservation momentum can be utilized [43].

As a result, having different domains with different material properties and different governing equations, and large solution domain size are the main challenges of the seismic dam analysis. Due to large solution domain size and having several degrees of freedom at each node during monolithic solution of the whole domain, high performance computing techniques must be utilized during the dynamic analysis of dam-reservoir interaction problem. As being a highly scalable solution method, FETI family iterative solvers, are good candidate for reducing the analysis time.

Guaranteeing the convergence of the FETI solvers and reducing the number of iterations at each time step of the dynamic analysis are the main challenges for problems having different types of domains. Thus, developing a robust, scalable and practical solution method for such large scale domains would significantly help improving the seismic design of dam structures.

1.2. Objectives and Scope

The main objective of this research is to develop a computationally efficient solution algorithm that can handle linear dynamic analysis of dam-reservoir interaction problem. In order to provide computational efficiency, domain decomposition methods implemented with parallel computing methodologies are utilized. Finite Element Tearing and Interconnecting (FETI) algorithm which is a Lagrange Multipliers based domain decomposition algorithm are used for the coupling of the subdomains and fully implicit monolithic solution algorithm is developed. Compressibility and viscosity of the water are taken into consideration during the reservoir modeling by two different Lagrangian fluid element formulations. Material heterogeneity, floating subdomains formed during the FETI solution, zero-energy modes of fluid elements, spectral drift of the dynamic solution and divergence/slow convergence in some cases are the main challenges for the fully Lagrangian, implicit monolithic parallel solution in the seismic analysis of dam-reservoir interaction problem. For this objective, the study will focus on three main research steps;

- Development of a Lagrange Multipliers based (FETI variant) fully implicit monolithic parallel solution algorithm that can successfully converge to the solution with an acceptable accuracy in a reasonable time for multi-physics problem of dam and reservoir by utilizing fully Lagrangian fluid element formulations for reservoir. This step involves the implementation and validation of finite element formulations and utilization of these elements in FETI family of solvers for the dynamic solution of dam-reservoir interaction problems. Two different Lagrangian fluid formulations are implemented and

validated for benchmark problems. State of the art techniques in the literature for handling floating subdomains and instabilities resulting from material heterogeneity, fluid element formulation and implicit time integration are implemented and their performance, strengths and weaknesses are examined on several benchmark problems.

- Investigation of the possible improvements in the algorithm for increasing the accuracy, improving the convergence rate and optimizing the computational efficiency. For this purpose, the use of deflation method on the convergence rate, performance and memory efficiency are investigated. Besides, a novel deflation vector generation method is also developed. Different fluid element formulations are tested for different material properties. Advantages and disadvantages of improvement methods are examined on representative sample problems by using two different FETI solvers i.e. FETI-1 and FETI-2.
- Extension of investigations to determine the efficiency and the scalability of solution algorithm with different deflation vectors. Several benchmark problems are solved by utilizing the parallel implementation of the solution approach based on FETI-DP solver. Advantages and disadvantages of discretization with different finite elements and different partitioning configurations are examined. Different deflation vector generation methods are compared with each other by changing the number of processors utilized on a computer cluster. Besides, overall performance details of the solution approach are presented for the solution of an actual large dam problem.

Within the scope of this thesis, serial implementation of the solution approach is implemented on MATLAB environment, whereas parallel implementation is implemented with utilizing C++ programming language and PETSc library. During the parallel solutions, a computer cluster with 48 processors is utilized. The scope of the research is limited to development of a computationally efficient solution approach for linear dynamic analysis of dam-reservoir interaction problems.

1.3. Thesis Outline

This thesis contains six chapters. The first chapter is an introductory chapter that encapsulates the problem definition and the objective and scope of the study. Second chapter summarizes the literature survey and overview on dam-reservoir systems, its solution approaches, and its discretization. Moreover, as a solution method; domain decomposition methods specifically FETI family of solution methods are summarized in this chapter. In the third chapter, details of the fully Lagrangian implicit monolithic solution algorithm for the seismic analysis of dam-reservoir systems are discussed. FETI-1 and FETI-2 algorithms based serial implementation of the overall solution framework is presented and convergence rate improving methods are discussed, additionally. Fourth chapter focuses on the conditioning methods for instabilities formed during the solution and especially, a novel deflation vector generation algorithm is proposed. Two different fluid element formulations are tested for different material properties on representative sample problems by the serial implementation. FETI-DP based parallel implementation of the solution framework by utilizing C++ programming language and PETSc library is presented in the fifth chapter. This implementation is utilized for the solution of more realistic and comparatively large problems by a computer cluster. Effect of utilizing different finite elements, different partitioning configurations and different number of processors are investigated by numerical experiments. Overall performance details of the solution approach are presented for the solution of an actual large dam problem. Finally, conclusions are summarized in the last chapter which is the sixth chapter.

Theoretical details of Lagrangian fluid element formulations, finite element validations and overall framework validation in terms of mechanics are demonstrated in APPENDIX A and APPENDIX B, respectively.

CHAPTER 2

BACKGROUND

2.1. Overview on Dam-Reservoir Systems

This section describes analysis and evaluation procedures required for assessing the seismic performance of dams-reservoir systems. One of the acceptable methods of analysis for computing deflections and stresses developed in the dam is finite element analyses. The finite element analysis should be conducted by developing an accurate model of the dam system. In the following subsections, modelling techniques for each domain that compose the dam system will be discussed.

2.1.1. Dam Model

The finite element analysis highly depends on the finite element mesh employed for the dam and the mesh is required to closely match the domain geometry and is suitable for application of the various loads. Displacement based solid finite elements are generally utilized for the dam structure [18]. The type of finite element mesh developed is mostly dependent on the geometry of the dam and the ability of the displacement field of the element to capture the displacement and stress fields that one is attempting to model. Therefore, it is not possible to strictly define the number of elements of which that constitute a representative finite element mesh for all cases [18, 44]. However, generally, high order elements are utilized to build relatively coarse meshes. In case of 3D analysis, for example, the linear 8-node solid hexahedron element requires finer meshing to obtain the same displacements. The foundation profile sometimes may dictate the size of elements. For instance, considerably irregular foundation profiles generally require smaller elements to

model the dam geometry. When elements get smaller, generally they become increasingly sensitive to geometric discontinuities [44]. For example, large stress concentrations that are fictitious because of the formation of cracks in the foundation material are obtained on the re-entrant corners at the dam-foundation interface. Dynamic characteristics and the response of the dam to earthquake loading are also affected by the size of elements. For example, in order to compute the contribution of all significant modes of large thin arch dams, finite element mesh should be sufficiently fine [2, 4]. As a rule of thumb, a finite element mesh of a concrete dam should embrace at least 5 rows of elements along the dam height and sufficient number of elements along the dam axis. The number of elements along the dam axis is determined so that the aspect ratios of the elements is less than 2 [1].

2.1.2. Foundation Model

In order to account for the effects of foundation flexibility on the deflections and stresses of the dam, an appropriate volume of the foundation rock should be considered during the modeling. In general, this volume extends to a large enough distance beyond which the effect of foundation on deflections and stresses of the structure become negligible. Although, it is possible to develop a foundation model that closely matches the site topography, such an extravagant model is not required in practice [1, 2]. In general, finite element mesh becomes finer near the dam-foundation interface where the largest deformations and stresses are formed, whereas coarser mesh is employed away from the dam, where the interaction with the dam is reduced.

As a rule of thumb, the ratio of the foundation deformation modulus to the concrete modulus of elasticity (E_f/E_c) is utilized to determine the size of the foundation model. For example in case of a competent foundation rock with E_f/E_c equal or greater than 1, the foundation model should extend one dam height in the upstream, downstream, and downward directions. In case of flexible foundation rocks, the

foundation model should be finer and it is required to extend twice the dam height in all directions [1].

2.1.3. Reservoir Model

In addition to dam-foundation interaction, damping, and the characteristics of earthquake ground motion, dam-reservoir interaction significantly affects the seismic response of concrete dams to earthquake excitation. One of the first studies on this topic was carried out by Westergaard in 1933 [45]. By assuming a rigid gravity dam with a vertical upstream face in 2D, he demonstrated that the hydrodynamic pressures applied on the upstream face of the dam because of the seismic motion equals to the inertia forces of some water body deforming with the dam whereas the rest of reservoir water remains stationary. He suggested that these hydrodynamic pressures can be expressed as a parabolic shape along the upstream face with a base width equal to $7/8$ of the dam height.

More accurate added-mass representation of the reservoir can be obtained by a finite element solution of the pressure wave equation that is fully considering the complex geometry of the dam and the reservoir. In these methodologies, incompressible fluid finite elements are utilized, surface waves are neglected and rigid reservoir boundaries are assumed [4, 46]. Although finite element mesh of the incompressible water can closely match the reservoir topography, in general and for most cases a prismatic model constructed by projecting the dam in the upstream direction would be sufficient. As a rule of thumb, the reservoir model should at least extend three times the dam height in the upstream direction and should have at least three layers of elements in that direction. Then, the pressure values computed for the nodal points on the upstream face of the dam are converted into equivalent nodal forces, from which an added-mass matrix demonstrating the inertial effects of the reservoir is obtained. The added-mass matrix obtained is a symmetric matrix coupling all the nodal DOFs on the upstream face of the dam structure [47]. Several important factors

on the seismic response of dam systems such as the effects of water compressibility and reservoir boundary absorption are ignored in aforementioned methods. However, it is shown that the reservoir boundary absorption and the compressibility of water can considerably influence the hydrodynamic pressures and therefore the seismic response of concrete dams [2].

Interaction of a concrete dam with the reservoir increases the dam vibration periods [1, 4, 46] since the dam structure cannot move without deforming the water in contact with it. Therefore, the total mass in motion increases with the water added and hence the natural periods of the dam increases, which in turn affects the effective earthquake inertia forces. Damping is also increased because of the partial absorption of pressure waves at the reservoir boundaries [19, 48]. In conclusion, reservoir is a significant factor that can change the earthquake response of the dam with respect to that for the dam with empty reservoir, and it should be considered carefully in seismic dam analysis [1, 2, 48].

2.1.4. Finite Element Time-History Analysis

The seismic response of concrete dams can be evaluated by the time-history analysis. In most cases, linear time history analysis coupled with engineering judgement is sufficient to evaluate the seismic safety of a concrete dam [1, 2]. The linear time-history response analysis can analyze time dependent characteristics of the dynamic response. Besides, acceleration time-histories can be utilized as the seismic input, and complete response histories (i.e. nodal displacements and element stresses) of the dam for the entire duration of the earthquake can be obtained by the solution of the equations of motion. The finite elements that are utilized for the discretization of the idealized dam-reservoir-foundation system is used for the assembly of the equations of motion [44].

2.2. Dam-Reservoir System Solutions

The behavior of dam-reservoir interaction can be considered as a multi-physics problem where the fundamental equations for solid and fluid domains are considered in a single problem. Thus, the both solid and fluid domains cannot be approximated as independent due to the tight coupling in between them. Such problems are defined as fluid-structure interaction (FSI) problems where "the flow around a body has a strong impact on the structure and/or on the movement of the body and the modification of the structure or the position of the body or a component of the body due to the flow has an influence on the flow which is not negligible" [18]. Therefore, both the fluid and the structure equations should be defined by the relationships of continuum mechanics in order to represent the physical phenomena of both bodies deforming together.

2.2.1. Analysis Approaches for Fluid-Structure Interaction

Depending on the physical nature of the interaction, analysis approaches can be classified in two groups [13, 15, 16, 49–52] as monolithic and partitioned analysis methods. In fact, for only small sized linear problems that can be solved by hand calculations or by computer symbolically, it is possible to eliminate the field variables at the level of differential equations by substituting one into the other. In other words, a common discretization field variable (i.e. displacements) for both domains can be eliminated from the set of equations obtained from solid mechanics by inserting the rewritten set of equations obtained from fluid mechanics. Even if these are small sized problems, as the number of equations are considered, it could be a difficult task to solve such complex equations.

Former group of analysis methods is the monolithic analysis methods [5–12] which require a simultaneous solution for all unknowns of the coupled overall system i.e. solid and fluid domains. In other words, both solid and fluid domains are analyzed

with a consistent discretization in space and time in a single iteration. This way, all interaction effects between the fundamental equations of both domains are covered without any approximation [9] since coupling conditions are enforced strictly as part of the algebraic system.

Latter group of analysis methods is the partitioned analysis methods [13–17]. For these methods, solution domain is partitioned according to the type of governing equations. For fluid-structure interaction case, first group is governed by the fluid dynamics, whereas the other one is governed by the structure dynamics. In this approach, the fluid and structure domains are each solved multiple times at each time step and exchanging the field variables at the common interface, until some predefined tolerance is obtained, at which point the algorithm can progress to the next time step [10].

Iterative resolutions uncoupling large systems can be carried out in partition methods and hence these methods are generally preferred to monolithic approaches. Uncoupling of the different physical domains forms the basis for all of the partitioned methods. This property makes it possible to utilize different solution algorithms for each uncoupled domain. In case of strong coupling, efficiency of these methods may degrade due to excessive number of iterations required. Besides, in case of 3D problems, the dimension of the pure mechanical problem is considerably larger than the fluid domain dimension and that leads to unbalanced subsystems. Besides these methods may also lose the numerical stability for large time steps, and can decrease the accuracy of the time integration algorithm in some cases [53]. Another advantage of monolithic solution methods is that they are immune to the added mass effect which is stated as the numerical instabilities occurred during the solution of interaction problems where solid and fluid materials have similar densities and the solid structure is thin. In this case, partitioned methods may become unstable or very expensive by requiring a large number of sub-iterations [54]. Therefore, due to the mentioned concerns above, a monolithic approach can be

preferred for fluid-structure interaction problems even if it requires considerable amount of computational resources [9].

2.2.2. Discretization of Dam-Reservoir Systems

As one of the acoustic fluid-structure interaction problems, dam-reservoir interaction contains the analysis of the solid structure domain i.e. dam, the fluid domain i.e. reservoir and the coupling in between these two domains. Displacement formulation is generally utilized for the discretization of solid structure [18]. On the other hand, several finite element formulations are proposed to represent the fluid domain for the analysis of fluid-structure interaction problems such as pure displacement, velocity potential, and displacement potential and pressure. Choosing a scalar variable such as pressure for the fluid field considerably reduces the size of problem compared to the displacement formulation. In case of dynamic analysis, it is well known that the pressure formulation results in a non-symmetric matrix [3]. The non-symmetry of the matrix can be removed by utilizing the velocity potential formulation or the pressure-displacement potential formulation on the expense of an added damping matrix [55]. However, these formulations are developed for inviscid fluid only. The displacement formulation can model the viscosity of a fluid, and the coupling condition can be easily implemented at the interface between fluid and structure. Moreover, since the field variables are same for fluid and structural domain, displacement based formulations do not necessitate any special condition at the interface or new solution methodologies for the coupling. However, the displacement based formulations suffers from the presence of the non-zero frequency modes without any physical meaning (i.e. spurious modes [56]), and locking in the frequency analysis of a solid vibrating in a nearly incompressible fluid [57]. Furthermore Cheng et.al [3] stated that the displacement formulation locks in the analysis of a nearly incompressible fluid interacting with a flexible boundary.

Several improved formulations are proposed by introducing different approaches. The penalty method is one of these approaches and has been shown that it gives good solutions for the cases considered [58]. Accordingly, Olson and Bathe [57] showed that this method “locks up” in the frequency analysis of a solid vibrating in a fluid cavity. Besides, it is demonstrated that reduced integration applied on the penalty formulation improves the results but does not guarantee the convergence to solution for all cases. A pure displacement based formulation with rotational constraints and a reduced integration technique is proposed by Wilson and Khalvati [59], and a four-node element with a reduced integration technique and an element mass matrix projection is developed by Chen and Taylor [60]. Besides, another promising method to model the fluid domain is conducted by Bermudez and Rodriguez [61]. In this research, a three-node triangular edge element is proposed but the DOFs of these elements are not those of the structure and hence special considerations are required for the coupling. Because of the lack of generality and the spurious modes encountered in the mentioned methods, Bathe [62] investigated the causes of the spurious non-zero frequencies and hence showed that they are originated due to the use of the pure displacement based formulation (including the penalty formulations) and the mishandling of the fluid-structure interface conditions. They proposed displacement/pressure based with mixed formulation elements that satisfy the inf-sup condition. This formulation is also applicable in case of the analysis of incompressible or almost incompressible materials. Because of the additional field variable of pressure to the fluid element, computational requirements of this formulation are higher than that of the displacement based formulation. However, in case of almost incompressible fluid material, the pressure DOFs can be statically condensed out in the element level, and hence same matrix size as in the displacement based formulation is obtained as a result [3].

From these various formulations, pure displacement based formulation with rotational constraints and a reduced integration technique [59] is heavily used by the dam behavior researchers [63]. In addition to this formulation, displacement/pressure based mixed formulation [62] is presented in APPENDIX A

2.2.3. Overview on Domain Decomposition Methods

The increasing problem sizes in the finite element analysis led researchers to spend extensive efforts on the development of efficient and high performance solvers. Due to its high potential for the utilization of available computational resources, domain decomposition methods [64] receive great attention especially in computational mechanics. Domain decomposition methods can be briefly summarized as the solution methods that redefine a global domain problem as a set of subdomain problems. Performance gain of the domain decomposition methods increases when the problems on each subdomain become independent from each other [65]. Provided that subdomains are intersecting only at their interfaces, solution method is called non-overlapping domain decomposition method. Otherwise, it is called overlapping domain decomposition method.

Main stages of the non-overlapping domain decomposition methods can be summarized as the decomposition of the problem domain, condensation of the problem on the interface between subdomains to guarantee the continuity of primal unknowns and the equilibrium of fluxes, and then the solution of the interface problem by a direct or iterative solver. In case of solution with iterative solver, when utilized with an appropriate preconditioner, a domain decomposition method is mostly scalable with respect to the mesh size h (or the effective element edge length) of the given problem. In order to be scalable with respect to the subdomain size H (or the effective subdomain edge length), it must also be utilized with a "coarse space" preconditioner [66] whose mathematical formulation is similar to that of the "coarse grids" defined in multigrid methods [67–72]. Obviously, scalability with respect to the number of subdomains is a necessary condition for achieving parallel scalability—which is, delivering larger speed-ups as the number of processors utilized for the solution of a given problem increases.

Most known non-overlapping domain decomposition methods are the primal approach, Balancing Domain Decomposition (BDD) [73], and the dual approach,

Finite Element Tearing and Interconnection (FETI) [21]. In case of a classical elastic problem, the former consists in computation of the continuity of interface displacements that guarantees the force equilibrium among the subdomains, while the latter involves the computation of the equilibrated interface forces that ensures the continuity of the displacements among the subdomains.

2.2.3.1. Equilibrium Definitions for the Partitioned System

Let's consider a domain Ω partitioned into N_s number of non-overlapping subdomains $\Omega^{(s)}$ and assume that a linear static analysis of this partitioned system will be performed. Thus, the equilibrium equations of the partitioned system is given by

$$K^{(s)}\mathbf{u}^{(s)} = \mathbf{f}^{(s)} + \mathbf{g}^{(s)} \quad s = 1, \dots, N_s, \quad (1)$$

where $K^{(s)}$, $\mathbf{u}^{(s)}$ and $\mathbf{f}^{(s)}$ are the subdomain stiffness matrices, displacements and applied forces, respectively. $\mathbf{g}^{(s)}$ are the connecting forces on the interface between subdomains (note that they are equal to zero on the internal DOFs). For the sake of simplicity, also assume that the subdomain meshes are conforming at the interfaces.

When the interface forces are assembled on the interface, the resultant should be zero (static equilibrium equation):

$$\sum_{s=1}^{N_s} L^{(s)T} \mathbf{g}^{(s)} = 0 \quad (2)$$

where $L^{(s)}$ is a Boolean transformation (or assembly) matrix. The interface forces are such that the interface DOFs satisfy the compatibility condition, i.e.:

$$\sum_{s=1}^{N_s} B^{(s)} \mathbf{u}^{(s)} = 0 \quad (3)$$

This relation states that for any pair $(u^{(i)}, u^{(j)})$ of DOFs matching on the interface, $u^{(i)} - u^{(j)} = 0$. Hence, $B^{(s)}$ are signed Boolean matrices designating the compatibility constraints at the interface.

The equilibrium of domain Ω is fully described by the set of local equilibrium equations, Equation (1), and by the interface constraints, Equations (2) and (3). In block diagonal notations, Equation (1), (2), and (3) can be written as;

$$\begin{cases} K\mathbf{u} = \mathbf{f} + \mathbf{g}, \\ L^T \mathbf{g} = 0, \\ B\mathbf{u} = 0, \end{cases} \quad (4)$$

where

$$K = \begin{bmatrix} K^{(1)} & & \\ & \ddots & \\ & & K^{(N_s)} \end{bmatrix} \mathbf{u} = \begin{Bmatrix} \mathbf{u}^{(1)} \\ \vdots \\ \mathbf{u}^{(N_s)} \end{Bmatrix} \mathbf{f} = \begin{Bmatrix} \mathbf{f}^{(1)} \\ \vdots \\ \mathbf{f}^{(N_s)} \end{Bmatrix} \mathbf{g} = \begin{Bmatrix} \mathbf{g}^{(1)} \\ \vdots \\ \mathbf{g}^{(N_s)} \end{Bmatrix},$$

$$L^T = [L^{(1)T} \quad \dots \quad L^{(N_s)T}],$$

$$B = [B^{(1)} \quad \dots \quad B^{(N_s)}],$$

It is obvious that in Equation (4), for each subdomain, one set of interface displacements and one set of interface forces are defined. Therefore, mainly two different system solution approaches for the domain decomposition problems: with primal variables (interface displacements) and with dual variables (interface forces) can be developed. Other than these approaches, there is a hybrid approach which utilizes both primal and dual variables for the solution.

2.2.3.1.1. Definition with Primal Variables

In order to solve the domain decomposition problem defined in Equation (4), displacements that satisfy the interface compatibility, Equation (3) a priori can be utilized. Therefore, a unique global set of DOFs on the interface, \mathbf{u}_g is defined as;

$$\mathbf{u}^{(s)} = L^{(s)}\mathbf{u}_g \text{ or } \mathbf{u} = L\mathbf{u}_g \quad (5)$$

where $L^{(s)}$ is the transformation matrix presented also in Equation (2) that can be utilized to extract subdomain (i.e. local) DOFs from the global set.

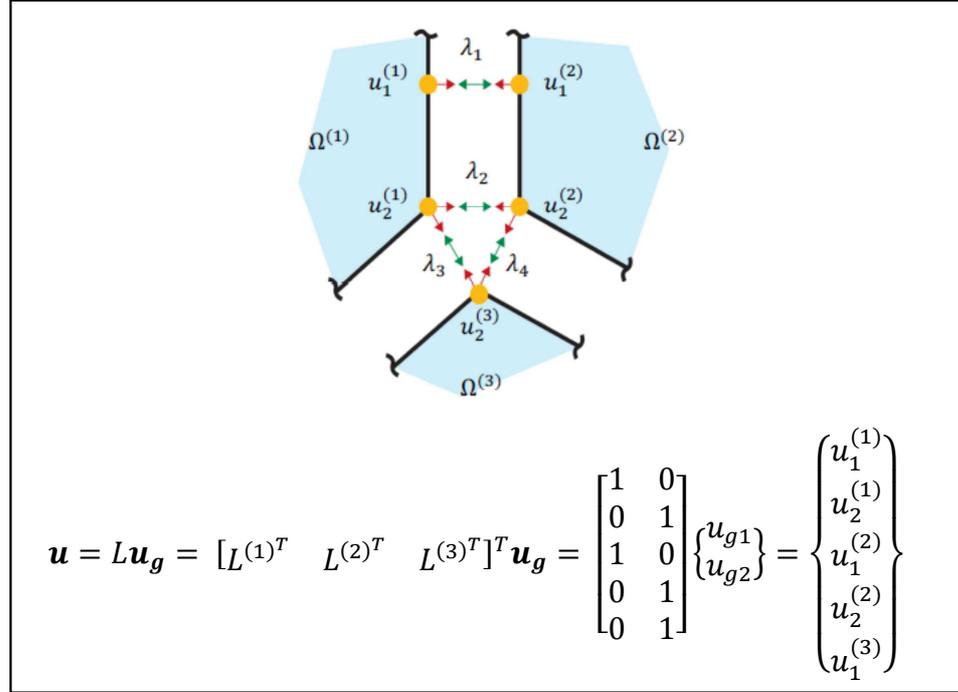


Figure 2.1. Interface transformation

Obtaining $\mathbf{u}^{(s)}$ from a unique set implies the interface compatibility, Equation (3) and Equation (4) and can be expressed as

$$B\mathbf{u} = BL\mathbf{u}_g = 0 \quad (6)$$

for any set of global displacements, \mathbf{u}_g . Similarly, all compatible displacements can be expressed as in Equation (5). Thus

$$L = null(B) \quad (7)$$

In other words, L is formed by the all solution vectors x to the equation $Bx = 0$. An example is presented in Figure 2.1 to demonstrate these concepts. Introducing Equation (5) in Equation (4) yields

$$\begin{aligned} KLu_g &= \mathbf{f} + \mathbf{g}, \\ L^T \mathbf{g} &= 0, \end{aligned} \quad (8)$$

Solving the equilibrium equations of all primal variables of the system defined in Equation (8), simultaneously is not only impractical but also eliminates the advantage of domain decomposition methods as the number of interface equations increases [74]. A basic approach in domain decomposition method is to first condense the internal contributions of each subdomain to their boundaries and assemble the interface system with primal variables such as displacements. Following the solution of the interface displacements, internal displacements of each subdomain can be recovered. Static equilibrium equation for each subdomain can be given as;

$$K^{(s)} \mathbf{u}^{(s)} = \mathbf{f}^{(s)}, \quad s = 1, \dots, N_s \quad (9)$$

By first numbering the internal DOFs and then the DOFs at the subdomain interfaces, the assembled system of equations will have the form presented in;

$$\begin{bmatrix} K_{ii}^{(s)} & K_{ib}^{(s)} \\ K_{bi}^{(s)} & K_{bb}^{(s)} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_i^{(s)} \\ \mathbf{u}_b^{(s)} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_i^{(s)} \\ \mathbf{f}_b^{(s)} \end{Bmatrix}, \quad s = 1, \dots, N_s \quad (10)$$

where subscripts i and b denote the internal and interface DOFs, respectively. Hence, internal stiffness and force contributions can be condensed to interfaces by using Schur Complements to obtain;

$$S^{(s)} = K_{bb}^{(s)} - K_{bi}^{(s)} K_{ii}^{(s)-1} K_{ib}^{(s)}, \quad s = 1, \dots, N_s \quad (11)$$

and

$$\tilde{\mathbf{f}}_b^{(s)} = \mathbf{f}_b^{(s)} - K_{bi}^{(s)} K_{ii}^{(s)-1} \mathbf{f}_i^{(s)}, \quad s = 1, \dots, N_s \quad (12)$$

Interface system contributions of each subdomain can be assembled by using block matrix notation and utilizing transformation matrices;

$$\tilde{S} = L_b^T S^{(s)} L_b \text{ and } \widetilde{\mathbf{f}}_b = L_b^T \tilde{\mathbf{f}}_b^{(s)} \quad (13)$$

in order to form the system equilibrium;

$$\tilde{S} \mathbf{u}_b = \widetilde{\mathbf{f}}_b \quad (14)$$

Multi-frontal solution algorithms utilize direct solvers for the interface solution in Equation (14) whereas Primal Substructuring Methods uses iterative solvers such as preconditioned conjugate gradient (PCG) method.

2.2.3.1.2. *Definition with Dual Variables*

Another way of defining the domain decomposition problem above is choosing a set of interface forces that satisfies the interface equilibrium $L^T \mathbf{g} = 0$ a priori while redundant interface DOFs are stored in \mathbf{u} . By utilizing Equation (6), these interface forces can be expressed as

$$\mathbf{g}^{(s)} = -B^{(s)T} \boldsymbol{\lambda} \text{ or } \mathbf{g} = -B^T \boldsymbol{\lambda} \quad (15)$$

$\boldsymbol{\lambda}$ values are interface forces that act in opposite directions between any pair of conforming DOFs on the interface and thus they are in equilibrium as it is shown in Figure 2.2. Equation (4) becomes

$$\begin{cases} K\mathbf{u} + B^T \boldsymbol{\lambda} = \mathbf{f}, \\ B\mathbf{u} = 0, \end{cases} \quad (16)$$

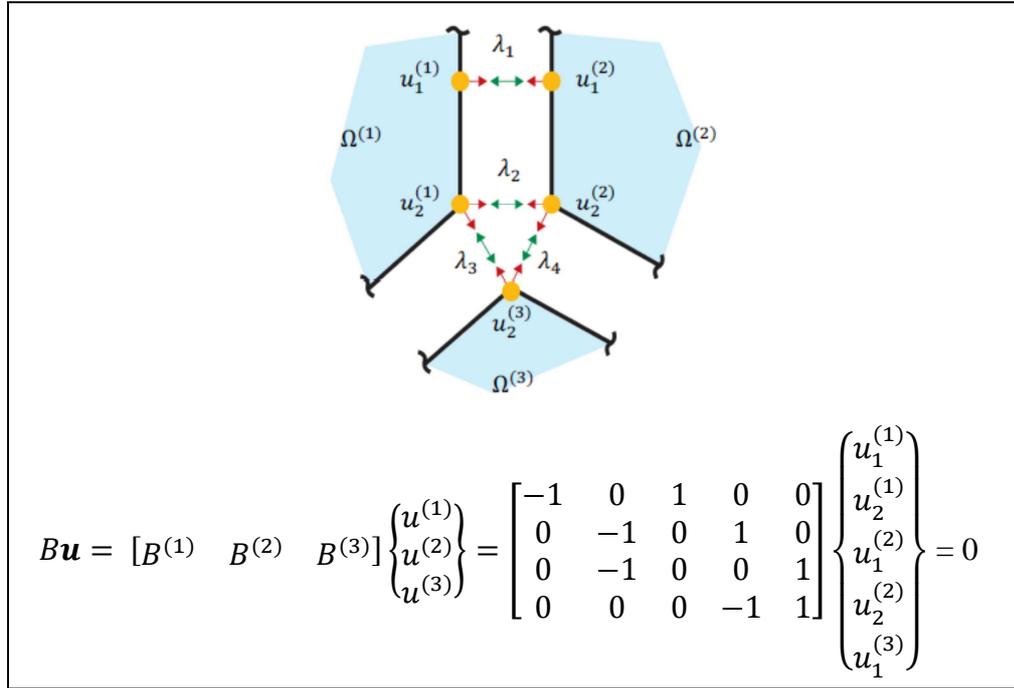


Figure 2.2. Interface compatibility

The Lagrange multipliers related to the interface compatibility constraints are shown as λ values. Presented decomposed problem formation is utilized for dual variable solution methods such as FETI. In these methods, iterative algorithms are utilized to compute interface forces λ such that the displacements resulting from the subdomain equilibrium satisfy the compatibility condition on the interface. Solution methods with dual variables require the solution of local subdomain problems of the form

$$K^{(s)}\mathbf{u}^{(s)} = \mathbf{f}^{(s)} - B^{(s)T}\boldsymbol{\lambda} \quad s = 1, \dots, N_s \quad (17)$$

where $K^{(s)}$ is the stiffness matrices of subdomains. Equation (17) may be gathered to form one matrix equation using block diagonal notation;

$$\begin{aligned} K\mathbf{u} &= \mathbf{f} - B^T\boldsymbol{\lambda} \\ &= \begin{bmatrix} K^{(1)} & & \\ & \ddots & \\ & & K^{(N_s)} \end{bmatrix} \begin{Bmatrix} \mathbf{u}^{(1)} \\ \vdots \\ \mathbf{u}^{(N_s)} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}^{(1)} \\ \vdots \\ \mathbf{f}^{(N_s)} \end{Bmatrix} - \begin{bmatrix} B^{(1)T} \\ \vdots \\ B^{(N_s)T} \end{bmatrix} \boldsymbol{\lambda} \end{aligned} \quad (18)$$

The solution of Equation (17) may have two different variations depending on the existence of zero energy modes in the subdomains. When the external constraints of a subdomain prevent all possible zero energy motions, such a subdomain is called non-floating and the solution of Equation (17) is

$$\mathbf{u}^{(s)} = K^{(s)-1}(\mathbf{f}^{(s)} - B^{(s)T} \boldsymbol{\lambda}) \quad (19)$$

Otherwise, subdomain is a floating subdomain and Equation (17) is solvable if the loads $\mathbf{f}^{(s)} - B^{(s)T} \boldsymbol{\lambda}$ are self-equilibrated; this condition can be represented as;

$$R^{(s)T}(\mathbf{f}^{(s)} - B^{(s)T} \boldsymbol{\lambda}) = 0 \quad (20)$$

where matrix $R^{(s)}$ stores the zero energy mode vectors of subdomain s . If the condition stated in Equation (20) holds, then the general solution of Equation (17) is given by

$$\mathbf{u}^{(s)} = K^{(s)+}(\mathbf{f}^{(s)} - B^{(s)T} \boldsymbol{\lambda}) + R^{(s)} \boldsymbol{\alpha}^{(s)} \quad (21)$$

where $K^{(s)+}$ is a generalized inverse of $K^{(s)}$ and a $\boldsymbol{\alpha}^{(s)}$ is a vector of arbitrary entries that represent the amplitudes of the zero energy modes. Rewriting Equations from (19) to (21) in block diagonal form as follows;

$$\begin{aligned} R^T(\mathbf{f} - B^T \boldsymbol{\lambda}) &= 0 \\ \mathbf{u} &= K^+(\mathbf{f} - B^T \boldsymbol{\lambda}) + R \boldsymbol{\alpha} \end{aligned} \quad (22)$$

where

$$K^+ = \begin{bmatrix} K^{(1)+} & & \\ & \ddots & \\ & & K^{(N_s)+} \end{bmatrix}, R = \begin{bmatrix} R^{(1)} & & \\ & \ddots & \\ & & R^{(N_s)} \end{bmatrix} \text{ and } \boldsymbol{\alpha} = \begin{Bmatrix} \boldsymbol{\alpha}^{(1)} \\ \vdots \\ \boldsymbol{\alpha}^{(N_s)} \end{Bmatrix}$$

For subdomains that are sufficiently supported (non-floating), submatrices $K^{(s)+}$ of K^+ are substituted by $K^{(s)-1}$ and $R^{(s)}$ and $\boldsymbol{\alpha}$ are modified to satisfy Equation (19) and Equation (21) consequently.

In domain decomposition methods, the equations associated with the internal DOFs of the subdomains are eliminated first generally. Hence, Equation (17) is rewritten as

$$S^{(s)} \mathbf{u}_b^{(s)} = \tilde{\mathbf{f}}_b^{(s)} - B_b^T \boldsymbol{\lambda} \quad (23)$$

where

$$S^{(s)} = K_{bb}^{(s)} - K_{bi}^{(s)} K_{ii}^{(s)-1} K_{ib}^{(s)}, \quad s = 1, \dots, N_s$$

and

$$\tilde{\mathbf{f}}_b^{(s)} = \mathbf{f}_b^{(s)} - K_{bi}^{(s)} K_{ii}^{(s)-1} \mathbf{f}_i^{(s)}$$

Subscripts i and b symbolize the restriction of the variables to internal and interface DOFs, respectively. $S^{(s)}$ is a condensed stiffness matrix, also called Schur complement [75].

Similarly to Equations (22), Equation (23) is solvable under the condition

$$R_b^{(s)T} (\tilde{\mathbf{f}}_b^{(s)} - B_b^T \boldsymbol{\lambda}) = 0 \quad (24)$$

and it is expressed as

$$\mathbf{u}_b^{(s)} = S^{(s)+} (\hat{\mathbf{f}}_b^{(s)} - B_b^T \boldsymbol{\lambda}) + R_b^{(s)} \boldsymbol{\alpha} \quad (25)$$

where $R_b^{(s)}$ designates the restriction of matrix $R^{(s)}$ to the interface DOFs

Solution methods with dual variables such as FETI-1 and FETI-2 will be discussed in Sections 2.2.4.1 and 2.2.4.2, respectively.

2.2.3.1.3. Hybrid Definition with Primal and Dual Variables

Third method that can be preferred for the solution of domain decomposition problem in Equation (4) is to choose interface displacements that are unique on part of the interface, whereas equilibrated connecting forces are defined on the remainder of the interface. This type of approaches named as hybrid dual/primal such as the FETI-DP solution algorithm [23]. FETI-DP procedure will be discussed in Section 2.2.4.3 in detail.

2.2.4. Overview on FETI Family Solution Methods

The FETI family of solution methods [21, 76] and related BDD methods [73] are among the first non-overlapping domain decomposition methods that have achieved numerical scalability with respect to both the mesh and subdomain sizes, for both second-order elasticity [77] and fourth-order plate and shell problems [22, 78]. Especially, the parallel scalability of the FETI method and its ability to outperform several popular direct and iterative solution algorithms on both sequential and parallel computers is comprehensively presented [42, 79].

2.2.4.1. One-Level FETI Method (FETI-1)

The FETI method [21, 73] is a Lagrange multiplier based domain decomposition method. As its name implies it composed of tearing the solution domain into subdomains and then interconnecting them by utilizing Lagrange multipliers. It utilizes preconditioned conjugate projected gradient (PCPG) iterative solution method in order to compute the dual variables (Lagrange multipliers, λ) defined in Section 2.2.3.1.2. Once λ is determined, the subdomain displacements can be recovered by solving the equilibrium equations;

$$\mathbf{u}^{(s)} = K^{(s)+} (\mathbf{f}^{(s)} - B^{(s)T} \lambda) + R^{(s)} \alpha^{(s)} \quad (26)$$

where $K^{(s)+}$ designates the inverse of $K^{(s)}$ if $\Omega^{(s)}$ has sufficiently supported to prevent $K^{(s)}$ from being singular, or a generalized inverse of $K^{(s)}$ if $\Omega^{(s)}$ is a floating subdomain, in other words partially supported or totally unsupported domains. In the latter case, the columns of $R^{(s)}$ contains the rigid body (or more generally zero energy) modes of $\Omega^{(s)}$, i.e. $R^{(s)} = \text{null}(K^{(s)})$ and $\boldsymbol{\alpha}^{(s)}$ is the set of amplitudes that represents the contribution of the null space $R^{(s)}$ to the solution $\mathbf{u}^{(s)}$.

These coefficients can be computed if and only if each subdomain problem is mathematically solvable—that is, each floating subdomain is self-equilibrated—which can be expressed by

$$R^{(s)T}(\mathbf{f}^{(s)} - B^{(s)T}\boldsymbol{\lambda}) = 0 \quad (27)$$

Substituting Equation (26) into the compatibility equation and exploiting the solvability condition, Equation (27), transforms problem (18) into the interface problem

$$\begin{bmatrix} F_I & -G_I \\ -G_I^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \mathbf{d} \\ -\mathbf{e} \end{bmatrix} \quad (28)$$

where

$$\begin{aligned} F_I &= \sum_{s=1}^{N_s} B^{(s)} K^{(s)+} B^{(s)T}, \\ \mathbf{d} &= \sum_{s=1}^{N_s} B^{(s)} K^{(s)+} \mathbf{f}^{(s)}, \\ G_I &= [B^{(s)} R^{(s)} \quad \dots \quad B^{(N_f)} R^{(N_f)}], \\ \boldsymbol{\alpha} &= [\boldsymbol{\alpha}^{(1)T} \quad \dots \quad \boldsymbol{\alpha}^{(N_f)T}]^T, \\ \mathbf{e} &= [\mathbf{f}^{(1)T} R^{(1)} \quad \dots \quad \mathbf{f}^{(N_f)T} R^{(N_f)}]^T \end{aligned}$$

and where N_f denotes the number of floating subdomains. Since $\boldsymbol{\lambda}$ is a dual variable to the primal unknowns $\mathbf{u}^{(s)}$ of the system, the interface problem, Equation (28), is called a dual interface problem. The interface problem is comparatively smaller than

the total number of DOFs and in the original FETI method [21], the interface problem is solved by a preconditioned conjugate projected gradient (PCPG) algorithm. For this purpose, the indefinite interface problem in Equation (28) is transformed into a semi-definite system of equations by eliminating the self-equilibrium condition $G_I^T \boldsymbol{\lambda} = \mathbf{e}$ using the splitting;

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}^0 + P(Q)\bar{\boldsymbol{\lambda}} \quad (29)$$

where $\boldsymbol{\lambda}^0$ is a particular solution of $G_I^T \boldsymbol{\lambda} = \mathbf{e}$ of the form

$$\boldsymbol{\lambda}^0 = QG_I(G_I^T QG_I)^{-1}\mathbf{e} \quad (30)$$

and $P(Q)$ is a projection operator and it is a function of a given matrix Q by

$$P(Q) = I - QG_I(G_I^T QG_I)^{-1}G_I^T \quad (31)$$

where for any matrix Q ,

$$P^2(Q) = P(Q) \text{ and } G_I^T P(Q) = 0 \quad (32)$$

equalities are satisfied [74]. Throughout the remainder of this study, the projector presented in Equation (31) will be simply denoted as P . Performing the splitting Equation (29) on the interface problem, Equation (28) yields the following projected interface problem;

$$(P^T F_I P)\bar{\boldsymbol{\lambda}} = P^T(\mathbf{d} - F_I \boldsymbol{\lambda}^0) \quad (33)$$

The projected interface problem is semi-positive and semi-definite [21, 74] for any given matrix Q . Hence, Preconditioned Conjugate Gradient (PCG) algorithm can be utilized to solve it. Iterating directly with $\mathbf{y}^k = P\bar{\mathbf{y}}^k$ and $\mathbf{p}^k = P\bar{\mathbf{p}}^k$, and exploiting

the properties mentioned above lead to the well-known FETI-1 PCPG iterative solution [21] presented in Table 2.1.

In this algorithm, \tilde{F}_I^{-1} designates a chosen preconditioner (several of them are presented in Section 2.2.4.4). This solution algorithm has been named as “one-level” FETI algorithm in order to separate it from the two-level FETI algorithm [22, 79, 80] that is presented in Section 2.2.4.2.

Table 2.1. Pseudocode for FETI-1 PCPG algorithm

Pseudocode: PCPG Solution Algorithm for FETI-1

Initialize:

$$\boldsymbol{\lambda}^0 = \mathbf{Q}\mathbf{G}_I(\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I)^{-1}\mathbf{e}$$

$$\mathbf{w}^0 = \mathbf{P}^T(\mathbf{d} - \mathbf{F}_I\boldsymbol{\lambda}^0)$$

for $k = 0, 1, \dots$ until convergence

$$\mathbf{y}^k = \mathbf{P}\tilde{\mathbf{F}}_I^{-1}\mathbf{w}^k$$

$$\mathbf{p}^k = \mathbf{y}^k - \sum_{i=0}^{k-1} \frac{\mathbf{y}^{kT}\mathbf{F}_I\mathbf{p}^i}{\mathbf{p}^{iT}\mathbf{F}_I\mathbf{p}^i} \mathbf{p}^i$$

$$\eta^k = \frac{\mathbf{p}^{kT}\mathbf{w}^k}{\mathbf{p}^{kT}\mathbf{F}_I\mathbf{p}^k}$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \eta^k\mathbf{p}^k$$

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \eta^k\mathbf{P}^T\mathbf{F}_I\mathbf{p}^k$$

end for

Finally, computation of null space $R^{(s)}$ (or zero energy mode computation of subdomain s) is a discrete computation apart from the algorithm of FETI. In case of solid subdomains without any internal mechanisms, computation of zero energy modes (i.e. "rigid body modes" for this case) is a straightforward task [33, 37]. Even, for subdomains having internal mechanisms composed of mechanism free subparts, algebraic/geometric methodologies can be utilized [38, 81]. However, in case of displacement based fluid elements, these methodologies cannot be applicable [59, 62] and direct computation of null spaces is required and it is a computationally expensive procedure that utilizes either Singular Value Decomposition or a full rank revealing QR decomposition [82].

2.2.4.2. Two-Level FETI Method (FETI-2)

Since one-level FETI solution algorithm is not numerically scalable for fourth-order elasticity problems [22, 24], a numerically scalable extension of the FETI method is presented by enforcing the continuity of the displacements at the subdomain cross points throughout the PCPG iterations. Moreover, such a constraint can be prescribed by solving another auxiliary coarse problem that contains not only the subdomain zero energy modes as in the original FETI method, but also the so-called subdomain corner modes [22, 24]. This enriched coarse problem, that converts the original FETI method into a two-level algorithm, has produced an even more powerful FETI method known as the two-level FETI method [22, 79, 80]. The two-level FETI method is presented in [80] as a one-level FETI PCPG algorithm where an optional admissible constraint of the form;

$$C^T \mathbf{w}^k = 0 \quad (34)$$

is enforced at each iteration k . Matrix C is rectangular and demonstrates some subspace to be determined, and $\mathbf{w}^k = \mathbf{P}^T(\mathbf{d} - \mathbf{F}_I \boldsymbol{\lambda}_k)$ (see FETI-1 PCPG Algorithm in Table 2.1). Iterative solution of two-level dual interface problem initializes with the enriched condition of $\overline{\boldsymbol{\lambda}}_0 = 0$ by the quantity $C\boldsymbol{\gamma}^{init}$ so that $\boldsymbol{\lambda}_0$ obtained as;

$$\boldsymbol{\lambda}_0 = QG_I(G_I^T QG_I)^{-1} \mathbf{e} + PC\boldsymbol{\gamma}^{init} \quad (35)$$

Using the second condition in Equation (32), it can be checked that the above starting value satisfies the necessary condition $G_I^T \boldsymbol{\lambda}_0 = \mathbf{e}$. The value of $\boldsymbol{\gamma}^{init}$ is obtained with the following minimization;

$$(C^T P^T F_I P C) \boldsymbol{\gamma}^{init} = C^T P^T (\mathbf{d} - F_I Q G_I (G_I^T Q G_I)^{-1}) \quad (36)$$

at each iteration k , a correction parameter originated from the second constraint $\boldsymbol{\gamma}^k$ can be computed as;

$$(C^T P^T F_I P C) \gamma^k = -C^T P^T F_I P p^k \quad (37)$$

Since p^k is generated by $y^k = P F_I w^k$ and by using the first of Equation (32), the right hand side of Equation (37) can be replaced with the following;

$$-C^T P^T F_I P p^k = -C^T P^T F_I p^k \quad (38)$$

Utilizing the concepts presented above, PCPG algorithm obtained for two-level FETI [80] is presented in Table 2.2.

Table 2.2. Pseudocode for FETI-2 PCPG algorithm

Pseudocode: PCPG Solution Algorithm for FETI-2

Initialize:

$$\text{Solve; } (C^T P^T F_I P C) \gamma^{init} = C^T P^T (d - F_I Q G_I (G_I^T Q G_I)^{-1} e)$$

$$\lambda^0 = Q G_I (G_I^T Q G_I)^{-1} e + P C \gamma^{init}$$

$$w^0 = P^T (d - F_I \lambda^0)$$

for $k = 0, 1, \dots$ until convergence

$$y^k = P \bar{F}_I^{-1} w^k$$

$$p^k = y^k - \sum_{i=0}^{k-1} \frac{y^{kT} F_I \hat{p}^i}{\hat{p}^{iT} F_I \hat{p}^i} \hat{p}^i$$

$$\text{Solve; } (C^T P^T F_I P C) \gamma^k = -C^T P^T F_I p^k$$

$$\hat{p}^k = p^k + P C \gamma^k$$

$$\eta^k = \frac{\hat{p}^{kT} w^k}{\hat{p}^{kT} F_I \hat{p}^k}$$

$$\lambda^{k+1} = \lambda^k + \eta^k \hat{p}^k$$

$$w^{k+1} = w^k - \eta^k P^T F_I \hat{p}^k$$

end for

To sum up, at each iteration k ; $u^{(s)k}$ is evaluated by substituting in Equation (16) the Lagrange multipliers λ^k generated by the PCPG algorithm. Hereafter, at each iteration k , the subdomains are in equilibrium, but the jump of the subdomain displacement w^k iterates across the subdomain interfaces is not necessarily zero. This jump vanishes in the usual numerical sense only at convergence. However, at each FETI-2 iteration, the constraint $C^T w^k = 0$ forces some prescribed components of the subdomain displacements to be continuous across the interfaces. As a result,

although the computational cost increased for each iteration, in general, the overall convergence of two-level FETI algorithm is accelerated [22, 83].

2.2.4.3. Dual-Primal FETI Method (FETI-DP)

As a dual-primal method, in addition to the concepts discussed for the dual methods FETI-1 and FETI-2, FETI-DP additionally utilizes primal variables. In the dual-primal FETI methods [23, 83–85], dual and primal variables are distinguished according to the way of defining the continuity of the solution in those variables. Dual displacement variables are those, for which the continuity is enforced by a continuity constraint and Lagrange multipliers λ and thus, continuity is not established until convergence of the iterative method is reached similar to the classical FETI-1 method [23, 27, 84];

On the other hand, continuity of the primal displacement variables is explicitly enforced at each iteration step by the subassembly of the local stiffness matrices $K^{(s)}$ of neighbor subdomains at the primal displacement variables. This subassembly produces a symmetric, positive definite stiffness matrix K which is not block diagonal but is coupled at the primal displacement variables. This coupling forms a global problem which is necessary to attain a numerically scalable algorithm [24, 83].

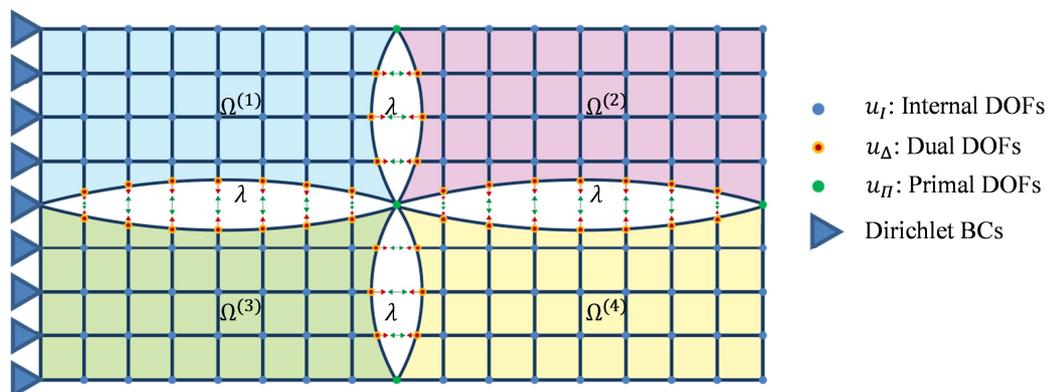


Figure 2.3. Variable types of dual-primal system definition

Let's use the subscripts I , Δ , and Π presented in Figure 2.3, to denote the internal, dual, and primal displacement variables, respectively, and obtain for the local stiffness matrices, load vectors, and solution vectors of nodal values;

$$K^{(s)} = \begin{bmatrix} K_{II}^{(s)} & K_{\Delta I}^{(s)} & K_{\Pi I}^{(s)} \\ K_{\Delta I}^{(s)} & K_{\Delta\Delta}^{(s)} & K_{\Pi\Delta}^{(s)} \\ K_{\Pi I}^{(s)} & K_{\Pi\Delta}^{(s)} & K_{\Pi\Pi}^{(s)} \end{bmatrix}, \mathbf{u}^{(s)} = \begin{bmatrix} \mathbf{u}_I^{(s)} \\ \mathbf{u}_\Delta^{(s)} \\ \mathbf{u}_\Pi^{(s)} \end{bmatrix}, \mathbf{f}^{(s)} = \begin{bmatrix} \mathbf{f}_I^{(s)} \\ \mathbf{f}_\Delta^{(s)} \\ \mathbf{f}_\Pi^{(s)} \end{bmatrix} \quad (39)$$

By introducing the following notation;

$$\begin{aligned} \mathbf{u}_B &= [\mathbf{u}_I \quad \mathbf{u}_\Delta]^T, \mathbf{f}_B = [\mathbf{f}_I \quad \mathbf{f}_\Delta]^T, \mathbf{u}_B^{(s)} \\ &= [\mathbf{u}_I^{(s)} \quad \mathbf{u}_\Delta^{(s)}]^T, \text{ and } \mathbf{f}_B^{(s)} = [\mathbf{f}_I^{(s)} \quad \mathbf{f}_\Delta^{(s)}]^T \end{aligned} \quad (40)$$

and accordingly;

$$\begin{aligned} K_{BB} &= \text{diag}_{s=1}^{N_s} (K_{BB}^{(s)}), K_{BB}^{(s)} = \begin{bmatrix} K_{II}^{(s)} & K_{\Delta I}^{(s)} \\ K_{\Delta I}^{(s)} & K_{\Delta\Delta}^{(s)} \end{bmatrix}, K_{\Pi B} \\ &= [K_{\Pi B}^{(1)} \quad \dots \quad K_{\Pi B}^{(N)}] \end{aligned} \quad (41)$$

where K_{BB} is a block diagonal matrix. By subassembly in the primal displacement variables, coefficient matrix can be obtained as;

$$\tilde{K} = \begin{bmatrix} K_{BB} & \tilde{K}_{\Pi B}^{(s)} \\ \tilde{K}_{\Pi B}^{(s)} & \tilde{K}_{\Pi\Pi}^{(s)} \end{bmatrix} \quad (42)$$

where a tilde indicates the subassembled matrices and where

$$\tilde{K}_{\Pi B} = [\tilde{K}_{\Pi B}^{(1)} \quad \dots \quad \tilde{K}_{\Pi B}^{(N_s)}] \quad (43)$$

Introducing local assembly operators $L_{\Pi}^{(s)}$ which map from the local primal displacement variables $\mathbf{u}_{\Pi}^{(s)}$ to the global, assembled $\tilde{\mathbf{u}}_{\Pi}^{(s)}$, following definition is obtained;

$$\begin{aligned}\tilde{K}_{\Pi B}^{(s)} &= L_{\Pi}^{(s)} K_{\Pi B}^{(s)}, \tilde{\mathbf{u}}_{\Pi}^{(s)} = \sum_{s=1}^{N_s} L_{\Pi}^{(s)} \mathbf{u}_{\Pi}^{(s)}, \tilde{K}_{\Pi \Pi}^{(s)} \\ &= \sum_{s=1}^{N_s} L_{\Pi}^{(s)} K_{\Pi \Pi}^{(s)} L_{\Pi}^{(s)T}\end{aligned}\quad (44)$$

for $s = 1, \dots, N_s$. Due to the subassembly of the primal displacement variables, Lagrange multipliers have to be used only for the dual displacement variables \mathbf{u}_{Δ} to enforce continuity. A discrete jump operator B is introduced such that the solution \mathbf{u}_{Δ} , associated with more than one subdomain, coincides when $B\mathbf{u}_B = 0$; the internal variables \mathbf{u}_I remain unchanged and thus the corresponding entries in B remain zero. Since the grids are assumed to be continuous across the interface Γ , matrix B is a Boolean matrix with entries of 0, 1, and -1 [74].

Reformulating the finite element discretization of Equation (42) yields;

$$\begin{bmatrix} K_{BB} & \tilde{K}_{\Pi B}^{(s)} & B^T \\ \tilde{K}_{\Pi B}^{(s)} & \tilde{K}_{\Pi \Pi}^{(s)} & 0 \\ B & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_B^{(s)} \\ \mathbf{u}_{\Pi}^{(s)} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_B^{(s)} \\ \mathbf{f}_{\Pi}^{(s)} \\ \mathbf{0} \end{bmatrix}\quad (45)$$

Elimination of the primal variables $\tilde{\mathbf{u}}_{\Pi}$ and the internal and dual displacement variables \mathbf{u}_B leads to a reduced linear system of the form;

$$F\boldsymbol{\lambda} = \mathbf{d}\quad (46)$$

where the coefficient matrix F and the right hand side vector \mathbf{d} are formally obtained by block Gauss elimination. Here, the matrix F is never built explicitly [83, 86] but, in each iteration, appropriate linear systems are solved under the constraints of selected primal variables by utilizing a Krylov subspace solver with preconditioning [87, 88].

The selection of primal variables, also named as corners in literature, should satisfy two essential conditions. Former one is that none of the subdomain stiffness matrices should be singular. Latter one is that the resulting coefficient matrix for the coarse problem should not be singular. In addition to these essential conditions, keeping the number of primal variables low reduces the overall cost of the computation and improves its scalability [89].

In the literature, a straightforward algorithm that guarantees the non-singularity of each subdomain $K_{BB}^{(s)}$ is making sure that every subdomain has either 3 non-collinear corner nodes in 3D or 2 non-coincidental corner nodes in 2D [33, 37]. However this is true only for problems with homogeneous material properties in solid mechanics, since in case of Lagrangian fluid finite elements there are internal mechanisms in the elements [59, 62]. Because of the spurious modes in pure displacement based fluid formulation or by nearly incompressible condition for both fluid formulations, problems may become ill-conditioned. Therefore, in these cases, convergence rate and efficiency of iterative solution methods decrease [90, 91].

2.2.4.4. Preconditioners

One of the most important subjects in order to obtain efficient iterative methods is utilizing well-posed preconditioners for the problem. Although numerous preconditioners are proposed in literature [35, 66, 92–95], subdomain-based preconditioners such as Dirichlet, lumped and super-lumped preconditioners are the mostly utilized preconditioners due to efficiency and parallel scalability [74].

It is now a well-known fact that the Dirichlet preconditioner guarantees scalability with respect to the mesh size h for most of the FETI family methods. Lumped preconditioner is a more economical version of the Dirichlet preconditioner that, for homogeneous second-order elasticity problems, supplies a scalable computational performance [74, 86, 92].

2.2.4.4.1. Dirichlet Preconditioner:

A numerically scalable preconditioner can be written as;

$$\bar{F}_I^{D^{-1}} = B_{p_b} S^{(s)} B_{p_b}^{-1} \quad (47)$$

where $S^{(s)} = K_{bb}^{(s)} - K_{bi}^{(s)} K_{ii}^{(s)^{-1}} K_{ib}^{(s)}$ and the subscripts i and b denote the internal and interface boundary unknowns, respectively. Operator B_{p_b} is similar to B_b defined in Section 2.2.3.1.2 but includes a scaling that is obtained from interface multiplicity or relative interface stiffness [86]. The Dirichlet preconditioner approximates the inverse of the sum (sum over the substructures) by the sum of the inverses. Although it is mechanically consistent and numerically scalable, it is expensive to form the condensed stiffness of each subdomain [74].

2.2.4.4.2. Lumped Preconditioner

Lumped preconditioner lies on the same mechanical interpretation as the Dirichlet preconditioner, except that in this case all the stiffness of a subdomain is lumped at its interface DOFs. Therefore it can be formulated as follows;

$$\bar{F}_I^{L^{-1}} = B_{p_b} K_{bb}^{(s)} B_{p_b}^{-1} \quad (48)$$

Lumped preconditioner is more economical than the Dirichlet preconditioner and has proved to be more efficient generally for second-order elasticity problems [77]. However, the Dirichlet preconditioner is more efficient for fourth-order plate and shell problems [22, 24].

2.2.4.4.3. Super-Lumped Preconditioner

A smoothed version of lumped preconditioner by the stiffness of neighbor subdomains is called the super-lumped preconditioner and represented as follows;

$$\bar{F}_I^{SL-1} = B_{p_b} \text{diag}(K_{bb}^{(s)}) B_{p_b}^{-1} \quad (49)$$

which means for each interface DOF, the assembled stiffness is computed by gathering the diagonal stiffness coefficients from all neighboring DOFs [92].

2.2.4.5. Extension to Dynamic Analysis

Using a standard Galerkin procedure where the displacement field is approximated by suitable shape functions and the equations of dynamic equilibrium are linearized around \mathbf{u}^{n+1} and following Differential/Algebraic equations;

$$M^{(s)} \ddot{\mathbf{u}}^{(s)} + C^{(s)} \dot{\mathbf{u}}^{(s)} + K^{(s)} \mathbf{u}^{(s)} = \mathbf{F}^{(s)} - B^{(s)T} \boldsymbol{\lambda} \quad s = 1, \dots, N_s \quad (50)$$

subject to;

$$\sum_{s=1}^{N_s} B^{(s)} \mathbf{u}^{(s)} = 0 \quad \text{or} \quad \sum_{s=1}^{N_s} B^{(s)} \dot{\mathbf{u}}^{(s)} = 0 \quad \text{or} \quad \sum_{s=1}^{N_s} B^{(s)} \ddot{\mathbf{u}}^{(s)} = 0$$

are obtained. $M^{(s)}$ and $K^{(s)}$ are, respectively, the subdomain mass and stiffness matrices, $\mathbf{F}^{(s)}$ is the subdomain vector of prescribed forces and $B^{(s)}$ are the subdomain constraint matrices. Dynamic equilibrium equations are generally solved by Newmark trapezoidal time integration ($\beta = \frac{1}{4}, \gamma = \frac{1}{4}$). Farhat and Crivelli [43] showed that when the displacement constraints $\sum_{s=1}^{N_s} B^{(s)} \mathbf{u}^{(s)} = 0$ are enforced, the stability of the Newmark trapezoidal integrator is lost for any time step value, Δt due to the dynamics of the dual variables (Lagrange multipliers). Therefore, they proposed to restore unconditional stability by bypassing the dynamics of these multipliers.

By denoting $\mathbf{v}^{(s)}$ and \mathbf{v} as the subdomain and global momentum variables, respectively, thus;

$$\mathbf{v}^{(s)} = M^{(s)}\dot{\mathbf{u}}^{(s)} \quad s = 1, \dots, N_s, \quad (51)$$

and

$$\mathbf{v} = M\dot{\mathbf{u}} \quad (52)$$

Consider Equations (50) with the displacement constraint equations $\sum_{s=1}^{N_s} B^{(s)}\mathbf{u}^{(s)} = 0$. Using the Newmark trapezoidal integration ($\beta = \frac{1}{4}$ and $\gamma = \frac{1}{4}$), and by ignoring damping terms for simplicity, the dynamics of each subdomain can be independently integrated as it is presented in Table 2.3.

Table 2.3. Pseudocode for Extension of FETI Methods to Dynamics

Pseudocode: Extension of FETI methods to Dynamics

for $n = 0, \Delta t, \dots$ until the end of dynamic analysis duration

Solve;

$$\left(\frac{4}{\Delta t^2}M^{(s)} + K^{(s)}\right)\mathbf{u}^{(s)n+\frac{1}{2}} = F^{(s)n+\frac{1}{2}} + \frac{4}{\Delta t^2}M^{(s)}\mathbf{u}^{(s)n} + \frac{2}{\Delta t}\mathbf{v}^{(s)n} - B^{(s)T}\boldsymbol{\lambda} \quad s = 1, \dots, N_s$$

subject to; $\sum_{s=1}^{N_s} B^{(s)}\mathbf{u}^{(s)} = 0$

$$\dot{\mathbf{v}}^{(s)n+\frac{1}{2}} = F^{(s)n+\frac{1}{2}} - K^{(s)}\mathbf{u}^{(s)n+\frac{1}{2}}$$

$$\mathbf{v}^{(s)n+1} = \mathbf{v}^{(s)n} - \Delta t\dot{\mathbf{v}}^{(s)n+\frac{1}{2}} \quad \text{or} \quad \mathbf{v}^{(s)n+1} = \mathbf{v}^{(s)n+\frac{1}{2}} - \frac{\Delta t}{2}\dot{\mathbf{v}}^{(s)n+\frac{1}{2}}$$

$$\mathbf{u}^{(s)n+1} = 2\mathbf{u}^{(s)n+\frac{1}{2}} - \mathbf{u}^{(s)n}$$

end for

Computing the momentum from the assembled equations of motion rather than from the subdomain equations of dynamic equilibrium, Equation (50), removes the dynamics of $\boldsymbol{\lambda}$ from the system. The above time integration algorithm is unconditionally stable [43]. It is important to note that unconditional stability is restored, neither by using the midpoint rule, nor the momentum formulation, but by bypassing the dynamics of the Lagrange multipliers with replacing the accelerations with momentums. The formulation with momentum variables allows the evaluation of both $\mathbf{u}^{(s)n+1}$ and $M^{(s)}\dot{\mathbf{u}}^{(s)n+1}$ from the equations of dynamic equilibrium without having to assemble and factor M , and also it eliminates the need for computing the

acceleration terms from the differentiation of the displacement and velocity fields, which is known to cause an oscillatory behavior of the acceleration $\ddot{\mathbf{u}}$ [96].

Returning back to system of Differential/Algebraic equations in Equation (50), it can be rewritten as;

$$\widehat{K}^{(s)}\mathbf{u}^{(s)} = \widehat{\mathbf{f}}^{(s)} \quad (53)$$

under the same constraints and $\widehat{K}^{(s)}$, $\mathbf{u}^{(s)}$ and $\widehat{\mathbf{f}}^{(s)}$ designates the effective stiffness matrix, displacements and effective forces for the dynamic analysis case. In case of a standard Newmark scheme, the solution of dynamic equilibrium equations takes the form;

$$\left(\frac{1}{\beta\Delta t^2} M^{(s)} + \frac{\gamma}{\beta\Delta t} C^{(s)} + K^{(s)} \right) {}^{t+\Delta t}\mathbf{u}^{(s)} = {}^{t+\Delta t}\mathbf{f}^{(s)} \quad (54)$$

where t and Δt are the current time and the time increment, respectively, while $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$ for generally used Newmark trapezoidal integration. Therefore, by comparison;

$$\widehat{K}^{(s)} = \frac{1}{\beta\Delta t^2} M^{(s)} + \frac{\gamma}{\beta\Delta t} C^{(s)} + K^{(s)} \quad (55)$$

where $C^{(s)}$ and $M^{(s)}$ are the damping and mass matrices of each subdomain. The effective stiffness matrix $\widehat{K}^{(s)}$ is usually equal to a linear combination of the mass, damping and stiffness matrices, while its exact expression depends on the adopted time integration scheme.

The addition of mass and damping terms to the stiffness matrix makes the resulting effective matrix $\widehat{K}^{(s)}$, considerably less ill-conditioned than the static stiffness matrix $K^{(s)}$ due to the fact that mass and damping terms in general remove the zero energy modes of subdomain stiffness matrices [43]. Therefore, iterative methods utilized for the solution of Equation (53) usually requires much less iterations than the corresponding problem $K^{(s)}\mathbf{u}^{(s)} = \mathbf{f}^{(s)}$ of static analysis.

This assumption is in fact correct, with one important exception: with regard to the solution of local subdomain problems discussed in Section 2.2.3.1.2, Equation (20) and (21) as well as Equations (24) and (25) do not make sense in the case of implicit dynamics because, contrary to a subdomain stiffness matrix $K^{(s)}$, matrix $\widehat{K}^{(s)}$ is in general positive definite and thus has an empty null space and no zero energy modes $R^{(s)}$. Therefore, for implicit dynamics Equations (21) and (25) must be replaced by following equations;

$$\mathbf{u}^{(s)} = \widehat{K}^{(s)-1}(\widehat{\mathbf{f}}^{(s)} - B^T \widehat{\boldsymbol{\lambda}}) \quad (56)$$

and

$$\mathbf{u}_b^{(s)} = \widehat{S}^{(s)-1}(\widehat{\mathbf{f}}_b^{(s)} - B_b^T \widehat{\boldsymbol{\lambda}}) \quad (57)$$

It should be considered that $K^{(s)}$ is substituted by the corresponding $\widehat{K}^{(s)}$, the generalized inverses $K^{(s)+}$ by the inverses $\widehat{K}^{(s)-1}$ and the forces $\mathbf{f}^{(s)}$ by the effective forces $\widehat{\mathbf{f}}^{(s)}$ of each step of the time integration scheme. Furthermore, the zero energy modes $R^{(s)}$ of the corresponding static problem $K^{(s)}\mathbf{u}^{(s)} = \mathbf{f}^{(s)}$ should still be used wherever $R^{(s)}$ appears [74].

2.2.4.5.1. Dynamic Analysis with FETI-1

Let Equation (21) be replaced by Equation (56), due to the absence of subdomain zero energy modes in the dynamic problem. If the same steps for the derivation of Equation (33) of FETI-1 (Section 2.2.4.1) are recomputed with Equation (56) instead of Equation (21), then the following equation is derived in block matrix notation as:

$$\widehat{F}_l \widehat{\boldsymbol{\lambda}} = \widehat{\mathbf{d}} \quad (58)$$

where $\widehat{\boldsymbol{\lambda}}$ are the dynamic Lagrange multipliers, while

$$\widehat{F}_l = B\widehat{K}^{-1}B^T \text{ and } \widehat{\mathbf{d}} = B\widehat{K}^{-1}\mathbf{f}^s \quad (59)$$

A comparison of Equation (33) to Equation (58) demonstrates that in the latter the projections and the initial Lagrange multiplier vector λ_0 have vanished. Shortly, the coarse problem of FETI-1 has been removed. This is due to the absence of subdomain zero energy modes in the dynamic case. In [43], Farhat and Crivelli utilized the PCG to solve Equation (58) with the preconditioners defined in Section 2.2.4.4 by replacing the static stiffness terms with the dynamics terms as in Equation (58).

2.2.4.5.2. *Dynamic Analysis with FETI-2*

The resulting dynamic problem defined in Equation (58) can be solved with FETI-2 solution methods by employing the matrix $G = BR^{(s)}$ as the constraint matrix C [97]. By following the reasoning in Section 2.2.4.2, these admissible constraints obtained from static case are utilized to form a projection on Equation (58) as follows;

$$(\hat{F}_I - \hat{F}_I C (C^T \hat{F}_I C)^{-1} C^T \hat{F}_I) \hat{\lambda} = (I - \hat{F}_I C (C^T \hat{F}_I C)^{-1} C^T) \hat{d} \quad (60)$$

And it can be solved by utilizing PCG iterations with the preconditioners defined in Section 2.2.4.4 by replacing the static stiffness terms with the dynamics terms as in Equation (58). Moreover, it was proven that if the time step $\Delta t \rightarrow \infty$, this method converges towards the FETI-1 method [98]. Besides, Fragakis and Papadrakakis are proposed to set $C = QG$, where matrix Q is set equal to one of the preconditioners defined in Section 2.2.4.4, depending on the type of problem under investigation [86].

CHAPTER 3

SERIAL IMPLEMENTATION OF SOLUTION FRAMEWORK

3.1. Introduction

In this chapter, the serial implementation of solution framework for seismic analysis of dam-reservoir systems is discussed. Mainly, solution relies on the FETI family solution methods utilized to solve the implicit monolithic problem represented by Lagrangian finite elements.

Serial implementation of solution framework on MATLAB (a commercial software package; MATLAB Release 2014a, The MathWorks, Inc., Natick, Massachusetts, United States) environment is presented and then the concerns about the implementation and convergence rate are discussed in detail. In addition to that, performance improvements by utilizing several methods that are already available in literature are investigated on this implementation. Validations of implemented finite elements and overall solution framework are presented in APPENDIX B.

3.2. Implementation

Solution framework mainly relies on the fully Lagrangian representation of whole dam system that involves dam structure, foundation and the reservoir. Dam and the foundation are generally modelled by linear Lagrangian solid finite elements [18]. During the Lagrangian representation of the reservoir, two fluid finite elements with different formulations presented in APPENDIX A are utilized. Pure displacement fluid formulations which are based on bulk modulus and utilize inviscid assumptions are frequently utilized on dam-reservoir problems. Since this formulation results in

spurious energy modes, reduced integration and penalty formulations are used to stabilize them for a wide range of frequencies [59]. In order to prevent spurious modes totally, fluid elements with mixed formulation by displacements and pressures are developed by Bathe [62]. In such formulations, pressure DOFs are condensed out during element stiffness formation hence it is possible to utilize these elements in fully Lagrangian solvers. Since the field variables of the solid and fluid domains are the same, there is no need for any coupling elements or methods in between subdomains with different media.

Monolithic time domain analysis requires large memory capacity and computational resources, hence, high performance computing techniques are generally preferred [10]. In this serial implementation on MATLAB environment, domain decomposition based high performance solution method, FETI family methods [25] are implemented. Since these solution methods are iterative, in general, specially chosen preconditioners are utilized to satisfy carefully selected termination criteria. Coarse problem defined in the FETI-1 method (Section 2.2.4.1) vanishes in case of time history analysis of a structure (Section 2.2.4.5.1). Although it is not crucial for the solution, it is the main cause of the convergence rate and global error propagation of these solution methods [86]. Therefore another auxiliary coarse problem is defined in FETI-2 by utilizing the admissible constraints on interface (Section 2.2.4.5.2). However, this solution method produces a diverging trend called spectral drift when it is utilized with standard Newmark integration method, hence, a special consideration (Section 2.2.4.5) is implemented to extend the FETI solution methods to dynamics.

3.2.1. Finite Element Implementations

Both formulations presented in APPENDIX A are implemented for two and three dimensional finite elements with bilinear and quadratic approximations. At this point, it is beneficial to define a naming convention for finite elements that are

mentioned. xYNIPMZz mapping is defined to name all finite elements used in this study where

- x; the media (“s” is for solid and “f” for fluid media),
- Y; solution space (“Q” for Quadrilateral, “H” for Hexahedron),
- N; number of displacement nodes,
- I; optional designation for incompatible modes,
- P; shows up for all as a designation for pressure,
- M; number of pressure nodes,
- Zz; type of element formulation (“u” for pure displacement, “u/p” for displacement/pressure formulation).

As an example, “sQ4P0u” designates 4-node pure displacement formulation quadrilateral finite element for solid media. Similarly “fQ9P3u/p” designates 9-node displacement/pressure formulation finite element for fluid domain with 3 pressure nodes (Figure 3.1-a) and “fH27P4u/p” designates 27-node displacement/pressure formulation finite element for fluid domain with 4 pressure nodes (Figure 3.1-b).

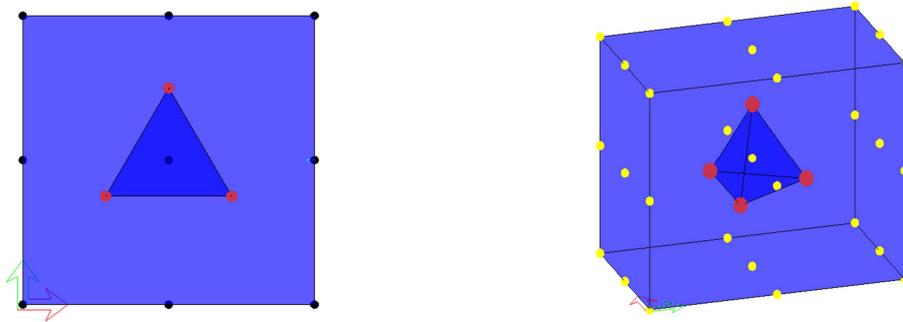


Figure 3.1. fQ9P3u/p and fH27P4u/p elements

Using this naming convention, the list of implemented finite elements is presented in Table 3.1. Note that, incompatible modes can only be defined for linear pure displacement approximation formulation of solid media. In addition to that mixed formulation can be applied for plane stain, general 3D and axisymmetric stress conditions [44, 62].

Table 3.1. List of implemented Finite Elements

Solid Finite Elements		Fluid Finite Elements	
sQ4P0u	sH8P0u	fQ4P1u	fH8P1u
sQ4IP0u	sH8IP0u	fQ9P1u	fH27P1u
sQ9P0u	sH27P0u	fQ4P1u/p	fH8P1u/p
sQ4P1u/p	sH8P1u/p	fQ9P3u/p	fH27P4u/p
sQ9P3u/p	sH27P4u/p		

3.2.2. Implementation for Static Analysis

For the serial implementation of the monolithic solution algorithm for static analysis of dam-reservoir systems presented in Figure 3.2, built-in functions and data types of MATLAB are employed. Algorithm initiates with reading the model input. Input file contains the active DOFs for the problem, position and restraint information of nodes, material definitions, discretization elements with their required properties and connectivity, and finally the external forces. Besides, multiple input files may be given to designate that problem is partitioned into subdomains. By using the information supplied for the nodes, nodes of each subdomain is sorted as the internal DOFs are grouped before the interface DOFs. Local equation numbering is prepared by utilizing the sorted local nodes array. Following that the transformation and compatibility matrices are generated as defined in Section 2.2.3.1. Local stiffness matrices and external load vectors are assembled and boundary conditions are applied exactly the same way in the standard finite elements procedure [18, 62].

As it is discussed in Section 2.2.4.1, subdomains may be floating or sufficiently supported. In case of floating substructures, since the local stiffness matrix is singular, inverse of it cannot be computed. Therefore, pseudo-inverse of local stiffness matrix is computed by "pinv" function and null space is obtained from "null" function which utilizes Singular Value Decomposition. Hence, the dual system defined in Section 2.2.3.1.2 can be reduced to interface by utilizing the procedure presented in Section 2.2.4.1. The resulting dual interface system is solved by PCPG algorithm demonstrated in Table 2.1 by utilizing one of the preconditioners discussed

in Section 2.2.4.4 and the projection matrix defined in Equation (31). In order to terminate the solution iterations criteria presented in Section 3.2.4 is utilized. Finally, the displacements of overall system can be recovered by utilizing the resulting interface equilibrium forces λ from these iterations by using Equation (19) and/or Equation (21). Then, element forces and stresses can be obtained by utilizing standard finite elements procedures [18, 62].

The implementation of this algorithm contains several details that require special consideration. One of them is that when the definition of $B^{(s)}$ given in Section 2.2.3.1 is applied, redundant constraints are formed at the corner nodes of a mesh partition (nodes belong to more than two subdomains) and this prevents $B^{(s)}$ to have a full column rank [98]. Since F_I and \hat{F}_I are in general semi definite, in this case, the solution λ of the dual interface problem is not unique. However, the corresponding subdomain displacements $\mathbf{u}^{(s)}$ are unique. Obviously, this phenomenon can be avoided with programming tricks, but in this implementation, it is preferred to keep the full redundancy in compatibility constraint definitions [99].

Another concern is about the selection of matrix Q that is utilized for the definition of projection space P given in Equation (31). If the overall solution domain is fully restrained—that is, if K is not singular— G_I has full column rank and $G_I^T G_I$ is non-singular [100]. In that case, it is preferable to restrict the choice of the matrix Q by the condition that $G_I^T Q G_I$ be also non-singular. Besides, if matrix Q is chosen among symmetric matrices, $G_I^T Q G_I$ becomes symmetric and thus it becomes more economical to handle. The simplest choice $Q = I$ is the most computationally efficient one and this choice is adopted for all numerical examples in this study.

For the utilization of FETI-2 the flowchart presented in Figure 3.2 also descriptive with a few changes. FETI-2 relies on an additional projection with admissible constraints defined by matrix C (Section 2.2.4.2) and then PCPG algorithm presented in Table 2.2 is utilized instead of the algorithm demonstrated in Table 2.1.

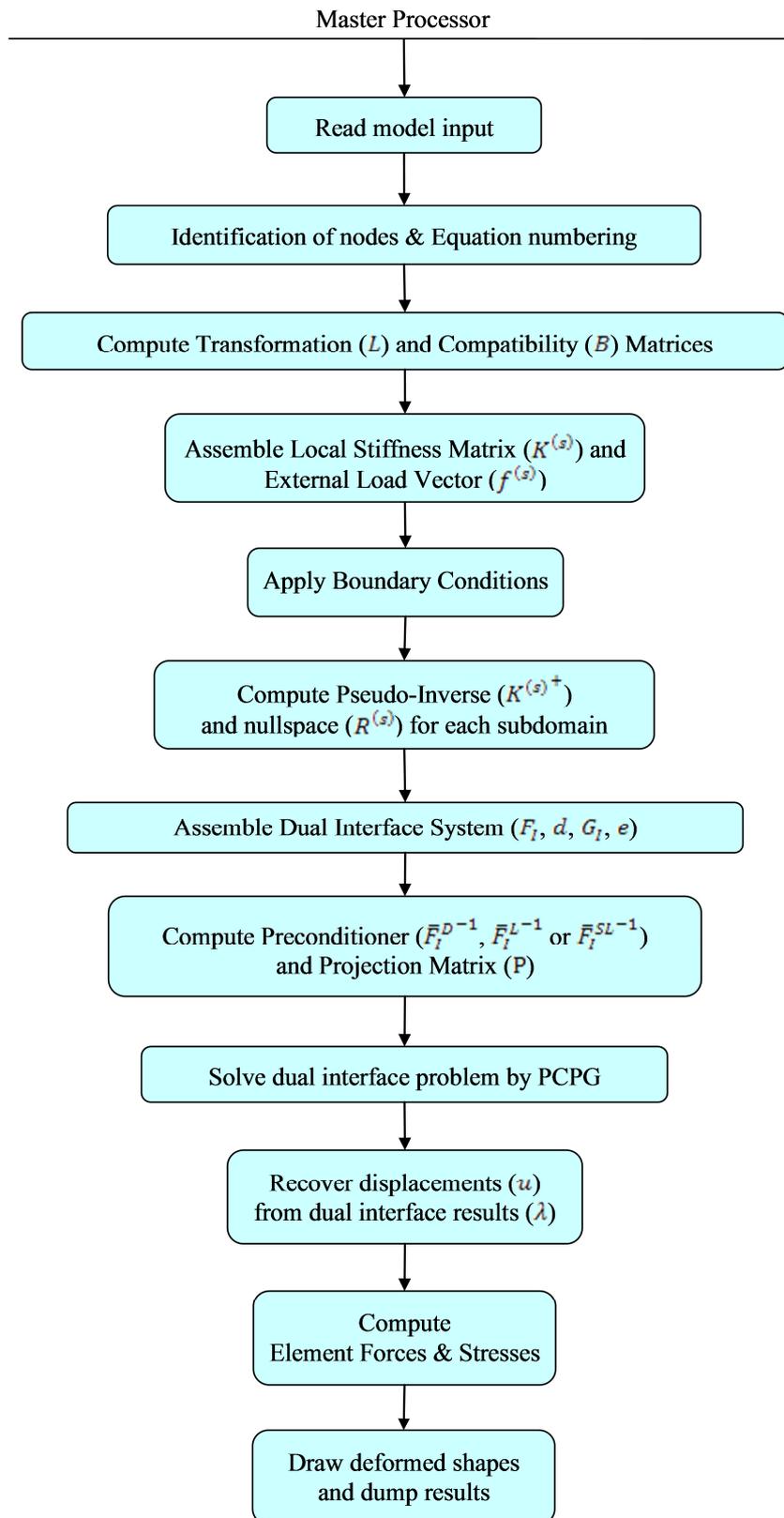


Figure 3.2. Flowchart of solution framework for static analysis

3.2.3. Implementation for Dynamic Analysis

The serial implementation of the implicit monolithic solution algorithm for dynamic analysis of dam-reservoir systems is presented in Figure 3.3. This implementation is almost same with the static case up to the time integration iterations. Additional parameters such as local mass and damping matrices and global mass matrix are computed as in standard finite elements procedures [18, 62]. For the rest of the steps up to time integration, instead of parameters originated from static definitions, dynamic definitions are computed and utilized as defined in Section 2.2.4.5.1 and Section 2.2.4.5.2. For example, coefficient matrix of dual interface system and preconditioner is computed by utilizing local effective stiffness matrices. In addition to definition of these computations, Section 2.2.4.5.1 states that projection space which exists in static analysis is vanished in dynamic case.

In order to compute the dynamic response history of the domain, implicit time integration algorithm that is presented in Section 2.2.4.5 is utilized in this implementation. These time history iterations are mainly relies on the solution of dual interface system for dynamics defined at each time step. For this purpose, at the beginning of each time iteration, displacements and momentums defined in Section 2.2.4.5 is computed. In addition to that, the resulting displacements and momentums are utilized together with the forces at that time step to compute the dual interface system right hand side. Obtained dual interface system for dynamics is solved by utilizing the PCPG algorithm demonstrated in Table 2.1 by utilizing one of the preconditioners (Section 2.2.4.4) that computed by using local effective stiffness matrix. But, for the implementation with FETI-1 solution algorithm, this time projection space parameter P is taken as unity since there is no projection space for the dynamic analysis with this algorithm (Section 2.2.4.5.1). However, in case of implementation with FETI-2 algorithm, PCPG algorithm presented in Table 2.2 is utilized and additional projection space is supplied by matrix C .

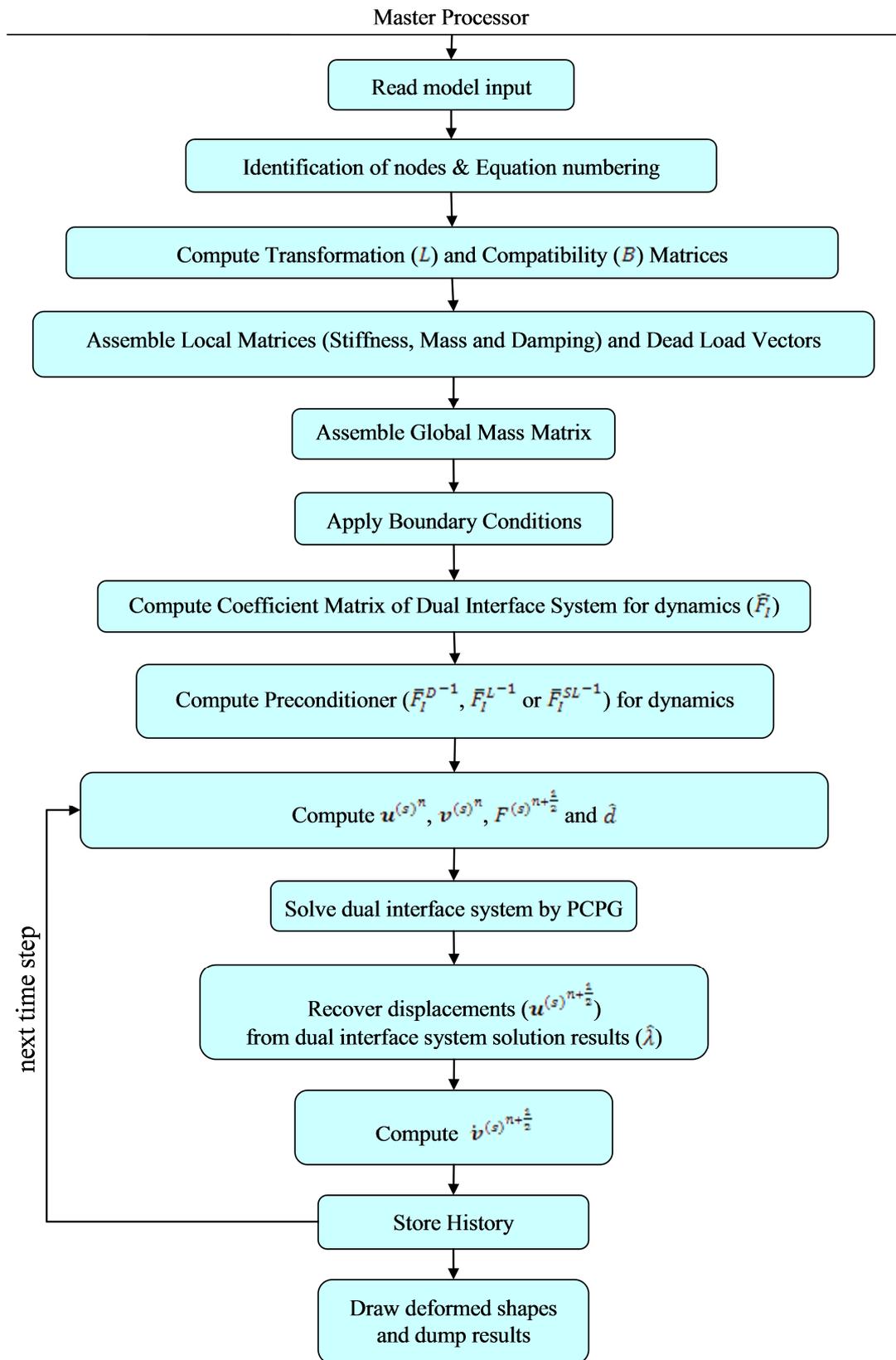


Figure 3.3. Flowchart of solution framework for dynamic analysis

For dynamics problems, it was proposed in [97] to choose $C = G_I$, with $R^{(s)}$ defined as the rigid body modes obtained for subdomain $\Omega^{(s)}$ when any prescribed Dirichlet boundary condition is ignored. Such a choice for matrix C is equivalent to forcing at each iteration k the constant components of $\mathbf{u}^{(s)k}$ to be continuous across the subdomain interfaces. It was also shown in [97] that the resulting two-level FETI method is numerically scalable for transient dynamics problems. After the solution of dual interface system, displacements are recovered by utilizing Equation (19) and parameters required to compute the displacements and momentums of the next time are computed. Finally, computed response for this time step is stored and then next time iteration initiates.

3.2.4. Coarse Problem Termination Criteria

In order to assess the performance of the new estimate for the Lagrange multipliers (see Table 2.1 and Table 2.2), convergence is monitored through the evaluation of the global primal residual:

$$\frac{\|K_g \mathbf{u}_g - \mathbf{f}_g\|}{\|\mathbf{f}_g\|} \leq \varepsilon = 10^{-6}, \quad (61)$$

where K_g , \mathbf{u}_g and \mathbf{f}_g are global assembled stiffness matrix, displacement field and forces as described in Section 2.2.3.

3.3. Methods for Improving Convergence Rate

In this section, several convergence rate improving methodologies are applied to the serial implementation of the solution framework and the change of behavior with these methods is investigated for this specific implementation on the sample problems within the scope of this study.

3.3.1. Re-orthogonalization of Direction Search Vectors

In following sections, mathematical background of re-orthogonalization is presented and performance improvements attained by re-orthogonalization of direction search vectors is discussed.

3.3.1.1. Mathematical Background

From their definition, Conjugate Gradient solvers are based on the orthogonality of the consecutive search vectors, which can be written as;

$$\mathbf{p}^k = \mathbf{y}^k - \frac{\mathbf{y}^{kT} P \mathbf{w}^k}{\mathbf{y}^{k-1T} P \mathbf{w}^{k-1}} \mathbf{p}^{k-1} \quad (62)$$

and

$$\eta^k = \frac{\mathbf{y}^{kT} \mathbf{w}^k}{\mathbf{p}^{kT} F_I \mathbf{p}^k} \quad (63)$$

However, a rapid loss of orthogonality between the direction vectors can be observed due to the numerical errors such as round-off and algorithmic errors such as spectral patterns in the dynamics [43]. In such cases, increasing the numerical precision does not restore the orthogonality of the direction search vectors since the propagation of the errors is a function of the ratio between consecutive eigenvalues [90].

The convergence rate of the PCG algorithm for the solution of the interface problem is badly influenced by the loss of orthogonality of the search directions. In order to overcome this problem, a re-orthogonalization procedure [86] within the PCG algorithm is introduced as;

$$\mathbf{p}^k = \mathbf{y}^k - \sum_{i=0}^{k-1} \frac{\mathbf{y}^{kT} F_I \mathbf{p}^i}{\mathbf{p}^{iT} F_I \mathbf{p}^i} \mathbf{p}^i \quad (64)$$

and

$$\eta^k = \frac{\mathbf{p}^{kT} \mathbf{w}^k}{\mathbf{p}^{kT} F_I \mathbf{p}^k} \quad (65)$$

However, determination of the new direction vector causes additional computational resources at each iteration k such as the storage of the direction vector \mathbf{p}^k and the product $F_I \mathbf{p}^k$. Besides, the computation of k dot products of the form $r_j^T [F_I \mathbf{p}^k]$ where $[F_I \mathbf{p}^k]$ is readily available and $1 \leq j < k$, and of an $n_i \times j$ matrix-vector product where n_i is the number of interface unknowns.

Obviously, such a re-orthogonalization procedure is not feasible if it is introduced during the solution with the PCG algorithm of a global finite element problem, as it would require unreasonable amount of memory and computation power. It is however quite affordable within the context of a domain decomposition algorithm as it applies iterative solution only to the interface problem. Particularly, the additional computational costs mentioned above are small compared to the cost of the pair of forward and backward substitutions that are required at each iteration k of the PCG algorithm in order to evaluate the product $F_I \mathbf{p}^k$. In other words, such a strategy is cost-effective for subdomain problems because it is applied only to the interface Lagrange multiplier unknowns [86].

3.3.1.2. Improvement Achieved by Re-orthogonalization of Direction Vectors

In order to investigate the improvement achieved by re-orthogonalization of direction vectors, three different problems discussed in Section 3.3.3 are solved with and without of the re-orthogonalization. In addition to the configurations presented in that section, iteration counts for solutions of the problem with different element discretization is also presented in the following figures.

Figure 3.4 demonstrates the improvement achieved by utilization of re-orthogonalization for cantilever problem. Number of iterations required for sQ4P0u meshing is higher than the meshing with sQ9P0u elements, as expected because of the greater number of sQ4P0u elements used in order to acquire the same accuracy as sQ9P0u elements. Apparently, re-orthogonalization reduces the number of iterations

required for the sQ4P0u meshing from around 70 to 50 which is approximately 28% reduction. However, for meshing with sQ9P0u elements reduction is around 20%.

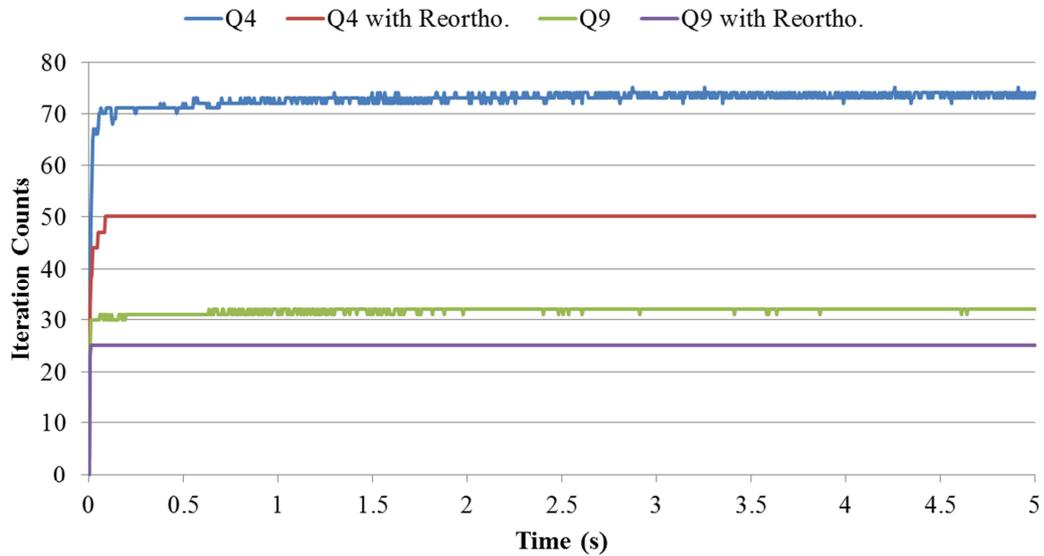


Figure 3.4. Improvement by re-orthogonalization on cantilever problem

In Figure 3.5, iteration counts for Tall Water Column modelled with fQ9P1u are presented. By utilization of re-orthogonalization, iteration counts required for the solution is reduced from 70 to 28 which equals to 60% reduction.

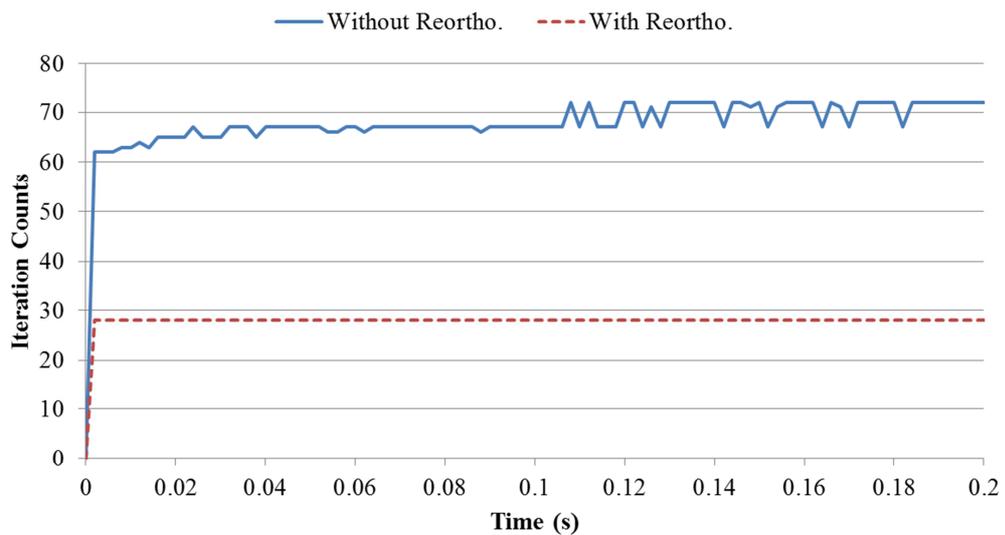


Figure 3.5. Improvement by re-orthogonalization on tall water column problem

Finally, Figure 3.6 demonstrates improvements attained by re-orthogonalization for the solution of the Water Tank problem. Since the number of elements is fixed for the two meshes, it is reasonable to obtain that number of iterations required for meshing with fQ9P1u elements is greater than the number of iterations required for meshing with fQ4P1u elements. For meshing with fQ9P1u elements, iteration required for the solution reduced from around 44 to 40 which equals to approximately 10%. However, reduction in meshing with fQ4P1u elements is not so apparent.

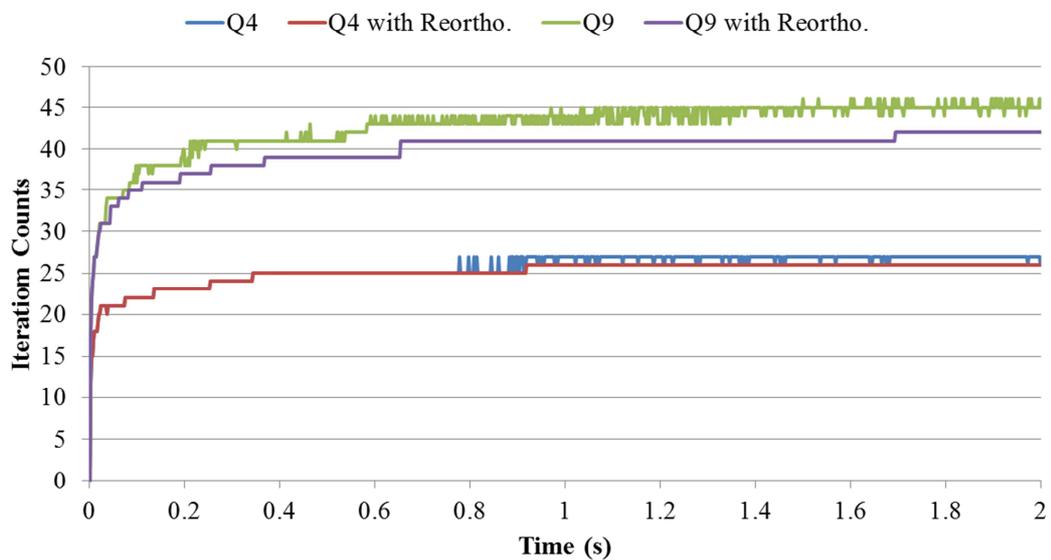


Figure 3.6. Improvement by re-orthogonalization on water tank problem

The performance improvement of the re-orthogonalization procedure presented above is given in Table 3.2 as the total number of iterations required for the solutions. Average improvement obtained by re-orthogonalization for fluid media problem is around 41% and for solid media problem, it is around 26%. However, for the fluid-solid interaction problem, improvement drops to 4%. Since re-orthogonalization procedure aims at enhancing the counts of PCG iterations carried for dual interface problem at each time step, presented discussion limited to iteration counts are descriptive.

Table 3.2. Improvement by re-orthogonalization with total iteration counts

Model	w/o	w	Improvement
Water Tank Q4	25636	25100	2%
Water Tank Q9	42791	39878	7%
Cantilever Q9	72792	49902	31%
Cantilever Q4	31742	24998	21%
Tall Water Column Q4	6826	2800	59%
Tall Water Column Q9	6777	5199	23%

In practice, the number of direction vectors that is stored for re-orthogonalization is determined by the memory space that is available after all of the other storage requirements of the FETI method has been satisfied. When only a few directions can be stored, a partial re-orthogonalization is implemented. In this case, the optimal strategy consists in storing the first few directions instead of the most recent ones, because the subspace generated by the first directions is closer to the subspace associated with the highest eigenvalues [86]. In summary, the FETI algorithm is always used with an explicit full re-orthogonalization procedure in order to accelerate convergence. In [42], it was shown that such a strategy is cost-effective for subdomain problems because re-orthogonalization is applied only to the interface Lagrange multiplier unknowns.

3.3.2. Improvement of Convergence by Krylov Subspaces

3.3.2.1. Mathematical Background

In FETI family domain decomposition methods, dual interface system is solved by a Krylov iterative solver which is generally a variation of the conjugate gradient method. In such solvers, iterations are based on searching for a direction vector and step length by using a residual vector originated from the previous iteration [90]. During the solution direction vector searches at each step, new direction is re-orthogonalized to the previous directions to improve the convergence as it is

described in Section 3.3.1. Thus, at each step a set of search directions are produced by conjugate gradient method. In this section, utilization of these Krylov subspaces in between solution of linearized dynamic equations at each time step will be discussed [41].

Supposing a system of N_A equations has been solved ($A\mathbf{x}_1 = \mathbf{b}_1$) in r_1 iterations, and that the $N_A \times r_1$ matrix S_1 associated with the Krylov subspace S_1 is readily available;

$$S_1 = [\mathbf{s}_1^{(1)} \quad \dots \quad \mathbf{s}_1^{(r_1)}] \quad (66)$$

where $\mathbf{s}_1^{(k)}$ and $r_1 < N_A$ denote the search direction vector at iteration k , and the number of iterations for convergence of that solution, respectively. In case of another system of equations, the same coefficient matrix but different right hand side ($A\mathbf{x}_2 = \mathbf{b}_2$), there can be a \mathbf{y}_2^0 vector to define the initial estimate value of \mathbf{x}_2^0 as;

$$\mathbf{x}_2^0 = S_1 \mathbf{y}_2^0 \quad (67)$$

It is shown that \mathbf{y}_2^0 is given as the solution of following system;

$$S_1^T A S_1 \mathbf{y}_2^0 = \tilde{\mathbf{b}}_2 \quad (68)$$

where

$$\tilde{\mathbf{b}}_2 = S_1^T \mathbf{b}_2 \quad (69)$$

It has been shown that $S_1^T A S_1$ is diagonal [41]. Therefore, the components $[\mathbf{y}_2^0]_j$ of \mathbf{y}_2^0 can be simply computed as follows;

$$[\mathbf{y}_2^0]_j = \frac{[\tilde{\mathbf{b}}_2]_j}{d_{1j}} \quad j = 1, \dots, r_1 \quad (70)$$

In case of the solution for the second time step with conjugate gradient method, since, \mathbf{x}_2^0 is orthogonal to S_1 , at each iteration k , the search directions $\mathbf{s}_2^{(k)}$ must be

explicitly orthogonalized to S_1 . This results in replacing the computation of modified search directions $\hat{s}_2^{(k)}$ as follows:

$$\hat{s}_2^{(k)} = s_2^{(k)} + \sum_{q=1}^{q=r_1} \alpha_q s_1^{(q)} \quad (71)$$

where

$$\alpha_q = -\frac{s_1^{(q)T} A s_2^{(k)}}{s_1^{(q)T} A s_1^{(q)}} = -\frac{s_2^{(k)T} A s_1^{(q)}}{s_1^{(q)T} A s_1^{(q)}}$$

Except for the above modifications, the original PCG algorithm is unchanged. However, convergence is expected to be much faster for the second and subsequent solution steps than for the first one, because S_1 and the subsequent supplementary spaces have smaller dimensions than possible solution dimension, and a significant number of the solution components are included in the startup solutions of the form of x_2^0 [41].

3.3.2.2. Improvement Achieved by Utilization of Krylov Subspaces

Water tank model defined in Section 3.3.3.1.3 is solved by utilizing both FETI-1 algorithm and FETI-2 algorithm. Although both algorithms produced the same results, total number of iterations and improvements obtained by utilization of Krylov subspaces are different. FETI-2 algorithm was always capable of solving each time step in a single PCG iteration. Obviously, this is because of the additional solution level inserted to the coarse problem. Since, this successive convergence of FETI-2 algorithm hinders the investigation of improvement achieved by the Krylov subspaces, for the rest of the discussions; FETI-1 solution algorithm is utilized.

In, Figure 3.7 iteration counts required for FETI-1 algorithm to converge a solution for each time step is shown for different model combinations. Krylov subspace improvement described above is not utilized, thus, number of iterations required

increases until the system gains a stable oscillating response that is similar to the response presented in Figure B.7.

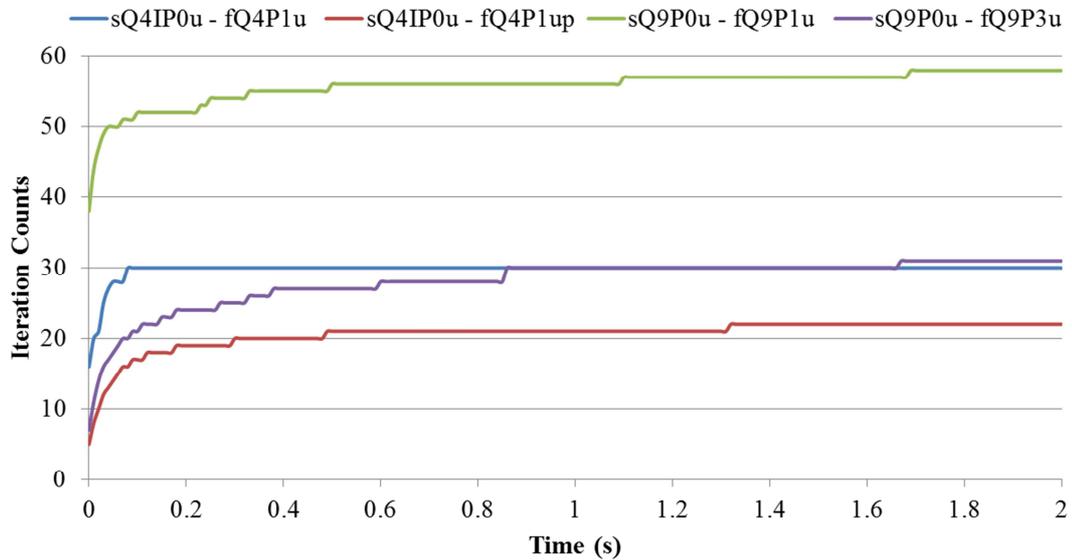


Figure 3.7. Iteration counts without Krylov subspaces improvement

As it is demonstrated in Figure 3.7, number of iterations required for the solution of models with Q4 elements with the same formulations is drastically lower than those of Q9 elements. It is acceptable because of the additional DOFs inserted to the system by using quadratic approximation. Another interesting comment is that displacement/pressure based formulation decreases the required number of iterations for the solution of the same model with the pure displacement formulations. This behavior can be explained by the decrease with the utilization of u/p formulation in the zero energy modes of floating subdomains. Henceforth, PCPG algorithm does not misdirected during the solution direction searches by non-realistic modes of subdomain.

In, Figure 3.8 again, the iteration counts required for FETI-1 algorithm to converge a solution for each time step is shown for different model combinations. However, in this case Krylov subspace improvements are applied to the iterations as described in the previous section. Obviously, during the beginning of the time integration

($t < 0.05$ and $t_{total} = 2.0$), there are meaningless up and downs in the iteration counts. This is due to skirmishing effects of the increasing trend observed in Figure 3.7 and the decreasing trend of the subspace improvement. After 0.005 seconds, decreasing trend of subspace improvements governs and finally after 0.35 seconds a single iteration is enough to converge for the solution.

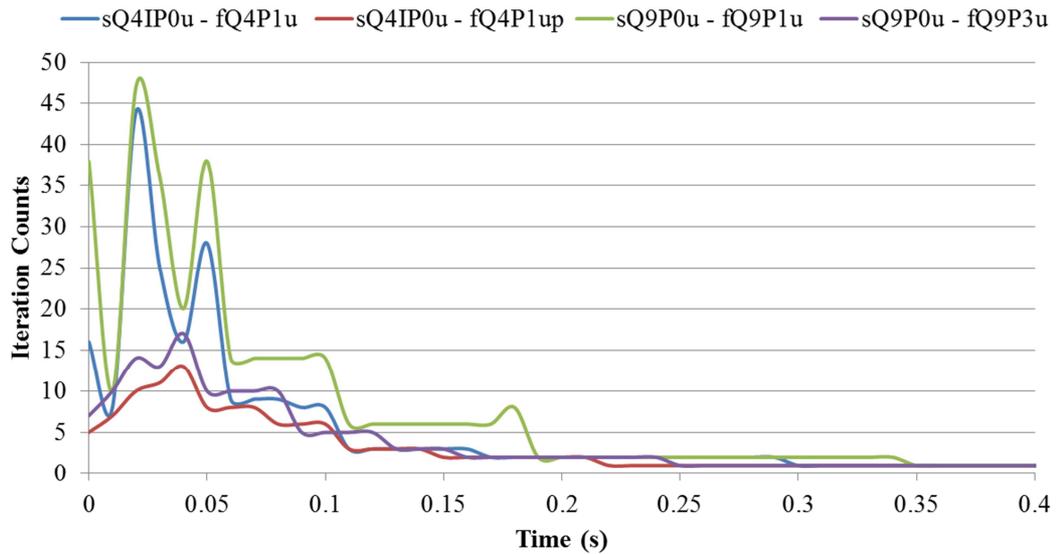


Figure 3.8. Iteration counts with Krylov subspaces improvement

Due to the stored search directions computed by the solutions of previous time steps, iterations required for the solution of the next time steps converges faster than before. Obviously, this behavior is drastic because of the similarities in between consecutive loadings at each time step. In case of solution of different multiple right hand sides, this behavior possibly not that effective since the solution domain will be totally different [41]. However, still, it is quite beneficial when the iteration counts required presented in Figure 3.8 is taken into consideration even for such a small problem.

3.3.3. Performance Tests with Different Preconditioners

Although, the scalability of FETI on different homogeneous problems are presented in the literature several times, for heterogeneous problems situation is not yet completely clarified. For arbitrarily heterogeneous problems, the optimal conditioning limits does not hold for most otherwise scalable domain decomposition methods [74, 86], and the performance of the Dirichlet, lumped, and similar preconditioners can be expected to deteriorate. Coefficient jumps are utilized in various techniques for preconditioning subdomain problems [65]. However, most if not all of these techniques are applicable only to domain decomposed problems where each subdomain is characterized by a single coefficient, in other words, for the problems that are globally heterogeneous but locally homogeneous.

In order to investigate the optimization possibilities of an implicit monolithic solution algorithm for dynamic analysis of dam-reservoir systems, different preconditioning configurations are chosen. Three possible interface situations are solved with three commonly utilized preconditioners presented in Section 2.2.4.4 such as Dirichlet, lumped and super-lumped.

3.3.3.1.1. *Solid Mechanics Problem*

In order to investigate the performances of chosen preconditioners on the structure to structure configuration, a cantilever beam subjected to a tip moment problem is modeled as in Figure B.3. By restricting the span to depth ratio of beam, shear deformations are assumed to be negligible. Therefore, Euler-Bernoulli Beam Theory can be used as a guide for determining reasonable problem dimensions. According to mentioned theory, deflection at any point of the span is given as $\vartheta(x) = Mx^2/2EI$. Finally, material and geometric properties are chosen according to equate tip deflection to unity. In order to prevent from the Saint-Venant's effect, displacement

comparisons are carried out at the mid-span. The analytical result at this point is 0.25 and the numerical solutions should converge to this result.

The cantilever beam is modeled with 2x6 mesh of sQ9P3u/p elements. The loading is applied to the system for just 0.1 seconds and then released. Rayleigh damping is not applied ($\alpha = 0.0$ and $\beta = 0.0$). Model is partitioned into four subdomains as it is shown in Figure 3.9.



Figure 3.9. Partitioning of cantilever beam problem

Iteration counts required for the solution of this problem by FETI-1 algorithm with different preconditioners are presented in Figure 3.10 and Figure 3.11. In these tests, Krylov subspaces are not utilized since this improvement hides the performance of preconditioners.

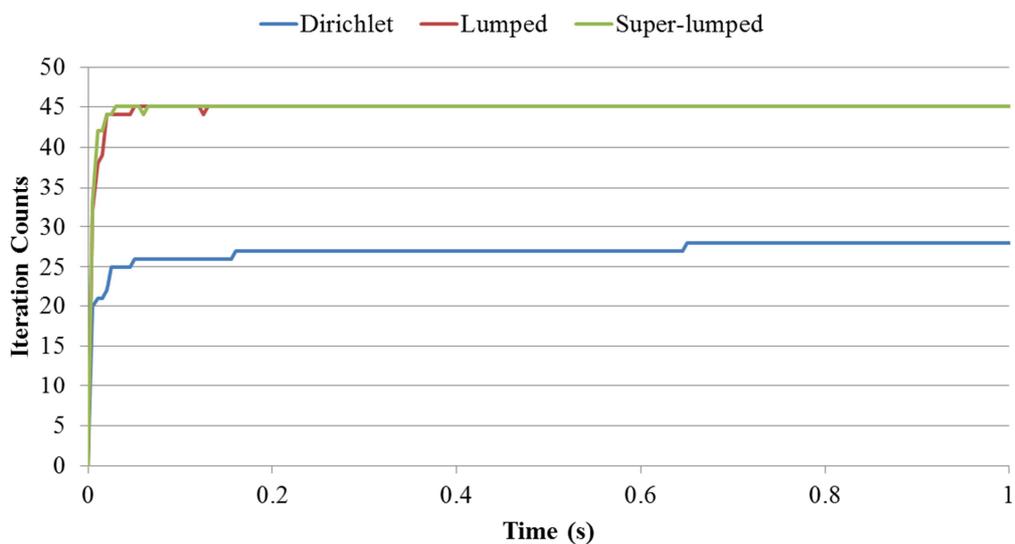


Figure 3.10. Cantilever beam solved by FETI-1 with different preconditioners

As it is demonstrated in Figure 3.10, Dirichlet preconditioner demonstrates its superiority on solid structural mechanics problems. It converges to the solution with almost half of the iterations required by other preconditioners. Super-lumped preconditioner has no significant contribution to performance when it is compared to lumped preconditioner.

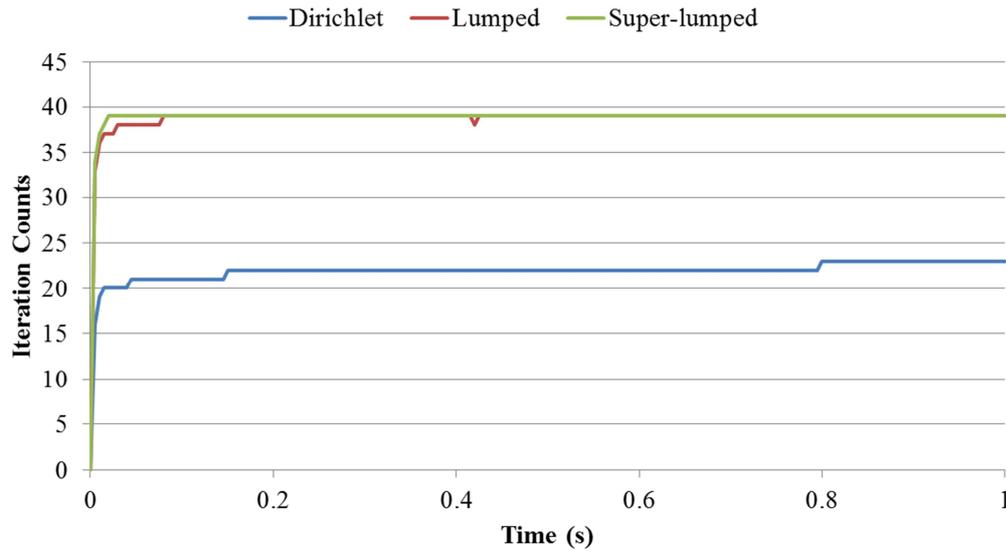


Figure 3.11. Cantilever beam solved by FETI-2 with different preconditioners

In Figure 3.11, results for the solution by FETI-2 algorithm are presented. When the results are compared with each other, similar conceptual outcomes are obtained. Besides, as it is expected, iteration counts required for FETI-2 is more less than the ones obtained from FETI-1 solution because of the additional solution level exist in FETI-2 algorithm as it is demonstrated in Section 2.2.4.2.

3.3.3.1.2. Hydrostatic Problem

The water column problem is modeled by utilizing mixed displacement/pressure formulations for fluid media under dynamic loading as in Figure B.5. Fluid media is meshed with 10 fluid finite elements of type fQ9P1u and restrained in X direction to represent the rigid tank boundary. Tank has 1/10 width height ratio.

Uniform area load which is lumped to the nodes using tributary area is applied to the system with the time function shown in Figure B.6. Rise time is taken as 0.01 where natural period of the system is around 0.0016. T/T_n ratio is in between 0.5 and 1 hence the dynamic response of mid-span displacement is in correspondence with the given figure by Chopra [40] as given in Figure B.7. Iteration counts required for the solution of this problem by FETI algorithms with different preconditioners are given in Figure 3.12 and Figure 3.13. In these tests, Krylov subspaces presented in Section 3.3.2 are not utilized.

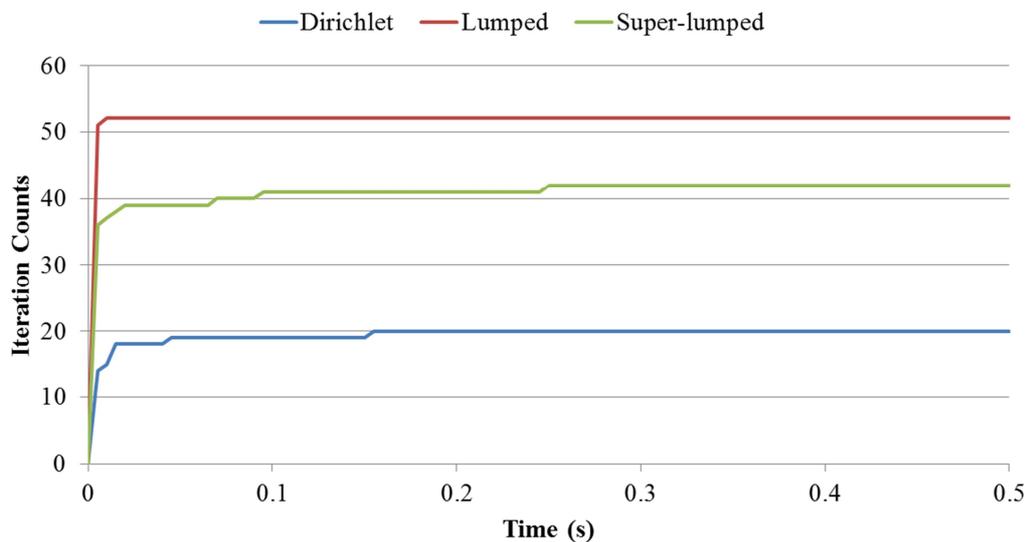


Figure 3.12. Hydrostatic problem solved by FETI-1 with different preconditioners

As it is presented in Figure 3.12, Dirichlet preconditioner results in the minimum iteration counts. The performance improvement by Dirichlet preconditioner against lumped preconditioner is increased when it is compared with cantilever problem.

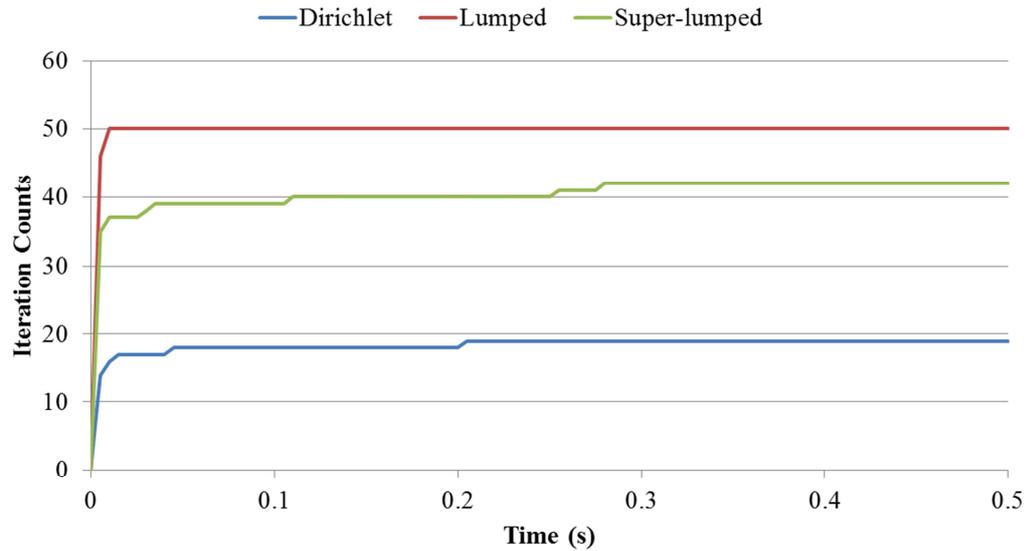


Figure 3.13. Hydrostatic problem solved by FETI-2 with different preconditioners

As it is demonstrated in Figure 3.13, FETI-2 results in slightly less number of iterations. Since it is known that performance improvement of FETI-2 results from second level of constrained solution by C parameter, it reveals that C parameter used for this problem is not efficient as in the cantilever problem.

3.3.3.1.3. Fluid-Structure Interaction Problem

Water tank with flexible walls is accepted as one of the characteristic problem of acoustic fluid-structure interaction and used as a benchmark problem for heterogeneous domains. Several fluid elements are contained in a one element thick of solid elements where it is base is hinged from an end and released in one direction. Dynamic loading is applied to the free end of the tank (Figure 4.18) by linearly increasing it to full capacity in 0.5 seconds (similar to Figure B.6(a), but with 0.5 seconds of rise time).

Problem is modeled with “sQ9P0u - fQ9P3u/p” elements which designates that solid media is modelled with Q9 elements with displacement formulation and fluid media

is modelled with Q9 elements with displacement/pressure based elements. Domain is partitioned into two subdomains according to media. Since, fluid media has no restraints, it is floating. Complete set of rigid body modes of this floating subdomain are computed by using "null" function of MATLAB in which Single Value Decomposition algorithm is utilized. Thus, the coarse problem formed is in between solid-fluid interface which is heterogeneous.

Iteration counts required for the solution of this problem by FETI solution algorithms with different preconditioners are given in Figure 3.14 and Figure 3.15. In these tests, Krylov subspaces presented in Section 3.3.2 are not utilized.

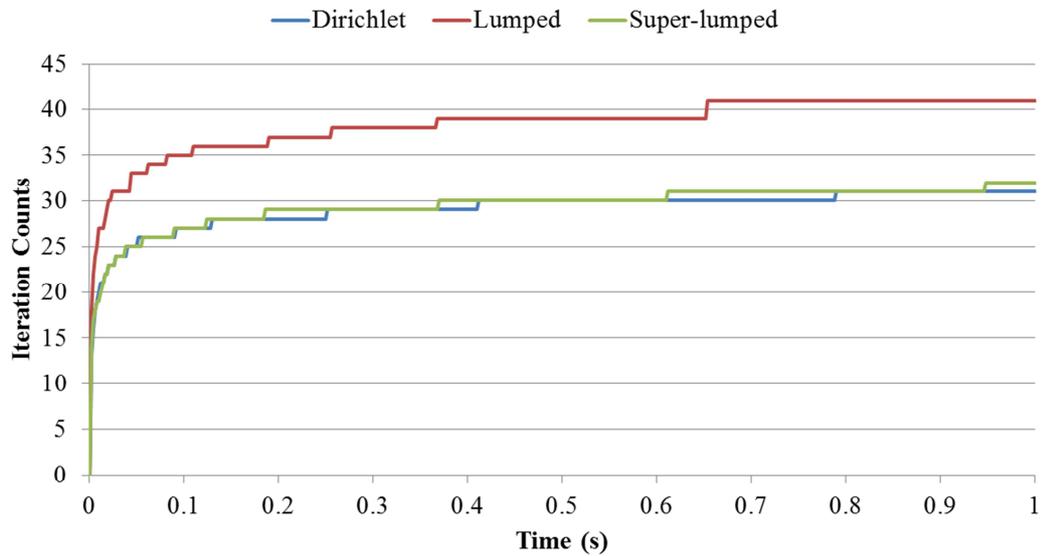


Figure 3.14. Water tank problem solved by FETI-1 with different preconditioners

As Figure 3.14 demonstrates, Dirichlet preconditioner exhibits a superior performance on this problem also. It requires 40% less number of iterations than lumped preconditioner. Besides, super-lumped preconditioner exhibits a similar behavior as Dirichlet preconditioner. As it is stated in Section 2.2.4.4.3, super-lumped preconditioner is very cheap when compared to Dirichlet preconditioner.

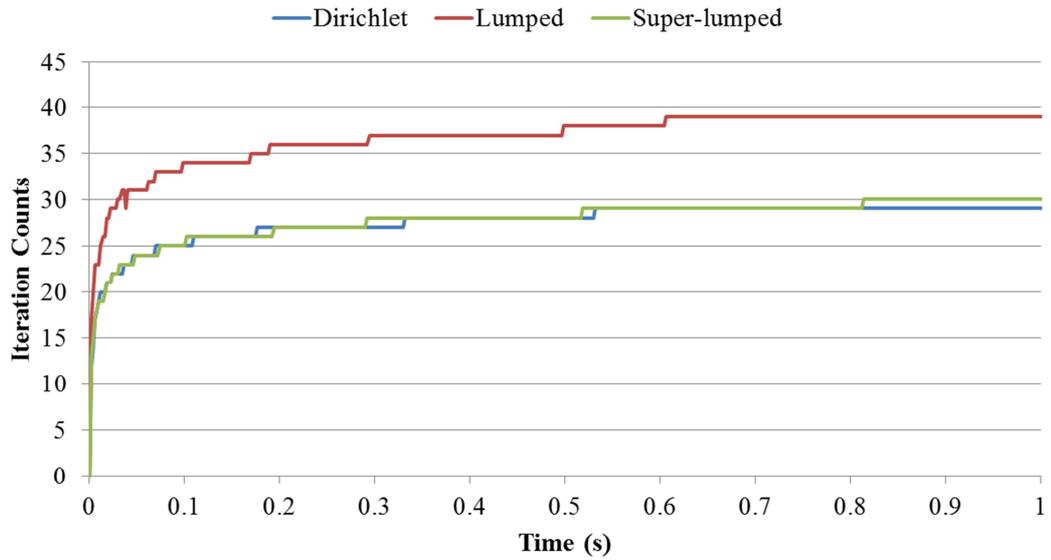


Figure 3.15. Water tank problem solved by FETI-2 with different preconditioners

If the FETI-2 results are investigated from Figure 3.15, FETI-2 requires slightly less number of iterations for the solution and the iteration count trends are similar to the ones in Figure 3.14.

To sum up, utilized preconditioner significantly effects the iteration counts required to obtain the solutions. Although, it is stated that Dirichlet preconditioner is quite expensive computationally when it is compared with lumped and super-lumped preconditioners, it performs the best in all cases. However, for the heterogeneous problems partitioned into homogeneous subdomains, super-lumped preconditioner can be preferred.

CHAPTER 4

CONDITIONING THE INSTABILITIES

4.1. Introduction

As one of the acoustic fluid-structure interaction problems, dam-reservoir interaction contains the analysis of the solid structure domain i.e. dam, the fluid domain i.e. reservoir and the coupling in between these two domains. Displacement formulation is generally utilized for the discretization of solid structure [18]. On the other hand, several fluid formulations are utilized for modeling the fluid domain depending on the assumptions made on fluid properties and flow/problem conditions. For a considerable amount of structural engineering problems; the behavior of the fluid domain can be considered within the limits of acoustics and the fluid domain discretization can be greatly reduced to a simpler form. Therefore, a simplified Lagrangian representation of the fluid domain is possible and highly preferable in case of dam-reservoir interaction problems [1, 63]. Although it is computationally expensive, representing the both sides of the interaction with the same unknown field eliminates the necessity to develop special coupling methodologies for the fluid-structure interaction.

Monolithic solution of a fully Lagrangian representation of acoustic fluid-structure interactions is composed of the simultaneous solution of fluid and solid equilibrium equations without inserting any additional energy to the system or any time lag in between time steps [9]. Since fluid field requires smaller time steps for the solution, additional computational requirements are appealed as a result [101]. In order to limit the high computational requirements of the monolithic Lagrangian solution of dam-reservoir problems, fluid properties are simplified by ignoring compressibility and/or

viscosity. On the other hand, iterative solution methods and parallel computing techniques provide considerable computational power. Therefore, iterative solution methods highly suitable for parallelization [102] like FETI (Finite Element Tearing and Interconnecting) family of solution methods [25] can be of interest to solve dam-reservoir interaction problems.

Iterative, fully Lagrangian monolithic solution of dam-reservoir interaction has several disadvantages. One of the main disadvantages is the considerable change in behavior of materials along the fluid-structure interface which may cause ill-conditioning of the problem [103]. Since the interface system defined by FETI is generally solved by PCG (Preconditioned Conjugate Gradient) algorithm, in addition to preconditioning, it is possible to condition the solution by deflation methods in case of ill-conditioning [31, 33, 36, 104].

Deflation is the projection of the original problem to a better conditioned representative state, such that the convergence rate of the iterative solution is improved or the solution is carried out successfully with less computational requirements [30]. Deflation methods have several algebraic connections with multi-grid methods and projection preconditioners [29, 31, 32]. Efficiency of the deflation methods highly depends on the deflation vectors utilized to define the projection space [33–36]. In case of subdomains composed of solid elements, computation of rigid body modes and utilizing them as deflation vectors is a straight forward task [33, 37, 38]. However, in case of subdomains with Lagrangian fluid finite elements, such methods are not applicable and a fully rank revealing Eigen solution is required [39]. Therefore, for such problems, more efficient and robust deflation vector generation methods are needed to obtain an improvement in terms of computation and memory requirements [29]. Thus, the main focus of this chapter is to examine the effect of different deflation vector generation methods on the efficiency of the dam-reservoir interaction problems in terms of condition number, iteration count, operation count, and memory requirements. Moreover, a novel deflation vector generation method is proposed. In this method, the pre-selected a priori known zero

energy modes of the fluid finite elements used in the discretization of the fluid domain are utilized in order to obtain a deflation space for the solution and finally enhance the convergence of the iterative solution of the dam-reservoir system. Along with the proposed method, several other deflation vector generation methods are utilized to solve the water tank with flexible walls problem which is a classical benchmark problem for acoustic fluid-structure interaction. Then, an actual dam-reservoir interaction problem is solved by utilizing FETI-2 algorithm [22, 24] and the performance of each deflation vector generation method is discussed in detail.

4.2. Theory

This study focuses on monolithic time domain analysis of dam-reservoir interaction problems where both dam body and reservoir are modeled with finite elements having displacements as their DOFs. The solid domain discretization is performed with classical displacement based finite elements [18] whereas two different formulations, pure displacement [59] and displacement/pressure formulations [62] are utilized for the fluid domain discretization. Monolithic time domain analysis requires large memory capacity and computational resources, hence, high performance computing techniques are generally preferred [10]. In this study, domain decomposition based iterative, high performance solution method, FETI family methods [25] are implemented. Due to the nature of the problem, solid and fluid subdomains not only have large stiffness differences but also have different governing equations that describe their behavior. Because of this reason, the system may be highly ill-conditioned which requires additional precautions to achieve acceptable convergence rates.

4.2.1. Discretization

The solid domain, i.e. the dam body and foundation rock, is modeled with solid finite elements with displacement based formulation [18] but two different formulations

for fluid elements are implemented and tested. The first fluid element formulation is pure displacement based formulation [59], where the fluid strains are computed by the linear strain-displacement relationship and the constraint of zero fluid rotation at the integration points are introduced. The only strain energy taken into account is related to the compressibility of the fluid. It is reported that although pure displacement based fluid elements are stable for a wide-range of frequencies, they have spurious zero energy modes [62]. The other fluid element is based on mixed formulation [62], approach. The formulation is based on Hu-Washizu principle with displacement/pressure DOFs such that spurious zero energy modes are prevented (inf-sup condition). The formulation has both displacements and pressure quantities as DOFs but pressure DOFs are condensed out prior to system assembly.

Pure displacement based fluid formulation ensures the solvability and stability if the bulk modulus and shear modulus of fluid are of the same order. In case of almost incompressible material analysis, displacement/pressure based mixed formulation that satisfies the inf-sup condition is well established [49, 50]. However, in case of inviscid acoustic fluid model analysis, due to the loss of ellipticity, zero energy modes corresponding to the zero deviatoric strain energy are formed [62].

4.2.2. System Solution

In FETI family of solution methods [21, 22, 24], solution domain Ω is partitioned into N_s number of non-overlapping subdomains $\Omega^{(s)}$. For each subdomain, local stiffness matrix $K^{(s)}$, local load vector $\mathbf{f}^{(s)}$ and local unknowns $\mathbf{u}^{(s)}$ can be grouped in block diagonal format as follows;

$$K = \begin{bmatrix} K^{(1)} & & \\ & \ddots & \\ & & K^{(N_s)} \end{bmatrix}, \mathbf{u} = \begin{Bmatrix} \mathbf{u}^{(1)} \\ \vdots \\ \mathbf{u}^{(N_s)} \end{Bmatrix}, \mathbf{f} = \begin{Bmatrix} \mathbf{f}^{(1)} \\ \vdots \\ \mathbf{f}^{(N_s)} \end{Bmatrix} \quad (72)$$

Continuity of displacements along the interface can be separated as constraints and Lagrange multipliers $\boldsymbol{\lambda}$ which represent the interface forces can be defined as dual variables to obtain the following minimization problem;

$$\begin{cases} Ku + B^T \boldsymbol{\lambda} = \mathbf{f}, \\ B\mathbf{u} = 0, \end{cases} \quad (73)$$

where $B = [B^{(1)} \dots B^{(N_s)}]$ and $B^{(s)}$ is a signed matrix that represents the interface continuity. This minimization problem can be solved by condensing the displacements and solving the generated dual variable system by an iterative solution method like preconditioned conjugate gradient (PCG) method. Here, K is generally a positive semi-definite matrix [21].

In the original FETI algorithm, generalized inverse is utilized for local subdomain solutions and consistency of the linear system is enforced by a projection matrix, P which builds up a coarse space. In general sense, for any given Q matrix, P can be expressed as follows;

$$P(Q) = I - QG_I(G_I^T Q G_I)^{-1} G_I^T \quad (74)$$

where I represents the identity matrix of appropriate size, $G_I = BR$ and R spans the null space of K . This procedure is called one-level FETI or FETI-1 [21, 25].

In two-level FETI algorithms [22, 24], also named as FETI-2, an additional level of coarse space is formed by defining admissible constraints on the interface of the form;

$$C^T \mathbf{w}^k = 0 \quad (75)$$

which is enforced at each k^{th} PCG iteration for the solution of coarse problem. In Equation (75), matrix C contains vectors of some subspace to be determined, and $\mathbf{w}^k = P^T(\mathbf{d} - F_I \boldsymbol{\lambda}_k)$ represents the projected unbalanced right hand side of the coarse problem for the k^{th} iteration.

In all of the aforementioned variants of FETI, a coarse space is formed to assure the scalability with respect to the number of subdomains by global error propagation over the whole domain and accelerate convergence.

As the coarse problem is generally solved with iterative solvers, one of the most significant indicators for the accuracy of the iterative solutions and the convergence rate is the condition number of the problem. In monolithic solution of fully Lagrangian fluid-structure interaction problems, coefficient matrices are symmetric; therefore condition number can be approximated as the ratio of the maximum eigenvalue to the minimum eigenvalue of the system:

$$\kappa_2(K) = \left| \frac{\lambda_{max}}{\lambda_{min}} \right| \quad (76)$$

In case of domain decomposition methods, the condition number increases according to the increase in the H/h ratio where H denotes the size of subdomains and h denotes the size of the elements [10]. Moreover, ill-conditioned matrices reduce the convergence rate of an iterative solution method. This is the case, especially, in fully Lagrangian acoustic fluid-structure interaction problems because there are considerable differences in the behavior of materials along the solid-fluid interfaces. In this case, scaling methods [74, 86] and special preconditioners [92] are utilized to have a better conditioned problems and increase the convergence rate of the solution.

Another source of ill-conditioning for this specific problem is that if the fluid domain is modeled as almost incompressible, condition number is asymptotically increasing as the Poisson's ratio of the fluid element is modeled as being closer to 0.5. Therefore, if the compressibility of the fluid domain is of interest and the fluid domain is modeled as almost incompressible or inviscid, the problem may become significantly ill-conditioned [105, 106].

In both FETI-1 and FETI-2 formulations, the local stiffness matrix of subdomains, $K^{(s)}$, can be singular, i.e. floating subdomain [22, 24]. Therefore, the null space of the local stiffness matrices which represent the rigid body modes of subdomains is required for the elimination of singularities. In case of solid floating subdomains immune to internal mechanisms, 3 rigid body modes (2 translations + 1 rotation) in 2D, 6 rigid body modes (3 translations + 3 rotations) in 3D may exist at most. Otherwise, direct computation of null spaces is required and it is a computationally expensive procedure. Hence, several other methods, heuristic or approximate, are proposed [38]. The main assumption of these methods is the subdomains or the finite elements composing the subdomains do not contain any internal mechanisms, or “Zero Energy Modes”. Therefore, only a few of them can be utilized for finite elements that are not compatible with this condition like the pure displacement based fluid elements [59].

In this study, deflation method is utilized to condition the solution against the aforementioned instabilities in addition to preconditioning and this way not only the convergence of the solution is guaranteed but also the number of iterations is also reduced.

4.2.3. Deflation Method

Deflation method, first proposed by Nicolaides [107] in 1987, is a conditioning method that can be utilized together with preconditioning method even though it is not compulsory. The main idea behind the deflation method is to define a projection matrix which projects the extreme eigenvalues out of the system and this way the condition number of the problem is reduced. Therefore, iterative method used for the system solution can converge with less number of iterations than before or successfully converge to a solution with less computational cost.

The main steps of the deflation method are presented in Figure 4.1.

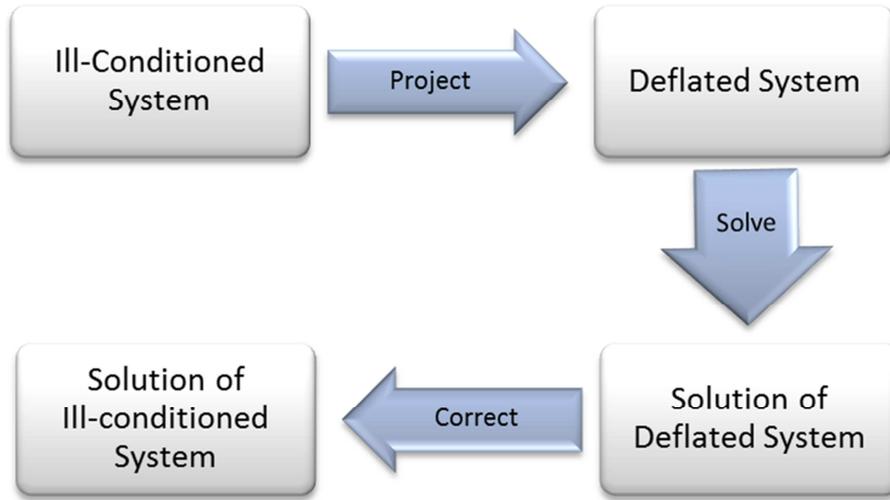


Figure 4.1. Flowchart for deflated system solution

As a starting point, unknowns of linear system of equations;

$$K\mathbf{u} = \mathbf{f} \quad (77)$$

are split into two parts;

$$\mathbf{u} = (I - P^T)\mathbf{u} + P^T\mathbf{u} \quad (78)$$

where

$$P = I - KQ, Q = ZE^{-1}Z^T \text{ and } E = Z^TKZ \quad (79)$$

In Equation (79), I represents the identity matrix of appropriate size, Q is $n \times n$ correction matrix, Z is the deflation subspace, i.e., the space to be projected out of the system, E is $k \times k$ symmetric positive definite matrix and P is $n \times n$ projection matrix if k is the number of deflation vectors and n is the number of unknowns. By assuming $k \ll n$ and Z is full rank, also considering that KP^T is symmetric, deflated system can be written as;

$$PK\hat{\mathbf{u}} = P\mathbf{f} \quad (80)$$

where $\hat{\mathbf{u}}$ designates the unknowns of the deflated system. Deflated system in Equation (80) can be solved by using PCG and the final displacements can be obtained from the result of deflated system solution by using following correction;

$$\mathbf{u} = Q\mathbf{f} + P^T\hat{\mathbf{u}} \quad (81)$$

Note that deflated system in Equation (80) is singular. However, the projected solution $P^T \hat{\mathbf{u}}$ is unique; it has no components in the null space $\mathcal{N}(PK) = \text{span}\{Z\}$. Moreover, the null space of PK do not go into the iterations and the solution is not influenced by the corresponding zero eigenvalues [68]. Therefore, as long as the system is consistent it can be solved by CG variant methods.

An effective error bound for $n \times n$ deflated system PK with $n - m$ rank is defined [105, 108] as;

$$\kappa_{eff}(PK) = \frac{\lambda_n(K)}{\lambda_{m+1}(C)} \quad (82)$$

by supposing a splitting $K = C + R$, where C and R are symmetric semi-definite with null space of C , $\mathcal{N}(C) = \text{span}\{Z\}$ and by ordering eigenvalues as;

$$\lambda_i(C) \leq \lambda_i(PK) \leq \lambda_i(C) + \lambda_{max}(PR) \quad (83)$$

When deflation is utilized together with preconditioning, Equation (82) extends for the preconditioner $M = LL^T$ as;

$$\kappa_{eff}(L^{-1}PKL^{-T}) = \frac{\lambda_n(L^{-1}KL^{-T})}{\lambda_{m+1}(L^{-1}CL^{-T})} \quad (84)$$

4.2.4. Implementation of Deflation

Deflated version of preconditioned CG algorithm [109] is given in Table 4.1. Replacement of original coefficient space, K with deflated coefficient space, PK at the beginning of iterations designates the ‘‘Project’’ step in Figure 4.1 and hence residuals, search directions and step lengths become defined for the deflated solution space. As in the ‘‘Correct’’ step in Figure 4.1, after the convergence of iterations, deflated system solution $\hat{\mathbf{u}}_{k+1}$ is utilized to obtain the solution \mathbf{u} .

Table 4.1. Pseudocode for Deflated PCG algorithm

Pseudocode: DPCG for solving $K\mathbf{u} = \mathbf{f}$

Choose \mathbf{u}_0 , compute $\mathbf{r}_0 = (\mathbf{f} - K\mathbf{u}_0)$ and $\hat{\mathbf{r}}_0 = P\mathbf{r}_0$
Solve $M\mathbf{y}_0 = \hat{\mathbf{r}}_0$ and $\mathbf{p}_0 = \mathbf{y}_0$
for $k = 0, 1, \dots$ until convergence
 $\hat{\mathbf{w}}_k = PK\mathbf{p}_k$
 $\alpha_k = \frac{(\hat{\mathbf{r}}_k, \mathbf{y}_k)}{(\hat{\mathbf{w}}_k, \mathbf{p}_k)}$
 $\hat{\mathbf{u}}_{k+1} = \hat{\mathbf{u}}_k + \alpha_k \mathbf{p}_k$
 $\hat{\mathbf{r}}_{k+1} = \hat{\mathbf{r}}_k - \alpha_k \hat{\mathbf{w}}_k$
Solve $M\mathbf{y}_{k+1} = \hat{\mathbf{r}}_{k+1}$
 $\beta_k = \frac{(\hat{\mathbf{r}}_{k+1}, \mathbf{y}_{k+1})}{(\hat{\mathbf{r}}_k, \mathbf{y}_k)}$
 $\hat{\mathbf{p}}_{k+1} = \hat{\mathbf{y}}_{k+1} + \beta_k \mathbf{p}_k$
end for
 $\mathbf{u} = ZE^1Z^T\mathbf{f} + P^T\hat{\mathbf{u}}_{k+1}$

In practice, factorization of matrix E and matrix multiplication of KZ can be computed beforehand and used repetitively during iterations. Besides, explicit computation of P is not required, by rearranging the terms in Equation (79), it is obvious that $P\mathbf{v} = \mathbf{v} - KQ\mathbf{v}$ and $P^T\mathbf{v} = \mathbf{v} - QK\mathbf{v}$ for any \mathbf{v} .

In case of almost all variants of FETI algorithm, coarse problem is generally solved by utilizing PCG. In original FETI algorithm, FETI-1, projection space is already implemented in the algorithm in order to handle floating subdomains [21]. Computed zero energy modes (a generalization to rigid body modes in solid mechanics) which correspond to zero eigenvalues of subdomains are stored in rectangular matrix R . These subspace vectors are taken into account during the G_I parameter computation in Equation (74). On the other hand, in case of FETI-2, algorithm is mainly built upon the fact of utilizing a set of subspace vectors through C matrix in order to project the problem. Thus, deflation method can easily be implemented to FETI-1 and FETI-2 by enriching the projection subspace by the predetermined additional deflation vectors. Moreover, the projection matrix, $P(Q)$ in Equation (74) is actually the transpose of the P in Equation (79). In other words, the projection matrix generated with deflation vectors can be used as projection matrix of FETI coarse problem solver.

4.3. Deflation Vector Generation

Generation of vectors that define the projection space for the deflation method is a difficult task. The main challenge is to choose sufficient number of deflation vectors that would result in a better convergence rate in a computationally cheaper way. One way of obtaining the deflation vectors is a fully rank-revealing algorithm by utilizing QR or Singular Value Decomposition (SVD) methods. These methods are, however, computationally expensive which makes them impractical to use in the solution of large scale models. Hence, alternative deflation vector generation methods are crucial to increase the improvement obtained by the deflation method.

Estimating the effect of the generated deflation vectors to the solution time a priori is not a straight forward task. Firstly, the deflation vector generation procedure and deflation method itself will require extra computation. Likewise, stored deflation parameters will require additional memory space. In order to minimize these disadvantages, it is necessary to set several prerequisites for the generated deflation vectors.

The main requirement for the deflation vectors is that, they should be as sparse as possible. This way, the projected matrices would be also sparse and thus requires less memory space and computation. Moreover, deflation vectors should be able to approximate the eigenspace corresponding to the extreme eigenvalues especially the smallest ones as much as possible. Otherwise, the efficiency of the deflation method cannot be guaranteed. Last but not least, generation of the deflation vectors should be cheap in the sense of computational resources and memory. Besides, generation algorithm should be suitable for parallel programming in order to utilize the available hardware efficiently.

In general, preconditioning treats the largest eigenvalues of the system, effectively. Therefore, targeting the smallest eigenvalues during the deflation vector generation

enables the deflation matrix to behave as a complementary part of the preconditioning by projecting the smallest eigenvalues out of the system [109].

4.3.1. Subdomain Deflation Method

Subdomain deflation method which is one of the arbitrary deflation vector generation algorithms utilizes the partitioning information of subdomains to construct deflation vectors. In this method, each subdomain has at least one corresponding deflation vector. In the simplest case which is the zero order approximation, the corresponding row of the deflation vector has value “1” if the corresponding DOFs are on the owner subdomain and has value “0” otherwise. In general sense, the deflation subspace Z is defined as;

$$Z = \bigcup_0^k Z^k \text{ where } Z_{ij}^k = \begin{cases} \alpha^k, & i \in \mathbb{I}_j \\ 0, & i \notin \mathbb{I}_j \end{cases} \quad (85)$$

by assuming that solution domain Ω is partitioned with index set $\mathbb{I} = \{i \mid u_i \in \Omega\}$ into N_s number of non-overlapping subdomains $\Omega_j, j = 1, \dots, N_s$, with respective index sets $\mathbb{I}_j = \{i \in \mathbb{I} \mid u_i \in \Omega_j\}$ where k designates the order of approximation and u_i is not necessarily an actual partitioning point or interface DOFs, it could be an imaginary grid point [37].

For instance, suppose that, for a 1D domain Ω , consists of solution points x_1, \dots, x_6 , that is divided into two subdomains such that $\Omega_{h_1} = \{x_1, x_2, x_3\}$ and $\Omega_{h_2} = \{x_4, x_5, x_6\}$. Subdomain deflation vectors are aggregation of the piecewise constant, linear and second order etc. deflation vectors as presented in Equation (86). Obviously, for each deflation vector, number of non-zeros is fixed according to partitioning, all vectors are moderately sparse, orthogonal to each other and disjoint [109].

$$\begin{aligned}
Z &= Z^0 + Z^1 + Z^2 + \dots \\
&= \begin{Bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{Bmatrix} + \begin{Bmatrix} 1 & 0 \\ 2 & 0 \\ 3 & 0 \\ 0 & 1 \\ 0 & 2 \\ 0 & 3 \end{Bmatrix} + \begin{Bmatrix} 1 & 0 \\ 4 & 0 \\ 9 & 0 \\ 0 & 1 \\ 0 & 4 \\ 0 & 9 \end{Bmatrix} + \dots
\end{aligned} \tag{86}$$

4.3.2. Recursive Deflation Method for Heterogeneous Problems

In [37], a recursive deflation vector generation method is presented in order to decouple the eigenspace of different materials. Sets of finite elements which have the same material properties are chosen to define a matrix C and deflation space is defined by the null space of the C matrix.

The group of elements that forms up a body l of the material j is defined as Ω_j^l , where $\Omega = \cup_{j=1}^m \{ \cup_{l=1}^{j_k} \Omega_j^l \}$. Utilizing the splitting presented in Equation (82) for the first material body, it is decoupled from all other materials. The rigid body modes of all elements corresponding to the first body of material l are contained in $\mathcal{N}(C_0)$ where the matrix C_0 consists of the assembly of all finite elements that belongs to it. Consequently, that yields, $\mathcal{N}(C_0) = \tilde{Z}_1$ and $P_1 = I - \tilde{A}_0 \tilde{Z}_1 (\tilde{Z}_1^T \tilde{A}_0 \tilde{Z}_1)^{-1} \tilde{Z}_1^T$.

Continuing the splitting recursively [105], for each body of each material, at splitting m ;

$$\mathcal{N}(C_{m-1}) = \tilde{Z}_m \tag{87}$$

and

$$P_{m-1} = I - \tilde{K}_{m-1} \tilde{Z}_m \left(\tilde{Z}_m^T \tilde{K}_{m-1} \tilde{Z}_m \right)^{-1} \tilde{Z}_m^T = P \tag{88}$$

with $P = I - AZ(Z^T AZ)^{-1}Z^T$ and $span(Z) = \cup_{j=1}^m span\{\tilde{Z}_j\}$.

In short, deflation removes the corresponding extreme eigenvalues from the system by removing the rigid body modes of the subdomains corresponding to the different materials. In this method, it is assumed that each finite element has at most 3 rigid body modes in 2D, 6 rigid body modes in 3D. Therefore, these rigid body modes of each element are utilized for computing the null space of subdomains [37].

4.3.3. A Novel Strategy: Semi-heuristic Deflation Method

Recursive deflation vector generation strategy relies on the assumption that each finite element has at most 3 rigid body modes in 2D, 6 rigid body modes in 3D. It is apparent and applicable for subdomains which have no internal mechanism like solid domains. For the specific problem examined in this study; there are two main problems with this assumption. Firstly, Lagrangian fluid finite elements not satisfying the inf-sub condition may have additional zero energy modes [59, 62] like the ones presented in Figure 4.2. Secondly, even if mixed formulation fluid elements that satisfy inf-sub condition are utilized, in almost incompressible state, problem becomes ill-conditioned. Therefore, methodology given in Section 4.3.2 can only be applied with a local eigenspace solution for subdomains [105].

However, even if it is approximate, an Eigen problem solution for each subdomain is an expensive computation. On the other hand, additional zero energy modes (other than the rigid body modes) of a fluid element can be predicted from its element formulation [59]. Hence, a semi-heuristic deflation vector generation algorithm based on a set of known zero energy modes is proposed.

In order to follow the reasoning behind the recursive deflation vector generation strategy, solution domain Ω is partitioned into N_s number of non-overlapping subdomains Ω_j^l such that each subdomain composed of a single homogeneous material where $\cup_{j=1}^m \{\cup_{l=1}^{j_k} \Omega_j^l\}$. Let V_j be a priori known set of zero energy mode of

the material j for a specific type of finite element formulation as $V_j = [\mathbf{v}_0 \mathbf{v}_1 \dots \mathbf{v}_n]$. Therefore;

$$Z = \bigcup_{k=0}^{N_s} Z^k \text{ where } Z^k = Z_{jl} = [\mathbf{z}_0 \mathbf{z}_1 \dots \mathbf{z}_n] \quad (89)$$

where

$$\mathbf{z}_i = \sum_{e_q \in \Omega_j^l} \mathbf{v}_i \quad (90)$$

where e_q represents the q^{th} element. In other words, each \mathbf{z}_i in Z_{jl} is obtained by the assembly of pre-selected vector \mathbf{v}_i over all elements of subdomain Ω_j^l . Hence, a set of arbitrary deflation vectors is obtained and by utilizing these deflation vectors, projection space can be defined as $P = I - A\tilde{Z}(\tilde{Z}^T A\tilde{Z})^{-1}\tilde{Z}^T$. In summary, the main idea behind this deflation vector generation approach is based on the assumption that the standard assembly of each element's a priori selected zero energy modes according to the element connectivity information of each subdomain would build up a projection space for that subdomain.

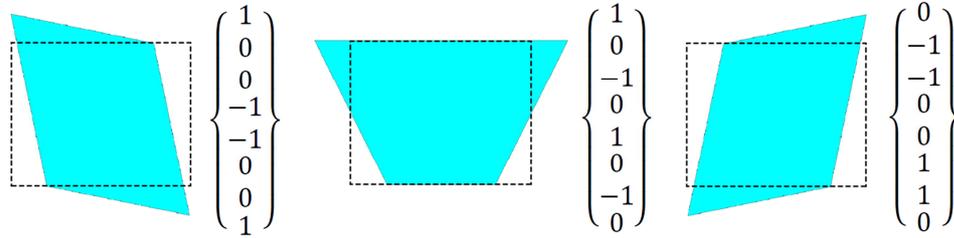


Figure 4.2. Some basic a priori known zero energy modes

Some sample vectors of V_j (a priori known zero energy modes) for the pure displacement based fluid element from its formulation [59] are presented in Figure 4.2. Since, u/p based mixed fluid formulation satisfies the inf-sub condition and rest of the small valued eigenvalues cannot be known without a full rank-revealing Eigen solution, zero energy modes of pure displacement based fluid formulation could be utilized also for u/p based mixed fluid formulation.

4.3.4. Further Heuristics; Deflation Vector Assembly

Although both of the recursive and semi-heuristic deflation methods are given as the aggregation of the deflation vectors for each subdomain, apparently one more alternative for the methods discussed in Section 4.3.2 and 4.3.3 can be obtained by a further heuristic. Alternatively, computed eigenvectors of corresponding eigenvalues that are close to each other with a tolerance can be assembled into a single vector even if they are belong to different subdomains. Proposed additional heuristic of assembly can be expressed as follows;

$$\tilde{\mathbf{Z}} = \bigcup_{k=0}^{N_s} \tilde{\mathbf{Z}}^k \text{ where } \tilde{\mathbf{Z}}^k = \tilde{\mathbf{Z}}_{jl} = [\tilde{\mathbf{z}}_0 \tilde{\mathbf{z}}_1 \dots \tilde{\mathbf{z}}_n] \quad (91)$$

where

$$\tilde{\mathbf{z}}_i = \sum_{j=0}^{N_s} \sum_{k=j+1}^{N_s} \sum_{l=0}^{n_j} \sum_{m=0}^{n_k} \mathbb{H}(\mathbf{z}_l, \mathbf{z}_m) \quad (92)$$

where n_j and n_k are the number of generated vectors for j^{th} and k^{th} subdomains and \mathbb{H} function defined as follows;

$$\mathbb{H}(\mathbf{z}_l, \mathbf{z}_m) = \begin{cases} (L_j^T \mathbf{z}_l L_j + L_k^T \mathbf{z}_m L_k) & \text{if } C \text{ satisfied} \\ 0 & \text{otherwise} \end{cases} \quad (93)$$

where L is the transformation matrix from local problem to global problem. However, in case of FETI family of solution methods, L designates the transformation matrix from local problem to coarse problem which is considerably smaller in size than the global domain. In Equation (93), condition C is $[\lambda_l, \lambda_m] < \varepsilon$ for ‘‘Recursive Deflation Method for Heterogeneous Problems’’ and λ designates the corresponding eigenvalue. On the contrary, in case of ‘‘Semi-heuristic Deflation Method’’, it is sufficient to check the equality of pre-selected deflation vectors.

4.4. Numerical Experiments

In order to investigate the performance of the deflation method on the specific problem considered in this study, several problems are first validated and solved with different deflation vector generation methods mentioned in Section 4.3. Although validations and solutions were carried out for the whole time domain, performance data presented in this section belongs to the first time step of the analysis in order to eliminate the effects of Krylov subspace utilization methodology [41] during the time history analysis.

4.4.1. Water Tank Problem

The schematic representation of the water tank with flexible walls problem accepted as one of the benchmark problems of acoustic fluid-structure interaction analysis is presented in Figure 4.18. The model has fluid elements contained in flexible walls. The base of the tank is restrained fully at one end and restrained only in vertical direction at other base nodes. A gradually rising force $P(t)$ was applied from one of the flexible walls.

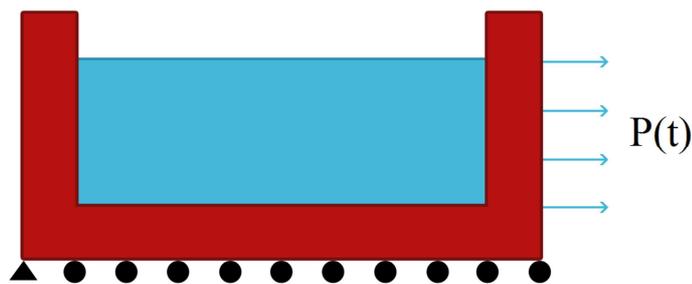


Figure 4.18. Water tank with flexible walls

Several models are created and tested where the fluid elements are modeled with quadrilateral finite elements with either displacement based (u) formulation or displacement/pressure (u/p) based mixed formulation. Each model is partitioned into

subdomains $\Omega^{(s)}$, $s = 1, \dots, N_s$ as in Figure 4.4 either $N_s = 4$ or $N_s = 42$, with similar sizes and homogeneous material properties.

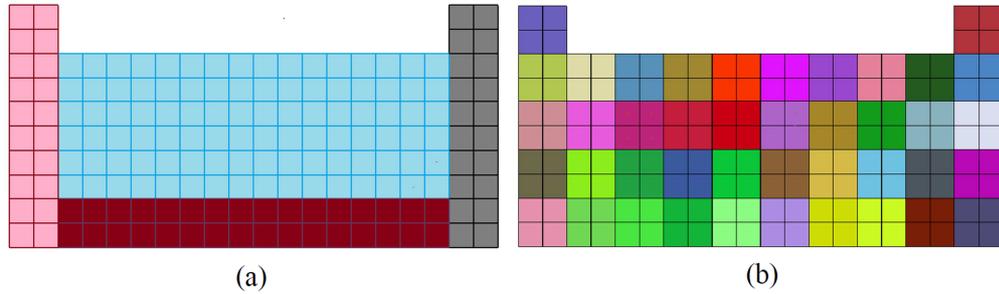


Figure 4.4. Water tank problem partitioned into (a) 4 and (b) 42 subdomains

Aforementioned deflation vector generation algorithms, PCG variants and FETI family solvers are implemented in MATLAB by using sparse storage definitions. Deflated PCG is implemented as it is given in Table 4.1 by adding condition number estimation functionality as in [90].

Through the discussion of results, SD, RD and SH are used as abbreviations for “Subdomain Deflation Method” given in Section 4.3.1, “Recursive Deflation Method for Heterogeneous Problems” given in Section 4.3.2 and “Semi-heuristic Deflation Method” given in Section 4.3.3, respectively. SD-0, SD-1 and SD-2 designate the order of approximation for subdomain deflation method. In order to utilize RD for this specific problem, it is required to replace the geometric rigid body modes computation with QR decomposition. For SH, pre-selected vectors, V_j are chosen from the element formulation of the pure displacement based fluid element. For RD and SH, two alternative methods that rely on aggregation (AGG) and assembly (ASM) are also investigated as given in Section 4.3.4.

Deflation methods insert an additional layer of computation which is the projection to a better conditioned solution space. Therefore, its efficiency relies on the improvements achieved by the projection and the additional computational/memory requirements. Hence, there are several parameters to be utilized as performance

indicators. Operation counts that are one of the main performance indicators composed of the operation counts (additions + multiplications) required for the projection/correction steps in Figure 4.1 and the computations during the iterations for deflated system solution [109]. Apparently, operation counts include any computation during the solution and in general linearly depend on the iteration counts. Another important performance indicator is the additional memory requirement for deflation methods which is composed of the storage required for deflation vectors, factorization of E and resultant matrix KZ defined in Equation (79). Obviously, the number and the sparsity of deflation vectors are the main actors for the additional memory required.

Although extensive numerical experiments are held, in order to exhibit the basic behavior and performance of aforementioned methods, three different cases of water tank problem are selected. First case demonstrates the details of the solution with u-displacement based fluid formulation. In this case, the relationship between deflation vector count and iteration counts are discussed and operation counts and additional memory consumption for each deflation method are presented. In the second case, fluid formulation is chosen to be almost incompressible, therefore u/p-mixed formulation is used. As an ill-conditioning indicator, condition numbers and change in condition numbers for each deflation method is presented and its relationship with iteration counts is discussed. As a third case, solution method is changed to FETI-2 algorithm instead of PCG iterations and the change in the performance indicators are investigated for different number of subdomains.

4.4.1.1. Model with Pure Displacement Based Fluid Elements

In this case, water tank problem discretized with u-displacement based fluid elements in Figure 4.18 is solved with FETI-2 by utilizing the standard FETI-2 algorithm and different deflation vector generation methods. First drawback of deflation methods is the necessity to store deflation vectors and some additional relatively small-sized

matrices through the iterations. Depending on the deflation vector generation method utilized generated number of vectors is different as it is represented in Figure 4.5. Standard FETI-2 procedure equipped with a QR decomposition based deflation vector generation produces the minimum number of vectors (exactly, 6) for the coarse problem.

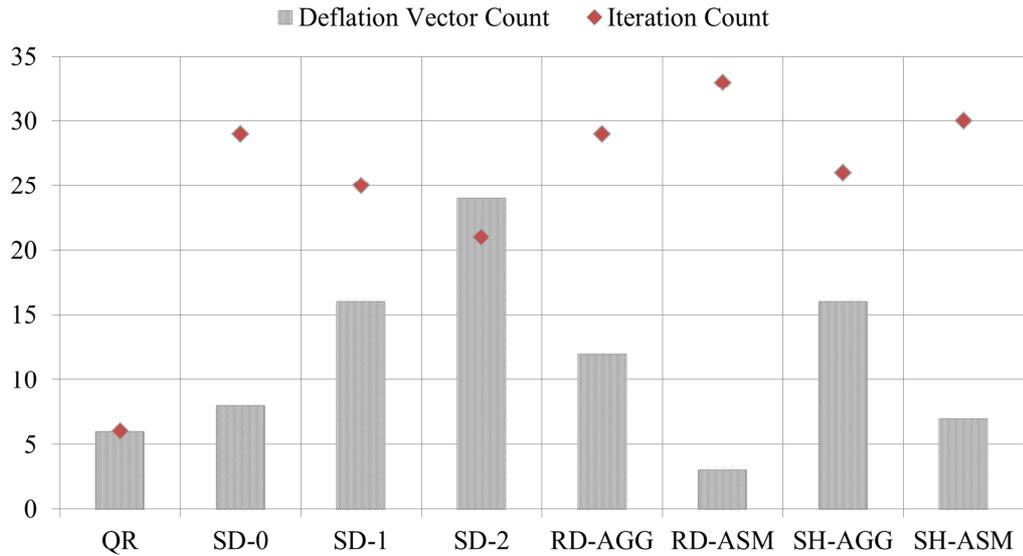


Figure 4.5. Deflation vector counts and iteration counts

Number of deflation vectors generated with SD method is directly proportional to the subdomain count which is 4 for this case and the fact that problem is in 2D. Therefore, 8, 16 and 24 deflation vectors are generated by constant, linear and second order variants of SD method. RD method generated 12 deflation vectors which are composed of 3 rigid body modes for each of the subdomains. Since, SH method relies on fixed number of pre-selected deflation vectors for each finite element, 3 vectors are generated for each of the 3 solid subdomains and 7 vectors for the single fluid subdomain. Therefore, 16 deflation vectors are generated totally when SH method is utilized. In ASM variants of these methods, generated vectors are assembled to 3 and 7 vectors for solid and fluid subdomains, respectively.

Main purpose of the deflation is to remove the most extremal eigenvalues from the system. By utilizing a fully rank-revealing algorithm like QR decomposition which exactly targets the extremal eigenvalues, there is no need to compute many deflation vectors. In contrast, for deflation methods mentioned in this study, the distribution and the number of eigenvalues are unpredictable for especially floating fluid domains. Therefore, it is necessary to generate at least enough number of deflation vectors to span a space in which the extremal modes are sufficiently approximated. Iteration counts obtained for each of the deflation vector generation method presented in Figure 4.5 in correspondence with the reasoning mentioned above. When the deflation vector generation method becomes more approximate, the number of vectors required to obtain a similar performance is increasing. Besides, it is obvious that when the number of deflation vectors is increasing, iteration counts are decreasing independently from the generation strategy. It is also expected due to the fact that additional linearly independent vectors enlarge the projection space spanned.

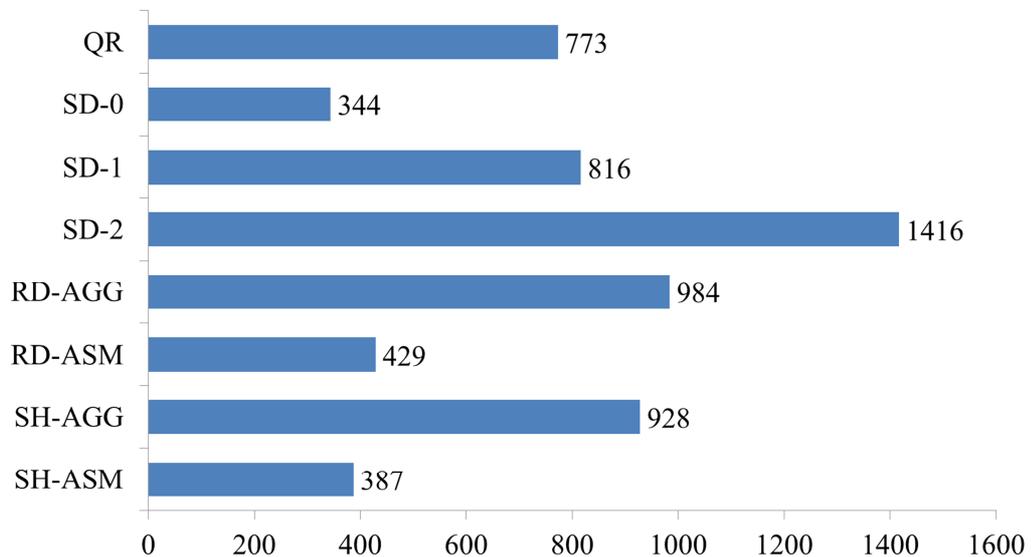


Figure 4.6. Additional memory requirements (in # of non-zeros)

Memory requirements of the deflation methods are another important aspect for the efficiency. Not only the number of deflation vectors but also the sparsity of the

vectors generated is significant. Obviously, vectors generated by methods originated from a fully rank revealing algorithm like QR decomposition have drastically lesser number of non-zero values. Although their number of vectors is smaller, in overall, as it is presented in Figure 4.6, standard FETI-2 and RD-AGG requires moderate memory storages. SD methods generate moderately sparse vectors but their memory requirements are increasing rapidly with an increase in the order of approximation. Memory requirements of standard SH method is similar to SD method but as it is in SH-ASM and RD-ASM, memory requirements can be limited by utilizing ASM heuristics.

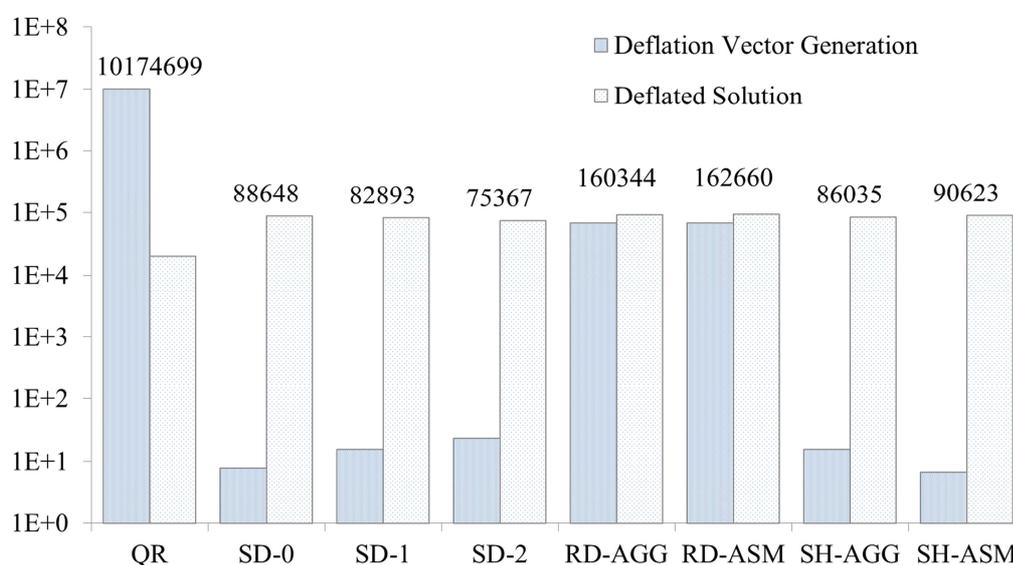


Figure 4.7. Operation counts for deflation vector generation, deflated solution

Operation count is another important factor on the efficiency of deflated solution with different deflation vector generation approaches. In Figure 4.7, operation counts for deflation vector generation and deflated solution is given separately. Overall operation counts are given on top of each bar couple. Apparently, when QR decomposition takes place, operation counts for deflation vector generation stage are dominant. On the other hand, in case of arbitrary deflation vector generation methods such as SD and SH methods, deflation vector generation requires considerably less number of operation counts.

4.4.1.2. Almost Incompressible Case by CG Variants

In order to investigate the improvements achieved by mentioned deflation methods for almost incompressible case, water tank problem presented in Figure 4.18 is solved by Conjugate Gradient (CG) with and without an Incomplete Cholesky (IC) preconditioner for a fluid material with Poisson's ratio of 0.499 and 0.49999.

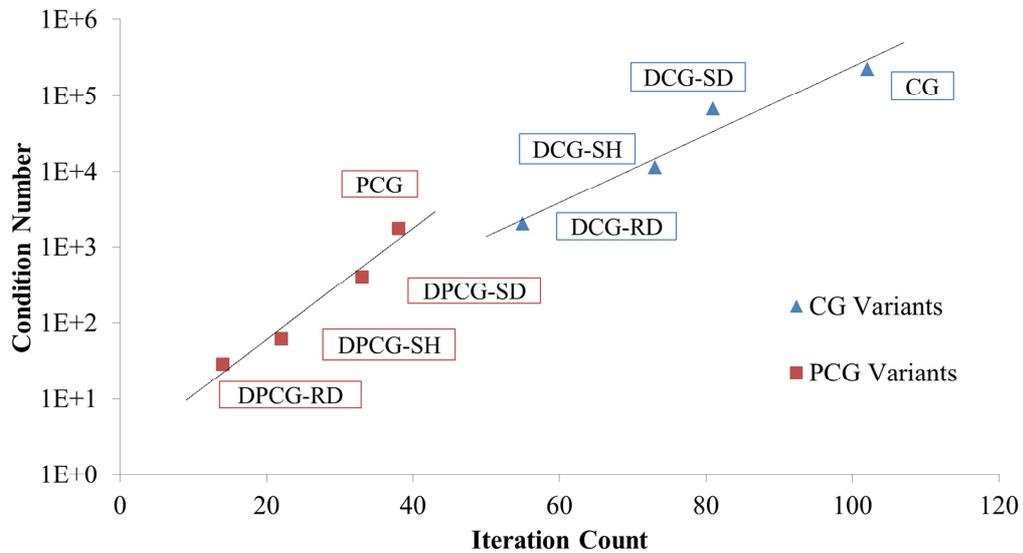


Figure 4.8. Solution by CG variants w/o $PC = IC$ ($\nu = 0.499$)

First set of results presented in Figure 4.8 is obtained from the first time step solution of water tank problem for a fluid material with Poisson's ratio of 0.499. First of all, the correlation in between condition number and iteration counts is apparent and hence iteration counts decrease with the reduction in condition number. In addition to that, as expected, preconditioning reduces the condition number drastically. However, for both preconditioned and non-preconditioned cases, with the utilization of any deflation method mentioned convergence of the CG/PCG method is improved due to the reduction in the estimated condition number. Obviously, in correspondence with the better representation of extremal eigenspace, the improvement achieved increases. For instance, SD which is an arbitrary deflation vector generation method behaves worst in contrast to RD. On the other hand, SH which is also an arbitrary deflation vector generation method improves the

convergence rate more than the popular SD method and this improvement is even considerably closer to the improvement achieved by RD.

Same problem is solved with same solution parameters but only Poisson's ratio (ν) is shifted to 0.49999, in short, to the almost incompressible state. Iteration counts and condition number estimates are presented in Figure 4.9. As expected condition number estimates are drastically increased due to ill-conditioning of the problem at the edge of incompressibility. Accordingly, it is reflected to solution as a drastic increase in iteration counts.

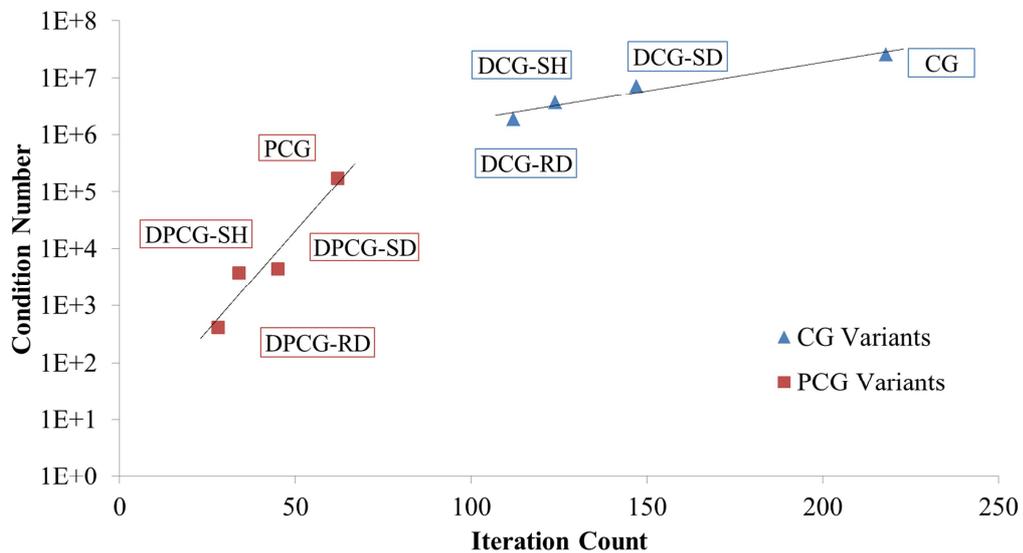


Figure 4.9. Solution by CG variants w/o PC = IC ($\nu = 0.49999$)

When Figure 4.9 is compared with Figure 4.8, for the almost incompressible case, reduction in estimated condition numbers for non-preconditioned solutions is degraded. On the contrary, it is improved for the preconditioned solutions. Obviously, deflation is not as effective as preconditioning but it is a complementary method for ill-conditioned systems. Therefore, both preconditioning and deflation is required to remove the eigenspace corresponding to the smallest and largest eigenvalues formed as a result of the almost incompressibility.

4.4.1.3. Almost Incompressible Case by FETI Variants

Fully Lagrangian representation of acoustic fluid-structure interaction problems requires considerable amount of computational power. Therefore, instead of DPCG, FETI variant solvers are of interest for this type of specific problems. In order to investigate the computational efficiency, memory requirements and overall performance improvement in practice, deflation methods mentioned in previous section is implemented on FETI-2 solution algorithm. Since FETI family of solvers are highly scalable, in general, the number of subdomains chosen to be equal or higher than the number of processors. Therefore, change in operation counts and memory requirements of deflation methods utilized are significant.

Water tank problem in almost incompressible state ($\nu = 0.49999$) is solved by FETI-2 solution method which utilizes Dirichlet Preconditioner [86] as preconditioner of coarse problem and QR factorization for computing the zero energy modes, R of floating subdomains. Deflation is implemented into the solution by using matrix C defined in Equation (75).

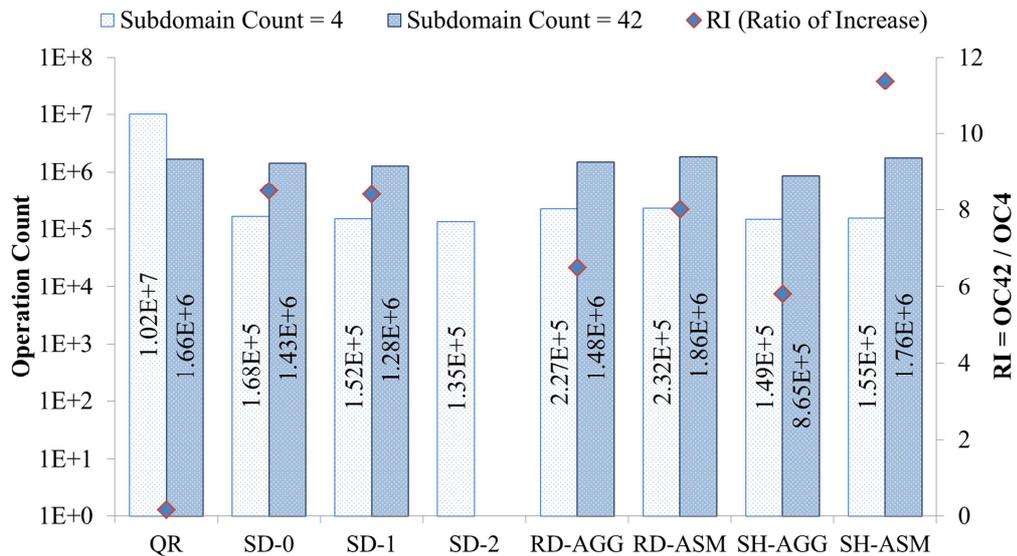


Figure 4.10. Operation counts for different number of subdomains

In Figure 4.10, operation counts for deflation method with different deflation vector generation approaches are given for 4 and 42 subdomains and the change in operation counts are demonstrated as RI (the ratio of increase in operation counts) from 4 subdomains to 42 subdomains. An important observation is that solution with SD-2 did not converge to a solution for 42 subdomains although it was converged for 4 subdomains. Since, SD method does not explicitly generate orthonormal deflation vectors, arbitrarily generated deflation space caused the solution to diverge. Unpredicted behavior of arbitrary deflation generation methods is apparent in this case. As the number of subdomains is increased from 4 to 42, the size of the subdomains is about 10 times smaller than the ones in 4 subdomain case. Since in FETI-2, QR factorization for computing the zero energy modes, R of floating subdomains are computed in subdomain level, operation counts are reduced about 60 times. Arbitrary deflation vector generation methods like SD and SH, however, require about 6 to 10 times more operations since the number of vectors increases proportional to the subdomain count which directly affects the operation counts of deflated system solution. In case of methods RD-AGG and RD-ASM, although the operation counts for deflation vector generation are quite similar, due to the drastic increase in number of vectors, overall operation counts are increased about 8 times.

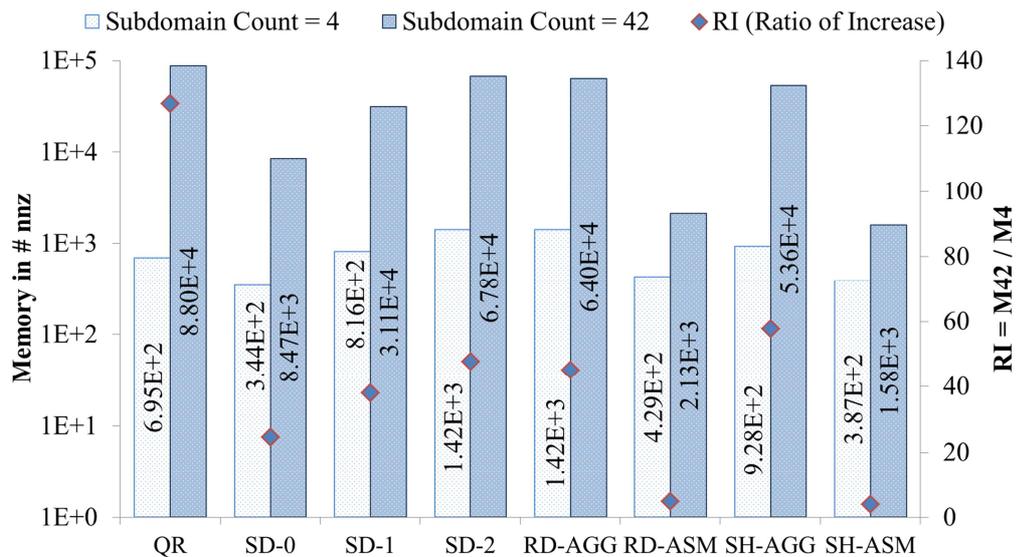


Figure 4.11. Additional memory requirements for different number of subdomains

Additional memory requirements of the water tank model partitioned into 4 and 42 subdomains and RI (the ratio of increase in memory) are presented in Figure 4.11. Sparsity of the generated vectors and the increase in the number of vectors are the main determinant on this figure. For example, although the number of generated vectors for SD-0 and RD-AGG increased about 10 times, memory requirements of RD-AGG increased about 45 times that is almost 2 times higher than the increase in SD-0. Limited memory usage property of ASM heuristics is apparent again.

4.4.2. Actual Dam Model

As a final numerical experiment, Pine Flat Dam model (Figure 4.12), as being utilized as a benchmark problem in literature [110, 111], is analyzed using the deflation methods mentioned.

Pine Flat Dam [111] is a concrete gravity dam on the Kings River of central California. It is 122 m high and the freeboard of reservoir is at level of 116 m. The upstream face of the dam is nearly vertical having a slope 0.05H:1V, while the inclined downstream side has slope of 0.78H:1V. The dam is modeled in a similar manner as in [110] with the same material properties. Finite element mesh of the dam that is composed of 1 m to 15 m sized elements is given in Figure 4.12.

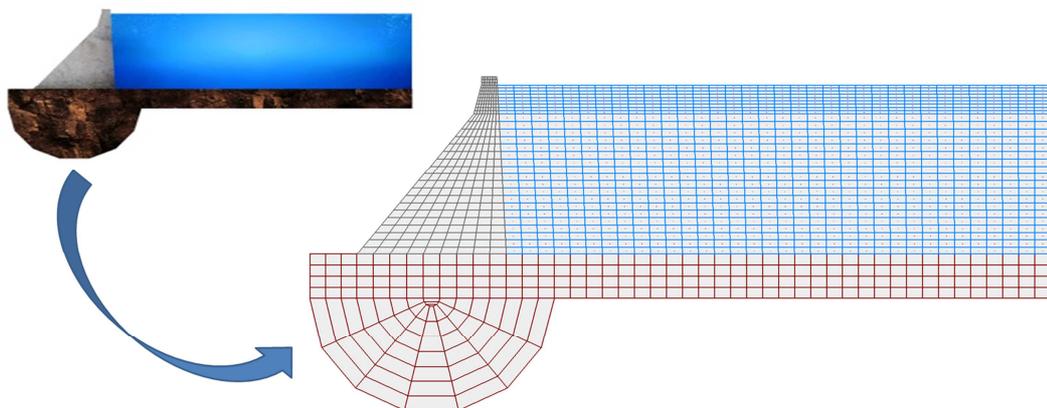


Figure 4.12. Pine Flat Dam and its finite element mesh

Base width of the concrete block is 95.8 m and top width is one-tenth of the base. Modulus of elasticity, Poisson's ratio and density for concrete block is taken as 22.5 GPa, 0.2 and 25 kN/m³, respectively. Foundation rock is modelled as 30 m depth layer of rock and a 101.8 m diameter half circle rock toe. Modulus of elasticity, Poisson's ratio and density of foundation rock is chosen as 68.94 GPa, 0.33 and 26 kN/m³. Bulk modulus, Poisson's ratio and density of the water are taken as 2.541 GPa, 0.49999 and 999.97 kg/m³, respectively.

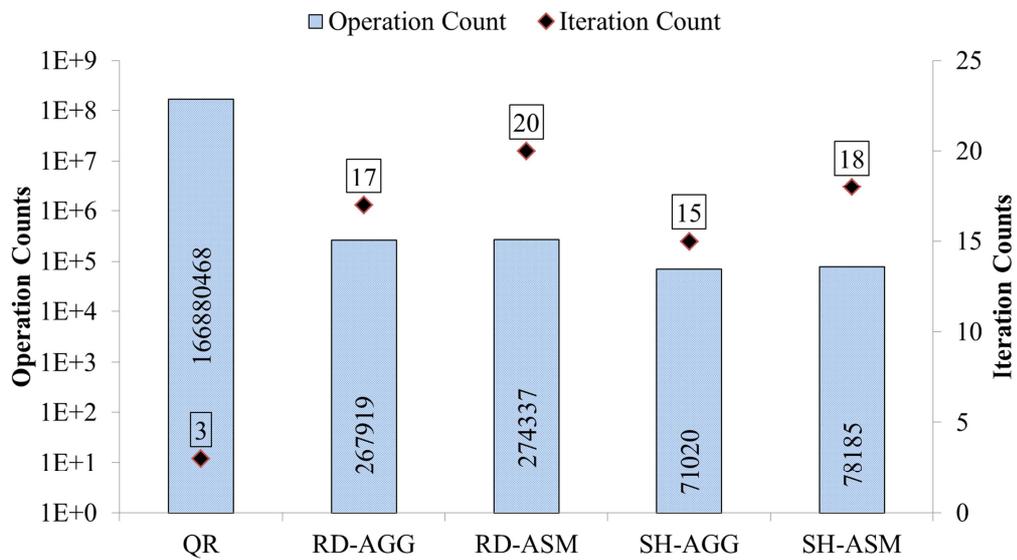


Figure 4.13. Solution with displacement based fluid elements

In order to demonstrate the practical improvement on an actual case, reservoir is modelled with either pure displacement based fluid elements or mixed formulation based fluid elements. In the first case, only compressibility of the water is taken into account, however, in second case, viscosity of the water is considered, too. Results obtained by the utilization of the proposed deflation vector generation strategies for these two cases are given in Figure 4.13, Figure 4.14 and Figure 4.15, respectively. As it is shown, obtained results are in correspondence with the ones obtained for water tank problem. However, in this case, solutions with the projection space obtained from SD-0, SD-1 and SD-2 are diverged.

Operation counts and iteration counts obtained for each solution is given in Figure 4.13 and Figure 4.14. Although iteration counts are increased about 5 to 10 times, operation counts are reduced about 500 times for RD variants. Reduction in operation counts for semi-heuristic deflation vector generation methods is more than 2,000 times for the solution with pure displacement based fluid elements and about 1,000 times for the solution with mixed formulation fluid elements. This is expected because pre-selected vectors are obtained from pure displacement based fluid element formulation.

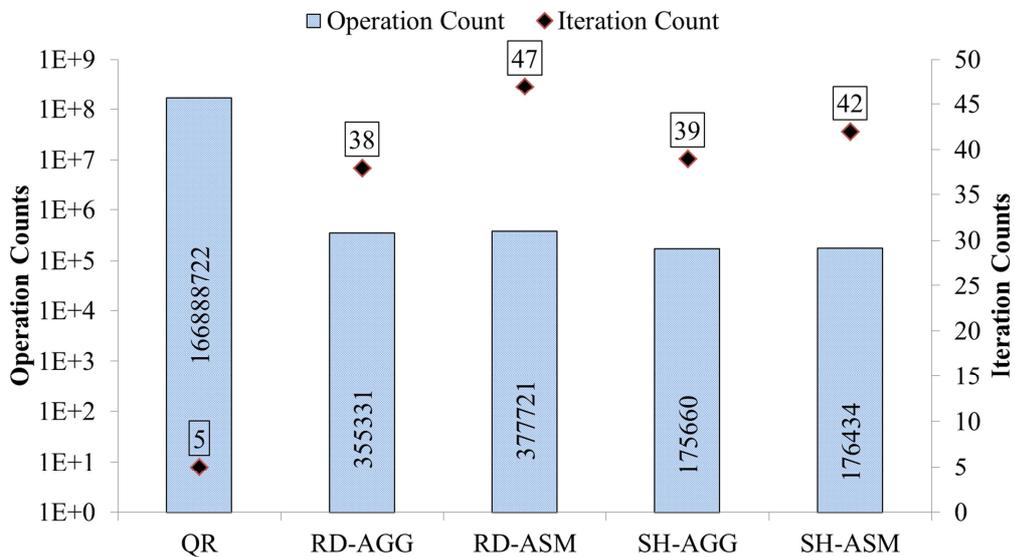


Figure 4.14. Solution with mixed formulation fluid elements

Obviously, semi-heuristic deflation vector generation method is even 2 to 4 times efficient than the recursive deflation vector generation method for heterogeneous problems for the case of dam-reservoir interaction problems.

According to the memory storage requirements presented in Figure 4.15, fully rank-revealing Eigen solution based methods like QR and RD require more memory space than the proposed arbitrary vector generation method, SH. Except from the SH methods, in all other methods, memory requirements for the solution with mixed formulation fluid elements are quite lower than the one with displacement based

fluid formulation fluid elements because of the use of reduced number of deflation vectors. In case of SH, memory requirement is not changing in between different element formulation due to the fact that same vectors are selected for the both cases. Apparently, SH is the most memory efficient method among the aforementioned methods.

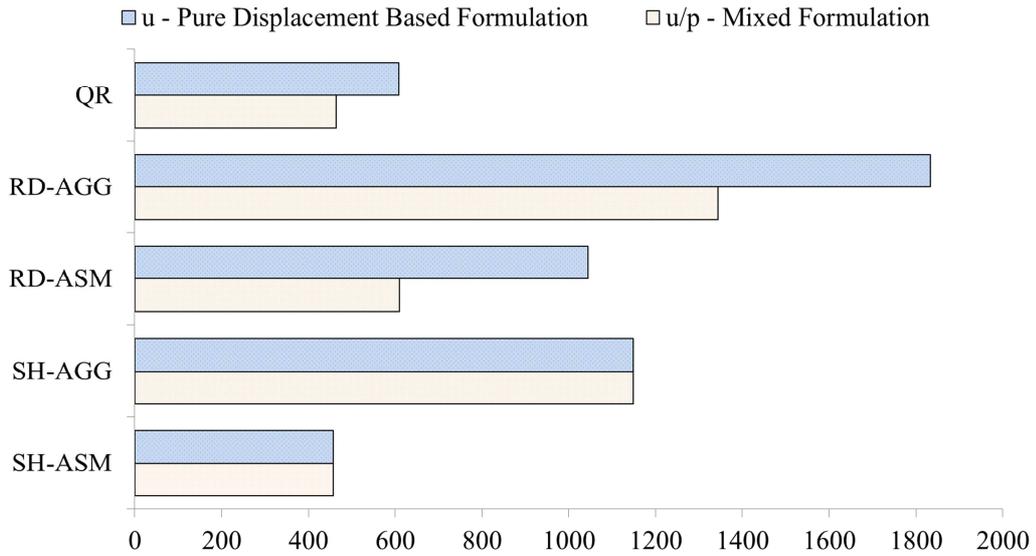


Figure 4.15. Additional memory required (in # of non-zeros) for the solution

As an arbitrary deflation vector generation strategy, first drawback of the proposed semi-heuristic deflation vector generation method is that for each subdomain of a material, n deflation vectors are computed without knowing their efficiency during the iterative solution. However, the generation of deflation vectors is comparatively straightforward and parallelizable since it is based on discrete computations on each pre-selected vector. It is also cheap to store. Finally, the improvements obtained with the generated deflation space are comparable to previously mentioned methods. In addition to that, assembly heuristic is suggested in case of increasing subdomain counts and memory limited hardware.

4.5. Conclusion

In this chapter, monolithic fully Lagrangian solution of the dam-reservoir systems is presented. The analysis is performed with FETI family iterative solvers in order to overcome the computational resource requirements. Sources of ill-conditioning are presented and deflation method is utilized to improve the convergence rate of the solution. Different deflation generation methods are presented and finally a novel approach for deflation vector generation is proposed. Two different benchmark problems are analyzed with different conditions and performance of the mentioned deflation methods are discussed in terms of iteration counts, additional memory requirements and operation counts.

Considering the test cases examined in this chapter, the main conclusions of this chapter are as follows:

- The monolithic fully Lagrangian solution of dam-reservoir systems can be solved with iterative methods such as PCG (preconditioned conjugate gradient) method and FETI family methods. Adding another projection level with deflation method enhance the solution in terms of number of iterations and operation count.
- Among the examined deflation methods, subdomain deflation vector generation method (SD) is the fastest in terms of operation count but the solution did not always converge especially in an actual dam-reservoir interaction problem.
- Solutions with recursive deflation vector generation method for heterogeneous problems (RD) converged to a solution for all cases but it was the most costly method especially for the fluid subdomains.
- A priori known zero energy modes of a finite element formulation can be utilized to generate a projection space. Deflation method utilized with projection space obtained from the proposed semi-heuristic approach converged to a solution for all cases. Semi-heuristic deflation vector

generation method is a cheaper alternative among the other deflation vector generation methods discussed.

- Additional memory requirements of the generation methods can be limited by utilizing an assembly heuristic with a trade off from the number of iterations and overall performance.
- Proposed semi-heuristic deflation vector generation method and further assembly heuristic method are not problem dependent and successfully applicable to dam-reservoir interaction problems.

CHAPTER 5

PARALLEL IMPLEMENTATION OF SOLUTION FRAMEWORK

5.1. Introduction

This chapter focuses on the parallel implementation of the implicit monolithic solution algorithm for dynamic analysis of dam-reservoir systems. Parallel implementation is developed with C++ programming language and parallel solution of system equations is mainly handled by PETSc library [112]. Developed algorithms are added to the general purpose finite element framework, Panthalassa [113]. During the test runs, FETI-DP solution algorithm [114] available in PETSc library is utilized with different deflation vector generation methods. The effect of using different finite elements, different partitioning approaches, and different types of fluid-structure interaction problems on the efficiency and scalability of the developed solution framework are discussed in detail.

5.2. Implementation

The solution framework is composed of two main parts. First part, named as Zargana, is responsible from the pre-processing and post-processing steps of computations before and after the linear algebraic solution for the system of equations. While the partitioning, equation numbering and the generation of system equations are considered as the pre-processing computations, computation of element forces and stresses are the post-processing computations. Orkinos solver is the second part of the solution framework that performs the algebraic solution of system of equations with FETI-DP solution algorithm. Zargana and Orkinos are programmed as dynamic link libraries (i.e. dll) with C++ programming language.

Moreover, Eigen linear algebra library [115] and PETSc [112] are utilized for data storage and high performance numerical computations, respectively. Both Zargana and Orkinos are developed to be a part of the general purpose finite element analysis environment, Panthalassa [113]. Panthalassa environment provides common algorithms and functionalities for a typical finite element program and it allows addition of different solution algorithms, finite elements, and material models as plugins in the form of dynamic link libraries. The finite elements utilized in this study, i.e. displacement based and mixed formulations of Q4, Q9, H8 elements (APPENDIX A), are added to the element library of Panthalassa.

The main steps of the solution by Zargana are presented in Figure 5.1. Solution starts with the execution of processes with same input file. The input file may or may not contain partitioning information of subdomains. If partitioning information is not provided, the solution domain is partitioned by utilizing METIS partitioning library [116]. The number of partitions is taken as equal to the number of processes. Then, each process read the data related to its own subdomain from input file to its local memory. Equation numbering of each subdomain is carried out such that internal DOFs are numbered before the interface DOFs. Any matrix reordering methods [117, 118] for equation numbering are not utilized since suitable equation numbering method for the chosen solution method is automatically handled by PETSc [112]. Following that, mapping in between the local (i.e. subdomain level) numbering and global (i.e. overall solution domain level) numbering is formed. Then, subdomain level stiffness and mass matrices are assembled for each subdomain. Loads applied to each subdomain are also assembled in a similar way. If it is applicable, any boundary conditions on the subdomain are applied to the subdomain system of equations as in the standard finite elements procedure [18, 44]. Before initiating the solution on Orkinos solver, deflation vectors should be generated by utilizing either the partitioning information or element level data or subdomain level data depending on the type of the deflation vector generation method. These vectors are generated once at the beginning of the solution. All the computations at the pre-processing step are performed simultaneously on each process.

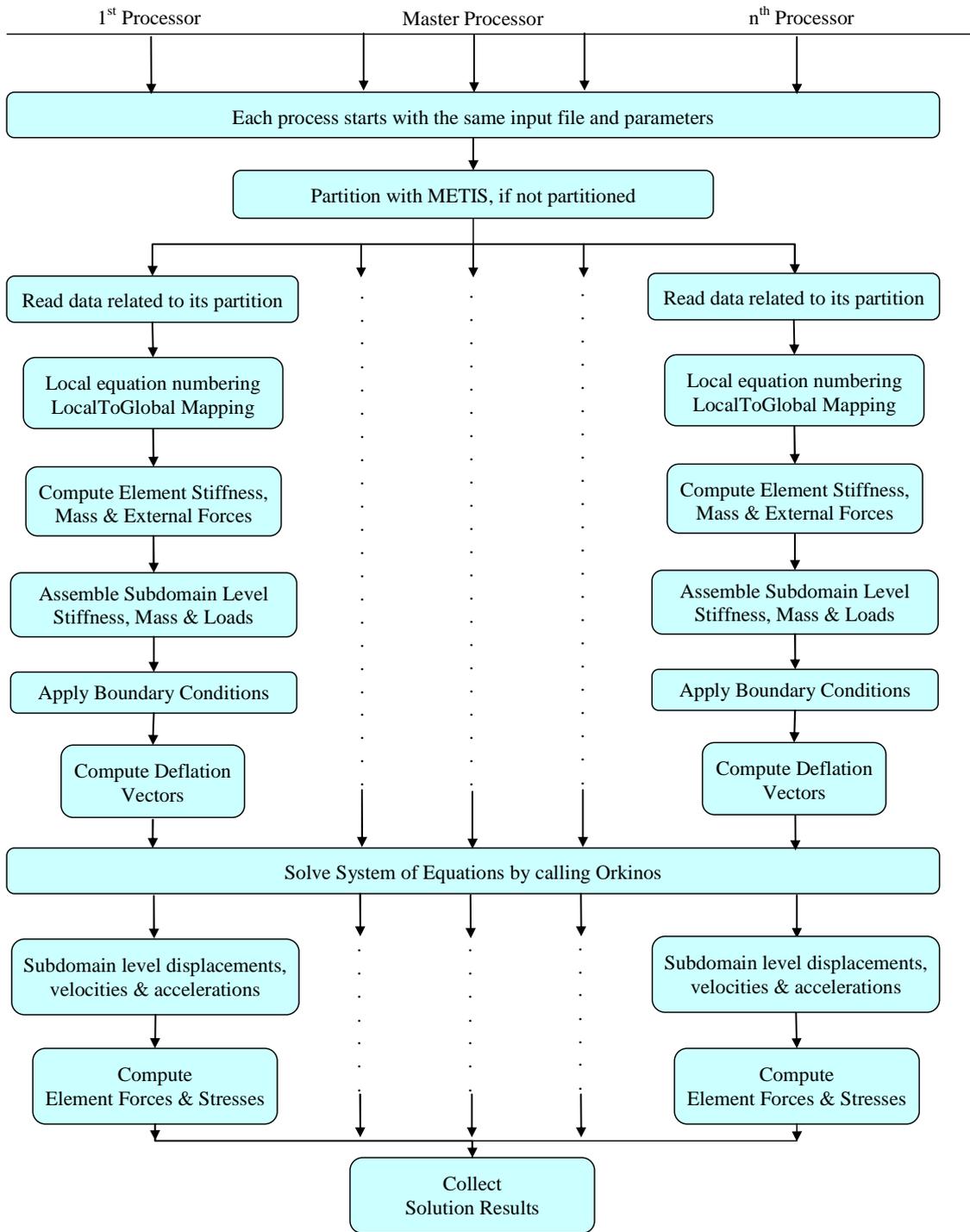


Figure 5.1. Flowchart for overall solution on Zargana

Subdomain level data; stiffness and mass matrices and load vectors are passed to the corresponding instance of Orkinos solver with accompanying the mapping information and deflation vectors of the corresponding subdomain. At this step, aforementioned data are passed to each processor by reference; therefore any memory duplication is prevented. Flowchart for the solution of equations by Orkinos is demonstrated in Figure 5.2. Solution initiates with the initialization of PETSc environment on each processor. PETSc library contains data structures for both storage and parallel linear algebra routines. Besides, it handles the data transfers required for the parallel execution and gathers statistics about the execution of solution [112]. Initially, the globally distributed system is defined by using the subdomain level data and corresponding mappings. The globally distributed system is not actually assembled but only the dependency in between the data of contributing subdomains is described. In case of a data required from another subdomain (for example, stiffness contribution of a subdomain to an interface DOF), this definition reveals the owning process of this data and its location on the memory of its owning process. After the global system definition, the required memory for right hand side and solution vectors are allocated beforehand to decrease repetitive memory allocations. In case of a solution with deflation method, deflation vectors provided by Zargana are orthonormalized by using Gram-Schmidt iterations [119]. These iterations are carried out in parallel since every subdomain has its own deflation vectors. Thus, this step does not require any communication among the processes. Orthonormalized set of vectors are declared as near null space in PETSc library.

Solution of the global distributed system is handled by the FETI-DP solution algorithm [112, 120, 121] available in PETSc library. This algorithm composed of two main steps. Former is the local solution which involves the computation of the stiffness contributions for each subdomain to the interface and the preconditioner for the interface problem. Latter is the interface problem solution. PETSc has its unique flexibility to choose any solver available in its library for local system solution or interface system solution. However, in this study, local system solution is performed

by utilizing direct sparse Cholesky factorization available in PETSc library [112]. These computations are explicitly parallel which means that no communication among the processes is required. By utilizing the results of the local solutions, coefficient matrix for the interface problem and the corresponding preconditioner are obtained. Right hand side vector of the interface problem should be computed at each time step depending on the time stepping algorithm. In this implementation, implicit time integration is utilized by replacing accelerations with momentums in order to prevent the spectral drift and the obtained interface problem is solved by Preconditioned Conjugate Gradient (PCG) iterations available in PETSc library [112]. During these iterations Dirichlet preconditioner is utilized and the convergence to the solution is monitored similarly as in the serial implementation (Section 3.2.4). After the successive convergence of the interface problem solution, each process stores the displacements computed for that time step and proceeds to the next time step. In order to enhance the convergence of following solutions, the first twenty Krylov subspace vectors of the previous solution are utilized by the Fischer's algorithm of initial guess generator for repeated linear solves [122].

Finally, at the end of time steps, the solution is completed by finalizing the PETSc environment and the results obtained by Orkinos are passed back to Zargana for computing the element forces and stresses and preparing the necessary output files.

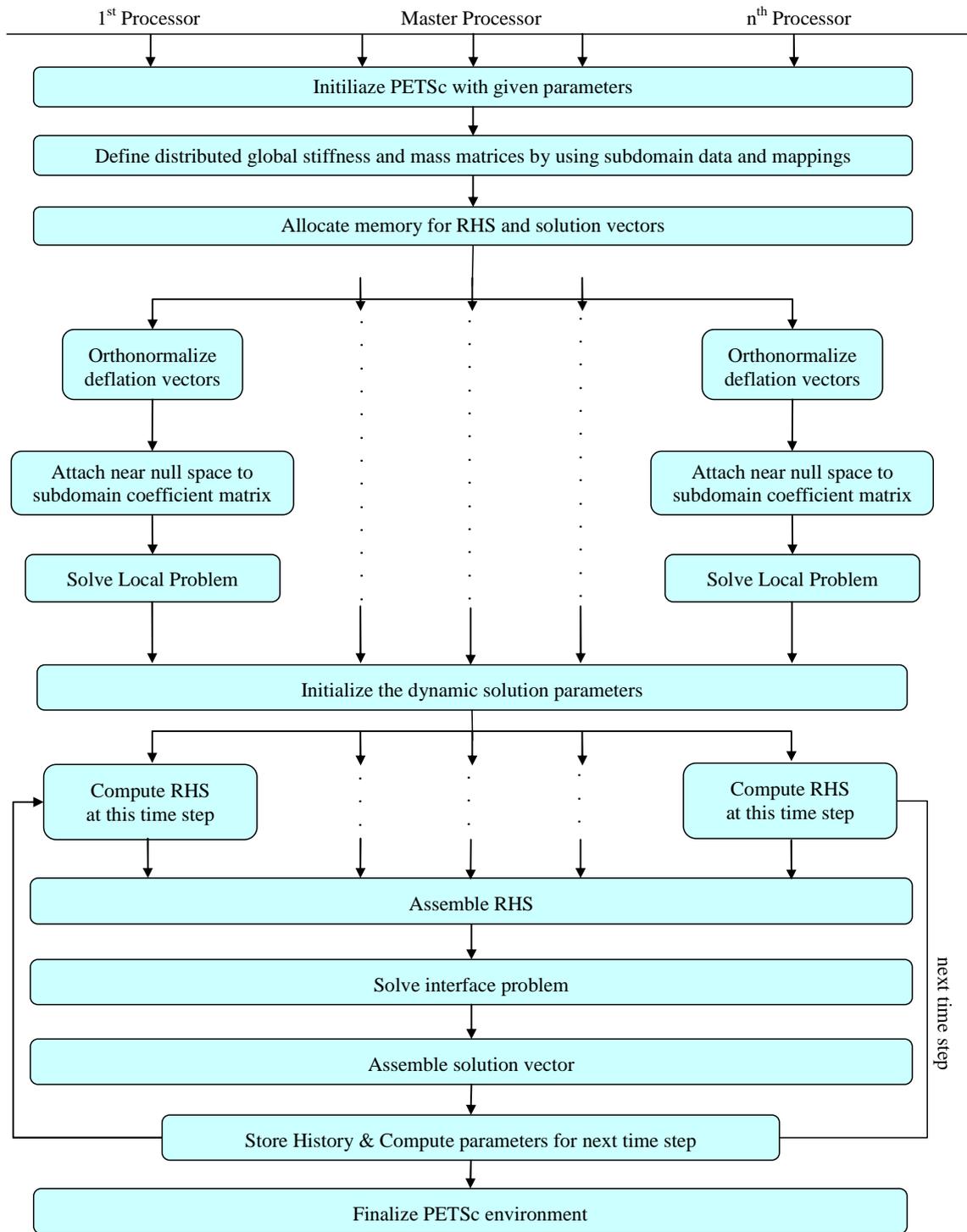


Figure 5.2. Flowchart for solution of equations on Orkinos

5.3. Numerical Experiments

The parallel solution framework is tested by analyzing several fluid-structure interaction problems on a homogeneous computer cluster. The cluster is composed of 6 identical computers each having Intel Xeon E5630 processors and 24 GB local memory. The Intel Xeon E5630 family processor has four cores that are theoretically working at 2.53 GHz and the cores share the local memory. Nodes of cluster are connected to each other with InfiniBand network switch having 40 GBit/s communication speed. The start-up latency for communication is measured as $3.37e-06$ seconds. Besides, all nodes are running Windows Server 2008 R2 HPC Edition with SP1.

The test problems involves structural models with 2D and 3D elements, different partitioning configurations, and different finite element formulations to investigate the convergence characteristics and performance of the analysis in terms of scalability and memory consumptions of different solution methods. Throughout these numerical experiments, problems are solved by FETI-DP algorithm with/without different deflation vectors. First solution method is the standard FETI-DP solution algorithm [114] available in PETSc library. In case of "standard FETI-DP" solution algorithm, no deflation vectors are utilized. However, this algorithm guarantees the non-singularity of each subdomain $K_{BB}^{(s)}$ by making sure that every subdomain has either 3 non-collinear corner nodes in 3D or 2 non-coincidental corner nodes in 2D [33, 37]. In other words, in default, this implementation selects rigid body modes (not zero energy modes) as constraints. Second solution method named as "QR" relies on the explicit computation of null spaces of each subdomain by a fully rank-revealing QR factorization algorithm (SPQR of SuitSparse library [123]). Rest of the solution methods are obtained by utilizing the deflation vector generation methods like SD-0, RD-AGG, RD-ASM, SH-AGG and SH-ASM that are already described in Section 4.4.

For all of the numerical experiments conducted in this section, pure displacement based fluid formulation is utilized because it requires more conditioning than the mixed formulation (Section 2.2.2). Finally, for simplicity, the names of different finite element formulation sets are defined as follows:

- "Q4 elements"; sQ4IP0u for solid domains and fQ4P1u for fluid domains,
- "Q9 elements"; sQ9P0u for solid domains and fQ9P1u for fluid domains,
- "H8 elements"; sH8IP0u for solid domains and fH8P1u for fluid domains.

5.3.1. Effect of Different Partitioning Configurations

Performance of parallel algorithms is highly governed by the workload balance in between processors and the amount of data transfer required for computing the solution [124]. In case of multi-physics problems such as fluid-structure interactions, partitioning can be done in such a way that each partition is composed of "homogeneous" elements and material properties. In this case, standard partitioning algorithms cannot be directly utilized since each domain having different properties must be partitioned within itself. This approach might violate the workload balancing among subdomains and increase the communication cost. On the other hand, the stiffness coefficients at the partition interfaces would be more uniform which might reduce the number of iteration for convergence [27, 86]. On the contrary, in case of heterogeneous partitioning, standard partitioning algorithms can be directly utilized. As a result, partitions that have similar workloads are obtained and the interface problem size is optimized for minimum communication requirement during the parallel interface solution.

All of the deflation vector generation methods are also applicable to the solutions of problem domains partitioned into heterogeneous subdomains. Implementation for deflation vector generation methods other than SH variants are apparent since they are all either based on partitioning information or QR decompositions. In case of SH variants; the rigid body modes of solid finite elements are aggregated in the same

way as the pre-selected zero energy modes of fluid finite elements (Section 4.3.3). Similarly, during the assembly heuristic, rigid body modes of solid finite elements are assembled with the corresponding rigid body modes of fluid finite elements (Section 4.3.4).

In order to investigate the effect of different partitioning approaches on the efficiency of different solution methods, water tank problem discretized with H8 elements (and $H/h = 16$) are analyzed statically by utilizing 8 processors. First, the domain is partitioned into fluid and solid subdomains and each domain is then partitioned manually so that each subdomain has the same number of elements. Such partitioning is called homogeneous partitioning and the subdomains are presented in Figure 5.3 (a). The other partitioning approach is to use a standard partitioning algorithm for the whole domain. In this case, the subdomain may have finite elements having different formulations and material properties. METIS partitioning library is utilized for this approach and subdomains are presented in Figure 5.3 (b).

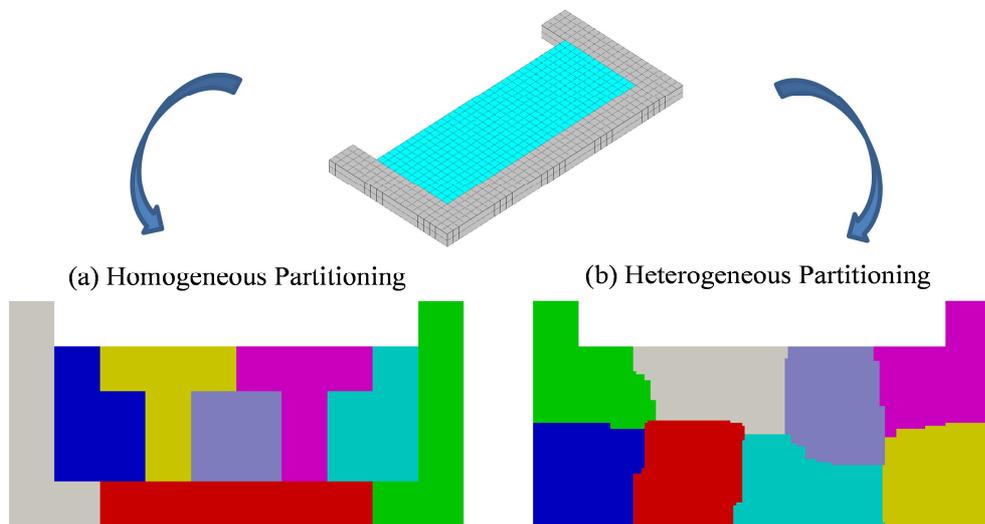


Figure 5.3. Homogeneous and heterogeneous partitioning of water tank model

Since the model has two layers of H8 elements, it has totally 21,504 finite elements which totally generate 99,081 DOFs. The minimum and maximum local problem sizes for both approaches are presented in Table 5.1. If the local problem size is

considered as the only parameter affecting the workload for simplicity, unbalancing among the subdomains can be defined as the ratio of max/min local problem sizes. Therefore, in case of heterogeneous partitioning in which METIS partitioning library is utilized, workload unbalance ratio is 1.02. On the contrary, the homogeneous partitioning is done manually and the workload unbalance ratio is 1.51. In other words, local problem solutions for each subdomain are not completed at the same time and hence some processors wait for others to initiate the interface problem solution. As it can be seen from the local solution times in Table 5.1, workload unbalancing results in a considerable increase in case of homogeneous partitioning. Even if the homogeneous partitioning is done by a special algorithm by utilizing METIS partitioning library, due to the geometrical limitations, i.e. the solid domain is composed of three narrow rectangular prisms whereas the fluid domain has a wide rectangular prism shape, it is very likely to have significant unbalance in the local solution times since the shape of the solid subdomains would be rectangular and the shape of the fluid subdomains would be more close to squares.

Table 5.1. Partitioning details for different approaches

Partitioning	Min. Local DOF Count	Max. Local DOF Count	Workload Unbalance Ratio	Local Solution Time (s)	Interface DOF Count
Heterogeneous	12726	13032	1.02	288	3894
Homogeneous	9801	14841	1.51	349	4359

Similar to local problem sizes, interface problem size is also intended to be minimized by METIS partitioning library. For example, interface problem size for heterogeneous partitioning is 3,894 whereas interface problem size for homogeneous partitioning is 4,359. Since the projection space formed by deflation vectors defines admissible constraints on the interface (i.e. primal interface DOFs), the number of dual variables is changing with the utilized deflation vector generation method. Dual variables are the interface equilibrium forces that are computed by the parallel solution iterations. Number of dual variables on the interfaces of both partitioning approaches for different deflation vector generation methods is presented in Figure 5.4. Since only the rigid body modes of subdomains are constrained, highest number

of dual variables exists in standard FETI-DP iterations. Since the RD-AGG method based on eigenvectors computed by QR decomposition, lowest number of dual variables exists in this method. Besides, when the assembly heuristic is utilized, the number of dual variables obtained is increasing since it assembles the resultant vectors on each other and degrades their quality of representing the smaller Eigen modes of the subdomain. For all deflation methods investigated in this section, representation of smaller Eigen modes are handled more successfully when the elements and material properties are homogeneous in the subdomain. For example, in case of utilizing SH-AGG method, the number of dual variables is decreased from 3,870 to 3,557 for heterogeneous partitioning approach whereas same parameter is decreased from 4,335 to 4,013 for homogeneous case. Therefore, it can be concluded that deflation vector generation methods handles homogenous subdomains slightly better than the heterogeneous subdomains.

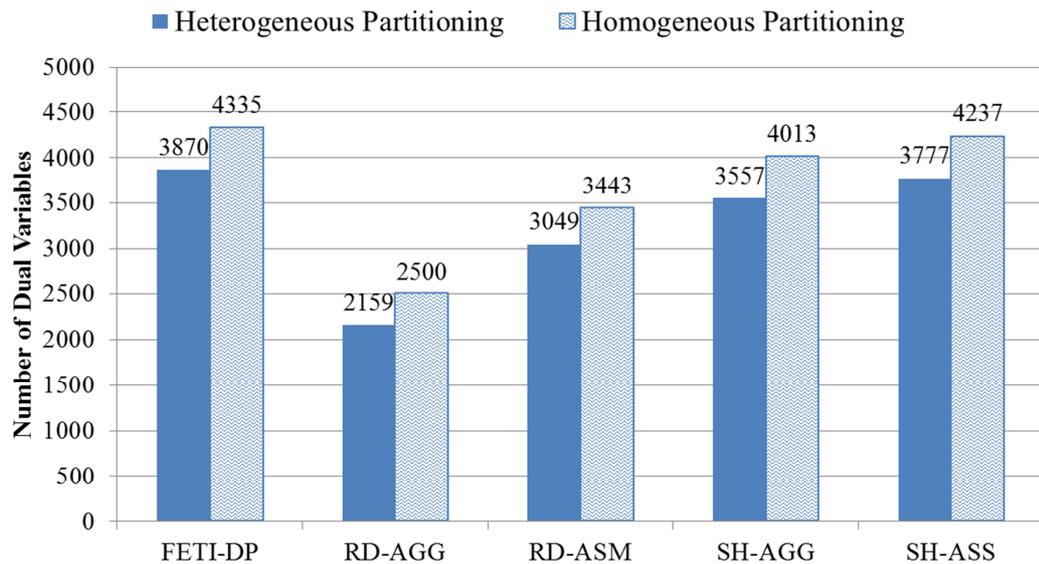


Figure 5.4. Number of dual variables for different partitioning approaches

In Figure 5.5, interface solution timings and iteration counts required for homogeneous and heterogeneous partitioning approaches are presented. For all cases, iteration counts required for the interface solution is decreased when homogeneous partitioning is utilized. Similarly, interface solution timings are also decreasing;

however, the obtained decrease in the interface solution timings is not proportional to the decrease in iteration counts due to the fact that interface solution timings also depend on the utilized deflation vector counts and the number of dual variables on the interfaces that computed during the iterations. Another factor that affects the interface solution timings is the data transfer during the solution. The total amount of memory consumption and data transfer for both partitioning approaches are presented in Figure 5.6. For all cases, the amount of data transfer and the memory consumption is increased when the homogeneous partitioning is utilized since the interface problem size is larger for this case.

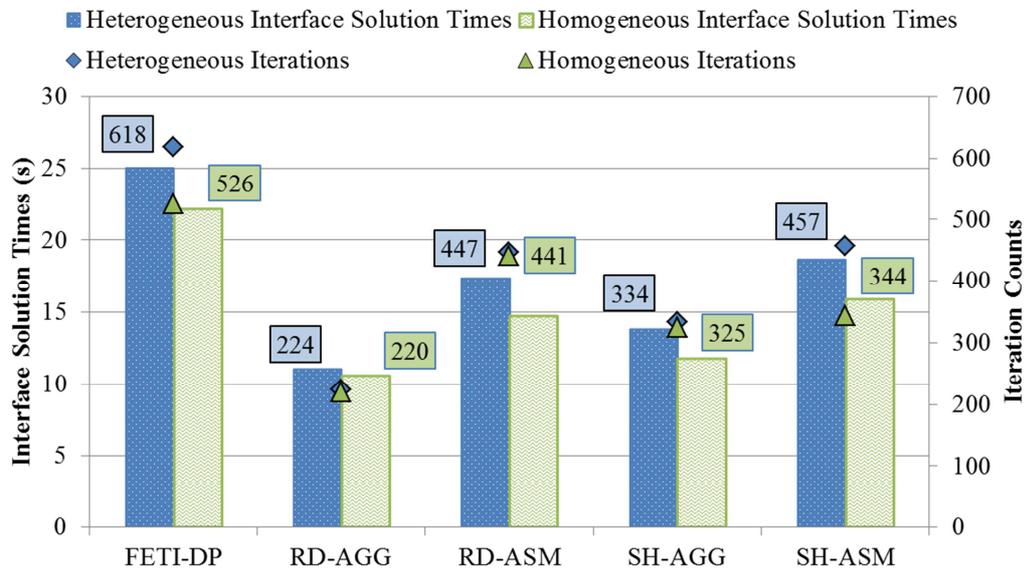


Figure 5.5. Interface solution times and iteration counts for both partitioning approaches

In order to investigate the effect of different partitioning approaches on solution performance in detail, Figure 5.4, Figure 5.5 and Figure 5.6 need to be considered together. As it can be seen from Figure 5.5, iteration counts for the standard FETI-DP is decreased when the homogeneous partitioning is utilized due to the uniform coefficient jumps along the interface. Although the number of dual variables is larger for this case, due to the considerable decrease in iteration counts interface solution timing is also decreased.

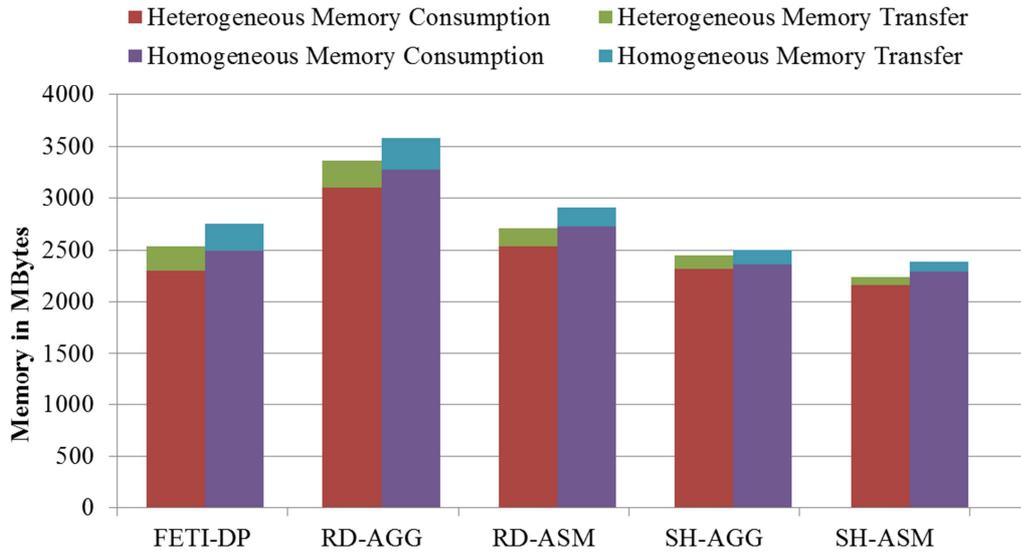


Figure 5.6. The total amount of data transfer and memory consumption for both partitioning approaches

When the RD-AGG deflation vectors are utilized, the size of the dual interface problems increases, but the deflation vector counts are decreased, the interface solution timing decreased slightly. However, when the ASM heuristic is utilized, the number of deflation vectors is decreased considerably and the decrease in the interface solution timings become apparent since the assembly of deflation vectors results in a more successful representation when the resultant vectors are obtained from subdomains that have similar material properties. Percentage of the amount of data transfer to memory consumption for RD variants is lower than others because of the minimum dual interface problem formed by these methods. However, these methods require more memory because the number of deflation vectors generated with these methods is greater than the ones generated with other methods.

SH variant methods generate vectors from only the rigid body modes for solid subdomains and pre-selected number of vectors from zero energy modes for fluid subdomains. In case of heterogeneous subdomains, vectors generated by solid elements are assembled together with the corresponding vectors from fluid finite elements for each subdomain. Therefore, the deflation vector counts for

homogeneous partitioning are slightly lower than the heterogeneous case for SH-AGG. When ASM heuristic is utilized, the number of deflation vector count is equal to the number of pre-selected vectors for fluid finite elements. Therefore, the amount of data transfer and memory consumption is not affected from the deflation vector counts but depends on the interface problem size and the number of dual variables. However, the decrease in solution timings is apparent, because of the decrease in deflation vector counts and the number of iterations in homogeneous case.

In the light of the performance results obtained for this specific problem with the mentioned partitioning configurations, it can be concluded that homogeneous partitioning improves the interface solution timings. However, since the interface solution timings are considerably lower than local solution timings for the developed solution method, workload balancing of the subdomains is much more important for the overall solution performance. Homogeneous partitioning requires a special algorithm that first separates elements into groups according to their material properties and connectivity. Then each subdomain should be partitioned into smaller subdomains by considering the workload balancing of all subdomains. Hence, this is a challenging task need to be investigated in detail. Therefore, if such an algorithm that has a lower computational cost than the improvement obtained as a result of the utilization of homogeneous partitioning is developed, homogeneous partitioning can be beneficial.

5.3.2. Effect of Finite Element Formulations

Determination of the finite elements for model discretization is important since the accuracy of the results obtained at the end, the convergence rate of the solution, and solution timings may vary depending on the finite elements preferred for the model discretization. Water tank problem presented in Section 4.4.1 is modelled with three different finite element formulations, i.e. Q4, Q9, and H8, implemented in this framework. The static analysis performed with a constant uniform load on the

exterior of wall on the right by using 8 processors with standard FETI-DP method. The aspect ratio of the subdomains (H/h) is fixed to 16 for all cases. The properties of the models and the parameters related to convergence characteristics of the solution, and solution times of different steps of the overall analysis are presented in Table 5.2.

Table 5.2. 2D vs 3D finite element formulations

Element Type	Q4	Q9	H8
Total DOF Count	22018	87042	66054
Interface DOF Count	758	1552	2313
Condition Number	1.92E+04	2.08E+04	9.62E+06
Number of Iterations	80	90	347
Local Solution (s)	1.13	39.33	89.86
Interface Solution (s)	0.31	1.25	5.68
Total Solution (s)	1.45	40.58	95.54
Overall Analysis (s)	3.09	106.16	178.16

First, since the aspect ratio of the problem (H/h) is fixed to 16 for all cases, the number of elements in each model is the same. However, in practice, the number of elements that is required for the model discretization depends on the mechanical accuracy of the element. In other words, more elements are required when linear finite elements are utilized to achieve a similar accuracy in the solution when compared the quadratic finite elements. In this case, since the number of elements is the same for all models, total number of DOFs depends only on the number of DOFs for each finite element. For example, using Q9 elements instead of using Q4 elements multiplies the number of equations by approximately 4, whereas using H8 elements multiplies the number of equations by exactly 3 since totally a single layer of H8 elements are utilized for the discretization in the third dimension. On the other hand, using Q9 elements instead of using Q4 elements multiplies the number of interface DOFs by approximately 2. Therefore, it can be concluded that higher order elements results in a comparatively small interface to total DOF count ratio. In case of discretization with H8 elements, interface to total DOF count ratio also depends on the number of elements utilized to model the third dimension. In this case, single layer of H8 elements are used to model the thickness of the water tank, thus, number

of interface DOFs with H8 elements are exactly 3 times of the ones with Q4 elements. Addition of the third dimension to the discretization also directly affects the complexity of the solution due to the additional coupling with the third dimension. Although condition number estimation is approximate, it reveals the increasing complexity as 3D discretization is utilized instead of 2D discretization. Condition number estimation of Q9 element solutions are slightly greater than that of Q4 element solutions, in contrast H8 element solutions has condition numbers approximately 500 times greater than the 2D problems. In other words, this increase in the condition number of system solution results in more than 4 times of iteration counts and about 50 times of solution times.

Another important conclusion from these comparisons is that the ratio of number of interface DOFs to total number of DOFs at the solution with Q9 elements is approximately half of that ratio for the solution with Q4 elements. As it can be seen from Table 5.2, percentage of time required for the interface solution in the total solution time is reduced from 21% to 3% when Q9 elements are utilized. As it is demonstrated in Figure 5.2, local solution is an explicitly parallel computation in which there is no data transfer is required among the processors and the workload for each local solution is balanced by the partitioning algorithm. For example, unbalancing among the subdomains—ratio of the maximum number of elements to the minimum number of elements—for the problem modelled with Q9 elements is 1.014 whereas it is equal to 1.018 for both problems modelled with Q4 and H8 elements. In contrast, interface problem solution requires data transfer among the processors which is a degrading effect on the scalability of the solution because of the latencies for the data communication. Therefore, it can be concluded that the utilization of higher order elements increases the percentage of explicitly parallel computations in the overall solution.

In conclusion, when deciding about the discretization element for the problem within the scope of this section's considerations, following remarks should be considered:

- 3D discretization has considerably larger condition number than the 2D discretization and that effects the number of iterations required to converge to the solution, proportionally.
- Discretization with higher order elements results in significantly larger problem sizes and this will directly affect the solution timing.
- Discretization with higher order elements results in a comparatively small ratio of interface problems size to total problem size; hence, the percentage of explicitly parallel computations in the overall solution is increased.

5.3.3. Effect of Utilizing Different Processor Counts

In this section, static analysis of water tank with flexible walls problem (presented in Section 4.4.1) with a fixed aspect ratio ($H/h = 16$) is analyzed with different number of processors ranging from 2 to 48. METIS partitioning library v5.1.0 is utilized with default parameters for the partitioning of each model. In these series of tests, efficiency and the scalability of solutions with different finite element formulations and different deflation methods (Section 4.3) are investigated.

Total solution times considering both local and interface solution for the problem discretized with Q9 elements are presented in Table 5.3. For almost all cases, iterations converge to the solution successfully, however, for the standard FETI-DP method and solution by SD-0 method with 48 processors, iterations did not converge to the solution until 2,000 iterations. This shows that the interface problem was not conditioned sufficiently in these cases. Although AGG versions of RD and SH converge to the solution successfully when 48 processors utilized, the time required for the solution is higher than the solutions by less number of processors. This is because as the number of subdomains increase, the number of deflation vectors increase drastically which cause a significant increase in computations required for deflation method. For example, total number of deflation vectors generated by RD-AGG algorithm increases from 28 to 844 as the number of processors increases from

4 to 48. However, ASM variants of RD and SH decrease the solution times continuously as the processor counts increase for all cases because these algorithms are immune to the degradation caused by the increasing number of deflation vector numbers. For example, by performing ASM heuristic algorithm on 844 deflation vectors generated by RD-AGG algorithm, the projection space is reduced to 18 deflation vectors. In the light of these results, it can be concluded that the number of deflation vectors utilized and their capability to condition the problem significantly affect not only the efficiency of the solution but also the convergence of the solution.

Table 5.3. Solution times (s) of models discretized with Q9 elements

Processor Count	4	8	16	24	36	48
FETI-DP	138.1	40.9	14.3	6.5	3.5	Not Conv.
QR	133.1	43.5	17.0	8.1	5.1	5.1
SD-0	131.0	36.9	12.0	5.5	2.6	Not Conv.
RD-AGG	141.6	40.2	15.6	11.3	14.5	22.9
RD-ASM	138.6	40.8	13.3	6.5	5.5	5.4
SH-AGG	131.8	40.5	12.6	7.8	6.3	7.5
SH-ASM	132.4	39.5	12.4	6.3	4.4	4.0

Furthermore, when the increase in processor counts and the decrease in solution times are considered, it can be seen that solution algorithm achieved a super-linear speed-up. For instance, solution time decreased from 131.8 seconds to 12.6 seconds when the number of processors utilized in SH-AGG solution increased from 4 to 16. Within these timing, 131.0 seconds and 10.8 seconds are required for local solution with 4 and 16 processors, respectively. Interface problems having 846 and 2,392 equations are solved in 0.8 seconds and 1.8 seconds, for 4 and 16 processors solutions, respectively. Thus, the main cause of this behavior is the improvement in local problem solution in which a direct solution algorithm is utilized. Therefore, such an improvement is expected when the decrease in bandwidth of the local solution as the number of processors increases is considered. Moreover, PETSc utilizes asynchronous message passing routines so that data transfer does not need to block the computations. In other words, the communications and the computations are overlapped.

In case of solutions for the problem discretized with H8 elements, condition number increases 500 times and the resulting effect of solution time increase can be seen from Table 5.4. For example, when the problem is discretized with H8 elements instead of Q9 elements, number of iterations required for the SH-AGG solution with 16 processors increases from 179 to 802 and the solution time increases from 12.6 seconds to 33.2 seconds. In these series of tests, solutions with standard FETI-DP and SD-0 diverges in most of the cases due to the loss of orthogonality originated from increased condition number of the problem with H8 elements. The solutions by 36 processors and excluding solutions by RD-AGG, for the rest of the deflation methods, solution time enhances as the processor counts increase. Solutions by 36 processors suffer from the degradation resulting from the unbalanced workload of subdomains for this case. For example, although the number of elements is balanced for the subdomains, the ratio of maximum to minimum operation counts for this solution is recorded as 3.8 whereas this ratio is changing from 1.2 to 1.8 for the rest of the cases. The local solution is degraded for this case and this verifies the insufficient representation of subdomain workload only by the number of elements contained. Similar to the case with Q9 elements, in RD-AGG case, total number of deflation vectors in solution with 24 processors increase from 576 to 864 when 36 processors are utilized. Therefore, projection of problem system to the deflated system increases accordingly. Except from the solution by 36 processors, solutions with QR method and SH variants improves as the number of processors increase and SH-ASM delivers the solution with the highest speed when the solution is carried out by 48 processors.

Table 5.4. Solution times (s) of models discretized with H8 elements

Processor Count	4	8	16	24	36	48
FETI-DP	Not Conv.	95.5	42.2	Not Conv.	Not Conv.	Not Conv.
QR	285.1	109.8	43.8	24.3	26.1	12.1
SD-0	Not Conv.	86.3	37.3	Not Conv.	Not Conv.	Not Conv.
RD-AGG	304.4	96.1	41.2	27.2	38.3	52.1
RD-ASM	291.4	85.8	35.8	15.9	27.3	18.1
SH-AGG	279.1	84.3	33.2	19.0	21.6	10.3
SH-ASM	301.6	92.8	38.3	18.5	21.0	7.4

In order to investigate the efficiency of utilizing deflation methods over standard FETI-DP and the scalability of the parallel solution algorithm, timings of solutions with different deflation vector generation methods are scaled with respect to the timings of successive solutions with standard FETI-DP method or with QR deflation method. These timings are obtained by utilizing processor counts ranging from 2 to 48. Since standard FETI-DP algorithm converges for only two cases for the problems discretized with H8 elements (Table 5.4), these timings are scaled with respect to timings of solutions with QR deflation method (demonstrated in Figure 5.7) whereas solution timings of the problems discretized with Q9 elements are scaled with respect to the timings of standard FETI-DP solution method (demonstrated in Figure 5.8).

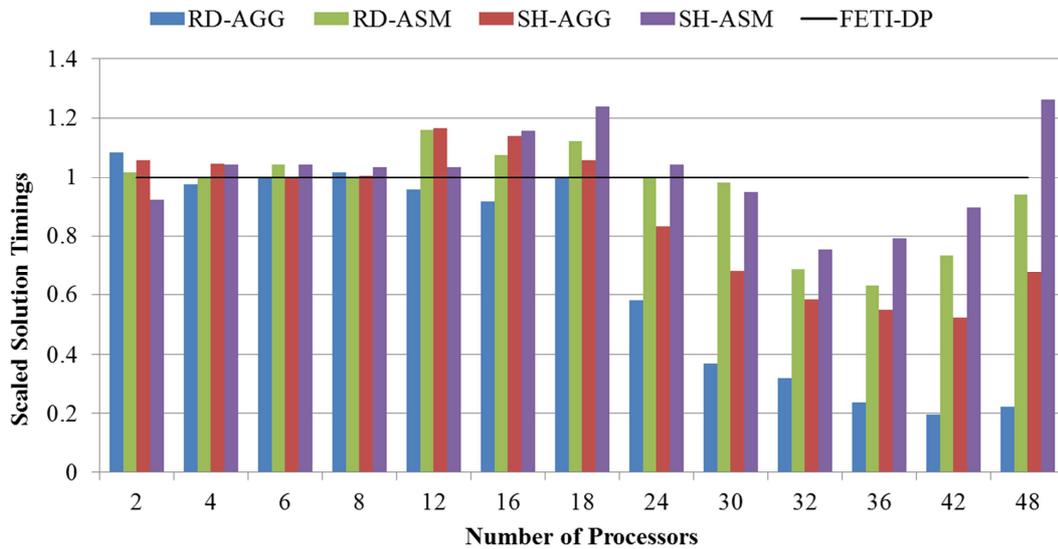


Figure 5.7. Scaled solution timings for Q9 elements discretization

As it is presented in Table 5.3, for this specific type of problem considered in this section, solution method with SD-0 deflation enhances the solution time of standard FETI-DP for all cases if it converges. However, for the solution by 48 processors, solution method with SD-0 deflation and the standard FETI-DP method are both did not converge to a solution. Although the solution method with QR deflation converged to a solution for all cases, it exhibits slower performance than the standard FETI-DP methods due to its requirement of the factorization of large subdomain level stiffness matrices. Scaled timings for the solutions with RD and SH variants are

presented in Figure 5.7. Similarly, solution with RD-AGG also exhibits slower performance than the standard FETI-DP methods for most of the cases. This degrading performance becomes more apparent when the number of processors increases since the total number of deflation vectors utilized increase drastically with the number of processor counts. In case of solutions with RD-ASM deflation, this degrading performance of RD-AGG is enhanced with the utilization of ASM heuristics to decrease the total number of deflation vectors utilized.

In case of solutions with SH variant deflation vectors, solution times are faster than the standard FETI-DP solutions up to 24 processors in almost all cases. However, the solutions when more than 24 processors are utilized, performance improvement by SH deflation vector generation methods vanish. This is due to the increasing deflation vector counts as the number of processors increase for SH-AGG method or the insufficient conditioning with the generated deflation vectors for SH-ASM method. Similar to RD variants, ASM heuristics enhances the timings of solutions with SH-AGG deflation vectors.

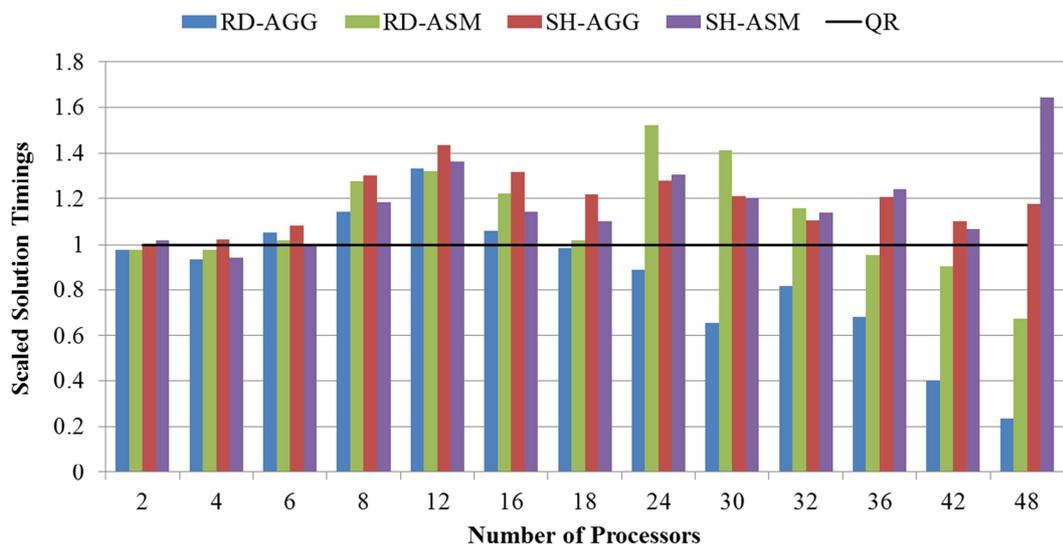


Figure 5.8. Scaled solution timings for H8 elements discretization

Same test cases are conducted by utilizing H8 elements for discretization. As it is mentioned before, the standard FETI-DP and the solution with SD-0 deflation

vectors did not converged to a solution for most of the cases discretized with H8 elements (Table 5.4). Therefore, in Figure 5.8, solution times are scaled with respect to the timings of solutions with QR deflation vectors. Although, solutions with RD variant vectors enhances the required times for the solutions by less than 24 processors, these improvements vanish as the total number of deflation vectors increase with the processors counts increase. For almost all cases, solutions with SH variant deflation vectors improve the solution times with respect to the timings of solutions with QR deflation vectors.

When the solution performances for the problems with H8 elements are compared with the performances for the Q9 elements, improvements obtained by the utilization of deflation vectors are more obvious. Since the condition number of the problem increases when H8 elements are utilized, conditioning of the problem is more important in this case. Especially with the utilization of SH variant deflation vectors, solution is significantly improved as it can be seen from Figure 5.8.

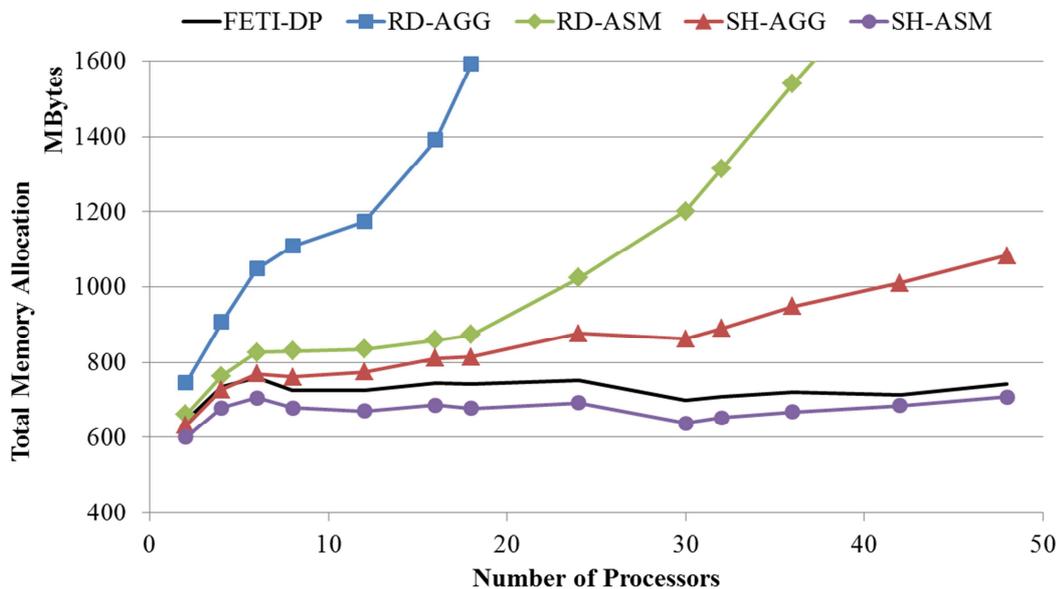


Figure 5.9. Total memory allocated for the solutions with Q9 elements

Another important parameter about the performance of a parallel solution algorithm is the memory required for the execution. Total memory allocated on all processors

for the solution of test model discretized with Q9 and H8 elements by changing processor counts are presented in Figure 5.9 and Figure 5.10, respectively.

For both cases, although the iterations converge to the solution, RD-ASM and RD-AGG requires considerable amount of memory allocations due to increasing number of deflation vectors. For instance, total number of deflation vectors utilized by RD-AGG for 16 processors case is approximately 4 times higher than that for 4 processors because, as the number of subdomains increase, the number of deflation vectors generated increase. Although solutions with SH-AGG deflation vectors required less amount of memory allocations when compared to RD variant methods, its memory requirements are higher than the standard FETI-DP method and this requirement is gradually increasing as the number of processors increase. Within these methods, SH-ASM requires the minimum sizes of memory for almost all cases and that verifies the success of ASM heuristic in order to limit the memory requirements by limiting the generated deflation vector counts.

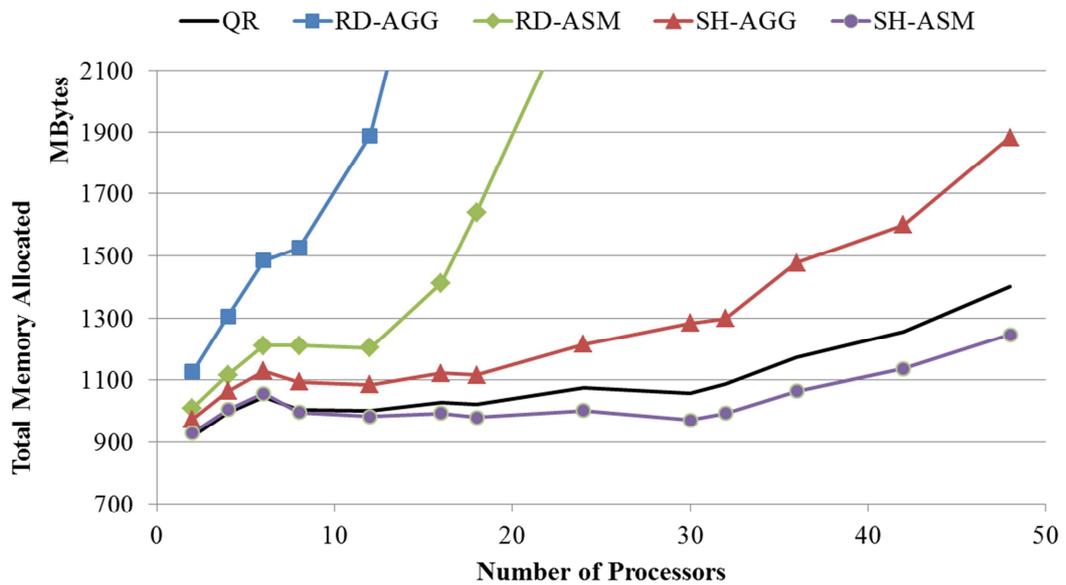


Figure 5.10. Memory allocated for solutions with H8 elements

Another performance measurement for a parallel solution algorithm is the amount of data transfer during the solution. PETSc utilizes asynchronous message passing routines so that data transfer does not need to block the computations. In other words, the communications and the computations are overlapped; therefore, time spent for the communication cannot be fully tracked. In Figure 5.11, the amount of data transfer during the system solution is presented. Linear interpolations that show the amount of data transfer with respect to processor counts are calculated for each solution method. Obviously, the amount of data transfer through the iterative solution is mainly governed by the number of iterations and number of deflation vectors utilized for the solution. Similar data transfer trends are expected for Standard FETI-DP, RD-ASM and SH-ASM since the utilization deflation vectors requires more memory but in return decreases the iteration counts. However, solutions with RD-AGG and SH-AGG vectors requires the most amount of data transfer since the number of deflation vectors for these methods are considerably larger than the other ones. For example, the total number of deflation vectors generated by the RD-AGG method is 96 whereas the SH-ASM method generated 10 deflation vectors for the solution of the problem discretized with H8 elements with four processors.

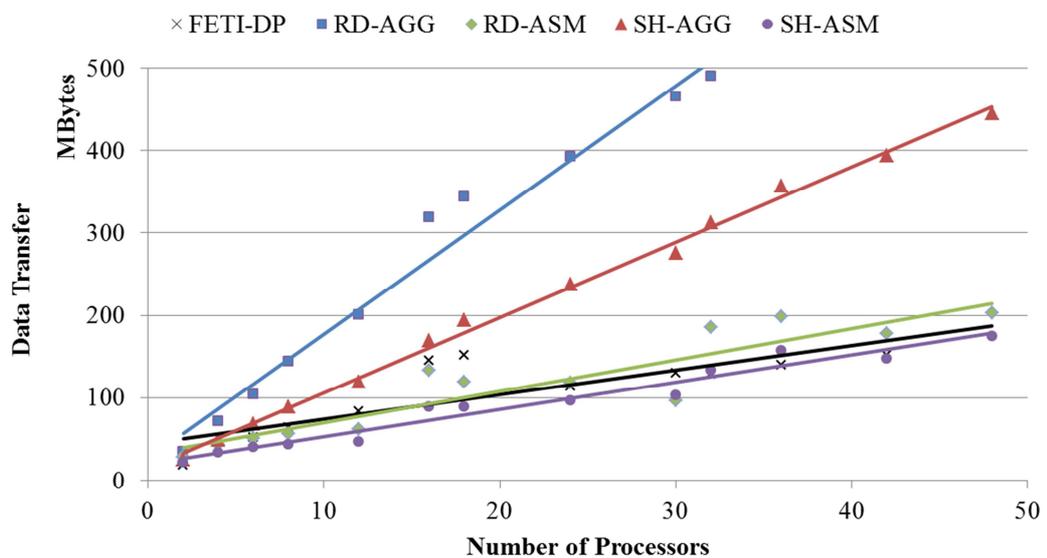


Figure 5.11. The amount of data transfer for the solutions with Q9 elements

According to the results presented in this section, the following remarks can be listed:

- Standard FETI-DP method and solutions with SD-0 deflation vectors did not converge to a solution for most of the cases, especially in 3D discretization.
- In case of solutions with clusters having less than 8 processors, the solution method with QR deflation vectors requires the minimum amount of data transfer and memory consumption. Otherwise, the solutions with SH-ASM deflation vectors require the minimum amount of data transfer and memory consumption.
- In all test cases, QR deflation vectors did not enhance the solution time when compared with the standard FETI-DP method due to its requirement of the factorization of large subdomain level matrices.
- Enhancement in solution times obtained by RD variant vectors degrades with the increasing number of processors because the total number of deflation vectors also increase accordingly. Because of this reason, for cases where the processor count is larger than 16, the solution time was not reduced.
- When Q9 elements are utilized, deflation vectors generated by SH method improve the solution times up to 24 processors. When the number processors utilized exceeds 24, SH variants did not improve the solution time with respect to standard FETI-DP. However, when H8 elements are used, deflation vectors generated by SH method improve the solution times for almost all cases and performs faster than the standard FETI-DP method.

5.3.4. Utilization of Krylov Subspaces

In order to investigate the effect of Krylov subspace utilization during dynamic analysis, water tank model is discretized with three different elements, i.e. Q4, Q9 and H8 elements, analyzed for 2 seconds with 0.001 second time steps. A step force with finite rise time function ($t_r = 0.1$ s) is applied as shown in Figure 5.12. The problem is analyzed by the solution method with SH-ASM deflation vectors and

iteration counts throughout the dynamic analysis for the three models are presented in Figure 5.12.

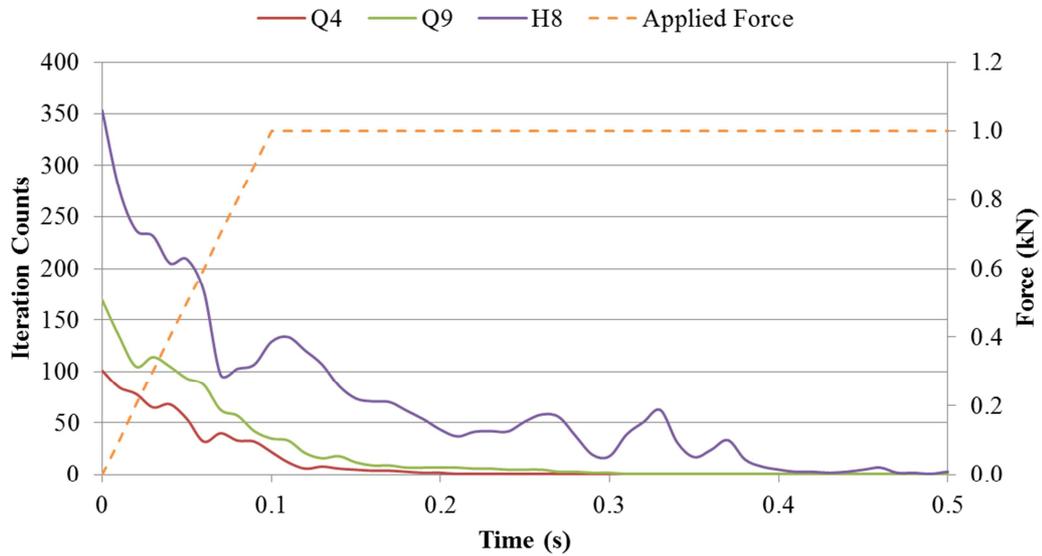


Figure 5.12. Iteration counts for the first 0.5 seconds of the dynamic analysis with step force

When Krylov subspace vectors are utilized, the number of iterations required at each time step decreased as the number of time steps increased. This is actually an expected result since the applied loading is simple and unidirectional step force. The number of iterations reduced up to 1 iteration per each time-step for the steady state response. Such stabilization is observed earlier for the solutions of models discretized with Q4 and Q9 elements. However, this behavior is postponed for dynamic analysis of models with H8 elements due to the increased condition number for these cases. As the time step exceeds 0.212 seconds, 0.307 seconds and 0.514 seconds for the models discretized with Q4, Q9 and H8 elements, respectively, a single iteration is enough for each time step to obtain the response at that step.

For investigating the performance change in the solution for seismic loading, same problem is analyzed with the same solution method and same dynamic parameters but this time El-Centro NS ground acceleration data is utilized instead of a step force.

Accelerations are applied as horizontal forces on each node proportional to the lumped mass value at that node. In order to compare the performance with the problem with step force and also zoom to the seismic data, dynamic analysis is carried out for only first 2 seconds of the earthquake. Ground acceleration data and the iteration counts required for the solution throughout the time history is presented in Figure 5.13. Iteration counts for the solutions of all models are oscillating and this is expected since the applied force at each time step is generally different than the previous time step. However, iteration counts are slightly decreasing when the ground acceleration is gradually increasing or decreasing. For example, when the iteration counts after 1.8 seconds are considered, obviously a decreasing trend in iteration counts for all solutions is apparent. Similar to the solution with step force case, iteration counts are decreased much for the solution of model with Q4 elements and decreased least for the solution of model with H8 elements.

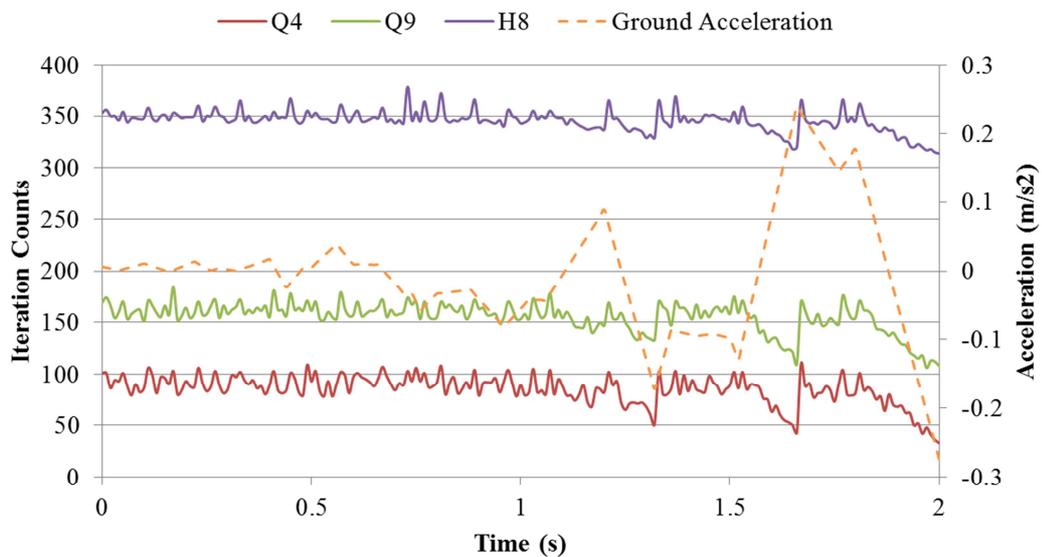


Figure 5.13. Iteration counts for the first 2 seconds of seismic analysis

During these tests, first 20 Krylov subspace vectors are stored and they are utilized to compute a better initial guess for the iterations of the following time step. The performance improvement obtained with the utilization of Krylov subspace vectors highly depends on the count of subspace vectors and the way they are selected. To

illustrate, for this number of Krylov subspace vectors, the way they are selected and the problem investigated in this section, the utilization of Krylov subspace vectors causes 95.8%, 96.0%, and 94.6% reduction in total iteration counts for the solutions of models with Q4, Q9 and H8 elements, respectively, when the step force is applied. However, when the earthquake excitation is applied to the problems, the reduction in total iteration counts are 21%, 10%, and 2%. As a result, Krylov subspaces can be stored for the improvements of solution iterations for the following time steps; but the improvement obtained is degraded considerably when the applied force is changing rapidly as it is in earthquake loading.

5.3.5. Solution Performance of a Large Actual Dam Model

In order to investigate the ultimate performance of the parallel solution algorithm on a computer cluster, proposed SH-ASM deflation vector generation method is tested with the dynamic analysis of an actual dam model presented in Figure 4.12, with a much finer mesh. Problem domain is partitioned by using METIS partitioning library with default parameters into heterogeneous subdomains and the dynamic analysis is carried out for 5 seconds with 0.01 second time steps with the utilization of Krylov subspaces. Table 5.5 contains the details of each mathematical model and the time required to obtain the results.

Table 5.5. Performance of SH-ASM on actual cases

Element Type	Q4			Q9	H8
H/h	128	128	256	128	128
Processor Count	24	48	24	24	24
Total # of DOFs	1,380,354	1,380,354	11,023,356	5,513,218	4,141,062
# of Interface DOFs	13,390	20,540	27,472	26,782	40,198
Element Count	688,128	688,128	2,752,512	688,128	688,128
Solution Times (s)	12 min.	7 min.	3 hr.	19 hr.	17 hr.
Total Times (s)	1 hr.	30 min.	16 hr.	28 hr.	35 hr.

Three types of finite elements, Q4, Q9 and H8 elements are utilized as different cases. Corresponding mathematical models for H8 elements is obtained by defining a

single layer of H8 elements on both sides of the 2D models utilized. Solution times given in Table 5.5 involves only the time required for the time stepping algorithm, however, total times additionally contains any other pre-processing or post-processing required for the analysis.

When Q4 elements are utilized with H/h ratio equal to 128, the model has about 1.4 million DOFs and the time integration for dynamic analysis of this model was completed in roughly 12 minutes and overall solution with all pre-processing and post-processing computations required about one hour with 24 processors. These timings are roughly halved when the processor count is doubled. When H/h is doubled, element count and total number of DOFs reaches 2.7 million and 11 million, respectively. Approximately 3 hours is required for the repetitive solution of the system and 16 hours for overall analysis by utilizing 24 processors. When Q9 and H8 elements are utilized for discretization, the number of total DOFs increased to 4 times and the number of interface DOFs increased to 2 and 3 times, respectively. This jump and increased complexity results in a drastic increase in the time required for the time stepping iterations; 19 and 17 hours are required for the time stepping algorithm and 28 hours and 35 hours required for the overall analysis, respectively.

In conclusion, implemented solution framework with proposed deflation vector generation methods sufficiently converged to a solution for the dynamic analysis of a large actual dam-reservoir system in an acceptable time frame. Mechanical validations of these solutions are presented in Section B.2.3.

CHAPTER 6

CONCLUSIONS

6.1. Summary

In this thesis, a parallel iterative solution method which can be utilized for the dynamic analysis of fully Lagrangian solution of the dam-reservoir systems is proposed. The iterative solution is mainly based on FETI solvers to have a scalable solution framework. The problem itself is highly ill-conditioned due to having domains with different materials and governing equations. Having an ill-conditioned system disturbs the convergence rate of an iterative solver. Thus, deflation method is utilized with different deflation vector generation algorithms to improve the convergence rate. In addition to that, a novel semi-heuristic deflation vector generation method is proposed. The method is based on generating deflation vectors utilizing the pre-selected zero energy modes of the fluid finite elements that are used to model the reservoir. Furthermore, proposed heuristic method is extended to reduce the memory requirements significantly without disturbing the convergence rate. Performance of utilizing existing and proposed deflation methods in terms of iteration counts, memory requirements, and operation counts are discussed and compared by analyzing water tank with flexible wall problem. The effect of having different finite element formulations and material properties are also examined. As a final investigation, an actual dam-reservoir model is analyzed.

Developed solution method is extended to utilize computer cluster for all analysis steps by utilizing C++ programming language and PETSc library. The implemented program is added to the general finite element framework Panthalassa. The efficiency and the scalability of the parallel solution framework are discussed for

different types of finite elements, for different partitioning approaches, and for different number of processors. Finally, the solution performance is tested for a large scale actual dam model having an order of million elements.

6.2. Concluding Remarks

Conclusions that are extracted from the studies performed throughout the thesis are summarized as in the following:

- A priori known zero energy modes of a finite element formulation can be utilized to generate a deflation space. Utilization of deflation method with the resulting projection space reduces the condition number of the problems considered in this study and this enhances the iteration counts and time required for the solutions of such problems. Besides, additional memory requirements of the generation methods can be limited by utilizing an assembly heuristic.
- Although solution methods without deflation and solution methods with subdomain deflation vectors perform better than others for some test cases, iterations did not converge to a solution for all cases, especially when 3D finite elements are utilized. On the other hand, solution methods with recursively generated deflation vectors and QR deflation vectors converge to a solution for all cases, but, generally perform worse than the other solution methods. Among the examined deflation vector generation methods, proposed deflation vector generation algorithms mostly perform the best and always converge to a solution for the problems considered in this study.
- Developed solution method with deflation vectors generated by proposed methods successfully converged to a solution for the dynamic analysis of a large actual dam-reservoir system in an acceptable time frame; within days not months.

6.3. Recommendations for Future Study

The developed solution method is tested for static and dynamic analysis of dam-reservoir systems discretized with Lagrangian quadrilateral and hexahedron finite elements. Authentication of the developed solution method is needed to be made for different types of Lagrangian finite elements. Even several other analyses like non-linear analysis and dam failure analysis can be investigated.

Since the developed solution framework involves several interchangeable sub-algorithms, it provides a beneficial infrastructure for investigating alternatives. For example, different solvers can be investigated for the solution of local problems or interface problems. Besides, plentiful convergence improvement methods for iterative solutions exist in literature. More detailed studies might be conducted to obtain workload balanced homogeneous partitioning and optimized selection of Krylov subspace vectors.

Last but not the least, developed solution method is investigated for dam-reservoir interaction problems, however it might be applied on other types of acoustic fluid-structure interaction problems.

REFERENCES

- [1] P. on E.E. for C. Dams, C. on E. Engineering, D. of N.H. Mitigation, C. on E. and T. Systems, D. on E. and P. Sciences, N.R. Council, "Earthquake Engineering for Concrete Dams: Design, Performance, and Research Needs", National Academies Press, 1991.
- [2] K.-L. Fok, "Earthquake analysis and response of concrete arch dams.", (1986).
- [3] L. Cheng, R.D. White, K. Gosh, "Three Dimensional Viscous Finite Element Formulation For Acoustic Fluid Structure Interaction", *Comput. Methods Appl. Mech. Eng.* 197 (2008) 4160–4172.
- [4] Y. Ghanaat, "Theoretical manual for analysis of arch dams", DTIC Document, 1993.
- [5] Y. Bazilevs, V.M. Calo, T.J.R. Hughes, Y. Zhang, "Isogeometric fluid-structure interaction: theory, algorithms, and computations", *Comput. Mech.* 43 (2008) 3–37.
- [6] Y. Bazilevs, V.M. Calo, Y. Zhang, T.J. Hughes, "Isogeometric fluid–structure interaction analysis with applications to arterial blood flow", *Comput. Mech.* 38 (2006) 310–322.
- [7] B. Hübner, E. Walhorn, D. Dinkler, "A monolithic approach to fluid–structure interaction using space–time finite elements", *Comput. Methods Appl. Mech. Eng.* 193 (2004) 2087–2104.
- [8] M. Heil, "An efficient solver for the fully coupled solution of large-displacement fluid–structure interaction problems", *Comput. Methods Appl. Mech. Eng.* 193 (2004) 1–23.
- [9] C. Michler, S.J. Hulshoff, E.H. van Brummelen, R. de Borst, "A monolithic approach to fluid–structure interaction", *Comput. Fluids.* 33 (2004) 839–848.
- [10] A.T. Barker, "Parallel monolithic fluid-structure interaction algorithms with application to blood flow simulation", University of Colorado at Boulder, 2009.
- [11] T. Dunne, R. Rannacher, T. Richter, "Numerical simulation of fluid-structure interaction based on monolithic variational formulations", *Fundam. Trends Fluid–structure Interact. Contemp Chall Math Fluid Dyn Appl.* 1 (2010) 1–75.

- [12] J. Hron, S. Turek, "A monolithic FEM solver for an ALE formulation of fluid-structure interaction with configuration for numerical benchmarking", in: ECCOMAS CFD 2006 Proc. Eur. Conf. Comput. Fluid Dyn. Egmond Aan Zee Neth. Sept. 5-8 2006, Delft University of Technology; European Community on Computational Methods in Applied Sciences (ECCOMAS), 2006.
- [13] S. Piperno, C. Farhat, B. Larrouturou, "Partitioned procedures for the transient solution of coupled aroelastic problems Part I: Model problem, theory and two-dimensional application", *Comput. Methods Appl. Mech. Eng.* 124 (1995) 79–112.
- [14] C.A. Felippa, K.C. Park, "Partitioned simulation of coupled system: I. tools for methods design", *Comput. Asp. Struct. Acoust. Vib. CISM-Cent. Int. Sci. Mécaniques.* (2006).
- [15] K. Park, C. Felippa, J. DeRuntz, "Stabilization of staggered solution procedures for fluid-structure interaction analysis", *Comput. Methods Fluid-Struct. Interact. Probl.* (1977) 95–124.
- [16] C.A. Felippa, K.C. Park, "Staggered transient analysis procedures for coupled mechanical systems: Formulation", *Comput. Methods Appl. Mech. Eng.* 24 (1980) 61–111.
- [17] C.A. Felippa, K.C. Park, C. Farhat, "Partitioned analysis of coupled mechanical systems", *Comput. Methods Appl. Mech. Eng.* 190 (2001) 3247–3270.
- [18] O.C. Zienkiewicz, R.L. Taylor, "Finite Element Method: Volume I", 5th edition, Butterworth-Heinemann, Oxford ; Boston, 2000.
- [19] R.B. Jansen, ed., "Advanced Dam Engineering for Design, Construction, and Rehabilitation", Springer US, Boston, MA, 1988.
- [20] B. Engquist, A. Majda, "Absorbing Boundary Conditions for the Numerical Simulation of Waves", *Math Comp.* 31 (1977) 629–651.
- [21] C. Farhat, F.-X. Roux, "A method of finite element tearing and interconnecting and its parallel solution algorithm", *Int. J. Numer. Methods Eng.* 32 (1991) 1205–1227.
- [22] C. Farhat, P.-S. Chen, J. Mandel, F.X. Roux, "The two-level FETI method Part II: Extension to shell problems, parallel implementation and performance results", *Comput. Methods Appl. Mech. Eng.* 155 (1998) 153–179.
- [23] C. Farhat, M. Lesoinne, P. Le Tallec, K. Pierson, D. Rixen, "FETI-DP: A Dual-Primal Unified FETI Method – Part I: A Faster Alternative to the Two-Level FETI Method.", *Int J Numer Meth Engrg.* 50 (2001) 1523–1544.

- [24] C. Farhat, J. Mandel, "The two-level FETI method for static and dynamic plate problems Part I: An optimal iterative solver for biharmonic systems", *Comput. Methods Appl. Mech. Eng.* 155 (1998) 129–151.
- [25] M. Bhardwaj, D. Day, C. Farhat, M. Lesoinne, K. Pierson, D. Rixen, "Application of the FETI method to ASCII problems—scalability results on 1000 processors and discussion of highly heterogeneous problems", *Int. J. Numer. Methods Eng.* 47 (2000) 513–535.
- [26] A. Klawonn, O.B. Widlund, "FETI and Neumann–Neumann Iterative Substructuring Methods: Connections and New Results", *Comm Pure Appl Math.* 54 (2001) 57–90.
- [27] A. Klawonn, O.B. Widlund, "FETI-DP Methods for Three-Dimensional Elliptic Problems with Heterogeneous Coefficients", Courant Institute of Mathematical Sciences, 2000.
- [28] N. Bićčanić, E. Hinton, "Spurious modes in two-dimensional isoparametric elements", *Int. J. Numer. Methods Eng.* 14 (1979) 1545–1557.
- [29] R. Nabben, C. Vuik, "A comparison of abstract versions of deflation, balancing and additive coarse grid correction preconditioners", *Numer. Linear Algebra Appl.* 15 (2008) 355–372.
- [30] O. Coulaud, L. Giraud, P. Ramet, X. Vasseur, "Deflation and augmentation techniques in Krylov subspace methods for the solution of linear systems", *ArXiv13035692 Math.* (2013).
- [31] A. Klawonn, O. Rheinbach, "Deflation, Projector Preconditioning, and Balancing in Iterative Substructuring Methods: Connections and New Results", *SIAM J. Sci. Comput.* 34 (2012).
- [32] J.M. Tang, R. Nabben, C. Vuik, Y.A. Erlangga, "Comparison of Two-Level Preconditioners Derived from Deflation, Domain Decomposition and Multigrid Methods", *J. Sci. Comput.* 39 (2009) 340–370.
- [33] T.B. Jönsthövel, D.U. of T.D. of A.M. Analysis, "Preconditioned Conjugate Gradient Methods Enhanced by Deflation of Rigid Body Modes Applied to Composite Materials", Delft University of Technology, 2009.
- [34] C. Vuik, R. Nabben, J. Tang, "Deflation acceleration for domain decomposition preconditioners", 2006.
- [35] J. Frank, C. Vuik, "On the Construction of Deflation-Based Preconditioners", *SIAM J. Sci. Comput.* 23 (2001) 442–462.

- [36] J.M. Tang, R. Nabben, C. Vuik, Y.A. Erlangga, "Theoretical and numerical comparison of various projection methods derived from deflation, domain decomposition and multigrid methods", Delft University of Technology, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft Institute of Applied Mathematics, 2007.
- [37] T.B. Jonsthovel, M.B. van Gijzen, C. Vuik, A. Scarpas, "On the use of rigid body modes in the deflated preconditioned conjugate gradient method", *SIAM J. Sci. Comput.* 35 (2013) B207–B225.
- [38] M. Papadrakakis, Y. Fragakis, "An integrated geometric–algebraic method for solving semi-definite problems in structural mechanics", *Comput. Methods Appl. Mech. Eng.* 190 (2001) 6513–6532.
- [39] C. Farhat, M. Géradin, "On the general solution by a direct method of a large-scale singular system of linear equations: application to the analysis of floating structures", *Int. J. Numer. Methods Eng.* 41 (1998) 675–696.
- [40] A.K. Chopra, "Dynamics of structures", 4th ed., Prentice Hall New Jersey, 1995.
- [41] C. Farhat, P. CHEN, "Tailoring Domain Decomposition Methods", in: *Domain Decompos. Methods Sci. Eng. Comput. Proc. Seventh Int. Conf. Domain Decompos.* Oct. 27-30 1993 Pa. State Univ., American Mathematical Soc., 1994: p. 401.
- [42] C. Farhat, F.X. Roux, "Implicit parallel processing in structural mechanics.", *Comput Mech Adv.* 2 (1994) 1–124.
- [43] C. Farhat, L. Crivelli, F.X. Roux, "Extending substructure based iterative solvers to multiple load and repeated analyses", *Comput. Methods Appl. Mech. Eng.* 117 (1994) 195–209.
- [44] K.-J. Bathe, "Finite Element Procedures", Klaus-Jurgen Bathe, 2006.
- [45] H.M. Westergaard, "Water Pressures on Dams during Earthquakes", *Trans. Am. Soc. Civ. Eng.* 98 (1933) 418–433.
- [46] J.S.-H. Kuo, "Fluid-Structure interactions: added mass computations for incompressible fluid", University of California, College of Engineering, Earthquake Engineering Research Center, 1982.
- [47] F.E.R. Commission, others, "Engineering guidelines for the evaluation of hydropower projects", Wash. DC. (2002).

- [48] H. Ren, T. Li, L. Zhao, X. Wang, D. Liang, "Study on the Effect of Fluid-Solid Coupling and Foundation Radiation Damping on Seismic Response of Arch Dam", in: *Adv. Water Resour. Hydraul. Eng.*, Springer, 2009: pp. 1756–1761.
- [49] D.P. Mok, W.A. Wall, E. Ramm, "Accelerated iterative substructuring schemes for instationary fluid-structure interaction", *Comput. Fluid Solid Mech.* (2001) 1325–1328.
- [50] F. Nobile, "Numerical approximation of fluid-structure interaction problems with application to haemodynamics", (2001).
- [51] S. Piperno, "Explicit/implicit fluid/structure staggered procedures with a structural predictor and fluid subcycling for 2D inviscid aeroelastic simulations", *Int. J. Numer. Methods Fluids.* 25 (1997) 1207–1226.
- [52] D. Peric, W.G. Dettmer, "A computational strategy for interaction of fluid flow with spatial structures", in: *Proc. 5th Int. Conf. Comput. Shell Spat. Struct. IASS IACM'05*, 2005.
- [53] E. Kuhl, S. Hulshoff, R. De Borst, "An arbitrary Lagrangian Eulerian finite-element approach for uid–structure interaction phenomena", *Int J Numer Meth Engng.* 57 (2003) 117–142.
- [54] P. Causin, J-F. Gerbeau, F. Nobile, "Added-mass effect in the design of partitioned algorithms for fluid–structure problems", *Comput. Methods Appl. Mech. Eng.* 194 (2005) 4506–4527.
- [55] G.C. Everstine, "A symmetric potential formulation for fluid-structure interaction", *J. Sound Vib.* 79 (1981) 157–160.
- [56] L. Kiefling, G.C. Feng, "Fluid-Structure Finite Element Vibrational Analysis", *AIAA J.* 14 (1976) 199–203.
- [57] L.G. Olson, K.-J. Bathe, "A study of displacement-based fluid finite elements for calculating frequencies of fluid and fluid-structure systems", *Nucl. Eng. Des.* 76 (1983) 137–151.
- [58] M.A. Hamdi, Y. Ousset, G. Verchery, "A displacement method for the analysis of vibrations of coupled fluid-structure systems", *Int. J. Numer. Methods Eng.* 13 (1978) 139–150.
- [59] E.L. Wilson, M. Khalvati, "Finite elements for the dynamic analysis of fluid-solid systems", *Int. J. Numer. Methods Eng.* 19 (1983) 1657–1668.
- [60] H.C. Chen, R.L. Taylor, "Vibration analysis of fluid-solid systems using a finite element displacement formulation", *Int. J. Numer. Methods Eng.* 29 (1990) 683–698.

- [61] A. Bermúdez, R. Rodríguez, "Finite element computation of the vibration modes of a fluid—solid system", *Comput. Methods Appl. Mech. Eng.* 119 (1994) 355–370.
- [62] X. Wang, K.-J. Bathe, "Displacement/Pressure Based Mixed Finite Element Formulations for Acoustic fluid–structure Interaction Problems", *Int. J. Numer. Methods Eng.* 40 (1997) 2001–2017.
- [63] F. Hamdan, "Near-field fluid–structure interaction using Lagrangian fluid finite elements", *Comput. Struct.* 71 (1999) 123–141.
- [64] R. Glowinski, others, "First international symposium on domain decomposition methods for partial differential equations", SIAM, 1988.
- [65] P. Le Tallec, "Domain decomposition methods in computational mechanics", in: J.T. Oden (Ed.), *Comput. Mech. Adv.*, North-Holland, 1994: pp. 121–220.
- [66] J.H. Bramble, J.E. Pasciak, A.H. Schatz, "The construction of preconditioners for elliptic problems by substructuring, I", *Math Comp.* 47 (1986) 103–134.
- [67] J. Bey, G. Wittum, "Downwind Numbering: Robust Multigrid for Convection-Diffusion Problems", *Appl Numer Math.* 23 (1997) 177–192.
- [68] M. Pennacchio, V. Simoncini, "Algebraic multigrid preconditioners for the bidomain reaction–diffusion system", *Appl. Numer. Math.* 59 (2009) 3033–3050.
- [69] R.E. Bank, T.F. Dupont, H. Yserentant, "The Hierarchical Basis Multigrid Method", *Numer Math.* 52 (1988) 427 – 458.
- [70] J. Bramble, "Multigrid methods", Longman Scientific, 1993.
- [71] W. Hackbusch, "Multigrid Methods and Applications", Springer, Berlin, 1985.
- [72] L.C. Cowsar, A. Weiser, M.F. Wheeler, "Parallel Multigrid and Domain Decomposition Algorithms for Elliptic Equations", in: D.E. Keyes, T.F. Chan, G.A. Meurant, J.S. Scroggs, R.G. Voigt (Eds.), *Fifth Int. Symp. Domain Decompos. Methods Partial Differ. Equ.*, SIAM, Philadelphia, PA, 1992: pp. 376–385.
- [73] J. Mandel, "Balancing Domain Decomposition", *Comm Numer Meth Engrg.* 9 (1993) 233–241.
- [74] Y. Fragakis, M. Papadrakakis, "The mosaic of high performance domain Decomposition Methods for Structural Mechanics: Formulation, interrelation and numerical efficiency of primal and dual methods", *Comput. Methods Appl. Mech. Eng.* 192 (2003) 3799–3830.

- [75] T.P.A. Mathew, "Schur complement and iterative substructuring algorithms", *Domain Decompos. Methods Numer. Solut. Partial Differ. Equ.* (2008) 107–230.
- [76] C. Farhat, "A Lagrange multiplier based on divide and conquer finite element algorithm", *J Comput Syst. Engrg.* 2 (1991) 149–156.
- [77] C. Farhat, J. Mandel, F.-X. Roux, "Optimal convergence properties of the FETI domain decomposition method", *Comput Methods Appl Mech Engrg.* 115 (1994) 367–388.
- [78] P.L. Tallec, J. Mandel, M. Vidrascu, "A Neumann-Neumann domain decomposition algorithm for solving plate and shell problems.", *SIAM J Numer Math.* 35 (1998) 836–867.
- [79] M. Geradin, D. Coulon, J.-P. Delsemme, "Parallelization of the Samcef Finite Element Software Through Domain Decomposition and Feti Algorithm", *Int. J. High Perform. Comput. Appl.* 11 (1997) 286–298.
- [80] C. Farhat, P.-S. Chen, F. Risler, F.-X. Roux, "A unified framework for accelerating the convergence of iterative substructuring methods with Lagrange multipliers", *Intern. J Numer Methods Engrg.* 42 (1998) 257–288.
- [81] G. Shklarski, S. Toledo, "Computing the null space of finite element problems", *Comput. Methods Appl. Mech. Eng.* 198 (2009) 3084–3095.
- [82] W. Ford, "Numerical Linear Algebra with Applications: Using MATLAB", 1 edition, Academic Press, Waltham, MA, 2014.
- [83] K.P. C. Farhat M. Lesoinne, P. LeTallec, D. Rixen, "FETI-DP: A Dual-Primal Unified FETI Method - Part I: A Faster Alternative to the Two-Level FETI Method", *Int. J. Numer. Methods Eng.* (2000).
- [84] M. Lesoinne, K. Pierson, "FETI-DP: An Efficient, Scalable, and Unified Dual-Primal FETI Method", in: T. Chan, T. Kako, H. Kawarada, O. Pironneau (Eds.), *Domain Decompos. Methods Sci. Eng.*, DDM.org, 1999: pp. 421–428.
- [85] O. Rheinbach, "Parallel Scalable Iterative Substructuring: Robust Exact and Inexact FETI-DP Methods with Applications to Elasticity", Department of Mathematics, University of Duisburg-Essen, 2006.
- [86] Y. Fragakis, M. Papadrakakis, "The mosaic of high-performance domain decomposition methods for structural mechanics—Part II: Formulation enhancements, multiple right-hand sides and implicit dynamics", *Comput. Methods Appl. Mech. Eng.* 193 (2004) 4611–4662.

- [87] J. Mandel, B. Sousedík, "Adaptive selection of face coarse degrees of freedom in the BDDC and the FETI-DP iterative substructuring methods", *Comput. Methods Appl. Mech. Eng.* 196 (2007) 1389–1399.
- [88] J. Mandel, B. Sousedík, "Coarse spaces over the ages", in: *Domain Decompos. Methods Sci. Eng.* XIX, Springer, 2011: pp. 213–220.
- [89] M. Lesoinne, "19. a feti-dp corner selection algorithm for three-dimensional problems", *Domain Decompos. Methods Sci. Eng.* (2003) 217.
- [90] Y. Saad, "Iterative Methods for Sparse Linear Systems", PWS Publishing Company, 1996.
- [91] O. Axelsson, "Iterative Solution Methods", Cambridge University Press, New York, 1994.
- [92] D.J. Rixen, "Extended preconditioners for the FETI method applied to constrained problems", *Int. J. Numer. Methods Eng.* 54 (2002) 1–26.
- [93] L.M. Carvalho, L. Giraud, P.L. Tallec, "Algebraic two-level preconditioners for the Schur complement method", *SIAM J Sci. Comput.* 22 (2001) 1987–2005.
- [94] T.F. Chan, T.P. Mathew, "Domain Decomposition Preconditioners for Convection Diffusion Problems", in: A. Quarteroni, J. Périaux, Y.A. Kuznetsov, O. Widlund (Eds.), *Domain Decompos. Methods Sci. Eng.*, AMS, Providence, RI, 1994: pp. 157–175.
- [95] L.M. Carvalho, L. Giraud, L.M. Carvalho, L. Giraud, "Block diagonal preconditioners for the Schur complement method", in: *Innov. Comput. Methods Struct. Mech.*, 1999: pp. 61–82.
- [96] C. Farhat, L. Crivelli, M. Géradin, "Implicit time integration of a class of constrained hybrid formulations—Part I: Spectral stability theory", *Comput. Methods Appl. Mech. Eng.* 125 (1995) 71–107.
- [97] C. Farhat, P.-S. Chen, J. Mandel, "A Scalable Lagrange Multiplier Based Domain Decomposition Method for Time-Dependent Problems", *Intern. J Numer Methods Engrg.* 38 (1995) 3831–3853.
- [98] C. Farhat, A. Macedo, M. Lesoinne, F.-X. Roux, F. Magoulès, A. de L. Bourdonnaie, "Two-level domain decomposition methods with Lagrange multipliers for the fast iterative solution of acoustic scattering problems", *Comput. Methods Appl. Mech. Eng.* 184 (2000) 213–239.
- [99] A. de La Bourdonnaye, C. Farhat, A. Macedo, F. Magoules, F.-X. Roux, "A non-overlapping domain decomposition method for the exterior Helmholtz problem", *Contemp. Math.* 218 (1998) 42–66.

- [100] M. Schäfer, G. Sieber, R. Sieber, I. Teschauer, "Coupled Fluid-Solid Problems: Examples and Reliable Numerical Simulation", in: Trends in Computational Structural Mechanics, Barcelona, 2001.
- [101] C. Michler, E.H. van Brummelen, S.J. Hulshoff, R. de Borst, "The relevance of conservation for stability and accuracy of numerical methods for fluid-structure interaction", *Comput. Methods Appl. Mech. Eng.* 192 (2003) 4195–4215.
- [102] O. Kurc, "Parallel Computing in Structural Engineering: A Substructure Based Approach", VDM Verlag, Saarbrücken, 2008.
- [103] M.R. Ross, "Coupling and simulation of acoustic fluid-structure interaction systems using localized Lagrange multipliers", 2006.
- [104] T.B. Jönsthövel, M.B. van Gijzen, S. MacLachlan, C. Vuik, A. Scarpas, "Comparison of the deflated preconditioned conjugate gradient method and algebraic multigrid for composite materials", *Comput. Mech.* 50 (2011) 321–333.
- [105] L. Conen, V. Dolean, R. Krause, F. Nataf, "A coarse space for heterogeneous Helmholtz problems based on the Dirichlet-to-Neumann operator", *J. Comput. Appl. Math.* 271 (2014) 83–99.
- [106] B. Vereecke, H. Bavestrello, D. Dureisseix, "An extension of the FETI domain decomposition method for incompressible and nearly incompressible problems", *Comput. Methods Appl. Mech. Eng.* 192 (2003) 3409–3430.
- [107] R.A. Nicolaides, "Deflation of Conjugate Gradients with Applications to Boundary Value Problems", *SIAM J. Numer. Anal.* 24 (1987) 355–365.
- [108] J.M. Tang, C. Vuik, TU Delft: Electrical Engineering, Mathematics and Computer Science, TU Delft, Delft University of Technology, "Two-level preconditioned conjugate gradient methods with applications to bubbly flow problems", (2008).
- [109] J.M. Tang, C. Vuik, TU Delft: Electrical Engineering, Mathematics and Computer Science, "An Efficient Deflation Method applied on 2-D and 3-D Bubbly Flow Problems", (2006).
- [110] M.B. Sawant, A. Dey, "Hydrodynamic Behavior of Pine Flat Dam Subjected to El-centro Strong Motion", in: Proc. Indian Geotech. Conf., Roorke, 2013.
- [111] "Pine Flat Dam", Wikipedia Free Encycl. (2015). https://en.wikipedia.org/w/index.php?title=Pine_Flat_Dam&oldid=653916804 (accessed July 9, 2015).

- [112] S. Balay, S. Abhyankar, M.F. Adams, J. Brown, P. Brune, K. Buschelman, et al., "PETSc Users Manual", Argonne National Laboratory, 2015. <http://www.mcs.anl.gov/petsc>.
- [113] T. Bahçecioğlu, U. Albostan, Ö. Kurç, "An Extensible Parallel Finite Element Analysis Environment: Panthalassa", in: Adv. Civ. Eng., Ankara, 2012.
- [114] S. Balay, S. Abhyankar, M.F. Adams, J. Brown, P. Brune, K. Buschelman, et al., "PETSc Web page", 2015. <http://www.mcs.anl.gov/petsc>.
- [115] G. Guennebaud, B. Jacob, others, "Eigen v3", 2010. <http://eigen.tuxfamily.org>.
- [116] G. Karypis, V. Kumar, "METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices, Version 4.0", 1998.
- [117] E. Cuthill, J. McKee, "Reducing the bandwidth of sparse symmetric matrices", in: ACM Press, 1969: pp. 157–172.
- [118] A. George, "Nested Dissection of a Regular Finite Element Mesh", SIAM J Numer Anal. 10 (1973) 345–363.
- [119] Y. Saad, "Numerical Methods for Large Eigenvalue Problems: Revised Edition", Siam, 2011.
- [120] A. Klawonn, O. Rheinbach, "A parallel implementation of Dual-Primal FETI methods for three dimensional linear elasticity using a transformation of basis", SIAM J Sci Comput. 28 (2006) 1886–1906.
- [121] J. Mandel, C.R. Dohrmann, R. Tezaur, "An algebraic theory for primal and dual substructuring methods by constraints", Appl. Numer. Math. 54 (2005) 167–193.
- [122] P.F. Fischer, "Projection techniques for iterative solution of $Ax = b$ with successive right-hand sides", Comput. Methods Appl. Mech. Eng. 163 (1998) 193–204.
- [123] T.A. Davis, "Direct Methods for Sparse Linear Systems (Fundamentals of Algorithms 2)", Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2006.
- [124] O. Kurc, S. Ozmen, "An efficient parallel solution framework for the linear solution of large systems on PC clusters", Tsinghua Sci. Technol. 13 (2008) 65–70.
- [125] M.N. Viladkar, A. Al-Assady, "Nonlinear Analysis Of Pine Flat Dam Including Base Sliding and Separation", in: 15th World Conf. Earthq. Eng. 15 WCEE, Lisbon, 2012.

APPENDIX A

FINITE ELEMENT FORMULATIONS

A.1 Pure Displacement Based Formulation

Since Wilson and Khalvati formulation [59] is widely utilized in the literature, this formulation is presented here. In this formulation, the elements are based on a pure displacement formulation in terms of the displacement DOFs at the nodes of the element. The surface sloshing motion and the effects of compressible wave propagation are included in the formulation. Although authors reported that the introduction of rotational constraints and the use of reduced integration techniques in the formulation of the element stiffness eliminate all zero energy modes, Bathe showed that for wide range frequencies it is correct but not whole range [62]. In this formulation small displacements are assumed for the fluid flow.

A.1.1 Theory

The proposed element is based on a formulation in which the fluid strains are obtained from the linear strain-displacement equations. This formulation removes the zero energy modes selectively. The method introduces the constraint of zero fluid rotation at the integration points and only the strain energy associated with the compressibility of the fluid is considered. The movement of the fluid particles approximated by the finite element displacement functions forms the basis for the kinetic energy. The formulation involves the change in the potential energy due to the low frequency sloshing of the fluid system. Besides, by introducing a rotational stiffness, the displacement field is constrained to be irrotational. In terms of the 3D elasticity, the relationship between pressure and volume for a linear fluid is given by

$$p = C_{11}e \quad (94)$$

where the pressure p denotes the magnitude of the mean stress. The constant C_{11} is the bulk modulus of the fluid. The volume change e can be stated in terms of the displacements by the following strain-displacement relationship equation:

$$e = \frac{dU_x}{dx}x + \frac{dU_y}{dy}y + \frac{dU_z}{dz}z \quad (95)$$

For imposing the rotational constraints, following rotations are defined:

$$e_x = \left(\frac{dU_y}{dy}z - \frac{dU_z}{dz}y \right) / 2 \quad (96)$$

$$e_y = \left(\frac{dU_z}{dz}x - \frac{dU_x}{dx}z \right) / 2 \quad (97)$$

$$e_z = \left(\frac{dU_x}{dx}y - \frac{dU_y}{dy}x \right) / 2 \quad (98)$$

The force and stiffness terms related to these rotations are defined by

$$p_x = C_{22}e_x \quad (99)$$

$$p_y = C_{33}e_y \quad (100)$$

$$p_z = C_{44}e_z \quad (101)$$

where C_{ii} is a constraint parameter and suggested to be taken in between 10-1000 [63]. The total strain energy of the fluid system is written as;

$$\Pi_e = \frac{1}{2} \int e^T C e dV \quad (102)$$

where $e^T = [e \ e_x \ e_y \ e_z]$ and the diagonal terms of the 4×4 , C matrix are the formerly defined bulk modulus and constraint parameters. In case of a large constraint parameter, the rotation and the strain energy related to the rotation approach to zero.

The ability of deforming without a change in volume is an important behavior of fluids. This movement is in the form of sloshing waves in which the deformation takes place in vertical direction. Provided that, the average vertical displacement of the fluid column is $U_s/2$; hence, the increase in the potential energy of the system is

$$\Pi_s = \frac{1}{2} \int U_s w U_s dA + \frac{1}{2} \int w D U_s dA \quad (103)$$

where the former integral yields the surface stiffness coefficients and the latter integral symbolizes the weight of the fluid that is usually evaluated as an element volume integral rather than as a surface integral.

The kinetic energy of the fluid system is given by

$$T = \frac{1}{2} \int m v^T v dV \quad (104)$$

where $v^T = [v_x \quad v_y \quad v_z]$.

A.1.2 Finite Element Discretization

The displacements within a characteristic element are expressed in terms of the nodal displacements by equations of the form

$$U_{ij}(x, y, z) = h_i u_j \quad (105)$$

or in matrix form

$$U_i = H_i U \quad (106)$$

where h is an $1 \times n$ array of interpolation functions for a node element and H is a $3 \times n$ array for 3D. The application of Equations (95), (96), (97) and (98) produce the following equation for the element volume changes and rotations in terms of the nodal displacements:

$$e_i = B_i U \quad (107)$$

The strain energy of the system, Equation (102), can now be expressed in terms of element properties and nodal displacement by

$$\Pi_e = \frac{1}{2} U^T K U \quad (108)$$

The surface potential energy, Equation (103), is expressed in terms of the vertical node displacements at the surface as

$$\Pi_s = \frac{1}{2} U_s^T S U_s \quad (109)$$

and the kinetic energy, Equation (104), can be written in the form

$$T = \frac{1}{2} V^T M V \quad (110)$$

where U and V are the vectors of nodal displacements and velocities. By applying Lagrange's equation, following equilibrium can be obtained:

$$Ma + KU + S U_s = R \quad (111)$$

where

- M is an symmetric matrix formed by the assembly of nodal mass coefficients that is defined as $M_i = m \int H_i^t H_i dV_i$ where m is unit mass of the field,
- K is an symmetric matrix formed by the assembly of nodal compressibility and rotational constraint terms which is defined as $K_i = \int B_i^t C_i B_i dV_i$,
- S is an symmetric matrix formed by the assembly of nodal surface potential terms that is defined as $S_i = w \int h_i^t h_i dA_i$ where w is the unit weight of the fluid,
- R denotes the time-dependent nodal forces vector.

It is reported that pure displacement based fluid elements have spurious non-zero frequency modes [58]. Therefore, it is suggested to use reduced integration for the integration of the stiffness of the elements. More detailed information about the spurious non-zero frequency modes and the irrotationality or penalty methods of the pure displacement based elements can be found on these references [58, 62].

A.2 Mixed Formulation – u/p (Displacement/Pressure Based)

A.2.1 Theory

In order to prevent from spurious non-zero frequency modes, Bathe [62] proposed replacing the pure displacement based formulation by a mixed displacement/pressure

formulation that fulfill the inf-sup condition. Using Hu-Washizu principle they defined a new variational indicator;

$$\Pi = \int_V \left\{ \frac{p^2}{2\beta} - \mathbf{u} \cdot \mathbf{f}^B - \lambda_p \left(\frac{p}{\beta} + \nabla \cdot \mathbf{u} \right) \right\} dV + \int_S \bar{p} u_n^S dS \quad (112)$$

where the variables are p , \mathbf{u} , and the Lagrange multiplier λ_p . In this equation, the first term corresponds to the strain energy expressed in terms of the pressure and the second term corresponds to the potential of the externally applied body forces that includes gravity effects. Then, the constitutive relationship is imposed by the third term. Finally, the potential due to any applied boundary pressure on S is presented by the last term. The effects of surface gravity waves is included with a surface gravitational potential term $\int_{S_s} \frac{1}{2} \rho g u_s^2 dS$ where S_s denotes the free surface.

Imposing the stationarity of Π , the Lagrange multiplier λ_p is defined by the pressure p and the governing equations with $\mathbf{f}^B = -\rho \ddot{\mathbf{u}}$,

$$\nabla p + \rho \ddot{\mathbf{u}} = \mathbf{0} \quad (113)$$

$$\nabla \cdot \mathbf{u} + \frac{p}{\beta} = 0 \quad (114)$$

With the boundary conditions

$$\begin{aligned} \mathbf{u} \cdot \mathbf{n} &= \bar{u}_n \text{ on } S_u \\ p &= \bar{p} \text{ on } S_f \end{aligned} \quad (115)$$

It is clear that Equations (113) and (114) are the momentum and mass conservation equations. The pressure \bar{p} is usually considered as zero on the free surface if the effects of surface gravity waves are ignored.

A.2.2 Finite Element Discretization

Applying the Galerkin discretization procedure, following discretization is obtained for a typical finite element;

$$U = H\hat{U} \quad (116)$$

$$P = H_p\hat{P} \quad (117)$$

$$\nabla \cdot u = (\nabla \cdot H)\hat{U} = B\hat{U} \quad (118)$$

where H and H_p are the interpolation matrices, and \hat{U} and \hat{P} are the vectors of solution variables. The matrix equations of the u/p formulation are given as;

$$\begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \hat{U} \\ \hat{P} \end{Bmatrix} + \begin{bmatrix} K_{uu} & K_{up} \\ K_{up}^T & K_{pp} \end{bmatrix} \begin{Bmatrix} \hat{U} \\ \hat{P} \end{Bmatrix} = \begin{Bmatrix} R \\ 0 \end{Bmatrix} \quad (119)$$

where

$$M = \int_V \rho H^T H dV \quad (120)$$

$$K_{up} = - \int_V B^T H_p dV \quad (121)$$

$$K_{pp} = - \int_V \frac{1}{\beta} H_p^T H_p dV \quad (122)$$

$$R = - \int_S \rho H_n^S \bar{p} dS \quad (123)$$

If the bulk modulus is finite, the pressure unknowns can be condensed out statically to form the following definition;

$$M_{uu}\hat{U} + S_{uu}\hat{U} = R \quad (124)$$

where S_{uu} is obtained from the static condensation of K_{uu} .

In Equation (119) the first and the second subpart of the stiffness matrix originated from the deviatoric strain energy and the volumetric strain energy, respectively. It is apparent that K_{uu} is positive definite for the shear modulus $G > 0$ and the rank of S_{uu} is equal to the number of pressure DOFs. The following three categories of problems can be considered:

- the solid bulk modulus κ and solid shear modulus G are of the same order
- $\kappa \gg G$ and $\kappa, G > 0$
- $\kappa > 0$ and $G = 0$

In the first case, the solvability and stability can be assured by the standard displacement formulation. In the second case, i.e. (almost) incompressible material

analysis, the displacement/pressure based formulation with mixed elements that fulfill the inf-sup condition is well established [49, 50]. The third category contains the inviscid acoustic fluid analysis. In this case, zero frequency modes corresponding to the zero deviatoric strain energy (for this case $K_{uu} = 0$) is introduced by the loss of ellipticity. In order to identify whether or not non-zero frequency spurious modes are formed, a mathematical prediction of the exact number of zero frequencies is required. For n displacement unknowns, the exact number of zero frequencies is $n - m$, provided that the physical constant pressure mode arising with the boundary condition $u \cdot n = \bar{u}_n$ on S_u has been eliminated.

A.2.3 2D Pressure Interpolation

In this section, a typical two-dimensional mixed finite element with three pressure DOFs in a local natural coordinate system (ξ, η) is examined. It is shown in Figure A.1. Circles on the element boundary represent the nodes associated with displacements, whereas triangles inside the element represent the nodes associated with the linear pressure approximation. Notice that the displacement nodes and pressure nodes are completely independent. In this section, only interpolation functions associated with the three pressure nodes are derived.

The three pressure nodes associated with a Q9P3up element are located as shown in Figure A.1. The following assumptions are made:

- The triangle that is formed from the three pressure nodes is equilateral.
- The centroid is located at the origin of the natural coordinate system; at $(\xi, \eta) \rightarrow (0, 0)$.
- Pressure node 1 is located along the η -axis.
- The distance from pressure node 1 to the centroid is d .

by using geometry, point coordinates are can be computed as;

$$\text{Pressure Node 1; } (\xi_1, \eta_1) \rightarrow (0, d)$$

$$\text{Pressure Node 2: } (\xi_2, \eta_2) \rightarrow \left(-\frac{\sqrt{3}d}{2}, -\frac{d}{2}\right)$$

$$\text{Pressure Node 3: } (\xi_3, \eta_3) \rightarrow \left(\frac{\sqrt{3}d}{2}, -\frac{d}{2}\right)$$

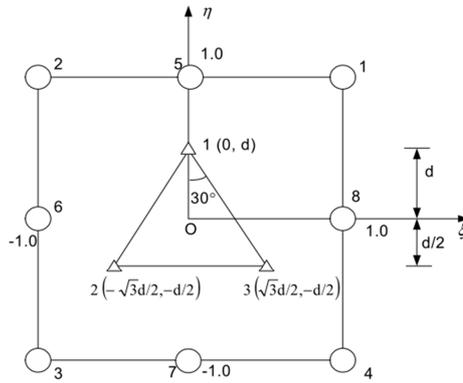


Figure A.1. Orientation of pressure nodes in 2D

by using the areal coordinates and assuming $d = 0.5$, interpolation functions for 2D pressure distribution can be obtained as;

$$N_1^{(p)} = \frac{4}{3}\eta + \frac{1}{3}$$

$$N_2^{(p)} = -\frac{2}{\sqrt{3}}\xi - \frac{2}{3}\eta + \frac{1}{3}$$

$$N_3^{(p)} = \frac{2}{\sqrt{3}}\xi - \frac{2}{3}\eta + \frac{1}{3}$$

A.2.4 3D Pressure Interpolation

In this section, a typical three-dimensional mixed finite element with four pressure DOFs in a local natural coordinate system (ξ, η, ζ) is considered. It is shown in

Figure A.2. Similarly, circles on the element boundary represent the nodes associated with displacements, and triangles inside the element represent the nodes associated with the pressures. The following assumptions are made:

- The tetrahedron that is formed from the four pressure nodes is equilateral.
- The centroid is located at the origin of the local natural coordinate system $(\xi, \eta, \zeta) \rightarrow (0, 0, 0)$
- Pressure node 1 is located along the ζ axis.
- Line 13 lies in the $\eta - \zeta$ plane.
- The distance from node 1 to the centroid is d .
- The angle between O1 and 13 is θ .

Notice that the displacement nodes and pressure nodes are completely independent. In this section, only interpolation functions associated with the four pressure nodes are derived. The length of each equilateral line of the tetrahedral can be computed using geometry as $3d/(2 \cos \theta)$.

From tetrahedral, it can be shown that $\sin \theta = \frac{\sqrt{3}}{3}$, hence;

$$\theta = \sin^{-1} \frac{\sqrt{3}}{3} \cong 35.264$$

The coordinates of each pressure nodes can be computed from the geometry of the tetrahedron as follows:

$$\text{Pressure Node 1; } (\xi_1, \eta_1, \zeta_1) \rightarrow (0, 0, d)$$

$$\text{Pressure Node 2; } (\xi_2, \eta_2, \zeta_2) \rightarrow \left(\frac{3d}{4 \cos \theta}, -3d \tan \frac{\theta}{4}, -\frac{d}{2} \right)$$

$$\text{Pressure Node 3; } (\xi_3, \eta_3, \zeta_3) \rightarrow \left(0, 3d \tan \frac{\theta}{2}, -\frac{d}{2} \right)$$

$$\text{Pressure Node 4; } (\xi_4, \eta_4, \zeta_4) \rightarrow \left(-\frac{3d}{4 \cos \theta}, -3d \tan \frac{\theta}{4}, -\frac{d}{2} \right)$$

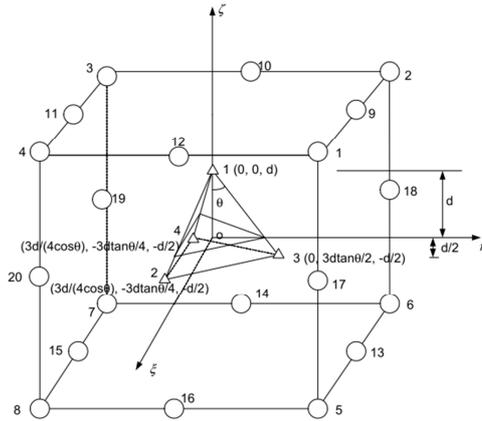


Figure A.2. Orientation of pressure nodes in 3D

by using the volume coordinates and assuming $d = 0.5$, interpolation functions for 3D pressure distribution can be obtained as;

$$N_1^{(p)} = -\frac{2}{3d}\zeta + \frac{1}{3}$$

$$N_2^{(p)} = \frac{2 \cos \theta}{3d}\xi - \frac{2}{9d \tan \theta}\eta - \frac{2}{9d}\zeta + \frac{2}{9}$$

$$N_3^{(p)} = \frac{4}{9d \tan \theta}\eta - \frac{2}{9d}\zeta + \frac{2}{9}$$

$$N_4^{(p)} = -\frac{2 \cos \theta}{3d}\xi - \frac{2}{9d \tan \theta}\eta - \frac{2}{9d}\zeta + \frac{2}{9}$$

APPENDIX B

VALIDATIONS

B.1 Validations of Finite Element Implementations

In order to validate the finite element implementations several benchmark problems solved by serial implementation of the solution framework (which is developed on MATLAB as discussed in CHAPTER 3) by utilizing a direct solver for system solutions and explicit time integration algorithm for time history analysis. Four problems are determined as benchmarks. In first case, static and dynamic analysis of a cantilever beam problem is carried out for the validation of structural domain finite elements. In the second case hydrostatic pressures of water tank is computed for the validation of hydrostatic pressure results obtained. Finally, a tall water column under dynamic loading is investigated.

B.1.1 Cantilever Beam Problem

This problem consists of a cantilever beam subjected to a tip moment. By restricting the span to depth ratio of beam, shear deformations are assumed to be negligible. Therefore, Euler-Bernoulli Beam Theory can be used as a guide for determining reasonable problem dimensions. According to mentioned theory, deflection at any point of the span is given as $\vartheta(x) = Mx^2/2EI$. Finally, material and geometric properties are chosen according to equate tip deflection to unity. In order to prevent from the Saint-Venant's effect, displacement comparisons are carried out at the mid-span. Therefore results obtain at this point should be around 0.25.

In order to test the performance of finite elements, 4x40 mesh is utilized for linear displacement approximation elements and 2x6 mesh for quadratic displacement approximation. Tests are carried out for three different Poisson ratios 0.25, 0.45 and 0.4999 and the results are compared with the respective ADINA finite element module.

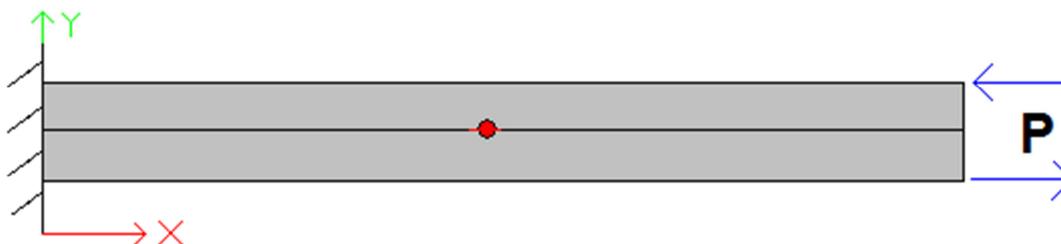


Figure B.1. Cantilever beam under static loading

Initial results are given for plane strain condition. As it can be seen displacements obtained are closely equal to the results obtained from ADINA and they decrease with the increasing Poisson ratio because of the plane strain constraint in transverse. Second outcome is that quadratic finite elements are successively produce better results than the linear ones. However, finite elements with incompatible modes are similarly equal results with the quadratic elements without any computational drawback. The last and the most important observation is that with the increasing Poisson ratio the displacement results obtained are diverging from the pseudo-theoretical displacement result 0.25. However, mixed formulation finite elements can represent the behavior even in nearly incompressible state.

Table B.1. Validation of 2D solid finite elements with plane strain assumption

		$\nu = 0.25$			$\nu = 0.45$			$\nu = 0.4999$		
		Value	ADINA	% Error	Value	ADINA	% Error	Value	ADINA	% Error
sQ4P0u	4x40	0.21203	0.21203	0.00024	0.16194	0.16194	0.00025	0.00214	0.00214	0.00094
sQ4IP0u	4x40	0.23356	0.23356	0.00021	0.19539	0.19539	0.00026	0.18189	0.18189	0.00016
sQ4P1u/p	4x40	0.21998	0.21998	0.00000	0.19096	0.19097	0.00021	0.17989	0.17990	0.00006
sQ9P0u	2x6	0.23230	0.23230	0.00000	0.18233	0.18233	0.00016	0.14442	0.14442	0.00014
sQ9P3u/p	2x6	0.23308	0.23308	0.00000	0.19298	0.19298	0.00016	0.17824	0.17824	0.00017

Second group of results are obtained for plane stress condition. Similar to the previous discussion, the displacements obtained are closely equal to the results obtained from ADINA. In contrast they do not apparently decrease with the increasing Poisson ratio because of the released constraint in transverse. Again, the quadratic finite elements are successively produce better results than the linear ones. However, finite elements with incompatible modes yields almost equal results with the quadratic elements. These test results verifies the assumption of plane stress condition for any Poisson ratio.

Table B.2. Validation of 2D solid finite elements with plane stress assumption

		$\nu = 0.25$			$\nu = 0.45$			$\nu = 0.4999$		
		Value	ADINA	% Error	Value	ADINA	% Error	Value	ADINA	% Error
sQ4P0u	4x40	0.22597	0.22597	0.00013	0.22539	0.22539	0.00004	0.22459	0.22459	0.00009
sQ4IP0u	4x40	0.24950	0.24950	0.00004	0.24845	0.24845	0.00016	0.24810	0.24810	0.00012
sQ9P0u	2x6	0.24894	0.24894	0.00004	0.24574	0.24574	0.00016	0.24457	0.24457	0.00016

Final result group is for 3D general stress state condition. As it can be seen displacements obtained are closely equal to the results obtained from ADINA and they decrease with the increasing Poisson ratio because of the plane strain constraint in transverse. Second outcome is that quadratic finite elements are successively produce better results than the linear ones. However, finite elements with incompatible modes produced almost equal results with the quadratic elements. The last and the most important observation is that with the increasing Poisson ratio the displacement results obtained are diverging from the pseudo-theoretical displacement result 0.25. However, mixed formulation finite elements can represent the behavior even in nearly incompressible state; especially sH27P4u/p produces the best results for any condition.

Table B.3. Validation of 3D solid finite elements

		$\nu = 0.25$			$\nu = 0.45$			$\nu = 0.4999$		
		Value	ADINA	% Error	Value	ADINA	% Error	Value	ADINA	% Error
sH8P0u	4x40	0.22464	0.22461	0.01545	0.21168	0.21162	0.02892	0.07562	0.07562	0.00106
sH8IP0u	4x40	0.24876	0.24876	0.00012	0.24459	0.24459	0.00008	0.22930	0.22930	0.00013
sH8P1u/p	4x40	0.23058	0.23054	0.01505	0.23024	0.23018	0.02585	0.22962	0.22955	0.02836
sH27P0u	2x6	0.24662	0.24663	0.00041	0.22400	0.22400	0.00000	0.18999	0.18999	0.00000
sH27P4u/p	2x6	0.24877	0.24877	0.00048	0.24499	0.24497	0.00935	0.24308	0.24303	0.01843

As a final comment, 2D and 3D implementations of pure displacement and mixed displacement/pressure formulation for solid mechanics are both verified in displacement sense.

B.1.2 Hydrostatic Pressure of Water Tank

In this problem, both pure displacement and mixed displacement/pressure formulations for fluid media is investigated. Water in the tank is meshed with 5 fluid elements and no external load is except from the self-weight of the fluid. Fluid elements are restrained in X direction and depth is measured from free surface.

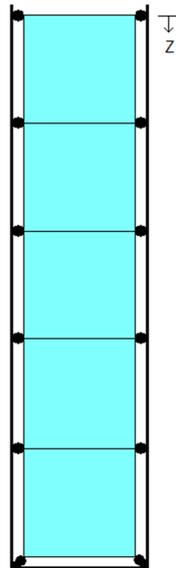


Figure B.2. Water tank with its self-weight

Pressure results obtained from the solution compared with the exact solution given from the hydrostatic pressure of reservoir which is $P(z) = \rho g z$ where ρ is the density of fluid, g is the gravitational acceleration and z is the depth. Missing results of mid points of Q4 elements are obtained by interpolation.

In the first test group, performance of the 2D elements is investigated. Although all results are extremely close to exact values, as it can be seen in Table B.4, the best

results are obtained with fQ4P1u element which utilizes linear displacement approximation and constant pressure. The reason of that is the hydrostatic behavior is a linear behavior.

Table B.4. Hydrostatic pressures obtained from 2D model

Depth	Exact	fQ4P1u	Rel.Error	fQ4P1u/p	Rel.Error	fQ9P1u	Rel.Error	fQ9P3u/p	Rel.Error
0	0.00000	0.00000	0.0E+00	0.00000	0.0E+00	0.00000	0.000000	0.00000	0.000000
500	0.00490	0.00490	-4.1E-13	0.00489	-2.7E-03	0.00480	-2.2E-02	0.00489	-2.7E-03
1000	0.00981	0.00981	-2.5E-13	0.00978	-2.7E-03	0.00959	-2.2E-02	0.00978	-2.7E-03
1500	0.01471	0.01471	-2.0E-13	0.01467	-2.7E-03	0.01470	-5.3E-04	0.01467	-2.7E-03
2000	0.01962	0.01962	-1.5E-13	0.01956	-2.7E-03	0.01960	-7.9E-04	0.01956	-2.7E-03
2500	0.02452	0.02452	-1.2E-13	0.02446	-2.7E-03	0.02452	-2.3E-05	0.02446	-2.7E-03
3000	0.02943	0.02943	-8.5E-14	0.02935	-2.7E-03	0.02942	-3.8E-05	0.02935	-2.7E-03
3500	0.03433	0.03433	-5.8E-14	0.03424	-2.7E-03	0.03433	-2.6E-06	0.03424	-2.7E-03
4000	0.03923	0.03923	-5.1E-14	0.03913	-2.7E-03	0.03923	-2.0E-06	0.03913	-2.7E-03
4500	0.04414	0.04414	-4.5E-14	0.04402	-2.7E-03	0.04414	-3.3E-07	0.04402	-2.7E-03
5000	0.04904	0.04904	-4.1E-14	0.04891	-2.7E-03	0.04904	-3.4E-08	0.04891	-2.7E-03

Second test group composed of the modals with 3D elements. Similarly all results are in a highly correspondence with the exact results. However, linear displacement element with constant pressure fH8P1u provides the best results.

Table B.5. Hydrostatic pressures obtained from 3D model

Depth	Exact	fH8P1u	Rel.Error	fH8P1u/p	Rel.Error	fH27P1u	Rel.Error	fH27P4u/p	Rel.Error
0	0.00000	0.00000	0.0E+00	0.00000	0.0E+00	0.00000	0.000000	0.00000	0.000000
500	0.00490	0.00490	-1.4E-12	0.00489	-2.7E-03	0.00490	-2.6E-06	0.00489	-2.7E-03
1000	0.00981	0.00981	-3.1E-12	0.00978	-2.7E-03	0.00981	1.1E-12	0.00978	-2.7E-03
1500	0.01471	0.01471	-3.7E-12	0.01467	-2.7E-03	0.01471	-2.6E-06	0.01467	-2.7E-03
2000	0.01962	0.01962	-2.5E-12	0.01956	-2.7E-03	0.01962	7.6E-12	0.01956	-2.7E-03
2500	0.02452	0.02452	-1.8E-12	0.02446	-2.7E-03	0.02452	1.5E-06	0.02446	-2.7E-03
3000	0.02943	0.02943	-1.6E-12	0.02935	-2.7E-03	0.02943	7.0E-12	0.02935	-2.7E-03
3500	0.03433	0.03433	-1.5E-12	0.03424	-2.7E-03	0.03433	3.1E-07	0.03424	-2.7E-03
4000	0.03923	0.03923	-1.5E-12	0.03913	-2.7E-03	0.03923	6.1E-12	0.03913	-2.7E-03
4500	0.04414	0.04414	-1.5E-12	0.04402	-2.7E-03	0.04414	-3.3E-07	0.04402	-2.7E-03
5000	0.04904	0.04904	-1.5E-12	0.04891	-2.7E-03	0.04904	5.5E-12	0.04891	-2.7E-03

As a final comment, with these tests, the pressure computation from fluid finite elements and their performances are verified for hydrostatic case.

B.1.3 Explicit Dynamic Analysis of Cantilever Beam Problem

In this problem, same model with the first example is utilized with just replacing the load with a dynamic load.

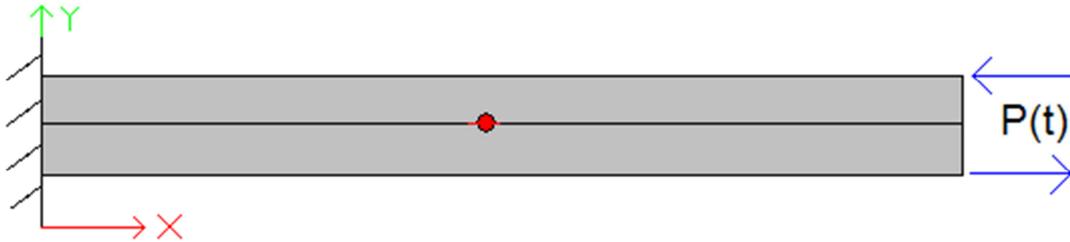
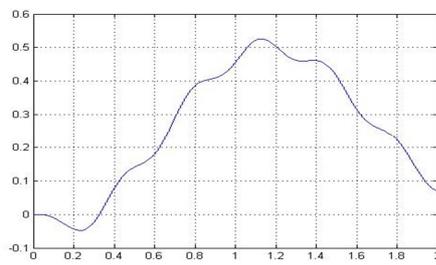
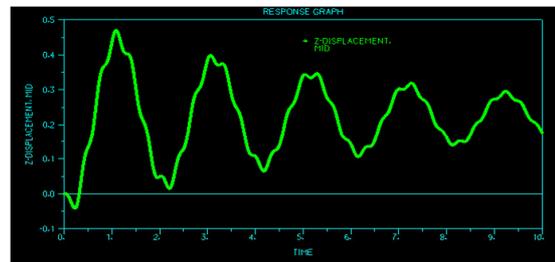


Figure B.3. Cantilever beam problem under dynamic loading

The same magnitude of load is applied to the system for just 0.1 seconds and then loading is released. Rayleigh damping is applied with $\alpha = 0$ and $\beta = 0.3$ and following graph is obtained.



(a)



(b)

Figure B.4. Tip deflection comparison with ADINA

Obtained results from this problem solution is in correspondence with the results obtained from ADINA.

B.1.4 Explicit Dynamic Analysis of Tall Water Column

In this problem, both pure displacement and mixed displacement/pressure formulations for fluid media under dynamic loading is investigated. Water tank meshed with 10 fluid elements and restraint in X direction to represent the rigid tank boundary. Tank has 1/10 width height ratio.

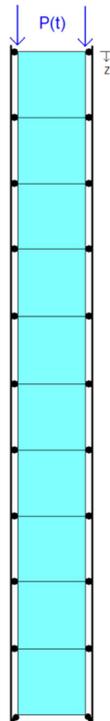


Figure B.5. Tall water column under dynamic loading

Uniform area load which is lumped to the nodes using tributary area is applied to the system with the following time function;

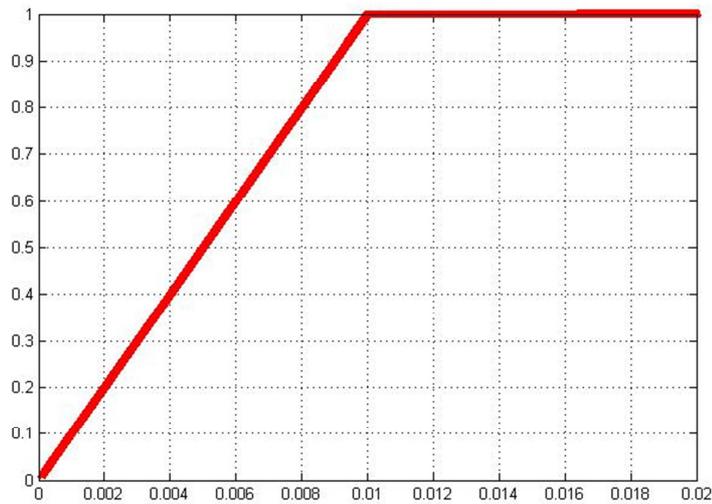
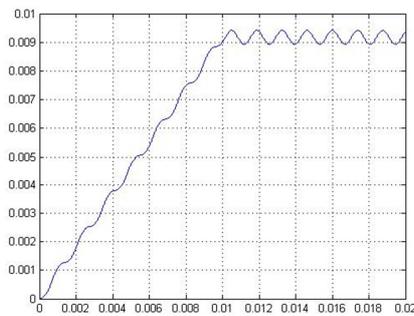
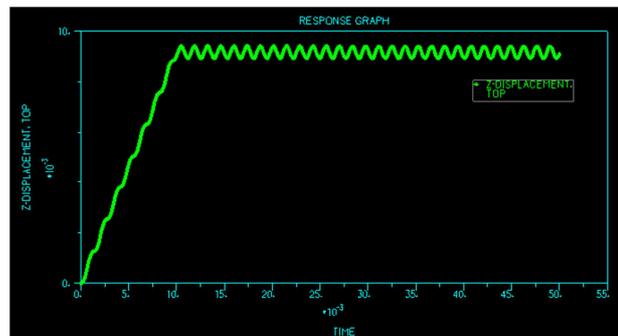


Figure B.6. Step force with finite rise time function ($t_r = 0.01$ s)

Rise time is taken as 0.01 where natural period of the system is around 0.0016. T/T_n ratio is in between 0.5 and 1, hence, the dynamic response of mid-span displacement is in correspondence with the given figure by Chopra [40].



(a)



(b)

Figure B.7. Free surface response (for $t_r = 0.01$ s) comparison with ADINA

Rise time is lowered 5 times and the following figure is obtained, as it can be seen mean of the displacements after rise time is exactly same with the static solution of the system.

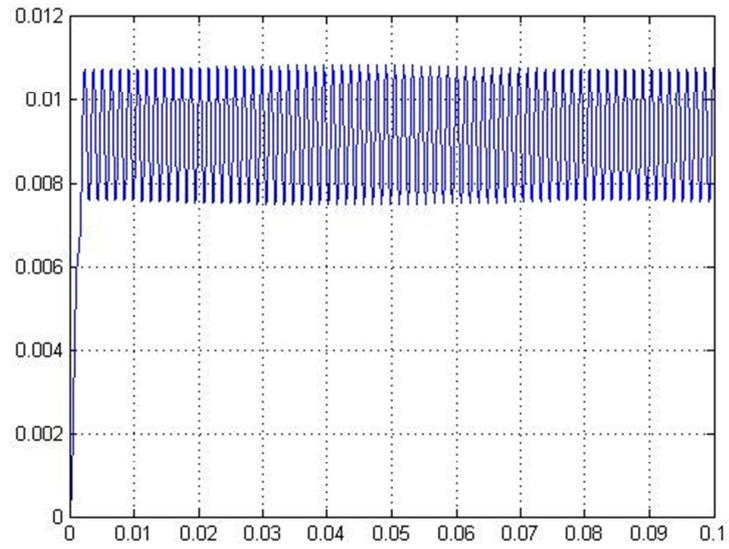


Figure B.8. Free surface response for $t_r = 0.002 s$

Obtained results from this problem solution is in correspondence with the results obtained from ADINA in the order of 0.01%.

B.2 Validations of Parallel Implementation of Solution Framework

In order to validate the parallel solution framework described in Section 5.2 mechanically, three benchmark problems are determined. During these tests, solution method with SH-ASM deflation presented in Section 5.3 is utilized by using 4 processors.

B.2.1 Cantilever Beam

The static analysis of a cantilever beam problem utilized in this validation is exactly same as the problem defined in Section B.1.1 except that quadrilaterals are assumed to be plain strain since the solutions with plane stress state results are almost same. sQ4IP0u elements are utilized for the discretization. In Figure B.9, displacements on X and Y directions and stress magnitude computed by Zargana solver is given. The same values obtained from ADINA are presented in Figure B.10 and when these two sets are compared, obviously results are same.

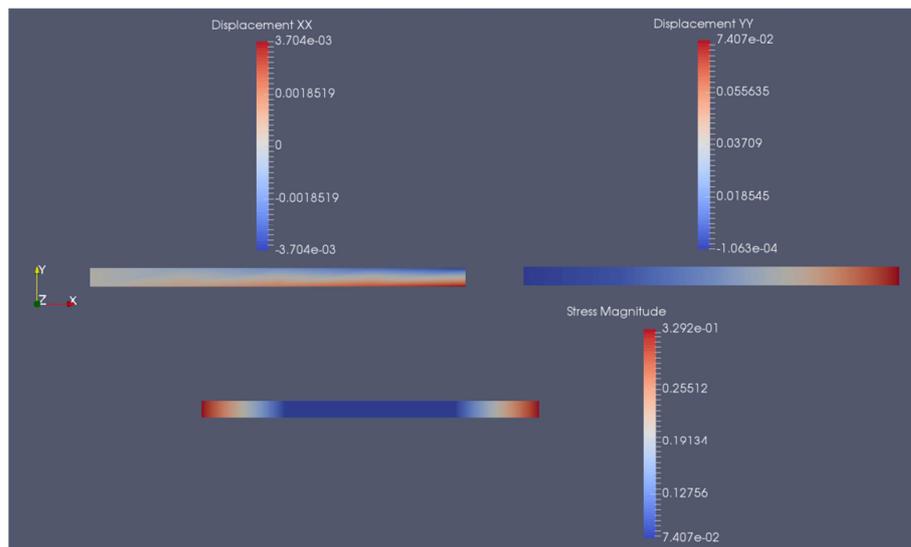


Figure B.9. Displacements on X and Y directions and stress magnitude by Zargana

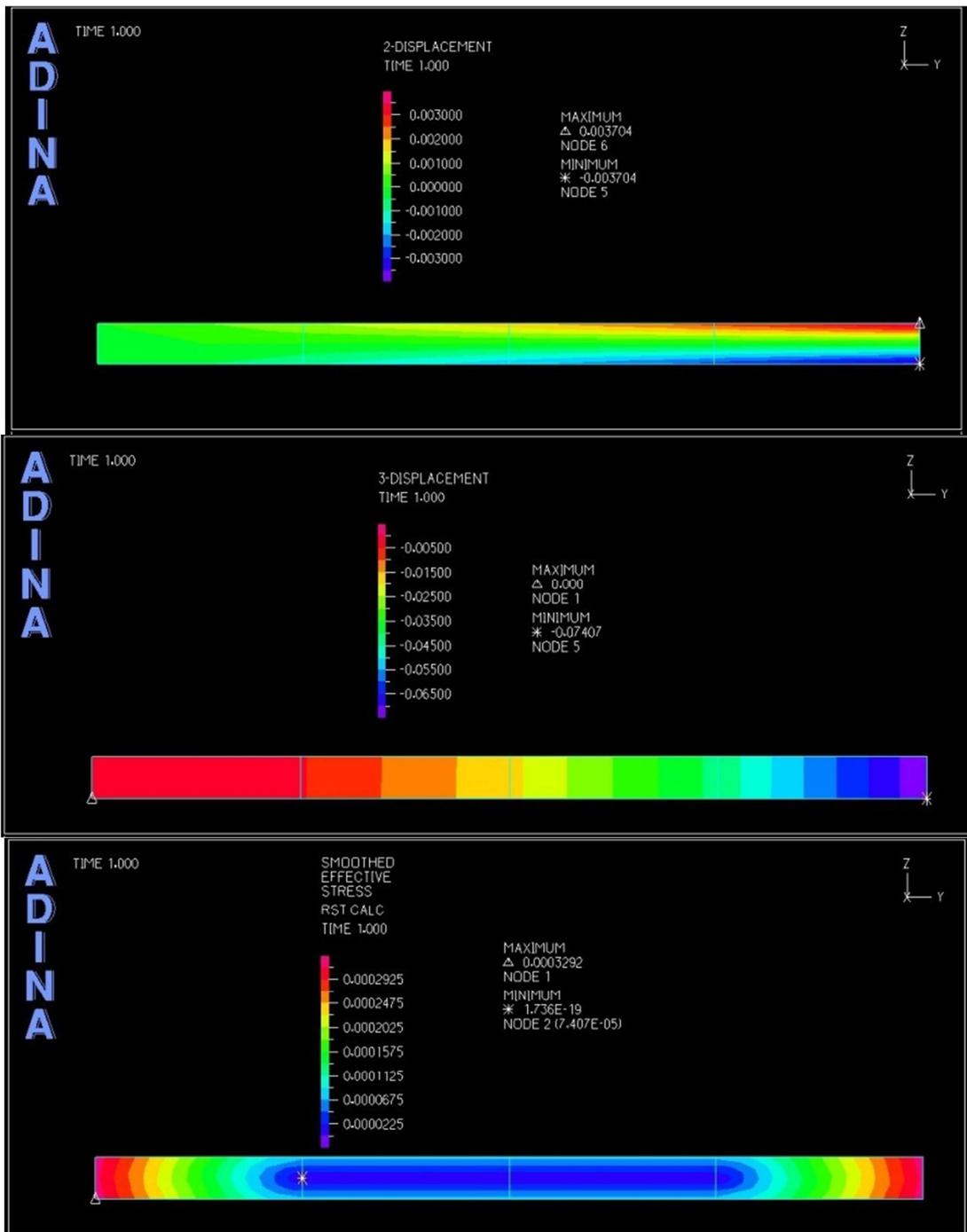


Figure B.10. Displacements on X and Y directions and stress magnitude by ADINA

B.2.2 Water Tank Problem

In this case, the water tank problem with flexible walls presented in Section 4.4.1 is solved, but for this case analysis is carried out as static with a constant uniform load on the exterior of right wall. Q4 elements are utilized for the discretization. Same problem is modelled in ADINA and “Displacement Based Fluid Element with Rotation Penalty” element is utilized. Displacements on X-Y directions and stress contour plots obtained from Zargana is given in Figure B.11, Figure B.12 and Figure B.13, whereas the ones obtained from ADINA are presented in Figure B.14, Figure B.15, and Figure B.16. As it can be seen results are almost same.

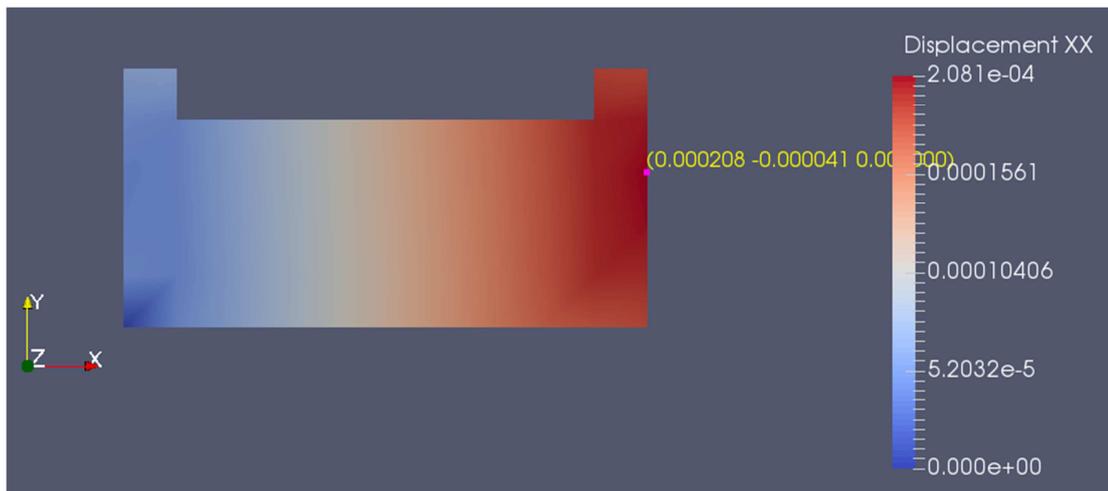


Figure B.11. Displacements on X direction by Zargana

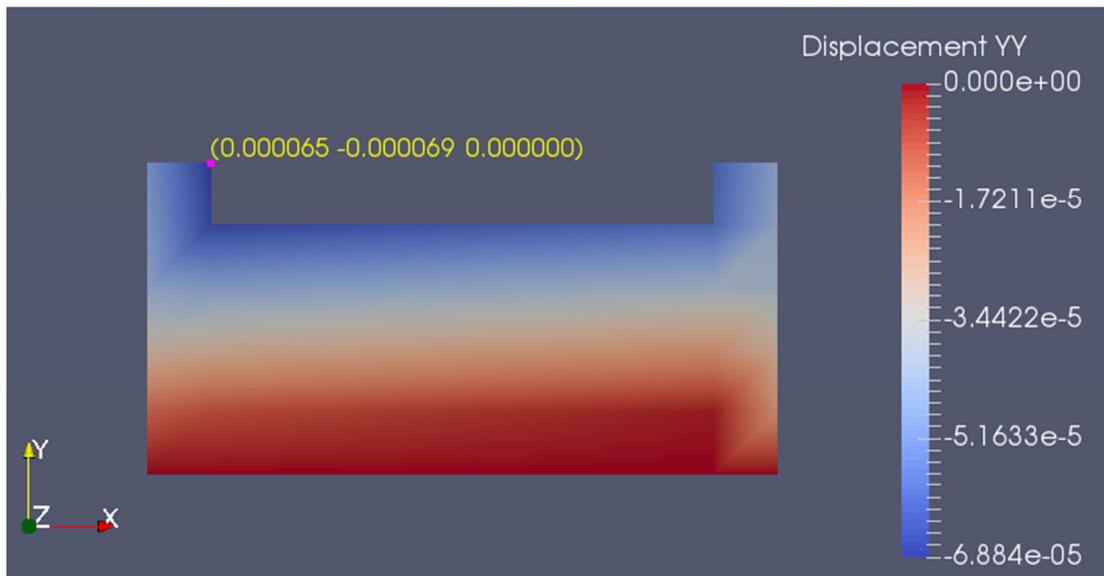


Figure B.12. Displacements on Y direction by Zargana

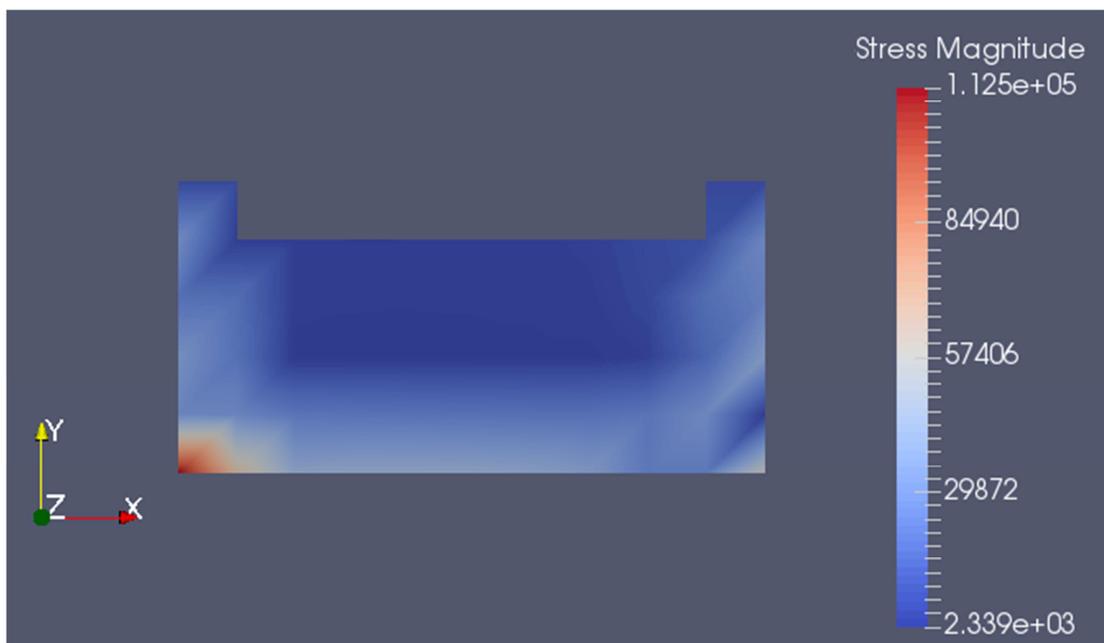


Figure B.13. Stress magnitude by Zargana

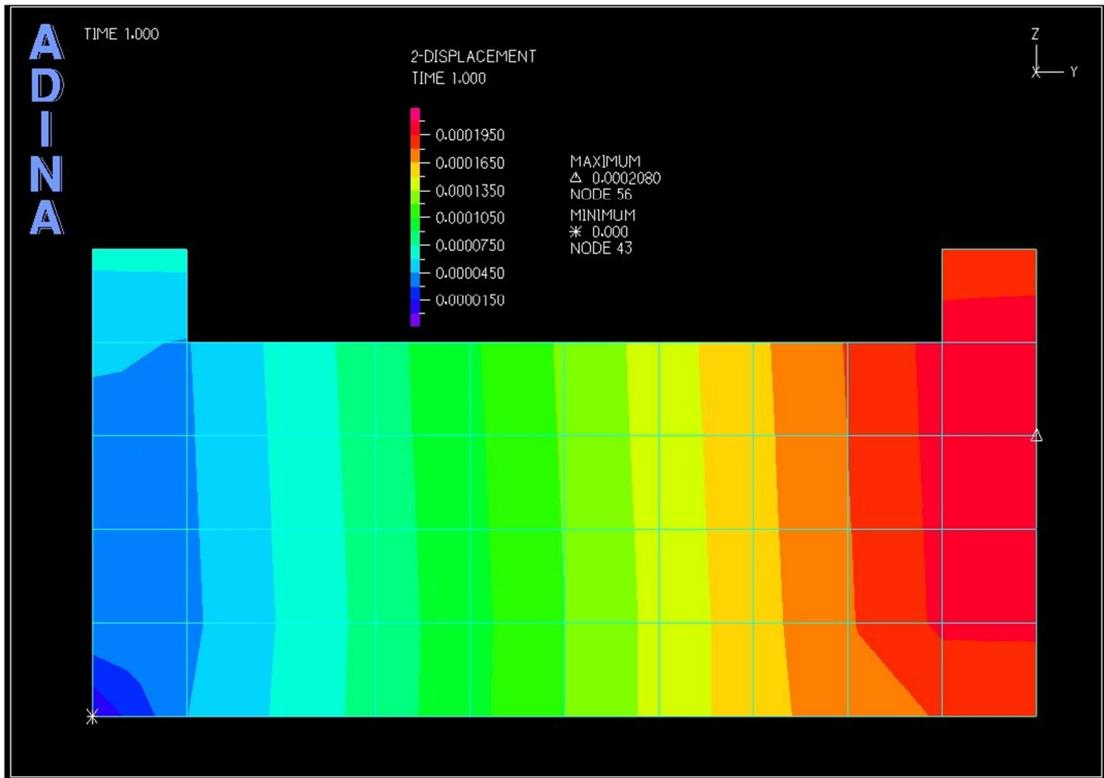


Figure B.14. Displacements on X direction by ADINA

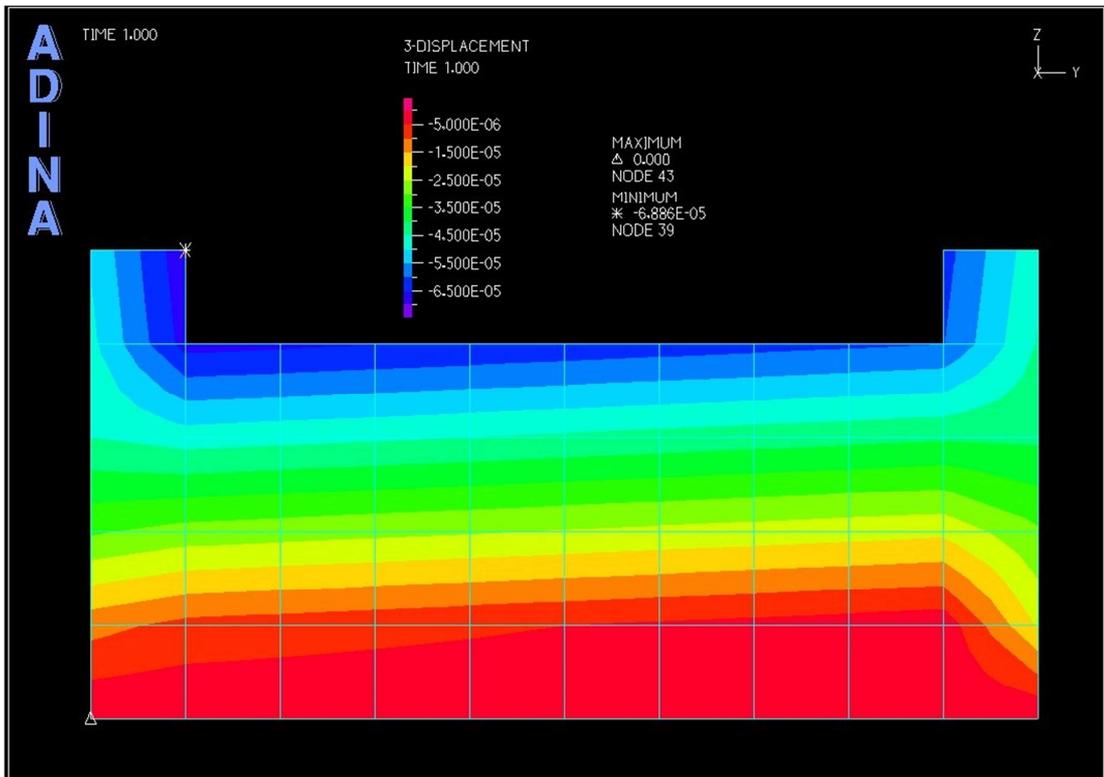


Figure B.15. Displacements on Y direction by ADINA

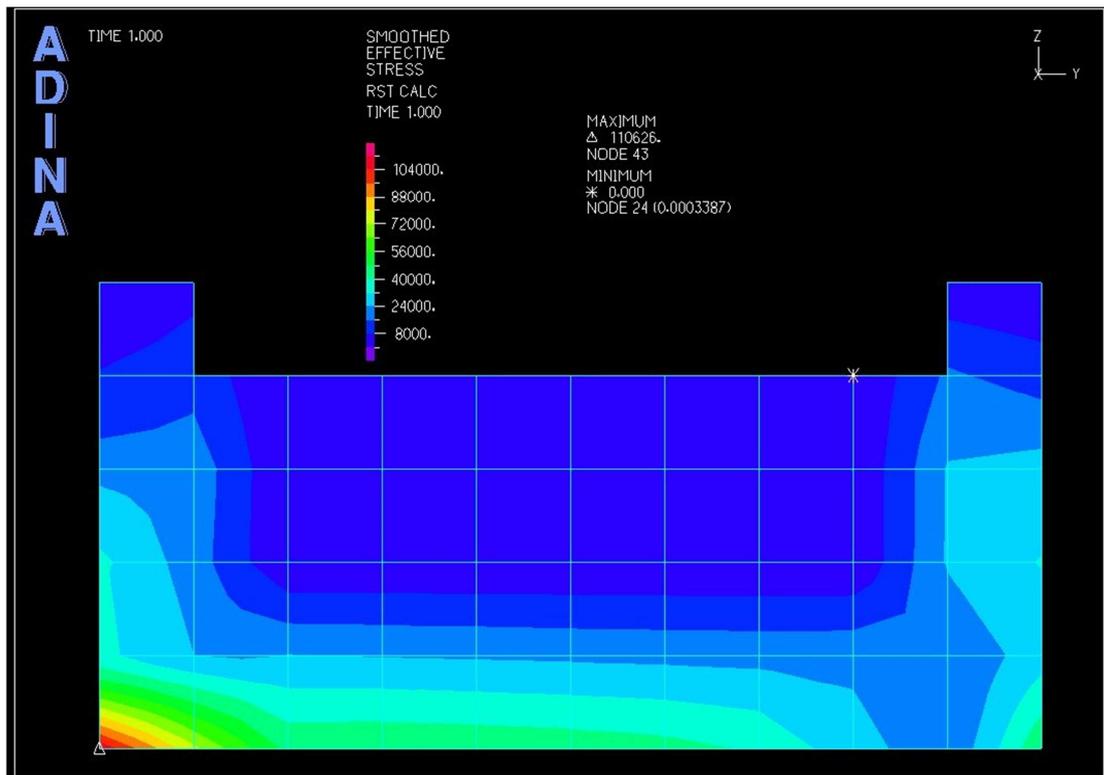


Figure B.16. Stress magnitude by ADINA

B.2.3 Pine Flat Dam Problem

As a final validation, Pine Flat dam problem presented in Section 4.4.2 is solved. However, in this case the half-circular extension under the foundation is removed for comparative reasons with other researches. Q4 elements are utilized for the discretization. By using the same dimensions and material properties, static analysis of the dam is carried out under its own self-weight. Displacements on X-Y directions and stress contour plots obtained from Zargana and ADINA are given in following figures. As it can be seen from the figures, the results obtained from Zargana are in correspondence with the results from ADINA. Besides, the same problem is investigated in literature [110, 125] and it is reported that horizontal crest displacement and maximum stress magnitude for this case is approximately 11 cm and 1300 KPa.

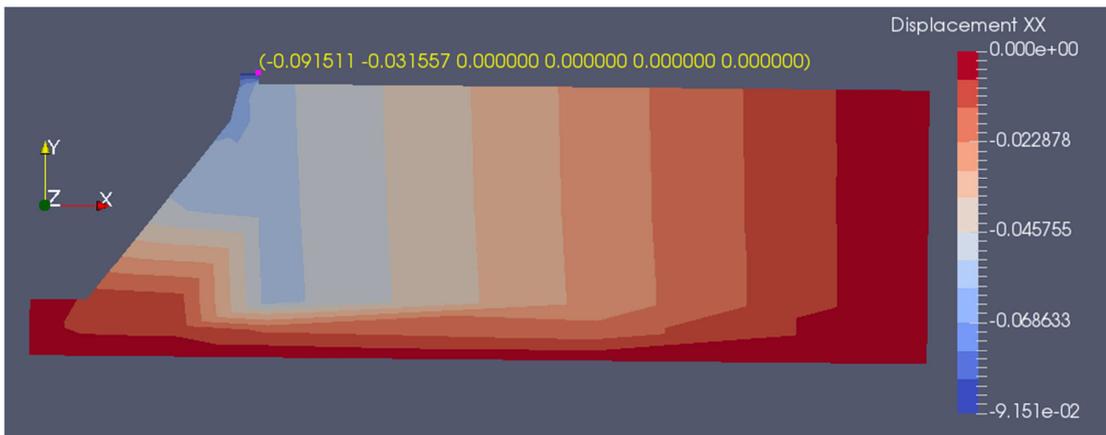


Figure B.17. Displacements on X direction by Zargana

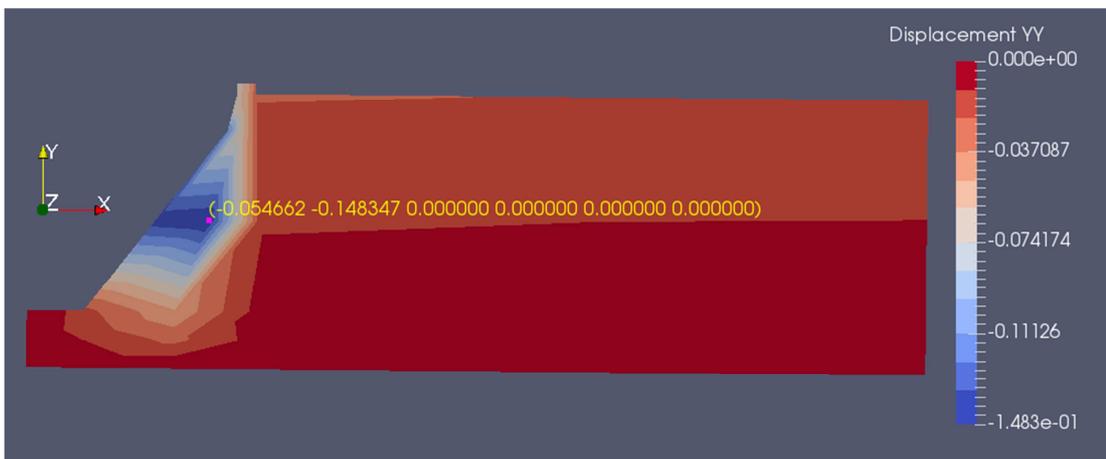


Figure B.18. Displacements on Y direction by Zargana



Figure B.19. Stress magnitude by Zargana

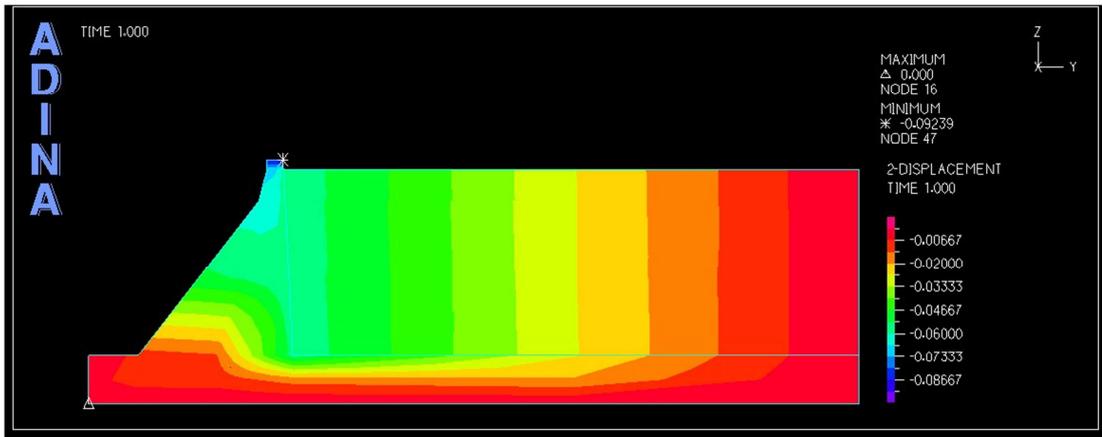


Figure B.20. Displacements on X direction by ADINA

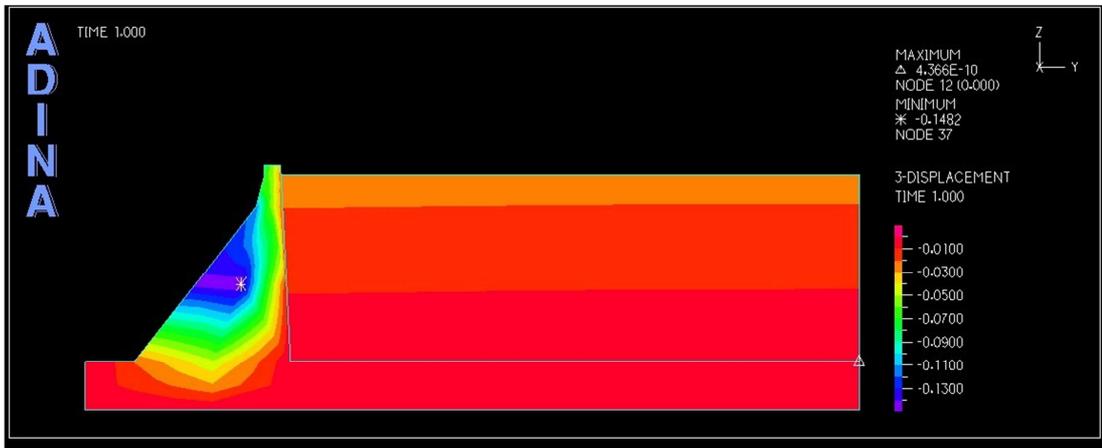


Figure B.21. Displacements on Y direction by ADINA

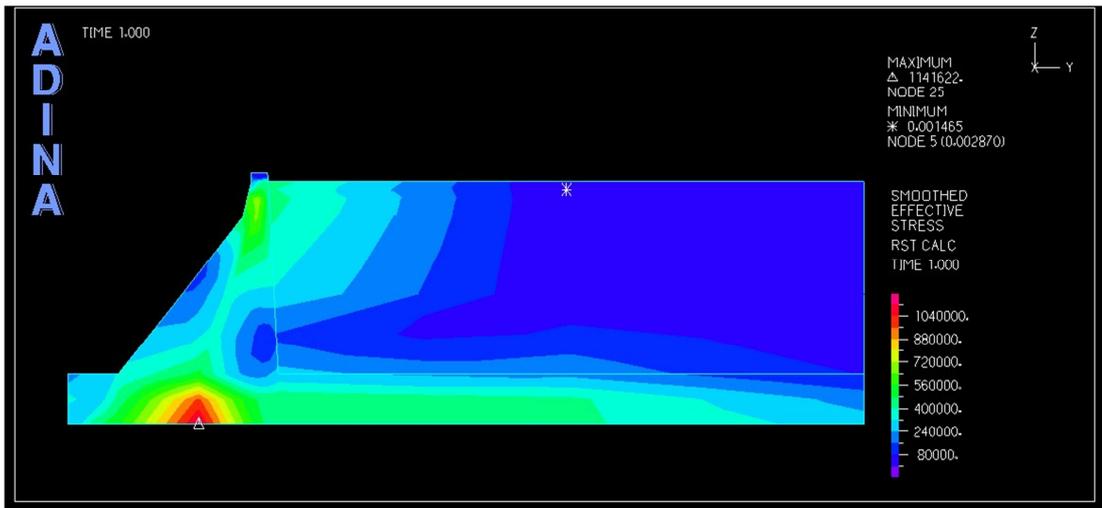


Figure B.22. Stress magnitude by ADINA

CURRICULUM VITAE

PERSONAL INFORMATION

Name, Surname: Semih Özmen

Date and Place of Birth: 1983, Eskişehir, Turkey

E-mail: semihozmen@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
MS	METU Civil Engineering	July 2009
BS Minor	METU RS/GIS	June 2006
BS	METU Civil Engineering	June 2006

WORK EXPERIENCE

Working as Senior Structural Software Engineer in Proya Software Development, one of the subsidiaries of PROKON-EKON group of companies.

PUBLICATIONS

- O. Kurc, S. Ozmen, “An Efficient Parallel Solution Framework for the Linear Solution of Large Systems on PC Clusters”, presented in 12th International Conference on Computing in Civil and Building Engineering, Beijing, China, October, 2008 & published in Journal of Tsinghua Science & Technology, Vol. 13, Supp. 1, Oct. 2008, Pg. 65-70
- T. Bahcecioglu, S. Ozmen, O. Kurc, “A Comparative Study on Two Different Direct Parallel Solution Strategies for Large-Scale Problems”, presented in The First International Conference on Parallel, Distributed and Grid Computing for Engineering, Pécs, Hungary, April, 2009
- S. Ozmen, T. Bahcecioglu, O. Kurc, “A Substructure Based Parallel Dynamic Solution of Large Systems on Homogeneous PC Clusters”, presented in Advances in Civil Engineering, Trabzon, Turkey, September, 2010
- S. Ozmen, T. Bahcecioglu, O. Kurc, “Yoğun Matrislerin Grafik İşlem Birimleri Yardımıyla İndirgenmesi”, presented in Başarım2012; Third National Conference on High Performance Computing, Ankara, Turkey, 2012