## UC Riverside UC Riverside Electronic Theses and Dissertations

#### Title

Improving the Performance of Partitioned Methods for Solid-Fluid Coupling

Permalink https://escholarship.org/uc/item/1383393s

**Author** Akbay, Muzaffer

Publication Date 2018

Peer reviewed|Thesis/dissertation

#### UNIVERSITY OF CALIFORNIA RIVERSIDE

Improving the Performance of Partitioned Methods for Solid-Fluid Coupling

A Dissertation submitted in partial satisfaction of the requirements for the degree of

Doctor of Philosophy

 $\mathrm{in}$ 

Computer Science

by

Muzaffer Akbay

September 2018

Dissertation Committee:

Dr. Tamar Shinar, Chairperson Dr. Jiasi Chen Dr. Michaelis Faloutsos Dr. Ertem Tuncel

Copyright by Muzaffer Akbay 2018 The Dissertation of Muzaffer Akbay is approved:

Committee Chairperson

University of California, Riverside

#### Acknowledgments

Foremost, I would like to sincerely thank Dr. Shinar, who has been guiding me through this long journey, and who has taught me the delight of working in this subject. Not only has she been a great academic advisor with her knowledge, experience, and enthusiasm, but also an amazing mentor for my personal life, helping me endure frequent frustration. I will always value her advice as much as her sympathy and support during distressing times.

I would like to express my gratitude to Dr. Zordan, who academically supervised my research in the first years of the Ph.D. Thanks to his passion for research, we were able to collaborate even after he left the university and accomplish acknowledged studies, but most of all, I would like to thank him for his care for me as a person, and for being such an enthusiastic, encouraging, and energetic person.

I would like to thank my fellow labmates, Nicholas (Nobles), Steve, William, Wojciech, Adriano, James, Marc and Lingli for all their effort and help.

I also have sincere gratitude toward all the esteemed members of my committee for their valuable time, constructive feedback, and indispensable encouragement and guidance.

To my parents, Fatma and Kemal, thank you from the bottom of my heart. Even though you are continents away, I always felt your endless support, encouragement, and love for me. Thank you for all your efforts in raising me and teaching me how to walk on this path and all the values and virtues that made me who I am today. There are no proper words to express my gratitude and respect for both of you.

I deeply thank my "little" brother Ertuğrul (Deniz), for sharing all the joy and grief with me during my life and for being not only a good brother, but also a great friend that I can rely on. I'm especially grateful for all the support and faith he puts in me. I need to add my special thanks to his lovely wife, Aylin, as well as my nephew Çıntar.

I would be forever thankful to all members of the Koksal's family. I'm really lucky to have them in a place far away from my home country. I would like to thank Makbule Abla for being a great "big sister" who keeps supporting and encouraging me all these years, for your tolerance, and for all the amazing meals. I thank Abdurrahman Abi for rushing for me whenever I need help. And, of course, the kids, my little brothers and sisters... Attila, Kayra and Omay, thank you for cheering me up and always keeping the house alive.

Of course, I can never forget the friends, who have been a major source of joy and support. I would like to thank all my friends, specifically, my taicho Emre Kale, Süco, Ali (Şakar), Mustafa (Yasir), Ali (Cırık), Selda, Ömer, Marina, and my brother-in-arms Steve, who went through the hard times with me.

Last but not least, I would like to thank Andrea. In the time we spent together, you have brought enthusiasm, humor, energy, and love into my life. I have always felt your comforting presence, and your cheerful smile has always helped wipe away my worries. I would, specifically, like to express my gratitude for all the joyous times and undoubtedly for enduring me during the stressful and grueling times towards the end of my studies. I feel very lucky to have you in my life. To my grandfathers, Muzaffer Akbay and Ali Varol, whom I miss dearly.

#### ABSTRACT OF THE DISSERTATION

Improving the Performance of Partitioned Methods for Solid-Fluid Coupling

by

Muzaffer Akbay

Doctor of Philosophy, Graduate Program in Computer Science University of California, Riverside, September 2018 Dr. Tamar Shinar, Chairperson

Partitioned approaches, where the fluid and solid solvers are treated as black boxes with limited exposed interfaces, provide one convenient way of coupling solid and fluid solvers due their modularity and black-box design. However, they are often not preferred due to their stability and performance issues.

In this thesis, some of the crucial problems of partitioned approaches are addressed. A novel extended partitioned method for two-way solid-fluid coupling is presented, where the method achieves improved stability and extended range of applicability over standard partitioned approaches through three techniques. First, the black-box solvers are coupled through a small, reduced-order monolithic system, which is constructed on the fly from input/output pairs generated by the solid and fluid solvers. Second, a conservative, impulsebased interaction term is proposed to couple the solid and fluid rather than typical pressurebased forces. We show that both of these techniques significantly improve stability and reduce the number of iterations needed for convergence. Finally, a novel boundary pressure projection method is presented that allows for the partitioned simulation of a fully enclosed fluid coupled to a dynamic solid, a scenario that has been problematic for partitioned methods. The benefits of the methods are demonstrated by coupling Eulerian fluid solvers for smoke and water to Lagrangian solid solvers for volumetric and thin deformable and rigid objects in a variety of challenging scenarios. The application of the extended partitioned method to Smoothed Particle Hydrodynamics fluids requires some simple modifications to the system. These modifications are described and demonstrated on a publicly available solver. Finally, the boundary pressure projection method is revised further to handle solids with higher densities. The benefits of the revised boundary pressure projection method are demonstrated on various scenarios including multiple neighboring closed regions.

# Contents

Li	st of	Figures	xi
Li	ist of Tables xv		
1	Intr	oduction	1
<b>2</b>	An	Extended Partitioned Method For Conservative Solid-Fluid Coupling	<b>5</b>
	2.1	Introduction	5
	2.2	Related Work	7
	2.3	Reduced Model Interface	10
		2.3.1 Reduced monolithic system	11
		2.3.2 Constructing the reduced Jacobians	12
		2.3.3 Applying the RMI in the partitioned coupling	14
		2.3.4 Rigid Bodies	14
		2.3.5 Thin Shells	18
	2.4	Conservative Impulse-Based Coupling	19
	2.5	Boundary Pressure Projection	23
	2.6	Results	29
		2.6.1 Improved stability and convergence with RMI, conservative impulses,	
		and XPM	31
		2.6.2 Boundary pressure projection method	36
		2.6.3 Variety of solids coupled to an Eulerian smoke and water	37
	2.7	Conclusion	40
	-		-
3	$\mathbf{Sm}$	ooth Particle Hydrodynamics with XPM	<b>42</b>
	3.1	Introduction	42
	3.2	Related Work	44
	3.3	Smoothed Particle Hydrodynamics Overview	46
	3.4	XPM on SPH	51
	3.5	Results	54
	3.6	Conclusion	59

4	Bou	ndary	Pressure Projection for Incompressible Dirichlet Fluid Do	)-
	mai	$\mathbf{ns}$		60
	4.1	Introd	uction	60
	4.2	Equati	ons and Methods	64
		4.2.1	Fluid Equations	64
		4.2.2	Fluid Solver	64
		4.2.3	Structure Equations	65
		4.2.4	Numerical Method $\ldots$	66
	4.3	Examp	bles	73
		4.3.1	Supported Rigid Body	73
		4.3.2	Simple balloon-like problem	79
		4.3.3	Damped Structural Instability	83
		4.3.4	Inflating a Thin-shell Balloon	88
		4.3.5	Hydraulic Piston	92
		4.3.6	Inflating Two Symmetric Coupled Thin-shell Balloons	98
		4.3.7	Inflating Nested Thin-shell Balloons	106
		4.3.8	Multi-section Piston	110
	4.4	Conclu	ision	114
5	Con	clusior	ıs	115
Bi	bliog	raphy		117

# List of Figures

2.1	Schematic representation of the partitioned coupling of black-box fluid (F) and solid (S) solvers through the reduced model interface. Reduced Jacobians $\hat{F}_X$ and $\hat{S}_p$ are computed from input/output pairs of F and S and tightly coupled in a low-rank monolithic system to generate improved boundary	
	conditions to F and S	10
2.2	A light rigid plank interacts with a jet of smoke. (fluid grid size: $100 \times 150 \times 100$ )	16
2.3	Schematic of solid in a mixed dual cell of the fluid simulation grid	18
2.4	Center: A thin, transparent elastic balloon is stressed and filled by a fast	
	stream of water. (fluid grid size: $100 \times 125 \times 100$ , 4520 solid nodes)	20
2.5	Rigid balls of varying densities dropped into a pool of water exhibit correct	
	buoyancy. Their densities relative to the water are (back row to front row)	
	0.1, 0.5, 0.9, and 10. (fluid grid size: $225 \times 300 \times 150$ )	22
2.6	Inflating a balloon. Left: a schematic of the fluid simulation, with source faces	
	(blue), where air is injected, and solid faces (cyan). Center: a partitioned	
	solve with compatibility projection but without $p_0$ fails to inflate. Right: the	
	same simulation using our BPP method inflates properly.	24
2.7	A thin, elastic balloon is filled with a fast jet of smoke, and then released. The	
	balloon inflates properly due to the use of our BPP method. Computation	
	of this challenging scenario is stabilized by our extended partitioned method.	
	(fluid grid size: $100 \times 150 \times 100$ , 4520 solid nodes)	26
2.8	Pressure applied to a small piston of a hydraulic press is transmitted undi-	
	minished to a large piston through a fully enclosed, incompressible medium,	
	using our BPP method to determine the constant component of fluid pressure	
	$p_0$ . (fluid grid size: $160 \times 160 \times 80$ )	28
2.9	A light piece of deformable foam interacts with a jet of smoke. (fluid grid	
	size: $100 \times 150 \times 100$ )	30
2.1	0 2D cloth interacting with smoke	32
2.1	1 Number of iterations of the underrelaxed partitioned (P) and reduced model	
	interface with conservative impulses (XPM) over 540 substeps of two-dimensional	l
	simulations of a balloon filling with smoke. M = 0.8, $\omega$ = 0.2. The values	
	are averaged over a window of 10 substeps to reduce noise.	36

2.12	Comparison of partitioned without BPP (P), partitioned with BPP (P-BPP), XPM and monolithic (M) methods for a 2D balloon filling scenario with fully
2.13	enclosed fluid
$3.1 \\ 3.2$	Eulerian (left) and Lagrangian (right) representations of fluid
0.2	deficit in nearby particles. (Right) Correct density calculations with solid sample particles.
3.3	Top row: Simulations of a rigid ball coupled to an SPH fluid, with ball/fluid density ratio of 0.1. All images are at frame 26. Left: Weakly coupled solution computed with SPIisHSPlasH is stable. Right: Solution computed with XPM, with a tolerance of 10%, is also stable. Bottom row: The density ratio has been decreased to a more challenging value of 0.05. Left: Weakly coupled solution computed with SPIisHSPlasH is now unstable. Right: Solution
3.4	computed with XPM, with tolerance of 10%, is still stable XPM simulation of an SPH fluid coupled to a rigid sphere (M=0.05), cylinder (M=2), and torus (M=0.7), with 24389 SPH fluid particles. Top to bottom,
35	left to right: frame 17, 61, 105, and 360
0.0	rigid ball with density ratio 0.05, using the underrelaxed partitioned scheme (P) and the XPM with tolerance 1% and 10%. The XPM requires fewer iterations on average than the underrelaxed partitioned scheme. Increasing
	the tolerance further reduces iteration counts while stability is maintained. The values are averaged over a window of 10 substeps to reduce noise
4.1	The boundary pressure projection method demonstrated on a representative thin-shell example. Top: Before the BPP, the fluid velocities are incompatible as there is a net inflow from the source on the left. The BPP is used to project the fluid boundary velocities and to determine the constant pressure mode to be applied to the solid. Middle: Fluid boundary velocities after compatibility projection. Bottom: Solid velocities (red) after application of the constant
12	pressure mode
4.3	Supported rigid solid example $p - p_0$ values at t=10s
4.4	Supported rigid solid example. $p_0$ values throughout the simulation
4.5	Error in fluid forces, computed as $F'_s - Mg$ , where $F_s$ is the total pressure
	force applied on the rigid body by the fluid.
4.6	The log of the outflow volume rate, which is used as another error metric for
4 17	the supported rigid example.
4.7	Failed piston example with fluid momentum based formulation is depicted at various times t in the simulation
4.8	Setup of simple balloon-like problem with a volumetric solid
4.9	Volumetric simple balloon-like example. $p - p_0$ values at $t = 15$ .
	$\mathbf{r}$

4.10	Snapshots of the fluid domain velocity magnitudes for the volumetric simple	
	balloon-like example.	81
4.11	Volumetric simple balloon-like example. $p_0$ over time throughout the simu-	
	lation.	82
4.12	The log of the volume change rate for volumetric simple balloon-like problem.	83
4.13	Damped structural instability problem.	84
4.14	The fluid velocity field magnitude $\ \mathbf{u}\ $ at different stages of the simulation	
	for the damped structural instability problem.	85
4.15	The pressure field before $p_0$ is added: $p - p_0$ at time=4s in the simulation of	
	the damped structural instability simulation.	86
4.16	The change in $p_0$ over time in the damped structural instability example.	87
4.17	The log of the outflow volume rate for the damped structural instability	
	example	87
4.18	Setup of simple balloon-like problem with a thin-shell solid	88
4.19	The fluid pressure field variation $p - p_0$ , at the end of simulation, $t = 10s$ ,	
	is illustrated with a color map for the simple balloon-like example with a	
	thin-shell solid.	89
4.20	Snapshots at various stages of the simulation demonstrating the interaction	
	between the fluid and the thin-shell solid in the simple balloon-like example.	
	Color coding represents the fluid velocity field magnitude, $\ \mathbf{u}\ $	90
4.21	Constant pressure mode, $p_0$ , of the enclosed fluid domain plotted against	
	simulation time for the duration of the simple balloon-like example with	
	thin-shell solid.	91
4.22	The log of the volume flux rate as an error metric for the simple balloon-like	
	example with thin-shell solid.	92
4.23	Setup of the piston examples. Top: At equilibrium without any external	
	forces. Bottom: At equilibrium with an external downward force applied to	
	the right piston.	93
4.24	Color-mapped illustration of the pressure values of the fluid without the	
	constant pressure mode, $p - p_0$ for the hydraulic piston example. (Left)	
	Without external forces at the end of the simulation $(t = 10s)$ . (Right) With	0.4
4.05	external forces at the end of the simulation $(t = 90s)$	94
4.25	Constant pressure mode, $p_0$ , during the simulation of the hydraulic piston	05
4.96	example without external forces.	95
4.20	Constant pressure mode, $p_0$ , during the simulation of the hydraunc piston ex-	
	airpe with external forces, smoothed with a Gaussian method over a window	06
4.97	Size of 100 substeps.	90
4.21	force. Color coding represents the fluid velocity fold magnitude.	07
1 90	Coupled thin shall balloon axample setup	91
4.20	Color map of the fluid pressure field without the constant pressure mode	90
4.49	$n_0$ for the symmetric (top) and nonsymmetric (bottom) tests of two coupled	
	thin-shell balloons	ga
		00

4.30	The change in the solid position and velocity magnitude of the fluid domain is down and the form the simulation of two second this deal	
	is demonstrated in snapshots from the simulation of two symmetric thin-shell balloops	101
4 31	The change in the solid position and velocity magnitude of the fluid domain	101
1.01	is demonstrated with snapshots from the simulation of two nonsymmetric	
	thin-shell balloons.	102
4.32	Constant pressure mode, $p_0$ , throughout the simulation of the symmetric test	
	of two thin-shell balloons example. Region #1 is the left region, and Region	
	#2 is the right region. $\ldots$	103
4.33	Constant pressure mode, $p_0$ , throughout the simulation of the nonsymmetric	
	tests of two thin-shell balloons example. Region $\#1$ is the left region, and	
	Region $#2$ is the right region.	104
4.34	Log of the volume outflow rate for the symmetric (top) and the nonsymmetric	
	(bottom) examples. The values are smoothed with a Gaussian method with	
	a window size of 10	105
4.35	Nested balloon rings problem setup	106
4.36	Snapshots from the nested balloon at various states of the simulation. The	
	velocity magnitudes are illustrated with a color map	107
4.37	Plot of constant pressure modes over time for all regions in the nested thin-	
	shell balloons example. Region $\#1$ : outermost region, Region $\#2$ : middle	100
4.80	region, Region #3: innermost region.	109
4.38	Net volume outlow rate for all regions in the nested thin-shell balloons ex-	
	ample. Region #1: outermost region, Region #2: middle region, Region #3:	
	innermost region. Log function as well as Gaussian smoothing with size 10	100
1 20	Is applied for cleaner visualization.	109
4.39	The pressure field of the fluid without the constant pressure mode at the end	110
4.40	of the simulation for the multi-section piston example	111
1 11	Constant pressure mode $n_0$ plotted against time during the simulation of	111
7.71	the multi-section piston example Begion $\#1$ : bottom region Begion $\#2$ :	
	left region, Region #3: right region, $\dots$	112
4.42	The log of net volumetric outflow rate is used as an error metric for the	<b>-</b>
_	multi-section piston example. Region $#1$ : bottom region, Region $#2$ : left	
	region, Region #3: right region	113

## List of Tables

2.1	Average number of iterations per substep in the simulation of Figure 2.10,	
	with fewest iterations for each $M$ in blue bold. The solvers compared were	
	the underrelaxed partitioned (P) with various underrelaxation parameters,	
	$\omega$ , underrelaxed partitioned with conservative impulses (P-C), reduced model	
	interface (RMI), and reduced model interface with conservative impulses	
	(XPM). Our XPM required the fewest iterations and was able to handle	
	the most challenging mass ratios where the underrelaxed partitioned ap-	
	proach became unstable. Use of conservative impulses instead of pressure-	
	based forces improved stability overall for both the underrelaxed partitioned	
	method (P vs. P-C) and for the RMI (RMI vs. XPM)	33
2.2	Percentage improvement when using conservative impulse terms, with – in-	
	dicating both simulations were unstable, and $\infty$ indicating the unstable simu-	
	lation was stabilized by using the conservative scheme. For the underrelaxed	
	partitioned scheme, using the conservative impulses generally improved iter-	
	ation counts and allowed for more challenging mass ratios, though smaller	
	ratios were still unstable. Using the XPM, all simulations were stable and	
	iteration counts were improved up to 30% over the RMI.	35
2.3	The number of frames, average simulation time per substep, and average	
	number of substeps per frame are shown for several examples	39
0 1		

## Chapter 1

# Introduction

Fluids and solids interact with each other in various real-life scenarios such as sailing, where the air is interacting with sails, and the hull exchanges forces with the water to float and change direction, or blood flow in the heart, where the heart muscles and tissues interplay with blood, or even when a skirt is waved through the air. In order to reproduce such rich interactions, various physically-based simulation methods that capture both the internal and coupling dynamics are devised. These methods are categorized according to the level of interaction as one-way coupled if either only the fluid is affecting the solid or vice-versa. Methods that allow exchange of dynamics in both directions are called two-way coupled.

Physically-based simulation of two-way coupled fluids and solids has been an area of active research in recent years, both in computer graphics and computational physics, and many solution approaches have been investigated [27]. The techniques generally fall into two categories, monolithic and partitioned approaches. In monolithic approaches, equations for the solid and fluid are combined into one system and solved simultaneously, implicitly coupling internal solid and fluid forces to the interaction forces between them. On the other hand, partitioned approaches employ separate solvers for the solid and fluid and interleave calls to each solver to achieve either a weak coupling, or if iterated to convergence, a strong coupling of the solvers.

Both the monolithic and partitioned approaches have advantages and disadvantages in the simulation of two-way solid-fluid coupling. Monolithic approaches have the advantage of better stability, as stiff coupling terms can be resolved simultaneously with other forces in the system, but they also incur significant cost in terms of development and do not fully leverage existing codes. On the other hand, partitioned approaches have the advantage of being able to reuse previously developed fluid and solid solvers in a black-box fashion as long as they expose appropriate interfaces. However, they suffer from poorer stability properties, and in challenging scenarios, they may require many iterations per time step or worse, fail to converge [5]. When strong coupling is desired, a large number of iterations may be required for convergence, negatively impacting performance. Despite their drawbacks, partitioned approaches remain an important class of methods due to the appeal of reusing existing software, and recent work on partitioned approaches is aimed at improving their stability properties [20, 5, 25].

The systems in monolithic approaches include the constraints acting on both solids and fluids. Therefore, the constraints are always satisfied in the solution. On the other hand, these constraints are typically handled by the respective solvers in the partitioned approaches and are invisible to the other solver. However, when the input-only boundary conditions constructed by the state of the other solver do not satisfy these constraints, the internal solver fails since the system would not have any solutions.

In fluid dynamics, one of the most commonly applied constraints is the one that ensures that the flow is divergence-free. This constraint is required due to incompressible nature of the simulated fluid such as water or air/smoke. The problem arises in partitioned Dirichlet-Neumann coupling schemes, where the fluid provides pressures, or pressure forces, to the solid, and the fluid boundary velocities are determined by the solid velocities to avoid inter-penetration. In these schemes, the input velocity boundary conditions are read-only, and a problem emerges when the fluid is fully enclosed by a solid, since the solid may disregard the incompressibility constraint and provide divergent boundary velocities, where the total flow integrated over the closed region is not 0. This creates an unsolvable system in the fluid solver which leads to either a system crash or incorrect results. Partitioned methods that employ Dirichlet-Neumann coupling schemes are required to handle this case if they are targeted to work in scenarios with closed regions, such as simulations capturing blood flow or a hydraulic piston. Unfortunately, these scenarios are not always predictable, and can occur spontaneously, for example, water or air entrapped between different solids or air pockets formed when a piece of cloth folds over itself.

In this thesis, we provide solutions for the performance and stability issues of the partitioned approaches, and address the closed region problem with novel techniques. In the following sections, the proposed solutions and their applications are presented. In Chapter 2, an extended partitioned method (XPM), with the three methods that it consists of, is introduced and described. The method not only improves the performance and stability of the partitioned solvers, but also addresses the closed region problem. The benefits and application of the method on Eulerian-based fluid solvers are demonstrated in various results and scenarios. In chapter 3, the application of XPM on a Lagrangian solver is investigated. The modifications to the method, as well as its benefits, are demonstrated. Lastly, in Chapter 4, the boundary pressure projection (BPP) method is revised and improved by considering the solid dynamics in the formulation, and an algorithm that allows underrelaxation is presented. The method is supported with results from various scenarios.

## Chapter 2

# An Extended Partitioned Method For Conservative Solid-Fluid Coupling

#### 2.1 Introduction

In this section, we present a novel extended partitioned method (XPM) for twoway solid-fluid coupling of incompressible fluids to rigid and deformable solids and shells. While this approach does not outperform monolithic methods, which are generally more efficient, it does mitigate several drawbacks of partitioned schemes.

Our approach builds on a framework first proposed in [22] and [53, 55], which seeks strongly coupled solutions through a partitioned solve. Rather than simple interleaving calls to the fluid and solid solvers, potentially with underrelaxation, a monolithic reduced-order model of the system is used as a intermediate layer to generate improved boundary conditions for the individual fluid and solid solvers, thus accelerating convergence and improving stability. Whereas [22] used a reduced model based on a simplified physical problem, [55] constructs reduced models on the fly by computing local least-squares estimates of the solver Jacobians. In this study, we follow the basic framework introduced in [53, 55] for deformable bodies, extending it free surfaces, thin shells, and rigid bodies.

Furthermore, we propose the use of a conservative, impulse-based interaction term, first developed in the context of the monolithic approach [46], instead of the pressure-based forces typically used in partitioned approaches.

Finally, we propose a novel solution to the partitioned simulation of incompressible fluid regions fully enclosed by a dynamic solid, a scenario that has previously been problematic [31]. We demonstrate that these novel contributions give stability and performance benefits over more standard partitioned approaches and extend the scope of scenarios that can be effectively solved in a partitioned fashion.

In summary, our partitioned method for two-way coupling of incompressible fluids to solids includes the following novel contributions:

- Extending the reduced-model interface approach of [55] to free surfaces, thin shells, and rigid bodies, demonstrating the benefits of this general approach on a variety of solid-fluid coupling scenarios.
- Proposing the use of a conservative, impulse-based interaction term rather than the standard pressure-based forces typically used in partitioned approaches, resulting in accelerated convergence.

• Proposing a novel boundary pressure projection method for solving incompressible fluid regions fully enclosed by dynamic solids, extending the range of scenarios that can be simulated using a partitioned scheme.

#### 2.2 Related Work

Computer graphics researchers have developed a variety of approaches to coupling Eulerian fluids with deformable and rigid Lagrangian solids. Monolithic approaches include a non-symmetric linear system to capture the two-way interactions between fluids and deformable solids [14], a symmetric system capturing the interaction of smoke and rigid bodies [30], a variational formulation based on kinetic energy minimization for coupling fluids to rigid bodies [7], a symmetric, momentum-conserving method for coupling fluids to volumetric and thin deformable and rigid bodies [46], a positive-definite formulation of that system [45], two-way coupling of fluids to reduced deformable bodies [36], and a cutcell method which implicitly couples fluid pressure with solid elasticity and damping [57]. Implicit coupling of solids and fluids was also achieved through a fully Eulerian treatment of both [51].

Partitioned approaches have also received considerable attention [27, 17]. One common method is that the solid solver provides velocity boundary conditions to the fluid solver, and the fluid solver provides pressure-based forces to the solid solver (also referred to as a Dirichlet-Neumann decomposition). Separate calls to the fluid and solid solvers are interleaved, treating each as a black box, to achieve either a weak coupling, or if iterated to convergence, a strong coupling of the solvers. A partitioned approach to coupling fluid and rigid bodies was proposed in [12], where additional forces are applied to the fluid velocity to enforce a rigid motion. In [23], thin deformable and rigid shells were weakly coupled to smoke and water in an interleaved fashion.

Partitioned approaches may suffer from stability issues under challenging scenarios such as high mass ratio of fluid to solid [13]. In the simplest interleaved approach, one might try to reduce the time step size to achieve better stability. However, it has been shown that the stability condition on the time step is not always achievable for certain problem parameters [33, 54]. Several methods attempt to mitigate the stability problems of partitioned solvers, for example, by incorporating relaxation into the fixed-point iteration [32]. When Gauss-Seidel subiterations with underrelaxation are used, a very small underrelaxation factor may be needed for convergence [54].

Another approach to improving stability of partitioned schemes is the use of a reduced monolithic model as an intermediary between the solvers [22, 55, 20, 17]. We build on the method introduced in [53, 55], where local, linear reduced-order models of the solid and fluid solvers are constructed on the fly using the input/output pairs collected from the solver calls. The reduced order models are then solved in a monolithic fashion to generate boundary conditions for each black-box solver, improving stability and reducing iterations counts. We note that this approach is closely related to quasi-Newton methods for partitioned solid-fluid coupling [18, 20, 17, 25], in which the black-box solver Jacobians are similarly approximated, and which have also been shown to improve performance of partitioned schemes. A variety of reduced-order models have been developed for physically-based animation, for example [52] for reduced fluid simulation, and [6] for reduced deformable solid

simulation. In contrast to these and related approaches, our approach uses the reduced models to stabilize and accelerate convergence of the iterated partitioned coupling, while the final fluid and solid state in each time step is determined by the full solvers.

Several authors have noted a limitation in handling fully enclosed fluid regions in partitioned or interleaved approaches for coupling incompressible flows to structures [7, 17]. Since the fluid cannot compress or expand, there should be no net flow through the boundary of any fluid region. In a typical partitioned solve, however, the solid solver is not aware of this constraint, and will supply the fluid solver with boundary velocities that violate it in the case of a fully enclosed fluid region. Internal to the fluid solver, this constraint manifests as a pressure Poisson equation with only Neumann boundary conditions, and hence with a nontrivial null-space, and constraint-violating velocities from the solid lead to an incompatible linear system for pressure. These "Neumann regions" occur in practice, sometimes spontaneously, and must be detected and handled for robustness.

In [23], a partial solution was implemented by projecting the solid velocities to be compatible. A related issue is that the fluid pressure inside the enclosed region is determined only up to an unknown constant  $p_0$ . Since the fluid is accelerated by the gradient of pressure,  $p_0$  is not needed for the fluid simulation. However, at the enclosed region boundary, the solid should be affected by  $p_0$ , and if it is not properly accounted for, the forces on the solid will be incorrect.



Figure 2.1: Schematic representation of the partitioned coupling of black-box fluid (F) and solid (S) solvers through the reduced model interface. Reduced Jacobians  $\hat{F}_X$  and  $\hat{S}_p$  are computed from input/output pairs of F and S and tightly coupled in a low-rank monolithic system to generate improved boundary conditions to F and S.

#### 2.3 Reduced Model Interface

Rather than simple Gauss-Seidel iterations between the fluid and solid solvers, we couple them through the Reduced Model Interface (RMI) (Figure 2.1). In this approach, the boundary conditions passed to the individual solvers are generated by an intermediate solver, a monolithic system coupling reduced models of the solid and fluid. Like other partitioned methods, this allows for the use of the fluid and solid solvers as black boxes. Input/output pairs collected from invocations of the fluid and solid solvers are used to approximate their Jacobians and build the monolithic reduced system. The RMI system is smaller than a full monolithic system, as it contains no more than the number of interface variables. Although dense, as opposed to the sparse monolithic discretizations such as [46], it is also low-rank, typically capturing a small number of modes, and this is exploited in the

matrix computations to avoid forming dense matrices the size of the number of interface variables. Stability analysis of RMI coupling for unsteady flow in an elastic tube showed that unstable components appear during the first iterations and are implicitly coupled by the reduced model interface [19].

#### 2.3.1 Reduced monolithic system

We follow the approach of [53, 55] for coupling a fluid and volumetric deformable solid and extend it to rigid bodies, free surfaces, and thin shells. The black-box fluid solver is represented as

$$p = F(X, V(X)), \tag{2.1}$$

taking as input the solid node positions and effective velocities and returning the fluid pressures on the solid. The black-box solid solver is represented as

$$X = S(p), \tag{2.2}$$

taking as input fluid pressures and returning the solid simulation state.

The approach is illustrated in Figure 2.1. We seek to construct a reduced-order model of the fluid and solid solvers, the RMI, to act as intermediary, generating improved boundary condition inputs to the black-box solvers. Since the RMI is reduced-order, it can be efficiently solved in a monolithic fashion. Its cost is much less than that of the individual fluid and solid solvers, as shown in Section 2.6.

We next describe the RMI. During each time step, at iteration i, a call to the fluid solver (2.1) yields an input/output pair  $(X_i^f, p_i^f)$ , and a call to the solid solver (2.2) yields an output/input pair  $(X_i^s, p_i^s)$ . After k iterations, the linear, reduced-order models of the fluid and solid solvers are given by

$$\hat{p} = p_k^f + \hat{F}_X(\hat{X} - X_k^f),$$
(2.3)

$$\hat{X} = X_k^s + \hat{S}_p(\hat{p} - p_k^s),$$
(2.4)

respectively, where  $\hat{F}_X$  is an approximation to the Jacobian of F with respect to X, and  $\hat{S}_p$  is an approximation to the Jacobian of S with respect to p, and  $(\hat{X}, \hat{p})$  are unknowns. Moving unknown terms to the left hand side, this becomes a monolithic set of equations

$$\begin{pmatrix} I & -\hat{F}_X \\ -\hat{S}_p & I \end{pmatrix} \begin{pmatrix} \hat{p} \\ \hat{X} \end{pmatrix} = \begin{pmatrix} p_k^f - \hat{F}_X X_k^f \\ X_k^s - \hat{S}_p p_k^s \end{pmatrix}.$$
 (2.5)

As only one of  $\hat{p}$  or  $\hat{X}$  is needed when invoking the solid or fluid solver, they may be obtained individually by applying block Gaussian elimination to Equation (2.5) as

$$\hat{X} = \left(I - \hat{S}_p \hat{F}_X\right)^{-1} \left[X_k^s + \hat{S}_p \left(p_k^f - p_k^s - \hat{F}_X X_k^f\right)\right],$$
(2.6)

$$\hat{p} = \left(I - \hat{F}_X \hat{S}_p\right)^{-1} \left[p_k^f + \hat{F}_X \left(X_k^s - X_k^f - \hat{S}_p p_k^s\right)\right].$$
(2.7)

#### 2.3.2 Constructing the reduced Jacobians

Next, we describe the computation of the reduced model Jacobians  $\hat{F}_X$  and  $\hat{S}_p$ . They are constructed from scratch at each iteration, using data from all calls to the fluid or solid solver in the current time step. The data pairs  $(X_i^f, p_i^f)$  and  $(X_i^s, p_i^s), i = 1, ..., k$ are collected from the fluid and solid solvers, respectively, over k iterations. Note that the RMI layer translates between the disparate discretizations of the solvers. In our examples, interface variables are located at solid nodes, and fluid variables are mapped to/from the nodes through interpolation and/or conservative distribution operators. From the k fluid solver data pairs, we construct a position difference matrix  $V = (\Delta x_1^f \dots \Delta x_{k-1}^f)$ , where  $\Delta x_j^f = X_k^f - X_j^f$ , and a pressure difference matrix  $P = (\Delta p_1^f \dots \Delta p_{k-1}^f)$  where  $\Delta p_j^f = p_k^f - p_j^f$ . A displacement  $\Delta X$  is approximated by the vectors in V by solving the least squares problem  $\min_{\alpha} ||\Delta X - V\alpha||_2$ . Taking the QR factorization of V,

$$V = QR, \quad R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}, \quad Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix},$$

it follows that

$$\alpha = R_1^{-1} Q_1^T \Delta X,$$

and setting the pressure differences consistently with the displacements as  $\Delta p = P\alpha$ , we get

$$\Delta p = P R_1^{-1} Q_1^T \Delta X.$$

The Jacobian  $\hat{F}_X$  is finally determined as

$$\hat{F}_X = \frac{\partial p}{\partial x} = \frac{\partial p}{\partial \alpha} \frac{\partial \alpha}{\partial x} = P R_1^{-1} Q_1^T.$$

In the same manner, the approximate Jacobian of the solid solver  $\hat{S}_p$  can be computed. Input/output pairs gathered from the solid solver are used to compute difference matrices  $\bar{V}$  and  $\bar{P}$ . Using a QR factorization  $\bar{P} = \bar{Q}\bar{R}$ , with  $\bar{R}_1$  and  $\bar{Q}_1$  analogous to  $R_1$ and  $Q_1$ , respectively, and the same procedure as for  $\hat{F}_X$  above, we get

$$\hat{S}_p = \bar{V}\bar{R}_1^{-1}\bar{Q}_1^T.$$

The least squares systems described above can become arbitrarily ill-conditioned as more basis vectors are incorporated. To stably compute the near rank-deficient least squares problem, we use the Eigen library's [24] QR decomposition with full pivoting.

#### 2.3.3 Applying the RMI in the partitioned coupling

Equations (2.6) and (2.7) are solved in a matrix-free manner using GMRES, and the expressions of the form  $R^{-1}Q^T x$  are evaluated through backward substitution rather than formed explicitly. Therefore, although for m interface nodes,  $\hat{F}_X$  is  $m \times 3m$  and  $\hat{S}_p$  is  $3m \times m$ , the algorithm can exploit their low rank (no more than k-1) in the computations to achieve efficiency, and the cost of solving these systems is a small fraction of the overall simulation time (0.2% - 4.2% in our examples).

Equation (2.6) is solved before each call to the fluid solver, while Equation (2.7) is solved before each call to the solid solver, in order to provide as input the RMI prediction of the converged state to the individual solvers. Effective velocities are computed as  $V_{k+1}^{f} = (X_{k+1}^{f} - X_{0})/\Delta t \text{ every time before calling the fluid solver. The overall procedure is summarized in Algorithm 1.$ 

#### 2.3.4 Rigid Bodies

To handle rigid bodies, the RMI samples them with embedded points, whose positions and velocities are determined from the rigid body state. The embedded points are spaced at a distance 1-1.5dx apart to be commensurate with the grid. Fluid interaction forces at the embedded points are mapped to a force and torque and passed to the rigid

### **ALGORITHM 1:** Single substep of the extended partitioned method, integrating from time

	$t^n$ to time $t^{n+1}$
1	extrapolate solid positions to get $X_1^f$
2	$p_1^s = p_1^f = F(X_1^f, (X_1^f - X^n) / \Delta t)$
3	$X_2^f = X_1^s = S(p_1^s)$
4	$p_2^s = p_2^f = F(X_2^f, (X_2^f - X^n)/\Delta t)$
5	$X_2^s = S(p_2^s)$
6	$R = X_2^s - X_1^s$
7	Construct $\hat{F}_X$ with 1 mode
8	k = 1
9	while $  R   > tolerance do$
10	Construct $\hat{S}_p$ with k modes
11	Solve Eq. (2.6) to get $X_{k+1}^f$
12	$p_{k+1}^f = F(X_{k+1}^f, (X_{k+1}^f - X^n) / \Delta t)$
13	Construct $\hat{F}_X$ with $k + 1$ modes
14	Solve Eq. (2.7) to get $p_{k+1}^s$
15	$X_{k+1}^{s} = S(p_{k+1}^{s})$
16	$R = X_{k+1}^s - X_k^s$
17	k = k + 1
18	end while



Figure 2.2: A light rigid plank interacts with a jet of smoke. (fluid grid size:  $100 \times 150 \times 100$ )

body solver. The process described above for solid nodes is then used without modification with the embedded points.

Reduced model interface formulation for rigid bodies: Alternatively, one could formulate the reduced model directly on the rigid body degrees of freedom. Following the work in [42], a quaternion, q, can be represented as a displacement vector, w, around a reference rotation,  $q_*$ , using logarithmic and exponential maps with the following conversion operations:  $w = log(q_*^{-1}q)$  and  $q = q_* \exp(w)$ . By selecting  $q_* = q_k$  We can now rewrite the rigid body coordinates as  $B = [t, w]^T$  where  $w = log(q_k - 1q)$ , and displacement vector, U, as  $U = u_m, m = 1 \dots k$  where  $u_m = B_k - B_m$ , noting the displacement vector for  $q_k$  is 0.

By selecting  $q_* = q_k$ , we can assume  $q_k$  is close to all  $q_m$ 's and is in the same hemi-

sphere, avoiding a search, proposed in [42], for a reference point that satisfies this condition

We factorize U as above by including a preconditioning matrix that balances the scales of t and w:  $M\Delta B = MU\alpha$  where  $M = \begin{pmatrix} I & 0 \\ 0 & rI \end{pmatrix}$  and r is the average radius of the rigid body scaling the unit sphere the quaternion operates on.

We can solve this least squares problem similar to before by using the QR factorization of MU, to find the column contributions  $\alpha$ :

$$MU = QR, \quad R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}, \quad Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix},$$

$$\alpha = R_1^{-1} Q_1^T M \Delta B,$$

and setting the pressure differences consistently with the displacements,

$$\Delta p = P\alpha.$$

The Jacobian  $\hat{F}_B$  is finally determined as

$$\hat{F}_B = \frac{\partial p}{\partial b} = \frac{\partial p}{\partial \alpha} \frac{\partial \alpha}{\partial b} = P R_1^{-1} Q_1^T M$$

The equation for  $\hat{S}_p$  in the original is changed slightly replacing  $\bar{V}$  with  $\bar{U}$ :  $\hat{S}_p = \bar{U}\bar{R}_1^{-1}\bar{Q}_1^T$ . Equations 2.5, 2.6 and 2.7 are followed exactly with new *B* representation, however, an additional step is required after equation 2.6 to convert the vector representation, w, in *B* to final orientation of the rigid body:  $q = q_k \exp(w)$ .



Figure 2.3: Schematic of solid in a mixed dual cell of the fluid simulation grid.

#### 2.3.5 Thin Shells

Similar to [23], we compute the pressure difference in the direction of the face normal,  $\Delta p_f \mathbf{n}_f$ , on each fluid grid face f that is marked as a solid boundary. For a solid node i, the pressure difference from face f is weighted by  $\mathcal{K}_{if} = \mathcal{K}(\mathbf{x}_i - \mathbf{x}_f)$ , a distance-based kernel function. The final pressure-based force on node i is  $f_i = \sum_f w_{if} \Delta p_f \mathbf{n}_f \cdot \mathbf{n}_i A_i$  where  $w_{if} = \mathcal{K}_{if} / \sum_{f'} \mathcal{K}_{if'}$  is normalized by the sum of weights of all faces influencing node i.

These nodal force components are used in place of the values  $p_i$  in Section 2.3.2. For volumetric solids, we use the same formulation with pressures inside the solid set to zero.

#### 2.4 Conservative Impulse-Based Coupling

The conservative monolithic coupling scheme of [46] identified the impulse,  $I_{DC}$ , implicitly applied in a mixed dual cell of the fluid simulation MAC grid in order to satisfy the no-slip boundary condition I. We review that here.

Consider a dual cell about a sample of an x-component of the velocity field, u, as shown in Figure 2.3. Assume that forces other than pressure and interaction with the solid at the interface  $\Gamma$  have been integrated to get an intermediate velocity  $u^*$ , as in a typical MAC grid-based Eulerian fluid simulation. Given pressure samples  $p_L$  and  $p_R$  along the left and right faces of the cell with areas  $A_L$  and  $A_R$ , respectively, the change in x-momentum due to pressure and interaction with the solid can be written as

$$M_f u^{n+1} - M_f u^* = p_L A_L - p_R A_R + I_{DC}, (2.8)$$

where  $I_{DC} \approx \int_{\Gamma} pn_x dS$ , and  $n_x$  is the x-component of the outward unit normal to the solid. As in [46], rather than explicitly computing  $-I_{DC}$  as an approximate boundary integral, and applying that to the solid, we infer its value from Equation (2.8). Since  $I_{DC}$  is the impulse applied on the fluid by the solid, the impulse applied on the solid by the fluid must be  $-I_{DC}$  in order to satisfy the dynamic boundary condition II. We propose to apply this impulse in the partitioned approach instead of explicitly approximating  $\int_{\Gamma} pn_x dS$ .

This has several potential benefits. First, this formulation is conservative, as the fluid/solid interaction is treated through a momentum exchange. Second, in [46], this formulation led to an "added mass" term in the solid momentum equation, which contributed to the improved stability properties of that semi-implicit scheme. The added



Figure 2.4: Center: A thin, transparent elastic balloon is stressed and filled by a fast stream of water. (fluid grid size:  $100 \times 125 \times 100$ , 4520 solid nodes)
mass effect, whereby the effect of the fluid on the solid can be modeled in part as an increase in solid mass, has been shown to have a destabilizing effect on fully explicit schemes [13]. While the added mass term is treated explicitly in the case of our partitioned solve, we also observe improved stability with the impulse-based formulation, which departs from the standard Dirichlet-Neumann scheme. Finally, as in [46], the impulse-based formulation applies directly to thin shells as well as volumetric solids, with  $M_f = V\rho$  and  $A_R = A_L = A$ , where V and A are the volume and face area, respectively, of the dual cell. As in [46], we take  $A_R = A_L = A$  in our implementation.

In order to apply the conservative formulation in a partitioned fashion, the fluid solver must supply as output the impulse  $-I_{DC}$  rather than an explicitly computed approximation of the boundary forces. This is a minor modification for fluid solvers that compute pressures through a pressure projection step, such as fractional step fluid solvers [15], or some recent SPH solvers [28, 44]. For other solvers, one would have to estimate the constraint force applied to enforce the solid boundary condition. We do not investigate this further in the present work.

When using the conservative formulation as part of the RMI, we replace the pressure values  $p_i$  in Section 2.3.2 with the impulses  $I_{DC}$ . Furthermore, rather than interpolating to solid nodes, the impulses are distributed in a conservative fashion to solid nodes as in [46]. We demonstrate in our examples that use of the conservative formulation leads to improved stability and lower iteration counts as compared with the non-conservative formulation, for both the partitioned scheme with underrelaxation, and the partitioned scheme with the RMI (see Table 2.2).



Figure 2.5: Rigid balls of varying densities dropped into a pool of water exhibit correct buoyancy. Their densities relative to the water are (back row to front row) 0.1, 0.5, 0.9, and 10. (fluid grid size:  $225 \times 300 \times 150$ )

## 2.5 Boundary Pressure Projection

We propose a novel boundary pressure projection (BPP) method, that to our knowledge has not been proposed elsewhere. In particular, we show that  $p_0$  is the Lagrange multiplier associated with the boundary velocity constraint, and show how it can be determined and incorporated into a partitioned solid/fluid coupling approach. Like [23], our solution involves a projection of the solid velocities to be compatible for the pressure solve. Our derivation is more general and reduces to the form they used if solids are voxelized to the grid and other boundary source terms are neglected. Additionally, unlike that work, we also solve for the unknown constant component of pressure  $p_0$  and apply this pressure to the solid. This causes the solid to evolve toward constraint-satisfying velocities.

Figure 2.6 demonstrates the partitioned approach with and without determining and applying  $p_0$  to the solid. The simulation contains a Neumann fluid region, a region fully enclosed by faces with velocities determined either from sources or the dynamic solid. Note that without  $p_0$ , the balloon fails to inflate, whereas with  $p_0$  determined through our BPP, the balloon inflates similarly to our ground truth monolithic simulation.

Let  $\Gamma$  be the boundary of a closed region in the fluid domain, for example, the surface of the balloon and air source as pictured in Figure 2.6. The divergence free constraint on the fluid implies that

$$\int_{\Gamma} \mathbf{u} \cdot \mathbf{n} dS = 0.$$

This can be discretized on the MAC grid as

$$\sum_{f} A_{f} u_{f} n_{f} = 0,$$



Figure 2.6: Inflating a balloon. Left: a schematic of the fluid simulation, with source faces (blue), where air is injected, and solid faces (cyan). Center: a partitioned solve with compatibility projection but without  $p_0$  fails to inflate. Right: the same simulation using our BPP method inflates properly.

where  $A_f$  is the face area,  $u_f$  is the velocity component on that face, and  $n_f$  is the closed region normal component on that face (±1). We write the velocities, area-weighted normal components, and dual cell masses associated with the boundary faces as U = $(u_1, u_2, \ldots, u_n)^T$ ,  $C = (A_1 n_1, A_2 n_2, \ldots, A_n n_n)^T$ , and  $M = \text{diag}(M_1, M_2, \ldots, M_n)$ . The constraint equation is

$$C^T U = 0, (2.9)$$

and the Jacobian of the constraint is given by  $C^T$ . We seek an unknown pressure  $p_0$  that when applied to the fluid in the enclosed region causes the fluid to satisfy the constraint (2.9):

$$M\hat{U} = MU^* + \triangle tCp_0. \tag{2.10}$$

Note that the pressure  $p_0$  is the unknown Lagrange multiplier for the constraint force  $Cp_0$ . See [56] for a general description of constrained particle dynamics and Lagrange multipliers. Multiplying both sides by  $M^{-1}$ , applying the constraint (2.9), and solving for  $p_0$ , we get

$$p_0 = -\frac{1}{\Delta t} \left( C^T M^{-1} C \right)^{-1} C^T U^*.$$
(2.11)

Plugging this result back into Equation (2.10) and rearranging gives the projection

$$\hat{U} = \mathbb{P}U^* \tag{2.12}$$

$$\mathbb{P} = I - M^{-1} C (C^T M^{-1} C)^{-1} C^T$$
(2.13)

If  $M_f = \rho V$  in all dual cells and  $A_f = A$  is a constant for all faces, then the projection operator in Equation (2.13) simplifies to  $(I - \frac{1}{n}NN^T)$ , where *n* is the number of boundary faces, and *N* is the vector of normal components, i.e., C = AN. This is the projection used in [23] to ensure compatible velocities for the pressure solve, though they did not calculate  $p_0$  to apply to the solid.

Note that some of the closed region boundary faces may be sources or kinematic objects whose velocities are not determined by the dynamic solid. In that case, those should be isolated from the other faces in (2.9), and the constraint force and projection should not be applied to those faces in (2.10) and (2.13). By partitioning the velocity vector as  $U = (U_s^T | U_k^T)^T$ , and the constraint equation as  $C = (C_s^T | C_k^T)^T$ , where the subscripts s, k represent solid and source/kinematic components, respectively, Equations (2.9)-(2.13) become



Figure 2.7: A thin, elastic balloon is filled with a fast jet of smoke, and then released. The balloon inflates properly due to the use of our BPP method. Computation of this challenging scenario is stabilized by our extended partitioned method. (fluid grid size:  $100 \times 150 \times 100$ , 4520 solid nodes)

$$C_{s}^{T}U_{s} = -C_{k}^{T}U_{k}$$

$$M_{s}\hat{U}_{s} = M_{s}U_{s}^{*} + \triangle tC_{s}p_{0}$$

$$p_{0} = -\frac{1}{\triangle t} \left(C_{s}^{T}M_{s}^{-1}C_{s}\right)^{-1} \left(C_{s}^{T}U_{s}^{*} + C_{k}^{T}U_{k}\right)$$

$$\hat{U} = \mathbb{P}U^{*} - M_{s}^{-1}C_{s}(C_{s}^{T}M_{s}^{-1}C_{s})^{-1}C_{k}^{T}U_{k}$$

$$\mathbb{P} = I - M_{s}^{-1}C_{s}(C_{s}^{T}M_{s}^{-1}C_{s})^{-1}C_{s}^{T}$$

The BPP method also applies directly to multiple coupled Neumann regions, as may occur, for example, with cloth folding over itself. In this case, m constraints lead to m rows in  $C^T$ ,  $C^T M^{-1}C$  is an  $m \times m$  matrix rather than a scalar, and  $p_0$  is an m-vector containing the unknown constant pressures for all of the enclosed regions. The procedure for finding the vector  $p_0$  is then identical to that described above.

[31] also addressed the problem of partitioned coupling of structures to incompressible fluids. They proposed several possible solutions: to enforce the velocity constraint directly on the solid inside the solid solver, while the fluid solver determines a displacement (coupling the fluid and solid in a Neumann-Dirichlet fashion for that one mode), or alternatively to relax the incompressibility constraint. Our solution instead projects the solid velocities based on the state of the fluid, and supplies the fluid with the compatible velocities, and the solid with the pressure  $p_0$  needed to project the velocities, maintaining the Dirichlet-Neumann structure of the partitioning. Hence, in our approach, as for the other fluid pressures, the determination of  $p_0$  is not implicitly coupled to other solid forces, and does not require a constraint to be imposed on the solid. When strong coupling is



Figure 2.8: Pressure applied to a small piston of a hydraulic press is transmitted undiminished to a large piston through a fully enclosed, incompressible medium, using our BPP method to determine the constant component of fluid pressure  $p_0$ . (fluid grid size:  $160 \times 160 \times 80$ )

achieved, the solid velocities on the enclosed fluid region boundary converge to satisfy the incompressibility constraint.

To incorporate the BPP, each call  $F(X, (X - X^n)/\Delta t)$  in Algorithm 1 is replaced by a call to Solve\_Fluid(X), shown in Algorithm 2. We demonstrate the BPP method by simulating a balloon being filled with smoke (Figure 2.7). The incompressible fluid inside the balloon is fully enclosed by the balloon surface and smoke source, and using the BPP it inflates properly. We further demonstrate the BPP method by simulating a hydraulic press, where a fully enclosed incompressible medium is used to transmit pressures between two pistons (Figure 2.8). In both cases, determining  $p_0$  and applying it to the solid is essential to capturing the dynamics of the system.

## ALGORITHM 2: Wrapper for the fluid solver which applies the boundary pressure projection

1 
$$p =$$
function Solve\_Fluid(X)

2 Compute effective velocities 
$$V = (X - X^n)/\Delta t$$

$$\mathbf{s} \qquad (V, p_0) = \mathrm{BPP}(V)$$

- $q \qquad p = F(X, V) + p_0$
- 5 end function

## 2.6 Results

Here, we demonstrate individually the performance of the RMI, the conservative impulses, the BPP, and the combination of these techniques, the XPM. In general, monolithic approaches are expected to be more efficient in computing a strongly coupled solution. The benefit of the partitioned approach lies not in fast computation time, but in the ability to reuse existing software with little modification. Therefore, we do not expect to outperform the monolithic solver, but aim instead to improve performance of the partitioned approach.



The cost of the RMI layer is relatively small compared with the cost of the fluid

Figure 2.9: A light piece of deformable foam interacts with a jet of smoke. (fluid grid size:  $100 \times 150 \times 100$ )

and solid solvers. Moreover, since we are using existing black-box fluid and solid solvers, our primary measure of performance gains due to our approach is in the reduction of iteration counts.

We computed our examples with MPI parallelism on a cluster of 16 nodes, each comprised of 2 AMD Opteron 6272 CPUs totaling 32 cores, 64GB ECC memory, and 40Gbps InfiniBand networks. In all of our simulations, we used the smaller of the time step sizes computed by the individual fluid and solid solvers, and did not impose additional stability conditions on the time step due to the coupling. We limited the number of reduced model subiterations to 30, although the typical number of subiterations was much less. We used a convergence tolerance of 5% of the fluid grid cell size.

# 2.6.1 Improved stability and convergence with RMI, conservative impulses, and XPM

In this section, we study the benefits of using the RMI, the conservative impulses, and both (the XPM), on a two-dimensional example of cloth interacting with a jet of smoke with source velocity U = 0.5, shown in Figure 2.10. We used the PhysBAM cloth simulator and PhysBAM smoke simulator. Though PhysBAM supports monolithic coupling [46], we treated the solvers as black boxes and coupled them in a partitioned fashion.

We compare the performance of our method with that of a partitioned solver with underrelaxation, shown in Algorithm 3, for different values of the underrelaxation parameter,  $\omega$ . For smaller values of  $\omega$ , we expect better stability at the expense of a greater number of iterations. Table 2.1 shows the average number of iterations for the solves, with a dash indicating the simulation became unstable before completion. M is the mass ratio of the balloon to the air, and problem difficulty increases with decreasing M. We found that for  $\omega \geq 0.5$ , the simple partitioned approach failed to converge from the outset for all values of M. As M was decreased, the partitioned approach required increasingly stringent underrelaxation for convergence, which led to a greater number of subiterations.

When partitioning with underrelaxation is used, performance can be improved by using Aitken relaxation, which dynamically determines the underrelaxation parameter [32]. Though we do not compare with that approach here, the RMI has been demonstrated to outperform Aitken relaxation as well [55].



Figure 2.10: 2D cloth interacting with smoke.

M	Р	Р	Р	Р	Р	P-C	RMI	XPM
	$\omega \ge .5$	$\omega = .2$	$\omega = .1$	$\omega = .05$	$\omega = .01$	$\omega = .05$	$\omega = 1$	$\omega = 1$
.1	-	-	-	-	28.2	5.5	5.5	3.8
.2	-	-	-	5.7	27.3	4.5	4.0	3.2
.5	-	-	2.6	4.8	23.8	4.6	3.0	2.6
.8	-	3.4	2.6	5.3	10.2	5.0	2.7	2.1
1	-	2.2	2.8	5.6	9.1	4.7	2.6	2.0
2	-	2.0	2.9	5.1	5.2	3.2	2.0	2.0

Table 2.1: Average number of iterations per substep in the simulation of Figure 2.10, with fewest iterations for each M in blue bold. The solvers compared were the underrelaxed partitioned (P) with various underrelaxation parameters,  $\omega$ , underrelaxed partitioned with conservative impulses (P-C), reduced model interface (RMI), and reduced model interface with conservative impulses (XPM). Our XPM required the fewest iterations and was able to handle the most challenging mass ratios where the underrelaxed partitioned approach became unstable. Use of conservative impulses instead of pressure-based forces improved stability overall for both the underrelaxed partitioned method (P vs. P-C) and for the RMI (RMI vs. XPM).

## **ALGORITHM 3:** Partitioned solver with underrelaxation

1 k = 1

L

<sup>2</sup> while ||R|| > tolerance do

$$\mathbf{3} \qquad X^{k+1} = S(p^k)$$

4 
$$X^{k+1} = \omega X^{k+1} + (1-\omega)X^k$$

5  $p^{k+1} = F(X^{k+1}, (X^{k+1} - X^n)/\Delta t)$ 

$$\mathbf{6} \qquad R = X^{k+1} - X$$

$$\mathbf{7} \quad k = k+1$$

8 end while

## Effect of RMI

Table 2.1 compares the partitioned approach (P columns) with the RMI (RMI column). Specifically, we compare the RMI with the partitioned method with the largest value of  $\omega$  that converged. The RMI outperformed the underrelaxed partitioned approach in 3 out of 6 cases, tied it in 1 case, and performed slightly worse in 2 cases. For M = .1, the RMI converged, whereas the underrelaxed partitioned method with  $\omega \geq 0.05$  did not. Therefore, the RMI appears to improve the stability of the partitioned approach over underrelaxation alone.

#### Effect of conservative impulses

The conservative impulse-based formulation is orthogonal to the RMI and can be used with the underrelaxed partitioned scheme. To illustrate the effect of using conservative impulses instead of the explicit pressure-based forces, Table 2.1 compares the partitioned approach with pressured based forces (column P,  $\omega = .05$ ) and with the conservative impulses (column P-C,  $\omega = .05$ ). In every case, using the conservative impulses reduced the number of iterations required. Furthermore, for M = .1,  $\omega \ge .05$ , the partitioned approach with pressure-based forces was unstable, whereas the partitioned approach with impulsebased forces was stable. For this example, RMI and XPM also differ only in the use of the conservative impulses. Comparison of Table 2.1 columns RMI and XPM shows that the use of conservative impulses reduced average iteration counts by up to 30%.

Tables 2.1 and 2.2 show only average performance for a problem that is initially easy. The significant benefits of the XPM are better illustrated in Figure 2.11, which

М		P vs	RMI vs.		
	$\omega = .05$	$\omega = .1$	$\omega = .2$	$\omega = .5, .8$	XPM (RMI-C)
.1	$\infty$	-	-	-	29.9
.2	21.3	$\infty$	-	-	21.4
.5	4.0	-13.3	$\infty$	-	12.4
.8	6.2	8.2	15.4	-	22.0
1	16.3	25.1	-18.1	-	21.6
2	38.4	30.0	0.0	$\infty$	0.3

Table 2.2: Percentage improvement when using conservative impulse terms, with - indicating both simulations were unstable, and  $\infty$  indicating the unstable simulation was stabilized by using the conservative scheme. For the underrelaxed partitioned scheme, using the conservative impulses generally improved iteration counts and allowed for more challenging mass ratios, though smaller ratios were still unstable. Using the XPM, all simulations were stable and iteration counts were improved up to 30% over the RMI.

shows the number of iterations of the underrelaxed partitioned and XPM solutions over 540 substeps of two-dimensional simulations of a balloon filling with smoke, with M = 0.8,  $\omega = 0.2$ . Initially, both methods perform comparably, but as the balloon stretches and the simulation becomes more challenging, the performance of the underrelaxed partitioned approach drops, while the XPM performance is largely unchanged, offering 3-4x better performance in the most challenging part of the simulation.

Table 2.2 gives further details on the performance improvement using the conservative impulses, comparing the partitioned scheme with and without conservative impulses for a variety of underrelaxation parameters, as well as the RMI and XPM schemes.



Figure 2.11: Number of iterations of the underrelaxed partitioned (P) and reduced model interface with conservative impulses (XPM) over 540 substeps of two-dimensional simulations of a balloon filling with smoke. M = 0.8,  $\omega = 0.2$ . The values are averaged over a window of 10 substeps to reduce noise.

## 2.6.2 Boundary pressure projection method

The boundary pressure projection method is orthogonal to the RMI or use of conservative impulses and can be used with any partitioned scheme. Figure 2.6 shows a two-dimensional simulation of a deformable balloon being inflated with air. Inside the balloon, the air is surrounded by either solid faces, or by the source boundary conditions, and hence constitutes a "Neumann region". Without the BPP, the balloon fails to properly inflate.

Further comparison is shown in the Figure 2.12, depicting results of the partitioned scheme without BPP (P), partitioned scheme with BPP (P-BPP), XPM, and monolithic scheme (M). Using the BPP, both the P-BPP and XPM approaches successfully inflate the balloon. XPM is better able than P-BPP to replicate the balloon shape produced with the



Figure 2.12: Comparison of partitioned without BPP (P), partitioned with BPP (P-BPP), XPM and monolithic (M) methods for a 2D balloon filling scenario with fully enclosed fluid.

"ground truth" monolithic method.

Figure 2.7 shows a three-dimensional simulation of a balloon filling with smoke, using XPM. As in the 2D case, the balloon does not properly inflate without the BPP. Figure 2.8 shows a three-dimensional simulation of a hydraulic press, where pressure applied to the small piston is transmitted undiminished to the large piston through a fully enclosed and incompressible medium. Without the BPP, both rigid plates fall downward and the simulation eventually fails. Passive tracer particles are included as a post-process to better visualize the flow. We note that they are advected with interpolated velocities and forward Euler time integration, leading to visual density artifacts that do not reflect density errors in the flow solver.

## 2.6.3 Variety of solids coupled to an Eulerian smoke and water

We demonstrate our method in three dimensions on a variety of challenging scenarios. We use PhysBAM for both the solid and fluid solvers. We again use a tolerance of 0.05dx for the XPM. Figure 2.9 demonstrates the coupling of a deformable object and



Figure 2.13: Complex example: Simulation of cloth and rigid bodies of varying densities coupled to Eulerian water. (fluid grid size:  $120 \times 120 \times 60$ )

	# framos	Avg. time	Avg. number	
Example	# frames	per substep	substeps	
		(s)	per frame	
smoke balloon (Fig. 2.7)	117	30.3	39.2	
water balloon (Fig. 2.4)	143	96.3	30.0	
water piston (Fig. 2.8)	143	140.6	3.2	
smoke rigid (Fig. 2.2)	80	96.6	10.5	
smoke deformable (Fig. 2.9)	500	14.6	12.1	
water rigid (Fig. 2.5)	495	30.7	11.8	
water complex (Fig. 2.13)	447	48.9	16.0	

Table 2.3: The number of frames, average simulation time per substep, and average number of substeps per frame are shown for several examples.

smoke through the RMI. Figure 2.2 demonstrates our extension of the RMI to handle rigid bodies by simulating a rigid plank coupled to smoke. In Figure 2.4, a deformable balloon is filled with water using our XPM. In Figure 2.5, the XPM is used to couple many rigid balls to water, capturing the buoyancy forces. Figure 2.13 depicts a complex example with interacting water, cloth, and rigid bodies of varying densities.

Table 2.3 summarizes the simulations times for each example. Overall, our 3D simulations took roughly between a few minutes per frame to up to a couple of hours per frame on up to 64 cores. A small amount of time is spent in the XPM interface layer as compared with the individual solid and fluid solvers (see Table 2.4). We also note that in our experimentation, without conservative impulses, some of these simulations did reach the maximum 30 iterations, whereas with the conservative impulses, the average number of iterations was 2-7.

Frample	Fluid	Solid	XPM	Avg
Example	solver	solver	interface	# modes
smoke balloon (Fig. 2.7)	95.5	0.3	4.2	2.1
water balloon (Fig. 2.4)	98.5	0.4	1.1	5.9
water piston (Fig. 2.8)	98.2	0.4	1.4	3.8
smoke rigid (Fig. 2.2)	99.7	0.1	0.2	2.5
smoke deformable (Fig. 2.9)	92.6	6.3	1.0	3.2
water rigid (Fig. 2.5)	99.1	0.4	0.5	1.0
water complex (Fig. 2.13)	94.5	3.8	1.7	5.1

Table 2.4: Percentage of time spent in the fluid solver, solid solver, and XPM interface, and average number of modes used by the RMI (average number of iterations minus one). The XPM interface computations require a relatively small amount of time, taking 0.2-4.2% of the total simulation time in our examples.

## 2.7 Conclusion

We have presented a novel extended partitioned method for two-way coupling of incompressible fluids and solids. We have demonstrated that the XPM allows one to solve with a partitioned solver, reusing existing black box solid/fluid solvers, scenarios that were previously solved with specialized monolithic solvers. The XPM could be used to couple other fluid and solid solvers in a black-box fashion, and the reduced model interface could further apply to coupling various materials efficiently. We also note that the reduced model is dynamically updated throughout the simulation and applied locally in a small time window, rather than using a fixed, precomputed basis over a long simulation time.

The RMI method is closely related to quasi-Newton methods for partitioned solidfluid coupling [18, 20, 17]. Recently, it was shown that saving solver state over multiple time steps can lead to improved estimates of the solver Jacobians [25]. It may also be interesting to consider the use of other types of reduced models in constructing the RMI. When using the XPM, we ran most of our examples to a convergence tolerance of 5% of the fluid grid cell size. Since convergence is not necessarily required for graphics applications, it may be practically better to run the XPM with a small, fixed number of iterations, or enforce a larger convergence tolerance, as demonstrated in the SPH example. This maintains stability benefits of XPM, but the reduces the cost at the expense of accuracy. While the XPM significantly improves stability over the partitioned approach with underrelaxation, it is not unconditionally stable.

In future work, we would like to further investigate the BPP on multiple coupled regions, as outlined above. We are also interested in applied our method to scenarios with surface tension.

# Chapter 3

# Smooth Particle Hydrodynamics with XPM

## 3.1 Introduction

In the Lagrangian approach, the fluid variables are stored and evolved on a set of particles that are sampled in the fluid domain, as an alternative representation to the Eulerian approach, where the fluid is simulated utilizing a grid. A sketch showing the representation of a fluid domain with both approaches is presented in Figure 3.1. Smoothed Particle Hydrodynamics (SPH) is one of the most well-known Lagrangian methods, where the mass-carrying volumetric particles are used to build a density field and a pressure force field to drive the fluid motion. SPH is mostly preferred in applications that require interactivity and faster frame rates such as games, since especially the pure Lagrangian formulation can be parallelized on the GPU and CPU very efficiently in a straightforward



Figure 3.1: Eulerian (left) and Lagrangian (right) representations of fluid.

way, and does not require any global solves such as the pressure solve in Eulerian methods.

The SPH framework has been employed to simulate fluids such as smoke and water. It has received significant attention due to its simplicity and performance, and several authors in computer graphics and the computational physics community have investigated the SPH method and its drawbacks and benefits.

Coupling SPH fluids to solids is an active research problem and is addressed in various works. In most of these studies, accurate and smooth boundary representation of the solid and free surface is sought. In this chapter, we look at the coupling problem from a different and orthogonal perspective, as we try to address the performance and stability issues due to the partitioned nature of the coupling. We apply the RMI method to SPH and demonstrate the stability and performance benefits on a scenario that requires strong coupling for stability using a publicly available SPH library, *SPlisHSPlasH*. Furthermore, the formulation and implementation details of the application are provided.

## 3.2 Related Work

The smoothed particle hydrodynamics method, as described in [37], simulates fluids by computing a density field on particles, then by using an equation of state (EOS) to find pressure forces that penalize density deviations to enforce incompressibility. The method has been studied by the computer graphics research community after the introductory work by Müller et al. [38] where free-surface fluids are simulated using a weakly compressible SPH method (WCSPH). Although it presents reasonable solutions, a major drawback of WCSPH is that it requires high stiffness values for incompressibility, demanding smaller time steps and leading to poor computational performance.

Splitting, a commonly used method in Eulerian-based fluids, is employed in many SPH methods in order to allow larger time steps. Typically,, it is accompanied by implicit solvers that incorporate Jacobi iterations to enforce the constant density constraint. Solenthaler and Pajarola developed a predictive corrective scheme (PCISPH) that corrects the particle density errors [48] with position and density predictions. In [28], the particle densities are predicted using velocities instead and are corrected by computing the pressures that are determined by solving a Poisson pressure equation (PPE) on the particles, in order to achieve larger time steps while avoiding the costly neighborhood search during a substep. The divergence-free SPH (DFSPH) method incorporates a velocity solver that enforces the divergence-free constraint along with a density correction solver [10]. DFSPH is demonstrated to achieve up to 20 times faster performance in comparison to PCISPH, especially when large time step sizes are used. In our experiments, we have adopted a publicly available solver for DFSPH, but, we would like to remind the reader that XPM treats solvers as black boxes, and can be applied to any available SPH solver with the procedure described in Section 3.4.

In the hybrid approaches presented in [44], [34], [35], the Laplacian is built on a coarse background grid where the face velocities are interpolated from the particle velocities. Then, a PPE is solved on the grid and the resulting pressure field is distributed to particles in order to speed up the propagation of information. This step is usually followed by an WCSPH step. Although this methods broadcasts global density errors quickly, local errors might still require small time steps.

Many methods have also been developed for the coupling of Lagrangian fluids to deformable and rigid solids. Müller et al. modeled the no-penetration and no-slip boundary conditions between a triangulated surface and fluid particles by using a convolution integral to over a set of particles to calculated distance-based repulsion forces [39]. Direct forcing to achieving boundary conditions was also done in [8]. In [41] a conservative, impulse-based interaction between rigid bodies and SPH fluids is proposed. Particle-based granular flows were two-way coupled with solids in [40]. Implicit two-way coupling of solids and fluids is naturally handled in frameworks that treat both in a unified fashion, such as with a fully Lagrangian approach [29, 49, 26, 47], or a hybrid approach [50]. A momentum-conserving approach for coupling SPH to thin and volumetric rigid bodies was proposed in [3]. In this work, the fluid densities are reflected by the solid particles and their density distributions are corrected to handle dense and irregular sampling of solid particles. The method was subsequently extended for coupling SPH to elastic solids in [2]. As an alternative to Lagrangian representation of the surface, in [10], the solid boundary conditions are implemented using a density map, which can be precomputed for rigid bodies, and smoother surface normals and boundary forces are achieved.

In these works, the coupling approaches are demonstrated on less-stiff scenarios, where a weak coupling approach is sufficient, as strong-coupling is beyond the scope of these works. In our study, we have investigated the stability and performance issues that arise due to the partitioned nature of coupling, and explained the application of the RMI method to the SPH framework. Our method is orthogonal to the solid-fluid coupling scheme and can be directly applied to any strategy that implements particle sampling at the solid-fluid interface.

## 3.3 Smoothed Particle Hydrodynamics Overview

The Navier-Stokes equations that govern fluid evolution are written as

$$\frac{Dv}{Dt} = -\frac{1}{\rho}\nabla p + \nu\nabla \cdot \nabla v + F^e$$
(3.1)

$$= F^p + F^\nu + F^e \tag{3.2}$$

with the incompressibility constraint

$$\nabla \cdot v = 0, \tag{3.3}$$

where  $F^e$  is the sum of external forces such as gravity or penalty forces,  $F^p = -\frac{1}{\rho} \nabla p$  is the pressure force,  $F^{\nu} = +\nu \nabla \cdot \nabla v$  is the viscosity force, and  $D(\cdot)/Dt$  is the material derivative. We will omit the  $F^{\nu}$  term in the latter parts, since it is not used in our simulations, and it is beyond the focus of this thesis. Unlike Eulerian approach, in Lagrangian methods, the velocities are carried on the particles, therefore advection and convection are handled seamlessly as the particle positions are integrated in time. In SPH, the incompressibility constraint,  $\nabla \cdot v = 0$ , is satisfied via the constant density constraint  $\frac{D\rho}{Dt} = 0$ . However, please note that the equality of these constraints may be violated in practice due to numerical integration errors, and if handled properly performance benefits can be gained [10]. Density and pressure are some of the other variables that are stored and evolved on particles. In order to calculate any of these variables at an arbitrary point in space, typically, isotropic nonlinear kernel weight functions, W, that are centered at particle locations are used; i.e. the variable e is calculated at a point x as a weighted sum of all particle values,  $e_j$ :

$$e(x) = \sum_{j} \frac{m_{j}}{\rho_{j}} e_{j} W(x - x_{j}), \qquad (3.4)$$

where  $m_j$  is the constant mass,  $\rho_j$  is the density, and  $x_j$  is the position of a particle j. The density of a particle, i, can be calculated with this formulation as:

$$\rho_i(x) = \sum_j \frac{m_j}{\rho_j} \rho_i W_{ij} = \sum_j m_i W_{ij}, \qquad (3.5)$$

where  $W_{ij} = W(x_j - x_i)$ .

According to [38], the equation of state (EOS) for the fluid is given by the equation

$$p_i = \frac{k\rho_0}{\gamma} \left( \left(\frac{\rho_i}{\rho_0}\right)^{\gamma} - 1 \right), \tag{3.6}$$

with stiffness parameters  $\gamma$  and k. In graphics applications,  $\gamma$  is generally picked to be close to 1. The formulation provides information propagation similar to travel of sound waves.

The pressure force,  $F_i^p$ , acting on a particle is found by the pressure gradient, just like the Eulerian approach, however, a gradient kernel,  $\nabla W$ , is used instead, befitting the SPH scheme:

$$F_i^p = -\sum_j m_j \frac{p_j}{\rho_j} \nabla W_{ij}.$$
(3.7)

Typically, a modified version of this equation is preferred, since it is not symmetric, and therefore, does not preserve momentum:

$$F_{i}^{p} = -\sum_{j} m_{j} \frac{p_{j} + p_{i}}{2\rho_{j}} \nabla W_{ij} \quad \text{or} \quad F_{i}^{p} = -\sum_{j} \rho_{i} m^{j} (\frac{p_{i}}{\rho_{i}^{2}} + \frac{p_{j}}{\rho_{j}^{2}}) \nabla W_{ij}.$$
(3.8)



Figure 3.2: (Left) The density is underestimated on the particles near the solid due to deficit in nearby particles. (Right) Correct density calculations with solid sample particles.

## Solid-fluid coupling

There are various ways to represent solids in a coupling scheme for SPH as discussed in Section 3.2. In the particle sampling approach, a set of particles is sampled over the solid surface and used either to build a repulsive force field, or to account for missing fluid particles near the solid boundary as illustrated in Figure 3.2. In the latter approach, sample particles are treated as special fluid particles that carry mass. In [3], the varying mass,  $\Psi_b$  of each boundary particle *b* is precalculated by using other sample particles, *k*, in its vicinity:

$$\Psi_b = \frac{\rho_0}{\sum_k W_{bk}}.\tag{3.9}$$

This formulation scales the masses of the sample particles and helps avoid large pressure forces and instability due to overly sampled regions. The density and pressure formulations are changed slightly to incorporate these masses:

$$\rho_i(x) = \sum_j m_i W_{ij} + \sum_b \Psi_b W_{ib} \tag{3.10}$$

$$F_i^p = -\sum_j m_j \frac{p_j}{\rho_j^2} \nabla W_{ij} - \sum_b \Psi_b \frac{p_i}{\rho_i^2} \nabla W_{ib}.$$
(3.11)

The fluid forces acting on the rigid bodies are found by reflecting the pressure forces in Eq. 3.11 on the solid particles in order to preserve momentum. These forces are then converted to torques and linear forces acting on the rigid bodies. This particular coupling method can be categorized as a Robin-Robin-like (RR) scheme, although the interaction is force-based, the pressure forces are directly affected by positions.

Algorithm 4 outlines a classic WCSPH step that evolves the fluid according to the Navier-Stokes equation (Eq. 3.2). The force and density calculations involving sampled solid particles b are included by extending Eqs. 3.10 and 3.11.

#### **Divergence-free SPH**

In our work, we preferred the DFSPH method [10] that allows larger time steps. The method achieves this by doing two solves with Jacobi iterations on all particles. The first solve enforces the constant density constraint by predicted velocities. Once densities are corrected, the particles are moved with an Euler step on positions. This is followed by

## **ALGORITHM 4:** A standard substep of WCSPH with time-step $\triangle t$

 $_1$  for all particles i do

```
2 find neighborhood N_i
```

```
3 end for
```

```
▷ Compute densities
```

- 4 for all particles i do
- 5 Compute density with Eq. 3.10
- 6 end for

▷ Compute pressure forces

- 7 for all particles i do
- 8 Compute pressure force with Eq. 3.11

9 end for

▷ Euler-step position and velocities

10 for all particles i do

11  $v_i = v_i + \triangle t(F_i^e + F_i^p)$ 12  $x_i = x_i + \triangle t v_i$ 

```
13 end for
```

standard density and neighborhood calculations. A DFSPH step is then finalized with the divergence-free solver that changes the particle velocities to enforce incompressibility. The algorithm is outlined in Algorithm 5.

## **ALGORITHM 5:** A DFSPH substep with time-step $\triangle t$

- 1 for all particles i do
- 2 compute forces  $F_i^e$  and  $F_i^{\nu}$
- 3 end for
- 4 predict velocities for the next time-step with non-advection forces
- 5 correct densities with predicted velocities using an implicit method
- 6 compute  $x_i$  for each particle *i* using forward Euler-step
- 7 find neighborhood of each particle as in Algorithm 4
- s compute density of each particle as in Algorithm 4
- 9 enforce divergence-free constraint by updating velocities using an implicit method

## 3.4 XPM on SPH

We have implemented the reduced model interface (RMI) for coupling an SPH fluid solver and a rigid body solver. For SPH, we used the DFSPH [10] implementation in the open source *SPlisHSPlasH* library [9], and for the rigid bodies, a position-based dynamics [21] solver, also included in the *SPlisHSPlasH* library, is used. Internal to the library, solid-fluid coupling is implemented using the method of Akinci et al. [3].

Strong coupling is implemented by adding underrelaxation and rollback (save/restore) procedures and calling the solid and SPH solvers successively as outlined in Algorithm 6.

The solid positions and velocities at the end of a time step are determined after the underrelaxation step, therefore, an interpolation method for rigid degrees of freedom is implemented, while the sample particles can still be used to calculate and store the effective velocities of the solids since these velocities are needed by the predictive-corrective scheme.

## ALGORITHM 6: A substep of the strongly-coupled partitioned approach for SPH.

1 k = 1

- $_2$  F0  $\leftarrow$  SPH solver state at the beginning of sub-step
- $\mathbf{s} \mathbf{S0} \leftarrow \text{solid solver state at the beginning of sub step}$
- 4 while ||R|| > tolerance do

5	if $k > 1$ then			
6	restore the SPH solver state to ${\bf F0}$			
7	end if			
8	$X_s^{k+1} = \text{Solid}\_\text{Solve}(f_p^k, \triangle t)$			
9	$X_{s}^{k+1} = \omega X_{s}^{k+1} + (1-\omega) X_{s}^{k}$			
10	if $k > 1$ then			
11	restore the solid solver state to <b>S0</b>			
12	end if			
13	$f_p^{k+1} = \operatorname{SPH}(X_s^{k+1}, \triangle t)$			
14	$R = X_s^{k+1} - X_s^k$			
15	k = k + 1			
16 end while				

#### Reduced model interface on SPH

The reduced model interface is implemented as an additional layer between the solid and fluid solvers. The formulations in Chapter 2 need to be modified slightly for SPH-rigid coupling. This begins by modifying the fluid solver representation in Eqs. 2.1 and 2.2 to:

$$f_s = SPH(X_s), \tag{3.12}$$

$$X_s = S(f_s), \tag{3.13}$$

where  $X_s$  is the sample positions, and  $f_s$  is the fluid pressure forces sampled on the solid particles.  $f_s$  is computed in the SPH solver without the need to modify the code.

After this modification, the rest of the equations in Section 2.3 can be written in terms of  $f_s$  instead of the pressure values p, and  $X_s$  replaces X. Please note that the size of system is changed since  $f_s$  values are vectors while the p are scalars. The RMI blocks, now, try to predict the  $X_s$  and  $f_s$  of the sampled particles.

**Rigid-bodies with RMI and SPH:** Although the coupling interface of SPH consists of only the sample particles, unlike the Eulerian implementation that required rasterization of rigid bodies on the fluid grid for coupling, the solution of the RMI that predicts the solid positions still only adjusts the boundary particles, and may not satisfy the rigid motion properties. Therefore a treatment, similar to the method described in Section 2.3.4 is still required.

## Impulse-terms and closed regions

For the standard SPH algorithms the impulse terms are not needed since the coupling schemes are already momentum preserving, however as mentioned in Section 2.4, the methods involving a PPE solve might still require minor modifications for momentum conservation. Additionally, closed regions are handled automatically by the constant density constraint solver since fluid densities are mirrored onto the solid particles and the pressure forces are reflected.

## Optimizations

In addition to straightforward parallelization of the particle iterations in RMI and strong coupling approaches, the computational performance can be further improved considering the split treatment of solid samples and fluid particles. Specifically, fluid rollback functions can be modified to save and restore the state of the SPH solver after the internal variables, such as fluid density contributions, are calculated, since they would result into the same values during the fluid solve immediately after restore call. Moreover, fluid solver can be redesigned to facilitate rollback function cleverly by updating only variables of the particles in vicinity of the solid boundary, based on the information propagation of the SPH scheme, i.e. the costly neighborhood search at every subiteration can be partially avoided.

## 3.5 Results

We compare the performance of an underrelaxed partitioned strong coupling scheme, the XPM, and the native weak coupling of SPlisHSPlasH

#### XPM improves stability over weak coupling.

Figure 3.3 (top row) shows snapshots from a simulation of a rigid ball interacting with fluid, where the density ratio of the ball to the fluid is 0.1. The XPM results are iterated until a convergence tolerance of 10% times the particle radii is achieved. In this case, both the weak coupling native to SPIisHSPIasH and the XPM solution are stable. Next, we decreased the density ratio to 0.05, making the problem more challenging. In this case, the weakly coupled SPIisHSPIasH became unstable. The XPM was able to compute the simulation stably. Figure 3.3 (bottom row) shows snapshots from the simulation with density ratio 0.05. In these examples, 5684 fluid particles were used. Figure 3.4 shows an XPM simulation of an SPH fluid coupled to a rigid sphere (M=0.05), cylinder (M=2), and torus (M=0.7), with 24389 SPH fluid particles.

#### XPM improves performance over underrelaxed strong coupling.

We also tested the underrelaxed partitioned strong coupling approach in Algorithm 3 and 6 on the SPH/rigid coupling. We chose  $\omega = 0.05$  for the underrelaxation parameter, since larger values of  $\omega$  had failed for density ratios less than 0.1 (see Table 2.1). As shown in Figure 3.5, strong coupling was able to simulate the example with density ratio of 0.05, but took more iterations than the XPM on average to do so. In the most challenging parts of the simulation, the underrelaxed partitioned coupling approach required 2-3x more iterations than the XPM. Therefore, the XPM improves performance when compared with the underrelaxed, partitioned strong coupling.



Figure 3.3: Top row: Simulations of a rigid ball coupled to an SPH fluid, with ball/fluid density ratio of 0.1. All images are at frame 26. Left: Weakly coupled solution computed with SPIisHSPlasH is stable. Right: Solution computed with XPM, with a tolerance of 10%, is also stable. Bottom row: The density ratio has been decreased to a more challenging value of 0.05. Left: Weakly coupled solution computed with SPIisHSPlasH is now unstable. Right: Solution computed with SPIisHSPlasH is now unstable. Right: Solution computed with SPIisHSPlasH is now unstable.


Figure 3.4: XPM simulation of an SPH fluid coupled to a rigid sphere (M=0.05), cylinder (M=2), and torus (M=0.7), with 24389 SPH fluid particles. Top to bottom, left to right: frame 17, 61, 105, and 360.



Figure 3.5: Number of iterations per substep for simulation of an SPH fluid coupled to a rigid ball with density ratio 0.05, using the underrelaxed partitioned scheme (P) and the XPM with tolerance 1% and 10%. The XPM requires fewer iterations on average than the underrelaxed partitioned scheme. Increasing the tolerance further reduces iteration counts while stability is maintained. The values are averaged over a window of 10 substeps to reduce noise.

### Strong versus weak XPM

So far in the thesis, we have not explored the tolerance parameter used by the XPM. Here, we illustrate that the XPM can be used with a larger tolerance to compute a weak coupling, which will be faster but less accurate. We computed the rigid ball/fluid simulation with the XPM with both a tolerance of 1% for strong coupling and a tolerance of 10% for weak coupling. Both computed stably, and the 10% tolerance required fewer iterations as expected. The larger tolerance may be a good choice when speed is preferred over accuracy and stability is not compromised. Figure 3.5 compares the iteration counts for the simulations of the underrelaxed, strongly coupled solution, XPM with tolerance 1%, and XPM with tolerance 10%.

## 3.6 Conclusion

In this section, we have investigated partitioned SPH-rigid coupling methods and described the application of RMI. RMI is quite modular, and only a few modifications were necessary for this adaptation. We have demonstrated the benefits of RMI on a state-ofthe-art SPH method (DFSPH) and coupling scheme on scenarios where the weakly coupled approach cannot handle. Similar to the Eulerian approaches, RMI produced improvements in stability and performance especially as the example becomes stiffer. As the future work, coupling schemes with density maps can be improved with RMI.

# Chapter 4

# Boundary Pressure Projection for Incompressible Dirichlet Fluid Domains

# 4.1 Introduction

Partitioned numerical schemes for fluid-structure interaction problems often employ a Dirichlet-Neumann decomposition, where the fluid equations are solved subject to Dirichlet boundary conditions from the structure, and the structure equations are solved subject to Neumann boundary conditions from the fluid. When the fluid is incompressible, the velocity field must satisfy the divergence-free condition everywhere. Consequently, the net flow through the boundary of any fluid region must also be zero. If the solution of the fluid equations in a region is subject to Dirichlet boundary conditions, it is necessary that the boundary conditions satisfy the constraint of zero net flow. Otherwise, they are incompatible with the flow equations and no solution exists. Examples of such a scenario include the inflation of an elastic balloon and the action of a hydraulic press. In general, enclosed fluid regions can arise spontaneously in free boundary fluid-structure interaction problems.

While monolithic approaches naturally deal with enclosed fluid regions [46], partitioned schemes require special treatment of this case for robustness. In partitioned fluidstructure interaction schemes with Dirichlet-Neumann decomposition, the structure is unaware of the incompressibility constraint on the fluid and will generally supply the fluid with incompatible velocities. Furthermore, the constant pressure mode is not determined by the fluid solver, although it is felt by the structure in the coupling. This problem was described in [31], which proposed a method whereby the boundary constraint is incorporated into the structure equations and solver, requiring their modification. Two alternatives were also discussed in [31]. The first is to replace the Dirichlet-Neumann decomposition with a Neumann-Dirichlet decomposition where the structure imposes forces on the fluid and the fluid imposes interfacial displacements on the structure. The second alternative approach is to incorporate artificial compressibility [15, 43] into the fluid equations to eliminate the issue of incompatibility while iterating the solution in pseudo-time toward an incompressible state. The performance of the artificial compressibility method is improved when combined with quasi-Newton methods, that are designed to stabilize and improve the performance of strongly coupled partitioned approaches [11].

In this section, we present a boundary pressure projection method which allevi-

ates boundary velocity incompatibility and computes the constant pressure mode, while maintaining the Dirichlet-Neumann structure of the decomposition. Our method takes incompatible velocities from the structure solver, projects them to be compatible while in the process computing the constant pressure mode for the region. The compatible velocities are then used as Dirichlet boundary conditions for the fluid while the constant pressure mode is added to the fluid solver pressures to be applied to the structure. We demonstrate our approach in a variety of scenarios including inflation of an elastic balloon and action of a hydraulic press. Our approach overcomes the incompressibility dilemma using a small intermediate computation while respecting the Dirichlet-Neumann decomposition.

We note that we first introduced the boundary pressure projection approach in [1] as explained in Chapter 2. In that work, the constant pressure mode was determined by considering the resulting change of momentum of the *fluid*. While this works for scenarios where the solid density was close to the fluid density, it fails in cases where the solid becomes more massive, as we demonstrate in this work. Here, we correct the formulation to use the change of momentum of the *solid* to determine the correct magnitude of the constant pressure mode. This allows the fluid to support a solid with large relative density. We also formulate the approach for the case of underrelaxed partitioned iterations, and study its behavior for multiple coupled enclosed regions separated by a thin shell or rigid body.



Figure 4.1: The boundary pressure projection method demonstrated on a representative thin-shell example. Top: Before the BPP, the fluid velocities are incompatible as there is a net inflow from the source on the left. The BPP is used to project the fluid boundary velocities and to determine the constant pressure mode to be applied to the solid. Middle: Fluid boundary velocities after compatibility projection. Bottom: Solid velocities (red) after application of the constant pressure mode.

## 4.2 Equations and Methods

## 4.2.1 Fluid Equations

The fluid equations are the incompressible Navier-Stokes equations

$$\rho\left(\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u}\right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{f}, \quad \mathbf{x} \in \Omega$$
(4.1)

$$\nabla \cdot \mathbf{u} = 0. \tag{4.2}$$

The fluid is subject to Dirichlet boundary conditions

$$\mathbf{u} = \bar{\mathbf{u}}, \quad \mathbf{x} \in \partial \Omega. \tag{4.3}$$

## 4.2.2 Fluid Solver

Our partitioned approach treats the fluid as a black-box solver. We denote the fluid solver as

$$p = F(\mathbf{X}, \mathbf{V}),$$

where **X** is the fluid-structure interface position, **V** is the structure velocity at the interface, and p are the fluid pressures inside the fluid domain  $\Omega$ . This interface for the fluid solver supports the Dirichlet-Neumann coupling of the solvers as it can accept as input the Dirichlet boundary conditions from the solid.

In our examples, we discretize the fluid equations on a standard, uniform MAC grid and solve them using the fractional-step projection method [15] as follows. First, the advection, viscosity, and force terms are integrated to obtain an intermediate velocity  $\mathbf{u}^{\star}$ ,

$$\rho\left(\frac{\mathbf{u}^{\star}-\mathbf{u}^{n}}{\bigtriangleup t}+(\mathbf{u}\cdot\nabla)\mathbf{u}\right)=\mu\nabla^{2}\mathbf{u}+\mathbf{f}$$

Next, from the equation

$$\rho\left(\frac{\mathbf{u}^{n+1}-\mathbf{u}^{\star}}{\Delta t}\right) = -\nabla p \tag{4.4}$$

and the incompressibility condition  $\nabla \cdot \mathbf{u}^{n+1} = 0$ , we get the Poisson equation

$$\nabla \cdot \left(\frac{\Delta t}{\rho} \nabla p\right) = \nabla \cdot \mathbf{u}^{\star}.$$

This is solved to obtain the pressure p which is then used in Eq. (4.4) to obtain  $\mathbf{u}^{n+1}$ .

## 4.2.3 Structure Equations

We consider both elastic and rigid structures. The equations of motion for an elastic structure are given by

$$\rho_s \eta_{tt} = \nabla \cdot \sigma_s + \mathbf{f},$$

where  $\rho_s$  is the density,  $\eta$  is the structure displacement,  $\sigma_s$  is the stress tensor and **f** accounts for external forces, including gravity and fluid forces. In our examples, we use the Neo-Hookean constitutive model for solid elasticity. The two-dimensional elastic solids are discretized into a triangular mesh and simulated using a finite volume method. One-dimensional thin shells are discretized into a segmented curve and simulated as a mass-spring system.

For rigid bodies, the equations of motion are

$$\frac{d}{dt} \begin{pmatrix} \mathbf{P}(t) \\ \mathbf{L}(t) \end{pmatrix} = \begin{pmatrix} \mathbf{f}(t) \\ \tau(t) \end{pmatrix},$$

where **P** is the linear momentum of the body, **L** is the angular momentum of the body, **f** is the net force on the body, and  $\tau$  is the net torque on the body. The position and orientation of the body are updated as

$$\frac{d}{dt} \begin{pmatrix} \mathbf{x}(t) \\ R(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \omega^{\star}(t)R(t) \end{pmatrix},$$

where  $\mathbf{x}$  is the position of the rigid body center of mass, R(t) is the orientation of the body,  $I_{r0}$  is the body space inertia tensor,  $I_r(t) = R(t)I_{r0}R(t)^T$  is the world space inertia tensor,  $\mathbf{v} = M^{-1}\mathbf{P}(t)$  is velocity of the rigid body center of mass, and  $\omega(t) = I(t)^{-1}\mathbf{L}(t)$  is the angular velocity of the body.

## 4.2.4 Numerical Method

Since the fluid and solid variables are not collocated, we define an interpolation operator W, such that

$$\mathbf{U} = W\mathbf{V} \tag{4.5}$$

interpolates solid velocities to fluid boundary faces. Conversely, pressure gradients or differences across the fluid boundary faces are conservatively distributed to solid degrees of freedom as

$$\mathbf{F} = W^T p.$$

To communicate quantities between the fluid MAC grid and the Lagrangian solid nodes used in the examples below, we define W as follows. Each element,  $W_{fi}$  in the matrix, defines the weight contribution of solid node i for fluid face f, and it is calculated using an isometric spherical cubic kernel distance function,  $w_{fi} = w(x_i, x_f, r)$ , with kernel radius  $r \in [dx, 2dx]$  in our examples. The weights are normalized so that W is an interpolation operator and  $W^T$  preserves momentum:

$$W_{fi} = \frac{w_{fi}}{\sum_j w_{fj}}.$$
(4.6)

For rigid bodies, auxiliary points are sampled on the surface of the rigid body and W interpolates the velocities of those points.

**ALGORITHM 7:** Single time step without underrelaxation  $(t^n \to t^{n+1})$ 

1  $\mathbf{p}^0 = \mathbf{p}^n 
ho$  Initialize fluid pressure field 2  $p_0^0 = p_0^n \, \triangleright \, \operatorname{Initialize}$  zero pressure 3 for  $k = 0, 1, 2, \dots$  do ▶ Solid solve and pressure correction 4  $(\mathbf{X}^{k+1}, \mathbf{V}^{k+1}) = S(\mathbf{p}^k, p_0^k) \triangleright$  Call solid solver 5  $\left( riangle t C_s M^{-1} C_s^T \right) riangle p_0^{k+1} = -C_s \mathbf{V}^{k+1} \triangleright$  Solve Eq. (4.9) for  $riangle p_0^{k+1}$ 6  $p_0^{k+1} = p_0^k + \bigtriangleup p_0^{k+1} \triangleright$  Correct zero pressure 7 > Compute compatible fluid boundary conditions and fluid solve 8  $ilde{\mathbf{V}}^{k+1} = rac{1}{ riangle t} \left( \mathbf{X}^{k+1} - \mathbf{X}^n 
ight) imes$  Solid effective velocity 9  $ar{\mathbf{U}}^{k+1} = W \mathbb{P} ilde{\mathbf{V}}^{k+1} imes$  Projection for compatibility 10  $\mathbf{p}^{k+1} = F(\mathbf{X}^{k+1}, \bar{\mathbf{U}}^{k+1}) \triangleright$  Call fluid solver 11 12 end for

## Fluid Boundary Constraint

Let  $\mathbf{U}$  be the vector of fluid velocities. The boundary constraints on the fluid velocity are

$$C\mathbf{U} = \mathbf{0}.\tag{4.7}$$

# **ALGORITHM 8:** Single time step with underrelaxation $(t^n \to t^{n+1})$

- 1  $\mathbf{p}^0 = \mathbf{p}^n 
  ightarrow$  Initialize fluid pressure field
- 2  $p_0^0 = p_0^n \triangleright$  Initialize zero pressure
- 3  $\mathbf{X}^0 = \mathbf{X}^n \triangleright$  Initialize solid position for underrelaxation

$$4 \quad \triangle t = t^{n+1} - t^n$$

**5** for 
$$k = 0, 1, 2, \dots$$
 do

6 > Solid solve and pressure correction

7 
$$(\mathbf{X}^{k+1},\mathbf{V}^{k+1})=S(\mathbf{p}^k,p_0^k) \triangleright$$
 Call solid solver

8 
$$\left( \bigtriangleup t C_s M^{-1} C_s^T \right) \bigtriangleup p_0^{k+1} = -C_s \mathbf{V}^{k+1} \triangleright$$
 Solve Eq. (4.9) for  $\bigtriangleup p_0^{k+1}$ 

9 
$$p_0^{k+1} = p_0^k + riangle p_0^{k+1} \triangleright$$
 Correct zero pressure

10 Compute compatible fluid boundary conditions and fluid solve

11 
$$\mathbf{X}^{k+1} \leftarrow \omega \mathbf{X}^{k+1} + (1-\omega) \mathbf{X}^k \triangleright$$
 Underrelaxation of solid positions

12 
$$| ilde{\mathbf{V}}^{k+1} = rac{1}{\wedge t} \left( \mathbf{X}^{k+1} - \mathbf{X}^n 
ight) imes$$
 Solid effective velocity

13 
$$ar{\mathbf{U}}^{k+1} = W \mathbb{P} ilde{\mathbf{V}}^{k+1} 
ightarrow extsf{Projection for compatibility}$$

14 
$$\mathbf{p}^{k+1} = F(\mathbf{X}^{k+1}, ar{\mathbf{U}}^{k+1})$$
 > Call fluid solver

15 end for

Note that C contains exactly one row for each separate Dirichlet fluid region.

### Solid Momentum Balance

We next describe the steps in integrating from time  $t^n$  to time  $t^{n+1}$ . Algorithm 12 outlines the steps in the basic algorithm, and Algorithm 15 includes underrelaxation. Let  $\Delta t = t^{n+1} - t^n$ , and let M and  $\mathbf{V}$  denote the generalized mass matrix and generalized velocity vector for the solid degrees of freedom.

At the start of iteration k, the solid solver is called with inputs of fluid pressures  $\mathbf{p}^k$ and constant pressure mode  $p_0^k$ , and returns positions  $\mathbf{X}^{k+1}$  and velocities  $\mathbf{V}^{k+1}$ , expressed as

$$(\mathbf{X}^{k+1}, \mathbf{V}^{k+1}) = S(\mathbf{p}^k, p_0^k).$$

Next, the solid momentum is corrected to satisfy the fluid velocity boundary constraints. We determine an impulse strength  $\Delta t \Delta p_0$ , such that when the impulse is applied to the solid, the new solid velocity will satisfy the fluid boundary constraints. The impulse is to be applied at the interface of the fluid and structure and is the impulse associated with enforcing the constraint (4.7). Hence, the impulse has the form  $\Delta t C^T \Delta p_0$ , where  $\Delta p_0$  is the Lagrange multiplier associated with the constraint, and corresponds to a change in the constant mode of the pressure field which is in the null space of the fluid equations. Note that the previous estimate  $p_0^k$  was passed to the solid solver, and hence the impulse required to adjust the solid momentum corresponds to a change in pressure  $\Delta p_0^{k+1}$ .

To determine the impulse, we write the solid momentum update due to the impulse

$$M\mathbf{V}' = M\mathbf{V}^{k+1} + \triangle t W^T C^T \triangle p_0^{k+1}, \qquad (4.8)$$

where  $W^T$  is the transpose of the interpolation matrix in (4.5), and acts to conservatively distribute the impulse to the solid degrees of freedom.

Here, we note that [1] instead determined the impulse by solving the equation

$$M_f \mathbf{U}' = M_f W \mathbf{V}^{k+1} + \triangle t C^T \triangle p_0^{k+1},$$

that is, using the fluid mass. While both formulations work for relative densities close to one, the latter fails when the relative density of the solid is large. The formulation in Eq. (4.8) is more accurate, because the force due to the constant pressure mode needs to accelerate the *solid* velocities to be compatible, and the fluid velocities are then determined from the solid velocities.

Multiplying both sides of (4.8) by  $M^{-1}$ , letting  $\mathbf{U}' = W\mathbf{V}'$ , and applying the constraint equation (4.7), we get

$$C\mathbf{U}' = CW\mathbf{V}' = CW\left(\mathbf{V}^{k+1} + \triangle tM^{-1}W^TC^T \triangle p_0^{k+1}\right) = \mathbf{0}.$$

The unknown Lagrange multiplier  $\triangle p_0^{k+1}$  is then determined by solving the linear system

$$\triangle t C W M^{-1} W^T C^T \triangle p_0^{k+1} = -C W \mathbf{V}^{k+1}.$$
(4.9)

This system is  $m \times m$ , where m is the number of Dirichlet fluid regions subject to the boundary constraint. The system (4.9) is solved, and the constant pressure mode is then updated as

$$p_0^{k+1} = p_0^k + \triangle p_0^{k+1}.$$

This pressure correction acts to drive the solid velocities to be compatible with the fluid boundary constraints.

#### **Fluid Boundary Conditions**

The fluid is solved subject to velocity boundary conditions from the solid. In order for the fluid system to be compatible, the boundary velocities must be projected to satisfy the boundary constraint. While the solid velocities  $\mathbf{V}'$  in Eq. (4.8) do satisfy the boundary constraint, in determining fluid boundary conditions we use the effective solid and velocity and may also apply underrelaxation to the solid state. Here we describe how the compatible fluid velocities are computed taking those two things into account.

We first apply any underrelaxation to the solid positions as

$$\mathbf{X}^{k+1} \leftarrow \omega \mathbf{X}^{k+1} + (1-\omega) \mathbf{X}^k,$$

where  $\omega \in (0, 1)$  if underrelaxation is being used, and  $\omega = 1$  corresponds to no underrelaxation. We then compute effective solid velocities as

$$\tilde{\mathbf{V}}^{k+1} = \frac{1}{\triangle t} \left( \mathbf{X}^{k+1} - \mathbf{X}^n \right).$$

Next, we project the effective velocities for compatibility. We rewrite Eq. (4.9) with  $\mathbf{V}^{k+1}$  replaced by  $\tilde{\mathbf{V}}^{k+1}$  and  $\triangle p_0^{k+1}$  replaced by  $\triangle p_0$ , and solve for  $\triangle p_0$  to obtain

$$\Delta p_0 = -\frac{1}{\Delta t} \left( CWM^{-1}W^T C^T \right)^{-1} CW \tilde{\mathbf{V}}^{k+1}.$$

Substituting this result into a solid momentum update equation analogous to Eq. (4.8), we get

$$M\mathbf{V}' = M\tilde{\mathbf{V}}^{k+1} - W^T C^T \left(CWM^{-1}W^T C^T\right)^{-1} CW\tilde{\mathbf{V}}^{k+1},$$

which gives

$$\mathbf{V}' = \tilde{\mathbf{V}}^{k+1} - M^{-1} W^T C^T \left( CWM^{-1}W^T C^T \right)^{-1} CW \tilde{\mathbf{V}}^{k+1},$$
$$= \left( I - M^{-1}W^T C^T \left( CWM^{-1}W^T C^T \right)^{-1} CW \right) \tilde{\mathbf{V}}^{k+1}.$$

Letting  $C_s = CW$  representing the constraint Jacobian with respect to the solid velocities, we write the above as

$$\mathbf{V}' = \left(I - M^{-1}C_s^T \left(C_s M^{-1}C_s^T\right)^{-1} C_s\right) \tilde{\mathbf{V}}^{k+1}$$
$$= \mathbb{P}_s \tilde{\mathbf{V}}^{k+1},$$

where the last equality define the projection operator  $\mathbb{P}_s$  for projecting solid velocities to be compatible with the fluid boundary constraint. The projected interface velocities are then given by

$$\bar{\mathbf{U}}^{k+1} = W \mathbb{P}_s \tilde{\mathbf{V}}^{k+1},$$

and the fluid solver is called with the compatible Dirichlet boundary conditions as

$$\mathbf{p}^{k+1} = F(\mathbf{X}^{k+1}, \bar{\mathbf{U}}^{k+1}).$$

We note that it is also possible to write the projection operator directly on the fluid velocities. In this case, we have

$$W\mathbf{V}' = \bar{\mathbf{U}}^{k+1} = \left(I - WM^{-1}W^T C^T (CWM^{-1}W^T C^T)^{-1}C\right) W\tilde{\mathbf{V}}^{k+1}.$$

If we let  $W_f = WM^{-1}W^T$  denote an effective mass inverse, we can write the fluid projection operator as

$$\mathbb{P}_f = I - W_f C^T \left( C W_f C^T \right)^{-1} C,$$

which has the same form as  $\mathbb{P}_s$ . In summary, the projection operator can equivalently be expressed on the solid velocities or fluid velocities, so that  $\bar{\mathbf{U}}^{k+1} = W \mathbb{P}_s \tilde{\mathbf{V}}^{k+1} = \mathbb{P}_f W \tilde{\mathbf{V}}^{k+1}$ .

## 4.3 Examples

In all examples, we use the minimum of the fluid and solid solver time steps that are determined by the respective solvers in every substep.

## 4.3.1 Supported Rigid Body



Figure 4.2: Supported rigid solid example setup.

For validation, we show an example of a single rigid body supported by an enclosed fluid. The body has total mass M, and is subject to gravitational forces and fluid forces. Therefore, the fluid pressure forces should exactly balance the force of gravity on the body, giving

$$-M\mathbf{g} + Ap_h\mathbf{n} = \mathbf{0},$$

where A is the surface area of the interface,  $p_h$  is the fluid pressure at the interface, and **n** is the fluid region outward normal.

In this example, a rigid body of dimensions  $0.6m \times 0.15m$  is placed on the top of a  $0.6m \times 0.8m$  enclosed fluid domain, as illustrated in Figure 4.2. The fluid is inviscid and has a density of 1  $kg/m^2$ . The solid density is  $10^4 kg/m^2$ . The gravitational acceleration acting on both the solid and the fluid is  $9.8m/s^2$  in the negative y-direction.



Figure 4.3: Supported rigid solid example.  $p - p_0$  values at t=10s.



Figure 4.4: Supported rigid solid example.  $p_0$  values throughout the simulation.

The fluid is simulated on a uniform grid with dx = 0.003125m. For coupling, sample points are distributed on the surface of the solid, at a separation distance commensurate with the fluid grid resolution in order to accurately capture the pressure forces. The algorithm is called iteratively until the convergence criteria are met. Specifically, the iterations terminate when  $\|\mathbf{X}^{k+1} - \mathbf{X}^k\| \leq 10^{-5} dx$  and  $\Delta p_0 \leq 0.1$ . We ran the simulation for 5 seconds with an underrelaxation parameter of 0.1. The fluid pressure field without the constant mode  $p_0$  is illustrated in Figure 4.3. Since the fluid solver has only Neumann faces, its solution has an arbitrary constant mode, and in this case it calculated values in range [-3.90469, 3.90469]N/m which is very close to the centered analytical hydrostatic solution. Our method computes  $p_0$  values that also compensate for this arbitrary solution as plotted in Figure 4.4.



Figure 4.5: Error in fluid forces, computed as  $F'_s - Mg$ , where  $F_s$  is the total pressure force applied on the rigid body by the fluid.

To further demonstrate the solution, we measure the fluid forces applied to the rigid body at the surface sample particles. The error in the total force is plotted over time in Figure 4.5 by finding the y-component difference of the sum of sampled forces, F', and the analytical solution F = Mg. In order to better visualize the error, the plot is split into two parts: on left the log of the error is plotted until t = 0.15s to demonstrate the evolution of the error; error values after t = 0.15s are plotted on the right. We have observed that these errors are related to the underrelaxation parameter. As another metric, we have used the volume change rate at the end of every time step, and the log of this error is shown in Figure 4.6. The same pattern as with the force errors can be observed as the errors get smaller and smaller from the beginning and stabilize at a point with some numerical noise.



Figure 4.6: The log of the outflow volume rate, which is used as another error metric for the supported rigid example.

## Comparison with Fluid Momentum Formulation

Here, we demonstrate the problem with the fluid momentum based formulation of [1] through studying the case of the supported rigid piston with solid/fluid density ratio ~ 1500. As shown in Figure 4.7, both piston heads start to fall down immediately since the constant pressure mode that is required to hold the system in equilibrium is underestimated. As the time passes, the accumulated  $p_0$  value increases and approaches the correct value, however the solution does not achieve equilibrium as the piston heads continue fluctuating up and down until the end of the simulation.



Figure 4.7: Failed piston example with fluid momentum based formulation is depicted at various times t in the simulation.

## 4.3.2 Simple balloon-like problem

Similar to the examples in [31] and [11], a thin volumetric balloon is inflated with a source as depicted in Figure 4.8. The balloon has a density of 1000  $kg/m^2$  and thickness of 0.2m. It has a 1m wide opening, which is fixed to the walls of a static rigid tube that leads to the source. A Neo-Hookean constitutive model is used for the solid, with Young's modulus  $E = 7 \times 10^5 N/m$ , and Poisson's ratio  $\nu_s = 0.45$ . The solid volume is constructed using triangulated areas with minimum side length of 0.02m at the rest state. The total mass is distributed to the triangle vertices equally.



Figure 4.8: Setup of simple balloon-like problem with a volumetric solid.

The fluid has density  $\rho_f = 1.1 kg/m^2$ , while it has a kinematic viscosity  $\nu_s = 0.146 m^2/s$ . The source velocity has a parabolic profile along the y-axis of the source.

Initially, the source velocity is  $0 \ m/s$  and it is increased gradually over 1 second according to the sinusoidal function  $v_c(t) = (\sin(\pi \ t + 1.5\pi) + 0.5) \ u_{\text{max}}$ , where  $v_c(t)$  is the maximum value of the source velocity profile (at the center of the source), at time t. The fluid grid has a cell size of  $0.02m \times 0.02m$ .



Figure 4.9: Volumetric simple balloon-like example.  $p - p_0$  values at t = 15.











Figure 4.10: Snapshots of the fluid domain velocity magnitudes for the volumetric simple balloon-like example.

To capture pressure forces affecting the solid, particles are sampled at the outer vertices of the triangulated solid volume. The underrelaxation parameter is 0.05, and a subiteration is terminated if the maximum vertex displacement between consecutive solid solutions is less than  $10^{-3}dx$ . The simulation is run for 15 seconds, and the state of the balloon and fluid velocity field at various times are illustrated in Figure 4.10. Figure 4.9 depicts the fluid pressures without the constant mode, and the plot in Figure 4.11 shows the evolution of the constant pressure mode,  $p_0$ , throughout the simulation. The outflow volume rate at the end of each time step is calculated by CWV, and the log of this metric throughout the simulation is plotted in Figure 4.12.



Figure 4.11: Volumetric simple balloon-like example.  $p_0$  over time throughout the simulation.



Figure 4.12: The log of the volume change rate for volumetric simple balloon-like problem.

## 4.3.3 Damped Structural Instability

We further demonstrate the method on an example with a nonsymmetric structure and fluid setup, as illustrated in Figure 4.13, similar to previous works [31, 11]. A deformable solid tube is constructed by fixing the ends of two volumetric bands to the walls of static rigid tubes, each of which has a source at its other entrance.

Both of the bands have a thickness of 0.1m and density of  $500 \ kg/m^2$ . A Neo-Hookean constitutive model is used for modeling the solid deformation where the Poisson's ratio,  $nu_s$  is 0.3 and the Young's moduli are  $9 \times 10^5 \ N/m$  for top band and  $9 \times 10^8 \ N/m$  for the bottom band. The fluid density is  $1 \ kg/m^2$  and the kinematic viscosity is  $9 \ m^2/s$ . For both of the fluid sources parabolic profiles are used along the x axis, where the center of



Figure 4.13: Damped structural instability problem.

each source has the velocity  $v_c(t) = (\sin(\pi t + 1.5\pi) + 0.5) u_{\text{max}}$ , which is gradually increased during the first second of the simulation to its  $u_{\text{max}}$  value. The  $u_{\text{max}}$  values are 10 m/s and 10.1 m/s for the left and right sources, respectively.

The solid mesh is built with triangulated areas with a minimum side length of 0.05m. Sample particles are distributed around the surface of the solid to capture the pressure forces from the fluid. The fluid grid has a fixed cell side length of 0.025m. The underrelaxation parameter is set to 0.05, and the solid position convergence tolerance is  $10^{-3} dx$ .

A few snapshots capturing the key frames of the simulation are displayed in Figure 4.14, where the fluid velocity field magnitude is shown along with the solid bands. Similar to the examples in the literature, initially the top band moves up as the tube is inflated, while the stiffer bottom band does not displace significantly. Once a critical pressure is reached, the bottom band starts to give toward the side with higher source velocity, creating a flow inside the tube towards the other side. After the bottom band stabilizes



t=0.25s

t = 1s





Figure 4.14: The fluid velocity field magnitude  $\|\mathbf{u}\|$  at different stages of the simulation for the damped structural instability problem.

the inner flow calms down, and the tube keeps its symmetrical shape till the end of the simulation as the tube is inflating. This behavior is also captured on the  $p_0$  plot, where sudden jumps can be observed after the bottom band gives around t = 2.5s. The value of  $p_0$  resumes its gradually increasing pattern after the solid settles. Since the pressure variations are negligible in magnitude compared to the constant pressure mode value, it does not significantly impact the coupling. Therefore, we only provide the pressure field at the end of the simulation in Figure 4.15.



Figure 4.15: The pressure field before  $p_0$  is added:  $p - p_0$  at time=4s in the simulation of the damped structural instability simulation.



Figure 4.16: The change in  $p_0$  over time in the damped structural instability example.



Figure 4.17: The log of the outflow volume rate for the damped structural instability example.

## 4.3.4 Inflating a Thin-shell Balloon

Our formulation solves for the coupled constant pressure modes for neighboring thin shells. Before testing it with multiple neighboring regions, we first study the simple balloon-like problem with a thin-shell solid. We make this distinction by categorizing solids that can have fluid cells on both sides of their surface as thin shells. In thin-shell solids, pressure force samples are affected by fluid regions on both sides of the surface. In this example, a thin-shell balloon is attached to a static rigid tube that leads to a source as shown in Figure 4.18. The balloon is inflated by the fluid source for 15 seconds.



Figure 4.18: Setup of simple balloon-like problem with a thin-shell solid.

The solid has one-dimensional density of 200 kg/m, calculated by multiplying the balloon density and thickness from the volumetric case. A mass-spring constitutive model that resists stretching and bending is used to model the balloon-like behavior with the following parameters:  $k_{stretch}=5 \times 10^4 \ kg/s^2$ ,  $b_{stretch}=1500 \ kg/s^2$ ,  $k_{bend}=5 \times 10^4 \ kg/s^2$ ,  $b_{bend}=800 \ kg/s^2$ . The restlength of the solid springs are set to 0.02m at the initial setup, and particles are sampled at spring edges for capturing pressure forces.

For the fluid, the density is  $1.1 \ kg/m^2$  and kinematic viscosity is  $0.146 \ m^2/s$ . The source profile is parabolic along the y-axis and the time-changing maximum value of the parabola is computed with the equation  $(\sin(\pi \ t + 1.5\pi) + 0.5) \ u_{\text{max}}$  for t < 1s, then kept constant at  $u_{\text{max}} = 1 \ m/s$ . A uniform grid with cell width dx = 0.02m is used for the fluid domain inside the closed region.



Figure 4.19: The fluid pressure field variation  $p - p_0$ , at the end of simulation, t = 10s, is illustrated with a color map for the simple balloon-like example with a thin-shell solid.



Figure 4.20: Snapshots at various stages of the simulation demonstrating the interaction between the fluid and the thin-shell solid in the simple balloon-like example. Color coding represents the fluid velocity field magnitude,  $\|\mathbf{u}\|$ .

For the simulation, the strong-coupling scheme is employed with a fixed underrelaxation parameter of 0.05 and the convergence tolerance on the solid positions is set to  $10^{-3} dx$ . The spring parameters used in this example result in balloon behavior that is less stiff than the volumetric one, as can be observed in the velocity field snapshots shown in Figure 4.20. The effect of the fluid motion on the solid is more localized in this example, and the balloon bounces back and forth as it is inflated. These movements create fluctuations in the calculated p0 values throughout the simulation as illustrated in the plot in Figure 4.11. The initial values of  $p_0$  are relatively small, but they start to increase significantly, especially after the balloon loses its initial square-like shape around t = 4s.



Figure 4.21: Constant pressure mode,  $p_0$ , of the enclosed fluid domain plotted against simulation time for the duration of the simple balloon-like example with thin-shell solid.

The fluid pressure field has relatively small values without its constant mode. We

provide a representative snapshot of the field at the end of the simulation in Figure 4.19. The volume flux rate is, once again, calculated at the end of each time step for this example as an error metric and is plotted in Figure 4.22. The order of the error is very similar but slightly better than in the volumetric case.



Figure 4.22: The log of the volume flux rate as an error metric for the simple balloon-like example with thin-shell solid.

## 4.3.5 Hydraulic Piston

We demonstrate our method on an example that models a hydraulic piston. Unlike the supported solid example, in the hydraulic piston case the gravitational force acting on the solid in one cylinder of the piston is balanced by the other solid under gravity placed in the other cylinder. The countering forces are transmitted through the fluid medium and are balanced when the system is at equilibrium.


Figure 4.23: Setup of the piston examples. Top: At equilibrium without any external forces. Bottom: At equilibrium with an external downward force applied to the right piston.

We study two different test cases. In the first test, the system is at equilibrium as depicted in Figure 4.23 (top). In the second test, the solid positions are displaced away from the equilibrium state as depicted in Figure 4.23 (right). At the beginning of the simulation, an external force,  $f_{\text{max}}$ , calculated analytically using the total solid-fluid mass of each of the pistons with Pascal's principle, is applied for the first second of the simulation and gradually removed over 5 seconds according to the function  $f(t) = (\cos(0.2 (t-1) pi) * 0.5 + 0.5) f_{\text{max}}$ . The outline of the second case is sketched in Figure 4.23 (bottom). In both scenarios, the fluid is inviscid with a density of  $1 kg/m^2$ . In the first scenario, the solid density is  $1000 kg/m^2$  for both of the rigid bodies, while in the second case, it is  $10 kg/m^2$ , a value closer to the fluid density, so that the simulation time until convergence is shorter.



Figure 4.24: Color-mapped illustration of the pressure values of the fluid without the constant pressure mode,  $p - p_0$  for the hydraulic piston example. (Left) Without external forces at the end of the simulation (t = 10s). (Right) With external forces at the end of the simulation (t = 90s).

For both tests, the fluid grid cell width is dx = 0.02m and the sample particles for the pressure forces are distributed on the rigid body surface with a matching resolution. The underrelaxation parameter is 0.1. The convergence tolerance on the solid position change is  $10^{-5}dx$ , and convergence does not depend on the value of  $\delta p_0$  unlike the rigid support case. The first test is simulated over 10 seconds, since it is already at equilibrium, and  $p_0$ quickly converges as seen in Figure 4.25. A longer duration of 60 seconds is selected for the second example, since it takes time to reach the equilibrium state, especially because the solid has a significantly higher density than the fluid, and the motion is driven by the fluid weight. After release of the external force, the constant pressure mode drops slowly as it approaches the state of equilibrium as shown in Figure 4.26, where a window of 100



Figure 4.25: Constant pressure mode,  $p_0$ , during the simulation of the hydraulic piston example without external forces.

substeps is used for smoothing the values. The fluid pressure field without  $p_0$  is shown in Figure 4.24 for both of the tests. Again, the  $p_0$  computation compensates for the arbitrary constant mode in the fluid solver pressure fields.



Figure 4.26: Constant pressure mode,  $p_0$ , during the simulation of the hydraulic piston example with external forces, smoothed with a Gaussian method over a window size of 100 substeps.

A few selected snapshots from the second example are shown in Figure 4.27. In the first snapshot, at t = 3.5s, the left cap of the piston starts to speed up away from its initial position, reaches its highest velocity around t = 6.5s, and drops down to 0 velocity around t = 9s. Images in the bottom row demonstrate the velocities as the system is cooling down.



Figure 4.27: Snapshots at various stages of the piston simulation with initial external force. Color coding represents the fluid velocity field magnitude,  $\|\mathbf{u}\|$ .





Figure 4.28: Coupled thin-shell balloon example setup.

In the previous examples, we demonstrated our method in scenarios with a single Dirichlet fluid region. As the first example with multiple neighboring regions, the simple thin-shell balloon-like problem is extended by adding another balloon and source that mirrors the original one along the y-axis. As in the simple balloon example, the density of the fluid is  $1.1 \ kg/m^2$  and its kinematic viscosity is  $0.146 \ m^2/s$ , while the one-dimensional solid density is 200kg/m. The parameters of the mass-spring system along with a detailed depiction of the example setup is given in Figure 4.28. The same parabolic profile is used for both of the fluid sources along their relative y-axes. The source velocities are gradually increased during the first second of the simulation with the following equation that gives the maximum value of the parabola:  $(\sin(\pi \ t + 1.5\pi) + 0.5) \ u_{\text{max}}$ , where  $u_{\text{max}}^L = 1 \ m/s$  is used for the left source and  $u_{\text{max}}^R$  is used for the right one.

We carried out two different tests with the same layout. In the first test, Test 1,  $u_{\text{max}}^R$  is set to the same magnitude as  $u_{\text{max}}^L$  in the opposite direction in a symmetrical setup. In the nonsymmetric case, Test 2, the right source maximum velocity is halved:



Figure 4.29: Color map of the fluid pressure field without the constant pressure mode,  $p_0$ , for the symmetric (top) and nonsymmetric (bottom) tests of two coupled thin-shell balloons.

 $u_{\text{max}}^R = 0.5 \ m/s$ , while it is still in the opposite direction of  $u_{\text{max}}^L$ . For both tests, the fluid grid dx is 0.02m, and the springs in the solid have a matching resolution. The vertices of the solid are used as sample locations for the pressure forces.

The sole convergence criterion for both tests is a tolerance value of  $10^{-3} dx$  for the maximum solid displacements in consecutive iterations, while the underrelaxation parameter is fixed to 0.05. The tests are run for 15 seconds, and the fluid pressures without the constant mode at the end of the simulation as well as the state of the balloons is illustrated in Figure 4.29, where the pressure field extrema are located around the edges of the balloons since they still bounce slightly at the end of the simulation. Evolution of the balloon shapes and the fluid velocity fields are depicted in snapshots in Figure 4.30 and Figure 4.31 for the symmetric and nonsymmetric cases, respectively.

In the symmetric case, both the balloon and the velocity fields evolve symmetrically until the end, and the opposing pressures forced the balloons to expand vertically creating a butterfly-like shape. The center line between the balloons does not move horizontally during the simulation and the fluid velocity around it is close to 0. The symmetry can be further observed on the  $p_0$  plot in Figure 4.32, where the constant modes of the two regions overlap for the whole simulation. Also, similar to the single balloon case, a sudden jump occurs around t = 3s, as the balloons lose their original shape totally and bounce back and forth slightly to adopt their new shape.

In the nonsymmetric case, the right region is inflated at half the rate of left region, which has the same source flow as the balloons in the symmetric case. However, in this case, the left region can grow horizontally into a wider shape as the centerline moves



Figure 4.30: The change in the solid position and velocity magnitude of the fluid domain is demonstrated in snapshots from the simulation of two symmetric thin-shell balloons.

t = 15s

1 m/s

t = 11s

0 m/s



Figure 4.31: The change in the solid position and velocity magnitude of the fluid domain is demonstrated with snapshots from the simulation of two nonsymmetric thin-shell balloons.

towards the right. The movement of the center line is mainly driven by the differences of the constant pressure modes of the left and right regions,  $p_0^L$  and  $p_0^R$ , respectively. These modes are illustrated in Figure 4.33, where the difference in values as well as the jumps after shape deformations can be observed. When compared with the symmetric case both  $p_0^L$  and  $p_0^R$  values are smaller, which is an obvious result for the right region as the input source velocities are smaller. On the other hand, it is also one of the main reasons for the lower values in the left region since the  $p_0$ 's are coupled.



Figure 4.32: Constant pressure mode,  $p_0$ , throughout the simulation of the symmetric test of two thin-shell balloons example. Region #1 is the left region, and Region #2 is the right region.



Figure 4.33: Constant pressure mode,  $p_0$ , throughout the simulation of the nonsymmetric tests of two thin-shell balloons example. Region #1 is the left region, and Region #2 is the right region.

The log of the volume outflow rate in plotted for each time step as an error metric for both the symmetric (top) and the nonsymmetric (bottom) examples in Figure 4.34. The values are smoothed with a Gaussian method with size of 100 substeps for clear visualization. While the order of the errors is similar to the single thin-shell balloon case, the errors in the less inflated region in the nonsymmetric case are smaller than the others as expected, as it has lower input from sources.



Nonsymmetric case

Figure 4.34: Log of the volume outflow rate for the symmetric (top) and the nonsymmetric (bottom) examples. The values are smoothed with a Gaussian method with a window size of 10.

### 4.3.7 Inflating Nested Thin-shell Balloons



Figure 4.35: Nested balloon rings problem setup.

In this example, three circular balloon-like thin-shell solids with different radii are centered around a circular fluid source forming a nested layout as depicted in Figure 4.35. In contrast to the two-balloon example, the Dirichlet regions inside the interior balloons are fully coupled to each other and all solid nodes of the interior balloons are affected by the pressures of two different closed fluid regions.

All three balloons have a one-dimensional density of 200  $kg/m^2$ , and are modeled with the same mass-spring system parameters:  $k_{stretch}=5 \times 10^4 \ kg/s^2$ ,  $b_{stretch}=1.5 \times 10^3 \ kg/s^2$ ,  $k_{bend}=5 \times 10^4 \ kg/s^2$ ,  $b_{bend}=800 \ kg/s^2$ . The radii of the solid balloons, from innermost to outermost, are 1.5 m, 2.5 m and 3.5 m. The circular fluid source at the center of balloons has a radius of 0.45 m. The source velocities are in the direction normal to the





Figure 4.36: Snapshots from the nested balloon at various states of the simulation. The velocity magnitudes are illustrated with a color map.

circle, and have a uniform profile along the surface with a gradually increasing magnitude. The maximum source velocity magnitude,  $u_{\text{max}}$  is  $1 \ m/s$ , which is reached in the first second of the simulation with equation:  $(\sin(\pi \ t+1.5\pi)+0.5) \ u_{\text{max}}$ . The fluid density is  $1.1 \ kg/m^2$  and the kinematic viscosity is  $0.146 \ m^2/s$ .

The fluid grid has cell width dx - 0.02m and all balloons have matching spring edge lengths initially. The sample points for the pressure forces are placed at the vertices of the balloon meshes. The solid convergence criterion is used for this example with a tolerance of  $10^{-3}dx$ , and the underrelaxation parameter is fixed at 0.05.

Evolution of the balloons and fluid velocity field is demonstrated in a few images from the simulation in Figure 4.36. The flow coming from the source increases the volume of the innermost region, meanwhile the volumes of the outer regions are unchanged as they are pushed outwards. The constant pressure mode of each region is coupled to its neighboring regions, and their differences move the balloons. Fig 4.37 shows the change of  $p_0$  values, where the regions are numbered in increasing order from outside to inside.

Log values of volume outflow rate of each region are plotted in Figure 4.38, where the error order is similar to the other thin-shell examples, but slightly decreases from innermost region to outermost region. The values are smoothed with a moving window size of 10 substeps with a Gaussian method for better visualization.



Figure 4.37: Plot of constant pressure modes over time for all regions in the nested thinshell balloons example. Region #1: outermost region, Region #2: middle region, Region #3: innermost region.



Figure 4.38: Net volume outlow rate for all regions in the nested thin-shell balloons example. Region #1: outermost region, Region #2: middle region, Region #3: innermost region. Log function as well as Gaussian smoothing with size 10 is applied for cleaner visualization.

#### 4.3.8 Multi-section Piston



Figure 4.39: Setup of piston with multiple sections at equilibrium.

As a final test, we study a hydraulic piston with multiple Dirichlet regions coupled through rigid bodies. In this example, two rigid bodies are placed in various locations of each cylinder of a  $10m \times 12m$  piston, while the regions between the rigid bodies are filled with fluid. The left cylinder has a diameter of 6m, and the right cylinder has a diameter of 2m. The bottom tube's height and width are both 2m. Different densities and sizes are chosen for each rigid body as shown in Figure 4.39, which illustrates the example setup and parameters. The system is designed to be in equilibrium at the initial state and expected to stay stable throughout the simulation since no external forces, other than the gravitational acceleration of  $9.8m/s^2$  that acts on both the fluid and solid, are applied.

The fluid medium between the solids has a density of  $1 kg/m^2$  and is inviscid, while the cell size, dx, of the fluid grid is 0.02m in all regions. Particles are sampled on the surface of the rigid bodies at a resolution matching that of the fluid grid. In the strong-coupling



Figure 4.40: The pressure field of the fluid without the constant pressure mode at the end of the simulation for the multi-section piston example.

scheme, and underrelaxation value of 0.1 is used, and the solid displacement tolerance is set to  $10^{-5} dx$ .

The example is run for approximately 16 seconds. The fluid pressure field without the the constant pressure mode is illustrated in Figure 4.40. Upon examination of the figure, one can observe that all pressure fields are determined up to different arbitrary pressure constant modes by the fluid solver. Our method not only computes the pressures required for the rigid bodies to stay still, but also compensates for the arbitrary constant modes generated by the fluid solver.



Figure 4.41: Constant pressure mode,  $p_0$ , plotted against time during the simulation of the multi-section piston example. Region #1: bottom region, Region #2: left region, Region #3: right region.

The constant pressure modes calculated by our method throughout the simulation are plotted in Figure 4.41 where region #1 is the bottom region, region #2 is the left region, and region #3 is the right region. The pressure on the upper boundaries of regions #1 and #2 can be calculated analytically by multiplying the mass above them with gravity and dividing by their respective tube area. A similar calculation can be done for region #3 to determine the pressure value at an arbitrary height in the region. The computed solutions for the constant pressure modes are very close to the expected analytical values, however, they are not exact since the pressure field of the solvers are compensated.

The log of the net volume outflow rate is plotted in Figure 4.42 at every time step of the simulation. The initial error is around  $10^{-4}$ , however it drops down quickly to  $10^{-10}$ or less. One way to avoid this error is to place a convergence criterion on it or on  $\Delta p_0$  in the strong-coupling scheme.



Figure 4.42: The log of net volumetric outflow rate is used as an error metric for the multisection piston example. Region #1: bottom region, Region #2: left region, Region #3: right region.

### 4.4 Conclusion

In this section, we have revisited the closed region problem in the partitioned methods with Dirichlet-Neumann coupling scheme. The earlier formulation of boundary projection method does not capture the solid dynamics properly and fails especially for high solid-fluid density ratios. We have rewritten the formulation based on solid momentum change instead and provided an algorithm to work with underrelaxation scheme that is commonly employed in strong-coupling. The method respects the black-box design for the solid and fluid solvers, as no internal variables are needed to be accessed or modified.

We have demonstrated the efficacy of our method on several examples. The revised formulation computes correct pressure forces for the solids as shown in supported rigid body and piston examples. The method works with multiple regions without any modification, and it is displayed in various test cases with multiple regions, where coupling of the constant pressure modes are required for correct results and behavior.

In future work, we would like to do a numerical analysis of our method, and investigate the convergence rates and criteria for the strong-coupling scheme. Combining the method with RMI-based partitioned methods would also be another essential study.

## Chapter 5

# Conclusions

In this thesis, we have addressed various issues in the partitioned approach to the two-way solid-fluid coupling problem. One of the main drawbacks of partitioned approaches is their poor performance and lack of stability, especially under stiff scenarios. We have addressed these issues with an extended partitioned approach (XPM) as presented in Chapter 2. The stability and performance benefits are demonstrated in cases where the strongly coupled partitioned approach fails to converge or does so very slowly. Moreover, various three-dimensional simulations are presented to showcase the method's applicability in the computer graphics domain. The application details and algorithm modifications that are required to apply XPM to SPH are explained in Chapter 3. The method is demonstrated on a publicly available simulation software, *SPlisHSPlasH*, using state-of-art solvers for SPH and rigid bodies. Finally, a boundary pressure projection method is presented to address the closed-region problem that arises in partitioned methods with a Dirichlet-Neumann scheme. In Chapter 4, the method is further improved by capturing solid dynamics and supporting underrelaxation, and the formulation is supported by results from various test scenarios including cases with multiple neighboring Dirichlet regions.

All of the methods presented are fairly easy to implement, and they do not require substantial change in the solver code. One requirement for our methods is support for rollbacks, which is implemented additionally on *SPlisHSPlasH*. This functionality is commonly provided by simulation software to allow restarts on interrupted simulations. Another requirement is access to intermediate interface velocities of the fluid solver. However, these velocities are typically calculated directly from the solid velocities and are available external to the solvers.

The methods runs reasonably fast and computational time of the partitioned methods are only affected marginally. For RMI, computational overhead was between 1%-5% in all of our simulations, and for BPP, the size of the system to be solved is equal to the number of closed regions, and can be solved quickly.

In future work, we would like to explore the cases where partitioned methods are faster than the monolithic solvers. We suspect that the partitioned approach, with the presented improvements, can outperform the monolithic approach especially when the number of solid nodes in the simulation is relatively high, making the system built by the monolithic solver significantly larger, while in the partitioned method only the interface, which is fast to solve, and the solid system, which typically requires less computation time for solution, would be affected.

## Bibliography

- Muzaffer Akbay, Nicholas Nobles, Victor Zordan, and Tamar Shinar. An extended partitioned method for conservative solid-fluid coupling. ACM Transactions on Graphics (TOG), 37(4):86, 2018.
- [2] Nadir Akinci, Jens Cornelis, Gizem Akinci, and Matthias Teschner. Coupling elastic solids with smoothed particle hydrodynamics fluids. *Computer Animation and Virtual* Worlds, 24(3-4):195–203, 2013.
- [3] Nadir Akinci, Markus Ihmsen, Gizem Akinci, Barbara Solenthaler, and Matthias Teschner. Versatile rigid-fluid coupling for incompressible sph. ACM Trans. Graph., 31(4):62:1–62:8, July 2012.
- [4] Stefan Band, Christoph Gissler, and Matthias Teschner. Moving Least Squares Boundaries for SPH Fluids. In Fabrice Jaillet and Florence Zara, editors, Workshop on Virtual Reality Interaction and Physical Simulation. The Eurographics Association, 2017.
- [5] Jeffrey W Banks, William D Henshaw, and Donald W Schwendeman. An analysis of a new stable partitioned algorithm for fsi problems. part i: Incompressible flow and elastic solids. *Journal of Computational Physics*, 269:108–137, 2014.
- [6] Jernej Barbič, Marco da Silva, and Jovan Popović. Deformable object animation using reduced optimal control. ACM Transactions on Graphics (TOG), 28(3):53, 2009.
- [7] Christopher Batty, Florence Bertails, and Robert Bridson. A fast variational framework for accurate solid-fluid coupling. *ACM Trans. Graph.*, 26(3), July 2007.
- [8] M. Becker, H. Tessendorf, and M. Teschner. Direct forcing for lagrangian rigid-fluid coupling. *IEEE Transactions on Visualization and Computer Graphics*, 15(3):493–503, May 2009.
- [9] Jan Bender. Splishsplash library, 2017.
- [10] Jan Bender and Dan Koschier. Divergence-free sph for incompressible and viscous fluids. *IEEE Transactions on Visualization and Computer Graphics*, 23(3):1193–1206, March 2017.

- [11] Alfred EJ Bogaers, Schalk Kok, B Dayanand Reddy, and Thierry Franz. Extending the robustness and efficiency of artificial compressibility for partitioned fluid-structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 283:1278–1295, 2015.
- [12] Mark Carlson, Peter J. Mucha, and Greg Turk. Rigid fluid: Animating the interplay between rigid bodies and fluid. ACM Trans. Graph., 23(3):377–384, August 2004.
- [13] Paola Causin, Jean-Frédéric Gerbeau, and Fabio Nobile. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Computer methods in* applied mechanics and engineering, 194(42):4506-4527, 2005.
- [14] Nuttapong Chentanez, Tolga G. Goktekin, Bryan E. Feldman, and James F. O'Brien. Simultaneous coupling of fluids and deformable bodies. In *Proceedings of the 2006* ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '06, pages 83–89, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.
- [15] Alexandre Joel Chorin. A numerical method for solving incompressible viscous flow problems. Journal of computational physics, 2(1):12–26, 1967.
- [16] Alexandre Joel Chorin. A numerical method for solving incompressible viscous flow problems. Journal of computational physics, 135(2):118–125, 1997.
- [17] Joris Degroote. Partitioned simulation of fluid-structure interaction. Archives of Computational Methods in Engineering, 20(3):185–238, 2013.
- [18] Joris Degroote, Klaus-Jürgen Bathe, and Jan Vierendeels. Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction. Computers & Structures, 87(11):793-801, 2009.
- [19] Joris Degroote, Peter Bruggeman, Robby Haelterman, and Jan Vierendeels. Stability of a coupling technique for partitioned solvers in fsi applications. *Computers & Structures*, 86(23):2224–2234, 2008.
- [20] Joris Degroote, Robby Haelterman, Sebastiaan Annerel, Peter Bruggeman, and Jan Vierendeels. Performance of partitioned procedures in fluid-structure interaction. Computers & structures, 88(7):446-457, 2010.
- [21] Crispin Deul, Patrick Charrier, and Jan Bender. Position-based rigid-body dynamics. Comput. Animat. Virtual Worlds, 27(2):103–112, March 2016.
- [22] Jean-Frédéric Gerbeau and Marina Vidrascu. A quasi-newton algorithm based on a reduced model for fluid-structure interaction problems in blood flows. ESAIM: Mathematical Modelling and Numerical Analysis, 37(4):631-647, 2003.
- [23] Eran Guendelman, Andrew Selle, Frank Losasso, and Ronald Fedkiw. Coupling water and smoke to thin deformable and rigid shells. ACM Trans. Graph., 24(3):973–981, July 2005.

- [24] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.
- [25] R Haelterman, Alfred EJ Bogaers, K Scheufele, B Uekermann, and M Mehl. Improving the performance of the partitioned qn-ils procedure for fluid-structure interaction problems: Filtering. *Computers & Structures*, 171:9–17, 2016.
- [26] Xiaowei He, Ning Liu, Guoping Wang, Fengjun Zhang, Sheng Li, Songdong Shao, and Hongan Wang. Staggered meshless solid-fluid coupling. ACM Transactions on Graphics (TOG), 31(6):149, 2012.
- [27] Gene Hou, Jin Wang, and Anita Layton. Numerical methods for fluid-structure interaction – a review. *Communications in Computational Physics*, 12(2):337–377, 2012.
- [28] Markus Ihmsen, Jens Cornelis, Barbara Solenthaler, Christopher Horvath, and Matthias Teschner. Implicit incompressible sph. *IEEE Transactions on Visualization* and Computer Graphics, 20(3):426–435, 2014.
- [29] Richard Keiser, Bart Adams, Dominique Gasser, Paolo Bazzi, Philip Dutré, and Markus Gross. A unified lagrangian approach to solid-fluid animation. In *Point-Based Graphics, 2005. Eurographics/IEEE VGTC Symposium Proceedings*, pages 125–148. IEEE, 2005.
- [30] Bryan M. Klingner, Bryan E. Feldman, Nuttapong Chentanez, and James F. O'Brien. Fluid animation with dynamic meshes. ACM Trans. Graph., 25(3):820–825, July 2006.
- [31] Ulrich Küttler, Christiane Förster, and Wolfgang A Wall. A solution for the incompressibility dilemma in partitioned fluid-structure interaction with pure dirichlet fluid domains. *Computational Mechanics*, 38(4-5):417-429, 2006.
- [32] Ulrich Küttler and Wolfgang A. Wall. Fixed-point fluid-structure interaction solvers with dynamic relaxation. *Computational Mechanics*, 43(1):61–72, 2008.
- [33] Patrick Le Tallec and Jean Mouro. Fluid structure interaction with large structural displacements. Computer Methods in Applied Mechanics and Engineering, 190(24):3039– 3067, 2001.
- [34] Jie Liu, Seiichi Koshizuka, and Yoshiaki Oka. A hybrid particle-mesh method for viscous, incompressible, multiphase flows. J. Comput. Phys., 202(1):65–93, January 2005.
- [35] Frank Losasso, Jerry Talton, Nipun Kwatra, and Ronald Fedkiw. Two-way coupled sph and particle level set fluid simulation. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):797–804, July 2008.
- [36] Wenlong Lu, Ning Jin, and Ronald Fedkiw. Two-way coupling of fluids to reduced deformable bodies. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium* on Computer Animation, pages 67–76. Eurographics Association, 2016.
- [37] J.J. Monaghan. Simulating free surface flows with sph. J. Comput. Phys., 110(2):399–406, February 1994.

- [38] Matthias Müller, David Charypar, and Markus Gross. Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '03, pages 154–159, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.
- [39] Matthias Müller, Simon Schirm, Matthias Teschner, Bruno Heidelberger, and Markus Gross. Interaction of fluids with deformable solids: Research articles. *Comput. Animat. Virtual Worlds*, 15(3-4):159–171, July 2004.
- [40] Rahul Narain, Abhinav Golas, and Ming C Lin. Free-flowing granular materials with two-way solid coupling. ACM Transactions on Graphics (TOG), 29(6):173, 2010.
- [41] Seungtaik Oh, Younghee Kim, and Byung-Seok Roh. Impulse-based rigid body interaction in sph. Comput. Animat. Virtual Worlds, 20(2-3):215–224, June 2009.
- [42] Sang Il Park, Hyun Joon Shin, and Sung Yong Shin. On-line locomotion generation based on motion blending. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics* Symposium on Computer Animation, SCA '02, pages 105–111, New York, NY, USA, 2002. ACM.
- [43] Peter Raback, Juha Ruokolainen, Mikko Lyly, and Esko Järvinen. Fluid-structure interaction boundary conditions by artificial compressibility. In *ECCOMAS CFD*, volume 2001, 2001.
- [44] Karthik Raveendran, Chris Wojtan, and Greg Turk. Hybrid smoothed particle hydrodynamics. In Proceedings of the 2011 ACM SIGGRAPH/Eurographics symposium on computer animation, pages 33–42. ACM, 2011.
- [45] Avi Robinson-Mosher, Craig Schroeder, and Ronald Fedkiw. A symmetric positive definite formulation for monolithic fluid structure interaction. *Journal of Computational Physics*, 230(4):1547–1566, 2011.
- [46] Avi Robinson-Mosher, Tamar Shinar, Jon Gretarsson, Jonathan Su, and Ronald Fedkiw. Two-way coupling of fluids to rigid and deformable solids and shells. ACM Trans. Graph., 27(3):46:1–46:9, August 2008.
- [47] X Shao, Z Zhou, Nadia Magnenat-Thalmann, and W Wu. Stable and fast fluid-solid coupling for incompressible sph. In *Computer Graphics Forum*, volume 34, pages 191– 204. Wiley Online Library, 2015.
- [48] B. Solenthaler and R. Pajarola. Predictive-corrective incompressible sph. ACM Trans. Graph., 28(3):40:1–40:6, July 2009.
- [49] Barbara Solenthaler, Jürg Schläfli, and Renato Pajarola. A unified particle model for fluid-solid interactions. Computer Animation and Virtual Worlds, 18(1):69–82, 2007.
- [50] Alexey Stomakhin, Craig Schroeder, Chenfanfu Jiang, Lawrence Chai, Joseph Teran, and Andrew Selle. Augmented mpm for phase-change and varied materials. ACM Trans. Graph., 33(4):138:1–138:11, July 2014.

- [51] Yun Teng, David I. W. Levin, and Theodore Kim. Eulerian solid-fluid coupling. ACM Trans. Graph., 35(6):200:1–200:8, November 2016.
- [52] Adrien Treuille, Andrew Lewis, and Zoran Popović. Model reduction for real-time fluids. ACM Trans. Graph., 25(3):826–834, July 2006.
- [53] Jan Vierendeels. Implicit Coupling of Partitioned Fluid-Structure Interaction Solvers using Reduced-Order Models, pages 1–18. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.
- [54] Jan Vierendeels, Joris Degroote, Sebastiaan Annerel, and Robby Haelterman. Stability issues in partitioned fsi calculations. In *Fluid Structure Interaction II*, pages 83–102. Springer, 2011.
- [55] Jan Vierendeels, Lieve Lanoye, Joris Degroote, and Pascal Verdonck. Implicit coupling of partitioned fluid-structure interaction problems with reduced order models. *Computers & structures*, 85(11):970–976, 2007.
- [56] Andrew Witkin. Physically based modeling: principles and practice constrained dynamics. *Computer graphics*, 1997.
- [57] Omar Zarifi and Christopher Batty. A positive-definite cut-cell method for strong two-way coupling between fluids and deformable bodies. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, page 7. ACM, 2017.