UC Berkeley
UC Berkeley Electronic Theses and Dissertations

Title
High Order Implicit Shock Tracking using an Optimization-Based Discontinuous Galerkin Method

Permalink
https://escholarship.org/uc/item/0rw139xf

Author
Shi, Andrew Justin

Publication Date
2022

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

High Order Implicit Shock Tracking using an Optimization-Based Discontinuous Galerkin Method

by

Andrew Shi

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Applied Mathematics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Per-Olof Persson, Co-chair
Assistant Professor Matthew Zahr, Co-chair
Professor Ming Gu
Assistant Professor Michael Lindsey

Fall 2022

High Order Implicit Shock Tracking using an Optimization-Based Discontinuous Galerkin Method

Copyright 2022
by
Andrew Shi

Abstract

High Order Implicit Shock Tracking using an Optimization-Based Discontinuous Galerkin Method

by

Andrew Shi

Doctor of Philosophy in Applied Mathematics

University of California, Berkeley

Professor Per-Olof Persson, Co-chair

Assistant Professor Matthew Zahr, Co-chair

We present a framework for resolving discontinuous solutions of conservation laws using *implicit tracking* and a high-order discontinuous Galerkin (DG) discretization. Central to the framework is an optimization problem and associated sequential quadratic programming solver which simultaneously solves for a discontinuity-aligned mesh and the corresponding high-order approximation to the flow that does not require explicit meshing of the *a priori* unknown discontinuity surface. We utilize an error-based objective function that penalizes violation of the DG residual in an enriched test space, which endows the method with r-adaptive behavior: mesh nodes move to track discontinuities with element faces and improve the conservation law approximation in smooth regions of the flow. This method is shown to deliver highly accurate solutions on coarse, high-order discretizations without nonlinear stabilization and recover optimal convergence rates $O(h^{p+1})$ for problems with discontinuous solutions. We demonstrate this framework on a series of inviscid steady and unsteady conservation laws, the latter of which using both a space-time and method of lines discretization. We also develop local mesh operations for curved meshes that are required to maintain mesh and solution quality as our high-order meshes deform to track the shock.

Contents

List of Figures

List of Tables

Acknowledgments

I have spent almost my entire adult life in Berkeley. Despite our differences (both with university and city), I will always be grateful for my time here. I came into graduate school with a cautious optimism, and correctly so. The pursuit of a research degree can be difficult, isolating, and hopeless. Years can be spent attempting to prove a theorem that turns out to be false. Countless hours spent debugging a code that returns complete nonsense, not knowing if you are just a small bug fix away or your method actually doesn't work. Nobody knows until you figure it out, or figure out why you can't. Such as the true nature of research that everyone should (in theory) come to appreciate before earning the degree. But in many ways, it is extremely rewarding ways both mathematical and otherwise. I can count on one hand the number of "successful" days I've had as a graduate student, yet instead of feeling discouraged, I for whatever reason felt the urge to come back for more. Put it in writing like this and it makes me seem like a masochist. Maybe that's just how it's supposed to be. Before my time in graduate school, I believed research was just about results. After all, nobody publishes their failures and the media doesn't report on "almost" breakthroughs. But my time here taught me to appreciate the process. The world doesn't stop for research; life goes on as usual, with its many challenges. But through it all, I always felt the support, encouragement, and friendship of the people I met during my time at Berkeley, and it is my great pleasure to acknowledge them here.

For most students, the most important factor that determines the quality of their experience in graduate school is their relationship with their advisor. I can easily say that I hit the jackpot in this regard with my advisor Per-Olof Persson. From him, I learned many important lessons. An early one was how to maximize productivity by procrastination. My first few projects hit quite a few dead ends. Everybody gets stuck, but then there's *really* stuck. So stuck, I could spend an entire semester of effort working on a "baby project" only to offer up a concrete explanation why our proposed approach was doomed to fail (and be forced chalk that up as a relative success). After a few moments of contemplation, he would look at me and say "I think you're right. Maybe it is *fundamental*" (the memory of which still sends chills down my spine). To me this used to seem like a death sentence, but without missing a beat, Persson would exclaim that "This is good news! Where you see a challenge, I see opportunity." At the time, I can only describe my feelings as a combination of being annoyed to no end by his attitude and so discouraged, that if I let my head hang on the way out of his office, I could make a dent in the floor. But over time, much to his credit, his infectious and unrelenting optimism came to rub off on me. I have always looked forward to our meetings where I would set up shop in his office, present my numerical results on his big monitor, and ramble on about my progress (or lack thereof) as he would patiently listen, offer comments, and draw pictures. I can say that almost every single time I left his office feeling re-energized with a whole slew of ideas and directions I couldn't wait to try out before we met again. He and I probably spent just as much (if not more) time discussing non-research things, mathematical, departmental, or really anything else. It has been a great pleasure and boon for me to interact with him as someone with whom I could always speak freely

and candidly. Beyond research, he imparted in me valuable insights on all things academic, such as how to give good talks or balance multiple responsibilities. Prof. Persson always looked out for opportunities for me whether it be new projects, collaborations, conferences, or grants. He always encouraged me to go in any direction I wanted, even if it did not necessarily align with his own. Without his guidance I would not have been able to develop my confidence and independence. After I hit a difficult and extended period in my personal life, he without fail helped me out in any way I asked of him, far beyond the scope of what can be expected from an academic advisor. I am proud to call him (in no particular order) my teacher, advisor, mentor, collaborator, advocate, co-instructor and friend.

In the same breath, I must mention my co-advisor Prof. Matthew Zahr. Our early interactions can be described as him enthusiastically and patiently explaining all of his various ideas on how to develop this shock tracking method on his extremely busy (but perfectly legible) whiteboard at the lab while I would stare blankly at him for an hour, only for me to come back a month later and ask him what a Jacobian was. But Matt's near limitless patience and availability as I fumbled through the early years eventually paid off, and I am very grateful to him for that. Matt is the ideal combination of mentor and collaborator, letting me find my own way and directions, while at the same time helping me clear unnecessarily cumbersome obstacles and gently reign me in when I've gone too far off. I have always been in awe of the sheer breadth and depth of his mathematical and technical knowledge as well as his academic savvy at such a young age. Just like with Prof. Persson, it makes life much easier when you can work with someone you are almost always on the same page with and can discuss potentially difficult issues very candidly. Matt's attention to detail and willingness to make large investments in developing systems such as his codebases rather than taking shortcuts or settling for less reflects on his dedication to his craft, his genuine investment in his students, and his integrity as a researcher and a person. It is because of him and his extremely well designed LaTeX templates that I am probably the only student filing their dissertation this semester to eschew the standard template and simply call my thesis "pap.tex". The surprisingly large amount of bureaucratic hurdles I had to clear to get him formally acknowledged as my co-advisor was easily worth it, because now I have the honor of calling myself his first PhD student.

I don't seem to recall learning all that much math from either Matt or Per. Whatever I did learn, pretty soon I'll certainly forget most of it. But I can say I got a genuine taste of what it really means to do research from the two of you. To not be discouraged or give up at the first sign (or major signs, plural) of trouble, be creative and independent, balance short term practical needs with long term visions, and maintain integrity and transparency while still "playing the game". It is great to work with people who embody a lot of the characteristics and goals that you have for yourself; they remind you that it's all possible. I will miss working with the two of you immensely.

I have had the great pleasure of interacting with many faculty during my decade+ at UC Berkeley. I first want to thank Prof. Alexandre Chorin. It is only through a very late Phase I Telebears appointment I reluctantly enrolled in his Numerical Analysis course in the Spring of 2013 my freshman year that met MWF at 8am (!!). But without fail I would

show up at 7:55am, and each day around 8:08am, he would neatly fold up his jacket, put away his newspaper, and began lecturing from memory and filling up the four chalkboards in LeConte 4. I would always attempt to follow his arguments with great interest, even that week I fell off the top bunk and sprained both my ankles (and then some). I recall fairly clearly one of our earliest personal interactions, when I made the trek to his office to seek an administrative signature approving my planned course schedule. I tapped on his closed door and he curtly informed me that his office hours had just ended and I had misread them. As I prepared to leave in embarrassment, he graciously waved me inside so I could get my signature. After a few minutes of silence as he examined my record, he asked me about my post-graduation plans and I offered up some vague response about finding a well-paying job. He then asked me if I had ever considered going further in mathematics, and I attempted to delicately reply that mathematics was not for me. When he asked why not, I thoughtlessly blurted out something to the effect of "What is there even left to do?". Almost a decade later, I remember very little about his actual answer, but what stuck with me was how much he had to say in response; the sudden excitement my näıve question sparked in this usually soft-spoken man. I can say that day, the seed was planted in my mind. Prof. Chorin piqued my interest in the subject and was supportive and encouraging of my decision to pursue graduate studies in the field. I owe a great debt to him as do countless other students he has advised and mentored over the years, and it is my great fortune that the beginning of my academic career could overlap with the end of his fifty year teaching career. I thank Prof. John Strain, from whom I've taken many courses (and taught just as many for). Prof. Strain has always given me excellent character building advice and challenges over the years, and I have always enjoyed working and interacting with him a great deal. Prof. Jon Wilkening has always been a friendly face in the department who I can count on for an encouraging word (and who I've long suspected was the only member on the admissions committee that year who wanted to admit me at all). I have always been in awe of his ability to take on many teaching, research, and service responsibilities (not to mention small children!), all while keeping his cool. Prof. Ming Gu has always been someone who I have found very easy to talk to, and even though we work in very different fields, I appreciate his confidence in me through his invitation to conduct research with him on his RRQR method and his last minute service on my dissertation committee. Even though Prof. Craig Evans and I did not interact as much as I would have liked, I thank him for his courses, his service on my qualifying committee, and graciously autographing my copy of his textbook that I can show off for generations to come. While his research and pedagogical accomplishments speak for themselves, I have always in particular been inspired by his commitment to education, pedagogy, and the mission of the University of California as a public institution and the vital role our department plays in that mission. Thanks to Prof. Shawn Shadden for serving on my qualifying exam committee and teaching some excellent courses that provided my first exposure to the finite element method. Prof. Michael Lindsey gets special mention, due to his recent ascent from the "friends" to the "faculty" section of these acknowledgements. I call everyone Prof. Last Name, but to me, he is and also will be Mike, even after he gets elected to the National Academy of Science. I never would have expected the 2nd year

graduate student that volunteered to host a prospective student (only because his roommate was out of town and there was a $50 Amazon gift card in it for him, so why not?) would end up serving on my dissertation committee. I also credit him for dispelling some of my misgivings about returning to Berkeley for a second tour of duty after a two-year gap. I have always greatly admired him as a scientific role model and more importantly as a person, and I consider it a great honor to have been the second most frequent visitor to the earliest and most primitive versions of your personal webpage (only behind your mother) as I had the pleasure of watching your early career rapidly ascend. But more importantly, I thank you for your friendship and support in your own second tour of duty at Berkeley as well as your willingness to help me out at the last minute without the slightest hesitation to get this dissertation through the finish line.

I also want to thank our wonderful staff. A now-retired professor once described his year long sabbatical at a company with the following quote:

> Amazingly, I came across no individual here who was less than competent at their job. This contrasts with my experience in academia, where 5% of employees should have been fired years ago.

This seems like an appropriate way to characterize the amazing staff in the math department. I first want to thank Jennifer Sixt-Pinney. She graciously picked up my call to help me sort out all of my course equivalencies and other unique situations before I had even arrived on campus as a freshman in Fall 2012, even though looking back it should have been my responsibility to Google 75% of the answers myself. For many years, Jennifer has been the glue that held the department together through her tireless efforts (only she knows where all the bodies are buried), and I and countless others have certainly benefited from her work from behind the scenes. There have been many instances of departmental gossip that only she and I could fully appreciate and discuss in hushed tones behind a barely cracked open door, and I wish her a happy and well deserved retirement. I also want to thank Marsha Snow. For many years, she always confused me some other guy who competed on the Jeopardy College Championship. But eventually, she figured out who I was and throughout the years I would always find an excuse to drop by the front office so we could chat, despite her and Jennifer's independent ridicule of me as "cowgirl". Thank god she finally listened to my advice and got rid of those awful Tootsie Rolls she used to stock and replaced them with Jolly Ranchers. It doesn't matter how many sizes or flavors they're available in Marsha, nobody likes them besides you! I also want to thank Jon Phillips. I don't think he and I have ever met face to face, but he was extremely helpful and responsive in helping me get this dissertation through the final stages. There were many strange bureaucratic hurdles that I believe neither of us had ever seen before, quite a few of which were caused by my own negligence and last minute change of plans, but Jon skillfully and deftly helped me navigate through all of them. Most of the time, my interaction with the graduate advisor doesn't amount anything personal, usually just filling out some Google form to enroll in units or get a GSI assignment. But Jon showed me that during crunch time, he is more than "just

a advisor", but a true advocate for all the graduate students in the department, and for that I will always be very grateful to him. I thank Vicky Lee, both from her previous role as graduate advisor, and now as her new position of Director of Student Services. I don't know what the official job description of the latter title entails, but both of us know this makes her the new keeper of all knowledge and responsibility in the department. I very clearly remember my first impression of Vicky during the graduate orientation, where she said something to the effect of "Cheesy as it may sound, I truly feel that my calling is to be here at the University and help students succeed." Years of my own observation have proven there is quite a lot of substance behind that cheesy statement. Vicky is the one to wear almost all the hats in the department, going above and beyond to pick up the slack and take on and take on any responsibility to help students succeed. Anything from working with a particular individual on their own situation to or some broader initiative that may impact the academic outcomes of many students, despite any number of uphill climbs or thankless tasks it takes. All while making it look easy (though I know it's anything but) and being incredibly approachable. I am grateful for our interactions and her help and support over the years. Finally, I also want to thank all the other members of our staff who I may not have directly interacted with as much, but I know are responsible for contributing to my experience here in one way or another. Ask any of the hundreds of students who have had the (mis)fortunate of having me as their GSI. I don't think I am one to give out praise very lightly or frequently. but the staff of the math department at the University of California, Berkeley is truly world class.

I want to thank all of the great friends I made at Berkeley. In theory, I should have been constantly discussing high level mathematics with my classmates at one of the top schools to study mathematics until the late hours of the night like some great melding of the minds. There were indeed many discussions until the late hours of the night (and early morning), but I don't recall math ever being mentioned. I wouldn't change anything about it.

First I want to thank all of my officemates, official and unofficial. First up is my only officemate of five years, Tong Yu. Even though I never really figured out what he does beyond my expert parroting of buzzwords (DMET/DMFT, Hartree-Fock *like...*), I have always admired him as a scientist and mathematician. He is a humble and truly capable person, and I can only hope the multiple donuts and sweets I split with him over the years will make me eligible for a hefty discount on the quantum computers I expect him to make breakthroughs in. I appreciate his (reluctant) willingness to let me bogart our shared office space to store all sorts of junk, as well as his selfless sacrifice in giving up an office with an window so I could keep my couch (not to mention his consideration in dimming the lights for me during the multiple times he caught me napping during business hours). I want to thank my first set of "nighttime" office mates Charles Wang, Angxiu Ni, and Chan Bae. Even though Chan Bae gave me terrible investment advice and our petty squabbles behind closed doors as he played Clash of Clans frequently echoed through the hall, I am (somewhat) grateful for the short time we spent together, nearly 7 days a week, within 15 feet from each other. And with grandpa Angxiu Ni, we formed the original "party four" (see [51]). I recall our many nights eating "Lavar's", discussing which acronyms to use for his latest algorithm,

and his inability to recite more US states than NBA teams (Golden State is still not a state), fondly. I also thank my second set of "daytime" officemates Michael Franco, Yanhe Huang, and Luke Corcos. Finding any any $\binom{3}{k}, k = 1, 2, 3$ subset of them, we could discuss anything from academic and non-academic career paths, to teaching Yanhe "real world English", and the fatal fantasy implications of drafting Mr. Big Chest with your first round pick. But of course, the case of $k = 3$ allowed us to form what the recent literature described as the "office party" (see [33]). Considering I spent many years lurking as an ever present shadow in the dimly lit halls of Evans and rarely changing sweaters, I would hardly consider myself a party person. But I am certainly flattered (if not bemused) to remembered as such.

Thanks to Nick Bhattacharya, who I found a distinct shared sense of humor few do in this lifetime, even though just seeing his name evokes the clear memory of him wolfing down those disgusting Popeyes biscuits on two-piece Tuesday. I want to thank boy wonder Xiaohan Yan. You might have been part of the gifted youth class at Peking University, but that doesn't mean squat when it comes to your complete lack of appreciation of the skill it takes to play Spider Solitaire with more than one suit. I thank Yao Yuan and Irit Huq-Kuruvilla for our multiple excursions to eat something far away from campus. But Yao Yuan I can only thank reluctantly, because that final excursion to Great China with the soggy duck skin still leaves a bad taste in my mouth. Actually, it's a wonder it left any taste in my mouth, because in the time it takes me to delicately assemble one duck pancake, you have already shoved five in your face hole. Splitting the bill equally with you has easily the biggest scam in Berkeley, and eventually Irit and I will find a way to get even. I acknowledge James Dix for our multiple Google maps tours of fast food eateries in remote college towns and our shared quest to find of high-quality Picanha in the bay area at bargain basement prices. If either of us gain any political influence in Berkeley, I know our first move would be to tear down Chez Panisse (Michelin stars be damned) without the slightest hesitation and replace it with a Sonic drive-through. Max Hlavacek has always been a source of cheerful optimism. I could always count on her to spy on me while I was absorbing sunlight on the patio, just as she could always count on me provide last minute lodging on my air mattress when she gets locked out of her office late at night. (I believe as of this writing, we are still the only two of our cohort to spend the night in Evans).

~~Anth~~ony Villafranca is the original co-host of late night in Evans and fellow prisoner during the COVID times in Cynthia's den. We always had a good time together, from "take a little disk" to "I saw a deer" and all the other random stuff in between. Here's to Tugba, $\frac{\partial u}{\partial t} = \Delta u$, Grimace, big boy and his wife, 35 year old, seasons 1-7 Lorelai (but only seasons 1-3 Rory, Yale just ruined her), Erika Cheung, Bibo or Biji or whatever, the middle eastern guy from Office Space that says "Peter, If you're so obsessed with her, why don't you just .... ask her out?", which reminds me of that cute waitress at Soi Four you blew any chance with because you looked particularly homeless that day and your card was declined, and all the other nonsense characters that only exist in our own demented world. (But here I have to admit I don't nearly like Thai food as much as you do, but since you had a car...)

Out of our cohort of thirty, Adele Padgett and I are probably the most different. Gender, ethnicity, appearance, personalities, social inclinations, mathematical interests, choice of

restaurants, and ability to hold our booze or lie in the sun for extended periods of time (exercise to the reader to guess who is better at which). In theory I don't think we should have been friends. But in reality, I have come to value her friendship a great deal over the years. She was someone I could always confide in and came to trust, and we could always healthily disagree without ever getting into a fight. As much as I will miss you, I know someone from Berkeley who will miss you even more than I. Because now that you are Canadian, Hotsy Totsy will need a small business loan to keep their doors open.

I thank Roy Zhao (@GSIRoyZhao) for always being someone I can count to help me out with anything and commiserate with on just how damn hard it is to get anything done in graduate school. Roy is a genuinely good guy who I admire for his dedication to his friends, students, and the department. In fact, you are such a good guy, you were the first person to rush to get those gift cards "Martin Olsson" urgently requested. Thanks for always sharing your questionably obtained URep haul with me (impeach Larsen). Once you finally find the next largest prime number (or whatever it is you actually do), I am sure 爸爸 will welcome back his prodigal son.

And finally, I thank my friend Charles 'les Wang (@c). Out of everyone on this list, I have easily spent the least amount of time in your physical presence because you decided to abandon us for *fancy-schmacy Harvard* after a single year. But you have been the one I have interacted with the most during my time in graduate school. Looking back, year one of graduate school was easily the most enjoyable. Probably because I don't seem to recall actually doing a single thing of value towards the pursuit of my graduate degree. What I do recall is spending countless hours in your freezing office watching you play as sillibrij, as well as the countless hours in my much nicer office I went all in on the river with nothing but 2♣ 9♡ high, saying that I'm sure this "jerk from Estonia" is going to fold. Only to get snapped off, reload, and repeat, completely undeterred by my own stupidity and stubbornness. I don't think I ever got a cavity filled on my back left molar from what I suspect resulted from our personal hoarding of multiple 5lb bags of candy after we lost faith in the previous model of communism due to certain bad actors (who are also mentioned in these acknowledgements but I will not call out by name.) My once sharp memory has become very fuzzy in the last few years, but I still remember clearly your last day at Berkeley. After a night in your office hiding from @gny and a disappointing trek to Long Life Vegi House, I lament that I only got to feel just how soft your mermaid hair was we said goodbye (We both know exactly why I had good reason to assume it was dirty). While the road ends for me here in the pursuit of our shared dream that formed the basis of our strong friendship over the last few years, I wish you the best of luck and hope you get everything you deserve, Wanglands and more. By the way, can you explain one more time exactly "hwat in the hell" a Grassmanian is?

In closing, there are far more important things in life than research, high-order implicit shock tracking or pretty much anything else. Some of us are lucky and never have to learn this lesson. Some of us learn this lesson just a little bit too late. To my late mother and her late mother, this dissertation is nothing but a small monument to my tremendous selfishness.

为了写这本天书，把你们俩送到天堂。

Chapter 1

Introduction

It is widely believed that higher fidelity is required for problems with propagating waves, turbulent fluid flow, nonlinear interactions, and multiple scales [75]. This has resulted in a significant interest in high-order accurate methods, such as discontinuous Galerkin (DG) methods [7, 31], which have the potential to produce accurate solutions on coarse meshes. Among the most significant challenges associated with high-order methods is their sensitivity to under-resolved features, in particular for nonlinear problems where the spurious oscillations often cause a breakdown of the numerical solvers. This is exacerbated for problems with shocks where the low dissipation associated with high-order methods is insufficient to stabilize the solution. Since shocks are present in many important problems in fields such as aerospace, astrophysics, and combustion, this poses a fundamental barrier to widespread adoption of these methods. Many solutions have been proposed, but most require excessively refined meshes to resolve all the features, which in practice makes it difficult to accurately predict high Reynolds, high Mach flows that feature shocks, boundary layers, and interactions between them. Therefore, new advances are required to make high-order schemes sufficiently robust and competitive for real-world problems. We first provide a simple motivation of our proposed approach to remedy these issues in context of the existing methods and issues (Section 1.1), and then go into more detail on the related and current state-of-the-art methods and discuss their relative advantages and drawbacks (Section 1.2).

1.1 Motivation

Consider a simple two element mesh in 1D (Figure 1.1). Typical numerical methods (such as the Finite Element Method) rely on a set of smooth polynomial basis functions defined on these elements (Figure 1.2) for numerical approximation. However, when the underlying function to be approximated is a discontinuity (Figure 1.3), a fundamental issue arises when you attempt to use a set of smooth basis functions. It is clear this does not approximate the discontinuity very well, not to mention the fact the approximation takes on a negative value, which is a significant problem when estimating physical quantities such as density for

which this makes no sense.

Figure 1.1: A two element mesh in 1D.

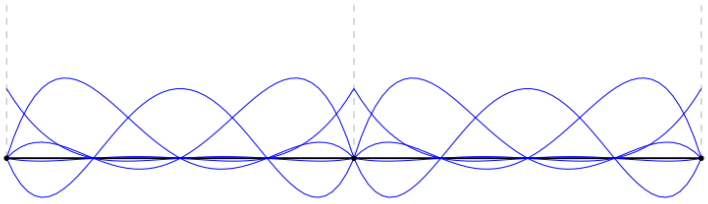


Figure 1.2: A polynomial basis on the mesh from Figure 1.1.

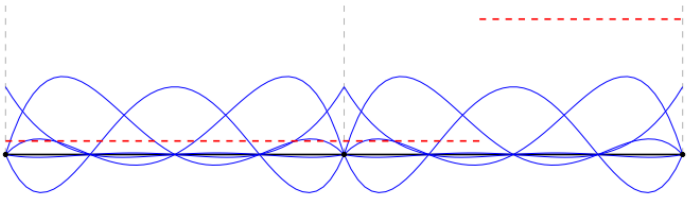


Figure 1.3: Polynomial basis from Figure 1.2 along with a discontinuous function.

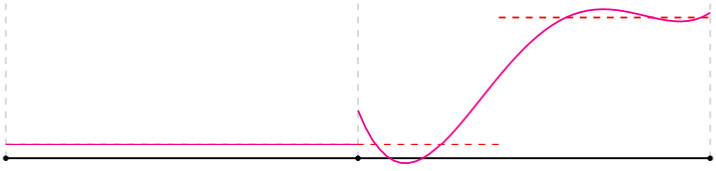


Figure 1.4: A numerical approximation of the discontinuous function from Figure 1.3 using a polynomial basis.

One approach is known as limiting (Section 1.2.1), but this has the drawback of order reduction (Figure 1.5), which is undesirable when the demand for high-order and high-fidelity solutions (especially along features such as discontinuities) are at an all time high. This approach can be augmented by local mesh refinement (Figure 1.6), but it doesn't really

do much to improve on the crux of the issue, which is high-order accuracy in the presence of a discontinuity.

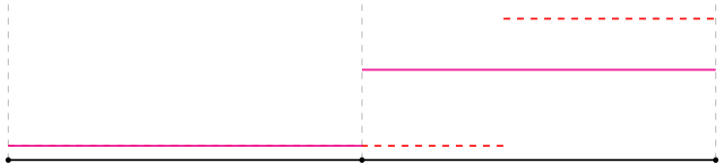


Figure 1.5: Approximation to the discontinuous function by limiting.

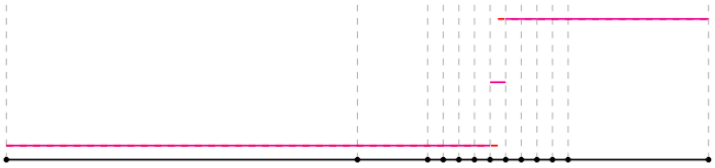


Figure 1.6: Approximation to the discontinuous function by limiting and mesh refinement.

The most popular approach today for stabilizing shocks is known as shock capturing (Section 1.2.2), in particular artificial viscosity. Rather on relying on the dissipation inherent in the numerical scheme (such as the discontinuous Galerkin approximation), an artificial viscosity term is added which is aimed at eliminating the high frequencies in the solution, thus eliminating Gibbs-type oscillations (Figure 1.7), which can be reflected in the corresponding numerical approximation (Figure 1.8).

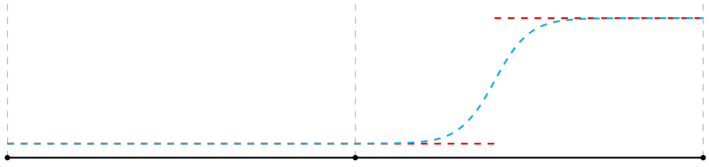


Figure 1.7: Artificial viscosity added to the discontinuous function.

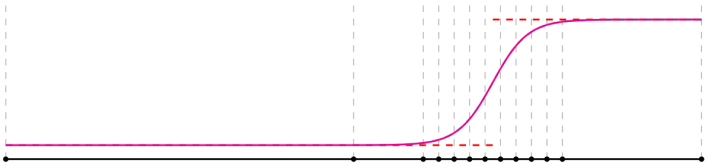


Figure 1.8: Numerical approximation to the viscosity enhanced function in Figure 1.7.

Revisiting our discontinuous function (Figure 1.9), in this thesis we present an alternative approach for the numerical approximation of discontinuous functions. Shock tracking (or shock fitting), aims to align the features of the solution basis with features in the solution using an optimization formulation and solver. In this simple example (Figure 1.10), we can see that our numerical solution can now exactly represent the function by aligning the mesh to the discontinuity and using smooth basis functions to represent the solution elsewhere.

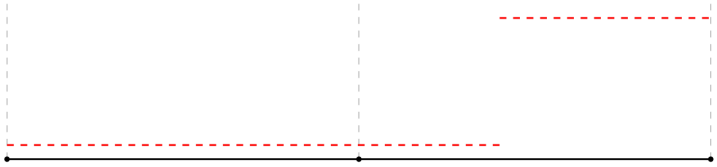
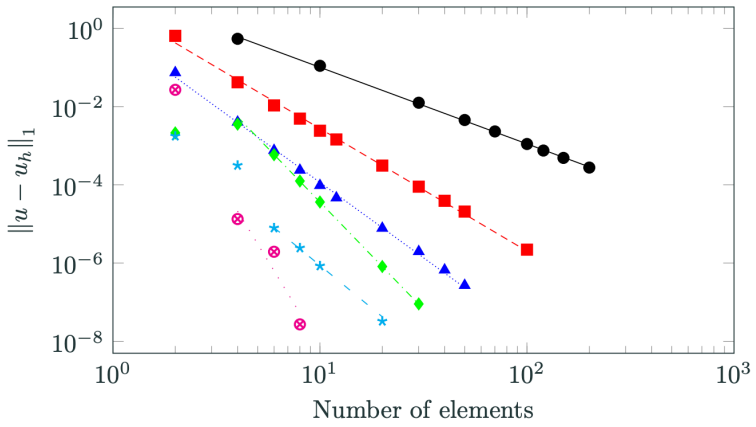


Figure 1.9: The discontinuous function on the original mesh from Figure 1.1.

Figure 1.10: Mesh aligned numerical solution with the discontinuous function.

Such an approach has been shown to have many key advantages, two of which are the ability to obtain optimal convergence rates of $O(h^{p+1})$ *even in the presence of discontinuous solutions* (Figure 1.11) [79], as well as gain orders of magnitudes of improvement in comparison to adaptive mesh refinement procedures (Figure 1.12).



Convergence of implicit shock tracking (Burgers' equation) with polynomial degrees $p = 1$ (●), $p = 2$ (■), $p = 3$ (▲), $p = 4$ (♦), $p = 5$ (*), $p = 6$ (⊗).

Figure 1.11: A demonstration of high-order convergence of implicit shock tracking (See [79] for details).

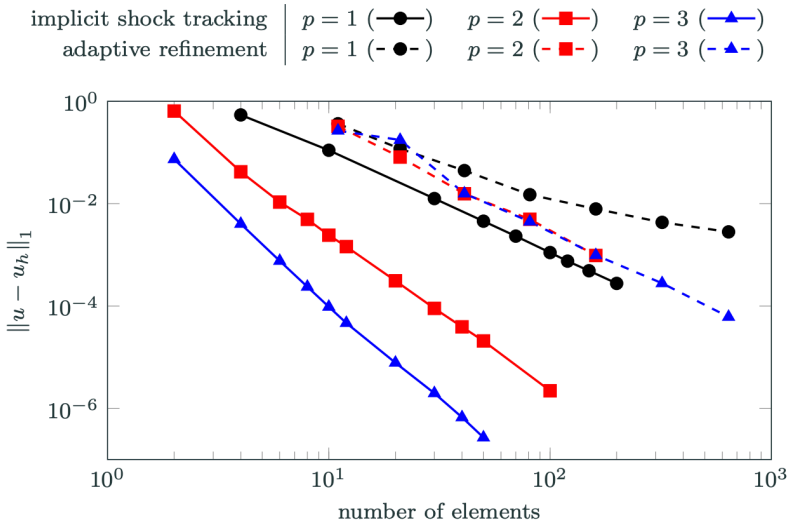


Figure 1.12: Convergence of implicit shock tracking (Burgers' equation): implicit shock tracking (solid) vs. adaptive mesh refinement (dashed).

In this work, we propose an approach to shock tracking that does not explicitly require generating a mesh of the unknown discontinuity surface. Rather, the conservation law is

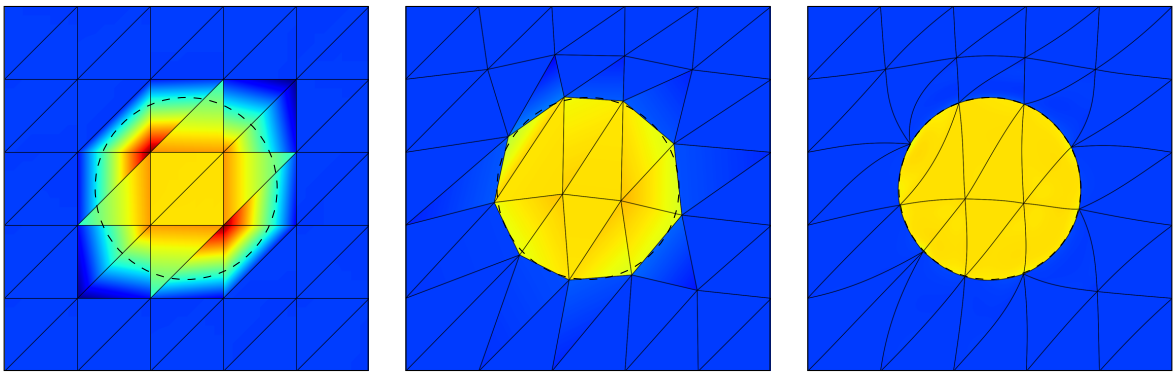


Figure 1.13: L^2 projection of a piecewise constant function onto a non-aligned (*left*) vs. discontinuity-aligned mesh with linear (*middle*) and cubic (*right*) elements and corresponding solution. See Example 4.1 of [79] for details.

discretized on a mesh without knowledge of the discontinuity surface and an optimization problem is formulated such that its solution is the pair $(\boldsymbol{u}, \boldsymbol{x})$, where $\boldsymbol{x}$ is the positions of the mesh nodes that cause element faces to align with discontinuities in the flow and $\boldsymbol{u}$ is the solution of the discretized conservation law on the mesh defined by $\boldsymbol{x}$. That is, discontinuity tracking is implicitly achieved through the solution of an optimization problem and will be referred to as *implicit shock tracking*. While this approach works with any discretization that allows for inter-element discontinuities, we focus on high-order DG methods due to the high degree of accuracy attainable on coarse meshes, proper treatment of discontinuities with approximate Riemann solvers, and the ability to use curved elements to track discontinuities with curvature (Fig. 1.13). The implicit tracking optimization problem proposed here minimizes the violation of the DG residual in an enriched test space while enforcing that the standard DG (same test and trial space) equations are satisfied. This objective function is a surrogate for violation of the infinite-dimensional weak formulation of the conservation law, which endows the method with r-adaptive behavior: it promotes alignment of the mesh with discontinuities and adjusts nodes in smooth region to improve approximation of the conservation law. The optimization problem is solved using a sequential quadratic programming method with a Levenberg-Marquardt Hessian approximation that simultaneously converges the mesh and solution to their optimal values, which never requires the fully converged DG solution on a non-aligned mesh and does not require nonlinear stabilization. The combination of implicit tracking with a DG discretization leads to a high-order accurate numerical method that has been shown to provide accurate approximations to high-speed inert [79, 81, 63] and reacting flows [80].

1.2 Related approaches

1.2.1 Limiting

One simple method is to use a sensor that identifies the mesh elements in the shock region and reduce the degree of the approximating polynomial [3, 6]. A more sophisticated approach includes limiting, such as the weighted essentially non-oscillatory (WENO) schemes [30, 45, 35], which perform a high-order reconstruction near discontinuities, but require a large computational stencil which offsets the attractive compactness properties of DG methods. For a more detailed discussion about the relative merits and drawbacks of limiting and ENO/WENO schemes in comparison to artificial viscosity, see the introduction of [56].

1.2.2 Shock capturing

For high-order methods, artificial viscosity has also proven to be competitive, since it can smoothly resolve the jumps in the solution without introducing additional discontinuities between the elements [56]. Indeed, artificial viscosity has been huge in the success of making progress towards previously intractable problems such as unsteady Large Eddy Simulation and Aeroacoustics. A recent comparative study of artificial viscosity models [77] discusses their relative merits, but conclude all of these artificial viscosity models suffer from a relatively strong dependency on a large number of empirical parameters which must be tuned. Parameter tuning notwithstanding, the main problem with all these approaches is they lead to globally first-order accurate schemes. This can be remedied by local mesh refinement around the shock (h-adaptivity) [14], although the anisotropic high-order mesh adaptation is challenging and requires highly refined elements near the shock. This issue is further complicated by transient problems where shocks and other local features propagate throughout the domain, requiring online adaptivity to be computationally feasible. Even with aggressive h- and p-adaptivity, simulations of complex, shock-dominated, unsteady flows are extremely challenging and expensive.

1.2.3 Explicit shock tracking/fitting

An alternative approach is *shock tracking* or *shock fitting*, where the computational mesh is moved such that its faces are aligned with the discontinuities in the solution. This is natural in the setting of a DG method since the numerical scheme already incorporates jumps between the elements and the approximate Riemann solvers employed on the element faces handle the discontinuities correctly. However, it is a difficult meshing problem since it essentially requires generating a fitted mesh to the (unknown) shock surface. Many previous approaches employ specialized formulations and solvers which are dimension-dependent and do not easily generalize [29, 28, 4] and/or are limited to relatively simple problems [65, 66, 71]. In addition, early approaches to shock fitting have been applied to low-order schemes where the relative advantage over shock capturing is smaller than for high-order methods

[69, 2]. Explicit shock tracking strategies [58, 48] have been proposed which are able to attain high-order accuracy in the presence of shocks, but require a specialized strategy to explicitly track the shock separately from the remainder of the flow. These methods are not easily applicable to discontinuities whose topologies not known *a priori*. For these reasons, shock tracking is largely not used in practical CFD today.

Chapter 2

Governing equations and high-order numerical discretization

Consider a general system of m inviscid conservation laws, defined on the fixed domain $\Omega \subset \mathbb{R}^d$ and subject to appropriate boundary conditions,

$$\frac{\partial U}{\partial t} + \nabla \cdot F(U) = 0 \quad \text{in} \quad \Omega \times [0,T] \tag{2.1}$$

where $U : \Omega \times [0,T] \to \mathbb{R}^m$ is the solution of the system of conservation laws, $F : \mathbb{R}^m \to \mathbb{R}^{m \times d}$ is the flux function, $\nabla := (\partial_{x_1}, \ldots, \partial_{x_d})$ is the gradient operator in the physical domain such that $\nabla W(x,t) = \begin{bmatrix} \partial_{x_1} W(x,t) & \cdots & \partial_{x_d} W(x,t) \end{bmatrix} \in \mathbb{R}^{N \times d}$ for any $W : \Omega \times [0,T] \to \mathbb{R}^N$ and $x \in \Omega$, $t \in [0,T]$, and the boundary of the domain $\partial\Omega$ has outward unit normal $n : \partial\Omega \to \mathbb{R}^d$. The conservation law in (2.1) is supplemented with the initial condition $U(x,0) = \bar{U}(x)$ for all $x \in \Omega$, where $\bar{U} : \Omega \to \mathbb{R}^m$. In particular, we will also consider the special case of

$$\nabla \cdot F(U) = 0 \quad \text{in} \quad \Omega, \tag{2.2}$$

which is sufficiently general enough to encapsulate steady conservation laws in a d-dimensional spatial domain or time-dependent conservation laws in a $(d-1)$ dimensional domain, i.e., a d-dimensional space-time domain.

In general, the solution $U(x)$ may contain discontinuities, in which case, the conservation law (2.2) holds away from the discontinuities and the Rankine-Hugoniot conditions [47]

$$F(U^+)n = F(U^-)n \tag{2.3}$$

hold for $x \in \Gamma_s$, where $\Gamma_s \subset \Omega$ is a surface along which U is discontinuous, $U^+(x), U^-(x) \in \mathbb{R}^M$ are the values of $U(x)$ on either side of the discontinuity, and n is a normal vector to the surface Γ_s.

We will construct a high-order numerical method that tracks discontinuities with the computational grid as they evolve through the domain, which places three requirements on the discretization: 1) a high-order, stable, and convergent discretization of the conservation

law in (2.1), 2) employs a solution basis that supports discontinuities between computational cells or elements, and 3) allows for deformation of the computational domain. As such, our method is based on a standard high-order DG-DIRK discretization of an Arbitrary Lagrangian-Eulerian (ALE) formulation of the governing equations. We insist on a high-order discretization given their proven ability [79, 8] to deliver accurate solutions on coarse discretizations provided discontinuities are tracked. The extension to other discretizations that support inter-element discontinuities such as finite volumes, flux reconstruction, hybridizable DG, and the DG spectral element method are possible, but beyond the scope of this work.

The remainder of this section will detail the discretization of the conservation law (2.1) using DG such that it reduces to the semi-discrete form

$$\boldsymbol{r}(\dot{\boldsymbol{u}}, \boldsymbol{u}, \dot{\boldsymbol{x}}, \boldsymbol{x}) = \boldsymbol{0} \tag{2.4}$$

where $\boldsymbol{r} : \mathbb{R}^{N_{\boldsymbol{u}}} \times \mathbb{R}^{N_{\boldsymbol{u}}} \times \mathbb{R}^{N_{\boldsymbol{x}}} \times \mathbb{R}^{N_{\boldsymbol{x}}} \to \mathbb{R}^{N_{\boldsymbol{u}}}$, $\boldsymbol{u}$ is the semi-discrete representation of the conservation law state U, and $\boldsymbol{x}$ is the semi-discrete representation of the conservation law domain Ω (nodal coordinates of mesh nodes). We will then apply a high-order temporal discretization to (2.4) by a diagonally implicit Runge-Kutta method to yield a complete discretization of (2.1). The same discretization process can be used to yield a semi-discretization from an enriched *test* space and corresponding temporal discretization, which will be used in the definition of the proposed objective function. For emphasis, this discretization simplifies considerably in the case of the steady conservation law (2.2), in which all the time-dependent terms drop out and we are left with the semi-discrete form

$$\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{x}) = \boldsymbol{0} \tag{2.5}$$

where $\boldsymbol{r} : \mathbb{R}^{N_{\boldsymbol{u}}} \times \mathbb{R}^{N_{\boldsymbol{x}}} \to \mathbb{R}^{N_{\boldsymbol{u}}}$. For a treatment of Sections 2 and 3 considering only the steady case corresponding to 2.2 and 2.5, see [78].

2.1 Arbitrary Lagrangian-Eulerian (ALE) formulation of conservation laws

We use an ALE formulation of the governing equations to account for the time-dependent domain deformations required to track discontinuities as they evolve. To this end, we introduce a time-dependent domain mapping (Fig. 2.1)

$$\mathcal{G} : \Omega_0 \times [0, T] \to \Omega; \qquad \mathcal{G} : (X, t) \mapsto \mathcal{G}(X, t), \tag{2.6}$$

where $\Omega_0 \subset \mathbb{R}^d$ is a fixed reference domain, T is the final time, and at each time $t \in [0, T]$, $\mathcal{G}(\,\cdot\,, t) : \Omega_0 \to \Omega$ is a diffeomorphism. We note that the domain Ω is fixed, i.e., Ω occupies the same region of $\mathbb{R}^d$ at any time $t \in [0, T]$; the time-dependent diffeomorphism is introduced as an integral part of the proposed numerical method to track discontinuities as they evolve. Under the domain mapping (2.6), the conservation law (2.1) becomes

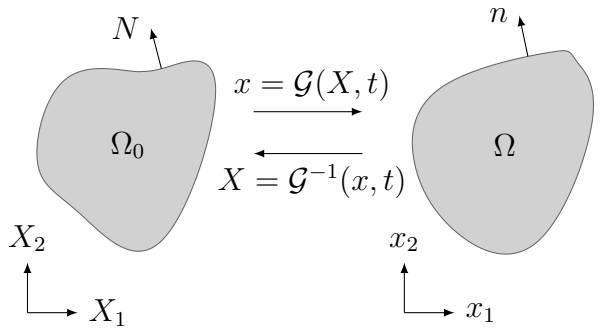


Figure 2.1: Mapping between reference and physical domains.

$$\frac{\partial U}{\partial t} + \nabla \cdot F(U) = 0 \quad \text{in} \quad \mathcal{G}(\Omega_0, t) \tag{2.7}$$

Following the approach in [54], the conservation law on the physical domain Ω is transformed to a conservation law on the reference domain Ω_0

$$\frac{\partial U_X}{\partial t} + \nabla_X \cdot F_X(U_X; G, v) = 0 \quad \text{in} \quad \Omega_0 \tag{2.8}$$

where $U_X : \Omega_0 \times [0,T] \to \mathbb{R}^m$ is the solution of the transformed conservation law, $F_X : \mathbb{R}^m \times \mathbb{R}^{d\times d} \times \mathbb{R}^d \to \mathbb{R}^{m\times d}$ is the transformed flux function, $\nabla_X \coloneqq (\partial_{X_1}, \ldots, \partial_{X_d})$ is the gradient operator on the reference domain, and the deformation gradient $G : \Omega_0 \times [0,T] \to \mathbb{R}^{d\times d}$, mapping Jacobian $g : \Omega_0 \times [0,T] \to \mathbb{R}$, and mapping velocity $v : \Omega_0 \times [0,T] \to \mathbb{R}^d$ are defined as

$$G = \nabla_X \mathcal{G}, \qquad g = \det G, \qquad v = \frac{\partial \mathcal{G}}{\partial t}. \tag{2.9}$$

The transformed and physical solutions are related, for any $X \in \Omega_0$ and $t \in [0,T]$, as

$$U_X(X,t) = g(X,t) U(\mathcal{G}(X,t), t) \tag{2.10}$$

and the transformed flux is defined as

$$F_X : (W_X; \Theta, \xi) \mapsto [(\det \Theta) F((\det \Theta)^{-1} W_X) - W_X \otimes \xi] \Theta^{-T}. \tag{2.11}$$

This is by an application of the divergence theorem in the physical domain, change of variables of surface integrals (Nanson's formula), and an application of the divergence theorem in the reference domain (See [78], equation (10) for details). The unit normals in the reference and physical domain are related by

$$n = \frac{g G^{-T} N}{\| g G^{-T} N \|}. \tag{2.12}$$

The transformed conservation law is supplemented with the initial condition $U_X(X,0) = \bar{U}_X(X)$ for all $X \in \Omega_0$, where $\bar{U}_X : \Omega_0 \to \mathbb{R}^m$ is $\bar{U}_X(X) = g(X,0)\bar{U}(\mathcal{G}(X,0))$. In this work, we take the reference domain to be the physical domain at time 0, which implies $g(X,0) = 1$.

2.2 Discontinuous Galerkin discretization of transformed conservation law

We use a nodal discontinuous Galerkin method [7, 31] to discretize the transformed conservation law (2.8). Let $\mathcal{E}_h$ represent a discretization of the reference domain Ω_0 into non-overlapping, potentially curved, computational elements, where h is a mesh element size parameter. The DG construction begins with the elementwise weak form of the conservation law (2.8) that results from multiplying each equation by a test function ψ_X, integrating over a single element $K \in \mathcal{E}_h$, and applying the divergence theorem

$$\int_K \psi_X \cdot \dot{U}_X \, dV + \int_{\partial K} \psi_X^+ \cdot F_X(U_X; G, v) N \, dS - \int_K F_X(U_X; G, v) : \nabla_X \psi_X \, dV = 0, \quad (2.13)$$

where N is the outward normal to the surface ∂K and ψ_X^+ denotes the trace of ψ *interior* to element K. To ensure the face integrals are single-valued, we replace $F_X(U_X; G, v)N$ in the second term with a numerical flux function $\mathcal{H}_X : \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R}^d \times \mathbb{R}^{d\times d} \times \mathbb{R}^d \to \mathbb{R}^m$ associated with the reference inviscid flux F_X

$$\int_K \psi_X \cdot \dot{U}_X \, dV + \int_{\partial K} \psi_X^+ \cdot \mathcal{H}_X(U_X^+, U_X^-, N; G, v) \, dS - \int_K F_X(U_X; G, v) : \nabla_X \psi_X \, dV = 0, \quad (2.14)$$

where U_X^+ (U_X^-) denotes the interior (exterior) trace of U_X to the element K; for points $X \in \partial K \cap \partial\Omega_0$, U_X^- is a boundary state constructed to enforce the appropriate boundary condition. In this work, we take the numerical flux to be a smoothed version of the Roe flux (Section 2.3). To establish the finite-dimensional (semi-discrete) form of (2.14), we introduce the DG approximation (trial) space of discontinuous piecewise polynomials associated with the mesh $\mathcal{E}_h$

$$\mathcal{V}_h^p = \left\{ v \in [L^2(\Omega_0 \times [0,T])]^m \;\middle|\; v(\,\cdot\,, t)|_K \in [\mathcal{P}_p(K)]^m, \; \forall K \in \mathcal{E}_h, \; t \in [0,T] \right\} \quad (2.15)$$

where $\mathcal{P}_p(K)$ is the space of polynomial functions of degree at most $p \geq 1$ on the element K, and we take the DG test space to be $\mathcal{V}_h^{p'}$, where $p' \geq p$. We also define the space of admissible domain mappings as the space of continuous piecewise polynomials of degree q associated with the mesh $\mathcal{E}_h$

$$\mathcal{W}_h = \left\{ v \in [C^0(\Omega_0 \times [0,T])]^d \;\middle|\; v(\,\cdot\,, t)|_K \in [\mathcal{P}_q(K)]^d, \; \forall K \in \mathcal{E}_h, \; t \in [0,T] \right\}. \quad (2.16)$$

We can now establish the finite dimensional form of (2.14) and the formal statement of DG as: given $\mathcal{G}_h \in \mathcal{W}_h$, find $U_{X,h} \in \mathcal{V}_h^p$ such that for all $\psi_{X,h} \in \mathcal{V}_h^{p'}$

$$\int_K \psi_{X,h} \cdot \dot{U}_{X,h} \, dV + \int_{\partial K} \psi_{X,h}^+ \cdot \mathcal{H}_X(U_{X,h}^+, U_{X,h}^-, N; G_h, v_h) \, dS - \int_K F_X(U_{X,h}; G_h, v_h) : \nabla_X \psi_{X,h} \, dV = 0, \quad (2.17)$$

where the DG residual form $r_h^{p'p} : \mathcal{V}_h^{p'} \times \mathcal{V}_h^p \times \mathcal{W}_h \to \mathbb{R}$ is given by

$$r_h^{p',p} : (\psi_{X,h}, W_{X,h}, \mathcal{Q}_h) \mapsto \sum_{K \in \mathcal{E}_{h,q}} r_K^{p',p}(\psi_{X,h}, W_{X,h}, \mathcal{Q}_h), \tag{2.18}$$

and the elemental DG form $r_K^{p',p} : \mathcal{V}_h^{p'} \times \mathcal{V}_h^p \times \mathcal{W}_h \to \mathbb{R}$ is given by

$$\begin{aligned} r_K^{p',p} : (\psi_{X,h}, W_{X,h}, \mathcal{Q}_h) \mapsto & \int_K \psi_{X,h} \cdot \dot{W}_{X,h} \, dV \\ & + \int_{\partial K} \psi_{X,h}^+ \cdot \mathcal{H}_X(W_{X,h}^+, W_{X,h}^-, N; \nabla_X \mathcal{Q}_h, \dot{\mathcal{Q}}_h) \, dS \\ & - \int_K F_X(W_{X,h}; \nabla_X \mathcal{Q}_h, \dot{\mathcal{Q}}_h) : \nabla_X \psi_{X,h} \, dV. \end{aligned} \tag{2.19}$$

Next, we introduce a (nodal) basis over each element for the test space ($\mathcal{V}_h^p$), trial space ($\mathcal{V}_h^{p'}$), and mapping space ($\mathcal{W}_h$) to reduce (2.19) to a system of ordinary differential equations (ODEs) in residual form. In the case where $p' = p$, we denote the residual $\boldsymbol{r} : \mathbb{R}^{N_{\boldsymbol{u}}} \times \mathbb{R}^{N_{\boldsymbol{u}}} \times \mathbb{R}^{N_{\boldsymbol{x}}} \times \mathbb{R}^{N_{\boldsymbol{x}}} \to \mathbb{R}^{N_{\boldsymbol{u}}}$, which is defined as

$$\boldsymbol{r} : (\mathring{\boldsymbol{w}}, \boldsymbol{w}, \mathring{\boldsymbol{y}}, \boldsymbol{y}) \mapsto \boldsymbol{m}\mathring{\boldsymbol{w}} + \boldsymbol{f}(\boldsymbol{w}, \boldsymbol{y}, \mathring{\boldsymbol{y}}), \tag{2.20}$$

where $N_{\boldsymbol{u}} = \dim \mathcal{V}_h^p$, $N_{\boldsymbol{x}} = \dim \mathcal{W}_h$, $\boldsymbol{m} \in \mathbb{R}^{N_{\boldsymbol{u}} \times N_{\boldsymbol{u}}}$ is the mass matrix associated with the test/trial space $\mathcal{V}_h^p$, and $\boldsymbol{f} : \mathbb{R}^{N_{\boldsymbol{u}}} \times \mathbb{R}^{N_{\boldsymbol{x}}} \times \mathbb{R}^{N_{\boldsymbol{x}}} \to \mathbb{R}^{N_{\boldsymbol{u}}}$ is the algebraic form of the second and third terms in (2.19). In this notation, the standard DG discretization reads: given $\boldsymbol{x} : [0, T] \to \mathbb{R}^{N_{\boldsymbol{x}}}$, find $\boldsymbol{u} : [0, T] \to \mathbb{R}^{N_{\boldsymbol{u}}}$ such that

$$\boldsymbol{r}(\dot{\boldsymbol{u}}(t), \boldsymbol{u}(t), \dot{\boldsymbol{x}}(t), \boldsymbol{x}(t)) = \boldsymbol{0}, \quad \boldsymbol{u}(0) = \bar{\boldsymbol{u}}, \tag{2.21}$$

for all $t \in [0, T]$, where $\boldsymbol{u}$ is the time-dependent coefficients of the DG solution, $\boldsymbol{x}$ is the time-dependent coefficients of the domain mapping (nodal coordinates of the mesh), and $\bar{\boldsymbol{u}} \in \mathbb{R}^{N_{\boldsymbol{u}}}$ is the algebraic representation of the initial condition $\bar{U}_X$; additionally, we define $\bar{\boldsymbol{x}} \in \mathbb{R}^{N_{\boldsymbol{x}}}$ as the initial condition for the nodal coordinates, i.e., $\bar{\boldsymbol{x}} = \boldsymbol{x}(0)$. Typically, the evolution of the mesh coordinates $\boldsymbol{x}(t)$ is known analytically or governed by a dynamical system (e.g., fluid-structure interaction); however, in this work, it will be determined as the solution of an optimization problem (after temporal discretization) such that discontinuities are tracked over time.

Finally, we use the expansions in the nodal bases to define the *enriched residual* $\boldsymbol{R} : \mathbb{R}^{N_{\boldsymbol{u}}} \times \mathbb{R}^{N_{\boldsymbol{u}}} \times \mathbb{R}^{N_{\boldsymbol{x}}} \times \mathbb{R}^{N_{\boldsymbol{x}}} \to \mathbb{R}^{N'_{\boldsymbol{u}}}$ associated with a trial space of degree p' as

$$\boldsymbol{R} : (\mathring{\boldsymbol{w}}, \boldsymbol{w}, \mathring{\boldsymbol{y}}, \boldsymbol{y}) \mapsto \boldsymbol{M}\mathring{\boldsymbol{w}} + \boldsymbol{F}(\boldsymbol{w}, \boldsymbol{y}, \mathring{\boldsymbol{y}}), \tag{2.22}$$

where $N'_{\boldsymbol{u}} = \dim \mathcal{V}_h^{p'}$, $\boldsymbol{M} \in \mathbb{R}^{N'_{\boldsymbol{u}} \times N_{\boldsymbol{u}}}$ is the mass matrix associated with the test space $\mathcal{V}_h^{p'}$ and trial space $\mathcal{V}_h^p$, and $\boldsymbol{F} : \mathbb{R}^{N_{\boldsymbol{u}}} \times \mathbb{R}^{N_{\boldsymbol{x}}} \times \mathbb{R}^{N_{\boldsymbol{x}}} \to \mathbb{R}^{N'_{\boldsymbol{u}}}$ is the algebraic form of the second and third terms in (2.19). In this work, we take $p' = p + 1$, but other choices are possible as well. The enriched residual will be used in Section 3 to define the implicit tracking objective function.

2.3 Numerical flux function

The numerical flux function corresponding to the reference flux ($\mathcal{H}_X$, i.e. reference numerical flux) is a quantity that replaces the transformed flux dotted with the outward unit normal ($F_X \cdot N$), as done from (2.13) to (2.14). It is given by

$$\mathcal{H}_X(U_X^+, U_X^-, N; G, v) = \left\| gG^{-T}N \right\| \tilde{\mathcal{H}}(U^+, U^-, n; v), \tag{2.23}$$

since by (2.10)-(2.12) we have

$$\mathcal{H}_X \sim F_X \cdot N = g(F - U \otimes v) \cdot G^{-T}N = \left\| gG^{-T}N \right\| (F - U \otimes v) \cdot n \sim \left\| gG^{-T}N \right\| \tilde{\mathcal{H}}, \tag{2.24}$$

where arguments have been dropped for brevity. Here, $\tilde{\mathcal{H}} : \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^m$ is a modified numerical flux which corresponds to the modified flux function, $\tilde{F} : \mathbb{R}^m \times \mathbb{R}^d \to \mathbb{R}^{m \times d}$, which is related to the physical flux as

$$\tilde{F}(U; v) = F(U) - U \otimes v. \tag{2.25}$$

The modified flux accounts for the domain motion and is obtained from the numerical flux function corresponding to the physical flux, $\mathcal{H} : \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R}^d \to \mathbb{R}^m$.

Notice that the Jacobian of the modified flux function only differs from the Jacobian of the physical flux by a scale multiple of the identity matrix (with scale factor $v \cdot n$), which makes implementation of many numerical fluxes that depend on the eigenvalue decomposition of the Jacobian matrix, e.g., local Lax-Friedrichs, Roe, Vijayasundaram, straightforward given the decomposition of the Jacobian of the physical flux. For the remainder of this section, we will discuss the numerical flux in the physical domain $\mathcal{H}(U^+, U^-, n)$ and it will be transformed to the reference domain according to (2.23).

For DG methods to be stable, the numerical flux may be any two-point monotone Lipschitz function that is

(i) *consistent* with the flux function, i.e., for any $U \in \mathbb{R}^M$

$$\mathcal{H}(U, U, n) = F(U) \cdot n \tag{2.26}$$

(ii) *conservative*, i.e., for any $U, U' \in \mathbb{R}^M$

$$\mathcal{H}(U, U', n) = -\mathcal{H}(U', U, -n). \tag{2.27}$$

Conditions (i)-(ii) are satisfied by all standard numerical fluxes as these are the minimum requirements for a stable and accurate DG method. However, it was observed in [79, 8] that the requirements are higher for tracking-based discretizations since inter-element jumps do not tend to zero under refinement. As introduced in [8], this requires the additional condition:

(iii) *preservation* of the Rankine-Hugoniot conditions, i.e., given $U^+, U^- \in \mathbb{R}^M$ such that $F(U^+) \cdot n = F(U^-) \cdot n$, then

$$\mathcal{H}(U^+, U^-, n) = F(U^+) \cdot n = F(U^-) \cdot n \tag{2.28}$$

Conditions (i)-(iii) are satisfied by exact Riemann solvers and a number of approximate Riemann solvers including the Roe [61] and HLLC [67] fluxes; however, many popular numerical flux functions such as the local Lax-Friedrichs, Rusanov, and Roe with entropy fix do not satisfy (iii) [8]. We highlight one additional desired property of the numerical flux function that we will show is important for optimization-based discontinuity tracking:

(iv) *smoothness* with respect to variations in the normal.

Since the numerical flux must have upwind-like properties, it is difficult to construct numerical fluxes to satisfy (i)-(iv), particularly the smoothness property. Therefore, we choose numerical fluxes that satisfy (i)-(iii) and replace non-smooth terms with smooth approximations to recover smoothness (iv). Strictly speaking, this will not preserve the Rankine-Hugoniot conditions (iii); in Section 4 we demonstrate this is a desirable trade-off.

As an example, consider linear advection of a scalar field $u : \Omega \to \mathbb{R}$ in a spatially varying direction $\beta : \Omega \to \mathbb{R}^d$ governed by conservation law of the form (2.2) (see (4.2) in Section 4.2) with flux function

$$F_{\text{adv}}(U) = U\beta^T \tag{2.29}$$

and the upwind numerical flux

$$\mathcal{H}_{\text{up}}(U^+, U^-, n) = \begin{cases} (\beta \cdot n)U^+ & \text{if } \beta \cdot n \geq 0 \\ (\beta \cdot n)U^- & \text{if } \beta \cdot n < 0, \end{cases} \tag{2.30}$$

which can equivalently be written in terms of the Heaviside function $H : \mathbb{R} \to \{0, 1\}$ as

$$\mathcal{H}_{\text{up}}(U^+, U^-, n) = (\beta \cdot n) \left[U^+ \cdot H(\beta \cdot n) + U^- \cdot (1 - H(\beta \cdot n)) \right]. \tag{2.31}$$

The upwind flux satisfies conditions (i)-(iii) for an admissible flux for shock tracking, but fails to satisfy condition (iv). Condition (i) follows from

$$\mathcal{H}_{\text{up}}(U, U, n) = (\beta \cdot n)U = F_{\text{adv}}(U)n \tag{2.32}$$

and condition (ii) follows from

$$-\mathcal{H}(U', U, -n) = (\beta \cdot n) \left[U' \cdot H(-\beta \cdot n) + U \cdot (1 - H(-\beta \cdot n)) \right] = \mathcal{H}(U, U', n), \tag{2.33}$$

for any $U, U' \in \mathbb{R}$, where the second equality follows from the property of the Heaviside function: $H(-s) = 1 - H(s)$ for $s \in \mathbb{R}$. To verify condition (iii), consider $U^+, U^- \in \mathbb{R}$ such that $U^+ \neq U^-$ (discontinuity) and assume $F_{\text{adv}}(U^+) \cdot n = F_{\text{adv}}(U^-) \cdot n$. This condition

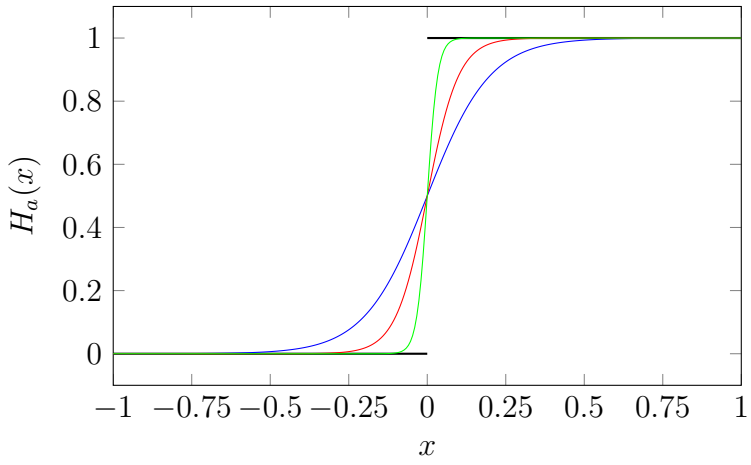


Figure 2.2: Smoothed Heaviside (logistic) function for $a = 5$ (——), $a = 10$ (——), $a = 30$ (——), $a = \infty$ (——).

implies $(U^+ - U^-)(\beta \cdot n) = 0$, which in turn implies $\beta \cdot n = 0$ from the assumption that $U^+ \neq U^-$. Therefore condition (iii) holds from

$$\mathcal{H}(U^+, U^-, n) = F_{\text{adv}}(U^+)n = F_{\text{adv}}(U^-)n = 0. \tag{2.34}$$

Finally, it is easy to see that condition (iv) is not satisfied; the upwind flux is continuous with respect to variations in the normal n, but not smooth due to the $H(\beta \cdot n)$ terms. A single isolated point of non-smoothness does not necessarily hinder the optimization solver because it is unlikely to be visited during the solution procedure. However, in this case, the kinks in the numerical flux function lie at points where $\beta \cdot n = 0$, which is precisely the requirement for a discontinuous solution to satisfy the Rankine-Hugoniot conditions and will certainly be approached as the mesh faces align with the discontinuity.

In Section 2.3 of [78], to recover condition (iv), we introduce a smoothed version of the upwind flux where the Heaviside function is replaced with a smoothed step function $H_a : \mathbb{R} \to \mathbb{R}$

$$\mathcal{H}^a_{\text{up}}(U^+, U^-, n) = (\beta \cdot n)\left[U^+ \cdot H_a(\beta \cdot n) + U^- \cdot (1 - H_a(\beta \cdot n))\right]. \tag{2.35}$$

where we used the logistic function as the smoothed step function

$$H_a(x) := \frac{1}{1 + e^{-2ax}}, \tag{2.36}$$

where $a \in \mathbb{R}$ is the smoothing parameter (Figure 2.2). Section 4 provides a detailed study of the impact of the smoothness of the numerical flux on the convergence of the tracking algorithm.

We later found in [63] that the smoothed absolute value function performs better numerically as we are no longer smoothing a discontinuous function. To repeat the previous procedure for clarity for this alternative method of smoothing, again consider linear advection of a scalar field $U : \Omega \to \mathbb{R}$ in a spatially varying direction $\beta : \Omega \to \mathbb{R}^d$ governed by a conservation law of the form (2.2) with physical flux function

$$F_{\text{adv}}(U) = U\beta^T \tag{2.29 revisited}$$

with the corresponding upwind numerical flux

$$\mathcal{H}_{\text{up}}(U^+, U^-, n) = \begin{cases} (\beta \cdot n)U^+ & \text{if } \beta \cdot n \geq 0 \\ (\beta \cdot n)U^- & \text{if } \beta \cdot n < 0. \end{cases} \tag{2.30 revisited}$$

This can equivalently be written in terms of the absolute value function $|\cdot| : \mathbb{R} \to \mathbb{R}_{\geq 0}$ as

$$\mathcal{H}_{\text{up}}(U^+, U^-, n) = 0.5\left[(\beta \cdot n)(U^+ + U^-) + (U^+ - U^-)|\beta \cdot n|\right]. \tag{2.37}$$

Due to the linearity of the flux, the modified flux function is

$$\tilde{F}_{\text{adv}}(U) = U(\beta - v)^T, \tag{2.38}$$

which has an identical form as (2.29), albeit with a modified velocity field. Thus, the corresponding modified upwind numerical flux function is

$$\tilde{\mathcal{H}}_{\text{up}}(U^+, U^-, n; v) = 0.5\left[((\beta - v) \cdot n)(U^+ + U^-) + (U^+ - U^-)|(\beta - v) \cdot n|\right]. \tag{2.39}$$

Following the approach in [81, 80], we introduce a smoothed version the upwind flux where the absolute value function is replaced with a smoothed absolute value function

$$\mathcal{H}^s_{\text{up}}(U^+, U^-, n) = 0.5\left[(\beta \cdot n)(U^+ + U^-) + (U^+ - U^-)|\beta \cdot n|_s\right], \tag{2.40}$$

where $|\cdot|_s : \mathbb{R} \to \mathbb{R}_{\geq 0}$ is a smooth approximation to the absolute value given by

$$|\cdot|_s : x \mapsto x \tanh(kx) \tag{2.41}$$

and k is a smoothness parameter (Fig. 2.3). In this work, we use $k = 100$. Then, the smoothed version of the modified upwind numerical flux is given by

$$\tilde{\mathcal{H}}^s_{\text{up}}(U^+, U^-, n; v) = 0.5\left[((\beta - v) \cdot n)(U^+ + U^-) + (U^+ - U^-)|(\beta - v) \cdot n|_s\right]. \tag{2.42}$$

Similarly, for the Euler equations, we use Roe's flux [61] with the absolute value of the eigenvalues replaced by the corresponding smoothed absolute value; a detailed derivation of the smoothed Roe flux for the reacting Euler equations (single reaction) is included in Appendix A of [80].

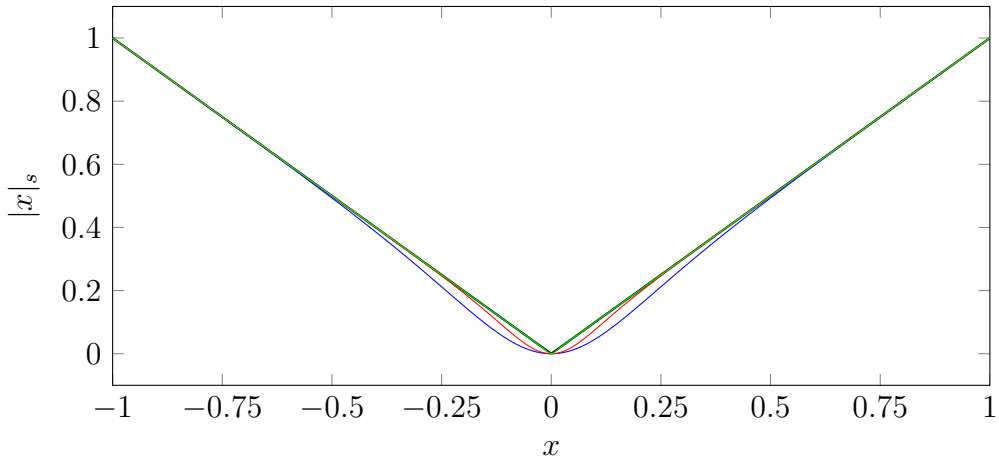


Figure 2.3: Smoothed absolute value function for $k = 5$ (—), $k = 10$ (—), and $k = 100$ (—). $k = \infty$ corresponds to the unsmoothed absolute value function (—).

2.4 High-order temporal discretization

Proceeding with the method of lines, we discretize the dynamical system in (2.21) using a diagonally implicit Runge-Kutta method (DIRK) to yield a sequence of algebraic systems of equations. Unlike fully implicit Runge-Kutta methods, an s-stage DIRK scheme has a Butcher tableau $(A, b, c) \in \mathbb{R}^{s\times s} \times \mathbb{R}^s \times \mathbb{R}^s$ where A is lower triangular (Table 2.1). As a result, the ith stage only depends on the solution for stages $1, \ldots, i$, allowing the stages to be solved sequentially. Because the nodal coordinates $\boldsymbol{x}$ are unknown in the proposed setting, we first recast (2.21) as the following coupled system of ODEs

$$\begin{aligned} \boldsymbol{m}\dot{\boldsymbol{u}}(t) + \boldsymbol{f}(\boldsymbol{u}(t), \boldsymbol{x}(t), \boldsymbol{\nu}(t)) &= \boldsymbol{0} \\ \dot{\boldsymbol{x}}(t) - \boldsymbol{\nu}(t) &= \boldsymbol{0} \\ \boldsymbol{u}(0) &= \bar{\boldsymbol{u}} \\ \boldsymbol{x}(0) &= \bar{\boldsymbol{x}} \end{aligned} \tag{2.43}$$

Table 2.1: Butcher tableau for DIRK schemes

$$\begin{array}{c|c} \boldsymbol{c} & \boldsymbol{A} \\ \hline & \boldsymbol{b}^T \end{array} \quad = \quad \begin{array}{c|ccc} c_1 & a_{11} & & \\ \vdots & \vdots & \ddots & \\ c_s & a_{s1} & \ldots & a_{ss} \\ \hline & b_1 & \ldots & b_s \end{array}$$

for $t \in [0,T]$, where $\boldsymbol{\nu} : [0,T] \to \mathbb{R}^{N_{\boldsymbol{x}}}$ is the unknown nodal velocity, defined as

$$\boldsymbol{\nu} : t \mapsto \dot{\boldsymbol{x}}(t). \tag{2.44}$$

Next, we partition the time interval $[0,T]$ into N_T intervals of equal size $\Delta t = T/N_T$ with endpoints $\{t_n\}_{n=0}^{N_T}$, where $t_0 = 0$ and $t_n = t_{n-1} + \Delta t$ for $n = 1, \dots, N_T$. In this setting, an s-stage DIRK discretization of (2.21) with Butcher tableau (A, b, c) reads: for $n = 1, \dots, N_T$ and $i = 1, \dots, s$,

$$\begin{aligned}
&\boldsymbol{u}_0 = \bar{\boldsymbol{u}}, \quad \boldsymbol{u}_{n+1} = \boldsymbol{u}_n + \sum_{j=1}^{s} b_i \boldsymbol{k}^{\boldsymbol{u}}_{n,j}, \quad \boldsymbol{u}_{n,i} = \boldsymbol{u}_n + \sum_{j=1}^{i} a_{ij} \boldsymbol{k}^{\boldsymbol{u}}_{n,j} \\
&\boldsymbol{x}_0 = \bar{\boldsymbol{x}}, \quad \boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \sum_{j=1}^{s} b_i \boldsymbol{k}^{\boldsymbol{x}}_{n,j}, \quad \boldsymbol{x}_{n,i} = \boldsymbol{x}_n + \sum_{j=1}^{i} a_{ij} \boldsymbol{k}^{\boldsymbol{x}}_{n,j} \\
&\boldsymbol{m}\boldsymbol{k}^{\boldsymbol{u}}_{n,i} = -\Delta t \boldsymbol{f}\left(\boldsymbol{u}_{n,i}, \boldsymbol{x}_{n,i}, \boldsymbol{\nu}_{n,i}\right), \quad \boldsymbol{k}^{\boldsymbol{x}}_{n,i} = \Delta t \boldsymbol{\nu}_{n,i}, \quad \boldsymbol{\nu}_{n,i} = \boldsymbol{\nu}(t_n + c_i \Delta t)
\end{aligned} \tag{2.45}$$

where $\boldsymbol{u}_0, \boldsymbol{u}_n, \boldsymbol{u}_{n,i}, \boldsymbol{k}^{\boldsymbol{u}}_{n,i} \in \mathbb{R}^{N_{\boldsymbol{u}}}$ and $\boldsymbol{x}_0, \boldsymbol{x}_n, \boldsymbol{x}_{n,i}, \boldsymbol{k}^{\boldsymbol{x}}_{n,i}, \boldsymbol{\nu}_{n,i} \in \mathbb{R}^{N_{\boldsymbol{x}}}$ are implicitly defined as the solution of (2.45); $\boldsymbol{u}_n \approx \boldsymbol{u}(t_n)$ and $\boldsymbol{x}_n \approx \boldsymbol{x}(t_n)$ are the state and mesh approximation at each time step $n = 0, \dots, T$; $\boldsymbol{u}_{n,i} \approx \boldsymbol{u}(t_n + c_i \Delta t)$ and $\boldsymbol{x}_{n,i} \approx \boldsymbol{x}(t_n + c_i \Delta t)$ are the state and mesh approximations at each stage $i = 1, \dots, s$ of each time interval $n = 1, \dots, T$; $\boldsymbol{k}^{\boldsymbol{u}}_{n,i}$ and $\boldsymbol{k}^{\boldsymbol{x}}_{n,i}$ are the solution and mesh stage updates, and $\boldsymbol{\nu}_{n,i} = \boldsymbol{\nu}(t_n + c_i \Delta t)$ is the mesh velocity. Because the mesh velocity $\boldsymbol{\nu}$ is unknown, the system in (2.45) can neither be solved nor evaluated in residual form.

Following the work in [23], we replace the unknown velocity function $\boldsymbol{\nu}$ with a modified velocity function, $\tilde{\boldsymbol{\nu}} : [0,T] \to \mathbb{R}^{N_{\boldsymbol{x}}}$, that ensures the corresponding modified position $\tilde{\boldsymbol{x}} : [0,T] \to \mathbb{R}^{N_{\boldsymbol{x}}}$, defined as the solution of

$$\dot{\tilde{\boldsymbol{x}}}(t) - \tilde{\boldsymbol{\nu}}(t) = \boldsymbol{0}, \qquad \tilde{\boldsymbol{x}}(0) = \bar{\boldsymbol{x}} \tag{2.46}$$

for $t \in [0,T]$, agree with the original stages $\boldsymbol{x}_{n,i}$ when discretized with the same DIRK method. Discretization of (2.46) using the DIRK scheme reads: for $n = 1, \dots, N_T$ and $i = 1, \dots, s$,

$$\tilde{\boldsymbol{x}}_0 = \bar{\boldsymbol{x}}, \quad \tilde{\boldsymbol{x}}_{n+1} = \tilde{\boldsymbol{x}}_n + \sum_{j=1}^{s} b_i \tilde{\boldsymbol{k}}^{\boldsymbol{x}}_{n,j}, \quad \tilde{\boldsymbol{x}}_{n,i} = \tilde{\boldsymbol{x}}_n + \sum_{j=1}^{i} a_{ij} \tilde{\boldsymbol{k}}^{\boldsymbol{x}}_{n,j}, \quad \tilde{\boldsymbol{k}}^{\boldsymbol{x}}_{n,i} = \Delta t \tilde{\boldsymbol{\nu}}_{n,i}, \quad \tilde{\boldsymbol{\nu}}_{n,i} = \tilde{\boldsymbol{\nu}}(t_n + c_i \Delta t), \tag{2.47}$$

where $\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{x}}_n, \tilde{\boldsymbol{x}}_{n,i}, \tilde{\boldsymbol{k}}^{\boldsymbol{x}}_{n,i}, \tilde{\boldsymbol{\nu}}_{n,i} \in \mathbb{R}^{N_{\boldsymbol{x}}}$ are implicitly defined as the solution of (2.47); $\tilde{\boldsymbol{x}}_n \approx \tilde{\boldsymbol{x}}(t_n)$ is the modified mesh approximation at each time step $n = 0, \dots, T$; $\tilde{\boldsymbol{x}}_{n,i} \approx \tilde{\boldsymbol{x}}(t_n + c_i \Delta t)$ is the modified mesh approximation at each stage $i = 1, \dots, s$ of each time interval $n = 1, \dots, T$; $\tilde{\boldsymbol{k}}^{\boldsymbol{x}}_{n,i}$ is the modified mesh stage update, and $\tilde{\boldsymbol{\nu}}_{n,i} = \tilde{\boldsymbol{\nu}}(t_n + c_i \Delta t)$ is the modified mesh velocity at each stage. Because the DIRK scheme only depends on the value of $\tilde{\boldsymbol{\nu}}$ at $t_n + c_i \Delta t$, we only need to define its value at these points, i.e., $\tilde{\boldsymbol{\nu}}_{n,i}$; any smooth function $\boldsymbol{z} : [0,T] \to \mathbb{R}^{N_{\boldsymbol{x}}}$

that satisfies $\boldsymbol{z}(t_n + c_i\Delta t) = \tilde{\boldsymbol{\nu}}_{n,i}$ will lead to the same DIRK solution and is a valid choice for $\tilde{\boldsymbol{\nu}}$. The requirement that the modified and original mesh stages agree, i.e., $\tilde{\boldsymbol{x}}_{n,i} = \boldsymbol{x}_{n,i}$ for $n = 1, \ldots, N_T$ and $i = 1, \ldots, s$, leads to

$$\boldsymbol{x}_{n,i} = \tilde{\boldsymbol{x}}_{n,i} = \tilde{\boldsymbol{x}}_n + \Delta t \sum_{j=1}^{i} a_{ij}\tilde{\boldsymbol{\nu}}_{n,j} = \boldsymbol{x}_n + \Delta t \sum_{j=1}^{i} a_{ij}\tilde{\boldsymbol{\nu}}_{n,j}, \tag{2.48}$$

where we used the definition of the stage $\tilde{\boldsymbol{x}}_{n,i}$ in (2.47) and the relationship $\tilde{\boldsymbol{x}}_n = \boldsymbol{x}_n$ for $n = 0, \ldots, N_T$, which follows directly from the stage-consistent requirement ($\tilde{\boldsymbol{x}}_{n,i} = \boldsymbol{x}_{n,i}$), the initial condition ($\tilde{\boldsymbol{x}}_0 = \boldsymbol{x}_0 = \bar{\boldsymbol{x}}$), and that the DIRK schemes used to integrate (2.43) and (2.46) are the identical (same Butcher tableau). In the case where A is full rank, which is the case for fully and diagonally implicit Runge-Kutta methods, this equation can be inverted to express the modified velocity stages as a linear combination of the mesh stages

$$\tilde{\boldsymbol{\nu}}_{n,i} = \sum_{j=1}^{i} (\boldsymbol{A}^{-1})_{ij} \frac{\boldsymbol{x}_{n,j} - \boldsymbol{x}_n}{\Delta t}. \tag{2.49}$$

This approach leads to a *stage consistent* velocity approximation [23] in the sense that the modified velocity function is chosen such that it is consistent with the original mesh positions at each stage.

With the stage consistent mesh velocity approximation, the DIRK discretization in (2.45) becomes

$$\begin{aligned}
&\boldsymbol{u}_0 = \bar{\boldsymbol{u}}, \quad \boldsymbol{u}_{n+1} = \boldsymbol{u}_n + \sum_{j=1}^{s} b_i \boldsymbol{k}^{\boldsymbol{u}}_{n,j}, \quad \boldsymbol{u}_{n,i} = \boldsymbol{u}_n + \sum_{j=1}^{i} a_{ij} \boldsymbol{k}^{\boldsymbol{u}}_{n,j} \\
&\boldsymbol{x}_0 = \bar{\boldsymbol{x}}, \quad \boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \sum_{j=1}^{s} b_i \boldsymbol{k}^{\boldsymbol{x}}_{n,j}, \quad \boldsymbol{x}_{n,i} = \boldsymbol{x}_n + \sum_{j=1}^{i} a_{ij} \boldsymbol{k}^{\boldsymbol{x}}_{n,j} \\
&\boldsymbol{m}\boldsymbol{k}^{\boldsymbol{u}}_{n,i} = -\Delta t \boldsymbol{f}\left(\boldsymbol{u}_{n,i}, \boldsymbol{x}_{n,i}, \tilde{\boldsymbol{\nu}}_{n,i}\right), \quad \boldsymbol{k}^{\boldsymbol{x}}_{n,i} = \Delta t \tilde{\boldsymbol{\nu}}_{n,i}
\end{aligned} \tag{2.50}$$

for $n = 1, \ldots, N_T$ and $i = 1, \ldots, s$, where the stage velocity is defined in (2.49). Even though the unknown velocity function has been eliminated using the stage consistent velocity approximation, the new system (2.50) is underdetermined because there are effectively $N_{\boldsymbol{u}}$ equations in $N_{\boldsymbol{u}} + N_{\boldsymbol{x}}$ unknowns at a fixed stage. This will be resolved by the optimization-based tracking formulation in the next section.

To close this section, we convert the modified DIRK system in (2.50) to residual form $\boldsymbol{r}_{n,i} : \mathbb{R}^{N_{\boldsymbol{u}}} \times \mathbb{R}^{N_{\boldsymbol{x}}} \to \mathbb{R}^{N_{\boldsymbol{u}}}$ at a fixed step $n \in \{1, \ldots, N_T\}$ and stage $i \in \{1, \ldots, s\}$ as

$$\boldsymbol{r}_{n,i} : (\boldsymbol{w}, \boldsymbol{y}) \mapsto \boldsymbol{m}\boldsymbol{\xi}_{n,i}(\boldsymbol{w}) + \Delta t \boldsymbol{f}(\boldsymbol{w}, \boldsymbol{y}, \boldsymbol{\zeta}_{n,i}(\boldsymbol{y})), \tag{2.51}$$

where $\boldsymbol{\xi}_{n,i} : \mathbb{R}^{N_{\boldsymbol{u}}} \to \mathbb{R}^{N_{\boldsymbol{u}}}$ maps the state stage ($\boldsymbol{u}_{n,i}$) to the corresponding stage update ($\boldsymbol{k}^{\boldsymbol{u}}_{n,i}$)

$$\boldsymbol{\xi}_{n,i} : \boldsymbol{w} \mapsto (A^{-1})_{ii}(\boldsymbol{w} - \boldsymbol{u}_n) + \sum_{j=1}^{i-1} (A^{-1})_{ij}(\boldsymbol{u}_{n,j} - \boldsymbol{u}_n) \tag{2.52}$$

Table 2.2: Butcher tableau for DIRK1 (*left*), DIRK2 (*middle*), and DIRK3 (*right*), where $\alpha = 1 - \frac{1}{\sqrt{2}}$, $\beta = 0.435866521508459$, $\gamma = -\frac{6\beta^2 - 16\beta + 1}{4}$, and $\omega = \frac{6\beta^2 - 20\beta + 5}{4}$.

1	1
	1

α	α	0
1	1-α	α
	1-α	α

β	β		
$\frac{1+\beta}{2}$	$\frac{1+\beta}{2} - \beta$	β	
$\gamma + \omega + \beta$	γ	ω	β
	γ	ω	β

and $\boldsymbol{\zeta}_{n,i} : \mathbb{R}^{N_{\boldsymbol{x}}} \to \mathbb{R}^{N_{\boldsymbol{x}}}$ maps the mesh stage ($\boldsymbol{x}_{n,i}$) to the corresponding stage-consistent velocity ($\tilde{\boldsymbol{\nu}}_{n,i}$) as

$$\boldsymbol{\zeta}_{n,i} : \boldsymbol{y} \mapsto (A^{-1})_{ii} \frac{\boldsymbol{y} - \boldsymbol{x}_n}{\Delta t} + \sum_{j=1}^{i-1} (A^{-1})_{ij} \frac{\boldsymbol{x}_{n,j} - \boldsymbol{x}_n}{\Delta t}. \tag{2.53}$$

Similarly, we define the corresponding fully discrete enriched residual function $\boldsymbol{R}_{n,i} : \mathbb{R}^{N_{\boldsymbol{u}}} \times \mathbb{R}^{N_{\boldsymbol{x}}} \to \mathbb{R}^{N'_{\boldsymbol{u}}}$ as

$$\boldsymbol{R}_{n,i} : (\boldsymbol{w}, \boldsymbol{y}) \mapsto \boldsymbol{M}\boldsymbol{\xi}_{n,i}(\boldsymbol{w}) + \Delta t \boldsymbol{F}(\boldsymbol{w}, \boldsymbol{y}, \boldsymbol{\zeta}_{n,i}(\boldsymbol{y})), \tag{2.54}$$

which we use in next section to define the objective function of the implicit tracking optimization problem. Notice that for $\boldsymbol{R}_{n,i}$ only the *spatial* test space is enriched; the temporal discretization in $\boldsymbol{r}_{n,i}$ and $\boldsymbol{R}_{n,i}$ is identical.

In this work, we consider three $L-$ stable DIRK schemes with order of accuracy equal to the number of stages s; Butcher tableaus given in Table 2.2. We denote the k-th order accurate DIRK scheme as DIRKk, i.e. 1st order DIRK is DIRK1, etc. We note that DIRK1 is equivalent to the Backward Euler scheme.

Chapter 3

Optimization formulation of r-adaptivity for implicit tracking of discontinuities

In this section, we introduce the main contribution of this work: an r-adaptivity framework that recasts the discrete conservation law as an optimization problem over the discrete solution and mesh that aims to align features in the solution basis with features in the solution itself. In this work these features are discontinuities since we only consider inviscid conservation laws; however, future work will consider steep gradients (viscous conservation laws) and interfaces. In the present setting, this amounts to aligning element faces with discontinuities. In this section, we focus on setting up the optimization formulation and address in detail the issue of solvers for the optimization problem. In particular, we introduce a new error-like objective function based on the DG residual using an enriched *test space* and a term to penalize mesh distortion. We first explain this framework in the case of the the steady conservation law (2.2) with corresponding discretized form (2.5), and in Section 3.5 explain how this this framework naturally extends to the unsteady case by the method of lines.

3.1 Constrained optimization formulation

We formulate the problem of tracking discontinuities as a constrained optimization problem over the PDE state and coordinates of the mesh nodes that minimizes some objective function $f : \mathbb{R}^{N_{\boldsymbol{u}}} \times \mathbb{R}^{N_{\boldsymbol{x}}} \to \mathbb{R}$ while enforcing the DG discretization of the conservation law

$$\begin{aligned} \underset{\boldsymbol{u}\in\mathbb{R}^{N_{\boldsymbol{u}}},\boldsymbol{x}\in\mathbb{R}^{N_{\boldsymbol{x}}}}{\text{minimize}} \quad & f(\boldsymbol{u},\boldsymbol{x}) \\ \text{subject to} \quad & \boldsymbol{r}(\boldsymbol{u},\boldsymbol{x}) = \boldsymbol{0}. \end{aligned} \tag{3.1}$$

The objective function is constructed such that the solution of the optimization problem is a mesh that aligns with discontinuities in the solution. The optimization-based tracking

method directly inherits the benefits of standard DG methods, i.e., high-order accuracy and conservation, due to the constraint that exactly enforces the DG discretization. Finally, the optimization formulation in (3.1) will provide nonlinear stability if all discontinuities are successfully tracked, which we will demonstrate using several examples in Sections 4 and 5.

The Lagrangian of the optimization problem in (3.1) $\mathcal{L} : \mathbb{R}^{N_{\boldsymbol{u}}} \times \mathbb{R}^{N_{\boldsymbol{x}}} \times \mathbb{R}^{N_{\boldsymbol{u}}} \to \mathbb{R}$ takes the form

$$\mathcal{L}(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{u}, \boldsymbol{x}) - \boldsymbol{\lambda}^T \boldsymbol{r}(\boldsymbol{u}, \boldsymbol{x}), \tag{3.2}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{N_{\boldsymbol{u}}}$ is a vector of Lagrange multipliers associated with the DG constraint in (3.1). The first-order optimality, or Karush-Kuhn-Tucker (KKT), conditions state that the $(\boldsymbol{u}^\star, \boldsymbol{x}^\star)$ is a first-order solution of the optimization problem if there exists $\boldsymbol{\lambda}^\star$ such that

$$\nabla_{\boldsymbol{u}} \mathcal{L}(\boldsymbol{u}^\star, \boldsymbol{x}^\star, \boldsymbol{\lambda}^\star) = \boldsymbol{0}, \qquad \nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{u}^\star, \boldsymbol{x}^\star, \boldsymbol{\lambda}^\star) = \boldsymbol{0}, \qquad \nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{u}^\star, \boldsymbol{x}^\star, \boldsymbol{\lambda}^\star) = \boldsymbol{0}, \tag{3.3}$$

or equivalently,

$$\frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}^\star, \boldsymbol{x}^\star)^T - \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}^\star, \boldsymbol{x}^\star)^T \boldsymbol{\lambda}^\star = \boldsymbol{0}, \qquad \frac{\partial f}{\partial \boldsymbol{x}}(\boldsymbol{u}^\star, \boldsymbol{x}^\star)^T - \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}}(\boldsymbol{u}^\star, \boldsymbol{x}^\star)^T \boldsymbol{\lambda}^\star = \boldsymbol{0}, \qquad \boldsymbol{r}(\boldsymbol{u}^\star, \boldsymbol{x}^\star) = \boldsymbol{0}. \tag{3.4}$$

Since the DG Jacobian with respect to the state variables $\boldsymbol{u}$ is assumed to be invertible, we define the estimate of the optimal Lagrange multiplier $\hat{\boldsymbol{\lambda}} : \mathbb{R}^{N_{\boldsymbol{u}}} \times \mathbb{R}^{N_{\boldsymbol{x}}} \to \mathbb{R}^{N_{\boldsymbol{u}}}$ such that the first equation ($\nabla_{\boldsymbol{u}} \mathcal{L} = 0$) (adjoint equation) is always satisfied

$$\hat{\boldsymbol{\lambda}}(\boldsymbol{u}, \boldsymbol{x}) = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{x})^{-T} \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{x})^T. \tag{3.5}$$

Then the optimality criteria becomes

$$\boldsymbol{c}(\boldsymbol{u}^\star, \boldsymbol{x}^\star) = \boldsymbol{0}, \qquad \boldsymbol{r}(\boldsymbol{u}^\star, \boldsymbol{x}^\star) = \boldsymbol{0}, \tag{3.6}$$

where $\boldsymbol{c} : \mathbb{R}^{N_{\boldsymbol{u}}} \times \mathbb{R}^{N_{\boldsymbol{x}}} \to \mathbb{R}^{N_{\boldsymbol{u}}}$ is defined as

$$\boldsymbol{c}(\boldsymbol{u}, \boldsymbol{x}) \coloneqq \nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{u}, \boldsymbol{x}, \hat{\boldsymbol{\lambda}}(\boldsymbol{u}, \boldsymbol{x})) = \frac{\partial f}{\partial \boldsymbol{x}}(\boldsymbol{u}, \boldsymbol{x})^T - \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}}(\boldsymbol{u}, \boldsymbol{x})^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{x})^{-T} \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{x})^T. \tag{3.7}$$

In Section 3.4.7, $\|\boldsymbol{c}(\boldsymbol{u}, \boldsymbol{x})\|$ and $\|\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{x})\|$ will be used to define the termination criteria for the proposed solver.

3.2 Choice of objective function

We propose an objective function that consists of two terms: one term penalizes a measure of the DG solution error $f_{\text{err}} : \mathbb{R}^{N_{\boldsymbol{u}}} \times \mathbb{R}^{N_{\boldsymbol{x}}} \to \mathbb{R}$ and the other term penalizes distortion of the mesh $f_{\text{msh}} : \mathbb{R}^{N_{\boldsymbol{x}}} \to \mathbb{R}$, i.e.,

$$f(\boldsymbol{u}, \boldsymbol{x}) = f_{\text{err}}(\boldsymbol{u}, \boldsymbol{x}) + \kappa^2 f_{\text{msh}}(\boldsymbol{x}), \tag{3.8}$$

where $\kappa \in \mathbb{R}_+$ is a parameter that weights the contribution of the two terms. Since a piecewise polynomial solution on an aligned mesh will have much lower error than on a non-aligned mesh, f_{err} promotes alignment of the mesh with discontinuities while f_{msh} prevents the mesh from entangling or becoming unacceptably skewed.

For the error-like tracking term, we use the norm of the DG residual corresponding to an enriched *test space*, i.e.,

$$f_{\text{err}}(\boldsymbol{u}, \boldsymbol{x}) \coloneqq \frac{1}{2}\boldsymbol{R}(\boldsymbol{u}, \boldsymbol{x})^T \boldsymbol{R}(\boldsymbol{u}, \boldsymbol{x}), \tag{3.9}$$

where we enrich the test space using polynomials of one degree higher than the trial space. This follows on a large body of work that uses residual-based error indicators to drive h-, p-, and r-adaptivity [20]. This is a reasonable choice for the objective function because the enriched test space adds additional constraints to the solution and even though a discrete solution $\boldsymbol{u}$ containing significant oscillations satisfies $\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{x}) = \boldsymbol{0}$, it will likely not minimize $\boldsymbol{R}(\,\cdot\,, \boldsymbol{x})$. Furthermore, the more the test space is enriched, the closer the DG residual comes to enforcing the true conservation law and therefore the minimum-residual solution approaches the exact solution of the PDE. Given that we are explicitly enforcing the constraint $\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{x}) = \boldsymbol{0}$, which fixes $\boldsymbol{u}$ for a given $\boldsymbol{x}$ (assuming $\boldsymbol{r}(\cdot, \boldsymbol{x}) = \boldsymbol{0}$ has a unique solution), the solution of the optimization problem must deform the mesh $\boldsymbol{x}$ to drive the pair $(\boldsymbol{u}, \boldsymbol{x})$ to a point where the enriched residual is minimized, which we expect to be a mesh that tracks sharp features in the solution. Therefore, we expect this choice of objective function to have desirable tracking properties, which will be confirmed by our numerical experiments (Section 4). Finally, this choice of objective function is agnostic to whether we are considering an inviscid or viscous conservation law, which is not the case for many popular physics-based feature indicators than rely, e.g., on the Rankine-Hugoniot conditions.

[79] considered the objective function in (3.9), but did not endorse it due to numerical experiments that showed it possessed non-aligned local minima that made it impractical. However, in that work only meshes with a fixed topology were considered, which contributed to these local minima. In the present work, we collapse elements with small volumes after each major optimization iteration, which either eliminates the local minima of (3.9) or suggests this strategy less sensitive to local minima.

Other choices for the objective function are possible, but are not considered here. The objective function proposed in [79] is the elementwise deviation of the DG solution from its mean, which was shown to have excellent tracking properties. However, it tends to move the mesh in regions of the domain where the solution is smooth and therefore not ideal when the entire mesh is parametrized. The work in [8] uses a physics-based objective function based on the Rankine-Hugoniot conditions at all element faces for inviscid conservation laws, which was shown to work well when combined with a DG-like discretization in a minimum-residual framework.

To not only prevent the degradation of the mesh quality, but actively promote mesh smoothing, we take $\kappa \in \mathbb{R}_+$ and define f_{msh} as the deviation of the distortion of the physical

mesh from the distortion of the reference mesh

$$f_{\text{msh}}(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{R}_{\text{msh}}(\boldsymbol{x}) - \boldsymbol{R}_{\text{msh}}(\boldsymbol{X}))^T(\boldsymbol{R}_{\text{msh}}(\boldsymbol{x}) - \boldsymbol{R}_{\text{msh}}(\boldsymbol{X})), \tag{3.10}$$

where $\boldsymbol{R}_{\text{msh}} : \mathbb{R}^{N_{\boldsymbol{x}}} \to \mathbb{R}^{|\mathcal{E}_{h,q}|}$ is the algebraic system corresponding to the elementwise mesh distortion used for high-order mesh generation [41, 24]

$$r^K_{\text{msh}}(\mathcal{G}_{h,q}) \coloneqq \int_K \left(\frac{\|G_{h,q}\|_F^2}{(\det G_{h,q})_+^{2/d}} \right)^2 dv \tag{3.11}$$

and $\boldsymbol{X} \in \mathbb{R}^{N_{\boldsymbol{x}}}$ are the nodal coordinates of the reference mesh. We define $\boldsymbol{R}_{\text{msh}}$ as the deviation from the distortion of the reference mesh rather than the mesh distortion itself because the Hessian approximation used for our solver performs best when the objective function approaches zero (Section 3.4.4). Even though the distortion term prevents the combined objective function from converging to zero, if the mesh is high-quality, we expect $\boldsymbol{R}_{\text{msh}}(\boldsymbol{x})$ to be close to its minimum value (component-wise) $\boldsymbol{R}_{\text{msh}}(\boldsymbol{X})$.

To close this section, we define the following vector-valued function

$$\boldsymbol{F}(\boldsymbol{u}, \boldsymbol{x}) \coloneqq \begin{bmatrix} \boldsymbol{R}(\boldsymbol{u}, \boldsymbol{x}) \\ \kappa(\boldsymbol{R}_{\text{msh}}(\boldsymbol{x}) - \boldsymbol{R}_{\text{msh}}(\boldsymbol{X})) \end{bmatrix} \tag{3.12}$$

and re-write the objective function as

$$f(\boldsymbol{u}, \boldsymbol{x}) = \frac{1}{2}\left\|\boldsymbol{F}(\boldsymbol{u}, \boldsymbol{x})\right\|_2^2 = \frac{1}{2}\boldsymbol{R}(\boldsymbol{u}, \boldsymbol{x})^T\boldsymbol{R}(\boldsymbol{u}, \boldsymbol{x}) + \frac{\kappa^2}{2}(\boldsymbol{R}_{\text{msh}}(\boldsymbol{x}) - \boldsymbol{R}_{\text{msh}}(\boldsymbol{X}))^T(\boldsymbol{R}_{\text{msh}}(\boldsymbol{x}) - \boldsymbol{R}_{\text{msh}}(\boldsymbol{X})), \tag{3.13}$$

to emphasize the objective function is the square two-norm of a residual function. This has implications in terms of available solvers for the optimization problem in (3.1) as will be discussed in Section 3.4.4.

3.3 Boundary constraint enforcement

In order to maintain a boundary-conforming mesh, the coordinates of all mesh nodes cannot be allowed to move freely; rather, we must add boundary constraints to ensure nodes *slide* along the domain boundaries. To this end, we write the mesh node coordinates as the result of a mapping $\chi : \mathbb{R}^{N_{\boldsymbol{\phi}}} \to \mathbb{R}^{N_{\boldsymbol{x}}}$ from the unconstrained degrees of freedom $\boldsymbol{\phi} \in \mathbb{R}^{N_{\boldsymbol{\phi}}}$ that incorporates all boundary constraints

$$\boldsymbol{x} = \chi(\boldsymbol{\phi}), \tag{3.14}$$

where the specific form of the mapping depends on the domain under consideration. This constraint is incorporated into the optimization problem (3.1) as:

$$\begin{aligned} \underset{\boldsymbol{u}\in\mathbb{R}^{N_{\boldsymbol{u}}}, \boldsymbol{\phi}\in\mathbb{R}^{N_{\boldsymbol{\phi}}}}{\text{minimize}} \quad & f(\boldsymbol{u}, \chi(\boldsymbol{\phi})) \\ \text{subject to} \quad & \boldsymbol{r}(\boldsymbol{u}, \chi(\boldsymbol{\phi})) = \boldsymbol{0}. \end{aligned} \tag{3.15}$$

As discussed in [79], the re-parametrized formulation can also be used to explicitly incorporate mesh smoothing into the mesh deformation. By introducing the following definitions

$$\tilde{f}(\boldsymbol{u},\boldsymbol{\phi}) \coloneqq f(\boldsymbol{u},\chi(\boldsymbol{\phi})), \qquad \tilde{\boldsymbol{r}}(\boldsymbol{u},\boldsymbol{\phi}) \coloneqq \boldsymbol{r}(\boldsymbol{u},\chi(\boldsymbol{\phi})), \qquad \tilde{\boldsymbol{F}}(\boldsymbol{u},\boldsymbol{\phi}) \coloneqq \boldsymbol{F}(\boldsymbol{u},\chi(\boldsymbol{\phi})), \tag{3.16}$$

the optimization problem with boundary enforcement in (3.15) becomes

$$\begin{aligned} \underset{\boldsymbol{u}\in\mathbb{R}^{N_{\boldsymbol{u}}},\boldsymbol{\phi}\in\mathbb{R}^{N_{\boldsymbol{\phi}}}}{\text{minimize}} \quad & \tilde{f}(\boldsymbol{u},\boldsymbol{\phi}) \\ \text{subject to} \quad & \tilde{\boldsymbol{r}}(\boldsymbol{u},\boldsymbol{\phi}) = \boldsymbol{0}, \end{aligned} \tag{3.17}$$

which has the same structure as the original optimization problem without boundary enforcement (3.1) with the following replacements: $\tilde{f} \leftarrow f$, $\tilde{\boldsymbol{r}} \leftarrow \boldsymbol{r}$, and $\boldsymbol{\phi} \leftarrow \boldsymbol{x}$.

Most of the problems in this work only require nodes to slide along boundaries aligned with coordinate directions; the nodes on all other boundaries are fixed. Nodes sliding along more general boundaries has been considered in [10, 78]. In this special case, we partition the mesh node coordinates into the constrained $\boldsymbol{x}_c \in \mathbb{R}^{N_{\boldsymbol{x}}-N_{\boldsymbol{\phi}}}$ and unconstrained $\boldsymbol{\phi} \in \mathbb{R}^{N_{\boldsymbol{\phi}}}$ coordinates

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{\phi} \\ \boldsymbol{x}_c \end{bmatrix}, \tag{3.18}$$

which implies the boundary mapping is the padded identity mapping

$$\boldsymbol{x} = \chi(\boldsymbol{\phi};\boldsymbol{x}_c) \coloneqq \begin{bmatrix} \boldsymbol{\phi} \\ \boldsymbol{x}_c \end{bmatrix}, \qquad \frac{\partial \chi}{\partial \boldsymbol{\phi}}(\boldsymbol{\phi};\boldsymbol{x}_c) = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{0} \end{bmatrix}. \tag{3.19}$$

3.4 Full space, minimum-residual solver for optimization-based discontinuity tracking

With the formulation of the optimization problem given in (3.17), this section introduces a robust, iterative solver. For simplicity, we introduce the solver for the optimization problem in (3.1), i.e., without boundary enforcement; however, due to the mirror structure between (3.1) and (3.17), the exact algorithm applies to solve (3.17) with the following replacements: $\tilde{f} \leftarrow f$, $\tilde{\boldsymbol{r}} \leftarrow \boldsymbol{r}$, and $\boldsymbol{\phi} \leftarrow \boldsymbol{x}$.

For brevity, we combine the PDE solution $\boldsymbol{u}$ and mesh coordinates $\boldsymbol{x}$ into a single vector

$$\boldsymbol{z} \coloneqq \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{x} \end{bmatrix} \in \mathbb{R}^{N_{\boldsymbol{z}}}, \tag{3.20}$$

where $N_{\boldsymbol{z}} = N_{\boldsymbol{u}} + N_{\boldsymbol{x}}$. In the remainder, we will replace $(\boldsymbol{u},\boldsymbol{x})$ with $\boldsymbol{z}$, and vice versa, as needed. We will also abbreviate the partial derivatives of the objective function and DG residual as

$$\begin{aligned} \boldsymbol{g}_{\boldsymbol{z}}(\boldsymbol{z}) &= \frac{\partial f}{\partial \boldsymbol{z}}(\boldsymbol{z})^T \quad \boldsymbol{g}_{\boldsymbol{u}}(\boldsymbol{z}) = \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{z})^T \quad \boldsymbol{g}_{\boldsymbol{x}}(\boldsymbol{z}) = \frac{\partial f}{\partial \boldsymbol{x}}(\boldsymbol{z})^T \\ \boldsymbol{J}_{\boldsymbol{z}}(\boldsymbol{z}) &= \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{z}}(\boldsymbol{z}), \quad \boldsymbol{J}_{\boldsymbol{u}}(\boldsymbol{z}) = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{z}), \quad \boldsymbol{J}_{\boldsymbol{x}}(\boldsymbol{z}) = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}}(\boldsymbol{z}). \end{aligned} \tag{3.21}$$

3.4.1 Sequential quadratic programming solver

The proposed solver is a sequential quadratic programming (SQP) method [5] that uses a sequence of quadratic programs to solve (3.1). Let $\boldsymbol{z}_k$ denote the current iterate and formulate a quadratic subproblem by linearizing the constraint about $\boldsymbol{z}_k$ and using a second-order Taylor series centered at $\boldsymbol{z}_k$ to approximate the Lagrangian function

$$\begin{aligned} \underset{\Delta \boldsymbol{z} \in \mathbb{R}^{N_{\boldsymbol{z}}}}{\text{minimize}} \quad & \boldsymbol{g}_{\boldsymbol{z}}(\boldsymbol{z}_k)^T \Delta \boldsymbol{z} + \frac{1}{2} \Delta \boldsymbol{z}^T \boldsymbol{B}(\boldsymbol{z}_k) \Delta \boldsymbol{z} \\ \text{subject to} \quad & \boldsymbol{r}(\boldsymbol{z}_k) + \boldsymbol{J}_{\boldsymbol{z}}(\boldsymbol{z}_k) \Delta \boldsymbol{z} = \boldsymbol{0}, \end{aligned} \tag{3.22}$$

where $\boldsymbol{B}(\boldsymbol{z}) \in \mathbb{R}^{N_{\boldsymbol{z}} \times N_{\boldsymbol{z}}}$ a symmetric positive definite approximation of the Hessian of $\mathcal{L}(\boldsymbol{z})$, i.e.,

$$\boldsymbol{B}(\boldsymbol{z}) \coloneqq \begin{bmatrix} \boldsymbol{B}_{\boldsymbol{uu}}(\boldsymbol{z}) & \boldsymbol{B}_{\boldsymbol{ux}}(\boldsymbol{z}) \\ \boldsymbol{B}_{\boldsymbol{ux}}(\boldsymbol{z})^T & \boldsymbol{B}_{\boldsymbol{xx}}(\boldsymbol{z}) \end{bmatrix} \tag{3.23}$$

such that

$$\begin{aligned} \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{u} \partial \boldsymbol{u}}(\boldsymbol{z}) &\approx \boldsymbol{B}_{\boldsymbol{uu}}(\boldsymbol{z}) \in \mathbb{R}^{N_{\boldsymbol{u}} \times N_{\boldsymbol{u}}} \\ \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{u} \partial \boldsymbol{x}}(\boldsymbol{z}) &\approx \boldsymbol{B}_{\boldsymbol{ux}}(\boldsymbol{z}) \in \mathbb{R}^{N_{\boldsymbol{u}} \times N_{\boldsymbol{x}}} \\ \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{x} \partial \boldsymbol{x}}(\boldsymbol{z}) &\approx \boldsymbol{B}_{\boldsymbol{xx}}(\boldsymbol{z}) \in \mathbb{R}^{N_{\boldsymbol{x}} \times N_{\boldsymbol{x}}}. \end{aligned} \tag{3.24}$$

and the dependence on the Lagrange multipliers has been dropped. Once the quadratic subproblem is solved to obtain the current search direction $\Delta \boldsymbol{z}_{k+1} \in \mathbb{R}^{N_{\boldsymbol{z}}}$, the current iterate is updated according to

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + \alpha_{k+1} \Delta \boldsymbol{z}_{k+1} \tag{3.25}$$

to yield a sequence of iterates $\{\boldsymbol{z}_k\}$ where $\alpha_{k+1} \in (0, 1]$ is a step length parameter (Section 3.4.3). If $\boldsymbol{B}$ is the true Hessian and $\alpha = 1$, this method is equivalent to Newton's method applied to the first-order optimality conditions of (3.1) [5] and therefore will converge quadratically provided the initial guess is sufficiently close to a local minima.

3.4.2 Linear subproblem

The first-order optimality condition of the quadratic program (3.22) leads to the following linear system of equations

$$\begin{bmatrix} \boldsymbol{B}_{\boldsymbol{uu}}(\boldsymbol{z}_k) & \boldsymbol{B}_{\boldsymbol{ux}}(\boldsymbol{z}_k) & \boldsymbol{J}_{\boldsymbol{u}}(\boldsymbol{z}_k)^T \\ \boldsymbol{B}_{\boldsymbol{ux}}(\boldsymbol{z}_k)^T & \boldsymbol{B}_{\boldsymbol{xx}}(\boldsymbol{z}_k) & \boldsymbol{J}_{\boldsymbol{x}}(\boldsymbol{z}_k)^T \\ \boldsymbol{J}_{\boldsymbol{u}}(\boldsymbol{z}_k) & \boldsymbol{J}_{\boldsymbol{x}}(\boldsymbol{z}_k) & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{u}_{k+1} \\ \Delta \boldsymbol{x}_{k+1} \\ \boldsymbol{\eta}_{k+1} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{g}_{\boldsymbol{u}}(\boldsymbol{z}_k) \\ \boldsymbol{g}_{\boldsymbol{x}}(\boldsymbol{z}_k) \\ \boldsymbol{r}(\boldsymbol{z}_k) \end{bmatrix}, \tag{3.26}$$

where $\boldsymbol{\eta}_{k+1} \in \mathbb{R}^{N_{\boldsymbol{u}}}$ are the Lagrange multipliers associated with the linearized constraint and the step $\Delta \boldsymbol{z}_{k+1}$ is decomposed as a step in the PDE state $\Delta \boldsymbol{u}_{k+1}$ and the nodal mesh

coordinates $\Delta \boldsymbol{x}_{k+1}$

$$\Delta \boldsymbol{z}_{k+1} = \begin{bmatrix} \Delta \boldsymbol{u}_{k+1} \\ \Delta \boldsymbol{x}_{k+1} \end{bmatrix}. \tag{3.27}$$

The state update $\Delta \boldsymbol{u}$ and Lagrange multipliers $\boldsymbol{\eta}$ can be eliminated from the linear system to obtain an explicit expression for the state and mesh updates

$$\Delta \boldsymbol{x}_{k+1} = -\boldsymbol{A}(\boldsymbol{z}_k)^{-1}\boldsymbol{b}(\boldsymbol{z}_k), \qquad \Delta \boldsymbol{u}_{k+1} = \Delta \boldsymbol{u}_0(\boldsymbol{z}_k) + \boldsymbol{C}(\boldsymbol{z}_k)\Delta \boldsymbol{x}, \tag{3.28}$$

where

$$\begin{aligned} \Delta \boldsymbol{u}_0(\boldsymbol{z}) &= -\boldsymbol{J}_{\boldsymbol{u}}(\boldsymbol{z})^{-1}\boldsymbol{r}(\boldsymbol{z}) \\ \boldsymbol{C}(\boldsymbol{z}) &= -\boldsymbol{J}_{\boldsymbol{u}}(\boldsymbol{z})^{-1}\boldsymbol{J}_{\boldsymbol{x}}(\boldsymbol{z}) \\ \boldsymbol{A}(\boldsymbol{z}) &= \boldsymbol{B}_{\boldsymbol{xx}}(\boldsymbol{z}) + 2\boldsymbol{C}(\boldsymbol{z})^T\boldsymbol{B}_{\boldsymbol{ux}}(\boldsymbol{z}) + \boldsymbol{C}(\boldsymbol{z})^T\boldsymbol{B}_{\boldsymbol{uu}}(\boldsymbol{z})\boldsymbol{C}(\boldsymbol{z}) \\ \boldsymbol{b}(\boldsymbol{z}) &= \boldsymbol{g}_{\boldsymbol{x}}(\boldsymbol{z}) + \boldsymbol{C}(\boldsymbol{z})^T\boldsymbol{g}_{\boldsymbol{u}}(\boldsymbol{z}) + \left(\boldsymbol{B}_{\boldsymbol{ux}}(\boldsymbol{z})^T + \boldsymbol{C}(\boldsymbol{z})^T\boldsymbol{B}_{\boldsymbol{uu}}(\boldsymbol{z})\right)\Delta \boldsymbol{u}_0. \end{aligned} \tag{3.29}$$

Notice $\Delta \boldsymbol{u}_0(\boldsymbol{z})$ is the Newton step for the nonlinear system $\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{x}) = \boldsymbol{0}$ if $\boldsymbol{x}$ is fixed and $\boldsymbol{C}(\boldsymbol{z})$ is the sensitivity of the solution $\boldsymbol{u}$ with respect to the mesh nodes $\boldsymbol{x}$.

The linear systems in (3.26) and (3.28) are two mathematically equivalent options to compute the solution to the SQP subproblem that place different requirements on the linear solver. The system in (3.26) is large (size: $2N_{\boldsymbol{u}} + N_{\boldsymbol{x}}$) since it simultaneously computes the steps $\Delta \boldsymbol{u}$, $\Delta \boldsymbol{x}$ and Lagrange multipliers $\boldsymbol{\eta}$, but each (block) entry is relatively easy to form. On the other hand, the system in (3.28) is smaller (size: $N_{\boldsymbol{x}}$), but formation of $\boldsymbol{A}(\boldsymbol{z}_k)$ requires computation of $\boldsymbol{C}(\boldsymbol{z}_k)$, which is impractical if a direct solver is not available. In this work, we use a direct solver to solve (3.26); however, this approach is not practical for large-scale problems. In future work, we will develop iterative solvers and preconditioners for the full system in (3.26).

3.4.3 Line search globalization

To ensure the sequence $\{\boldsymbol{z}_k\}$ converges to a first-order critical point of (3.1) from an arbitrary initial guess, the SQP algorithm must be globalized with e.g., a trust region strategy or line search. We choose to globalize the SQP method with a line search that computes the step length such that it minimizes a merit function that combines the objective function and a measure of constraint violation into a scalar function. In this work, we use the ℓ_1 penalty function $\varphi_k : \mathbb{R} \to \mathbb{R}$

$$\varphi_k(\alpha) \coloneqq f(\boldsymbol{z}_k + \alpha \Delta \boldsymbol{z}_k) + \mu \left\| \boldsymbol{r}(\boldsymbol{z}_k + \alpha \Delta \boldsymbol{z}_{k+1}) \right\|_1 \tag{3.30}$$

where $\mu > 0$ is the penalty parameter. This is an exact merit function in the sense that there exists a positive scalar $\hat{\mu}$ such that for any $\mu > \hat{\mu}$, any local solution of (3.1) is a local minimizer of (3.30) [53]. Furthermore, $\hat{\mu} = \left\| \boldsymbol{\lambda}^\star \right\|_\infty$, where $\boldsymbol{\lambda}^\star$ are the Lagrange multipliers associated with the optimal solution $(\boldsymbol{u}^\star, \boldsymbol{x}^\star)$ [53]. Therefore, in the present setting we take $\mu = 2 \left\| \hat{\boldsymbol{\lambda}}(\boldsymbol{z}_k) \right\|_\infty$, where $\hat{\boldsymbol{\lambda}}$ is defined in (3.5).

It is well-known that it is not necessary to find the exact minimizer of the merit function to obtain a convergent algorithm. Instead, we search for $\alpha_{k+1} \in (0, 1]$ that satisfies *sufficient decrease*

$$\varphi_k(\alpha_{k+1}) \leq \varphi_k(0) + c\alpha_{k+1}\varphi_k'(0), \tag{3.31}$$

where $c \in (0, 1)$. We use a backtracking strategy [53] to determine α_{k+1}: define $\alpha_{k+1} = \tau^{n-1}$ for $\tau \in (0, 1)$ and let $n \in \mathbb{N}$ be the smallest number such that sufficient decrease (3.31) holds. In this work we make standard choices for these parameters [53]: $c = 10^{-4}$ and $\tau = 0.5$.

3.4.4 Levenberg-Marquardt Hessian approximation

The true Hessian of the Lagrangian in (3.2) is

$$\boldsymbol{H}(\boldsymbol{z}) = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{z}}(\boldsymbol{z})^T \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{z}}(\boldsymbol{z}) + \boldsymbol{F}_i(\boldsymbol{z}) \frac{\partial^2 \boldsymbol{F}_i}{\partial \boldsymbol{z} \partial \boldsymbol{z}}(\boldsymbol{z}) - \boldsymbol{\lambda}_i \frac{\partial^2 \boldsymbol{r}_i}{\partial \boldsymbol{z} \partial \boldsymbol{z}}(\boldsymbol{z}) \tag{3.32}$$

where summation is implied over the repeated index. The second and third term involves the Hessian of the DG residual, which is a complicated third-order tensor that is rarely available in computational mechanics codes. Therefore, we use the Gauss-Newton assumption and approximate the Hessian as

$$\boldsymbol{H}(\boldsymbol{z}) \approx \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{z}}(\boldsymbol{z})^T \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{z}}(\boldsymbol{z}), \tag{3.33}$$

which is justified if the combined enriched residual and mesh distortion $\boldsymbol{F}(\boldsymbol{z})$ is small. While this approximation is convenient, it could also lead to singular or ill-conditioned Hessian approximations, which would in turn lead to poor search directions. Therefore, we use the Levenberg-Marquardt approach that adds a scaled multiple of a symmetric positive definite matrix to regularize the system. Based on the observations in [8] that Levenberg-Marquardt regularization improves the mesh motion and is not needed for the state, we only regularize the mesh components of the Hessian, i.e.,

$$\boldsymbol{B}_{\boldsymbol{uu}}(\boldsymbol{z}) = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}(\boldsymbol{z})^T \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}(\boldsymbol{z}), \quad \boldsymbol{B}_{\boldsymbol{ux}}(\boldsymbol{z}) = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}(\boldsymbol{z})^T \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}(\boldsymbol{z}), \quad \boldsymbol{B}_{\boldsymbol{xx}}(\boldsymbol{z}) = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}(\boldsymbol{z})^T \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}(\boldsymbol{z}) + \gamma \boldsymbol{D}, \tag{3.34}$$

where $\gamma \in \mathbb{R}_+$ and $\boldsymbol{D} \in \mathbb{R}^{N_{\boldsymbol{x}} \times N_{\boldsymbol{x}}}$ is a symmetric positive definite (SPD) matrix (Section 3.4.5).

3.4.5 Choice of regularization matrix

The classic Levenberg-Marquardt algorithm uses the identity matrix as the regularization matrix $\boldsymbol{D} = \boldsymbol{I}_{N_{\boldsymbol{x}}}$ to guard against poor search directions that could result if the Jacobian of $\boldsymbol{F}$ with respect to $\boldsymbol{x}$ is rank deficient or ill-conditioned. However, the same result can be achieved for any SPD matrix $\boldsymbol{D}$. We consider a regularization matrix that is known to possess smoothing properties: the stiffness matrix of a linear elliptic PDE. To this end,

define a vector-valued function $v : \Omega_0 \to \mathbb{R}^d$ in which each component satisfies the elliptic PDE with homogeneous Neumann boundary conditions

$$\nabla_X \cdot (k \nabla_X v_i) = 0 \quad \text{in} \quad \Omega_0, \qquad \nabla_X v_i \cdot N = 0 \quad \text{on} \quad \partial\Omega_0 \tag{3.35}$$

for $i = 1, \ldots, d$ and the coefficient $k : \Omega_0 \to \mathbb{R}_+$ is piecewise constant over each element in $\mathcal{E}_{h,q}$. Numerical experimentation on problems where elements in the reference mesh significantly vary in size indicate that the natural scaling of the stiffness matrix ($k = 1$) is not sufficient to provide search directions that are scaled according to the mesh resolution. Instead, we take the piecewise constant coefficient as

$$k(x) = \omega_K, \quad x \in K \tag{3.36}$$

for each $K \in \mathcal{E}_{h,q}$, where $\omega_K \in \mathbb{R}_+$ is inversely proportional to the size of the element

$$\omega_K \coloneqq \frac{\min\limits_{K' \in \mathcal{E}_{h,q}} |K'|}{|K|}. \tag{3.37}$$

The regularization matrix is the assembled stiffness matrix associated with the elliptic system in (3.35)-(3.36) over the global finite element space $\mathcal{W}_{h,q}$. This is similar to the regularization matrix used in [10] without volume-based scaling of the elliptic coefficient, which is important for problems with reference domain elements of varying size (Section 4.4.2). Boundary constraints are incorporated into the elliptic regularization matrix as homogeneous Dirichlet boundary conditions.

3.4.6 Adaptive regularization parameter

The ideal value for the regularization parameter γ in (3.34) is difficult to know *a priori*. Large values of γ produce highly regularized search directions $\Delta\boldsymbol{x}$, e.g., the search direction becomes the solution of the elliptic PDE with the steepest descent direction as the right-hand side; however, in this case, the Hessian approximation is poor. Alternatively, small values of γ, particularly in early iterations, can leads to bad search directions due to ill-conditioning of the Hessian approximation. Following the work in [8], we introduce an adaptive algorithm to minimize the importance of choosing the appropriate value for γ *a priori*. Our algorithm is based on the heuristic that γ should control the size of the mesh deformation at a given iteration. Therefore, if a step is too large, we increase the value of γ for the next iteration and vice versa. Let $\gamma_k > 0$ be the value of the regularization parameter at iteration k and define constants $\kappa_1, \kappa_2 > 0$, $\sigma \in (0, 1)$, $\gamma_{min} > 0$ then

$$\gamma_{k+1} = \max\{\bar{\gamma}_{k+1}, \gamma_{min}\}, \qquad \bar{\gamma}_{k+1} = \begin{cases} \sigma^{-1}\gamma_k & \text{if } \|\Delta\boldsymbol{x}_k\| < \kappa_1 \\ \sigma\gamma_k & \text{if } \|\Delta\boldsymbol{x}_k\| > \kappa_2 \\ \gamma_k & \text{otherwise.} \end{cases} \tag{3.38}$$

For domains with dimensions $\mathcal{O}(1)$, we take $\kappa_1 = 10^{-2}$ and $\kappa_2 = 10^{-1}$. In this work, we choose a conservative value $\sigma = 0.5$ to prevent γ from changing significantly between iterations. The appropriate value of γ_{min} is problem-dependent and will be addressed in Section 4.

3.4.7 Termination criteria

The termination criteria for the SQP method comes from the first-order optimality criteria discussed in Section 3.1 (3.6). That is, given tolerances $\epsilon_1, \epsilon_2 > 0$, $\boldsymbol{z}_k$ is a considered a numerical solution of (3.1) if

$$\|\boldsymbol{c}(\boldsymbol{z}_k)\| < \epsilon_1, \qquad \|\boldsymbol{r}(\boldsymbol{z}_k)\| < \epsilon_2. \tag{3.39}$$

We have empirically observed that feasibility can be driven to near machine tolerance (take $\epsilon_2 = 10^{-10}$); however, the optimality condition is a more difficult condition (take $\epsilon_1 = 10^{-5}$). For difficult problems, it may require a large number of iterations, so we safeguard the algorithm with a maximum number of iterations (N_{max}), i.e., iterations terminate when either the convergence criteria (3.39) are met or a N_{max} iterations have been completed. If termination is based on N_{max}, it is likely the DG constraint is not satisfied, i.e.,

$$\boldsymbol{r}(\boldsymbol{z}_{N_{max}}) > \epsilon_2, \tag{3.40}$$

which could lead to a non-conservative scheme. In this case, we solve the DG equations on the fixed mesh with nodal coordinates $\boldsymbol{x}_{N_{max}}$ using Newton's method.

3.5 Extension to the unsteady case by the method of lines

In this section, we extend the high-order implicit shock tracking framework from Sections 3.1 to 3.4 ([79, 81]) to time-dependent problems using a method of lines approach (in contrast to the space-time approach in [8, 81]). We recast the fully discrete conservation law as a PDE-constrained optimization problem over the discrete solution and mesh that aims to align element faces with discontinuities at each Runge-Kutta stage for every timestep.

We formulate the problem of tracking discontinuities at a given stage i and timestep n as a constrained optimization problem over the PDE state and mesh stage that minimizes an objective function, $f_{n,i} : \mathbb{R}^{N_{\boldsymbol{u}}} \times \mathbb{R}^{N_{\boldsymbol{x}}} \to \mathbb{R}$, while enforcing the DG-DIRK discretization. That is, we define $\boldsymbol{u}_{n,i} \in \mathbb{R}^{N_{\boldsymbol{u}}}$ and $\boldsymbol{x}_{n,i} \in \mathbb{R}^{N_{\boldsymbol{x}}}$ as

$$(\boldsymbol{u}_{n,i}, \boldsymbol{x}_{n,i}) := \underset{\boldsymbol{w}\in\mathbb{R}^{N_{\boldsymbol{u}}}, \boldsymbol{y}\in\mathbb{R}^{N_{\boldsymbol{x}}}}{\arg\min} \; f_{n,i}(\boldsymbol{w}, \boldsymbol{y}) \quad \text{subject to:} \quad \boldsymbol{r}_{n,i}(\boldsymbol{w}, \boldsymbol{y}) = \boldsymbol{0}. \tag{3.41}$$

for $n = 1, \ldots, N_T$ and $i = 1, \ldots, s$. For a fixed time step n, once the stage states $\{\boldsymbol{u}_{n,i}\}_{i=1}^s$ and meshes $\{\boldsymbol{x}_{n,i}\}_{i=1}^s$ are computed, the state and mesh can be advanced to the next time

step ($\boldsymbol{u}_{n+1}$ and $\boldsymbol{x}_{n+1}$) using the relationships in (2.50). The DIRK schemes considered in this work (Table 2.2) satisfy the property that $A_{si} = b_i$ for $i = 1, \ldots, s$, which implies the state and mesh at the final stage of time step n are identical to the state and mesh at time step $n+1$, i.e., $\boldsymbol{u}_{n+1} = \boldsymbol{u}_{n,s}$ and $\boldsymbol{x}_{n+1} = \boldsymbol{x}_{n,s}$.

The objective function is constructed such that the solution of the PDE-constrained optimization problem is a feature-aligned mesh by using the norm of the enriched DG-DIRK residual

$$f_{n,i} : (\boldsymbol{w}, \boldsymbol{y}) \mapsto \frac{1}{2} \left\| \boldsymbol{R}_{n,i}(\boldsymbol{w}, \boldsymbol{y}) \right\|_2^2 . \tag{3.42}$$

This is the same objective function as (3.1) for steady conservation laws, but unlike that case we do not include the mesh quality term in the objective function for the unsteady (method of lines) case. In the steady case, there is no information about the shock location *a priori*, which usually requires significant deformation to the initial mesh to align with shocks, necessitating the use of a mesh regularization term in the objective function. However, in the context of timestepping, we have useful information from the previous timestep to use as an initial guess for both the state and mesh. We can combine this information along with the Rankine-Hugoniot conditions and a high-order mesh smoothing procedure to obtain an excellent initial guess for the mesh at each Runge-Kutta stage (Section 5.1.1). In a sense, the additional time dimension allows the mesh regularization to be decoupled from the implicit shock tracking procedure, which is one advantage of the method of lines discretization over the space-time formulation for solving unsteady problems.

Because the optimization problem (3.41) *exactly matches the form* of the optimization problem for the steady case (3.1) with the identical form of the Lagrangian (3.2), we use the same SQP solver from Section 3.4 to define the sequences $\{\boldsymbol{u}^{(k)}\}_{k=0}^{\infty} \subset \mathbb{R}^{N_{\boldsymbol{u}}}$ and $\{\boldsymbol{x}^{(k)}\}_{k=0}^{\infty} \subset \mathbb{R}^{N_{\boldsymbol{x}}}$ as

$$\boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{(k)} + \alpha_{k+1} \Delta \boldsymbol{u}^{(k+1)}, \qquad \boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_{k+1} \Delta \boldsymbol{x}^{(k+1)}, \tag{3.43}$$

for $k = 0, 1, \ldots,$ where $\alpha_{k+1} \in (0, 1]$ is a step length which can be determined by a line search procedure, and $\Delta \boldsymbol{u}^{(k+1)} \in \mathbb{R}^{N_{\boldsymbol{u}}}$ and $\Delta \boldsymbol{x}^{(k+1)} \in \mathbb{R}^{N_{\boldsymbol{x}}}$ are search directions. At a given iteration k, the search directions $\Delta \boldsymbol{u}^{(k)}$ and $\Delta \boldsymbol{x}^{(k)}$ are computed simultaneously as the solution of a quadratic approximation to the optimization problem in (3.41) with a regularized Levenberg-Marquardt approximation of the Hessian. A pair $(\boldsymbol{u}, \boldsymbol{x})$ is considered a solution of (3.41) if $\|\boldsymbol{c}(\boldsymbol{u}, \boldsymbol{x})\| < \epsilon_1$ and $\|\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{x})\| < \epsilon_2$ for tolerances $\epsilon_1, \epsilon_2 > 0$.

Chapter 4

Numerical experiments: steady case

In this section we introduce three steady inviscid conservation laws and demonstrate the tracking framework on six problems with discontinuous solutions of varying difficulty. We also provide a detailed study of the various algorithmic parameters introduced in Sections 2 - 3, in particular, the choice of numerical flux ($\mathcal{H}$) and mesh distortion parameter (κ). We first discuss a variety of practical considerations needed (Section 4.1) and explain how each gradually factors into the solver in practice through a series of numerical experiments (Sections 4.2 - 4.4).

4.1 Practical considerations

4.1.1 Solution and mesh initialization

The discontinuity-tracking optimization problem in (3.1) is non-convex and therefore the initial guess for the SQP solver is critical to obtain a good solution. In the present context, this means we must provide a reasonable initial guess for the mesh coordinates $\boldsymbol{x}_0$ and DG solution $\boldsymbol{u}_0$.

Special case: straight-sided mesh ($q = 1$)

First consider the case of a straight-sided mesh ($q = 1$) combined with any finite element space for the solution ($p \geq 0$). The mesh is always initialized from the reference mesh which comes from mesh generation agnostic to the discontinuity because we usually do not have an estimate of the discontinuity surface. In special cases where an estimate of the discontinuity surface is available, this could be used to drive generation of the reference mesh. The DG solution is initialized from the DG ($p = 0$) solution on the reference mesh; $p = 0$ is used because nonlinear instabilities resulting from oscillations about discontinuities cannot arise with a piecewise constant solution field. For nonlinear problems the $p = 0$ solution is obtained using pseudo-transient continuation [38] with adaptive time steps, initialized from uniform flow.

General case: high-order meshes ($q > 1$)

To avoid local minima in the optimization problem (3.1) that arise when high-order meshes are used, we usually initialize the tracking problem for $p \geq 0$, $q > 1$ from the solution of the tracking problem for $p' \leq p$, $q' = 1$. That is, we solve the the tracking problem using a solution space with polynomial degree $p' \leq p$ and mesh deformation with polynomial degree $q' = 1$ and initialize the desired tracking problem (solution space of degree p and mesh of degree $q > 1$) from the resulting DG solution and mesh. This strategy comes from our observations that $q = 1$ tracking is quite robust and convergence of the $q > 1$ solution from a straight-sided tracking mesh is rapid. For difficult problems, it may be helpful to use continuation on the polynomial degree for the solution as well, i.e., take $p' < p$.

4.1.2 Edge collapses and solution transfer

As we will show in our numerical experiments in Sections 4.2, we typically start with an initial mesh that is far from alignment with the shock and iterate using an SQP solver which attempts to move nodes onto the shock. Since the deformation of the initial mesh can be quite large, this can result in severely ill-conditioned elements that drastically degrade the quality of our solution. To address this, we follow the approach in [8] and collapse elements once they become problematic. In particular, after each SQP iteration, we compute the volume of each element K in the physical domain, v^K, and compare it to the volume of the corresponding element in the reference domain, v_0^K, where

$$v_0^K = \int_K dV, \qquad v^K = \int_{\mathcal{G}_{h,q}(K)} dV. \tag{4.1}$$

If the volume of an element has decreased more than a certain factor, i.e., $v^K < \epsilon v_0^K$ (where $\epsilon = 0.2$ in this work), the element is tagged for removal. In principle, κ in the objective function (3.8) could be chosen large enough to prevent these ill-conditioned elements from ever appearing. However, such a κ would weight f_{msh} too heavily over f_{err} and interfere with the tracking capabilities of the method. Therefore, this weighting term κ is only chosen large enough to prevent unacceptably bad elements (such as tangled ones) which would cause the mesh distortion term f_{msh} to blow up dramatically. As long as we avoid this situation, we can handle the remainder of the ill-conditioned elements through collapses.

Each element in the physical domain $\boldsymbol{x}$ tagged for removal is eliminated by an edge collapse [46]. For a given tagged element, we choose to collapse the shortest edge into the longest edge (Figure 4.1). This is based off the principle that the shortest edge is likely to be transverse to the shock and the longest edge is likely to be aligned with the shock. A special case to note is when tagged elements have nodes or edges in common. Additional care must be taken to choose an edge to collapse that is consistent with all affected elements. For elements on the boundary, this logic is slightly modified to ensure that an edge collapse does not move nodes off the boundary. Note that an edge collapse must also be applied to

the corresponding element in the reference domain $\boldsymbol{X}$ to ensure the physical and reference domains always have the same topology.

The solution $\boldsymbol{u}$ is transferred to the new mesh by removing the entries corresponding to the degrees of freedom in the collapsed elements. To update the data structures for both the mesh and the solution, we simply need to delete the entries in the element connectivity matrix and solution vector corresponding to the collapsed elements and renumber based off the new node numbering. Because of our choice of a DG discretization, this update is particularly easy to do, and furthermore, removes the need to modify the degrees of freedom in the neighboring elements. In a CG framework, the degrees of freedom in the neighboring elements would need to be modified in order to guarantee continuity on the new mesh.

While these edge collapses do modify the objective function and formally result in a new optimization problem, in practice this does not lead to issues in the convergence of the solver. We observe that these collapses mostly occur in the initial iterations of the SQP solver when the mesh is far from convergence and each iteration results in a large update to the mesh. At some point, the line search described in Section 3.4.3 will reject steps that would cause large deformations in the mesh, hence precluding the need for further collapses and ensuring that they do not occur indefinitely.

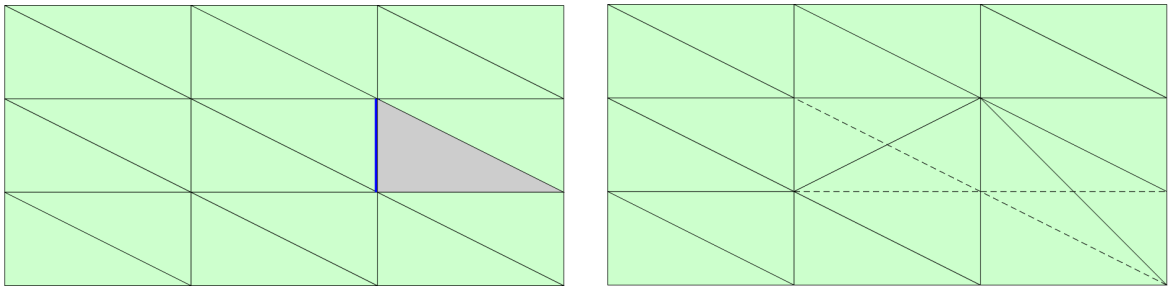


Figure 4.1: Demonstration of edge collapse algorithm: the element identified in the original mesh (*left*) is collapsed along the highlighted edge to produce the new mesh (*right*) with the original elements shown in dashed lines for reference.

4.2 Linear advection

The first conservation law we consider is steady linear advection of a scalar quantity $U : \Omega \to \mathbb{R}$ through a domain $\Omega \subset \mathbb{R}^d$

$$\nabla \cdot (\beta U) = 0 \quad \text{in } \Omega, \qquad U = U_\infty \quad \text{on } \Gamma_i, \tag{4.2}$$

where $\beta : \Omega \to \mathbb{R}^d$ is the local flow direction, $\Gamma_i \coloneqq \{x \in \partial\Omega \mid \beta \cdot n(x) < 0\}$ is the inflow boundary, $\partial\Omega : \mathbb{R}^d \to$ is the unit outward normal to the boundary, and $U_\infty : \Gamma_i \to \mathbb{R}$ is the inflow boundary condition. We consider the pure upwind numerical flux $\mathcal{H}_{\text{up}}$ introduced in (2.30) because it satisfies conditions (i)-(ii) (Section 2.3) for a stable DG discretization

and condition (iii) to be suitable for tracking, as well as its smoothed version $\mathcal{H}^a_{\text{up}}$ to recover smoothness with respect to variations of the mesh nodes (iv). In this section we take the smoothness parameter to be $a = 10$ to trade-off between smoothness and appropriate upwinding (using the smoothed Heaviside function 2.2) The boundary condition is enforced via the pure upwind numerical flux evaluated at the interior state U^+ and the boundary state $U^\partial = U_\infty$.

4.2.1 Straight shock, piecewise constant solution

First we consider a two-dimensional ($d = 2$) domain $\Omega \coloneqq (-1, 1) \times (0, 1)$ with constant advection field and piecewise constant boundary condition

$$\beta(x) = \begin{bmatrix} -1.25 \\ 1 \end{bmatrix}, \quad U_\infty(x) = H(x). \tag{4.3}$$

This leads to a linear discontinuity surface $\Gamma_s \subset \Omega$

$$\Gamma_s \coloneqq \{(-1.25s, s) \mid s \in (0, 0.8)\} \tag{4.4}$$

and piecewise constant solution field

$$U(x) = H(x_1 + 1.25x_2). \tag{4.5}$$

To ensure the Heaviside function on the bottom (inflow) boundary is accurately represented and integrated in the weak form, we require our computational mesh to have an element face that intersects $(0, 0)$ and do not allow the corresponding node to move throughout iterations using the boundary mapping described in Section 3.3.

Given the piecewise constant solution and linear shock surface, the solution lies in a $p = 0$ polynomial basis on a piecewise linear mesh $q = 1$, provided the mesh tracks the discontinuity. To demonstrate the performance of the tracking method, we use this minimal basis with $p = 0$, $q = 1$. The reference mesh is taken to be a uniform triangular mesh of the domain with 36 elements. The mesh and solution are initialized according to Section 4.1.1. The SQP solver is used with λ chosen adaptively (Section 3.4.6). The various DG/tracking parameters are set as follows: $\kappa = 0$ (no mesh smoothing), pure upwind numerical flux ($a = \infty$), $\gamma_0 = 10^{-2}$ and $\gamma_{min} = 10^{-8}$ (regularization parameter adaptivity), and $\epsilon_1 = 10^{-10}$, and $\epsilon_2 = 10^{-12}$ (termination criteria). After only 10 iterations, the mesh perfectly tracks the discontinuity (Figure 4.2) and the DG solution closely matches the exact solution; the L^1 error of the solution is 3.84×10^{-11}. For this problem where the finite element subspace contains the exact solution of the problem, the tracking method exhibits Newton-like convergence (Figure 4.3). Furthermore, the tracking solver is robust with regard to the various algorithmic parameters introduced, i.e., fast convergence to the exact solution was obtained without mesh smoothing, with the upwind numerical flux (not smooth with respect to domain deformation), and nearly independent of the choice of regularization adaptivity (γ_0, γ_{min}). The choice of these parameters becomes significant for curved discontinuities and nontrivial flows as we will demonstrate in subsequent sections.

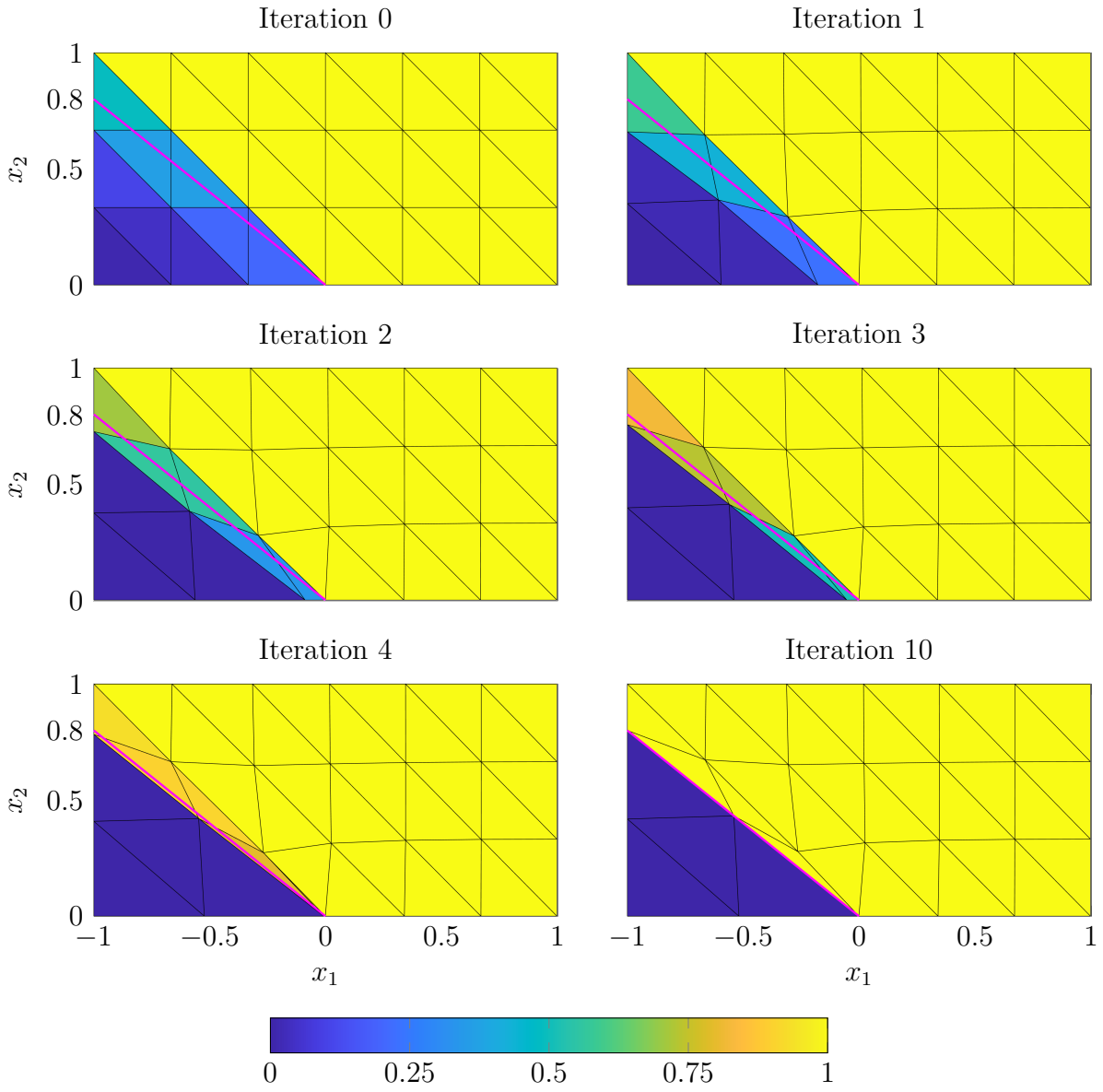


Figure 4.2: Solution of advection equation with a straight shock (constant advection field) using the tracking method at various iterations throughout the solution procedure using $p = 0$ basis for the solution and $q = 1$ basis for the mesh. The magenta line highlights the discontinuity surface of the exact solution. The method converges to the exact solution, which lies in the chosen finite element subspace, to near machine precision in only 10 iterations.

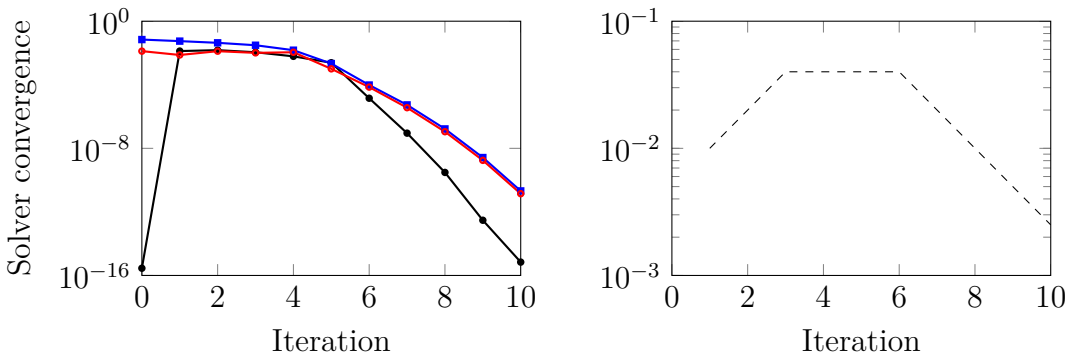


Figure 4.3: Convergence of the DG residual $\|\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{x})\|$ (—•—), enriched DG residual $\|\boldsymbol{R}(\boldsymbol{u}, \boldsymbol{x})\|$ (—■—), optimality condition $\|\boldsymbol{c}(\boldsymbol{u}, \boldsymbol{x})\|$ (—•—), and control of the regularization parameter (- - -) for the tracking method applied to the advection equation with a straight shock (constant advection field). For this simple problem, Newton-like convergence is achieved.

4.2.2 Curved shock, piecewise constant solution

Next we consider linear advection through a two-dimensional ($d = 2$) domain $\Omega := (-1, 1) \times (0, 1)$ with a spatially varying advection field and piecewise constant boundary condition

$$\beta(x) = \begin{bmatrix} -\sin(\pi x_2) \\ 1 \end{bmatrix}, \quad U_\infty(x) = H(x). \tag{4.6}$$

This leads to a trigonometric discontinuity surface $\Gamma_s \subset \Omega$

$$\Gamma_s := \left\{ \left(\frac{\cos(\pi s) - 1}{\pi}, s \right) \mid s \in (0, 1) \right\} \tag{4.7}$$

and piecewise constant solution field

$$U(x) = H(\pi x_1 - \cos(\pi x_2) + 1). \tag{4.8}$$

Unlike the previous problem, this solution cannot be represented exactly using polynomial basis functions since the discontinuity surface is non-polynomial (trigonometric). We use a piecewise constant solution basis ($p = 0$) and piecewise linear, quadratic, and cubic basis for the mesh ($q = 1, 2, 3$). The reference mesh is taken to be a uniform triangular mesh of the domain with 64 elements. The mesh and solution are initialized according to Section 4.1.1; however, for this problem we do not use continuation in the polynomial degree. The SQP solver is used with λ chosen adaptively (Section 3.4.6). The various DG/tracking parameters are set as follows: $\kappa = 0$ (mesh smoothing), smoothed upwind numerical flux ($a = 10$), $\gamma_0 = 10^{-2}$ and $\gamma_{min} = 10^{-3}$ (regularization parameter adaptivity), and $N_{max} = 80$, $\epsilon_1 = 10^{-10}$, $\epsilon_2 = 10^{-12}$ (termination criteria). For all polynomial degrees considered, the mesh tracks the discontinuity as accurately as possible given the resolution in the finite element space

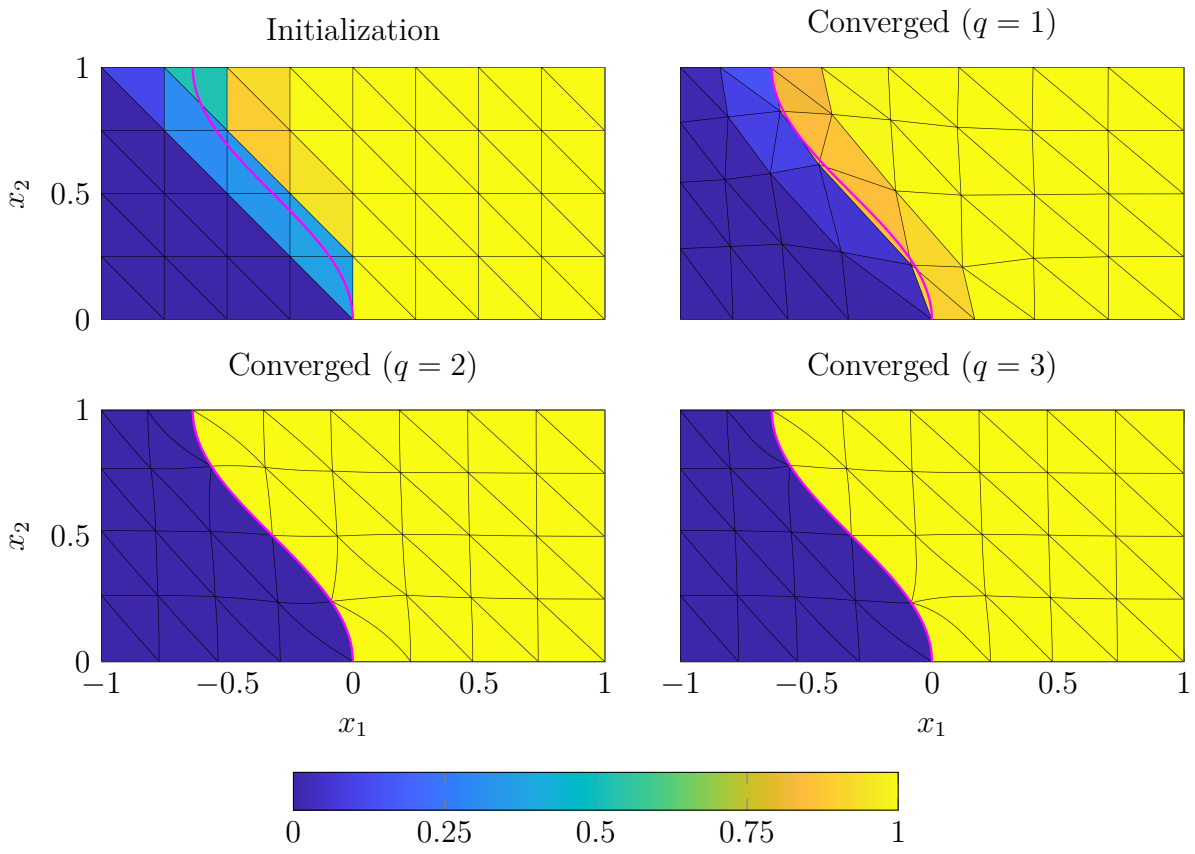


Figure 4.4: Solution of advection equation with the trigonometric shock using the using a $p = 0$ basis for the solution and $q = 1$ (*top right*), $q = 2$ (*bottom left*), and $q = 3$ (*bottom right*) basis for the mesh. The DG solution on a uniform triangular mesh with 64 elements is used to initialize the tracking method (*top left*). The magenta line highlights the discontinuity surface of the exact solution. The method converges to a mesh and solution that approximates the true solution as good as can be expected given the resolution of the finite element space; however, only the high-order elements provide a reasonable approximation on this coarse mesh.

(Figure 4.4). The $q = 1$ solution is under-resolved since the combination of the coarse mesh and straight-sided elements is not sufficient to resolve the discontinuity structure and as a result the solution exhibits over- and under-shoot; the L^1 error associated with this solution is 5.79×10^{-2}. However, the $q = 2$ and $q = 3$ solutions are extremely accurate even on the coarse mesh due to the high-order elements that curve to conform to the discontinuity structure; the L^1 error associated with each solution is 1.15×10^{-3} and 5.50×10^{-4}, respectively.

For all polynomial degrees considered, the solver is able to converge the KKT system to relatively tight tolerances ($\|\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{x})\| < 10^{-10}$ and $\|\boldsymbol{c}(\boldsymbol{u}, \boldsymbol{x})\| < 10^{-7}$). As the polynomial

degree increases, the enriched residual and mesh distortion converge to increasingly small values and the overall convergence of the solver becomes cleaner (Figure 4.5). This comes from the improved Hessian approximation, which comes from the mesh distortion and enriched residual converging to smaller values and justifies dropping of the second term in (3.33). The regularization parameter is adapted to control the size of $\Delta \boldsymbol{x}$ produced from the linear solve and the line search ensures sufficient progress is made with respect to the ℓ_1 merit function (Figure 4.5). For this problem, a non-unity step size is only required once the regularization parameter is small, which is required near convergence to have a decent approximation to the Hessian.

For this problem, the mesh smoothing parameter (κ) and minimum value for the regularization parameter (γ_{min}) were important for the $q = 1$ tracking problem; for $\kappa < 10^{-2}$ or $\gamma_{min} < 10^{-3}$, the tracking algorithm would continually collapse elements near the discontinuity as it pushed nodes to this region to improve the faceted approximation of the trigonometric discontinuity surface. For $q > 1$, the smoothing parameter played no role as the mesh is well-conditioned even with $\kappa = 0$ and γ_{min} can be taken much smaller, e.g., $\gamma_{min} = 10^{-8}$, without adversely affecting the performance of the solver or the final solution. The numerical flux plays a more significant role with regard to the behavior of the SQP solver, even for higher order elements. For the non-smooth (with respect to domain deformation) numerical flux, the solver does not converge because the first-order information is meaningless in regions near the kink in the numerical flux, whereas the convergence is much faster and cleaner for the smoothed upwind flux (Figure 4.6).

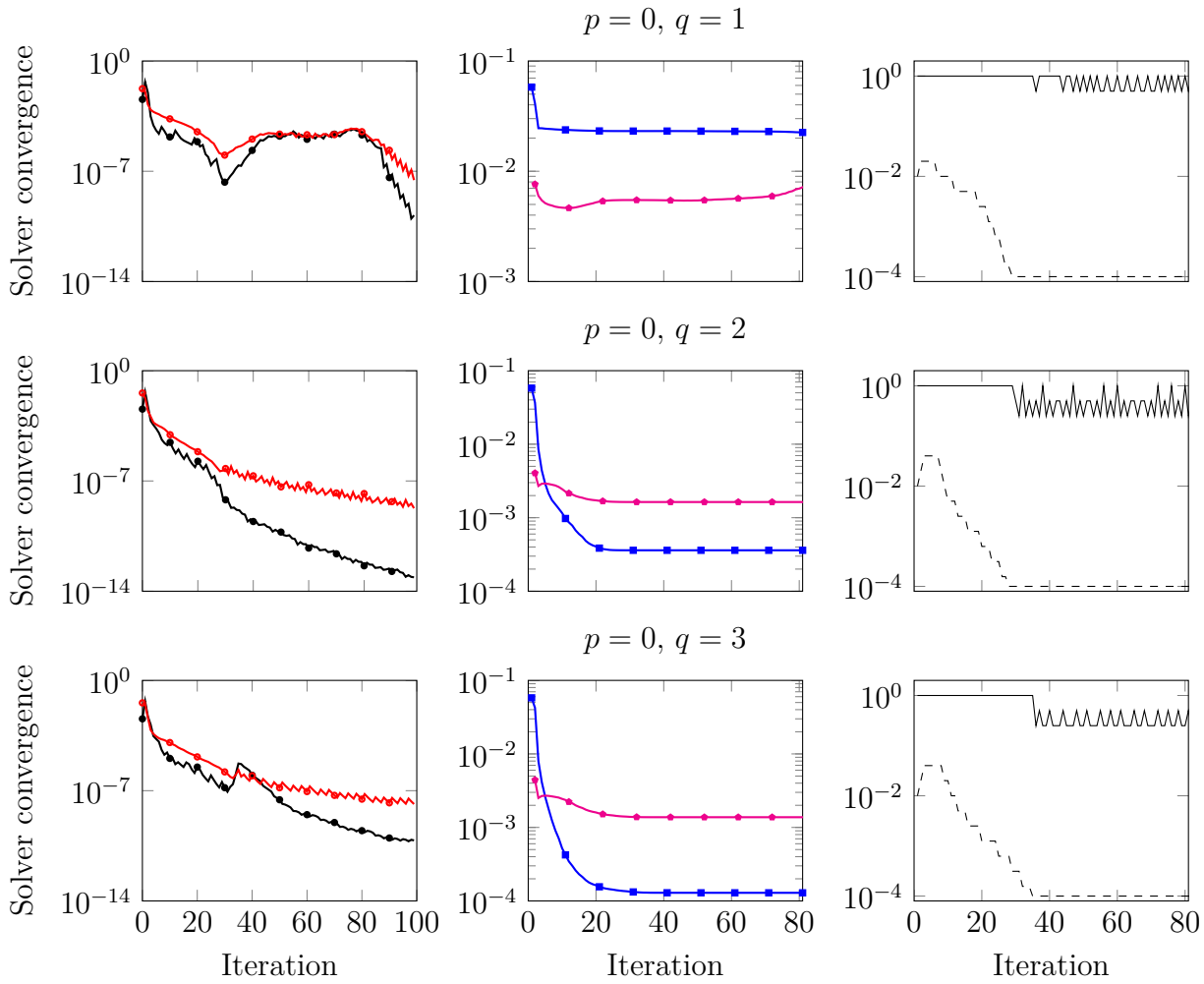


Figure 4.5: Convergence of the DG residual $\|\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{x})\|$ (—•—), enriched DG residual $\|\boldsymbol{R}(\boldsymbol{u}, \boldsymbol{x})\|$ (—■—), optimality condition $\|\boldsymbol{c}(\boldsymbol{u}, \boldsymbol{x})\|$ (—•—), mesh distortion $\|\kappa \boldsymbol{R}_{\text{msh}}(\boldsymbol{x})\|$ (—•—), and control of the regularization parameter (- - -) and step size (——) for the tracking method applied to the advection equation with the trigonometric shock. Even though Newton convergence is not achieved, the proposed solver converges the KKT system to tight tolerances in a reasonable number of iterations.

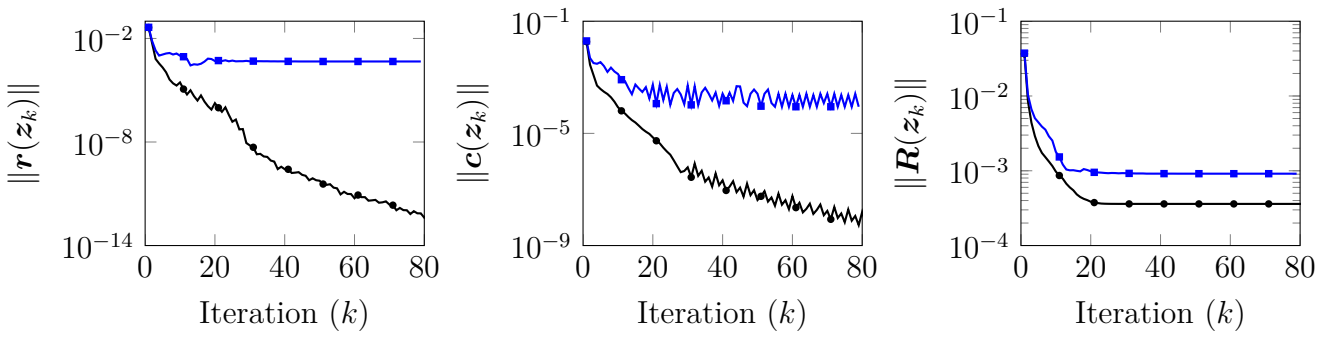


Figure 4.6: Convergence of the constraint (*left*), optimality condition (middle), and objective function (*right*) of the shock tracking optimization problem for linear advection with the trigonometric shock ($p = 0$, $q = 2$) when the non-smooth upwind flux $\mathcal{H}_{\text{up}}$ (—•—) and smoothed ($a = 10$) upwind flux $\mathcal{H}^a_{\text{up}}$ (—▪—) are used as the numerical flux function.

4.3 Time-dependent, inviscid Burgers' equation

Next, we consider the time-dependent, inviscid Burgers' equation that governs nonlinear advection of a scalar quantity $U : \Omega \to \mathbb{R}$ through the space-time domain $\Omega \coloneqq \mathcal{T} \times \bar{\Omega}$,

$$\frac{\partial U}{\partial t} + U\frac{\partial U}{\partial x} = 0 \quad \text{in } \Omega, \qquad U = U_\infty \quad \text{on } \Gamma_i(U) \tag{4.9}$$

where $\mathcal{T} \coloneqq (0, 1)$ is the time domain, $\bar{\Omega} \coloneqq (-1, 1)$ is the one-dimensional spatial domain, $\Gamma_i(U) \coloneqq \{(t, x) \in \partial\Omega \mid \beta(U) \cdot n(x) < 0\}$ is the inflow boundary, $\beta(U)$ is the space-time flow direction defined as

$$\beta(U) = \begin{bmatrix} 1 \\ U \end{bmatrix}, \tag{4.10}$$

$n : \partial\Omega \to \mathbb{R}^2$ is the space-time unit outward normal, and $U_\infty : \partial\Omega \to \mathbb{R}$ is the inflow boundary condition. This fits the form of a general, steady conservation law over the space-time domain Ω with flux function

$$F(U) = \begin{bmatrix} U & \frac{U^2}{2} \end{bmatrix}. \tag{4.11}$$

In the space-time setting, Burgers' equation has the same form as linear advection with the solution-dependent advection field (4.10). Similar to the advection equation, we consider a pure space-time upwind numerical flux since it satisfies conditions (i)-(iii) (Section 2.3)

$$\mathcal{H}(U^+, U^-, n) = \mathcal{H}_{\text{up}}\left(U^+, U^-, n; \beta\left(\frac{U^+ + U^-}{2}\right)\right) \tag{4.12}$$

and yields a stable DG discretization suitable for tracking. We also consider the smoothed version to satisfy condition (iv)

$$\mathcal{H}(U^+, U^-, n) = \mathcal{H}^a_{\text{up}}\left(U^+, U^-, n; \beta\left(\frac{U^+ + U^-}{2}\right)\right). \tag{4.13}$$

We consider the following piecewise quadratic boundary condition

$$U_\infty(t, x) = 2(x+1)^2(1 - H(x)), \tag{4.14}$$

which is enforced via the pure upwind numerical flux evaluated at the interior state U^+ and boundary state $U^\partial = U_\infty$.

For this problem, we consider a solution and mesh basis of equal polynomial degree $p = q$ up to $p = q = 4$. The reference mesh is taken to be a uniform triangular mesh of the domain with 64 elements. The mesh and solution are initialized according to Section 4.1.1, including continuation in the polynomial degree. The SQP solver is used with λ chosen adaptively (Section 3.4.6). The various DG/tracking parameters are set as follows: $\kappa = 10^{-4}$ (mesh smoothing), smoothed upwind numerical flux ($a = 10$), $\gamma_0 = 10^{-1}$ and $\gamma_{min} = 10^{-4}$ (regularization parameter adaptivity), and $N_{max} = 100$, $\epsilon_1 = 10^{-6}$, $\epsilon_2 = 10^{-10}$ (termination criteria). We initialize the $p = q = 1$ simulation from the $p = 1$ DG solution on the reference mesh. Even though the initial mesh is far from tracking the discontinuity (some faces are nearly orthogonal to the discontinuity, rather than parallel to it), our method tracks a faceted approximation to the discontinuity in only 40 iterations, requiring 7 element collapses (Figure 4.7).

The mesh and solution for the high-order elements ($p = q > 1$) are initialized from the $p = q = 1$ tracking mesh and solution. These high-order approximations provide high-quality approximations of the discontinuous space-time solution on the coarse mesh (Figure 4.8). This can further be verified from the temporal slices (Figure 4.9), which show the moving discontinuity is perfectly tracked and the solution is smooth and non-oscillatory away from the discontinuity.

The convergence of the SQP solver in general is similar to convergence behavior observed with the linear advection equation with a trigonometric shock in that the higher the polynomial degree, the cleaner the convergence, and the enriched residual and mesh distortion converge to smaller absolute values, indicating a solution that provides a better approximation to the continuous weak form on a higher quality mesh (Figure 4.10). The line search is active at some intermediate iterations, but plays less of a role near convergence. Furthermore, the same observations regarding the tracking parameters made for the advection equation with the trigonometric shock hold for this problem: the smoothed upwind flux significantly improves convergence of the solver, and if κ or γ_{min} are any smaller, too many elements will collapse onto the discontinuity surface for $p = q = 1$.

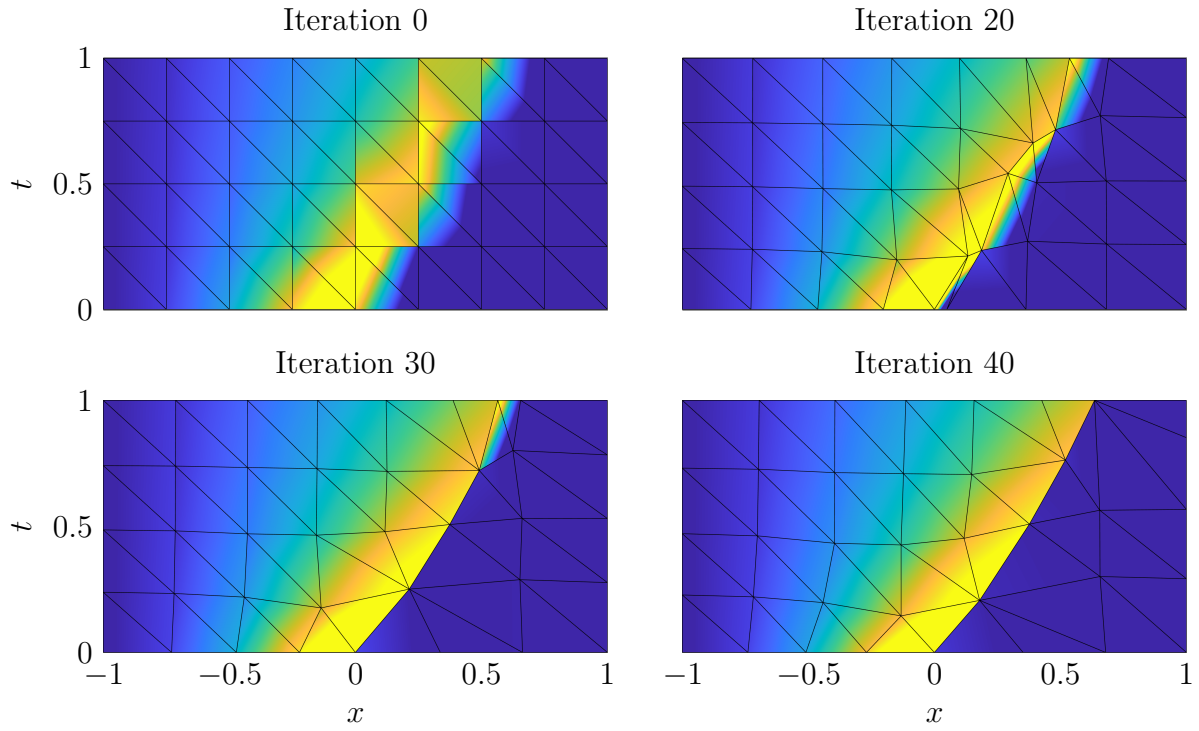


Figure 4.7: Space-time solution of one-dimensional, inviscid Burgers' equation using the tracking method at various iterations throughout the solution procedure using a $p = q = 1$ basis for the solution and mesh. The method collapses 7 elements throughout the solution procedure and tracks a faceted approximation of the shock trajectory using the $q = 1$ mesh in only 40 iterations.

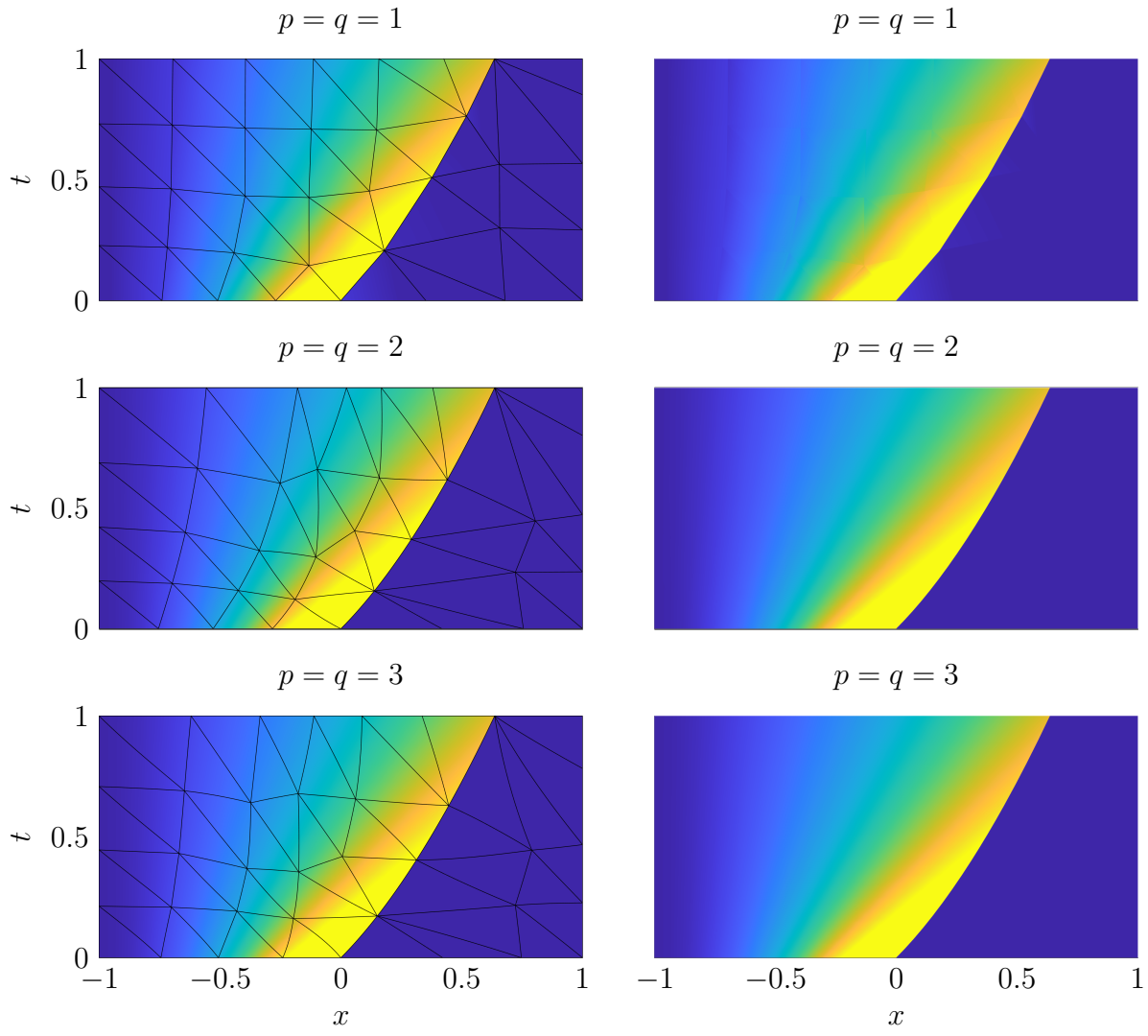


Figure 4.8: Space-time solution of one-dimensional, inviscid Burgers' equation using the proposed tracking method with a $p = q = 1$ (*top*), $p = q = 2$ (*middle*), and $p = q = 3$ (*bottom*) basis for the solution and mesh with (*left*) and without (*right*) element boundaries. In all cases, the finite element solution provides a high-quality approximation to the true solution by tracking the discontinuity with a well-conditioned mesh. This is particularly true for the high-order elements that curve to the space-time trajectory of the discontinuity.

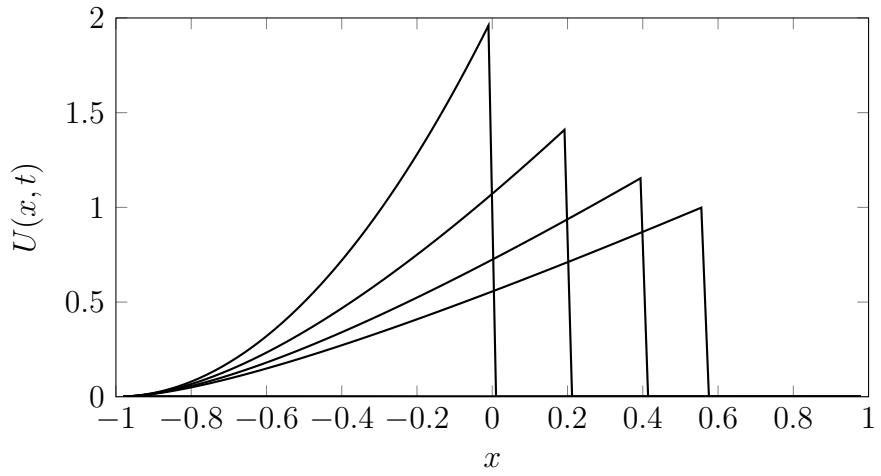


Figure 4.9: Temporal slices of the $p = q = 4$ tracking solution to the inviscid Burgers' equation at times $t = 0.05, 0.35, 0.65, 0.95$.

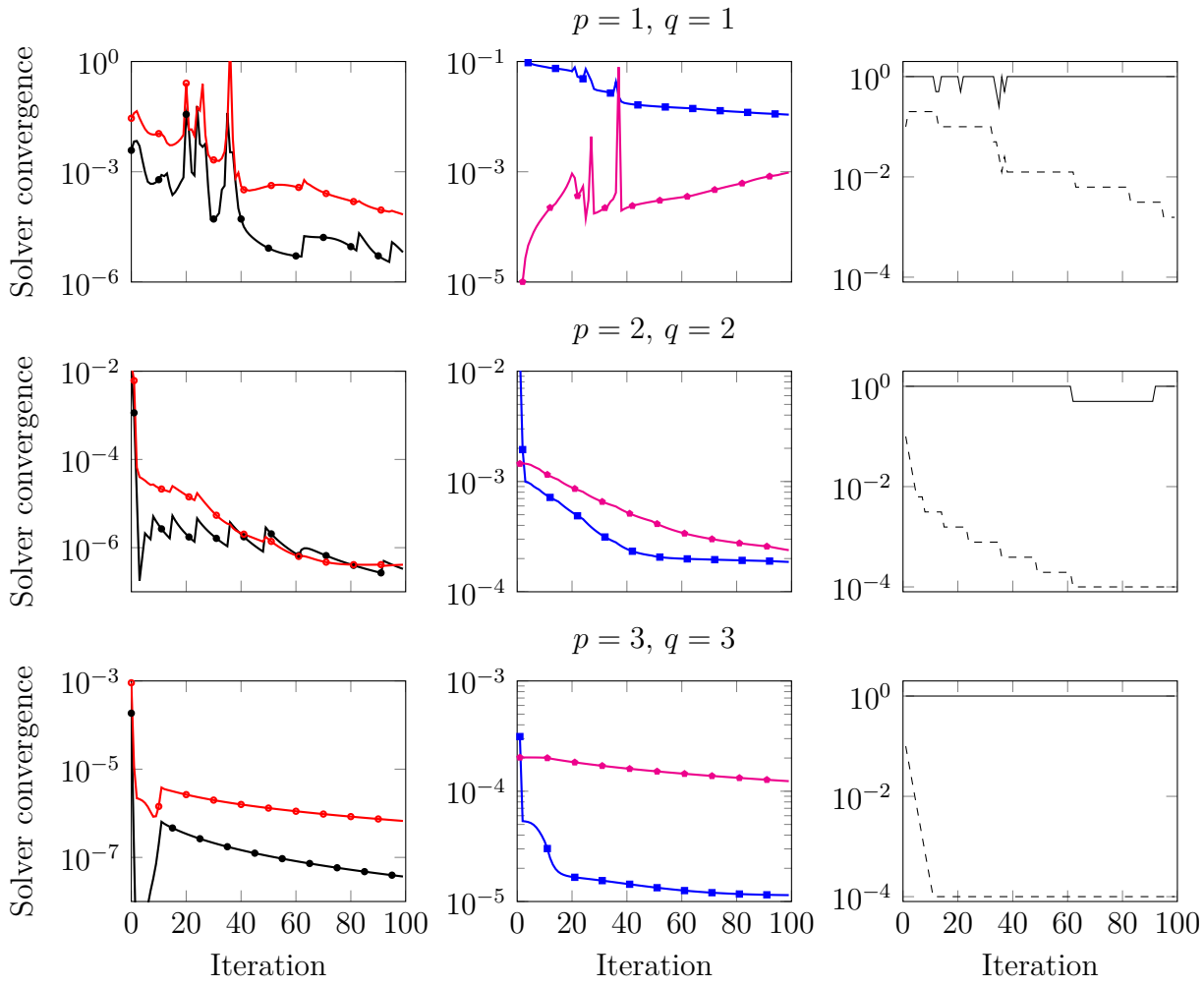


Figure 4.10: Convergence of the DG residual $\|\boldsymbol{r}(\boldsymbol{u},\boldsymbol{x})\|$ (—•—), enriched DG residual $\|\boldsymbol{R}(\boldsymbol{u},\boldsymbol{x})\|$ (—■—), optimality condition $\|\boldsymbol{c}(\boldsymbol{u},\boldsymbol{x})\|$ (—•—), mesh distortion $\|\kappa\boldsymbol{R}_{\text{msh}}(\boldsymbol{x})\|$ (—•—) and control of the regularization parameter (- - -) and step size (——) for the tracking method applied to the inviscid Burgers' equation.

4.4 Steady, compressible Euler equations

The Euler equations govern the steady flow of an inviscid, compressible fluid through a domain $\Omega \subset \mathbb{R}^d$

$$\begin{aligned} \frac{\partial}{\partial x_j}\left(\rho(x,t)v_j(x,t)\right) &= 0 \\ \frac{\partial}{\partial x_j}\left(\rho(x,t)v_i(x,t)v_j(x,t) + P(x,t)\delta_{ij}\right) &= 0 \\ \frac{\partial}{\partial x_j}\left(\left[\rho(x,t)E(x,t) + P(x,t)\right]v_j(x,t)\right) &= 0 \end{aligned} \tag{4.15}$$

where $\rho : \Omega \times (0,T) \to \mathbb{R}_+$ is the density of the fluid, $v_i : \Omega \times (0,T) \to \mathbb{R}$ for $i = 1, \ldots, d$ is the velocity of the fluid in the ith coordinate direction, and $E : \Omega \times (0,T) \to \mathbb{R}_+$ is the total energy of the fluid. The square velocity $q^2 : \Omega \times (0,T) \to \mathbb{R}_+$ and kinetic energy $k_e : \Omega \times (0,T) \to \mathbb{R}_+$ of the fluid are

$$q^2 = v_i v_i, \quad k_e = \frac{1}{2}\rho q^2. \tag{4.16}$$

The enthalpy of the fluid $H : \Omega \times (0,T) \to \mathbb{R}_+$ is defined as

$$\rho H = \rho E + P, \tag{4.17}$$

where $P : \Omega \times (0,T) \to \mathbb{R}_+$ is the pressure. For a calorically ideal fluid, the pressure and energy are related via the ideal gas law

$$P = (\gamma - 1)(\rho E - k_e) \tag{4.18}$$

and the speed of sound is

$$c = \sqrt{\gamma P/\rho}. \tag{4.19}$$

The density, velocity, and energy are combined into a vector of conservative variables U and the Euler equations take the form of an inviscid conservation law (2.1) with flux function $F(U)$

$$U = \begin{bmatrix} \rho \\ \rho v \\ \rho E \end{bmatrix}, \qquad F(U) = \begin{bmatrix} \rho v^T \\ \rho v v^T + P I_2 \\ \rho H v^T \end{bmatrix}, \tag{4.20}$$

where I_2 is the 2×2 identity matrix.

We use Roe's flux [61] as the numerical flux function to ensure a stable discretization suitable for tracking since it satisfies (i)-(iii) (Section 2.3); however, it is not smooth with respect to variations in the domain deformation. To eliminate entropy violating rarefaction shocks and improve the smoothness with respect to the domain deformations, we modify the interior numerical fluxes with the Harten-Hyman entropy fix [29]. This leads to a numerical

flux function that violates (iii), as observed in [8], but recovers smoothness with respect to mesh deformation, which proves to be a reasonable trade-off.

We consider three types of boundaries: slip wall (Γ_w), supersonic inflow (Γ_D), and supersonic outflow (Γ_N). The supersonic inflow is also known as a farfield or Dirichlet condition and the supersonic outflow is a Neumann condition. For a slip wall ($v \cdot n = 0$), the boundary state is defined as

$$U_w^\partial(U, n) \coloneqq \begin{bmatrix} \rho \\ \rho v^- \\ \rho E, \end{bmatrix} \tag{4.21}$$

where $v^- = (v - 2v \cdot n)n$ is the velocity reflected about the normal. For a supersonic inflow, all characteristics are coming into the domain and the boundary state depends solely on the prescribed density ρ_∞, velocity v_∞, and pressure p_∞

$$U_D^\partial(U) \coloneqq \begin{bmatrix} \rho_\infty \\ \rho_\infty v_\infty \\ \frac{p_\infty}{\gamma - 1} + \frac{\rho_\infty}{2} v_\infty \cdot v_\infty \end{bmatrix}. \tag{4.22}$$

Finally, at a supersonic outflow, all characteristics are leaving the domain and the boundary state is taken from the interior

$$U_N^\partial(U) \coloneqq U. \tag{4.23}$$

All boundary conditions are enforced via the pure Roe flux (without entropy fix) evaluated at the interior state U^+ and appropriate boundary state

$$U^\partial(U, n) = \begin{cases} U_w^\partial(U, n) & \text{on } \Gamma_w \\ U_D^\partial(U) & \text{on } \Gamma_D \\ U_N^\partial(U) & \text{on } \Gamma_N. \end{cases} \tag{4.24}$$

A useful property of the inviscid flows is the enthalpy is constant throughout the domain. Therefore, to quantify the error in the numerical solution obtained using the proposed method, we will use the deviation of the flow enthalpy from the inflow enthalpy, $H_\infty \coloneqq \frac{\gamma}{\gamma - 1} \frac{p_\infty}{\rho_\infty} + \frac{1}{2} v_\infty \cdot v_\infty$

$$e_H(U) \coloneqq \sqrt{\frac{\int_\Omega (H(U) - H_\infty)^2}{\int_\Omega dV}}. \tag{4.25}$$

4.4.1 Supersonic flow over wedge

First we consider supersonic flow ($M_\infty = 2$) over a $\theta = 10°$ inclined plane (Figure 4.11). Since all wall boundaries are straight-sided and the incoming flow is uniform, the flow is piecewise constant and the shocks are straight.

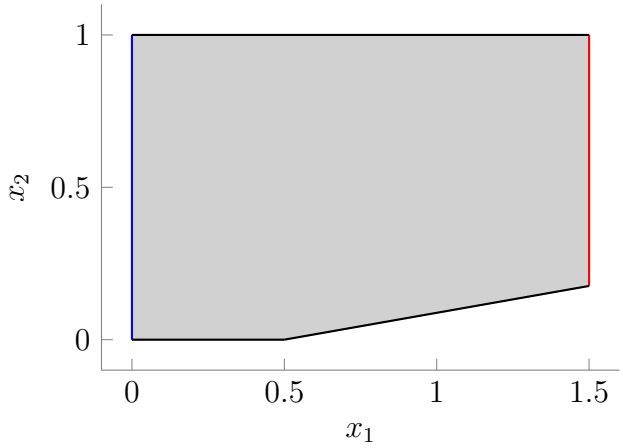


Figure 4.11: Geometry and boundary conditions of the wedge problem. Boundary conditions: slip wall (—), supersonic inflow with $\rho_\infty = 1.4$, $v_\infty = (2, 0)$, $p_\infty = 1$ ($M_\infty = 2$) (—), and supersonic outflow (—).

Therefore, we apply the tracking method with a $p = 0$ polynomial basis for the solution and $q = 1$ basis for the mesh. The reference mesh is taken to be a uniform triangular mesh of the domain with 48 elements. The solver is initialized with the $p = 0$ DG solution on the reference mesh (Section 4.1.1). The SQP solver is used with λ chosen adaptively (Section 3.4.6). The various DG/tracking parameters are set as follows: $\kappa = 0$ (no mesh smoothing), Roe flux (without entropy fix), $\gamma_0 = 1$ and $\gamma_{min} = 10^{-8}$ (regularization parameter adaptivity), and $\epsilon_1 = 10^{-8}$, and $\epsilon_2 = 10^{-12}$ (termination criteria). After only 20 iterations, the mesh perfectly tracks the shock (Figure 4.12) and the DG solution closely matches the exact solution; the enthalpy error of the solution is $e_H = 7.94 \times 10^{-10}$. The solver performs similarly to the advection equation with a straight discontinuity and therefore a full discussion is omitted for brevity.

4.4.2 Supersonic flow over airfoil

Next, we apply the proposed tracking method to solve for supersonic flow over a NACA0012 airfoil (Figure 4.13). This is a difficult problem because there are two distinct shocks that must be resolved: a bow shock ahead of the leading edge and an oblique shock off the tail. The reference mesh is taken as an unstructured triangular mesh of the domain with 160 elements generated by DistMesh [57]. The mesh and solution are initialized according to Section 4.1.1, including continuation in the polynomial degree. The SQP solver is used with λ chosen adaptively (Section 3.4.6). The various DG/tracking parameters are set as follows: $\kappa = 10^{-4}$ (mesh smoothing), Roe flux with entropy fix, $\gamma_0 = 1$ and $\gamma_{min} = 1$ (regularization parameter adaptivity), and $N_{max} = 100$, $\epsilon_1 = 10^{-6}$, $\epsilon_2 = 10^{-10}$ (termination criteria).

For the $p = q = 1$ simulation, the proposed method tracks a faceted approximation to the discontinuity in only 100 iterations, requiring 11 element collapses and maintain high-

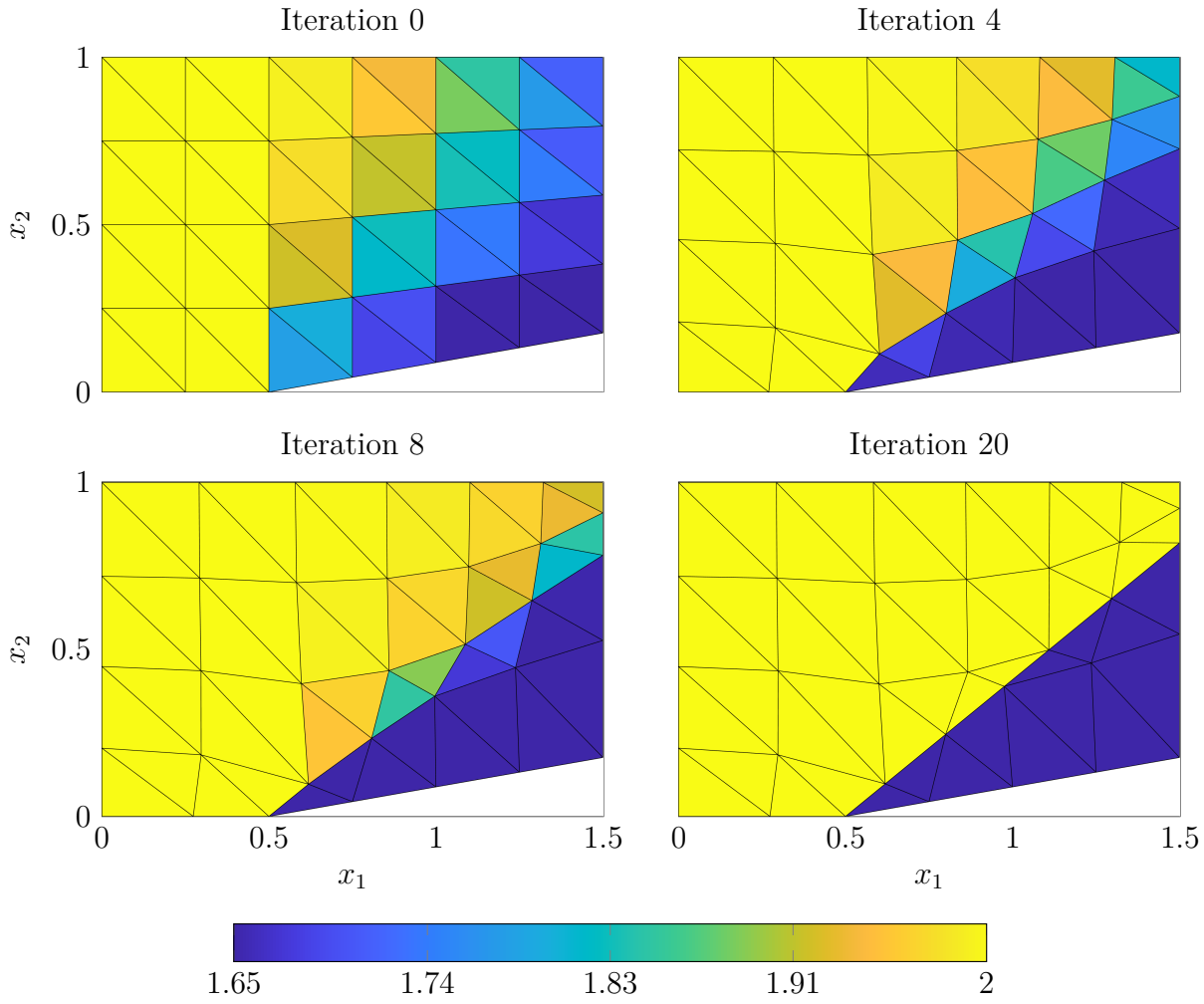


Figure 4.12: Solution (Mach) of Euler equations over a wedge (supersonic regime) using the tracking method at various iterations throughout the solution procedure using $p = 0$ basis for the solution and $q = 1$ basis for the mesh. The method converges to nearly the exact solution (enthalpy error $e_H = 7.94 \times 10^{-10}$), in only 20 iterations.

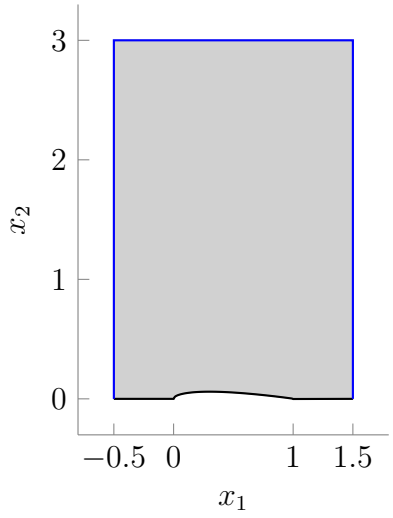


Figure 4.13: Geometry and boundary conditions of the airfoil problem. Boundary conditions: slip wall (——) and farfield (characteristic) conditions with $\rho_\infty = 1.4$, $v_\infty = (M_\infty, 0)$, $p_\infty = 1$ ($M_\infty = 0.85$ for the transonic case and $M_\infty = 1.5$ for the supersonic case). (——).

quality elements, despite the coarse elements in the initial mesh that do not conform to either shock (Figure 4.14). Even though the discontinuities are successfully tracked, the solution is under-resolved, particularly near the body, since the large $p = 1$ elements are not sufficient to resolve the solution in this region. As a result, the enthalpy error is large $e_H = 1.30 \times 10^{-3}$. The $p = q = 2$ and $p = q = 3$ tracking solutions provide highly accurate approximations to the true flow even on this coarse mesh due to the high-order resolution of the discontinuity surface with curved elements and high-order approximation of the flow (Figure 4.14); the enthalpy error for the $p = q = 2$ tracking solution is $e_H = 6.73 \times 10^{-5}$ and for $p = q = 3$ is $e_H = 1.02 \times 10^{-5}$. The perfect (zero-thickness) capturing of the shocks and high-order approximation of the solution can further be seen from the solution slices in Figure 4.15.

For this problem, the convergence of the solver is not as clean as the other problems (Figure 4.16) and heavily relies on the line search throughout the solution procedure to ensure sufficient decrease in the ℓ_1 merit function. However, the solver still drives the KKT conditions to reasonable tolerances within 100 iterations for the $p = q = 3$ simulation. The spikes in the KKT residuals around iterations 20 and 60 in the $p = q = 1$ simulation are due to element collapses. Our choice of algorithmic parameters were critical for this case. If κ is significantly larger, the element collapses would not occur and the mesh would not be able to track the shocks, particularly the weaker shock at the trailing edge. Furthermore, it was important to use the relatively large value of $\gamma_{min} = 1$, otherwise the line search would be taxed more heavily in the later iterations and completely stalled convergence.

Finally, we mention that this is the first problem where our choice of the elliptic regular-

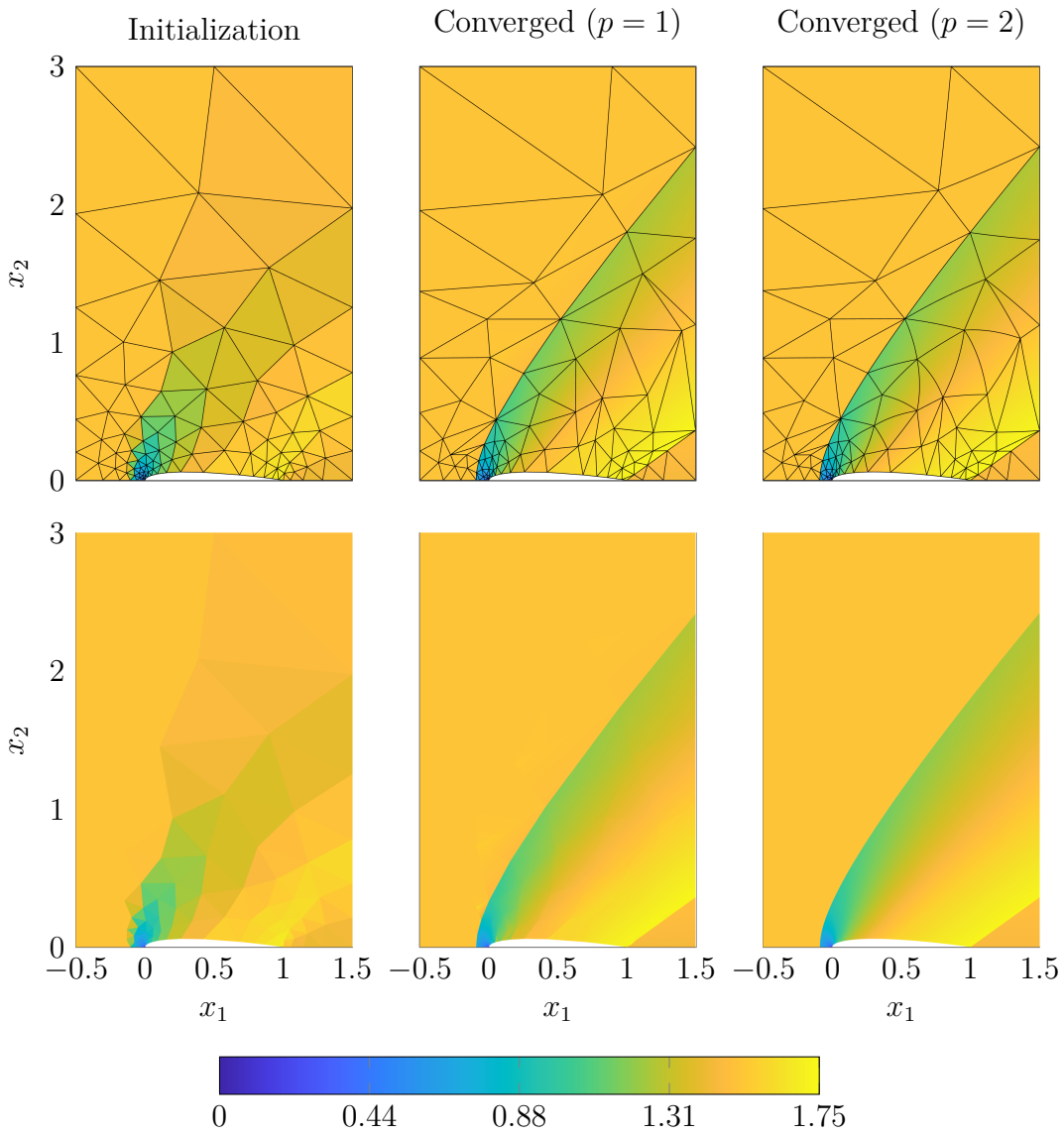


Figure 4.14: Solution (Mach) of Euler equations over the NACA0012 airfoil ($M_\infty = 1.5$) using the proposed tracking method with a $p = q = 1$ (*center*) and $p = q = 2$ (*right*) basis for the solution and mesh with (*top*) and without (*bottom*) element boundaries. In both cases, the tracking procedure successfully tracks the shocks given the resolution in the finite element space, despite the initial mesh and solution (*left*) being far from aligned with the shock. The high-order ($p = q = 2$) basis yields an accurate approximation to the flow on the coarse mesh while the low-order ($p = q = 1$) solution is under-resolved.

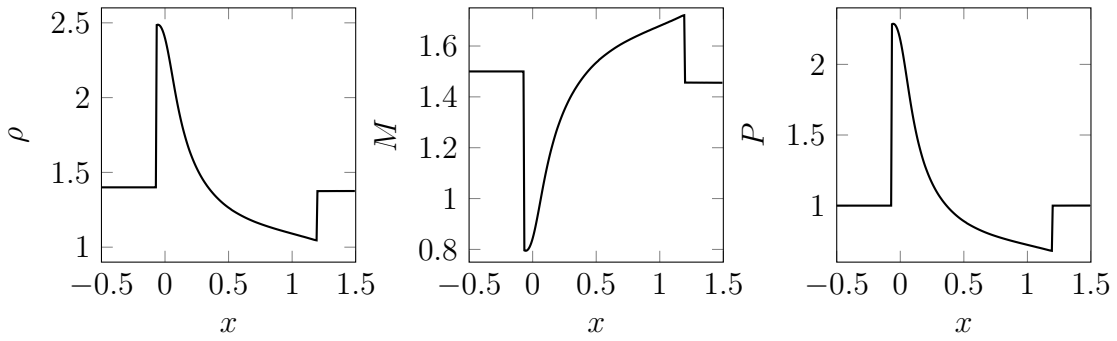


Figure 4.15: Slices of density (*left*), Mach number (*center*), and pressure (*right*) of the $p = q = 3$ tracking solution along the curve $\Gamma \coloneqq \{(s, 0.14) \mid s \in (-0.5, 1.5)\}$ for the NACA problem. The discontinuities are captured perfectly between DG elements and the solution is smooth and non-oscillatory away from the discontinuities, indicating that the solution is well-resolved and the discontinuities are successfully tracked.

ization matrix with the coefficients chosen inversely proportional to the size of the element in the reference domain (Section 3.4.5) is significant. The other choices we explored included the identity matrix [8], elliptic PDE stiffness matrix without volume-based weighting [10], and other choices involving the finite element mass matrix. All of these options performed similarly in creating unacceptably large search direction in regions near the leading and trailing edge (small elements) relative to other regions in the mesh (large elements). Without a line search, these steps would cause the mesh to entangle and the simulation to crash. With the line search, the step size in regions with larger elements would be driven nearly to zero causing the solver to fail to track the discontinuity in these regions. From this, we conclude the combination of the elliptic PDE stiffness matrix and the weighting of the coefficients inversely with respect to the element size in the reference domain is important for meshes with elements of significantly varying size.

4.4.3 Transonic flow over airfoil

Next, we consider transonic flow ($M_\infty = 0.85$) over the same NACA0012 airfoil and domain from the previous section (Figure 4.13). This problem has a shock attached to the curved airfoil profile, which requires a nontrivial boundary mapping (Section 3.3) to ensure nodes slide along the airfoil surface. We construct the mapping χ using the procedure in [78], i.e., the x-coordinates of the nodes on the surface are taken as optimization parameters and the y-coordinates are determined from the expression for the airfoil profile

$$y(x) = 0.6\left(0.2960\sqrt{x} - 0.126x - 0.3516x^2 + 0.2843x^3 - 0.1036x^4\right). \tag{4.26}$$

The reference mesh is taken as an unstructured triangular mesh of the domain with 127 elements generated by DistMesh [57]. The mesh and solution are initialized according to

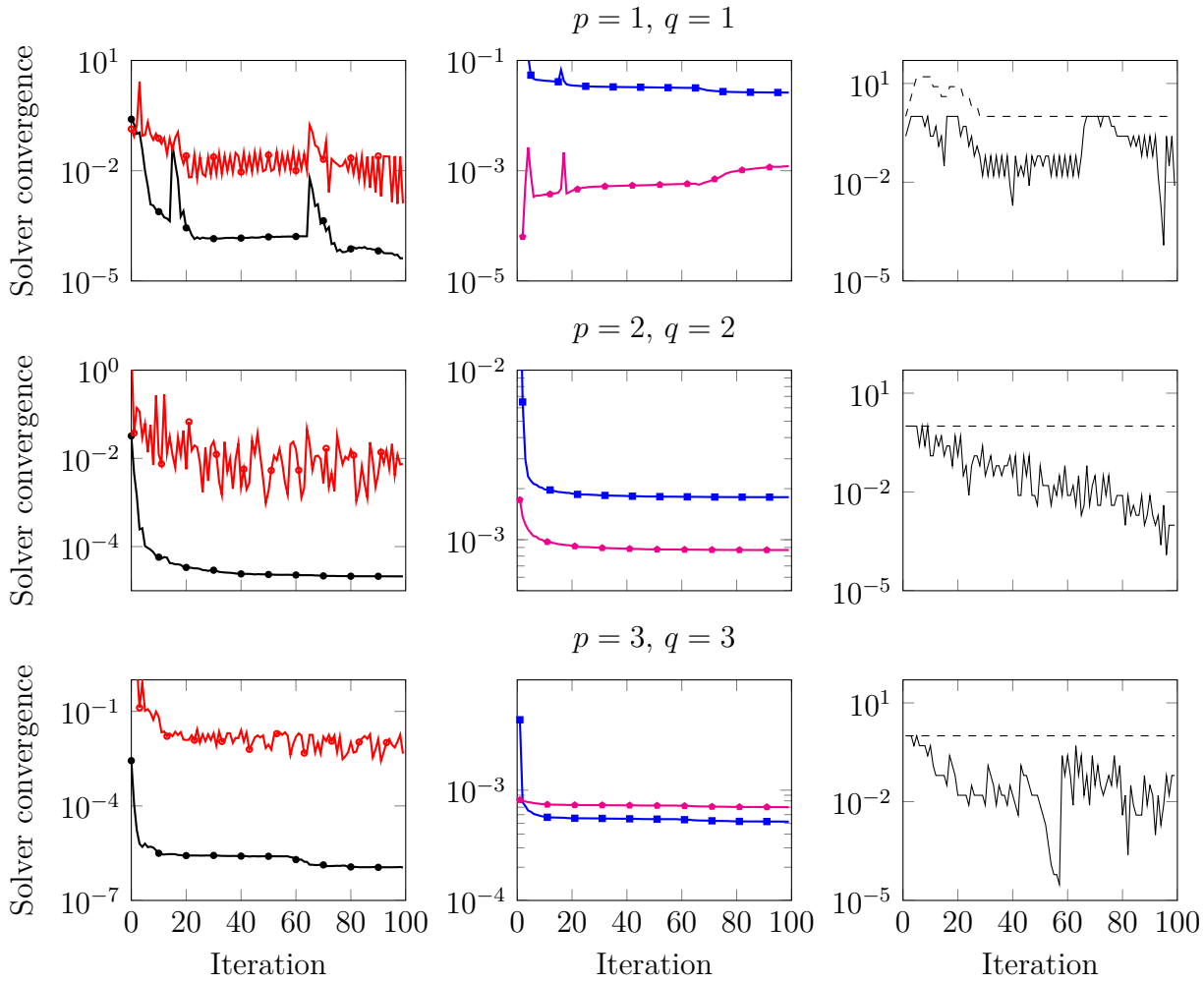


Figure 4.16: Convergence of the DG residual $\|\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{x})\|$ (—•—), enriched DG residual $\|\boldsymbol{R}(\boldsymbol{u}, \boldsymbol{x})\|$ (—▪—), optimality condition $\|\boldsymbol{c}(\boldsymbol{u}, \boldsymbol{x})\|$ (—•—), mesh distortion $\|\kappa \boldsymbol{R}_{\text{msh}}(\boldsymbol{x})\|$ (—•—) and control of the regularization parameter (- - -) and step size (——) for the tracking method applied to solve the supersonic flow around the NACA airfoil. For this difficult problem, the convergence of the KKT system is not as clean as the other problems and relies heavily on the line search; however, the solver still tracks the shocks and returns an accurate flow as seen from the solution plots (Figures 4.14-4.15). The spikes in the KKT conditions for $p = q = 1$ are due to element collapses.

Section 4.1.1, including continuation in the polynomial degree. The SQP solver is used with λ chosen adaptively (Section 3.4.6). The various DG/tracking parameters are set as follows: $\kappa = 10^{-3}$ (mesh smoothing), Roe flux with entropy fix, $\gamma_0 = 10^{-2}$ and $\gamma_{min} = 1$ (regularization parameter adaptivity), and $N_{max} = 30$, $\epsilon_1 = 10^{-6}$, $\epsilon_2 = 10^{-10}$ (termination criteria).

For polynomial degrees $p = q = 1, 2, 3$, the tracking method accurately tracks the attached discontinuity without requiring element collapses and produces an accurate flow solution given the resolution of the approximation space (Figure 4.17). The high resolution of the $p = q = 3$ simulation can be seen through the slices of the solution (Figure 4.18): the discontinuity is captured perfectly and the solution away from the discontinuity is smooth and non-oscillatory. The convergence of the method has been thoroughly studied in the previous sections and omitted for brevity.

4.4.4 Sod shock tube

Sod's shock tube is a Riemann problem for the unsteady Euler equations that models an idealized shock tube where the membrane separating a high pressure region from a low pressure one is instantaneously removed. This is a commonly used validation problem since it has an analytical solution and features a shock wave, a rarefaction wave, and contact discontinuity. The flow domain is $\Omega = [0, 1]$, the final time is $T = 0.2$, the initial condition is given in terms of the density, velocity, and pressure as

$$\rho(x,0) = \begin{cases} 1 & x < 0.5 \\ 0.125 & x \geq 0.5 \end{cases}, \quad v(x,0) = 0, \quad P(x,0) = \begin{cases} 1 & x < 0.5 \\ 0.1 & x \geq 0.5, \end{cases} \tag{4.27}$$

and the density, velocity, and pressure are prescribed at $x = 0$ and the velocity is prescribed at $x = 1$ (values can be read from the initial condition). The solution of this problem contains three waves (shock, contact, rarefaction) that emanate from $x = 0.5$ and move at different speeds, which is a generalized triple point in space-time. The method of lines approach cannot handle this case because a single node lies at $(x, t) = (0.5, 0)$, which cannot track all three waves at time $t > 0$. Thus, we use the space-time implicit shock tracking approach to solve this problem. The method is initialized with an unstructured mesh of the space-time domain $\bar{\Omega} = [0, 1] \times [0, 0.2]$ consisting of 173 simplex elements ($p = 3$, $q = 1$) with additional refinement near $(0.5, 0)$ to resolve the geometric complexity of the triple point for a total of 5190 spatiotemporal degrees of freedom. The DG solution is initialized with the $p = 0$ solution on the reference mesh. The final space-time tracking solution is shown in Figure 4.19 where all features (head and tail of rarefaction, shock, and contact) are tracked. A total of 47 element collapses are required, mostly near the triple point to obtain elements that do not cross between the five distinct regions (left state, rarefaction, between rarefaction and contact, between contact and shock, and right state); the final mesh contains 126 elements (3780 degrees of freedom). Despite the reduction in the number of degrees of freedom, the solution at the final time T agrees well with the exact solution (Figure 4.20)

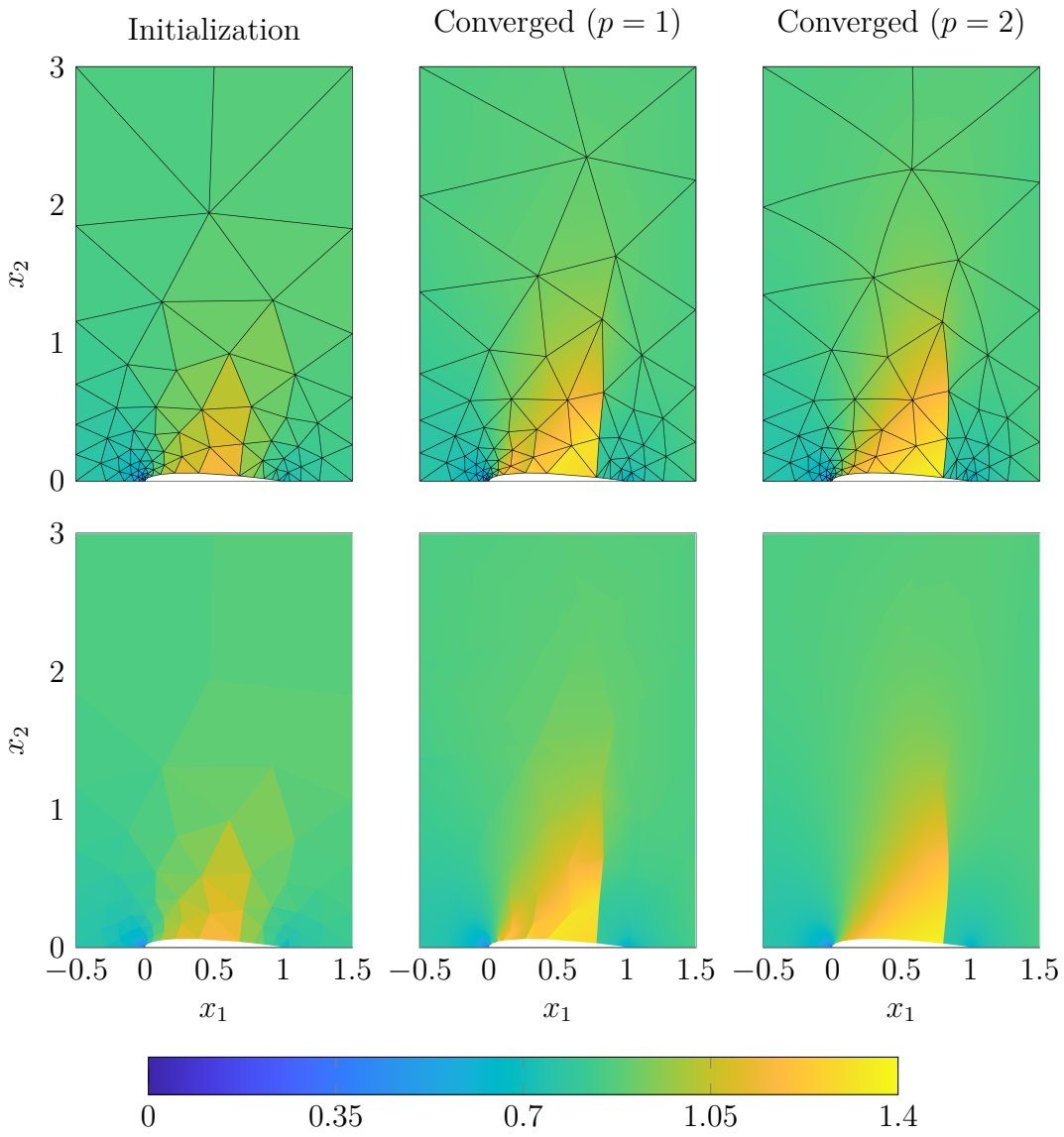


Figure 4.17: Solution (Mach) of Euler equations over the NACA0012 airfoil ($M_\infty = 0.85$) using the proposed tracking method with a $p = q = 1$ (*center*) and $p = q = 2$ (*right*) basis for the solution and mesh with (*top*) and without (*bottom*) element boundaries. In both cases, the tracking procedure successfully tracks the shocks given the resolution in the finite element space, despite the initial mesh and solution (*left*) being far from aligned with the shock.

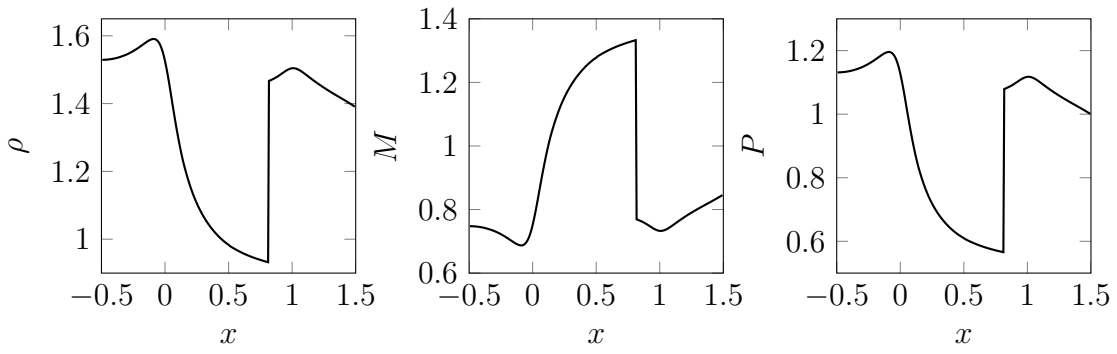


Figure 4.18: Slices of density (*left*), Mach number (*center*), and pressure (*right*) of the $p = q = 3$ tracking solution along the curve $\Gamma \coloneqq \{(s, 0.14) \mid s \in (-0.5, 1.5)\}$ for the NACA problem ($M_\infty = 0.85$). The discontinuity is captured perfectly between DG elements and the solution is smooth and non-oscillatory away from the discontinuity, indicating that the solution is well-resolved.

because the cubic basis functions do not cross discontinuities or kinks. This is an example of an unsteady conservation law which is best suited to be solved as a "steady" problem through a space-time approach vs. the method of lines approach described in Section 2.4 due to the multiple XXX. This distinction will be further highlighted in Section 5.

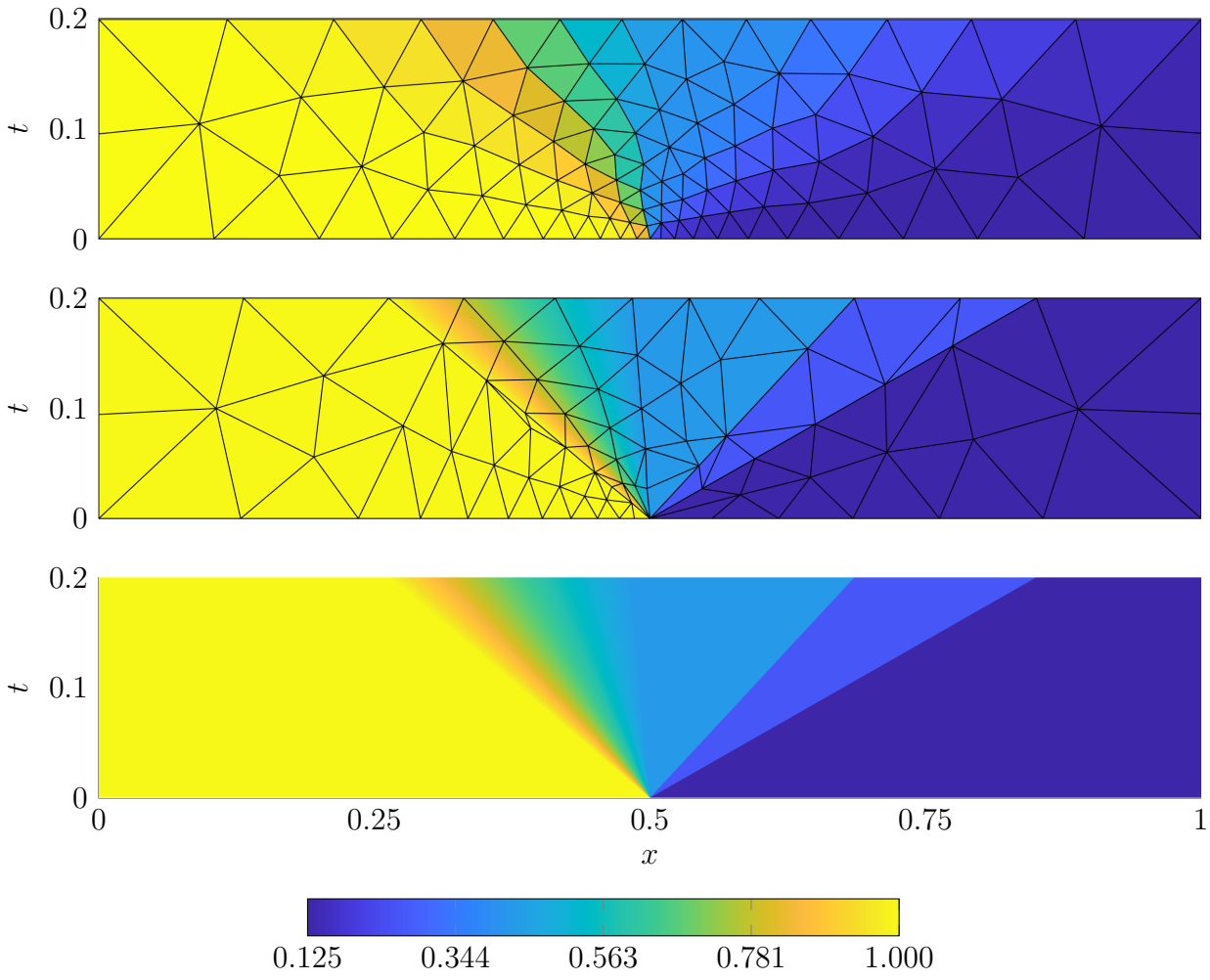


Figure 4.19: Space-time solution of Sod's shock tube (density) using implicit shock tracking using a $p = 3$ DG discretization (*center* with element boundaries, *bottom* without element boundaries), initialized from an unstructured mesh without knowledge of the discontinuity surfaces and a $p = 0$ DG solution (*top*).

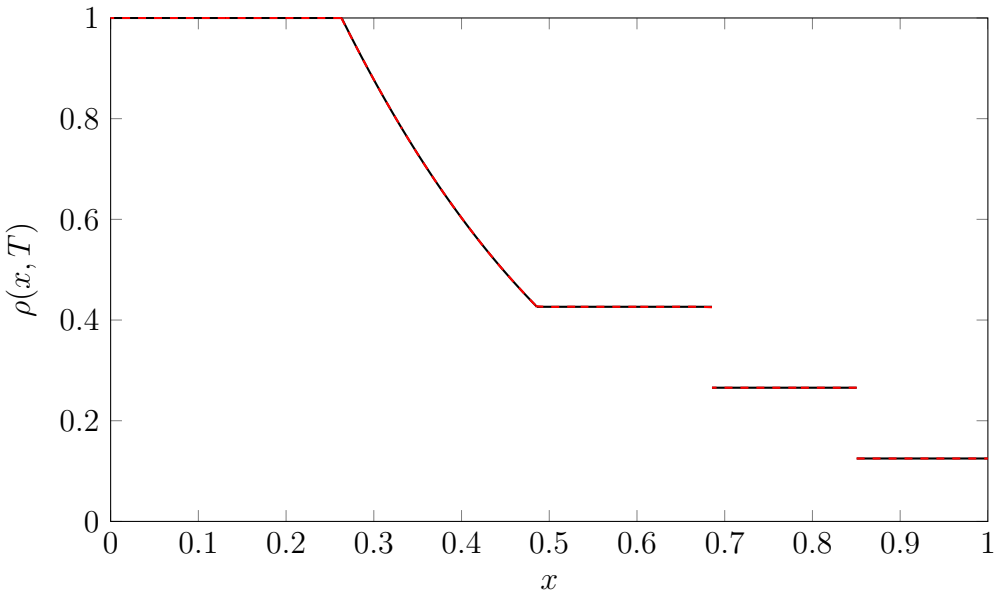


Figure 4.20: Slice of space-time implicit shock tracking solution (——) at time $T = 0.2$ relative to the exact solution (- - -).

Chapter 5

Numerical experiments: unsteady case

In this section, we extend the work of the previous section by considering the same three inviscid conservation laws but now in *unsteady* form (2.1). We demonstrate the tracking framework on five problems with discontinuous solutions of varying difficulty. These unsteady conservation laws are solved by the method of lines as described in (Section 3.5), where we "solve a steady problem at each timestep". We also demonstrate high-order convergence in time and present the convergence behavior of the SQP solver.

5.1 Practical considerations

5.1.1 Initial guess for optimization

The implicit shock tracking optimization problem (3.41) is non-convex and therefore the initial guess for the SQP solver is critical to obtain a good solution. In the method of lines setting where implicit shock tracking is performed at each time step, an obvious idea for the initial guess is the converged mesh and solution from the previous time t_n. In practice, this is too far off to attain good convergence properties except for prohibitively small choices of timestep Δt. Instead, we employ an initial guess where we advect each node on the discontinuity surface from the previous time by the instantaneous shock speed determined by the Rankine-Hugoniot conditions (Fig. 5.1). The remaining nodes are updated using a standard optimization-based mesh smoothing using the high-order mesh distortion metric $r_K^{\text{msh}} : \mathcal{W}_h \to \mathbb{R}$ developed in [24], defined as

$$r_{\text{msh}}^K : \mathcal{Q} \mapsto \int_K \left(\frac{\|\nabla_X \mathcal{Q}\|_F^2}{d(\det \nabla_X \mathcal{Q})_+^{2/d}} \right)^2 dv. \qquad \text{(3.11 revisited)}$$

This provides a good initial guess for the shock-aligned mesh for each stage of the optimization problem. However, the advection of these shock nodes can result in a tangled mesh

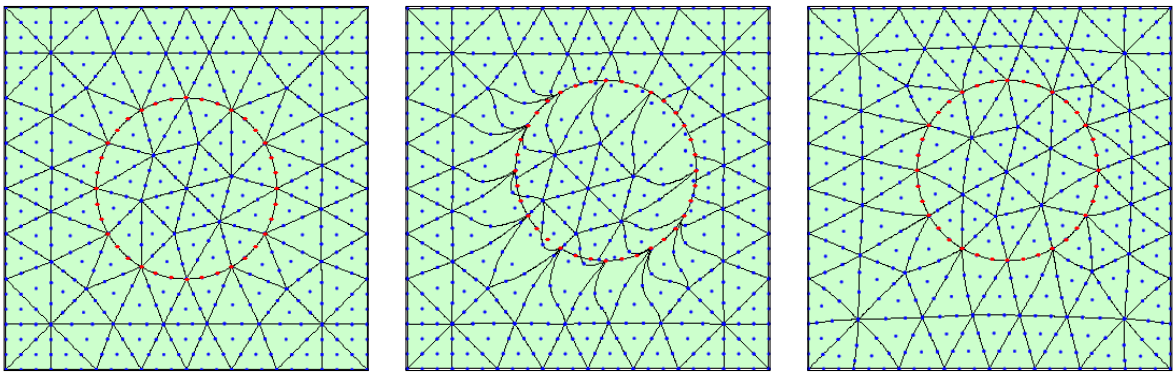


Figure 5.1: Shock nodes (·····) (*left*), advected by instantaneous shock speed (*middle*) and smoothed mesh (*right*).

(Fig. 5.1, middle). We modify the mesh distortion metric (3.11) following the approach in [26] where the Jacobian ($\det \nabla_X \mathcal{Q}$) is regularized to allow for the optimization procedure to recover from initially invalid configurations.

For the initial guess for the solution, we use the converged physical solution (U) from the previous timestep. However, since we applied our DG discretization to the transformed reference conservation law (2.8), we are solving for the reference solution (U_X). Therefore, to use the physical solution at time t_n as the initial guess for time t_{n+1}, we multiply it by the ratio of the Jacobian of the initial guess for the mapping at time t_{n+1} (advection based on Rankine-Hugoniot conditions and smoothing) to the Jacobian of the converged mapping at time t_n. More precisely, the initial guess for the ith Runge-Kutta stage in *physical* space, denoted $\check{U}_h$, is obtained by constant extrapolation of the physical state at time t

$$\check{U}_h(x, t + c_i\Delta t) \coloneqq U_h(x,t) = g_h(X,t)^{-1} U_{X_h}(X,t). \tag{5.1}$$

Let $\tilde{\mathcal{G}} \in \mathcal{W}_h$ be the mapping to the configuration obtained by the Rankine-Hugoniot-based procedure and associated smoothing, then the initial guess for the ith Runge-Kutta stage in the *reference* domain, denoted $\check{U}_{X_h}$, is

$$\check{U}_{X_h}(X, t + c_i\Delta t) \coloneqq \tilde{g}(X, t + c_i\Delta t)\check{U}_h(x, t + c_i\Delta t) = \frac{\tilde{g}(X, t + c_i\Delta t)}{g_h(X,t)} U_{X_h}(X,t). \tag{5.2}$$

Future work to enhance the robustness of these initial guesses might consider more advanced methods to approximate the shock velocity [58] and higher order extrapolation-based estimates for the solution.

5.1.2 Line search

In the steady case, the initial mesh is generated independently of the *a priori* unknown shock location, and as such can begin quite far from alignment with the shock. Element collapses

were required to remove small and poorly shaped elements that arose between iterations as the coarse high-order mesh deformed to align with the shock. Because of this, we chose the step length parameter α_{k+1} of (3.43) based off a backtracking line search procedure based on the ℓ_1 merit function in our previous work [81].

In this work, we simply take $\alpha_k = 1$ and do not require any topology changes to the mesh. This is possible because we are able to construct high-quality initial guesses for the optimization problem in the unsteady case that already contain information about the shock (unlike the steady case). In Sections 5.2 and 4.4, we demonstrate this choice leads to rapid convergence of the SQP solver.

5.2 Linear advection

The first problem we consider is linear advection of a scalar quantity U through a domain $\Omega \subset \mathbb{R}^d$

$$\begin{aligned} \frac{\partial}{\partial t}U(x,t) + \frac{\partial}{\partial x_j}\left(U(x,t)\beta_j(x)\right) = 0 \qquad & \text{for } x \in \Omega,\ t \in [0,T] \\ U(x,0) = \bar{U}(x) \quad & \text{for } x \in \Omega, \end{aligned} \tag{5.3}$$

where $U : \Omega \times [0,T] \to \mathbb{R}$ is the conserved quantity implicitly defined as the solution of (5.3), $\beta : \Omega \to \mathbb{R}^d$ is the flow direction, and $\bar{U} : \Omega \to \mathbb{R}$ is the initial condition.

5.2.1 1D spatially varying advection

As a simple benchmark problem to demonstrate the capabilities and performance of the unsteady shock tracking framework, we consider the advection equation in one spatial dimension with a spatially varying advection field $\beta : \Omega \to \mathbb{R}$

$$\beta : x \mapsto 1 + \frac{1}{2}\sin^2(2\pi x) \tag{5.4}$$

with initial condition $\bar{U} : \Omega \to \mathbb{R}$:

$$\bar{U}(x) = \begin{cases} \sin(\pi x) & x \leq 0.5 \\ \sin(\pi(x-1)) & x > 0.5 \end{cases} \tag{5.5}$$

and periodic boundary conditions. We initialize with a mesh of the reference domain $\Omega_0 = [0,1]$, which we construct such that an element interface lies at the initial shock location ($x = 0.5$), i.e., the shock in the initial condition is tracked. The shock tracking solution is computed using a DG discretization on this equispaced mesh with 20 elements of degree $p = 4$, $q = 1$ and a DIRK3 temporal discretization with 25 time steps with final time $T = 0.25$ (Figure 5.2). The SQP solver is used with tolerances $\epsilon_1 = 10^{-6}$ and $\epsilon_2 = 10^{-8}$. At each timestep $n+1$, we are able obtain a solution to within the specified tolerances, which corresponds to the solution of the optimization problem (3.41) at time step n and stage s,

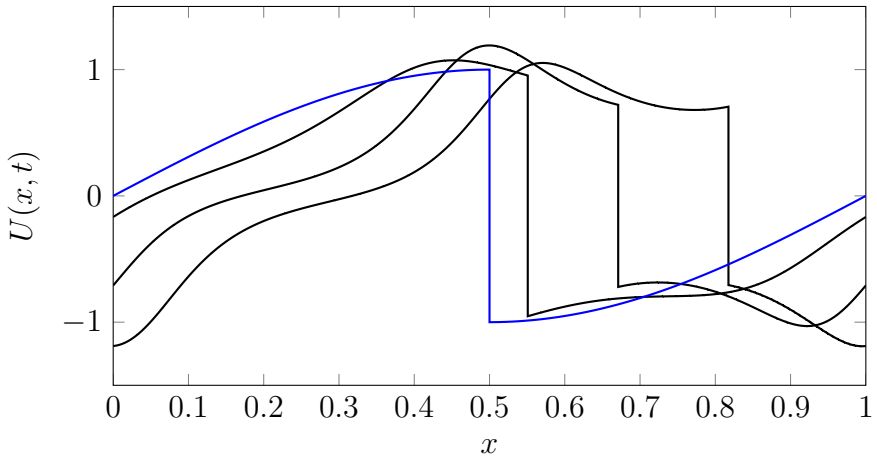


Figure 5.2: Method of lines solution of the one-dimensional, spatially varying advection equation with $p = 4, q = 1$. Initial condition $\bar{U}(x)$ (——) and tracking solution at times $t = 0.05, 0.15, 0.25$ (——).

and only need 2-3 iterations of the SQP solver to do so (Figure 5.3). Note that the sth stage tends to be more difficult for the SQP solver – one possible explanation is because each stage of the DIRK method has low stage order (first order), making it easier on the SQP solver compared to the sth stage, which is in fact the high-order solution because of our specific choice of DIRK schemes. We acknowledge this is an easy example for the SQP solver, and will demonstrate similar behavior for a more challenging example in the following sections.

We obtain a reference solution of (5.3) at $T = 0.25$ using the method of characteristics; the characteristic equations and corresponding solution are integrated using classical RK4 with 10,000 timesteps. The shock tracking solution compares well to the reference solution at the final time $T = 0.25$ (Figure 5.4). We use this reference solution to demonstrate high-order convergence in time of both the L^1 error of the solution at the final time and the shock location, verifying the design order of accuracy of the DIRKk schemes even in the presence of a discontinuity (Figure 5.5).

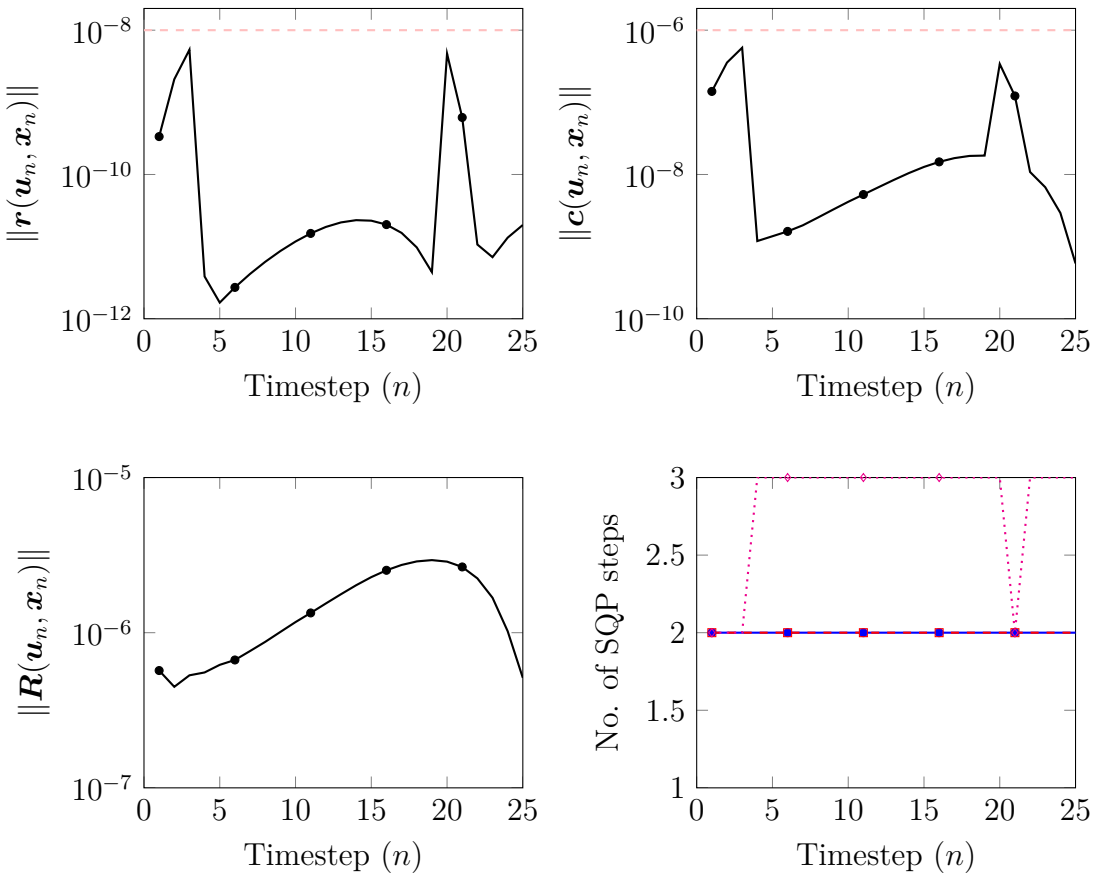


Figure 5.3: Final converged value of the constraint (*top left*), optimality condition (*top right*), and objective function (*bottom left*) for the solution to 1D spatially varying advection at each timestep (—•—) and the specified tolerances (- - -), and the number of SQP steps needed at each stage of each time step for convergence (*bottom right*); stage 1: (—•—), stage 2 (-▫-), stage 3 (···◇···).

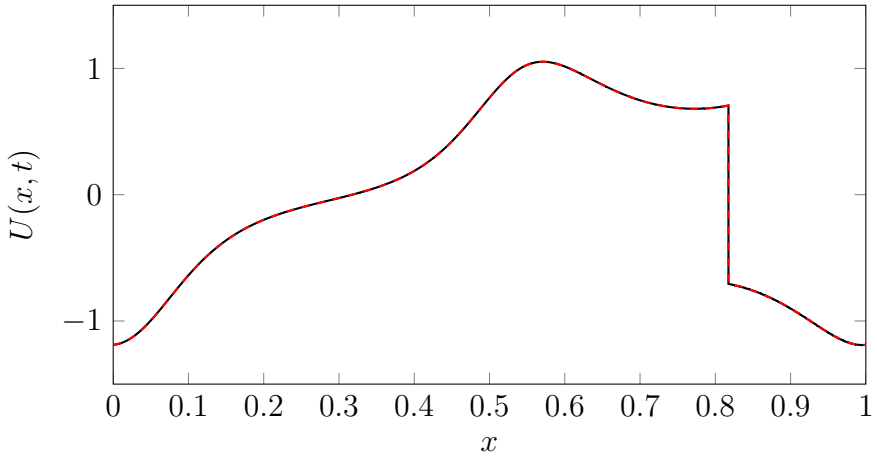


Figure 5.4: Comparison of shock tracking (——) and reference (- - -) solutions at $T = 0.25$.

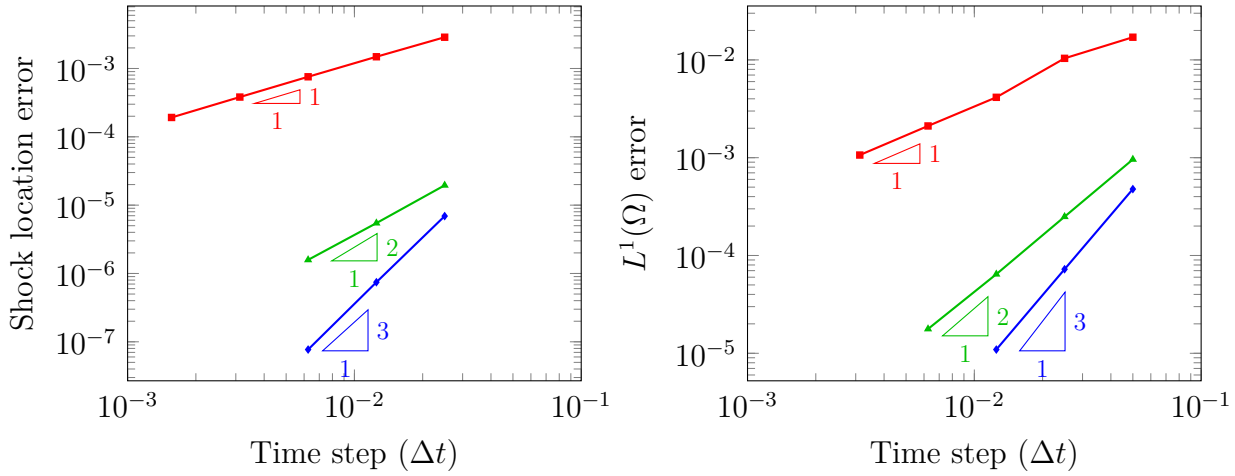


Figure 5.5: Temporal convergence of the DIRK1 (—■—), DIRK2 (—▲—), and DIRK3 (—◆—) schemes for the L^1 error of the shock location (*left*) and the solution (*right*) for the spatially varying advection equation.

5.3 Time-dependent, inviscid Burgers' equation

The time-dependent, inviscid Burgers' equation governs nonlinear advection of a scalar quantity through the domain $\Omega \subset \mathbb{R}^d$

$$\begin{aligned} \frac{\partial}{\partial t}U(x,t) + \frac{\partial}{\partial x_j}\left(\frac{1}{2}U(x,t)^2\beta_j\right) &= 0 \qquad \text{for } x \in \Omega,\ t \in [0,T] \\ U(x,t) &= 0 \qquad \text{for } x \in \partial\Omega,\ t \in [0,T] \\ U(x,0) &= \bar{U}(x) \quad \text{for } x \in \Omega, \end{aligned} \tag{5.6}$$

where $U : \Omega \times [0,T] \to \mathbb{R}$ is the conserved quantity implicitly defined as the solution of (5.6), $\beta \in \mathbb{R}^d$ is the flow direction, and $\bar{U} : \Omega \to \mathbb{R}$ is the initial condition. We investigate this problem in both one and two dimensions.

5.3.1 1D Burgers' equation

For Burgers' equation in one spatial dimension, we take $\beta = 1$ and consider the initial condition $\bar{U} : \Omega \to \mathbb{R}$

$$\bar{U} : x \mapsto 2(x+1)^2(1 - H(x)), \tag{5.7}$$

where $H : \mathbb{R} \to \{0,1\}$ is the Heaviside function. Our approach only requires a mesh of the reference domain $\Omega_0 = [-1,1]$, which we construct such that an element interface lies at the initial shock location ($x = 0$), i.e., the shock in the initial condition is tracked. The shock tracking solution is computed using a DG discretization on this equispaced mesh with 20 elements of degree $p = 4$, $q = 1$ and a DIRK3 temporal discretization with 20 time steps with final time $T = 1$ (Figure 5.6).

5.3.2 2D Burgers' equation

We now consider Burgers' equation in two spatial dimensions. We take $\Omega_0 = [-1,1]^2$ as our two-dimensional spatial domain, $\beta = (1,0)$ as the flow direction, and $\bar{U} : \Omega \to \mathbb{R}$ as the initial condition, defined as

$$\bar{U} : (x_1, x_2) \mapsto \begin{cases} (0.5 - 2(x_2^2 - 0.25))\left(\frac{4}{3}(x_1 + 0.75)\right) & x \in \Omega_\square \\ 0 & \text{elsewhere,} \end{cases} \tag{5.8}$$

where $\Omega_\square \coloneqq [-0.75, 0] \times [-0.5, 0.5]$. The problem is constructed such that the initially straight shock curves over time, which is tracked by the high-order mesh. The shock tracking solution is computed using a DG discretization on a mesh with 128 simplex elements of degree $p = 2$, $q = 2$ and a DIRK3 temporal discretization with 40 time steps with final time $T = 2$ (Figure 5.7). The mesh smoothing procedure described in Section 5.1.1 is important here to maintain high-quality elements as the shock moves across the domain.

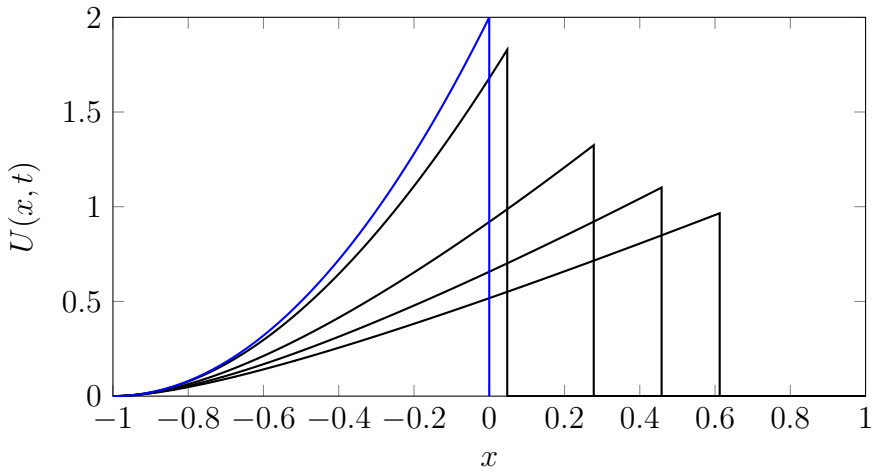


Figure 5.6: Method of lines solution of the one-dimensional, inviscid Burgers' equation with $p = 4, q = 1$, including initial condition $\bar{U}(x)$ (——) and tracking solution at times $t = 0.05, 0.35, 0.65, 0.95$ (——).

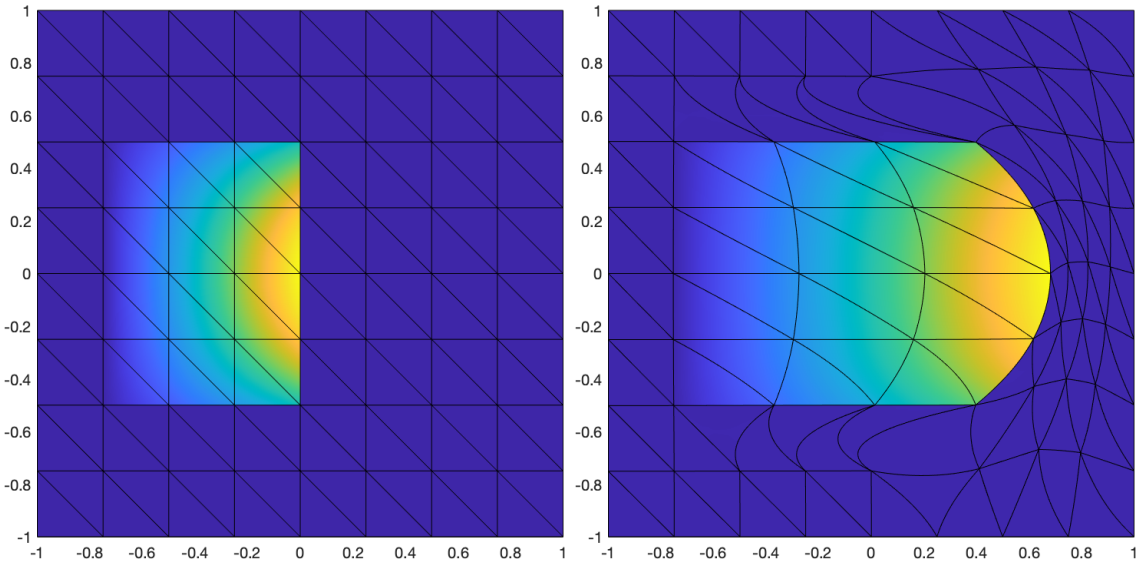


Figure 5.7: Method of lines solution of two-dimensional, inviscid Burgers' equation with $p = q = 2$. Initial condition $\bar{U}(x)$ (*left*) and solution at $T = 2$ (*right*)

5.4 Unsteady, compressible Euler equations

The Euler equations govern the flow of an inviscid, compressible fluid through a domain $\Omega \subset \mathbb{R}^d$

$$\begin{aligned} \frac{\partial}{\partial t}\rho(x,t) + \frac{\partial}{\partial x_j}\left(\rho(x,t)v_j(x,t)\right) &= 0 \\ \frac{\partial}{\partial t}\left(\rho(x,t)v_i(x,t)\right) + \frac{\partial}{\partial x_j}\left(\rho(x,t)v_i(x,t)v_j(x,t) + P(x,t)\delta_{ij}\right) &= 0 \\ \frac{\partial}{\partial t}\left(\rho(x,t)E(x,t)\right) + \frac{\partial}{\partial x_j}\left(\left[\rho(x,t)E(x,t) + P(x,t)\right]v_j(x,t)\right) &= 0 \end{aligned} \tag{5.9}$$

for all $x \in \Omega$, $t \in [0,T]$, $i = 1,\ldots,d$ and summation is implied over the repeated index $j = 1,\ldots,d$, where $\rho : \Omega \times (0,T) \to \mathbb{R}_+$ is the density of the fluid, $v_i : \Omega \times (0,T) \to \mathbb{R}$ for $i = 1,\ldots,d$ is the velocity of the fluid in the x_i direction, and $E : \Omega \times (0,T) \to \mathbb{R}_+$ is the total energy of the fluid, implicitly defined as the solution of (5.9). For a calorically ideal fluid, the pressure of the fluid, $P : \Omega \times (0,T) \to \mathbb{R}_+$, is related to the energy via the ideal gas law

$$P = (\gamma - 1)\left(\rho E - \frac{\rho v_i v_i}{2}\right), \tag{5.10}$$

where $\gamma \in \mathbb{R}_+$ is the ratio of specific heats. By combining the density, momentum, and energy into a vector of conservative variables $U : \Omega \times [0,T] \to \mathbb{R}^{d+2}$, defined as

$$U : (x,t) \mapsto \begin{bmatrix} \rho(x,t) \\ \rho(x,t)v(x,t) \\ \rho(x,t)E(x,t) \end{bmatrix} \tag{5.11}$$

the Euler equations are a conservation law of the form (2.1). We investigate the shock tracking framework on two benchmark examples: the Shu-Osher problem and a blast wave problem in 2D.

5.4.1 Shu-Osher problem

The Shu-Osher problem [64] is a one-dimensional idealization of shock-turbulence interaction where a Mach 3 shock moves into a field with a small sinusoidal density disturbance. The flow domain is $\Omega_0 = [-4.5, 4.5]$, and the initial condition is given in terms of the density,

velocity, and pressure as

$$\begin{aligned}
\rho(x) &= \begin{cases} 3.857143 & x < -4 \\ 1 + 0.2\sin(5x) & x \geq -4 \end{cases} \\
v(x) &= \begin{cases} 2.629369 & x < -4 \\ 0 & x \geq -4 \end{cases} \\
P(x) &= \begin{cases} 10.3333 & x < -4 \\ 1 & x \geq -4, \end{cases}
\end{aligned} \tag{5.12}$$

and the density, velocity, and pressure are prescribed at $x = -4.5$ and the velocity is prescribed at $x = 4.5$ (values can be read from the initial condition). The shock tracking solution is computed using a DG discretization on a mesh with 288 elements of degree $p = 4$, $q = 1$, half of which are equispaced to the left of the initial shock location from $[-4.5, -4]$, and the other half equispaced from $[-4, 4.5]$. The temporal discretization is done by the DIRK3 method with 110 time steps with final time $T = 1.1$. The final time is chosen such that waves trailing behind the primary shock do not steepen into shock waves; shock formation will be the subject of future work. In Figure 5.8, we present the shock tracking solution along with a reference solution computed using a fifth-order WENO method with 200 elements and temporal integration via RK4 with 110 timesteps [64]. The shock tracking solution actually overshoots the reference solution at the formation of the trailing waves, which suggests the reference solution is being overly dissipated by the WENO scheme (*left inset*). The shock is perfectly represented by the aligned mesh in the shock tracking solution compared to the reference (*right inset*).

The SQP solver is used with tolerances $\epsilon_1 = 10^{-4}$ and $\epsilon_2 = 10^{-8}$, which is easily attained at each Runge-Kutta stage and timestep and in very few SQP iterations (Figure 5.9). We note that the objective function is steadily growing because we are using a fixed amount of resolution to represent an increasingly oscillatory numerical solution. This is not an issue of any other fundamental concern, and can be addressed with adaptive element collapses and refinement, which is straightforward in the one-dimensional case.

The Shu-Osher problem is ideal for implicit shock tracking because it starts off with a well-defined shock that maintains its topology over time and the flow away from the shock is smooth. At present, one could imagine combining this shock tracking approach with a shock capturing method, where strong well-defined shocks could be tracked and weaker shocks which form over time could be captured [36]. Future work will focus on the simulation of the full Shu-Osher problem to additionally track the smaller shocks which form as the trailing waves steepen.

5.4.2 Blast wave

In the final example, we consider a spherical blast wave problem featuring a strong, radially expanding shock. Our problem is inspired by the classic Sedov problem, which is an idealized

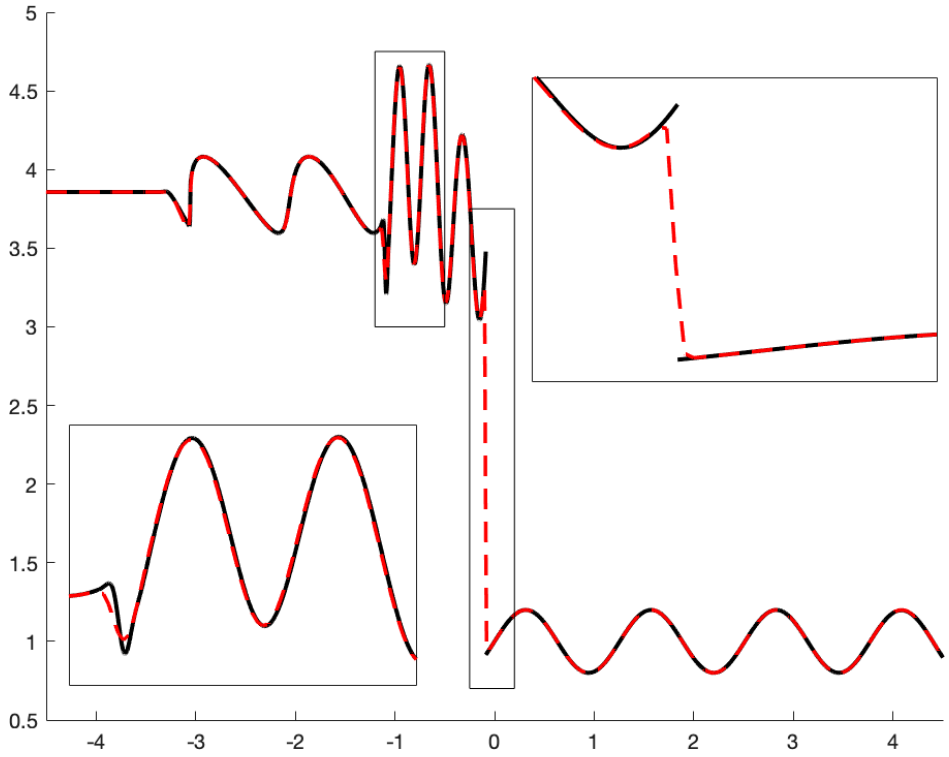


Figure 5.8: Density at $T = 1.1$ of Shu-Osher problem for the reference (- - -) and shock tracking (—) solutions.

model of the self-similar evolution of a cylindrical (or spherical) blast wave starting from a large total energy deposited at a single point placed into an otherwise homogenous medium of a uniform ambient density with negligible pressure. The Sedov problem is a workhorse verification test for traditional shock capturing methods since there is an analytical solution [62] available in one-, two-, and three-dimensions for comparison. It is primarily used to test geometrical concerns, such as the ability to track a curved shock and maintain spherical symmetry.

We take $\Omega = [-1, 1]^2$ as our two-dimensional spatial domain with the initial condition given in terms of the density, velocity, and pressure by a function of the distance from the origin $r = \sqrt{x_1^2 + x_2^2}$ as

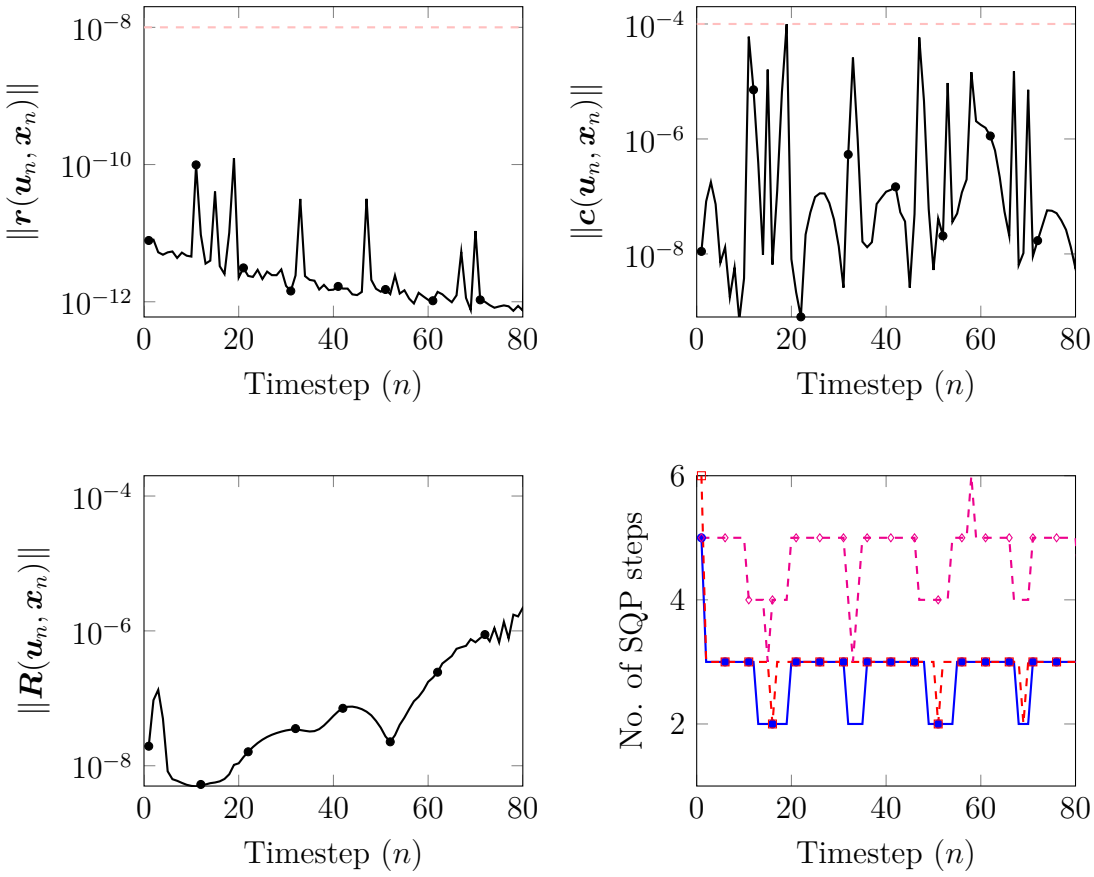


Figure 5.9: Final converged value of the constraint (*top left*), optimality condition (*top right*), and objective function (*bottom left*) for the solution to the Shu-Osher problem at each timestep (—•—) and the specified tolerances (- - -), and the number of SQP steps needed at each stage of each time step for convergence (*bottom right*); stage 1: (—•—), stage 2 (-▫-), stage 3 (-⋄-).

$$\begin{aligned}
\rho(r) &= \begin{cases} \frac{5.378}{0.25^2} r^2 & r \leq 0.25 \\ 1 & \text{elsewhere} \end{cases} \\
v(r) &= \begin{cases} \frac{1.304}{0.25} r & r \leq 0.25 \\ 0 & \text{elsewhere} \end{cases} \\
P(r) &= \begin{cases} \frac{0.978}{0.25^2} r^2 + 1 & r \leq 0.25 \\ 10^{-3} & \text{elsewhere} \end{cases}.
\end{aligned} \tag{5.13}$$

Like the other numerical examples considered in this work, the initial condition is constructed such that the shock is exactly meshed. This initial condition is inspired by the solution to the Sedov problem at a time when the primitive variables have already taken on a "bowl-like" structure. The shock tracking solution is computed using a DG discretization on a mesh with 422 simplex elements of degree $p = 2$, $q = 2$ and a DIRK2 temporal discretization with 140 time steps and final time $T = 0.14$ (Figure 5.10). In the shock tracking framework, the ability to track curved shocks accurately while maintaining radial symmetry comes very naturally. Since the initial value for pressure in the ambient region is close to zero and much smaller than the value inside the shock, a slight modification was made to the SQP solver to avoid negative values for energy in the ambient region. After each SQP step, the elements in the ambient region with a negative value for energy are reinitialized with their element-wise average. This helps stabilize the solver in its initial iterations, and it quickly converges from there.

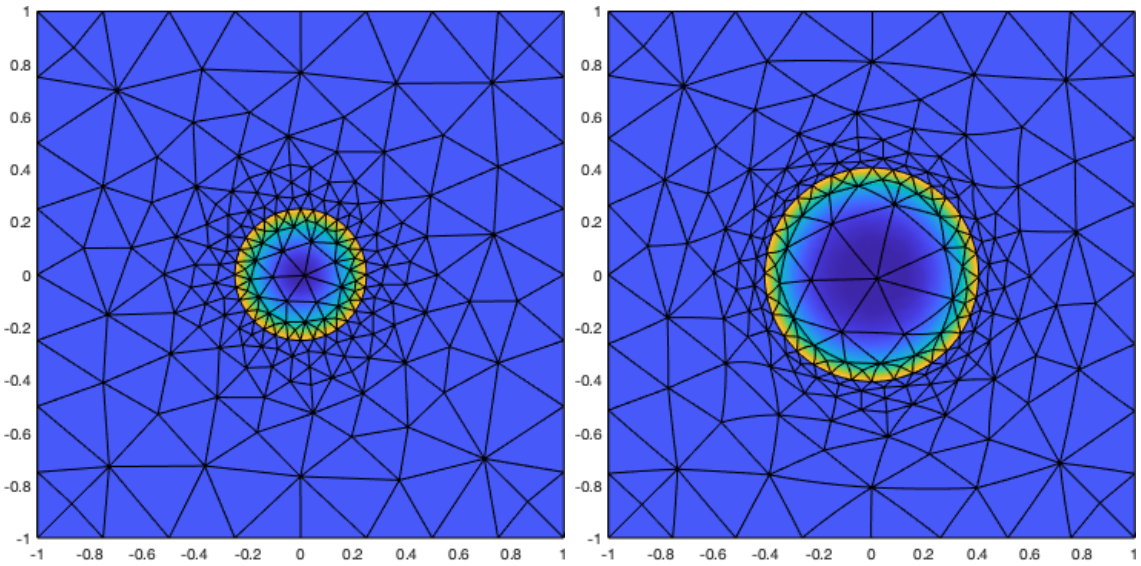


Figure 5.10: Method of lines solution of two-dimensional blast wave problem with $p = q = 2$. Initial condition $\bar{U}(x)$ (*left*) and solution at $T = 0.14$ (*right*)

Chapter 6

Local element operations for curved meshes

Mesh optimization procedures are generally a combination of node smoothing and discrete operations which affect a small number of elements to improve the quality of the mesh elements. These procedures are useful as a post-processing step in mesh generation procedures and in applications such as fluid simulations with severely deforming domains. In order to perform high-order mesh optimization, these ingredients must also be extended to high-order meshes. In this section, we present a method to perform local element operations on curved meshes. The mesh operations discussed in this work are edge/face swaps, edge collapses, and edge splitting (more generally refinement) for triangular meshes. These local operations are performed by first identifying the patch of elements which contain the edge/face being acted on, performing the operation as a straight-sided one, placing the high-order nodes by an isoparametric mapping from the master element, and smoothing the high-order nodes on the elements in the patch by minimizing a Jacobian-based mesh distortion measure. Since the initial straight-sided guess frequently results in invalid elements, a regularization of this distortion measure which allows for mesh untangling is critical for the optimization to succeed. We present several examples to demonstrate these local operations and how they can be combined with a high-order node smoothing procedure to maintain mesh quality for problems with severe deformations.

6.1 Background

Much of the great promise of high-order methods is due their ability to handle complex geometries with unstructured meshes. These methods require high-order (curved) meshes to ensure the curved geometry is represented with sufficient accuracy to enable the high-order convergence rate of these methods. However, the existence (or rather, lack thereof) of robust and automated high-order mesh generation procedures is still one of the main bottlenecks in the widespread adoption of these methods in the design process. To this end,

the high-order community has identified high-order mesh generation as a topic deserving of considerable research attention and as one of the top pacing items in high-order method research [75, 74]. One might say curved meshing solidified itself a "mainstream" topic of research in the meshing community in 2015, the first year it had its own dedicated session at the International Meshing Roundtable [42]. The introduction of [70] contains a recent survey of the current state of affairs of high-order meshing research and ample additional references. In terms of available software, Gmsh is a widely used free software with the capability to generate and display curved meshes for elements of arbitrarily high-order [27]. Up until very recently, there were no commercial high-order mesh generators available before Pointwise released support for this in 2019 [1].

Most straight sided mesh generation procedures proceed in two steps: 1) the initial creation of a mesh from geometry through a method like advancing front, Delaunay or octree, and 2) a mesh optimization post-processing step to improve element quality. There is a large body of literature on these heuristic mesh optimization procedures (sometimes referred to as a "mesh clean-up phase"). While those works may differ in their notions of optimality and specific techniques used, they largely agree the best way to perform mesh optimization is through a combination of node movement/smoothing and discrete, localized element operations. Freitag and Ollivier-Gooch [22] consider various node smoothing procedures and face and edge swapping operations, and show that each mechanism fails to significantly improve the mesh quality when used individually but result in very high quality meshes when combined. They offer a variety of empirical recommendations on how to do so and demonstrate the effectiveness of their schedules on a variety of tetrahedral mesh geometries. De Cougny and Shephard [11] combine edge collapses and splitting operations for the purpose of mesh adaptation. Klinger and Shewchuk [40] consider a richer set of operations in an effort to take these ideas to an extreme to find the highest quality mesh, assuming that speed is not the highest priority. We refer the reader to Section 3 of their paper where they survey many of the commonly (and some uncommonly) used mesh operations.

Now the question is how to extend these "mesh clean-up" ideas to high-order meshes. Existing work [59, 37, 15, 26] on high-order node smoothing mostly takes the approach of optimization-based minimization of some energy functional or objective function that quantifies the global high-order mesh distortion. The key issues here are the need for mesh distortion metrics for high-order elements [60, 25] and a procedure for high-order untangling [68, 16]. In principle, some of the approaches to curved mesh generation such as the solution of a linear/nonlinear elasticity analogy [55, 76, 52, 49] or a PDE-based approach [21] could also be adapted for the purpose of high-order mesh smoothing. There is practically no literature on the use of local element operations for curved meshes. To the best of the author's knowledge, the only existing work on high-order mesh operations [19] considers face and edge swaps for meshes with second-order elements.

It is clear that further developments in these areas will be required to advance the current state of curved mesh generation and adaptivity. The difficulty of directly acting on high-order meshes is one possible explanation for why most curved mesh generation approaches are of the *a priori* variety rather than direct methods which proceeds in the same two phases

as most linear mesh generators do. In the context of high-order finite element solutions, solution quality is frequently dictated by the mesh element of the worst quality. Local mesh operations for curved meshes could be an inexpensive and direct way to deal with this and other applications such as the repair of deforming high-order domains occurring in time-dependent fluid simulations.

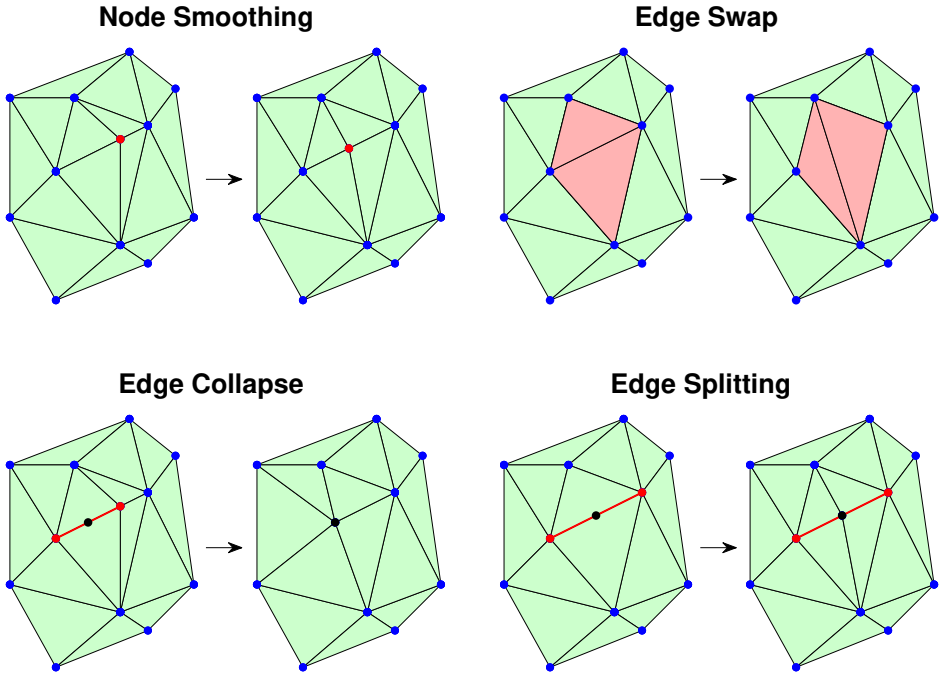


Figure 6.1: Straight sided mesh operations.

The main goal of this section is to answer the question, "How we can extend these local element operations (Figure 6.1) to high-order triangular meshes?" One key difference from the low-order case is that there are many valid ways to perform these operations due to the additional degrees of freedom afforded by the high-order nodes (Figure 6.2). So there is now an *optimal* way to perform these local operations by placing these additional nodes in a way to minimize a high-order Jacobian-based mesh distortion measure (Section 6.2). We go onto describe the actual process of performing these operations in Section (6.3) and some practical details related to their implementation, and illustrate their use in multiple numerical examples in Section (6.4) that show how these operations can be combined with high-order mesh smoothing and untangling to repair curved meshes that undergo severe deformations.

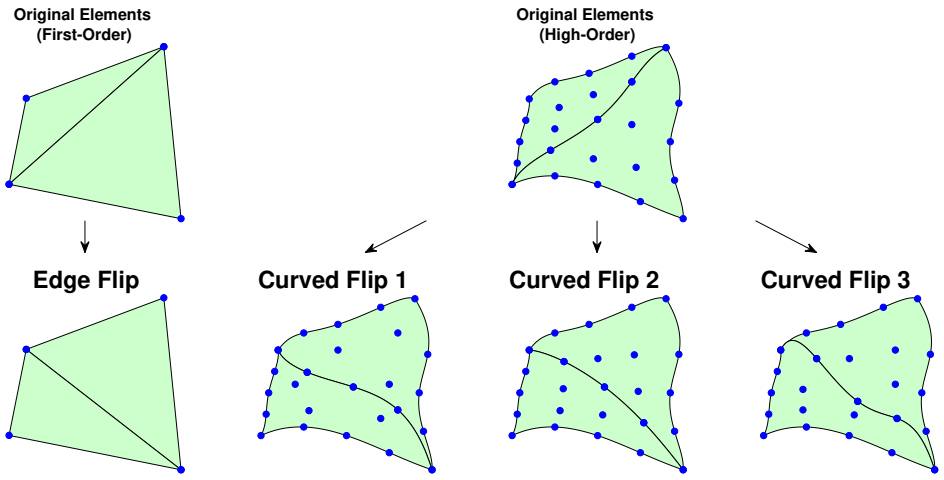


Figure 6.2: Only one possible flip for straight-sided elements, but many possible flips in the high-order setting.

6.2 Distortion measure and untangling

We describe the construction of the high-order distortion measure used in this work for node smoothing from a standard linear distortion measure, which is then regularized to allow for simultaneous smoothing and untangling. We also offer some comments on how this measure is used our setting for optimization.

6.2.1 High-order distortion measure

We introduce the distortion measure for linear elements (6.2). This measure – known as the inverse mean-ratio shape measure – was first introduced in [43, 44]. This measure is widely used in the meshing community as a shape distortion metric due to its many desirable properties such as the standard invariance to translation, rotation, and scaling, but is also special as a geometric shape measure that also enjoys many of the desirable properties of an *algebraic* mesh quality metric [41]. In particular, this measure is shown to be convex [50] and well-suited for optimization.

Consider a linear element E^I having the desired shape and size (the ideal element) and a corresponding element in physical space E^P. From these we can define a unique affine mapping $\phi_E : E^I \to E^P$. It is frequently convenient to introduce the notion of the master element E^M and define two additional mappings ϕ_I, ϕ_M (Figure 6.3a) that map from the master element to the ideal and physical element, respectively. We can define ϕ_E in terms of the these mappings as follows

$$\phi_E : E^I \xrightarrow{\phi_I^{-1}} E^M \xrightarrow{\phi_P} E^P. \tag{6.1}$$

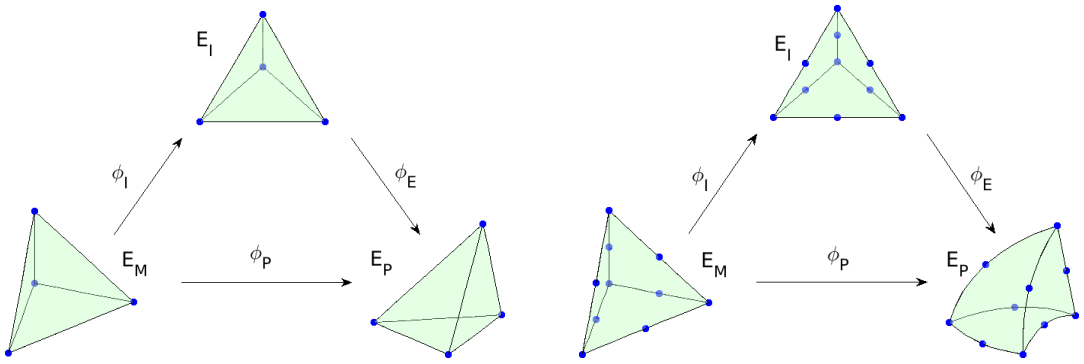


Figure 6.3: Mappings between the master, ideal and physical elements: (a) linear case *(left)* and (b) high-order case *(right)*.

The element shape distortion metric η is defined as

$$\eta(D\phi_E) = \frac{||D\phi_E||_F^2}{d|\sigma|^{2/d}} \tag{6.2}$$

where d is the spatial dimension, $||\cdot||_F$ is the Frobenius norm, and $\sigma = \det(D\phi_E)$. Note that for linear elements, since ϕ_E is affine, its Jacobian $D\phi_E$ is constant. This distortion measure quantifies the deviation of the shape of the physical element E^P with respect to the ideal element E^I. This distortion measure is 1 for the ideal element and tends to infinity as the physical element degenerates. It is often convenient to define a corresponding quality measure q as

$$q = \frac{1}{\eta} \in [0, 1]. \tag{6.3}$$

For high-order elements, we can no longer apply (6.2) directly because the mapping ϕ_E is no longer affine, so $D\phi_E$ is no longer constant. Roca et al. describes an extension of the linear distortion measure to high order triangular [60] and tetrahedral [25] elements. The idea is that since the linear measure quantifies the local deviation between the ideal and physical elements, we can obtain a high-order element-wise distortion $\hat{\eta}$ measure by integrating the linear measure on the physical high order element

$$\hat{\eta}(D\phi_E) = \left(\frac{1}{|t|} \int_t \eta^r(D\phi_E)\, dx \right)^{1/r} \tag{6.4}$$

where $|t|$ is the area of the physical element and $r = 2$ in this work. Explicit expressions for the isoparametric mappings for the high-order case (Figure 6.3b) in terms of shape functions are given in [60].

6.2.2 Regularization and untangling

The distortion measure η needs to be modified to eliminate the singularity so an optimization procedure can recover from an invalid initial configuration. We use the regularization introduced in [16] which replaces σ in the denominator of (6.2) by

$$\sigma_\delta(\sigma) = \frac{1}{2}\left(\sigma + \sqrt{\sigma^2 + 4\delta^2}\right) \tag{6.5}$$

where δ is a positive element-wise parameter. This regularized Jacobian $\sigma_\delta(\sigma)$ is a monotonically increasing function of σ such that $\sigma_\delta(0) = \delta$, which tends to 0 when σ tends to $-\infty$, which allows us to overcome the vertical asymptote at $\sigma = 0$ in the original measure. This parameter δ is only set for a non-zero value if there exists an invalid element in the mesh under consideration, otherwise it is set to zero for a mesh with all valid elements. Using the regularized Jacobian, we can now modify (6.2) to obtain a linear distortion measure capable of *simultaneous* smoothing and untangling

$$\eta_\delta(D\phi_E) = \frac{||D\phi_E||_F^2}{d|\sigma_\delta|^{2/d}} \tag{6.6}$$

and use this to define a high-order regularized measure $\hat{\eta}_\delta$ analogous to (6.4). Figures 5 and 6 of [26] provide a useful illustration of how this regularized distortion metric compares to the original for a simple example.

What remains is the issue of how to pick this positive elementwise regularization parameter δ. We certainly require the original ($\hat{\eta}$) and regularized ($\hat{\eta}_\delta$) distortion measures to have nearby minima, so δ needs to be sufficiently small. On the other hand, δ has to be large enough to ensure that $4\delta^2$ is significant compared to σ^2 in the expression for σ_δ. When this idea was originally introduced in [16], this parameter was chosen in a fairly ad hoc manner by testing a few values for a given initial tangled mesh. In [26], a heuristic was developed to choose a constant value of δ for each element. Their setting deals with high-order mesh generation by curving the boundaries of a well-shaped straight-sided mesh and minimizing the regularized distortion metric. For them, in the mapping ϕ_E, each physical element E^P in the curved mesh has the ideal element E^I as the corresponding element in the original straight-sided mesh. They propose to choose this parameter δ solely based off information from the straight-sided ideal element. Defining $\sigma^* = -\det \phi_I$ and imposing

$$\sigma_\delta(\sigma^*) = \frac{1}{2}\left(\sigma^* + \sqrt{(\sigma^*)^2 + 4\delta^2}\right) = \tau > 0 \tag{6.7}$$

for some given tolerance τ implies

$$\delta(\sigma^*) = \frac{1}{2}\sqrt{(2\tau + |\sigma^*|)^2 - (\sigma^*)^2} = \sqrt{\tau^2 + \tau|\sigma^*|}. \tag{6.8}$$

They argue that τ should be small compared to σ and select $\tau = \alpha|\sigma^*|$, giving the final value for δ as

$$\delta(\sigma^*) = |\sigma^*|\sqrt{\alpha^2 + \alpha}.$$

The value $\alpha = 10^{-3}$ is observed to work well in practice and accomplish the tradeoff required on the value of δ (Figure 6.4). Our setting deals with high-order mesh smoothing, so we choose the ideal element E^I to always be the equilateral triangle/tetrahedron. For the unregularized distortion measure, the actual size of E^I is irrelevant since the measure is invariant to scaling. This is no longer true for the regularized distortion measure. So in order to maintain the assumption from their setting that each E^I and E^P are roughly of the same size, we need to scale the ideal element to have the same volume as each corresponding physical element.

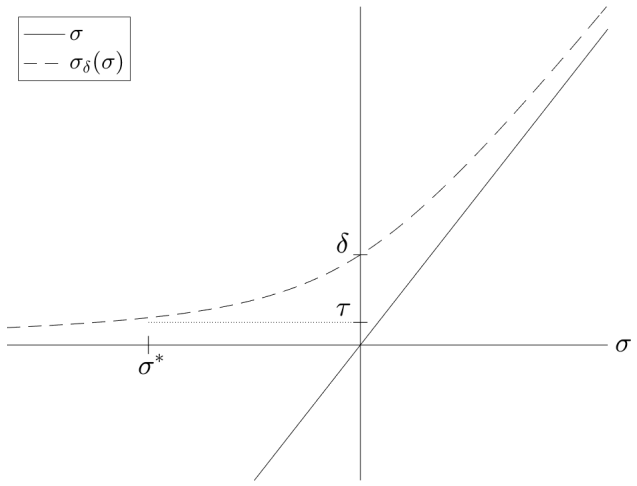


Figure 6.4: Representation of $\sigma_\delta(\sigma)$ (reproduced from [26])

6.2.3 Numerical optimization

We need to aggregate the distortion measures for single elements into a single quantity for a group of elements for optimization purposes. What we really want is to optimize for the worst element, since the distortion of the worst element in a mesh is often what drives the solution quality in the context of numerical simulation. But since minimizing the maximum distortion is difficult optimization-wise, we consider the normalized sum of squares of the elementwise distortion of each element e_i in a group of elements M

$$\hat{\eta}_{agg}(M) = \frac{1}{|M|} \sum_{e_i \in M} \hat{\eta}_\delta(D\phi_{e_i})^2. \tag{6.9}$$

The two settings we perform mesh smoothing are over a small "patch" of elements that share a single vertex, or over the entire mesh. For small problems such as patches, solving the global problem works well. For smoothing over the entire mesh, we largely follow the localized approach given in Appendix A of [26], which considers a "Gauss-Seidel" like method which iteratively updates each node by a Newton step with backtracking line search until either the

objective function or the amount of node movement satisfies some stopping criteria. Instead of looping over nodes, we loop over patches (Figure 6.5), which seems to help mitigate some of that "back and forth" which has been shown to empirically improve the robustness of the high-order optimization [15]. We note this solver offers no guarantees for especially tangled or pathological meshes, due to the choice of solver, the issues with the regularization method which many authors have pointed out fails quite often in practice such as if the boundary of the mesh is also tangled [34], or the possible failure of the high-order Gauss quadrature rule to detect inversion at the corners of our high-order elements.

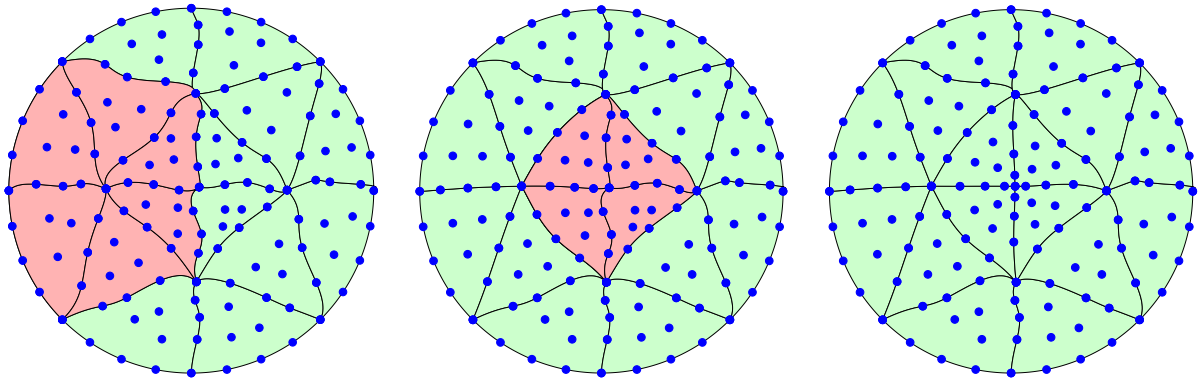


Figure 6.5: Patch-based high-order mesh smoothing

6.3 Local element operations

We describe how to perform the local element operations of edge/face swaps, edge collapses, and edge splitting for curved, simplex elements in 2D. The operations roughly proceed by performing the action as a straight-sided one and simultaneously untangling and smoothing the result. We note some practical issues along with differences in the high-order setting.

6.3.1 Edge/face swaps

Once a group of elements is identified to perform edge/face swaps on, we first do the straight-sided initial guess. This is done by freezing the boundary, discarding all the interior nodes and replacing them in the flipped configuration by the straight-sided information. This initial flip is extremely prone to tangling even when starting out with fairly well-shaped elements. By the high-order distortion measure (6.4), many high-order elements which seem valid on visual inspection with all their interior nodes contained in the boundary are actually invalid. Finally, we apply the simultaneous untangling and smoothing procedure described in the previous section to arrive at the curved flip (Figure 6.6). We decide to accept the resulting flip if it results in a lower aggregate distortion $\hat{\eta}_{agg}$ from the original configuration.

The normalization by the number of elements in the group in the definition is to allow comparison of the original and flipped configuration, since in 3D there are many possible flips which in general change the number of elements.

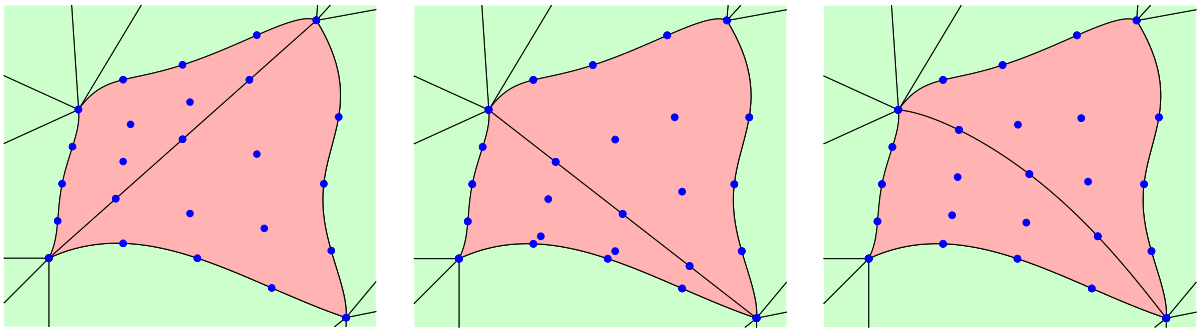


Figure 6.6: Edge flips in 2D: original elements *(left)*, initial straight-sided flip *(middle)*, curved flip *(right)*

6.3.2 Edge collapses

Once an edge is identified to collapse, we first do the straight-sided initial guess. This is done by identifying the patch of elements that contains either endpoint of the edge being collapsed, freezing its boundary, discarding all the interior nodes, and replacing them by the straight-sided edge collapse to the midpoint of the original curved edge. Then we apply the simultaneous untangling and smoothing procedure to arrive the final curved collapse (Figure 6.7). If the edge identified for collapse is on the boundary of the domain, information will inevitably be lost but can be well mitigated in the high-order case by performing projection based interpolation on the patch boundary by standard techniques encountered in *hp*-FEM methods [13].

6.3.3 Edge splitting

Depending on the setting, the edges to be split could either be directly identified or implied by choosing elements. Either way, once an element is identified to be split and the straight sided guess is performed according to one of the templates in 2D (Figure 6.8), its neighbors must also be split by a straight-sided guess with the corresponding template to prevent hanging nodes. Then, we apply simultaneous untangling and smoothing on this group of elements to arrive at the final curved split (Figure 6.9).

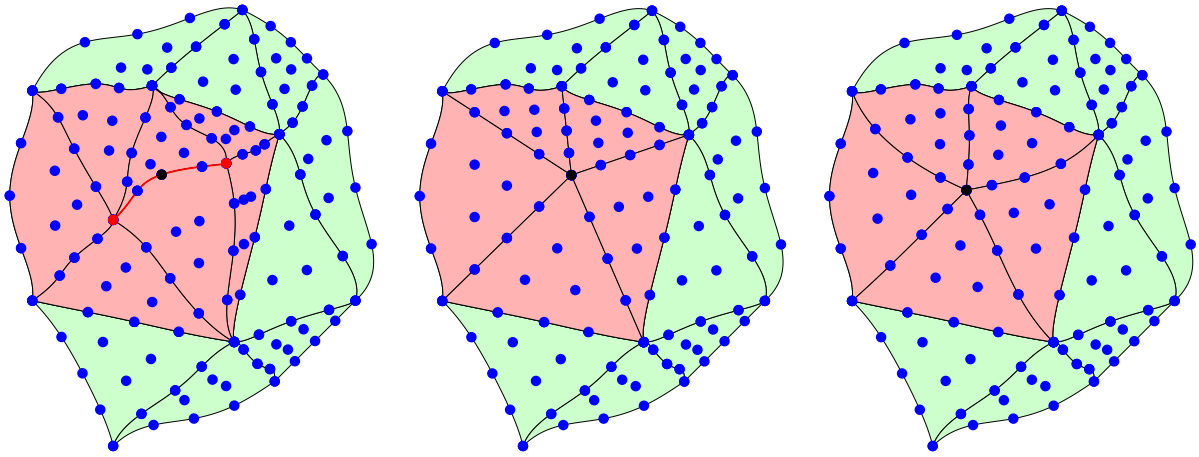


Figure 6.7: Edge collapses in 2D: original elements *(left)*, initial straight-sided collapse *(middle)*, curved collapse *(right)*

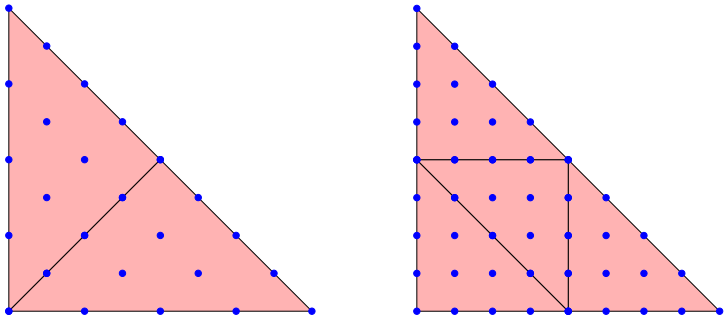


Figure 6.8: Element subdivision templates in 2D: $1 \to 2$ *(left)* and $1 \to 4$ *(right)*

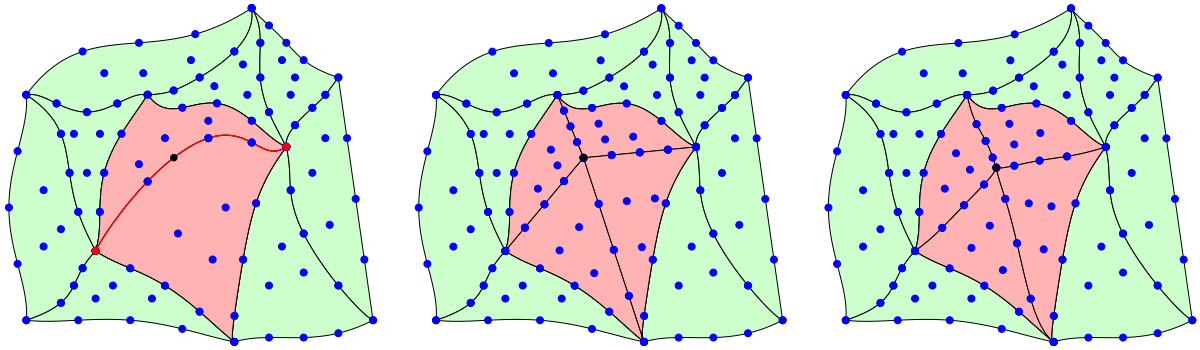


Figure 6.9: Edge splitting in 2D: original elements *(left)*, initial straight-sided split *(middle)*, curved split *(right)*

6.4 Numerical experiments

We demonstrate how curved local element operations can be combined with high-order mesh smoothing to maintain element quality and sizing for high-order meshes subject to severe deformations in 2D. All the meshes used for the numerical experiments are generated by Gmsh [27] and are of polynomial degree four.

6.4.1 Rotation

We first demonstrate how node smoothing combined with edge flips can maintain element quality under rotation. The 2D mesh is a circle-in-square mesh of 116 elements, where the circle is centered at the origin with radius 0.5 inside the square $[-1, 1]^2$ (Figure 6.10 *left*). At each step, we rotate the nodes on the circle, perform high-order node smoothing while keeping the nodes on the circle fixed, apply curved flips according to Algorithm 1, and smooth again. Algorithm 1 constructs a priority queue of elements with higher distortion than a floating threshold value and considers all topologically possible flips for each element, accepting the one (if any) that results in the greatest improvement. It performs sweeps over all the elements with a reset floating threshold for each sweep until no further improvement is possible.

Algorithm 1: Application of Curved Flips

```
Input curved mesh M with elements {e_i}
if dim = 2D then tmin = 1.5
Set distortion threshold T = min(tmin, min(D)), D(i) = distortion of e_i
Create priority queue P = {(e_j, D(j)), ∀j D(j) > T descending}
while true do
    for i in P do
        if dim = 2D then consider all possible 2 → 2 flips with e_i
        if best of topologically possible flips improves η̂_agg then
            Update the mesh M
            Remove all involved elements from the queue P and renumber
        end
    end
    if no flips done in sweep then
        return M
    end
    Recompute T and P
end
```

Rotation cannot exceed much more than a quarter turn with only node smoothing (Figure 6.10 *middle*). A straight sided mesh cannot rotate this far with smoothing alone; the presence

of high order nodes grant us the flexibility to maintain valid (albeit highly stretched) elements for more severe deformations. The 2D case can maintain high quality indefinitely with the application of flips. It is well known in the straight-sided case that such greedy flip algorithms like the one used here converge to an optimal triangulation (in the sense of Delaunay) in the 2D case, but offers no such guarantees in higher dimensions [12].

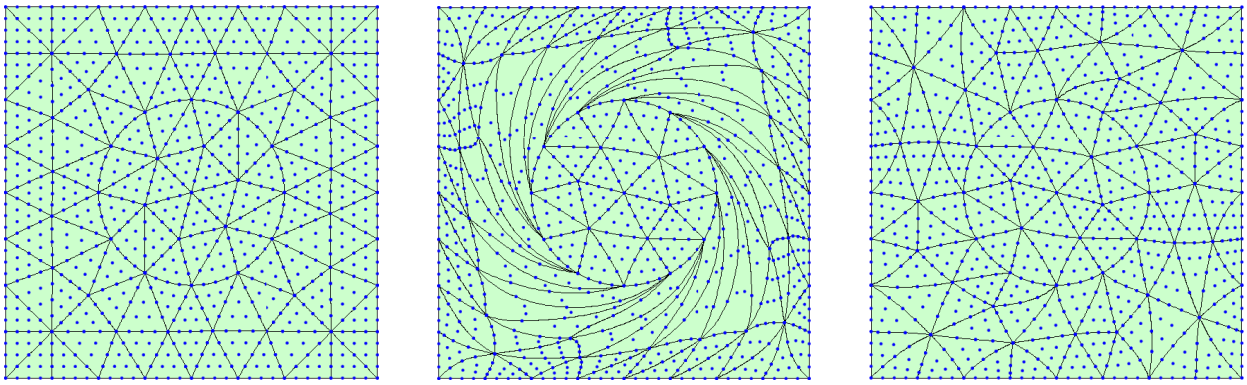


Figure 6.10: 2D rotation with flips: original mesh (*left*), quarter turn with smoothing only (*middle*), quarter turn with smoothing and flips (*right*).

Table 6.1: Mesh distortion of figure 6.10

	2D	
	max	agg
Original mesh	1.167	1.141
Smoothing only	5.327	7.876
With flips	1.197	1.179

6.4.2 Coarsening

Though this example does not deal with mesh deformation, we demonstrate how curved collapses can be combined with high-order node smoothing to coarsen a mesh (Figure 6.11). We begin with a rather fine 2D mesh with 690 elements of the square $[-1, 1] \times [-1, 1]$ with a circular hole at the origin with radius 0.5 (Figure 6.11, *top left*). We apply multiple passes of curved edge collapses to coarsen the mesh according to a uniform size function (Algorithm 2). We to also coarsen elements on the boundary with projection by interpolation in the L^2 error minimizing sense to maintain an accurate representation of the circular boundary with

fewer elements.

Algorithm 2: Application of Collapses

Input curved mesh M with edges $\{E_j\}$
Compute list of edge lengths L
Set ideal edge length $L_0 = 0.1$
Create priority queue $P = \{(E_k, L(E_k)), \forall k \ \frac{L_0}{L(E_k)} > 1.5 \text{ descending}\}$
while *true* **do**
  for *i in P* **do**
    Perform curved collapse on edge e_i and update the mesh M
    Remove all edges in collapse patch from P and renumber
  end
  if *no collapses done in sweep* **then**
    return M
  end
  Recompute L
end

6.4.3 Translation

We demonstrate the combination of all three curved element operations with mesh smoothing to maintain both element quality and uniform sizing. The 2D mesh is a circle-in-rectangle mesh of 314 elements, where the circle is centered at $(-1, 0)$ and has radius 0.5 inside the rectangle $[-2, 2] \times [-1, 1]$. At each step, we translate the nodes on the circle, perform high-order node smoothing while keeping the nodes on the circle fixed, apply all three types of local mesh operations according to Algorithm 4, and smooth again. Algorithm 4 applies sweeps of collapses and splits until convergence, and then applies sweeps of flips until convergence.

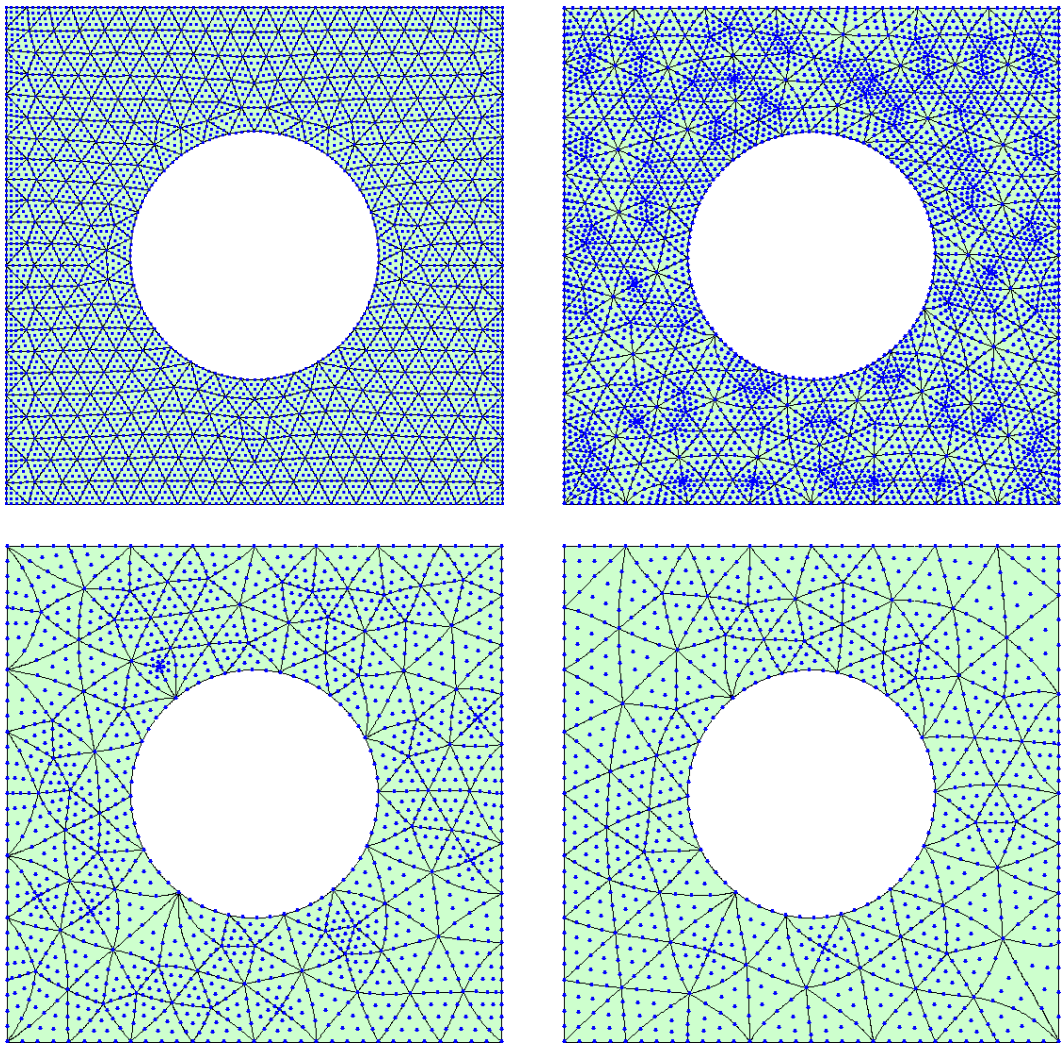


Figure 6.11: 2D coarsening with collapses: original mesh (*top left*), one sweep of collapses (*top right*), five sweeps of collapses (*bottom left*), final mesh (*bottom right*).

Algorithm 3: Application of Splitting

Input curved mesh M with edges $\{E_j\}$
Compute list of edge lengths L
Set ideal edge length as uniform $L_0 = \text{avg}(L)$
Create priority queue $P = \{(E_k, L(E_k)), \forall k \ \frac{L_0}{L(E_k)} > 1.5 \text{ descending}\}$
while *true* **do**
  for *i in P* **do**
    Identify elements K containing edge E_i
    for *j in K* **do**
      Identify all edges of element e_j in queue P to choose template
      Perform curved split on element e_j and update the mesh M
      Remove all edges in collapse patch from P and renumber
    end
  end
  if *no splits done in sweep* **then**
    return M
  end
  Recompute L
end

Algorithm 4: Application of all operations: flips, coarsening, and splitting

Input curved mesh M with elements $\{e_i\}$ and edges $\{e_j\}$
Perform sweeps of splits (Algorithm 3) with uniform ideal edge length
Perform sweeps of collapses (Algorithm 2) with uniform ideal edge length
Perform global smoothing of mesh M
Perform sweeps of flips (Algorithm 1)
Perform global smoothing of mesh M
return M

Translation cannot make it more than halfway across the prism with only node smoothing (Figure 6.12). With the addition of flips, we can make it across the entire domain but with some poorly sized elements. This can be remedied with the addition of collapses and splitting to maintain a uniform size, which allows us to practically maintain the original element sizes and qualities after translation across the entire domain (Table 6.2).

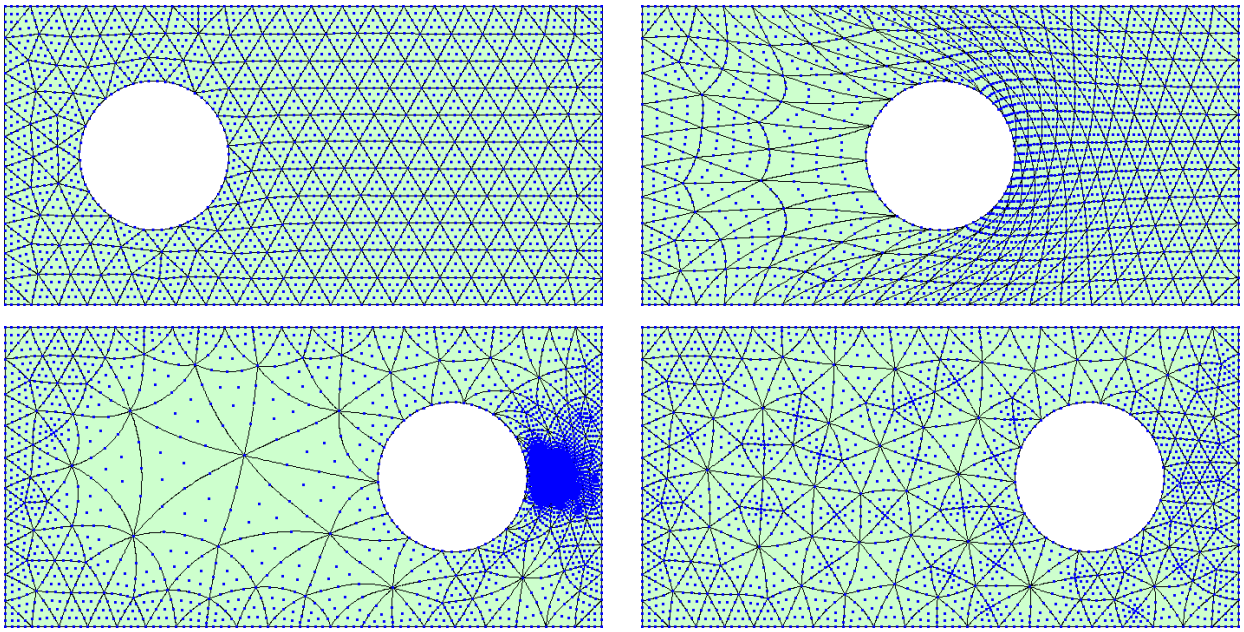


Figure 6.12: 2D translation: original mesh (*top left*), smoothing only (*top right*), smoothing with flips (*bottom left*), smoothing with flips, collapses, and splitting (*bottom right*).

Table 6.2: Mesh distortion of Figure 6.12

	2D	
	max	agg
Original mesh	1.1821	1.0473
Smoothing only	2.8378	2.3771
With flips	2.0643	1.2075
Flips, collapses, and splitting	1.1821	1.0498

Chapter 7

Conclusion and future research

7.1 Conclusion

We introduce a formulation of an optimization-based implicit shock tracking method and an associated solver that leverages the structure of the problem. The proposed optimization problem minimizes the DG residual in an *enriched* test and the distortion of the mesh, constrained by the standard DG residual (equal trial and test spaces). The enriched residual is a practical surrogate for the violation of the weak formulation of the conservation law; its magnitude serves as an error indicator for a DG solution. Therefore, penalizing the enriched DG residual promotes alignment of the element faces with discontinuities; otherwise, the DG solution would oscillate about the discontinuities and provide a poor approximation to the conservation law. The proposed solver for the constrained optimization problem over the DG solution and mesh coordinates is an SQP method that uses a Levenberg-Marquardt Hessian approximation and is globalized via a line search on the ℓ_1 merit function. The Hessian approximation is regularized with the stiffness matrix corresponding to a linear elliptic partial differential equation with coefficients inversely proportional to the local element size and adaptively chosen regularization parameter, which turns out to be significant for meshes containing elements of significantly different size.

For problems where the finite element space contains the exact solution, the SQP method exhibits Newton-like convergence to the exact solution, which was demonstrated via linear advection of a scalar through a constant advection field and supersonic, inviscid flow over a wedge. The framework was also shown to be effective in accurately resolving more complex flows with intricate shock structures using coarse, high-order meshes as demonstrated using the inviscid Burgers' equation (time-dependent), and transonic and supersonic flow over a NACA0012 airfoil. For these problems, the convergence of the solver is slower, but still drives the first-order optimality conditions to tight tolerances in a reasonable number of iterations.

We then extend the high-order implicit shock tracking framework for inviscid, steady conservation laws to the unsteady case by a method of lines discretization via a diagonally implicit Runge-Kutta (DIRK) method. In essence, we are "solving a steady problem at each

timestep", and as such inherit the desirable qualities from the steady case, namely a conservative and feature-aligned discretization at each timestep. We demonstrate our framework on a variety of one- and two- dimensional examples, and in particular show that our method is capable of preserving the design order of accuracy of our high-order temporal discretization for both the solution and the shock location, the latter of which is inaccessible to shock capturing methods because they do not maintain a perfect representation of discontinuities. The ability to construct good initial guesses for the optimization problem from information at the previous timestep is key to enable the SQP solver to converge rapidly.

This is in contrast to the steady case, where the initial mesh generation and corresponding guess for the solution has to be done independent of the shock location because we lack any *a priori* knowledge. Correspondingly, the steady case requires significant mesh deformation and edge collapses, which needs a more sophisticated line search procedure based on the ℓ_1 merit function to achieve convergence. The development of this method of lines approach gives us an additional tool to tackle unsteady problems in addition to the previously developed space-time approach. In general, the method of lines approach will be more practical and scale better as the size and difficulty of the problem increases. However, it is limited in that it cannot handle colliding shocks (triple points in space-time) without complex mesh operations and solution re-initialization. In these cases, the space-time approach is preferred due to its generality of tracking discontinuities in space-time, which naturally handles triple points.

7.2 Future research: shock tracking

Future work will develop iterative solvers and preconditioners for the SQP linear system in (3.26) to make the approach practical for large-scale problems. We also intend to further improve the robustness of the solver by incorporating pseudo-transient continuation to avoid the need to initialize from a $p = 0$ DG solution and use continuation on the polynomial degrees p (DG solution) and q (domain deformation). In [32], practical robustness measures are incorporated into the SQP solver that expand the capabilities of the high order implicit shock tracking framework to obtain results for more complex 2D problems such as the high-order resolution of steady, hypersonic (Mach 5) flow through a two-dimensional scramjet (Section 6.4.3 of [32]) and certain 3D problems.

Building on the method of lines, another direction would be the incorporation of topology changes, such as edge flips, refinement and coarsening as developed in Section 6 along with a corresponding high-order solution transfer procedure to the 2D unsteady examples. The issue is somewhat exacerbated by the fact that we are working with very coarse meshes to begin with. This difficultly manifests itself in our numerical experiments as seen in the unsteady case (Section 5.4), such as in the area immediately trailing the shock of the Shu-Osher problem as well as inside the circular region of the blast wave example as the shock expands radially; this loss of resolution is indicated by the gradual increase in the objective function over time (Figure 5.9). We emphasize that this is not a fundamental issue and the

method performs as expected under the constraint imposed by the fixed topology. However, the incorporation of adaptive mesh coarsening and refinement on curved meshes combined with L^2 error minimizing solution transfer between meshes will unlock the full potential of the method.

An entirely different direction would be the exploration of shock formation, most commonly illustrated by the time evolution of Burgers' equation from a smooth initial condition or the steepening of the waves trailing the shock in the Shu-Osher problem. Implicit shock tracking has the potential to cleanly form and subsequently track shocks due to its r-adaptive behavior and the optimization formulation that *implicitly* tracks shocks and simultaneously resolves the flow; this was shown by the authors of the MDG-ICE method for inviscid and viscous shocks in the space-time setting [9, 39]. In contrast, shock tracking approaches that handle shocks explicitly do not readily show potential to handle shock formation without significant specialization. A proof of concept using this implicit shock tracking framework has been demonstrated in Section 6.2.2 of [32], where a system governed by inviscid Burgers' equation takes a smooth initial condition and forms two distinct shock waves that eventually merge using a space-time formulation. However, analogous to the discussion in this thesis when contrasting the method of lines approach to the corresponding space-time formulation, when applicable, the former is preferable to the latter. The same reasoning applies to shock formation as well, and we believe our framework can be modified to do so.

7.3 Future work: high-order methods

Further work in the area of high-order numerical methods which would be useful in (but not limited to) shock tracking would be the extension of the local element operations for curved meshes introduced in Chapter 6 to 3D. To illustrate the general idea of this direction, the main difference here would be the introduction of many more possible "templates" for each of these element operations with the extra dimension. For example, for edge flips in 3D, some possibilities are shown by (but not limited to) the ones in Figure 7.1, because of the variety of many different possible initial topological configurations in 3D.

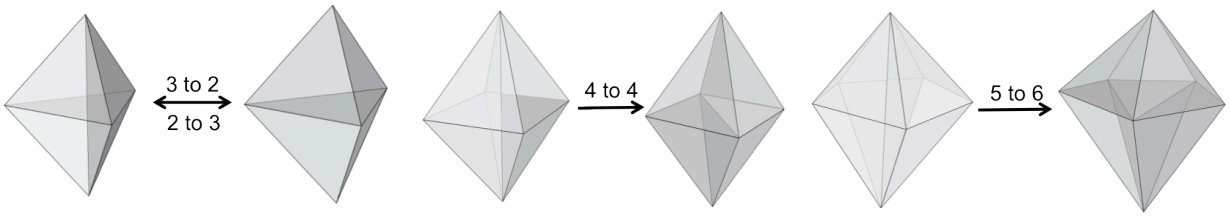


Figure 7.1: Edge/face swap templates in 3D.

One could proceed analogously with the procedure for 2D as shown in Figure 6.6 in 3D (Figure 7.2). In the same vein, the element subdivision templates in 2D (Figure 6.8) are

shown in 3D (Figure 7.3).

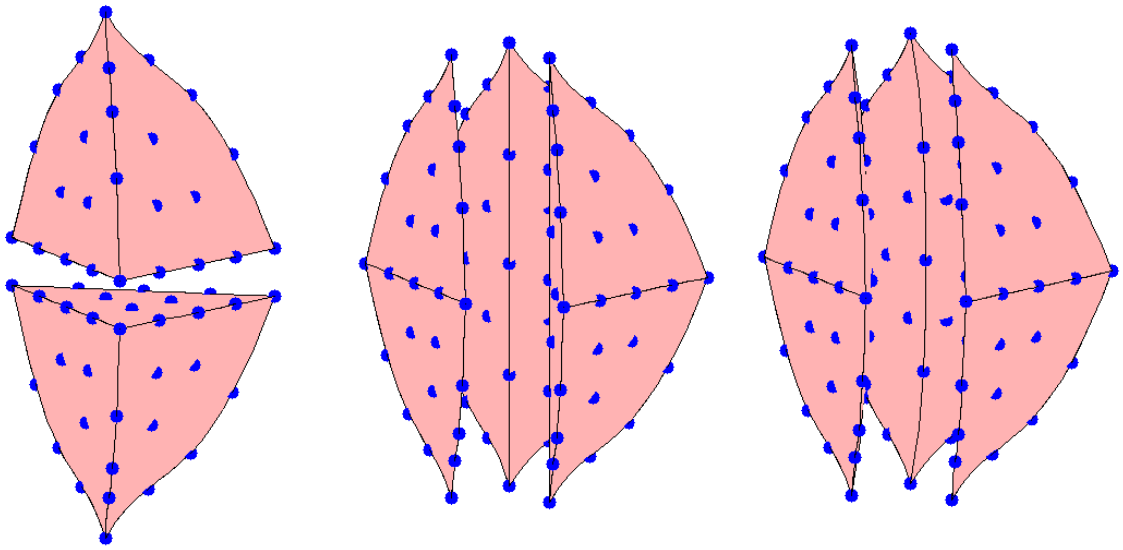


Figure 7.2: Face swaps in 3D: original elements *(left)*, initial straight-sided swap *(middle)*, curved swap *(right)*

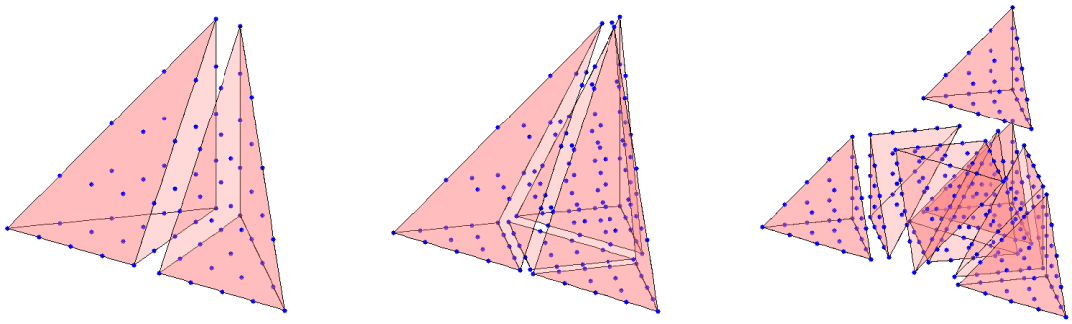


Figure 7.3: Element subdivision templates in 3D: $1 \to 2$ *(left)*, $1 \to 4$ *(middle)*, and $1 \to 8$ *(right)*

In order to incorporate these local element operations for curved meshes in high-order numerical simulation, it would need to be accompanied by a corresponding high-order solution transfer procedure. A conservative interpolation procedure between unstructured straight sided meshes that incorporate the construction of a "supermesh" as well as a corresponding L^2 minimizing error Galerkin projection was first introduced in ([18], [17]). Our case differs in two aspects: it is more difficult because we seek to do this for high-order meshes, but

simpler because our interpolation procedure would be localized (only for where the topology change occurs) rather than for the entire domain. The development of such a procedure would allow us to extend the numerical results in [72, 73] for the ALE based discontinuous Galerkin method for moving domains with large deformations to the high-order case along with a richer set of mesh operations.

The final direction we will mention is that of high-order unstructured mesh tangling and unsmoothing. This is practically taken for granted in this work, particularly in certain phases of the shock tracking framework and the execution of the high-order local element operations. However, the current procedure used in this work and others in the literature widely depend on the selection of a set of heuristically chosen parameters. This currently doesn't present any problems, but for more difficult 2D cases and 3D cases, robustness can be an issue. This is especially the case since our methods deal with large mesh deformations, such as in the initialization of our shock tracking procedure where mesh generation is done without any *a priori* knowledge of the shock, which can result in mesh edges and faces initialized that are nearly perpendicular to the final unknown shock location. We believe there is ample potential to be unlocked in the Winslow equations, building on the work of [21] for high-order unstructured mesh generation to mesh smoothing and untangling, particular when it comes to fast and robust solvers.

Bibliography

[1] https://blog.pointwise.com/2019/08/22/high-order-curved-meshes-for-more-accurate-cfd-solutions/. Accessed: 2020-09-16.

[2] M. J. Baines, S. J. Leary, and M. E. Hubbard. "Multidimensional Least Squares Fluctuation Distribution Schemes with Adaptive Mesh Movement for Steady Hyperbolic Equations". In: *SIAM Journal on Scientific Computing* 23.5 (2002), pp. 1485–1502. DOI: https://doi.org/10.1137/S1064827500370202.

[3] C. E. Baumann and J. T. Oden. "A Discontinuous *hp* Finite Element Method for the Euler and Navier-Stokes Equations". In: *International Journal for Numerical Methods in Fluids* 31.1 (1999). Tenth International Conference on Finite Elements in Fluids (Tucson, AZ, Jan 1998), pp. 79–95. DOI: https://doi.org/10.1002/(SICI)1097-0363(19990915)31:1<79::AID-FLD956>3.0.CO;2-C.

[4] J. B. Bell, G. R. Shubin, and J. M. Solomon. "Fully Implicit Shock Tracking". In: *Journal of Computational Physics* 48.2 (1982), pp. 223–245. DOI: https://doi.org/10.1016/0021-9991(82)90048-1.

[5] P. T. Boggs and J. W. Tolle. "Sequential Quadratic Programming for Large-Scale Nonlinear Optimization". In: *Journal of Computational and Applied Mathematics* 124.1-2 (2000). Numerical Analysis 2000. Vol. IV: Optimization and Nonlinear Equations, pp. 123–137. DOI: https://doi.org/10.1016/S0377-0427(00)00429-5.

[6] A. Burbeau, P. Sagaut, and C.-H. Bruneau. "A Problem-Independent Limiter for High-Order Runge-Kutta Discontinuous Galerkin Methods". In: *Journal of Computational Physics* 169.1 (2001), pp. 111–150. DOI: https://doi.org/10.1006/jcph.2001.6718.

[7] B. Cockburn and C.-W. Shu. "Runge-Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems". In: *Journal of Scientific Computing* 16.3 (2001), pp. 173–261. DOI: https://doi.org/10.1023/A:1012873910884.

[8] A. Corrigan, A. Kercher, and D. Kessler. "A Moving Discontinuous Galerkin Finite Element Method for Flows with Interfaces". In: *International Journal for Numerical Methods in Fluids* 89.9 (2019), pp. 362–406. DOI: https://doi.org/10.1002/fld.4697.

[9] A. Corrigan, A. Kercher, and D. Kessler. "The Moving Discontinuous Galerkin Method with Interface Condition Enforcement for Unsteady Three-Dimensional Flows". In: *AIAA Scitech 2019 Forum*. AIAA Paper 2019-0642. Jan. 2019. DOI: https://doi.org/10.2514/6.2019-0642.

[10] A. Corrigan et al. "Convergence of the Moving Discontinuous Galerkin Method with Interface Condition Enforcement in the Presence of an Attached Curved Shock". In: *AIAA Aviation 2019 Forum*. AIAA Paper 2019-3027. June 2019. DOI: https://doi.org/10.2514/6.2019-3207.

[11] H. L. De Cougny and M. S. Shephard. "Parallel Refinement and Coarsening of Tetrahedral Meshes". In: *International Journal for Numerical Methods in Engineering* 46.7 (1999), pp. 1101–1125. DOI: https://doi.org/10.1002/(SICI)1097-0207(19991110)46:7<1101::AID-NME741>3.0.CO;2-E.

[12] J. A. De Loera, J. Rambau, and F. Santos. *Triangulations: Structures for Algorithms and Applications*. Vol. 25. Algorithms and Computation in Mathematics. Springer Berlin, Heidelberg, 2010. DOI: https://doi.org/10.1007/978-3-642-12971-1.

[13] L. Demkowicz. *Computing with hp-ADAPTIVE FINITE ELEMENTS: Volume 1 One and Two Dimensional Elliptic and Maxwell Problems*. Chapman and Hall/CRC, 2006. DOI: https://doi.org/10.1201/9781420011685.

[14] A. Dervieux et al. "About Theoretical and Practical Impact of Mesh Adaptation on Approximation of Functions and PDE Solutions". In: *International Journal for Numerical Methods in Fluids* 43.5 (2003). ECCOMAS Computational Fluid Dynamics Conference, pp. 507–516. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.503.

[15] V. Dobrev et al. "The Target-Matrix Optimization Paradigm for High-Order Meshes". In: *SIAM Journal on Scientific Computing* 41.1 (2019), B50–B68. DOI: https://doi.org/10.1137/18M1167206.

[16] J. M. Escobar et al. "Simultaneous Untangling and Smoothing of Tetrahedral Meshes". In: *Computer Methods in Applied Mechanics and Engineering* 192.25 (2003), pp. 2775–2787. DOI: https://doi.org/10.1016/S0045-7825(03)00299-8.

[17] P. E. Farrell and J. R. Maddison. "Conservative Interpolation between Volume Meshes by Local Galerkin Projection". In: *Computer Methods in Applied Mechanics and Engineering* 200.1 (2011), pp. 89–100. DOI: https://doi.org/10.1016/j.cma.2010.07.015.

[18] P.E. Farrell et al. "Conservative Interpolation between Unstructured Meshes via Supermesh Construction". In: *Computer Methods in Applied Mechanics and Engineering* 198.33 (2009), pp. 2632–2642. DOI: https://doi.org/10.1016/j.cma.2009.03.004.

[19] R. Feuillet, A. Loseille, and F. Alauzet. "P^2 Mesh Optimization Operators". In: *27th International Meshing Roundtable*. Ed. by X. Roca and A. Loseille. Springer, 2018, pp. 3–21. DOI: https://doi.org/10.1007/978-3-030-13992-6_1.

[20] K. J. Fidkowski. "Output Error Estimation Strategies for Discontinuous Galerkin Discretizations of Unsteady Convection-Dominated Flows". In: *International Journal for Numerical Methods in Engineering* 88.12 (2011), pp. 1297–1322. DOI: https://doi.org/10.1002/nme.3224.

[21] M. Fortunato and P.-O. Persson. "High-Order Unstructured Curved Mesh Generation Using the Winslow Equations". In: *Journal of Computational Physics* 307 (2016), pp. 1–14. DOI: https://doi.org/10.1016/j.jcp.2015.11.020.

[22] L. A. Freitag and C. Ollivier-Gooch. "Tetrahedral Mesh Improvement Using Swapping and Smoothing". In: *International Journal for Numerical Methods in Engineering* 40.21 (1997), pp. 3979–4002. DOI: https://doi.org/10.1002/(SICI)1097-0207(19971115)40:21<3979::AID-NME251>3.0.CO;2-9.

[23] B. Froehle and P.-O. Persson. "Nonlinear Elasticity for Mesh Deformation with High-Order Discontinuous Galerkin Methods for the Navier-Stokes Equations on Deforming Domains". In: *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*. Ed. by R.M. Kirby, M. Berzins, and J.S. Hesthaven. Springer, 2015, pp. 73–85. DOI: https://doi.org/10.1007/978-3-319-19800-2_5.

[24] A. Gargallo-Peiró et al. "A Distortion Measure to Validate and Generate Curved High-Order Meshes on CAD Surfaces with Independence of Parameterization". In: *International Journal for Numerical Methods in Engineering* 106.13 (2016), pp. 1100–1130. DOI: https://doi.org/10.1002/nme.5162.

[25] A. Gargallo-Peiró et al. "Defining Quality Measures for Validation and Generation of High-Order Tetrahedral Meshes". In: *Proceedings of the 22nd International Meshing Roundtable*. Ed. by J. Sarrate and M. Staten. Springer, 2014, pp. 109–126. DOI: https://doi.org/10.1007/978-3-319-02335-9_7.

[26] A. Gargallo-Peiró et al. "Optimization of a Regularized Distortion Measure to Generate Curved High-Order Unstructured Tetrahedral Meshes". In: *International Journal for Numerical Methods in Engineering* 103.5 (2015), pp. 342–363. DOI: https://doi.org/10.1002/nme.4888.

[27] C. Geuzaine and J.-F. Remacle. "Gmsh: A 3-D Finite Element Mesh Generator with Built-In Pre-and Post-Processing Facilities". In: *International Journal for Numerical Methods in Engineering* 79.11 (2009), pp. 1309–1331. DOI: https://doi.org/10.1002/nme.2579.

[28] J. Glimm et al. "Conservative Front Tracking with Improved Accuracy". In: *SIAM Journal on Numerical Analysis* 41.5 (2003), pp. 1926–1947. DOI: https://doi.org/10.1137/S0036142901388627.

[29] A. Harten and J. M. Hyman. "Self Adjusting Grid Methods for One-Dimensional Hyperbolic Conservation Laws". In: *Journal of Computational Physics* 50.2 (1983), pp. 235–269. DOI: https://doi.org/10.1016/0021-9991(83)90066-9.

[30] A. Harten et al. "Uniformly High-Order Accurate Essentially Non-Oscillatory Schemes. III". In: *Journal of Computational Physics* 71.2 (1987), pp. 231–303. DOI: https://doi.org/10.1016/0021-9991(87)90031-3.

[31] J. Hesthaven and T. Warburton. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Texts in Applied Mathematics. Springer New York, NY, 2008. DOI: https://doi.org/10.1007/978-0-387-72067-8.

[32] T. Huang and M. J. Zahr. "A Robust, High-Order Implicit Shock Tracking Method for Simulation of Complex, High-Speed Flows". In: *Journal of Computational Physics* 454 (2022), p. 110981. DOI: https://doi.org/10.1016/j.jcp.2022.110981.

[33] Y. Huang. "A Boundary Integral Method for Modeling Axisymmetric Flow Around a Rising Bubble in a Vertical Tube and Accurate Numerical Evaluation of Orthogonal Polynomials". PhD thesis. University of California, Berkeley, 2020. URL: https://escholarship.org/uc/item/6vz710m8.

[34] G. Irving, J. Teran, and R. Fedkiw. "Invertible Finite Elements for Robust Simulation of Large Deformation". In: *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2004, pp. 131–140. DOI: https://doi.org/10.1145/1028523.1028541.

[35] G.-S. Jiang and C.-W. Shu. "Efficient Implementation of Weighted ENO Schemes". In: *Journal of Computational Physics* 126.1 (1996), pp. 202–228. DOI: https://doi.org/10.1006/jcph.1996.0130.

[36] E. Johnsen et al. "Assessment of High-Resolution Methods for Numerical Simulations of Compressible Turbulence with Shock Waves". In: *Journal of Computational Physics* 229.4 (2010), pp. 1213–1237. DOI: https://doi.org/10.1016/j.jcp.2009.10.028.

[37] S. L. Karman et al. "High-Order Mesh Curving Using WCN Mesh Optimization". In: *46th AIAA Fluid Dynamics Conference*. AIAA Paper 2016-3178. June 2016. DOI: https://doi.org/10.2514/6.2016-3178.

[38] C. T. Kelley and D. E. Keyes. "Convergence Analysis of Pseudo-Transient Continuation". In: *SIAM Journal on Numerical Analysis* 35.2 (1998), pp. 508–523. DOI: https://doi.org/10.1137/S0036142996304796.

[39] A. D. Kercher, A. Corrigan, and D. A. Kessler. "The Moving Discontinuous Galerkin Finite Element Method with Interface Condition Enforcement for Compressible Viscous Flows". In: *International Journal for Numerical Methods in Fluids* 93.5 (2021), pp. 1490–1519. DOI: https://doi.org/10.1002/fld.4939.

[40] B. M. Klingner and J. R. Shewchuk. "Aggressive Tetrahedral Mesh Improvement". In: *Proceedings of the 16th International Meshing Roundtable*. Ed. by M.L. Brewer and David Marcum. Springer, 2008, pp. 3–23. DOI: https://doi.org/10.1007/978-3-540-75103-8_1.

[41] P. M. Knupp. "Algebraic Mesh Quality Metrics". In: *SIAM Journal on Scientific Computing* 23.1 (2001), pp. 193–218. DOI: https://doi.org/10.1137/S1064827500371499.

[42] F. Ledoux and K. Lewis. *Proceedings of the 24th International Meshing Roundtable*. Vol. 124. Elsevier, 2005.

[43] A. Liu and B. Joe. "On the Shape of Tetrahedra from Bisection". In: *Mathematics of Computation* 63 (1994), pp. 141–154. DOI: https://doi.org/10.2307/2153566.

[44] A. Liu and B. Joe. "Relationship Between Tetrahedron Shape Measures". In: *BIT Numerical Mathematics* 34.2 (1994), pp. 268–287. DOI: https://doi.org/10.1007/BF01955874.

[45] X.-D. Liu, S. J. Osher, and T. Chan. "Weighted Essentially Non-Oscillatory Schemes". In: *Journal of Computational Physics* 115.1 (1994), pp. 200–212. DOI: https://doi.org/10.1006/jcph.1994.1187.

[46] R. Löhner. *Applied Computational Fluid Dynamics Techniques: An Introduction Based on Finite Element Methods*. John Wiley & Sons, 2008. DOI: https://doi.org/10.1002/9780470989746.

[47] A. Majda. *Compressible Fluid Flow and Systems of Conservation laws in Several Space Variables*. Vol. 53. Applied Mathematical Sciences. Springer New York, NY, 1984. DOI: https://doi.org/10.1007/978-1-4612-1116-7.

[48] G. Moretti. "Thirty-Six Years of Shock Fitting". In: *Computers & Fluids* 31.4-7 (2002), pp. 719–723. DOI: https://doi.org/10.1016/S0045-7930(01)00072-X.

[49] D. Moxey et al. "A Thermo-elastic Analogy for High-order Curvilinear Meshing with Control of Mesh Validity and Quality". In: *Procedia engineering* 82 (2014). 23rd International Meshing Roundtable (IMR23), pp. 127–135. DOI: https://doi.org/10.1016/j.proeng.2014.10.378.

[50] T. Munson. "Mesh Shape-Quality Optimization Using the Inverse Mean-Ratio Metric". In: *Mathematical Programming* 110.3 (2007), pp. 561–590. DOI: https://doi.org/10.1007/s10107-006-0014-3.

[51] A. Ni. "Numerical Differentiation of Stationary Measures of Chaos". PhD thesis. University of California, Berkeley, 2021. URL: https://escholarship.org/uc/item/9pm5k59h.

[52] E. J. Nielsen and W. K. Anderson. "Recent Improvements in Aerodynamic Design Optimization on Unstructured Meshes". In: *AIAA journal* 40.6 (2002), pp. 1155–1163. DOI: https://doi.org/10.2514/2.1765.

[53] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2006. DOI: https://doi.org/10.1007/978-0-387-40065-5.

[54] P.-O. Persson, J. Bonet, and J. Peraire. "Discontinuous Galerkin Solution of the Navier–Stokes Equations on Deformable Domains". In: *Computer Methods in Applied Mechanics and Engineering* 198.17-20 (2009), pp. 1585–1595. DOI: https://doi.org/10.1016/j.cma.2009.01.012.

[55] P.-O. Persson and J. Peraire. "Curved Mesh Generation and Mesh Refinement using Lagrangian Solid Mechanics". In: *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*. AIAA Paper 2009-949. Jan. 2009. DOI: https://doi.org/10.2514/6.2009-949.

[56] P.-O. Persson and J. Peraire. "Sub-Cell Shock Capturing for Discontinuous Galerkin Methods". In: *44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada*. AIAA Paper 2006-112. Jan. 2006. DOI: https://doi.org/10.2514/6.2006-112.

[57] P.-O. Persson and G. Strang. "A Simple Mesh Generator in MATLAB". In: *SIAM Review* 46.2 (2004), pp. 329–345. DOI: https://doi.org/10.1137/S0036144503429121.

[58] P. S. Rawat and X. Zhong. "On High-Order Shock-Fitting and Front-Tracking Schemes for Numerical Simulation of Shock-Disturbance Interactions". In: *Journal of Computational Physics* 229.19 (2010), pp. 6744–6780. DOI: https://doi.org/10.1016/j.jcp.2010.05.021.

[59] J.-F. Remacle et al. "Optimizing the Geometrical Accuracy of 2D Curvilinear Meshes". In: *Procedia Engineering* 82 (2014). 23rd International Meshing Roundtable (IMR23), pp. 228–239. DOI: https://doi.org/10.1016/j.proeng.2014.10.386.

[60] X. Roca, A. Gargallo-Peiró, and J. Sarrate. "Defining Quality Measures for High-Order Planar Triangles and Curved Mesh Generation". In: *Proceedings of the 20th International Meshing Roundtable*. Ed. by W.R. Quadros. Springer, 2011, pp. 365–383. DOI: https://doi.org/10.1007/978-3-642-24734-7_20.

[61] P. L. Roe. "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes". In: *Journal of Computational Physics* 43.2 (1981), pp. 357–372. DOI: https://doi.org/10.1016/0021-9991(81)90128-5.

[62] L. I. Sedov. *Similarity and Dimensional Methods in Mechanics*. Academic Press, 1959. DOI: https://doi.org/10.1016/C2013-0-08173-X.

[63] A. Shi, M. J. Zahr, and P.-O. Persson. "Implicit Shock Tracking for Unsteady Flows by the Method of Lines". In: *Journal of Computational Physics* 454 (2022), p. 110906. DOI: https://doi.org/10.1016/j.jcp.2021.110906.

[64] C.-W. Shu and S. Osher. "Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes, II". In: *Journal of Computational Physics* 83.1 (1989), pp. 32–78. DOI: https://doi.org/10.1016/0021-9991(89)90222-2.

[65] G. R. Shubin, A. B. Stephens, and H. M. Glaz. "Steady Shock Tracking and Newton's Method Applied to One-Dimensional Duct Flow". In: *Journal of Computational Physics* 39.2 (1981), pp. 364–374. DOI: https://doi.org/10.1016/0021-9991(81)90157-1.

[66] G. R. Shubin et al. "Steady Shock Tracking, Newton's Method, and the Supersonic Blunt Body Problem". In: *SIAM Journal on Scientific and Statistical Computing* 3.2 (1982), pp. 127–144. DOI: https://doi.org/10.1137/0903009.

[67] E.F. Toro, M. Spruce, and W. Speares. "Restoration of the Contact Surface in the HLL-Riemann Solver". In: *Shock Waves* 4.1 (1994), pp. 25–34. DOI: https://doi.org/10.1007/BF01414629.

[68] T. Toulorge et al. "Robust Untangling of Curvilinear Meshes". In: *Journal of Computational Physics* 254 (2013), pp. 8–26. DOI: https://doi.org/10.1016/j.jcp.2013.07.022.

[69] J.-Y. Trepanier et al. "A Conservative Shock Fitting Method on Unstructured Grids". In: *Journal of Computational Physics* 126.2 (1996), pp. 421–433. DOI: https://doi.org/10.1006/jcph.1996.0147.

[70] M. Turner. "High-Order Mesh Generation for CFD Solvers". PhD thesis. Imperial College London, 2017. DOI: https://doi.org/10.25560/57956.

[71] J. Van Rosendale. "Floating Shock Fitting via Lagrangian Adaptive Meshes". In: *12th Computational Fluid Dynamics Conference*. AIAA-95-1721-CP. June 1995. DOI: https://doi.org/10.2514/6.1995-1721.

[72] L. Wang. "Discontinuous Galerkin Methods on Moving Domains with Large Deformations". PhD thesis. University of California, Berkeley, 2015. URL: https://escholarship.org/uc/item/84j5b9j8.

[73] L. Wang and P.-O. Persson. "A High-Order Discontinuous Galerkin Method with Unstructured Space-Time Meshes for Two-Dimensional Compressible Flows on Domains with Large Deformations". In: *Computers & Fluids* 118 (2015), pp. 53–68. DOI: https://doi.org/10.1016/j.compfluid.2015.05.026.

[74] Z. J. Wang. "A Perspective on High-order Methods in Computational Fluid Dynamics". In: *Science China Physics, Mechanics & Astronomy* 59.1 (2016), pp. 614–701. DOI: https://doi.org/10.1007/s11433-015-5706-3.

[75] Z. J. Wang et al. "High-Order CFD Methods: Current Status and Perspective". In: *International Journal for Numerical Methods in Fluids* 72.8 (2013), pp. 811–845. DOI: https://doi.org/10.1002/fld.3767.

[76] Z. Q. Xie et al. "The Generation of Arbitrary Order Curved Meshes for 3D Finite Element Analysis". In: *Computational Mechanics* 51.3 (2013), pp. 361–374. DOI: https://doi.org/10.1007/s00466-012-0736-4.

[77] J. Yu and J. Hesthaven. "A Study of Several Artificial Viscosity Models Within the Discontinuous Galerkin Framework". In: *Communications in Computational Physics* 27.5 (2020), pp. 1309–1343. DOI: https://doi.org/10.4208/cicp.OA-2019-0118.

[78] M. J. Zahr and P.-O. Persson. "An r-Adaptive, High-Order Discontinuous Galerkin Method for Flows with Attached Shocks". In: *AIAA Scitech 2020 Forum*. AIAA Paper 2020-0537. Jan. 2020. DOI: https://doi.org/10.2514/6.2020-0537.

[79] M. J. Zahr and P.-O. Persson. "An Optimization-Based Approach for High-Order Accurate Discretization of Conservation Laws with Discontinuous Solutions". In: *Journal of Computational Physics* 365 (2018), pp. 105–134. DOI: https://doi.org/10.1016/j.jcp.2018.03.029.

[80] M. J. Zahr and J. M. Powers. "High-Order Resolution of Multidimensional Compressible Reactive Flow Using Implicit Shock Tracking". In: *AIAA Journal* 59.1 (2021), pp. 150–164. DOI: https://doi.org/10.2514/1.J059655.

[81] M. J. Zahr, A. Shi, and P.-O. Persson. "Implicit Shock Tracking Using an Optimization-Based High-Order Discontinuous Galerkin Method". In: *Journal of Computational Physics* 410 (2020), p. 109385. DOI: https://doi.org/10.1016/j.jcp.2020.109385.