

Numerical Methods for First and Second Order Fully Nonlinear Partial Differential Equations

Présentée le 14 juillet 2021

Faculté des sciences de base
Groupe Picasso
Programme doctoral en mathématiques

pour l'obtention du grade de Docteur ès Sciences

par

Dimitrios GOURZOULIDIS

Acceptée sur proposition du jury

Prof. T. Mountford, président du jury
Prof. M. Picasso, Prof. A. Caboussat, directeurs de thèse
Prof. R. Glowinski, rapporteur
Prof. O. Lakkis, rapporteur
Prof. A. Abdulle, rapporteur

Acknowledgements

First of all, I would like to express my gratitude and appreciation to my supervisors Prof. Alexandre Caboussat and Prof. Marco Picasso for their guidance and invaluable support. I am sincerely indebted to them for giving me the opportunity to work together. Special thanks to Prof. Caboussat for introducing me to the field of fully nonlinear equations, and particularly for leading me through all the stages of my PhD journey. I equally owe a great deal of appreciation to Prof. Picasso for the significant counselling and the useful guidance.

Aside from my supervisors, I would like to express my gratitude to the rest of my thesis committee: Prof. Assyr Abdulle, Prof. Roland Glowinski, and Prof. Omar Lakkis, for their insightful comments during the thesis defense, as well as to the thesis jury president, Prof. Thomas Mountford. I would also like to express my appreciation to Prof. Bernard Dacorogna for all the discussions and helpful suggestions on the orthogonal maps problem.

My gratitude extends to Geneva School of Business Administration (Haute école de gestion de Genève), University of Applied Sciences and Arts Western Switzerland (HES-SO) and the Swiss National Science Foundation (SNF grant #165785, “Variational and adaptive methods for Monge-Ampère-type equations”) for the funding opportunity to undertake my studies at EPFL as well as to the Institute of Mathematics at EPFL for hosting me.

I would like to thank my officemate Arwa for the nice time we spent working, as well as the squad, Samuel, Emile, Leo, and Paride, for building and sustaining such a positive atmosphere. Special thanks to Samuel for the interesting scientific and non-scientific discussions we had in the past years and to Panos for all the talks and experiences we shared during lunch and coffee breaks.

My thanks extend to Soumaya for supporting me throughout the thesis, and to Rika for always being around. I truly appreciate the time we spent together. Lastly, above ground, I am profoundly thankful to my parents Eleni & Anastasios, and my sister Menia who have been wonderfully supportive all those years.

Μαμά, μπαμπά, Μένια, σας ευχαριστώ από καρδιάς για όλα όσα έχετε κάνει για μένα όλα αυτά τα χρόνια.

Lausanne, March 14, 2021

D. G.

Abstract

This thesis focuses on the numerical analysis of partial differential equations (PDEs) with an emphasis on first and second-order fully nonlinear PDEs. The main goal is the design of numerical methods to solve a variety of equations such as orthogonal maps, the prescribed Jacobian equation and inequality, the elliptic and parabolic Monge-Ampère equations.

For orthogonal map we develop an *operator-splitting/finite element* approach for the numerical solution of the Dirichlet problem. This approach is built on the variational principle, the introduction of an associated flow problem, and a time-stepping splitting algorithm. Moreover, we propose an extension of this method with an *anisotropic mesh adaptation algorithm*. This extension allows us to track singularities of the solution's gradient more accurately. Various numerical experiments demonstrate the accuracy and the robustness of the proposed method for both constant and adaptive mesh.

For the prescribed Jacobian equation and the three-dimensional Monge-Ampère equation, we consider a *least-squares/relaxation finite element method* for the numerical solution of the Dirichlet problems. We then introduce a relaxation algorithm that splits the least-square problem, which stems from a reformulation of the original equations, into local nonlinear and variational problems. We develop dedicated solvers for the algebraic problems based on Newton method and we solve the differential problems using mixed low-order finite element method. Overall the least squares approach exhibits appropriate convergence orders in $L^2(\Omega)$ and $H^1(\Omega)$ error norms for various numerical tests.

We also design a *second-order time integration method* for the approximation of a parabolic two-dimensional Monge-Ampère equation. The space discretization of this method is based on low-order finite elements, and the time discretization is achieved by the implicit Crank-Nicolson type scheme. We verify the efficiency of the proposed method on time-dependent and stationary problems. The results of numerical experiments show that the method achieves nearly optimal orders for the $L^2(\Omega)$ and $H^1(\Omega)$ error norms when smooth solutions are approximated.

Finally, we present an adaptive mesh refinement algorithm for the elliptic Monge-Ampère equation based on the residual error estimate. The robustness of the proposed algorithm is verified using various test cases and two different solvers which are inspired from the two previous proposed numerical methods.

Key words. orthogonal maps, origami, operator splitting, finite element methods, adaptive mesh refinement, prescribed Jacobian equation, Monge-Ampère equation, least-squares method, Newton methods, parabolic Monge-Ampère equation.

Résumé

Cette thèse se consacre à l'analyse numérique des équations aux dérivées partielles (EDP) en mettant l'accent sur les EDP entièrement non linéaires du premier et du second ordre. Le but principal est la conception de méthodes numériques pour résoudre une variété d'équations telles que les représentations orthogonales, l'équation Jacobienne prescrite, les équations elliptiques et paraboliques de Monge-Ampère.

Pour les représentations orthogonales, nous développons une solution numérique du problème de Dirichlet basée sur le principe variationnel, l'introduction d'un problème de flux associé, et d'un algorithme de segmentation en étapes temporelles. De plus, nous proposons une extension de cette méthode avec un algorithme d'adaptation de maillage anisotrope. Cette extension nous permet de suivre plus précisément les singularités du gradient de la solution. Diverses analyses numériques démontrent la précision et la robustesse de la méthode proposée, tant pour le maillage constant que pour le maillage adaptatif.

Pour l'équation Jacobienne prescrite et l'équation tridimensionnelle de Monge-Ampère, nous considérons une méthode des moindres carrés/relaxation par éléments finis pour la solution numérique du problème de Dirichlet. Nous introduisons un algorithme de relaxation qui divise le problème des moindres carrés en deux : un problème local non linéaire et un problème variationnel. Les solveurs que nous proposons sont basés sur la méthode de Newton et la méthode des éléments finis d'ordre inférieur mixte. Globalement, l'approche des moindres carrés possède des ordres de convergence appropriés sous forme de normes d'erreur $L^2(\Omega)$ et $H^1(\Omega)$ pour divers tests numériques.

Nous développons également une méthode d'intégration temporelle du second ordre pour l'approximation d'une équation parabolique bidimensionnelle de Monge-Ampère. La discrétisation spatiale de cette méthode est basée sur des éléments finis d'ordre inférieur, et la discrétisation temporelle est obtenue par un schéma implicite de type Crank-Nicolson.

Nous présenterons enfin un algorithme de raffinement de maillage adaptatif pour l'équation elliptique de Monge-Ampère, basé sur l'estimation de l'erreur résiduelle. La robustesse des algorithmes proposés est vérifiée à l'aide de différents tests et de deux solveurs qui s'inspirent des deux méthodes numériques proposées précédemment.

Mots clés. représentations orthogonales, origami, méthodes des éléments finis, raffinement du maillage adaptatif, équation Jacobienne prescrite, équation de Monge-Ampère, méthode des moindres carrés, méthodes de Newton, équation parabolique de Monge-Ampère.

Contents

Acknowledgements	i
Abstract (English/Français)	iii
Introduction	1
1 Numerical Approximation of Orthogonal Maps	11
1.1 Mathematical Model	11
1.1.1 Initial problem	11
1.1.2 Determinant property and angle condition	12
1.1.3 Rigid maps problem	13
1.2 Numerical Algorithms	14
1.2.1 Regularization and Penalization	14
1.2.2 First Order Optimality Conditions	16
1.2.3 Flow Problem and Operator-Splitting Algorithm	17
1.2.4 Local Optimization Problems	18
1.2.5 Variational Problems	19
1.3 Finite Element Discretization	20
1.3.1 Generalities	20
1.3.2 Discretization of the flow problem	21
1.3.3 Solution of the discrete local optimization problems	21
1.3.4 Solution of discrete linear variational problems	22
1.4 Numerical Experiments	22
1.4.1 Smooth validation example	23
1.4.2 Simple folding and mesh dependency	24
1.4.3 Double diagonal folding	28
1.4.4 A non-smooth example with a point singularity	33
1.4.5 Curved boundaries	35
1.5 The Dirichlet problem with homogeneous boundary conditions	36
1.6 An anisotropic adaptive algorithm	39
1.6.1 General algorithm	42
1.7 Numerical experiments	43
1.7.1 Single fold	43
1.7.2 Double diagonal folding	49
	vii

1.7.3	Comparison with a standard adaptive approach	50
1.7.4	Non-smooth example with a point singularity	52
1.8	A decomposition approach for the homogeneous Dirichlet Problem	58
2	Numerical Approximation of the Prescribed Jacobian Equation/Inequality	67
2.1	Problem Formulation	67
2.2	Numerical Algorithm	68
2.2.1	Least squares method	68
2.2.2	Relaxation algorithm	69
2.2.3	Numerical solution of the local nonlinearly constrained minimization problems	70
2.2.4	Numerical solution of the linear variational problems	74
2.3	Finite Element Approximation	75
2.4	Numerical Experiments	77
2.4.1	Identity map	80
2.4.2	Identity map with periodic perturbation	82
2.4.3	Smooth solution with radial symmetric right-hand side	84
2.4.4	Smooth radial symmetric solution with non-smooth gradient	87
2.4.5	Identity map on non-convex domains	91
2.4.6	Smooth solution with radial symmetric right-hand side on non-convex domains	92
2.4.7	Nonsmooth right-hand side with a jump	97
2.4.8	Nonsmooth right-hand side with a Dirac delta function	102
3	Numerical Approximation of Monge-Ampère Equation	117
3.1	Mathematical Formulation and Least-Squares Approach	117
3.2	Relaxation Algorithm	118
3.3	Numerical Approximation of the Local Nonlinear Problems	119
3.3.1	Explicit Formulation of the Local Nonlinear Problems	119
3.3.2	A Reduced Newton Method	120
3.3.3	A Runge-Kutta Method for the Dynamical Flow Problem	120
3.4	Numerical Solution of the Linear Variational Problems	123
3.5	Mixed Finite Element Approximation	125
3.5.1	Finite Element Spaces	125
3.5.2	Finite Element Approximation Tools of the Monge-Ampère Equation	125
3.5.3	Discrete Formulation of the Least-Squares Method	127
3.5.4	A Discrete Relaxation Algorithm	127
3.5.5	Finite Element Approximation of the Local Nonlinear Problems	128
3.5.6	Finite Element Approximation of the Linear Variational Problems	128
3.5.7	Discrete Conjugate Gradient Algorithm	129
3.6	Numerical Results	131
3.6.1	Polynomial Examples	132
3.6.2	A Smooth Exponential Example	134

3.6.3	Non-Smooth Test Problems	135
3.6.4	Curved Boundaries and Non Convex Domains	139
3.7	An Alternative Discretization Method based on $\mathbb{Q}_1$ Finite Elements	143
4	Numerical Approximation of a 2D Parabolic Monge-Ampère Equation	147
4.1	Model problem	147
4.2	Numerical algorithm	148
4.3	Finite Element Discretization	150
4.4	Numerical Experiments for Time Dependent Solutions	151
4.4.1	A polynomial example	151
4.4.2	An exponential example	153
4.5	Numerical Experiments for Time Independent Solutions	153
4.5.1	Smooth examples	154
4.5.2	Non-smooth examples	155
4.5.3	Examples with singularities	159
4.5.4	Examples with Dirac functions on the right hand side	161
4.6	An adaptive algorithm for the elliptic Monge-Ampère equation	166
4.7	General Algorithm and Numerical Experiments	169
4.7.1	Smooth exponential example	170
4.7.2	Non-smooth examples	170
4.7.3	Examples with data involving a Dirac function	175
5	Conclusion	181
	Bibliography	183
	Curriculum Vitae	201

Introduction

Partial differential equations (PDEs) are often used to describe problems in scientific and engineering fields. Depending on their characteristics, PDEs can be classified as linear or nonlinear. Under the nonlinear class, PDEs can be further categorized as semi-linear, quasi-linear, or fully nonlinear. Semi-linear PDEs arise when the unknown variable is nonlinear, but has linear derivatives. Quasi-linear PDEs, on the other side, are characterized by a nonlinear low-order derivatives, and linear highest-order ones. Fully nonlinear equations are the ones where the highest-order derivatives are nonlinear. In this context, the order of a PDE refers to the order of its highest derivative. In this work, we focus on first and second-order fully nonlinear PDEs.

Fully nonlinear equations have much attention because of their importance in many applications. A non-exhaustive list of applications includes astrophysics [Ambrosio et al., 2003], differential geometry [Caffarelli and Milman, 1999b], fluid dynamics [Benamou and Brenier, 2000], image processing [Zitová and Flusser, 2003], mathematical finance [Stojanovic, 2004], mesh generation [Budd et al., 2009], meteorology [Hoskins, 1975], optimal transport [Villani, 2003; Figalli, 2018], alongside others.

In this work, we carry a thorough numerical study of some well-known fully nonlinear equations. We also explore in depth other less well-known equations in the literature of computational mathematics. The numerical methods that we design target primarily first and second-order fully nonlinear equations. The former involves folding or transforming a map whereas the latter addresses the prototypical Monge-Ampère equation and the parabolic Monge-Ampère equation in higher-order dimension. Throughout this study, we develop new algorithms based on variational and Galerkin approaches, which proved to be computationally efficient and reliable. These algorithms use a combination of standard methods such as splitting methods (Alternating direction method of multipliers (ADMM), Marchuk-Yanenko), different variations of Newton methods, and mixed methods with low order finite elements. Moreover, we extend those methods to tackle fully nonlinear equations with adaptive mesh refinement algorithms, which is to the best of our knowledge not well studied, if not at all.

Ultimately, this work paves the way to new applications, creates new tools to solve fully-nonlinear equations, and hopefully gives insightful perspectives to both theoretical analysis and numerical analysis communities. Below, we give a brief definition of fully nonlinear

equations, then we describe each of the equations that we tackle in the subsequent chapters.

Fully nonlinear equations

The general formulation of a scalar fully nonlinear equations can be expressed as

$$F[u] = F(x, u, Du, D^2u) = 0, \quad (1)$$

where $(x, u, Du, D^2u) \in (\Omega, \mathbb{R}, \mathbb{R}^d, \mathbb{R}^{d \times d})$.

For linear, semi-linear, and quasi-linear PDEs, the usual theory develops for classical and weak solutions. These solutions are difficult or impossible to derive for fully nonlinear PDEs, therefore another notion of solutions was established, and is referred to as viscosity solutions, see [Crandall et al., 1984]. There are numerous articles that investigate fully nonlinear PDE's from a theoretical point of view. One of the most cited work that includes extensive bibliography review until 1995 is [Caffarelli and Cabré, 1995]. Other important works on existence, uniqueness, regularity of classical or viscosity solutions can be found in [Evans, 1982; Jensen, 1988; Ishii, 1989; Ishii and Lions, 1990; Caffarelli et al., 1996; Dacorogna and Marcellini, 1999]. Recent works of well-posedness of fully nonlinear first and second order elliptic equations is discussed in [Katzourakis, 2015; Abugirda and Katzourakis, 2018; Katzourakis, 2016]. Results involving singularities are discussed in the following references [Labutin, 2001; Armstrong et al., 2012; Felmer et al., 2012; Birindelli and Galise, 2019].

Early numerical results for fully nonlinear equations dates back to the late 80' in [Oliker and Prussner, 1988]. The first work on the convergence of numerical schemes towards viscosity solutions is [Barles. and Souganidis, 1991]. Later, for more than a decade, most articles focused on convergence theory [Kuo and Trudinger, 1992; Krylov, 2000; Barles and Jakobsen, 2002; Krylov, 2005]. Starting from mid 2000s, many numerical schemes appeared in the literature. Oberman developed a finite difference method based on the framework in [Barles. and Souganidis, 1991], see [Oberman, 2008]. Feng and Neilan introduced the vanishing moment method for second order fully nonlinear equations in [Feng and Neilan, 2009c, 2014]. Glowinski and Dean showed numerical results using augmented Lagrangian and nonlinear least squares methods in [Dean and Glowinski, 2006]. In [Boehmer, 2008], Boehmer presented stability and convergence results using C^1 finite-elements methods. More recent works can be found in [Lakkis and Pryer, 2011; Davydov and Saeed, 2013; Jensen and Smears, 2013; Davydov and Saeed, 2017; Gallistl and Süli, 2019; Lakkis and Mousavi, 2019, 2020]. Some articles provide a summary of the work that has been done in the past years such as [Feng et al., 2013; Neilan et al., 2017].

Let us present some examples of second order fully nonlinear equations. For that, let $\Omega \subset \mathbb{R}^d$, $u : \Omega \rightarrow \mathbb{R}$ and f be a given function. In the literature, the first example considered as the prototype of fully nonlinear equations is the Monge-Ampère equation $F[u] = \det \mathbf{D}^2 u - f = 0$ for given f . We will examine this problem in a two and three-dimensional framework. Another equation

is the elliptic (σ_2) equation $F[u] = |\nabla^2 u|^2 - \mathbf{D}^2 u : \mathbf{D}^2 u - 2f = 0$. The (σ_2) equation appears in differential geometry and continuum mechanics. Numerical results are reported in [Caboussat, 2014], and theoretical results are in [Trudinger, 1997; Gilbarg and Trudinger, 2001; Chang et al., 2011]. Another second order equation is the Pucci $F[u] = \alpha |\nabla^2 u|^2 + (\alpha - 1)^2 \det \mathbf{D}^2 u = 0$ where $\alpha > 1$. This equation combines the Laplace operator and the Monge-Ampère operators with a weight α . Theoretical results of Pucci's equations are obtained in [Busca et al., 2005; Felmer and Quaas, 2006; Felmer et al., 2006; Quaas and Sirakov, 2008] while different numerical methods are developed in [Dean and Glowinski, 2005; Caffarelli and Glowinski, 2008; Lakkis and Pryer, 2013; Finlay and Oberman, 2018; Mishra and Kumar, 2018; Caboussat, 2019; Bonnans et al., 2020].

Although the literature of second order fully nonlinear equations is very rich, we can not claim the same for first order ones. We give few examples of first order fully nonlinear equations. A prototype example of this category is the steady scalar Eikonal equation. Let $u : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ then $F[u] = |\nabla u| - 1 = 0$. This equation appears in several mathematical models e.g. image processing, optics, wave propagation, differential geometry [Dacorogna et al., 2004, 2003; Glowinski et al., 2015; Gremaud and Kuster, 2006; Hysing and Turek, 2004; Qin et al., 1992; Zhao, 2005]. Numerical methods based for the solution of these type of nonlinear equations can be found in, e.g., [Caboussat and Glowinski, 2015b; Caboussat et al., 2015; Dacorogna et al., 2004, 2003; Glowinski et al., 2003]. The next equations equations is the orthogonal maps given by $F[\mathbf{u}] = \nabla \mathbf{u} (\nabla \mathbf{u})^T - \mathbf{I} = \mathbf{0}$, for $\mathbf{u} = [u_1, u_2]^T : \Omega \rightarrow \mathbb{R}^2$. Orthogonal maps has importance in modeling origami applications for instance. Another equation that falls in the category of first order fully nonlinear equation that we will investigate later is the prescribed Jacobian equation $F[\mathbf{u}] = \det \nabla \mathbf{u} - f = 0$ which appears in optimal transport applications.

Orthogonal maps

The art of paper folding inspire many applications in science e.g. self-assembling robots [Felton et al., 2014], mirrors and solar panels [Jasim and Taheri, 2018], DNA folding [Sanderson, 2010]. The mathematical analysis of paper-folding problems has been addressed in [Bern and Hayes, 1996; Basterrechea and Dacorogna, 2014; Dacorogna and Marcellini, 1999; Hull, 2003; Dacorogna et al., 2008a,b, 2010a]. Close related problems are discussed in [Nash, 1954; Ball and James, 1992]. A review paper that summarizes all the work in the mathematical analysis of paper-folding problems is [Dacorogna et al., 2010b]. From the mechanical point of view, an interactive tool for paper folding is developed by [Ghassaei et al., 2018] based on models developed in [Tachi, 2009, 2010; Schenk and Guest, 2011].

The equation that we are interested in here reads as: find $\mathbf{u} : \Omega \rightarrow \mathbb{R}^2$ satisfying

$$\begin{cases} \nabla \mathbf{u} \in \mathcal{O}(2) & \text{in } \Omega, \\ \mathbf{u} = \mathbf{g} & \text{on } \partial\Omega, \end{cases} \quad (2)$$

where $\mathbf{g}$ is a given sufficiently smooth boundary data, and $\mathcal{O}(2)$ the space of orthonormal 2×2

Introduction

matrix-valued functions. Note that finding $\nabla \mathbf{u} \in \mathcal{O}(2)$ is equivalent to the gradient $\nabla \mathbf{u}$ be an orthogonal 2×2 matrix ($\nabla \mathbf{u}(\nabla \mathbf{u})^T = \mathbf{I}$). Equation (2) is referred to in [Dacorogna et al., 2010b] as flat origami. Here we refer to it as orthogonal maps.

Although the initial application of orthogonal maps appears in the theory of paper folding, there exists similar related formulations in other fields, such as, computational geometry [Demaine and Tachi, 2017; Abel et al., 2018], rigid maps [Dudte et al., 2016; Qiu et al., 2019], or rigid displacements and bending [Afkham et al., 2018; Bartels et al., 2017; Bonito et al., 2020]. Due to the low regularity and the possible multiplicity of the solutions of orthogonal maps problem, these solutions have to be defined in a generalized sense, the most commonly accepted one being the sense of viscosity solutions see, e.g., [Crandall et al., 1984].

Orthogonal maps can be considered as a multivariate version of the classical, scalar, Eikonal equation. Therefore, our numerical method of orthogonal maps that developed here are inspired by previous works related to Eikonal equations [Caboussat and Glowinski, 2015b; Caboussat et al., 2015; Dacorogna et al., 2004, 2003; Glowinski et al., 2003]. Tentative approaches for orthogonal maps problems have been described in [Glowinski and Niu, 2017] and in [Glowinski, 2015b, Chapter 8].

The numerical method that we propose to solve (2) relies on a mix of classical variational techniques. The main ingredients of the method are the introduction of a variational principle, and an association of a flow problem, which is solved by operator-splitting techniques. Finally, the splitting strategy allows the decoupling of the local nonlinearities and of the differential operators. Thus, the methodology can be decomposed to the following steps:

- i Penalization method to relax the equation $\nabla \mathbf{u}(\nabla \mathbf{u})^T = \mathbf{I}$.
- ii Derivation of Euler-Lagrange equation of the regularized problem and introduction of a initial value problem (flow problem).
- iii First order operator-splitting scheme to time-discretize an initial value problem.
- iv Low order conforming finite elements approximation.

The operator-splitting approach allows the decoupling of the differential operators from the nonlinearities. This strategy has been successfully applied to other situations, such as Monge-Ampère and Pucci equations [Caboussat et al., 2013; Caffarelli and Glowinski, 2008; Dean and Glowinski, 2008; Glowinski, 2009; Glowinski et al., 2018], visco-plastic or particulate flow [Dean et al., 2007; Glowinski, 2003] and other problems involving non-smooth operators [Caboussat and Glowinski, 2008, 2012].

Prescribed Jacobian equation and inequality

Next, we consider the prescribed Jacobian equation. Let us assume that we have two sets $\Omega, \Omega' \subset \mathbb{R}^2$ (open and bounded) and two mass densities $f : \Omega \rightarrow \mathbb{R}$ and $p : \Omega' \rightarrow \mathbb{R}$, we look for a

map $\mathbf{u} : \Omega \rightarrow \Omega'$ that satisfies

$$\int_{\mathbf{u}(\Omega)} p(\mathbf{y}) d\mathbf{y} = \int_{\Omega} f(\mathbf{x}) d\mathbf{x}.$$

In other words, we look for a transformation $\mathbf{u}$ that spreads the mass f into the mass p . Applying integration by substitution the solution of the above problem is

$$\det \nabla \mathbf{u}(\mathbf{x}) p(\mathbf{u}(\mathbf{x})) = f(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega. \quad (3)$$

Equation (3) appears in many applications e.g. geometry, optics, economics (see [Guillen, 2019]). Theoretical investigations of (3) in all dimensions are done in [Csató et al., 2011]. It is worth mentioning that if one consider $\mathbf{u}(\mathbf{x}) = \nabla \psi(\mathbf{x})$ then (3) becomes the Monge-Ampère equation

$$\det \mathbf{D}^2 \psi = \frac{f(\mathbf{x})}{p(\nabla \psi(\mathbf{x}))} \quad \forall \mathbf{x} \in \Omega.$$

In this work we set $p(\mathbf{u}(\mathbf{x})) = 1$ and we look for a solution to the the Dirichlet problem

$$\begin{cases} \det \nabla \mathbf{u} = f & \text{in } \Omega, \\ \mathbf{u} = \mathbf{g} & \text{on } \partial\Omega, \end{cases} \quad (4)$$

where $\mathbf{g}$ is given and takes values in $\partial\Omega'$. The uniqueness and the regularity of a field $\mathbf{u}$ satisfying (4) are not guaranteed. Some results about (4) are available in literature, but under special cases. For instance, if we set in (4) $\mathbf{g}(\mathbf{x}) = \mathbf{x}$ and $f \in C^{k,a}$ Hölder space with $k \in \mathbb{N}_0, 0 < a < 1$ results are given in [Dacorogna and Moser, 1990] where the necessary condition $\int_{\Omega} f = |\Omega|$ is assumed. Later, other proofs were given in [Rivière and Ye, 1996; Avinyó et al., 2012; Carlier and Dacorogna, 2012]. Results that consider the right hand side f to be in the Sobolev spaces are given in [Ye, 1994; Burago and Kleiner, 1998; McMullen, 1998]. Numerical results using the augmented Lagrangian method for solution of (4) are given in [Caboussat and Glowinski, 2015a, 2018]. A related problem in incompressible elasticity has been addressed in [Glowinski and Tallec, 1989].

In order to demonstrate the flexibility of the proposed method we also consider the prescribed Jacobian inequality. The problem reads as

$$\begin{cases} \det \nabla \mathbf{u} \geq f & \text{in } \Omega, \\ \mathbf{u} = \mathbf{g} & \text{on } \partial\Omega. \end{cases} \quad (5)$$

Theoretical results of the prescribed Jacobian inequality are given in [Fischer and Kneuss, 2019], and to the best of our knowledge there are no proposed numerical schemes to solve these type of equations.

The numerical method that we propose solving (4) relies on reformulating the problem into a nonlinear least squares one. This is done by minimizing the L^2 -distance between $\nabla \mathbf{u}$ and a matrix-valued function $\mathbf{p} \in (L^2(\Omega))^{2 \times 2}$, where $\mathbf{u}$ satisfies the boundary conditions of the

problem, and $\mathbf{p}$ satisfies $\det \mathbf{p} = f$. Then, a relaxation algorithm allows the decoupling of the local nonlinearities and of the differential operators. The methodology can decompose to the following steps:

- i Reformulation of the problem to nonlinear least square formulation.
- ii Use of relaxation algorithm ADMM type that decouples to local nonlinear optimization problems, and linear variational problems.
- iii Use of Newton type methods to solve the local nonlinear problems.
- iv Use of a mixed finite element method using low order conforming finite elements approximations to solve the variational problem.

The numerical method that we propose to solve (5) follows closely the above framework and the only difference lies in the solution of the local nonlinear problems.

Numerical solutions obtained using the least squares formulation together with the ADMM algorithm have been used for other fully-nonlinear or quasi-linear equations, for instance in [Dean et al., 1996; Dean and Glowinski, 2004, 2005, 2006; Glowinski, 2009, 2008; Caboussat et al., 2013; Caboussat, 2014; Glowinski, 2015b; Caboussat et al., 2018; Caboussat, 2019]. The ADMM algorithm is firstly introduced in [Glowinski and Marroco, 1975] while there are similar algorithm developed in [Douglas and Rachford, 1956; Gabay and Mercier, 1976]. The ADMM in a nutshell is a convex optimization solver which breaks the problem into easy to handle sub-problems. The algorithm is used in numerous articles in different fields, for full bibliography review we refer to [Glowinski et al., 2018].

Monge-Ampère equation

The Monge-Ampère equation is the most established benchmark equation in the literature of second order fully nonlinear equation. The Monge-Ampère equation receives a lot of attention since it appears in many applications, for instance in finance [Stojanovic, 2004], in seismic wave propagation [Engquist et al., 2016], in geostrophic flows [Feng and Neilan, 2009b], in differential geometry [Feng et al., 2007], in mechanics and physics [Frisch et al., 2002], and it has some applications in optimal transport reported in [Caffarelli and Milman, 1999a]. The Dirichlet problem for the elliptic Monge-Ampère equation reads as follows

$$\begin{cases} \det \mathbf{D}^2 u = f & \text{in } \Omega, \\ u = g & \text{on } \partial\Omega, \end{cases} \quad (6)$$

where g and f are given sufficiently smooth data.

Theoretical results about existence of weak solutions in all dimensions of Monge-Ampère equation has been discussed in [Dairbekov et al., 2010; Aleksandrov, 1958]. Moreover, existence and uniqueness for the two dimensional case is discussed in [Aleksandrov, 1958]. The regularity of the solution is addressed in [Calabi, 1958; Pogorelov, 1971]. The existence of global smooth solution proved in [Ivchikina, 1985; Krylov, 1984; Caffarelli et al., 2014]. Results with non

smooth data are discussed in [Caffarelli, 1991; De Philippis and Figalli, 2013]. Dedicated books on Monge-Ampère equation are [Gutiérrez, 2001; Figalli, 2017] among others.

From a computational point of view, the last years witnessed an explosion in the proposed number of numerical solutions of the Monge-Ampère equation proposed in the literature. Mainly, new works consider the two-dimensional Monge-Ampère equation while, in three-dimensional case, the complexity increases a lot and only a handful of methods are available.

The first method that approximated numerically the two-dimensional Monge-Ampère equation has been introduced [Oliker and Prussner, 1988], which is based on a geometrical interpretation of the Monge-Ampère equation. The convergence of this method is discussed in [Nochetto and Zhang, 2019]. A finite difference monotone scheme is constructed in [Oberman, 2008] using wide stencils. This scheme is later improved and extended in three-dimensional case in [Froese and Oberman, 2011, 2012]. The Newton method to linearize the Monge-Ampère equation together with finite differences is used in [Loeper and Rapetti, 2005] and finite elements in [Feng and Neilan, 2009a; Lakkis and Pryer, 2013; Davydov and Saeed, 2013; Neilan, 2014b; Davydov and Saeed, 2017]. Other finite differences-based methods can be found in [Benamou et al., 2010; Awanou, 2016; Nochetto et al., 2018; Froese and Salvador, 2018; Nochetto et al., 2019; Nochetto and Ntougkas, 2019] and in three dimensions in [Mirebeau, 2015].

We consider Galerkin methods which are used in a variety of articles. For instance, in [Feng and Neilan, 2009c,a; Neilan, 2010; Feng and Neilan, 2011, 2014], they used the vanishing approach together with finite element method. In [Brenner et al., 2011], the authors constructed low-order penalty methods for the two-dimensional Monge-Ampère equation and then extended it to three-dimensional case in [Brenner and Neilan, 2012]. Related methods are discussed in [Neilan, 2014a; Awanou, 2020] for the two-dimensional setting. Standard finite elements methods for the solution of Monge-Ampère equation in two and three dimensions are reported in [Awanou, 2014]. A method using multigrid schemes is presented in [Liu et al., 2017], whereas methods with meshfree schemes are reported in [Liu and He, 2014; Böhmer and Schaback, 2019]. A method dedicated to non-convex 2d grids is detailed in [Jensen, 2018]. Other FE methods can be found in [Zheligovsky et al., 2010; Cossette and Smolarkiewicz, 2011; Weller et al., 2016; Feng and Jensen, 2017; Westphal, 2019].

Among the different approaches used to solve the two and three-dimensional Monge-Ampère equation, the operator-splitting method is a relevant in many works such as [Liu et al., 2019; Glowinski et al., 2019] and our work on orthogonal maps in [Caboussat et al., 2019]. One of the first Galerkin methods for the solution of Monge-Ampère equation in two dimensions using augmented Lagrangian or least-squares/relaxation approaches has been advocated in [Dean and Glowinski, 2008; Glowinski, 2009] and later improved in [Caboussat et al., 2013]. This method has been extended to optical design applications in [Prins et al., 2015; Thije ten Boonkamp et al., 2019; Yadav et al., 2019a,b]. The main contribution of our work has been to extend the existing method to a three-dimensional case. We also propose a new efficient

method for the two-dimensional case by solving the parabolic Monge-Ampère equation.

As mentioned above a nonlinear least squares formulation is used to solve (6). The reformulation is done by minimizing the L^2 -distance between $\mathbf{D}^2 u$ and a matrix-valued function $\mathbf{p} \in (L^2(\Omega))^{3 \times 3}$ where u satisfies the boundary conditions of the problem and $\mathbf{p}$ satisfies $\det \mathbf{p} = f$. Then a relaxation algorithm allows the decoupling of the local nonlinearities and of the differential operators. Finally, the methodology can decompose to the following steps

- i Reformulation of the problem to a nonlinear least squares optimization problem.
- ii Use of relaxation algorithm ADMM type that decouples the problem into local nonlinear optimization problems, and linear variational problems.
- iii Use of Newton type methods to solve the local nonlinear problems.
- iv Use of a mixed finite element method using low order conforming finite elements approximations to solve the variational problem.

Parabolic Monge-Ampère equation

Last, we consider parabolic fully nonlinear equations. For instance we examine equations expressed as

$$u_t + F(x, u, Du, D^2 u) = 0, \quad (7)$$

where $(x, u, Du, D^2 u) \in (\Omega, \mathbb{R}, \mathbb{R}^n, \mathbb{R}^{n \times n})$.

Theoretical properties of those equations are discussed in [Wang, 1992; Jakobsen and Karlsen, 2002; Caffarelli and Stefanelli, 2008; Imbert and Silvestre, 2013; Serra, 2015; Ekren et al., 2016; Krylov, 2018]. Numerical results for fully nonlinear parabolic equations are given in [González et al., 2002; Budd and Williams, 2006; Zhang and Zhuo, 2014; Feng and Lewis, 2014, 2018]. Some applications of interest arise for example in finance [Stojanovic, 2004; Koleva and Vulkov, 2013], or in mesh adaptation techniques [Budd et al., 2009; Budd and Williams, 2009; Sulman et al., 2011a; Sulman, 2017; Sulman et al., 2020].

We focus here on a time-evolutive parabolic Monge-Ampère equation where the problem reads: for Ω bounded domain, find $u : \Omega \times (0, T) \rightarrow \mathbb{R}$ satisfying

$$\begin{cases} \frac{\partial u}{\partial t} - \det \mathbf{D}^2 u = f & \text{in } \Omega \times (0, T), \\ u = g & \text{in } \partial\Omega \times (0, T), \\ u(0) = u_0 & \text{in } \Omega. \end{cases} \quad (8)$$

A numerical method to solve equation (8) has been developed in [Feng and Lewis, 2018] where the authors construct a scheme using high order local discontinuous Galerkin methods. Another perspective of equation (8) is given if we use the relationship $\det \mathbf{D}^2 u = \frac{1}{2} \nabla \cdot (\text{cof}(\mathbf{D}^2 u) \nabla u)$. The equation (8) can be seen as a nonlinear heat equation. Results for quasi-linear parabolic equations have been developed in [Thomee, 2007; Akrivis, 2015; Akrivis and Lubich, 2015].

Finally, we can use (8) or a close related form to derive methods for solving the steady state Monge-Ampère equation, see, e.g. in [Sulman et al., 2011b; Feng and Lewis, 2018; Liu et al., 2019].

The numerical method that we propose to solve (8) is as follows

- i Time discretization by using Crank–Nicolson type scheme.
- ii Linearization of the problem with the Newton method.
- iii Space discretization is achieved by continuous low order finite elements.

The Newton method for the linearization of the Monge-Ampère operator also successfully used in [Loeper and Rapetti, 2005; Lakkis and Pryer, 2013].

Adaptive methods

After developing, analyzing, implementing and testing our solvers for first and second order fully nonlinear equations, our goal is to design mesh adaptation algorithms based on heuristic error estimates. A posteriori error estimates for linear elliptic equations is discussed in [Babuška and Rheinboldt, 1978; Babuška et al., 1992; Verfürth, 1996; Iyer, 1999; Verfürth, 2013], and for nonlinear equations in [Akrivis et al., 2009; Verfürth, 1994]. Adaptive mesh refinements techniques are proposed in [Verfürth, 1994; Dorfler, 1996; Soner et al., 2003]. Anisotropic a posteriori error estimates are given in [Apel et al., 2011; Picasso, 2003b,a; Formaggia et al., 2004; Picasso, 2005, 2006]. To the best of our knowledge, there are no works dealing with mesh adaptation techniques for first order fully nonlinear equations. For the second order fully nonlinear equations, preliminary results are given in [Lakkis and Pryer, 2013] where the adaptive mesh refinement is based on Zienkiewicz–Zhu gradient recovery a posteriori estimator. Moreover, an adaptive finite difference method is developed in [Froese and Salvador, 2018].

In this work we will consider isotropic and anisotropic adaptive meshes. Isotropic adaptive meshes will be chosen mainly to tackle point singularities. Anisotropic adaptive meshes, on the other hand, will be favored when we deal with line singularities.

As an example of first-order fully nonlinear equations we consider the orthogonal maps equation. The simplified error estimate in this case is derived in the linear variational problem which corresponds to a Laplace equation $\nabla^2 \mathbf{u} = \mathbf{f}$. For this equation, anisotropic a posteriori error estimates and an adaptive algorithm has been developed in [Picasso, 2003a] which we closely follow here.

Next, we derive an error estimate for the Monge-Ampère operator $\det \mathbf{D}^2 u$. We write the Monge-Ampère equation with the equivalent formulation as $\frac{1}{2} \nabla \cdot (\text{cof}(\mathbf{D}^2 u) \nabla u) = f$. Then, we derive a heuristic a posteriori error estimate based on residual estimate. The derived error estimate is used as a criterion to the adaptive algorithm.

In the above adaptive algorithms, our goal is to reduce the number of elements and increase the accuracy of the solutions. Numerical examples are presented to verify the robustness of

the algorithms in each case. The adaptive algorithms designed for orthogonal maps and the Monge-Ampère operator can be used with minor changes to solve other equations such as Eikonal equations or Pucci equation.

Thesis outline

In Chapter 1 of this thesis we present a numerical method for the approximation of the solution of the orthogonal maps equation. We provide the mathematical formulation of the problem, the operator splitting method, and its finite element implementation. We also show the robustness of the method by presenting several numerical experiments. Then we extend the method by developing an adaptive algorithm. We conclude this chapter by discussing numerical results of the adaptive algorithm.

In Chapter 2, we present a numerical method for the approximation of the solution of the prescribed Jacobian equation and inequality. We provide details about the equality/inequality, and we reformulate it as a nonlinear least square problem. We develop the method by introducing a relaxation algorithm that splits the problem into local nonlinear and variational problems. We develop the solvers for the algebraic and differential problems. Then, we present an implementation in the finite element space and conclude with numerical examples to validate the method.

In Chapter 3, we present a numerical method for the solution of the three-dimension Monge-Ampère equation. We describe the mathematical formulation and detail of the nonlinear least-squares method that is solved using a relaxation algorithm. We develop the algebraic and differential solvers. Finally, we present the finite elements implementation and numerical examples using $\mathbb{P}_1$ and $\mathbb{Q}_1$ polynomials.

In Chapter 4, we present a numerical method for the solution of the parabolic Monge-Ampère equation. We describe the proposed method and we verify its efficiency on time-dependent, and stationary problems. In the second part of this chapter, we derive an error estimate for the steady-state Monge-Ampère equation, and develop an adaptive algorithm. The robustness of this proposed algorithm is checked using various test cases and two different solvers.

Chapters 1, 3 and a part of Chapter 4 are based on works were successfully published [Caboussat et al., 2018, 2019; Caboussat and Gourzoulidis, 2019] or submitted [Caboussat et al., 2020] during this study. Chapters 2 and part of Chapter 4 is intended to be reworked and published in the future.

1 Numerical Approximation of Orthogonal Maps

1.1 Mathematical Model

1.1.1 Initial problem

Let Ω be a open bounded domain of $\mathbb{R}^2$, and consider $\mathbf{g}: \partial\Omega \rightarrow \mathbb{R}^2$ a sufficiently smooth given function on its boundary. The unit square $\Omega = (0, 1)^2$ is a typical domain when considering origami applications. The problem of interest is written as follows. Find $\mathbf{u}: \Omega \rightarrow \mathbb{R}^2$ satisfying

$$\begin{cases} \nabla \mathbf{u} \in \mathcal{O}(2) & \text{in } \Omega, \\ \mathbf{u} = \mathbf{g} & \text{on } \partial\Omega, \end{cases} \quad (1.1)$$

where in (1.1)

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \quad \nabla \mathbf{u} = \begin{pmatrix} \frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial x_2} \\ \frac{\partial u_2}{\partial x_1} & \frac{\partial u_2}{\partial x_2} \end{pmatrix},$$

and $\mathcal{O}(2)$ the space of orthonormal 2×2 matrix-valued functions. Here u_1 , resp. u_2 , denote the first, resp. second, component of the mapping $\mathbf{u}$. Actually, $\mathbf{u}$ is a mapping that maps Ω (the original sheet) into another domain of $\mathbb{R}^2$. The image $\mathbf{u}(\Omega)$ of Ω through this mapping corresponds to the resulting paper sheet, described in the two-dimensional space, after the folding. Problem (1.1) can be equivalently written as:

$$\begin{cases} \nabla \mathbf{u} (\nabla \mathbf{u})^T = \mathbf{I} & \text{a.e in } \Omega, \\ \mathbf{u} = \mathbf{g} & \text{on } \partial\Omega, \end{cases} \quad (1.2)$$

or

$$\begin{cases} |\nabla u_1| = 1 & \text{a.e in } \Omega, \\ |\nabla u_2| = 1 & \text{a.e in } \Omega, \\ \nabla u_1 \cdot \nabla u_2 = 0 & \text{a.e in } \Omega, \\ \mathbf{u} = \mathbf{g} & \text{on } \partial\Omega. \end{cases} \quad (1.3)$$

We adopt the later formulation to set up the numerical method. Let us now develop equation (1.3) as follows:

$$\begin{cases} \left(\frac{\partial u_1}{\partial x_1} \right)^2 + \left(\frac{\partial u_1}{\partial x_2} \right)^2 = 1 \\ \left(\frac{\partial u_2}{\partial x_1} \right)^2 + \left(\frac{\partial u_2}{\partial x_2} \right)^2 = 1 & \text{in } \Omega, \\ \frac{\partial u_1}{\partial x_1} \frac{\partial u_2}{\partial x_1} + \frac{\partial u_1}{\partial x_2} \frac{\partial u_2}{\partial x_2} = 0 \\ \mathbf{u} = \mathbf{g} & \text{on } \partial\Omega. \end{cases} \quad (1.4)$$

The problem of interest (1.3) is reminiscent of the scalar Eikonal equation, but introduces a coupling between the two components; it can be seen as a *vectorial Eikonal problem*.

Theoretical considerations of (1.1) or (1.3) have been addressed, e.g., in [Ball and James, 1992; Dacorogna and Marcellini, 1999; Dacorogna et al., 2008b, 2010b]. For instance, an explicit solution of the homogeneous Dirichlet problem has been exhibited in [Dacorogna et al., 2018]. Existence conditions of a solution have been shown, but the uniqueness of that solution is generally not guaranteed (As we show in the numerical experiments section, we can easily construct problems with multiple solutions).

1.1.2 Determinant property and angle condition

A determinant representation of (1.1) can be found by the following algebraic computations

$$\begin{aligned} (\det \nabla \mathbf{u})^2 &= \left(\frac{\partial u_1}{\partial x_1} \frac{\partial u_2}{\partial x_2} - \frac{\partial u_1}{\partial x_2} \frac{\partial u_2}{\partial x_1} \right)^2 \\ &= \left(\frac{\partial u_1}{\partial x_1} \frac{\partial u_2}{\partial x_2} \right)^2 + \left(\frac{\partial u_1}{\partial x_2} \frac{\partial u_2}{\partial x_1} \right)^2 - 2 \frac{\partial u_1}{\partial x_1} \frac{\partial u_2}{\partial x_2} \frac{\partial u_1}{\partial x_2} \frac{\partial u_2}{\partial x_1}. \end{aligned} \quad (1.5)$$

Now, if we multiply the first two equations of (1.4) by each other we have

$$\left(\frac{\partial u_1}{\partial x_1} \frac{\partial u_2}{\partial x_1}\right)^2 + \left(\frac{\partial u_1}{\partial x_2} \frac{\partial u_2}{\partial x_1}\right)^2 + \left(\frac{\partial u_1}{\partial x_1} \frac{\partial u_2}{\partial x_2}\right)^2 + \left(\frac{\partial u_1}{\partial x_2} \frac{\partial u_2}{\partial x_2}\right)^2 = 1. \quad (1.6)$$

Combine (1.5) and (1.6) we obtain

$$(\det \nabla \mathbf{u})^2 + 2 \frac{\partial u_1}{\partial x_1} \frac{\partial u_2}{\partial x_2} \frac{\partial u_1}{\partial x_2} \frac{\partial u_2}{\partial x_1} + \left(\frac{\partial u_1}{\partial x_1} \frac{\partial u_2}{\partial x_1}\right)^2 + \left(\frac{\partial u_1}{\partial x_2} \frac{\partial u_2}{\partial x_2}\right)^2 = 1,$$

which can be written as

$$(\det \nabla \mathbf{u})^2 = 1 - \left(\frac{\partial u_1}{\partial x_1} \frac{\partial u_2}{\partial x_1} + \frac{\partial u_1}{\partial x_2} \frac{\partial u_2}{\partial x_2}\right)^2,$$

and using the third equation of (1.4) we can conclude that $\det \nabla \mathbf{u} = \pm 1$.

Therefore, the solution of (1.1) must satisfy

$$\det \nabla \mathbf{u}(\mathbf{x}) = \pm 1, \quad a.e. \mathbf{x} \in \Omega, \quad (1.7)$$

except on the folding lines. In the numerical experiments section, illustrations of $\det \nabla \mathbf{u}$ for different test cases show how the sign is fluctuating and is not defined on the folding lines. Moreover, for each vertex that lies at the intersection of folding lines, conditions apply on the number of singularity lines crossing each other (see [Dacorogna and Marcellini, 1999]). This property is sometimes called the angle condition and states that the number of edges coming from one vertex is even, and the determinant $\det \nabla \mathbf{u}$ alternates between 1 and -1 for adjacent regions touching the vertex.

1.1.3 Rigid maps problem

Problem (1.1) is actually a problem rewritten from the so-called rigid maps problem, which consists in finding the map $\tilde{\mathbf{u}} : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$ such that

$$\begin{cases} \nabla \tilde{\mathbf{u}} \in \mathcal{O}(2, 3) & a.e. \text{ in } \Omega, \\ \tilde{\mathbf{u}} = \tilde{\mathbf{g}} & \text{ on } \partial\Omega, \end{cases} \quad (1.8)$$

where $\mathcal{O}(2, 3) = \{\mathbf{A} : \Omega \rightarrow \mathbb{R}^{2 \times 3} : \mathbf{A}^T \mathbf{A} = \mathbf{I} \text{ in } \Omega\}$. The solution to this equivalent problem consists in finding the mapping $\tilde{\mathbf{u}}$ that embeds the folded paper $\tilde{\mathbf{u}}(\Omega)$ into $\mathbb{R}^3$. A graphical representation of problem (1.1) and (1.8) can be found in Figure 1.1.

Remark. In the general case map we consider $\tilde{\mathbf{u}} : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ such that is considered.

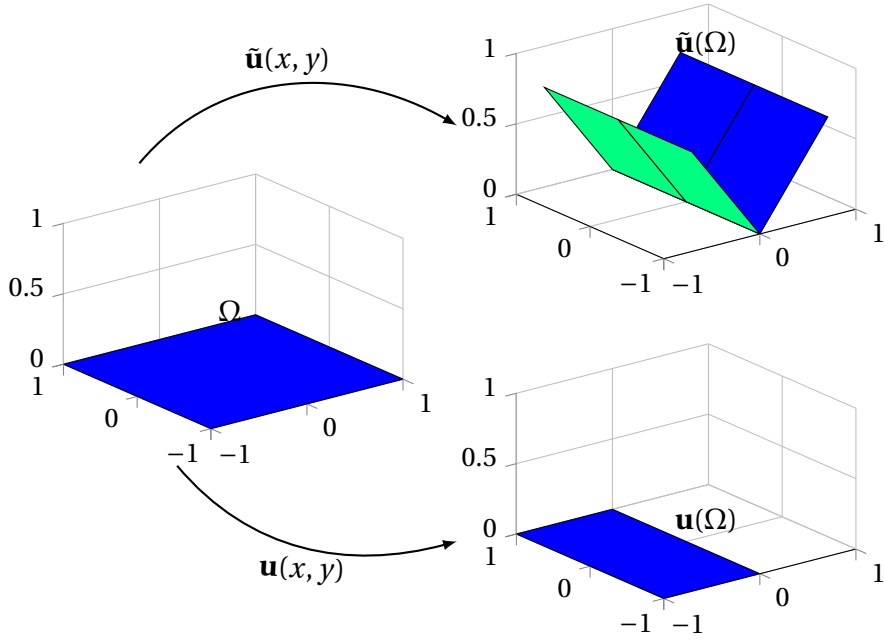


Figure 1.1 – Graphical representation (bottom) of the orthogonal map $\mathbf{u} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, where $\mathbf{u}(x, y) = (-|x|, y)$ is the solution to (1.1) when folding a paper along a centered axis oriented with the Ox axis; and, graphical representation of the rigid map $\tilde{\mathbf{u}} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$, where $\tilde{\mathbf{u}}(x, y) = (x/\sqrt{2}, y, |x|/\sqrt{2})$ is the solution to (1.8) when folding a paper in $\mathbb{R}^3$.

$$\begin{cases} \nabla \tilde{\mathbf{u}} \in \mathcal{O}(n, m) & \text{in } \Omega, \\ \tilde{\mathbf{u}} = \bar{\mathbf{g}} & \text{on } \partial\Omega, \end{cases}$$

where $\mathcal{O}(n, m) = \{\mathbf{A} : \Omega \rightarrow \mathbf{R}^{n \times m} : \mathbf{A}^T \mathbf{A} = \mathbf{I} \text{ in } \Omega\}$ and $(n, m) \in \mathbf{N}$. In this work, we examine only the cases where $n = m = 2$.

1.2 Numerical Algorithms

We propose a numerical method for (1.3) that relies on variational techniques. The underlying principles are the introduction of a variational principle, and of the corresponding flow problem, which is solved by operator-splitting techniques. Ultimately, the splitting strategy allows the decoupling of the local nonlinearities and of the differential operators.

1.2.1 Regularization and Penalization

In order to solve (1.3) and to enforce the uniqueness of the solution, we consider a variational problem. We denote $\mathbf{f}$ to be a given vector-valued function, which corresponds to a target solution, for instance given by prior information. Then the variational problem reads as: find

$\mathbf{u} \in \mathbf{E}_{\mathbf{g}}$ satisfying

$$J(\mathbf{u}) \leq J(\mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{E}_{\mathbf{g}}, \quad (1.9)$$

where

$$J(\mathbf{v}) = \frac{C}{2} \int_{\Omega} |\mathbf{v} - \mathbf{f}|^2 \, d\mathbf{x} + \frac{1}{2} \int_{\Omega} |\nabla \mathbf{v}|^2 \, d\mathbf{x}, \quad (1.10)$$

and

$$\mathbf{E}_{\mathbf{g}} = \{\mathbf{v} \in (H^1(\Omega))^2, \, \mathbf{v}|_{\partial\Omega} = \mathbf{g}, \, \nabla \mathbf{v} \in \mathcal{O}(2) \text{ a.e in } \Omega\}. \quad (1.11)$$

In (1.10), $C > 0$ is a positive constant. While the first term of (1.10) is a distance to be minimized, the second term ensures smoothness of the solution. In most numerical experiments, no a priori information is known therefore we consider $\mathbf{f} = \mathbf{0}$.

Note that the equation (1.3) itself has been transferred into a constraint in (1.11). To handle such constraints, we use an approach which has been successful with the scalar Eikonal equation in Caboussat and Glowinski [2015b], namely we penalize the constraint $\nabla \mathbf{v} \in \mathcal{O}(2)$ (or, equivalently, the first three equations of (1.3)). Let $\varepsilon_1 > 0$ be a given (regularization) parameter, and $\varepsilon_2 > 0$ be a given (penalization) parameter. We denote $(\varepsilon_1, \varepsilon_2)$ by ε . The modified objective function is defined as follows:

$$\begin{aligned} J_{\varepsilon}(\mathbf{v}) \quad := \quad & J(\mathbf{v}) + \frac{\varepsilon_1}{2} \int_{\Omega} |\nabla^2 \mathbf{v}|^2 \, d\mathbf{x} \\ & + \frac{1}{4\varepsilon_2} \int_{\Omega} [(|\nabla v_1|^2 - 1)^2 + (|\nabla v_2|^2 - 1)^2 + |\nabla v_1 \cdot \nabla v_2|^2] \, d\mathbf{x}. \end{aligned} \quad (1.12)$$

The second term is a biharmonic regularization designed to increase the robustness of the convergence of the numerical algorithm. The introduction of such a term has been already experimented for similar problems, see [Caboussat and Glowinski, 2018], and will be discussed numerically via experiments. The variational problem (1.9) becomes: Find $\mathbf{u}^{\varepsilon} \in \mathbf{V}_{\mathbf{g}}$ satisfying

$$J_{\varepsilon}(\mathbf{u}^{\varepsilon}) \leq J_{\varepsilon}(\mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{V}_{\mathbf{g}}, \quad (1.13)$$

where

$$\mathbf{V}_{\mathbf{g}} = \{\mathbf{v} \in (H^2(\Omega))^2, \, \mathbf{v}|_{\partial\Omega} = \bar{\mathbf{g}}\}. \quad (1.14)$$

The numerical approach to solve (1.13) relies on an appropriate reformulation of the problem when considering the first order optimality conditions, together with the introduction of a flow problem.

In (1.14), we used $\bar{\mathbf{g}}$ an extension of $\mathbf{g}$ belonging to $(H^{3/2}(\partial\Omega))^2$ see e.g. [Ern and Guermond, 2004], to make the boundary data compatible with the regularizing term in (1.10). However, numerical experiments in Section 1.2 suggest regularizing $\mathbf{g}$ is not essential.

1.2.2 First Order Optimality Conditions

We define the tensor-valued function $\mathbf{p}^\varepsilon := \nabla \mathbf{u}^\varepsilon$. Problem (1.13) is equivalent to: Find $\mathbf{p}^\varepsilon \in \mathbf{Q}_4 = (L^4(\Omega))^{2 \times 2}$ satisfying

$$j_\varepsilon(\mathbf{p}^\varepsilon) \leq j_\varepsilon(\mathbf{q}), \quad \forall \mathbf{q} \in \mathbf{Q}_4, \quad (1.15)$$

where

$$\begin{aligned} j_\varepsilon(\mathbf{q}) = & \frac{1}{2} \int_{\Omega} |\mathbf{q}|^2 + \frac{C}{2} \int_{\Omega} |\mathbf{B}\mathbf{q} + \mathbf{u}_{\mathbf{g}} - \mathbf{f}|^2 \, d\mathbf{x} + \mathbf{I}_{\nabla}(\mathbf{q}) \\ & + \frac{1}{4\varepsilon_2} \int_{\Omega} [(|q_1|^2 - 1)^2 + (|q_2|^2 - 1)^2 + |q_1 \cdot q_2|^2] \, d\mathbf{x}, \end{aligned} \quad (1.16)$$

and q_1, q_2 are the two column vectors of $\mathbf{q}$. Here we define:

- (i) the function $\mathbf{u}_{\mathbf{g}} \in (H^2(\Omega))^2$ as the unique solution (harmonic extension) of

$$\begin{cases} \nabla^2 \mathbf{u}_{\mathbf{g}} = \mathbf{0} & \text{in } \Omega, \\ \mathbf{u}_{\mathbf{g}} = \mathbf{g} & \text{on } \partial\Omega. \end{cases}$$

- (ii) the function $\mathbf{B}\mathbf{q}$ as the unique solution in $(H_0^1(\Omega))^2$ of

$$\nabla^2 \mathbf{B}\mathbf{q} = \nabla \cdot \mathbf{q} \text{ in } \Omega.$$

- (iii) the functional $\mathbf{I}_{\nabla}(\mathbf{q})$ as

$$\mathbf{I}_{\nabla}(\mathbf{q}) = \begin{cases} \frac{\varepsilon_1}{2} \int_{\Omega} |\nabla \cdot \mathbf{q}|^2 \, d\mathbf{x} & \text{if } \mathbf{q} \in \nabla \mathbf{V}_{\mathbf{g}}, \\ +\infty & \text{otherwise.} \end{cases}$$

Based on this *change of variables*, the first order optimality conditions (Euler-Lagrange equa-

tions) relative to (1.16) read as follows: Find $\mathbf{p}^\varepsilon \in \mathbf{Q}_4$ such that:

$$\begin{aligned} & \int_{\Omega} \mathbf{p}^\varepsilon : \mathbf{q} \, d\mathbf{x} + C \int_{\Omega} (\mathbf{B}\mathbf{p}^\varepsilon + \mathbf{u}_g - \mathbf{f}) \cdot \mathbf{B}\mathbf{q} \, d\mathbf{x} + \langle \partial \mathbf{I}_\nabla(\mathbf{p}^\varepsilon), \mathbf{q} \rangle \\ & + \frac{1}{\varepsilon_2} \int_{\Omega} \left[(|p_1^\varepsilon|^2 - 1) p_1^\varepsilon \cdot q_1 + (|p_2^\varepsilon|^2 - 1) p_2^\varepsilon \cdot q_2 \right. \\ & \left. + \frac{1}{2} p_1^\varepsilon \cdot p_2^\varepsilon (p_2^\varepsilon \cdot q_1 + p_1^\varepsilon \cdot q_2) \right] d\mathbf{x} = 0, \quad \forall \mathbf{q} \in \mathbf{Q}_4. \end{aligned} \quad (1.17)$$

Here, $\partial \mathbf{I}_\nabla(\cdot)$ denotes the subdifferential of the non-smooth proper lower semi-continuous (l.s.c.) convex functional $\mathbf{I}_\nabla$. In the sequel, the superscript ε will be dropped for simplicity.

1.2.3 Flow Problem and Operator-Splitting Algorithm

The solution method for (1.17) relies on an associated initial-value problem (*flow* in the dynamical systems terminology), to be integrated from $t = 0$ to $t = +\infty$. This initial value problem is defined as follows: Find $\mathbf{p}(t) \in \mathbf{Q}_4$ for a.e. $t \in (0, +\infty)$ satisfying

$$\begin{aligned} & \int_{\Omega} \frac{\partial \mathbf{p}(t)}{\partial t} : \mathbf{q} \, d\mathbf{x} + \int_{\Omega} \mathbf{p}(t) : \mathbf{q} \, d\mathbf{x} + C \int_{\Omega} (\mathbf{B}\mathbf{p}(t) + \mathbf{u}_g - \mathbf{f}) \cdot \mathbf{B}\mathbf{q} \, d\mathbf{x} + \langle \partial \mathbf{I}_\nabla(\mathbf{p}(t)), \mathbf{q} \rangle \\ & + \frac{1}{\varepsilon_2} \int_{\Omega} \left[(|p_1(t)|^2 - 1) p_1(t) \cdot q_1 + (|p_2(t)|^2 - 1) p_2(t) \cdot q_2 \right. \\ & \left. + \frac{1}{2} p_1(t) \cdot p_2(t) (p_2(t) \cdot q_1 + p_1(t) \cdot q_2) \right] d\mathbf{x} = 0, \quad \forall \mathbf{q} \in \mathbf{Q}_4. \end{aligned} \quad (1.18)$$

together with the initial condition $\mathbf{p}(0) = \mathbf{p}_0$ given.

We apply an operator-splitting strategy to solve (1.18) (namely, a first-order Marchuk-Yanenko scheme, see for example in [Glowinski, 2015b]). Let $\Delta t > 0$ be a constant given time step, $t^n = n\Delta t$, $n = 1, 2, \dots$, to define the approximations $\mathbf{p}^n \simeq \mathbf{p}(t^n)$. Starting from the initial condition $\mathbf{p}^0 = \mathbf{p}_0$, the Marchuk-Yanenko scheme allows, using $\mathbf{p}^n$ for all $n \geq 0$, to compute successively $\mathbf{p}^{n+1/2}$ and $\mathbf{p}^{n+1}$ with the two following intermediate steps:

(A) Prediction step: Find $\mathbf{p}^{n+1/2} \in \mathbf{Q}_4$ satisfying

$$\begin{aligned} & \int_{\Omega} \frac{\mathbf{p}^{n+1/2} - \mathbf{p}^n}{\Delta t} : \mathbf{q} \, d\mathbf{x} + \int_{\Omega} \mathbf{p}^{n+1/2} : \mathbf{q} \, d\mathbf{x} \\ & + \frac{1}{\varepsilon_2} \int_{\Omega} \left[(|p_1^{n+1/2}|^2 - 1) p_1^{n+1/2} \cdot q_1 + (|p_2^{n+1/2}|^2 - 1) p_2^{n+1/2} \cdot q_2 \right. \\ & \left. + \frac{1}{2} p_1^{n+1/2} \cdot p_2^{n+1/2} (p_2^{n+1/2} \cdot q_1 + p_1^{n+1/2} \cdot q_2) \right] d\mathbf{x} = 0, \end{aligned} \quad (1.19)$$

for all $\mathbf{q} \in \mathbf{Q}_4$. Notice here that this nonlinear problem does not involve any derivatives of the variable $\mathbf{p}$, a significant simplification,

(B) Correction step: Find $\mathbf{p}^{n+1} \in \mathbf{Q}_4$ satisfying

$$\begin{aligned} \int_{\Omega} \frac{\mathbf{p}^{n+1} - \mathbf{p}^{n+1/2}}{\Delta t} : \mathbf{q} \, dx + C \int_{\Omega} (\mathbf{B}\mathbf{p}^{n+1} + \mathbf{u}_g - \mathbf{f}) \cdot \mathbf{B}\mathbf{q} \, dx \\ + \langle \partial \mathbf{I}_{\nabla}(\mathbf{p}^{n+1}), \mathbf{q} \rangle = 0, \end{aligned} \quad (1.20)$$

for all $\mathbf{q} \in \mathbf{Q}_4$. This problem is actually a (hidden) linear variational problem whose solution will be addressed in Section 1.2.5.

1.2.4 Local Optimization Problems

The sub-problem (1.19) that arises in the splitting algorithm does not involve any derivatives of the variable $\mathbf{q}$. Therefore, it can be solved point-wise a.e. in Ω (see also, e.g., [Caboussat and Glowinski, 2018; Caboussat et al., 2018, 2013] for other instances of similar simplifications). To make (1.19) more explicit, we consider the four following test functions:

$$\mathbf{q}_{11} = q \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{q}_{12} = q \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \mathbf{q}_{21} = q \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \mathbf{q}_{22} = q \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix},$$

with $q \in L^2(\Omega)$. Taking $\mathbf{q} = \mathbf{q}_{ij}$ in (1.19), we obtain the following system of nonlinear equations:

$$\begin{aligned} (1 + \Delta t) p_{11}^{n+1/2} + \frac{\Delta t \mu^{n+1/2} p_{11}^{n+1/2}}{\varepsilon_2} - p_{11}^n + \frac{\Delta t \kappa^{n+1/2} p_{21}^{n+1/2}}{2 \varepsilon_2} &= 0, \\ (1 + \Delta t) p_{12}^{n+1/2} + \frac{\Delta t \mu^{n+1/2} p_{12}^{n+1/2}}{\varepsilon_2} - p_{12}^n + \frac{\Delta t \kappa^{n+1/2} p_{22}^{n+1/2}}{2 \varepsilon_2} &= 0, \\ (1 + \Delta t) p_{21}^{n+1/2} + \frac{\Delta t \lambda^{n+1/2} p_{21}^{n+1/2}}{\varepsilon_2} - p_{21}^n + \frac{\Delta t \kappa^{n+1/2} p_{11}^{n+1/2}}{2 \varepsilon_2} &= 0, \\ (1 + \Delta t) p_{22}^{n+1/2} + \frac{\Delta t \lambda^{n+1/2} p_{22}^{n+1/2}}{\varepsilon_2} - p_{22}^n + \frac{\Delta t \kappa^{n+1/2} p_{12}^{n+1/2}}{2 \varepsilon_2} &= 0, \end{aligned} \quad (1.21)$$

where

$$\begin{aligned} \mu^{n+1/2} &:= \left((p_{11}^{n+1/2})^2 + (p_{12}^{n+1/2})^2 - 1 \right), \\ \lambda^{n+1/2} &:= \left((p_{21}^{n+1/2})^2 + (p_{22}^{n+1/2})^2 - 1 \right), \\ \kappa^{n+1/2} &:= (p_{11}^{n+1/2} p_{21}^{n+1/2} + p_{12}^{n+1/2} p_{22}^{n+1/2}). \end{aligned}$$

Actually, system (1.21) can be reformulated in a more condensed form. Let us denote $[p_{11}, p_{12}]^T$

by α and $[p_{21}, p_{22}]^T$ by β ; then (1.21) becomes

$$\begin{aligned} (1 + \Delta t)\alpha^{n+1/2} + \frac{\Delta t(|\alpha^{n+1/2}|^2 - 1)\alpha^{n+1/2}}{\varepsilon_2} + \frac{\Delta t(\alpha^{n+1/2} \cdot \beta^{n+1/2})\beta^{n+1/2}}{2\varepsilon_2} &= \alpha^n, \\ (1 + \Delta t)\beta^{n+1/2} + \frac{\Delta t(|\beta^{n+1/2}|^2 - 1)\beta^{n+1/2}}{\varepsilon_2} + \frac{\Delta t(\alpha^{n+1/2} \cdot \beta^{n+1/2})\alpha^{n+1/2}}{2\varepsilon_2} &= \beta^n. \end{aligned} \quad (1.22)$$

The above nonlinear system consists of four cubic equations. It is solved using a Newton-Raphson method. If $\Delta t \leq \varepsilon_2$ then we can have a unique solution of system (1.22).

In practice (see Section 1.3), once a finite element discretization of Ω is constructed, (1.22) is solved point-wise on each element of the discretization. The number of systems to solve thus depends on the number of elements of the triangulation.

1.2.5 Variational Problems

The sub-problem (1.20) that arises in the splitting algorithm is a well-posed, linear variational problem. In order to highlight this statement, let us consider the reverse change of variable and take $\nabla \mathbf{u}^{n+1} := \mathbf{p}^{n+1}$. Problem (1.20) can be rewritten as follows : Find $\mathbf{u}^{n+1} \in \mathbf{V}_g$ such that

$$\begin{aligned} \varepsilon_1 \Delta t \int_{\Omega} (\nabla^2 \mathbf{u}^{n+1}) \cdot (\nabla^2 \mathbf{v}) \, \mathrm{d}\mathbf{x} + \int_{\Omega} \nabla \mathbf{u}^{n+1} : \nabla \mathbf{v} \, \mathrm{d}\mathbf{x} + C \Delta t \int_{\Omega} \mathbf{u}^{n+1} \cdot \mathbf{v} \, \mathrm{d}\mathbf{x} = \\ C \Delta t \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, \mathrm{d}\mathbf{x} + \int_{\Omega} \mathbf{p}^{n+1/2} : \nabla \mathbf{v} \, \mathrm{d}\mathbf{x}, \quad \mathbf{v} \in (H^2(\Omega) \cap H_0^1(\Omega))^2. \end{aligned} \quad (1.23)$$

Problem (1.23) is a fourth-order linear variational problem of the biharmonic type. We introduce a coupled problem with an auxiliary variable. The additional equation reads: Find $\mathbf{w}^{n+1} \in (H_0^1(\Omega))^2$ such that

$$\mathbf{w}^{n+1} = -\nabla^2 \mathbf{u}^{n+1} \quad \text{in } \Omega. \quad (1.24)$$

Aggregating (1.23) and (1.24) allows to obtain a coupled second order linear system that reads as follows: Find $(\mathbf{u}^{n+1}, \mathbf{w}^{n+1}) \in \mathbf{V}_g \times (H_0^1(\Omega))^2$ such that

$$\begin{cases} \varepsilon_1 \Delta t \int_{\Omega} \nabla \mathbf{w}^{n+1} : \nabla \mathbf{v} \, \mathrm{d}\mathbf{x} + \int_{\Omega} \nabla \mathbf{u}^{n+1} : \nabla \mathbf{v} \, \mathrm{d}\mathbf{x} + C \Delta t \int_{\Omega} \mathbf{u}^{n+1} \cdot \mathbf{v} \, \mathrm{d}\mathbf{x} \\ = C \Delta t \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, \mathrm{d}\mathbf{x} + \int_{\Omega} \mathbf{p}^{n+1/2} : \nabla \mathbf{v} \, \mathrm{d}\mathbf{x}, \\ \int_{\Omega} \nabla \mathbf{u}^{n+1} : \nabla \mathbf{q} \, \mathrm{d}\mathbf{x} - \int_{\Omega} \mathbf{w}^{n+1} \cdot \mathbf{q} \, \mathrm{d}\mathbf{x} = 0, \end{cases} \quad (1.25)$$

for all $(\mathbf{v}, \mathbf{q}) \in (H_0^1(\Omega))^2 \times (H_0^1(\Omega))^2$.

1.3 Finite Element Discretization

1.3.1 Generalities

The space approximation of the time-stepping algorithm (1.19)-(1.20) is addressed with piecewise linear continuous finite elements. The use of low-order finite elements is appropriate for problems such as (1.1), due to the low regularity of the solution. Indeed, piecewise linear finite element methods rely on spaces of Lipschitz continuous functions well-suited to the approximation of solutions to the orthogonal maps equation.

Let us define a space discretization step $h > 0$, and associate with h a *triangulation* $\mathcal{T}_h$ that satisfies the usual compatibility conditions (see, e.g., [Glowinski, 2008] for a complete definition). Let us denote by Σ_h the (finite) set of the vertices of $\mathcal{T}_h$, by N_h the number of elements in Σ_h , and by Σ_{0h} the subset of those elements in Σ_h not located on Γ (with $N_{0h} := \text{card}(\Sigma_{0h})$). From the triangulation $\mathcal{T}_h$ we define the following finite element spaces:

$$\begin{aligned}\mathbf{V}_h &= \{\mathbf{v} \in (C^0(\bar{\Omega}))^2, \mathbf{v} \in (\mathbb{P}_1)^2, \forall K \in \mathcal{T}_h\}, \\ \mathbf{V}_{\mathbf{g},h} &= \{\mathbf{v} \in \mathbf{V}_h, \mathbf{v}(Q) = \mathbf{g}(Q), \forall Q \text{ vertex of } \mathcal{T}_h \text{ belonging to } \Gamma\}, \\ \mathbf{Q}_h &= \{\mathbf{q} \in (L^\infty(\Omega))^{2 \times 2}, \mathbf{q}|_K \in \mathbb{R}^{2 \times 2}, \forall K \in \mathcal{T}_h\},\end{aligned}$$

where $\mathbb{P}_1$ is the space of two-variable polynomials of degree ≤ 1 . Note that we have $\nabla \mathbf{V}_h \subset \mathbf{Q}_h$.

Next, we equip $\mathbf{V}_h$, and its sub-spaces $\mathbf{V}_{\mathbf{g},h}$, with the following discrete inner product (based on classical quadrature formulas):

$$(\mathbf{v}, \mathbf{w})_{0h} = \sum_{k=1}^{N_h} \sum_{i=1}^{m_k} W_i \mathbf{v}(\zeta_i) \cdot \mathbf{w}(\zeta_i), \quad \forall \mathbf{v}, \mathbf{w} \in \mathbf{V}_h,$$

where W_i , resp. ζ_i are the weights, resp. evaluation points, of a Gauss quadrature rule of order ≥ 2 , and m_k is the number of quadrature points in the element k (supposed constant). The quadrature formulas used are implemented in the library `libmesh` ([Kirk et al., 2006]). The corresponding norm is $\|\mathbf{v}\|_{0h} := \sqrt{(\mathbf{v}, \mathbf{v})_{0h}}$, for all $\mathbf{v} \in \mathbf{V}_h$. In a similar fashion, we equip the space $\mathbf{Q}_h$ with the inner product and norm respectively defined as follows:

$$((\mathbf{p}, \mathbf{q}))_{0h} = \sum_{K \in \mathcal{T}_h} |K| \mathbf{p}|_K : \mathbf{q}|_K,$$

and $\|\mathbf{q}\|_{0h} = \sqrt{((\mathbf{q}, \mathbf{q}))_{0h}}$ (with $|K|$ = area of K).

The discrete version of the numerical algorithm uses the same steps as the continuous version presented in Section 1.2. However, let us sketch the main discretized milestones in the sequel.

1.3.2 Discretization of the flow problem

For the discrete analogous the operator-splitting strategy (1.19)-(1.20), we define the approximations $\mathbf{p}_h^n \simeq \mathbf{p}_h(t^n) \in \mathbf{Q}_h$. Starting from the initial condition $\mathbf{p}_h^0 = \mathbf{p}_{0,h}$, we compute successively $\mathbf{p}_h^{n+1/2}$ and $\mathbf{p}_h^{n+1}$ with the two intermediate steps:

(A) Prediction step: Find $\mathbf{p}_h^{n+1/2} \in \mathbf{Q}_h$ satisfying

$$\begin{aligned} & \int_{\Omega} \frac{\mathbf{p}_h^{n+1/2} - \mathbf{p}_h^n}{\Delta t} : \mathbf{q}_h \, d\mathbf{x} + \int_{\Omega} \mathbf{p}_h^{n+1/2} : \mathbf{q}_h \, d\mathbf{x} \\ & + \frac{1}{\varepsilon_2} \int_{\Omega} \left[(|p_{1,h}^{n+1/2}|^2 - 1) p_{1,h}^{n+1/2} \cdot q_{1,h} + (|p_{2,h}^{n+1/2}|^2 - 1) p_{2,h}^{n+1/2} \cdot q_{2,h} \right. \\ & \left. + \frac{1}{2} p_{1,h}^{n+1/2} \cdot p_{2,h}^{n+1/2} (p_{2,h}^{n+1/2} \cdot q_{1,h} + p_{1,h}^{n+1/2} \cdot q_{2,h}) \right] d\mathbf{x} = 0, \end{aligned} \quad (1.26)$$

for all $\mathbf{q}_h \in \mathbf{Q}_h$.

(B) Correction step: Find $\mathbf{p}_h^{n+1} \in \mathbf{Q}_h$ satisfying

$$\begin{aligned} & \int_{\Omega} \frac{\mathbf{p}_h^{n+1} - \mathbf{p}_h^{n+1/2}}{\Delta t} : \mathbf{q}_h \, d\mathbf{x} + C \int_{\Omega} (\mathbf{B} \mathbf{p}_h^{n+1} + \mathbf{u}_{\mathbf{g},h} - \mathbf{f}) \cdot \mathbf{B} \mathbf{q}_h \, d\mathbf{x} \\ & + \langle \partial \mathbf{I}_{\nabla}(\mathbf{p}_h^{n+1}), \mathbf{q}_h \rangle = 0, \end{aligned} \quad (1.27)$$

for all $\mathbf{q}_h \in \mathbf{Q}_h$.

1.3.3 Solution of the discrete local optimization problems

The finite dimensional nonlinear problem (1.26) can be solved *triangle-wise*; Indeed, if $\mathbf{p}_h^{n+1/2} := \{p_{K,h}^{n+1/2}\}_{K \in \mathcal{T}_h}$, one can rewrite (1.26) as follows: For each triangle $K \in \mathcal{T}_h$, solve

$$\begin{aligned} (1 + \Delta t) p_{K,h,11}^{n+1/2} + \frac{\Delta t \mu_h^{n+1/2} p_{K,h,11}^{n+1/2}}{\varepsilon_2} - p_{K,h,11}^n + \frac{\Delta t \kappa_h^{n+1/2} p_{K,h,21}^{n+1/2}}{2 \varepsilon_2} &= 0, \\ (1 + \Delta t) p_{K,h,12}^{n+1/2} + \frac{\Delta t \mu_h^{n+1/2} p_{K,h,12}^{n+1/2}}{\varepsilon_2} - p_{K,h,12}^n + \frac{\Delta t \kappa_h^{n+1/2} p_{K,h,22}^{n+1/2}}{2 \varepsilon_2} &= 0, \\ (1 + \Delta t) p_{K,h,21}^{n+1/2} + \frac{\Delta t \lambda_h^{n+1/2} p_{K,h,21}^{n+1/2}}{\varepsilon_2} - p_{K,h,21}^n + \frac{\Delta t \kappa_h^{n+1/2} p_{K,h,11}^{n+1/2}}{2 \varepsilon_2} &= 0, \\ (1 + \Delta t) p_{K,h,22}^{n+1/2} + \frac{\Delta t \lambda_h^{n+1/2} p_{K,h,22}^{n+1/2}}{\varepsilon_2} - p_{K,h,22}^n + \frac{\Delta t \kappa_h^{n+1/2} p_{K,h,12}^{n+1/2}}{2 \varepsilon_2} &= 0, \end{aligned} \quad (1.28)$$

where

$$\begin{aligned}\mu_h^{n+1/2} &:= \left(\left(p_{K,h,11}^{n+1/2} \right)^2 + \left(p_{K,h,12}^{n+1/2} \right)^2 - 1 \right), \\ \lambda_h^{n+1/2} &:= \left(\left(p_{K,h,21}^{n+1/2} \right)^2 + \left(p_{K,h,22}^{n+1/2} \right)^2 - 1 \right), \\ \kappa_h^{n+1/2} &:= \left(p_{K,h,11}^{n+1/2} p_{K,h,21}^{n+1/2} + p_{K,h,12}^{n+1/2} p_{K,h,22}^{n+1/2} \right).\end{aligned}$$

System (1.28) is similar to (1.21) and can be solved by Newton techniques, taking $\mathbf{p}_{K,h}^n$ as an initial guess. When applied to the solution of problem (1.28), the Newton method always converges and never requires more than 10 iterations for the test problems considered in Section 1.4.

1.3.4 Solution of discrete linear variational problems

Problem (1.27) is discretized as follows: Find $(\mathbf{u}_h^{n+1}, \mathbf{w}_h^{n+1}) \in \mathbf{V}_{\mathbf{g},h} \times \mathbf{V}_{\mathbf{0},h}$ such that

$$\begin{cases} \varepsilon_1 \Delta t ((\nabla \mathbf{w}_h^{n+1}, \nabla \mathbf{v}_h))_{0h} + ((\nabla \mathbf{u}_h^{n+1}, \nabla \mathbf{v}_h))_{0h} + C \Delta t (\mathbf{u}_h^{n+1}, \mathbf{v}_h)_{0h} \\ = C \Delta t (\mathbf{f}, \mathbf{v}_h)_{0h} + ((\mathbf{p}_h^{n+1/2}, \nabla \mathbf{v}_h))_{0h}, \\ ((\nabla \mathbf{u}_h^{n+1}, \nabla \mathbf{q}_h))_{0h} - (\mathbf{w}_h^{n+1}, \mathbf{q}_h)_{0h} = 0, \end{cases} \quad (1.29)$$

for all $(\mathbf{v}_h, \mathbf{q}_h) \in \mathbf{V}_{\mathbf{0},h} \times \mathbf{V}_{\mathbf{0},h}$.

Let φ_i , $i = 1 \dots N_h$ be the piecewise linear basis functions of $\mathbf{V}_h$. Note that, by defining the classical rigidity and mass matrices A and M by $A_{i,j} = \int_{\Omega} \nabla \varphi_i \cdot \nabla \varphi_j d\mathbf{x}$, $M_{i,j} = \int_{\Omega} \varphi_i \varphi_j d\mathbf{x}$, and suitable right-hand side $\mathbf{F}$, the linear system can be compactly written as

$$\begin{pmatrix} A + C \Delta t M & \varepsilon_1 \Delta t A \\ A & -M \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{w} \end{pmatrix} = \begin{pmatrix} \mathbf{F} \\ \mathbf{0} \end{pmatrix}$$

Among the many methods to solve the above block-structured linear system here we use a monolithic approach, however a Scur complement is another widely use method.

1.4 Numerical Experiments

In this section we are going to report on the results of numerical experiments aimed at assessing the robustness, the accuracy and the efficiency of our methodology. The computational domain (i.e. the paper sheet to be folded) is chosen either as the unit square $\Omega = (0, 1)^2$ or as the unit disk $\Omega = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 < 1\}$. All the experiments have been performed on a desktop computer with Intel Xeon E5-1650 (3.50 GHz $\times$ 12) and 64 GB memory.

We consider several discretizations of the computational domain, using various finite element meshes as shown in Figure 1.2. The influence of the chosen mesh on the solution obtained by the algorithm is also discussed. For all the numerical experiments we consider $\mathbf{f} = (0, 0)^T$, $\varepsilon_2 = 5 \times 10^{-10}$, $\Delta t = \varepsilon_2/2$, $C = 10$, and $\varepsilon_1 = \frac{h^2}{5\Delta t}$ (unless stated otherwise). The stopping criterion we use to decide on the flow stationarity is either $n \leq 1000$, or $\|\mathbf{p}^{n+1} - \mathbf{p}^n\|_{L^2(\Omega)} \leq 5 \times 10^{-4}$ (unless stated otherwise). We observe numerically that, for this stopping criterion, $\|\mathbf{u}^{n+1} - \mathbf{u}^n\|_{L^2(\Omega)}$ is of the order of 10^{-7} .

The choice of $\mathbf{f}$ allows to give an a priori estimate of the solution. With another choice of $\mathbf{f}$, another solution may be obtained, and numerical experiments have shown some sensitivity with respect to the parameters $\mathbf{f}$ and C . The penalization constant ε_2 is chosen in order to guarantee that the orthogonality conditions are satisfied accurately. Values of ε_2 ranging from 10^{-5} to 10^{-10} are suitable. The number of iterations of the Newton method in Section 1.2.4 may decrease when ε_2 is larger. The choice of ε_1 allows to have a regularization term in (1.23) of the order h^2 . In the light of these comments, a thorough sensitivity analysis has not been performed, but the influence of some parameters will be discussed in the numerical experiments.

1.4.1 Smooth validation example

In the first experiment, the algorithm is validated with a boundary condition that corresponds to a smooth mapping, without any singularities. We consider

$$\mathbf{g}(\mathbf{x}) = \mathbf{x}, \quad \forall \mathbf{x} \in \partial\Omega.$$

An exact solution corresponding to this given boundary is the identity mapping $\mathbf{u}(\mathbf{x}) = \mathbf{x}$. This corresponds to an origami without folding.

For such an example, when choosing $\varepsilon_1 = 0.0$ as a smoothing parameter, the problem is solved up to machine precision for all meshes. Figure 1.3 visualizes the graph of the two components of the computed approximate solution. Since the two components of the solution are piecewise affine, the continuous piecewise affine finite element approximation we advocated in Section 1.3 is ideally suited to the solution of the problem under consideration.

Note that a very accurate (up to machine precision) approximation of the solution can be obtained with ε_1 of the order h^2 , together with $\varepsilon_2 \approx 10^{-7}$ and a stopping criterion of $\|\mathbf{p}^{n+1} - \mathbf{p}^n\|_{L^2(\Omega)} \leq 5 \times 10^{-10}$. Table 1.1 illustrates the results obtained with the same setup as described previously. It shows that orthogonality conditions are not jeopardized by the introduction of the well-chosen smoothing parameter.

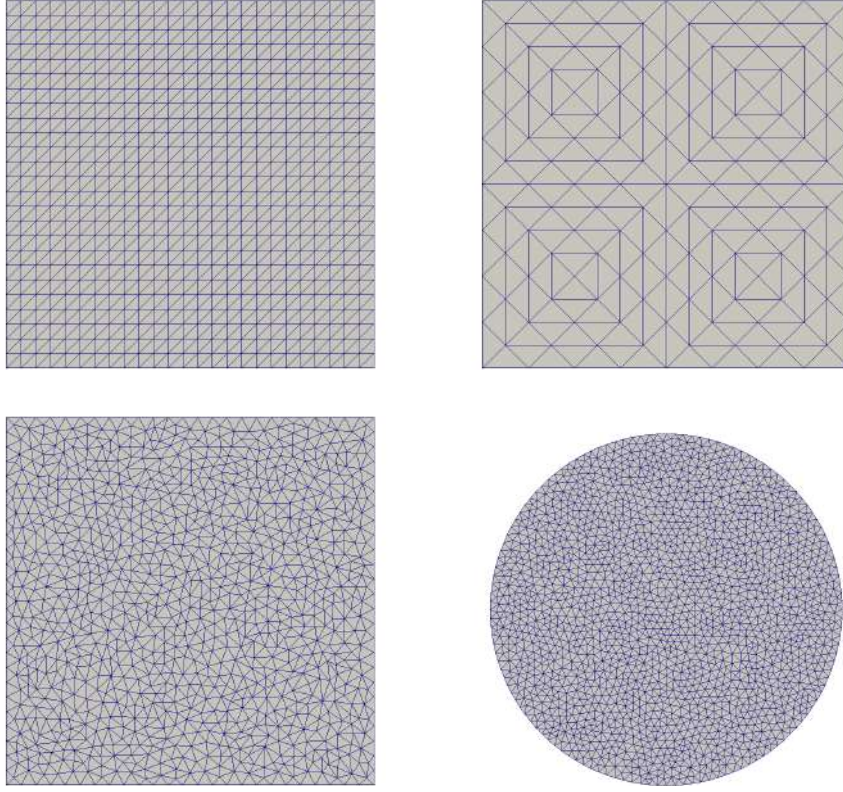


Figure 1.2 – Finite element triangulations used for the numerical experiments. Top left: Structured asymmetric mesh ($\Omega = (0, 1)^2$, $h = 0.04$); top right: Structured symmetric mesh ($\Omega = (0, 1)^2$, $h = 0.125$); bottom left: isotropic unstructured mesh ($\Omega = (0, 1)^2$, $h \approx 0.02$); bottom right: isotropic unstructured mesh ($\Omega = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 < 1\}$, $h \approx 0.08$).

Table 1.1 – Smooth validation example ($\varepsilon_1 \approx h^2$). (i) Variations with respect to h of the approximate orthogonality conditions verified by $\nabla u_{1,h}$ and $\nabla u_{2,h}$ (columns 2,3 and 4). (ii) Variations with respect to h of the $L^2(\Omega)$ norm of the computed approximation error $\mathbf{u} - \mathbf{u}_h$ and related convergence orders (columns 5 and 6). (iii) Variations with respect to h of the number of time steps necessary to achieve convergence (column 7). ($\Omega = (0, 1)^2$, structured asymmetric meshes).

h	$\int_{\Omega} \nabla u_{1,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{2,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{1,h} \cdot \nabla u_{2,h} d\mathbf{x}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$	rate	iter
0.02	1.0	1.0	1.89e-6	2.35e-07	-	31
0.01	1.0	1.0	8.94e-7	1.04e-07	1.17	36
0.005	1.0	1.0	4.38e-7	4.93e-08	1.07	40
0.0025	1.0	1.0	2.46e-7	2.72e-08	0.85	43

1.4.2 Simple folding and mesh dependency

For the second experiment, we consider the unit square and the boundary data $\mathbf{g}$ given by:

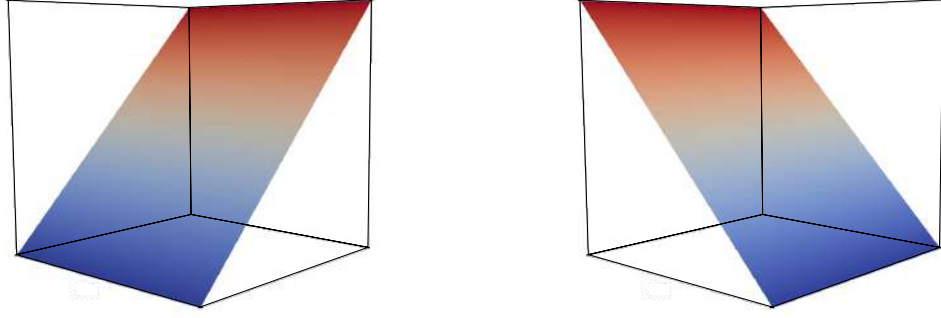


Figure 1.3 – Smooth validation example. Graphs of the two components of the computed approximate solution of (1.1) with $\mathbf{g}(\mathbf{x}) = \mathbf{x}$ on $\partial\Omega$; left: approximation of the first component u_1 ; right: approximation of the second component u_2 . These results have been obtained using an structured asymmetric triangulation with $h = 0.01$.

$$\begin{aligned} g_1(x_1, x_2) &= \begin{cases} x_1 & \text{if } x_1 < 0.5 \text{ and } x_2 = 0 \text{ or } x_2 = 1, \\ 1 - x_1 & \text{if } x_1 \geq 0.5 \text{ and } x_2 = 0 \text{ or } x_2 = 1, \\ 0 & \text{otherwise,} \end{cases} \\ g_2(x_1, x_2) &= x_2 \text{ on } \partial\Omega. \end{aligned} \quad (1.30)$$

In this case, the exact solution to (1.1) in Ω is given by $\mathbf{u} = (u_1, u_2)^T$, with

$$\begin{aligned} u_1(x_1, x_2) &= \begin{cases} x_1 & \text{if } x_1 < 0.5, \\ 1 - x_1 & \text{if } x_1 \geq 0.5, \end{cases} \\ u_2(x_1, x_2) &= x_2. \end{aligned} \quad (1.31)$$

The function $\mathbf{u}$ defined by (1.31) corresponds to a single folding of the domain Ω along the middle line. This implies that the singularity is a line singularity along $x_1 = 0.5$. Thus, when choosing $\varepsilon_1 = 0$, the algorithm obtains a discretized solution that is accurate up to machine precision. In the case of the structured meshes of Figure 1.2, the mesh edges can be aligned with this line singularity or not, depending if the vertical mesh edges are along $x_1 = 0.5$. In the case of unstructured meshes, the edges are never aligned with the line $x_1 = 0.5$. Tables 1.2, 1.3 and 1.4 show results using $\varepsilon_1 = \frac{h^2}{5\Delta t}$. One can observe that the results are more accurate when using a structured mesh with edges aligned with the line singularity; when the line singularity is not aligned with the (structured) mesh edges, the behavior of the algorithm is actually similar to when the mesh is unstructured. Moreover, the number of iterations (time steps) to obtain a stationary solution is larger for the two latter cases. Finally, the convergence to the exact solution is super-linear when $h \rightarrow 0$, but more stable for structured meshes in

Chapter 1. Numerical Approximation of Orthogonal Maps

Table 1.2.

We have visualized on Figure 1.4: (i) The graphs of the two components of the computed approximate solution (first row). (ii) The values of $\det \nabla \mathbf{u}$ (second row (left)). (iii) The image $\mathbf{u}(\Omega) \subset \mathbb{R}^2$ (second row (right)), which shows that the initial domain Ω is folded in half (note that this illustration is post processed from the computed approximations of u_1 and u_2).

Table 1.2 – Simple folding. (i) Variations with respect to h of the approximate orthogonality conditions verified by $\nabla u_{1,h}$ and $\nabla u_{2,h}$ (columns 2,3 and 4). (ii) Variations with respect to h of the $L^2(\Omega)$ norm of the computed approximation error $\mathbf{u} - \mathbf{u}_h$ and related convergence orders (columns 5 and 6). (iii) Variations with respect to h of the number of time steps necessary to achieve convergence (column 7). ($\Omega = (0, 1)^2$, structured asymmetric meshes, line singularity aligned with mesh edges).

h	$\int_{\Omega} \nabla u_{1,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{2,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{1,h} \cdot \nabla u_{2,h} d\mathbf{x}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$	rate	iter
0.02	0.9732	1.0	0.0028	1.87e-03	-	57
0.01	0.9866	1.0	0.0008	6.43e-04	1.54	65
0.005	0.9933	1.0	0.0002	2.22e-04	1.53	72
0.0025	0.9966	1.0	5.92e-05	7.76e-05	1.51	79

Table 1.3 – Simple folding. (i) Variations with respect to h of the approximate orthogonality conditions verified by $\nabla u_{1,h}$ and $\nabla u_{2,h}$ (columns 2,3 and 4). (ii) Variations with respect to h of the $L^2(\Omega)$ norm of the computed approximation error $\mathbf{u} - \mathbf{u}_h$ and related convergence orders (columns 5 and 6). (iii) Variations with respect to h of the number of time steps necessary to achieve convergence (column 7). ($\Omega = (0, 1)^2$, isotropic unstructured meshes).

h	$\int_{\Omega} \nabla u_{1,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{2,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{1,h} \cdot \nabla u_{2,h} d\mathbf{x}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$	rate	iter
0.05	0.9413	1.0	0.0167	6.20e-03	-	48
0.026	0.9700	1.0	0.0024	2.45e-03	1.33	96
0.013	0.9849	1.0	0.0011	1.10e-03	1.15	184
0.006	0.9925	1.0	0.0003	3.13e-04	1.82	117

1.4.3 Double diagonal folding

Let us consider now the unit square $\Omega = (0, 1)^2$ and the boundary data defined by:

$$\mathbf{g}(x_1, x_2) = (0, |x_1 - x_2|), \quad (x_1, x_2) \in \partial\Omega. \quad (1.32)$$

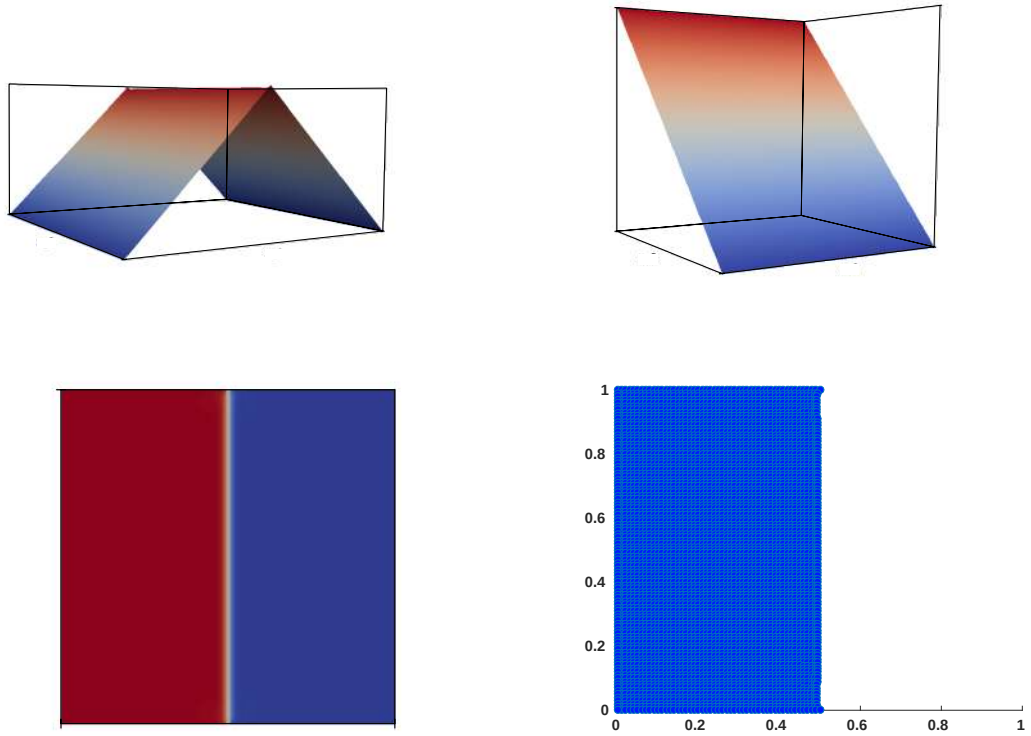


Figure 1.4 – Simple folding. Visualization of the graph of the first and second component of the computed approximate solution $\mathbf{u}_h (= (u_{1,h}, u_{2,h}))$ of (1.1) with $\Omega = (0, 1)^2$ and $\mathbf{g}$ given by (1.30). Top left: First component of the computed approximate solution $u_{1,h}$; top right: Second component of the computed approximate solution $u_{2,h}$; bottom left: visualization of $\det \nabla \mathbf{u}_h$; bottom right: visualization of the domain $\mathbf{u}_h(\Omega)$.

Table 1.4 – Simple folding. (i) Variations with respect to h of the approximate orthogonality conditions verified by $\nabla u_{1,h}$ and $\nabla u_{2,h}$ (columns 2,3 and 4). (ii) Variations with respect to h of the $L^2(\Omega)$ norm of the computed approximation error $\mathbf{u} - \mathbf{u}_h$ and related convergence orders (columns 5 and 6). (iii) Variations with respect to h of the number of time steps necessary to achieve convergence (column 7). ($\Omega = (0, 1)^2$, structured asymmetric meshes, line singularity not aligned with mesh edges).

h	$\int_{\Omega} \nabla u_{1,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{2,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{1,h} \cdot \nabla u_{2,h} d\mathbf{x}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$	rate	iter
0.0196	0.9724	1.0	0.0094	4.72e-03	-	58
0.0099	0.9864	1.0	0.0048	2.44e-03	0.95	66
0.0049	0.9932	1.0	0.0024	1.25e-03	0.96	157
0.0024	0.9966	1.0	0.0027	4.39e-04	1.50	553

A corresponding exact solution is

$$\begin{aligned}
 u_1(x_1, x_2) &= d(\mathbf{x}, \partial\Omega) \quad \forall \mathbf{x} \text{ in } \Omega, \\
 u_2(x_1, x_2) &= \begin{cases} \min(x_2, 1 - x_1) & \text{if } x_1 < x_2, \\ \min(x_1, 1 - x_2) & \text{otherwise.} \end{cases}
 \end{aligned} \tag{1.33}$$

For this test example, the line singularities are aligned with the two diagonals of Ω ($x_1 = x_2$ and $1 - x_1 = x_2$). When using the structured symmetric mesh (see Figure 1.2, top right) and $\varepsilon_1 = 0.0$, the algorithm obtains an approximated solution that is highly accurate, since the singularities are aligned with mesh edges. Indeed after 170 iterations (time steps of the splitting algorithm), the numerical solution satisfies $\|\mathbf{u} - \mathbf{u}_h\|_{L^2(\Omega)} = 1.45 \times 10^{-8}$, $\int_{\Omega} |\nabla u_1| d\mathbf{x} = 1.0$, $\int_{\Omega} |\nabla u_2| d\mathbf{x} = 1.0$, and $\int_{\Omega} |\nabla u_1 \cdot \nabla u_2| d\mathbf{x} = 3.88 \times 10^{-8}$.

Numerical results are reported in Tables 1.5 and 1.6 for the structured asymmetric and unstructured triangulations, respectively. Similar convergence orders are observed for both types of triangulations, since none of them match exactly the singularity of the gradient solution. Figure 1.5 illustrates the approximate solution $\mathbf{u}_h$. One can see that the two components actually satisfy the orthogonality conditions. Figure 1.5 (bottom left) visualizes $\det \nabla \mathbf{u}_h$ and shows that the angle condition is satisfied. Figure 1.5 (bottom right) visualizes the image $\mathbf{u}(\Omega)$ of Ω through the mapping $\mathbf{u}$. Table 1.7 illustrates the dependency of the solution with respect to the smoothing parameter ε_1 . One observes, that, as $\varepsilon_1 \rightarrow 0$, the $L^2(\Omega)$ -approximation error decreases and the orthogonality properties verified by ∇u_1 and ∇u_2 are better satisfied by $\nabla u_{1,h}$ and $\nabla u_{2,h}$. On the other hand, the number of iterations necessary to achieve convergence increases as $\varepsilon_1 \rightarrow 0$. This numerical example shows that the introduction of the regularization term, such that $\varepsilon_1 \neq 0$, lead to, overall, better convergence properties of the time-stepping algorithm. Numerical results have consistently shown that the introduction of this term not only helps the convergence of the time stepping algorithm towards a stationary solution, but also allows to reduce drastically the number of time iterations in some cases.

Table 1.5 – Double diagonal folding. (i) Variations with respect to h of the approximate orthogonality conditions verified by $\nabla u_{1,h}$ and $\nabla u_{2,h}$ (columns 2,3 and 4). (ii) Variations with respect to h of the $L^2(\Omega)$ norm of the computed approximation error $\mathbf{u} - \mathbf{u}_h$ and related convergence orders (columns 5 and 6). (iii) Variations with respect to h of the number of time steps necessary to achieve convergence (column 7) ($\Omega = (0,1)^2$, structured asymmetric meshes).

h	$\int_{\Omega} \nabla u_{1,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{2,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{1,h} \cdot \nabla u_{2,h} d\mathbf{x}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$	rate	iter
0.02	0.9672	0.9615	0.0710	3.86e-03	-	64
0.01	0.9837	0.9804	0.0370	1.55e-03	1.31	71
0.005	0.9918	0.9901	0.0189	6.58e-04	1.23	75
0.0025	0.9959	0.9950	0.0095	2.95e-04	1.15	84

Table 1.6 – Double diagonal folding. (i) Variations with respect to h of the approximate orthogonality conditions verified by $\nabla u_{1,h}$ and $\nabla u_{2,h}$ (columns 2,3 and 4). (ii) Variations with respect to h of the $L^2(\Omega)$ norm of the computed approximation error $\mathbf{u} - \mathbf{u}_h$ and related convergence orders (columns 5 and 6). (iii) Variations with respect to h of the number of time steps necessary to achieve convergence (column 7) ($\Omega = (0,1)^2$, isotropic unstructured meshes).

h	$\int_{\Omega} \nabla u_{1,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{2,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{1,h} \cdot \nabla u_{2,h} d\mathbf{x}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$	rate	iter
0.05	0.9346	0.9279	0.1305	9.93e-03	-	61
0.026	0.9677	0.9618	0.0724	3.93e-03	1.33	83
0.013	0.9840	0.9805	0.0393	1.73e-03	1.18	313
0.006	0.9920	0.9901	0.0208	7.63e-04	1.18	292

The sensitivity of the results with respect to the target function $\mathbf{f}$ is now investigated. When modifying the target function $\mathbf{f}$, the algorithm is able to track different solutions of the orthogonal maps problem. Let us consider $\mathbf{f}_1(x_1, x_2) = (-2000, 0)^T$. Figure 1.6 (left column) illustrates snapshots of the two components of the numerical solution $\mathbf{u}_h$, for $h = 0.01$, and of the determinant $\det \nabla \mathbf{u}_h$, and shows that the first component is the concave up version of the one illustrated in Figure 1.6 when $\mathbf{f} = \mathbf{0}$, while the second component remains unchanged. The numerical solution satisfies

$$\int_{\Omega} |\nabla u_1| d\mathbf{x} = 1.01, \quad \int_{\Omega} |\nabla u_2| d\mathbf{x} = 0.98, \quad \int_{\Omega} |\nabla u_1 \cdot \nabla u_2| d\mathbf{x} = 0.047$$

after 120 iterations. A second variation in the fidelity term is illustrated by a perturbation of the target function $\mathbf{f}$ around the null function $\mathbf{0}$, through a local perturbation around the mid-point $(0.5, 0.5)$. Figure 1.6 (right column) illustrates the snapshots of the two components

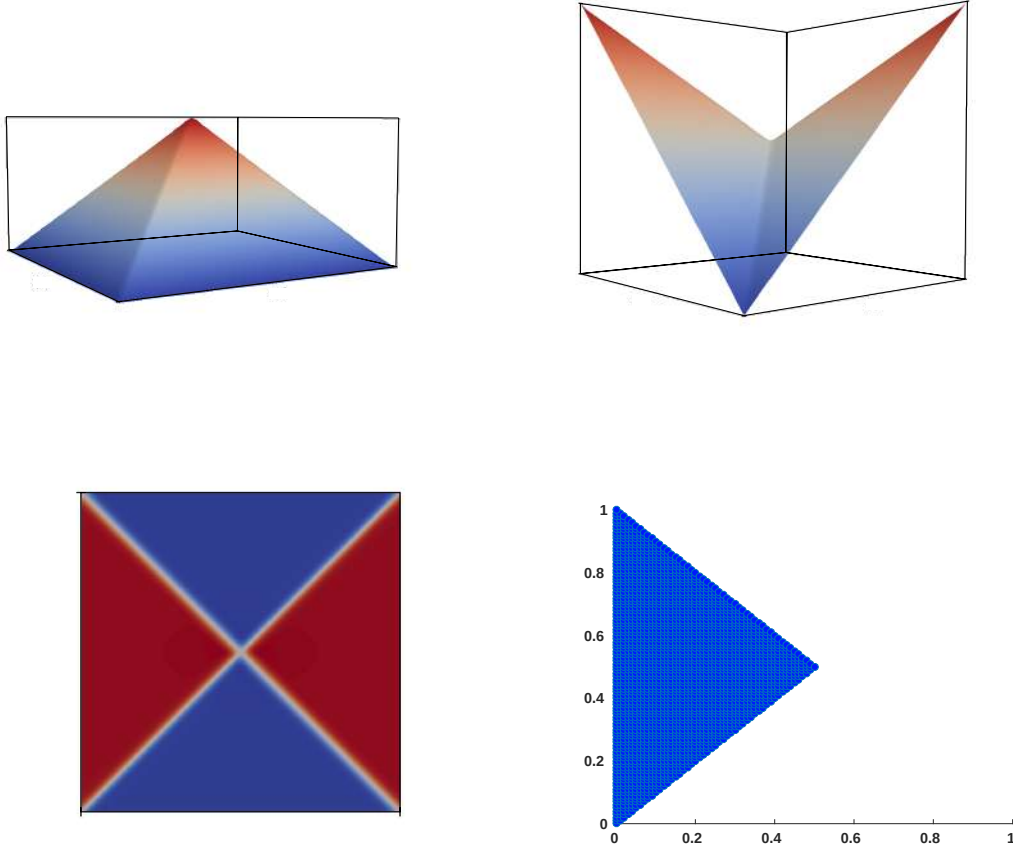


Figure 1.5 – Double diagonal folding. Snapshots of the numerical stationary solution of (1.1) with $\mathbf{g}$ given by (1.32). Top left: First component u_1 ; top right: Second component u_2 ; bottom left: visualization of $\det \nabla \mathbf{u} = \pm 1$; bottom right: visualization of the domain $\mathbf{u}(\Omega)$.

of the numerical solution $\mathbf{u}_h$, and of the determinant $\det \nabla \mathbf{u}_h$, obtained when

$$\mathbf{f}_2(x_1, x_2) = \begin{cases} -2000 & \text{if } (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 2h, \\ 0 & \text{otherwise,} \end{cases}$$

where $h = 0.01$ is the mesh size. The numerical solution satisfies

$$\int_{\Omega} |\nabla u_1| d\mathbf{x} = 0.96, \quad \int_{\Omega} |\nabla u_2| d\mathbf{x} = 0.97, \quad \int_{\Omega} |\nabla u_1 \cdot \nabla u_2| d\mathbf{x} = 0.040$$

after 615 iterations. We can observe that the local perturbation of the target function induces a local perturbation of the approximation of the first component of the solution, which corresponds to an additional folding in the mapping. The sensitivity of the solution with respect to the target function is thus limited to the neighborhood of the perturbation. This is a consequence the low regularity of the, piecewise affine, solutions.

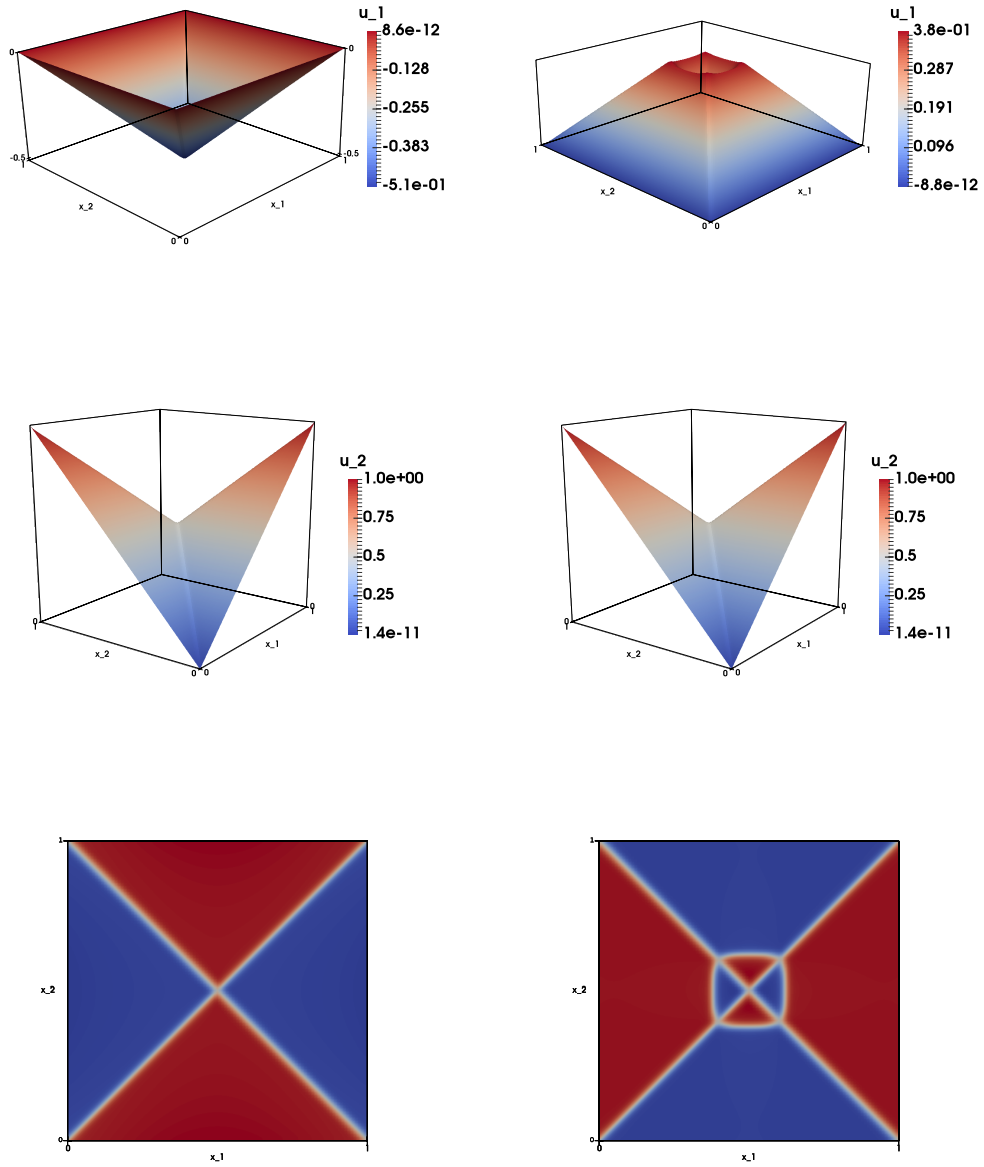


Figure 1.6 – Double diagonal folding. Snapshots of the numerical stationary solution of (1.1) with $\mathbf{g}$ given by (1.32) and non-zero target functions $\mathbf{f}$ (first row: first component: $u_{1,h}$, second row: second component: $u_{2,h}$, third row: $\det \nabla \mathbf{u}_h$). Left column: numerical results when $\mathbf{f} = \mathbf{f}_1$; Right column: numerical results when $\mathbf{f} = \mathbf{f}_2$.

Chapter 1. Numerical Approximation of Orthogonal Maps

Table 1.7 – Double diagonal folding. (i) Variations with respect to ε_1 of the approximate orthogonality conditions verified by $\nabla u_{1,h}$ and $\nabla u_{2,h}$ (columns 2,3 and 4). (ii) Variations with respect to ε_1 of the $L^2(\Omega)$ norm of the computed approximation error $\mathbf{u} - \mathbf{u}_h$ and related convergence orders (column 5). (iii) Variations with respect to ε_1 of the number of time steps necessary to achieve convergence (column 6) ($\Omega = (0, 1)^2$, structured asymmetric mesh, $h = 0.01$).

ε_1	$\int_{\Omega} \nabla u_{1,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{2,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{1,h} \cdot \nabla u_{2,h} d\mathbf{x}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$	iter
2×10^{-3}	0.8337	0.8554	0.2242	2.99e-02	37
10^{-3}	0.8873	0.8880	0.1878	1.87e-02	41
2×10^{-4}	0.9517	0.9444	0.1023	6.32e-03	52
10^{-4}	0.9658	0.9595	0.0758	4.03e-03	55
0.0	0.9960	0.9965	0.0142	1.84e-03	413

1.4.4 A non-smooth example with a point singularity

For this example, we have $\Omega = (0, 1)^2$, and the boundary data $\mathbf{g}$ defined by

$$\begin{aligned} g_1(x_1, x_2) &= \begin{cases} \min(x_1, 1 - x_1) & \text{if } x_2 = 1, \\ 0 & \text{otherwise,} \end{cases} \\ g_2(x_1, x_2) &= \begin{cases} x_2 & \text{if } x_2 \geq x_1 \text{ and } x_1 \leq 0.5, \\ x_2 & \text{if } x_1 > 0.5 \text{ and } x_2 > -x_1 + 1, \\ x_1 & \text{if } x_2 \leq x_1 \text{ and } x_1 \leq 0.5, \\ 1 - x_1 & \text{if } x_2 > x_1 \text{ and } x_2 \leq -x_1 + 1. \end{cases} \end{aligned} \quad (1.34)$$

An exact solution to the related problem (1.1) is

$$\begin{aligned} u_1(x_1, x_2) &= \begin{cases} x_1 & \text{if } x_2 \geq x_1 \text{ and } x_1 \leq 0.5, \\ 1 - x_1 & \text{if } x_1 > 0.5 \text{ and } x_2 > -x_1 + 1, \\ x_2 & \text{if } x_2 \leq x_1 \text{ and } x_2 \leq -x_1 + 1, \end{cases} \\ u_2(x_1, x_2) &= \begin{cases} x_2 & \text{if } x_2 \geq x_1 \text{ and } x_1 \leq 0.5, \\ x_2 & \text{if } x_1 > 0.5 \text{ and } x_2 > -x_1 + 1, \\ x_1 & \text{if } x_2 \leq x_1 \text{ and } x_1 \leq 0.5, \\ 1 - x_1 & \text{if } x_2 > x_1 \text{ and } x_2 \leq -x_1 + 1. \end{cases} \end{aligned} \quad (1.35)$$

The main numerical difficulty here is the point singularity, located at $(0.5, 0.5)$. In the neighborhood of this point, the numerical approximation of $\det \nabla \mathbf{u}$ oscillates between the values $+1$ and -1 , which increases significantly the number of iterations required to achieve convergence.

However, using the structured symmetric mesh, the line singularities are again aligned with the mesh edges, and a very accurate solution is obtained. Indeed, with $\varepsilon_1 = 0$, and after 130 time steps, we have

$$\begin{aligned} \|\mathbf{u} - \mathbf{u}_h\|_{L^2(\Omega)} &= 2.77 \cdot 10^{-10}, \\ \int_{\Omega} |\nabla u_{1,h}| d\mathbf{x} &= 1.0, \quad \int_{\Omega} |\nabla u_{2,h}| d\mathbf{x} = 1.0, \quad \int_{\Omega} |\nabla u_{1,h} \cdot \nabla u_{2,h}| d\mathbf{x} = 1.25 \cdot 10^{-9}. \end{aligned}$$

We obtained the results reported in Table 1.8 using structured asymmetric meshes. These results show a convergence of order $0.9 - 1.0$, approximately. One observes also that the number of time steps required to achieve quasi-stationarity increases with $1/h$; this is due to the oscillatory behavior of function $t \rightarrow \mathbf{u}_h(t)$. Various geometrical aspects of the computed approximate solution have been visualized in Figure 1.7. In particular, Figure 1.7 (bottom left) shows that $\det \nabla \mathbf{u}$ satisfies accurately the angle condition, however, accuracy deteriorates close to the singular point $(0.5, 0.5)$.

Chapter 1. Numerical Approximation of Orthogonal Maps

Table 1.8 – Non-smooth example with point singularity. (i) Variations with respect to h of the approximate orthogonality conditions verified by $\nabla u_{1,h}$ and $\nabla u_{2,h}$ (columns 2,3 and 4). (ii) Variations with respect to h of the $L^2(\Omega)$ norm of the computed approximation error $\mathbf{u} - \mathbf{u}_h$ and related convergence orders (columns 5 and 6). (iii) Variations with respect to h of the number of time steps necessary to achieve convergence (column 7). ($\Omega = (0, 1)^2$, structured asymmetric meshes).

h	$\int_{\Omega} \nabla u_{1,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{2,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{1,h} \cdot \nabla u_{2,h} d\mathbf{x}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$	rate	iter
0.02	0.9703	0.9667	0.0422	5.71e-03	-	333
0.01	0.9852	0.9831	0.0219	2.89e-03	0.98	675
0.005	0.9926	0.9915	0.0113	1.52e-03	0.93	1438
0.0025	0.9963	0.9957	0.0058	7.88e-04	0.94	3316

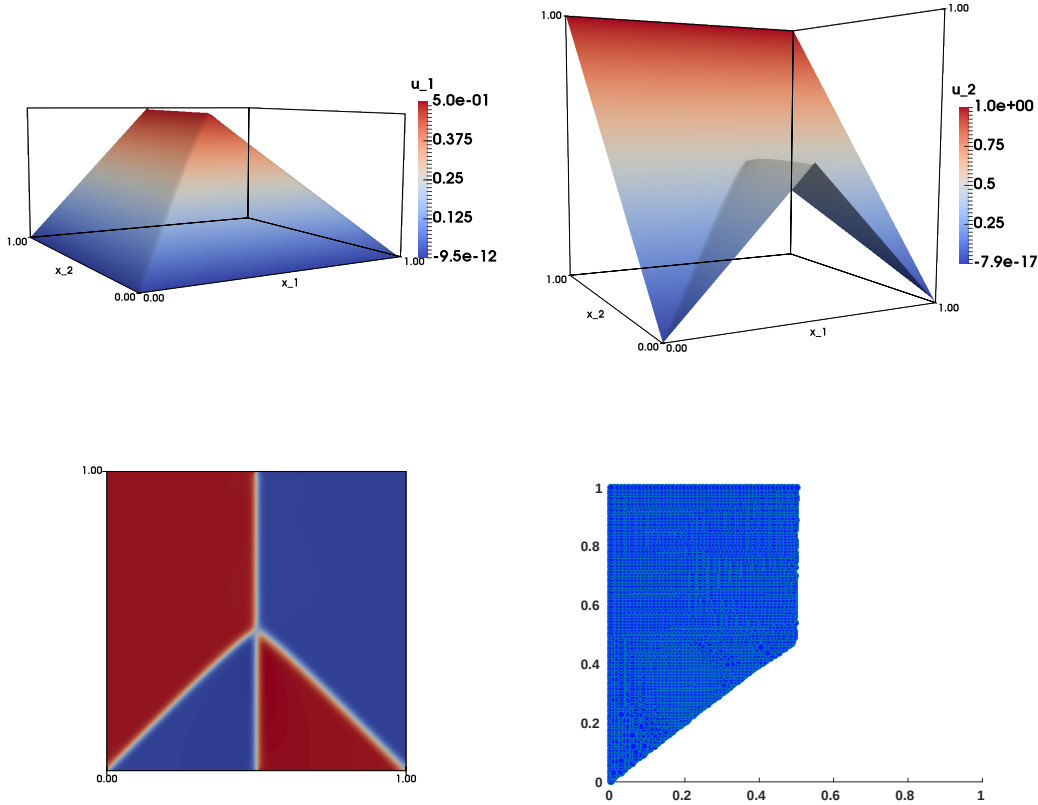


Figure 1.7 – Non-smooth example with point singularity. Visualization of the solution $\mathbf{u}_h$ of problem (1.1) with $\mathbf{g}$ given by (1.34). Top left: component $u_{1,h}$; top right: component $u_{2,h}$; bottom left: visualization of $\det \nabla \mathbf{u}_h$; bottom right: visualization of the domain $\mathbf{u}_h(\Omega)$.

1.4.5 Curved boundaries

Finally, in order to investigate the capabilities of our methodology at handling domains Ω with curved boundaries, we consider the particular problem (1.1) where Ω is the open unit disk centered at $(0,0)$, and $\mathbf{g}$ is the restriction to $\partial\Omega$ of the vector-valued function $\mathbf{u}$ defined by

$$\begin{aligned} u_1(x_1, x_2) &= \begin{cases} x_1 & \text{if } x_1 < 0.5, \\ 1 - x_1 & \text{if } x_1 \geq 0.5, \end{cases} \\ u_2(x_1, x_2) &= \begin{cases} x_2 & \text{if } x_2 < 0.5, \\ 1 - x_2 & \text{if } x_2 \geq 0.5. \end{cases} \end{aligned} \quad (1.36)$$

with $(x_1, x_2) \in \Omega$. The function $\mathbf{u}$ we just defined is an exact solution to the above problem. Actually, this solution corresponds to a double folding: one folding along the horizontal diameter of Ω (Ox_1 axis) and one folding along the vertical diameter of Ω (Ox_2 axis). For various values of h , we have reported in Table 1.9, the results we obtained, taking $\varepsilon_1 = \frac{h^2}{5\Delta t}$. These results (obtained using isotropic unstructured meshes) show first order convergence (actually, slightly better than first). They show also that the gradients of the components $u_{1,h}$ and $u_{2,h}$ of $\mathbf{u}_h$ verify accurately the orthogonality properties of $\nabla u_{1,h}$ and $\nabla u_{2,h}$. Geometrical aspects of $\mathbf{u}_h$ have been visualized in Figure 1.8, $\mathbf{u}_h(\Omega_h)$ in particular (in Figure 1.8 (bottom right)), Ω_h being a polygonal approximation of Ω .

Table 1.9 – Double folding of the unit disk. Variations with h of the: (i) Approximate orthogonality conditions verified by $\nabla u_{1,h}$ and $\nabla u_{2,h}$ (columns 2,3 and 4). (ii) $L^2(\Omega)$ -norm of the approximation error $\mathbf{u} - \mathbf{u}_h$ and related convergence rates (columns 5 and 6). (iii) Number of iterations necessary to achieve convergence (column 7). These results have been obtained using an isotropic unstructured triangulation of the unit disk.

h	$\int_{\Omega} \nabla u_{1,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{2,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{1,h} \cdot \nabla u_{2,h} d\mathbf{x}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$	rate	iter
0.05	0.9623	0.9623	0.0719	1.27e-02	-	75
0.025	0.9807	0.9807	0.0430	5.76e-03	1.03	379
0.013	0.9902	0.9902	0.0217	2.72e-03	1.08	557
0.006	0.9949	0.9949	0.0110	1.11e-03	1.29	427

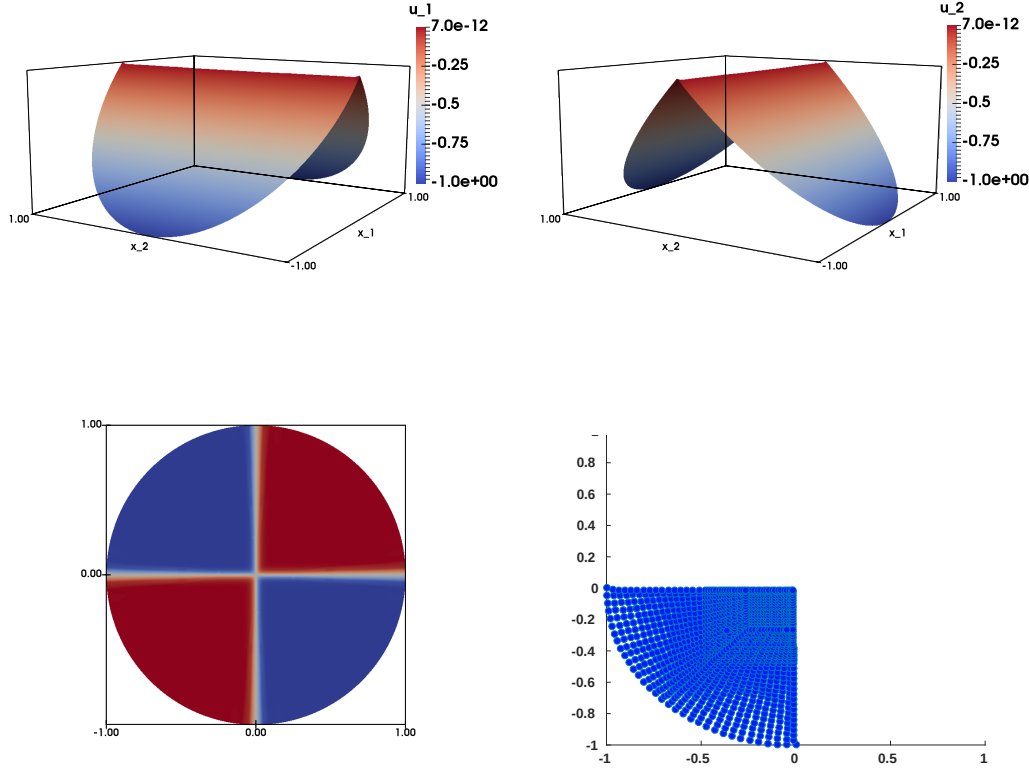


Figure 1.8 – Double folding of the unit disk. Visualization of the computed approximate solution $\mathbf{u}_h = (u_{1,h}, u_{2,h})$ of problem (1.1) (1.32), Ω being the unit disk centered at $(0, 0)$. Top left: Graph of $u_{1,h}$; top right: Graph of $u_{2,h}$; bottom left: visualization of $\det \nabla \mathbf{u}_h$; bottom right: visualization (post-processing) of the domain $\mathbf{u}_h(\Omega_h)$. These results have been obtained using an isotropic unstructured triangulation with $h = 0.08$.

1.5 The Dirichlet problem with homogeneous boundary conditions

In this section, we will elaborate on the solution of the orthogonal map problem (1.1) with homogeneous Dirichlet boundary conditions, that is: Find a vector-valued function $\mathbf{u} (= [u_1, u_2]^T) : \Omega \rightarrow \mathbb{R}^2$ verifying

$$\begin{cases} \nabla \mathbf{u} \in \mathcal{O}(2) & \text{in } \Omega, \\ \mathbf{u} = \mathbf{0} & \text{on } \partial\Omega. \end{cases} \quad (1.37)$$

The choice of homogeneous boundary conditions actually corresponds to the case where the image of the whole boundary $\partial\Omega$ is the single point $\mathbf{0} = (0, 0)$. If the condition $\nabla \mathbf{u} \nabla \mathbf{u}^T = \mathbf{I}$ is incompatible with the homogeneous boundary condition (as it is the case if Ω is a rectangle), the solution of the regularized least-squares problem (1.9) will develop a fractal behavior near the boundary Ω [Dacorogna and Marcellini, 1999; Dacorogna et al., 2010b], in a similar fashion

1.5. The Dirichlet problem with homogeneous boundary conditions

as in [Caboussat et al., 2015; Dacorogna et al., 2004; Glowinski, 2015b] for a scalar Eikonal equation. Such a fractal behavior makes the convergence of our time-stepping method not possible, and makes this example a very stringent one.

For this test example, we use a uniform asymmetric triangulation with $h = 1/200$ and $\varepsilon_1 = 0$. In Figure 1.9 is visualized the numerical solution. A close inspection shows that the angle condition is not satisfied everywhere, therefore a another approach must be favored.

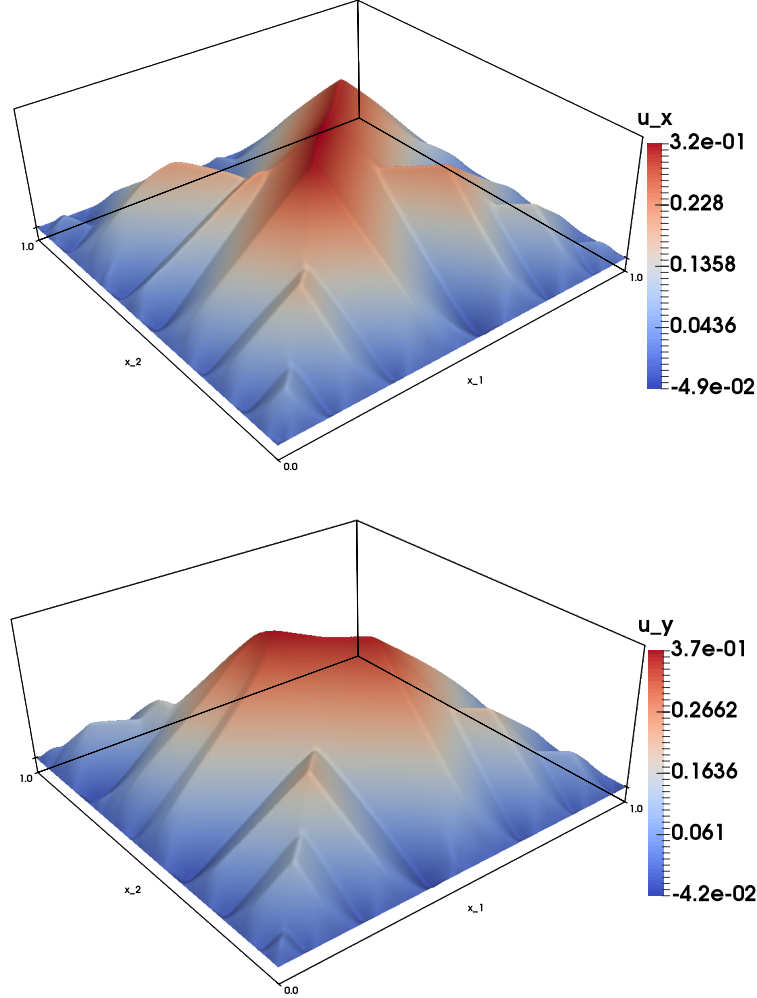


Figure 1.9 – Homogeneous orthogonal map-Dirichlet problem (1.37) with $\Omega = (0, 1)^2$. Visualization of the approximate solutions computed without and with mesh adaptation. Top: component $u_{1,h}$. Bottom: component $u_{2,h}$. Results obtained using a uniform asymmetric triangulation with 40,401 vertices and 80,000 triangles.

We are going to describe a first approach to solve (1.37). The approach would be to first fold the domain Ω into the segment, e.g. $\{0\} \times [0, 1]$. This could be achieved by first folding repeatedly

along the axis Ox_2 , as illustrated in Figure 1.10 when folding once, twice and three times.

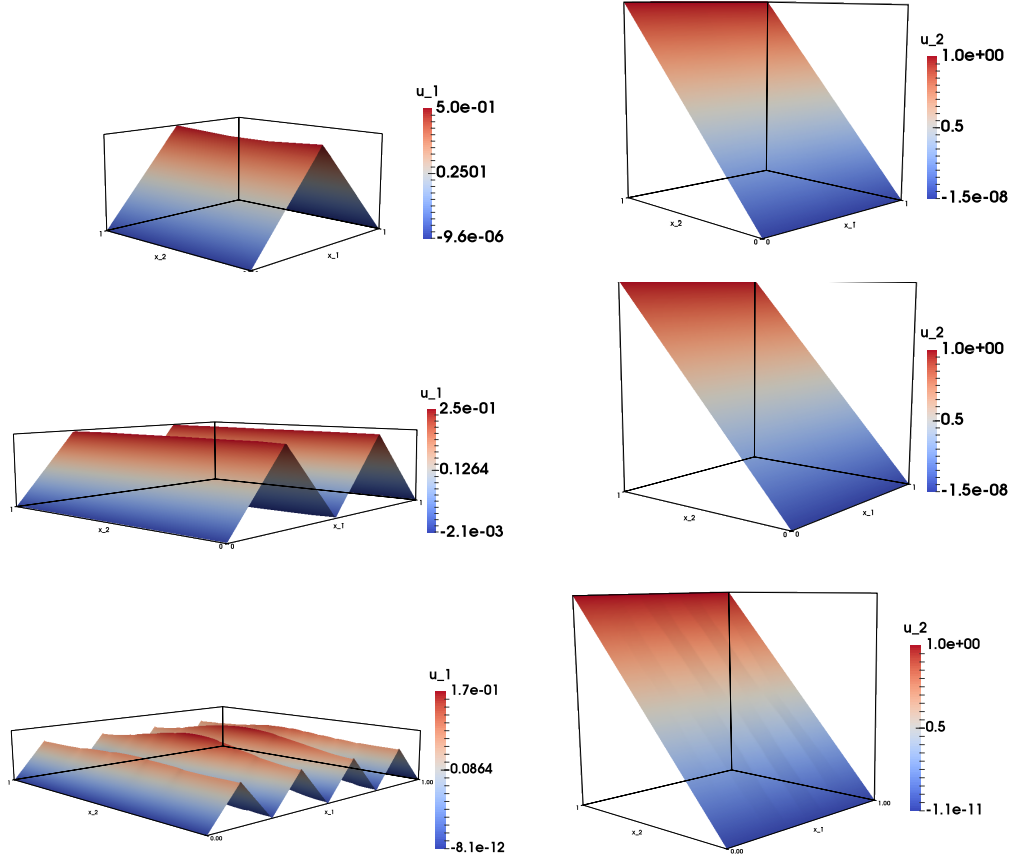


Figure 1.10 – Multiple folding for the construction of a sequence of numerical approximations of (1.1), with boundary data converging to $\mathbf{g} = \mathbf{0}$. Snapshots of the numerical stationary solution (left: first component: $u_{1,h}$, right: second component: $u_{2,h}$), when folding once, twice or three times along one axis (first through third row), and when folding three times along one axis, and once along the other axis (fourth row).

However, the limit solution obtained when proceeding like this iteratively will not be a solution to (1.37) as the solution mapping is the null mapping, with a range restricted to the point $(0,0)$. In the end of this Chapter we present an effective approach that can solve (1.37). We see in Figure (1.10), the method has a limitation when it comes to solutions of complicated problems like (1.37). In order to increase the robustness of the method, we equip it with adaptive techniques, as presented in the next section.

1.6 An anisotropic adaptive algorithm

Following [Picasso, 2003a,b], our goal is now to build an anisotropic mesh such that the estimated relative error is close to a preset tolerance TOL , namely:

$$0.75 \, TOL \leq \frac{\eta^{A,n+1}}{\|\nabla \mathbf{u}_h^{n+1}\|_{L^2(\Omega)}} \leq 1.25 \, TOL, \quad (1.38)$$

where the anisotropic error estimation $\eta^{A,n+1}$ is described hereafter.

In order to construct the anisotropic error estimate $\eta^{A,n+1}$, we set the smoothing term $\varepsilon_1 = 0$, since the adaptive algorithm is dedicated to capturing non-smooth solutions and $C = 0$.

Therefore, (1.23) writes: find $\mathbf{u}^{n+1} \in \mathbf{V}_{\mathbf{g}}$

$$\int_{\Omega} \nabla \mathbf{u}^{n+1} : \nabla \mathbf{v} \, d\mathbf{x} = \int_{\Omega} \mathbf{p}^{n+1/2} : \nabla \mathbf{v} \, d\mathbf{x}, \quad \forall \mathbf{v} \in (H_0^1(\Omega))^2 \quad (1.39)$$

and the corresponding discrete equation from (1.29) as: find $\mathbf{u}_h^{n+1} \in \mathbf{V}_{\mathbf{g},h}$

$$\int_{\Omega_h} \nabla \mathbf{u}_h^{n+1} : \nabla \mathbf{v}_h \, d\mathbf{x} = \int_{\Omega_h} \mathbf{p}_h^{n+1/2} : \nabla \mathbf{v}_h \, d\mathbf{x}, \quad \forall \mathbf{v}_h \in \mathbf{V}_{0,h}. \quad (1.40)$$

To simplify the notation from now on we denote $\mathbf{u} := \mathbf{u}(t^{n+1})$, $\mathbf{u}_h := \mathbf{u}_h^{n+1}$, $\mathbf{p} := \mathbf{p}^{n+1/2}$ and $\mathbf{p}_h := \mathbf{p}_h^{n+1/2}$

Before constructing $\eta^{A,n+1}$, let us first recall some required geometrical definitions of the finite element method. For any triangle K of the discretization $\mathcal{T}_h$, let $T_K : \hat{K} \rightarrow K$ be the affine transformation which maps the reference triangle $\hat{K}$ into K . Let M_K be the Jacobian of the mapping T_K . Since M_K is invertible, it admits a singular value decomposition $M_K = R_K^T \Lambda_K P_K$, where R_K and P_K are orthogonal and where Λ_K is diagonal with positive entries. In the following, we set

$$\Lambda_K = \begin{pmatrix} \lambda_{1,K} & 0 \\ 0 & \lambda_{2,K} \end{pmatrix}, \quad R_K = \begin{pmatrix} \mathbf{r}_{1,K}^T \\ \mathbf{r}_{2,K}^T \end{pmatrix}, \quad (1.41)$$

with the choice $\lambda_{1,K} \geq \lambda_{2,K}$. These geometrical quantities are illustrated in Figure 1.11.

Moreover, let us recall some interpolation results from [Formaggia and Perotto, 2001]. Let $I_h : (H^1(\Omega))^2 \rightarrow \mathbf{V}_h$ be the Clément interpolant [Clément, 1975; Carstensen, 2006]; there is a constant $C_0 = C_0(\hat{K})$ such that, $\forall \mathbf{v} \in (H^1(\Omega))^2$ and $\forall K \in \mathcal{T}_h$ we have

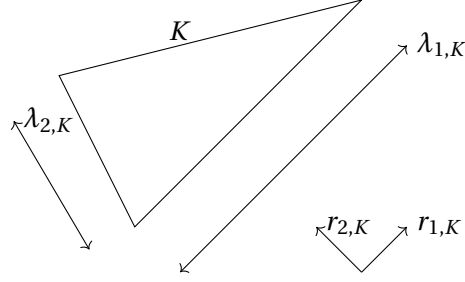


Figure 1.11 – Reference element, main directions, and element dimensions

$$\|\mathbf{v} - I_h \mathbf{v}\|_{L^2(K)} \leq C_0 \left(\lambda_{1,K}^2 (\mathbf{r}_{1,K}^T G_K(\mathbf{v}) \mathbf{r}_{1,K}) + \lambda_{2,K}^2 (\mathbf{r}_{2,K}^T G_K(\mathbf{v}) \mathbf{r}_{2,K}) \right)^{1/2}, \quad (1.42)$$

$$\|\mathbf{v} - I_h \mathbf{v}\|_{L^2(\partial K)} \leq C_0 h_K^{1/2} \left(\frac{\lambda_{1,K}}{\lambda_{2,K}} (\mathbf{r}_{1,K}^T G_K(\mathbf{v}) \mathbf{r}_{1,K}) + \frac{\lambda_{2,K}}{\lambda_{1,K}} (\mathbf{r}_{2,K}^T G_K(\mathbf{v}) \mathbf{r}_{2,K}) \right)^{1/2}, \quad (1.43)$$

where the error gradient matrix $G_K(\cdot)$ is defined as:

$$G_K(\mathbf{v}) = \sum_{T \in \Delta K} \begin{pmatrix} \int_T \left(\frac{\partial \mathbf{v}}{\partial x_1} \right)^2 d\mathbf{x} & \int_T \frac{\partial \mathbf{v}}{\partial x_1} \frac{\partial \mathbf{v}}{\partial x_2} d\mathbf{x} \\ \int_T \frac{\partial \mathbf{v}}{\partial x_1} \frac{\partial \mathbf{v}}{\partial x_2} d\mathbf{x} & \int_T \left(\frac{\partial \mathbf{v}}{\partial x_2} \right)^2 d\mathbf{x} \end{pmatrix},$$

with ΔK to be set as all triangles T that have a vertex common with K .

The anisotropic error estimate is based on the residual. Thus we can show for $\mathbf{u}$ and $\mathbf{u}_h$, we have

$$\begin{aligned} \|\nabla(\mathbf{u} - \mathbf{u}_h)\|_{L^2(\Omega)}^2 &= \int_{\Omega} \mathbf{p} : \nabla(\mathbf{u} - \mathbf{u}_h) d\mathbf{x} - \int_{\Omega} \nabla \mathbf{u}_h : \nabla(\mathbf{u} - \mathbf{u}_h) d\mathbf{x}, \\ &= \int_{\Omega} (\mathbf{p}_h - \nabla \mathbf{u}_h) : \nabla(\mathbf{u} - \mathbf{u}_h) d\mathbf{x} + \int_{\Omega} (\mathbf{p} - \mathbf{p}_h) : \nabla(\mathbf{u} - \mathbf{u}_h) d\mathbf{x} \\ \frac{1}{2} \|\nabla(\mathbf{u} - \mathbf{u}_h)\|_{L^2(\Omega)}^2 &\leq \int_{\Omega} (\mathbf{p}_h - \nabla \mathbf{u}_h) : \nabla(\mathbf{u} - \mathbf{u}_h) d\mathbf{x} + \frac{1}{2} \|\mathbf{p} - \mathbf{p}_h\|_{L^2(\Omega)}^2 \end{aligned}$$

Taking $\mathbf{v} = \mathbf{u} - \mathbf{u}_h$ and using (1.40), we have

$$\begin{aligned} \frac{1}{2} \|\nabla(\mathbf{u} - \mathbf{u}_h)\|_{L^2(\Omega)}^2 &\leq \int_{\Omega} (\mathbf{p}_h - \nabla \mathbf{u}_h) \nabla(\mathbf{v} - I_h \mathbf{v}_h) d\mathbf{x}, \\ &\leq \sum_{K \in \mathcal{T}_h} \left(\int_K \operatorname{div}(\nabla \mathbf{u}_h - \mathbf{p}_h) (\mathbf{v} - I_h \mathbf{v}_h) d\mathbf{x} + \frac{1}{2} \int_{\partial K} [(\nabla \mathbf{u}_h - \mathbf{p}_h) \cdot \mathbf{n}] (\mathbf{v} - I_h \mathbf{v}_h) d\mathbf{x} \right) \\ &\quad + \frac{1}{2} \|\mathbf{p} - \mathbf{p}_h\|_{L^2(\Omega)}^2, \end{aligned}$$

where $[\cdot]$ denotes the jump of the bracketed quantity across an internal edge ($[\cdot] = 0$ for an

edge on the boundary $\partial\Omega$). Using Cauchy-Schwarz inequality and the interpolation results in (1.42), we get

$$\begin{aligned} \frac{1}{2} \|\nabla(\mathbf{u} - \mathbf{u}_h)\|_{L^2(\Omega)}^2 &\leq C_0 \sum_{K \in \mathcal{T}_h} \left(\|\operatorname{div}(\nabla \mathbf{u}_h - \mathbf{p}_h)\|_{L^2(K)} + \frac{1}{2} \left(\frac{h_K}{\lambda_{1,K} \lambda_{2,K}} \right)^{1/2} \|[(\nabla \mathbf{u}_h - \mathbf{p}_h) \cdot \mathbf{n}]\|_{L^2(\partial K)} \right) \\ &\quad \times \left(\lambda_{1,K}^2 (\mathbf{r}_{1,K}^T G_K(\mathbf{v}) \mathbf{r}_{1,K}) + \lambda_{2,K}^2 (\mathbf{r}_{2,K}^T G_K(\mathbf{v}) \mathbf{r}_{2,K}) \right)^{1/2} + \frac{1}{2} \|\mathbf{p} - \mathbf{p}_h\|_{L^2(\Omega)}^2, \end{aligned}$$

then it is sufficient to take $\lambda_{2,K} h_{\hat{K}} \leq h_K \leq \lambda_{1,K} h_{\hat{K}}$ to obtain

$$\begin{aligned} \frac{1}{2} \|\nabla(\mathbf{u} - \mathbf{u}_h)\|_{L^2(\Omega)}^2 &\leq C_0 \sum_{K \in \mathcal{T}_h} \left(\|\operatorname{div}(\nabla \mathbf{u}_h - \mathbf{p}_h)\|_{L^2(K)} + \frac{1}{2 \lambda_{2,K}^{1/2}} \|[(\nabla \mathbf{u}_h - \mathbf{p}_h) \cdot \mathbf{n}]\|_{L^2(\partial K)} \right) \\ &\quad \times \left(\lambda_{1,K}^2 (\mathbf{r}_{1,K}^T G_K(\mathbf{v}) \mathbf{r}_{1,K}) + \lambda_{2,K}^2 (\mathbf{r}_{2,K}^T G_K(\mathbf{v}) \mathbf{r}_{2,K}) \right)^{1/2} + \frac{1}{2} \|\mathbf{p} - \mathbf{p}_h\|_{L^2(\Omega)}^2. \end{aligned}$$

the first term of the residual could practically become negligible when using piecewise linear finite elements

The first term of the residual is zero since we use $\mathbb{P}_1$ polynomials. Moreover, we recall that $\mathbf{p}$ and $\mathbf{p}_h$ are the solutions of (1.20) and (1.27), respectively. Therefore we neglect $\|\mathbf{p} - \mathbf{p}_h\|_{L^2(\Omega)}^2$ term because the approximation $\mathbf{p}_h$ is solved exactly. We conclude by replacing $\mathbf{v}$ to $\mathbf{u} - \mathbf{u}_h$ and we define our error estimate by:

$$(\eta^{A,n+1})^2 = \sum_{K \in \mathcal{T}_h} \left(\eta_K^{A,n+1} \right)^2 \text{ where } \left(\eta_K^{A,n+1} \right)^2 = \rho_K(\mathbf{u}_h) \times \omega_K(\mathbf{u} - \mathbf{u}_h),$$

with

$$\rho_K(\mathbf{u}_h) = \frac{1}{2 \lambda_{2,K}^{1/2}} \|[(\nabla \mathbf{u}_h - \mathbf{p}_h) \cdot \mathbf{n}]\|_{L^2(\partial K)}$$

and

$$\begin{aligned} \omega_K(\mathbf{u} - \mathbf{u}_h)^2 &= \lambda_{1,K}^2 (\mathbf{r}_{1,K}^T G_K(\mathbf{u} - \mathbf{u}_h) \mathbf{r}_{1,K}) \\ &\quad + \lambda_{2,K}^2 (\mathbf{r}_{2,K}^T G_K(\mathbf{u} - \mathbf{u}_h) \mathbf{r}_{2,K}). \end{aligned}$$

The estimate $\eta_K^{A,n+1}$ still contains the exact solution $\mathbf{u}$ in $\omega_K(\mathbf{u} - \mathbf{u}_h)$. In order to overcome this we apply the Zienkiewicz-Zhu (ZZ) post-processing technique to approximate $G_K(\mathbf{u} - \mathbf{u}_h)$, see [Zienkiewicz and Zhu, 1987, 1992]. Thus, we estimate

$$\frac{\partial(\mathbf{u} - \mathbf{u}_h)}{\partial x_i} \text{ by } \Pi_h^{ZZ} \frac{\partial \mathbf{u}_h}{\partial x_i} - \frac{\partial \mathbf{u}_h}{\partial x_i}, \quad i = 1, 2$$

where, for any $\mathbf{v}_h \in \mathbf{V}_h$, and for any vertex P of the mesh

$$\Pi_h^{ZZ} \frac{\partial \mathbf{v}_h}{\partial x_i}(P) = \frac{\sum_{K \in \mathcal{T}_h} |K| \left. \frac{\partial \mathbf{v}_h}{\partial x_i} \right|_K}{\sum_{K \in \mathcal{T}_h} |K|}$$

is an approximate $L^2(\Omega)$ projection of $\partial \mathbf{v}_h / \partial x_i$ onto $\mathbf{V}_h$.

Ultimately, we use BL2D mesh generator [Laug and Borouchaki, 1996] to reconstruct an adapted mesh at each iteration. It requires a metric to be given at the vertices of $\mathcal{T}_h$ for the update of the mesh, and thus the anisotropic error estimate on the elements is translated into an error estimate on each node of the mesh, as detailed, e.g., in [Picasso, 2003a].

1.6.1 General algorithm

The time splitting algorithm (1.26) (1.27) is revisited with additional mesh adaptivity techniques. Unlike what has been done in [Bourgault et al., 2009; Hassan and Picasso, 2015], the algorithm relies on the fact that the mesh is refined at each time step. When the stationary solution is reached, it is stabilized by performing additional iterations with the final adapted mesh. Let us consider given values of Δt , ε_1 and ε_2 ; and let us denote by n_{mesh} the number of time iterations achieved with mesh adaptation at each time step, and by n_{final} the (maximal) number of additional time steps performed with the given final adapted mesh. The general time stepping algorithm is sketched as follows:

- ▷ Set given initial conditions, with the initial finite element mesh $\mathcal{T}_h^0$.
- ▷ For $n = 0, 1, \dots, n_{mesh}$, do
 1. **(Local optimization)** Solve the local algebraic optimization problems (1.26) at each grid point of $\mathcal{T}_h^n$;
 2. **(Variational)** Solve the global linear variational problem (1.29) with the current discretization $\mathcal{T}_h^n$;
 3. **(Adapt)** Update the finite element mesh $\mathcal{T}_h^n \rightarrow \mathcal{T}_h^{n+1}$;
- ▷ For $n = n_{mesh}, \dots, n_{final}$ (or until $\|\mathbf{u}_h^{n+1} - \mathbf{u}_h^n\|_{0h} < 10^{-7}$), solve (1.26) and (1.27) on the fixed final adapted mesh $\mathcal{T}_h^{n_{mesh}}$.

Numerical experiments have shown that adapting the finite element mesh at each time step helps to converge faster to a stationary solution. This effect is documented in the next section.

1.7 Numerical experiments

We present some test cases to illustrate the convergence of our algorithm, and to perform a sensitivity analysis. For all examples considered, the computational domain is the unit square $\Omega = (0, 1)^2$, and the parameters in (1.19)–(1.20) are given by $\varepsilon_2 = 10^{-11}$ and $\Delta t = \varepsilon_2/2$. Although to construct the estimator we used $\varepsilon = 0$, in the numerical section we conduct experiments for various values of it. Because of the low regularity of the solution, small oscillations are created when the mesh is adapted and the stationary solution is difficult to catch. Therefore, the stopping criterion for the iterative method reads as follows: the number of time steps is fixed as $n_{mesh} = 450$ and $n_{final} = 500$ for all test cases. All error indicators are averaged over the last 200 time steps in the tables detailing the convergence behavior of the algorithm. Finally, the results on the effectivity index are given with a margin of error ($\pm$ standard deviation).

1.7.1 Single fold

The first example corresponds to a single fold, for which the singular set consists of the segment $\{1/2\} \times [0, 1]$. The exact solution of this problem is

$$u_1(x_1, x_2) = \begin{cases} x_1 & \text{if } x_1 < 0.5, \\ 1 - x_1 & \text{if } x_1 \geq 0.5, \end{cases} \quad \forall (x_1, x_2) \in \Omega,$$

$$u_2(x_1, x_2) = x_2,$$

and the boundary conditions are defined accordingly. Figure 1.12 illustrates a snapshot of the stationary solution, together with an illustration of an adaptive mesh (in that case, $h_{\min} = 4.53 \cdot 10^{-3}$, $h_{\max} = 9.81 \cdot 10^{-1}$, $TOL = 0.03125$, $\varepsilon_1 = 0$). We can observe that the orthogonality conditions are accurately satisfied ($\int_{\Omega} |\nabla u_{1,h}| = 1.000407$, $\int_{\Omega} |\nabla u_{2,h}| = 1.000000$, and $\int_{\Omega} \nabla u_{1,h} \cdot \nabla u_{2,h} = 0.0001$).

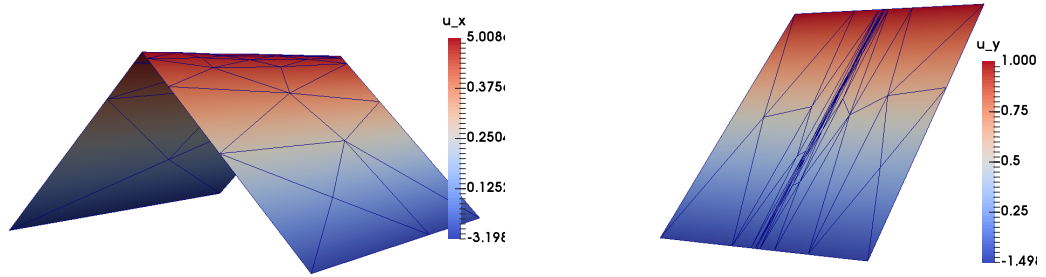


Figure 1.12 – Single folding. Snapshots of the approximated solution (left: first component $u_{1,h}$; right: second component $u_{2,h}$), with illustration of the final adapted mesh.

The method we advocate adapts the mesh at each time iteration. Figure 1.13 illustrates the evolution of the finite element discretization after 20, 21 and 40 time iterations. Even though

the transient solution at one given time step is not accurate and has not converged yet to the stationary solution, the mesh refinement allows to track for the singularities and obtain more robust convergence properties of the global outer loop algorithm.

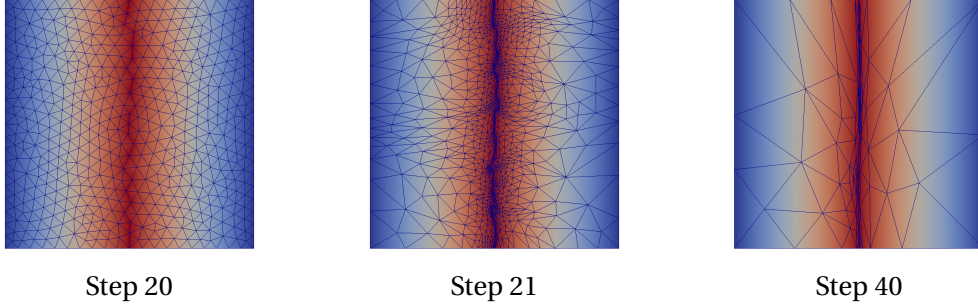


Figure 1.13 – Iterative mesh adaptation within the time stepping algorithm ($\Delta t = 0.5 \cdot 10^{-11}$, $\varepsilon_1 = 0.0$, $TOL = 0.03125$). Left: mesh at time step $n = 20$; middle: mesh at time step $n = 21$; right: mesh at time step $n = 40$.

Figures 1.14 and 1.15 illustrate the snapshots of the solution for various values of TOL ($\varepsilon_1 = 0$ fixed), and various values of ε_1 (TOL fixed). The conclusions are the following: i) Figure 1.14 shows that the smaller the tolerance, the thinner the region where elements are generated along the discontinuity line, and the larger the number of those elements; ii) Figure 1.15 shows that, for a given tolerance, if ε_1 is too large the solution becomes very smooth and thus the anisotropic mesh refinement is not accurate anymore as there is no privileged direction in the solution. Actually, when ε_1 becomes smaller, the mesh converges to the same mesh as when $\varepsilon_1 = 0$. Numerical results confirm that the jump terms in $\rho_K(\mathbf{u}_h)$ and $\Omega_K(\mathbf{u} - bu_h)$ in (1.44) are the crucial ones for mesh adaptation.

Table 1.10 shows the numerical behavior of the algorithm for varying parameters, namely the values of the parameters and final adapted meshes for all tolerances and for $\varepsilon_1 = 2 \cdot 10^{-6}$ and $\varepsilon_1 = 0$. The mesh sizes are defined as $h_{min} = \min_{K \in \mathcal{T}_h} \lambda_{1,K}$ and $h_{max} = \max_{K \in \mathcal{T}_h} \lambda_{2,K}$. The aspect ratio AR represents the maximal aspect ratio defined as $AR = \max_{K \in \mathcal{T}_h} \frac{\lambda_{1,K}}{\lambda_{2,K}}$.

As illustrated, the aspect ratio, the number of elements and the number of nodes increase when the tolerance decreases. The number of elements and nodes is larger when $\varepsilon_1 = 0$, showing that the convergence of the mesh adaptive algorithm is more difficult to reach. On the other hand, for the same number of time iterations, the L^2 -error is smaller when $\varepsilon_1 = 0$.

We observe that the effectivity index becomes smaller than one when the tolerance decreases, meaning that the H^1 -error is not bounded by the estimator η_K^A . This effect will be observed in all numerical experiments in the sequel. This behavior means that a contribution is missing in the estimator to have optimal convergence orders. This effect is actually expected, since we are not incorporating neither the splitting error, nor the contribution from the nonlinear operators in the estimator η_K^A , but only the linear variational operator. Table 1.11 shows an appropriate

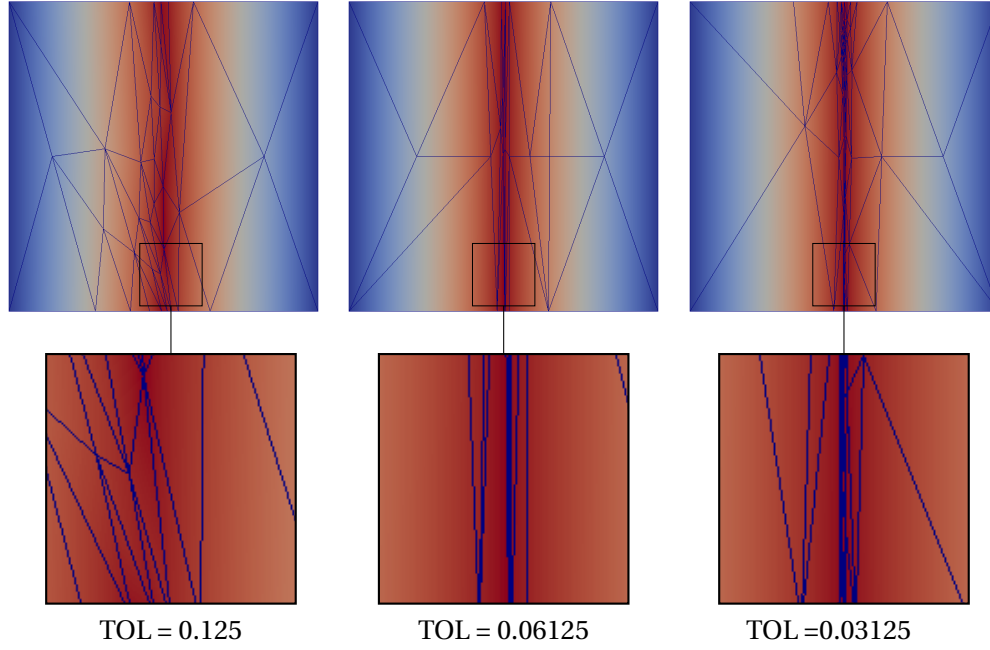


Figure 1.14 – Single folding. Snapshots of the final adapted mesh after 500 time iterations, for various values of the tolerance TOL ($\varepsilon_1 = 0.0$). The colormap represents the values of the first component $u_{1,h}$. The second row corresponds to a zoom in the squared region indicated in the first row.

convergence behavior for the orthogonality conditions, with even some super-convergence behavior in some cases.

Finally, Figure 1.16 illustrates the iterative behavior of the time-stepping algorithm for various tolerances when $\varepsilon_1 = 0$; it shows that the time evolution of indicators is indeed oscillating when the mesh is adapted, due to the low regularity of the solution. The top left figure shows the time evolution of the number of elements; the top right figure shows the time evolution of the error $\|\mathbf{u} - \mathbf{u}_h\|_{L^2(\Omega)}$; both allow to conclude to the convergence of the algorithm when the tolerance decreases. The bottom left figure shows the relationship between the error $\|\mathbf{u} - \mathbf{u}_h\|_{L^2(\Omega)}$ and the number of elements (at each time iteration). It shows that, in average in time, a smaller tolerance leads to a smaller error and a larger number of elements, and the relationship seems to be linear but with a high variability. The bottom right figure illustrates the behavior of $h_{\min}$ versus time.

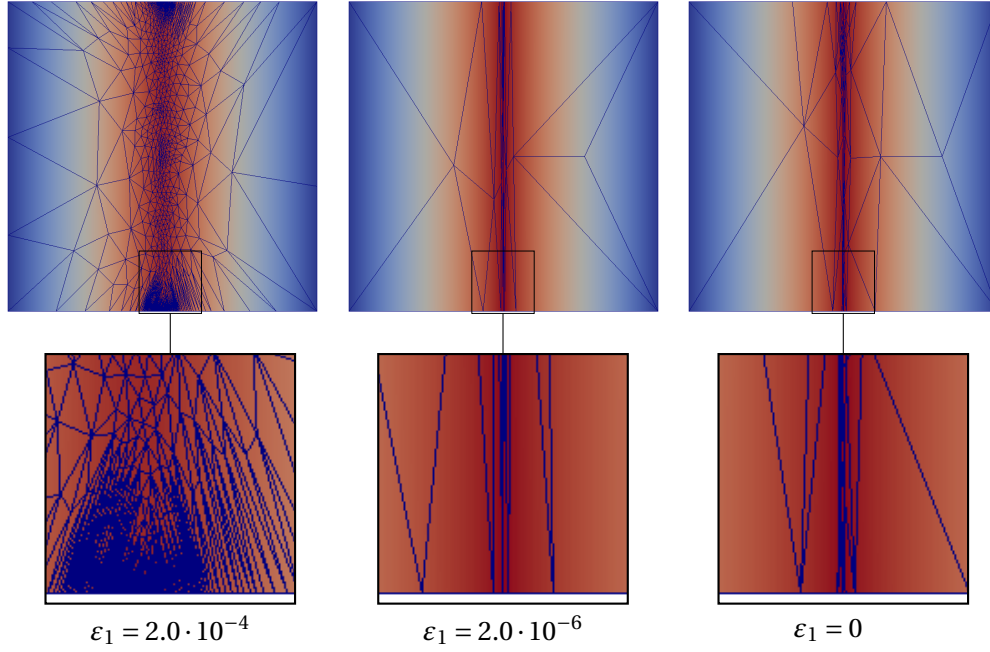


Figure 1.15 – Simple folding. Snapshots of the final adapted mesh after 500 time iterations, for various values of the regularization parameter ε_1 ($TOL = 0.03125$). The colormap represents the values of the first component $u_{1,h}$. The second row corresponds to a zoom in the squared region indicated in the first row.

Table 1.10 – Simple folding. Convergence behavior of the algorithm for various values of parameter ε_1 , as a function of the tolerance TOL . The columns contain the final minimal and maximal mesh size, the final numbers of elements and nodes, the maximal value of the aspect ratio, the value of the estimator, the effectivity index, and the L^2 -norm on the approximation $\mathbf{u}_h$ of the solution map $\mathbf{u}$.

Regularization term: $\varepsilon_1 = 2 \cdot 10^{-6}$								
TOL	h_{min}	h_{max}	AR	# elem	# nodes	η_K^A	$\frac{\eta_K^A}{\ \nabla(\mathbf{u} - \mathbf{u}_h)\ _{L^2}}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$
0.250000	7.86e-02	9.96e-01	10.57	24	19	4.49e-01	1.33 ± 0.4224	1.52e-02
0.125000	3.50e-02	1.00e+00	11.11	35	26	2.34e-01	1.18 ± 0.2705	4.56e-03
0.006250	1.09e-02	9.84e-01	29.27	40	28	1.38e-01	1.11 ± 0.3452	2.03e-03
0.031250	3.22e-03	9.83e-01	94.54	64	43	6.51e-02	0.86 ± 0.1584	8.88e-04
Regularization term: $\varepsilon_1 = 0.0$								
TOL	h_{min}	h_{max}	AR	# elem	# nodes	η_K^A	$\frac{\eta_K^A}{\ \nabla(\mathbf{u} - \mathbf{u}_h)\ _{L^2}}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$
0.250000	1.09e-01	1.00e+00	5.10	21	17	4.46e-01	1.39 ± 0.3156	1.36e-02
0.125000	2.45e-02	9.95e-01	19.24	39	28	2.14e-01	1.03 ± 0.2432	4.89e-03
0.006250	1.20e-02	9.90e-01	26.58	53	36	1.34e-01	0.93 ± 0.2517	2.85e-03
0.031250	4.53e-03	9.81e-01	46.53	61	40	7.47e-02	0.86 ± 0.1453	1.12e-03

Table 1.11 – Simple folding. Convergence behavior of the algorithm for various regularization parameters ε_1 , as a function of the tolerance TOL . The columns contain the constraints for the orthogonality of the solution.

Regularization term: $\varepsilon_1 = 2 \cdot 10^{-6}$			
TOL	$\int_{\Omega} \nabla u_{1,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{2,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{1,h} \cdot \nabla u_{2,h} d\mathbf{x}$
0.2500	0.9914	1.0002	0.0417
0.1250	0.9975	1.0001	0.0192
0.0625	0.9950	1.0000	0.0101
0.03125	0.9982	1.0000	0.0049
Regularization term: $\varepsilon_1 = 0.0$			
TOL	$\int_{\Omega} \nabla u_{1,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{2,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{1,h} \cdot \nabla u_{2,h} d\mathbf{x}$
0.0312	0.9982	1.0000	0.0049
0.1250	0.9929	1.0000	0.0221
0.0625	0.9959	1.0000	0.0165
0.03125	1.0004	1.0000	0.0010

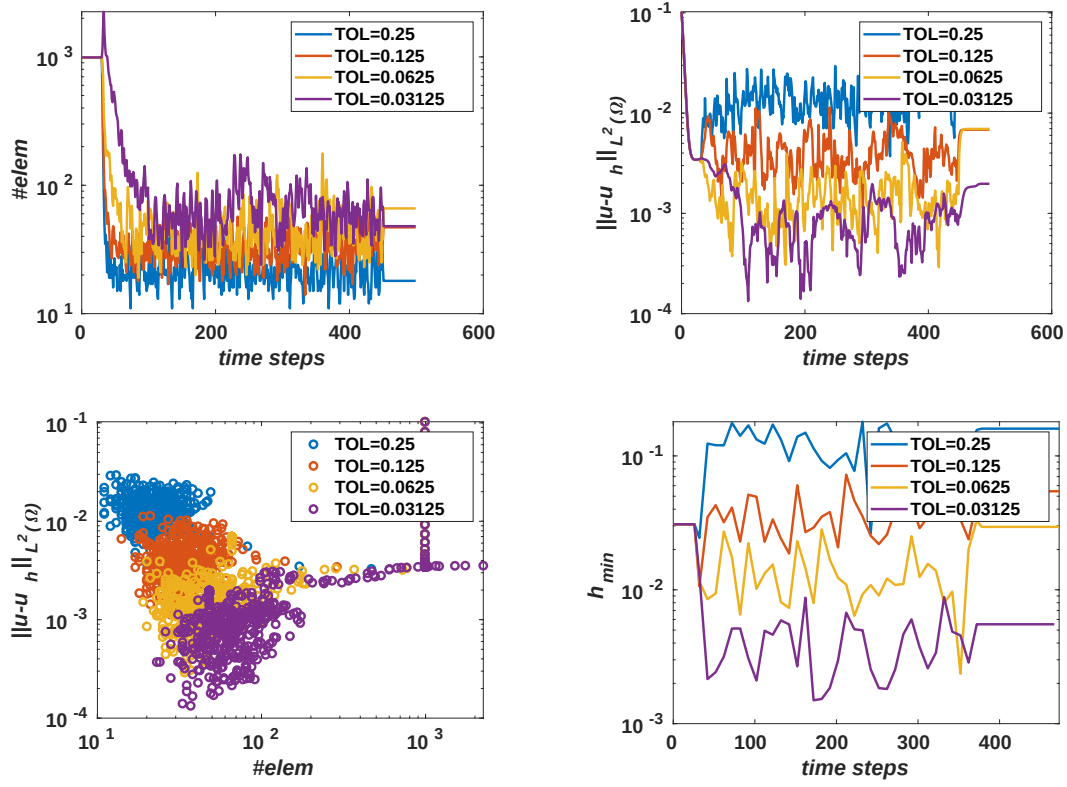


Figure 1.16 – Simple folding (case $\varepsilon_1 = 0.0$). Visualization of the behavior of the iterative algorithm. Top left: Visualization of the time evolution of the number of elements; Top right: Visualization of the time evolution of the error $\|\mathbf{u} - \mathbf{u}_h\|_{L^2(\Omega)}$; Bottom left: Visualization of the relationship between the error $\|\mathbf{u} - \mathbf{u}_h\|_{L^2(\Omega)}$ vs the number of elements; Bottom right: Visualization of the time evolution of h_{min} .

1.7.2 Double diagonal folding

The second example corresponds to a double folding, along the diagonals of the unit square domain. The exact solution of this problem is

$$\begin{aligned} u_1(x_1, x_2) &= d(\mathbf{x}, \partial\Omega), \\ u_2(x_1, x_2) &= \begin{cases} \min(x_2, 1 - x_1) & \text{if } x_1 < x_2, \\ \min(x_1, 1 - x_2) & \text{otherwise,} \end{cases} \quad \forall \mathbf{x} = (x_1, x_2) \in \Omega, \end{aligned}$$

and the boundary conditions are set accordingly. The additional difficulty lies in the intersection of two lines of the singular set. Figure 1.17 illustrates a snapshot of the stationary solution, together with an illustration of an adaptive mesh (in that case, $h_{\min} = 1.57 \cdot 10^{-3}$, $h_{\max} = 5.14 \cdot 10^{-1}$, $TOL = 0.03125$, $\varepsilon_1 = 0.0$).

Figure 1.18 illustrates, for $\varepsilon_1 = 0.0$, the refined mesh when the tolerance decreases. Again, the number of elements increases in a narrow neighborhood around the line singularities. Table 1.12 numerically confirms this statement. It also emphasizes that the convergence properties are comparable when $\varepsilon_1 \neq 0$ and when $\varepsilon_1 = 0$ (both in terms of accuracy and convergence rate).

Remark. *Using smaller tolerances usually requires a larger number of iterations, and may require a continuation approach (namely starting the time iterations with a larger tolerance and decreasing it as the iterations go, as, e.g., in [Caboussat et al., 2019]).*

Figure 1.19 illustrates the influence of the regularization parameter ε_1 . When ε_1 too large, the singularities of the solution are lost, and the anisotropic mesh adaptation algorithm does not converge easily.

Table 1.12 numerically confirms that (i) the smaller the tolerance, the larger the number of elements, nodes and aspect ratio; (ii) the estimator η_K^A and the error in the L^2 -norm are divided by $1.8 \sim 2.6$ when the tolerance is divided by two; (iii) however, the effectivity index does not remain constant.

Table 1.13 shows that the orthogonality constraints are satisfied and converge asymptotically. The accuracy in approximating the orthogonality conditions is better when $\varepsilon_1 = 0.0$.

Figure 1.20 illustrates the iterative behavior of the time-stepping algorithm for various tolerances when $\varepsilon_1 = 0.0$; it shows that the time evolution of indicators is indeed oscillating when the mesh is adapted, due to the low regularity of the solution. The conclusions are similar to those of the single folding example.

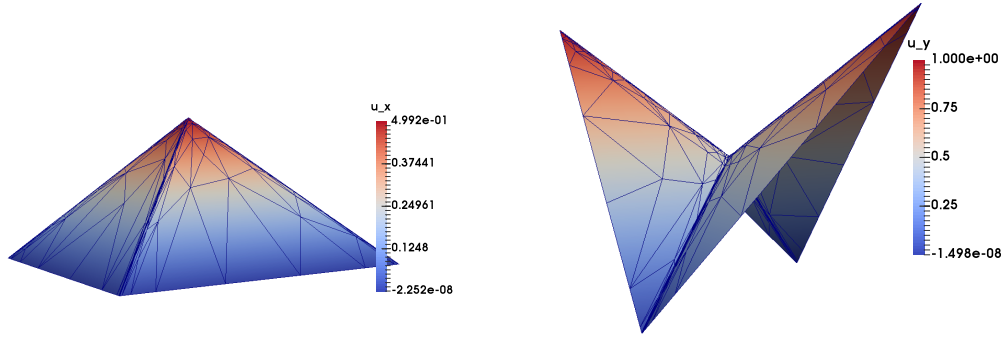


Figure 1.17 – Double diagonal folding. Snapshots of the approximated solution (left: first component $u_{1,h}$; right: second component $u_{2,h}$), with illustration of the final adapted mesh.

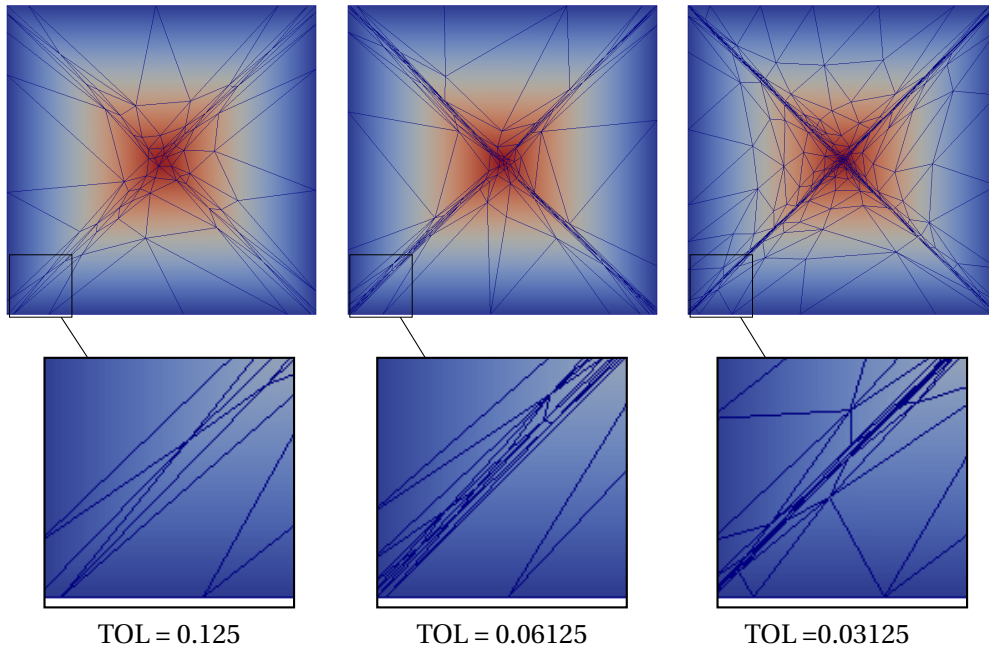


Figure 1.18 – Double diagonal folding. Snapshots of the final adapted mesh after 500 time iterations, for various values of the tolerance TOL ($\varepsilon_1 = 0.0$). The colormap represents the values of the first component $u_{1,h}$. The second row corresponds to a zoom in the squared region indicated in the first row.

1.7.3 Comparison with a standard adaptive approach

We actually advocate here a non-standard adaptive method when compared to the literature about mesh adaptive methods for elliptic problems. Indeed, the more standard approach [Bourgault et al., 2009; Hassan and Picasso, 2015] would be to solve the entire time-stepping problem, then to adapt the mesh and re-apply the whole solution method for the time-

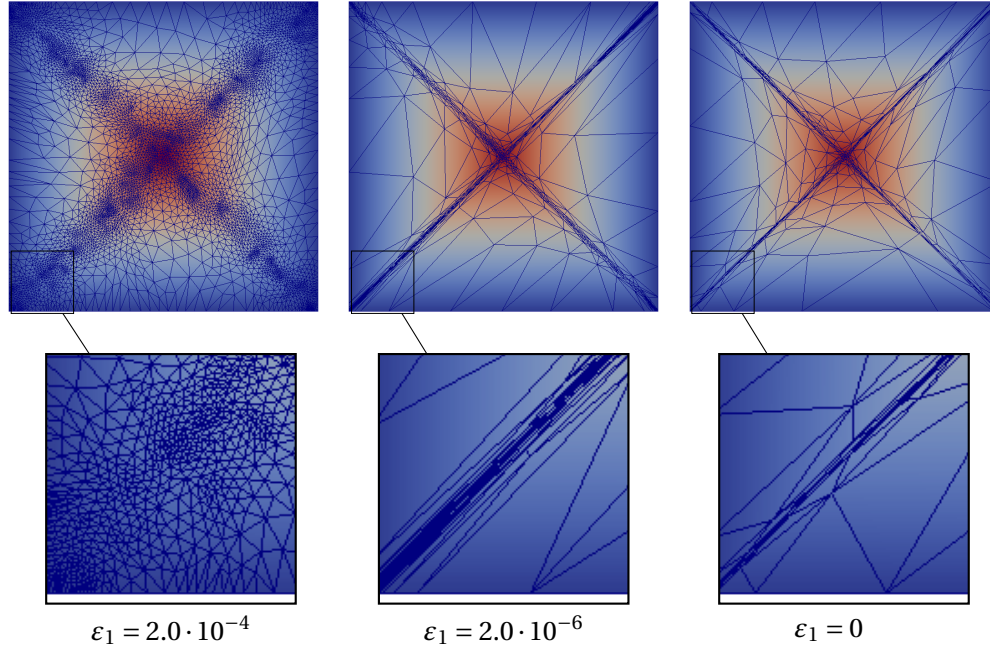


Figure 1.19 – Double diagonal folding. Snapshots of the final adapted mesh after 500 time iterations, for various values of the regularization parameter ε_1 ($TOL = 0.03125$). The colormap represents the values of the first component $u_{1,h}$. The second row corresponds to a zoom in the squared region indicated in the first row.

dependent problem. In the adaptive strategy described in Section 1.6.1, a few time iterations are performed without reaching the stationary solution with a fixed mesh, then the mesh is adapted at each time iteration. Actually the variations in the solution occur slowly and locally (on the edges where the singularities are formed), which favors a mesh adaptivity at each iteration. The other advantage of the adaptive approach described in Section 1.6.1 is that it allows to recover a suitable mesh faster compare to the standard strategy.

Figure 1.21 illustrates a comparison between the different approaches (with $TOL = 0.625$ and $\varepsilon_1 = 0$). The figure on the left shows the numerical solution using the adaptive approach (Section 1.6.1) after 500 iterations (total of time iterations and adaptive remeshing steps at each time iteration). The figure on the right shows the numerical solution applying the standard approach after 4000 iterations (same total). Figure 1.21 indicates that the mesh obtained with the adaptive approach tracks more efficiently the singularities with a smaller total number of iterations.

Table 1.14 shows the numerical behavior of the algorithm using the standard adaptive approach for varying parameters, namely the values of the parameters and final adapted meshes for all tolerances and $\varepsilon_1 = 2 \cdot 10^{-6}$ and $\varepsilon_1 = 0$.

Chapter 1. Numerical Approximation of Orthogonal Maps

Table 1.12 – Double diagonal folding. Convergence behavior of the algorithm for various values of parameter ε_1 , as a function of the tolerance TOL . The columns contain the final minimal and maximal mesh size, the final numbers of elements and nodes, the maximal value of the aspect ratio, the value of the estimator, the effectivity index, and the L^2 -norm on the approximation $\mathbf{u}_h$ of the solution map $\mathbf{u}$.

Regularization term: $\varepsilon_1 = 2 \cdot 10^{-6}$								
TOL	h_{min}	h_{max}	AR	# elem	# nodes	η_K^A	$\frac{\eta_K^A}{\ \nabla(\mathbf{u} - \mathbf{u}_h)\ _{L^2}}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$
0.250000	4.48e-02	4.80e-01	8.72	82	54	5.28e-01	1.18 ± 0.2049	1.74e-02
0.125000	1.42e-02	4.96e-01	18.63	157	93	2.96e-01	0.97 ± 0.0853	6.75e-03
0.006250	3.65e-03	4.81e-01	66.06	309	173	1.48e-01	0.81 ± 0.0993	2.92e-03
0.031250	7.75e-04	4.97e-01	208.35	537	300	7.42e-02	0.61 ± 0.0518	1.09e-03
Regularization term: $\varepsilon_1 = 0.0$								
TOL	h_{min}	h_{max}	AR	# elem	# nodes	η_K^A	$\frac{\eta_K^A}{\ \nabla(\mathbf{u} - \mathbf{u}_h)\ _{L^2}}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$
0.250000	4.42e-02	4.53e-01	7.69	86	56	5.25e-01	1.15 ± 0.1645	1.70e-02
0.125000	1.52e-02	4.71e-01	15.93	164	97	2.97e-01	1.03 ± 0.1109	6.60e-03
0.006250	4.82e-03	5.05e-01	50.25	265	150	1.56e-01	0.88 ± 0.1259	2.63e-03
0.031250	1.57e-03	5.14e-01	120.01	456	248	7.76e-02	0.69 ± 0.0767	1.15e-03

Table 1.13 – Double diagonal folding. Convergence behavior of the algorithm for various regularization parameters ε_1 , as a function of the tolerance TOL . The columns contain the constraints for the orthogonality of the solution.

Regularization term: $\varepsilon_1 = 2 \cdot 10^{-6}$			
TOL	$\int_{\Omega} \nabla u_{1,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{2,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{1,h} \cdot \nabla u_{2,h} d\mathbf{x}$
0.2500	0.9785	0.9899	0.1111
0.1250	0.9957	0.9960	0.0432
0.0625	0.9961	0.9957	0.0288
0.03125	0.9957	0.9946	0.0180
Regularization term: $\varepsilon_1 = 0.0$			
TOL	$\int_{\Omega} \nabla u_{1,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{2,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{1,h} \cdot \nabla u_{2,h} d\mathbf{x}$
0.2500	0.9896	0.9893	0.0763
0.1250	0.9976	0.9988	0.0409
0.0625	0.9988	0.9986	0.0237
0.03125	0.9997	0.9995	0.0100

1.7.4 Non-smooth example with a point singularity

The third example also corresponds to a double, re-entrant, folding. The exact solution of this problem is

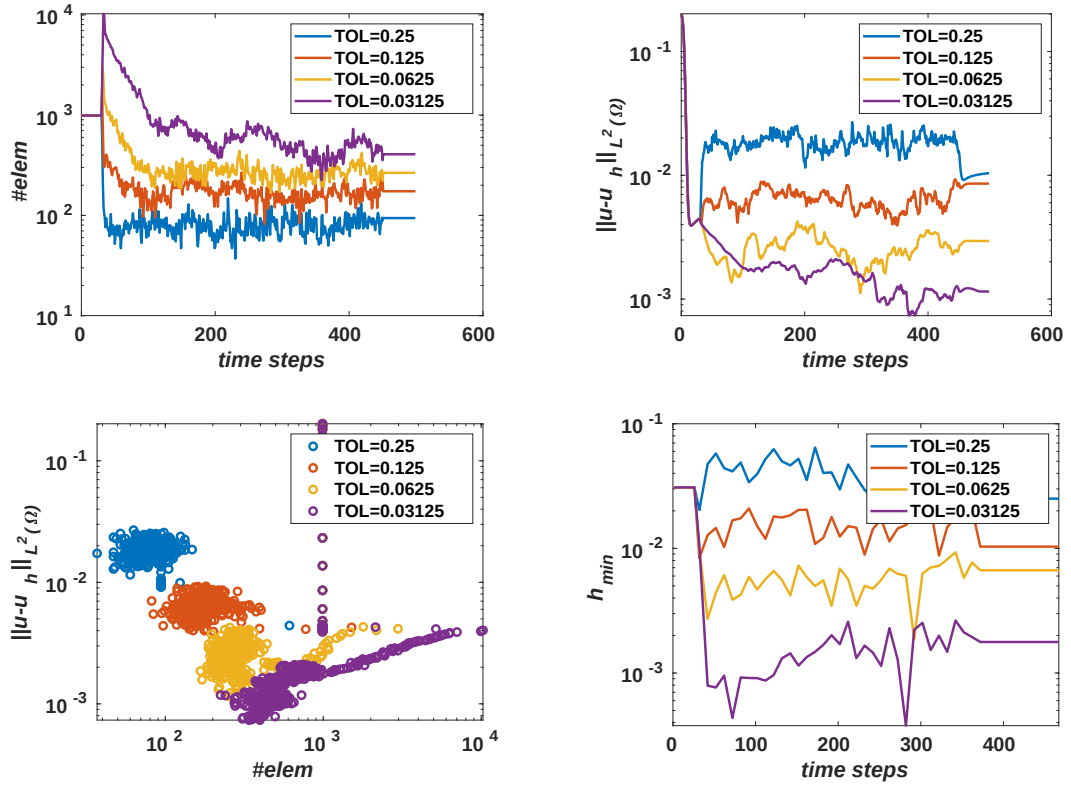


Figure 1.20 – Double diagonal folding (case $\varepsilon_1 = 0.0$). Visualization of the behavior of the iterative algorithm. Top left: Visualization of the time evolution of the number of elements; Top right: Visualization of the time evolution of the error $\|\mathbf{u} - \mathbf{u}_h\|_{L^2(\Omega)}$; Bottom left: Visualization of the relationship between the error $\|\mathbf{u} - \mathbf{u}_h\|_{L^2(\Omega)}$ vs the number of elements; Bottom right: Visualization of the time evolution of h_{min} ;

$$\begin{aligned}
 u_1(x_1, x_2) &= \begin{cases} x_1 & \text{if } x_2 \geq x_1 \text{ and } x_1 \leq 0.5, \\ 1 - x_1 & \text{if } x_1 > 0.5 \text{ and } x_2 > -x_1 + 1, \\ x_2 & \text{if } x_2 \leq x_1 \text{ and } x_2 \leq -x_1 + 1, \end{cases} \\
 u_2(x_1, x_2) &= \begin{cases} x_2 & \text{if } x_2 \geq x_1 \text{ and } x_1 \leq 0.5, \\ x_2 & \text{if } x_1 > 0.5 \text{ and } x_2 > -x_1 + 1, \\ x_1 & \text{if } x_2 \leq x_1 \text{ and } x_1 \leq 0.5, \\ 1 - x_1 & \text{if } x_2 > x_1 \text{ and } x_2 \leq -x_1 + 1, \end{cases} \quad \forall (x_1, x_2) \in \Omega,
 \end{aligned}$$

and the boundary conditions are set accordingly. We consider a fixed number of $n_{mesh} = 500$ time steps, and, for this example, we take $C = 0$. The additional difficulty lies in the refolding with a re-entrant corner, which causes a new type of point singularity. For this specific test problem, in the case when $TOL = 0.03125$, the algorithm fails to converge for $n_{mesh} = 500$. Therefore to accelerate the convergence of the adaptive algorithm, it uses gradually decreasing tolerances TOL (i.e. first a few iterations are performed with $TOL = 0.125$ then the tolerance

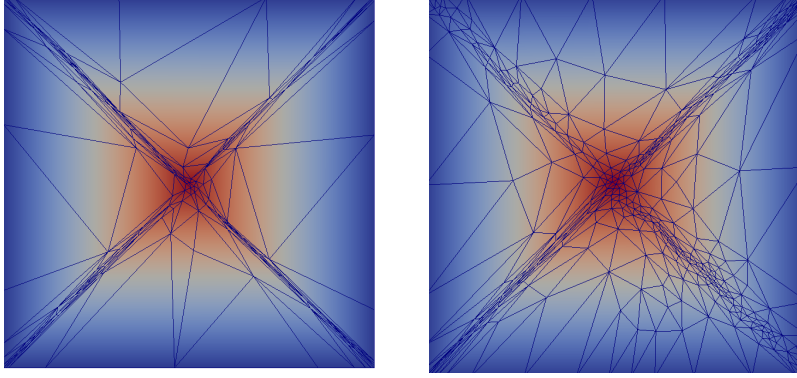


Figure 1.21 – Double diagonal folding. Snapshots of the final adapted mesh (the colormap represents the values of the first component $u_{1,h}$). Left: non-standard approach advocated here, after 500 time iterations; right: standard approach, after 4000 time iterations. ($TOL = 0.625$, $\varepsilon_1 = 0$).

Table 1.14 – Double diagonal folding. Convergence behavior of the algorithm for various values of parameter ε_1 , as a function of the tolerance using the standard adaptive strategy. The columns contain the final minimal and maximal mesh size, the final numbers of elements and nodes, the maximal value of the aspect ratio, the value of the estimator, the effectivity index, and the L^2 -norm on the approximation $\mathbf{u}_h$ of the solution map $\mathbf{u}$.

Regularization term: $\varepsilon_1 = 2 \cdot 10^{-6}$								
TOL	h_{min}	h_{max}	AR	# elem	# nodes	η_K^A	$\frac{\eta_K^A}{\ \nabla(\mathbf{u} - \mathbf{u}_h)\ _{L^2}}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$
0.250000	5.00e-02	4.82e-01	7.34	72	47	5.51e-01	1.44 ± 0.2884	1.42e-02
0.125000	1.54e-02	5.20e-01	20.72	161	95	2.74e-01	1.16 ± 0.1154	4.61e-03
0.006250	4.42e-03	3.96e-01	27.64	447	242	1.51e-01	0.81 ± 0.0623	4.05e-03
0.031250	1.48e-03	2.86e-01	23.36	1564	815	7.43e-02	0.67 ± 0.0597	7.96e-04
Regularization term: $\varepsilon_1 = 0.0$								
TOL	h_{min}	h_{max}	AR	# elem	# nodes	η_K^A	$\frac{\eta_K^A}{\ \nabla(\mathbf{u} - \mathbf{u}_h)\ _{L^2}}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$
0.250000	4.86e-02	4.56e-01	5.52	95	59	5.45e-01	1.37 ± 0.1456	1.35e-02
0.125000	1.65e-02	4.65e-01	13.18	157	92	2.90e-01	1.05 ± 0.1022	7.49e-03
0.006250	3.60e-03	3.98e-01	25.83	514	277	1.49e-01	0.84 ± 0.0618	2.35e-03
0.031250	1.07e-03	2.84e-01	30.35	1462	759	7.60e-02	0.55 ± 0.0357	1.62e-03

decreases to $TOL = 0.0625$ and $TOL = 0.03125$ successively).

Figure 1.22 illustrates a snapshot of the stationary solution, together with an illustration of an adaptive mesh (in that case, $h_{min} = 1.78 \cdot 10^{-3}$, $h_{max} = 5.81 \cdot 10^{-1}$, $TOL = 0.03125$, $\varepsilon_1 = 0.0$). Figures 1.23 and 1.24 illustrate the adapted mesh for $\varepsilon_1 = 0.0$ when the tolerance decreases, and for $TOL = 0.03125$ when the penalization parameter varies. Similar conclusions to the previous test cases hold.

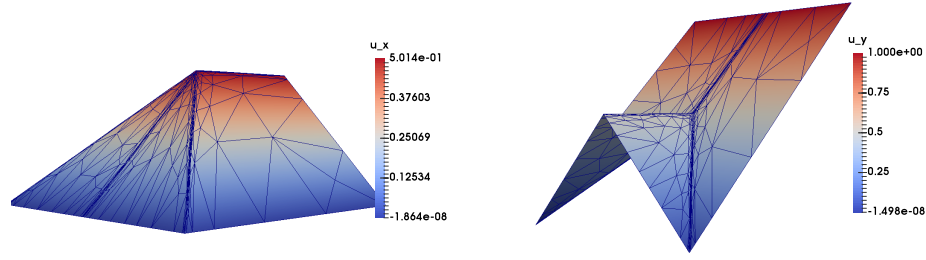


Figure 1.22 – Non-smooth folding with point singularity. Snapshots of the approximated solution (left: first component $u_{1,h}$; right: second component $u_{2,h}$), with illustration of the final adapted mesh.

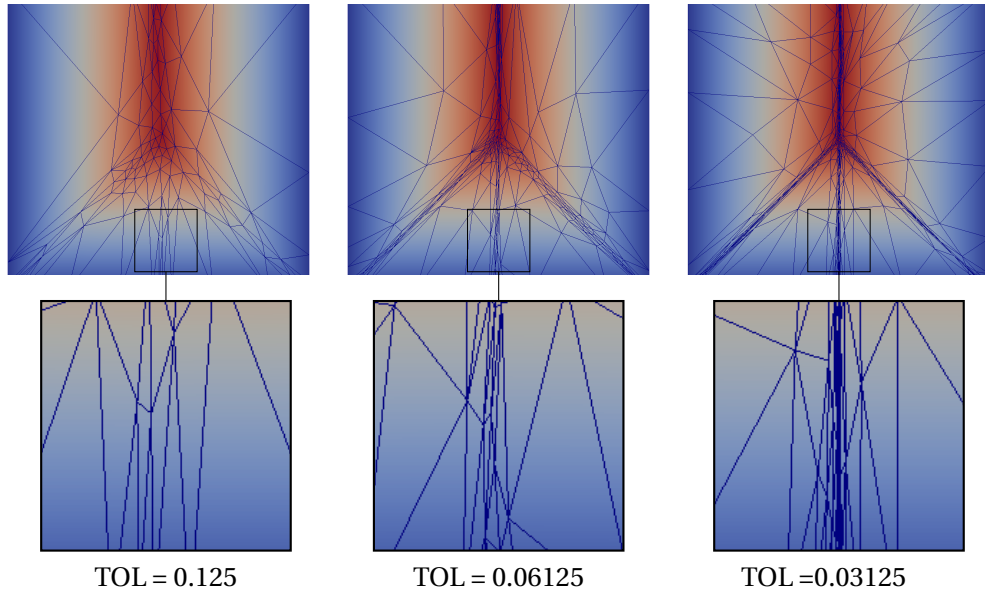


Figure 1.23 – Non-smooth folding with point singularity. Snapshots of the final adapted mesh after 500 time iterations, for various values of the tolerance TOL ($\varepsilon_1 = 0.0$). The colormap represents the values of the first component $u_{1,h}$. The second row corresponds to a zoom in the squared region indicated in the first row.

Table 1.15 numerically confirms the conclusions reached earlier. Table 1.16 shows an appropriate behavior for the orthogonality constraints that converge asymptotically when the tolerance decreases (except for one result for the smaller tolerance and $\varepsilon_1 \neq 0.0$). As expected, the accuracy in approximating the orthogonality conditions is higher when $\varepsilon_1 = 0.0$.

Figure 1.25 illustrates the iterative behavior of the time-stepping algorithm for various tolerances when $\varepsilon_1 = 0.0$ respectively; it confirms the oscillatory behavior of the time evolution of indicators when the mesh is adapted, due to the low regularity of the solution. Table 1.17

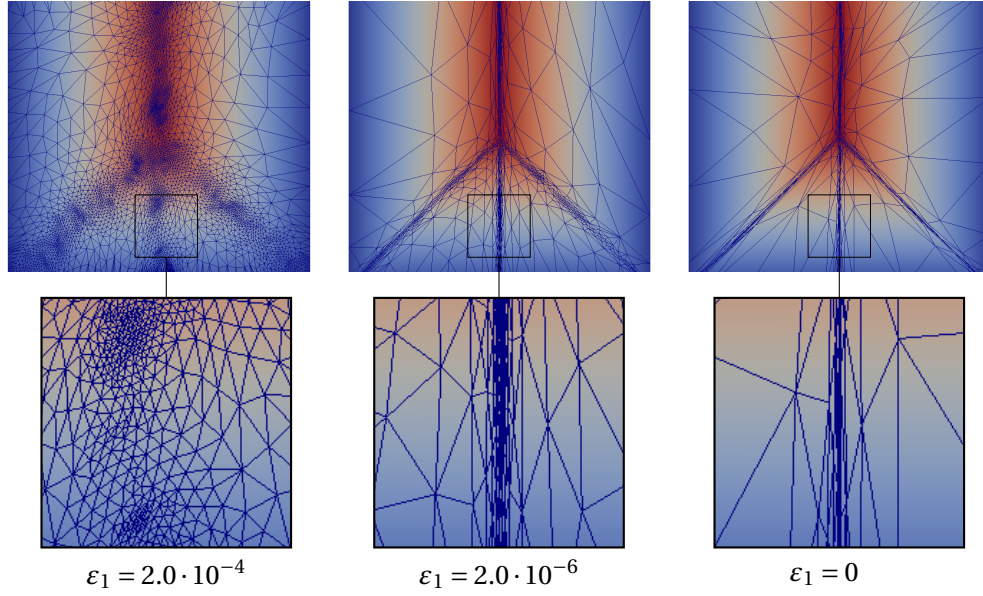


Figure 1.24 – Non-smooth folding with point singularity. Snapshots of the final adapted mesh after 500 time iterations, for various values of the regularization parameter ε_1 ($TOL = 0.03125$). The colormap represents the values of the first component $u_{1,h}$. The second row corresponds to a zoom in the squared region indicated in the first row.

Table 1.15 – Non-smooth folding with point singularity. Convergence behavior of the algorithm for various values of parameter ε_1 , as a function of the tolerance TOL . The columns contain the final minimal and maximal mesh size, the final numbers of elements and nodes, the maximal value of the aspect ratio, the value of the estimator, the effectivity index, and the L^2 -norm on the approximation $\mathbf{u}_h$ of the solution map $\mathbf{u}$.

Regularization term: $\varepsilon_1 = 2 \cdot 10^{-6}$								
TOL	h_{min}	h_{max}	AR	# elem	# nodes	η_K^A	$\frac{\eta_K^A}{\ \nabla(\mathbf{u} - \mathbf{u}_h)\ _{L^2}}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$
0.250000	4.83e-02	6.66e-01	8.14	60	41	5.11e-01	1.15 ± 0.1605	2.00e-02
0.125000	1.55e-02	6.58e-01	11.97	136	82	2.86e-01	0.98 ± 0.0875	8.18e-03
0.006250	4.59e-03	6.81e-01	37.89	259	149	1.44e-01	0.81 ± 0.1469	3.39e-03
0.031250	6.55e-04	5.81e-01	225.45	545	301	7.71e-02	0.64 ± 0.0697	1.48e-03
Regularization term: $\varepsilon_1 = 0.0$								
TOL	h_{min}	h_{max}	AR	# elem	# nodes	η_K^A	$\frac{\eta_K^A}{\ \nabla(\mathbf{u} - \mathbf{u}_h)\ _{L^2}}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$
0.250000	3.59e-02	6.92e-01	15.49	64	44	5.08e-01	1.08 ± 0.1142	2.16e-02
0.125000	1.61e-02	7.11e-01	15.97	122	75	2.88e-01	1.06 ± 0.1573	6.25e-03
0.006250	4.81e-03	6.82e-01	33.88	233	134	1.52e-01	0.86 ± 0.0989	2.59e-03
0.031250	1.78e-03	5.81e-01	49.02	427	234	7.71e-02	0.77 ± 0.1375	9.73e-04

illustrates the numerical behavior of the approximated solution for various ε_1 . Again, when

Table 1.16 – Non-smooth folding with point singularity. Convergence behavior of the algorithm for various regularization parameters ε_1 , as a function of the tolerance TOL . The columns contain the constraints for the orthogonality of the solution.

Regularization term: $\varepsilon_1 = 2 \cdot 10^{-6}$			
TOL	$\int_{\Omega} \nabla u_{1,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{2,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{1,h} \cdot \nabla u_{2,h} d\mathbf{x}$
0.2500	0.9780	0.9907	0.0826
0.1250	0.9933	0.9945	0.0489
0.0625	0.9968	0.9979	0.0174
0.03125	0.9959	0.9951	0.0092
Regularization term: $\varepsilon_1 = 0.0$			
TOL	$\int_{\Omega} \nabla u_{1,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{2,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{1,h} \cdot \nabla u_{2,h} d\mathbf{x}$
0.2500	0.9754	0.9947	0.1126
0.1250	0.9899	0.9956	0.0493
0.0625	0.9984	0.9994	0.0187
0.0312	0.9992	0.9995	0.0074

ε_1 is too large, the algorithm under-performs. Then, we can observe that, when $\varepsilon_1 \rightarrow 0.0$, the number of elements and nodes remains bounded despite the loss of regularity of the solution. Similarly the orthogonality conditions become more accurate when $\varepsilon_1 \rightarrow 0.0$.

Table 1.17 – Non-smooth folding with point singularity. Convergence behavior of the algorithm as a function of the regularization parameter ε_1 (tolerance: $TOL = 0.3125$).

ε_1	h_{min}	h_{max}	# elem	# nodes	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$
2.0e-04	7.66e-04	2.35e-01	3869	2095	1.25e-04
2.0e-06	6.55e-04	5.81e-01	545	301	1.48e-03
2.0e-08	1.70e-03	4.70e-01	422	232	4.16e-04
0.0	1.78e-03	5.81e-01	427	234	9.73e-04
ε_1	$\int_{\Omega} \nabla u_{1,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{2,h} d\mathbf{x}$	$\int_{\Omega} \nabla u_{1,h} \cdot \nabla u_{2,h} d\mathbf{x}$		
2.0e-04	0.9529	0.9496	0.0641		
2.0e-06	0.9960	0.9954	0.0113		
2.0e-08	0.9992	0.9996	0.0117		
0.0	0.9991	0.9998	0.0074		

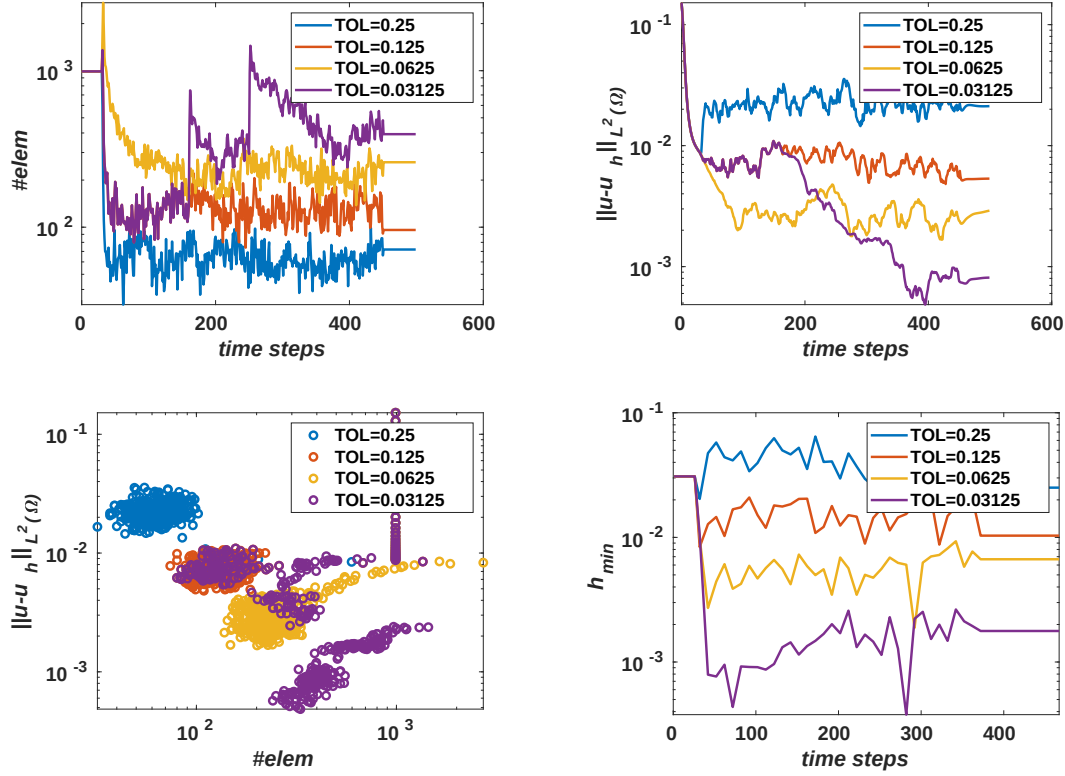


Figure 1.25 – Non-smooth folding with point singularity (case $\varepsilon_1 = 0.0$). Visualization of the behavior of the iterative algorithm. Top left: Visualization of the time evolution of the number of elements; Top right: Visualization of the time evolution of the error $\|u - u_h\|_{L^2(\Omega)}$; Bottom left: Visualization of the relationship between the error $\|u - u_h\|_{L^2(\Omega)}$ vs the number of elements; Bottom right: Visualization of the time evolution of h_{min} ;

1.8 A decomposition approach for the homogeneous Dirichlet Problem

In this final section, let us consider the homogeneous Dirichlet problem:

$$\begin{cases} \nabla \mathbf{u} \in \mathcal{O}(2) & \text{a.e. in } \Omega, \\ \mathbf{u} = \mathbf{0} & \text{on } \partial\Omega. \end{cases} \quad (1.44)$$

In this particular case, the solution becomes fractal near the boundary (see, e.g., [Dacorogna et al., 2018], but also [Caboussat et al., 2015] for a similar behavior for a scalar Eikonal equation). Preliminary numerical results reported in [Caboussat et al., 2019] have shown that adaptive mesh refinement is required to obtain the convergence of the time-stepping algorithm, and recover an admissible solution. Figure 1.26 illustrates the snapshot of the numerical approximation $\mathbf{u}_h$ of $\mathbf{u}$, obtained with the anisotropic mesh adaptivity algorithm. However,

1.8. A decomposition approach for the homogeneous Dirichlet Problem

sharp edges cannot be recovered exactly near the boundary where strong oscillations arise.

One property of the solution is that every singular point should be adjacent an even number of edges, and this number is at least four. This property is not satisfied for the numerical approximation in Figure 1.26 due to these inaccuracies near the boundary.

In order to overcome the introduction of such inaccuracies and numerically capture one solution, we advocate a domain decomposition algorithm to approximate one given solution of (1.44). Let us consider $\Omega = (0, 4)^2$, using the geometric information about the expected oscillatory behavior of the solution near the boundary, we define a sequence of domains Ω_k , $k \geq 0$, such that

$$\overline{\Omega} = \bigcup_{k \geq 0} \overline{\Omega_k},$$

with

$$\Omega_0 = (1, 3)^2, \quad \text{and} \quad \Omega_k = \left(1 - \frac{1}{k}, 3 + \frac{1}{k}\right)^2 \setminus \overline{\Omega_{k-1}}, \quad k \geq 1.$$

Numerically, the sequence is truncated to M domains, and $\Omega_M = (0, 4)^2 \setminus \overline{\Omega_{M-1}}$. Figure 1.27 (left) illustrates the situation for a decomposition in four domains. This decomposition allows to enforce the fractal behavior through boundary conditions. Figure 1.27 (right) illustrates schematically the shape of successive boundary conditions (only on half of the boundary).

The algorithm reads as follows: for each k , the orthogonal maps problem is solved on Ω_k with given boundary conditions. We use the following notation: $\Omega_k = (a_k, b_k)^2$, $N_k = 2N_{k-1} + 4$ for $k \geq 1$, $N_0 = 4$, $h_k = \frac{b_k - a_k}{N_k}$ and $p_j = a_k + j h_k$, where $j = 2\ell + 1$, $\ell = 0, \dots, \frac{N_k}{2} - 1$.

The boundary conditions are written, for all $\mathbf{x} = (x_1, x_2)$ on $\partial\Omega_k$ as $\mathbf{u} = \mathbf{g}_k$ in $\partial\Omega_k$, with

$$\mathbf{g}_k = \begin{pmatrix} g_{1,k} \\ g_{2,k} \end{pmatrix},$$

where

$$g_{1,k}(x_1, x_2) = 0 \tag{1.45}$$

and

$$g_{2,k}(x_1, x_2) = \begin{cases} y - p_{2\ell}, & \text{if } p_{2\ell} \leq y \leq p_{2\ell+1}, \\ p_{2\ell+2} - y, & \text{if } p_{2\ell+1} \leq y \leq p_{2\ell+2}, \end{cases} \quad \forall \ell = 0, 1, 2, \dots, \frac{N_k}{2} - 1, \tag{1.46}$$

where

$$y = \begin{cases} x_1 & \text{when } x_2 = a_k \text{ or } x_2 = b_k, \\ x_2 & \text{when } x_1 = a_k \text{ or } x_1 = b_k. \end{cases}$$

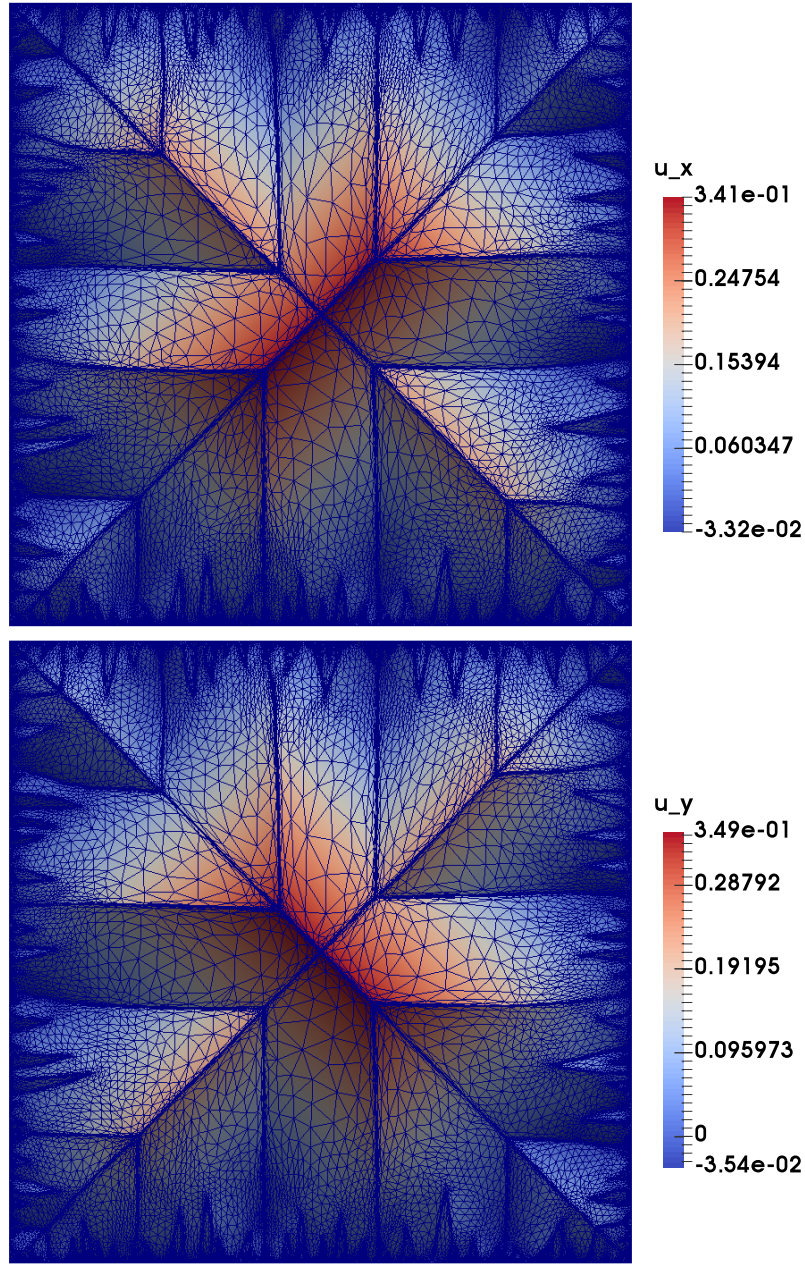


Figure 1.26 – Homogeneous Dirichlet test case. Snapshots of the approximated solution (top: first component $u_{1,h}$; bottom: second component $u_{2,h}$), with illustration of the final adapted mesh consisting of an unstructured adapted triangulation with 22,871 vertices and 42,987 triangles ($C = 0$, $\Delta t = 5 \cdot 10^{-12}$, $\varepsilon_1 = 0$, $TOL = 0.625$, approx. 3000 – 5000 time steps).

On the internal boundary, the external boundary conditions $\mathbf{g}_{k-1}$ are reproduced.

Figures 1.28-1.29 illustrate the snapshot of the numerical approximation $\mathbf{u}_h$ obtained by coupling the anisotropic adaptive algorithm with this domain decomposition method and

1.8. A decomposition approach for the homogeneous Dirichlet Problem

$M = 4$. Results show that the symmetries are perfectly respected, and that the spurious oscillations are controlled. As a side effect, let us note that this approach allows us to observe that the proposed adapted algorithm behaves well when solving the orthogonal maps problem in non-convex domains, as emphasized in Figure 1.30. Finally, Figure 1.31 visualizes $\det \nabla \mathbf{u}_h$ and shows that the angle conditions is satisfied.

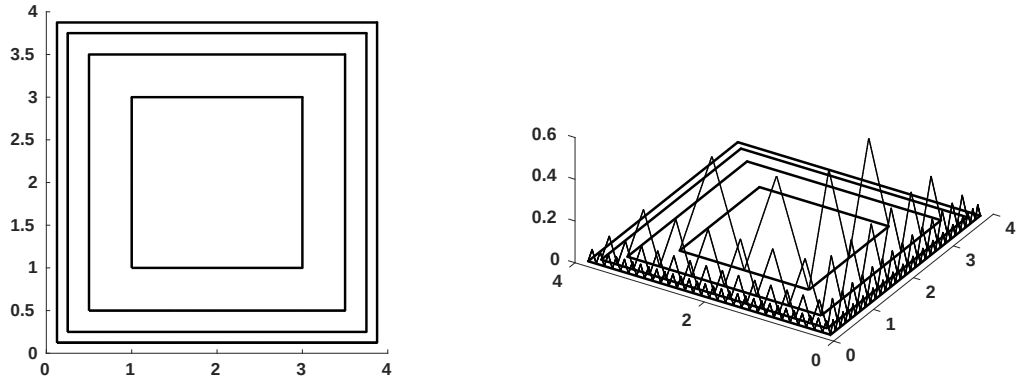


Figure 1.27 – Decomposition approach for the solution of the homogeneous Dirichlet problem. Left: sketch of the sequence of domains Ω_k ; Right: sketch of the shape of (piecewise linear) boundary conditions.

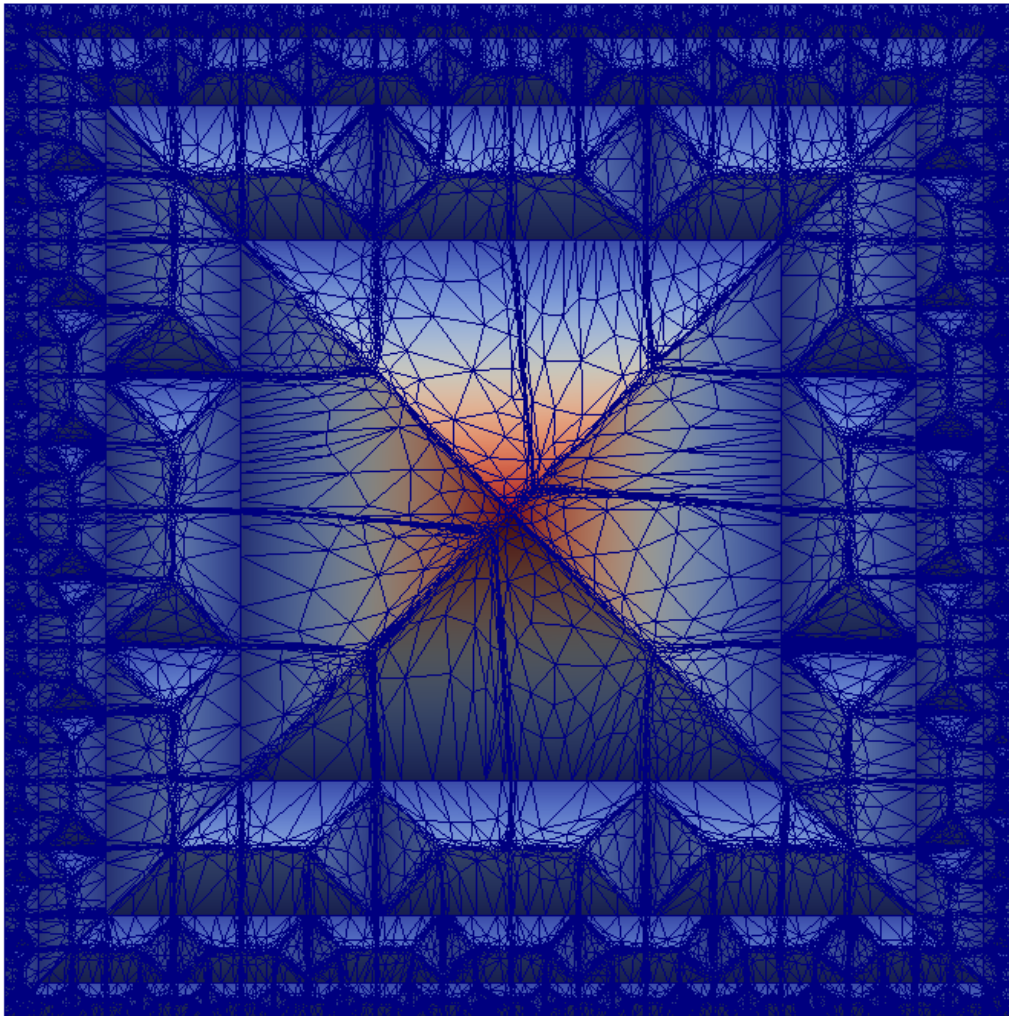


Figure 1.28 – Decomposition approach for the solution of the homogeneous Dirichlet problem. Snapshot of the approximated solution $u_{1,h}$, with illustration of the final adapted mesh after 4150 time iterations on each subdomain, superimposed after individual computation on each subdomain. The resulting mesh obtained by superimposition of the several meshes is non-conforming.

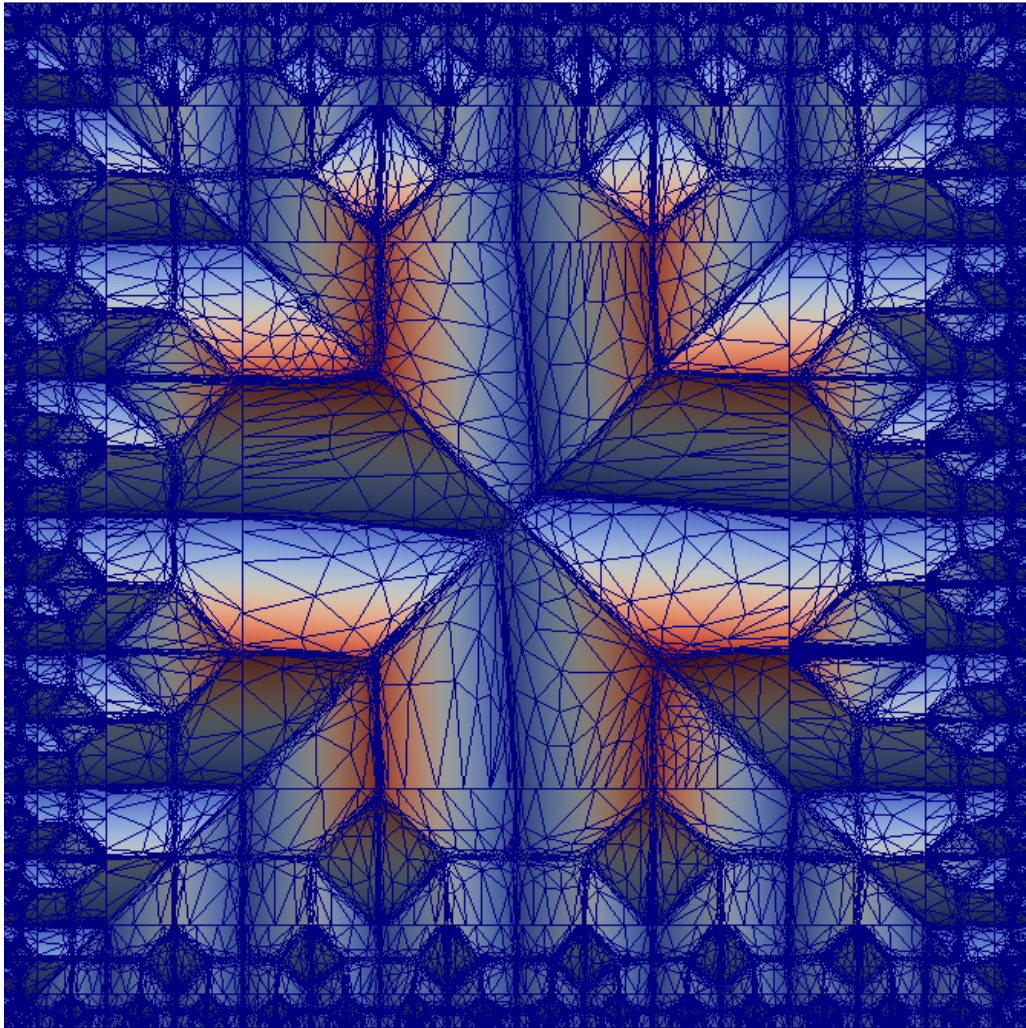


Figure 1.29 – Decomposition approach for the solution of the homogeneous Dirichlet problem. Snapshot of the approximated solution $u_{2,h}$, with illustration of the final adapted mesh after 4150 time iterations on each subdomain, superimposed after individual computation on each subdomain. The resulting mesh obtained by superimposition of the several meshes is non-conforming.

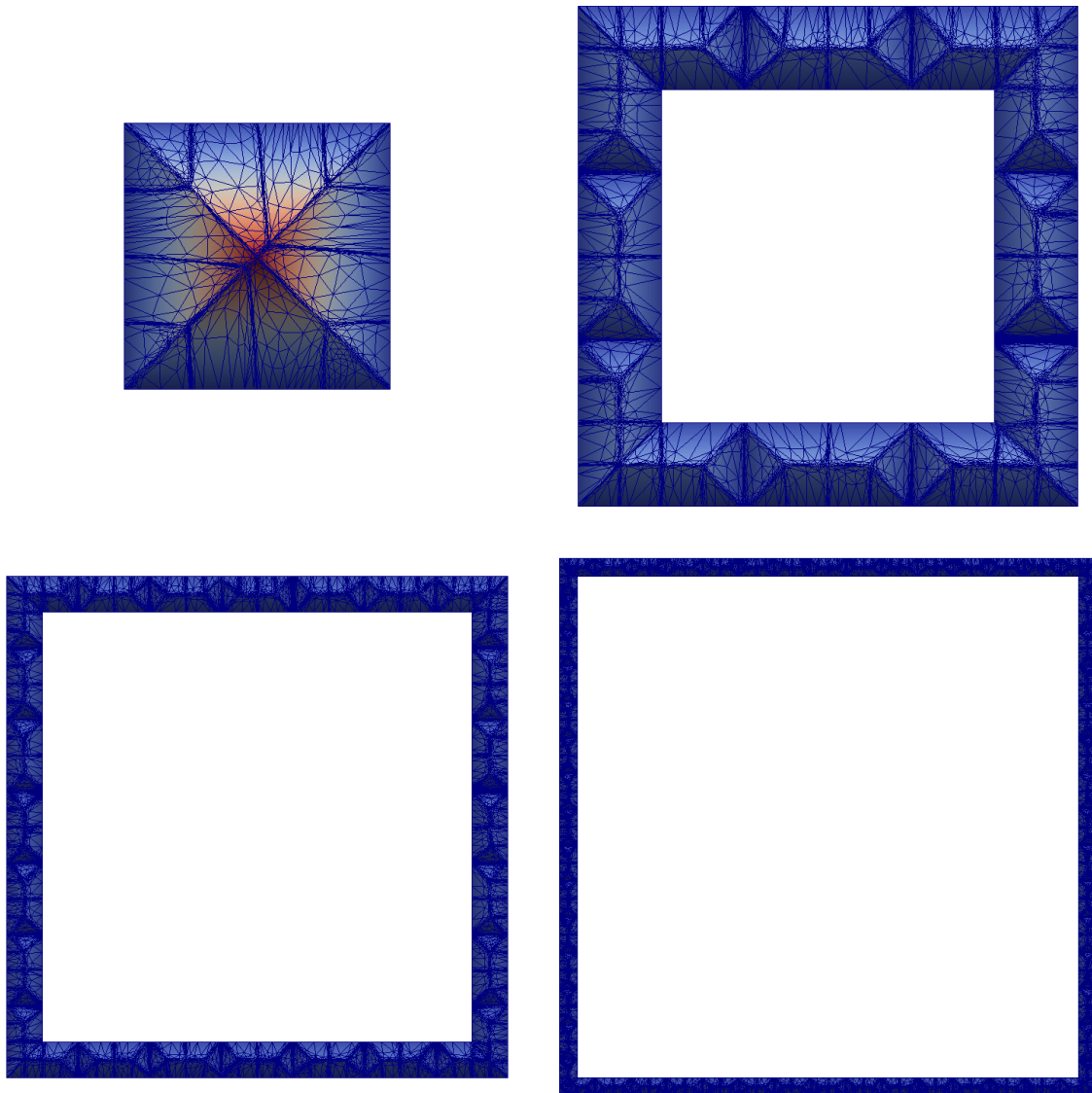


Figure 1.30 – Decomposition approach for the solution of the homogeneous Dirichlet problem. Snapshots of the individual solution computed on each subdomain. The colormap represents the values of the first component $u_{1,h}$.

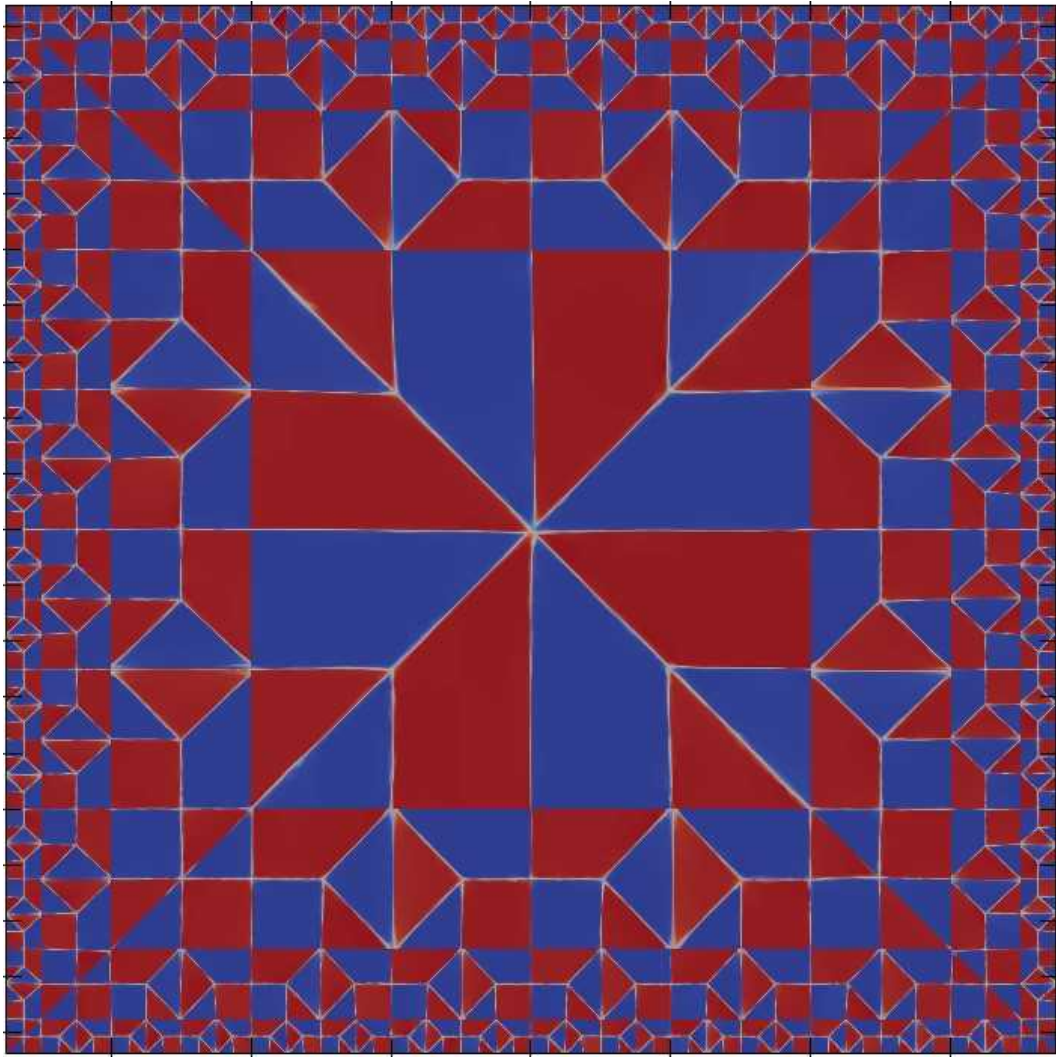


Figure 1.31 – Decomposition approach for the solution of the homogeneous Dirichlet problem. Visualization of $\det \nabla \mathbf{u}_h$.

2 Numerical Approximation of the Prescribed Jacobian Equation/Inequality

2.1 Problem Formulation

Let Ω be a bounded domain of $\mathbb{R}^2$; we denote by Γ the boundary of Ω . Let $f : \Omega \rightarrow \mathbb{R}_+$ and $\mathbf{g} : \Gamma \rightarrow \mathbb{R}^2$ be given sufficiently regular functions. The partial differential equation involving the Jacobian determinant we want to solve reads as follows: find $\mathbf{u} : \Omega \rightarrow \mathbb{R}^2$ satisfying

$$\begin{cases} \det \nabla \mathbf{u} = f & \text{in } \Omega, \\ \mathbf{u} = \mathbf{g} & \text{on } \Gamma. \end{cases} \quad (2.1)$$

In particular, we are interested in the identity boundary condition (i.e. $\mathbf{g}(\mathbf{x}) = \mathbf{x}$ for $\mathbf{x} \in \Gamma$) that has been considered in [Caboussat and Glowinski, 2018], and in [Dacorogna and Moser, 1990] from a theoretical point of view. The problem is of the following type:

$$\begin{cases} \det \nabla \mathbf{u} = f & \text{in } \Omega, \\ \mathbf{u}(\mathbf{x}) = \mathbf{x} & a.e. \mathbf{x} \text{ on } \Gamma. \end{cases} \quad (2.2)$$

This problem has been addressed in [Caboussat and Glowinski, 2018] with a numerical approach based on augmented Lagrangian techniques, which will be revisited here with a least-squares approach. Problem (2.2) corresponds to finding a mapping $\mathbf{u}$ that preserves both the boundary data and some kind of volume (up to some stretching of the mapping). Note that the solution to (2.2) is not necessarily unique and the same remark holds for (2.1). Indeed, let us consider (2.2) with $f = 1$ and Ω the unit disk centered at the origin; in this case, $\mathbf{u}(\mathbf{x}) = \mathbf{x}$ is an obvious solution. However, when using the polar coordinates (ρ, θ) , one can see that $\mathbf{v}$ defined by $\mathbf{v}(\rho, \theta) = (\rho \cos(\theta + 2k\pi\rho^2), \rho \sin(\theta + 2k\pi\rho^2))^T$ is also a solution.

The proof of the existence of a solution to (2.1) (via the divergence theorem) requires data to be compatible with the geometrical domain, see [Dacorogna and Moser, 1990]. When the

boundary conditions are given by $\mathbf{u}(\mathbf{x}) = \mathbf{x}$ on Γ , this compatibility condition reads as:

$$\int_{\Omega} f d\mathbf{x} = \text{measure}(\Omega). \quad (2.3)$$

The positiveness of the right-hand side f is useful from an analytical point of view to prove existence results; however, it has been recently loosened slightly to accept locally negative data (see [Cupini et al., 2009]). Moreover, it makes problem (2.1) elliptic, an important feature for the solution methodology discussed in this chapter. From now on we will assume that (2.3) holds.

In parallel, in this chapter we will also consider the following partial differential inequality : find $\mathbf{u} : \Omega \rightarrow \mathbb{R}^2$ satisfying

$$\begin{cases} \det \nabla \mathbf{u} \geq f & \text{in } \Omega, \\ \mathbf{u} = \mathbf{g} & \text{on } \Gamma. \end{cases} \quad (2.4)$$

with, in particular, the case of the identity boundary condition (i.e. $\mathbf{g}(\mathbf{x}) = \mathbf{x}$ for $\mathbf{x} \in \Gamma$). This problem has been addressed in [Fischer and Kneuss, 2019] where existence results have been established under the condition

$$\int_{\Omega} f d\mathbf{x} \leq \text{measure}(\Omega). \quad (2.5)$$

We will show that the numerical techniques developed for (2.1) also apply naturally to (2.4), with small modifications. Thus, in the sequel, the least-squares method we advocate is described when applied to the initial problem (2.1), and variations in the algorithms to address (2.4) are highlighted.

2.2 Numerical Algorithm

We propose a numerical algorithm based on the least-squares approach. The iterative relaxation algorithm allows to split the minimization problem into two separate sub problems. The first problem consist of a sequence of low dimensional local nonlinear problems, the number of them being determined from the chosen mesh grid. The second problem is a linear variational problem and it results in a fourth-order partial differential equation.

2.2.1 Least squares method

Let us define

$$\begin{aligned} \mathbf{Q}_f &= \left\{ \mathbf{q} \in \left(L^2(\Omega) \right)^{2 \times 2}, \det \mathbf{q} = f \right\}, \\ \mathbf{V}_{\mathbf{g}} &= \left\{ \mathbf{v} \in \left(H^1(\Omega) \right)^2, \mathbf{v}|_{\partial\Omega} = \mathbf{g} \right\}. \end{aligned}$$

The regularity of f and g implies that $\mathbf{Q}_f$ and $\mathbf{V}_g$ are non-empty.

The main idea of the numerical method is to find a solution at the intersection of $\mathbf{Q}_f$ and $\nabla \mathbf{V}_g$ (if there is an intersection). Finding the intersection is equivalent to solving (2.1). The least-squares method thus relies on the introduction of an auxiliary variable $\mathbf{q} \in \mathbf{Q}_f$, and reads as: find $\{\mathbf{u}, \mathbf{p}\} \in \mathbf{V}_g \times \mathbf{Q}_f$ such that

$$J(\mathbf{u}, \mathbf{p}) \leq J(\mathbf{v}, \mathbf{q}), \quad \forall \{\mathbf{v}, \mathbf{q}\} \in \mathbf{V}_g \times \mathbf{Q}_f, \quad (2.6)$$

where, in (2.6), J is defined by

$$J(\mathbf{v}, \mathbf{q}) = \frac{1}{2} \int_{\Omega} |\nabla \mathbf{v} - \mathbf{q}|^2 dx, \quad (2.7)$$

and $|\cdot|$ denotes the Frobenius norm $|\mathbf{T}| = (\mathbf{T} : \mathbf{T})^{1/2}$, with the inner product $\mathbf{S} : \mathbf{T} = \sum_{i,j=1}^2 s_{ij} t_{ij}$ where $\mathbf{T}, \mathbf{S}$ are 2×2 matrices with elements s_{ij}, t_{ij} for $i, j = 1, 2$, respectively.

We propose a biharmonic regularization of the objective function. The added term is motivated by previous works that involve first-order elliptic equations, see [Caboussat and Glowinski, 2015b, 2018; Caboussat et al., 2019]. As illustrated in the numerical experiments introducing the biharmonic regularization accelerates the convergence of the numerical algorithm. The modified objective function is defined as

$$J_{\varepsilon}(\mathbf{v}, \mathbf{q}) = J(\mathbf{v}, \mathbf{q}) + \frac{\varepsilon}{2} \int_{\Omega} |\nabla^2 \mathbf{v}|^2 dx, \quad (2.8)$$

where $\varepsilon \geq 0$. The modified minimization problem reads as: find $\{\mathbf{u}, \mathbf{p}\} \in \bar{\mathbf{V}}_g \times \mathbf{Q}_f$ such that

$$J_{\varepsilon}(\mathbf{u}, \mathbf{p}) \leq J_{\varepsilon}(\mathbf{v}, \mathbf{q}), \quad \forall \{\mathbf{v}, \mathbf{q}\} \in \bar{\mathbf{V}}_g \times \mathbf{Q}_f, \quad (2.9)$$

where $\bar{\mathbf{V}}_g = \mathbf{V}_g \cap (H^2(\Omega))^2$.

In the case the inequality (2.4), we modify the functional space $\mathbf{Q}_f$ as

$$\tilde{\mathbf{Q}}_f = \left\{ \mathbf{q} \in (L^2(\Omega))^{2 \times 2}, \det \mathbf{q} \geq f \right\}.$$

The objective function $J_{\varepsilon}(\mathbf{v}, \mathbf{q})$ remains the same as in (2.8), and the minimization problem reads as: find $\{\mathbf{u}, \mathbf{p}\} \in \bar{\mathbf{V}}_g \times \tilde{\mathbf{Q}}_f$ such that

$$J_{\varepsilon}(\mathbf{u}, \mathbf{p}) \leq J_{\varepsilon}(\mathbf{v}, \mathbf{q}), \quad \forall \{\mathbf{v}, \mathbf{q}\} \in \bar{\mathbf{V}}_g \times \tilde{\mathbf{Q}}_f. \quad (2.10)$$

2.2.2 Relaxation algorithm

For the solution of the minimization problems (2.9) and (2.10) respectively, we propose a relaxation algorithm. It reads as follows

Step 1. The initial guess of the algorithm is obtained by solving:

$$\begin{cases} -\Delta \mathbf{u}^1 = \tilde{\mathbf{f}} & \text{in } \Omega, \\ \mathbf{u}^1 = \mathbf{g} & \text{on } \Gamma, \end{cases} \quad (2.11)$$

where $\tilde{\mathbf{f}} = (1, 1)^T \forall \mathbf{x} \in \Omega$. The solution of (2.11) is smooth, convex, and matches the boundary conditions of (2.1).

For $n \geq 1$;

Step 2. Assuming that $\mathbf{u}^n$ is known, we look for

$$\mathbf{p}^n = \arg \min_{\mathbf{q} \in \mathbf{Q}_f} J_\varepsilon(\mathbf{u}^n, \mathbf{q}); \quad (2.12)$$

Step 3. Assuming that $\mathbf{p}^n$ is known, we look for

$$\mathbf{u}^{n+1/2} = \arg \min_{\mathbf{v} \in \mathbf{V}_g^\varepsilon} J_\varepsilon(\mathbf{v}, \mathbf{p}^n); \quad (2.13)$$

Step 4. Update the solution as

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \omega(\mathbf{u}^{n+1/2} - \mathbf{u}^n), \quad (2.14)$$

where $\omega \in (0, 2)$ is a relaxation parameter that helps controlling the convergence speed of the algorithm.

For the Jacobian inequality (2.4), the solution can be found by replacing in (2.12) the functional space $\mathbf{Q}_f$ by $\tilde{\mathbf{Q}}_f$ defined in (2.10). Therefore, the only modification needed in the algorithm is in **Step 2**.

2.2.3 Numerical solution of the local nonlinearly constrained minimization problems

The main focus of this subsection is the solution of (2.12) in the relaxation algorithm. Since $\mathbf{u}^n$ is known, we have

$$J_\varepsilon(\mathbf{u}^n, \mathbf{q}) = \frac{1}{2} \int_{\Omega} |\nabla \mathbf{u}^n - \mathbf{q}|^2 d\mathbf{x} + \frac{\varepsilon}{2} \int_{\Omega} |\nabla^2 \mathbf{u}^n|^2 dx, \quad (2.15)$$

and expanding the first term we get

$$J_\varepsilon(\mathbf{u}^n, \mathbf{q}) = \frac{1}{2} \int_{\Omega} |\nabla \mathbf{u}^n|^2 - \int_{\Omega} \nabla \mathbf{u}^n : \mathbf{q} d\mathbf{x} + \frac{1}{2} \int_{\Omega} |\mathbf{q}|^2 d\mathbf{x} + \frac{\varepsilon}{2} \int_{\Omega} |\nabla^2 \mathbf{u}^n|^2 dx. \quad (2.16)$$

Then, the solution of $\mathbf{p}^n$ is obtained by solving the following minimization problem

$$\min_{\mathbf{q} \in \mathbf{Q}_f} \left[\frac{1}{2} \int_{\Omega} |\nabla \mathbf{u}^n|^2 - \int_{\Omega} \nabla \mathbf{u}^n : \mathbf{q} d\mathbf{x} + \frac{1}{2} \int_{\Omega} |\mathbf{q}|^2 d\mathbf{x} + \frac{\varepsilon}{2} \int_{\Omega} |\nabla^2 \mathbf{u}^n|^2 dx \right], \quad (2.17)$$

Problem (2.17) is equivalent to the following optimization problem where we remove the constant terms.

$$\mathbf{p}^n = \arg \min_{\mathbf{q} \in \mathbf{Q}_f} \left[\int_{\Omega} \frac{1}{2} |\mathbf{q}|^2 d\mathbf{x} - \int_{\Omega} \nabla \mathbf{u}^n : \mathbf{q} d\mathbf{x} \right]. \quad (2.18)$$

Problem (2.18) can be solved pointwise because it does not involve any derivative for the variable $\mathbf{q}$. The solution can be obtained for all $\mathbf{x} \in \Omega$ as

$$\mathbf{p}(\mathbf{x}) = \arg \min_{\mathbf{q} \in \mathbf{E}_{f(\mathbf{x})}} \left[\frac{1}{2} |\mathbf{q}|^2 - \mathbf{b} : \mathbf{q} \right], \quad (2.19)$$

where $\mathbf{b} = \nabla \mathbf{u}^{n-1}(\mathbf{x})$ and $\mathbf{E}_{f(\mathbf{x})} = \{\mathbf{q}(\mathbf{x}) \in \mathbb{R}^{2 \times 2}, q_{11}(\mathbf{x}) q_{22}(\mathbf{x}) - q_{12}(\mathbf{x}) q_{21}(\mathbf{x}) = f(\mathbf{x})\}$, with

$$\mathbf{q} = \begin{pmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{pmatrix}.$$

Following [Glowinski and Tallec, 1989; Caboussat and Glowinski, 2018] with a proper change of variables, we can reduce the dimension of the problem and the corresponding solution. Let us denote the vectors $\vec{\mathbf{b}} = (b_{11}, b_{22}, b_{12}, b_{21})$, $b_{ij}, i, j = 1, 2$ to be the elements of $\mathbf{b}$, $\vec{\mathbf{q}} = (q_{11}, q_{22}, q_{12}, q_{21})$ and a 4×4 matrix as

$$\mathbf{S} = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 & 0 \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & 1/\sqrt{2} \\ 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} \end{pmatrix}.$$

We then introduce the new variables $\mathbf{y} = \mathbf{S} \vec{\mathbf{q}}^T$ and $\mathbf{a} = \mathbf{S} \vec{\mathbf{b}}^T$, so that the minimization problem in (2.19) is equivalent to

$$\min_{\mathbf{y} \in \mathbf{F}_f} \left[\frac{1}{2} |\mathbf{y}|^2 - \mathbf{a} \cdot \mathbf{y} \right], \quad (2.20)$$

with $\mathbf{F}_f = \{\mathbf{y} \in \mathbb{R}^4, y_1^2 - y_2^2 - y_3^2 + y_4^2 = 2f (> 0)\}$. In order to solve (2.20), we define the Lagrangian functional

$$L(\mathbf{y}, \tilde{\lambda}) = \frac{1}{2} |\mathbf{y}|^2 - \mathbf{a} \cdot \mathbf{y} - \frac{\tilde{\lambda}}{2} (y_1^2 - y_2^2 - y_3^2 + y_4^2 - 2f). \quad (2.21)$$

Let $\mathbf{z}$ denotes the solution of (2.20) and λ is the corresponding Lagrange multiplier. The first

order optimality conditions of (2.21) read:

$$\begin{aligned}
 z_1 - a_1 &= \lambda z_1, \\
 z_2 - a_2 &= -\lambda z_2, \\
 z_3 - a_3 &= -\lambda z_3, \\
 z_4 - a_4 &= \lambda z_4, \\
 z_1^2 - z_2^2 - z_3^2 + z_4^2 &= 2f.
 \end{aligned} \tag{2.22}$$

We re-write the first 4 equations of (2.22) in terms of $\mathbf{z}$ and we replace the variables $z_1, \dots, z_4$ into the fifth equation

$$\begin{aligned}
 z_1 &= \frac{a_1}{1 - \lambda}, \\
 z_2 &= \frac{a_2}{1 + \lambda}, \\
 z_3 &= \frac{a_3}{1 + \lambda}, \\
 z_4 &= \frac{a_4}{1 - \lambda}, \\
 \frac{a_1^2 + a_4^2}{(1 - \lambda)^2} - \frac{a_2^2 + a_3^2}{(1 + \lambda)^2} &= 2f.
 \end{aligned} \tag{2.23}$$

Finding the value of λ from the last equation allows to obtain $\mathbf{z}$, and then $\mathbf{q}$. Actually, the last equation of (2.23) can be written as a polynomial function of the variable λ

$$A\lambda^4 + B\lambda^2 + C\lambda + D = 0, \tag{2.24}$$

where

$$\begin{aligned}
 A &= -f, \\
 B &= \frac{1}{2}(a_1^2 - a_2^2 - a_3^2 + a_4^2 + 2f), \\
 C &= a_1^2 + a_2^2 + a_3^2 + a_4^2, \\
 D &= \frac{1}{2}(a_1^2 - a_2^2 - a_3^2 + a_4^2 - 2f).
 \end{aligned}$$

One can solve (2.24) exactly, although this task can be complex. We favor the use of the Newton method, and we target a solution λ that is close to zero. The underlying idea is that, if $\lambda = 0$, then the coefficient D should also be zero. The rationale behind targeting $D = 0$ is that D is actually equal to

$$D = \frac{1}{2}(\det \nabla \mathbf{u}^{n-1}(\mathbf{x}) - f(\mathbf{x})) \quad \forall \mathbf{x} \in \Omega,$$

and therefore $D = 0$ implies that $\mathbf{u}^{n-1}$ is the solution of (2.1) and $\lambda = 0$ from (2.23) implies that $\mathbf{q} = \nabla \mathbf{u}$. In the mentioned experiments, we observe that λ is close to zero.

When we consider the Jacobian inequality (2.4), the solution of the minimization problem for

all $\mathbf{x} \in \Omega$ reads as

$$\mathbf{p}^n(\mathbf{x}) = \arg \min_{\mathbf{q} \in \tilde{\mathbf{E}}_{f(\mathbf{x})}} \left[\frac{1}{2} |\mathbf{q}|^2 - \mathbf{b} : \mathbf{q} \right], \quad (2.25)$$

where

$$\tilde{\mathbf{E}}_{f(\mathbf{x})} = \{ \mathbf{q}(\mathbf{x}) \in \mathbb{R}^{2 \times 2}, \det \mathbf{q} = q_{11}(\mathbf{x}) q_{22}(\mathbf{x}) - q_{12}(\mathbf{x}) q_{21}(\mathbf{x}) \geq f(\mathbf{x}) \}$$

and $\mathbf{b} = \nabla \mathbf{u}^{n-1}$. The minimization problem (2.25) has four unknowns and one inequality constraint. In order to solve it, we introduce a slack variable and re-write (2.25) as

$$\begin{aligned} \min_{\mathbf{q} \in \mathbb{R}^{2 \times 2}} \quad & \left[\frac{1}{2} |\mathbf{q}|^2 - \mathbf{b} : \mathbf{q} \right] \\ \text{s.t.} \quad & \det \mathbf{q} - f - s = 0, \\ & s \geq 0. \end{aligned} \quad (2.26)$$

We use a logarithmic barrier function to eliminate the inequality constraint (see e.g. [Nocedal and Wright, 2006])

$$\begin{aligned} \min_{\mathbf{q} \in \tilde{\mathbf{E}}_f} \quad & \left[\frac{1}{2} |\mathbf{q}|^2 - \mathbf{b} : \mathbf{q} \right] - \mu \log s \\ \text{s.t.} \quad & \det \mathbf{q} - f - s = 0, \end{aligned} \quad (2.27)$$

where $\mu \geq 0$.

The minimization problem (2.27) has the same form as in (2.19), thus we can introduce the same change of variables. Namely, $\mathbf{y} = \mathbf{S}\tilde{\mathbf{q}}^T$ and $\mathbf{a} = \mathbf{S}\tilde{\mathbf{b}}^T$, therefore, the modified problem reads as

$$\arg \min_{\mathbf{y} \in \tilde{\mathbf{F}}_f} \left[\frac{1}{2} |\mathbf{y}|^2 - \mathbf{a} : \mathbf{y} \right], \quad (2.28)$$

with $\tilde{\mathbf{F}}_f = \{ \mathbf{y} \in \mathbb{R}^4, y_1^2 - y_2^2 - y_3^2 + y_4^2 = 2f + 2s \}$. As before, in order to minimize the objective function subjected to the constraint $\mathbf{y} \in \tilde{\mathbf{F}}_f$ we define the Lagrangian functional

$$L(\mathbf{y}, \tilde{\lambda}, s) = \frac{1}{2} |\mathbf{y}|^2 - \mathbf{a} : \mathbf{y} - \frac{\tilde{\lambda}}{2} (y_1^2 - y_2^2 - y_3^2 + y_4^2 - 2f - 2s) - \mu \log s. \quad (2.29)$$

Let $\mathbf{z}$ to be the solution of (2.28) and λ is associated to Lagrange multiplier. Applying the first order optimality conditions we have:

$$\begin{aligned} z_1 - a_1 &= \lambda z_1, \\ z_2 - a_2 &= -\lambda z_2, \\ z_3 - a_3 &= -\lambda z_3, \\ z_4 - a_4 &= \lambda z_4, \\ \frac{\mu}{s} &= \lambda, \\ z_1^2 - z_2^2 - z_3^2 + z_4^2 &= 2f + 2s. \end{aligned}$$

We isolate the first 5 equations with respect of $\mathbf{z}$ and replace them in the sixth equation

$$\begin{aligned} z_1 &= \frac{a_1}{1-\lambda}, \\ z_2 &= \frac{a_2}{1+\lambda}, \\ z_3 &= \frac{a_3}{1+\lambda}, \\ z_4 &= \frac{a_4}{1-\lambda}, \\ s &= \frac{\mu}{2\lambda}, \\ \frac{a_1^2 + a_4^2}{(1-\lambda)^2} - \frac{a_2^2 + a_3^2}{(1+\lambda)^2} &= 2f + 2s. \end{aligned}$$

The last equation of the above system can be written as a polynomial degree five

$$A\lambda^5 + B\lambda^4 + C\lambda^3 + D\lambda^2 + E\lambda + F = 0 \quad (2.30)$$

where

$$\begin{aligned} A &= 2f, \\ B &= 2\mu, \\ C &= -a_1^2 + a_2^2 + a_3^2 - a_4^2 - 4f, \\ D &= -2a_1^2 - 2a_2^2 - 2a_3^2 - 2a_4^2 - 4\mu, \\ E &= -a_1^2 + a_2^2 + a_3^2 - a_4^2 + 2f, \\ F &= 2\mu. \end{aligned}$$

Again we use a Newton method to solve (2.30).

2.2.4 Numerical solution of the linear variational problems

In this subsection we focus on the solution of **Step 3** of the relaxation algorithm. With the assumption that $\mathbf{p}^n$ is known, we look for the solution of $\mathbf{u}^n$ by solving the following optimization problem

$$\min_{\mathbf{v} \in \tilde{\mathbf{V}}_g} \left[\frac{1}{2} \int_{\Omega} |\nabla \mathbf{v} - \mathbf{p}^n|^2 d\mathbf{x} + \frac{\varepsilon}{2} \int_{\Omega} |\nabla^2 \mathbf{v}|^2 dx \right]. \quad (2.31)$$

In order to solve (2.31), we derive the first optimality conditions which read as: find $\mathbf{u}^{n+1/2} \in \tilde{\mathbf{V}}_g$ such that

$$\varepsilon \int_{\Omega} \nabla^2 \mathbf{u}^{n+1/2} : \nabla^2 \mathbf{v} d\mathbf{x} + \int_{\Omega} \nabla \mathbf{u}^{n+1/2} : \nabla \mathbf{v} d\mathbf{x} = \int_{\Omega} \mathbf{p}^n : \nabla \mathbf{v} d\mathbf{x}, \quad (2.32)$$

for all $\mathbf{v} \in \mathbf{V}_0$, where $\mathbf{V}_0 = \left\{ \mathbf{v} \in (H_0^1(\Omega) \cap H^2(\Omega))^2 \right\}$. Problem (2.32) is a fourth order partial differential equation. We solve it using a mixed method by introducing an auxiliary variable

$\mathbf{w} = -\nabla^2 \mathbf{u}^{n+1/2}$, see [Ciarlet, 2002].

Problem (2.32) becomes : find $\mathbf{w} \in (H_0^1(\Omega))^2$ such that

$$\varepsilon \int_{\Omega} \nabla \mathbf{w} : \nabla \mathbf{v} d\mathbf{x} + \int_{\Omega} \mathbf{w} \cdot \mathbf{v} d\mathbf{x} = \int_{\Omega} \mathbf{p}^n : \nabla \mathbf{v} d\mathbf{x}, \quad \forall \mathbf{v} \in (H_0^1(\Omega))^2, \quad (2.33)$$

together with: find $\mathbf{u}^{n+1/2} \in \mathbf{V}_g$ such that

$$\int_{\Omega} \nabla \mathbf{u}^{n+1/2} : \nabla \mathbf{v} d\mathbf{x} = \int_{\Omega} \mathbf{w} \cdot \mathbf{v} d\mathbf{x}, \quad \forall \mathbf{v} \in (H_0^1(\Omega))^2. \quad (2.34)$$

Problem (2.32) now splits in two well-posed second order elliptic equations (2.33) and (2.34), which can be solved using a standard piecewise linear finite element method.

2.3 Finite Element Approximation

Let $h > 0$ be a space discretization step and let $\{\mathcal{T}_h\}_h$ be family of conformal triangulations of Ω (see [Glowinski, 2008, Appendix 1]). On $\mathbf{Q}_h = \{\mathbf{q} \in L^2(\Omega)^{2 \times 2}, \forall T \in \mathcal{T}_h\}$, we define, respectively, the corresponding discrete inner product and corresponding norm as

$$((\mathbf{p}, \mathbf{q}))_{0h} = \sum_{T \in \mathcal{T}_h} |T| \mathbf{p}|_T : \mathbf{q}|_T, \quad |||\mathbf{q}|||_{0h} = \sqrt{((\mathbf{q}, \mathbf{q}))_{0h}}.$$

Let $\mathbf{Q}_{fh}$ and $\tilde{\mathbf{Q}}_{fh}$ be the finite dimensional subspaces of $\mathbf{Q}_f$ and $\tilde{\mathbf{Q}}_f$ given by

$$\mathbf{Q}_{fh} = \{\mathbf{q} \in \mathbf{Q}_h, \mathbf{q}|_T \in \mathbb{R}^{2 \times 2}, \det \mathbf{q}|_T = \bar{f}_T, \forall T \in \mathcal{T}_h\},$$

$$\tilde{\mathbf{Q}}_{fh} = \{\mathbf{q} \in \mathbf{Q}_h, \mathbf{q}|_T \in \mathbb{R}^{2 \times 2}, \det \mathbf{q}|_T \geq \bar{f}_T, \forall T \in \mathcal{T}_h\},$$

where $\bar{f}_T = \frac{1}{|T|} \int_T f(\mathbf{x}) d\mathbf{x}$. Let $\mathbf{V}_{g,h}$ and $\mathbf{V}_{0h}$ be the finite dimensional subspaces of $\mathbf{V}_g$ and $\mathbf{V}_0$ given by

$$\mathbf{V}_{g,h} = \left\{ \mathbf{v} \in \left(C^0(\bar{\Omega}) \right)^2, \mathbf{v}|_T \in (\mathbb{P}_1)^2, \forall T \in \mathcal{T}_h, \mathbf{v}|_{\Gamma_h} = \mathbf{g}_h \right\},$$

$$\mathbf{V}_{0h} = \left\{ \mathbf{v} \in \left(C^0(\bar{\Omega}) \right)^2, \mathbf{v}|_T \in (\mathbb{P}_1)^2, \forall T \in \mathcal{T}_h, \mathbf{v}|_{\Gamma_h} = \mathbf{0} \right\},$$

where $\mathbb{P}_1$ the space of the two-variable polynomials of degree ≤ 1 , and $\mathbf{g}_h$ is a piecewise linear interpolation of $\mathbf{g}$. We denote a discrete inner product and norm of $\mathbf{V}_{g,h}$ and $\mathbf{V}_{0h}$ as

$$(\mathbf{u}, \mathbf{v})_{0h} = \sum_{T \in \mathcal{T}_h} \sum_{i=1}^m W_i \mathbf{u}(\zeta_i) \cdot \mathbf{v}(\zeta_i), \quad |||\mathbf{u}|||_{0h} = \sqrt{(\mathbf{u}, \mathbf{u})_{0h}},$$

with W_i the weights and ζ_i the evaluation points of a Gauss quadrature rule, m denotes the number of points of the quadrature rule.

The discrete formulation of the least squares method to solve (2.9) reads as: find $\{\mathbf{u}, \mathbf{p}\} \in \mathbf{V}_{\mathbf{g},h} \times \mathbf{Q}_{f,h}$ such that

$$J_{\varepsilon,h}(\mathbf{u}, \mathbf{p}) \leq J_{\varepsilon,h}(\mathbf{v}, \mathbf{q}), \forall \{\mathbf{v}, \mathbf{q}\} \in \mathbf{V}_{\mathbf{g},h} \times \mathbf{Q}_{f,h},$$

with

$$J_{\varepsilon,h}(\mathbf{u}, \mathbf{p}) = |||\nabla \mathbf{u}_h - \mathbf{p}_h|||_{0h}^2.$$

Similarly, the discrete formulation of the least squares method to solve (2.10) reads as: find $\{\mathbf{u}, \mathbf{p}\} \in \mathbf{V}_{\mathbf{g},h} \times \tilde{\mathbf{Q}}_{f,h}$ such that

$$J_{\varepsilon,h}(\mathbf{u}, \mathbf{p}) \leq J_{\varepsilon,h}(\mathbf{v}, \mathbf{q}), \forall \{\mathbf{v}, \mathbf{q}\} \in \mathbf{V}_{\mathbf{g},h} \times \tilde{\mathbf{Q}}_{f,h}.$$

The discrete formulation of the ADMM method that is described in Section 2.2.2 becomes:

Step 1. The initialization of the algorithm reads as: find $\mathbf{u}_h^1 \in \mathbf{V}_{\mathbf{g},h}$

$$((\nabla \mathbf{u}_h^1, \nabla \mathbf{v}_h))_{0h} = (\tilde{f}, \mathbf{v}_h), \forall \mathbf{v}_h \in \mathbf{V}_{0h}, \quad (2.35)$$

where $\tilde{f} = (1, 1) \forall \mathbf{x} \in \Omega$.

For $n \geq 1$,

Step 2. The discrete minimization problem of problem (2.18) is written as

$$\mathbf{p}_h^n = \arg \min_{\mathbf{q}_h \in \mathbf{Q}_{f,h}} \left[\frac{1}{2} |||\mathbf{p}_h|||_{0h}^2 - ((\mathbf{u}_h^n, \mathbf{q}_h))_{0h} \right], \quad (2.36)$$

while in the case of Jacobian inequality (2.4), we replace $\mathbf{Q}_{f,h}$ in (2.36) by $\tilde{\mathbf{Q}}_{f,h}$. The solution of the minimization problem $\mathbf{p}_h^n$ can be obtained as described in Section 2.2.3. We could compute a solution of $\mathbf{p}_h^n$ for multiple points in any $T \in \mathcal{T}_h$ however, because $\nabla \mathbf{u}_h^n$ is a constant per $T \in \mathcal{T}_h$, it is sufficient to compute $\mathbf{p}_h^n$ only once per T . Therefore, we look for solution $\mathbf{p}_h^n(T), \forall T \in \mathcal{T}_h$, by providing $\nabla \mathbf{u}_h^n(T)$ and $\tilde{f}_T$ for every triangle in the given mesh.

Step 3. To obtain $\mathbf{u}_h^{n+1/2}$, we discretize (2.32) and get

$$\varepsilon (\nabla^2 \mathbf{u}_h^{n+1/2}, \nabla^2 \mathbf{v}_h)_{0h} + ((\nabla \mathbf{u}_h^{n+1}, \nabla \mathbf{v}_h))_{0h} = ((\mathbf{p}_h^n, \nabla \mathbf{v}_h))_{0h}. \quad (2.37)$$

By applying a mixed finite element method to (2.37), as applied to (2.32) in Section 2.2.4, we obtain: find $(\mathbf{u}_h^{n+1/2}, \mathbf{w}_h^n) \in \mathbf{V}_{\mathbf{g},h} \times \mathbf{V}_{0h}$ such that,

$$\begin{cases} \varepsilon ((\nabla \mathbf{w}_h^n, \nabla \mathbf{v}_h))_{0h} + (\mathbf{w}_h^n, \mathbf{v}_h)_{0h} = ((\mathbf{p}_h^n, \nabla \mathbf{v}_h))_{0h}, \forall \mathbf{v}_h \in \mathbf{V}_{0h}, \text{ and} \\ ((\nabla \mathbf{u}_h^{n+1/2}, \nabla \boldsymbol{\varphi}_h))_{0h} = (\mathbf{w}_h^n, \boldsymbol{\varphi}_h)_{0h}, \forall \boldsymbol{\varphi}_h \in \mathbf{V}_{0h}. \end{cases}$$

Step 4. The update of the solution is given by

$$\mathbf{u}_h^{n+1} = \mathbf{u}_h^n + \omega (\mathbf{u}_h^{n+1/2} - \mathbf{u}_h^n).$$

We study the convergence properties of the above algorithm in the next section with several numerical experiments.

2.4 Numerical Experiments

In this section, we test our algorithm for different numerical examples. We consider the following domains: the unit square

$$\Omega_q = (0, 1)^2,$$

the unit disk domain,

$$\Omega_d = \{\mathbf{x} \in \mathbb{R}^2, \|\mathbf{x}\|_2 < 1\},$$

the so-called, pacman domain

$$\Omega_p = \Omega_d \setminus \{(x_1, x_2) \in \mathbb{R}^2, x > 0, |x_2| < x_1\},$$

and the cracked unit disk

$$\Omega_c = \Omega_d \setminus \{(x_1, x_2) \in \mathbb{R}^2, x > 0, |x_2| < \tan\left(\frac{\pi}{100}\right) x_1\}.$$

The discretization (meshes) of these domains are showed in Figure 2.1.

The algorithm initialization is performed as following: for the coarsest mesh, we solve equation (2.35) with the given boundary data. For every other mesh size, we use the numerical solution obtained by the coarser mesh. In all experiments the smoothing parameter ε is chosen as h^2 unless otherwise specified.

The relaxation parameter ω is chosen to be initially close to 1, and gradually increases to 2. The stopping criterion for the relaxation algorithm is when the difference between two successive iterations are smaller than the tolerance 10^{-8} , i.e., $\|\mathbf{u}^n - \mathbf{u}^{n-1}\|_{L^2(\Omega)} < 10^{-8}$.

Newton's method stopping criterion for the local nonlinear problems is when two successive iterations are smaller than 10^{-15} . Usually, the number of iterations of the Newton method does not exceed 5. In order to have an estimation of the multipliers λ on each mesh based on values associated with every triangle we denote λ_T to be the value of the λ for the triangle $T \in \mathcal{T}_h$ and N the number of triangles in the mesh then

$$\bar{\lambda} = \frac{\sum_{T \in \mathcal{T}_h} \lambda_T}{N},$$

Chapter 2. Numerical Approximation of the Prescribed Jacobian Equation/Inequality

is the average value. Another relevant quantity that describes $\bar{\lambda}$ is the standard deviation, estimated by

$$\bar{\sigma} = \sqrt{\frac{\sum_{T \in \mathcal{T}_h} (\lambda_T - \bar{\lambda})^2}{N - 1}}.$$

The parameter μ used to solve (2.4) will specified later.

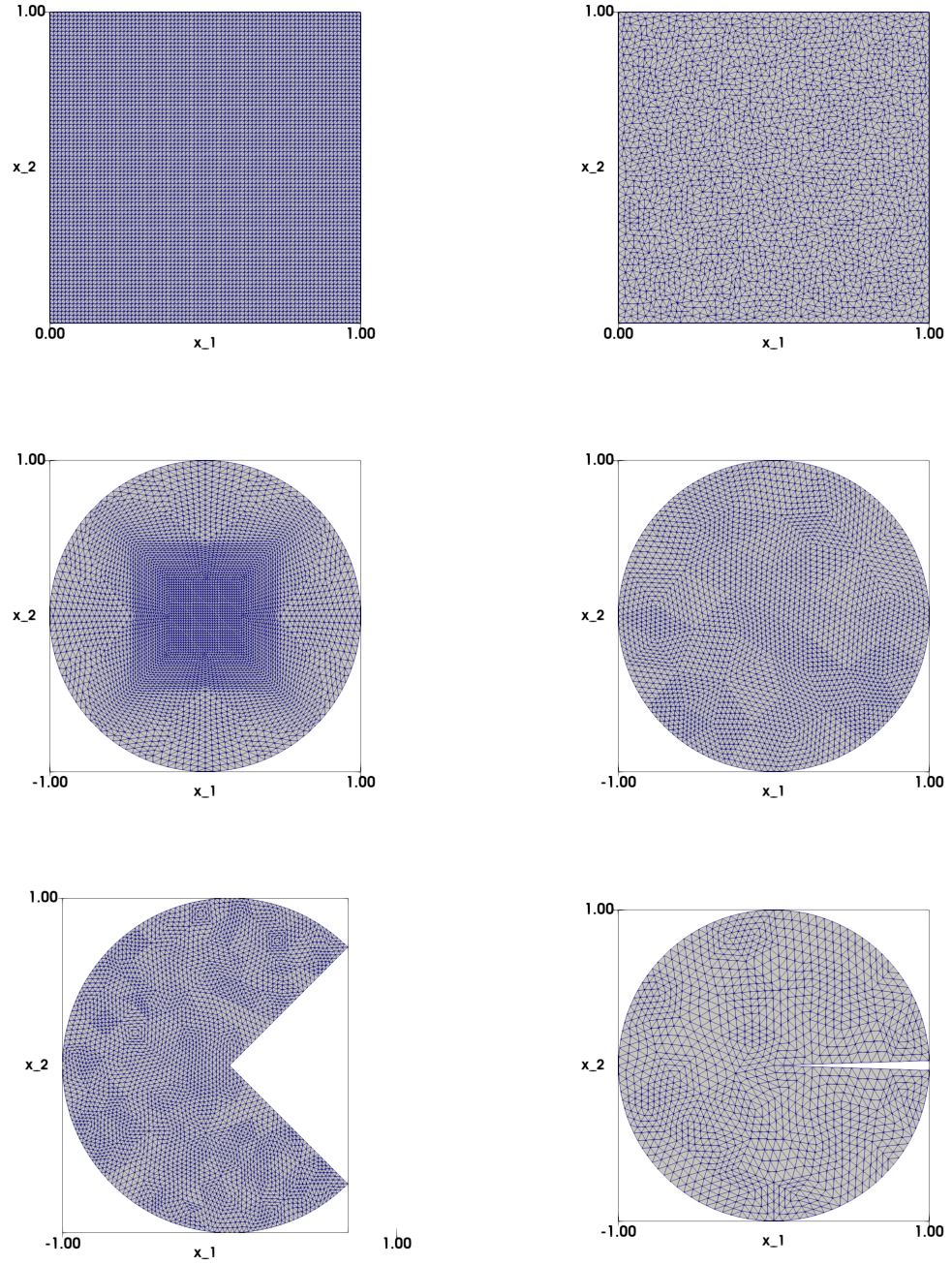


Figure 2.1 – Finite element meshes used for the numerical experiments. Top left: structured mesh for the unit square ($\Omega_q = (0, 1)^2$, $h = 0.0125$); Top right: unstructured mesh for the unit square ($\Omega_q = (0, 1)^2$, $h = 0.01882$); Middle left: structured mesh for the unit disk ($\Omega_d = \{(x_1, x_2) \in \mathbb{R}^2 : x_1^2 + x_2^2 < 1\}$, $h \simeq 0.0209$); Middle right: unstructured mesh for the unit disk ($\Omega_d = \{(x_1, x_2) \in \mathbb{R}^2 : x_1^2 + x_2^2 < 1\}$, $h \simeq 0.08$); Bottom left: unstructured mesh for the pacman domain ($\Omega_p = \Omega_d \setminus \{(x_1, x_2) \in \mathbb{R}^2, x_1 > 0, |x_2| < x_1\}$, $h = 0.0252$); Bottom right: unstructured mesh for the cracked unit disk ($\Omega_c = \Omega_d \setminus \{(x_1, x_2) \in \mathbb{R}^2, x_1 > 0, |x_2| < \tan(\frac{\pi}{100})x_1\}$, $h = 0.0486$).

2.4.1 Identity map

The first experiment corresponds to considering the identity map, $\mathbf{u}(\Omega) = \Omega$ as the exact solution. The data are chosen as $f = 1$ and $\mathbf{u}(\mathbf{x}) = \mathbf{x}$ on Γ . The problem can be written as

$$\begin{cases} \det \nabla \mathbf{u}(\mathbf{x}) = 1 & \text{in } \Omega, \\ \mathbf{u}(\mathbf{x}) = \mathbf{x} & \text{on } \Gamma. \end{cases} \quad (2.38)$$

When $\Omega = (0, 1)^2$, we use structured meshes with mesh size $h = \{0.00625, 0.025, 0.0125, 0.05\}$, and obtain estimations in the $L^2(\Omega)$ error norm of order 10^{-10} and $H^1(\Omega)$ error norm of order 10^{-9} to 10^{-10} . In addition, $\|\nabla \mathbf{u}_h - \mathbf{p}_h\|_{L^2(\Omega)}$ and $\bar{\lambda}$ are of the order of 10^{-10} . Figure 2.2 illustrates the two components of the numerical solution when using the structured mesh. More precisely, with $h = 0.025$, after 29 iterations, we obtain

$$\|\mathbf{u} - \mathbf{u}_h\|_{L^2(\Omega)} = 1.24 \cdot 10^{-10}, \quad |\mathbf{u} - \mathbf{u}_h|_{H^1(\Omega)} = 7.09 \cdot 10^{-10},$$

and

$$\|\nabla \mathbf{u}_h - \mathbf{p}_h\|_{L^2(\Omega)} = 4.21 \cdot 10^{-10}, \quad \bar{\lambda} = 2.96 \cdot 10^{-10}.$$

When

$$\Omega = \{(x_1, x_2) \in \mathbb{R}^2, x_1^2 + x_2^2 < 1\},$$

we use structured meshes with mesh size $h = \{0.0831, 0.0421, 0.0209, 0.0104\}$. Similar to the previous setting, we get estimations in the $L^2(\Omega)$ error norm of order 10^{-10} and $H^1(\Omega)$ error norm of order 10^{-9} to 10^{-10} . As well, $\|\nabla \mathbf{u}_h - \mathbf{p}_h\|_{L^2(\Omega)}$ and $\bar{\lambda}$ are of the order of 10^{-10} to 10^{-11} . Figure 2.3 illustrates the two components of the numerical solution in the disk structured mesh. In particular, for $h = 0.025$, and after 29 iterations, we obtain

$$\|\mathbf{u} - \mathbf{u}_h\|_{L^2(\Omega)} = 2.06 \cdot 10^{-10}, \quad |\mathbf{u} - \mathbf{u}_h|_{H^1(\Omega)} = 7.24 \cdot 10^{-10},$$

and

$$\|\nabla \mathbf{u}_h - \mathbf{p}_h\|_{L^2(\Omega)} = 2.06 \cdot 10^{-10}, \quad \bar{\lambda} = 9.77 \cdot 10^{-11}.$$

Remark. Note that solving the above test case is the same as solving (1.1) in Chapter 1 with data

$$\begin{cases} \nabla \mathbf{u} \in \mathcal{O}(2) & \text{in } \Omega, \\ \mathbf{u} = \mathbf{x} & \text{on } \partial\Omega. \end{cases}$$

We showed that (1.1) satisfies (1.7), that is $\det \nabla \mathbf{u}(\mathbf{x}) = \pm 1$, a.e. $\mathbf{x} \in \Omega$. In this specific test case, we have only the positive sign $\det \nabla \mathbf{u}(\mathbf{x}) = 1$, $\mathbf{x} \in \Omega$, which is equivalent to (2.38).

Next, we test the algorithm when (2.4) is considered. Therefore we examine the identity map

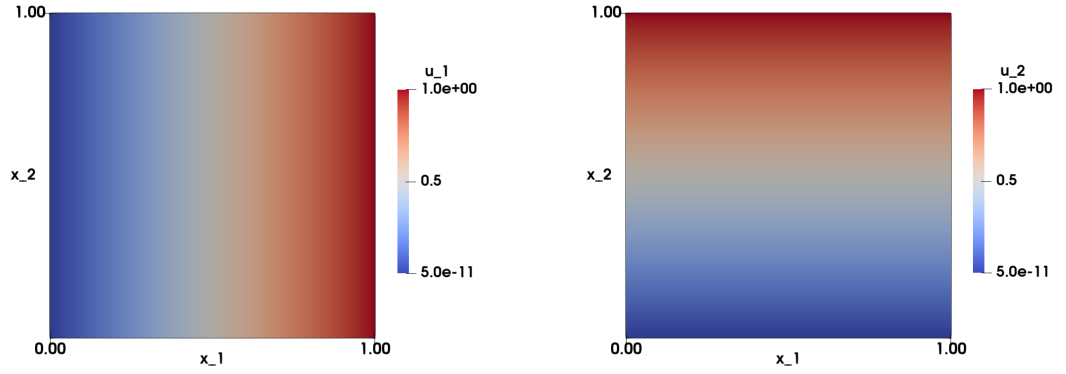


Figure 2.2 – Identity map test case with data $f = 1$ and $\mathbf{g} = \mathbf{x}$. Visualization of the numerical approximation of the solution $\mathbf{u}_h$; Left: the component $u_{1,h}$; Right: the component $u_{2,h}$. Results are obtained on structured mesh with $h = 0.0125$.

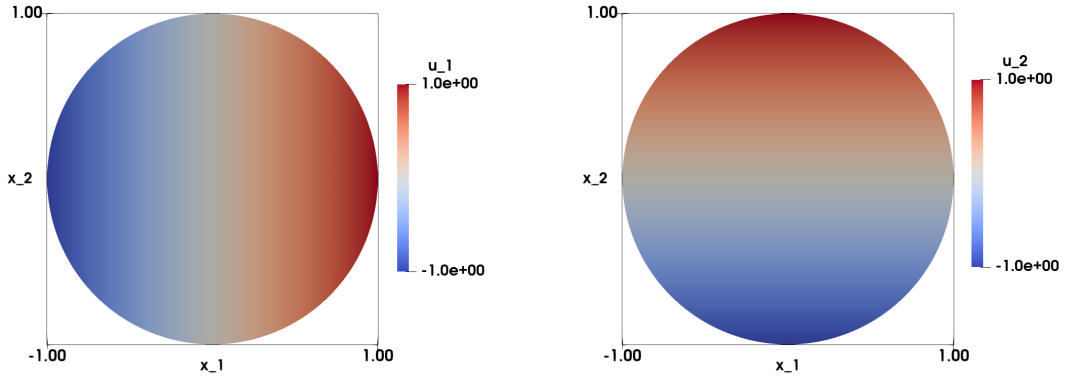


Figure 2.3 – Identity map test case with data $f = 1$ and $\mathbf{u} = \mathbf{x}$ on Γ . Visualization of the numerical approximation of the solution $\mathbf{u}_h$; Left: the component $u_{1,h}$; Right: the component $u_{2,h}$. Results are obtained on structured mesh with $h = 0.0209$.

test case with $f = 0$. The problem is written as

$$\begin{cases} \det \nabla \mathbf{u}(\mathbf{x}) \geq 0 \text{ in } \Omega, \\ \mathbf{u}(\mathbf{x}) = \mathbf{x} \text{ on } \Gamma, \end{cases} \quad (2.39)$$

with $\Omega = \{(x_1, x_2) \in \mathbb{R}^2, x_1^2 + x_2^2 < 1\}$.

In (2.39) there is no information on the right-hand side since $f = 0$, the only available informa-

tion is on the boundary. Despite this lack of information, the algorithm converges to an exact solution $\mathbf{u}(\mathbf{x}) = \mathbf{x}$, similarly as in (2.38). The results obtained solving (2.39) are similar to those obtained by solving (2.38).

By contrast to solving the Jacobian equality in (2.38), numerical errors in $L^2(\Omega)$ and $H^1(\Omega)$ norms of solving the inequality (2.39) using the same mesh size values are of the order of 10^{-2} and 10^{-3} , respectively. These parameters depend on the choice of the μ . If initially μ is chosen to be large $\|\nabla \mathbf{u}_h - \mathbf{p}_h\|_{L^2(\Omega)}$ will be also large. On the other hand, if μ is small, this may cause some convergence problems. In this test case we choose $\mu = 0.1$ and we decrease it with a factor of $\sqrt{h}$. For example, for $h = 0.025$, after 69 iterations we obtain

$$\|\mathbf{u} - \mathbf{u}_h\|_{L^2(\Omega)} = 6.86 \cdot 10^{-10}, \quad |\mathbf{u} - \mathbf{u}_h|_{H^1(\Omega)} = 2.67 \cdot 10^{-9},$$

and

$$\|\nabla \mathbf{u}_h - \mathbf{p}_h\|_{L^2(\Omega)} = 1.04 \cdot 10^{-2}, \quad \bar{\lambda} = 2.08 \cdot 10^{-3}.$$

2.4.2 Identity map with periodic perturbation

In this test case, we have the same boundaries as before, and we introduce some perturbation inside the domain. We define the right-hand side to be the following function

$$\begin{aligned} f(\mathbf{x}) = & 1 + \alpha\pi[\cos(\pi x_1) \sin(2\pi x_2) + \sin(2\pi x_2) \cos(\pi x_2)] \\ & + \alpha^2\pi^2[\cos(\pi x_1) \cos(\pi x_2) \sin(2\pi x_1) \sin(2\pi x_2)] \\ & - 4 \sin(\pi x_1) \sin(\pi x_2) \cos(2\pi x_1) \cos(2\pi x_2), \end{aligned} \quad (2.40)$$

with $\alpha = 1/(5\pi)$, chosen to ensure that f is greater than 0 for all $\mathbf{x} \in \Omega$. Then, the problem is written as

$$\begin{cases} \det \nabla \mathbf{u} = f \text{ in } \Omega, \\ \mathbf{u}(\mathbf{x}) = \mathbf{x} \text{ on } \Gamma. \end{cases} \quad (2.41)$$

The exact solution of (2.41) is

$$\mathbf{u}(\mathbf{x}) = \begin{pmatrix} x_1 + a \sin(\pi x_1) \sin(2\pi x_2) \\ x_2 + a \sin(2\pi x_1) \sin(\pi x_2) \end{pmatrix}.$$

Although the algorithm converges to the provided exact solution, we can not prove the uniqueness of the solution of (2.41). Table 2.1 illustrates the convergence of the numerical solution and gives information about the relaxation algorithm with a structured mesh on the unit square. More precisely, Table 2.1 shows that :

- i the numerical solution converges in L^2 -norm with a rate of $O(h)$ to $O(h^{1.4})$ and in H^1 -semi norm with an optimal rate of $O(h)$ when $h \rightarrow 0$.
- ii the error between of the numerical solution $\nabla \mathbf{u}_h$ and the auxiliary variable $\mathbf{p}_h$ in L^2 -norm

is decreasing with an order greater than $O(h)$.

- iii the number of iterations of the relaxation algorithm is linearly increasing.
- iv based on the estimated average and standard deviation $\bar{\lambda}$ and $\bar{\sigma}$, the range of the 95% confidence interval $[\bar{\lambda} - 2\bar{\sigma}, \bar{\lambda} + 2\bar{\sigma}]$ converges to zero when $h \rightarrow 0$. This is in adequation with the theory outlined before.

Similar results are observed in Table 2.2, where the unstructured mesh is used. We observe that the numerical solution converges in L^2 -norm and H^1 -semi norm with a rate greater than $O(h)$. Note that the error between the numerical solution $\mathbf{u}_h$ and the auxiliary variable $\mathbf{p}_h$ in L^2 -norm decreases for decreasing h . The iterations of the relaxation algorithm are linearly increasing. Although we cannot directly compare the results of Table 2.1 and Table 2.2 because of the difference in mesh size and the number of elements, we can claim that the algorithm is not sensitive to the structure of the mesh.

Table 2.1 – Identity map with periodic perturbation test case with data f given in (2.40) and $\mathbf{u} = \mathbf{x}$ on Γ . Computations with various mesh sizes h ; L^2 and H^1 error norms with the corresponding rates; Computation of the difference between $\nabla \mathbf{u}_h$ and $\mathbf{p}_h$ in L^2 norm; Average value of $\bar{\lambda}$ with its standard deviation $\bar{\sigma}$; Iterations of the relaxation algorithm. The results are obtained with a structured mesh.

h	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2(\Omega)}$		$\ \mathbf{u} - \mathbf{u}_h\ _{H^1(\Omega)}$		$\ \nabla \mathbf{u}_h - \mathbf{p}_h\ _{L^2(\Omega)}$	$\bar{\lambda}(\bar{\sigma})$	iter
1/20	2.71e-02		1.93e-01		7.58e-02	-6.97e-03(0.054)	19
1/40	1.29e-02	1.03	9.37e-02	1.00	3.42e-02	-3.08e-03(0.024)	49
1/80	4.84e-03	1.40	3.85e-02	1.18	1.11e-02	-8.19e-04(0.008)	149
1/160	2.22e-03	1.18	2.26e-02	0.94	3.76e-03	-1.35e-04(0.002)	359

Table 2.2 – Identity map with periodic perturbation test case with data f given in (2.40) and $\mathbf{u} = \mathbf{x}$ on Γ . Computations with various mesh sizes h ; L^2 and H^1 error norms with the corresponding rates; Computation of the difference between $\nabla \mathbf{u}_h$ and $\mathbf{p}_h$ in L^2 norm; Average value of $\bar{\lambda}$ with its standard deviation $\bar{\sigma}$; Iterations of the relaxation algorithm. The results are obtained with a unstructured mesh.

h	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2(\Omega)}$		$\ \mathbf{u} - \mathbf{u}_h\ _{H^1(\Omega)}$		$\ \nabla \mathbf{u}_h - \mathbf{p}_h\ _{L^2(\Omega)}$	$\bar{\lambda}(\bar{\sigma})$	iter
0.0622	3.49e-02		2.45e-01		1.02e-01	-2.13e-02(0.076)	19
0.0377	2.06e-02	0.76	1.46e-01	0.74	5.76e-02	-8.35e-03(0.046)	39
0.0188	8.57e-03	1.26	6.33e-02	1.20	2.24e-02	-4.38e-03(0.018)	89
0.0094	3.28e-03	1.38	2.87e-02	1.14	7.06e-03	-6.75e-04(0.005)	249

Figure 2.4 illustrates the two components of the numerical solution on the quad structured mesh. Figure 2.5 visualizes $\det \nabla \mathbf{u}_h$ (top left) and $\det \mathbf{p}_h$ (top right), where we can observe that both quantities are similar with small difference on the boundaries. On the bottom left shows

the normalized numerical solution of $\|\mathbf{u}_h\|$. On the bottom right the vectors are defined by the components $[u_1(x_1, x_2), u_2(x_1, x_2)]$ and are displayed at the node of the mesh $[x_1, x_2] \in \Omega$. The direction of the vectors shows the stretch of Ω that results to the map $\mathbf{u}(\Omega)$.

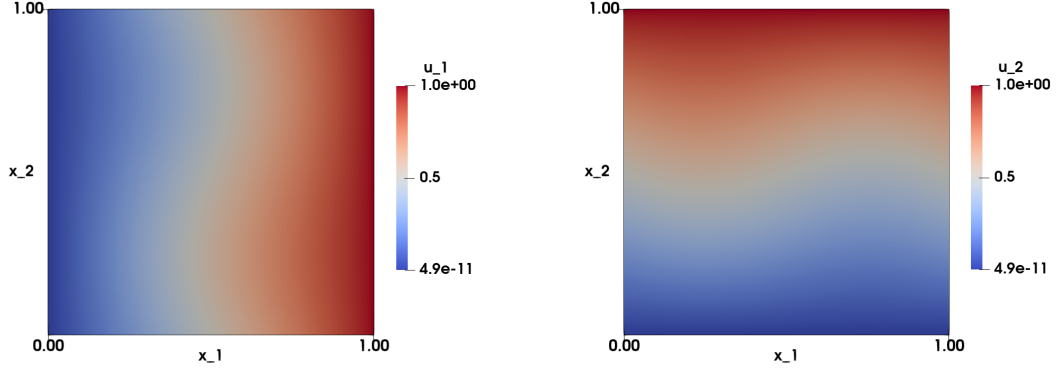


Figure 2.4 – Identity map with periodic perturbation test case with data f given in (2.40) and $\mathbf{u} = \mathbf{x}$ on Γ . Visualization of the numerical approximation of the solution $\mathbf{u}_h$; Left: the component $u_{1,h}$; Right: the component $u_{2,h}$. The results are obtained on structured mesh with $h = 0.0125$.

2.4.3 Smooth solution with radial symmetric right-hand side

Let $\Omega = \{(x_1, x_2) \in \mathbb{R}^2, x_1^2 + x_2^2 < 1\}$ be the unit disk. We consider the following problem

$$\begin{cases} \det \nabla \mathbf{u} = 2(x_1^2 + x_2^2) & \text{in } \Omega, \\ \mathbf{u}(\mathbf{x}) = \mathbf{g}(\mathbf{x}) & \text{on } \Gamma, \end{cases} \quad (2.42)$$

where

$$\mathbf{g}(\mathbf{x}) = \sqrt{2} \begin{pmatrix} \frac{1}{2}(x_1^2 - x_2^2) \\ x_1 x_2 \end{pmatrix}.$$

The exact solution satisfies

$$\mathbf{u}(\mathbf{x}) = \sqrt{2} \begin{pmatrix} \frac{1}{2}(x_1^2 - x_2^2) \\ x_1 x_2 \end{pmatrix}. \quad (2.43)$$

The non-uniqueness of the solution of equation (2.42) makes this test case challenging. For instance, this may cause the algorithm to oscilate between two different solutions. Later on, we show that our algorithm converges to a solution for every different set of parameters. The numerical solution of (2.42) is illustrated in the top row of Figure 2.6. Note that both the right-hand side of this equation, $2(x_1^2 + x_2^2)$, and $\|\mathbf{u}_h\|$ (bottom left), are radial symmetric while the solution, $\mathbf{u}_h$, does not have the same symmetry pattern. The numerical approximation of $\det \nabla \mathbf{u}_h$ and $\det \mathbf{p}_h$ are presented in the second row, we see that the solutions are identical.

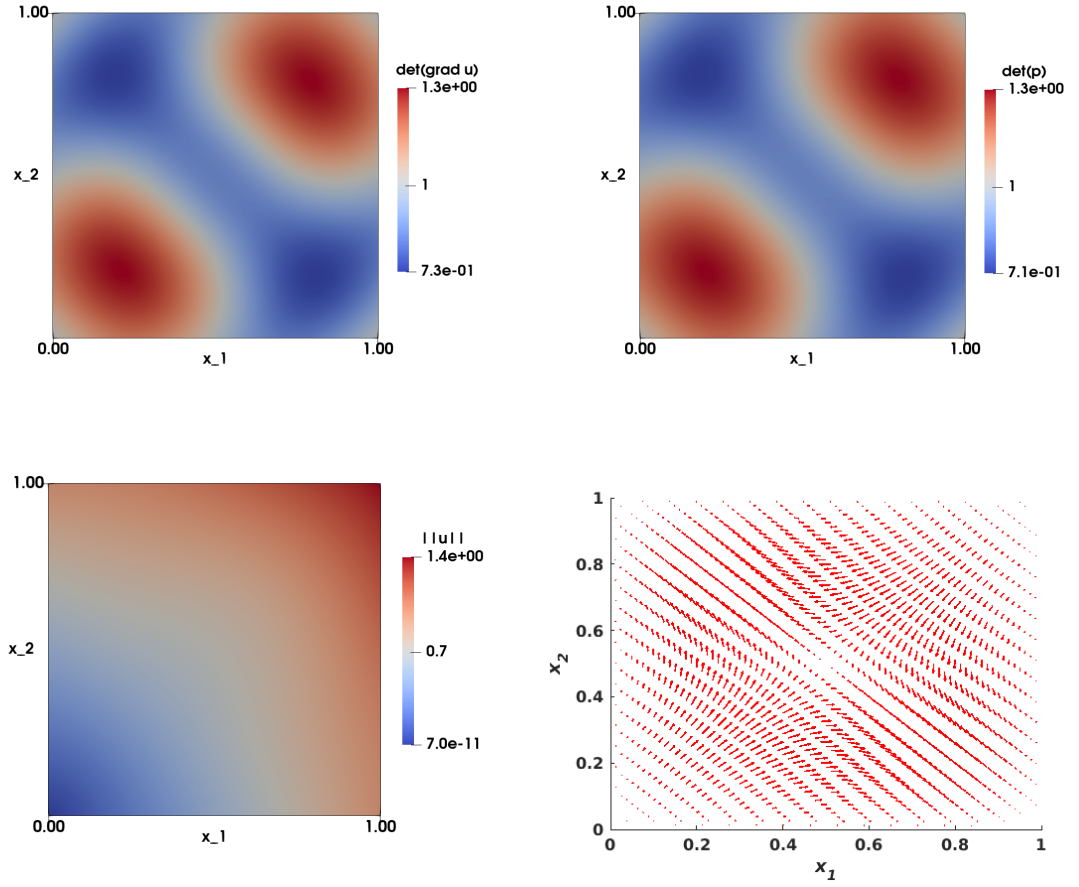


Figure 2.5 – Identity map with periodic perturbation test case with data f given in (2.40) and $\mathbf{u} = \mathbf{x}$ on Γ . Top left: Numerical approximation of the solution $\det \nabla \mathbf{u}_h$. Top right: Numerical approximation of the solution $\det \nabla \mathbf{p}_h$. Bottom left: Numerical approximation of the solution $\|\mathbf{u}_h\|$. Bottom right: Visualization of the vector field $\mathbf{u}_h$. The results are obtained on structured mesh with $h = 0.0125$.

Table 2.3 provides insights about the convergence of the numerical solution and the relaxation algorithm on the structured mesh for the unit disk. The numerical aspects of the algorithm enumerated in the previous section still holds with the only difference being that the numerical solution converges in L^2 -norm with a rate of $O(h^{1.7})$ to $O(h^2)$ and in H^1 semi-norm with an optimal rate of $O(h)$.

Similar results are observed in Table 2.4, where the unstructured mesh is used. We observe that the numerical solution converges in L^2 -norm and H^1 -semi norm with rate of $O(h^{1.9})$ and $O(h)$, respectively. This confirms that the algorithm does not depend on the structure of the mesh.

Chapter 2. Numerical Approximation of the Prescribed Jacobian Equation/Inequality

Table 2.3 – Smooth solution with radial symmetric right-hand side test case with data $f = 2(x^2 + y^2)$ and $\mathbf{u} = \sqrt{2}(\frac{1}{2}(x_1^2 - x_2^2), x_1 x_2)^T$ on Γ . Computations with various mesh sizes h ; L^2 and H^1 error norms with the corresponding rates; Computation of the difference between $\nabla \mathbf{u}_h$ and $\mathbf{p}_h$ in L^2 norm; Average value of $\bar{\lambda}$ and the standard deviation $\bar{\sigma}$; Iterations of the relaxation algorithm. The results are obtained on a structured mesh.

h	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2(\Omega)}$		$ \mathbf{u} - \mathbf{u}_h _{H^1(\Omega)}$		$\ \nabla \mathbf{u}_h - \mathbf{p}_h\ _{L^2(\Omega)}$	$\bar{\lambda}(\bar{\sigma})$	iter
0.0831	5.95e-03		2.14e-01		9.05e-02	1.37e-02(0.063)	19
0.0421	1.27e-03	2.19	1.03e-01	1.05	4.25e-02	7.50e-03(0.036)	39
0.0209	3.38e-04	1.91	5.22e-02	0.98	2.17e-02	3.58e-03(0.020)	79
0.0104	1.05e-04	1.70	2.63e-02	0.98	1.10e-02	1.76e-03(0.011)	219

Table 2.4 – Smooth solution with radial symmetric right-hand side test case with data $f = 2(x^2 + y^2)$ and $\mathbf{u} = \sqrt{2}(\frac{1}{2}(x_1^2 - x_2^2), x_1 x_2)^T$ on Γ . Computations with various mesh sizes h ; L^2 and H^1 error norms with the corresponding rates; Computation of the difference between $\nabla \mathbf{u}_h$ and $\mathbf{p}_h$ in L^2 norm; Average value of $\bar{\lambda}$ and the standard deviation $\bar{\sigma}$; Iterations of the relaxation algorithm. The results are obtained on an unstructured mesh.

h	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2(\Omega)}$		$ \mathbf{u} - \mathbf{u}_h _{H^1(\Omega)}$		$\ \nabla \mathbf{u}_h - \mathbf{p}_h\ _{L^2(\Omega)}$	$\bar{\lambda}(\bar{\sigma})$	iter
0.1327	5.79e-03		2.25e-01		1.14e-01	1.31e-02(0.062)	19
0.0665	1.53e-03	1.92	1.13e-01	1.00	5.74e-02	5.88e-03(0.038)	29
0.0332	4.06e-04	1.92	5.65e-02	1.00	2.85e-02	2.15e-03(0.022)	69
0.0166	1.30e-04	1.64	2.83e-02	1.00	1.42e-02	1.02e-03(0.012)	199

Let us briefly consider the same problem without providing the right-hand side ($f = 0$), and we solve the Jacobian inequality. Again, let $\Omega = \{(x_1, x_2) \in \mathbb{R}^2, x_1^2 + x_2^2 < 1\}$ to be the unit disk, the right-hand side is given by $f = 0$, and we have the same function $\mathbf{g}(\mathbf{x})$ on the boundary. The problem reads as

$$\begin{cases} \det \nabla \mathbf{u} \geq 0 \text{ in } \Omega, \\ \mathbf{u}(\mathbf{x}) = \mathbf{g}(\mathbf{x}) \text{ on } \Gamma. \end{cases} \quad (2.44)$$

The exact solution is estimated numerically, and it is equal to the solution of solving the Jacobian equality (2.43). Table 2.5 shows results obtained on the structured mesh. The parameter μ for the solution of the local non-linear problems is chosen to be 0.1 and we decrease it by $0.001\sqrt{h}$ at each iteration of the interior point method. We can observe that the numerical solution in this case converges in L^2 -norm and H^1 -semi norm with a rate of $O(h^{1.9})$ and $O(h)$, respectively.

Table 2.5 – Smooth solution with radial symmetric right-hand side inequality test case with data $f = 0$ and $\mathbf{u} = \sqrt{2} \left(\frac{1}{2} (x_1^2 - x_2^2), x_1 x_2 \right)^T$ on Γ . Computations with various mesh sizes h ; L^2 and H^1 error norms with the corresponding rates; Computation of the difference between $\nabla \mathbf{u}_h$ and $\mathbf{p}_h$ in L^2 norm; Average value of $\bar{\lambda}$ and the standard deviation $\bar{\sigma}$; Iterations of the relaxation algorithm. The results are obtained on the unit disk equipped by a structured mesh.

h	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2(\Omega)}$		$\ \mathbf{u} - \mathbf{u}_h\ _{H^1(\Omega)}$		$\ \nabla \mathbf{u}_h - \mathbf{p}_h\ _{L^2(\Omega)}$	$\bar{\lambda}(\bar{\sigma})$	iter
0.0831	6.12e-03		2.14e-01		1.08e-03	7.42e-04(0.010)	59
0.0421	1.33e-03	2.20	1.03e-01	1.05	4.74e-04	2.94e-04(0.007)	99
0.0209	3.48e-04	1.93	5.23e-02	0.98	3.20e-04	2.42e-04(0.007)	119
0.0104	9.36e-05	1.89	2.63e-02	0.99	2.09e-04	1.62e-04(0.005)	219

2.4.4 Smooth radial symmetric solution with non-smooth gradient

Let $\Omega = \{(x_1, x_2) \in \mathbb{R}^2, x_1^2 + x_2^2 < 1\}$ be the unit disk, the right hand side is given by $f = 2(x_1^2 + x_2^2)$, the boundary data is given by $\mathbf{g}(\mathbf{x}) = \mathbf{x}$. The problem (2.1) with these data reads as

$$\begin{cases} \det \nabla \mathbf{u} = 2(x_1^2 + x_2^2) & \text{in } \Omega, \\ \mathbf{u}(\mathbf{x}) = \mathbf{x} & \text{on } \Gamma. \end{cases} \quad (2.45)$$

Problem (2.45) admits an exact solution

$$\mathbf{u}(\mathbf{x}) = \sqrt{x_1^2 + x_2^2} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix},$$

and the gradient is

$$\nabla \mathbf{u}(\mathbf{x}) = \frac{1}{\sqrt{x_1^2 + x_2^2}} \begin{pmatrix} 2x_1^2 + x_2^2 & x_1 x_2 \\ x_1 x_2 & x_2^2 + 2x_1^2 \end{pmatrix}.$$

Note that the solution $\mathbf{u}$ is a smooth radial function, but $\nabla \mathbf{u}$ is not defined at the origin $(0, 0)$; this will be observed later when we compare this case with the previous test case.

The numerical solution of (2.45) is illustrated in Figure 2.7. As the previous test cases, the main comments are: (1) the smoothness of the solutions (first row), (2) the similarity between numerical solutions of $\det \nabla \mathbf{u}_h$ and $\det \mathbf{p}_h$ (second row), and (3) the radial symmetry of $\|\mathbf{u}_h\|$ (bottom right), along with the direction of the arrows (bottom left), which point to the center of the domain. Converges properties of the relaxation algorithm on the disk structured mesh are presented in Table 2.6 where we see that the numerical solution converges in L^2 -norm with a rate of $O(h^{1.1})$ to $O(h^{1.5})$ and in H^1 semi norm with an optimal rate of $O(h)$. The number of iterations in this test case may reach the maximum allowed number of iterations corresponding to the stopping criterion of 1000 iterations. This is relatively more computationally expensive than the previous cases. Similar results are reported in Table 2.7 for the unstructured mesh.

Chapter 2. Numerical Approximation of the Prescribed Jacobian Equation/Inequality

Table 2.6 – Smooth radial symmetric solution with non-smooth gradient test case with data $f = 2(x^2 + y^2)$ and $\mathbf{u} = \mathbf{x}$ on Γ . Computations with various mesh sizes h ; L^2 and H^1 error norms with the corresponding rates; Computation of the difference between $\nabla \mathbf{u}_h$ and $\mathbf{p}_h$ in L^2 norm; Average value of $\bar{\lambda}$ and the standard deviation $\bar{\sigma}$; Iterations of the relaxation algorithm. The results are obtained on a structured mesh on the unit disk.

h	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2(\Omega)}$		$\ \mathbf{u} - \mathbf{u}_h\ _{H^1(\Omega)}$		$\ \nabla \mathbf{u}_h - \mathbf{p}_h\ _{L^2(\Omega)}$	$\bar{\lambda}(\bar{\sigma})$	iter
0.0831	1.57e-01		6.44e-01		3.84e-01	-2.72e-01(0.270)	999
0.0421	6.60e-02	1.25	3.18e-01	1.02	1.73e-01	-1.13e-01(0.129)	199
0.0209	2.62e-02	1.33	1.56e-01	1.02	7.87e-02	-3.72e-02(0.057)	309
0.0104	9.20e-03	1.51	7.07e-02	1.14	3.31e-02	-1.07e-02(0.023)	999

Table 2.7 – Smooth radial symmetric solution with non-smooth gradient test case with data $f = 2(x^2 + y^2)$ and $\mathbf{u} = \mathbf{x}$ on Γ . Computations with various mesh sizes h ; L^2 and H^1 error norms with the corresponding rates; Computation of the difference between $\nabla \mathbf{u}_h$ and $\mathbf{p}_h$ in L^2 norm; Average value of $\bar{\lambda}$ and the standard deviation $\bar{\sigma}$; Iterations of the relaxation algorithm. The results are obtained on an unstructured mesh on the unit disk .

h	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2(\Omega)}$		$\ \mathbf{u} - \mathbf{u}_h\ _{H^1(\Omega)}$		$\ \nabla \mathbf{u}_h - \mathbf{p}_h\ _{L^2(\Omega)}$	$\bar{\lambda}(\bar{\sigma})$	iter
0.1327	1.71e-01		6.77e-01		4.40e-01	-9.82e-02(0.244)	359
0.0665	7.39e-02	1.21	3.19e-01	1.09	1.93e-01	-5.24e-02(0.130)	259
0.0332	2.65e-02	1.48	1.39e-01	1.20	7.44e-02	-1.86e-02(0.059)	489
0.0166	9.51e-03	1.48	6.17e-02	1.17	2.87e-02	-5.31e-03(0.026)	999

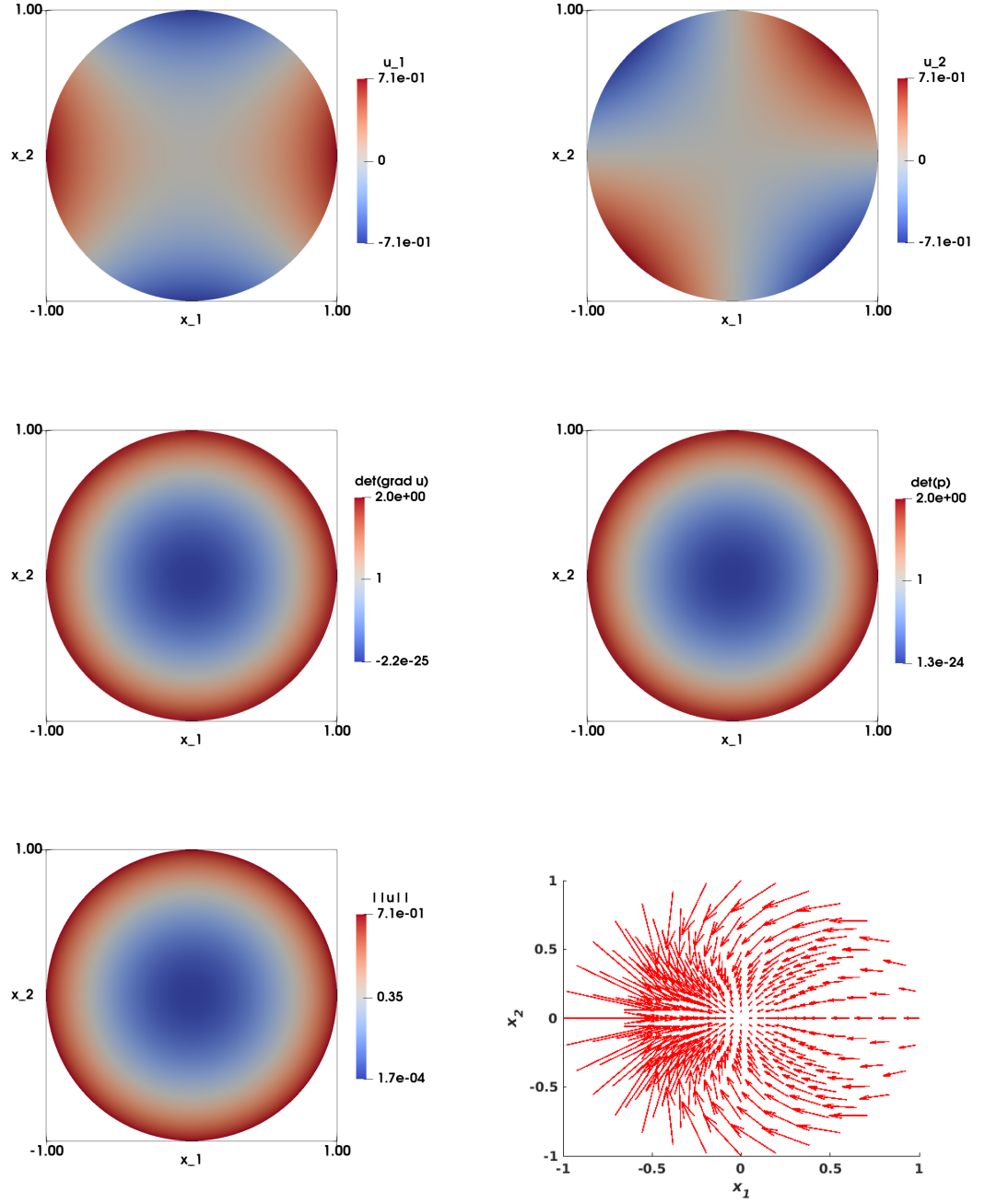


Figure 2.6 – Smooth solution with radial symmetric right-hand side test case with data $f = 2(x^2 + y^2)$ and $\mathbf{u} = \sqrt{2}(\frac{1}{2}(x_1^2 - x_2^2), x_1 x_2)^T$ on Γ . Graphs of the numerical approximations; Top Left: Numerical approximation of the solution of the component $u_{1,h}$. Top right: Numerical approximation of the solution of the component $u_{2,h}$. Middle left: Numerical approximation of $\det \nabla \mathbf{u}_h$. Middle right: Numerical approximation of $\det \mathbf{p}_h$. Bottom left: Numerical approximation of $\|\mathbf{u}_h\|$. Bottom right: Visualization of the vector field $\mathbf{u}_h$. The results are obtained on structured mesh on the unit disk with $h = 0.0209$.

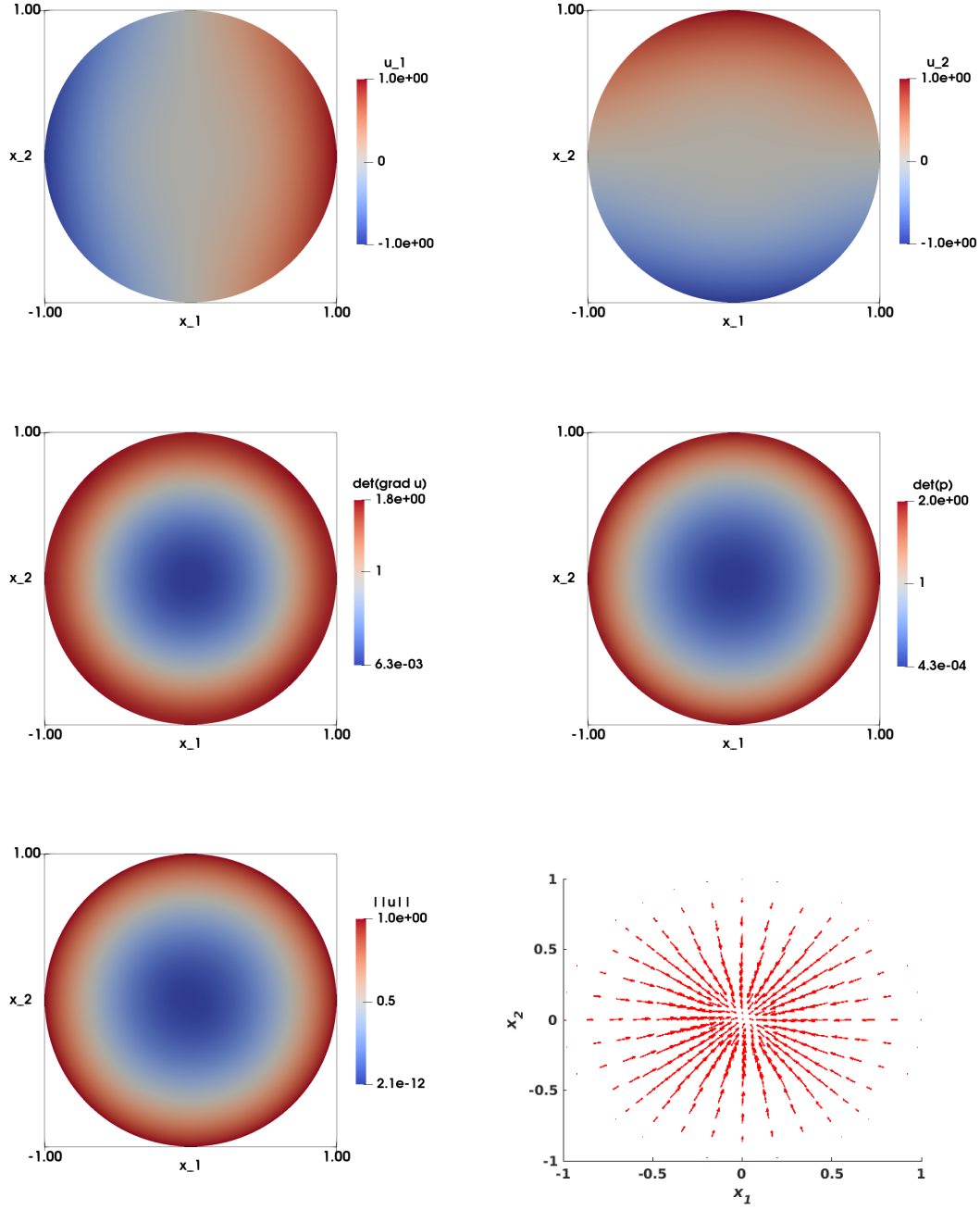


Figure 2.7 – Smooth radial symmetric solution with non-smooth gradient test case with data $f = 2(x^2 + y^2)$ and $\mathbf{u} = \mathbf{x}$ on Γ . Graphs of the numerical approximations; Top Left: Numerical approximation of the solution of the component $u_{1,h}$. Top right: Numerical approximation of the solution of the component $u_{2,h}$. Middle left: Numerical approximation of $\det \nabla \mathbf{u}_h$. Middle right: Numerical approximation of $\det \mathbf{p}_h$. Bottom left: Numerical approximation of $\|\mathbf{u}_h\|$. Bottom right: Visualization of the vector field $\mathbf{u}_h$. The results are obtained on a structured mesh on the unit disk with $h = 0.0209$.

2.4.5 Identity map on non-convex domains

In this subsection, we consider the identity map as in the first experiment

$$\begin{cases} \det \nabla \mathbf{u}(\mathbf{x}) = 1 & \text{in } \bar{\Omega}, \\ \mathbf{u}(\mathbf{x}) = \mathbf{x} & \text{on } \Gamma. \end{cases} \quad (2.46)$$

The exact solution of (2.46) is $\mathbf{u}(\mathbf{x}) = \mathbf{x}$. We will choose two non-convex domains. The first one is a pacman shaped domain obtained by eliminating a piece of angular size $2a$ where $a = \pi/4$, from a unit disk, so

$$\bar{\Omega} = \{(x_1, x_2) \in \mathbb{R}^2, x_1^2 + x_2^2 < 1\} \setminus \{(x_1, x_2) \in \mathbb{R}^2, x_1 > 0, |x_2| < x_1\}.$$

The finite element triangulation of $\bar{\Omega}$ is displayed in Figure 2.1 (bottom right). By using the pacman unstructured mesh with a size $h \in \{0.0747, 0.0503, 0.0252, 0.0126\}$, we get estimations in the $L^2(\Omega)$ and $H^1(\Omega)$ error norms of order 10^{-10} and 10^{-9} to 10^{-10} , respectively. Figure 2.8 illustrates the two components of the numerical solution. In addition, $\|\nabla \mathbf{u}_h - \mathbf{p}_h\|_{L^2(\Omega)}$ and $\bar{\lambda}$ are of the order of 10^{-10} . For $h = 0.0252$, after 29 iterations we obtain

$$\|\mathbf{u} - \mathbf{u}_h\|_{L^2(\Omega)} = 2.39 \cdot 10^{-10}, \quad \|\mathbf{u} - \mathbf{u}_h\|_{H^1(\Omega)} = 9.44 \cdot 10^{-10},$$

and

$$\|\nabla \mathbf{u}_h - \mathbf{p}_h\|_{L^2(\Omega)} = 6.51 \cdot 10^{-10}, \quad \bar{\lambda} = 1.55 \cdot 10^{-10}.$$

The second domain $\bar{\Omega}$ that we consider is a cracked unit disk domain. It is obtained by removing a piece of angular size $2a$ where $a = \pi/100$. The domain is then

$$\bar{\Omega} = \{(x_1, x_2) \in \mathbb{R}^2, x_1^2 + x_2^2 < 1\} \setminus \left\{ (x_1, x_2) \in \mathbb{R}^2, x_1 > 0, |x_2| < \tan\left(\frac{\pi}{100}\right) \right\},$$

The finite element triangulation of $\bar{\Omega}$ is displayed in Figure 2.1 (bottom right). We consider a mesh size $h \in \{0.1525, 0.0971, 0.0486, 0.0243\}$, for which we obtained error estimates in the $L^2(\Omega)$ and $H^1(\Omega)$ of order 10^{-10} and 10^{-9} , respectively. As well, $\|\nabla \mathbf{u}_h - \mathbf{p}_h\|_{L^2(\Omega)}$ and $\bar{\lambda}$ are of the order of 10^{-10} . Figure 2.9 shows the two components of the numerical solution. For instance, for $h = 0.0243$ after 29 iterations, we obtain

$$\|\mathbf{u} - \mathbf{u}_h\|_{L^2(\Omega)} = 3.74 \cdot 10^{-10}, \quad \|\mathbf{u} - \mathbf{u}_h\|_{H^1(\Omega)} = 1.23 \cdot 10^{-09},$$

and

$$\|\nabla \mathbf{u}_h - \mathbf{p}_h\|_{L^2(\Omega)} = 8.13 \cdot 10^{-10}, \quad \bar{\lambda} = 1.84 \cdot 10^{-10}.$$

To summarize, the behavior of our algorithm is similar of such non convex domains, both in terms of accuracy and efficiency.

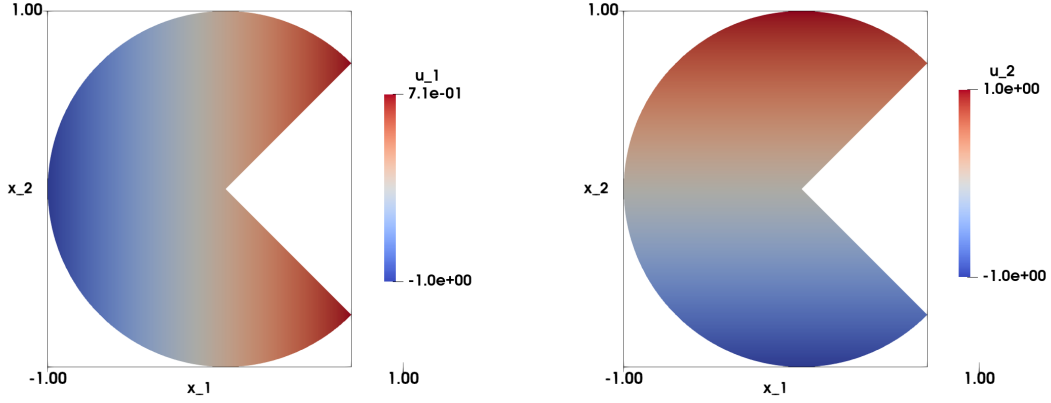


Figure 2.8 – Identity map on non-convex domains (pacman domain) with data $f = 1$ and $\mathbf{g} = \mathbf{x}$. Visualization of the numerical approximation of the solution $\mathbf{u}_h$; Left: the component $u_{1,h}$; Right: the component $u_{2,h}$. Results are obtained with an unstructured mesh with $h = 0.0252$.

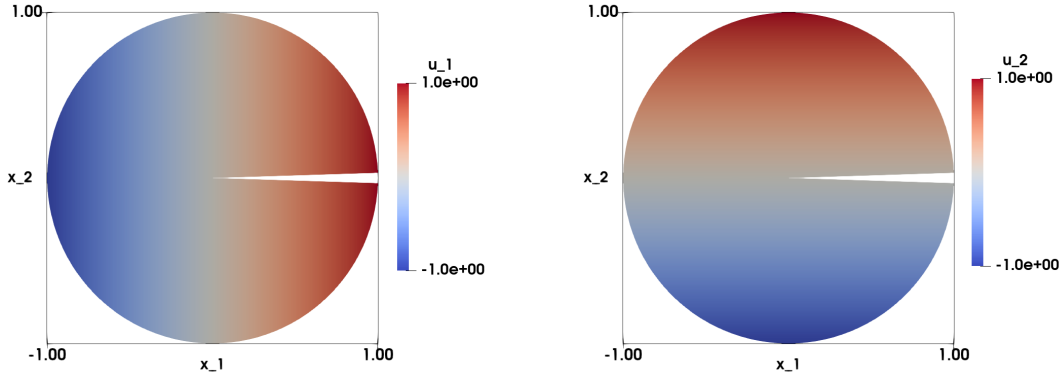


Figure 2.9 – Identity map on non-convex domains (cracked domain) with data $f = 1$ and $\mathbf{g} = \mathbf{x}$. Visualization of the numerical approximation of the solution $\mathbf{u}_h$; Left: the component $u_{1,h}$; Right: the component $u_{2,h}$. The results are obtained with an unstructured mesh with $h = 0.0486$.

2.4.6 Smooth solution with radial symmetric right-hand side on non-convex domains

In this experiment we use the same non-convex domains as in the previous section, and we focus on the following problem

$$\begin{cases} \det \nabla \mathbf{u} = 2\|\mathbf{x}\|_2^2 \text{ in } \bar{\Omega}, \\ \mathbf{u}(\mathbf{x}) = \mathbf{g}(\mathbf{x}) \text{ on } \Gamma. \end{cases} \quad (2.47)$$

where

$$\mathbf{g}(\mathbf{x}) = \sqrt{2} \begin{pmatrix} \frac{1}{2}(x_1^2 - x_2^2) \\ x_1 x_2 \end{pmatrix}.$$

In this case, the exact solution is

$$\mathbf{u}(\mathbf{x}) = \sqrt{2} \begin{pmatrix} \frac{1}{2}(x_1^2 - x_2^2) \\ x_1 x_2 \end{pmatrix}$$

For both domains (pacman with unstructured mesh, and cracked disk with an unstructured mesh), results are reported in Table 2.8 and Table 2.9, respectively. We see that the numerical solution converges in L^2 -norm with a rate of $O(h^{1.9})$ to $O(h^{1.7})$ and $O(h^{1.8})$, respectively. Convergence in H^1 semi norm is of order of $O(h)$ on both domains. The number of iterations of the relaxation algorithm is linearly increasing for decreasing h . Comparing these two tables, we can say that the performance of algorithm is the same for the two non-convex domains. Recall that in Table 2.3, we solved the same problem on a convex domain. We can conclude that the algorithm has the same level of performance for both algorithms, on either convex and non-convex domains.

The numerical solution of (2.47) on both domains is illustrated in the top row of Figure 2.10 and 2.11, respectively. The numerical approximation of $\nabla \mathbf{u}_h$ and $\det \mathbf{p}_h$ in the second row of both figures are identical on both domains, and the vector field $\mathbf{u}_h$ (bottom right) of the figures point towards the origin

Table 2.8 – Smooth solution with radial symmetric right-hand side test case with data $f(x_1, x_2) = 2(x_1^2 + x_2^2)$ and $\mathbf{u}(\mathbf{x}) = \sqrt{2}(\frac{1}{2}(x_1^2 - x_2^2), x_1 x_2)^T$ on Γ . Computations with various mesh sizes h ; L^2 and H^1 error norms with the corresponding rates; Computation of the difference between $\nabla \mathbf{u}_h$ and $\mathbf{p}_h$ in L^2 norm; Average value $\bar{\lambda}$ and the corresponding standard deviation $\bar{\sigma}$; Iterations of the relaxation algorithm. The results are obtained on unstructured mesh for the pacman domain.

h	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2(\Omega)}$		$\ \mathbf{u} - \mathbf{u}_h\ _{H^1(\Omega)}$		$\ \nabla \mathbf{u}_h - \mathbf{p}_h\ _{L^2(\Omega)}$	$\bar{\lambda}(\bar{\sigma})$	iter
0.0747	3.24e-03		1.51e-01		7.68e-02	3.43e-03(0.045)	19
0.0503	8.65e-04	1.90	7.59e-02	1.00	3.90e-02	1.47e-03(0.025)	29
0.0252	2.26e-04	1.94	3.80e-02	1.00	1.95e-02	7.16e-04(0.013)	59
0.0126	6.67e-05	1.76	1.90e-02	1.00	9.73e-03	2.99e-04(0.007)	119

Chapter 2. Numerical Approximation of the Prescribed Jacobian Equation/Inequality

Table 2.9 – Smooth solution with radial symmetric right-hand side test case with data $f(x_1, x_2) = 2(x_1^2 + x_2^2)$ and $\mathbf{u} = \sqrt{2}(\frac{1}{2}(x_1^2 - x_2^2), x_1 x_2)^T$ on Γ . Computations with various mesh sizes h ; L^2 and H^1 error norms with the corresponding rates; Computation of the difference between $\nabla \mathbf{u}_h$ and $\mathbf{p}_h$ in L^2 norm; Average value $\bar{\lambda}$ and the corresponding standard deviation $\bar{\sigma}$; Iterations of the relaxation algorithm. The results are obtained on unstructured mesh for the pacman domain.

h	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2(\Omega)}$		$\ \mathbf{u} - \mathbf{u}_h\ _{H^1(\Omega)}$		$\ \nabla \mathbf{u}_h - \mathbf{p}_h\ _{L^2(\Omega)}$	$\bar{\lambda}(\bar{\sigma})$	iter
0.1525	1.42e-02		3.44e-01		1.81e-01	1.89e-02(0.101)	19
0.0971	3.98e-03	1.83	1.72e-01	1.00	9.15e-02	1.11e-02(0.056)	19
0.0486	1.12e-03	1.83	8.63e-02	1.00	4.53e-02	5.62e-03(0.030)	39
0.0243	3.33e-04	1.75	4.32e-02	1.00	2.23e-02	2.89e-03(0.016)	69

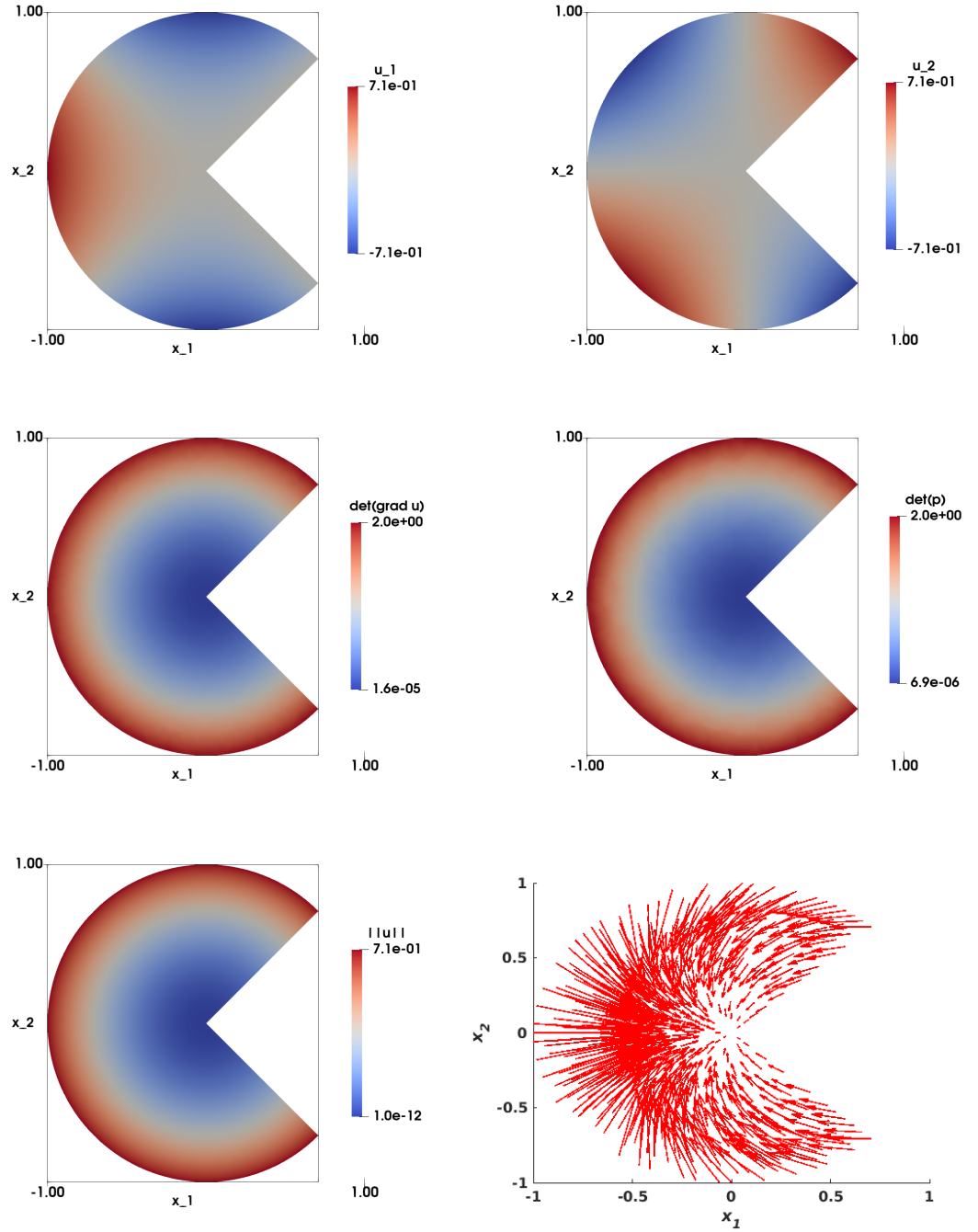


Figure 2.10 – Smooth solution with radial symmetric right-hand side on non-convex domains test case with data $f(x_1, x_2) = 2(x^2 + y^2)$ and $\mathbf{u}(x_1, x_2) = \sqrt{2} \left(\frac{1}{2}(x_1^2 - x_2^2), x_1 x_2 \right)^T$ on Γ . Top Left: Numerical approximation of the solution of the component $u_{1,h}$. Top right: Numerical approximation of the solution of the component $u_{2,h}$. Middle left: Numerical approximation of $\det \nabla \mathbf{u}_h$. Middle right: Numerical approximation of $\det \mathbf{p}_h$. Bottom left: Numerical approximation of $\|\mathbf{u}_h\|$. Bottom right: Visualization of the vector field $\mathbf{u}_h$. The results are obtained on pacman unstructured mesh with $h = 0.0252$.

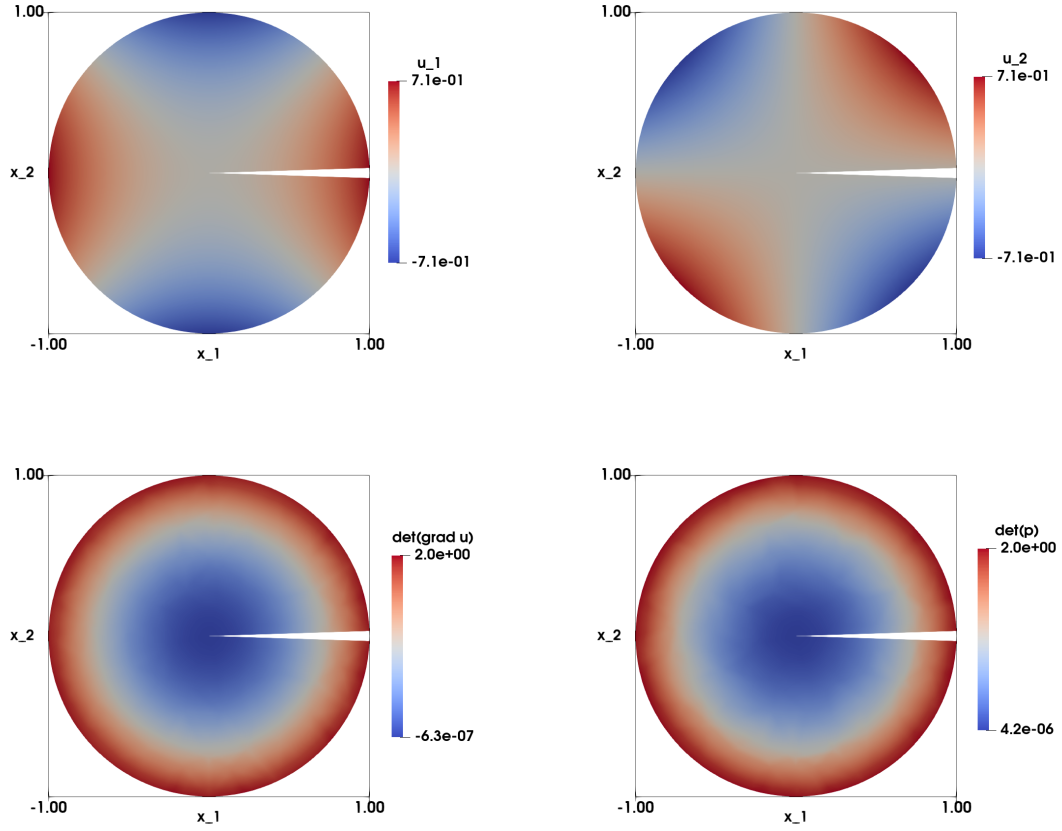


Figure 2.11 – Smooth solution with radial symmetric right-hand side on non-convex domains test case with data $f(x_1, x_2) = 2(x^2 + y^2)$ and $\mathbf{u}(x_1, x_2) = \sqrt{2}(\frac{1}{2}(x_1^2 - x_2^2), x_1 x_2)^T$ on Γ . Graphs of the numerical approximations; Top Left: Numerical approximation of the solution of the component $u_{1,h}$. Top right: Numerical approximation of the solution of the component $u_{2,h}$. Middle left: Numerical approximation of $\det \nabla \mathbf{u}_h$. Middle right: Numerical approximation of $\det \mathbf{p}_h$. The results are obtained on cracked disk unstructured mesh with $h = 0.0486$.

2.4.7 Nonsmooth right-hand side with a jump

Let Ω be the unit disk

$$\Omega = \{(x_1, x_2) \in \mathbb{R}^2, x_1^2 + x_2^2 < 1\}.$$

In this experiment, we validate the algorithm with a non-smooth right hand side by introducing a jump on f given by

$$f(\mathbf{x}) = \begin{cases} 0.1 & \text{if } x_1 \leq 0, \\ 1.9 & \text{if } x_1 > 0. \end{cases} \quad (2.48)$$

Note that f still satisfies the necessary condition $\int_{\Omega} f = \text{measure}(\Omega)$. On the boundary we enforce the identity function $\mathbf{g} = \mathbf{x}$, and the problem reads as

$$\begin{cases} \det \nabla \mathbf{u} = f & \text{in } \Omega, \\ \mathbf{u} = \mathbf{g} & \text{on } \Gamma. \end{cases} \quad (2.49)$$

It is important to mention that an exact solution of problem (2.49) is unknown. However, the low regularity on the right-hand side indicates that the solution, if it exists, will also have low regularity.

Table 2.10 shows results for $\varepsilon = 0$ and $\varepsilon = h^2$ for the disk structured meshes. We observe that the error between the numerical solution $\nabla \mathbf{u}_h$ and the auxiliary variable $\mathbf{p}_h$ in L^2 is of the order of $O(h)$ for both values of ε , although, more accurate for $\varepsilon = 0$. Same observations can be made for $\tilde{\lambda}$ and $\tilde{\sigma}$. In addition, the iterations of the relaxation algorithm are reaching the limit of stopping criterion for $\varepsilon = 0$. and for $\varepsilon = h^2$, the iterations are well controlled. This shows that the ε -regularization helps the convergence of the algorithm.

The numerical solution of (2.49) on the disk domain with $\varepsilon = 0$ is illustrated in Figure 2.12 (top row). A close inspection of this figure shows that the $u_{1,h}$ component (top left) is discontinuous in $x_2 = 0$, as expected, and the $u_{2,h}$ component (top right) is smooth. The numerical approximation of $\det \nabla \mathbf{u}_h$ and $\det \mathbf{p}_h$ are displayed in the second row, and are identical. In the bottom row, we illustrate $\|\mathbf{u}_h\|_2$ on the left side, and we see on the right side that the vector fields

The numerical solution of (2.49) with ε set to h^2 is displayed in the top row of Figure 2.13, where both components can be regarded as smooth. The second row on the other side, shows the numerical solution of $\det \nabla \mathbf{u}_h$ and $\det \mathbf{p}_h$. These solutions look different, for instance $\det \nabla \mathbf{u}_h$ is smoother in $x_2 = 0$. The left panel in the bottom row of Figure 2.13 represents $\|\mathbf{u}_h\|_2$, where we see that vector fields is directed towards the center.

A comparison with respect to the different figures associated with the choice of ε is made in Figure 2.14, where we plot the solutions as a function of x_1 for $x_2 = 0$. In the top row, we plot the solution of the component $u_{1,h}$ (left) and $\|\mathbf{u}_h\|$ (right). We can see a slight difference between the two curves along with the discontinuity point mentioned above. In the bottom row, we

Chapter 2. Numerical Approximation of the Prescribed Jacobian Equation/Inequality

Table 2.10 – Nonsmooth right-hand side with a jump test case with the data f given by (2.48). Results for $\varepsilon = 0$ and $\varepsilon = h^2$; Computations with various mesh sizes h ; Computation of the difference of $\nabla \mathbf{u}_h$ and $\mathbf{p}_h$ in L^2 norm; Average value of $\bar{\lambda}$ and the corresponding standard deviation $\bar{\sigma}$; Number of iterations of the relaxation algorithm. The results are obtained with structured meshes.

h	$\ \nabla \mathbf{u}_h - \mathbf{p}_h\ _{L^2(\Omega)}$	$\bar{\lambda}(\bar{\sigma})$	iter
$\varepsilon = 0$			
0.0831	3.61e-02	7.27e-04(0.007)	999
0.0421	1.77e-02	1.81e-04(0.005)	999
0.0209	1.15e-02	-1.52e-05(0.003)	999
0.0104	8.65e-03	-1.40e-05(0.002)	999
$\varepsilon = h^2$			
0.0831	4.68e-01	-1.31e-01(0.166)	189
0.0421	2.69e-01	-3.36e-02(0.090)	129
0.0209	1.73e-01	-8.84e-03(0.059)	269
0.0104	1.14e-01	-2.42e-03(0.041)	779

represent $\det \nabla \mathbf{u}_h$ (on the left), where we see the effect of increasing ε on the smoothness of the corresponding solution. The visualization of $\det \mathbf{p}_h$ (right) shows that the choice of ε is irrelevant, as the two curves coincide pointwise.

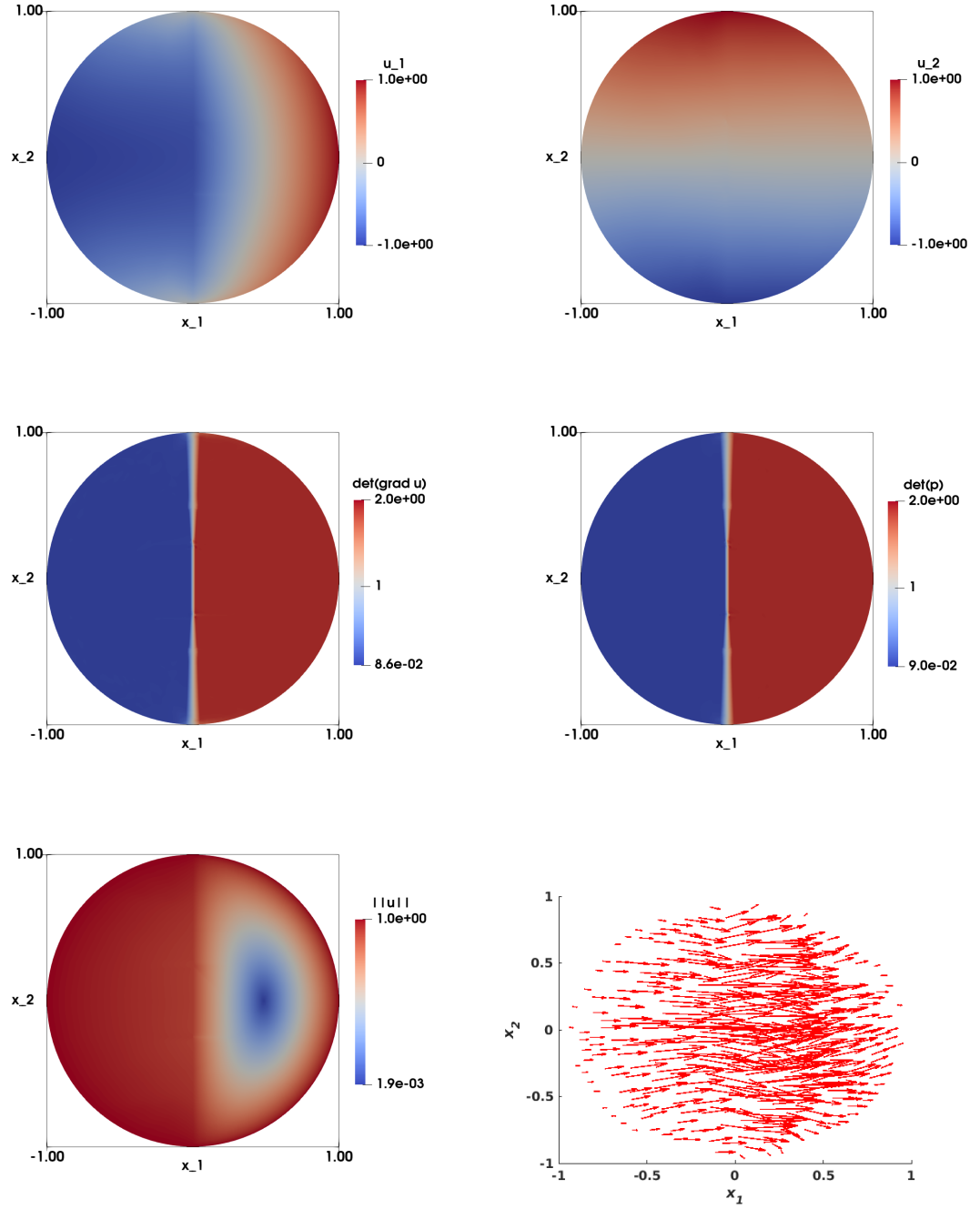


Figure 2.12 – Nonsmooth right-hand side with a jump test case with data f as in (2.48). Top Left: Numerical approximation of the solution of the component $u_{1,h}$. Top right: Numerical approximation of the solution of the component $u_{2,h}$. Middle left: Numerical approximation of $\det \nabla \mathbf{u}_h$. Middle right: Numerical approximation of $\det \mathbf{p}_h$. Bottom left: Numerical approximation of $\|\mathbf{u}_h\|$. Bottom right: Visualization of the vector field $\mathbf{u}_h$. The results are obtained on a disk structured mesh with $h = 0.0209$ and $\varepsilon = 0$.

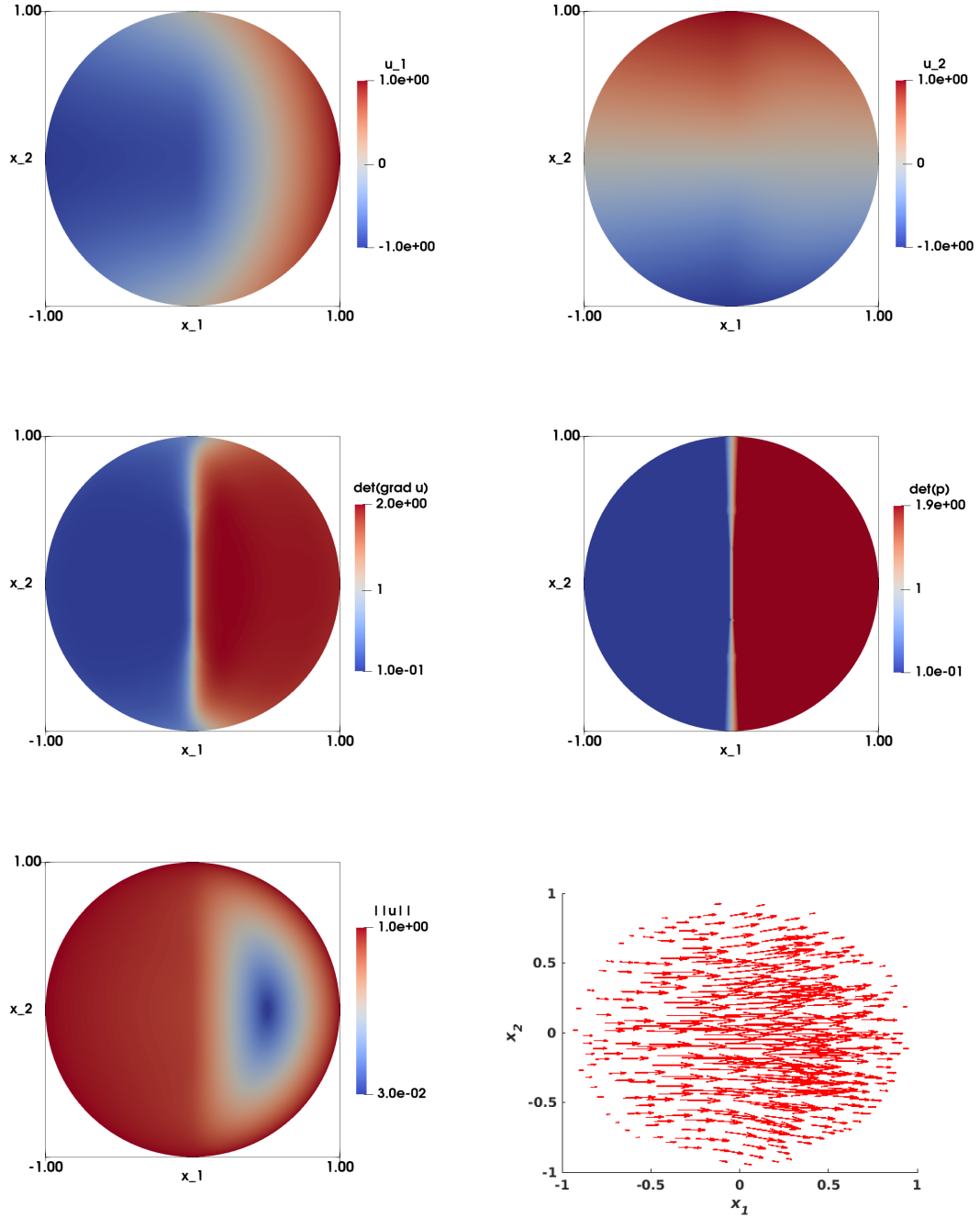


Figure 2.13 – Nonsmooth right-hand side with a jump test case with data f as in (2.48). Top Left: Numerical approximation of the solution of the component $u_{1,h}$. Top right: Numerical approximation of the solution of the component $u_{2,h}$. Middle left: Numerical approximation of $\det \nabla \mathbf{u}_h$. Middle right: Numerical approximation of $\det \mathbf{p}_h$. Bottom left: Numerical approximation of $\|\mathbf{u}_h\|$. Bottom right: Visualization of the vector field $\mathbf{u}_h$. The results are obtained on a disk structured mesh with $h = 0.0209$ and $\varepsilon = h^2$.

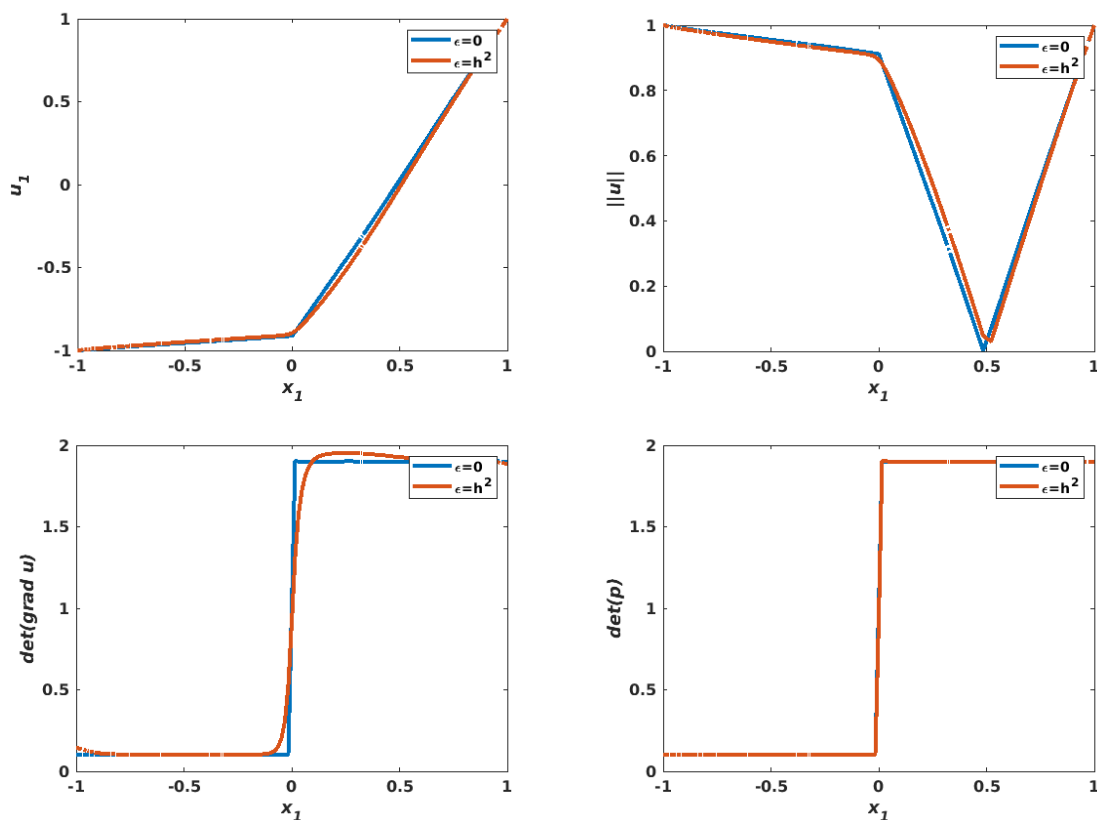


Figure 2.14 – Nonsmooth right-hand side with a jump test case with data f as in (2.48). Comparing plot between $\epsilon = 0$ and $\epsilon = h^2$. Computation on a structured mesh with $h = 0.00209$. All data extracted along the line $x_2 = 0$ line; Top left : $u_{1,h}$ component versus x_1 . Top right: $\|\mathbf{u}_h\|$ vs x_1 . Bottom left: $\det \nabla \mathbf{u}_h$ vs x_1 . Bottom right: $\det \mathbf{p}_h$ vs x_1 .

2.4.8 Nonsmooth right-hand side with a Dirac delta function

Let Ω be the unit disk defined by

$$\Omega = \{(x_1, x_2) \in \mathbb{R}^2, x_1^2 + x_2^2 < 1\}.$$

In this experiment, we consider for the right hand side a Dirac delta function centered at the origin, i.e.,

$$f = \pi \delta_{(0,0)}$$

and $\mathbf{g}(\mathbf{x}) = \mathbf{x}$ on Γ ; therefore the problem reads :

$$\begin{cases} \det \nabla \mathbf{u} = f & \text{in } \Omega, \\ \mathbf{u} = \mathbf{g} & \text{on } \Gamma. \end{cases} \quad (2.50)$$

The exact solution of problem (2.50) is

$$\mathbf{u}(\mathbf{x}) = \frac{\mathbf{x}}{||\mathbf{x}||_2}. \quad (2.51)$$

In order to apply our methodology, we approximate f by f_η defined by

$$f_\eta(\mathbf{x}) = \frac{\eta^2}{(\eta^2 + ||\mathbf{x}||_2^2)^2},$$

where η is a small positive value, see [Caboussat et al., 2013; Liu et al., 2019]. When $\eta \rightarrow 0$, the approximate solution f_η converges to f . Note also that f_η satisfies the necessary condition

$$\int_{\Omega} f_\eta d\mathbf{x} = \frac{\pi}{\eta + 1} = \text{measure}(\Omega), \quad \text{when } \eta \rightarrow 0.$$

The modified problem reads as

$$\begin{cases} \det \nabla \mathbf{u} = f_\eta & \text{in } \Omega, \\ \mathbf{u}_\eta(\mathbf{x}) = \mathbf{x} & \text{on } \Gamma, \end{cases} \quad (2.52)$$

and the exact solution is

$$\mathbf{u}_\eta(\mathbf{x}) = \mathbf{x} \sqrt{\frac{1 + \eta^2}{\eta^2 + ||\mathbf{x}||_2^2}}. \quad (2.53)$$

We can also show that

$$\lim_{\eta \rightarrow 0} \mathbf{u}_\eta(\mathbf{x}) = \mathbf{u}(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega.$$

We will examine this test case for various values of η, ε and h . Table 2.11 shows computations with different η and h for a unstructured mesh on a unit disk. We can see, for $\eta \in \{1/8, 1/16, 1/32, 1/64\}$, the error of the numerical solution in L^2 norm decreases with an order equal or larger than $O(h)$. For $\eta = 1/4$, the convergence rate drops to 0.83 when $h = 0.0104$. The same comments hold for the difference between $\nabla \mathbf{u}_h$ and $\mathbf{p}_h$ in L^2 norm, and the estimates of λ and $\bar{\sigma}$. For most values of η , the maximum number of iterations is reached when the mesh size is equal to the coarsest and finest values, i.e., $h = 0.0571$ and $h = 0.0104$. For other choices of h , we obtained a reasonable number of iterations.

It is worth mentioning that large values of η such as $1/4$, do not satisfy the necessary condition $\int_{\Omega} f_{\eta} \leq \pi$. This eventually causes numerical problems when it comes to the convergence of the algorithm. Small values on the other hand, such as $\eta = 1/64$, are associated with large gradients which consequently implies that we need a finer mesh in order to have a better convergence.

Chapter 2. Numerical Approximation of the Prescribed Jacobian Equation/Inequality

Table 2.11 – Nonsmooth right-hand side with a Dirac delta function test case with data $f = \frac{\eta^2}{(\eta^2 + \|\mathbf{x}\|_2^2)^2}$, $\mathbf{u}_\eta = \mathbf{x} \sqrt{\frac{1+\eta^2}{\eta^2 + \|\mathbf{x}\|_2^2}}$; Results for different η . Computations with various mesh sizes h ; L^2 error norm with the corresponding rates; Computation of the difference between $\nabla \mathbf{u}_h$ and $\mathbf{p}_h$ in L^2 norm; Average value of $\bar{\lambda}$ and the standard deviation $\bar{\sigma}$; Iterations of the relaxation algorithm.

h	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2(\Omega)}$		$\ \nabla \mathbf{u}_h - \mathbf{p}_h\ _{L^2(\Omega)}$	$\bar{\lambda}(\bar{\sigma})$	iter
$\eta = 1/4$					
0.0831	1.69e-01		5.69e-01	-4.94e-02(0.178)	59
0.0421	5.80e-02	1.54	2.01e-01	-2.87e-02(0.057)	99
0.0209	2.33e-02	1.31	8.55e-02	-2.16e-02(0.019)	239
0.0104	1.31e-02	0.83	6.01e-02	-1.87e-02(0.006)	999
$\eta = 1/8$					
0.0831	2.13e-01		9.20e-01	-1.09e-01(0.181)	119
0.0421	7.73e-02	1.46	4.19e-01	-4.60e-02(0.069)	39
0.0209	2.66e-02	1.54	1.64e-01	-1.79e-02(0.026)	299
0.0104	9.71e-03	1.45	5.66e-02	-7.54e-03(0.009)	999
$\eta = 1/16$					
0.0831	2.68e-01		1.43e+00	-2.08e-01(0.177)	359
0.0421	1.14e-01	1.23	8.40e-01	-9.08e-02(0.083)	249
0.0209	4.37e-02	1.39	4.11e-01	-3.37e-02(0.035)	329
0.0104	1.47e-02	1.57	1.61e-01	-1.09e-02(0.013)	999
$\eta = 1/32$					
0.0831	3.21e-01		2.03e+00	-3.04e-01(0.167)	999
0.0421	1.61e-01	0.99	1.38e+00	-1.46e-01(0.092)	79
0.0209	7.33e-02	1.14	8.37e-01	-6.16e-02(0.044)	589
0.0104	2.85e-02	1.36	4.11e-01	-2.26e-02(0.018)	379
$\eta = 1/64$					
0.0831	3.68e-01		2.39e+00	-3.88e-01(0.153)	999
0.0421	2.09e-01	0.81	1.99e+00	-2.05e-01(0.097)	229
0.0209	1.13e-01	0.89	1.38e+00	-1.00e-01(0.051)	999
0.0104	5.29e-02	1.09	8.36e-01	-4.39e-02(0.023)	999

Results reported in Table 2.12 are for fixed $h = 0.0209$ and different η . We note that for η decreasing, the estimated errors $\|\mathbf{u} - \mathbf{u}_h\|_{L^2(\Omega)}$, $\|\nabla \mathbf{u}_h - \mathbf{p}_h\|_{L^2(\Omega)}$, the sample mean $\bar{\lambda}$, and the sample standard deviation $\bar{\sigma}$ along with the number of iterations are consistently increasing.

Table 2.13 represents the L^∞ norm of f_η , $\det \mathbf{p}_h$ and $\det \nabla \mathbf{u}_h$. We observe that the values of $\det \mathbf{p}_h$ are close to f_η for $\eta = \{1/4, 1/8, 1/16, 1/32\}$. Last row of Table 2.13 exhibits large difference from $\|f_\eta\|_{L^\infty(\Omega)}$, and as mentioned above, refining the mesh will fix this problem. Large differences between $\|\det \nabla \mathbf{u}_h\|_{L^\infty(\Omega)}$ and the other columns are mainly due to the choice of $\varepsilon = h^2$.

Table 2.12 – Nonsmooth right-hand side with a Dirac delta function test case with data $f = \frac{\eta^2}{(\eta^2 + \|\mathbf{x}\|_2^2)^2}$, $u_\eta = \mathbf{x} \sqrt{\frac{1+\eta^2}{\eta^2 + \|\mathbf{x}\|_2^2}}$; Results for different η . Computations with various mesh sizes h ; L^2 error norms with the corresponding rates; Computation of the difference between $\nabla \mathbf{u}_h$ and $\mathbf{p}_h$ in L^2 norm; Average value of $\bar{\lambda}$ and the standard deviation $\bar{\sigma}$; Iterations of the relaxation algorithm.

η	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2(\Omega)}$	$\ \nabla \mathbf{u}_h - \mathbf{p}_h\ _{L^2(\Omega)}$	$\bar{\lambda}(\bar{\sigma})$	iter
$h = 0.0209$				
1/4	2.33e-02	8.55e-02	-2.16e-02(0.019)	239
1/8	2.66e-02	1.64e-01	-1.79e-02(0.026)	299
1/16	4.37e-02	4.11e-01	-3.37e-02(0.035)	329
1/32	7.33e-02	8.37e-01	-6.16e-02(0.044)	589
1/64	1.13e-01	1.38e+00	-1.00e-01(0.051)	999

In Table 2.14, the parameter η is fixed and we vary h and ε . We observe that for $\varepsilon = \{0, h^2\}$, the numerical solution converges in L^2 -norm at least with a rate $O(h)$. Moreover, we note that the error between the numerical solution $\nabla \mathbf{u}_h$ and the auxiliary variable $\mathbf{p}_h$ in L^2 norm decreases with an order of $O(h)$. Same occurs to $\bar{\lambda}$ and $\bar{\sigma}$. The number of iterations for the relaxation algorithm reaches the maximum that corresponds to the stopping criterion for $\varepsilon = 0$. Perturbations in the number of iteration for $\varepsilon = h^2$ are due to large gradients.

Table 2.13 – Nonsmooth right-hand side with a Dirac delta function test case with data $f = \frac{\eta^2}{(\eta^2 + \|\mathbf{x}\|_2^2)^2}$, $u_\eta = \mathbf{x} \sqrt{\frac{1+\eta^2}{\eta^2 + \|\mathbf{x}\|_2^2}}$; Results for different η . L^∞ -norm for $\det \mathbf{p}_h$, $\det \nabla \mathbf{u}_h$, and f_η ; Computations with mesh size $h = 0.0209$.

η	$\ \det \mathbf{p}_h\ _{L^\infty(\Omega)}$	$\ \det \nabla \mathbf{u}_h\ _{L^\infty(\Omega)}$	$\ f_\eta\ _{L^\infty(\Omega)}$
$h = 0.0209$			
1/4	15.99	14.81	16
1/8	63.88	46.97	64
1/16	254.23	119.25	256
1/32	996.13	218.06	1024
1/64	3676.18	280.17	4096

Chapter 2. Numerical Approximation of the Prescribed Jacobian Equation/Inequality

Table 2.14 – Nonsmooth right-hand side with a Dirac delta function test case with data $f = \frac{\eta^2}{(\eta^2 + \|\mathbf{x}\|_2^2)^2}$, $u_\eta = \mathbf{x} \sqrt{\frac{1+\eta^2}{\eta^2 + \|\mathbf{x}\|_2^2}}$; Results for different η . Computations with various mesh size h ; L^2 and H^1 error norms with the corresponding rate; Computation of the difference of $\nabla \mathbf{u}_h$ and $\mathbf{p}_h$ in L^2 norm; Average value of $\bar{\lambda}$ and the standard deviation $\bar{\sigma}$; Iterations of the relaxation algorithm.

h	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2(\Omega)}$		$\ \nabla \mathbf{u}_h - \mathbf{p}_h\ _{L^2(\Omega)}$	$\bar{\lambda}(\bar{\sigma})$	iter
$\eta = 1/32, \varepsilon = 0$					
0.0831	7.90e-01		4.98e-01	9.51e-03(0.042)	999
0.0421	2.14e-01	1.89	3.08e-01	-1.24e-02(0.026)	999
0.0209	9.60e-02	1.16	1.65e-01	-7.98e-04(0.013)	999
0.0104	4.87e-02	0.98	9.73e-02	-2.04e-04(0.007)	999
$\eta = 1/32, \varepsilon = h^2$					
0.0831	3.21e-01		2.03e+00	-3.04e-01(0.167)	999
0.0421	1.61e-01	0.99	1.38e+00	-1.46e-01(0.092)	79
0.0209	7.33e-02	1.14	8.37e-01	-6.16e-02(0.044)	589
0.0104	2.85e-02	1.36	4.11e-01	-2.26e-02(0.018)	379

The numerical solutions of (2.47) for all η examined here are illustrated in Figures 2.15-2.19. We see from the figures that the solution of the two components $u_{1,h}$ and $u_{2,h}$ are smooth for $\eta = \{1/4, 1/8\}$ and the singularity in $(0,0)$ is visible for $\eta = \{1/16, 1/32, 1/64\}$. The numerical solution of $\det \nabla \mathbf{u}_h$ and $\det \mathbf{p}_h$ illustrated in the second row of all figures, show that as η gets smaller, both determinants become larger. The largest values of these determinants are shown on Table 2.13.

A comparison of the different figures associated with the choice of η is made in Figure 2.20, where we plot the solutions as a function of x_1 for $x_2 = 0$. In the top row, we plot both, the solution of the component $u_{1,h}$ (on the left) and $\|\mathbf{u}_h\|$ (on the right). As η gets smaller, we see that the tangent of $u_{1,h}$ at the singularity point $(0,0)$ becomes vertical, which implies that the gradient of the curve is undefined. This singularity point significantly appears on the right panel in the plot of $\|\mathbf{u}_h\|$. The bottom row of Figure 2.20 shows $\det \nabla \mathbf{u}_h$ (left) and $\det \mathbf{p}_h$ (right). In our case, the determinant of the corresponding quantities is related to the Dirac delta function which is better approximated by small η , as shown in both figures.

In Figure 2.21 we report the numerical solution of (2.47) for the following case scenario: $\eta = 1/32$, $\varepsilon = 0$, $h = 0.0209$. In the first row, we remark that the solutions of the two components $u_{1,h}$ and $u_{2,h}$ are non-smooth and the singularity at the origin is visible, this point also appears in the plots of $\det \nabla \mathbf{u}_h$ and $\det \mathbf{p}_h$ in the second row.

For the same case scenario, a comparison between the solution associated with $\varepsilon = \{0, h^2\}$ is made in Figure 2.22, where we plot the data as a function of x_1 for $x_2 = 0$. In the top row we plot the $u_{1,h}$ component (left) and $\|\mathbf{u}_h\|$ (right), and we see that the curves of $\varepsilon = 0$ have oscillations and look less smooth than those of $\varepsilon = h^2$. The same observation holds for $\det \nabla \mathbf{u}_h$ illustrated in the bottom right. In the bottom left of Figure 2.22, the choice of ε seems to be of second importance since both curves when $\varepsilon = 0$ and $\varepsilon = h^2$ are identical.

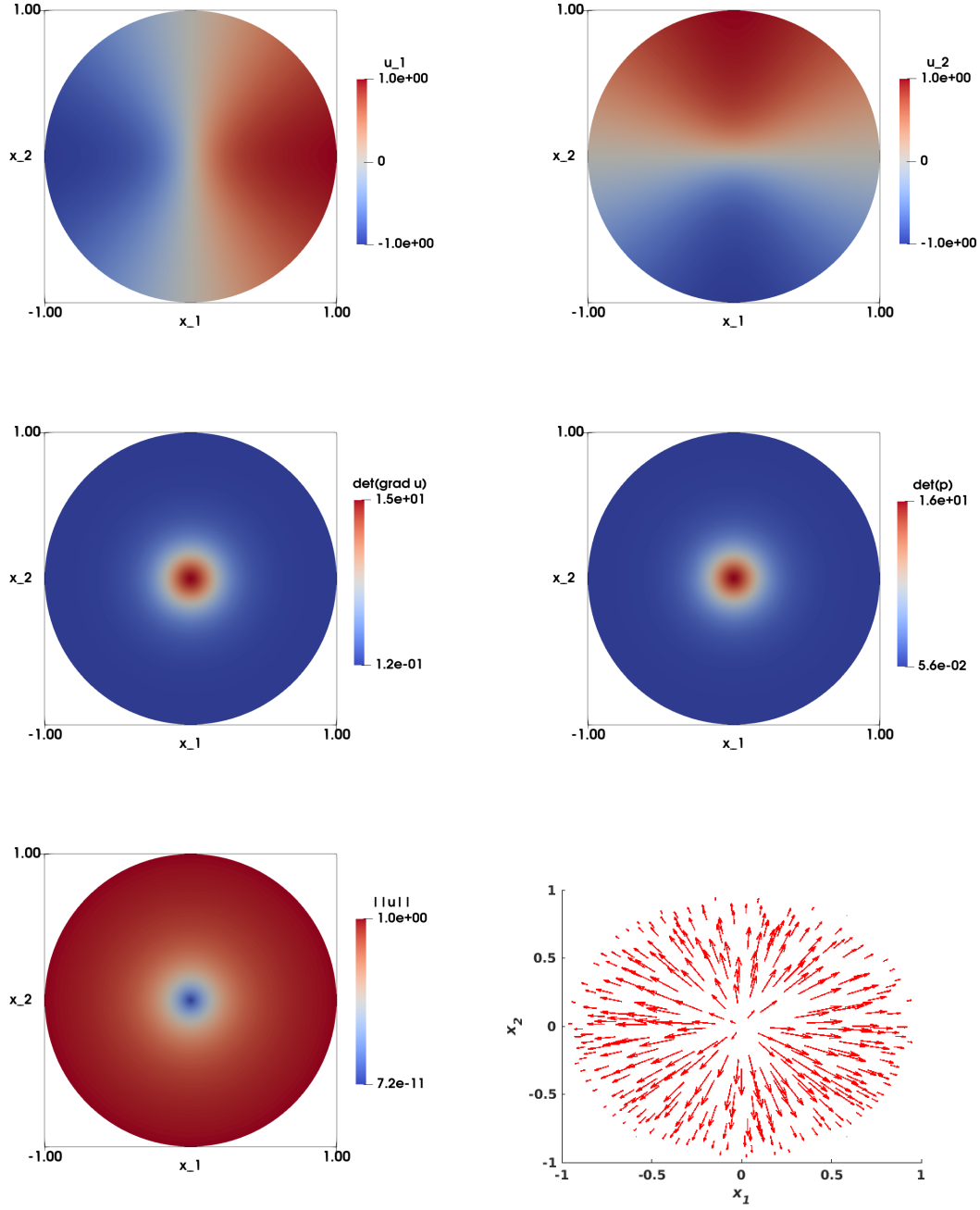


Figure 2.15 – Non-smooth right-hand side with a Dirac delta function test case with data $f = \frac{\eta^2}{(\eta^2 + \|x\|_2^2)^2}$, $\mathbf{u}\eta = \mathbf{x}\sqrt{\frac{1+\eta^2}{\eta^2 + \|x\|_2^2}}$ and $\eta = 1/4$. Top Left: Numerical approximation of the solution of the component $u_{1,h}$. Top right: Numerical approximation of the solution of the component $u_{2,h}$. Middle left: Numerical approximation of $\det \nabla \mathbf{u}_h$. Middle right: Numerical approximation of $\det \mathbf{p}_h$. Bottom left: Numerical approximation of $\|\mathbf{u}_h\|$. Bottom right: Visualization of the vector field $\mathbf{u}_h$. The results are obtained on structured mesh of the unit disk with $h = 0.0209$ and $\varepsilon = h^2$.

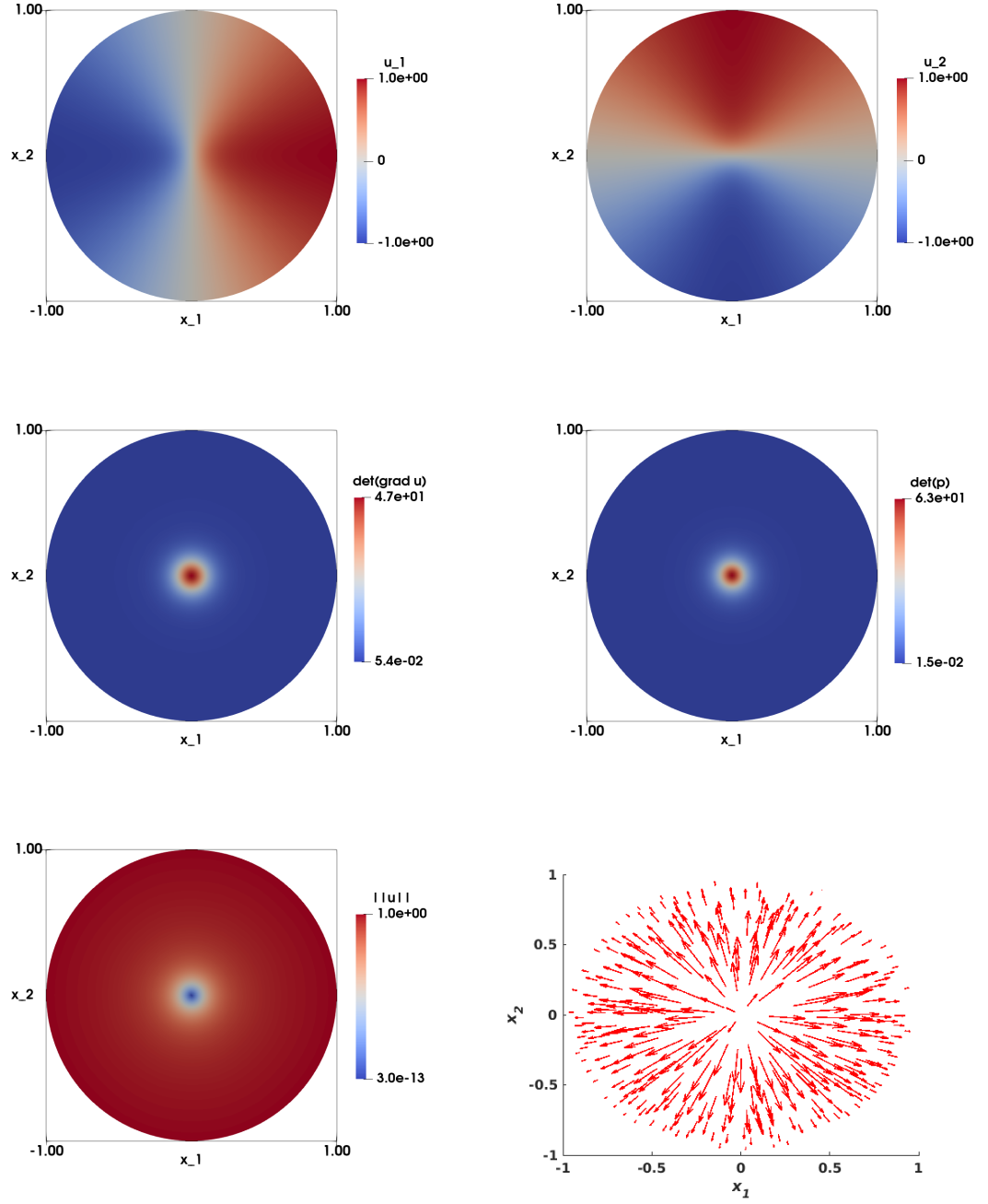


Figure 2.16 – Non-smooth right-hand side with a Dirac delta function test case with data $f = \frac{\eta^2}{(\eta^2 + \|\mathbf{x}\|_2^2)^2}$, $\mathbf{u}_\eta = \mathbf{x} \sqrt{\frac{1+\eta^2}{\eta^2 + \|\mathbf{x}\|_2^2}}$ and $\eta = 1/8$. Top Left: Numerical approximation of the solution of the component $u_{1,h}$. Top right: Numerical approximation of the solution of the component $u_{2,h}$. Middle left: Numerical approximation of $\det \nabla \mathbf{u}_h$. Middle right: Numerical approximation of $\det \mathbf{p}_h$. Bottom left: Numerical approximation of $\|\mathbf{u}_h\|$. Bottom right: Visualization of the vector field bu_h . The results are obtained on structured mesh of the unit disk with $h = 0.0209$ and $\varepsilon = h^2$.

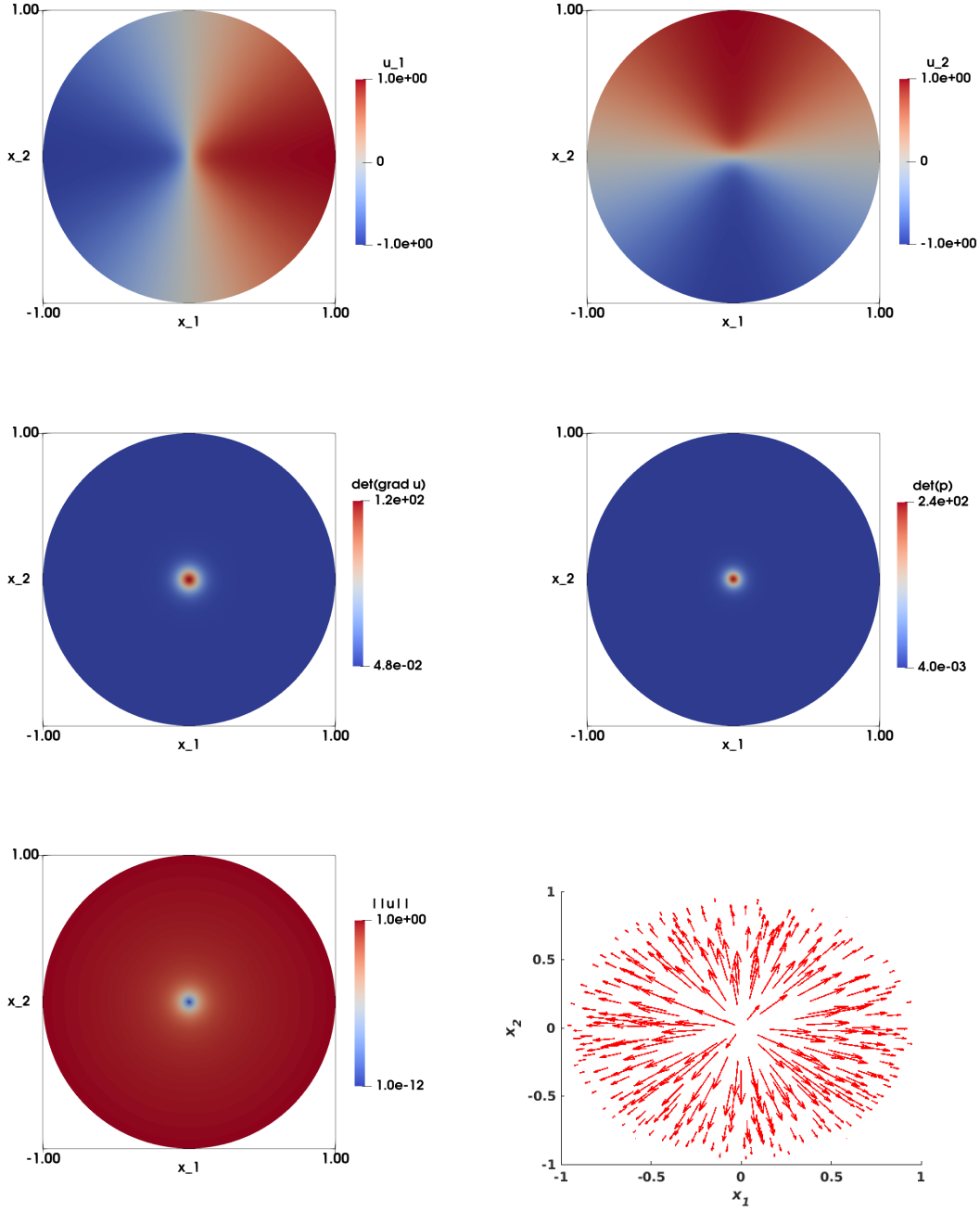


Figure 2.17 – Non-smooth right-hand side with a Dirac delta function test case with data $f = \frac{\eta^2}{(\eta^2 + \|\mathbf{x}\|_2^2)^2}$, $\mathbf{u}_\eta = \mathbf{x} \sqrt{\frac{1+\eta^2}{\eta^2 + \|\mathbf{x}\|_2^2}}$ and $\eta = 1/16$. Top Left: Numerical approximation of the solution of the component $u_{1,h}$. Top right: Numerical approximation of the solution of the component $u_{2,h}$. Middle left: Numerical approximation of $\det \nabla \mathbf{u}_h$. Middle right: Numerical approximation of $\det \mathbf{p}_h$. Bottom left: Numerical approximation of $\|\mathbf{u}_h\|$. Bottom right: Visualization of the vector field $\mathbf{u}_h$. The results are obtained on structured mesh of the unit disk with $h = 0.0209$ and $\varepsilon = h^2$.

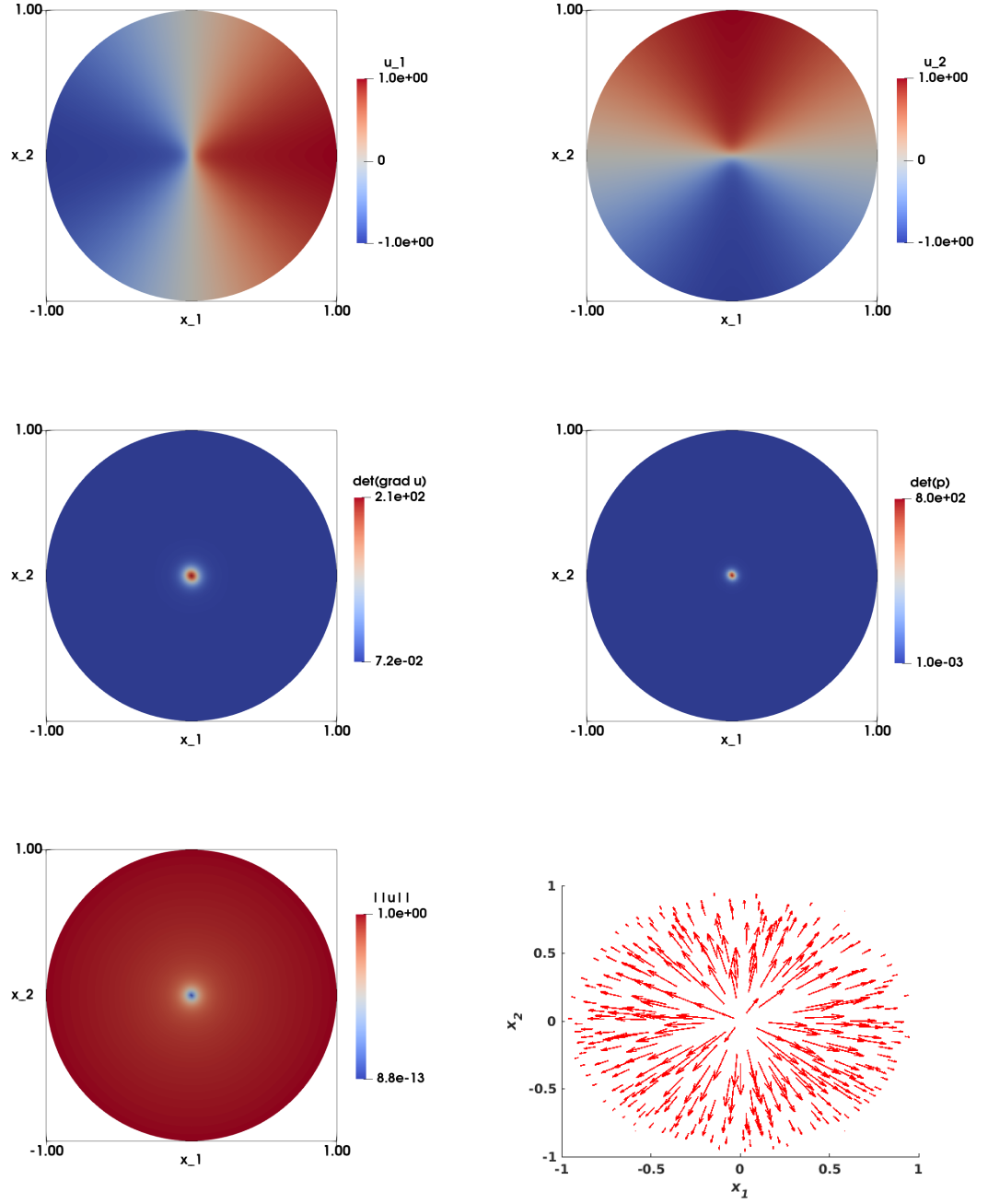


Figure 2.18 – Non-smooth right-hand side with a Dirac delta function test case with data $f = \frac{\eta^2}{(\eta^2 + \|\mathbf{x}\|_2^2)^2}$, $\mathbf{u}_\eta = \mathbf{x} \sqrt{\frac{1+\eta^2}{\eta^2 + \|\mathbf{x}\|_2^2}}$ and $\eta = 1/32$. Top Left: Numerical approximation of the solution of the component $u_{1,h}$. Top right: Numerical approximation of the solution of the component $u_{2,h}$. Middle left: Numerical approximation of $\det \nabla \mathbf{u}_h$. Middle right: Numerical approximation of $\det \mathbf{p}_h$. Bottom left: Numerical approximation of $\|\mathbf{u}_h\|$. Bottom right: Visualization of the vector field $\mathbf{u}_h$. The results are obtained on structured mesh of the unit disk with $h = 0.0209$ and $\varepsilon = h^2$.

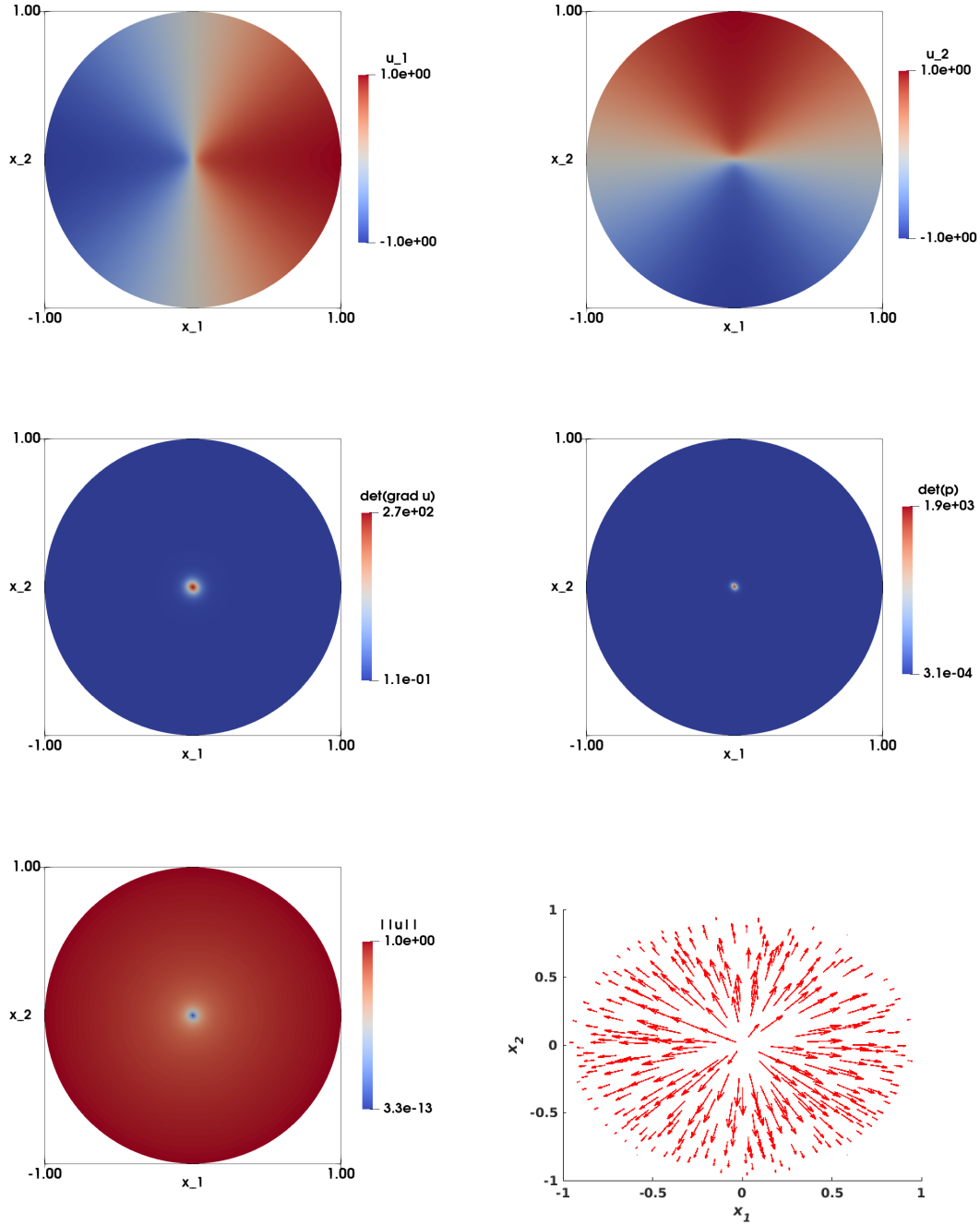


Figure 2.19 – Non-smooth right-hand side with a Dirac delta function test case with data $f = \frac{\eta^2}{(\eta^2 + ||\mathbf{x}||_2^2)^2}$, $\mathbf{u}_\eta = \mathbf{x} \sqrt{\frac{1+\eta^2}{\eta^2 + ||\mathbf{x}||_2^2}}$ and $\eta = 1/64$. Top Left: Numerical approximation of the solution of the component $u_{1,h}$. Top right: Numerical approximation of the solution of the component $u_{2,h}$. Middle left: Numerical approximation of $\det \nabla \mathbf{u}_h$. Middle right: Numerical approximation of $\det \mathbf{p}_h$. Bottom left: Numerical approximation of $||\mathbf{u}_h||$. Bottom right: Visualization of the vector field $\mathbf{u}_h$. The results are obtained on structured mesh of the unit disk with $h = 0.0209$ and $\varepsilon = h^2$.

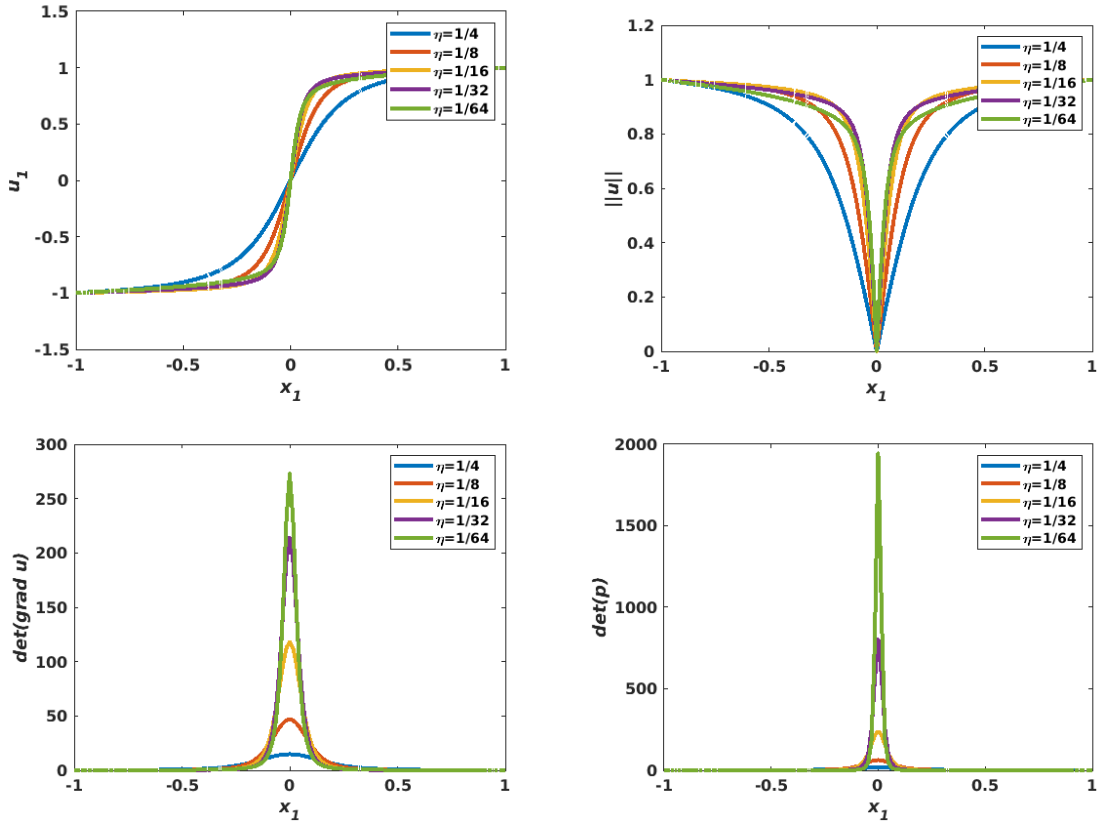


Figure 2.20 – Non-smooth right-hand side with a Dirac delta function test case with data $f = \frac{\eta^2}{(\eta^2 + \|\mathbf{x}\|_2^2)^2}$, $\mathbf{u}_\eta = \mathbf{x} \sqrt{\frac{1+\eta^2}{\eta^2 + \|\mathbf{x}\|_2^2}}$ and various value of the parameter η . Computation on structured mesh of the unit disk with $h = 0.00209$. Comparing profiles for different values of η along the line $x_2 = 0$; Top left : $u_{1,h}$ component versus x_1 . Top right: $\|\mathbf{u}_h\|$ versus x_1 . Bottom left: $\det \nabla \mathbf{u}_h$ versus x_1 . Bottom right: $\det \mathbf{p}_h$ versus x_1 .

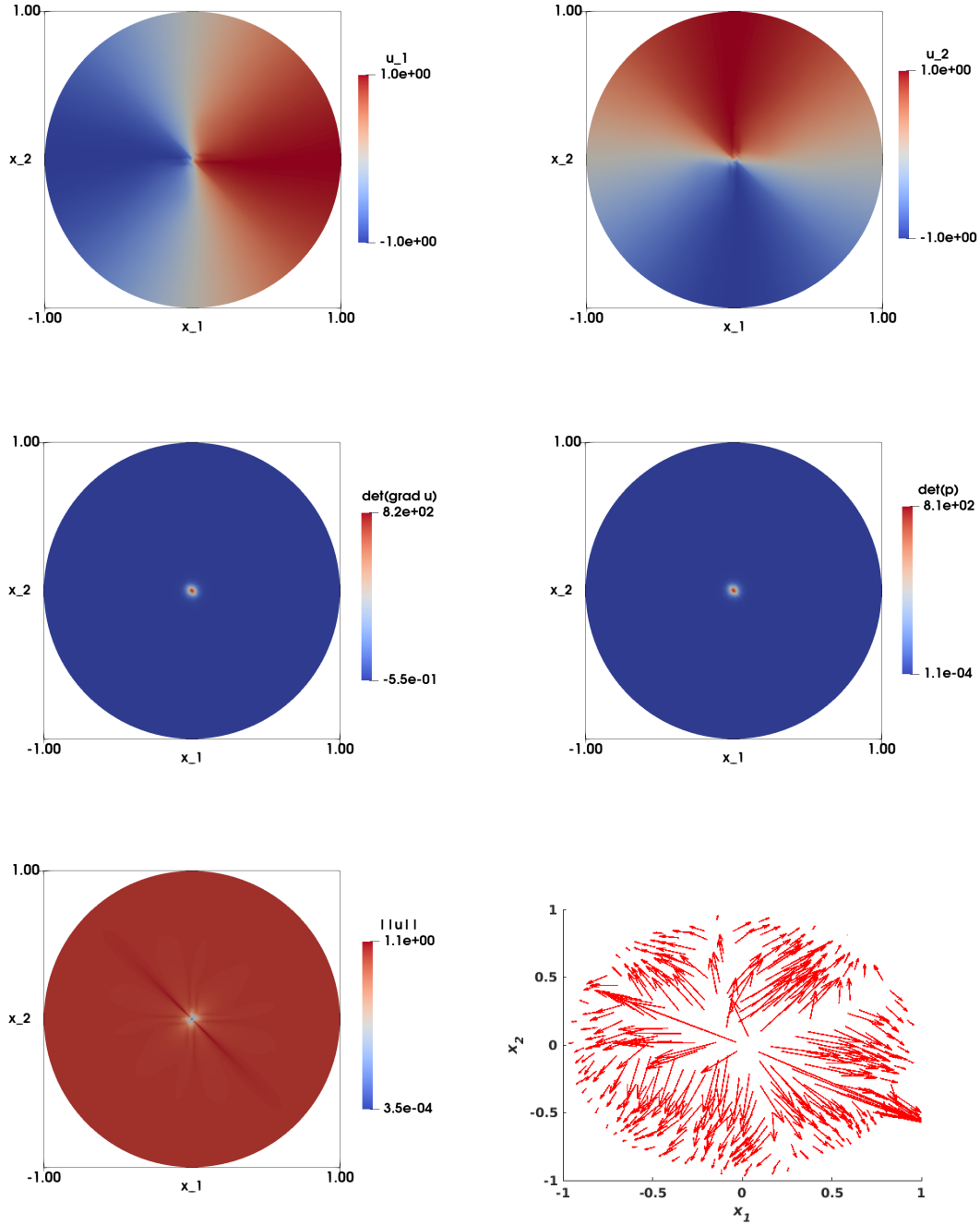


Figure 2.21 – Non-smooth right-hand side with a Dirac delta function test case with data $f = \frac{\eta^2}{(\eta^2 + \|\mathbf{x}\|_2^2)^2}$, $\mathbf{u}_\eta = \mathbf{x} \sqrt{\frac{1+\eta^2}{\eta^2 + \|\mathbf{x}\|_2^2}}$ and $\eta = 1/32$. Top Left: Numerical approximation of the solution of the component $u_{1,h}$. Top right: Numerical approximation of the solution of the component $u_{2,h}$. Middle left: Numerical approximation of $\det \nabla \mathbf{u}_h$. Middle right: Numerical approximation of $\det \mathbf{p}_h$. Bottom left: Numerical approximation of $\|\mathbf{u}_h\|$. Bottom right: Visualization of the vector field $\mathbf{u}_h$. The results are obtained on structured mesh of the unit disk with $h = 0.0209$ and $\varepsilon = 0$.

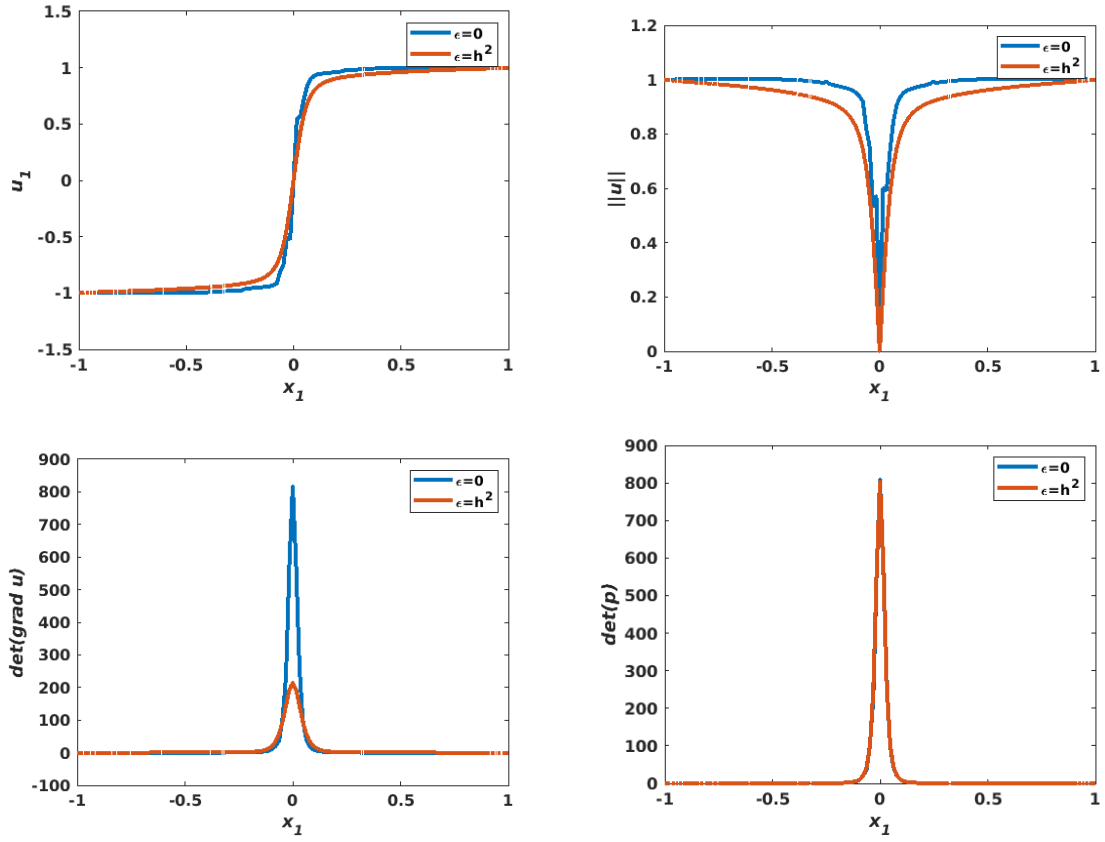


Figure 2.22 – Non-smooth right-hand side with a Dirac delta function test case with data $f = \frac{\eta^2}{(\eta^2 + \|\mathbf{x}\|_2^2)^2}$, $\mathbf{u}_\eta = \mathbf{x} \sqrt{\frac{1+\eta^2}{\eta^2 + \|\mathbf{x}\|_2^2}}$ and various value of the parameter η . Computation on structured mesh of the unit disk with $h = 0.00209$. Comparing plots between $\epsilon = 0$ and $\epsilon = h^2$ along the line $x_2 = 0$; Top left : $u_{1,h}$ component versus x_1 . Top right: $\|\mathbf{u}_h\|$ versus x_1 . Bottom left: $\det \nabla \mathbf{u}_h$ versus x_1 . Bottom right: $\det \mathbf{p}_h$ versus x_1 .

3 Numerical Approximation of Monge-Ampère Equation

3.1 Mathematical Formulation and Least-Squares Approach

Let Ω be a bounded convex domain of $\mathbb{R}^3$; we denote by Γ the boundary of Ω . The Dirichlet problem for the elliptic Monge-Ampère equation reads as follows:

$$\begin{cases} \det \mathbf{D}^2 u = f (> 0) & \text{in } \Omega, \\ u = g & \text{on } \Gamma, \end{cases} \quad (3.1)$$

where $\mathbf{D}^2 u = \left(\frac{\partial^2 u}{\partial x_i \partial x_j} \right)_{1 \leq i, j \leq 3}$ is the Hessian of the unknown function u .

As in the previous chapter, we consider a nonlinear least-squares method that relies on the introduction of an additional auxiliary variable. We define the functional spaces by :

$$\begin{aligned} V_g &= \{ \varphi \in H^2(\Omega), \varphi = g \text{ on } \Gamma \}, \\ \mathbf{Q} &= \{ \mathbf{q} \in L^2(\Omega)^{3 \times 3}, \mathbf{q} = \mathbf{q}^t \}, \\ \mathbf{Q}_f &= \{ \mathbf{q} \in \mathbf{Q}, \det \mathbf{q} = f, \mathbf{q} \text{ is a positive definite matrix-valued function} \}. \end{aligned}$$

The problem is written as

$$\text{Find } (u, \mathbf{p}) \in V_g \times \mathbf{Q}_f \text{ such that } J(u, \mathbf{p}) \leq J(v, \mathbf{q}), \quad \forall (v, \mathbf{q}) \in V_g \times \mathbf{Q}_f, \quad (3.2)$$

where

$$J(v, \mathbf{q}) = \frac{1}{2} \int_{\Omega} |\mathbf{D}^2 v - \mathbf{q}|^2 d\mathbf{x}, \quad (3.3)$$

and $|\cdot|$ denotes the Frobenius norm $|\mathbf{T}| = (\mathbf{T} : \mathbf{T})^{1/2}$, with the inner product $\mathbf{S} : \mathbf{T} = \sum_{i,j=1}^3 s_{ij} t_{ij}$ where $\mathbf{T}, \mathbf{S}$ are 3×3 matrices with elements s_{ij}, t_{ij} for $i, j = 1, 2, 3$, respectively.

We assume that $f \in L^{2/3}(\Omega)$ and $g \in H^{3/2}(\Gamma)$, so that V_g and $\mathbf{Q}_f$ are both non-empty. The space $\mathbf{Q}$ is a Hilbert space for the inner product $(\mathbf{q}, \mathbf{q}') \rightarrow \int_{\Omega} \mathbf{q} : \mathbf{q}' d\mathbf{x}$, and the associated norm.

3.2 Relaxation Algorithm

For the solution of the minimization problem (3.2), we propose a relaxation algorithm, which decouples the differential operators from the nonlinearities. Moreover the proposed algorithm allows to compute convex solutions or to force the convexity of the solution.

Step 1. The initialization is performed by solving:

$$\begin{cases} \Delta u^0 = 3\sqrt[3]{f} & \text{in } \Omega, \\ u^0 = g & \text{on } \Gamma. \end{cases} \quad (3.4)$$

For the rationale behind this problem see the remark below.

Then, for $n \geq 0$

Step 2. Assuming that u^n is known, we look for

$$\mathbf{p}^n = \arg \min_{\mathbf{q} \in \mathbf{Q}_f} J(u^n, \mathbf{q}); \quad (3.5)$$

Step 3. Assuming that $\mathbf{p}^n$ is known, we look for

$$u^{n+1/2} = \arg \min_{v \in V_g} J(v, \mathbf{p}^n); \quad (3.6)$$

Step 4. Update the solution as

$$u^{n+1} = u^n + \omega(u^{n+1/2} - u^n), \quad (3.7)$$

with $1 \leq \omega \leq \omega_{\max} < 2$. For the numerical experiments presented in Section 3.6, we have used $\omega \equiv 1$ (unless otherwise specified).

In the next section, we discuss the numerical algorithms that are used for the solution of problems (3.5) and (3.6). The question of the uniqueness of the solution to the local problem (3.5) still remains an open question and it is not addressed here.

Remark. The initialization procedure (3.4) is based on the following assumption: If we denote the eigenvalues of $\mathbf{D}^2 u$ by λ_i , $i = 1, 2, 3$, the Monge-Ampère equation reads $\lambda_1 \lambda_2 \lambda_3 = f$. If λ_1, λ_2 and λ_3 are 'close' from each other (and thus all equal to, let's say, λ), we have $\lambda^3 = f$, and thus $\lambda = \sqrt[3]{f}$. Therefore:

$$\Delta u = \lambda_1 + \lambda_2 + \lambda_3 = 3\lambda = 3\sqrt[3]{f}.$$

Note here that the case when the eigenvalues are very different, the efficiency of this initialization is not guaranteed.

3.3 Numerical Approximation of the Local Nonlinear Problems

3.3.1 Explicit Formulation of the Local Nonlinear Problems

An explicit formulation of problem (3.5) reads as

$$\mathbf{p}^n = \arg \min_{\mathbf{q} \in \mathbf{Q}_f} \left[\frac{1}{2} \int_{\Omega} |\mathbf{q}|^2 d\mathbf{x} - \int_{\Omega} \mathbf{D}^2 u^n : \mathbf{q} d\mathbf{x} \right]. \quad (3.8)$$

Since the objective function in (3.8) does not contain derivatives of $\mathbf{q}$, this minimization problem can be solved point-wise (in practice at the vertices of a finite element or finite difference grid). This leads, a.e. in Ω , to the solution of the following finite dimensional minimization problem:

$$\mathbf{p}^n(\mathbf{x}) = \arg \min_{\mathbf{q} \in \mathbf{E}_f(\mathbf{x})} \left[\frac{1}{2} |\mathbf{q}|^2 - \mathbf{D}^2 u^n(\mathbf{x}) : \mathbf{q} \right], \quad (3.9)$$

where

$$\mathbf{E}_f(\mathbf{x}) = \{ \mathbf{q} \in \mathbb{R}^{3 \times 3}, \mathbf{q} = \mathbf{q}^t, \det \mathbf{q} = f(\mathbf{x}), \mathbf{q} \text{ is positive definite} \}.$$

Problem (3.9) is similar to the nonlinear problem examined in Chapter 2. In that case, the problem was in 2D, and it did not require $\mathbf{q}$ to be a positive definite symmetric matrix. Therefore the approach that is considered in Chapter 2 can not be applied here. The same problem as in (3.9) but in the 2D case is considered in [Caboussat et al., 2013; Sorensen and Glowinski, 2010]. The methods that are developed there consist of a proper change of variables and the use of a class of quadratically constrained minimization problems. None of these two methods can be extended for the 3D case since the nonlinearity $\lambda_1 \lambda_2 \lambda_3 = f$ is cubic, implying that other approaches must be considered. The two approaches that we develop below rely on an appropriate re-parameterization of the problem, which allows transforming the constrained minimization problem into an unconstrained one. Finally, both approaches end up using Newton methods. Moreover, the approaches below can be applied to the case of the 2D Monge-Ampère equation with small modifications.

3.3.2 A Reduced Newton Method

For a.e. $\mathbf{x} \in \Omega$, (3.9) is an algebraic optimization problem. Using a Cholesky decomposition of $\mathbf{q}$, we write $\mathbf{q} = \mathbf{L}\mathbf{D}\mathbf{L}^t$, where

$$\mathbf{L} = \begin{pmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ b & c & 1 \end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix} \sqrt[3]{f(\mathbf{x})}e^{\rho_1} & 0 & 0 \\ 0 & \sqrt[3]{f(\mathbf{x})}e^{\rho_2} & 0 \\ 0 & 0 & \sqrt[3]{f(\mathbf{x})}e^{-\rho_1-\rho_2} \end{pmatrix}. \quad (3.10)$$

This re-parameterization is arbitrary but serves two purposes: first, it guarantees that all eigenvalues are strictly positive (convexity of the local solution). Second, the constraint $\det \mathbf{q} = f(\mathbf{x})$ is automatically satisfied. It thus allows one to replace (3.9) by an unconstrained minimization problem in the variable $\mathbf{X} := (a, b, c, \rho_1, \rho_2)$. For the sake of simplicity, we do not write the dependency on $\mathbf{x} \in \Omega$ anymore. The problem becomes:

$$\min_{\mathbf{X} \in \mathbb{R}^5} G(\mathbf{X}) = \left\{ \frac{1}{2} \mathbf{L}\mathbf{D}\mathbf{L}^t : \mathbf{L}\mathbf{D}\mathbf{L}^t - \mathbf{D}^2 u^n \right\}. \quad (3.11)$$

The first order optimality conditions corresponding to (3.11) can formally be written as

$$\nabla_{\mathbf{X}} G(\mathbf{X}) = \mathbf{0}.$$

This nonlinear system can be solved with a safeguarded Newton method for the variable $\mathbf{X}$. Namely, given $\mathbf{X}^0 \in \mathbb{R}^5$, solve, for $k \geq 0$:

$$\nabla_{\mathbf{X}}^2 G(\mathbf{X}^{k+1}) \delta \mathbf{X}^k = -\nabla_{\mathbf{X}} G(\mathbf{X}^k),$$

followed by

$$\mathbf{X}^{k+1} = \mathbf{X}^k + \lambda^k \delta \mathbf{X}^k,$$

where $\lambda^k \in \mathbb{R}_+$ is a step-length to be adapted according to some Armijo rule (see, e.g., Botsaris [1978]). Typically, we update the step-length if $\|\nabla G(\mathbf{X}^{k+1})\| > (1 - \alpha \lambda^k) \|\nabla G(\mathbf{X}^k)\|$, where $\alpha = 10^{-4}$ and $\|\cdot\|$ denotes the canonical Euclidean norm of $\mathbb{R}^5$, and set in that case $\lambda^{k+1} = \frac{1}{2} \lambda^k$.

The stopping criterion is based on the residual value $\|\nabla G(\mathbf{X}^k)\|$, and the iterations are stopped if $\|\nabla G(\mathbf{X}^k)\| < \varepsilon_{\text{Newton}}$, where $\varepsilon_{\text{Newton}}$ is a given tolerance.

3.3.3 A Runge-Kutta Method for the Dynamical Flow Problem

An alternative for the re-parameterization of the nonlinear problem (3.9) can be considered, based on a eigenvalues-eigenvectors decomposition, in the spirit of the approach in Sorensen and Glowinski [2010]. Namely, consider $\mathbf{q} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^t$, where $\mathbf{Q} \in \mathcal{O}(3) \subset \mathbb{R}^{3 \times 3}$ is the orthogonal matrix whose columns represent the eigenvectors of $\mathbf{q}$ ($\mathcal{O}(3)$ being the group of the 3×3

3.3. Numerical Approximation of the Local Nonlinear Problems

orthogonal matrices), and $\Lambda \in \mathbb{R}^{3 \times 3}$ is the diagonal matrix whose diagonal elements are the corresponding eigenvalues. We can denote

$$\mathbf{Q} = \left(\mathbf{T}_1 \mid \mathbf{T}_2 \mid \mathbf{T}_3 \right), \Lambda = \begin{pmatrix} \sqrt[3]{f(\mathbf{x})}e^{\rho_1} & 0 & 0 \\ 0 & \sqrt[3]{f(\mathbf{x})}e^{\rho_2} & 0 \\ 0 & 0 & \sqrt[3]{f(\mathbf{x})}e^{-\rho_1-\rho_2} \end{pmatrix}$$

This parameterization is not unique; it ensures that the relation $\det \mathbf{q} = f(\mathbf{x})$ is automatically satisfied and that the eigenvalues are positive in order to ensure convexity of the local solutions. The property $\mathbf{Q} \in \mathcal{O}(3)$ implies the following constraints for its column vectors $\mathbf{T}_i, i = 1, 2, 3$ (where $\mathbf{T}_i \cdot \mathbf{T}_j$ denotes the dot product of vectors $\mathbf{T}_i$ and $\mathbf{T}_j$):

$$\mathbf{T}_i \cdot \mathbf{T}_j = \delta_{ij}, \quad i, j = 1, 2, 3.$$

Let us define the variables $\mathbf{Y} = (\rho_1, \rho_2, \mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3) \in \mathbb{R}^{11}$. Problem (3.9) can be rewritten as

$$\begin{aligned} \min_{\mathbf{Y} \in \mathbb{R}^{11}} \quad & \left\{ \frac{1}{2} \mathbf{Q} \Lambda \mathbf{Q}^t : \mathbf{Q} \Lambda \mathbf{Q}^t - \mathbf{Q} \Lambda \mathbf{Q}^t : \mathbf{D}^2 u^n \right\} \\ \text{s. t.} \quad & \mathbf{T}_1 \cdot \mathbf{T}_1 = \mathbf{T}_2 \cdot \mathbf{T}_2 = \mathbf{T}_3 \cdot \mathbf{T}_3 = 1 \\ & \mathbf{T}_1 \cdot \mathbf{T}_2 = \mathbf{T}_1 \cdot \mathbf{T}_3 = \mathbf{T}_2 \cdot \mathbf{T}_3 = 0 \end{aligned}$$

Below, for $i = 1, 2, 3$, we will denote by $|\mathbf{T}_i|$ the quantity $(\mathbf{T}_i \cdot \mathbf{T}_i)^{1/2}$. We penalize the equality constraints in order to obtain an unconstrained problem that can be solved by a Newton approach. Let $\varepsilon_1, \varepsilon_2 > 0$ be two given (small) parameters. The constraints are taken into account by penalization, leading to the following unconstrained minimization problem:

$$\min_{\mathbf{Y} \in \mathbb{R}^{11}} G_{\varepsilon_1, \varepsilon_2}(\mathbf{Y}) \tag{3.12}$$

where

$$\begin{aligned} G_{\varepsilon_1, \varepsilon_2}(\mathbf{Y}) = & \frac{1}{2} \mathbf{Q} \Lambda \mathbf{Q}^t : \mathbf{Q} \Lambda \mathbf{Q}^t - \mathbf{Q} \Lambda \mathbf{Q}^t : \mathbf{D}^2 u^n \\ & + \frac{1}{\varepsilon_1} ((|\mathbf{T}_1|^2 - 1)^2 + (|\mathbf{T}_2|^2 - 1)^2 + (|\mathbf{T}_3|^2 - 1)^2) \\ & + \frac{1}{\varepsilon_2} ((\mathbf{T}_1 \cdot \mathbf{T}_2)^2 + (\mathbf{T}_1 \cdot \mathbf{T}_3)^2 + (\mathbf{T}_2 \cdot \mathbf{T}_3)^2). \end{aligned}$$

Similarly to the solution of (3.11), the first order optimality conditions associated with (3.12)

can be written as

$$\nabla G_{\varepsilon_1, \varepsilon_2}(\mathbf{Y}) = \mathbf{0}.$$

In order to smoothen the transition to critical point(s), we favor an evolutive formulation of the first order optimality conditions (in the sense of a flow problem in the dynamical systems terminology), which read as follows: find $\mathbf{Y} : (0, +\infty) \rightarrow \mathbb{R}^{11}$ such that

$$\frac{d\mathbf{Y}}{dt} + \nabla G_{\varepsilon_1, \varepsilon_2}(\mathbf{Y}) = \mathbf{0}, \quad t \in (0, +\infty) \quad (3.13)$$

$$\mathbf{Y}(0) = \mathbf{Y}^0 \text{ given.} \quad (3.14)$$

The steady state solution of (3.13) (3.14) corresponds to the desired critical point. In order to increase the stability of the numerical scheme and allow larger time steps and therefore a faster convergence to the steady state solution, it is customary to modify (3.13) into an equivalent modified flow problem [Kelley, 1995], namely: find $\mathbf{Y} : (0, +\infty) \rightarrow \mathbb{R}^{11}$ such that

$$\frac{d\mathbf{Y}}{dt} + (\nabla^2 G_{\varepsilon_1, \varepsilon_2}(\mathbf{Y}))^{-1} \nabla G_{\varepsilon_1, \varepsilon_2}(\mathbf{Y}) = \mathbf{0}, \quad t \in (0, +\infty) \quad (3.15)$$

with the same initial condition. The stability of the scheme is important here since we are aiming at solving such a flow problem for a.e. $\mathbf{x} \in \Omega$, which requires an efficient numerical algorithm. The additional computational cost induced by the introduction of the term $(\nabla^2 G_{\varepsilon_1, \varepsilon_2}(\mathbf{Y}))^{-1}$ is estimated in the sequel.

System (3.15) is solved by a two-stage (second order explicit) Runge-Kutta method (see, e.g., [Hairer et al., 1993]) in order to capture steady state solutions to quickly reach the stationary solution. Let Δt be a given time step, $t_n = n\Delta t$ and $\mathbf{Y}_n \simeq \mathbf{Y}(t_n)$, $n = 0, 1, \dots$. Let us define $\mathbf{Y}_0 = \mathbf{Y}^0$; then, at each time step, solve

$$\begin{aligned} \mathbf{k}_1 &= -(\nabla^2 G_{\varepsilon_1, \varepsilon_2}(\mathbf{Y}_n))^{-1} \nabla G_{\varepsilon_1, \varepsilon_2}(\mathbf{Y}_n), \\ \mathbf{k}_2 &= -\left(\nabla^2 G_{\varepsilon_1, \varepsilon_2}\left(\mathbf{Y}_n + \frac{2}{3}\Delta t \mathbf{k}_1\right)\right)^{-1} \nabla G_{\varepsilon_1, \varepsilon_2}\left(\mathbf{Y}_n + \frac{2}{3}\Delta t \mathbf{k}_1\right), \\ \mathbf{Y}_{n+1} &= \mathbf{Y}_n + \Delta t \left(\frac{1}{4}\mathbf{k}_1 + \frac{3}{4}\mathbf{k}_2\right). \end{aligned}$$

An adaptive time stepping strategy for Runge-Kutta methods is incorporated to the numerical algorithm; numerical experiments will show that the adaptive time step is particularly useful at the beginning of the outer iterations loop, when the initial solution is not close to the final steady state solution.

Remark. If we treat the modified flow problem with a first order Euler explicit scheme, it leads to solving at each time step

$$(\nabla^2 G_{\varepsilon_1, \varepsilon_2}(\mathbf{Y}^n)) \frac{\mathbf{Y}^{n+1} - \mathbf{Y}^n}{\Delta t} = -\nabla G_{\varepsilon_1, \varepsilon_2}(\mathbf{Y}^n);$$

this problem corresponds actually to a classical safeguarded Newton method, reminiscent to the one we presented in Section 3.3.2, with Δt playing the role of the step-length λ . With this remark, the adaptive time stepping algorithm for Runge-Kutta schemes can be seen as an adaptive Armijo-like rule, with $\Delta t = \lambda$. Furthermore, one can see that the Runge-Kutta approach is slower than the reduced Newton strategy (since it corresponds to solving two Newton-type systems at each time step), but it is more accurate since the two-step Runge-Kutta scheme is a higher order method than the Euler scheme. Finally, a study of the stability of Runge-Kutta schemes [Hairer et al., 1993] shows that their stability properties are better than those of the Euler scheme.

3.4 Numerical Solution of the Linear Variational Problems

Written in variational form the Euler-Lagrange equation of the sub-problem (3.6) reads as follows: find $u^{n+1/2} \in V_g$ satisfying:

$$\int_{\Omega} \mathbf{D}^2 u^{n+1/2} : \mathbf{D}^2 \varphi d\mathbf{x} = \int_{\Omega} \mathbf{p}^n : \mathbf{D}^2 \varphi d\mathbf{x}, \quad \forall \varphi \in V_0, \quad (3.16)$$

where $V_0 = H^2(\Omega) \cap H_0^1(\Omega)$. The linear variational problem (3.16) is well-posed and belongs to the following family of linear variational problems:

$$\text{Find } u \in V_g \text{ such that } \int_{\Omega} \mathbf{D}^2 u : \mathbf{D}^2 v d\mathbf{x} = L(v), \quad \forall v \in V_0, \quad (3.17)$$

with the functional $L(\cdot)$ linear and continuous over $H^2(\Omega)$; problem (3.17) is a bi-harmonic type problem, which can be solved by a conjugate gradient algorithm operating in well-chosen Hilbert spaces (see, e.g., [Glowinski, 2003, Chapter 3]). Here, our conjugate gradient algorithm operates in the spaces V_0 and V_g , both spaces being equipped with the inner product defined by $(v, w) \rightarrow \int_{\Omega} \Delta v \Delta w d\mathbf{x}$, and the corresponding norm. It reads as follows:

Step 1

$$u^0 \in V_g \text{ given.} \quad (3.18)$$

Step 2 Solve: find $g^0 \in V_g$ satisfying

$$\int_{\Omega} \Delta g^0 \Delta v d\mathbf{x} = \int_{\Omega} \mathbf{D}^2 u^0 : \mathbf{D}^2 v d\mathbf{x} - L(v), \quad \forall v \in V_0, \quad (3.19)$$

and set

$$w^0 = g^0. \quad (3.20)$$

Then, for $k \geq 0$, u^k, g^k and w^k being known, the last two different from zero, we compute u^{k+1}, g^{k+1} and, if necessary, w^{k+1} as follows.

Step 3 Solve: find $\bar{g}^k \in V_0$ satisfying

$$\int_{\Omega} \Delta \bar{g}^k \Delta v d\mathbf{x} = \int_{\Omega} \mathbf{D}^2 w^k : \mathbf{D}^2 v d\mathbf{x}, \quad \forall v \in V_0, \quad (3.21)$$

and compute

$$\rho_k = \frac{\int_{\Omega} |\Delta g^k|^2 d\mathbf{x}}{\int_{\Omega} \Delta \bar{g}^k \Delta w^k d\mathbf{x}}, \quad (3.22)$$

$$u^{k+1} = u^k - \rho_k w^k, \quad (3.23)$$

Step 4 Compute

$$g^{k+1} = g^k - \rho_k \bar{g}^k. \quad (3.24)$$

$$\delta_k = \frac{\int_{\Omega} |\Delta g^{k+1}|^2 d\mathbf{x}}{\int_{\Omega} |\Delta g^0|^2 d\mathbf{x}}. \quad (3.25)$$

If $\delta_k < \varepsilon$, take $u = u^{k+1}$; otherwise, compute:

$$\gamma_k = \frac{\int_{\Omega} |\Delta g^{k+1}|^2 d\mathbf{x}}{\int_{\Omega} |\Delta g^k|^2 d\mathbf{x}}; \quad (3.26)$$

and

$$w^{k+1} = g^{k+1} + \gamma_k w^k. \quad (3.27)$$

Step 5 Do $k+1 \rightarrow k$ and return to Step 3.

Steps 1,2 are the initialization of the algorithm; **Step 3** computes the steepest descent and updates the solution; **Step 4** checks the convergence and constructs the new descent direction.

3.5 Mixed Finite Element Approximation

We use a mixed finite element approximation (similarly discussed in, e.g., [Glowinski, 2015a] for the solution of linear and nonlinear bi-harmonic problems) with low order $\mathbb{P}_1$ (piecewise linear and globally continuous) finite elements on a partition of Ω made of tetrahedra. Moreover, for comparison, we also provide numerical examples based on $\mathbb{Q}_1$ finite elements on a partition of Ω made of hexahedra, in Section 3.7.

3.5.1 Finite Element Spaces

For simplicity, let us assume that Ω is a bounded polyhedral domain of $\mathbb{R}^3$, and define $\mathcal{T}_h$ as a finite element partition of Ω made out of tetrahedra (see, e.g., [Glowinski, 2008, Appendix 1]). Let Σ_h be the set of the vertices of $\mathcal{T}_h$, $N_h = \text{Card}(\Sigma_h)$, $\Sigma_{0h} = \{P \in \Sigma_h, P \notin \Gamma\}$, and $N_{0h} = \text{Card}(\Sigma_{0h})$. We suppose that $\Sigma_{0h} = \{P_j\}_{j=1}^{N_{0h}}$ and $\Sigma_h = \Sigma_{0h} \cup \{P_j\}_{j=N_{0h}+1}^{N_h}$.

From $\mathcal{T}_h$, we approximate the spaces $L^2(\Omega)$, $H^1(\Omega)$ and $H^2(\Omega)$ by the finite dimensional space V_h defined by:

$$V_h = \{v \in C^0(\bar{\Omega}), v|_T \in \mathbb{P}_1, \forall T \in \mathcal{T}_h\},$$

with $\mathbb{P}_1$ the space of the three-variable polynomials of degree ≤ 1 . We define also V_{0h} as

$$V_{0h} = V_h \cap H_0^1(\Omega) = \{v \in V_h, v = 0 \text{ on } \Gamma\}.$$

In the sequel, V_{0h} will be used to approximate both $H_0^1(\Omega)$ and $H^2(\Omega) \cap H_0^1(\Omega)$.

3.5.2 Finite Element Approximation Tools of the Monge-Ampère Equation

To use first-order $\mathbb{P}_1$ finite elements, we require to be able to compute second derivatives. For instance, when solving (3.17) by the conjugate gradient algorithm (3.18)-(3.27), one has to i) compute the discrete analogues of the second order derivatives, e.g., $\mathbf{D}^2 w^k$ and $\mathbf{D}^2 u^0$, and ii) solve biharmonic problems such as (3.19) and (3.21).

Concerning i) we will approximate the second order derivatives by functions belonging to V_{0h} . For a function φ belonging to $H^2(\Omega)$, it follows from Green's formula that, for $i, j = 1, 2, 3$:

$$\int_{\Omega} \frac{\partial^2 \varphi}{\partial x_i \partial x_j} v d\mathbf{x} = -\frac{1}{2} \int_{\Omega} \left[\frac{\partial \varphi}{\partial x_i} \frac{\partial v}{\partial x_j} + \frac{\partial \varphi}{\partial x_j} \frac{\partial v}{\partial x_i} \right] d\mathbf{x}, \forall v \in H_0^1(\Omega), \quad (3.28)$$

Consider now $\varphi \in V_h$. We define the discrete analogue $D_{hij}^2 \varphi \in V_{0h}$ of the second derivative $\frac{\partial^2 \varphi}{\partial x_i \partial x_j}$ by : for all $i, j, 1 \leq i, j, \leq 3$, $D_{hij}^2 \varphi \in V_{0h}$ is defined by

$$\int_{\Omega} D_{hij}^2 \varphi v d\mathbf{x} = -\frac{1}{2} \int_{\Omega} \left[\frac{\partial \varphi}{\partial x_i} \frac{\partial v}{\partial x_j} + \frac{\partial \varphi}{\partial x_j} \frac{\partial v}{\partial x_i} \right] d\mathbf{x}, \quad \forall v \in V_{0h}. \quad (3.29)$$

The functions $D_{hij}^2 \varphi$ are thus uniquely defined; Since the Hessian matrix $\mathbf{D}^2 u$ is in $(L^2(\Omega))^{3 \times 3}$, and since $H_0^1(\Omega)$ is dense in $L^2(\Omega)$, the approximation $\mathbf{D}_h^2 u$ is considered to be in V_{0h} .

As emphasized in [Caboussat et al., 2013; Picasso et al., 2011], when using piecewise linear mixed finite elements, the approximation of the error on the second derivatives of the solution u is, in general, $\mathcal{O}(1)$ in the L^2 -norm. Therefore, the convergence properties of the global algorithm strongly depends on the type of partition of Ω one employs. A way to improve the approximation properties of the discrete second order derivatives $D_{hij}^2 \varphi$ is to use, a Tychonoff-like regularization [Tychonoff, 1963], as in [Caboussat et al., 2013]. Let us introduce a stabilization constant C , and replace the previous variational problem by: for all $i, j, 1 \leq i, j \leq 3$, $D_{hij}^2 \varphi \in V_{0h}$ and verifies

$$\begin{aligned} \int_{\Omega} D_{hij}^2 \varphi v d\mathbf{x} + C \sum_{K \in \mathcal{T}_h} |K|^{2/3} \int_K \nabla D_{hij}^2 \varphi \cdot \nabla v d\mathbf{x} \\ = -\frac{1}{2} \int_{\Omega} \left[\frac{\partial \varphi}{\partial x_i} \frac{\partial v}{\partial x_j} + \frac{\partial \varphi}{\partial x_j} \frac{\partial v}{\partial x_i} \right] d\mathbf{x}, \quad \forall v \in V_{0h}. \end{aligned} \quad (3.30)$$

The regularization is shown in numerical examples that it is essential for the algorithm to converge when we use an unstructured partition of Ω . Similar ideas have been used for approximations of the Stokes problem in incompressible fluid mechanics when using essentially the same finite element spaces for the velocity components and the pressure (see, e.g., [Glowinski, 2008, Chapter 5]).

Concerning issue ii), that is the solution of the bi-harmonic problems encountered in the conjugate gradient algorithm (18)-(27), after space discretization the resulting discrete bi-harmonic problems are all particular cases of

$$\text{Find } r \in V_{0h} \text{ such that } \int_{\Omega} \Delta_h r \Delta_h \varphi d\mathbf{x} = \Lambda_h(\varphi), \quad \forall \varphi \in V_{0h}, \quad (3.31)$$

with $\Lambda_h \in \mathcal{L}(V_{0h}, \mathbb{R})$ and $\Delta_h \in \mathcal{L}(V_{0h}, V_{0h})$ defined by

$$\Delta_h \varphi \in V_{0h}, \quad - \int_{\Omega} \Delta_h \varphi \theta d\mathbf{x} = \int_{\Omega} \nabla \varphi \cdot \nabla \theta d\mathbf{x} \quad \forall (\varphi, \theta) \in V_{0h} \times V_{0h},$$

It follows from this definition that the discrete bi-harmonic problem (3.31) is equivalent to the following system of discrete Poisson-Dirichlet problems:

$$\begin{aligned} \text{Find } \omega \in V_{0h}, \quad & \int_{\Omega} \nabla \omega \cdot \nabla \varphi d\mathbf{x} = \Lambda(\varphi), \quad \forall \varphi \in V_{0h}, \\ \text{Find } r \in V_{0h}, \quad & \int_{\Omega} \nabla r \cdot \nabla \varphi d\mathbf{x} = \int_{\Omega} \omega \varphi d\mathbf{x}, \quad \forall \varphi \in V_{0h}. \end{aligned}$$

3.5.3 Discrete Formulation of the Least-Squares Method

We define the discrete analogues of spaces $\mathbf{Q}$ and $\mathbf{Q}_f$ as follows:

$$\begin{aligned} \mathbf{Q}_h &= \{\mathbf{q}_h \in (V_{0h})^{3 \times 3}, \mathbf{q}_h(P_k) = \mathbf{q}_h^t(P_k), k = 1, \dots, N_{0h}\}, \\ \mathbf{Q}_{fh} &= \{\mathbf{q}_h \in \mathbf{Q}_h, \det \mathbf{q}_h(P_k) = f_h(P_k), \\ &\quad \mathbf{q}_h(P_k) \text{ is positive definite}, k = 1, \dots, N_{0h}\}. \end{aligned}$$

We associate with V_h (or V_{0h} and V_{gh}) and $\mathbf{Q}_h$, the discrete inner products with corresponding norm as:

$$(v, w)_{0h} = \frac{1}{4} \sum_{k=1}^{N_h} A_k v(P_k) w(P_k), \quad ||v||_{0h} = \sqrt{(v, v)_h} \quad \forall v, w \in V_{0h},$$

and

$$((\mathbf{S}, \mathbf{T}))_{0h} = \frac{1}{4} \sum_{k=1}^{N_h} A_k \mathbf{S}(P_k) : \mathbf{T}(P_k), \quad |||\mathbf{S}|||_{0h} = \sqrt{((\mathbf{S}, \mathbf{S}))_{0h}} \quad \forall \mathbf{S}, \mathbf{T} \in \mathbf{Q}_h,$$

where A_k is the volume of the polyhedral domain which is the union of those tetrahedra of $\mathcal{T}_h$ which have P_k as a common vertex.

The discrete nonlinear least-squares method is addressed as: find $(u_h, \mathbf{p}_h) \in V_{gh} \times \mathbf{Q}_{fh}$ such that

$$J_h(u_h, \mathbf{p}_h) \leq J_h(\varphi_h, \mathbf{q}_h), \quad \forall (\varphi_h, \mathbf{q}_h) \in V_{gh} \times \mathbf{Q}_{fh} \quad (3.32)$$

where:

$$J_h(\varphi_h, \mathbf{q}_h) = \frac{1}{2} |||\mathbf{D}_h^2 \varphi_h - \mathbf{q}_h|||_{0h}^2 \quad (3.33)$$

3.5.4 A Discrete Relaxation Algorithm

The discrete relaxation algorithm we employ reads as follows:

Step 1. The initialization is performed by solving:

$$u_h^0 \in V_{gh} \text{ such that } \int_{\Omega_h} \nabla u_h^0 \cdot \nabla \varphi_h d\mathbf{x} = -(3\sqrt[3]{f_h}, \varphi_h)_{0h}, \quad \forall \varphi_h \in V_{0h}.$$

Then, for $n \geq 0$

Step 2. Assuming that u_h^n is known, we look for

$$\mathbf{p}_h^n = \arg \min_{\mathbf{q}_h \in \mathbf{Q}_{fh}} J_h(u_h^n, \mathbf{q}_h); \quad (3.34)$$

Step 3. Assuming that $\mathbf{p}_h^n$ is known, we look for

$$u_h^{n+1/2} = \arg \min_{\varphi_h \in V_{gh}} J_h(\varphi_h, \mathbf{p}_h^n); \quad (3.35)$$

Step 4. Update the solution as

$$u_h^{n+1} = u_h^n + \omega(u_h^{n+1/2} - u_h^n), \quad (3.36)$$

with $1 \leq \omega \leq \omega_{\max} < 2$.

3.5.5 Finite Element Approximation of the Local Nonlinear Problems

The finite dimensional minimization problems, discrete analogues of (3.9), are approximated, at each grid point $P_k \in \Sigma_{0h}$, by:

$$\mathbf{p}_h^n(P_k) = \arg \min_{\mathbf{q} \in \mathbf{E}_f(P_k)} \left[\frac{1}{2} |\mathbf{q}|^2 - \mathbf{D}_h^2 u^n(P_k) : \mathbf{q} \right].$$

The methods discussed in Section 3.3 still apply.

3.5.6 Finite Element Approximation of the Linear Variational Problems

The variational problems arising in the discrete version of the relaxation algorithm can be solved similarly as in the continuous case using a conjugate gradient algorithm. Let us point out however a particularity that arises in the discrete case. The discrete version of (3.16) reads as follows: find $u_h^{n+1/2} \in V_{gh}$ satisfying:

$$((\mathbf{D}_h^2 u_h^{n+1/2}, \mathbf{D}_h^2 \varphi_h))_{0h} = ((\mathbf{p}_h^n, \mathbf{D}_h^2 \varphi_h))_{0h}, \quad \forall \varphi_h \in V_{0h}. \quad (3.37)$$

The linear problem (3.37) leads to excessive computer resource requirements, which could be acceptable for two-dimensional problems, but become prohibitive for three dimensional ones. (Indeed, to derive the linear system equivalent to (3.37), we need to compute-via the solution of (3.30)-the matrix-valued functions $\mathbf{D}_h^2 w^j$, where the functions w^j form a basis of V_{0h} .) To avoid this difficulty, we are going to employ, as previously discussed in [Caboussat et al., 2013], an adjoint equation approach (see, e.g., [Glowinski et al., 2008]) to derive an equivalent formulation of (3.37), well-suited to a solution by a conjugate gradient algorithm.

This adjoint approach reads as

$$\text{Find } u_h^{n+1/2} \in V_{gh} \text{ such that } \langle \frac{\partial J_h}{\partial \varphi}(u_h^{n+1/2}, \mathbf{p}_h^n), \theta_h \rangle = 0, \quad \forall \theta_h \in V_{0h}, \quad (3.38)$$

where $\langle \frac{\partial J_h}{\partial \varphi}(\varphi, \mathbf{q}), \theta \rangle$ denotes the action of the partial derivative $\frac{\partial J_h}{\partial \varphi}(\varphi, \mathbf{q})$ on the test function θ . In order to solve (3.38), we first determine $D_{hij}^2 \varphi$ via (3.30). Then, we find $\lambda_{hij} \in V_{0h}$, $1 \leq i, j \leq 3$, by solving the (adjoint) systems:

$$(\lambda_{hij}, \theta_h)_{0h} + C \sum_{K \in \mathcal{T}_h} |K|^{2/3} \int_K \nabla \lambda_{hij} \cdot \nabla \theta_h d\mathbf{x} = (\mathbf{p}_{hij} - D_{hij}^2 \varphi, \theta_h)_{0h}, \quad \forall \theta_h \in V_{0h},$$

and we can show (see, e.g., [Glowinski et al., 2008]) that, for all $(\varphi_h, \mathbf{p}_h) \in V_{gh} \times \mathbf{Q}_h$:

$$\langle \frac{\partial J_h}{\partial \varphi}(\varphi_h, \mathbf{p}_h), \theta_h \rangle = \int_{\Omega} \left[\sum_{i=1}^3 \sum_{j=1}^3 \frac{\partial \lambda_{hij}}{\partial x_i} \frac{\partial \theta_h}{\partial x_j} \right] d\mathbf{x}, \quad \forall \theta_h \in V_{0h}.$$

3.5.7 Discrete Conjugate Gradient Algorithm

The relations that are developed in (3.30), (3.31) and (3.38) can be used directly in the conjugate gradient algorithm (3.18)–(3.27).

Step 1.

$$u_h^{n+1/2,0} = u_h^n \in V_{gh}.$$

Step 2.

- Compute $D_{hij}^2 u_h^{n+1/2,0} \in V_{0h}$, $1 \leq i, j \leq 3$, by solving

$$\begin{aligned} \int_{\Omega} D_{hij}^2 u_h^{n+1/2,0} v d\mathbf{x} + C \sum_{K \in \mathcal{T}_h} |K|^{2/3} \int_K \nabla D_{hij}^2 u_h^{n+1/2,0} \cdot \nabla v d\mathbf{x} \\ = -\frac{1}{2} \int_{\Omega} \left[\frac{\partial u_h^{n+1/2,0}}{\partial x_i} \frac{\partial v}{\partial x_j} + \frac{\partial u_h^{n+1/2,0}}{\partial x_j} \frac{\partial v}{\partial x_i} \right] d\mathbf{x}, \quad \forall v \in V_{0h}. \end{aligned}$$

- Compute $\lambda_{hij} \in V_{0h}$, $1 \leq i, j \leq 3$, by solving

$$(\lambda_{hij}, \theta_h)_{0h} + C \sum_{K \in \mathcal{T}_h} |K|^{2/3} \int_K \nabla \lambda_{hij} \cdot \nabla \theta_h d\mathbf{x} = (\mathbf{p}_{hij} - D_{hij}^2 u_h^{n+1/2,0}, \theta_h)_{0h}, \quad \forall \theta_h \in V_{0h}.$$

- Find $g_h^0 \in V_{0h}$ satisfying

$$(\Delta g_h^0, \Delta \theta_h)_{0h} = \int_{\Omega} \left[\sum_{i=1}^3 \sum_{j=1}^3 \frac{\partial \lambda_{ij}^k}{\partial x_i} \frac{\partial \theta_h}{\partial x_j} \right] d\mathbf{x}, \quad \forall \theta_h \in V_{0h}.$$

- Set $w_h^0 = g_h^0$

Then, for $k \geq 0$, $u_h^{n+1/2,k}$, g_h^k and w_h^k being known, the last two different from zero, we compute $u_h^{n+1/2,k+1}$, g_h^{k+1} and, if necessary, w_h^{k+1} as follows.

Step 3.

- Compute $D_{hij}^2 w_h^k \in V_{0h}$, $1 \leq i, j \leq 3$, by solving

$$\begin{aligned} \int_{\Omega} D_{hij}^2 w_h^k v d\mathbf{x} + C \sum_{K \in \mathcal{T}_h} |K|^{2/3} \int_K \nabla D_{hij}^2 w_h^k \cdot \nabla v d\mathbf{x} \\ = -\frac{1}{2} \int_{\Omega} \left[\frac{\partial w_h^k}{\partial x_i} \frac{\partial v}{\partial x_j} + \frac{\partial w_h^k}{\partial x_j} \frac{\partial v}{\partial x_i} \right] d\mathbf{x}, \quad \forall v \in V_{0h}. \end{aligned}$$

- Compute $\bar{\lambda}_{hij} \in V_{0h}$, $1 \leq i, j \leq 3$, by solving

$$(\bar{\lambda}_{hij}, \theta_h)_{0h} + C \sum_{K \in \mathcal{T}_h} |K|^{2/3} \int_K \nabla \bar{\lambda}_{hij} \cdot \nabla \theta_h d\mathbf{x} = -(D_{hij}^2 w_h^k, \theta_h)_{0h}, \quad \forall \theta_h \in V_{0h}.$$

- Find $\bar{g}_h^k \in V_{0h}$ satisfying

$$(\Delta \bar{g}_h^k, \Delta \theta_h)_{0h} = \int_{\Omega} \left[\sum_{i=1}^3 \sum_{j=1}^3 \frac{\partial \bar{\lambda}_{ij}^k}{\partial x_i} \frac{\partial \theta_h}{\partial x_j} \right] d\mathbf{x}, \quad \forall \theta_h \in V_{0h}.$$

- Compute the steepest descent

$$\rho_k = \frac{\|\Delta_h g_h^k\|_{0h}^2}{(\Delta_h \bar{g}_h^k, \Delta_h w_h^k)_{0h}}.$$

- Update the solution

$$u_h^{n+1/2,k+1} = u_h^{n+1/2,k} - \rho_k w_h^k.$$

Step 4.

- Compute

$$g_h^{k+1} = g_h^k - \rho_k \bar{g}_h^k.$$

- Compute

$$\delta_k = \frac{\|\Delta_h g_h^{k+1}\|_{0h}^2}{\|\Delta_h g_h^0\|_{0h}^2}.$$

- Check the convergence

if $\delta_k < \epsilon$ then take $u_h^{n+1/2} = u_h^{n+1/2,k+1}$; otherwise compute

$$\gamma_k = \frac{\|\Delta_h g_h^{n+1/2,k+1}\|_{0h}^2}{\|\Delta_h g_h^{n+1/2,k}\|_{0h}^2},$$

and

$$w_h^{k+1} = g_h^{k+1} + \gamma_k w_h^k.$$

Step 5.

- Do $k \leftarrow k + 1$ and go to **Step 3**.

3.6 Numerical Results

The first numerical results we are going to report, in order to validate our methodology, are associated with the unit cube $\Omega = (0, 1)^3$. Two types of partitions of the unit cube are considered, in order to study the mesh-dependence of our methods. These partitions have been constructed by using either advancing front 3D procedures, or successive extrusions [Geuzaine and Remacle, 2009], and are visualized in Figure 3.1. All experiments were performed on a desktop computer with Intelcx Xeon(R) CPU E5-1650 v3 @ 3.50GHz $\times$ 12.

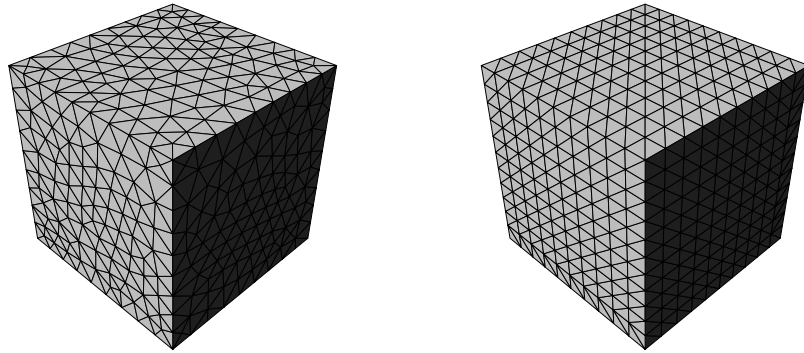


Figure 3.1 – Typical partitions of the unit cube $\Omega = (0, 1)^3$; left: isotropic mesh ($h \simeq 0.01826$); right: structured mesh ($h \simeq 0.00938$).

In the numerical examples presented hereafter, we consider $C = 0$ for the structured mesh and $C = 1$ for the isotropic mesh. The local nonlinear problems are solved with a stopping criterion of $\varepsilon_{\text{Newton}} = 10^{-9}$ on the residual for the Newton method, with a maximal number of iterations equal to 1000. When using the Runge-Kutta method for the dynamical flow approach, the time step is set to $\Delta t = 0.1$, and is reduced only if needed (only for the first 2-3 times steps usually); the maximal number of iterations is 20000 and the stopping criterion is $\varepsilon = 10^{-7}$ on two successive iterates.

Unless otherwise specified, the relaxation parameter is set to $\omega = 1$ at the beginning of the outer iterations, and gradually increased to 2. The conjugate gradient algorithm for the solution of the variational problems has a stopping criterion of $\varepsilon = 10^{-8}$ on successive iterates, with a maximal number of iterations equal to 100. Numerical experiments show that the number of conjugate gradient iterations is never larger than 35. The outer relaxation algorithm has a stopping criterion of $\varepsilon = 5 \times 10^{-4}$ on the residual $\| |\mathbf{D}_h^2 u_h^n - \mathbf{p}_h^n| |_{0h}$, or on successive iterates if the problem does not admit a classical solution (see Section 3.6.3), with a maximal number of iterations equal to 5000.

3.6.1 Polynomial Examples

Let us consider a first example involving a smooth exact solution. More precisely, let us consider as exact solution

$$u(x, y, z) = \frac{1}{2} (x^2 + 5y^2 + 15z^2), \quad (x, y, z) \in \Omega. \quad (3.39)$$

By a direct calculation, one obtains $\lambda_1 = 1, \lambda_2 = 5, \lambda_3 = 15$, and, therefore the data for the Monge-Ampère problem correspond to

$$f(x, y) = 75 \quad \text{and} \quad g(x, y, z) = \frac{1}{2} (x^2 + 5y^2 + 15z^2).$$

Figure 3.2 visualizes the $L^2(\Omega)$ and $H^1(\Omega)$ computed approximation errors obtained by using both approaches for the solution of the local nonlinear problems (Newton stands for the reduced Newton approach presented in Section 3.3.2, while RK stands for the dynamical flow approach presented in Section 3.3.3). Both Newton and Runge-Kutta algorithms provide exactly the same results. For the structured mesh, the method is globally second-order, resp. first-order, convergent for the L^2 (resp. H^1) norm of the approximation error. Table 3.1 confirms those convergence results, for both structured and isotropic meshes and for the approach based on the Runge-Kutta approximation for the dynamical flow problem in $\mathbb{R}^{11}$. We observe that, for the structured meshes, we have text-book second and first order convergence, while the orders of convergence deteriorate for the isotropic unstructured meshes.

Table 3.2 provides CPU times vs the number of degrees of freedom involved in the numerical approximation. Comparing mainly with [Brenner and Neilan, 2012; Liu et al., 2017] (who also

performed CPU times investigations), this test case is more stringent since the eigenvalues of the Hessian are not close from each other, which makes the problem less isotropic than examples used in the literature (typically the example presented in Section 3.6.2).

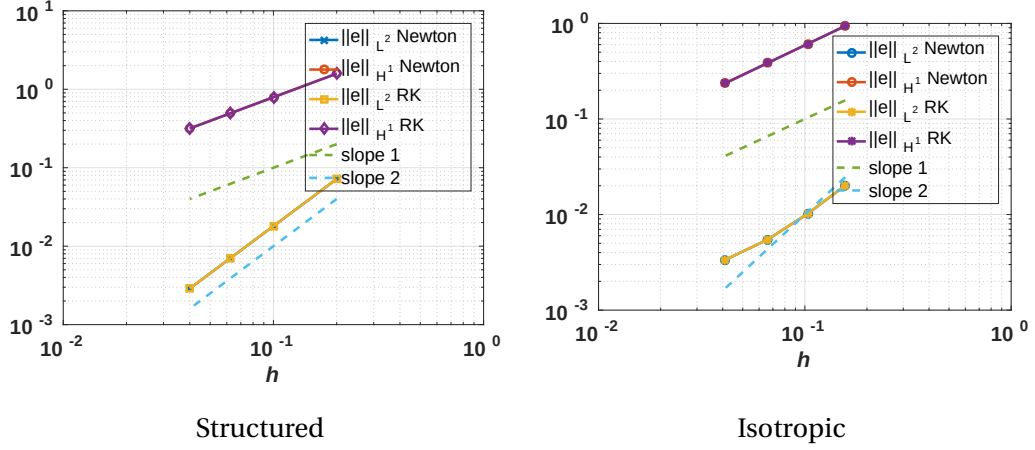


Figure 3.2 – Visualization of the variations with respect to h of the $L^2(\Omega)$ and $H^1(\Omega)$ norms of the computed approximation error $u_h - u$, with $u(x, y, z) = \frac{1}{2}(x^2 + 5y^2 + 15z^2)$ ($\Omega = (0, 1)^3$). Both Newton and Runge-Kutta algorithms provide exactly the same results.

Table 3.1 – (i) Variations with respect to h of the $L^2(\Omega)$ and $H^1(\Omega)$ norms of the computed approximation error $u_h - u$, with $u(x, y, z) = \frac{1}{2}(x^2 + 5y^2 + 15z^2)$ and related convergence orders. (ii) Variations with respect to h of the number of relaxation iterations necessary to achieve convergence. The local optimization problems are solved using the Runge-Kutta based method described in Section 3.3.3 ($\Omega = (0, 1)^3$).

Structured mesh					
h	$\ u_h - u\ _{L_2}$		$\ u_h - u\ _{H_1}$		iter
2.00e-01	7.19e-02	-	1.58e-00	-	71
1.00e-01	1.80e-02	1.99	7.91e-01	0.99	228
6.25e-02	7.06e-03	1.99	4.95e-01	1.00	314
4.00e-02	2.89e-03	1.99	3.16e-01	0.99	375
Isotropic unstructured mesh					
h	$\ u_h - u\ _{L_2}$		$\ u_h - u\ _{H_1}$		iter
1.57e-01	1.97e-02	-	9.39e-01	-	84
1.03e-01	1.01e-02	1.75	6.11e-01	1.03	137
6.58e-02	5.44e-03	1.63	3.84e-01	1.03	220
4.10e-02	3.35e-03	1.38	2.35e-01	1.03	314

The second test problem that we consider still has a polynomial exact solution, but this

Chapter 3. Numerical Approximation of Monge-Ampère Equation

Table 3.2 – CPU time results and numbers of degrees of freedom for the smooth test case with $u(x, y, z) = \frac{1}{2}(x^2 + 5y^2 + 15z^2)$ (the number of DOFs specified corresponds to the number of vertices of the finite element mesh). The reduced Newton method is used for the solution of the local nonlinear problems.

Structured mesh						
h	#DOFs	Algebraic solver [s]	Variational solver [s]	# outer iter.	Max # CG iter.	Total CPU [s]
0.2000	216	0	16	71	12	16
0.1000	1331	3	655	228	19	658
0.0625	4913	26	6070	314	16	6096
0.0400	17576	163	38993	375	15	39156

solution is much more anisotropic than the one in (3.39), since it is given by

$$u(x, y, z) = \frac{1}{2}(x^2 + 10y^2 + 100z^2), (x, y, z) \in \Omega. \quad (3.40)$$

Here $\lambda_1 = 1, \lambda_2 = 10, \lambda_3 = 100$, and, the data for the Monge-Ampère problem are given by $f(x, y) = 1000$ and $g(x, y, z) = \frac{1}{2}(x^2 + 10y^2 + 100z^2)$. This time, the initialisation (3.4) of the relaxation algorithm is not close to the solution, implying, as expected, that more iterations are needed to achieve convergence. Figure 3.3 illustrates the convergence orders for the computed approximation error for both types of meshes. Despite the anisotropy of the solution of this second test problem, the approximation errors are similar to those associated with the first test problem, that is perfect second and first orders with the structured meshes and slightly lower convergence orders for the anisotropic unstructured meshes. Note that here we have to choose $\omega = 0.5$ initially (under-relaxation), and increase it gradually to 2, to ensure convergence of the relaxation algorithm, and $C = 2.5$ for the isotropic mesh.

3.6.2 A Smooth Exponential Example

The third test problem we consider has a smooth exponential exact solution, namely the radial function u defined by

$$u(x, y, z) = e^{\frac{1}{2}(x^2 + y^2 + z^2)}, (x, y, z) \in \Omega. \quad (3.41)$$

This test problem generalizes to three dimensions a two-dimensional one commonly used in the community for Monge-Ampère solver benchmarking (see, e.g., [Caboussat et al., 2013; Feng and Neilan, 2009c]). Let us denote $\sqrt{x^2 + y^2 + z^2}$ by r . The data for the Monge-Ampère-Dirichlet problem (3.1) associated with the above function u are $f(x, y, z) = (1 + r^2)e^{3r^2/2}$, and $g(x, y, z) = e^{r^2/2}$. The stopping criterion for the relaxation algorithm is $|||\mathbf{D}_h^2 u_h^n - \mathbf{p}_h^n|||_{0h} <$

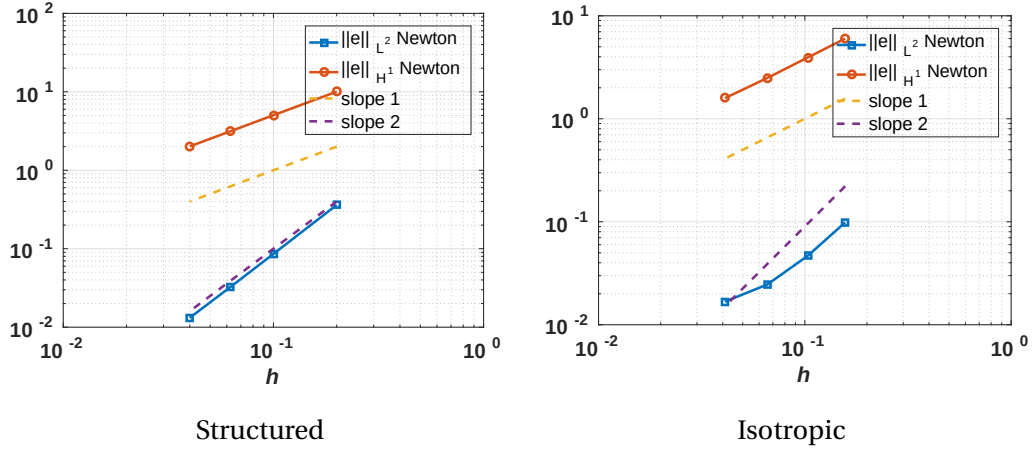


Figure 3.3 – Visualization of the variations with respect to h of the $L^2(\Omega)$ and $H^1(\Omega)$ norms of the computed approximation error $u_h - u$, with $u(x, y, z) = \frac{1}{2}(x^2 + 10y^2 + 100z^2)$ ($\Omega = (0, 1)^3$).

5×10^{-4} , and $C = 1$ for the isotropic unstructured mesh (as for the first test problem).

Figure 3.4 visualizes the $L^2(\Omega)$ and $H^1(\Omega)$ computed approximation errors for both approaches for the solution of the local nonlinear problems. The conclusions are similar: both Newton and Runge-Kutta methods provide exactly the same results, and the method is globally second-order convergent for the L^2 norm. Table 3.3 confirms these convergence results, showing in particular no loss of convergence orders for the unstructured isotropic mesh.

Table 3.4 provides CPU times vs the number of degrees of freedom involved in the numerical approximation for both structured and unstructured discretizations of the unit cube. When using a structured mesh of the unit cube, the performance of the algorithm is comparable to the other algorithms from the literature, albeit slightly less efficient. Using an unstructured isotropic mesh degrades the performance of the algorithm; computational performance for the algebraic part is identical, the difference coming from the increased number of conjugate gradient iterations. Note that, for this test case, the Hessian matrix $\mathbf{D}^2 u$ admits the eigenvalues $\lambda_1 = e^{r^2/2}$, $\lambda_2 = e^{r^2/2}$ and $\lambda_3 = (1 + r^2)e^{r^2/2}$. This example is thus rather isotropic (the eigenvalues of the Hessian are close to each other).

3.6.3 Non-Smooth Test Problems

Some of the test problems we are going to consider in this section do not have exact solution with the $H^2(\Omega)$ -regularity or may have no solution at all (but may have generalized solutions). These non-smooth problems are therefore suited to test the robustness of our methodology, and its ability at capturing generalized solutions when no exact solution does exist.

With Ω still being the unit cube $(0, 1)^3$, the first problem that we consider is the particular case

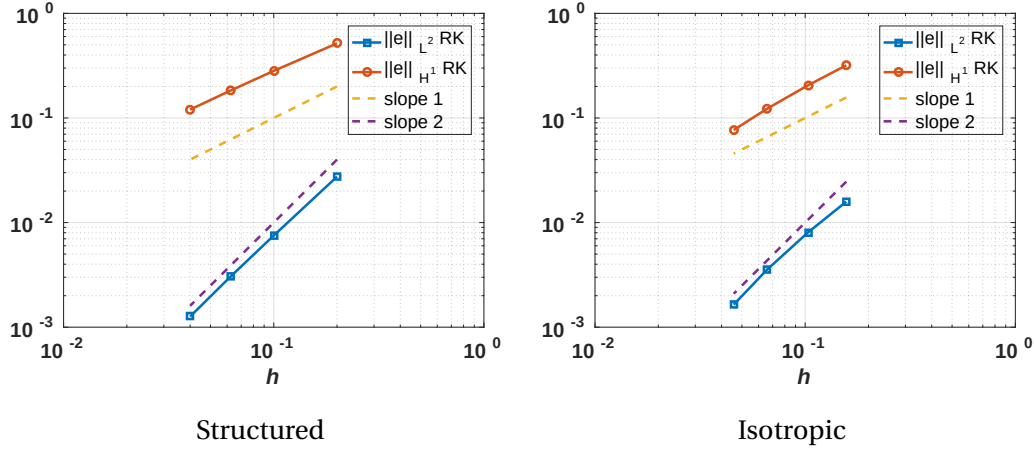


Figure 3.4 – Visualization of the variations with respect to h of the $L^2(\Omega)$ and $H^1(\Omega)$ norms of the computed approximation error $u_h - u$, with $u(x, y, z) = e^{\frac{1}{2}(x^2+y^2+z^2)}$ ($\Omega = (0, 1)^3$).

Table 3.3 – (i) Variations with respect to h of the $L^2(\Omega)$ and $H^1(\Omega)$ norms of the computed approximation error $u_h - u$, with $u(x, y, z) = e^{\frac{1}{2}(x^2+y^2+z^2)}$ and related convergence orders. (ii) Variations with respect to h of the number of relaxation iterations necessary to achieve convergence. The local optimization problems are solved using the Runge-Kutta based method described in Section 3.3.3 ($\Omega = (0, 1)^3$).

Structured mesh					
h	$\ u_h - u\ _{L_2}$		$\ u_h - u\ _{H_1}$		iter
2.00e-01	2.74e-02	-	5.16e-01	-	12
1.00e-01	7.52e-03	1.87	2.81e-01	0.87	20
6.25e-02	3.06e-03	1.91	1.83e-01	0.91	25
4.00e-02	1.26e-03	1.98	1.20e-01	0.95	28
Isotropic mesh					
h	$\ u_h - u\ _{L_2}$		$\ u_h - u\ _{H_1}$		iter
1.57e-01	1.58e-02	-	3.19e-01	-	24
1.03e-01	8.07e-03	1.61	2.05e-01	1.05	34
6.58e-02	3.54e-03	1.83	1.22e-01	1.15	41
4.57e-02	1.64e-03	2.11	7.67e-02	1.28	42

of problem (3.1) which has the convex function u defined, for $R \geq \sqrt{3}$, by

$$u(x, y, z) = -\sqrt{R^2 - (x^2 + y^2 + z^2)}, \quad \forall (x, y, z) \in \Omega.$$

When $R > \sqrt{3}$, this function u belongs to $C^\infty(\bar{\Omega})$, while $u \in C^0(\bar{\Omega}) \cap W^{1,s}(\Omega)$, with $1 \leq s < 2$, if $R = \sqrt{3}$. It is therefore interesting to see how our methodology can handle the possi-

Table 3.4 – CPU time results and numbers of degrees of freedom for the smooth test case with $u(x, y, z) = e^{\frac{1}{2}(x^2+y^2+z^2)}$ (the number of DOFs specified corresponds to the number of vertices of the finite element mesh). The reduced Newton method is used for the solution of the local nonlinear problems.

h	#DOFs	Algebraic solver [s]	Variational solver [s]	# outer iter.	Max # CG iter.	Total CPU [s]
Structured mesh						
0.2000	216	0	1	12	5	1
0.1000	1331	0	19	20	5	19
0.0625	4913	1	90	25	4	92
0.0400	17576	8	423	28	4	431
Isotropic unstructured mesh						
0.1570	1043	1	70	24	19	71
0.1030	3339	1	525	34	20	526
0.0658	12191	5	3568	41	22	3573
0.0457	42176	24	20926	42	22	20950

ble non-smoothness of the particular problem (3.1) associated with f and g defined by $f(x, y, z) = \frac{R^2}{(R^2 - r^2)^{5/2}}$ and $g(x, y, z) = -\sqrt{R^2 - r^2}$, with, as earlier, $r = \sqrt{x^2 + y^2 + z^2}$. Of particular interest will be the behavior of our methodology when $R \rightarrow \sqrt{3}$ from above (or even when $R = \sqrt{3}$).

On Tables 3.5 and 3.6, we have reported, for $R = \sqrt{6}$ and $R = \sqrt{3}$, computed approximation errors and orders of convergence as h varies, together with the number of iterations necessary to achieve convergence of the relaxation algorithm. For $R = \sqrt{6}$, the convergence orders of the approximation errors are the ones we expect, namely second order (resp., first order) for the L^2 -norm (resp., H^1 -norm), the number of iterations being pretty low. The case $R = \sqrt{3}$ is more challenging; indeed despite the solution singularity at point $(1, 1, 1)$, the $L^2(\Omega)$ -norm of the computed approximation error $u_h - u$ still decreases super-linearly with respect to h (for both mesh families), while the related $H^1(\Omega)$ -norm stays stable around 0.62 for the same values of h and shows no convergence order.

To conclude this section, we will consider the particular problem (3.1) associated with $\Omega = (0, 1)^3$, $f = 1$ and $g = 0$. For these particular data, problem (3.1) has no smooth solution (the arguments developed in [Caboussat et al., 2013; Glowinski, 2008] for the related two-dimensional problem still apply here).

Figure 3.5 shows different features of the approximated solution inside the unit cube. The stopping criterion for this particular case without a classical solution is $\|u_h^{n+1} - u_h^n\|_{0,h} < 10^{-5}$. When studying the number of outer iterations of the relaxation algorithm, we observe that the number of iterations is larger for structured meshes than isotropic ones, and that it increases

Chapter 3. Numerical Approximation of Monge-Ampère Equation

Table 3.5 – (i) Variations with respect to h of the $L^2(\Omega)$ and $H^1(\Omega)$ norms of the computed approximation error $u_h - u$, with $u(x, y, z) = -\sqrt{R^2 - (x^2 + y^2 + z^2)}$ ($R = \sqrt{6}$) and related convergence orders. (ii) Variations with respect to h of the number of relaxation iterations necessary to achieve convergence. The local optimization problems are solved using the Runge-Kutta based method described in Section 3.3.3 ($\Omega = (0, 1)^3$).

Structured mesh					
h	$\ u_h - u\ _{L_2}$		$ u_h - u _{H_1}$		iter
2.00e-01	4.96e-03	-	8.60e-02	-	4
1.00e-01	1.28e-03	1.95	4.41e-02	0.96	5
6.25e-02	5.09e-04	1.96	2.78e-02	0.97	6
4.00e-02	2.10e-04	1.97	1.79e-02	0.98	7
Isotropic unstructured mesh					
h	$\ u_h - u\ _{L_2}$		$ u_h - u _{H_1}$		iter
1.57e-01	3.81e-03	-	8.60e-02	-	13
1.03e-01	1.81e-03	1.78	3.62e-02	2.07	16
6.58e-02	7.51e-04	1.94	2.12e-02	1.18	19
4.10e-02	3.35e-04	2.21	1.30e-02	1.33	19

Table 3.6 – (i) Variations with respect to h of the $L^2(\Omega)$ and $H^1(\Omega)$ norms of the computed approximation error $u_h - u$, with $u(x, y, z) = -\sqrt{R^2 - (x^2 + y^2 + z^2)}$ ($R = \sqrt{3}$) and related convergence orders. (ii) Variations with respect to h of the number of relaxation iterations necessary to achieve convergence. The local optimization problems are solved using the Runge-Kutta based method described in Section 3.3.3 ($\Omega = (0, 1)^3$).

Structured mesh					
h	$\ u_h - u\ _{L_2}$		$ u_h - u _{H_1}$		iter
2.00e-01	1.15e-02	-	6.60e-01	-	9
1.00e-01	3.06e-03	1.91	6.31e-01	-	14
6.25e-02	1.24e-03	1.92	6.25e-01	-	17
4.00e-02	5.17e-04	1.96	6.22e-01	-	19
Isotropic unstructured mesh					
h	$\ u_h - u\ _{L_2}$		$ u_h - u _{H_1}$		iter
1.57e-01	6.76e-03	-	6.31e-01	-	13
1.03e-01	3.31e-03	1.69	6.25e-01	-	16
6.58e-02	1.39e-03	1.93	6.22e-01	-	19
4.10e-02	6.41e-04	1.63	6.21e-01	-	19

as expected when $h \rightarrow 0$. Figure 3.5 (bottom row) visualizes graphs of the computed solutions restricted to the lines $y = z = 1/2$ and $x = y, z = 1/2$ for $x \in (0, 1)$, and shows little influence of the type of partition on the solution.

We can also observe that $\mathbf{D}^2 u$ is symmetric positive definite for 100% of the grid points, independently of the nature of the discretization when $h \simeq 0.04$, even though the Monge-Ampère equations does not have a classical solution, that is $\mathbf{D}^2 u \neq \mathbf{p}$. The (necessary) loss of convexity of the solution is thus located (near the corners) in a region smaller than the mesh size. When arbitrarily refining the mesh in a corner of the domain, we observe that the Hessian $\mathbf{D}^2 u$ is not symmetric positive definite when evaluated in some grid points in a neighborhood of size 10^{-3} around that corner. This effect is highlighted when calculating $\|\mathbf{D}^2 u_h - \mathbf{p}_h\|_{L^2}$, using a structured mesh of the unit cube, both on Ω , but also on $\Omega' \subset \Omega$, as illustrated in Table 3.7 for $\Omega' = (0.2, 0.8)^3$. These results show that the error inside the domain $\Omega' = (0.2, 0.8)^3$ is significantly smaller than the error on Ω , implying that the error is mainly committed near the boundary.

Table 3.7 – (i) Variations with respect to h of the norm of the residuals $\|\mathbf{D}^2 u_h - \mathbf{p}_h\|_{L^2(\Omega)}$ and $\|\mathbf{D}^2 u_h - \mathbf{p}_h\|_{L^2(\Omega')}$ when $\Omega = (0, 1)^3$, $\Omega' = (0.2, 0.8)^3$, $f = 1$ and $g = 0$. (ii) Variations with respect to h of the number of relaxation iterations necessary to achieve convergence.

h	$\ \mathbf{D}^2 u_h - \mathbf{p}_h\ _{L^2((0,1)^3)}$	$\ \mathbf{D}^2 u_h - \mathbf{p}_h\ _{L^2((0.2,0.8)^3)}$	# iter.
1.00e-01	4.29931e-04	4.20694e-05	467
6.25e-02	4.32211e-04	8.47222e-06	1857
4.00e-02	4.32995e-04	2.42009e-06	37522

3.6.4 Curved Boundaries and Non Convex Domains

In order to further validate the robustness and flexibility of our methodology, we are going to consider test problems where Ω has a curved boundary and/or is non-convex. The first domain with a curved boundary we consider is the unit ball $B_1 = \{(x, y, z) \in \mathbb{R}^3, x^2 + y^2 + z^2 < 1\}$. Assuming that $\Omega = B_1$, $f = \frac{1}{3\sqrt{3}}$ and $g = 0$, the convex solution u of the related Monge-Ampère-Dirichlet problem (3.1) is given by

$$u(x, y, z) = -\frac{1}{2\sqrt{3}} (1 - x^2 - y^2 - z^2). \quad (3.42)$$

On Figure 3.6 (left) we have visualized a typical finite element mesh used for computation and some cuts of the computed solution. On Figure 3.6 (right) and Table 3.8 we have provided information on the $L^2(\Omega)$ and $H^1(\Omega)$ -norms of the approximation error $u_h - u$ and of the related rates of convergence, and on the number of relaxation iterations necessary to achieve convergence. Albeit the $L^2(\Omega)$ -approximation error is $\mathcal{O}(h^{1.8})$, approximately, these numerical results show that our methodology can handle rather accurately domains Ω with curved boundaries. Table 3.9 provides CPU times vs the number of degrees of freedom involved in the numerical approximation of the solution on the unit sphere. Results are comparable to those obtained when using unstructured meshes on the unit cube, and thus show that the curved boundaries are handled appropriately.

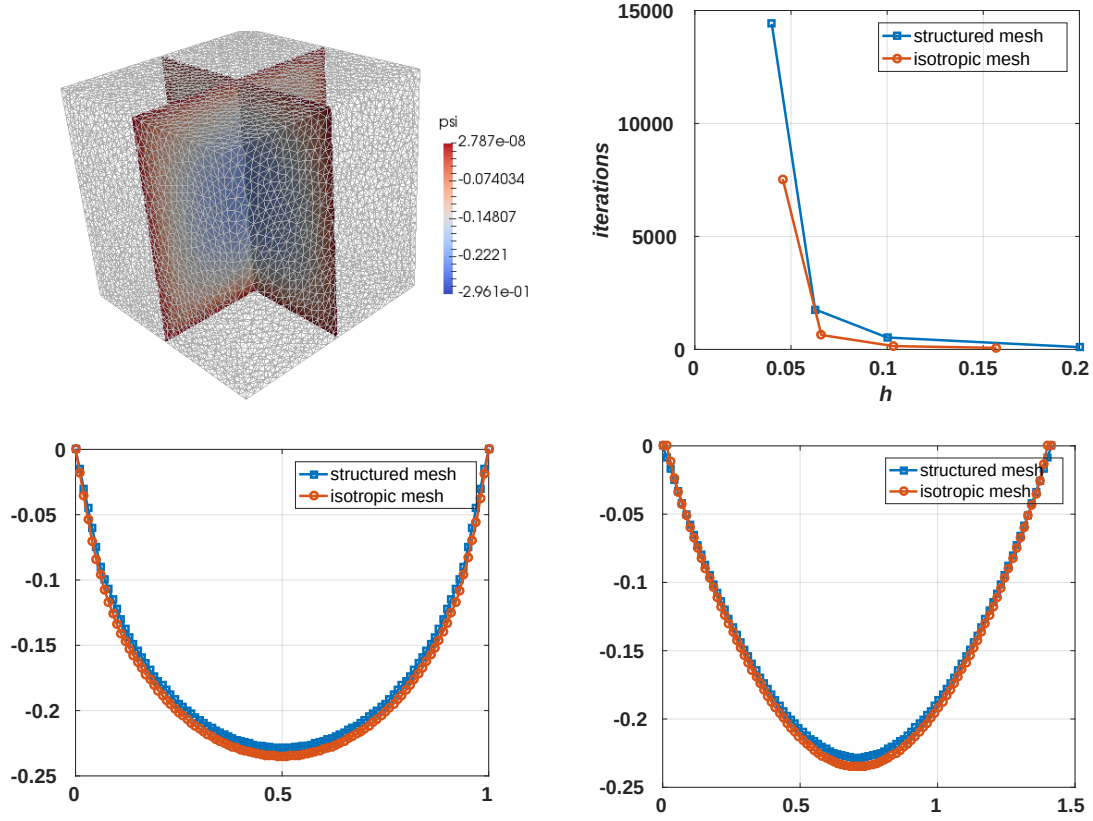


Figure 3.5 – Visualization of the numerical solution u_h for $f(x, y, z) = 1$, $g(x, y, z) = 0$ on the unit cube; top left: along cuts for $x = 1/2$ and $y = 1/2$ ($h \approx 0.0625$); top right: number of iterations needed for the convergence of the relaxation method for the stopping criterion $\|u_h^{n+1} - u_h^n\|_{0,h} < 10^{-5}$; bottom left: graphs of the computed solutions restricted to the line $y = z = 1/2$; bottom right: graphs of the computed solutions restricted to the lines $x = y$, $z = 1/2$.

The non-convexity of Ω may prevent problem (3.1) to have solutions (see, e.g., [Caffarelli and Cabré, 1995]). However, it makes sense to assess the capabilities of our methodology at handling problems having smooth solutions despite the non-convexity of Ω . To do so, we consider the particular problem (3.1) where: (i) Ω is the subset of B_1 obtained by removing from this ball a part of angular size θ , symmetric about Ox and oriented along the Oz axis (as shown on Figure 3.7 for $\theta = \pi/2$ and $\theta = \pi/9$), (ii) $f = 1/(3\sqrt{3})$, g being the restriction to $\partial\Omega$ of the function u defined by (3.42). The function u defined by (3.42) is clearly a convex solution

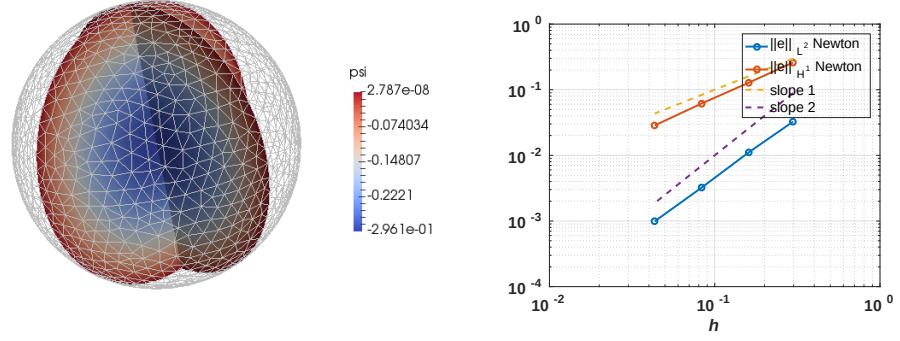


Figure 3.6 – Left: Visualization of the finite element mesh and of computed solution cuts ($h \approx 0.1610$) Right: Visualization of the variations with respect to h of the $L^2(\Omega)$ and $H^1(\Omega)$ norms of the computed approximation error $u_h - u$, with $u(x, y, z) = -\frac{1}{2\sqrt{3}} (1 - x^2 - y^2 - z^2)$ ($\Omega = B_1$).

Table 3.8 – (i) Variations with respect to h of the $L^2(\Omega)$ and $H^1(\Omega)$ norms of the computed approximation error $u_h - u$, with $u(x, y, z) = -\frac{1}{2\sqrt{3}} (1 - x^2 - y^2 - z^2)$ and related convergence orders. (ii) Variations with respect to h of the number of relaxation iterations necessary to achieve convergence. The local optimization problems are solved using the reduced Newton method described in Section 3.3.2 ($\Omega = B_1$).

h	$\ u_h - u\ _{L_2}$		$\ u_h - u\ _{H_1}$		iter
2.98e-01	3.26e-02		2.60e-01	-	14
1.61e-01	1.11e-02	1.74	1.28e-01	1.14	19
8.32e-02	3.22e-03	1.88	6.16e-02	1.11	21
4.34e-02	9.89e-04	1.80	2.86e-02	1.17	20

Table 3.9 – CPU time results and numbers of degrees of freedom for the smooth test case with $u(x, y, z) = -\frac{1}{2\sqrt{3}} (1 - x^2 - y^2 - z^2)$ on the unit sphere (the number of DOFs specified corresponds to the number of vertices of the finite element mesh). The reduced Newton method is used for the solution of the local nonlinear problems.

Structured mesh						
h	#DOFs	Algebraic solver [s]	Variational solver [s]	# outer iter.	Max # CG iter.	Total CPU [s]
0.2980	631	0	34	14	25	34
0.1610	3570	0	327	19	19	327
0.0832	22640	4	4385	21	22	4399
0.0434	184034	23	66027	20	23	66050

of the above problem (3.1). The solution methodology discussed in Sections 3.2 to 3.5 still applies for this case where an exact smooth solution does exist, some of the numerical results we obtained being reported in Figure 3.7. We observe in particular that the convergence orders are essentially independent of the value of the re-entrant angle θ .

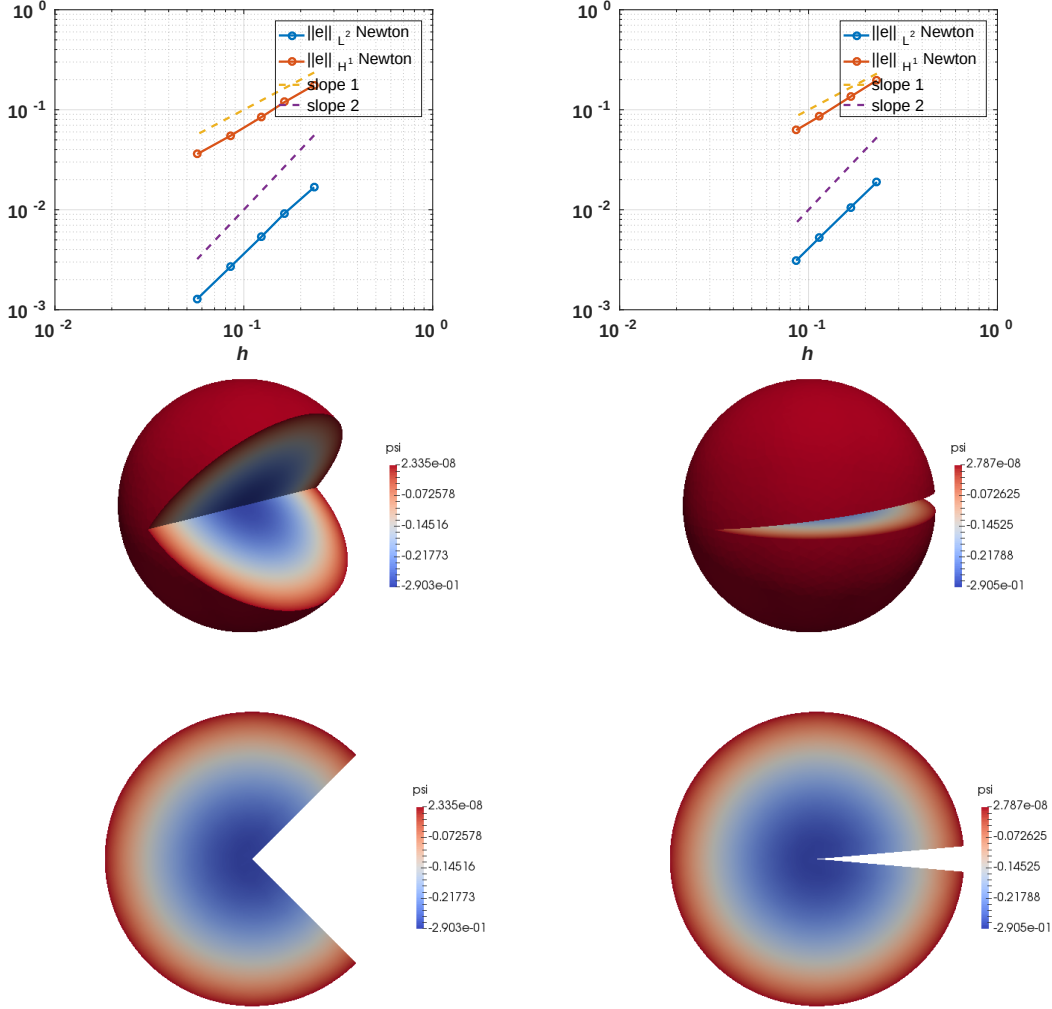


Figure 3.7 – First row: Visualization of the variations with respect to h of the $L^2(\Omega)$ and $H^1(\Omega)$ norms of the computed approximation error $u_h - u$, with u defined by (3.42), Ω being the truncated unit ball (left: $\theta = \pi/2$, right: $\theta = \pi/9$). Second row: Visualization of the truncated balls (left: $\theta = \pi/2$, right: $\theta = \pi/9$). Third row: Visualization of the restrictions of the computed solutions to the plane $z = 0$ (left: $\theta = \pi/2$, right: $\theta = \pi/9$).

3.7 An Alternative Discretization Method based on $\mathbb{Q}_1$ Finite Elements

We finally report some numerical results that were obtained using $\mathbb{Q}_1$ finite elements for the space discretization instead of the $\mathbb{P}_1$ finite elements used earlier. The finite element library `libmesh` [Kirk et al., 2006] has been used for implementation. The discretization of the unit cube $\Omega = (0, 1)^3$ is based on a structured mesh of elementary cubes, as visualized in Figure 3.8. The least-squares/relaxation methodology is still applicable. The nonlinear problems (3.9) are solved for each vertex of the hexahedral mesh with the Newton and Runge-Kutta methods discussed in Sections 3.3.2 and 3.3.3, respectively. The variational problem (3.17) is solved by a conjugate gradient algorithm, using Gauss quadrature rules (of order up to 4) for the numerical computation of integrals; all other techniques and approaches remain the same.

All the numerical results reported below are related to $\Omega = (0, 1)^3$. On Table 3.10 we have reported the variations with respect to h of the $L^2(\Omega)$ and $H^1(\Omega)$ norms of the computed approximation error $u_h - u$, for u defined by $u(x, y, z) = e^{\frac{1}{2}(x^2+y^2+z^2)}$ and $u(x, y, z) = \frac{1}{2}(x^2 + 5y^2 + 15z^2)$, the related convergence orders, and the number of relaxation iterations necessary to achieve convergence. The local optimization problems are solved using the Newton method described in Section 3.3.2. As expected, nearly optimal orders of convergence are obtained for both the $L^2(\Omega)$ and $H^1(\Omega)$ norms of the computed approximation error. Both solutions exhibit comparable orders of convergence, however, the larger anisotropy of the second one implies a larger number of iterations for the relaxation algorithm to achieve its convergence. Approximation errors and iteration numbers are consistent with those reported in Section 3.6.2 for the same test problems.

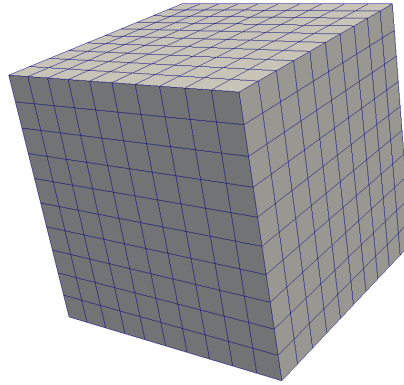


Figure 3.8 – A uniform structured hexahedral partition of the unit cube $\Omega = (0, 1)^3$ ($h = 0.1$).

The next test problem we consider, is the one, already investigated in Section 3.6.3, whose exact solution u is given by $u(x, y, z) = -\sqrt{R^2 - (x^2 + y^2 + z^2)}$, with $R \geq \sqrt{3}$, Ω still being the

Chapter 3. Numerical Approximation of Monge-Ampère Equation

Table 3.10 – (i) Variations with respect to h of the $L^2(\Omega)$ and $H^1(\Omega)$ norms of the computed approximation error $u_h - u$, with $u(x, y, z) = e^{\frac{1}{2}(x^2+y^2+z^2)}$ and $u(x, y, z) = \frac{1}{2}(x^2 + 5y^2 + 15z^2)$ and related convergence orders. (ii) Variations with respect to h of the number of relaxation iterations necessary to achieve convergence. The space approximation relies on $\mathbb{Q}_1$ based finite element spaces while the local optimization problems are solved using the Newton method described in Section 3.3.2 ($\Omega = (0, 1)^3$).

Exact solution $u(x, y, z) = e^{\frac{1}{2}(x^2+y^2+z^2)}$					
h	$\ u_h - u\ _{L_2}$		$ u_h - u _{H_1}$		iter
1/10	8.09e-03	-	1.18e-01	-	56
1/20	2.28e-03	1.82	5.67e-02	1.06	50
1/30	1.05e-03	1.90	3.71e-02	1.04	46
1/40	3.90e-04	1.93	2.75e-02	1.03	44
1/50	6.02e-04	1.95	2.18e-02	1.03	42

Exact solution $u(x, y, z) = \frac{1}{2}(x^2 + 5y^2 + 15z^2)$					
h	$\ u_h - u\ _{L_2}$		$ u_h - u _{H_1}$		iter
1/10	1.26e-02	-	1.65e-01	-	713
1/20	3.62e-03	1.79	7.88e-02	1.06	716
1/30	1.71e-03	1.85	5.15e-02	1.05	696
1/40	9.91e-04	1.88	3.82e-02	1.03	681
1/50	6.48e-04	1.90	3.03e-02	1.03	671

unit cube $(0, 1)^3$. Tables 3.11 and 3.12 show that, for $R = \sqrt{6}$ and $R = \sqrt{3}$, the $L^2(\Omega)$ and $H^1(\Omega)$ -norms of the approximation error $u_h - u$ are nearly of optimal order; moreover, the above tables show that $\|\mathbf{D}_h^2 u_h^n - \mathbf{D}^2 u\|_{L^2(\Omega)} \simeq \mathcal{O}(h^{3/2})$ if $R = \sqrt{6}$, while $\|\mathbf{D}_h^2 u_h^n - \mathbf{D}^2 u\|_{L^2(\Omega)} \simeq \mathcal{O}(1)$ if $R = \sqrt{3}$.

Table 3.11 – (i) Variations with respect to h of the $L^2(\Omega)$ and $H^1(\Omega)$ norms of the computed approximation error $u_h - u$, with $u(x, y, z) = -\sqrt{R^2 - (x^2 + y^2 + z^2)}$ with $R = \sqrt{6}$; related convergence orders. (ii) Variations with respect to h of the number of relaxation iterations necessary to achieve convergence. The space approximation relies on $\mathbb{Q}_1$ based finite element spaces while the local optimization problems are solved using the Newton method described in Section 3.3.2 ($\Omega = (0, 1)^3$).

h	$\ u_h - u\ _{L_2}$		$ u_h - u _{H_1}$		$\ \mathbf{D}_h^2 u_h^n - \mathbf{D}^2 u\ _{L_2}$		iter
1/10	1.63e-03		2.24e-02	-	8.18e-03	-	22
1/20	4.49e-04	1.85	1.06e-02	1.07	2.95e-03	1.47	18
1/30	2.06e-04	1.91	6.94e-03	1.05	1.62e-03	1.48	16
1/40	1.18e-04	1.94	5.14e-03	1.03	1.05e-03	1.48	15
1/50	7.65e-05	1.94	4.09e-03	1.03	7.55e-04	1.49	15

The numerical results we have just reported show that, as long as accuracy and number of

3.7. An Alternative Discretization Method based on $\mathbb{Q}_1$ Finite Elements

Table 3.12 – (i) Variations with respect to h of the $L^2(\Omega)$ and $H^1(\Omega)$ norms of the computed approximation error $u_h - u$, with $u(x, y, z) = -\sqrt{R^2 - (x^2 + y^2 + z^2)}$ with $R = \sqrt{3}$; related convergence orders. (ii) Variations with respect to h of the number of relaxation iterations necessary to achieve convergence. The space approximation relies on $\mathbb{Q}_1$ based finite element spaces while the local optimization problems are solved using the Newton method described in Section 3.3.2 ($\Omega = (0, 1)^3$).

h	$\ u_h - u\ _{L_2}$		$\ u_h - u\ _{H_1}$		$\ \mathbf{D}_h^2 u_h^n - \mathbf{D}^2 u\ _{L_2}$	iter
1/10	3.06e-03		4.70e-02	-	2.71e-01	35
1/20	8.66e-04	1.82	2.30e-02	1.02	2.59e-01	34
1/30	4.00e-04	1.90	1.53e-02	1.01	2.55e-01	31
1/40	2.29e-04	1.93	1.14e-02	1.00	2.55e-01	30

iterations are concerned, $\mathbb{Q}_1$ based finite element approximations of problem (3.1) compared well with $\mathbb{P}_1$ based ones if Ω is a cube and uniform structured partitions of Ω are used to define the finite element spaces. However the $\mathbb{P}_1$ based methods can easily handle domains Ω of arbitrary shapes and unstructured finite element partitions, properties that the $\mathbb{Q}_1$ based methods do not share.

4 Numerical Approximation of a 2D Parabolic Monge-Ampère Equation

4.1 Model problem

Let Ω be a smooth bounded convex domain of $\mathbb{R}^2$, and $T > 0$ a fixed time horizon. We consider a time evolutive two-dimensional Monge-Ampère equation, with Dirichlet boundary conditions, which reads as follows: find $u : \Omega \times (0, T) \rightarrow \mathbb{R}$ satisfying

$$\begin{cases} \frac{\partial u}{\partial t} - \det \mathbf{D}^2 u = f & \text{in } \Omega \times (0, T), \\ u = g & \text{in } \partial\Omega \times (0, T), \\ u(0) = u_0 & \text{in } \Omega. \end{cases} \quad (4.1)$$

In (4.1), $f = f(\mathbf{x}, t)$, $g = g(\mathbf{x}, t)$ and $u_0 = u_0(\mathbf{x})$ are given functions with the required regularity, and $\mathbf{D}^2 u (:= \mathbf{D}_{\mathbf{x}}^2 u)$ is the Hessian of the unknown function u (with respect to the space variable $\mathbf{x}$), defined by $\mathbf{D}^2 u = (D_{ij}^2 u)_{1 \leq i, j \leq 2}$, and $D_{ij}^2 u = \frac{\partial^2 u}{\partial x_i \partial x_j}$.

We assume in the sequel that u_0 is convex. (Note that a constraint on the time step may have to be enforced to make sure that the numerical solution remains convex at all times). The right-hand side f may change sign when u depends on time, otherwise f should be negative. Moreover, if u does not depend on time, equation (4.1) reads as: find $u : \Omega \rightarrow \mathbb{R}$ satisfying

$$\begin{cases} -\det \mathbf{D}^2 u = f \leq 0 & \text{in } \Omega, \\ u = g & \text{in } \partial\Omega, \end{cases} \quad (4.2)$$

where it is the steady state Monge-Ampère equation. In numerical experiments we will tackle separately the cases of solutions that depend on time from those that are time independent.

It can be shown that the Monge-Ampère operator can be rewritten under a divergence form,

as follows,

$$\det \mathbf{D}^2 u = \frac{1}{2} \nabla \cdot (\text{cof}(\mathbf{D}^2 u) \nabla u). \quad (4.3)$$

Then (4.1) is equivalent to

$$\frac{\partial u}{\partial t} - \frac{1}{2} \nabla \cdot (\text{cof}(\mathbf{D}^2 u) \nabla u) = f \quad \text{in } \Omega \times (0, T), \quad (4.4)$$

Equation (4.1) is a fully non-linear parabolic equation, and its equivalent form (4.4) is a some type of nonlinear heat equation. Equation (4.4) is well-posed when the nonlinearity $\text{cof}(\mathbf{D}^2 u)$ remains positive definite. This implies that $\mathbf{D}^2 u$ should remain positive definite. It is thus crucial that the numerical method maintains the positive definiteness of the Hessian $\mathbf{D}^2 u$. Like in Chapter 3, when we have solved the elliptic Monge-Ampère equation, we force the convexity of the Hessian matrix when solving constrained local non-linear minimization problems.

Remark. Following [Liu et al., 2019], one can consider an alternative formulation that consists in augmenting the differential equation into a differential system. By introducing an auxiliary variable $\mathbf{p} := \mathbf{D}^2 u$, and a penalization parameter $\varepsilon > 0$, (4.1) is weakly equivalent to: find $(u, \mathbf{p}) : (\Omega \times (0, T)) \rightarrow \mathbb{R} \times \mathbb{R}^{2 \times 2}$ satisfying:

$$\begin{cases} \frac{\partial u}{\partial t} - \det \mathbf{p} = f & \text{in } \Omega \times (0, T), \\ \frac{\partial \mathbf{p}}{\partial t} + \frac{1}{\varepsilon} (\mathbf{p} - \mathbf{D}^2 u) = 0, & \text{in } \Omega \times (0, T), \end{cases} \quad (4.5)$$

together with the additional initial condition $\mathbf{p}(0) = \mathbf{D}^2 u_0$. This approach has proved to be very efficient in capturing the stationary solution. However, accordingly to numerical experiments it is not clear if it is able to approximate accurately the whole transient trajectory of the evolutive problem. The parameter ε would have to be fine tuned with respect to the time-step. $\square$

In the sequel, we thus propose a second-order numerical method for the numerical approximation of the solution of (4.1), which relies on an implicit time-stepping scheme and a Newton method.

4.2 Numerical algorithm

Let $\Delta t > 0$ be a constant given time step, $t^n = n\Delta t$, $n = 1, 2, \dots$, to define the approximations $u^n \simeq u(t^n)$. The numerical algorithm proposed hereafter relies on a discretization of the formulation (4.1). In order to handle the stiff behavior of the Monge-Ampère operator, a semi-implicit time discretization of (4.1) is considered. In this case, we advocate a *midpoint rule* and, u^n being known, we look for the next time step approximation u^{n+1} satisfying

$$\frac{u^{n+1} - u^n}{\Delta t} - \det(\mathbf{D}^2 u^{n+1/2}) = f^{n+1/2}, \quad n = 0, 1, \dots, \quad (4.6)$$

where $u^{n+1/2} := \frac{u^{n+1} + u^n}{2}$ and $f^{n+1/2} := f\left(\frac{t^{n+1} + t^n}{2}\right)$. Then (4.6) can be written as

$$u^{n+1/2} - \frac{1}{2}\Delta t \det \mathbf{D}^2 u^{n+1/2} = u^n + \frac{1}{2}\Delta t f^{n+1/2}, \quad (4.7)$$

and

$$u^{n+1} = 2u^{n+1/2} - u^n. \quad (4.8)$$

Let us define $b^n := u^n + \frac{1}{2}\Delta t f^{n+1/2}$. Relationship (4.7) is rewritten at each time step as

$$F(u^{n+1/2}) := u^{n+1/2} - \frac{\Delta t}{2} \det(\mathbf{D}^2 u^{n+1/2}) - b^n = 0.$$

At each time step this nonlinear problem is solved with a safeguarded Newton method. For the ease of notation, we denote $u^{n+1/2}$ by v . Starting from the initial guess $v^0 = u^n$, the increments δv^k of the Newton method are obtained by solving

$$DF(v^k)\delta v^k = -F(v^k), \quad k = 0, 1, 2, \dots, \quad (4.9)$$

then, the next iterate is given by $v^{k+1} = v^k + \delta v^k$, until some stopping criterion is satisfied at step M ; then we set $u^{n+1/2} := v^M$. At the end of the Newton iteration loop, the approximation of the solution at the next time step is given by (4.8). In order to write the variational formulation corresponding to (4.9) we use the following identity which holds for 2×2 symmetric matrices (see, e.g., [Brenner and Neilan, 2012]):

$$\det \mathbf{D}^2(a + b) = \det(\mathbf{D}^2 a) + \det(\mathbf{D}^2 b) + \text{tr}(A^* \mathbf{D}^2 b), \quad (4.10)$$

where $A^* = \text{cof}(\mathbf{D}^2 a) = \det(\mathbf{D}^2 a)(\mathbf{D}^2 a)^{-1}$. This yields

$$\text{tr}(A^* \mathbf{D}^2 b) = \text{cof}(\mathbf{D}^2 a) : \mathbf{D}^2 b = \nabla \cdot (\text{cof}(\mathbf{D}^2 a) \nabla b),$$

where $\mathbf{A} : \mathbf{B} := \text{tr}(\mathbf{A}^T \mathbf{B})$ is the Frobenius inner product for $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{2 \times 2}$. Equation (4.10) becomes,

$$\det \mathbf{D}^2(a + b) = \det(\mathbf{D}^2 a) + \nabla \cdot (\text{cof}(\mathbf{D}^2 a) \nabla b) + \det(\mathbf{D}^2 b). \quad (4.11)$$

We thus have, for $s \in \mathbb{R}$,

$$F(v^k + s\delta v) = v^k + s\delta v - \frac{\Delta t}{2} \left(\det(\mathbf{D}^2 v^k) + \nabla \cdot (\text{cof}(\mathbf{D}^2 v^k) s \nabla \delta v) + s^2 \det(\mathbf{D}^2 \delta v) \right) - b^n.$$

We thus compute $DF(v^k)$ as follows:

$$DF(v^k)\delta v = \lim_{s \rightarrow 0} \frac{F(v^k + s\delta v) - F(v^k)}{s} = \delta v - \frac{\Delta t}{2} \nabla \cdot (\text{cof}(\mathbf{D}^2 v^k) \nabla \delta v). \quad (4.12)$$

In order to incorporate (4.12) in the variational formulation corresponding to (4.9), let us define $V_g = \{w \in H^1(\Omega) : w|_{\partial\Omega} = g\}$, and $V_0 = H_0^1(\Omega)$. Using (4.12), the variational formulation corresponding to the Newton system (4.9) can be explicit into : find $\delta v^k \in V_0$, for $k = 0, 1, 2, \dots$, such that

$$\int_{\Omega} \delta v^k w d\mathbf{x} + \frac{\Delta t}{2} \int_{\Omega} \text{cof}(\mathbf{D}^2 v^k) \nabla(\delta v^k) \cdot \nabla w d\mathbf{x} = - \int_{\Omega} \left(v^k - \frac{\Delta t}{2} \det(\mathbf{D}^2 v^k) - b^n \right) w d\mathbf{x}, \quad (4.13)$$

for all $w \in V_0$. This Newton's variational problem is coupled with a safeguarding strategy (Armijo's rule) when needed.

In addition, we make sure that the method guarantees that the matrix $\text{cof}(\mathbf{D}^2 v^k)$ remains positive definite. This procedure is achieved by computing the SVD of this matrix, and truncating its negative eigenvalues to zero. More precise, let A to be 2×2 symmetric matrix then we can compute $A = USU^T$, where S is the diagonal matrix with eigenvalues λ_1, λ_2 in the diagonal and U is an orthogonal matrix with columns that represent the eigenvectors of A . We then set $\lambda_i := \max(0, \lambda_i)$, for $i = 1, 2$ and we recompute $A = USU^T$. The procedure is important for the first iterations of the algorithm when looking for a solution of the steady state Monge-Ampère equation.

4.3 Finite Element Discretization

In order to avoid the construction of finite element sub-spaces of $H^2(\Omega)$ and to handle arbitrary shaped domains, we consider a mixed low order finite element method for the approximation of (4.13) see, e.g., [Caboussat et al., 2013, 2018]. Let us thus denote by $\mathcal{T}_h$ a regular finite element discretization of $\Omega \subset \mathbb{R}^2$ in triangles. From $\mathcal{T}_h$, we approximate the spaces $L^2(\Omega)$, $H^1(\Omega)$ and $H^2(\Omega)$, respectively $H_0^1(\Omega)$ and $H^2(\Omega) \cap H_0^1(\Omega)$, by the finite dimensional space V_h , respectively V_{0h} , defined by:

$$V_h = \left\{ v \in C^0(\overline{\Omega}), v|_K \in \mathbb{P}_1, \forall K \in \mathcal{T}_h \right\}, \quad V_{0,h} = V_h \cap H_0^1(\Omega), \quad (4.14)$$

with $\mathbb{P}_1$ the space of the two-variables polynomials of degree one. Moreover, let us define $V_{g,h} = \left\{ v \in C^0(\overline{\Omega}), v|_K \in \mathbb{P}_1, \forall K \in \mathcal{T}_h, v|_{\partial\Omega} = \pi_h g \right\}$ where $\pi_h g$ the interpolant of g .

In order to approximate second derivatives with low order polynomials we proceed as in Chapter 3 or [Caboussat et al., 2013]. Therefore, let a function φ being given in $H^1(\Omega)$, we approximate the differential operators D_{ij}^2 by $D_{h,ij}^2$, for $1 \leq i, j \leq 2$, defined by $D_{h,ij}^2(\varphi) \in V_{0h}$ and

$$\int_{\Omega} D_{h,ij}^2(\varphi) v d\mathbf{x} + C \sum_{K \in \mathcal{T}_h} |K| \int_K \nabla D_{h,ij}^2(\varphi) \cdot \nabla v d\mathbf{x} = -\frac{1}{2} \int_{\Omega} \left[\frac{\partial \varphi}{\partial x_i} \frac{\partial v}{\partial x_j} + \frac{\partial \varphi}{\partial x_j} \frac{\partial v}{\partial x_i} \right] d\mathbf{x},$$

where $C \geq 0$ and $|K| = \text{meas}(K)$,

Set u_h^0 be an approximation of u^0 in $V_{g,h}$. At each time step, the numerical approximation of (4.13) is computed as follows: let $v_h^0 := u_h^n$ at each time iteration; then, for $k = 0, 1, 2, \dots$, we search for $\delta v_h^k \in V_{0,h}$ such that:

$$\int_{\mathcal{T}_h} \delta v_h^k w_h d\mathbf{x} + \frac{\Delta t}{2} \int_{\mathcal{T}_h} \text{cof}(\mathbf{D}_h^2 v_h^k) \nabla(\delta v_h^k) \cdot \nabla w_h d\mathbf{x} = - \int_{\mathcal{T}_h} \left(v_h^k - \frac{\Delta t}{2} \det(\mathbf{D}_h^2 v_h^k) - b_h^n \right) w_h d\mathbf{x}, \quad (4.15)$$

for all $w_h \in V_{0,h}$. Then we set $v_h^{k+1} := v_h^k + \delta v_h^k$; when some stopping criterion is satisfied at step M , we set $u_h^{n+1/2} := v_h^M$. To progress to the next time step, we set $u_h^{n+1} = 2u_h^{n+1/2} - u_h^n$.

4.4 Numerical Experiments for Time Dependent Solutions

Numerical results are presented to validate the method for convex solutions. In the following examples, $\Omega = (0, 1)^2$ and $T = 1$. Both a triangular structured asymmetric mesh and an unstructured isotropic mesh are used. The mesh size h and the time step Δt vary together. The stopping criterion for the Newton method is $\|v_h^{k+1} - v_h^k\|_{L_2(\Omega)} \leq 10^{-12}$, with a maximal number of 200 Newton iterations. The Newton method typically needs 9 – 12 iterations to converge, depending on the mesh size and the time step. The parameter C is set to 1 (unless specified otherwise). The convergence of the error $e = u - u_h$ is quantified by the following quantities

$$\|e\|_{L^2(L^2)} := \int_0^T \|u - u_h\|_{L^2} dt, \quad \|e\|_{L^2(H^1)} := \int_0^T \|\nabla u - \nabla u_h\|_{L^2} dt,$$

In the tables below, those norms are approximated using the trapezoidal rule in time, and quadrature formulas in space (see [Caboussat et al., 2019]).

4.4.1 A polynomial example

Let us consider $T = 1$, and the exact solution:

$$u(x, y, t) = 0.5(0.5 + t)(x^2 + 5y^2), \quad (x, y) \in \Omega, \quad t \in (0, T). \quad (4.16)$$

This function is the solution of (4.1) with the data

$$f(x, y, t) := 0.5(x^2 + 5y^2) - 5(0.5 + t)^2,$$

$$g(x, y, t) := 0.5(0.5 + t)(x^2 + 5y^2),$$

and

$$u_0(x, y) := 0.25(x^2 + 5y^2).$$

The solution (4.16) is convex for all $t \in (0, T)$. Note that the eigenvalues of the Hessian $\mathbf{D}^2 u$ are $\lambda_1 = (0.5 + t)^2$ and $\lambda_2 = 5(0.5 + t)^2$, and are both positive for all $t \in (0, T)$. Figure 4.1 illustrates $u_0(x, y)$ (left) and $u_h(x, y, T)$ (right), while Table 4.1 shows that the solution method exhibits appropriate convergence orders (for the discrete version of the norms $\|u - u_h\|_{L^2(0, T; H^1(\Omega))}$ and $\|u - u_h\|_{L^2(0, T; L^2(\Omega))}$).

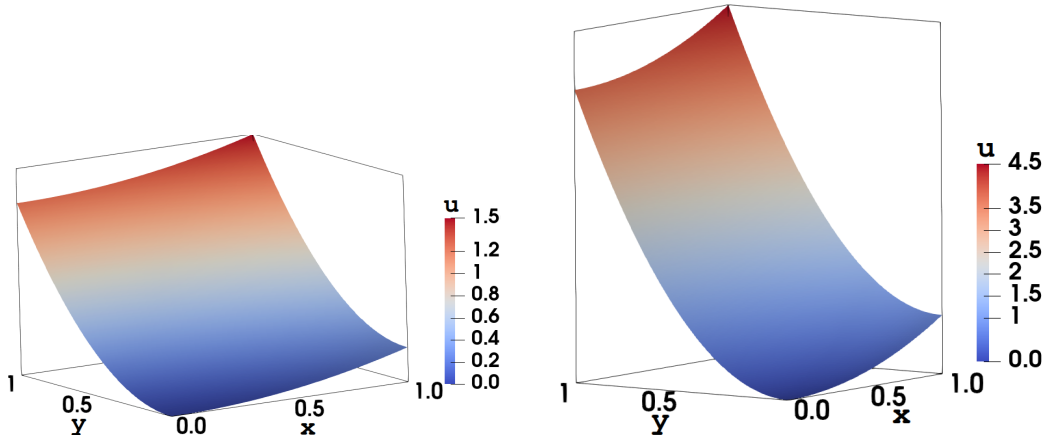


Figure 4.1 – A polynomial example corresponding to the exact solution (4.16). Numerical approximation of the solution for $h = 1/80$ and $\Delta t = 0.25 \cdot 10^{-3}$. Left: initial condition at time $t = 0$. Right: final solution at time $t = 1$.

Table 4.1 – A polynomial example. Estimated errors of $u - u_h$ in corresponding norms, and related convergence orders for various h and Δt . Top: structured meshes (with $C = 0$), Bottom: unstructured meshes.

Structured	h	Δt	$\ e\ _{L^2(L^2)}$		$\ e\ _{L^2(H^1)}$	
	1/20	1.00e-03	1.55e-03	-	7.37e-02	-
	1/40	0.50e-03	3.81e-04	2.02	3.68e-02	1.00
	1/80	0.25e-03	9.01e-05	2.08	1.84e-02	1.00
	1/160	0.125e-03	1.99e-05	2.17	9.20e-03	1.00
Unstructured	h	Δt	$\ e\ _{L^2(L^2)}$		$\ e\ _{L^2(H^1)}$	
	0.062	2.00e-03	2.10e-02	-	3.19e-01	-
	0.031	1.00e-03	7.28e-03	1.52	1.51e-01	1.07
	0.015	0.50e-03	1.90e-03	1.93	6.14e-02	1.29
	0.010	0.33e-03	8.29e-04	2.04	3.49e-02	1.39

4.4.2 An exponential example

Let us consider $T = 1$, and the exact solution

$$u(x, y, t) = e^{-t} e^{\frac{1}{2}(x^2+y^2)}, \quad (x, y) \in \Omega, \quad t \in (0, T). \quad (4.17)$$

This function is the solution of (4.1) with the data

$$f(x, y, t) := -e^{-t} e^{\frac{1}{2}(x^2+y^2)} \left(1 + e^{-t} (x^2 + y^2 + 1) e^{\frac{1}{2}(x^2+y^2)} \right),$$

$$g(x, y, t) := e^{-t} e^{\frac{1}{2}(x^2+y^2)},$$

$$u_0(x, y) := e^{\frac{1}{2}(x^2+y^2)}.$$

The solution (4.17) is convex for all time $t \in (0, T)$, since the eigenvalues of $\mathbf{D}^2 u$ are $\lambda_1 = e^{-t} e^{\frac{1}{2}(x^2+y^2)}$, and $\lambda_2 = e^{-t} e^{\frac{1}{2}(x^2+y^2)} (x^2 + y^2 + 1)$, which are both positive for all times $t \in (0, T)$. Figure 4.2 illustrates $u_0(x, y)$ (left) and $u_h(x, y, T)$ (right), while Table 4.2 shows that the solution method exhibits nearly optimal convergence orders (for structured and unstructured mesh we have $\mathcal{O}(h)$ and $\mathcal{O}(h^{1.5})$ for the discrete version of the norm $\|e\|_{L^2(H^1)}$ and $\mathcal{O}(h^{1.8})$ and $\mathcal{O}(h^2)$ for $\|e\|_{L^2(L^2)}$, respectively)

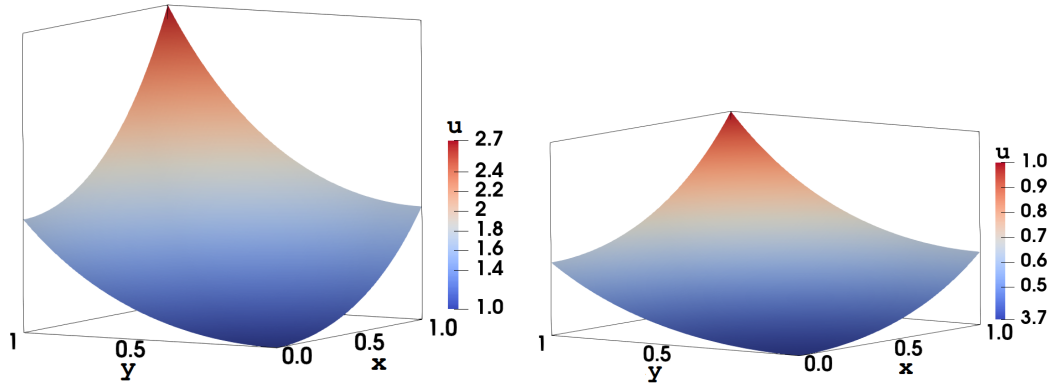


Figure 4.2 – Exponential example corresponding to the exact solution (4.17). Numerical approximation of the solution for $h = 1/80$ and $\Delta t = 0.25 \cdot 10^{-3}$. Left: initial condition at time $t = 0$. Right: the final solution at time $t = 1$.

4.5 Numerical Experiments for Time Independent Solutions

In this section, we consider the numerical solutions that are not depend on time. As in the beginning of this Chapter, we said this corresponds to solving the steady state Monge-Ampère equation (4.2).

We check the efficiency of the numerical algorithm for various test cases. The test cases of primary importance in this section are degenerate solutions, solutions with non-smooth

Table 4.2 – Exponential example. Estimated errors of $u - u_h$ in corresponding norms, and related convergence orders for various h and Δt . Top: structured meshes (with $C = 0$ when $h \geq 1/80$, and $C = 0.1$ when $h = 1/160$), Bottom: unstructured meshes.

Structured	h	Δt	$\ e\ _{L^2(L^2)}$		$\ e\ _{L^2(H^1)}$	
	1/20	1.00e-03	8.96e-04	-	3.58e-02	-
	1/40	0.50e-03	2.40e-04	1.90	1.79e-02	1.00
	1/80	0.25e-03	6.69e-05	1.80	8.96e-03	0.99
	1/160	1.25e-03	9.97e-06	2.74	4.44e-03	1.01
Unstructured	h	Δt	$\ e\ _{L^2(L^2)}$		$\ e\ _{L^2(H^1)}$	
	0.062	2.00e-03	1.49e-02	-	2.02e-01	-
	0.031	1.00e-03	5.31e-03	1.48	8.93e-02	1.17
	0.015	0.50e-03	1.25e-03	2.08	3.27e-02	1.44
	0.010	0.33e-03	5.26e-04	2.13	1.81e-02	1.45

data or data with Dirac function, etc. These cases are considered as challenging ones in the literature [Caboussat et al., 2013; Glowinski et al., 2019].

Let $\Omega = (0, 1)^2$ be the unit square. In all the test cases we use a triangular structured asymmetric mesh. The mesh size h and the time step Δt vary together. The stopping criterion for the Newton method is $\|v_h^{k+1} - v_h^k\|_{L_2(\Omega)} \leq 10^{-9}$, with a maximal number of 400 iterations. The parameter C is set to 1. We consider $T = \infty$, and we stop the algorithm when $\|u_h^{n+1} - u_h^n\|_{L_2(\Omega)} \leq 5 \times 10^{-8}$. The initial guess of the algorithm $u_0(x, y)$ is obtained by solving the following problem.

$$\begin{cases} \Delta u_0 = a & \text{in } \Omega, \\ u_0 = g & \text{on } \Gamma. \end{cases} \quad (4.18)$$

We use $a = 5$ in most of the examples except the ones with Dirac right-hand sides where we choose $a = 10$. In the literature, the initialization is usually chosen as a solution of $\Delta u_0 = 2\sqrt{f}$, see, for example, [Caboussat et al., 2013]. Here we need an initial condition that is more convex than the one from solving $\Delta u_0 = 2\sqrt{f}$. The choice of a influences the computational time of the algorithm, to obtain a stationary solution.

In order to simplify the notation we set $u := u^N$ and $u_h := u_h^N$ where N is the last time-step. In the sequel, we define $f(x, y)$ in Ω , $g(x, y)$ on Γ and $u(x, y)$ in $\bar{\Omega} = \Omega \cup \Gamma$.

4.5.1 Smooth examples

First we consider $f(x, y)$ to be a smooth function. The test cases we consider are:

$$(A) \quad f(x, y) = -5 \text{ and } g(x, y) = \frac{1}{2}(x^2 + 5y^2).$$

4.5. Numerical Experiments for Time Independent Solutions

(B) $f(x, y) = -(x + y + 1)e^{(x^2+y^2)}$ and $g(x, y) = e^{\frac{1}{2}(x^2+y^2)}$.

These examples are similar to those used in the previous section. Example (A) corresponds to a polynomial smooth solution, namely,

$$u(x, y) = \frac{1}{2}(x^2 + 5y^2). \quad (4.19)$$

The eigenvalues of the Hessian $\mathbf{D}^2 u$ are $\lambda_1 = 1$ and $\lambda_2 = 5$.

Example (B) is a classical example that is often seen in the literature. It corresponds to an exponential smooth solution

$$u(x, y) = e^{\frac{1}{2}(x^2+y^2)}. \quad (4.20)$$

The eigenvalues of the Hessian $\mathbf{D}^2 u$ are $\lambda_1 = e^{\frac{1}{2}(x^2+y^2)}$, and $\lambda_2 = e^{\frac{1}{2}(x^2+y^2)}(x^2 + y^2 + 1)$.

Table 4.3 illustrates the convergence of the numerical solutions and the number of timesteps for examples (A) and (B). Table 4.3 shows that the numerical solutions for both test cases converge in L^2 -norm with a rate of $O(h^{1.9})$ and in H^1 semi norm with an optimal rate of $O(h^{1.2})$ when $h, \Delta t \rightarrow 0$. The number of timesteps is multiplied by 2 as h and Δt are divided by two.

Figure 4.3 shows the obtained numerical solution for example (A) on the left and example (B) on the right.

Table 4.3 – Smooth examples. Estimated errors of $u - u_h$ in corresponding norms, and related convergence orders for various h and Δt . Number of timesteps of the algorithm. Top table: (A) $f(x, y) = 5$ and $g(x, y) = \frac{1}{2}(x^2 + 5y^2)$; Bottom table: (B) $f(x, y) = (x + y + 1)e^{(x^2+y^2)}$ and $g(x, y) = e^{\frac{1}{2}(x^2+y^2)}$.

	h	Δt	$\ u - u_h\ _{L^2(\Omega)}$		$ u - u_h _{H^1(\Omega)}$		Timesteps
(A)	1/20	1.00e-03	4.28e-03	-	1.15e-01	-	201
	1/40	0.50e-03	1.15e-03	1.89	5.12e-02	1.16	320
	1/80	0.25e-03	2.96e-04	1.95	2.22e-02	1.20	576
	h	Δt	$\ u - u_h\ _{L^2(\Omega)}$		$ u - u_h _{H^1(\Omega)}$		Timesteps
(B)	1/20	1.00e-03	4.56e-03	-	9.49e-02	-	298
	1/40	0.50e-03	1.19e-03	1.93	3.99e-02	1.25	550
	1/80	0.25e-03	3.10e-04	1.94	1.74e-02	1.19	1018

4.5.2 Non-smooth examples

We consider examples that often appear in the literature on the numerical solution of two dimensional Monge-Ampère equation. The problems we consider are:

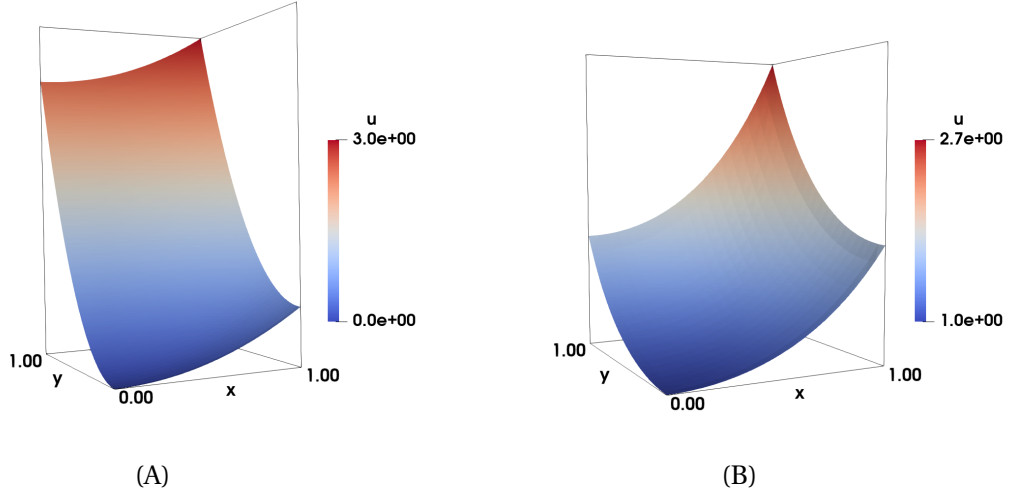


Figure 4.3 – Smooth examples. Left: Numerical approximation of the solution of test case (A) $f(x, y) = -5$ and $g(x, y) = \frac{1}{2}(x^2 + 5y^2)$. Right: Numerical approximation of the solution of test case (B) $f(x, y) = -(x + y + 1)e^{(x^2 + y^2)}$ and $g(x, y) = e^{\frac{1}{2}(x^2 + y^2)}$. The results are obtained with $h = 1/40$ and $\Delta t = 0.50 \cdot 10^{-3}$.

(C) $f(x, y) = 0$ and $g(x, y) = |x - 0.5|$.

(D) $f(x, y) = -1$ and $g(x, y) = 0$.

Test case (C) is a degenerate case where the solution has low regularity properties, and we expect the convergence order in L^2 -norm to be of order one or less. Table 4.4 illustrates the numerical solutions of example (C) for various h and Δt . We observe that, in L^2 -norm although the numerical solution is accurate of order 10^{-3} , there is no convergence order. Moreover, we measure the absolute error at $(0.5, 0.5)$, and we see that it goes to zero as $h, \Delta t \rightarrow 0$. Concerning the timesteps, we note that they increase as h decreases.

In order to improve the results and reduce the number of timesteps, we use a simple adaptive strategy heuristics in time. We reduce the time-step when the Newton iterations are smaller than five. The results are reported in Table 4.5. We observe that the convergence in L^2 -error norm is of the order of $O(h)$, and the number of timesteps are reduced remarkably. The point error at $(0.5, 0.5)$ remained the same. In Table 4.5, we report the first time-step Δt_0 and the last time-step Δt_N . Graphs of the numerical solution of test case (C) without an adaptive time-step strategy are illustrated in Figure 4.4.

Example (D) does not have a classical solution but only admits a generalized one, even though the data are smooth. This problem is theoretically discussed in [Gutiérrez, 2001] and numerically investigated in numerous works e.g. [Feng and Neilan, 2009c; Oberman, 2008; Benamou et al., 2010; Caboussat et al., 2013; Awanou, 2014; Glowinski et al., 2019]. We compare our numerical solution with the numerical solution given in the previously cited

4.5. Numerical Experiments for Time Independent Solutions

Table 4.4 – Non-smooth examples. Test case (C) with data $f(x, y) = 0$ and $g(x, y) = |x - 0.5|$. Estimated errors $u - u_h$ in corresponding norms, and related convergence orders for various h and Δt . Number of timesteps of the algorithm.

	h	Δt	$\ u - u_h\ _{L^2(\Omega)}$		$ u(0.5, 0.5) - u_h(0.5, 0.5) $	Timesteps
(C)	1/20	1.00e-03	2.00e-03	-	0.0138	3957
	1/40	0.50e-03	1.19e-03	0.74	0.0092	6991
	1/80	0.25e-03	1.16e-03	-	0.0057	10991

Table 4.5 – Non-smooth examples. Test case (C) with data $f(x, y) = 0$ and $g(x, y) = |x - 0.5|$. Estimated errors $u - u_h$ in corresponding norms, and related convergence orders for various h and adaptive time-step. First and the last time-step. Number of timesteps of the algorithm.

	h	Δt_0	Δt_N	$\ u - u_h\ _{L^2(\Omega)}$		$ u(0.5, 0.5) - u_h(0.5, 0.5) $	Timesteps
(C)	1/20	1.00e-03	4194.3	1.98e-03	-	0.0138	358
	1/40	0.50e-03	8388.61	1.01e-03	0.97	0.0093	536
	1/80	0.25e-03	8388.61	4.85e-04	1.05	0.0057	755

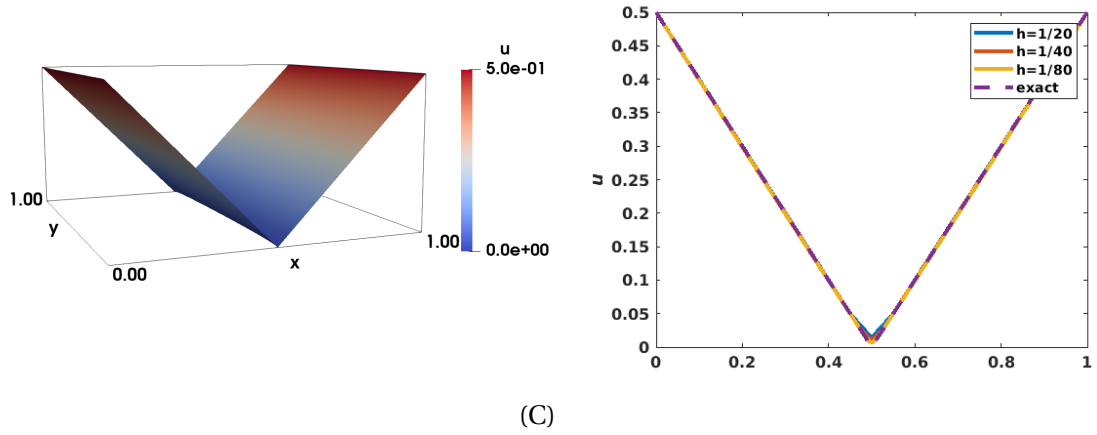


Figure 4.4 – Non-smooth examples. Numerical approximation of the solution of (C) $f(x, y) = 0$ and $g(x, y) = |x - 0.5|$. Left: Graph of the numerical solution (with $h = 1/40$ and $\Delta t = 0.50 \cdot 10^{-3}$). Right: Profile plot of u as a function x and $y = 0.5$.

articles. The numerical solution of (D) is illustrated in Figure 4.5 (top left) and contours of the solution are given in Figure 4.5 (top right). Comparison of the amplitude of the numerical solution for different h is done in the bottom row of Figure 4.5, where we plot the solutions as a function of x for $y = 0.5$ on the left (resp $x = y$ on the right). The numerical solution obtained aligns with the ones given in the above mentioned references. In [Glowinski et al., 2019] by

Chapter 4. Numerical Approximation of a 2D Parabolic Monge-Ampère Equation

comparing [Oberman, 2008; Benamou et al., 2010], the authors provide a range of points for the minimal solution that is $[-0.184475, -0.17315]$. In Table 4.6, we show that we are in that range for different h and Δt . Moreover, the tables show that the number of timesteps is doubled as h and Δt decrease by half.

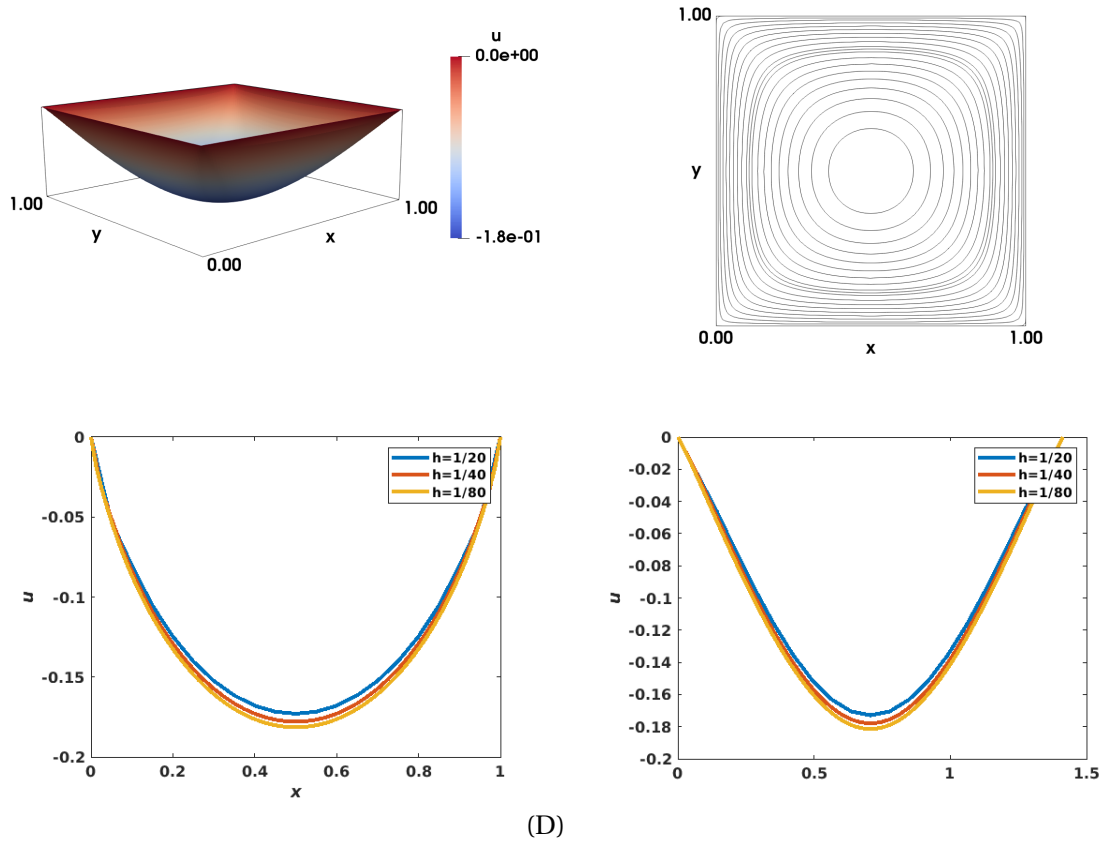


Figure 4.5 – Non-smooth examples. Numerical approximation of the solution of (D) $f(x, y) = 0$ and $g(x, y) = |x - 0.5|$. Top row: Graph of the numerical solution and contours of the numerical solution. Bottom row: Profile plot of $u_h(x, 0.5)$ and $u_h(0.5, y)$. The results are obtained with $h = 1/40$ and $\Delta t = 0.50 \cdot 10^{-3}$.

Table 4.6 – Non-smooth examples. Example (D) with data $f(x, y) = -1$ and $g(x, y) = 0$. Estimation of the minimal solution u_h along with number of timesteps for various h and Δt .

	h	Δt	$\min(u)$	timesteps
(D)	1/20	1.00e-03	-0.1730	640
	1/40	0.50e-03	-0.1780	1238
	1/80	0.25e-03	-0.1814	2349

4.5.3 Examples with singularities

We consider examples with singularities. The problems are defined as

$$\begin{aligned} \text{(E)} \quad f(x, y) &= -\frac{R^2}{(R^2 - (x - 0.5)^2 - (y - 0.5)^2)^2} \text{ and } g(x, y) = -\sqrt{R^2 - (x - 0.5)^2 - (y - 0.5)^2}. \\ \text{(F)} \quad f(x, y) &= -\frac{1 - (1 - 2x)^2(1 - 2y)^2}{16xy(1 - x)(1 - y)} \text{ and } g(x, y) = 0. \end{aligned}$$

Test case (E) is a classical example with the exact solution given by

$$u(x, y) = -\sqrt{R^2 - (x - 0.5)^2 - (y - 0.5)^2}, \quad (4.21)$$

we also considered a similar test case for the three dimensional Monge-Ampère in Chapter 3. For $R > 1/\sqrt{2}$, the solution belongs to $C^\infty(\bar{\Omega})$. Here we are going to consider the case that $R = 1/\sqrt{2}$ where ∇u is discontinuous in the four corners of $\bar{\Omega}$, while the solution remains smooth.

Test case (F) has the following the exact solution

$$u(x, y) = -\sqrt{x(1 - x)y(1 - y)}. \quad (4.22)$$

Similarly to the previous test case, the solution is smooth, but ∇u is discontinuous on the whole boundary.

Graphs of the numerical solution of test cases (E) and (F) are presented in Figure 4.6 for $h = 1/40$ and $\Delta t = 0.50 \times 10^{-3}$. Table 4.7 provides information about the error norms with the corresponding rates and the number of timesteps for various h and Δt , for the test cases (E) and (F). In the top table for the test case (E), the L^2 error norm decreases with an order of $O(h^{1.8})$, and the H^1 error norm decreases with an order of $O(h^{0.7})$. Since ∇u is not regular, it is logical to expect a convergence order less than one. In the top table for the test case (F), we see that it is more challenging than the previous one. The convergence order in the L^2 error norm drops to 0.5, while there is no convergence order for the H^1 error norm. Finally, we observe that for both test cases, the number of timesteps is increasing linearly.

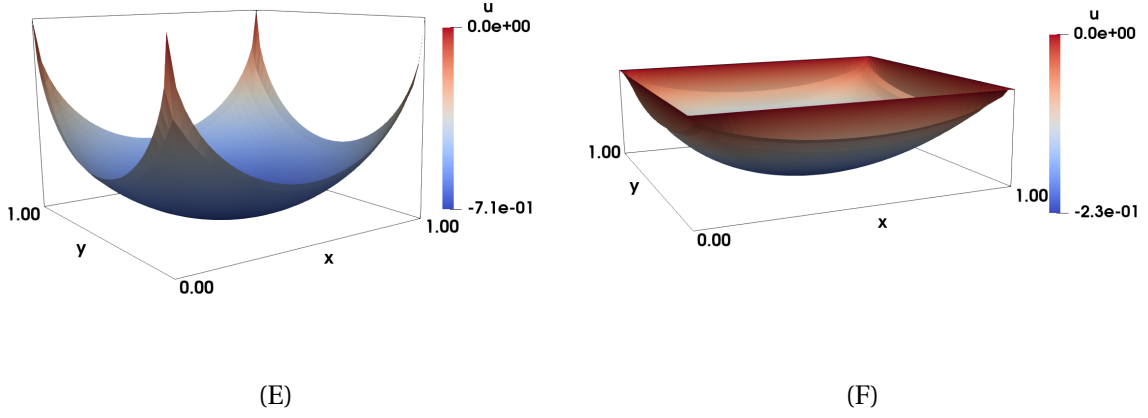


Figure 4.6 – Examples with singularities. Left: Graph of the numerical approximation of the solution of test case (E) with $f(x, y) = -\frac{R^2}{(R^2 - (x-0.5)^2 - (y-0.5)^2)^2}$, $g(x, y) = -\sqrt{R^2 - (x-0.5)^2 - (y-0.5)^2}$ and $R = \frac{1}{\sqrt{2}}$. Right: Graph of the numerical approximation of the solution of test case (F) with $f(x, y) = -\frac{1-(1-2x)^2(1-2y)^2}{16xy(1-x)(1-y)}$ and $g(x, y) = 0$. The results are visualized with $h = 1/40$ and $\Delta t = 0.50 \cdot 10^{-3}$.

Table 4.7 – Examples with singularities on the data. Estimated errors $u - u_h$ in corresponding norms, and related convergence orders for various h and Δt . Number of timesteps of the algorithm. Top table: Test case (E) with data $f(x, y) = -\frac{R^2}{(R^2 - (x-0.5)^2 - (y-0.5)^2)^2}$, $g(x, y) = -\sqrt{R^2 - (x-0.5)^2 - (y-0.5)^2}$ and $R = \frac{1}{\sqrt{2}}$; Bottom table: Test case (F) with data $f(x, y) = -\frac{1-(1-2x)^2(1-2y)^2}{16xy(1-x)(1-y)}$ and $g(x, y) = 0$.

(E)	h	Δt	$\ u - u_h\ _{L^2(\Omega)}$		$ u - u_h _{H^1(\Omega)}$		Timesteps
	1/20	1.00e-03	8.47e-03	-	2.88e-01	-	277
	1/40	0.50e-03	2.41e-03	1.77	1.79e-01	0.68	512
	1/80	0.25e-03	6.90e-04	1.80	1.11e-01	0.68	943

(F)	h	Δt	$\ u - u_h\ _{L^2(\Omega)}$		$ u - u_h _{H^1(\Omega)}$		Timesteps
	1/20	1.00e-03	2.07e-02	-	4.12e-01	-	428
	1/40	0.50e-03	1.46e-02	0.50	4.03e-01	-	803
	1/80	0.25e-03	9.41e-03	0.63	3.99e-01	-	1477

4.5.4 Examples with Dirac functions on the right hand side

We consider examples with Dirac functions on the right hand side. The problems are defined as

$$(G) \quad f(x, y) = -2\delta_{(1/2, 1/2)} \text{ and } g(x, y) = 0.$$

$$(H) \quad f(x, y) = -\pi\delta_{(1/2, 1/2)} \text{ and } g(x, y) = \sqrt{(x-0.5)^2 + (y-0.5)^2}.$$

$$(I) \quad f(x, y) = -\frac{\pi}{2}\delta_{(1/4, 1/2)} - \frac{\pi}{2}\delta_{(3/4, 1/2)} \text{ and}$$

$$g(x, y) = \begin{cases} |y-0.5| & \text{if } 1/4 < x < 3/4, \\ \min\left(\sqrt{(x-1/4)^2 + (y-0.5)^2}, \sqrt{(x-3/4)^2 + (y-0.5)^2}\right) & \text{Otherwise.} \end{cases}$$

In order to apply our methodology, we approximate the Dirac function $\delta_{(\alpha, \beta)}$ by

$$f_{\eta}^{(\alpha, \beta)}(x, y) = \frac{\eta^2}{\pi\left(\eta^2 + (x-\alpha)^2 + (y-\beta)^2\right)^2},$$

where η is a small positive value.

In example (G) we approximate $-2\delta_{(1/2, 1/2)}$ by $-2f_{\eta}^{(1/2, 1/2)}(x, y)$. The exact solution given in [Glowinski et al., 2019] is

$$u = -\min(x, 1-x, y, 1-y). \quad (4.23)$$

The solution has low regularity properties. To estimate the error norms, we assume that $\lim_{h \rightarrow 0} \lim_{\eta \rightarrow 0} u_h^{\eta} = u$. For this test case, we choose $\eta = 10^{-3/2}$ as suggested in [Glowinski et al., 2019]. Figure 4.7 displays the graph of the numerical solution for $h = 1/40$ and $\delta = 0.5 \times 10^{-3}$. Moreover, the profile of the numerical solutions as a function of $(x, y = 0.5)$ and $(x, y = x)$ shows the behavior of the numerical solution for different h and Δt . In Table 4.8, we observe that the convergence in L^2 -norm drops to $O(h^{0.5})$ and the minimal numerical solution does not consistently converges to -0.5 as expected. In order to reduce the timesteps we can use the adaptive strategy that we described for the test case (D). To improve the convergence order of the method we would need to investigate a better choice of the parameter η .

Test case (H) and (I) are the so-called Pogorelov solutions [Benamou et al., 2014; Benamou and Froese, 2014], involving one and then two Dirac functions. Numerical investigation for test case (H) and (I) is done in [Glowinski et al., 2019] and [Awanou and Leopold, 2015], respectively. In test cases (H) and (I) we respectively approximate $-\pi\delta_{(1/2, 1/2)}$ and $-\frac{\pi}{2}\delta_{(1/4, 1/2)} - \frac{\pi}{2}\delta_{(3/4, 1/2)}$ by $-\pi f_{\eta}^{(1/2, 1/2)}(x, y)$ and $-\frac{\pi}{2}f_{\eta}^{(1/4, 1/2)}(x, y) - \frac{\pi}{2}f_{\eta}^{(3/4, 1/2)}(x, y)$.

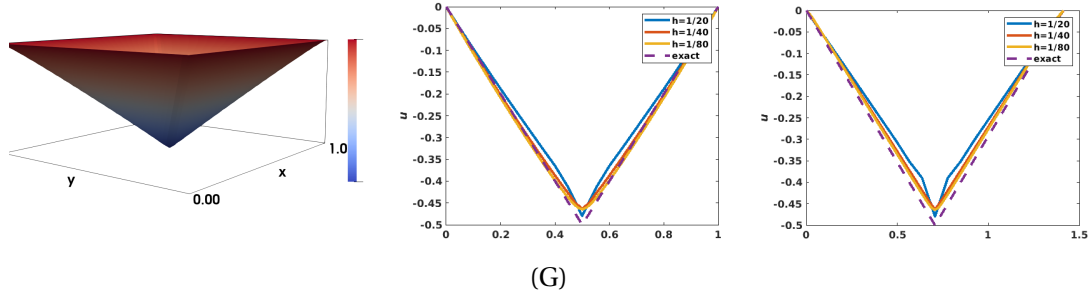


Figure 4.7 – Examples with Dirac function on right hand side. Numerical approximation of the solution of test case (G) $f(x, y) = -2\delta_{(1/2, 1/2)}$ and $g = 0$. Left: Graph of the numerical solution. Middle: Profile plot of u as a function x and $y = 0.5$. Right: Profile plot of u along the diagonal $x = y$. The results are obtained with $h = 1/40$, $\Delta t = 0.50 \cdot 10^{-3}$ and $\eta = 10^{-3/2}$.

Table 4.8 – Examples with Dirac function on right hand side. Test case (G) with data $f(x, y) = -2\delta_{(1/2, 1/2)}$ and $g = 0$. Estimated error $u - u_h$ in corresponding norm, and related convergence orders for various h and Δt . Estimation of the minimal solution u_h along with number of timesteps.

	h	Δt	$\ u - u_h\ _{L^2(\Omega)}$		$\min(u_h)$	Timesteps
(G)	1/20	1.00e-03	2.12e-02	-	-0.4798	559
	1/40	0.50e-03	9.40e-03	1.17	-0.4636	4658
	1/80	0.25e-03	6.74e-03	0.47	-0.4645	9930

Example (H) admits the exact solution

$$u(x, y) = \sqrt{(x - 0.5)^2 + (y - 0.5)^2}, \quad (4.24)$$

and test case (I) admits the exact solution

$$u(x, y) = \begin{cases} |y - 0.5| & \text{if } 1/4 < x < 3/4, \\ \min\left(\sqrt{(x - 1/4)^2 + (y - 0.5)^2}, \sqrt{(x - 3/4)^2 + (y - 0.5)^2}\right) & \text{Otherwise.} \end{cases} \quad (4.25)$$

As before, we assume that $\lim_{h \rightarrow 0} \lim_{\eta \rightarrow 0} u_h^\eta = u$, in order to compute error norms.

For the example (H), we use $\eta = 10^{-3/2}$ as suggested in [Glowinski et al., 2019] and results are reported in Table 4.9. We observe that the L^2 error norm converges as $O(h^{0.8})$. Moreover, the provided minimal solution is close to zero, as expected. We are able to reduce the number of timesteps by using the adaptive strategy as described before. The results are reported in Table 4.10.

4.5. Numerical Experiments for Time Independent Solutions

For $h = 1/40$ and $\Delta t = 0.5 \times 10^{-3}$, Figure 4.8 displays the graph of the numerical solution. Moreover, we profile the solutions as function of $(x, y = 0.5)$ and $(x, y = x)$, in order to show the behavior of numerical solution along these cuts for various h and Δt .

For the test case (G), we use $\eta = h$. The results are reported in Table 4.11. The same observations made before hold in this case. Moreover, we use the the adaptive strategy to reduce the number of timesteps, the results reported in Table 4.12.

Figure 4.9 displays the graph of the numerical solution for $h = 1/40$ and $\delta = 0.5 \times 10^{-3}$. Moreover, we show the solutions as a function of y for $x = 0.5$ in order to show the behavior of the numerical solution for different h and Δt .

Table 4.9 – Examples with Dirac function on the right hand side. Test case (H) with data $f(x, y) = -\pi\delta_{(1/2, 1/2)}$ and $g(x, y) = \sqrt{(x-0.5)^2 + (y-0.5)^2}$. Estimated error $u - u_h$ in corresponding norms, and related convergence orders for various h and Δt . Estimation of the minimal solution u_h along with the number of timesteps.

	h	Δt	$\ u - u_h\ _{L^2(\Omega)}$		$\min(u_h)$	Timesteps
(H)	1/20	1.00e-03	8.24e-03	-	-0.0263	337
	1/40	0.50e-03	4.17e-03	0.98	0.0178	564
	1/80	0.25e-03	2.42e-03	0.78	0.0283	1055

Table 4.10 – Examples with Dirac function on the right hand side. Test case (H) with data $f(x, y) = -\pi\delta_{(1/2, 1/2)}$ and $g(x, y) = \sqrt{(x-0.5)^2 + (y-0.5)^2}$. Estimated error $u - u_h$ in corresponding norms, and related convergence orders for various h and adaptive time-step. Indication of the first and the last time-step. Estimation of the minimal solution u_h along with the number of timesteps.

	h	Δt_0	Δt_N	$\ u - u_h\ _{L^2(\Omega)}$		$\min(u_h)$	Timesteps
(H)	1/20	1.00e-03	0.256	8.24e-03	-	-0.0263	165
	1/40	0.50e-03	16.384	4.17e-03	0.98	0.0178	195
	1/80	0.25e-03	0.512	2.42e-03	0.78	0.0283	257

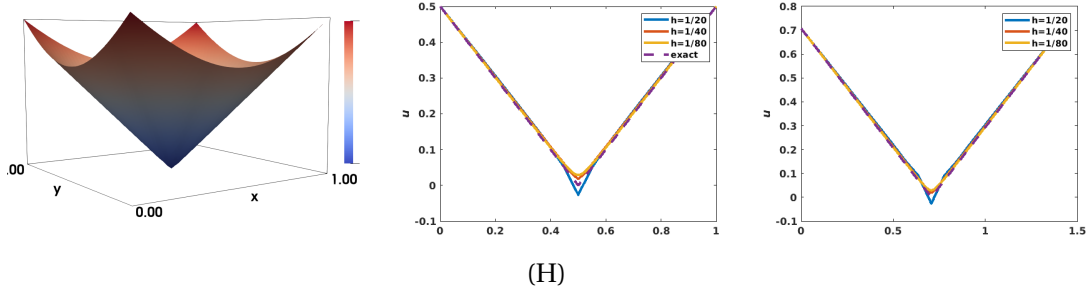


Figure 4.8 – Examples with Dirac function on the right hand side. Numerical approximation of the solution of test case (H) with data $f(x, y) = -\pi\delta_{(1/2, 1/2)}$ and $g(x, y) = \sqrt{(x-0.5)^2 + (y-0.5)^2}$. Left: Graph of the numerical solution. Middle: Profile plot of u as a function x and $y = 0.5$. Right: Profile plot of u along the diagonal $x = y$. The results are obtained with $h = 1/40$, $\Delta t = 0.50 \cdot 10^{-3}$ and $\eta = 10^{-3/2}$.

Table 4.11 – Examples with Dirac function on the right hand side. Bottom table: Test case (I) with data $f(x, y) = -\frac{\pi}{2}\delta_{(1/4, 1/2)} - \frac{\pi}{2}\delta_{(3/4, 1/2)}$ and $g(x, y) = u(x, y)$ where $u(x, y)$ is given in (4.25). Estimated error $u - u_h$ in corresponding norms, and related convergence orders for various h and Δt . Estimation of the minimal solution u_h along with the number of timesteps.

	h	Δt	$\ u - u_h\ _{L^2(\Omega)}$		$\min(u_h)$	timesteps
(I)	1/20	1.00e-03	1.08e-02	-	0.0024	468
	1/40	0.50e-03	7.05e-03	0.61	0.0039	1261
	1/80	0.25e-03	4.04e-03	0.80	0.0047	3451

Table 4.12 – Examples with Dirac function on the right hand side. Test case (I) with data $f(x, y) = -\frac{\pi}{2}\delta_{(1/4, 1/2)} - \frac{\pi}{2}\delta_{(3/4, 1/2)}$ and $g(x, y) = u(x, y)$ where $u(x, y)$ is given in (4.25). Estimated error $u - u_h$ in corresponding norms, and related convergence orders for various h and adaptive time-step. Indication of the first and the last time-step. Estimation of the minimal solution u_h along with the number of timesteps.

	h	Δt_0	Δt_N	$\ u - u_h\ _{L^2(\Omega)}$		$\min(u_h)$	timesteps
(I)	1/20	1.00e-03	0.512	1.08e-02	-	0.0024	166
	1/40	0.50e-03	0.512	7.05e-03	0.61	0.0039	249
	1/80	0.25e-03	4.096	4.04e-03	0.80	0.0047	352

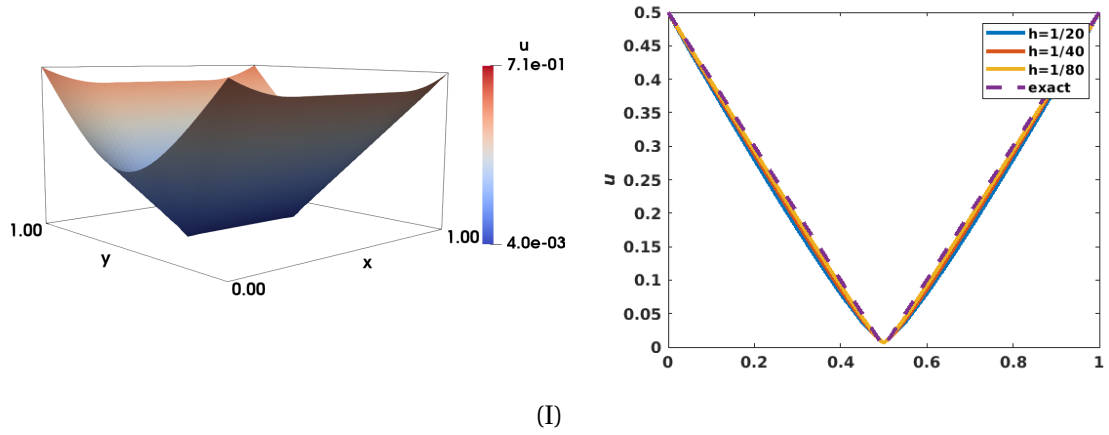


Figure 4.9 – Examples with Dirac function on the right hand side. Numerical approximation of the solution of test case (I) with data $f(x, y) = -\frac{\pi}{2}\delta_{(1/4, 1/2)} - \frac{\pi}{2}\delta_{(3/4, 1/2)}$ and $g(x, y) = u(x, y)$ where $u(x, y)$ is given in (4.25). Left: Graph of the numerical solution. Right: Profile plot of u as a function x and $y = 0.5$. The results are obtained with $h = 1/40$, $\Delta t = 0.50 \cdot 10^{-3}$ and $\eta = h$.

4.6 An adaptive algorithm for the elliptic Monge-Ampère equation

In this section we design an adaptive algorithm for the steady state Monge-Ampère equation. We use two different solvers to test the adaptive algorithm. The first solver, described throughout this chapter, has shown its efficiency in capturing the stationary solution of Monge-Ampère equation; we will refer to it as Parabolic Monge-Ampère (PMA). The second one is the two-dimensional version of the solver developed in Chapter 3, we will refer it as Least-squares Monge-Ampère (LSMA) and we describe it briefly in the subsequent paragraph.

LSMA is an iterative method that is based on the least-squares approach. Let the functional spaces be defined as

$$\begin{aligned} V_g^2 &= \{\varphi \in H^2(\Omega), \varphi = g \text{ on } \partial\Omega\}, \\ \mathbf{Q}_f &= \{\mathbf{q} \in L^2(\Omega)^{2 \times 2}, \mathbf{q} = \mathbf{q}^t, \det \mathbf{q} = f, q_{ii} > 0, i = 1, 2\}. \end{aligned}$$

The method relies on introducing an auxiliary variable $\mathbf{q} \in \mathbf{Q}_f$: find $(u, \mathbf{p}) \in V_g^2 \times \mathbf{Q}_f$ such that

$$J(u, \mathbf{p}) \leq J(v, \mathbf{q}), \quad \forall (v, \mathbf{q}) \in V_g^2 \times \mathbf{Q}_f, \quad (4.26)$$

where

$$J(v, \mathbf{q}) = \frac{1}{2} \int_{\Omega} |\mathbf{D}^2 v - \mathbf{q}|^2 dx.$$

In order to solve (4.26) we advocate a relaxation algorithm of the Gauss-Seidel-type. First, the initialization is obtained by solving:

$$\begin{aligned} \Delta u^0 &= 2\sqrt{f} \text{ in } \Omega, \\ u^0 &= g \text{ on } \partial\Omega. \end{aligned}$$

Then, for $n \geq 0$, $u^n \rightarrow \mathbf{p}^n \rightarrow u^{n+1/2} \rightarrow u^{n+1}$:

$$\mathbf{p}^n = \arg \min_{\mathbf{q} \in \mathbf{Q}_f} J(u^n, \mathbf{q}); \quad (4.27)$$

$$u^{n+1/2} = \arg \min_{v \in V_g^2} J(v, \mathbf{p}^n); \quad (4.28)$$

$$u^{n+1} = u^n + \omega(u^{n+1/2} - u^n); \quad (4.29)$$

with $1 \leq \omega \leq 2$.

The minimization problem (4.27) can be solved point-wise and it is solved with the $\mathbf{Q}_{min}$ algorithm, see [Sorensen and Glowinski, 2010]. The minimization problem (4.28) is a linear variational problem and results in a fourth-order partial differential equation, namely find

4.6. An adaptive algorithm for the elliptic Monge-Ampère equation

$u^{n+1/2} \in V_g^2$ satisfying

$$\int_{\Omega} \mathbf{D}^2 u^{n+1/2} : \mathbf{D}^2 v d\mathbf{x} = \int_{\Omega} \mathbf{p}^n : \mathbf{D}^2 v d\mathbf{x}, \quad \forall v \in V_0^2, \quad (4.30)$$

where $V_0^2 = H^2(\Omega) \cap H_0^1(\Omega)$. Equation (4.30) is solved by using a conjugate gradient algorithm.

The details of the method are in [Caboussat et al., 2013], where they show that it is robust. The solvers PMA and LSMA are both using low-order finite elements and compute the Hessian matrix as described in Section 4.3.

Our goal, like in Chapter 1, is to build an adaptive mesh such that the estimated relative error is close to a preset tolerance TOL , namely:

$$0.5 \, TOL \leq \frac{\eta^I}{\|\nabla u_h\|_{L^2(\Omega)}} \leq 1.5 \, TOL,$$

where the error estimate η^I is based on the finite element mesh and u_h computed either by PMA or LSMA. There are many options for estimating η^I . For instance for PMA, we can derive an estimate to the linearized equation (4.13). A posteriori error estimates for similar equations are developed in [Verfürth, 1994]. For LSMA, we can derive an estimate to the linear variational problem (4.30). Although, a posteriori error estimates for fourth-order elliptic problems exist in the literature e.g [Georgoulis et al., 2009], to the best of our knowledge, they consider high-order finite element approximations.

Another feasible option for estimating η^I is the following: in the conjugate gradient algorithm that is used to solve (4.30), we have to solve in the following problem several times: find $\varphi \in V_g^2$ satisfying

$$\int_{\Omega} \Delta \varphi \Delta v d\mathbf{x} = \int_{\Omega} \mathbf{D}^2 u : \mathbf{D}^2 v d\mathbf{x} - \int_{\Omega} \mathbf{p}^n : \mathbf{D}^2 v, \quad \forall v \in V_0^2. \quad (4.31)$$

To solve (4.31) we use mixed finite elements methods. For equation (4.31) with a mixed method formulation a posteriori and a priori error estimates are given in [Charbonneau et al., 1997]. However, as mentioned above, the obtained estimates cannot be used directly since the authors also consider high-order finite element approximations.

The estimate of η^I we propose here is derived directly from the Monge-Ampère equation. Let us re-write this equation using the expression in (4.3). The problem reads as follows: find $u : \Omega \rightarrow \mathbb{R}$ such that

$$\begin{cases} \frac{1}{2} \nabla \cdot (\text{cof}(\mathbf{D}^2 u) \nabla u) = f & \text{in } \Omega, \\ u = g & \text{on } \partial\Omega, \end{cases} \quad (4.32)$$

where f, g are given. The weak formulation of this equation can be obtained by multiplying

(4.32) by $v \in H_0^1(\Omega)$ and integrate by parts over Ω

$$-\frac{1}{2} \int_{\Omega} \text{cof}(\mathbf{D}^2 u) \nabla u \cdot \nabla v \, d\mathbf{x} = \int_{\Omega} f v \, d\mathbf{x}. \quad (4.33)$$

The error estimate that we derive is based on the residual. Therefore, we re-express the residual based on an integral term for which we can provide an upper bound

$$\begin{aligned} \frac{1}{2} \int_{\Omega} (\text{cof}(\mathbf{D}^2 u) \nabla u - \text{cof}(\mathbf{D}_h^2 u_h) \nabla u_h) \cdot \nabla (u - u_h) \, d\mathbf{x} &= - \int_{\Omega} f (u - u_h) \, d\mathbf{x} \\ &\quad - \frac{1}{2} \int_{\Omega} \text{cof}(\mathbf{D}_h^2 u_h) \nabla u_h \cdot \nabla (u - u_h) \, d\mathbf{x}, \\ &= R(u - u_h). \end{aligned} \quad (4.34)$$

Following [Verfürth, 2013], an upper bound of $R(u - u_h)$ is given by

$$\begin{aligned} R(u - u_h) \leq c \left(\sum_{K \in \mathcal{T}_h} \left(h_K^2 \left\| -f + \frac{1}{2} \nabla \cdot (\text{cof}(\mathbf{D}_h^2 u_h) \nabla u_h) \right\|_{L^2(K)}^2 \right. \right. \\ \left. \left. + \frac{1}{16} h_K \left\| [\text{cof}(\mathbf{D}_h^2 u_h) \nabla u_h \cdot \mathbf{n}] \right\|_{L^2(\partial K)}^2 \right) \right)^{\frac{1}{2}} \|\nabla (u - u_h)\|_{L^2(\Omega)}, \end{aligned}$$

where c is a constant that depends the mesh aspect ratio. Recall that the second order derivatives $\mathbf{D}_h^2 u_h$ are approximated as discussed in Section 4.3.

We denote the error estimate by η^I which is given by

$$\eta^I = \left(\sum_{K \in \mathcal{T}_h} (\eta_K^I)^2 \right)^{\frac{1}{2}}, \quad (4.35)$$

where

$$(\eta_K^I)^2 = h_K^2 \left\| -f + \frac{1}{2} \nabla \cdot (\text{cof}(\mathbf{D}_h^2 u_h) \nabla u_h) \right\|_{L^2(K)}^2 + \frac{1}{16} h_K \left\| [\text{cof}(\mathbf{D}_h^2 u_h) \nabla u_h \cdot \mathbf{n}] \right\|_{L^2(\partial K)}^2. \quad (4.36)$$

If we plug expression (4.3) in (4.36), we get

$$(\eta_K^I)^2 = h_K^2 \left\| -f + \det \mathbf{D}_h^2 u_h \right\|_{L^2(K)}^2 + \frac{1}{16} h_K \left\| [\text{cof}(\mathbf{D}_h^2 u_h) \nabla u_h \cdot \mathbf{n}] \right\|_{L^2(\partial K)}^2.$$

This provides us with an upper bound for the residual in terms of η^I

$$R(u - u_h) \leq c \left(\sum_{K \in \mathcal{T}_h} (\eta_K^I)^2 \right)^{\frac{1}{2}} \|\nabla (u - u_h)\|_{L^2(\Omega)}. \quad (4.37)$$

Next, we find a lower bound of the residual throughout its integral re-expression in (4.34) in order to investigate the relationship between the error estimate and the $H^1(\Omega)$ error semi-

norm.

Another expression for the residual can be obtained by adding and subtracting $\text{cof}(\mathbf{D}^2 u) \nabla u_h$ in the integral (4.34) and proceeding as following

$$\begin{aligned} \frac{1}{2} \int_{\Omega} (\text{cof}(\mathbf{D}^2 u) \nabla u - \text{cof}(\mathbf{D}_h^2 u_h) \nabla u_h) \cdot \nabla (u - u_h) \, d\mathbf{x} &= \frac{1}{2} \int_{\Omega} \text{cof}(\mathbf{D}^2 u) \nabla (u - u_h) \cdot \nabla (u - u_h) \\ &\quad + \frac{1}{2} \int_{\Omega} (\text{cof}(\mathbf{D}^2 u) - \text{cof}(\mathbf{D}_h^2 u_h)) \nabla u_h \cdot \nabla (u - u_h) \, d\mathbf{x}. \end{aligned}$$

A lower bound for this term can be obtained as a sum of two lower bounds of the right terms. Assuming that u is strictly convex we obtain

$$\begin{aligned} \frac{1}{2} \int_{\Omega} (\text{cof}(\mathbf{D}^2 u) \nabla u - \text{cof}(\mathbf{D}_h^2 u_h) \nabla u_h) \cdot \nabla (u - u_h) \, d\mathbf{x} &\geq \frac{\lambda_1}{2} \|\nabla (u - u_h)\|_{L^2(\Omega)}^2 \\ &\quad - \frac{1}{2} \|(\text{cof}(\mathbf{D}^2 u) - \text{cof}(\mathbf{D}_h^2 u_h)) \nabla u_h\|_{L^2(\Omega)} \|\nabla (u - u_h)\|_{L^2(\Omega)}, \end{aligned}$$

where λ_1 is the smallest eigenvalue of $\text{cof}(\mathbf{D}^2 u)$. Last, we plug into the above inequality expression (4.34) and (4.37) to get

$$\frac{\lambda_1}{2} \|\nabla (u - u_h)\|_{L^2(\Omega)} \leq c \left(\sum_{K \in \mathcal{T}_h} (\eta_K^I)^2 \right)^{\frac{1}{2}} + \frac{1}{2} \|(\text{cof}(\mathbf{D}^2 u) - \text{cof}(\mathbf{D}_h^2 u_h)) \nabla u_h\|_{L^2(\Omega)}, \quad (4.38)$$

The decision of mesh refinement in the adaptive algorithm will be based on the first term of (4.38). In [Picasso et al., 2011], the authors consider a $\mathbb{P}_1$ finite elements method and they compute the second derivatives similar to this setting. They show that they can obtain an order of convergence $O(h)$ for the H^2 error semi-norm for various mesh types. Therefore, the second term in (4.38) that involves the cofactor of the Hessian is expected to have rate of convergence $O(h)$. However, in special cases it can be $O(h^2)$ (see [Picasso et al., 2011, Remark 2.1]) then the left term in (4.38) could become negligible.

4.7 General Algorithm and Numerical Experiments

In this section, we present different test cases in order to examine the efficiency of the estimator η^I . The examples we consider here have already been presented in the previous section, therefore, we focus on interpreting the numerical results. For all test cases, the computational domain is the unit square $\Omega = (0, 1)^2$. For the PMA solver, the initial time step is $\Delta t = 5 \times 10^{-4}$. We decrease this timestep in the Newton method if the number of iterations exceed 100, and we increase it if the number of iterations is below 5. For the LSMA solver, ω is initially chosen to be one and gradually increases to two. For both solvers the parameter C is set to one.

We initially consider a structured asymmetric mesh of size $h_K = 1/20$, then we perform the first mesh refinement when $\|u_h^{n+1} - u_h^n\|_{L_2(\Omega)} \leq \nu$, where $\nu = 5 \times 10^{-4}$ for PMA and 5×10^{-5}

for LSMA. Both solvers reach this condition in 50 to 100 iterations or timesteps. Then, we adapt the mesh every 50 iterations (timesteps) or if the previous condition is satisfied. The algorithm stops when reaches 800 timesteps for PMA or 500 iterations for LSMA. The adaptive strategy will have to be improved in a future work, but the aim of this part is to investigate the performance of the estimator η^I .

We define the second term of (4.38)

$$\gamma = \frac{1}{2} \| (\text{cof}(\mathbf{D}^2 u) - \text{cof}(\mathbf{D}_h^2 u_h)) \nabla u_h \|_{L^2(\Omega)}.$$

For the sequel, we define as $h_{\min} = \min_{K \in \mathcal{T}_h} h_K$ and $h_{\max} = \max_{K \in \mathcal{T}_h} h_K$. In order to prevent the adaptive algorithm to infinitely refine, we set $(h_{\max}/h_{\min}) \leq 40$.

4.7.1 Smooth exponential example

First we consider a smooth exponential example

$$\begin{cases} \det \mathbf{D}^2 u = (x + y + 1) e^{(x^2 + y^2)} & \text{in } \Omega, \\ u = e^{\frac{1}{2}(x^2 + y^2)} & \text{on } \partial\Omega. \end{cases}$$

The convex solution u is defined by $u(x, y) = e^{\frac{1}{2}(x^2 + y^2)}$, $\forall (x, y) \in \Omega$.

In this example we use a triangular unstructured mesh and we vary h_K uniformly. We do not consider a mesh refinement, instead our stopping criterion is $\|u_h^{n+1} - u_h^n\|_{L_2(\Omega)} \leq 10^{-7}$, for both solvers. The purpose of this example is to verify that the error estimator η^I works on smooth solutions and unstructured refined mesh. Table 4.13 shows that the H^2 error norm, the estimations γ , and the error estimator η^I converge with order $O(h)$ for both solvers. The effectivity index settles at around 14. For both solvers we observe that the estimator η^I exhibits similar behavior.

4.7.2 Non-smooth examples

The first non-smooth problem reads as

$$\begin{cases} \det \mathbf{D}^2 u = \frac{R^2}{(R^2 - (x - 0.5)^2 - (y - 0.5)^2)^2} & \text{in } \Omega, \\ u = -\sqrt{R^2 - (x - 0.5)^2 - (y - 0.5)^2} & \text{on } \partial\Omega, \end{cases} \quad (4.39)$$

where $R = 1/\sqrt{2}$. The exact solution u of problem (4.39) is given by

$$u(x, y) = -\sqrt{R^2 - (x - 0.5)^2 - (y - 0.5)^2}, \quad \forall (x, y) \in \bar{\Omega},$$

Note that the solution u is smooth in $\bar{\Omega}$ but ∇u is discontinuous in the four corners of $\bar{\Omega}$.

4.7. General Algorithm and Numerical Experiments

Table 4.13 – Smooth exponential example with uniformly refined mesh. Estimated errors of $u - u_h$ in H^1 error norm, values of the estimator η^I and the corresponding convergence orders for various h_K . Number of timesteps or iterations. Top table: Numerical results obtained with PMA solver; Bottom table: Numerical results obtained with LSMA solver .

PMA	h_K	η^I		$\frac{\eta^I}{ u - u_h _{H^1(\Omega)}}$	$ u - u_h _{H^2(\Omega)}$		γ		Timesteps
	0.03125	6.29e-01	-	8.51	7.17e-01	-	4.81e-01	-	150
	0.01561	3.33e-01	0.91	11.40	4.74e-01	0.59	3.42e-01	0.49	247
	0.01035	2.18e-01	1.04	12.72	3.06e-01	1.07	2.26e-01	1.02	378
LSMA	0.00776	1.63e-01	1.01	13.60	2.30e-01	0.99	1.69e-01	1.01	379
	h_K	η^I		$\frac{\eta^I}{ u - u_h _{H^1(\Omega)}}$	$ u - u_h _{H^2(\Omega)}$		γ		Iterations
	0.03125	4.90e-01	-	9.73	5.96e-01	-	3.21e-01		15
	0.01561	2.77e-01	0.82	12.55	3.44e-01	0.79	2.18e-01	0.56	16
	0.01035	1.84e-01	1.00	13.43	1.91e-01	1.45	1.39e-01	1.10	16
	0.00776	1.38e-01	1.00	14.07	1.47e-01	0.91	1.05e-01	0.97	15

According to the regularity of the solution, the adaptive algorithm is supposed to track the discontinuities and refine the mesh in the corners. This is indeed observed in Figure 4.10 which illustrates the refined meshes for various TOL and for both solvers.

Numerical results obtained by both solvers are displayed in Table 4.14. When TOL decreases, h_{min} , h_{max} and the L^2 error norm decrease while the number of elements and nodes increase for both solvers. Unlike LSMA, for PMA, the estimator η_K^I and the H^1 error norm do not decrease for decreasing TOL and L^2 error norm. More specifically, for LSMA if we estimate the effectivity index $\eta^I/|u - u_h|_{H^1}$, we get a value close to 10 for all TOL . For $TOL = 0.25$, we can observe that for PMA the number of elements and nodes are larger than LSMA.

Another non-smooth problem that we consider is

$$\begin{cases} \det \mathbf{D}^2 u = 1 & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \end{cases} \quad (4.40)$$

Despite the smooth data, as highlighted in the previous section, this problem does not admit a classical solution but only a generalized one. In this example, as mentioned in [Dean and Glowinski, 2006], the main difficulties occur nearby the boundary of the domain. This is confirmed in Figure 4.11, where we display the refined meshes for both solvers and different TOL . In this figure, we see that for a tolerance level smaller than 0.5, the algorithm successfully refines the mesh around the boundary and keeps a coarser mesh in the center of the domain.

Table 4.15 shows numerical results obtained using PMA and LSMA solvers. Observations made about h_{min} , h_{max} , the number of elements and nodes holds in this setting. The estimator η^I does not decreases for decreasing TOL . One possible explanation could be that the bound we

Chapter 4. Numerical Approximation of a 2D Parabolic Monge-Ampère Equation

Table 4.14 – Non-smooth example with data $f(x, y) = \frac{R^2}{(R^2 - (x-0.5)^2 - (y-0.5)^2)^2}$, $g(x, y) = -\sqrt{R^2 - (x-0.5)^2 - (y-0.5)^2}$ and $R = \frac{1}{\sqrt{2}}$. Convergence behavior of the algorithm for various values of parameter TOL . The columns contain the final minimal and maximal mesh size, the final numbers of elements and nodes, the value of the estimator, the H^1 and L^2 error norms. Top table: Results obtained with PMA solver after 800 timesteps. Bottom table: Results obtained with LSMA solver after 500 iterations.

PMA	TOL	h_{min}	h_{max}	# elem	# nodes	η^I	$\ \mathbf{u} - \mathbf{u}_h\ _{H^1}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$
	1.0	6.65e-03	1.43e-01	916	523	0.36e+01	1.32e-01	4.41e-03
	0.5	3.00e-03	9.60e-02	2949	1612	0.65e+01	9.30e-02	1.23e-03
	0.25	2.03e-03	5.57e-02	9217	4889	1.18e+01	1.03e-01	5.75e-04
LSMA	TOL	h_{min}	h_{max}	# elem	# nodes	η^I	$\ \mathbf{u} - \mathbf{u}_h\ _{H^1}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$
	1.0	4.81e-03	1.89e-01	825	471	1.31e+00	1.23e-01	2.21e-02
	0.5	3.41e-03	9.33e-02	2565	1399	0.92e+00	9.32e-02	6.49e-03
	0.25	1.74e-03	6.41e-02	7281	3862	0.81e+00	8.84e-02	3.48e-03

set for refinement $\frac{h_{max}}{h_{min}} \leq 40$ prevents η^I from decreasing. For the minimal solution presented in the table, we see that it alligns with the discussion in subsection 4.5.2

Table 4.15 – Non-smooth example with data $f(x, y) = 1$ and $g(x, y) = 0$. Convergence behavior of the algorithm for various values of parameter TOL . The columns contain the final minimal and maximal mesh size, the final numbers of elements and nodes, the value of the estimator, and the minimal solution of u_h error norm. Top table: Results obtained with PMA solver after 800 timesteps. Bottom table: Results obtained with LSMA solver after 500 iterations.

PMA	TOL	h_{min}	h_{max}	# elem	# nodes	η^I	$\ u_h\ _{L^\infty(\Omega)}$
	1.0	1.31e-01	3.16e-01	46	30	4.41e-01	-0.168726
	0.5	2.89e-03	8.61e-02	4454	2496	2.82e-01	-0.173368
	0.25	2.76e-03	5.40e-02	9916	5279	5.83e-01	-0.179461
LSMA	TOL	h_{min}	h_{max}	# elem	# nodes	η^I	$\ u_h\ _{L^\infty(\Omega)}$
	1.0	7.13e-02	2.55e-01	110	69	4.02e-01	-0.166722
	0.5	1.03e-02	5.90e-01	1189	663	1.96e-01	-0.167842
	0.25	1.33e-03	7.41e-02	5516	3059	2.72e-01	-0.174659

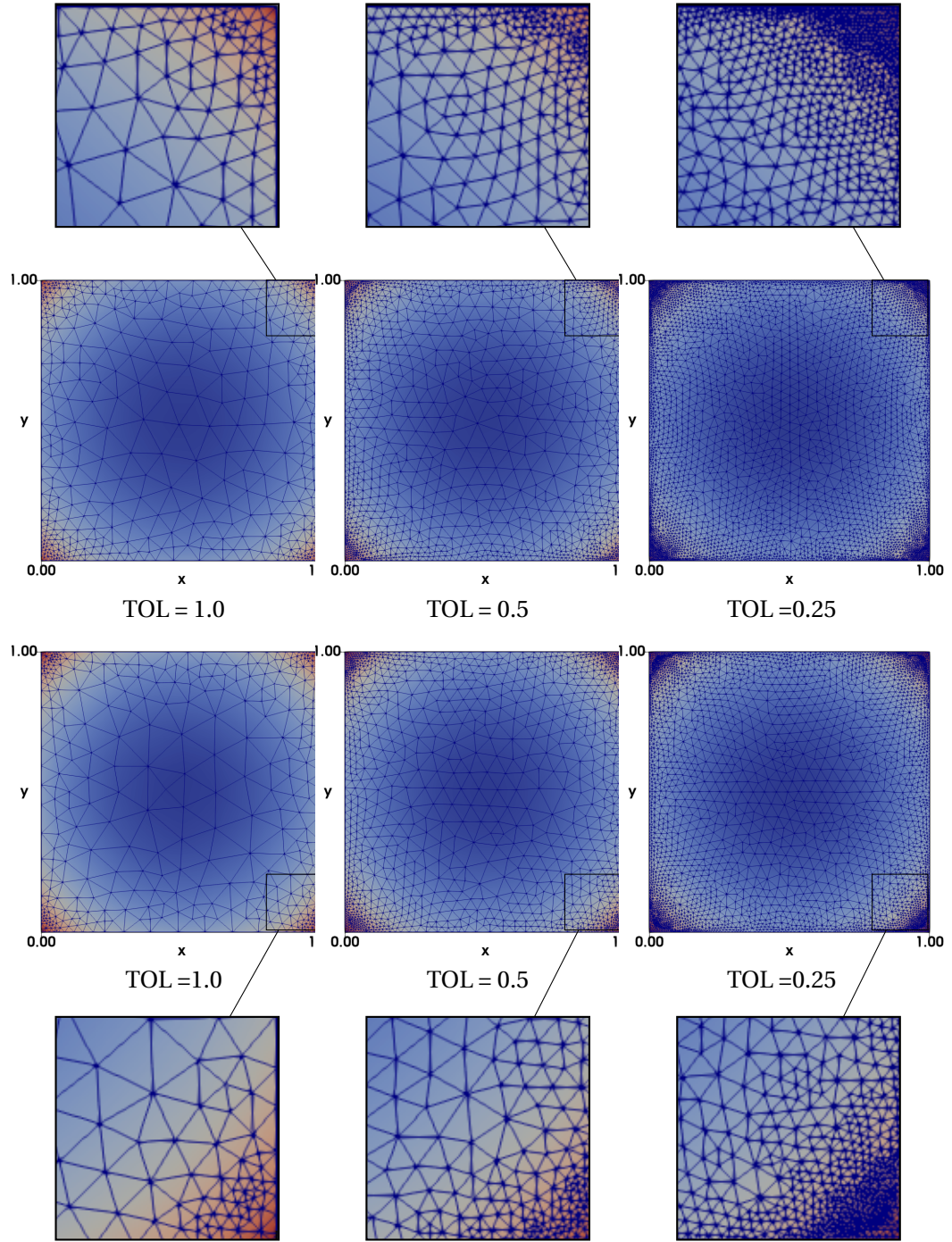


Figure 4.10 – Non-smooth example with data $f(x, y) = \frac{R^2}{(R^2 - (x-0.5)^2 - (y-0.5)^2)^2}$, $g(x, y) = -\sqrt{R^2 - (x-0.5)^2 - (y-0.5)^2}$ and $R = \frac{1}{\sqrt{2}}$. Graphs of the final adapted mesh and zoomed in parts of the corresponding graphs for various values of TOL . Top row: Graphs obtained with PMA solver after 800 timesteps. Bottom row: Graphs obtained with LSMA solver after 500 iterations.

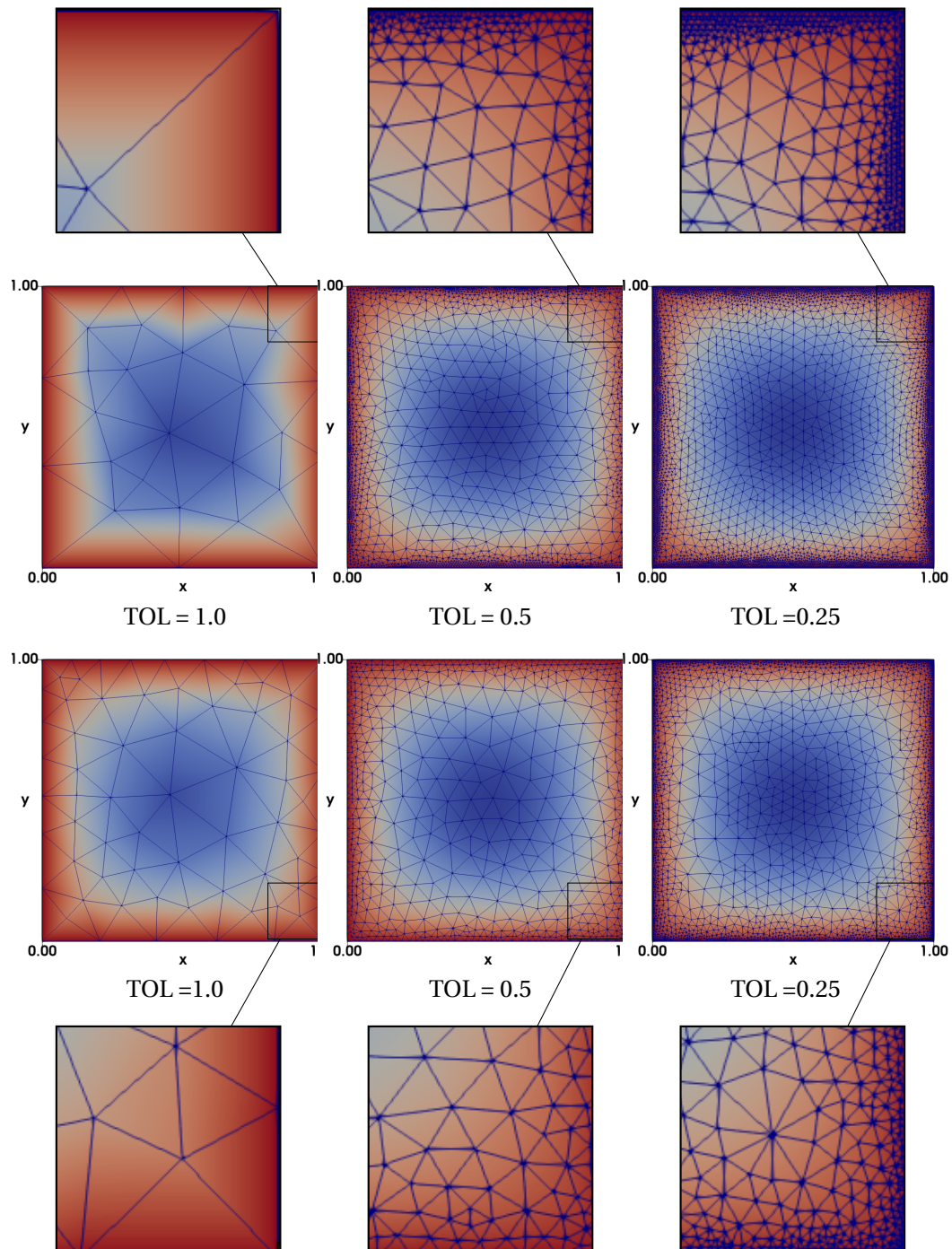


Figure 4.11 – Non-smooth example with data $f(x, y) = 1$ and $g(x, y) = 0$. Graphs of the final adapted mesh and zoomed in parts of the corresponding graphs for various values of TOL . Top row: Graphs obtained with PMA solver after 800 timesteps. Bottom row: Graphs obtained with LSMA solver after 500 iterations.

4.7.3 Examples with data involving a Dirac function

The first problem with a Dirac function on the right hand side reads as

$$\begin{cases} \det \mathbf{D}^2 u = \pi \delta_{(1/2, 1/2)} & \text{in } \Omega, \\ u = \sqrt{(x-0.5)^2 + (y-0.5)^2} & \text{on } \partial\Omega, \end{cases} \quad (4.41)$$

The exact solution u of problem (4.41) is defined by

$$u(x, y) = \sqrt{(x-0.5)^2 + (y-0.5)^2} \quad \forall (x, y) \in \bar{\Omega},$$

As in subsection 4.5.4 for both solvers we approximate the Dirac function $\delta_{(\alpha, \beta)}$ by

$$f_\varepsilon^{(\alpha, \beta)}(x, y) = \frac{\varepsilon^2}{\pi \left(\varepsilon^2 + (x - \alpha)^2 + (y - \beta)^2 \right)^2},$$

where ε is a small positive value. Therefore, we approximate $\pi \delta_{(1/2, 1/2)}$ by $f_\varepsilon^{(1/2, 1/2)}(x, y)$. To estimate the error norms, we assume that $\lim_{h \rightarrow 0} \lim_{\varepsilon \rightarrow 0} u_\varepsilon^\eta = u$. Moreover, as in subsection 4.5.4, we set $\varepsilon = 10^{-3/2}$.

In this experiment, we expect the adaptive algorithm to refine the mesh around $(0.5, 0.5)$, i.e., where the Dirac function occurs. Figure 4.12 shows indeed the refined meshes for both solvers and different TOL .

Table 4.16 shows numerical results obtained by PMA and LSMA solvers. For both solvers, when TOL decreases, h_{min} , h_{max} , η_K^I , L^2 error norm decrease, and the number of elements and nodes increase. For PMA, the number of elements and nodes is larger than LSMA. — Figure 4.13 displays the refined meshes for LSMA with fixed $TOL = 0.25$ and different values of ε . In this figure, we observe that the smaller the parameter ε is, the larger the number of elements. More precisely, Table 4.17 shows that as ε decreases, the h_{min} , h_{max} , L^2 error norm decreases and the number of elements and nodes increases. As ε gets smaller, the problem becomes more challenging, and therefore, η_K^I increases.

In this problem we consider two Dirac function on the right hand side

$$\begin{cases} \det \mathbf{D}^2 u = \frac{\pi}{2} \delta_{(1/4, 1/2)} + \frac{\pi}{2} \delta_{(3/4, 1/2)} & \text{in } \Omega, \\ u = g & \text{on } \partial\Omega, \end{cases} \quad (4.42)$$

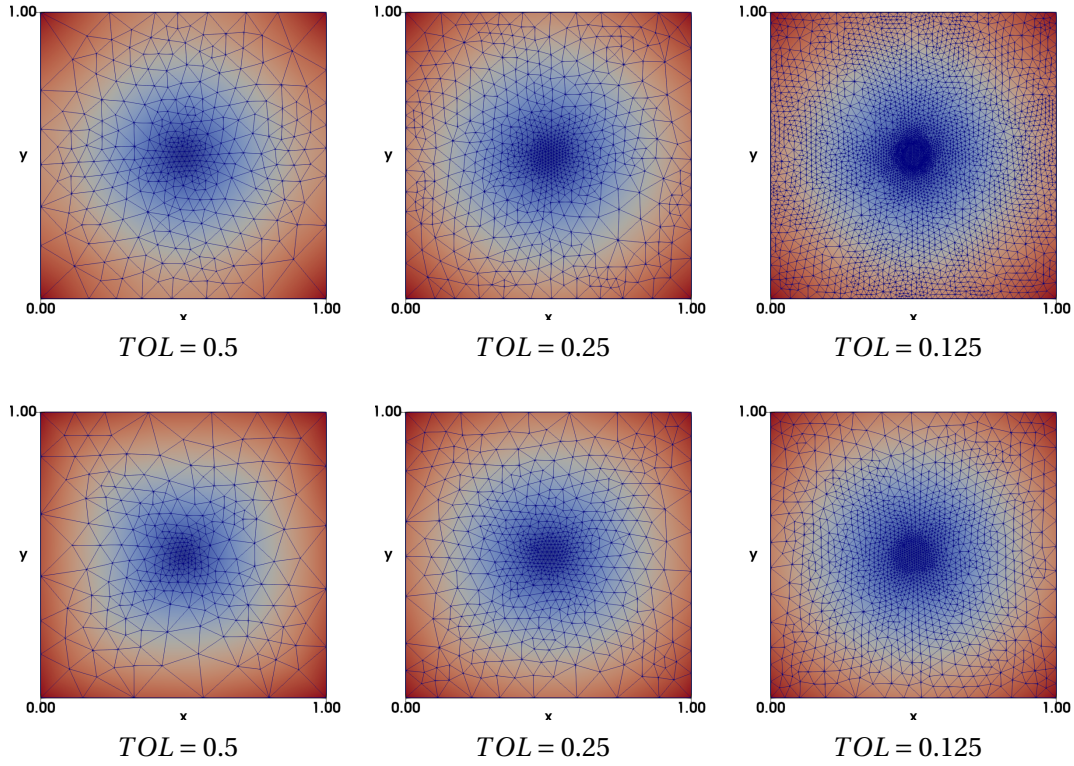


Figure 4.12 – Examples with Dirac function for data $f(x, y) = \pi\delta_{(1/2, 1/2)}$ and $g(x, y) = \sqrt{(x - 0.5)^2 + (y - 0.5)^2}$. Graphs of the final adapted mesh for various values of TOL and fixed $\varepsilon = 10^{-3/2}$. Top row: Graphs obtained with PMA solver after 800 timesteps. Bottom row: Graphs obtained with LSMA solver after 500 iterations.

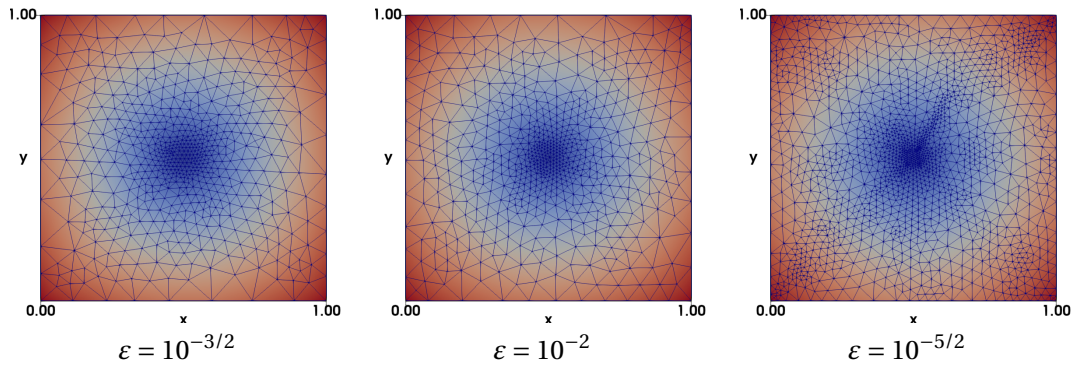


Figure 4.13 – Examples with Dirac function for data $f(x, y) = \pi\delta_{(1/2, 1/2)}$ and $g(x, y) = \sqrt{(x - 0.5)^2 + (y - 0.5)^2}$. Graphs of the final adapted mesh for various values of ε and fixed $TOL = 0.25$. The graphs obtained with LSMA solver after 500 iterations.

4.7. General Algorithm and Numerical Experiments

Table 4.16 – Examples with Dirac function for data $f(x, y) = \pi\delta_{(1/2, 1/2)}$ and $g(x, y) = \sqrt{(x - 0.5)^2 + (y - 0.5)^2}$. Convergence behavior of the algorithm for various values of parameter TOL and fixed $\varepsilon = 10^{-3/2}$. The columns contain the final minimal and maximal mesh size, the final numbers of elements and nodes, the value of the estimator, and L^2 error norms. Top table: Results obtained with PMA solver after 800 timesteps. Bottom table: Results obtained with LSMA solver after 500 iterations.

PMA	TOL	h_{min}	h_{max}	# elem	# nodes	η^I	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$
	0.5	1.21e-02	1.90e-01	627	328	7.24e-01	3.71e-02
	0.25	7.85e-03	1.23e-01	1331	739	5.23e-01	1.37e-02
	0.125	3.58e-03	5.57e-02	3425	1748	3.50e-01	6.68e-03
LSMA	TOL	h_{min}	h_{max}	# elem	# nodes	η^I	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$
	0.5	1.36e-02	1.77e-01	509	269	8.71e-01	7.73e-02
	0.25	8.41e-03	1.24e-01	1128	582	5.50e-01	3.84e-02
	0.125	4.89e-03	7.78e-02	2920	1489	3.70e-01	1.92e-02

Table 4.17 – Examples with Dirac function for data $f(x, y) = \pi\delta_{(1/2, 1/2)}$ and $g(x, y) = \sqrt{(x - 0.5)^2 + (y - 0.5)^2}$. Convergence behavior of the algorithm for various values of parameter $\varepsilon = 10^{-3/2}$ and fixed $TOL = 0.25$. The columns contain the final minimal and maximal mesh size, the final numbers of elements and nodes, the value of the estimator, and L^2 error norms. The results obtained with LSMA solver after 500 iterations.

LSMA	ε^2	h_{min}	h_{max}	# elem	# nodes	η^I	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$
	1e-03	8.41e-03	1.24e-01	1128	582	5.50e-01	3.84e-02
	1e-04	7.93e-03	1.40e-01	1224	636	0.12e-01	2.29e-02
	1e-05	4.19e-03	7.02e-02	2986	1544	0.15e-01	1.29e-02

where

$$g(x, y) = \begin{cases} |y - 0.5| & \text{if } 1/4 < x < 3/4, \\ \min\left(\sqrt{(x - 1/4)^2 + (y - 0.5)^2}, \sqrt{(x - 3/4)^2 + (y - 0.5)^2}\right) & \text{Otherwise.} \end{cases} \quad (4.43)$$

The exact solution is given as $u = g$ in Ω .

We approximate $\frac{\pi}{2}\delta_{(1/4, 1/2)} + \frac{\pi}{2}\delta_{(3/4, 1/2)}$ by $\frac{\pi}{2}f_\varepsilon^{(1/4, 1/2)}(x, y) + \frac{\pi}{2}f_\varepsilon^{(3/4, 1/2)}(x, y)$. To estimate the error norms, we assume that $\lim_{h \rightarrow 0} \lim_{\varepsilon \rightarrow 0} u_\varepsilon^\eta = u$. Moreover, as before we set $\varepsilon = 10^{-3/2}$.

In this experiment, we expect the adaptive algorithm to refine the mesh around the neighbor-

hood of the line between $(1/4, 1/2)$ and $(3/4, 1/2)$. Figure 4.12 displays the refined meshes for both solvers and different TOL , which shows the expected behavior.

Table 4.16 shows numerical results obtained by PMA and LSMA solvers. For the PMA, although the L^2 error norm is accurate to the order of 10^{-3} , it does not decrease with TOL . On the other hand, for LSMA, the L^2 error norm is of order 10^{-2} , and it decreases monotonically with TOL .

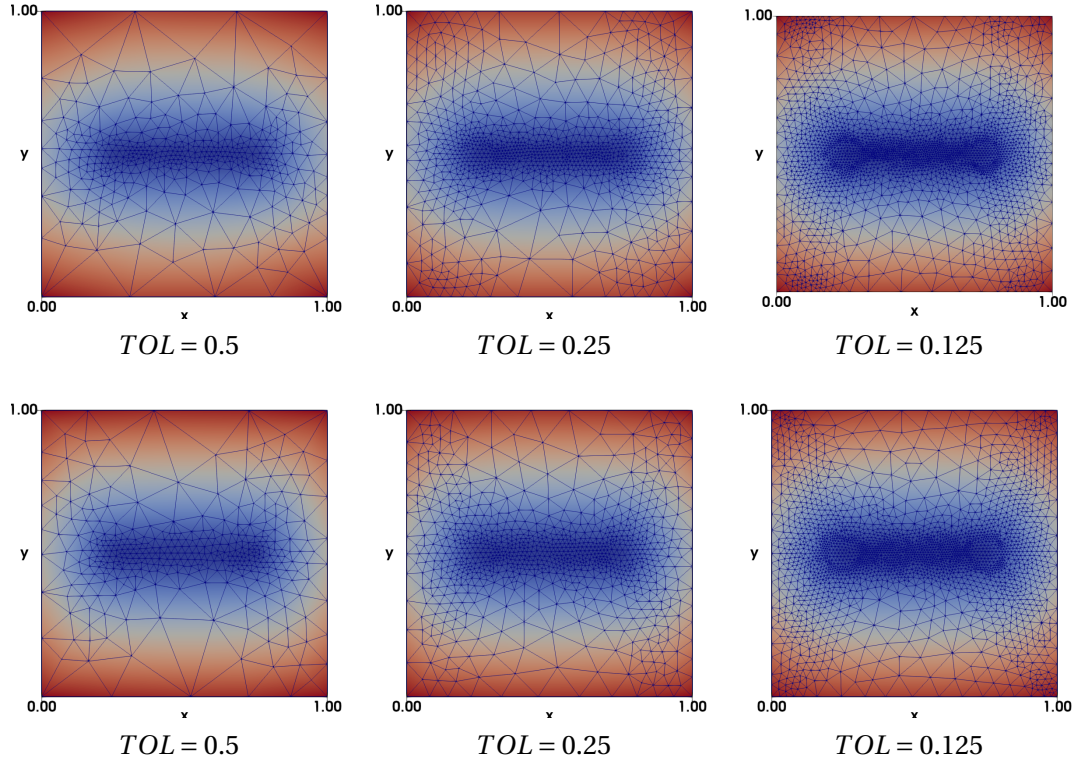


Figure 4.14 – Examples with Dirac function for data $f(x, y) = \frac{\pi}{2}\delta_{(1/4, 1/2)} + \frac{\pi}{2}\delta_{(3/4, 1/2)}$ and $g(x, y)$ is given in (4.43). Graphs of the final adapted mesh for various values of TOL and fixed $\varepsilon = 10^{-3/2}$. Top row: Graphs obtained with PMA solver after 800 timesteps. Bottom row: Graphs obtained with LSMA solver after 500 iterations.

4.7. General Algorithm and Numerical Experiments

Table 4.18 – Examples with Dirac function for data $f(x, y) = \frac{\pi}{2}\delta_{(1/4, 1/2)} + \frac{\pi}{2}\delta_{(3/4, 1/2)}$ and $g(x, y)$ is given in (4.43). Convergence behavior of the algorithm for various values of parameter TOL and fixed $\varepsilon = 10^{-3/2}$. The columns contain the final minimal and maximal mesh size, the final numbers of elements and nodes, the value of the estimator, and L^2 error norms. Top table: Results obtained with PMA solver after 800 timesteps. Bottom table: Results obtained with LSMA solver after 500 iterations.

PMA	TOL	h_{min}	h_{max}	# elem	# nodes	η^I	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$
	0.5	1.11e-02	3.27e-01	722	373	7.72e-01	7.75e-03
	0.25	5.96e-03	2.00e-01	1886	969	4.85e-01	7.71e-03
	0.125	3.96e-03	1.26e-02	4664	2371	2.85e-01	8.42e-03
LSMA	TOL	h_{min}	h_{max}	# elem	# nodes	η^I	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$
	0.5	1.23e-02	2.17e-01	687	356	5.98e-01	3.84e-02
	0.25	8.14e-03	2.25e-01	1617	828	4.82e-01	2.08e-02
	0.125	4.89e-03	7.78e-02	2920	1489	3.70e-01	1.92e-02

5 Conclusion

In this thesis, numerical methods for solutions of first and second-order fully nonlinear equations have been presented. We have discussed an operator-splitting/ finite element method for the numerical solution of the Dirichlet problem for orthogonal maps, and we have extended it by introducing an anisotropic space adaptivity method. Then, we have derived least-squares/relaxation methods for the numerical solutions of Jacobian equation/inequality, and the three dimensional Monge-Ampère equation. We have also developed a second order time integration method for the approximation of a parabolic and elliptic 2D Monge-Ampère equation. Lastly, we have introduced an isotropic adaptive method for the two dimensional elliptic Monge-Ampère equation.

In Chapter 1, we tackled the orthogonal map equation and we proposed an operator-splitting/ finite element method for the numerical solution of the Dirichlet problem. This method is based on a variational principle, the introduction of the associated flow problem, and a time-stepping splitting algorithm. Results show the robustness and the flexibility of this method and its ability to approximate solutions with line singularities on convex domains, with convergence of order one for the $L^2(\Omega)$ norm of the approximation error. Then the method has been extended by proposing an anisotropic adaptive mesh algorithm that allows to track the singularities of the gradient of the solution more accurately. Numerical experiments have shown the robustness of the derived estimator and the adaptive algorithm.

In Chapter 2, we focused on the numerical approximation of the solutions of the prescribed Jacobian equation and inequality. The proposed method relies on a least-squares reformulation of the original problem and on a relaxation algorithm. The method is based on variational approaches where the use of relaxation algorithm allows the decoupling of the least-squares problem into local nonlinear and to variational problems. We have presented the implementation in the finite element space and numerical examples to validate the method.

In Chapter 3, a least-squares/relaxation finite element method also has been introduced to solve the Dirichlet problem for the three-dimensional elliptic Monge-Ampère equation. The method resembles the least squares framework that has been developed for several fully

nonlinear equations, and in Chapter 2. In this case, new solvers have been derived to provide solutions for the local nonlinear and variational problems. Numerical results have been validated by various test cases using $\mathbb{P}_1$ and $\mathbb{Q}_1$ polynomials. The method achieves nearly optimal orders for the $L^2(\Omega)$ and $H^1(\Omega)$ error norms when smooth solutions are approximated.

In Chapter 4, we presented a numerical method for the solution of a parabolic 2D Monge-Ampère equation. Moreover, we have shown the efficiency of the method in capturing stationary solutions, such as the solutions of the elliptic Monge-Ampère equation. In the numerical examples, when tested to smooth solutions, the method had nearly optimal orders for the $L^2(\Omega)$ and $H^1(\Omega)$ error norms. The method proved to be also robust at approximating non-smooth solutions. We have derived a heuristic error estimate, and we have designed an adaptive algorithm for the two-dimensional elliptic Monge-Ampère equation. Numerical experiments confirmed the robustness of the estimators and adaptive algorithm by using two different solvers.

It is possible to extend this work in different directions by working further on the following subtopics. For example, the operator-splitting/finite element method that used to solve the orthogonal maps can be broadened to include the rigid maps problems. The prescribed Jacobian equation can be extended to include a transformation map of the form $\mathbf{u} : \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}^3$. This will allow us to tackle problems in optimal transportation in 3 dimensional setting. The parabolic and elliptic Monge-Ampère equation can also be extended to solve not only applications in optimal transport, but also financial ones. Last, the derived error estimates and the adaptive algorithms that are essentially designed for orthogonal maps and Monge-Ampère equation can be improved and extended to include other fully nonlinear equations such as Eikonal and Pucci equations.

Bibliography

- Abel, Z., Demaine, E. D., Demaine, M. L., Eppstein, D., Lubiw, A., and Uehara, R. (2018). Flat foldings of plane graphs with prescribed angles and edge lengths. *Journal of Computational Geometry*, 9(1):74–93.
- Abugirda, H. and Katzourakis, N. (2018). On the well-posedness of global fully nonlinear first order elliptic systems. *Advances in Nonlinear Analysis*, 7(2):139–148.
- Afkham, B. M., Bhatt, A., Haasdonk, B., and Hesthaven, J. S. (2018). Symplectic model-reduction with a weighted inner product. *arXiv preprint arXiv:1803.07799*.
- Akrivis, G. (2015). Stability of implicit-explicit backward difference formulas for nonlinear parabolic equations. *SIAM Journal on Numerical Analysis*, 53(1):464–484.
- Akrivis, G. and Lubich, C. (2015). Fully implicit, linearly implicit and implicit–explicit backward difference formulae for quasi-linear parabolic equations. *Numerische Mathematik*, 131(4):713–735.
- Akrivis, G., Makridakis, C., and Nochetto, R. H. (2009). Optimal order a posteriori error estimates for a class of Runge–Kutta and Galerkin methods. *Numerische Mathematik*, 114(1):133.
- Aleksandrov, A. D. (1958). Dirichlet’s problem for the equation $\text{Det}\|z_{ij}\| = \varphi(z_1, \dots, z_n, z, x_1, \dots, x_n)$. I. *Vestnik Leningradskogo Universiteta. Matematika, Mekhanika, Astronomiya*, 13(1):5–24.
- Ambrosio, L., Caffarelli, L., Salsa, S., Brenier, Y., Buttazzo, G., and Villani, C. (2003). *Optimal Transportation and Applications*. Springer-Verlag Berlin Heidelberg.
- Apel, T., Nicaise, S., and Sirch, D. (2011). A posteriori error estimation of residual type for anisotropic diffusion–convection–reaction problems. *Journal of Computational and Applied Mathematics*, 235(8):2805–2820.
- Armstrong, S. N., Sirakov, B., and Smart, C. K. (2012). Singular solutions of fully nonlinear elliptic equations and applications. *Archive for Rational Mechanics and Analysis*, 205(2):345–394.

Bibliography

- Avinyó, A., Solà-Morales, J., and València, M. (2012). A singular initial value problem to construct density-equalizing maps. *Journal of Dynamics and Differential Equations*, 24(1):51–59.
- Awanou, G. (2014). Standard finite elements for the numerical resolution of the elliptic Monge-Ampère equation: classical solutions. *IMA Journal of Numerical Analysis*, 35(3):1150–1166.
- Awanou, G. (2016). On standard finite difference discretizations of the elliptic Monge-Ampère equation. *Journal of Scientific Computing*, 69(2):892–904.
- Awanou, G. (2020). A two-grid method for the $\mathcal{C}^0$ interior penalty discretization of the Monge-Ampère equation. *Journal of Computational Mathematics*, 38(4):547–564.
- Awanou, G. and Leopold, M. M. (2015). A variational method for computing numerical solutions of the Monge-Ampère equation. *arXiv preprint arXiv:1510.00453*.
- Babuška, I., Durán, R., and Rodríguez, R. (1992). Analysis of the efficiency of an a posteriori error estimator for linear triangular finite elements. *SIAM Journal on Numerical Analysis*, 29(4):947–964.
- Babuška, I. and Rheinboldt, W. C. (1978). A-posteriori error estimates for the finite element method. *International Journal for Numerical Methods in Engineering*, 12(10):1597–1615.
- Ball, J. M. and James, R. D. (1992). Proposed experimental tests of a theory of fine microstructure and the two-well problem. *Philosophical Transactions: Physical Sciences and Engineering*, 338(1650):389–450.
- Barles, G. and Jakobsen, E. R. (2002). On the convergence rate of approximation schemes for Hamilton-Jacobi-Bellman equations. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 36(1):33–54.
- Barles, G. and Souganidis, P. E. (1991). Convergence of approximation schemes for fully nonlinear equations. *Asymptotic Analysis*, 4(3):271–283.
- Bartels, S., Bonito, A., and Nochetto, R. H. (2017). Bilayer plates: Model reduction, Γ -convergent finite element approximation and discrete gradient flow. *Communications on Pure and Applied Mathematics*, 70(3):547–589.
- Basterrechea, S. and Dacorogna, B. (2014). Existence of solutions for Jacobian and Hessian equations under smallness assumptions. *Numerical Functional Analysis and Optimization*, 35(7-9):868–892.
- Benamou, J., Froese, B. D., and Oberman, A. M. (2014). Numerical solution of the optimal transportation problem using the Monge-Ampère equation. *Journal of Computational Physics*, 260:107–126.
- Benamou, J. D. and Brenier, Y. (2000). A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numerische Mathematik*, 84:375–393.

- Benamou, J.-D. and Froese, B. D. (2014). A viscosity framework for computing Pogorelov solutions of the Monge-Ampère equation. *arXiv preprint arXiv:1407.1300*.
- Benamou, J. D., Froese, B. D., and Oberman, A. M. (2010). Two numerical methods for the elliptic Monge-Ampère equation. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 44(4):737–758.
- Bern, M. W. and Hayes, B. (1996). The complexity of flat origami. In *SODA '96*.
- Birindelli, I. and Galise, G. (2019). The Dirichlet problem for fully nonlinear degenerate elliptic equations with a singular nonlinearity. *Calculus of Variations and Partial Differential Equations*, 58(5):180.
- Boehmer, K. (2008). On finite element methods for fully nonlinear elliptic equations of second order. *SIAM Journal on Numerical Analysis*, 46(3):1212–1249.
- Bonito, A., Guignard, D., Nochetto, R. H., and Yang, S. (2020). LDG approximation of large deformations of prestrained plates. *arXiv preprint arXiv:2011.01086*.
- Bonnans, J. F., Bonnet, G., and Mirebeau, J.-M. (2020). Monotone and second order consistent scheme for the two dimensional pucci equation. *hal archives hal-02383521*.
- Botsaris, C. A. (1978). A class of methods for unconstrained minimization based on stable numerical integration techniques. *Journal of Mathematical Analysis and Applications*, 63(3):729–749.
- Bourgault, Y., Picasso, M., Alauzet, F., and Loseille, A. (2009). On the use of anisotropic a posteriori error estimators for the adaptative solution of 3D inviscid compressible flows. *International Journal For Numerical Methods In Fluids*, 59:47–74.
- Brenner, S. C., Gudi, T., Neilan, M., and Sung, L.-Y. (2011). $\mathcal{C}^0$ penalty methods for the fully nonlinear Monge-Ampère equation. *Mathematics of Computation*, 80(276):1979–1995.
- Brenner, S. C. and Neilan, M. (2012). Finite element approximations of the three dimensional Monge-Ampère equation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46(5):979–1001.
- Budd, C., Huang, W., and Russell, R. (2009). Adaptivity with moving grids. *Acta Numerica*, 18:111–241.
- Budd, C. J. and Williams, J. F. (2006). Parabolic Monge-Ampère methods for blow-up problems in several spatial dimensions. *Journal of Physics A: Mathematical and General*, 39(19):5425–5444.
- Budd, C. J. and Williams, J. F. (2009). Moving mesh generation using the parabolic Monge-Ampère equation. *SIAM Journal on Numerical Analysis*, 31(5):3438–3465.

Bibliography

- Burago, D. and Kleiner, B. (1998). Separated nets in Euclidean space and Jacobians of bilipschitz maps. *Geometric and Functional Analysis*, 8(2):273–282.
- Busca, J., Esteban, M. J., and Quaas, A. (2005). Nonlinear eigenvalues and bifurcation problems for Pucci’s operators. *Annales de l’Institut Henri Poincaré (C) Non Linear Analysis*, 22(2):187–206.
- Böhmer, K. and Schaback, R. (2019). A meshfree method for solving the Monge–Ampère equation. *Numerical Algorithms*, 82(2):539–551.
- Caboussat, A. (2014). On the numerical solution of the Dirichlet problem for the elliptic (σ_2) equation. In Fitzgibbon, W., Kuznetsov, Y. A., Neittaanmäki, P., and Pironneau, O., editors, *Modeling, Simulation and Optimization for Science and Technology*, volume 34, page 23–40. Springer Netherlands.
- Caboussat, A. (2019). Least-squares/relaxation method for the numerical solution of a 2D Pucci’s equation. *Methods and Applications of Analysis*, 26(2):113–132.
- Caboussat, A. and Glowinski, R. (2008). A numerical method for a non-smooth advection-diffusion problem arising in sand mechanics. *Communications on Pure & Applied Analysis*, 8(1):161–178.
- Caboussat, A. and Glowinski, R. (2012). Regularization methods for the divergence equation $\nabla \cdot \mathbf{u} = f$. *Journal of Computational Mathematics*, 30(4):354–380.
- Caboussat, A. and Glowinski, R. (2015a). A numerical algorithm for a fully nonlinear PDE involving the Jacobian determinant. In *Numerical Mathematics and advanced applications - ENUMATH 2013*, volume 103 of *Lecture Notes in Computational Science and Engineering*, page 143–151. Springer.
- Caboussat, A. and Glowinski, R. (2015b). A penalty-regularization-operator splitting method for the numerical solution of a scalar Eikonal equation. *Chinese Annals of Mathematics Series B*, 36(5):659–688.
- Caboussat, A. and Glowinski, R. (2018). An alternating direction method of multipliers for the numerical solution of a fully nonlinear partial differential equation involving the Jacobian determinant. *SIAM Journal on Scientific Computing*, 40:A52–A80.
- Caboussat, A., Glowinski, R., and Gourzoulidis, D. (2018). A least-squares/relaxation method for the numerical solution of the three-dimensional elliptic Monge–Ampère equation. *Journal of Scientific Computing*, 77(1):1–26.
- Caboussat, A., Glowinski, R., Gourzoulidis, D., and Picasso, M. (2019). Numerical approximation of orthogonal maps. *SIAM Journal on Scientific Computing*, 41(6):B1341–B1367.
- Caboussat, A., Glowinski, R., and Pan, T. W. (2015). On the numerical solution of some Eikonal equations: An elliptic solver approach. *Chinese Annals of Mathematics Series B*, 36(5):689–702.

- Caboussat, A., Glowinski, R., and Sorensen, D. C. (2013). A least-squares method for the numerical solution of the Dirichlet problem for the elliptic Monge-Ampère equation in dimension two. *ESAIM: Control, Optimization and Calculus of Variations*, 19(3):780–810.
- Caboussat, A. and Gourzoulidis, D. (2019). A second order time integration method for the approximation of a parabolic 2d Monge-Ampère equation. In *Proceedings of the European numerical mathematics and advanced applications*.
- Caboussat, A., Gourzoulidis, D., and Picasso, M. (2020). An anisotropic adaptive method for the numerical approximation of orthogonal maps. *Journal of Computational and Applied Mathematics*, (submitted).
- Caffarelli, L., Crandall, M. G., Kocan, M., and Świąch, A. (1996). On viscosity solutions of fully nonlinear equations with measurable ingredients. *Communications on Pure and Applied Mathematics*, 49(4):365–398.
- Caffarelli, L. and Milman, M. (1999a). *Monge Ampere Equation: Applications to Geometry and Optimization: Applications to Geometry and Optimization*. American Mathematical Society.
- Caffarelli, L., Nirenberg, L., and Spruck, J. (2014). The Dirichlet problem for nonlinear second-order elliptic equations I. Monge-Ampère equation. *Communications on Pure and Applied Mathematics*, 37:369–402.
- Caffarelli, L. A. (1991). Some regularity properties of solutions of Monge-Ampère equation. *Communications on Pure and Applied Mathematics*, 44(8-9):965–969.
- Caffarelli, L. A. and Cabré, X. (1995). *Fully Nonlinear Elliptic Equations*. American Mathematical Society.
- Caffarelli, L. A. and Glowinski, R. (2008). Numerical solution of the Dirichlet problem for a Pucci equation in dimension two. Application to homogenization. *Journal of Numerical Mathematics*, 16(3):185–216.
- Caffarelli, L. A. and Milman, M., editors (1999b). *Monge-Ampère Equation: Application to Geometry and Optimization*. American Mathematical Society, Providence, RI.
- Caffarelli, L. A. and Stefanelli, U. (2008). A counterexample to $C^{2,1}$ regularity for parabolic fully nonlinear equations. *Communications in Partial Differential Equations*, 33(7):1216–1234.
- Calabi, E. (1958). Improper affine hyperspheres of convex type and a generalization of a theorem by k. Jörgens. *Michigan Mathematical Journal*, 5:105–126.
- Carlier, G. and Dacorogna, B. (2012). Solution of the Dirichlet problem for the prescribed Jacobian equation using the Monge-Ampère equation. *Comptes Rendus Mathematique*, 350(7):371–374.
- Carstensen, C. (2006). *Clément Interpolation and Its Role in Adaptive Finite Element Error Control*, page 27–43. Birkhäuser Basel, Basel.

Bibliography

- Chang, S.-Y. A., Han, Z.-C., and Yang, P. (2011). On the prescribing σ_2 curvature equation on $\mathbb{S}^4$. *Calculus of Variations and Partial Differential Equations*, 40(3):539–565.
- Charbonneau, A., Dossou, K., and Pierre, R. (1997). A residual-based a posteriori error estimator for the Ciarlet-Raviart formulation of the first biharmonic problem. *Numerical Methods for Partial Differential Equations*, 13(1):93–111.
- Ciarlet, P. G. (2002). *The Finite Element Method for Elliptic Problems*. Society for Industrial and Applied Mathematics.
- Clément, P. (1975). Approximation by finite element functions using local regularization. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 9(R2):77–84.
- Cossette, J.-F. and Smolarkiewicz, P. K. (2011). A Monge-Ampère enhancement for semi-Lagrangian methods. *Computers & Fluids*, 46(1):180–185.
- Crandall, M., Evans, L., and Lions, P.-L. (1984). Some properties of viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society*, 282(2):487–502.
- Csató, G., Dacorogna, B., and Kneuss, O. (2011). *The Pullback Equation for Differential Forms*. Birkhäuser Boston.
- Cupini, G., Dacorogna, B., and Kneuss, O. (2009). On the equation $\det \nabla u = f$ with no sign hypothesis. *Calculus of Variations and Partial Differential Equations*, 36:251–283.
- Dacorogna, B., Glowinski, R., Kuznetsov, Y., and Pan, T.-W. (2004). On a conjugate gradient/Newton/penalty method for the solution of obstacle problems. application to the solution of an Eikonal system with Dirichlet boundary conditions. In M. Křížek, P. Neittaanmäki, R. G. and Korotov, S., editors, *Conjugate Gradient Algorithms and Finite Element Methods*, page 263–283. Springer-Verlag, Berlin, Heidelberg.
- Dacorogna, B., Glowinski, R., and T.-W. Pan (2003). Numerical methods for the solution of a system of Eikonal equations with Dirichlet boundary conditions. *C. R. Acad. Sci. Paris, Sér. I*, 336:511–518.
- Dacorogna, B. and Marcellini, P. (1999). *Implicit Partial Differential Equations*. Birkhäuser, Basel.
- Dacorogna, B., Marcellini, P., and Paolini, E. (2008a). An explicit solution to a system of implicit differential equations. *Annales de l'Institut Henri Poincaré (C) Non Linear Analysis*, 25:163–171.
- Dacorogna, B., Marcellini, P., and Paolini, E. (2008b). Lipschitz-continuous local isometric immersions: rigid maps and origami. *Journal de Mathématiques Pures et Appliquées*, 90:66–81.

- Dacorogna, B., Marcellini, P., and Paolini, E. (2010a). Functions with orthogonal Hessian. *Differential and Integral Equations*, 23:51–60.
- Dacorogna, B., Marcellini, P., and Paolini, E. (2010b). Origami and partial differential equations. *Notices of the American Mathematical Society*, 57:598–606.
- Dacorogna, B., Marcellini, P., and Paolini, E. (2018). On the n -dimensional Dirichlet problem for isometric maps. *Journal of Functional Analysis*, 255:3274–3280.
- Dacorogna, B. and Moser, J. (1990). On a partial differential equation involving the Jacobian determinant. *Annales de l'I.H.P. Analyse non linéaire*, 7:1–26.
- Dairbekov, N., Alexandrov, A., Kutateladze, S., and Sossinsky, A. (2010). *Convex Polyhedra*. Springer Monographs in Mathematics. Springer Berlin Heidelberg.
- Davydov, O. and Saeed, A. (2013). Numerical solution of fully nonlinear elliptic equations by Böhmer's method. *Journal of Computational and Applied Mathematics*, 254:43–54.
- Davydov, O. and Saeed, A. (2017). C^1 quintic splines on domains enclosed by piecewise conics and numerical solution of fully nonlinear elliptic equations. *Applied Numerical Mathematics*, 116:172–183.
- De Philippis, G. and Figalli, A. (2013). $W^{2,1}$ regularity for solutions of the Monge–Ampère equation. *Inventiones mathematicae*, 192(1):55–69.
- Dean, E. J. and Glowinski, R. (2004). Numerical solution of the two-dimensional elliptic Monge–Ampère equation with Dirichlet boundary conditions: a least-squares approach. *C. R. Acad. Sci. Paris, Sér. I*, 339(12):887–892.
- Dean, E. J. and Glowinski, R. (2005). Numerical solution of a two-dimensional elliptic Pucci's equation with Dirichlet boundary conditions: a least-squares approach. *Comptes Rendus Mathématique*, 341:374–380.
- Dean, E. J. and Glowinski, R. (2006). Numerical methods for fully nonlinear elliptic equations of the Monge–Ampère type. *Computer Methods in Applied Mechanics and Engineering*, 195(13):1344–1386.
- Dean, E. J. and Glowinski, R. (2008). On the numerical solution of the elliptic Monge–Ampère equation in dimension two: A least-squares approach. In Glowinski, R. and Neittaanmäki, P., editors, *Partial Differential Equations: Modeling and Numerical Simulation*, volume 16 of *Computational Methods in Applied Sciences*, page 43–63. Springer.
- Dean, E. J., Glowinski, R., and Guidoboni, G. (2007). On the numerical simulation of Bingham visco-plastic flow: old and new results. *Journal of Non Newtonian Fluid Mechanics*, 142:36–62.

Bibliography

- Dean, E. J., Glowinski, R., and Trevas, D. (1996). An approximate factorization/least squares solution method for a mixed finite element approximation of the Cahn-Hilliard equation. *Japan Journal of Industrial and Applied Mathematics*, 13(3):495–517.
- Demaine, E. D. and Tachi, T. (2017). Origamizer: A practical algorithm for folding any polyhedron. In *33rd International Symposium on Computational Geometry, SoCG 2017, July 4-7, 2017, Brisbane, Australia*, page 1–34.
- Dorfler, W. (1996). A convergent adaptive algorithm for Poisson’s equation. *SIAM Journal on Numerical Analysis*, 33(3):1106–1124.
- Douglas, J. and Rachford, H. H. (1956). On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American Mathematical Society*, 82(2):421–439.
- Dudte, L. H., Vouga, E., Tachi, T., and Mahadevan, L. (2016). Programming curvature using origami tessellations. *Nature Materials*, 15:583–588.
- Ekren, I., Touzi, N., and Zhang, J. (2016). Viscosity solutions of fully nonlinear parabolic path dependent pdes: Part i. *Annals of Probability*, 44(2):1212–1253.
- Engquist, B., Froese, B. D., and Yang, Y. (2016). Optimal transport for seismic full waveform inversion. *Communications in Mathematical Science*, 14(8):2309–2330.
- Ern, A. and Guermond, J. (2004). *Theory and Practice of Finite Elements*. Applied Mathematical Sciences. Springer New York.
- Evans, L. C. (1982). Classical solutions of fully nonlinear, convex, second-order elliptic equations. *Communications on Pure and Applied Mathematics*, 35(3):333–363.
- Felmer, P. and Quaas, A. (2006). *Some Recent Results on Equations Involving the Pucci’s Extremal Operators*, page 263–281. Birkhäuser Basel, Basel.
- Felmer, P., Quaas, A., and Sirakov, B. (2012). Existence and regularity results for fully nonlinear equations with singularities. *Mathematische Annalen*, 354(1):377–400.
- Felmer, P. L., Quaas, A., and Tang, M. (2006). On uniqueness for nonlinear elliptic equation involving the Pucci’s extremal operator. *Journal of Differential Equations*, 226(1):80–98.
- Felton, S., Tolley, M., Demaine, E., Rus, D., and Wood, R. (2014). A method for building self-folding machines. *Science*, 345(6197):644–646.
- Feng, X. and Jensen, M. (2017). Convergent semi-Lagrangian methods for the Monge–Ampère equation on unstructured grids. *SIAM Journal on Numerical Analysis*, 55(2):691–712.
- Feng, X. and Lewis, T. (2014). Mixed interior penalty discontinuous Galerkin methods for fully nonlinear second order elliptic and parabolic equations in high dimensions. *Numerical Methods for Partial Differential Equations*, 30(5):1538–1557.

- Feng, X. and Lewis, T. (2018). Nonstandard local discontinuous Galerkin methods for fully nonlinear second order elliptic and parabolic equations in high dimensions. *Journal of Scientific Computing*, 77(3):1534–1565.
- Feng, X. and Neilan, M. (2009a). Mixed finite element methods for the fully nonlinear Monge-Ampère equation based on the vanishing moment method. *SIAM Journal on Numerical Analysis*, 47(2):1226–1250.
- Feng, X. and Neilan, M. (2009b). A modified characteristic finite element method for a fully nonlinear formulation of the semigeostrophic flow equations. *SIAM Journal on Numerical Analysis*, 47(4):2952–2981.
- Feng, X. and Neilan, M. (2009c). Vanishing moment method and moment solutions of second order fully nonlinear partial differential equations. *Journal of Scientific Computing*, 38(1):74–98.
- Feng, X. and Neilan, M. (2011). Analysis of Galerkin methods for the fully nonlinear Monge-Ampère equation. *Journal of Scientific Computing*, 47(3):303–327.
- Feng, X. and Neilan, M. (2014). Finite element approximations of general fully nonlinear second order elliptic partial differential equations based on the vanishing moment method. *Computers and Mathematics with Applications*, 68(12):2182–2204.
- Feng, X., Neilan, M., and Glowinski, R. (2013). Recent developments in numerical methods for fully nonlinear 2nd order PDEs. *SIAM Review*, 55:1–64.
- Feng, X., Neilan, M., and Prohl, A. (2007). Error analysis of finite element approximations of the inverse mean curvature flow arising from the general relativity. *Numerische Mathematik*, 108:93–119.
- Figalli, A. (2017). *The Monge-Ampère Equation and Its Applications*. Zurich lectures in advanced mathematics. European Mathematical Society.
- Figalli, A. (2018). On the Monge-Ampère equation. *Séminaire Bourbaki*.
- Finlay, C. and Oberman, A. M. (2018). Approximate homogenization of fully nonlinear elliptic pdes: Estimates and numerical results for Pucci type equations. *Journal of Scientific Computing*, 77(2):936–949.
- Fischer, J. and Kneuss, O. (2019). Bi-sobolev solutions to the prescribed Jacobian inequality in the plane with L^p data and applications to nonlinear elasticity. *Journal of Differential Equations*, 266(1):257–311.
- Formaggia, L., Micheletti, S., and Perotto, S. (2004). Anisotropic mesh adaptation in computational fluid dynamics: Application to the advection-diffusion-reaction and the Stokes problems. *Applied Numerical Mathematics*, 51(4):511–533.

Bibliography

- Formaggia, L. and Perotto, S. (2001). New anisotropic a priori error estimates. *Numerische Mathematik*, 89:641–667.
- Frisch, U., Matarrese, S., Mohayaee, R., and Sobolevski, A. (2002). A reconstruction of the initial conditions of the Universe by optimal mass transportation. *Nature*, 417(6886):260–262.
- Froese, B. D. and Oberman, A. M. (2011). Fast finite difference solvers for singular solutions of the elliptic Monge-Ampère equation. *Journal of Computational Physics*, 230(3):818–834.
- Froese, B. D. and Oberman, A. M. (2012). Convergent finite difference solvers for viscosity solutions of the elliptic Monge-Ampère equation in dimensions two and higher. *SIAM Journal on Scientific Computing*, 49(4):1692–1714.
- Froese, B. H. and Salvador, T. (2018). Higher-order adaptive finite difference methods for fully nonlinear elliptic equations. *Journal of Scientific Computing*, 75(3):1282–1306.
- Gabay, D. and Mercier, B. (1976). A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40.
- Gallistl, D. and Süli, E. (2019). Mixed finite element approximation of the Hamilton-Jacobi-Bellman equation with cordes coefficients. *SIAM journal on numerical analysis*, 57(2):592–614.
- Georgoulis, E. H., Houston, P., and Virtanen, J. (2009). An a posteriori error indicator for discontinuous Galerkin approximations of fourth-order elliptic problems. *IMA Journal of Numerical Analysis*, 31(1):281–298.
- Geuzaine, C. and Remacle, J.-F. (2009). Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331.
- Ghassaei, A., Demaine, E., and Gershenfeld, N. (2018). Fast , interactive origami simulation using gpu computation. In *Proceedings of the 7th International Meeting on Origami in Science, Mathematics and Education (OSME 2018)*, volume 4, page 1151–1166.
- Gilbarg, D. and Trudinger, N. (2001). *Elliptic Partial Differential Equations of Second Order*. Springer, Berlin.
- Glowinski, R. (2003). *Finite Element Methods For Incompressible Viscous Flow*, volume IX of *Handbook of Numerical Analysis (P.G. Ciarlet, J.L. Lions eds)*, page 3–1176. Elsevier, Amsterdam.
- Glowinski, R. (2008). *Numerical Methods for Nonlinear Variational Problems*. Springer-Verlag, New York, NY, second edition.

- Glowinski, R. (2009). Numerical methods for fully nonlinear elliptic equations. In *Invited Lectures, 6th Int. Congress on Industrial and Applied Mathematics, Zürich, Switzerland, 16-20 July 2007*, page 155–192. EMS.
- Glowinski, R. (2015a). *Variational Methods for the Numerical Solution of Nonlinear Elliptic Problem*. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics.
- Glowinski, R. (2015b). *Variational Methods for the Numerical Solution of Nonlinear Elliptic Problems*. SIAM, Philadelphia, PA.
- Glowinski, R., Kuznetsov, Y., and T.-W. Pan (2003). A penalty/Newton/conjugate gradient method for the solution of obstacle problems. *Comptes Rendus Mathématique*, 336:435–440.
- Glowinski, R., Lions, J.-L., and He, J. W. (2008). *Exact and Approximate Controllability for Distributed Parameter Systems: A Numerical Approach*. Encyclopedia of Mathematics and its Applications. Cambridge University Press.
- Glowinski, R., Liu, H., Leung, S., and Qian, J. (2019). A finite element/operator-splitting method for the numerical solution of the two dimensional elliptic Monge–Ampère equation. *Journal of Scientific Computing*, 79(1):1–47.
- Glowinski, R., Lueng, T., and Qian, J. (2015). A penalization-regularization-operator splitting method for Eikonal based traveltime tomography. *SIAM Journal on Imaging Sciences*, 8:1263–1292.
- Glowinski, R. and Marroco, A. (1975). Sur l’approximation, par éléments finis d’ordre un, et la résolution, par pénalisation-dualité d’une classe de problèmes de Dirichlet non linéaires. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 9(R2):41–76.
- Glowinski, R. and Niu, X. (2017). On the numerical solution of a vector Eikonal equation. Private communication.
- Glowinski, R., Osher, S., and Yin, W. (2018). *Splitting Methods in Communication, Imaging, Science, and Engineering*. Scientific Computation. Springer International Publishing.
- Glowinski, R. and Tallec, P. L. (1989). *Augmented Lagrangians and Operator-Splitting Methods in Nonlinear Mechanics*. SIAM, Philadelphia.
- González, C., Ostermann, A., Palencia, C., and Thälhammer, M. (2002). Backward Euler discretization of fully nonlinear parabolic problems. *Mathematics of Computation*, 71(237):125–145.
- Gremaud, P. A. and Kuster, C. M. (2006). Computational study of fast methods for the Eikonal equation. *SIAM Journal on Scientific Computing*, 27:1803–1816.

Bibliography

- Guillen, N. (2019). A primer on generated Jacobian equations: Geometry, optics, economics. *Notices of the American Mathematical Society*, 66:1.
- Gutiérrez, C. E. (2001). *The Monge-Ampère Equation*. Birkhäuser, Boston.
- Hairer, E., Norsett, S. P., and Wanner, G. (1993). *Solving Ordinary Differential Equations I. Nonstiff Problems*, volume 8 of *Springer Series in Comput. Mathematics*. Springer-Verlag.
- Hassan, W. and Picasso, M. (2015). An anisotropic adaptive finite element algorithm for transonic viscous flows around a wing. *Computers and Fluids*, 111:33–45.
- Hoskins, B. J. (1975). The geostrophic momentum approximation and the semi-geostrophic equations. *Journal of the Atmospheric Sciences*, 32(2):233–242.
- Hull, T. (2003). Counting mountain-valley assignments for flat folds. *Ars Combinatoria*, 67.
- Hysing, S. and Turek, S. (2004). *The Eikonal Equation: Numerical Efficiency Vs. Algorithmic Complexity on Quadrilateral Grids*. *Ergebnisberichte angewandte Mathematik*.
- Imbert, C. and Silvestre, L. (2013). *An Introduction to Fully Nonlinear Parabolic Equations*, page 7–88. Springer International Publishing, Cham.
- Ishii, H. (1989). On uniqueness and existence of viscosity solutions of fully nonlinear second-order elliptic PDE's. *Communications on Pure and Applied Mathematics*, 42(1):15–45.
- Ishii, H. and Lions, P.-L. (1990). Viscosity solutions of fully nonlinear second-order elliptic partial differential equations. *Journal of Differential Equations*, 83(1):26–78.
- Ivohkina, N. M. (1985). Classical solvability of the Dirichlet problem for the Monge-Ampère equation. *Journal of Soviet Mathematics*, 30(4):2287–2292.
- Iyer, N. R. (1999). A posteriori error estimation and adaptive mesh refinement for reliable finite element solutions. *Current Science*, 77(10):1319–1324.
- Jakobsen, E. R. and Karlsen, K. H. (2002). Continuous dependence estimates for viscosity solutions of fully nonlinear degenerate parabolic equations. *Journal of Differential Equations*, 183(2):497–525.
- Jasim, B. and Taheri, P. (2018). An origami-based portable solar panel system. In *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*.
- Jensen, M. (2018). Numerical solution of the simple Monge-Ampère equation with nonconvex Dirichlet data on non-convex domains. *arXiv preprint arXiv:1705.04653*.
- Jensen, M. and Smears, I. (2013). On the convergence of finite element methods for Hamilton-Jacobi-Bellman equations. *SIAM Journal on Numerical Analysis*, 51(1):137–162.

- Jensen, R. (1988). The maximum principle for viscosity solutions of fully nonlinear second order partial differential equations. *Archive for Rational Mechanics and Analysis*, 101(1):1–27.
- Katzourakis, N. (2015). Existence and uniqueness of global solutions to fully nonlinear first order elliptic systems. *Nonlinear Analysis: Theory, Methods and Applications*, 115:50–61.
- Katzourakis, N. (2016). Existence and uniqueness of global strong solutions to fully nonlinear second order elliptic systems. *Nonlinear Differential Equations and Applications NoDEA*, 23(3):33.
- Kelley, C. (1995). *Iterative Methods for Linear and Nonlinear Equations*. Society for Industrial and Applied Mathematics.
- Kirk, B. S., Peterson, J. W., Stogner, R. H., and Carey, G. F. (2006). libMesh: A c++ library for parallel adaptive mesh refinement/coarsening simulations. *Engineering with Computers*, 22(3–4):237–254.
- Koleva, M. N. and Vulkov, L. G. (2013). Quasilinearization numerical scheme for fully nonlinear parabolic problems with applications in models of mathematical finance. *Mathematical and Computer Modelling*, 57(9):2564–2575.
- Krylov, N. (2018). *Sobolev and Viscosity Solutions for Fully Nonlinear Elliptic and Parabolic Equations*. Mathematical Surveys and Monographs. American Mathematical Society.
- Krylov, N. V. (1984). Boundedly nonhomogeneous elliptic and parabolic equations in a domain. *Mathematics of the USSR-Izvestiya*, 22(1):67–97.
- Krylov, N. V. (2000). On the rate of convergence of finite-difference approximations for Bellmans equations with variable coefficients. *Probability Theory and Related Fields*, 117(1):1–16.
- Krylov, N. V. (2005). The rate of convergence of finite-difference approximations for Bellman equations with Lipschitz coefficients. *Applied Mathematics and Optimization*, 52(3):365–399.
- Kuo, H.-J. and Trudinger, N. S. (1992). Discrete methods for fully nonlinear elliptic equations. *SIAM Journal on Numerical Analysis*, 29(1):123–135.
- Labutin, D. A. (2001). Isolated singularities for fully nonlinear elliptic equations. *Journal of Differential Equations*, 177(1):49–76.
- Lakkis, O. and Mousavi, A. (2019). A least-squares Galerkin approach to gradient and hessian recovery for nondivergence-form elliptic equations. *arXiv preprint arXiv:1909.00491*.
- Lakkis, O. and Mousavi, A. (2020). A least-squares Galerkin gradient recovery method for fully nonlinear elliptic equations. *arXiv preprint arXiv:2007.15498*.

Bibliography

- Lakkis, O. and Pryer, T. (2011). A finite element method for second order nonvariational elliptic problems. *SIAM Journal on Scientific Computing*, 33(2):786–801.
- Lakkis, O. and Pryer, T. (2013). A finite element method for fully nonlinear elliptic problems. *SIAM Journal on Scientific Computing*, 35(4):A2025–A2045.
- Laug, P. and Borouchaki, H. (1996). The bl2d mesh generator: Beginner’s guide, user’s and programmer’s manual. Technical Report RT-0194, INRIA.
- Liu, H., Glowinski, R., Leung, S., and Qian, J. (2019). A finite element/operator-splitting method for the numerical solution of the three dimensional Monge-Ampère equation. *Journal of Scientific Computing*, 81(3):2271–2302.
- Liu, J., Froese, B. D., Oberman, A. M., and Xiao, M. (2017). A multigrid scheme for 3D Monge-Ampère equations. *International Journal of Computer Mathematics*, 94(9):1850–1866.
- Liu, Z. and He, Y. (2014). An iterative meshfree method for the elliptic Monge-Ampère equation in 2D. *Numerical Methods for Partial Differential Equations*, 30(5):1507–1517.
- Loeper, G. and Rapetti, F. (2005). Numerical solution of the Monge-Ampère equation by a Newton’s algorithm. *Comptes Rendus Mathématique*, 340(4):319–324.
- McMullen, C. (1998). Lipschitz maps and nets in Euclidean space. *Geometric and Functional Analysis*, 8(2):304–314.
- Mirebeau, J.-M. (2015). Discretization of the 3D Monge-Ampère operator, between wide stencils and power diagrams. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(5):1511–1523.
- Mishra, G. and Kumar, M. (2018). Numerical simulation of two-dimensional Pucci’s equation with Dirichlet boundary conditions using nonvariational finite element method. *Differential Equations and Dynamical Systems*.
- Nash, J. (1954). C^1 isometric imbeddings. *Annals of Mathematics*, 60(3):383–396. Full publication date: Nov., 1954.
- Neilan, M. (2010). A nonconforming Morley finite element method for the fully nonlinear Monge-Ampère equation. *Numerische Mathematik*, 115(3):371–394.
- Neilan, M. (2014a). Finite element methods for fully nonlinear second order PDEs based on a discrete Hessian with applications to the Monge-Ampère equation. *Journal of Computational and Applied Mathematics*, 263:351–369.
- Neilan, M. (2014b). A unified analysis for three finite element methods for the Monge-Ampère equation. *Electronic Transactions on Numerical Analysis*, 41:262–288.
- Neilan, M., Salgado, A. J., and Zhang, W. (2017). Numerical analysis of strongly nonlinear PDEs. *Acta Numerica*, 26:137–303.

- Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, New York, NY, USA, second edition.
- Nochetto, R. H. and Ntoggas, D. (2019). Convergent two-scale filtered scheme for the Monge-Ampère equation. *SIAM Journal on Scientific Computing*, 41(2):B295–B319.
- Nochetto, R. H., Ntoggas, D., and Zhang, W. (2018). Two-scale method for the Monge-Ampère equation: pointwise error estimates. *IMA Journal of Numerical Analysis*, 39(3):1085–1109.
- Nochetto, R. H., Ntoggas, D., and Zhang, W. (2019). Two-scale method for the Monge-Ampère equation: Convergence to the viscosity solution. *Mathematics of Computation*, 88:637–664.
- Nochetto, R. H. and Zhang, W. (2019). Pointwise rates of convergence for the Osher-Prussner method for the Monge-Ampère equation. *Numerische Mathematik*, 141(1):253–288.
- Oberman, A. (2008). Wide stencil finite difference schemes for the elliptic Monge-Ampère equations and functions of the eigenvalues of the Hessian. *Discrete and Continuous Dynamical Systems, B*, 10(1):221–238.
- Osher, V. I. and Prussner, L. D. (1988). On the numerical solution of the equation $z_{xx}z_{yy} - z_{xy}^2 = f$ and its discretization, I. *Numerische Mathematik*, 54:271–293.
- Picasso, M. (2003a). An anisotropic error indicator based on Zienkiewicz-Zhu error estimator: Application to elliptic and parabolic problems. *SIAM Journal on Scientific Computing*, 24(4):1328–1355.
- Picasso, M. (2003b). Numerical study of the effectivity index for an anisotropic error indicator based on Zienkiewicz-Zhu error estimator. *Communications in Numerical Methods in Engineering*, 19(1):13–23.
- Picasso, M. (2005). An adaptive algorithm for the Stokes problem using continuous, piecewise linear stabilized finite elements and meshes with high aspect ratio. *Applied Numerical Mathematics*, 54(3):470–490.
- Picasso, M. (2006). Adaptive finite elements with large aspect ratio based on an anisotropic error estimator involving first order derivatives. *Computer Methods in Applied Mechanics and Engineering*, 196(1):14–23.
- Picasso, M., Alauzet, F., Borouchaki, H., and George, P.-L. (2011). A numerical study of some hessian recovery techniques on isotropic and anisotropic meshes. *SIAM Journal on Scientific Computing*, 33:1058–1076.
- Pogorelov, A. V. (1971). On the regularity of generalized solutions of the equation $\det(\partial^2 u / \partial x^i \partial x^j) = \varphi(x^1, x^2, \dots, x^n) > 0$. *Doklady Akademii Nauk SSSR*, 200:534–537.
- Prins, C. R., Beltman, R., Thije ten Boonkamp, J. H. M., IJzerman, W. L., and Tukker, T. W. (2015). A least-squares method for optimal transport using the Monge-Ampère equation. *SIAM Journal on Scientific Computing*, 37(6):B937–B961.

Bibliography

- Qin, F., Luo, Y., Olsen, K., Cai, W., and Schuster, G. (1992). Finite-difference solution of the Eikonal equation along expanding wavefronts. *Geophysics*, 57(3):478–487.
- Qiu, D., Lam, K.-C., and Lui, L.-M. (2019). Computing quasiconformal folds. *arXiv preprint arXiv:1804.03936*.
- Quaas, A. and Sirakov, B. (2008). Principal eigenvalues and the Dirichlet problem for fully nonlinear elliptic operators. *Advances in Mathematics*, 218(1):105–135.
- Rivière, T. and Ye, D. (1996). Resolutions of the prescribed volume form equation. *Nonlinear Differential Equations and Applications*, 3(3):323–369.
- Sanderson, K. (2010). Bioengineering: What to make with DNA origami. *Nature*, 464(7286):158–159.
- Schenk, M. and Guest, S. (2011). Origami folding: A structural engineering approach. In *Conference: 5OSM*.
- Serra, J. (2015). Regularity for fully nonlinear nonlocal parabolic equations with rough kernels. *Calculus of Variations and Partial Differential Equations*, 54(1):615–629.
- Soner, H., Bangerth, W., Rannacher, R., Foellmer, H., and Rogers, L. (2003). *Adaptive Finite Element Methods for Differential Equations*. Lectures in Mathematics. Springer.
- Sorensen, D. C. and Glowinski, R. (2010). A quadratically constrained minimization problem arising from PDE of Monge-Ampère type. *Numerical Algorithms*, 53(1):53–66.
- Stojanovic, S. (2004). Optimal momentum hedging via hypoelliptic reduced Monge-Ampère PDE. *SIAM J. Control and Optimization*, 43:1151–1173.
- Sulman, M. (2017). A comparison study of parabolic Monge-Ampère equations adaptive grid methods. In Huang, Z., Stynes, M., and Zhang, Z., editors, *Boundary and Interior Layers, Computational and Asymptotic Methods BAIL 2016*, page 183–196, Cham. Springer International Publishing.
- Sulman, M., Nguyen, T., Haynes, R., and Huang, W. (2020). Domain decomposition parabolic Monge-Ampère approach for fast generation of adaptive moving meshes. *arXiv preprint arXiv:2006.14602*.
- Sulman, M., Williams, J., and Russell, R. (2011a). Optimal mass transport for higher dimensional adaptive grid generation. *Journal of Computational Physics*, 230:3302–3330.
- Sulman, M. M., Williams, J., and Russell, R. D. (2011b). An efficient approach for the numerical solution of the Monge-Ampère equation. *Applied Numerical Mathematics*, 61(3):298–307.
- Tachi, T. (2009). Simulation of rigid origami. *Origami*, 4(8):175–187.
- Tachi, T. (2010). Freeform variations of origami. *Journal for Geometry and Graphics*, 14(2):203–215.

- Thije ten Boonkamp, J. H. M., Romijn, L. B., and IJzerman, W. L. (2019). Generalized Monge-Ampère equations for illumination freeform design. In Wang, Y., Benítez, P., and Matoba, O., editors, *Optical Design and Testing IX*, volume 11185, page 11 – 28. International Society for Optics and Photonics, SPIE.
- Thomee, V. (2007). *Galerkin Finite Element Methods for Parabolic Problems*. Springer Berlin Heidelberg.
- Trudinger, S. (1997). Weak solutions of Hessian equations. *Communications in Partial Differential Equations*, 22(7-8):25–54.
- Tychonoff, A. N. (1963). The regularization of incorrectly posed problems. *Doklady Akademii Nauk SSSR*, 153:42–52.
- Verfürht, R. (1996). *A review of a posteriori error estimation and adaptive mesh-refinement techniques*, page 1–127. Advances in numerical mathematics. Wiley.
- Verfürth, R. (1994). A posteriori error estimates for nonlinear problems. finite element discretizations of elliptic equations. *Mathematics of Computation*, 62(206):445–475.
- Verfürth, R. (2013). *A Posteriori Error Estimation Techniques for Finite Element Methods*. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford.
- Villani, C. (2003). *Topics in Optimal Transportation*. Graduate studies in mathematics. American Mathematical Society.
- Wang, L. (1992). On the regularity theory of fully nonlinear parabolic equations: I. *Communications on Pure and Applied Mathematics*, 45(1):27–76.
- Weller, H., Browne, P., Budd, C., and Cullen, M. (2016). Mesh adaptation on the sphere using optimal transport and the numerical solution of a Monge-Ampère type equation. *Journal of Computational Physics*, 308:102–123.
- Westphal, C. R. (2019). A Newton div-curl least-squares finite element method for the elliptic Monge-ampère equation. *Computational Methods in Applied Mathematics*, 19:631–643.
- Yadav, N. K., Romijn, L. B., Thije ten Boonkamp, J. H. M., and IJzerman, W. L. (2019a). A least-squares method for the design of two-reflector optical systems. *Journal of Physics: Photonics*, 1(3):034001.
- Yadav, N. K., Thije ten Boonkamp, J. H. M., and IJzerman, W. L. (2019b). A Monge-Ampère problem with non-quadratic cost function to compute freeform lens surfaces. *Journal of Scientific Computing*, 80(1):475–499.
- Ye, D. (1994). Prescribing the Jacobian determinant in Sobolev spaces. *Annales de l'I.H.P. Analyse non linéaire*, 11(3):275–296.

Bibliography

- Zhang, J. and Zhuo, J. (2014). Monotone schemes for fully nonlinear parabolic path dependent PDEs. *Journal of Financial Engineering*, 01(01):1450005.
- Zhao, H. (2005). A fast sweeping method for Eikonal equations. *Mathematics of Computation*, 74:603–627.
- Zheligovsky, V., Podvigina, O., and Frisch, U. (2010). The Monge-Ampère equation: Various forms and numerical solution. *Journal of Computational Physics*, 229(13):5043–5061.
- Zienkiewicz, O. C. and Zhu, J. Z. (1987). A simple error estimator and adaptive procedure for practical engineering analysis. *International Journal for Numerical Methods in Engineering*, 24:337–357.
- Zienkiewicz, O. C. and Zhu, J. Z. (1992). The superconvergent patch recovery and a posteriori error estimates. I. The recovery technique. *International Journal for Numerical Methods in Engineering*, 33:1331–1364.
- Zitová, B. and Flusser, J. (2003). Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000.

Dimitrios Gourzoulidis

✉ dimigourz@gmail.com
📄 gr.linkedin.com/in/dimigourz

Education

- 2021 **PhD in Mathematics**, *Mathematics in Computational Science and Engineering Section*, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland.
PhD Thesis: “*Numerical Methods for First and Second Order Fully Nonlinear Partial Differential Equations*”.
- 2015 **MSc in Computational and Applied Mathematics**, *Mathematics and Applied Mathematics Department*, University of Crete, Heraklion, Greece.
MSc Thesis: “*Adaptive Discontinuous Galerkin Finite Elements Method for the Non Linear Schrödinger Equation in critical dimension*”.
- 2013 **BSc in Applied Mathematics**, *Department of Applied Mathematics*, University of Crete, Heraklion, Greece.
Specialized in Mathematical Modeling and Computational Techniques.

Experience

Research Internship

- Jan. 2016–
Aug. 2016 **Research Fellow**, *Institute for Applied and Numerical Mathematics*, Department of Mathematics, Karlsruhe Institute of Technology, Karlsruhe, Germany.
Project: “*Discontinuous Galerkin Finite Elements Method for Acoustics Wave Equations in Heterogenous Media*”.

Teaching

- 2017–2021 **Doctoral Assistant**, *EPFL*, Lausanne and *HES-SO Geneva*, Switzerland.
◦ Advanced Numerical Analysis Winter Semester 2019-2020.
◦ Numerical Solution of PDE's 2 Spring Semester 2017-2018-2018-2019 .
- 2013–2015 **Graduate Assistant**, *Department of Applied Mathematics*, Heraklion, Greece.
◦ Numerical Analysis Winter Semester 2013-2014-2015
◦ Numerical Solution of ODE's Spring Semester 2013-2014

Computer skills

Programming C, C++, MATLAB

Experience with Parallel programming(MPI, OPENMP), Math libraries (lapack,PETSc/Tao,...), Intel compilers, HPC systems, FEM libraries, SHELL programming, Unix, Windows

Publications

- 2021 *Numerical Approximation of the Prescribed Jacobian Equation*
A. Caboussat, R Glowinski, and D. Gourzoulidis, (in preparation)

- 2020 *An Anisotropic Adaptive Method for the Numerical Approximation of Orthogonal Maps*
A. Caboussat, D. Gourzoulidis, and M. Picasso, *Journal of Computational and Applied Mathematics* (submitted)
- 2019 *A Second Order Method for the Approximation of a Parabolic 2D Monge-Ampère Equation*
A. Caboussat, D. Gourzoulidis, *Proceedings of ENUMATH 2019*, Springer
- 2019 *Numerical Approximation of Orthogonal Maps*
A. Caboussat, R Glowinski, D. Gourzoulidis, and M. Picasso, *SIAM Journal on Scientific Computing*, **vol** 41 (2019)
- 2018 *A Least-Squares/Relaxation Method for the Numerical Solution of the Three-Dimensional Elliptic Monge-Ampère Equation*
A. Caboussat, R Glowinski, and D. Gourzoulidis, *Journal of Scientific Computing* **vol** 77 (2018)