

Fast Adaptive Augmented Lagrangian Digital Image Correlation

Thesis by
Jin Yang

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy

The logo for the California Institute of Technology (Caltech), featuring the word "Caltech" in a bold, orange, sans-serif font.

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2019
Defended Sep 6, 2018

© 2019

Jin Yang

ORCID: 0000-0002-5967-980X

All rights reserved

ACKNOWLEDGEMENTS

First, I would like to express my deep appreciation for my advisor, Prof. Kaushik Bhattacharya, for giving me five years' mentorship and support. I am so lucky to be a student of Kaushik, who is open-minded, patient, kind, and intelligent. I learned a lot from him about doing research, finding good problems, and focusing on the most important goals not only in academia; I also learned lots of tips from him in my personal life about how to be a good husband, a good friend, and a good father in the future. I am very thankful to my chair of committee, Prof. Guruswami Ravichandran, for his guidance and suggestions throughout my research. Ravi is a knowledgeable, optimistic and patient person. He always gives me encouragement on my research progress and helps me build confidence. I would like to thank Prof. Jose Andrade and Prof. Jean-Philippe Avouac for serving on my committee and providing valuable feedback on my work.

I would like to thank Prof. Michael Ortiz, Prof. Rodney Clifton, Prof. Dennis Kochmann, Prof. Ares J. Rosakis, and many other brilliant professors at Caltech who taught me math and solids courses. I would like to thank Prof. Nadia Lapusta, who let me TA and have teaching experience in her classes. I would like to thank Dr. Michael Mello and Dr. Vito Rubino for insightful discussions about my research. I also would like to thank our experienced engineer, Petros Arakelian, for helping me set up experimental equipment.

I would like to thank my friends and colleagues in Kaushik-Ravi group: Paul Mazur, Hao Zhou, Chun-Jen Hsueh, Dingyi, Victoria(Tori) Lee, Ying Shi Teh, Kevin Korner, Sai Sharan Injeti, Swarnava Ghosh, Stella Brach, Louisa Avellar, Paul Plucinsky, Gal Shmuel, Aaron Stebner, Likun Tan, Xin(Cindy) Wang, Lincoln Collins, Srivatsan Hulikal Sampath Kumaran, Vinamra Agrawal, Mauricio Ponga, M. Zubaer Hossain, Andrew W. Richards, Ha T. Giang, Jacob Notbohm, Noy Cohen and Ruobing Bai. Thank you all for your help and insightful discussions throughout my PhD years.

I would like to thank our MCE staff: Jennifer Campbell, Leslie Rico, Lynn Seymour, Carolina Oseguera, Cheryl L. Geer, Holly Golcher, and Sonya Lincoln, for all of their assistance and help during my PhD years. I would like to thank the Caltech ISP staff Laura and Daniel, for giving me lots of help as an international student in the United States.

I would like to thank all my friends from Caltech MCE, CaltechC, and CaltechY. I had a lot of good times with you guys, especially Wei Mao, Zheng Li, Yanan Sui, Feng Bi, Jiaming Li, Lucy Yin, John Pang, Yuanlong Hunag, Xiaobing Xiong, Luis Phillipe C Tosi, John K. (Kenny) Buyco, Kazuki Maeda, Nicholas Burali, Nikola Z. Georgiev, Brandon Runnels, Kimberly A. Mac Donald, Tomoyuki Oniyama, Christian Kettenbeil, Liuchi Li, Wen Chen, Cheng Li, Wen Yan, Charles S. (Stan) Wojnar, Zachary Erickson and Jacob Lin.

Finally and the most importantly, I am deeply grateful to my family for their support, especially my wife, Lu Xu. I am so lucky I married her during my third PhD year. She always accompanies me to study, workout, and explore the world. She always supports my academic career dream and helps me build my confidence. I could not have the achievements I have right now without her love and endless support.

ABSTRACT

Digital image correlation (DIC) is a powerful experimental technique for measuring full-field displacement and strain. The basic idea of the method is to compare images of an object decorated with a speckle pattern before and after deformation in order to compute the displacement and strain fields. Local Subset DIC and finite element-based Global DIC are two widely used image matching methods; however there are some drawbacks to these methods. In Local Subset DIC, the computed displacement field may not satisfy compatibility, and the deformation gradient may be noisy, especially when the subset size is small. Global DIC incorporates displacement compatibility, but can be computationally expensive. In this thesis, we propose a new method, the augmented-Lagrangian digital image correlation (ALDIC), that combines the advantages of both the local (fast and in parallel) and global (compatible) methods. We demonstrate that ALDIC has higher accuracy and behaves more robustly compared to both Local Subset DIC and Global DIC.

DIC requires a large number of high resolution images, which imposes significant needs on data storage and transmission. We combined DIC algorithms with image compression techniques and show that it is possible to obtain accurate displacement and strain fields with only 5 % of the original image size. We studied two compression techniques (discrete cosine transform (DCT) and wavelet transform), and three DIC algorithms (Local Subset DIC, Global DIC, and our newly proposed augmented Lagrangian DIC (ALDIC)). We found the Local Subset DIC leads to the largest errors and ALDIC to the smallest when compressed images are used. We also found wavelet-based image compression introduces less error compared to DCT image compression.

To further speed up and improve the accuracy of DIC algorithms, especially in the study of complex heterogeneous strain fields at various length scales, we apply an adaptive finite element mesh to DIC methods. We develop a new h-adaptive technique and apply it to ALDIC. We show that this adaptive mesh ALDIC algorithm significantly decreases computation time with only little loss in accuracy.

PUBLISHED CONTENT AND CONTRIBUTIONS

- [1] J Yang and K Bhattacharya. “Fast adaptive global digital image correlation”. In: *Advancement of Optical Methods & Digital Image Correlation in Experimental Mechanics*. Vol. 3. 2019. Chap. 7. DOI: https://doi.org/10.1007/978-3-319-97481-1_7.
J. Yang derived the main result of this paper.
- [2] J Yang and K Bhattacharya. “Augmented Lagrangian digital image correlation”. In: *Experimental Mechanics*. Under review (2018).
J. Yang derived the main result of this paper.
- [3] J Yang and K Bhattacharya. “Combining image compression with digital image correlation”. In: *Experimental Mechanics*. Under review (2018).
J. Yang derived the main result of this paper.
- [4] J Yang and K Bhattacharya. “Efficient FFT-based digital image correlation”. In: *To be submitted* (2018).
J. Yang derived the main result of this paper.
- [5] J Yang and K Bhattacharya. “Multigrid adaptive mesh digital image correlation”. In: *To be submitted* (2018).
J. Yang derived the main result of this paper.
- [6] AP Stebner et al. “Transformation strains and temperatures of a nickel-titanium-hafnium high temperature shape memory alloy”. In: *Acta Materialia* 76 (2014), pp. 40–53. DOI: <https://doi.org/10.1016/j.actamat.2014.04.071>. URL: <http://www.sciencedirect.com/science/article/pii/S1359645414003346>.
J. Yang performed the theoretical calculation results in this paper.

TABLE OF CONTENTS

Acknowledgements	iii
Abstract	v
Published Content and Contributions	vi
Table of Contents	vii
List of Illustrations	ix
List of Tables	xii
Chapter I: Introduction	1
Chapter II: Review current DIC methods	6
2.1 Digital image correlation problem formulation	6
2.2 Local Subset DIC method	7
2.3 Inverse Compositional Gauss-Newton (IC-GN) scheme	8
2.4 Global DIC method	10
2.5 Global DIC with regularization	12
2.6 Conclusion	14
Chapter III: Augmented Lagrangian DIC method	15
3.1 ALDIC formulation	15
3.2 Alternating direction method of multipliers (ADMM)	16
3.3 Extensions of ALDIC algorithm	18
3.4 Convergence and optimal conditions of ADMM	19
3.5 Demonstration	21
3.6 Computational cost	29
3.7 Conclusion	32
Chapter IV: Combining DIC with image compression techniques	35
4.1 Introduction	35
4.2 Review image compression techniques	36
4.3 Combining image compression and DIC methods	37
4.4 Demonstration	39
4.5 Conclusion	45
Chapter V: Adaptive mesh ALDIC	48
5.1 Introduction	48
5.2 Variational formulation of regularized Global DIC	49
5.3 Variational formulation of ALDIC Subproblem 2	51
5.4 Discretization	52
5.5 Finite element spaces	54
5.6 A priori error estimate	63
5.7 A posteriori error estimate	66
5.8 Mesh refinement	69
5.9 Convergence and stopping criterion	77
5.10 Demonstration	79

5.11 Computation cost	83
5.12 Conclusion	89
Chapter VI: Conclusion and future work	91
6.1 Conclusion	91
6.2 Future work	91
Appendix A: The operator \mathbf{D}	93
Appendix B: Proof of Theorem 1	95
Appendix C: 3D Kuhn triangulation and Octree adaptive mesh	98
Bibliography	106

LIST OF ILLUSTRATIONS

<i>Number</i>	<i>Page</i>
2.1 DIC reference image $f(\mathbf{X})$ deforms into deformed image $g(\mathbf{y}(\mathbf{X}))$. . .	6
2.2 The change of variables involved in the IC-GN update.	10
2.3 Comparison between Local Subset DIC and Global DIC.	11
3.1 Convergence of the ALDIC method for the SEM 2D-DIC synthetic images, Sample 1 representing translation.	23
3.2 Comparison of RMS error in displacement (a) and strain (b) computed with the three methods for the synthetic images in the SEM 2D-DIC, Sample 1 or translation.	24
3.3 Exact horizontal x -displacement and strain e_{xx} field associated with Sample 14 images L1, L3, and L5.	24
3.4 Convergence check of ALDIC algorithm in Sample 14 L1, L3, and L5.	26
3.5 The horizontal displacement (u) obtained using the three methods from the synthetic images of SEM 2D-DIC Sample 14: L1, L3, and L5.	27
3.6 The horizontal longitudinal strain (e_{xx}) obtained using the three methods from the synthetic images of SEM 2D-DIC Sample 14: L1, L3, and L5.	28
3.7 Heterogeneous specimen used for the third case study. (a) Front view of designed fracture specimen with brick architecture, where the box area will be captured using a CCD camera. (b) A speckle pattern is applied using white spray paint onto the surface of the specimen where the length scale of the digital image is 0.037mm/pixel. (c) One deformed image of the sample as the crack propagates under loading. (d) The local subsets/global finite element mesh used in all three DIC methods.	29
3.8 Convergence of ALDIC method in heterogeneous fracture experiment.	30
3.9 Contour plot of three DIC algorithms solved displacement and strain fields in heterogeneous fracture experiment.	30
3.10 Comparison of the deformation gradient (F_{11} component) obtained using the Local Subset DIC (a) and ALDIC (b) for Sample 14 L1. . .	33
3.11 Comparison of Local Subset DIC, Global DIC, and ALDIC.	34

4.1	Image compression applied to the image from the SEM DIC Challenge Sample 14, Case L1. (a) Original reference image (Size: 1.2 MB). (b) Original deformed image (Size: 1.2 MB). (c-f) Wavelet compressed image whose size is only 10% of original reference (c) and deformed (d) image (Size: 120 KB) and 5% of original reference (e) and deformed (f) image (Size: 59 KB). (g-j) DCT compressed image whose size is only 10% of original reference (g) and deformed (h) image (Size: 124 KB) and 5% of original reference (i) and deformed (j) image (Size: 62 KB).	37
4.2	Errors in the images induced by wavelet and DCT image compression. (a-b) Histogram of greyscale values of wavelet (a) and DCT (b) compressed images. (c-d) Both wavelet (c) and DCT (d) compression induced greyscale value errors obey standard Gaussian distributions with different standard deviations.	38
4.3	The displacement obtained from Sample 14 L1 data using the three DIC algorithms and the original as well as compressed images. The subset/mesh size is 20×20	40
4.4	The strains obtained from Sample 14 L1 data using the three DIC algorithms and the original as well as compressed images. The subset/mesh size is 20×20 . The strains for the Local Subset DIC are not shown because they are very noisy.	41
4.5	The displacement fields obtained from Local Subset DIC, Global DIC, and ALDIC using original, wavelet, and DCT compressed images of the heterogeneous fracture specimen.	43
4.6	The strain fields obtained from Global DIC and ALDIC using original, wavelet, and DCT compressed images of the heterogeneous fracture specimen.	44
5.1	Comparison of adaptive Kuhn triangulation mesh and adaptive Quadtree mesh.	55
5.2	All the 2D Kuhn triangles belong to one similarity class with different orientations.	57
5.3	All kinds of elements in the Quadtree mesh, where nodes 1-4 are regular corner nodes from Q4 finite element and nodes 5-8 are hanging nodes to keep C^0 continuity of the deformation field.	61
5.4	Isoparametric mapping in Q4 elements.	61
5.5	Example of recursive refinement of 2D Kuhn triangulation.	72

	xi	
5.6	Example of recursive refinement of 2D Quadtree mesh.	77
5.7	Comparison of solved Sample 14 L5 deformation field using adaptive regularized Global DIC with Kuhn triangulation and Quadtree mesh.	80
5.8	Comparison of solved Sample 14 L5 deformation field using adaptive ALDIC with Kuhn triangulation and Quadtree mesh.	81
5.9	Plot of a posteriori error estimates of SEM 2D-DIC synthetic images of Sample 14 L5 based on ADMM Subproblem 1 and Subproblem 2.	82
5.10	Comparison of a posteriori error estimate based on ADMM subproblem 1 local update and subproblem 2 global update.	82
5.11	(a) Collective cells under microscope. (b) DIC image with speckle pattern to measure collective cell migration. (Images courtesy of Jacob Notbohm.)	82
5.12	Adaptive ALDIC solved collective cell migration using Kuhn adaptive triangulation mesh.	84
5.13	Adaptive ALDIC solved collective cell migration using Quadtree adaptive mesh.	85
5.14	Comparison between solved collective cell migration using uniform Local Subset DIC, adaptive Kuhn triangulation ALDIC, and adaptive Quadtree mesh ALDIC.	86
5.15	Plot of a posteriori error estimates of cell migration based on ADMM Subproblem 1 and Subproblem 2.	87
5.16	Comparison of adaptive Local Subset DIC method, adaptive regularized Global DIC method and adaptive ALDIC method.	90
A.1	Matrix \mathbf{D} in (a) 4×4 and (b) 10×10 FEM Q4 meshes where both element length $h = 1$	94
C.1	Kuhn 3D simplex. (a) A 3D cube can be subdivided by three planar cuts in 6 similar Kuhn triangulation simplices. (b) Binary bisection tree of a 3D Kuhn triangulation simplex.	98
C.2	Adaptive 3D Octree mesh refinement.	99
C.3	3D Octree element template.	99
C.4	Isoparametric mapping in Octant elements.	100

LIST OF TABLES

<i>Number</i>		<i>Page</i>
3.1	List of symbols used in the demonstration section	22
3.2	Comparison of the RMS errors in displacement and strain for the SEM 2D-DIC synthetic images of Sample 14: L1, L3, and L5	25
3.3	DIC displacement and strain RMS errors of Sample 14 L1 with different window sizes	26
3.4	List of symbols used in the analysis of computational cost	31
3.5	The computation cost of the Local Subset DIC IC-GN iteration	31
3.6	The computation cost of the Global DIC FEM iteration	31
3.7	The computation cost of the ALDIC ADMM iteration	32
3.8	Computation time using three DIC algorithms	32
4.1	Mean and standard deviation of image compression induced greyscale errors	36
4.2	Absolute RMS errors of Sample 14 L1 data using three DIC algorithms applied to the original images and the compressed images with different subset/mesh	45
4.3	Relative RMS errors of Sample 14 L1 data using three DIC algorithms applied to the original images and the compressed images with different subset/mesh	46
4.4	Relative RMS errors of fracture experiment using three DIC algorithms applied to the compressed images compared with results using the original images	46
4.5	Computation time of the numerical demonstrations	47
5.1	Comparison of the RMS errors in different DIC algorithms solved displacement and strain for the SEM 2D-DIC synthetic images of Sample 14 L5	79
5.2	Computation time of adaptive regularized Global DIC using Quadtree mesh to solve SEM-2D synthetic images of Sample 14 L5	83
5.3	Computation time of adaptive ALDIC using Quadtree mesh to solve SEM-2D synthetic images of Sample 14 L5	87
5.4	Computation time of adaptive ALDIC using Kuhn triangulation mesh to solve SEM-2D synthetic images of Sample 14 L5	87
5.5	Computation time of adaptive regularized Global DIC using Quadtree mesh to solve collective cell migration	88

5.6	Computation time of adaptive ALDIC using Quadtree mesh to solve collective cell migration	88
5.7	Computation time of adaptive ALDIC using adaptive Kuhn triangulation mesh to solve collective cell migration	88

Chapter 1

INTRODUCTION

Digital image correlation (DIC) is a popular optical experimental technique for measuring deformation and strain in solids. In this method, we decorate the surface of the sample with a speckle pattern. We then take a sequence of grayscale digital images of a test specimen during the deformation. Finally, by comparing images in the sequence, we determine the displacement and strain fields of the specimen using image tracking algorithms [1, 2, 3, 4].

DIC has several advantages compared with other strain measurement methods. First, unlike electrical resistance strain gauges that need to be glued on the sample surface, taking digital images does not require contact with the specimen. This is especially advantageous for soft materials where contact may affect strain fields. Second, each electrical resistance strain gauge only measures the strain status of one point but DIC can easily provide full field displacement and strain values. Compared with other non contact and full field optical strain measurement methods such as holographic methods, speckle methods, and interferometric methods [5, 6, 7, 8], DIC experiments do not require a very sophisticated experimental environment.

DIC has been applied to study the behavior of diverse solids systems such as biological material [9, 10, 11], metal alloys [12], shape memory alloys [13], porous metals [14], polymers [15], and polymer foams [16]. It has provided insight into the very nonlinear behavior of solids like slip bands [13, 17] and crack tips [18]. This method can also be combined with other diagnostic tools to enable investigation of complex phenomena with very heterogeneous and complex strain fields at various length scales from nanometers to kilometers. For example, DIC has been used to measure nonuniform phase transformation by combining scanning electron microscopy (SEM) and electron backscatter diffraction (EBSD) [19]. It has also been used with atomic force microscopy (AFM) to measure in-plane displacement at the nanometer scale [20]. At the other extreme, DIC has been used in earthquake and glacier monitoring [21, 22, 23, 24] at the scale of tens of kilometers.

Over the last thirty years, various DIC algorithms to compare images and obtain displacement and strain have been proposed and implemented. Most algorithms can be cast into two categories: Local Subset DIC method and Global DIC method

[25, 26, 27, 28, 29]. In Local Subset DIC, as its name implies, we first break up both reference image and deformed image into many subsets and then find the deformation of each subset independently. Since the subsets are limited in size, the deformation of each subset can be solved very fast; moreover the subsets can be analyzed in parallel. Therefore, Local Subset DIC can be very fast. However, since the deformation of each subset is obtained independently, the overall deformation may not be compatible and the strain field can be extremely noisy. In Global DIC, we represent the global deformation using a basis set (often based on a finite element discretization), and then analyze the global image to obtain the coefficients relative to this basis set. However, this is expensive.

These considerations have led to a number of attempts to improve these methods. A number of filtering and smoothing schemes have been proposed to address the noisiness of the Local Subset DIC methods [30, 31, 32]. Broadly, filtering of both the images and the displacements not only reduces the noise but also can improve the accuracy because it incorporates information from surrounding regions. While this can be effective, the critical choice of filter is unrelated to the underlying mathematical structure and may be experiment dependent. Similarly, a number of sophisticated numerical methods have been introduced to address the computational cost of global methods. These have followed two key ideas, or a combination of the two. The first is to use either gradient [33, 34] or elastic [35, 36, 37] regularization. The second is to use domain decomposition where the domain is broken up into a number of sub-domains, the correlation is performed compatibly in each sub-domain and the compatibility between the sub-domains is enforced using either Lagrange multipliers [36, 38] or the finite element tearing and interconnecting (FETI) procedure [35, 37]. These can then be used in parallel implementation (see [39] for a review). These significantly speed up the convergence and reduce computational time. However, these require sophistication in their implementation and must be adopted to the problem at hand.

In this thesis, we propose and demonstrate a new image comparison algorithm: augmented Lagrangian DIC or ALDIC. This method seeks to combine the advantages of both the Local Subset DIC (speed and parallel implementation) and the Global DIC (displacement compatibility and strain smoothness). The basic idea is to match subsets locally as in the Local Subset DIC, but use compatibility as a constraint. Specifically, we introduce an auxiliary globally compatible displacement field and introduce the constraint that this auxiliary globally compatible displacement field

and its gradient equal the locally correlated values. We implement the constraint using the augmented Lagrangian method.

The augmented Lagrangian method, also known as the method of multipliers, has been used to solve constrained minimization problems in diverse fields [40, 41]. It adds to the objective functional a term that is linear in the constraint as in the method of Lagrange multipliers and a term that is quadratic in the constraint as in the penalty method. The addition of the quadratic term makes the numerical implementation easier than the method of Lagrange multipliers. However, unlike the penalty method, one does not need to take the limit of infinitely large penalty coefficients. For this reason, the augmented Lagrangian method has found widespread acceptance in both image precessing [42] and in mechanics [43, 44].

We implement the augmented Lagrangian using the alternating direction method of multipliers (ADMM) that is a form of operator splitting [45]. In this method, we successively perform the local correlation, optimize the auxiliary displacement, update the multiplier and iterate. The convergence and other numerical issues of ADMM have been carefully studied [46], and this method is widely used in image processing [45, 47, 48] and in mechanics [49]. The second problem, the optimization over the auxiliary displacement, is global. However, it leads to a universal, sparse, well-conditioned operator (sum of the Laplacian and identity). This can be treated very efficiently using established methods.

We compare the performance (accuracy and efficiency) of the new ALDIC algorithm with that of Local Subset DIC method and Global DIC method using both synthetic data (where the exact displacement is known) and experimental data. We show that ALDIC provides the most accurate displacement and strain. It is only a few times more computationally expensive compared to Local Subset DIC and significantly cheaper than Global DIC.

ALDIC also provides two additional advantages: it allows combining DIC with image compression, and it allows a multi-resolution approach.

In practice, DIC methods usually require a large number of high resolution images and this imposes significant needs on data storage and transmission. In the recent decades, a number of sophisticated image and data compression algorithms have been developed to represent images with less data size. We tested the combination of different DIC methods with two popular lossy image compression techniques based on discrete cosine transform (DCT) and wavelet transform and found that it

is possible to obtain accurate displacement and strain fields with only 5% of the original image size. Among all three DIC methods (Local Subset DIC method, Global DIC method, and our proposed ALDIC method), Local Subset DIC leads to the largest errors and ALDIC to the smallest errors when compressed images are used. We also found that wavelet-based image compression introduces less error compared to DCT image compression techniques [50].

To further save the computation cost, especially in the study of complex heterogeneous strain fields at various length scales, we apply h-adaptive finite element mesh with our proposed ALDIC method, which can be solved quickly. Compared to the current Global DIC method, this new adaptive ALDIC algorithm significantly decreases computation time with little loss (and some gain) in accuracy.

In this thesis, we review two current DIC algorithms, Local Subset DIC method and Global DIC method, in Chapter 2. Next, we propose and describe a new image comparison algorithm: augmented Lagrangian DIC (ALDIC) in Chapter 3. We verify and evaluate the accuracy of the proposed ALDIC method using a series of case studies using synthetic data and experimental data in Section 3.5. These examples demonstrate the superior accuracy of the proposed ALDIC algorithm. We analyze the computation cost of the proposed method in Section 3.6. We show that the computational effort of the ALDIC is at worst a factor of two to four more expensive compared to Local Subset DIC method, and less expensive than the Global DIC method.

Chapter 4 introduces combining digital image correlation with image compression techniques. Specifically, Section 4.2 introduces two popular image compression techniques based on discrete cosine transform (DCT) and wavelet transform which we use in this thesis. We analyze the combination of the image compression techniques with three types of DIC algorithms in Section 4.3. Section 4.4 provides various case studies with synthetic and experimental images. It shows that ALDIC combines naturally with wavelet transform based image compression.

Chapter 5 studies the combination of adaptive mesh strategies with various DIC algorithms. We develop a new robust adaptive mesh Global DIC method based on finite element a posteriori error estimate in Section 5.2. To further save computation cost, we apply adaptive mesh onto ALDIC method in Section 5.3. In Section 5.5, we use Kuhn triangulation and Quadtree adaptive mesh specifically for DIC problems to avoid image grayscale value interpolation bias errors. We demonstrate the superior efficiency of the proposed adaptive mesh ALDIC method using synthetic

and experimental examples in Section 5.10 and analyze their computation cost in Section 5.11. We recall the main results of this thesis and point out future directions in Chapter 6.

REVIEW CURRENT DIC METHODS

In this chapter, we review DIC problem formulation and two popular categories of DIC algorithms which are Local Subset DIC method and Global DIC method. Local Subset DIC decomposes the domain into many local subsets, ignores the dependency between neighboring subsets, and can be solved fast and in parallel. In Global DIC method, the whole domain with all the subsets' unknowns are solved together where the kinematic connection between neighboring subsets are automatically considered but it's more expensive to solve and easily gets stuck into local minima.

2.1 Digital image correlation problem formulation

Consider a domain $\Omega \subset \mathbb{R}^n$ undergoing a deformation $\mathbf{y} : \Omega \rightarrow \mathbb{R}^n$, ($n = 2, 3$). As seen in Figure 2.1, let \mathbf{X} denotes the reference or undeformed position of a particle in Ω and $\mathbf{y}(\mathbf{X})$ denotes the image or current position of the particle. Suppose we have a speckle pattern with grayscale value $f(\mathbf{X})$ in the reference domain, and the corresponding grayscale value $g(\mathbf{y})$ in the current configuration. If the deformation convects the grayscale, then we have

$$f(\mathbf{X}) = g(\mathbf{y}(\mathbf{X})). \quad (2.1)$$

The problem of digital image correlation is the *inverse problem* of finding the deformation $\mathbf{y}(\mathbf{X})$ that satisfies (2.1) given grayscale images $f(\mathbf{X})$ and $g(\mathbf{y})$. We pose it as one of optimization, or one of finding the deformation map that minimizes

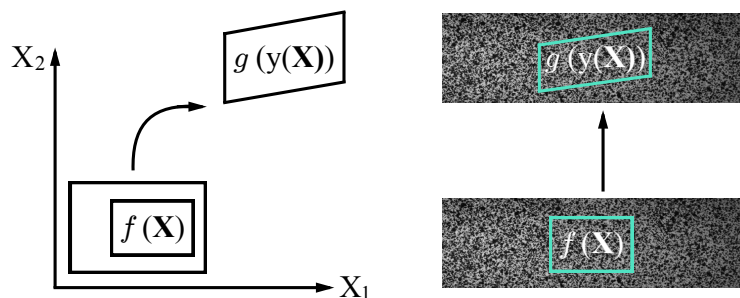


Figure 2.1: DIC reference image $f(\mathbf{X})$ deforms into deformed image $g(\mathbf{y}(\mathbf{X}))$.

the squared difference:

$$C = \int_{\Omega} |f(\mathbf{X}) - g(\mathbf{y}(\mathbf{X}))|^2 d\mathbf{X} \rightarrow \text{minimize over } \mathbf{y} : \Omega \rightarrow \mathbb{R}^n. \quad (2.2)$$

A few comments are in order. First, the images are pixelated with f, g taking discrete values. So we can either replace the integrals above with a sum, or interpolate the images (we use bi-cubic interpolation whenever we need sub-pixel values). Second, due to illumination artifacts and gain errors in real experiments, it is useful to normalize the images. This normalization depends on the knowledge of illumination and other experimental details. A simple example is to normalize both images to have the same mean and standard deviation:

$$f(\mathbf{X}) \mapsto \frac{f(\mathbf{X}) - \bar{f}}{\sigma_f}, \quad g(\mathbf{y}) \mapsto \frac{g(\mathbf{y}) - \bar{g}}{\sigma_g}, \quad (2.3)$$

where \bar{f}, \bar{g} are the mean values of f, g , and σ_f, σ_g are their standard deviations [51]. Henceforth, we assume that we are always working with normalized images. Third, in light of the normalization, note that minimizing C is equivalent to maximizing the cross correlation

$$\int_{\Omega} f(\mathbf{X})g(\mathbf{y}(\mathbf{X}))d\mathbf{X}. \quad (2.4)$$

Finally, in practice, there are different ways in which the correlation can be performed. One can take a series of images as the deformation proceeds and do the correlation between consecutive images, or one correlate the first and final image, or one can do something intermediate. The incremental correlation between successive image can lead to easier convergence and smaller individual errors due to small displacement, but can lead to the accumulation of systemic errors and add to the cost. These issues are common to all three algorithms that we discuss presently.

2.2 Local Subset DIC method

The Local Subset DIC method is the most widely used algorithm in DIC software packages [26, 27, 29]. As the name indicates, the idea is to break up the domain into local subsets and perform the correlation or optimization independently in each subset. Mathematically, we break up our domain into a finite number of subsets $\Omega = \bigcup_i \Omega_i$, and make the ansatz that the deformation is piecewise constant

translation or piecewise affine.

$$\mathbf{y}(\mathbf{X}) = \mathbf{X} + \mathbf{u}(\mathbf{X}) = \mathbf{X} + \sum_i (\mathbf{u}_i) \chi_i(\mathbf{X}) \quad (\text{Piecewise constant translation}), \quad (2.5)$$

$$\mathbf{y}(\mathbf{X}) = \mathbf{X} + \mathbf{u}(\mathbf{X}) = \mathbf{X} + \sum_i (\mathbf{u}_i + \mathbf{F}_i(\mathbf{X} - \mathbf{X}_{i0})) \chi_i(\mathbf{X}) \quad (\text{Piecewise affine}), \quad (2.6)$$

where \mathbf{u}_i is the translation vector of the center of the local subset Ω_i , \mathbf{F}_i is affine deformation gradient tensor, \mathbf{X}_{i0} is the coordinates of the center of each local subset, χ_i is the characteristic or index function

$$\chi_i = \begin{cases} 1 & \mathbf{X} \in \Omega_i, \\ 0 & \mathbf{X} \notin \Omega_i. \end{cases} \quad (2.7)$$

Using piecewise translation ansatz (2.5), the optimization problem (2.2) decomposes into a number of decoupled problems of optimizing over four ($n = 2$) or six ($n = 6$) scalar variables:

$$C_i = \int_{\Omega_i} |f(\mathbf{X}) - g(\mathbf{X} + \mathbf{u}_i)|^2 d\mathbf{X} \rightarrow \text{minimize over } \mathbf{u}_i. \quad (2.8)$$

Using piecewise affine deformation ansatz 2.6, the optimization problem (2.2) decomposes into a number of decoupled problems of optimizing over six ($n = 2$) or twelve ($n = 3$) scalar variables:

$$C_i = \int_{\Omega_i} |f(\mathbf{X}) - g(\mathbf{X} + \mathbf{u}_i + (\mathbf{F}_i(\mathbf{X} - \mathbf{X}_{i0})))|^2 d\mathbf{X} \rightarrow \text{minimize over } \mathbf{F}_i, \mathbf{u}_i. \quad (2.9)$$

Problem 2.8 can be solved very fast using fast Fourier transform method [10]. As for problem 2.9, there are a number of methods that have been used to solve this problem including the Inverse Compositional Gauss-Newton(IC-GN) [25, 26, 52] and Inverse Compositional Levenberg-Marquardt(IC-LM) scheme [52]. In this thesis, we use IC-GN and this is described in detail in 2.3 and summarized in Algorithm 1.

2.3 Inverse Compositional Gauss-Newton (IC-GN) scheme

Here we summarize the Inverse Compositional Gauss-Newton (IC-GN) scheme to solve Local Subset DIC optimization problem.

Given the current iterate of deformation map \mathbf{y}^k , we seek the updated deformation map \mathbf{y}^{k+1} . It is convenient to define the inverse maps ϕ^k and ϕ^{k+1} , where

Algorithm 1: Local Subset DIC

Input: Reference image f , deformed image g
Output: Displacement \mathbf{u}_i , affine deformation gradient tensor \mathbf{F}_i of each local subset

Step 1: Initialization using FFT integer pixel search method;

Step 2: Precompute image gradients ∇g ;

Step 3: For each local subset, compute $a_{ip}, b_{lqr}, c_{jkqr}$ using (2.14), (2.15), (2.16);

while $\|d_i\|, \|e_{jk}\| > \varepsilon$ **do**

 Step 4: Warp deformed image g with current deformation $\mathbf{F}_i, \mathbf{u}_i$;

 Step 5: Compute d_i, e_{jk} using (2.17), (2.18);

 Step 6: Compute \mathbf{v}, \mathbf{H} using (2.13);

 Step 7: Update ϕ using (2.11)

end

$\phi^k(\mathbf{y}^k(\mathbf{X})) = \mathbf{X}$. We define the increment ψ^k through $\mathbf{y}^{k+1} = \psi^k \circ \mathbf{y}^k$ as shown in Figure 2.2. We make a change of configuration and rewrite as

$$C_i = \int_{\Omega_i^k} |f(\phi^k(\mathbf{z})) - g(\psi(\mathbf{z}))|^2 d\mathbf{z}, \quad (2.10)$$

where \mathbf{z} is the current iterate of deformation map \mathbf{y}^k . We obtain ψ^k as the minimizer of this functional and the updated deformation map as

$$\phi^{k+1} = \phi^k \circ (\psi^k)^{-1}. \quad (2.11)$$

To minimize (2.10), we assume $\psi^k \approx \mathbf{z} + \mathbf{v} + \mathbf{H}(\mathbf{z} - \mathbf{z}_0)$ for small \mathbf{v} and \mathbf{H} . Therefore,

$$C_i = \int_{\Omega_i^k} |f(\phi^k(\mathbf{z})) - g(\mathbf{z}) - \nabla g(\mathbf{z}) \cdot (\mathbf{v} + \mathbf{H}(\mathbf{z} - \mathbf{z}_0))|^2 d\mathbf{z}. \quad (2.12)$$

Minimizing over \mathbf{v} and \mathbf{H} , we obtain

$$\begin{pmatrix} a_{lp} & b_{lqr} \\ b_{mnp} & c_{mnqr} \end{pmatrix} \begin{pmatrix} v_p \\ H_{qr} \end{pmatrix} = \begin{pmatrix} d_l \\ e_{mn} \end{pmatrix}, \quad (2.13)$$

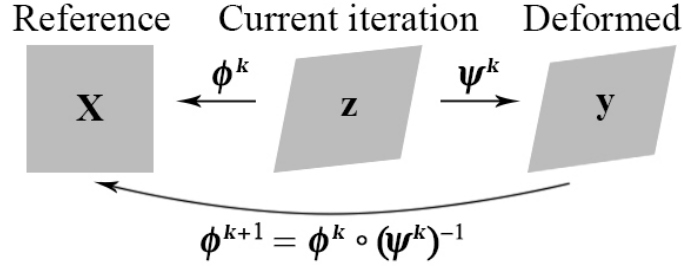


Figure 2.2: The change of variables involved in the IC-GN update.

where

$$a_{lp} = 2 \int_{\Omega_i^k} g_{,l} g_{,p} d\mathbf{z}, \quad (2.14)$$

$$b_{lqr} = \int_{\Omega_i^k} g_{,l} g_{,q} (z_r - z_{0r}) d\mathbf{z}, \quad (2.15)$$

$$c_{mnqr} = 2 \int_{\Omega_i^k} g_{,m} (z_n - z_{0n}) g_{,q} (z_r - z_{0r}) d\mathbf{z}, \quad (2.16)$$

$$d_l = \int_{\Omega_i^k} (f - g) g_{,l} d\mathbf{z}, \quad (2.17)$$

$$e_{mn} = \int_{\Omega_i^k} (f - g) g_{,m} (z_n - z_{0n}) d\mathbf{z}, \quad (2.18)$$

and $g_{,l} = \frac{\partial g}{\partial z_l}$ etc. We solve (2.13) for \mathbf{v} , \mathbf{H} to obtain ψ^k . We then obtain the new (inverse) deformation ϕ^{k+1} using (2.11). In practice, we don't need to compute Ω_i^k domain at each iteration, instead we directly compute all the integrations (or discrete summations) over the final deformed configuration, which also gives us good results and saves lots of computation time.

Since the problems are decoupled, i.e., can be solved independently for each i , local subset DIC is extremely fast and easily parallelized. Further, in practice, the subsets can overlap. However, since each problem is solved independently, the results can be noisy, susceptible to local imaging problems, and lead to discontinuous strain fields.

2.4 Global DIC method

Not like in the Local Subset DIC method where we break up the whole domain using densely distributed independent local subsets and ignore their compatibility between neighbor subsets, (see Figure 2.3 left), in the Global DIC method, we represent the global deformation using a global basis set, often based on a finite element discretization, where the compatibility can be guaranteed automatically

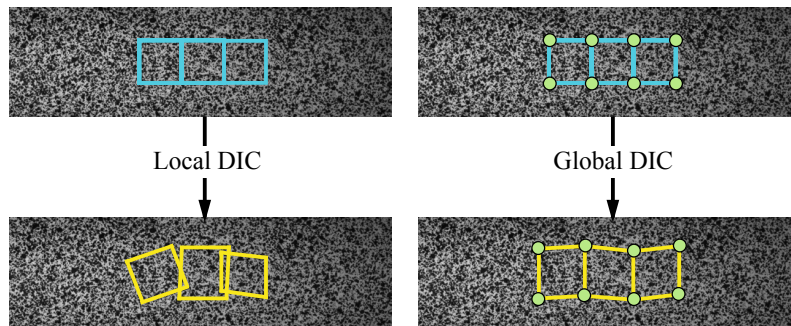


Figure 2.3: Comparison between Local Subset DIC and Global DIC.

(see Figure 2.3 right):

$$\mathbf{y}(\mathbf{X}) = \mathbf{X} + \mathbf{u}(\mathbf{X}) = \mathbf{X} + \sum_p u_p \boldsymbol{\psi}_p(\mathbf{X}), \quad (2.19)$$

where $\boldsymbol{\psi}_p(\mathbf{X})$ are the chosen global basis functions and u_p are the unknown degrees of freedom. Thus, the problem (2.2) becomes

$$C_g = \int_{\Omega} \left| f(\mathbf{X}) - g\left(\mathbf{X} + \sum_p u_p \boldsymbol{\psi}_p(\mathbf{X})\right) \right|^2 d\mathbf{X} \rightarrow \text{minimize over } \{u_p\}. \quad (2.20)$$

We can solve this problem iteratively by setting $\mathbf{u}_{k+1} = \mathbf{u}_k + \delta\mathbf{u}$ and using the first order approximation

$$g(\mathbf{y}(\mathbf{X})) = g(\mathbf{X} + \mathbf{u}_k(\mathbf{X}) + \delta\mathbf{u}) \approx g(\mathbf{X} + \mathbf{u}_k(\mathbf{X})) + \nabla g \cdot \delta\mathbf{u}(\mathbf{X}) \quad (2.21)$$

so that

$$C_g \approx \int_{\Omega} \left| f(\mathbf{X}) - g(\mathbf{X} + \mathbf{u}_k(\mathbf{X})) - \left(\sum_p \delta u_p \boldsymbol{\psi}_p(\mathbf{X}) \right) \cdot \nabla g(\mathbf{X}) \right|^2 d\mathbf{X}. \quad (2.22)$$

This leads to the linear equation in $\delta\mathbf{u}$

$$M_{pq} \delta u_q = b_p, \quad (2.23)$$

where

$$M_{pq} = \int_{\Omega} \boldsymbol{\psi}_p^T(\mathbf{X}) (\nabla g) (\nabla g)^T \boldsymbol{\psi}_q(\mathbf{X}) d\mathbf{X}, \quad (2.24)$$

$$b_p = \int_{\Omega} (f(\mathbf{X}) - g(\mathbf{X} + \mathbf{u}_k(\mathbf{X}))) \boldsymbol{\psi}_p^T(\mathbf{X}) \nabla g(\mathbf{X}) d\mathbf{X}. \quad (2.25)$$

Alternately, if the displacements are small, we can treat (2.23) as a linear problem with $\delta\mathbf{u}$ as the incremental displacement.

Note that the size of the linear problem (2.23) is equal to the number of basis functions or the size of the finite element discretization. This can be large if we seek fine resolution. Thus, Global DIC is expensive and difficult solve in parallel. However, it leads to compatible solutions. And there are methods to reduce the computational expense as discussed in the introduction. In practice, it is common to replace ∇g with ∇f or to use IC-GN which deals with the inverse map. This has the advantage that the matrix M_{pq} is independent of iteration thereby reducing the effort.

We remark that the procedure described in (2.23-2.25) may result in noisy displacement fields because of the conditioning of the matrix M . So it is common practice to add a weighted higher order penalty (regularizer) to the objective function. This needs experience and expertise. Further, this requires boundary conditions whose choice can lead to errors.

Finally, note that the Global DIC is not limited to smooth fields. It can be used to study discontinuous fields like cracks and shear bands by using enriched basis [53, 54]. In this paper, we use a Q4 finite element mesh in Global DIC, and this is summarized in Algorithm 2.

2.5 Global DIC with regularization

In Global DIC method, besides the guarantee of displacement continuity, we can also assume there is certain smoothness in the displacement field. Thus, the regularized Global DIC is the method which modifies Global DIC to prefer displacement field with some smoothness by adding higher order regularizer term $\alpha \langle \mathbf{B}\mathbf{u}(\mathbf{X}), \mathbf{u}(\mathbf{X}) \rangle$ onto the original correlation function in (2.20).¹

The new correlation function C_{g-RG} changes to be

$$C_{g-RG} = \int_{\Omega} [f(\mathbf{X}) - g(\mathbf{X} + \mathbf{u}(\mathbf{X}))]^2 + \alpha \langle \mathbf{B}\mathbf{u}(\mathbf{X}), \mathbf{u}(\mathbf{X}) \rangle. \quad (2.26)$$

Notation $\langle \mathbf{B}\mathbf{u}(\mathbf{X}), \mathbf{u}(\mathbf{X}) \rangle$ means the inner product between $\mathbf{B}\mathbf{u}(\mathbf{X})$ and $\mathbf{u}(\mathbf{X})$, which explicitly is that $\mathbf{B}\mathbf{u}(\mathbf{X})$ multiplies $\mathbf{u}(\mathbf{X})$ and then integrate over the whole domain. There are various choices of operator \mathbf{B} to add this smoothness penalty term. For

¹If there exist cracks in the displacement field, we can still add smoothness regularization terms in Global DIC method, excluding crack paths.

Algorithm 2: Q4 Global DIC**Input:** Reference image f , deformed image g **Output:** Displacement \mathbf{u} **Step 1:** Initialization using FFT integer pixel search method;**Step 2:** Precompute image gradients ∇g ;**foreach** *pixel in each finite element* **do** **Step 3:** Compute isoparametric element local coordinates ; **Step 4:** Compute isoparametric element ψ matrix ; **Step 5:** Compute isoparametric element Jacobian \mathbf{J} matrix; **Step 6:** Compute spatial gradient of ψ matrix: $D\psi$; **Step 7:** Assemble onto stiffness matrix $\mathbf{M} = \mathbf{M} + [\psi^T \nabla g][\psi^T \nabla g]^T$ using (2.24); **Step 8:** (Optional) Add regularizer term onto stiffness matrix, e.g.
 $\alpha[D\psi]^T[D\psi]$ **end****while** $\|\delta\mathbf{u}\| > \varepsilon$ **do** **Step 9:** Warp deformed image g with current displacement \mathbf{u}_n ; **Step 10:** Assemble vector \mathbf{b} using (2.25); **Step 11:** Add regularizer term onto vector \mathbf{b} if **Step 8** has been done ; **Step 12:** Solve $\delta\mathbf{u}$ by (2.23) ; **Step 13:** Update displacement $\mathbf{u}_{k+1} = \mathbf{u}_k + \delta\mathbf{u}$;**end**

example, when \mathbf{B} equals the identity matrix \mathbf{I} , this approach is also called Tikhonov regularization [55]. When \mathbf{B} equals the Laplace operator, this method is gradient regularization [33, 34]. Ref [34]§9 uses elastic potential energy as a Global DIC regularizer, where uses a positive definite bilinear operator, to introduce a global constraint of smoothness as:

$$\mathbf{B}[\mathbf{u}, \mathbf{v}] = \int_{\Omega} \lambda \operatorname{div}(\mathbf{u}) \operatorname{div}(\mathbf{v}) + 2\mu \sum_{i,j=1}^2 e_{ij}(\mathbf{u}) e_{ij}(\mathbf{v}). \quad (2.27)$$

Here $e_{ij}(\mathbf{u}) = 1/2(\partial u_i / \partial x_j + \partial u_j / \partial x_i)$ is the small strain tensor, and $\lambda > 0, \mu > 0$ are so-called Lamé constants of the materials' elastic properties. All the above regularization choices are good for small deformations. As for large deformations, fluid regularization [56], hyper-elastic regularization [57], and curvature regularization [58] have also been proposed recently.

The benefits of regularization in Global DIC method are not only from the additional assumption that the deformation field has a certain smoothness and belongs to more

strict Sobolev space, but also that it helps optimization converge faster and avoid local minima in the numerical respect.

Theoretically, all these regularized Global DIC could more or less help improve displacements result's smoothness. However, if the displacements field is heterogeneous and not quite uniform, the displacements' accuracy is quite sensitive to the choice of \mathbf{B} operator and the regularization term weight α . Too large α will smooth out the heterogeneity of displacement fields, while too small α doesn't help improve the noise obviously. Choosing operator \mathbf{B} and parameter α is still mostly based on the researcher's experience and preference. What's more, as for displacement field with high heterogeneity, varying regularization term coefficients α is needed instead of just one fixed value [59].

Adding regularizer terms can help Global DIC converges faster, however, it is still an expensive method. Note that the size of the linear problem (2.23) is equal to the number of basis functions or the size of the finite element discretization. This can be large if we seek fine resolution. And since the highly oscillating characteristics of the speckle patterns, standard Gaussian quadrature usually fails in the computation of stiffness matrix $\{M_{pq}\}$ and external force vector $\{b_p\}$. Instead, these numerical integral in are approximated by direct pixelwise summations, which make the Global DIC method expensive. And since all the unknowns in finite element discretization are solved together, the procedure described in Eqs(2.23-2.25) may still need a large number of iterations until converged.

2.6 Conclusion

In this chapter, we review two popular DIC methods: Local Subset DIC method and Global DIC method. Local Subset DIC decomposes the optimization problem into many independent local problems, which can be computed in parallel and converge fast but the solved displacements field usually is not compatible, and directly solved strain field is very noisy. Global DIC method solves the whole domain at the same time using finite element method and can be added with global smoothness regularization to guarantee the compatibility and a certain smoothness of the solved deformation field results; however, this method usually is expensive and understanding how to add regularization terms requires extensive expertise and experience.

AUGMENTED LAGRANGIAN DIC METHOD

- [1] J Yang and K Bhattacharya. “Augmented Lagrangian digital image correlation”. In: *Experimental Mechanics. Under review* (2018).

We introduce a new image comparison algorithm, augmented Lagrangian DIC (ALDIC) in this chapter, which seeks to combine the advantages of both the Local Subset DIC (speed and parallel implementation) and the Global DIC (displacement compatibility and strain smoothness).

3.1 ALDIC formulation

Recall the ansatz (2.5) we made in Local Subset DIC method. In this ansatz, the local displacement \mathbf{u}_i and local displacement gradient \mathbf{F}_i in the subdomain Ω_i are independent of each other, and independent for each i . Thus, there is no guarantee of compatibility for the deformation. However, if the displacement field were compatible, then the displacements and the displacement gradients would not be independent, but instead satisfy a global constraint

$$\{\mathbf{F}\} = \mathbf{D}\{\mathbf{u}\}, \quad (3.1)$$

where \mathbf{D} is an appropriate discrete gradient operator (see Appendix A for examples of first order finite differences). The Local Subset DIC ignores this constraint while the Global DIC enforces this constraint by kinematic construction.

The key idea of ALDIC is to treat this constraint (3.1) efficiently. We do so by leaving \mathbf{F}_i and \mathbf{u}_i discrete as before, and introduce an auxiliary compatible displacement field $\hat{\mathbf{u}}$ such that

$$\mathbf{F}_i = \nabla \hat{\mathbf{u}}(\mathbf{X}_{i0}), \quad \mathbf{u}_i = \hat{\mathbf{u}}(\mathbf{X}_{i0}). \quad (3.2)$$

In other words, we minimize (2.2) subject to the ansatz (2.5) and constraints (3.2). We do so using an augmented Lagrangian method. Specifically, we consider the correlation functional

$$\begin{aligned} \mathcal{L}_0 = & \sum_i \int_{\Omega_i} \left(|f(\mathbf{X}) - g(\mathbf{X} + \mathbf{u}_i + (\mathbf{F}_i(\mathbf{X} - \mathbf{X}_{i0})))|^2 \right. \\ & \left. + \frac{\beta}{2} |(\mathbf{D}\hat{\mathbf{u}})_i - \mathbf{F}_i|^2 + \nu_i : ((\mathbf{D}\hat{\mathbf{u}})_i - \mathbf{F}_i) + \frac{\mu}{2} |\hat{\mathbf{u}}_i - \mathbf{u}_i|^2 + \lambda_i \cdot ((\hat{\mathbf{u}})_i - \mathbf{u}_i) \right) d\mathbf{X}, \end{aligned} \quad (3.3)$$

where we use the matrix or Frobenius norm for matrices $|\mathbf{A}|^2 = \sum_{ij} |a_{ij}|^2$, vector norm for vectors $|\mathbf{a}|^2 = \sum_i a_i^2$, and $:$ for double dot product between two matrices $\mathbf{A} : \mathbf{B} = \sum_{ij} A_{ij}B_{ij}$. Above, $\{\boldsymbol{\nu}_i\}$, $\{\boldsymbol{\lambda}_i\}$ are Lagrange multipliers that enforce the constraints (3.2). Finally, β and μ are two positive real scalars. If $\beta = \mu = 0$, then this functional gives the traditional Lagrange multiplier formulation if we change the sign of $\{\boldsymbol{\nu}_i\}$, $\{\boldsymbol{\lambda}_i\}$. On the other hand if β and μ were very large with $\boldsymbol{\nu}_i = \boldsymbol{\lambda}_i = \mathbf{0}$, then we have a penalty method. Choosing β and μ to be positive real scalars while retaining the Lagrange multipliers is referred to as the augmented Lagrangian method, and gives rise to well-conditioned numerical problems [40].

Given β and μ , we iteratively minimize \mathcal{L}_0 over $\{\mathbf{F}_i\}$, $\{\mathbf{u}_i\}$ and $\{\hat{\mathbf{u}}_i\}$ and update $\{\boldsymbol{\nu}_i\}$ and $\{\boldsymbol{\lambda}_i\}$. Before we proceed, it is convenient to make the following modification to the functional above. We set $\mathbf{W}_i := \boldsymbol{\nu}_i/\beta$, $\mathbf{v}_i := \boldsymbol{\lambda}_i/\mu$ and define

$$\begin{aligned} \mathcal{L} = \sum_i \int_{\Omega_i} & \left(|f(\mathbf{X}) - g(\mathbf{X} + \mathbf{u}_i + (\mathbf{F}_i(\mathbf{X} - \mathbf{X}_{i0})))|^2 \right. \\ & \left. + \frac{\beta}{2} |(\mathbf{D}\hat{\mathbf{u}})_i - \mathbf{F}_i + \mathbf{W}_i|^2 + \frac{\mu}{2} |\hat{\mathbf{u}}_i - \mathbf{u}_i + \mathbf{v}_i|^2 \right) d\mathbf{X}. \end{aligned} \quad (3.4)$$

Notice that minimizing \mathcal{L}_0 over $\{\mathbf{F}_i\}$, $\{\mathbf{u}_i\}$ and $\{\hat{\mathbf{u}}_i\}$ is the same as minimizing over \mathcal{L} since they differ by quadratic terms independent of $\{\mathbf{F}_i\}$, $\{\mathbf{u}_i\}$ and $\{\hat{\mathbf{u}}_i\}$.

The functional \mathcal{L} is *not* local because the discrete derivative $(\mathbf{D}\hat{\mathbf{u}})_i$ depends on the value of $\hat{\mathbf{u}}_i$ in neighboring subsets. Thus, we still obtain a global problem as in the Global DIC method. However, there are two reasons why this problem is computationally easier. First, the only non-local term is quadratic with a constant scalar coefficient. Therefore it can be minimized very easily. Second, we can solve this problem using an alternating direction method of multipliers (ADMM) that allows us to break it up into simpler problems.

3.2 Alternating direction method of multipliers (ADMM)

We use alternating direction method of multipliers (ADMM) where local subproblems are coordinated to find a solution to a large global problem [46] to iteratively solve the problem.

Given $\{\mathbf{F}_i^k\}$, $\{\mathbf{u}_i^k\}$, $\{\hat{\mathbf{u}}_i^k\}$, $\{\mathbf{W}_i^k\}$, $\{\mathbf{v}_i^k\}$, we find the $(k+1)$ th update as follows:

- Subproblem 1: local update. While holding $\{\hat{\mathbf{u}}_i^k\}$, $\{\mathbf{W}_i^k\}$, $\{\mathbf{v}_i^k\}$ fixed, minimize \mathcal{L} over $\{\mathbf{F}_i\}$, $\{\mathbf{u}_i\}$, to obtain $\{\mathbf{F}_i^{k+1}\}$, $\{\mathbf{u}_i^{k+1}\}$:

$$\{\mathbf{F}_i^{k+1}\}, \{\mathbf{u}_i^{k+1}\} = \arg \min_{\{\mathbf{F}_i\}, \{\mathbf{u}_i\}} \mathcal{L}(\{\mathbf{F}_i\}, \{\mathbf{u}_i\}, \{\hat{\mathbf{u}}_i^k\}, \{\mathbf{W}_i^k\}, \{\mathbf{v}_i^k\}). \quad (3.5)$$

Since $\{\hat{\mathbf{u}}_i^k\}$ and hence $\{(\mathbf{D}\hat{\mathbf{u}}_i^k)\}$ are known, this problem breaks into a series of local problems that can be solved independently for each i :

$$\begin{aligned} \mathbf{F}_i^{k+1}, \mathbf{u}_i^{k+1} = \arg \min_{\mathbf{F}_i, \mathbf{u}_i} \mathcal{L}_i = \arg \min_{\mathbf{F}_i, \mathbf{u}_i} \int_{\Omega_i} & \left(|f(\mathbf{X}) - g(\mathbf{X} + \mathbf{u}_i + (\mathbf{F}_i(\mathbf{X} - \mathbf{X}_{i0})))|^2 \right. \\ & \left. + \frac{\beta}{2} |(\mathbf{D}\hat{\mathbf{u}}_i^k - \mathbf{F}_i + \mathbf{W}_i^k)|^2 + \frac{\mu}{2} |\hat{\mathbf{u}}_i^k - \mathbf{u}_i + \mathbf{v}_i^k|^2 \right) d\mathbf{X}. \end{aligned} \quad (3.6)$$

This is similar to Local Subset DIC and can be solved by any of the methods described in Section 2.2.

- Subproblem 2: global update. While holding $\{\mathbf{F}_i^{k+1}\}, \{\mathbf{u}_i^{k+1}\}, \{\mathbf{W}_i^k\}, \{\mathbf{v}_i^k\}$ fixed, we minimize \mathcal{L} over $\{\hat{\mathbf{u}}_i\}$ to obtain $\{\hat{\mathbf{u}}_i^{k+1}\}$:

$$\begin{aligned} \{\hat{\mathbf{u}}_i^{k+1}\} &= \arg \min_{\{\hat{\mathbf{u}}_i\}} \mathcal{L}(\{\mathbf{F}_i^{k+1}\}, \{\mathbf{u}_i^{k+1}\}, \{\hat{\mathbf{u}}_i\}, \{\mathbf{W}_i^k\}, \{\mathbf{v}_i^k\}) \\ &= \arg \min_{\{\hat{\mathbf{u}}_i\}} \sum_i \int_{\Omega_i} \left(\frac{\beta}{2} |(\mathbf{D}\hat{\mathbf{u}}_i - \mathbf{F}_i^{k+1} + \mathbf{W}_i^k)|^2 + \frac{\mu}{2} |\hat{\mathbf{u}}_i - \mathbf{u}_i^{k+1} + \mathbf{v}_i^k|^2 \right) d\mathbf{X}. \end{aligned} \quad (3.7)$$

Note that this is a global problem, but is independent of the images f, g . Indeed, it leads to the linear problem

$$(\beta \mathbf{D}^T \mathbf{D} + \mu \mathbf{I}) \hat{\mathbf{u}}^{k+1} = (\beta \mathbf{D}^T \mathbf{a} + \mu \mathbf{b}), \quad (3.8)$$

where $\mathbf{a} = \{\mathbf{F}_i^{k+1} - \mathbf{W}_i^k\}$ and $\mathbf{b} = \{\mathbf{u}_i^{k+1} - \mathbf{v}_i^k\}$. The solution is given by

$$\hat{\mathbf{u}}^{k+1} = (\beta \mathbf{D}^T \mathbf{D} + \mu \mathbf{I})^{-1} (\beta \mathbf{D}^T \mathbf{a} + \mu \mathbf{b}). \quad (3.9)$$

Since β and μ are fixed, the matrix $(\beta \mathbf{D}^T \mathbf{D} + \mu \mathbf{I})^{-1}$ can be precomputed and stored, and therefore this step becomes a simple matrix-vector multiplication. Further, the matrix \mathbf{D} has a structure, and therefore this matrix-vector multiplication can be carried out very efficiently.

- Subproblem 3: Lagrange multiplier update. We finally update $\{\mathbf{W}_i\}, \{\mathbf{v}_i\}$ as follows:

$$\mathbf{W}_i^{k+1} = \mathbf{W}_i^k + ((\mathbf{D}\hat{\mathbf{u}}_i^{k+1} - \mathbf{F}_i^{k+1})), \quad (3.10)$$

$$\mathbf{v}^{k+1} = \mathbf{v}^k + (\hat{\mathbf{u}}^{k+1} - \mathbf{u}^{k+1}). \quad (3.11)$$

- Stopping criterion. Theoretically, we should check the convergence of all quantities during ALDIC iterations. However, in practice, we care most about the displacements. Therefore, we simply check $(\hat{\mathbf{u}}^{k+1} - \hat{\mathbf{u}}^k)$, and stop if this happens to be smaller than a given tolerance.

The overall algorithm is summarized in Algorithm 3.

Algorithm 3: ALDIC

Input: Reference image f , deformed image g

Output: Displacement \mathbf{u}_i , deformation gradient tensor \mathbf{F}_i of each local subset, global displacement $\hat{\mathbf{u}}$

Step 1: Initialization using FFT integer pixel search method;

Step 2: Precompute finite difference operator \mathbf{D} ;

Step 3: Choose initial parameters β, μ . Set dual variables \mathbf{W}, \mathbf{v} to be zero;

while $\|\mathbf{u}^{k+1} - \mathbf{u}^k\| > \varepsilon$ *and* $\|\hat{\mathbf{u}}^{k+1} - \hat{\mathbf{u}}^k\| > \varepsilon$ **do**

Step 4: Solve subproblem 1 (3.6) as in Algorithm 1 for $\mathbf{u}^{k+1}, \mathbf{F}^{k+1}$;

Step 5: Solve subproblem 2 (3.9) for $\hat{\mathbf{u}}^{k+1}$;

Step 6: Update dual variables \mathbf{W}, \mathbf{v} by (3.10), (3.11);

end

3.3 Extensions of ALDIC algorithm

A simplification of Subproblem 1

While holding $\{\hat{\mathbf{u}}_i^k\}, \{\mathbf{W}_i^k\}, \{\mathbf{v}_i^k\}$ fixed, minimize \mathcal{L} over $\{\mathbf{F}_i\}, \{\mathbf{u}_i\}$, to obtain $\{\mathbf{F}_i^{k+1}\}, \{\mathbf{u}_i^{k+1}\}$. Since $\{\hat{\mathbf{u}}_i^k\}$ and hence $\{(\mathbf{D}\hat{\mathbf{u}}_i^k)\}$ are known, this problem breaks into a series of local problems that can be solved independently for each i as in Local Subset DIC, which can be solved in parallel and converges fast near the 6-parameter optimal solution using IC-GN scheme analogously with (2.13), where a_{lp}, d_l, c_{mnqr} and e_{mn} are replaced with

$$a'_{lp} = 2 \int_{\Omega_i^k} (g_{,l}g_{,p} + \frac{\mu}{2}\delta_{lp})d\mathbf{z}, \quad (3.12)$$

$$d'_l = \int_{\Omega_i^k} ((f-g)g_{,l} + \frac{\mu}{2}(u_l - v_l^k - \hat{u}_l^k))d\mathbf{z}, \quad (3.13)$$

$$c'_{mnqr} = 2 \int_{\Omega_i^k} \left[g_{,m}(z_n - z_{0n})g_{,q}(z_r - z_{0r}) + \frac{\beta}{2}\delta_{mq}\delta_{nr} \right] d\mathbf{z}, \quad (3.14)$$

$$e'_{mn} = \int_{\Omega_i^k} [(f-g)g_{,m}(z_n - z_{0n}) + \beta(F_{mn} - (\mathbf{D}\hat{\mathbf{u}})_{mn}^k - W_{mn}^k)] d\mathbf{z}. \quad (3.15)$$

Practically, we can make the following final simplification of Subproblem 1 to speed up ALDIC algorithm. The local problem (3.5) requires us to minimize \mathcal{L} over both $\{\mathbf{F}_i\}$ and $\{\mathbf{u}_i\}$: this makes the local problem large and the overall convergence slow. Further, the high dimensionality can lead to local minima and thus poor accuracy. This is consistent with the practice of using only the displacements in most commonly used Local Subset DIC softwares. Therefore, we simplify the

ALDIC Subproblem 1 as follows: in the $(k + 1)$ th iteration step, we update \mathbf{F}^{k+1} to be exactly equal to $\mathbf{D}\hat{\mathbf{u}}^k$ and only solve for \mathbf{u}^{k+1} . We still use the same IC-GN iteration method as introduced previously in Section 2.2 Local Subset DIC, where \mathbf{u}^{k+1} are solved by

$$\{\mathbf{u}_p^{k+1}\}_i = \{a'_{lp}\}_i^{-1} \{d'_l\}_i. \quad (3.16)$$

Extension ALDIC to arbitrary mesh

In the second subproblem (3.7), in the alternating direction, we fix \mathbf{F} , \mathbf{u} and try to update $\hat{\mathbf{u}}$ by:

$$\hat{\mathbf{u}}^{k+1} = \arg \min_{\hat{\mathbf{u}}} \frac{\beta}{2} \|\mathbf{D}\hat{\mathbf{u}} - \mathbf{F}^{k+1} + \mathbf{W}^k\|_F^2 + \frac{\mu}{2} \|\hat{\mathbf{u}} - \mathbf{u}^{k+1} + \mathbf{v}^k\|_{L_2}^2 \quad (3.17)$$

Using regular uniform mesh, analytical solution for $\hat{\mathbf{u}}$ can be computed fast using sparse matrix computation.

$$\hat{\mathbf{u}}^{k+1} = (\beta\mathbf{D}^T\mathbf{D} + \mu\mathbf{I})^{-1} (\beta\mathbf{D}^T\mathbf{a} + \mu\mathbf{b}). \quad (3.18)$$

Since β, μ are both positive scalars, it's easy to prove $(\beta\mathbf{D}^T\mathbf{D} + \mu\mathbf{I})$ is symmetric positive definite, therefore it always has inverse.

ALDIC algorithm ADMM scheme works not only for uniform square mesh, but also for arbitrary general mesh. If we are using non-uniform mesh, Subproblem 1 can be easily extended to solve local update problems around each node of the non-uniform mesh. Subproblem 2 can still be solved fast through

$$(\beta\Delta - \mu\mathbf{I})\hat{\mathbf{u}} = \beta\nabla \cdot (\mathbf{F}^{k+1} - \mathbf{W}^k) - \mu(\mathbf{u}^{k+1} - \mathbf{v}^{k+1}). \quad (3.19)$$

(3.19) will be discussed in details in Section 5.3 where we apply adaptive mesh onto ALDIC.

Using non-uniform mesh, the Subproblem 3 Lagrange multipliers update step will also be changed to

$$\begin{cases} \mathbf{W}_i^{k+1} = \mathbf{W}_i^k + ((\nabla\hat{\mathbf{u}})_i^{k+1} - \mathbf{F}_i^{k+1}) \\ \mathbf{v}_i^{k+1} = \mathbf{v}_i^k + (\hat{\mathbf{u}}_i^{k+1} - \mathbf{u}_i^{k+1}) \end{cases} \quad (\text{Lagrangian multipliers update}). \quad (3.20)$$

3.4 Convergence and optimal conditions of ADMM

Convergence of ADMM

We briefly recall some results from Boyd *et al.* [46] that apply to the ADMM algorithm proposed above. Assume that the following conditions are true:

- Assumption 1. The functional C_i in (2.9) or the first term of \mathcal{L} can be approximated by a closed, proper, and convex functional near the optimal solution.
- Assumption 2. The Lagrangian \mathcal{L}_0 with $\beta = \mu = 0$ has a saddle point; i.e., there exist $(\{\mathbf{F}_i^*\}, \{\mathbf{u}_i^*\}, \{\hat{\mathbf{u}}_i^*\}, \{\boldsymbol{\nu}_i^*\}, \{\boldsymbol{\lambda}_i^*\})$, for which

$$\begin{aligned} \mathcal{L}_0(\{\mathbf{F}_i^*\}, \{\mathbf{u}_i^*\}, \{\hat{\mathbf{u}}_i^*\}, \{\boldsymbol{\nu}_i^*\}, \{\boldsymbol{\lambda}_i^*\}) &\leq \mathcal{L}_0(\{\mathbf{F}_i^*\}, \{\mathbf{u}_i^*\}, \{\hat{\mathbf{u}}_i^*\}, \{\boldsymbol{\nu}_i^*\}, \{\boldsymbol{\lambda}_i^*\}) \\ &\leq \mathcal{L}_0(\{\mathbf{F}_i\}, \{\mathbf{u}_i\}, \{\hat{\mathbf{u}}_i\}, \{\boldsymbol{\nu}_i^*\}, \{\boldsymbol{\lambda}_i^*\}) \end{aligned}$$

for all $(\{\mathbf{F}_i\}, \{\mathbf{u}_i\}, \{\hat{\mathbf{u}}_i\}, \{\boldsymbol{\nu}_i\}, \{\boldsymbol{\lambda}_i\})$.

Then, we have the following convergence results:

- Primal residual convergence. $(\mathbf{D}\hat{\mathbf{u}}^k - \mathbf{F}^k) \rightarrow 0$ and $(\hat{\mathbf{u}}^k - \mathbf{u}^k) \rightarrow 0$ as $k \rightarrow \infty$, i.e., the constraints are satisfied asymptotically;
- Dual residual convergence. $(\hat{\mathbf{u}}^{k+1} - \hat{\mathbf{u}}^k) \rightarrow 0$ as $k \rightarrow \infty$, i.e., the dual feasibility is satisfied asymptotically, see (3.26).
- Objective convergence. $\mathcal{L}^k \rightarrow \mathcal{L}^*$ as $k \rightarrow \infty$, i.e., the Lagrangian approaches its optimal value;
- Dual variable convergence. $\mathbf{W}^k \rightarrow \mathbf{W}^*$, $\mathbf{v}^k \rightarrow \mathbf{v}^*$ as $k \rightarrow \infty$, where $(\mathbf{W}^*, \mathbf{v}^*)$ is dual optimal point.

Note that the local functional C_i can be highly oscillatory and is thus not convex. However, if the initial guess for the local variables $(\{\mathbf{F}_i\}, \{\mathbf{u}_i\})$ is in the convergence basin of Local Subset DIC, then the first assumption is true. If this assumption is false, then subproblem 1 (3.5) above diverges; this provides a check that this assumption holds.

Optimality conditions of ADMM

Set the first term in (3.4) to be

$$\Phi(\mathbf{F}, \mathbf{u}) = \sum_i \int_{\Omega_i} |f(\mathbf{X}) - g(\mathbf{X} + \mathbf{u}_i + (\mathbf{F}_i(\mathbf{X} - \mathbf{X}_{i0})))|^2 d\mathbf{X}. \quad (3.21)$$

The necessary and sufficient optimality conditions for the ALDIC ADMM formulation are primal feasibility,

$$\begin{bmatrix} \mathbf{D}\hat{\mathbf{u}}^* - \mathbf{F}^* \\ \hat{\mathbf{u}}^* - \mathbf{u}^* \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \quad (3.22)$$

and dual feasibility,

$$\begin{bmatrix} \frac{\partial \Phi(\mathbf{F}^*, \mathbf{u}^*)}{\partial \mathbf{F}} \\ \frac{\partial \Phi(\mathbf{F}^*, \mathbf{u}^*)}{\partial \mathbf{u}} \end{bmatrix} - \begin{bmatrix} \beta \mathbf{W}^* \\ \mu \mathbf{v}^* \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \quad (3.23)$$

Since $\mathbf{F}^{k+1}, \mathbf{u}^{k+1}$ minimize $\mathcal{L}(\mathbf{F}, \mathbf{u}, \hat{\mathbf{u}}^k, \mathbf{W}^k, \mathbf{v}^k)$ in the ADMM Subproblem 1, we have that

$$\begin{aligned} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} &= \begin{bmatrix} \frac{\partial \Phi(\mathbf{F}^{k+1}, \mathbf{u}^{k+1})}{\partial \mathbf{F}} \\ \frac{\partial \Phi(\mathbf{F}^{k+1}, \mathbf{u}^{k+1})}{\partial \mathbf{u}} \end{bmatrix} - \begin{bmatrix} \beta \mathbf{W}^k \\ \mu \mathbf{v}^k \end{bmatrix} - \begin{bmatrix} \beta (\mathbf{D} \hat{\mathbf{u}}^k - \mathbf{F}^{k+1}) \\ \mu (\hat{\mathbf{u}}^k - \mathbf{u}^{k+1}) \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial \Phi(\mathbf{F}^{k+1}, \mathbf{u}^{k+1})}{\partial \mathbf{F}} \\ \frac{\partial \Phi(\mathbf{F}^{k+1}, \mathbf{u}^{k+1})}{\partial \mathbf{u}} \end{bmatrix} - \begin{bmatrix} \beta \mathbf{W}^k \\ \mu \mathbf{v}^k \end{bmatrix} - \begin{bmatrix} \beta (\mathbf{D} \hat{\mathbf{u}}^{k+1} - \mathbf{F}^{k+1}) \\ \mu (\hat{\mathbf{u}}^{k+1} - \mathbf{u}^{k+1}) \end{bmatrix} - \begin{bmatrix} \beta (\mathbf{D} \hat{\mathbf{u}}^k - \mathbf{D} \hat{\mathbf{u}}^{k+1}) \\ \mu (\hat{\mathbf{u}}^k - \hat{\mathbf{u}}^{k+1}) \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial \Phi(\mathbf{F}^{k+1}, \mathbf{u}^{k+1})}{\partial \mathbf{F}} \\ \frac{\partial \Phi(\mathbf{F}^{k+1}, \mathbf{u}^{k+1})}{\partial \mathbf{u}} \end{bmatrix} - \begin{bmatrix} \beta \mathbf{W}^{k+1} \\ \mu \mathbf{v}^{k+1} \end{bmatrix} - \begin{bmatrix} \beta (\mathbf{D} \hat{\mathbf{u}}^k - \mathbf{D} \hat{\mathbf{u}}^{k+1}) \\ \mu (\hat{\mathbf{u}}^k - \hat{\mathbf{u}}^{k+1}) \end{bmatrix}. \end{aligned} \quad (3.24)$$

Or equivalently,

$$\begin{bmatrix} \frac{\partial \Phi(\mathbf{F}^{k+1}, \mathbf{u}^{k+1})}{\partial \mathbf{F}} \\ \frac{\partial \Phi(\mathbf{F}^{k+1}, \mathbf{u}^{k+1})}{\partial \mathbf{u}} \end{bmatrix} - \begin{bmatrix} \beta \mathbf{W}^{k+1} \\ \mu \mathbf{v}^{k+1} \end{bmatrix} = \begin{bmatrix} \beta (\mathbf{D} \hat{\mathbf{u}}^k - \mathbf{D} \hat{\mathbf{u}}^{k+1}) \\ \mu (\hat{\mathbf{u}}^k - \hat{\mathbf{u}}^{k+1}) \end{bmatrix}. \quad (3.25)$$

This means that the quantity

$$\mathbf{s}^{k+1} = \begin{bmatrix} \beta (\mathbf{D} \hat{\mathbf{u}}^k - \mathbf{D} \hat{\mathbf{u}}^{k+1}) \\ \mu (\hat{\mathbf{u}}^k - \hat{\mathbf{u}}^{k+1}) \end{bmatrix} \quad (3.26)$$

can be viewed as a residual for the dual feasibility condition (3.23). We will refer to \mathbf{s}^{k+1} as the dual residual at ADMM $(k+1)$ th iteration, and to

$$\mathbf{r}^{k+1} = \begin{bmatrix} \mathbf{D} \hat{\mathbf{u}}^{k+1} - \mathbf{F}^{k+1} \\ \hat{\mathbf{u}}^{k+1} - \mathbf{u}^{k+1} \end{bmatrix} \quad (3.27)$$

as the primal residual at ADMM $(k+1)$ th iteration. And these two residuals converge to zeros as ADMM proceeds.

3.5 Demonstration

We now demonstrate the ALDIC method, and compare it to both Local Subset DIC and Global DIC methods. All algorithms are implemented in Matlab. We use the following parameters unless it is specified otherwise. We use bi-cubic interpolations for the grayscale value at subpixel positions. In the Local Subset DIC, we stop IC-GN iterations when $\|d_i\|, \|e_{jk}\| < 10^{-6}$. Usually the IC-GN reaches convergence point within several iteration steps. In the Global DIC, we use Q4 finite element

Table 3.1: List of symbols used in the demonstration section

\mathbf{F}^k	Solved deformation gradient tensor in the k -th ADMM iteration Subproblem 1
\mathbf{u}^k	Solved displacement vector in the k -th ADMM iteration Subproblem 1
\mathbf{u}^k	Solved displacement vector in the k -th ADMM iteration Subproblem 2
$\mathbf{W}^k, \mathbf{v}^k$	Dual variables in the k -th ADMM iteration
$u,$	x -direction displacement component
$v,$	y -direction displacement component
e_{xx}, e_{yy}, e_{xy}	The “ xx ,” “ yy ” and “ xy ” components of infinitesimal strain

with a bilinear form of the domain’s displacement field trying to approximate the exact nonuniform one. We stop the iteration when the average magnitude of the nodal displacement update is smaller than 10^{-6} pixels. In ALDIC, we start \mathbf{W} and \mathbf{v} from zero. We choose μ to be $O(10^{-3}) \sim O(10^{-1})$ times diagonal terms of a'_{ip} . We take $\beta = [O(10^{-1}) \sim O(10^0) \cdot \text{element size}^2 \cdot \mu]$ to balance the relevant terms. We use the same stop criteria in subproblem 1 as Local Subset DIC ($\|d'_i\|_{L_2} < 10^{-6}$), and the whole ALDIC iteration stops when $\|\hat{\mathbf{u}}^{k+1} - \hat{\mathbf{u}}^k\|_{L_2} < 10^{-4}$.

When studying synthetic images where the exact deformation is known, we use the root-mean-square (RMS) error,

$$\text{RMS error} := \sqrt{\frac{\sum_{\# \text{ of nodes}} |\text{Numerical result} - \text{Exact value}|^2}{\# \text{ of nodes}}} \quad (3.28)$$

in both the displacement and strain. RMS error reflects globally how far the computed results are away from the exact values and is a measure of the standard variance of the computation error.

In Local Subset DIC, we report the deformation gradients/strains obtained directly by the Local Subset DIC correlation. In Global DIC, we compute nodal strains by extrapolating the strains from the finite element Gauss points. In ALDIC, the strain field is obtained directly from $\mathbf{D}\hat{\mathbf{u}}$. We summarize the symbols we used in the demonstration section in Table 3.1.

Case study I: Synthetic images from the SEM 2D-DIC Challenge

We study synthetic images from the SEM 2D-DIC challenge, samples 1 & 14 [60]¹. Sample 1 represents a series of pure translations while Sample 14 represents a sinusoidal deformation with changing frequency.

¹<https://sem.org/dic-challenge/2d-test-image-sets.asp>

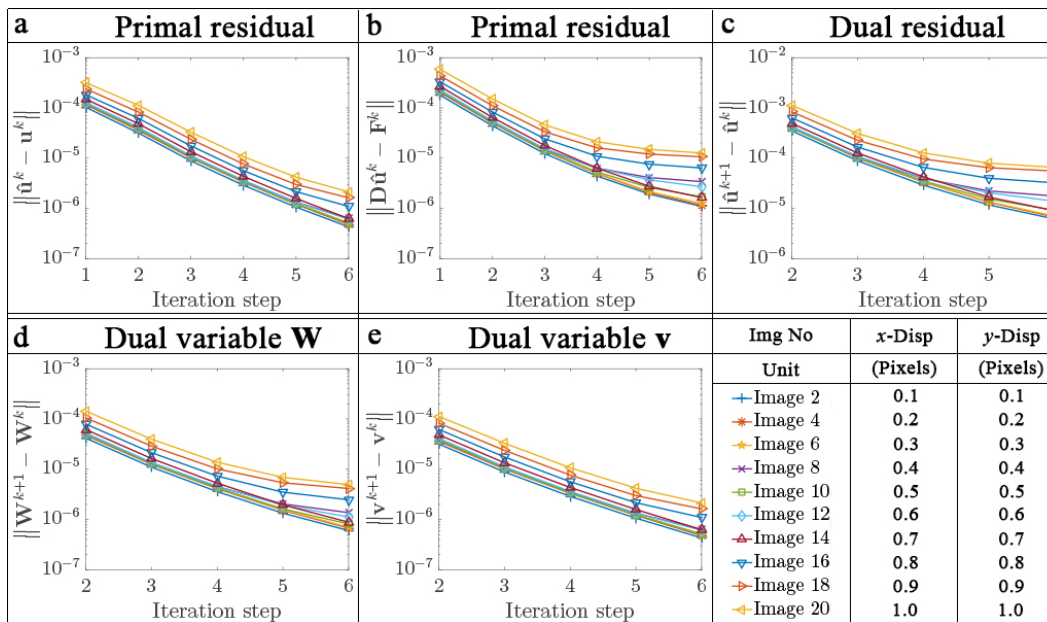


Figure 3.1: Convergence of the ALDIC method for the SEM 2D-DIC synthetic images, Sample 1 representing translation.

Translation: Sample 1

The deformations in Sample 1 are pure translations in both x and y directions with amplitudes ranging from 0 to 1 pixel in increments of 0.1 pixels. We set all the local window sizes to be 20×20 pixels, and set both the local neighboring windows distance and global element size to be 5×5 pixels. Figure 3.1 shows the convergence of the various quantities (without a stopping criterion). We see that ALDIC behaves well and converges within 6 steps. Figure 3.2 shows the RMS errors in displacements and strains, and compares with the corresponding errors in the Local Subset DIC and Global DIC methods. We observe that ALDIC has the smallest errors in all cases.

We make a couple of comments. First, we see that the error of the Global DIC method is high in this case. This is because we do not use a regularization since a regularizer forces zero gradient which artificially forces the desired answer. Second, when using synthetic images, a bias can be introduced if the interpolation used for subpixel shifting is different from those used for creating the images. The sinusoidal variation with image number is a reflection of this bias. We use a bicubic interpolation in our work based on the study of Bornert *et al.* [61].

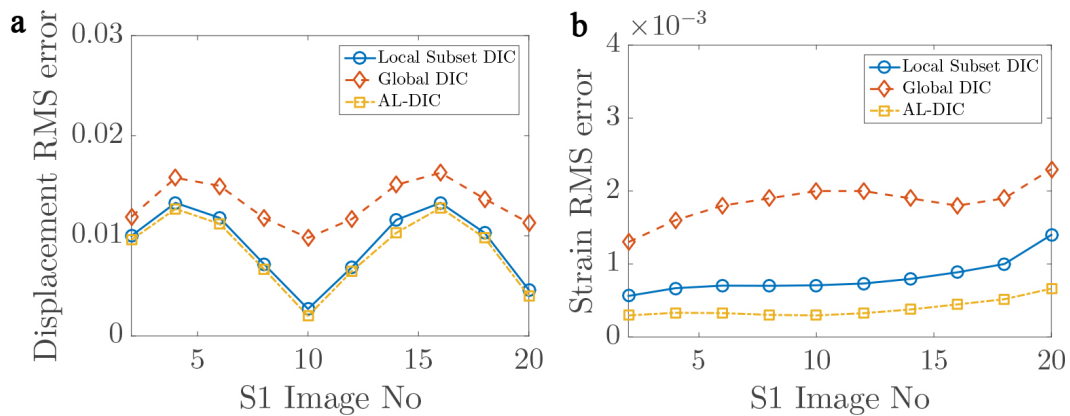


Figure 3.2: Comparison of RMS error in displacement (a) and strain (b) computed with the three methods for the synthetic images in the SEM 2D-DIC, Sample 1 or translation.

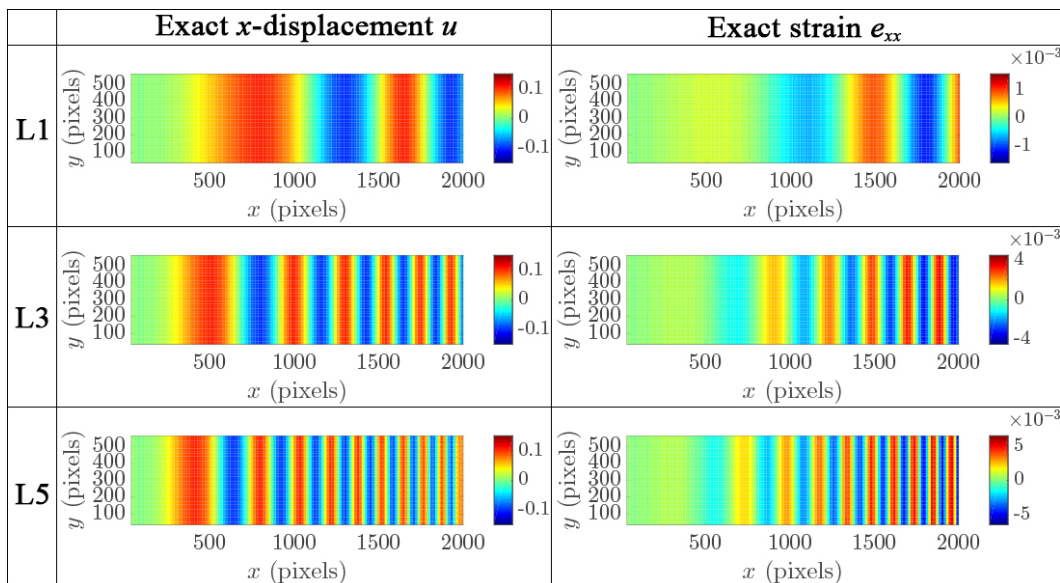


Figure 3.3: Exact horizontal x -displacement and strain e_{xx} field associated with Sample 14 images L1, L3, and L5.

Table 3.2: Comparison of the RMS errors in displacement and strain for the SEM 2D-DIC synthetic images of Sample 14: L1, L3, and L5

	Image No	local subset DIC	Regularized Global DIC	ALDIC
x displacement (pixels)	L1	0.0212	0.0080	0.0139
	L3	0.0202	0.0162	0.0136
	L5	0.0201	0.0234	0.0141
Strain e_{xx}	L1	3.18×10^{-3}	3.15×10^{-4}	9.83×10^{-4}
	L3	3.19×10^{-3}	1.90×10^{-3}	9.84×10^{-4}
	L5	3.20×10^{-3}	3.10×10^{-3}	1.00×10^{-3}

Heterogeneous deformation: Sample 14

The deformations in Sample 14 are sinusoidal with varying frequency in the x direction as shown in Figure 3.3 for the three images — L1, L3, and L5 — that we use. It has zero displacement in the y direction. We set all the local window sizes to be 30×30 pixels, and set both the local neighboring windows distance and global element size to be 5×5 pixels. As before, the ALDIC method converges in about six iterations (we have omitted the figure for brevity). Figures 3.5 and 3.6 show the horizontal displacement u and the horizontal longitudinal strain e_{xx} for the three images and the three methods. These figures show that the ALDIC leads to smooth displacement fields and this is reflected in the strain. Table 3.2 shows the RMS errors for strain and displacement, and shows that the ALDIC method leads to smaller errors compared to the other two methods.

We also use the image L1 from this set to study the effect of the subset size in the ALDIC method. Table 3.3 shows the RMS errors in displacement and strain using three window sizes. Not surprisingly, the errors increase with decreasing window size.

Case study II: Experimental heterogeneous fracture deformation

We conclude our case studies by analyzing data from an experiment on the fracture of a heterogeneous material taken from Avellar [62]. The $50.8 \times 45.7 \times 9.5$ mm specimen that is shown in Figure 3.7 (a) is 3D printed using a compliant material (Stratasys proprietary acrylic DM9895, $E=45$ MPa) shown in dark and a stiff material (Stratasys proprietary acrylic RGD835, $E=1960$ MPa, $\nu = 0.399$) shown in grey. We choose this example because the heterogeneous stiffness leads to complex strain and displacement fields. A speckle pattern is applied using white spray paint and the

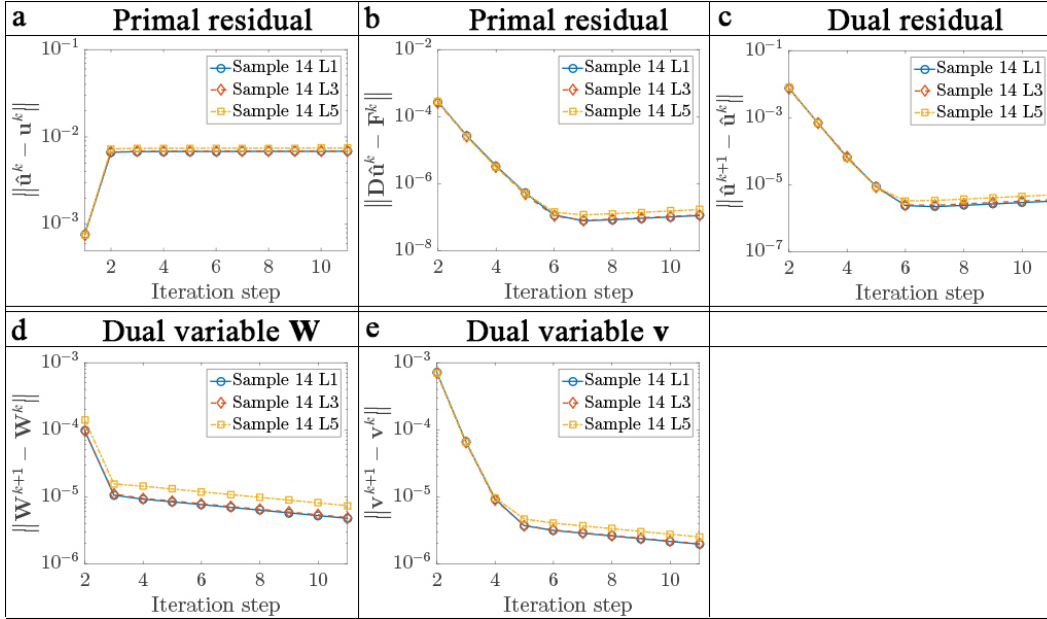


Figure 3.4: Convergence check of ALDIC algorithm in Sample 14 L1, L3, and L5.

Table 3.3: DIC displacement and strain RMS errors of Sample 14 L1 with different window sizes

	Subset Size	Local Subset DIC	Regularized Global DIC	ALDIC
x displacement (pixels)	30 × 30	0.0199	0.0079	0.0056
	20 × 20	0.0323	0.0082	0.0067
	10 × 10	0.0990	0.0088	0.0165
Strain e_{xx}	30 × 30	3.30×10^{-3}	2.08×10^{-4}	9.57×10^{-5}
	20 × 20	7.78×10^{-3}	2.35×10^{-4}	1.62×10^{-4}
	10 × 10	9.90×10^{-2}	2.84×10^{-4}	8.19×10^{-4}

specimen is loaded using the two pins inserted into the holes and pulled to failure. The reference image and one deformed image of the sample (the area in the red box of Figure 3.7 (a)) under loading are shown in Figure 3.7 (b-c), where the length scale in the digital image is 0.037mm/pixel. We set all the local window sizes to be 16×16 pixels, and set both the local neighboring windows distance and global element size also to be 16×16 pixels, see Figure 3.7 (d). The resulting images are analyzed using all three DIC methods. The convergence of the ALDIC method is shown in Figure 3.8. The displacement and strain fields obtained using all three methods are shown in Figure 3.9. We see little difference in the horizontal displacement u , but the vertical displacement v differs in the noise. ALDIC is less noisy than Global

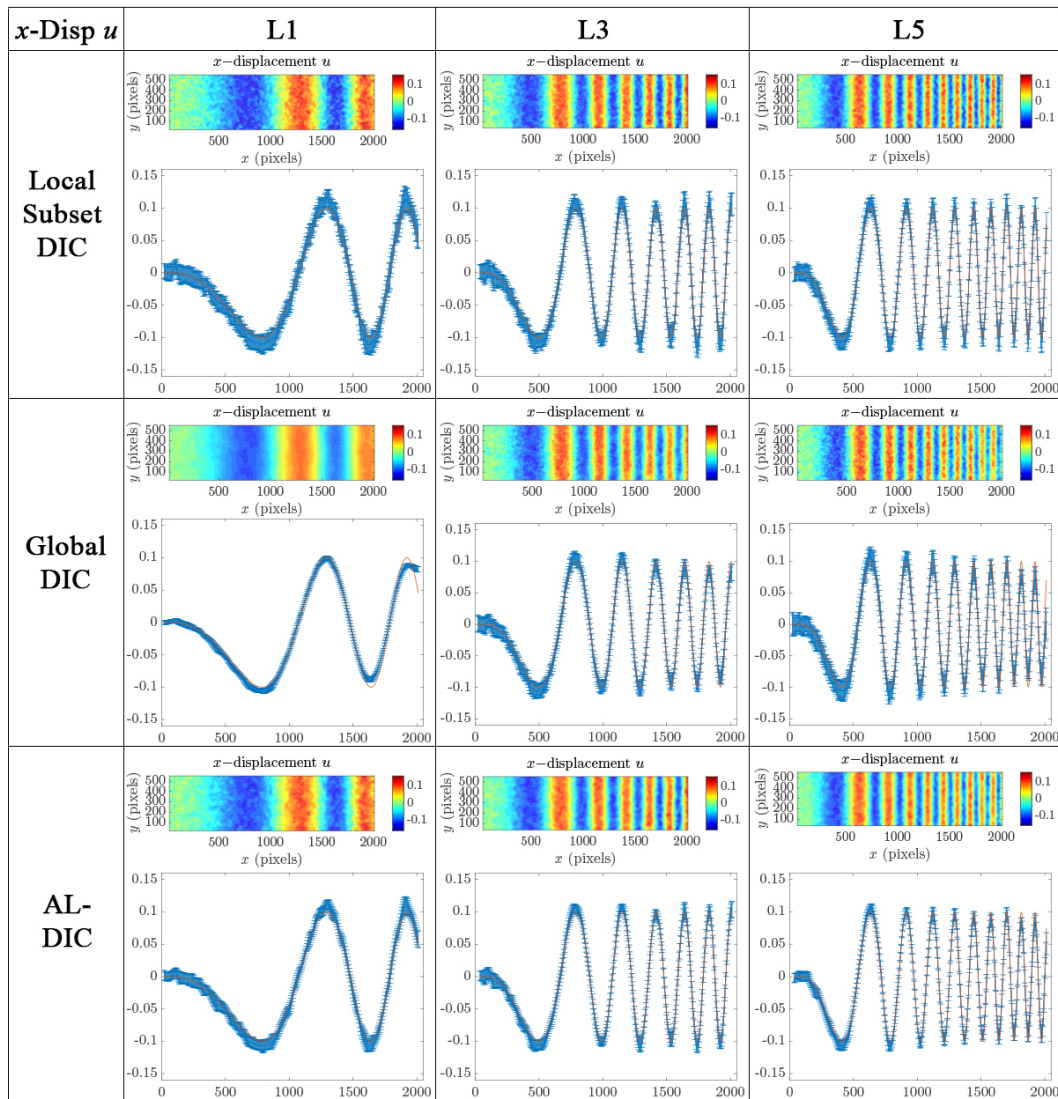


Figure 3.5: The horizontal displacement (u) obtained using the three methods from the synthetic images of SEM 2D-DIC Sample 14: L1, L3, and L5.

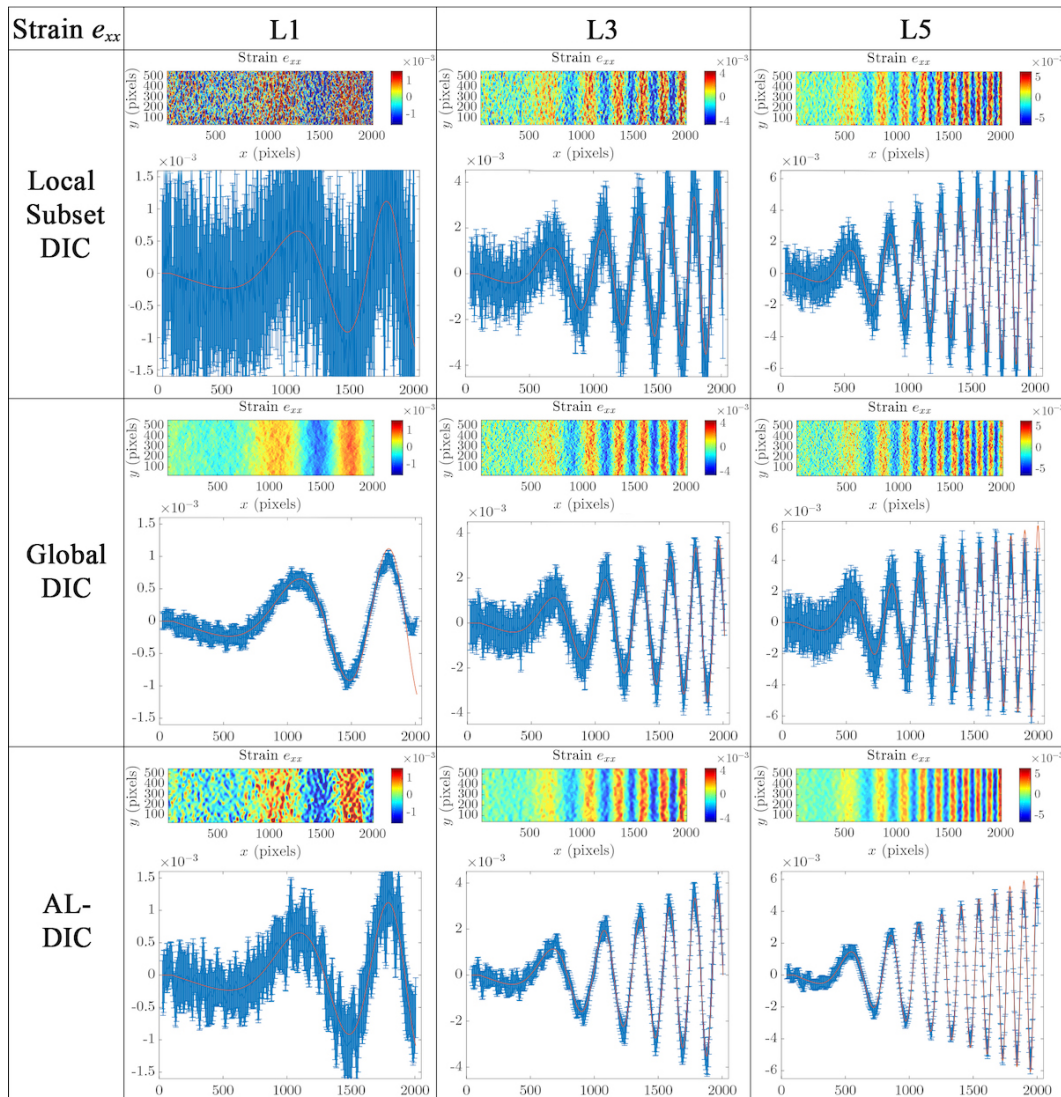


Figure 3.6: The horizontal longitudinal strain (e_{xx}) obtained using the three methods from the synthetic images of SEM 2D-DIC Sample 14: L1, L3, and L5.

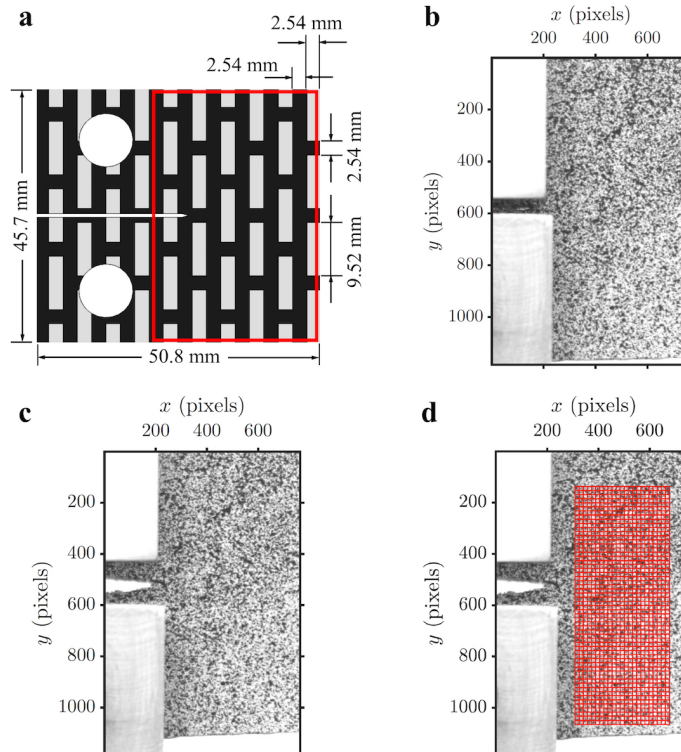


Figure 3.7: Heterogeneous specimen used for the third case study. (a) Front view of designed fracture specimen with brick architecture, where the box area will be captured using a CCD camera. (b) A speckle pattern is applied using white spray paint onto the surface of the specimen where the length scale of the digital image is 0.037mm/pixel. (c) One deformed image of the sample as the crack propagates under loading. (d) The local subsets/global finite element mesh used in all three DIC methods.

DIC, which in turn is less noisy than the Local Subset DIC. This is also reflected in the strain fields.

3.6 Computational cost

We compare the computational cost of the three DIC algorithms. The symbols used are listed in Table 3.4. We estimate the cost of each step in each algorithm, and these are listed in Tables 3.5-3.7. We then use the dominant terms (assuming that $k_1 \ll k_2$) to estimate the total cost, and these are also listed in the tables. We observe that all algorithms, properly implemented, scale linearly with the size of the image mN . Thus, the differences are in the pre-factors, and these can be significant as we presently demonstrate. We also note that the Local Subset DIC and ALDIC can be easily parallelized.

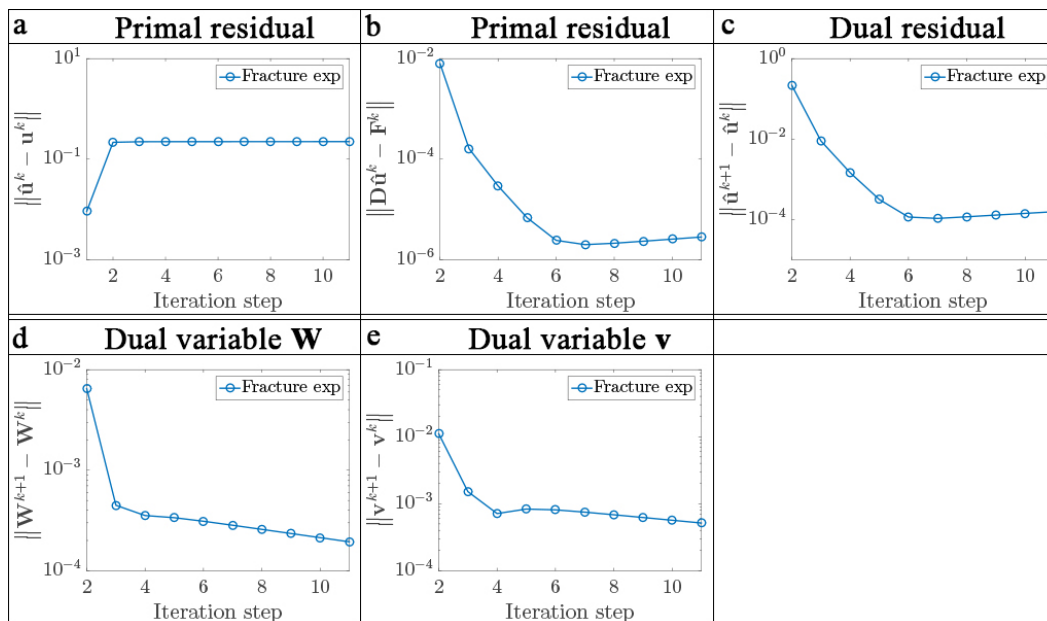


Figure 3.8: Convergence of ALDIC method in heterogeneous fracture experiment.

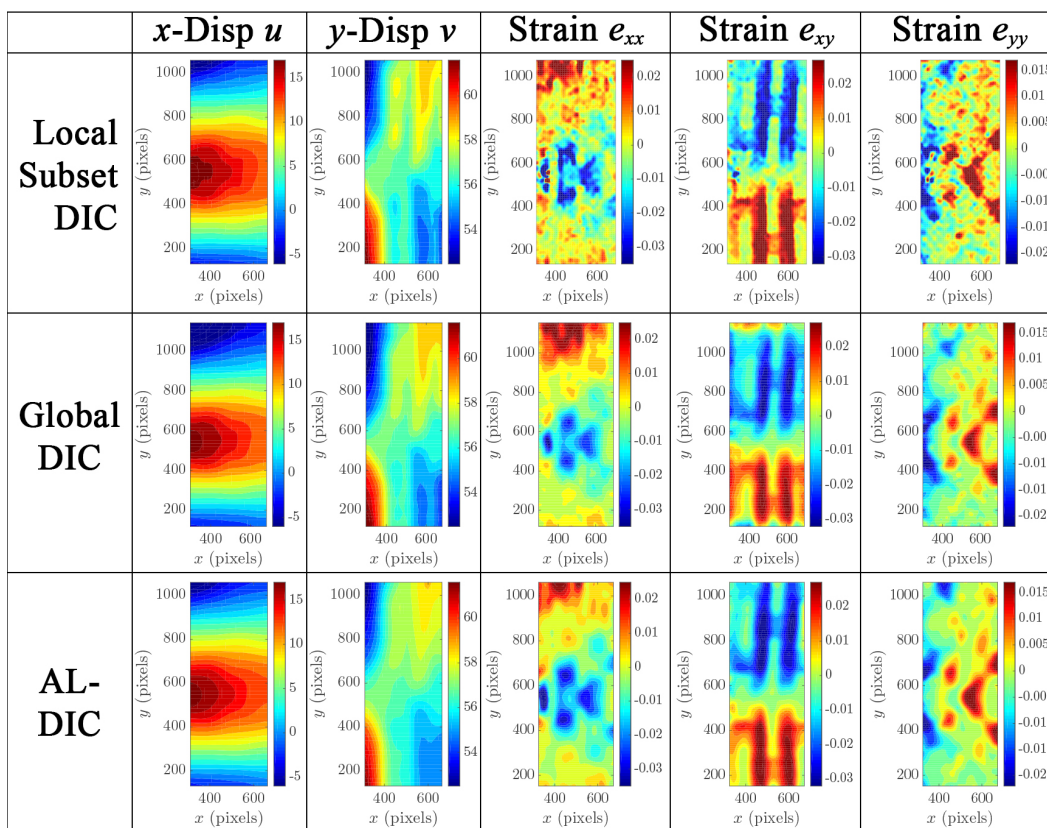


Figure 3.9: Contour plot of three DIC algorithms solved displacement and strain fields in heterogeneous fracture experiment.

Table 3.4: List of symbols used in the analysis of computational cost

N	# of pixels in each local subset or each finite element
m	# of total local subsets or finite elements
d	The dimension of images, e.g. $d = 2$ for 2D pixel images
n_L	Length of parameter vectors of each local subset
n_G	Length of parameter vectors in finite element
k_1	Computation cost to compute image grayscale derivatives
k_2	Computation cost to interpolate grayscale value at sub-pixel position
k_3	# of iterations in Local Subset DIC algorithm
k_4	# of iterations in Global DIC algorithm IC-GN scheme
k_5	# of iterations in ALDIC ADMM scheme
k_6	# of inside iterations in ALDIC Subproblem 1 IC-GN scheme
C	# of clusters used in Local Subset DIC and ALDIC Subproblem 1 for parallel computation

Table 3.5: The computation cost of the Local Subset DIC IC-GN iteration

Pre-computation	Step 2 $O(k_1 dmN)$	Step 3 $O(n_L^2 mN)$		
Per IC-GN iteration	Step 4 $O(k_2 n_L mN)$	Step 5 $O(n_L mN)$	Step 6 $O(n_L^3 m)$	Step 7 $O(n_L^3 m)$
Total				$O(k_2 k_3 n_L mN / C)$

Table 3.6: The computation cost of the Global DIC FEM iteration

Pre-computation	Step 2 $O(k_1 dmN)$	Step 3 $O(dn_G^3 mN)$	Step 4 $O(dn_G mN)$	Step 5 $O(d^2 n_G mN)$
	Step 6 $O(d^6 n_G mN)$	Step 7 $O(d^4 n_G^2 mN)$	Step 8 $O(d^4 n_G^2 mN)$	
Per FEM iteration	Step 9 $O(k_2 dn_G mN)$	Steps 10-11 $O(d^2 n_G mN)$	Step 12 $O(dn_G m)$	Step 13 $O(dn_G m)$
Total				$O(k_2 k_4 d^2 n_G mN)$

Table 3.8 lists the computational clock time for the case studies. All studies are performed on the same workstation with Intel (R) Xeon(R) CPU E5-2650 v3 2.30 GHz (2 Processors), RAM 32.0 GB Memory, 64-bit nodes. In the Local Subset DIC and ALDIC Subproblem 1 IC-GN iterations, we use 20 clusters and process it in parallel in Matlab. It is clear from the table that Local Subset DIC is the least expensive, and Global DIC is the most expensive as expected.

Table 3.7: The computation cost of the ALDIC ADMM iteration

Pre-computation	Steps 2-3 $O(2d^2m)$		
Per ALDIC iteration	Step 4 $O(k_2k_6n_LmN)$	Step 5 $O(2d^3m^2)$	Step 6 $O(n_Lm)$
Total	$O(k_2k_5k_6n_LmN/C)$		

Table 3.8: Computation time using three DIC algorithms

Example Name	Parameter	Local Subset DIC time cost(s)	Parameter	Global DIC time cost(s)	Parameter	Parameter	ALDIC time cost(s)
Theory	k_3	$O(0.3k_3k_2mN)$	k_4	$O(16k_4k_2mN)$	k_5	k_6	$O(0.3k_5k_6k_2mN)$
S14 L1 30 × 30	12.2	28.55	6	478.33	7	6.5	121.63
S14 L1 20 × 20	16.0	37.65	7	630.19	7	7.4	145.95
S14 L1 10 × 10	40.8	95.28	7	1402.3	7	13.3	294.99
S14 L1 5 × 5	12.2	743.36	7	11026	7	6.1	3065.02
S14 L3 5 × 5	12.1	743.36	7	11368	7	6.1	2923.05
S14 L5 5 × 5	12.1	743.36	8	11967	7	6.1	2960.71
Fracture exp	11.6	12.46	8	246.20	7	7.9	52.96
S1 img 2	7.3	46.07	5	338.27	6	2.9	150.25
S1 img 4	6.4	40.78	4	303.01	6	2.5	131.31
S1 img 6	5.9	38.48	4	307.18	6	2.5	127.08
S1 img 8	6.2	40.76	4	305.13	6	2.5	133.19
S1 img 10	6.3	39.95	5	331.99	6	2.6	133.03
S1 img 12	6.2	40.07	4	305.27	6	2.5	132.52
S1 img 14	6.0	38.40	4	304.49	6	2.5	126.21
S1 img 16	6.5	41.51	4	305.07	6	2.6	134.10
S1 img 18	7.4	47.16	5	330.98	6	3.1	156.03
S1 img 20	8.5	52.98	5	335.80	5	3.5	175.51

3.7 Conclusion

In this chapter, we have presented a new method, the augmented Lagrangian digital image correlation (ALDIC), for image matching. It combines the advantages of the two established methods, the speed of Local Subset DIC and the kinematic compatibility of Global DIC. We show in Section 3.5 through a series of case-studies using synthetic images that ALDIC provides superior accuracy compared to the established methods. We show in Section 3.6 that the computational cost of ALDIC is only a few times that of Local Subset DIC and less than that of Global DIC.

All the three DIC algorithms are summarized in Figure 3.11. ALDIC correlates subsets locally to find a displacement field as in Local Subset DIC, but then ties them together by introducing an auxiliary compatible displacement field. This leads to superior accuracy compared to Local Subset DIC for two reasons. First, the local

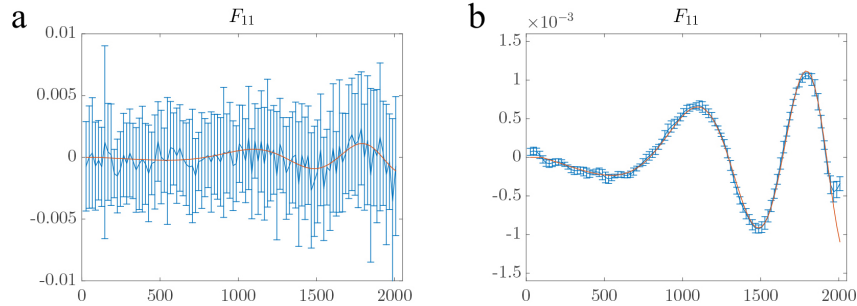


Figure 3.10: Comparison of the deformation gradient (F_{11} component) obtained using the Local Subset DIC (a) and ALDIC (b) for Sample 14 L1.

correlation or sub-problem 1 has some global information through the augmented Lagrangian and auxiliary field — see equation (3.6). Second, the auxiliary field leads to less noisy deformation gradients as shown in Figure 3.10. However, ALDIC is more expensive than Local Subset DIC because it requires the solution of a global problem (3.8) and it undergoes ADMM iterations. Still, this is not prohibitive: we see in Section 3.6 that it is only a few times that of Local Subset DIC.

Both ALDIC and Global DIC seek to impose compatibility. However, the point of departure is that ALDIC does not use a basis set to impose compatibility anywhere, but does so using an augmented Lagrangian. Therefore, the resulting operator ($\beta \mathbf{D}^T \mathbf{D} + \mu \mathbf{I}$) is the sum of the Laplacian and identity. This is universal, i.e., independent of the problem, displacement or image (though the matrix depends on the discretization). The nature of the operator and the universality allows us to either precompute the inverse (as we do here), or use a variety of established efficient methods (see for example [63]). In contrast, the operator M in (2.24) depends on the image, and moreover may be poorly conditioned depending on the image. Regularization can help with the conditioning and it is possible to address the computational cost. However, these require sophistication in their implementation and must be adopted to the problem at hand.

We hold β and μ fixed during ALDIC iterations, but it can be also updated at each iteration step to further speed up iterations, see Ref [46] §3.4.1. When varying penalty parameters are used, the scaled dual variables $\mathbf{W} = \boldsymbol{\nu}/\beta$ and $\mathbf{v} = \boldsymbol{\lambda}/\mu$ must also be rescaled after updating parameters β and μ .

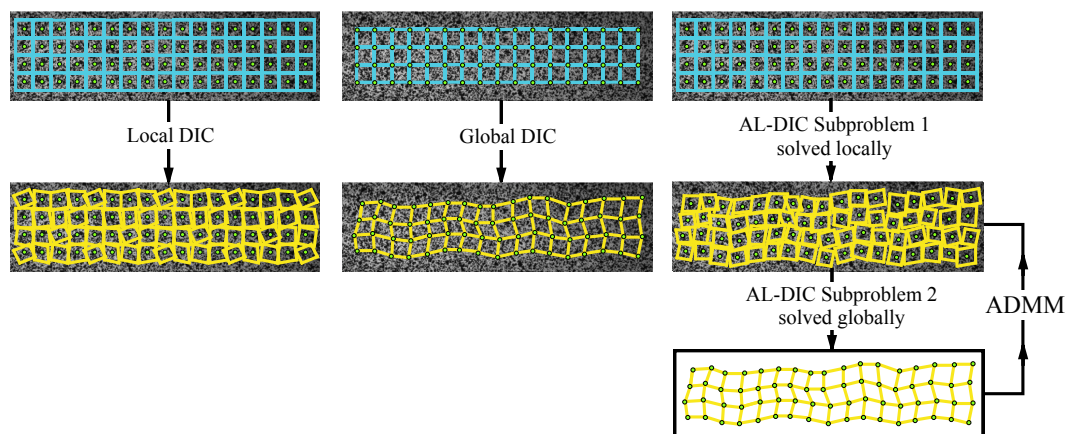


Figure 3.11: Comparison of Local Subset DIC, Global DIC, and ALDIC.

*Chapter 4***COMBINING DIC WITH IMAGE COMPRESSION
TECHNIQUES**

- [1] J Yang and K Bhattacharya. “Combining image compression with digital image correlation”. In: *Experimental Mechanics. Under review* (2018).

4.1 Introduction

Digital image correlation (DIC) infers the deformation from changes in the images of speckle patterns. Therefore, it is necessary to have high resolution images to obtain accurate displacement and strain values. Further, for the convergence and efficacy of the image conversion algorithms, it is useful to keep the relative deformation small. So it is common to acquire a series of images of the deformation process. For all these reasons, DIC requires the acquisition, transmission, and storage of large data-sets. The problem is only amplified in dynamical processes and the closely related digital volume correlation (DVC) method in three dimensions.

In the recent decades, a number of sophisticated image and data compression algorithms have been developed (see for example, [64]) that seek to represent digital images with less data in order to reduce storage and transmission costs. They are broadly divided into two categories: lossless and lossy. Lossless compression seeks to compress the data in such a way that the key original information can be reconstructed. This is useful, for example, where only the outline or borders of a particular image is critical. However, lossless methods cannot guarantee compression for all input data sets. This is indeed the case for speckle patterns. Lossy compression seeks to approximate the original data with a limited loss of information that is not perceptible to the eye. Lossy compression is broadly applicable and can lead to significant reduction in image sizes. Two widely used compression techniques use the wavelet transform (used in JPEG 2000 [65]) and the discrete cosine transform (DCT, used in JPEG [66]).

In this chapter, we study the combination of image compression and digital image correlation with the goal of reducing storage and transmission needs while maintaining the fidelity of the deformation information. We study two popular compression techniques – discrete cosine transform (DCT) and wavelet transform – with three

Table 4.1: Mean and standard deviation of image compression induced greyscale errors

	Mean value of Δf	Standard deviation of Δf
10 % wavelet compression	0.0532	3.5618
5 % wavelet compression	0.0408	6.7807
10 % DCT compression	-0.0163	5.4129
5 % DCT compression	0.0493	10.8765

image comparison methods described in Chapters 2 and 3.

4.2 Review image compression techniques

We consider two popular image compression techniques. The first technique is based on wavelet transform. A wavelet transform that localizes a signal both in space and frequency is applied to the image and only a fixed percentage of coefficients that have the greatest values are saved [65]. The second image compression technique is based on discrete cosine transform (DCT): the field is divided into blocks, discrete cosine transform is applied to each block, and only a fixed percentage of coefficients that have the greatest values are saved in each block [66]. Figure 4.1 shows the results of applying these techniques to a speckle pattern from the Society for Experimental Mechanics (SEM) DIC Challenge [60].

We compare the histogram of greyscale values of original images and compressed images in Figure 4.2 (a-b). Both wavelet and DCT image compression maintain the greyscale value distribution well. However, we find that there are sparse spikes in the histogram of DCT compressed images and the magnitude of some spikes can be large in the 5 % DCT compressed image. In contrast, there are no obvious spikes in the histogram of wavelet compressed images. To further understand the nature of the errors, let \tilde{f} denote the compression of the reference image f and Δf the difference: $f = \tilde{f} + \Delta f$. We list the mean and standard deviation of Δf in Table 4.1, and plot the histogram of Δf in Figure 4.2 (c-d). Both wavelet and DCT compression induced greyscale value errors obey standard Gaussian distributions with zero mean values. Further, wavelet compression outperforms DCT compression with smaller standard deviations at the same compression ratio.

The development of image compression algorithms gives us motivations to test whether we can combine image compression techniques with DIC methods to reduce image sizes while maintain the fidelity of the deformation information or not.

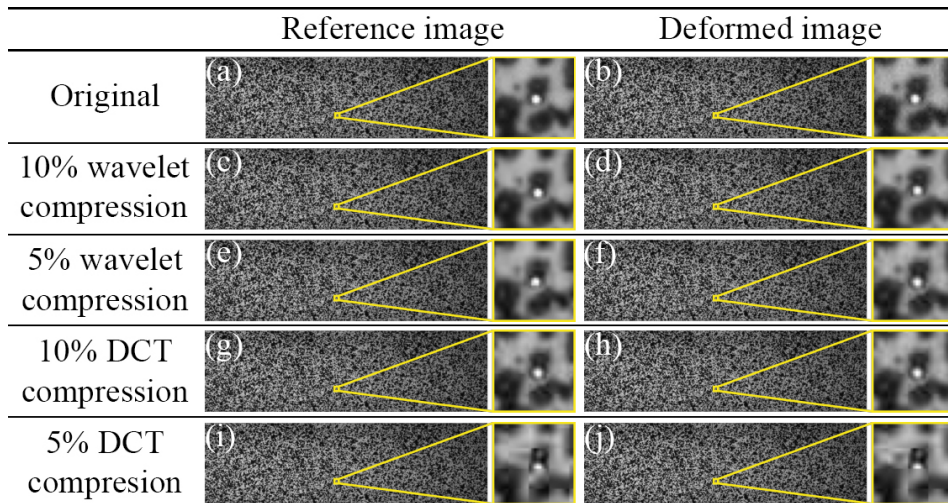


Figure 4.1: Image compression applied to the image from the SEM DIC Challenge Sample 14, Case L1. (a) Original reference image (Size: 1.2 MB). (b) Original deformed image (Size: 1.2 MB). (c-f) Wavelet compressed image whose size is only 10% of original reference (c) and deformed (d) image (Size: 120 KB) and 5% of original reference (e) and deformed (f) image (Size: 59 KB). (g-j) DCT compressed image whose size is only 10% of original reference (g) and deformed (h) image (Size: 124 KB) and 5% of original reference (i) and deformed (j) image (Size: 62 KB).

4.3 Combining image compression and DIC methods

Recall the discussion of the DIC algorithms in Chapter 2. In each of the DIC algorithms, we solve a system of linear equations: (2.13) in Local Subset DIC, (2.23) in Global DIC and (2.13) modified with (3.12), (3.13) in subproblem 1 of ALDIC. We may write each of these equations compactly as

$$Ax = b, \quad (4.1)$$

where the stiffness matrix A and the force vector b depends quadratically on the greyscale images and their gradients $f, g, \nabla f, \nabla g$.

Now, recall $f = \tilde{f} + \Delta f$, $g = \tilde{g} + \Delta g$ where \tilde{f}, \tilde{g} are the compressed images. If we pick any of our DIC algorithms and apply it to our compressed images \tilde{f}, \tilde{g} , we obtain a linear problem

$$\tilde{A}\tilde{x} = \tilde{b}, \quad (4.2)$$

where \tilde{A}, \tilde{b} are obtained using the compressed images. Since these quadratic, we find that $A = \tilde{A} + \Delta A$, $b = \tilde{b} + \Delta b$ where $\Delta A, \Delta b = O(\Delta f, \Delta g, \nabla(\Delta f), \nabla(\Delta g))$. Substituting this in (4.1) and using (4.2), we find that the error in the solution

$$\Delta x = x - \tilde{x} \approx A^{-1}[(\Delta A)x - \Delta b] = O(\Delta f, \Delta g, \nabla(\Delta f), \nabla(\Delta g)) \quad (4.3)$$

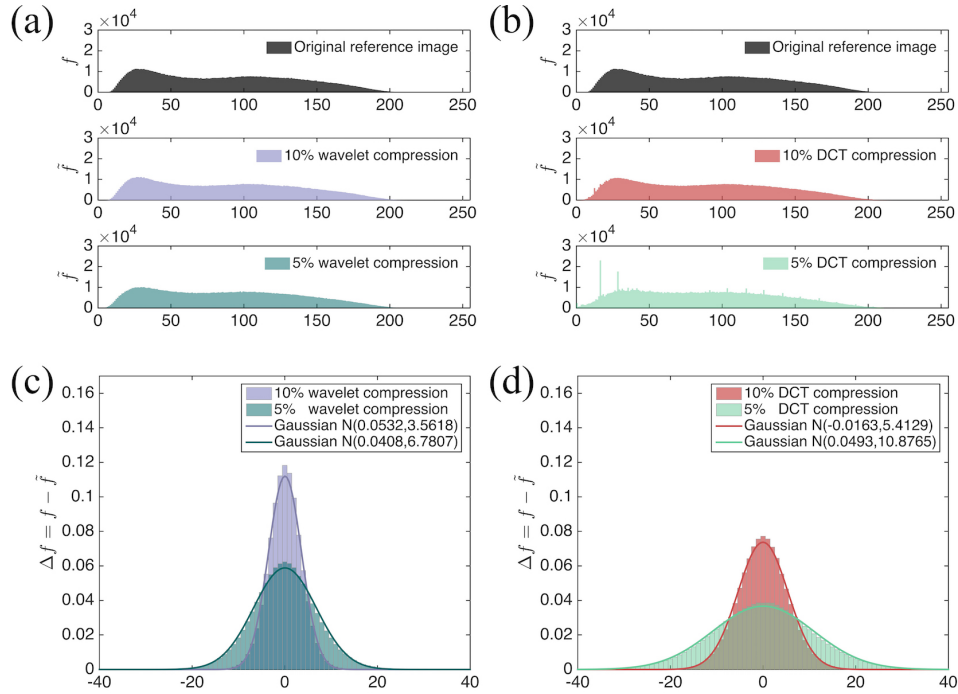


Figure 4.2: Errors in the images induced by wavelet and DCT image compression. (a-b) Histogram of greyscale values of wavelet (a) and DCT (b) compressed images. (c-d) Both wavelet (c) and DCT (d) compression induced greyscale value errors obey standard Gaussian distributions with different standard deviations.

since A, \tilde{A} are invertible. In other words, the error in displacements and strains is controlled by the compression error.

Still, there are important differences between the methods.

Local Subset DIC. In Local Subset DIC, the matrix A (or a, b, c shown in (2.14-2.16) in Section 2.2) involve the integration of terms that are quadratic in the image gradients ∇g . Since this is a speckle pattern, ∇g is noisy and so are the compression errors. Further, since the integration is on a small subset, ΔA can be large. Thus, though the errors are controlled, they can be large in Local Subset DIC. This is borne out by the examples in Section 4.4.

Global DIC. Here, the matrix A (or M in (2.24) in Section 2.4) also involve the integration of terms that are quadratic in the image gradients ∇f , but against the gradient of the shape function that is smooth on the scale of the speckle pattern. Further, the integration is over the entire image. Therefore, we expect smaller errors $\Delta A, \Delta b$, and thus small errors in displacement. This is also borne out by the

examples in Section 4.4.

ALDIC. Recall that the subproblem 1 of ALDIC is similar to Local Subset DIC. However, the matrix a is modified by μI (cf 3.12). This reduces the error. Further subproblem 2 of ALDIC or (3.8) is completely independent of the images; further, it involves the discrete Laplace operator $\mathbf{D}^T \mathbf{D}$ which is smoothing. For these reasons, we expect much smaller errors in ALDIC. This is also borne out by the examples in Section 4.4.

4.4 Demonstration

In this section, we study the combination of image compression and DIC through examples involving both synthetic images and real experimental images. We use two measures to evaluate the performance. The first is the *absolute root mean square error* when the exact deformation is known as in synthetic images:

$$\eta_1 = \sqrt{\frac{\sum_{\text{node}\#} |\mathbf{u} - \mathbf{u}_{\text{Exact}}|^2}{\text{node}\#}} \quad \text{and} \quad \xi_1 = \sqrt{\frac{\sum_{\text{node}\#} |\mathbf{e} - \mathbf{e}_{\text{Exact}}|^2}{\text{node}\#}}, \quad (4.4)$$

where \mathbf{u} is the inferred displacement with components u and v , and $\mathbf{u}_{\text{Exact}}$ is the exact displacement; \mathbf{e} is the inferred infinitesimal strain with components e_{xx} , e_{yy} , and e_{xy} , and $\mathbf{e}_{\text{Exact}}$ is the exact strain. This measures the accuracy of both the inherent DIC and the combined compression-DIC errors. To assess the latter, i.e., the errors induced in the DIC algorithm as a result of the compression, we use the *relative root mean square error*,

$$\eta_2 = \sqrt{\frac{\sum_{\text{node}\#} |\mathbf{u}_{\text{Compressed}} - \mathbf{u}_{\text{Original}}|^2}{\text{node}\#}} \quad \text{and} \quad \xi_2 = \sqrt{\frac{\sum_{\text{node}\#} |\mathbf{e}_{\text{Compressed}} - \mathbf{e}_{\text{Original}}|^2}{\text{node}\#}}, \quad (4.5)$$

where $\mathbf{u}_{\text{Compressed}}$ and $\mathbf{e}_{\text{Compressed}}$ are the DIC method solved deformation field using compressed images, while $\mathbf{u}_{\text{Original}}$ and $\mathbf{e}_{\text{Original}}$ are the solved displacement and strain field using original uncompressed images.

Synthetic example: Sample 14 of SEM 2D-DIC Challenge

SEM 2D-DIC Challenge is a series of benchmark problems [60]. We study the Sample 14 dataset which involve sinusoidal deformation with changing frequency. The reference image is taken from a real experiment, which is a typical DIC speckle

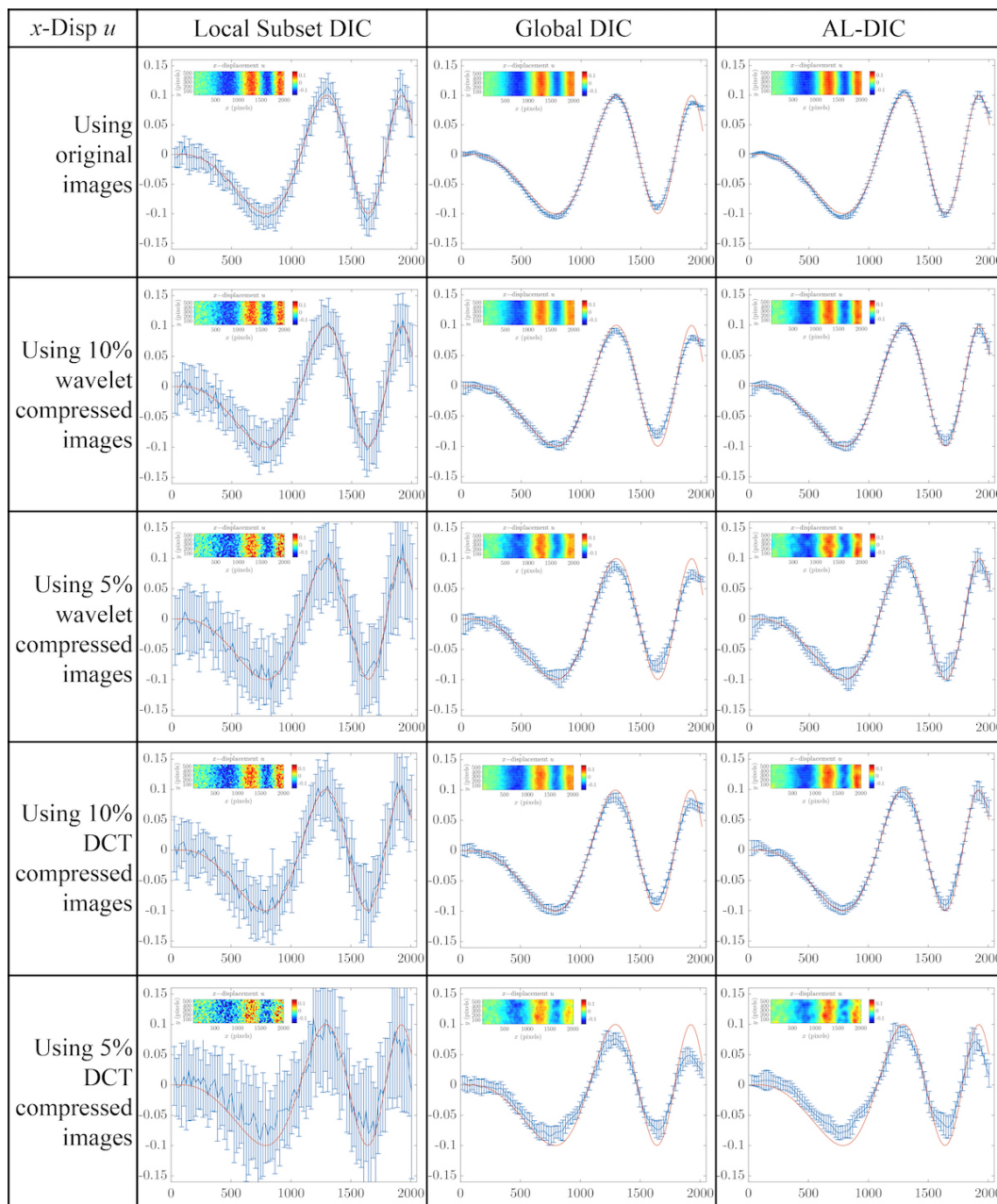


Figure 4.3: The displacement obtained from Sample 14 L1 data using the three DIC algorithms and the original as well as compressed images. The subset/mesh size is 20×20 .

pattern image with average radius around 8 pixels. The deformed images are formed synthetically by applying a known displacement field to the reference image.

We describe our results for the case labelled L1 in the dataset. We apply all three algorithms to the original images as well as compressed images. We show the displacements in Figure 4.3 and (the e_{xx}) strain in Figure 4.4. As expected,

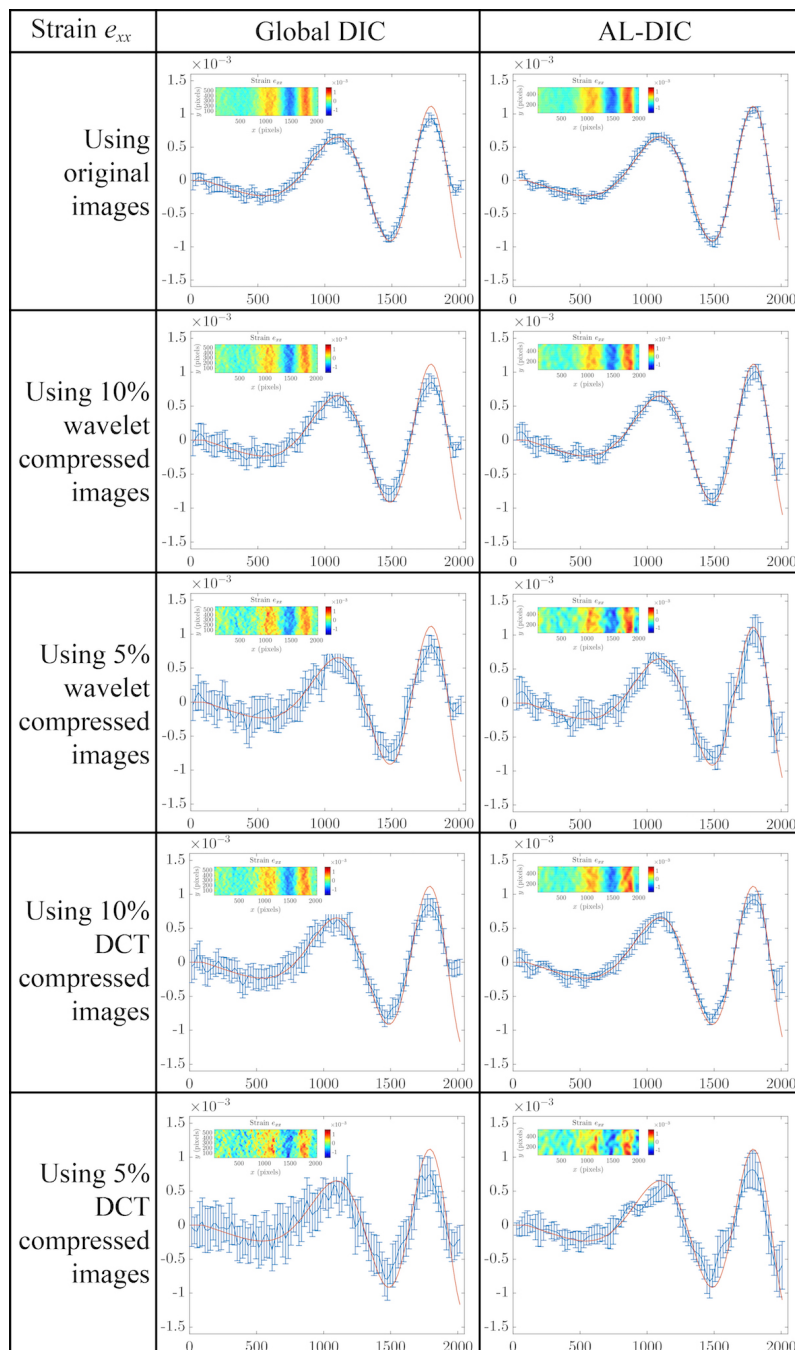


Figure 4.4: The strains obtained from Sample 14 L1 data using the three DIC algorithms and the original as well as compressed images. The subset/mesh size is 20×20 . The strains for the Local Subset DIC are not shown because they are very noisy.

Local Subset DIC is noisy, and the noise increases significantly with compression. Indeed, it is not possible to compute any meaningful strain without smoothing the data, but this requires experience and empirical intuition (too little smoothing leaves noise while too much smoothing introduces artifacts) and difficult to quantify. Therefore, we do not include the strains from Local Subset DIC. In the context of this work, we conclude that it is not possible to combine Local Subset DIC with image compression.

The two methods with global input, Global DIC and ALDIC, on the other hand do quite well even when we use compressed images. We see that the noise increases with compression, but not significantly. Global DIC method has more noise than ALDIC method, especially in strain at the high compression. We see that both methods have an artifact on the right boundary due to the boundary condition we use. It is more pronounced in Global DIC. Further, we begin to see artifacts in regions of rapidly changing strain (see the first peak from the right) in Global DIC. ALDIC on the other hand continues to accurately find the displacements and the strains even with only 5% of the original image with wavelet compression. DCT compression produces more error. Overall, we conclude that ALDIC and wavelet compression provide the optimal results.

This is confirmed in the absolute RMS errors shown in Table 4.2 and the relative RMS errors are shown in Table 4.3. These tables also show the effect of choosing different subset/mesh sizes. Importantly, the absolute RMS errors increase by about a factor two with the most parsimonious wavelet compression in Global DIC and ALDIC, but more in Local Subset DIC. Once again, we conclude that ALDIC and wavelet compression provide the optimal results.

In conclusion, this example shows that ALDIC and wavelet compression up to 5% of the original image provide good results for the displacement and strain fields. We have also analyzed other datasets from the SEM 2D-DIC Challenge. The results are similar as described in the supplementary material.

Experimental example: Heterogeneous fracture

We now turn to real experimental data from the fracture of a heterogeneous media shown in Figure 3.7 [62] (also [67] for more details).

Figures 4.5 and 4.6 show the displacement and strain fields obtained from the three algorithms using original, wavelet, and DCT compressed images of the heterogeneous fracture specimen. Once again, we do not show the strain fields for Local

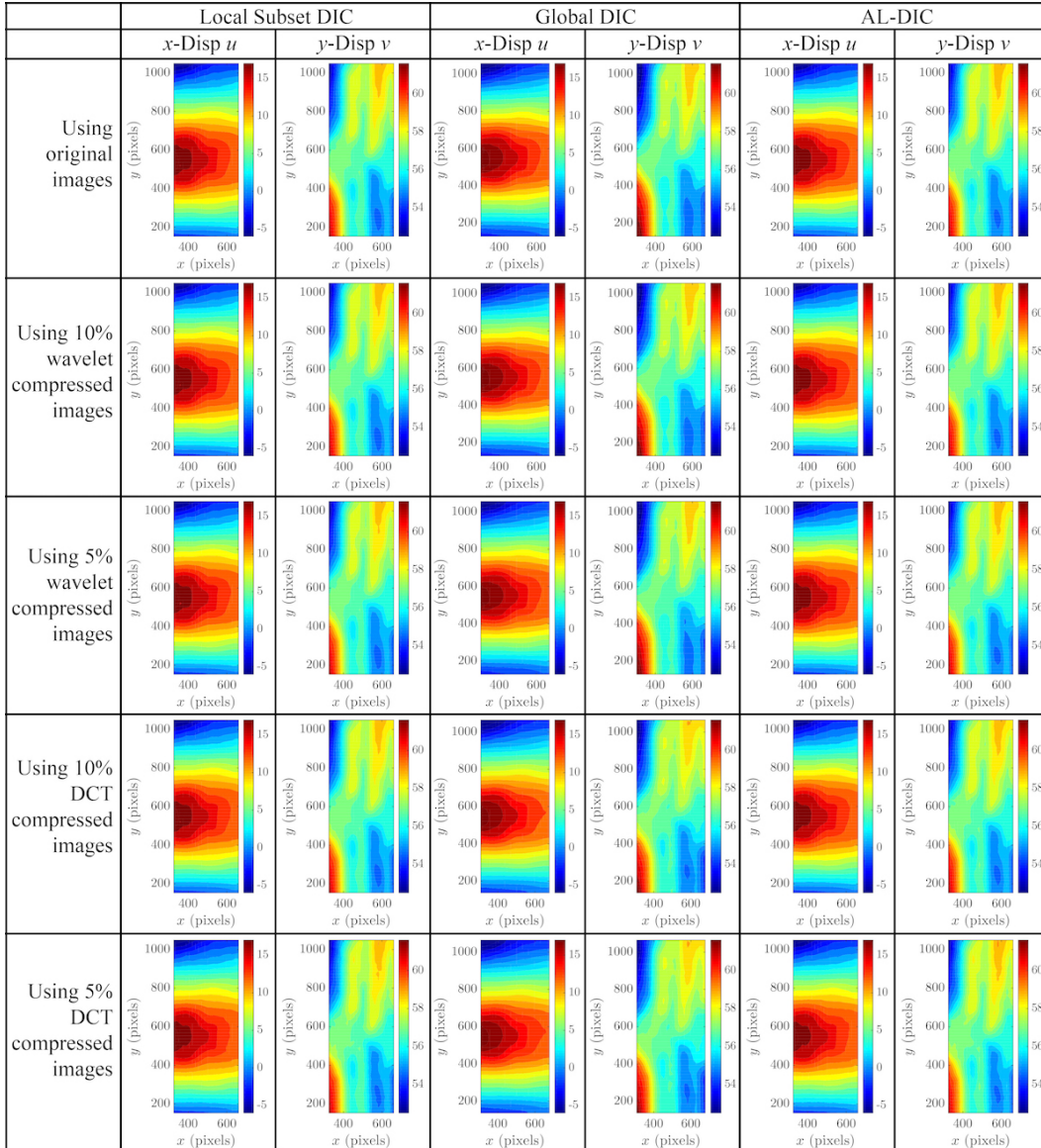


Figure 4.5: The displacement fields obtained from Local Subset DIC, Global DIC, and ALDIC using original, wavelet, and DCT compressed images of the heterogeneous fracture specimen.

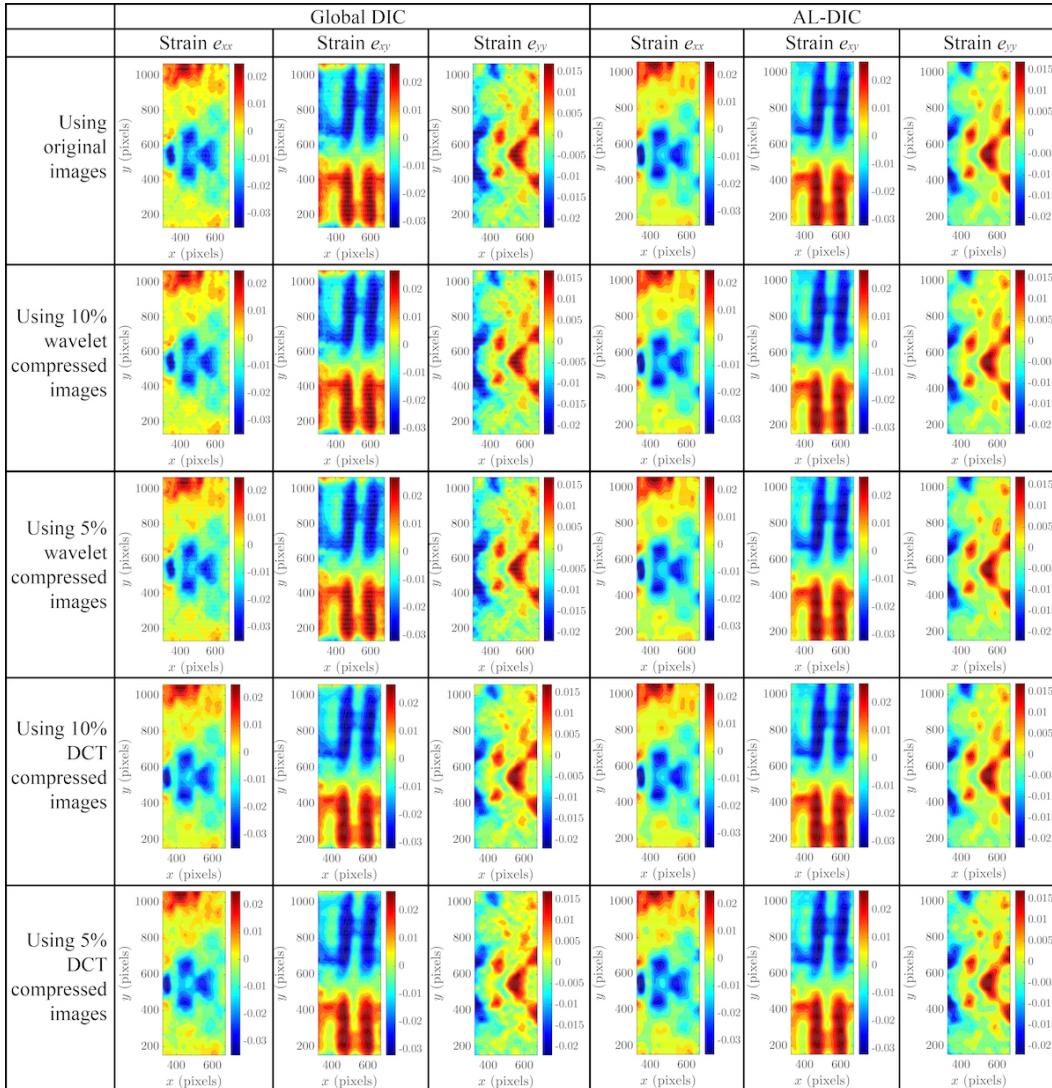


Figure 4.6: The strain fields obtained from Global DIC and ALDIC using original, wavelet, and DCT compressed images of the heterogeneous fracture specimen.

Table 4.2: Absolute RMS errors of Sample 14 L1 data using three DIC algorithms applied to the original images and the compressed images with different subset/mesh

Index	Mesh	Image compression	Local Subset DIC	Global DIC	ALDIC
Displacement η_1 (pixels)	30×30	Original	0.0199	0.0079	0.0056
		Wavelet 10 %	0.0312	0.0108	0.0080
		Wavelet 5 %	0.0514	0.0162	0.0127
		DCT 10 %	0.0409	0.0130	0.0103
		DCT 5 %	0.0652	0.0251	0.0207
	20×20	Original	0.0323	0.0082	0.0067
		Wavelet 10 %	0.0514	0.0112	0.0088
		Wavelet 5 %	0.0846	0.0162	0.0164
		DCT 10 %	0.0674	0.0131	0.0126
		DCT 5 %	0.1055	0.0249	0.0260
	10×10	Original	0.0990	0.0088	0.0165
		Wavelet 10 %	0.1416	0.0111	0.0232
		Wavelet 5 %	0.2086	0.0157	0.0351
		DCT 10 %	0.2176	0.0139	0.0288
		DCT 5 %	—	—	—
Strain ξ_1	30×30	Original	—	2.08×10^{-4}	9.57×10^{-5}
		Wavelet 10 %	—	2.65×10^{-4}	1.53×10^{-4}
		Wavelet 5 %	—	3.81×10^{-4}	2.10×10^{-4}
		DCT 10 %	—	3.22×10^{-4}	1.85×10^{-4}
		DCT 5 %	—	5.20×10^{-4}	2.50×10^{-4}
	20×20	Original	—	2.35×10^{-4}	1.62×10^{-4}
		Wavelet 10 %	—	3.11×10^{-4}	2.30×10^{-4}
		Wavelet 5 %	—	4.40×10^{-4}	3.72×10^{-4}
		DCT 10 %	—	3.79×10^{-4}	2.56×10^{-4}
		DCT 5 %	—	6.05×10^{-4}	3.79×10^{-4}
	10×10	Original	—	2.84×10^{-4}	8.19×10^{-4}
		Wavelet 10 %	—	3.77×10^{-4}	0.0010
		Wavelet 5 %	—	5.47×10^{-4}	0.0015
		DCT 10 %	—	4.71×10^{-4}	8.85×10^{-4}
		DCT 5 %	—	—	—

Subset DIC because they are very noisy. Table 4.4 shows the relative RMS errors. Once again, we see that ALDIC and wavelet compression gives good results.

Computation time

The computation times for the various examples are given in Table 4.5. All computations were conducted on a personal computer with an Intel (R) Xeon(R) CPU E5-2650 v3 2.30 GHz (2 Processors), RAM 32.0 GB Memory, 64-bit operating system. As we reported earlier, Local Subset DIC is the fastest and Global DIC is the slowest. We note that compression can marginally reduce the computation time. However, if we compress too much, then we have difficulty with convergence and this increases the computation time, especially in ALDIC.

4.5 Conclusion

Digital image compression requires high resolution images to accurately capture the speckle patterns, and this makes storage and transmission expensive. In this

Table 4.3: Relative RMS errors of Sample 14 L1 data using three DIC algorithms applied to the original images and the compressed images with different subset/mesh

Index	Mesh	Image compression	Local Subset DIC	Global DIC	ALDIC
Displacement η_2 (pixels)	30×30	Wavelet 10 %	0.0275	0.0078	0.0064
		Wavelet 5 %	0.0503	0.0145	0.0121
		DCT 10 %	0.0370	0.0101	0.0083
		DCT 5 %	0.0658	0.0248	0.0210
	20×20	Wavelet 10 %	0.0445	0.0078	0.0085
		Wavelet 5 %	0.0828	0.0140	0.0167
		DCT 10 %	0.0619	0.0101	0.0117
		DCT 5 %	0.1063	0.0249	0.0280
	10×10	Wavelet 10 %	0.1399	0.0076	0.0235
		Wavelet 5 %	0.2181	0.0138	0.0371
		DCT 10 %	0.2184	0.0104	0.0280
		DCT 5 %	—	—	—
Strain ξ_2	30×30	Wavelet 10 %	—	1.27×10^{-4}	6.05×10^{-5}
		Wavelet 5 %	—	2.34×10^{-4}	1.23×10^{-4}
		DCT 10 %	—	1.74×10^{-4}	7.90×10^{-4}
		DCT 5 %	—	3.45×10^{-4}	1.53×10^{-4}
	20×20	Wavelet 10 %	—	1.56×10^{-4}	1.32×10^{-4}
		Wavelet 5 %	—	2.72×10^{-4}	2.51×10^{-4}
		DCT 10 %	—	2.11×10^{-4}	1.56×10^{-4}
		DCT 5 %	—	4.08×10^{-4}	2.73×10^{-4}
	10×10	Wavelet 10 %	—	1.99×10^{-4}	6.79×10^{-4}
		Wavelet 5 %	—	3.51×10^{-4}	0.0011
		DCT 10 %	—	2.76×10^{-4}	5.89×10^{-4}
		DCT 5 %	—	—	—

Table 4.4: Relative RMS errors of fracture experiment using three DIC algorithms applied to the compressed images compared with results using the original images

Index	Mesh	Image compression	Local Subset DIC	Global DIC	ALDIC
Displacement η_2 (pixels)	16×16	Wavelet 10 %	0.0082	0.0030	0.0074
		Wavelet 5 %	0.0296	0.0118	0.0262
		DCT 10 %	0.0213	0.3888	0.0191
		DCT 5 %	0.0803	0.3908	0.0654
Strain ξ_2	16×16	Wavelet 10 %	—	1.71×10^{-4}	2.28×10^{-4}
		Wavelet 5 %	—	6.59×10^{-4}	8.06×10^{-4}
		DCT 10 %	—	0.0066	5.65×10^{-4}
		DCT 5 %	—	0.0067	0.0020

paper, we study the combination of DIC with image compression. We study three DIC algorithms (Local Subset DIC, Global DIC, and augmented Lagrangian DIC (ALDIC)), and two image compression methodologies (discrete cosine transform (DCT) and wavelet transform). Overall, we find through both synthetic and real data that the DIC algorithms are able to recover deformation information from images compressed to 5% of their original size albeit with some errors and noise.

Local Subset DIC is the most error-prone and noisy. This is to be expected since Local Subset DIC relies only on local information, and there is no mechanism by which any loss of information due to compression can be filled in. Further, since the loss of information is not uniform, image compression can lead to noise in

Table 4.5: Computation time of the numerical demonstrations

DIC Image	Mesh	Image compression	Local Subset DIC	Global DIC	ALDIC
Sample 14 L1	30×30	Original	28.55	478.33	121.63
		Wavelet 10 %	23.06	380.56	102.37
		Wavelet 5 %	22.78	370.80	106.40
		DCT 10 %	24.80	433.42	112.25
		DCT 5 %	25.58	435.38	118.67
Sample 14 L1	20×20	Original	37.65	630.19	145.95
		Wavelet 10 %	29.18	476.22	119.69
		Wavelet 5 %	30.03	472.20	122.29
		DCT 10 %	30.86	550.76	128.74
		DCT 5 %	33.53	560.79	157.71
Sample 14 L1	10×10	Original	95.28	1402.3	294.99
		Wavelet 10 %	84.99	1284.3	265.38
		Wavelet 5 %	95.56	1284.3	285.03
		DCT 10 %	86.99	1137.3	275.56
		DCT 5 %	—	—	—
Fracture Exp	16×16	Original	13.67	156.21	55.67
		Wavelet 10 %	11.02	182.66	45.55
		Wavelet 5 %	11.16	179.19	45.49
		DCT 10 %	11.00	147.99	46.60
		DCT 5 %	13.41	198.21	52.47

Local Subset DIC. The other two methods – Global DIC and ALDIC – are able to fill to some extent missing local information from neighboring regions. Therefore they are less error-prone and less noisy. However, this can introduce some artifacts due to mollification. ALDIC appears to find the right balance between filling in information and keeping mollification to a minimum.

Among the image compression algorithms, wavelet transform do better than DCT. In DCT, we first cut the whole image into 8×8 pixel blocks and ignore the high frequency coefficients within each block. Therefore, the interfaces between the blocks are not necessarily smooth (cf Figure 4.1). Wavelet transform, on the other hand, does not have this difficulty.

Chapter 5

ADAPTIVE MESH ALDIC

- [1] J Yang and K Bhattacharya. “Multigrid adaptive mesh digital image correlation”. In: *To be submitted* (2018).
- [2] J Yang and K Bhattacharya. “Fast adaptive global digital image correlation”. In: *Advancement of Optical Methods & Digital Image Correlation in Experimental Mechanics*. Vol. 3. 2019. Chap. 7. DOI: https://doi.org/10.1007/978-3-319-97481-1_7.

DIC is especially useful in the study of complex phenomena with very heterogeneous strain fields at various length scales. In heterogeneous strain fields, a uniform local subset or global finite element mesh discretization leads to very different accuracy in the different regions; however, the precision only depends on the worst subset/element. In other words, a high precision requires a very fine mesh if it is uniform. We can overcome this by using an adaptive mesh in DIC to save computation time with little loss in accuracy. In this chapter, we apply two adaptive approaches, Kuhn triangulation and Quadtree mesh, to ALDIC.

5.1 Introduction

In DIC, adaptivity can have three different meanings.

1. **Image.** First, images can be taken with adaptive spatial resolutions. Passieux *et al.* [68] use more than one camera to take DIC experimental images with different focal distances and image scale ratios. They take relatively low spatial resolutions farfield images using one camera far away to cover larger areas for small strain deformation region, and they take relatively high resolution nearfield images with another close camera to analyze large strain deformation regions. Images can also be adaptively filtered to have various resolutions to achieve multiscale resolution analysis [69].
2. **Mesh.** Second, the deformation field can be represented using a finite number of sampling points in a self-adaptive manner. For example, Yuan *et al.* [70] modified Local Subset DIC to automatically sample more points where the weighted zero-normalized summation of squared differences correlation

function is poorly satisfied. Ronovsky *et al.* [35] extended the Global DIC method to refine finite elements where the modified the summation of squared differences correlation function is relatively large.

3. **Subset.** Third, the deformation of different subsets/elements can be characterized using hierarchical bases. For example, Wittevrongel *et al.* [71] applies p-adaptive finite element method in Global DIC where larger gradient deformation regions are described with higher order finite elements. P-adaptive finite element-based Global DIC method has only been reported to work well with very big subsets/elements whose sizes are from 50×50 pixels to 100×100 pixels. Similarly in Local Subset DIC, the order of local subset shape function and the size of local subset can also be chosen adaptively to improve the overall accuracy [72, 73].

In this chapter, we consider the second aspect described above, where the adaptivity is in mesh used for representation of the deformation.

5.2 Variational formulation of regularized Global DIC

Recall from Section 2.5 that Global DIC optimization problem is usually severely ill-posed. Thus regularization terms are added to the original correlation function. Here we apply a gradient regularier to constraint the deformation field with certain smoothness. At the $(n + 1)$ th step,

$$C_g = \int_{\Omega} [f(\mathbf{X}) - g(\mathbf{X} + \mathbf{u}_n + \delta\mathbf{u})]^2 + \alpha |\nabla(\mathbf{u}_n + \delta\mathbf{u})|^2 d\mathbf{X}, \quad \alpha > 0, \quad (5.1)$$

where \mathbf{u}_n is the solved displacements at the previous n -th step, $\delta\mathbf{u}$ is the update of displacements at the $(n + 1)$ th step. The corresponding stationarity principle in weak form is given by

$$\begin{aligned} \int_{\Omega} \left\{ \alpha \nabla(\mathbf{u}_0 + \delta\mathbf{u}) : \nabla\varphi - [f(\mathbf{X}) - g(\mathbf{X} + \mathbf{u}_n + \delta\mathbf{u})] \frac{\partial g(\mathbf{X} + \mathbf{u}_n)}{\partial \mathbf{X}} \cdot \varphi \right\} d\mathbf{X} = 0, \\ \int_{\partial\Omega} \{ (\nabla(\mathbf{u}_n + \delta\mathbf{u})^T \varphi) \cdot \mathbf{n} \} dS = 0, \quad \forall \varphi \in H^1(\Omega). \end{aligned} \quad (5.2)$$

The “:” operator follows computation rule $\mathbf{A} : \mathbf{B}_G = \sum_i \sum_j A_{ij} B_{ij}$.

In the DIC experiments, if the deformation field is not large, we can take Taylor expansion of the deformed image grayscale and only keep the first order term

$g(\mathbf{X} + \mathbf{u}_n + \delta \mathbf{u}) \approx g(\mathbf{X} + \mathbf{u}_n) + \partial g(\mathbf{X} + \mathbf{u}_n)/\partial \mathbf{X} \cdot \delta \mathbf{u}$ to linearize the above equation. We can further approximate $\partial g(\mathbf{X} + \mathbf{u}_n)/\partial \mathbf{X}$ using $\partial f(\mathbf{X})/\partial \mathbf{X}$, which only needs to compute once to save computation time:

$$\int_{\Omega} \left\{ \left[\frac{\partial f(\mathbf{X})}{\partial \mathbf{X}} \cdot \delta \mathbf{u} \right] \left[\frac{\partial f(\mathbf{X})}{\partial \mathbf{X}} \cdot \varphi \right] \right\} d\mathbf{X} = \int_{\Omega} \left\{ -\alpha \nabla(\mathbf{u}_n + \delta \mathbf{u}) : \nabla \varphi + [f(\mathbf{X}) - g(\mathbf{X} + \mathbf{u}_n)] \left[\frac{\partial f(\mathbf{X})}{\partial \mathbf{X}} \cdot \varphi \right] \right\} d\mathbf{X}, \quad \forall \varphi \in H^1(\Omega). \quad (5.3)$$

Assuming there are natural boundary conditions, the above variational problem can further be written into the following form where $\mathfrak{B}_G : H^1 \times H^1 \rightarrow \mathbb{R}$ is a symmetric continuous bilinear form:

$$\delta \mathbf{u} \in H^1, \text{ s.t. } \quad \mathfrak{B}_G[\delta \mathbf{u}, \varphi] = \langle \mathbf{p}, \varphi \rangle, \quad \text{for all } \varphi \in H^1(\Omega), \quad (5.4)$$

where

$$\begin{cases} \mathfrak{B}_G[\delta \mathbf{u}, \varphi] = \int_{\Omega} \left\{ \delta \mathbf{u} \cdot \left[\frac{\partial f(\mathbf{X})}{\partial \mathbf{X}} \right] \left[\frac{\partial f(\mathbf{X})}{\partial \mathbf{X}} \right]^T \cdot \varphi + \alpha \nabla(\delta \mathbf{u}) : \nabla \varphi \right\} d\mathbf{X} \\ \langle \mathbf{p}, \varphi \rangle = \int_{\Omega} \left\{ -\alpha \nabla \mathbf{u}_n : \nabla \varphi + [f(\mathbf{X}) - g(\mathbf{X} + \mathbf{u}_n)] \left[\frac{\partial f(\mathbf{X})}{\partial \mathbf{X}} \cdot \varphi \right] \right\} d\mathbf{X}. \end{cases} \quad (5.5)$$

Lemma 1. The symmetric bilinear operator $\mathfrak{B}_G : H^1(\Omega) \times H^1(\Omega) \rightarrow \mathbb{R}$ in (5.4-5.5) is coercive if displacement $\delta \mathbf{u}$ is not orthogonal to image grayscale gradients $\nabla f(\mathbf{X})$ a.e. in Ω .

Proof. $\forall \delta \mathbf{u} \in H^1(\Omega)$, there exists $0 < M < \infty$, such that

$$\int_{\Omega} |\delta \mathbf{u}|^2 d\mathbf{X} < M. \quad (5.6)$$

By further assuming $\delta \mathbf{u}$ is not orthogonal to $\frac{\partial f(\mathbf{X})}{\partial \mathbf{X}}$ a.e. in Ω , thus there exists a local small sphere domain $\Omega_0 \subset \Omega$, where $\left| \delta \mathbf{u} \cdot \frac{\partial f(\mathbf{X})}{\partial \mathbf{X}} \right| > 0$ a.e. inside Ω_0 . So we have

$$\begin{aligned} \mathfrak{B}_G(\delta \mathbf{u}, \delta \mathbf{u}) &= \int_{\Omega} \left\{ \left| \delta \mathbf{u} \cdot \frac{\partial f(\mathbf{X})}{\partial \mathbf{X}} \right|^2 + \alpha |\nabla \delta \mathbf{u}|^2 \right\} d\mathbf{X} \\ &\geq \int_{\Omega_0} \left| \delta \mathbf{u} \cdot \frac{\partial f(\mathbf{X})}{\partial \mathbf{X}} \right|^2 d\mathbf{X} + \int_{\Omega} \alpha |\nabla \delta \mathbf{u}|^2 d\mathbf{X} \\ &> \frac{\int_{\Omega_0} \left| \delta \mathbf{u} \cdot \frac{\partial f(\mathbf{X})}{\partial \mathbf{X}} \right|^2 d\mathbf{X}}{M} \int_{\Omega} |\delta \mathbf{u}|^2 d\mathbf{X} + \int_{\Omega} \alpha |\nabla \delta \mathbf{u}|^2 d\mathbf{X} \\ &> C_1 |\delta \mathbf{u}|_{H^1(\Omega)}^2, \end{aligned} \quad (5.7)$$

where $C_1 = \min\left\{\frac{\int_{\Omega_0} |\delta \mathbf{u} \cdot \frac{\partial f(\mathbf{X})}{\partial \mathbf{X}}|^2 d\mathbf{X}}{M}, \alpha\right\}$ is a finite positive number. \blacksquare

Theorem 1. (Modified Lax-Milgram Theorem) Let $\mathfrak{B}_G : H^1(\Omega) \times H^1(\Omega) \rightarrow \mathbb{R}$ be a continuous bilinear form that is coercive, then the equation system Eqs(5.4-5.5) has a unique solution $\delta \mathbf{u} \in H^1(\Omega)$, which depends continuously on \mathbf{p} .

Proof. See Appendix B. \blacksquare

5.3 Variational formulation of ALDIC Subproblem 2

In this section, we apply adaptive mesh onto ALDIC method with ADMM scheme. In ADMM, there are two subproblems, and our mesh refinement criteria are based on a posteriori error estimates of both these two subproblems. Before discussing how to estimate a posteriori errors, we first describe how to solve ALDIC ADMM scheme for arbitrary non-uniform mesh. The first subproblem of ADMM decouples all the nodes into isolated ones, which can be easily solved with non-uniform mesh. The second subproblem is a global update whose variational formulation is derived as follows, and we can solve it using the finite element method.

In ALDIC subproblem 2 global update, while holding $\{\mathbf{F}^{k+1}\}$, $\{\mathbf{u}^{k+1}\}$, $\{\mathbf{W}^k\}$, $\{\mathbf{v}^k\}$ fixed, we minimize \mathcal{L} over $\{\hat{\mathbf{u}}\}$ to obtain $\{\hat{\mathbf{u}}^{k+1}\}$, which is a global problem independent of the reference and deformed images f, g .

$$\begin{aligned} \{\hat{\mathbf{u}}^{k+1}\} &= \arg \min_{\{\hat{\mathbf{u}}\}} \mathcal{L}(\{\mathbf{F}^{k+1}\}, \{\mathbf{u}^{k+1}\}, \{\hat{\mathbf{u}}\}, \{\mathbf{W}^k\}, \{\mathbf{v}^k\}) \\ &= \arg \min_{\{\hat{\mathbf{u}}\}} \int_{\Omega} \left(\frac{\beta}{2} |\mathbf{D}\hat{\mathbf{u}} - \mathbf{F}^{k+1} + \mathbf{W}^k|^2 + \frac{\mu}{2} |\hat{\mathbf{u}} - \mathbf{u}^{k+1} + \mathbf{v}^k|^2 \right) d\mathbf{X}. \end{aligned} \quad (5.8)$$

We define functional $\tilde{\mathcal{L}} := \int_{\Omega} \left(\frac{\beta}{2} |\mathbf{D}\hat{\mathbf{u}} - \mathbf{F}^{k+1} + \mathbf{W}^k|^2 + \frac{\mu}{2} |\hat{\mathbf{u}} - \mathbf{u}^{k+1} + \mathbf{v}^k|^2 \right) d\mathbf{X}$. The Euler-Lagrange Equation in weak form is given by

$$\begin{aligned} \int_{\Omega} \{\beta \nabla \hat{\mathbf{u}} : \nabla \boldsymbol{\varphi} + \mu \hat{\mathbf{u}} \cdot \boldsymbol{\varphi}\} d\mathbf{X} &= \int_{\Omega} \{\beta(\mathbf{F} - \mathbf{W}) : \nabla \boldsymbol{\varphi} + \mu(\mathbf{u} - \mathbf{v}) \cdot \boldsymbol{\varphi}\} d\mathbf{X} \\ \int_{\partial\Omega} \{\nabla \hat{\mathbf{u}}^T \boldsymbol{\varphi} \cdot \mathbf{n}\} dS &= 0, \quad \int_{\partial\Omega} \{(\mathbf{F} - \mathbf{W})^T \boldsymbol{\varphi} \cdot \mathbf{n}\} dS = 0, \quad \forall \boldsymbol{\varphi} \in H^1(\Omega). \end{aligned} \quad (5.9)$$

Assuming there are natural boundary conditions, the above variational problem can further be written into the following form where $\mathfrak{B}_{AL} : H^1 \times H^1 \rightarrow \mathbb{R}$ is a

continuous symmetric bilinear form:

$$\hat{\mathbf{u}} \in H^1, \text{ s.t.} \quad \mathfrak{B}_{AL}[\hat{\mathbf{u}}, \varphi] = \langle \mathbf{q}, \varphi \rangle, \text{ for all } \varphi \in H^1(\Omega), \quad (5.10)$$

where

$$\begin{cases} \mathfrak{B}_{AL}[\hat{\mathbf{u}}, \varphi] = \int_{\Omega} \{\beta \nabla \hat{\mathbf{u}} : \nabla \varphi + \mu \hat{\mathbf{u}} \cdot \varphi\} d\mathbf{X} \\ \langle \mathbf{q}, \varphi \rangle_{H^1} = \int_{\Omega} \{\beta(\mathbf{F} - \mathbf{W}) : \nabla \varphi + \mu(\mathbf{u} - \mathbf{v}) \cdot \varphi\} d\mathbf{X}. \end{cases} \quad (5.11)$$

Lemma 2. The symmetric bilinear operator $\mathfrak{B}_{AL} : H^1(\Omega) \times H^1(\Omega) \rightarrow \mathbb{R}$ in (5.10-5.11) is coercive.

Proof. $\forall \hat{\mathbf{u}} \in H^1(\Omega)$, there holds

$$\mathfrak{B}_{AL}(\hat{\mathbf{u}}, \hat{\mathbf{u}}) = \int_{\Omega} \{\beta |\nabla \hat{\mathbf{u}}|^2 + \mu |\hat{\mathbf{u}}|^2\} d\mathbf{X} \geq C_3 |\hat{\mathbf{u}}|_{H^1(\Omega)}^2, \quad (5.12)$$

where C_3 is some finite positive constant. ■

Using the similar proof with Theorem 1, we can apply Lax-Milgram theorem again to guarantee the well-posedness of (5.10-5.11).

Theorem 2. Equation system (5.10-5.11) has a unique solution $\hat{\mathbf{u}} \in H^1(\Omega)$, which depends continuously on \mathbf{q} .

5.4 Discretization

In this section, we recall some important concepts from [74] about Galerkin discretization. To solve the variational problem in the regularized Global DIC system (5.4-5.5) and the ALDIC subproblem 2 (5.10-5.11) numerically, we restrict the continuous space $H^1(\Omega)$ to finite dimensional subspaces of dimension $N < \infty$. For simplicity, we use symbol $\{\mathfrak{B}, \mathbf{u}, \mathbf{U}_N, \mathbf{p}, C\}$ to represent the notations $\{\mathfrak{B}_G, \delta \mathbf{u}, \delta \mathbf{U}_N, \mathbf{p}, C_1\}$ in the regularized Global DIC variational formulation (Section 5.2), and to represent $\{\mathfrak{B}_{AL}, \hat{\mathbf{u}}, \hat{\mathbf{U}}_N, \mathbf{q}, C_3\}$ in ALDIC subproblem 2 (Section 5.3).

Definition 1. For $N \in \mathbb{N}$, let $H_N^1(\Omega) \subset H^1(\Omega)$ be subspace of dimension N , e.g., $H_N^1(\Omega) := \{v \in C(\bar{\Omega}) | v_T \in \mathbb{P}_n(T), \text{ for all elements } T \in \mathcal{J}\}$, where \mathcal{J} is a conforming triangulation or Quadtree mesh of Ω (cf Definitions 7, 14) and $\mathbb{P}_n(T)$ stands for the space of polynomials with degree not greater than n over element T . Then a solution \mathbf{U}_N to

$$\mathbf{U}_N \in H_N^1(\Omega) : \quad \mathfrak{B}[\mathbf{U}_N, \varphi_N] = \langle \mathbf{p}, \varphi_N \rangle \quad \forall \varphi_N \in H_N^1(\Omega) \quad (5.13)$$

is called *Galerkin solution*.

For conforming discretizations (cf Definitions 7, 14), the bilinear form \mathfrak{B} is well defined and continuous on the discrete spaces $H^1(\Omega) \times H^1(\Omega)$. Lemma 1 can be extended in the conforming discretizations such that

$$\mathfrak{B}[\mathbf{U}, \mathbf{U}] \geq C |\mathbf{U}|_{H^1(\Omega)}^2, \quad \forall \mathbf{U} \in H_N^1(\Omega), \quad (5.14)$$

where $C_1 > 0$ is the same coercivity constant with the continuous spaces case in Lemma 1. We can further extend Theorem 1 into the discrete version.

Theorem 3. (Existence and uniqueness of the Galerkin solution). Let $H_N^1(\Omega) \subset H^1(\Omega)$, then for any $\mathbf{p} \in H_N^{-1}(\Omega)$ there exists a unique Galerkin solution $\mathbf{U}_N \in H_N^1(\Omega)$:

$$\mathbf{U}_N \in H_N^1(\Omega) : \quad \mathfrak{B}[\mathbf{U}_N, \boldsymbol{\varphi}] = \langle \mathbf{p}, \boldsymbol{\varphi} \rangle \quad \forall \boldsymbol{\varphi} \in H_N^1(\Omega) \quad (5.15)$$

if (5.14) is satisfied.

Proof. Applying the Riesz representation theorem, there exists a linear discrete operator $\mathbf{B}_N \in L(H_N^1(\Omega); H_N^1(\Omega))$ such that

$$\langle \mathbf{B}_N \mathbf{U}, \mathbf{W} \rangle = \mathfrak{B}[\mathbf{U}, \mathbf{W}], \quad \forall \mathbf{U}, \mathbf{W} \in H_N^1(\Omega). \quad (5.16)$$

Then the existence and uniqueness of a discrete solution $\mathbf{U}_N \in H_N^1(\Omega)$ for any $\mathbf{p} \in H_N^{-1}$ is equivalent to the invertibility of the operator $\mathbf{B}_N : H_N^1(\Omega) \rightarrow H_N^1(\Omega)$. Since the domain and the range of linear operator \mathbf{B}_N have the same dimension N , we only need to prove the injectivity of \mathbf{B}_N . Using coercivity,

$$\begin{aligned} \forall 0 \neq \mathbf{U}_N \in H_N^1(\Omega), \text{ with } C > 0, \\ |\mathbf{B}_N \mathbf{U}_N|_{H_N^1(\Omega)} &= \sup_{\mathbf{V}_N \in H_N^1(\Omega)} \frac{\langle \mathbf{B}_N \mathbf{U}_N, \mathbf{V}_N \rangle_{H_N^1(\Omega)}}{|\mathbf{V}_N|_{H_N^1(\Omega)}} \\ &\geq \frac{\langle \mathbf{B}_N \mathbf{U}_N, \mathbf{U}_N \rangle_{H_N^1(\Omega)}}{|\mathbf{U}_N|_{H_N^1(\Omega)}} = \frac{\mathfrak{B}[\mathbf{U}_N, \mathbf{U}_N]}{|\mathbf{U}_N|_{H_N^1(\Omega)}} \geq C |\mathbf{U}_N|_{H_N^1(\Omega)} \geq 0. \end{aligned} \quad (5.17)$$

So \mathbf{B}_N is invertible and Theorem 3 holds, i.e. for any $\mathbf{p} \in H_N^{-1}(\Omega)$, there exists unique solution $\mathbf{U}_N \in H_N^1(\Omega)$ of (5.15), which further satisfies

$$|\mathbf{U}_N|_{H_N^1(\Omega)} \leq C^{-1} |\mathbf{p}|_{H_N^{-1}(\Omega)}. \quad (5.18)$$

■

Corollary 1. (Galerkin orthogonality).

$$\mathfrak{B}[\mathbf{u} - \mathbf{U}_N, \mathbf{W}] = 0, \quad \forall \mathbf{W} \in H_N^1(\Omega). \quad (5.19)$$

Proof. We are allowed to use any $\mathbf{W} \in H_N^1(\Omega)$ as a test function in the definition of the continuous weak form (5.4-5.5). This gives

$$\mathfrak{B}[\mathbf{u}, \mathbf{W}] = \langle \mathbf{p}, \mathbf{W} \rangle, \quad \forall \mathbf{W} \in H_N^1(\Omega). \quad (5.20)$$

Then recalling the definition of the Galerkin solution (5.15) and taking the difference yields (5.19). ■

Corollary 2. (Cea's Lemma). Galerkin solution \mathbf{U}_N in (5.15) is up to a constant as close to exact solution \mathbf{U}_N as the best approximation. The error satisfies

$$|\mathbf{u} - \mathbf{U}_N|_{H^1(\Omega)} \leq \frac{|\mathfrak{B}|}{C} \min_{\mathbf{v} \in H_N^1(\Omega)} |\mathbf{u} - \mathbf{V}|_{H^1(\Omega)}. \quad (5.21)$$

5.5 Finite element spaces

Practically, in this thesis, we choose Lagrange elements of polynomial degree $n = 1$ as our finite element bases. These are easy to construct and are locally supported, the latter property leads to sparse matrices. We use two types of adaptive mesh systems: *Kuhn triangulation* and *Quadtree adaptive mesh*. Both of these two adaptive meshes only involve image grayscale values at integer pixels, which help us void the interpolation bias error of image grayscale value located at subpixels. In 2D, Kuhn triangle simplices are all isosceles right triangles (cf Figure 5.1 first row), where all descendants of Kuhn triangles belong to one similarity class with various orientations (cf Figure 5.2), while elements in Quadtree mesh are all squares (cf Figure 5.1 second row). Kuhn triangulation doesn't require hanging nodes and can be easily refined with longest-edge bisection recursively. However, Quadtree mesh needs hanging nodes along transition edges between elements with different sizes. adaptive mesh ALDIC method.

First, we give out some useful definitions and concepts of Kuhn triangulation and Quadtree mesh and review some of their properties in this section. After that, we will discuss the refinement process of a given initial Kuhn triangulation using bisection and the refinement of a given initial Quadtree mesh.

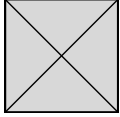
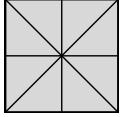
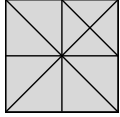
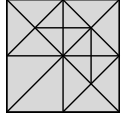
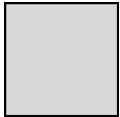
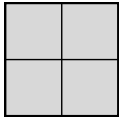
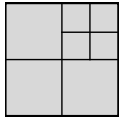
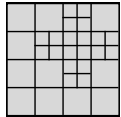
	Level 1	Level 2	Level 3	Level 4
Adaptive Kuhn triangle mesh				
Adaptive Quad tree mesh				

Figure 5.1: Comparison of adaptive Kuhn triangulation mesh and adaptive Quadtree mesh.

Kuhn simplex triangulation

We at first summarize the definition of *Kuhn simplex* (In 2D, it is also called *Kuhn triangle*).

Definition 2. (Simplex and Subsimplices). Let $d \in \mathbb{N}$ denotes the dimension. A subset T of \mathbb{R}^d is an n -simplex in \mathbb{R}^d if there exist $n + 1$ points $z_0, z_1, \dots, z_n \in \mathbb{R}^d$ such that

$$T = \text{conv hull}\{z_0, z_1, \dots, z_n\} = \left\{ \sum_{i=1}^n \lambda_i z_i \mid \lambda_i \geq 0 \text{ for } i = 0, \dots, d, \sum_{i=0}^n \lambda_i = 1 \right\}, \quad (5.22)$$

where $z_1 - z_0, \dots, z_n - z_0$ are linearly independent vectors in \mathbb{R}^d . By convention, points are defined as 0-simplices.

A subset T' of T is a (proper) k -subsimplex of T if T' is a k -simplex such that

$$T' = \text{conv hull}\{z'_0, \dots, z'_k\} \subset \partial T \quad (5.23)$$

with $k < n$ and $z'_0, \dots, z'_k \in \{z_0, \dots, z_d\}$. The 0-simplices are the *vertices* of a simplex. 1-simplices are *edges* and 2-simplices of 3-simplices are *faces*. Moreover, the $(n - 1)$ -simplices of n -simplices are referred as *sides*.

Definition 3. (Kuhn simplex). Kuhn simplex is a special type of d -simplex T with (ordered) vertices and its type t by

$$T = \{z_0, z_1, \dots, z_d\}_t, \quad t \in \{0, 1, \dots, d - 1\}, \quad (5.24)$$

where

$$z_0^\pi = 0, \quad z_i^\pi := \sum_{j=1}^i e_{\pi(j)} \quad \text{for all } i = 1, \dots, d, \quad (5.25)$$

where d is the dimension, π is a permutation of $\{1, \dots, d\}$.

There are three size measures of a simplex, which could further assist us to describe its shape regularity.

Definition 4. (Three size measures of a simplex). We define three quantities to measure the size of the given simplex T .

$$\text{Outball diameter: } \bar{h}_T := \sup\{|x - y| \mid x, y \in T\}, \quad (5.26)$$

$$\text{Inball diameter: } \underline{h}_T := \sup\{2r \mid B_r \subset T \text{ is a ball of radius } r\}, \quad (5.27)$$

$$\text{Element diameter: } h_T := |T|^{1/d}. \quad (5.28)$$

All the three quantities in Definition 4 satisfy the following inequality and they are equivalent up to the following quantity *Shape Coefficient*:

$$\underline{h}_T \leq h_T \leq \bar{h}_T \quad (5.29)$$

Definition 5. (Shape Coefficient). The shape coefficient of a d -simplex T in \mathbb{R}^d , σ_T , is defined as the ratio of the outball diameter and the inball diameter of T .

$$\sigma_T := \frac{\bar{h}_T}{\underline{h}_T}. \quad (5.30)$$

Generally, the shape coefficient is closely related with upper bounds for the element error of the solved Galerkin solutions, see Section 5.6. However, in the Kuhn triangulations, the shape of all the elements is always fixed with the same shape coefficient, which is one of the benefits for choosing Kuhn triangles as our elements (also see Theorem 4). For the 2D Kuhn triangle with triangle leg length h , $\bar{h}_T = \sqrt{2}h$, $\underline{h}_T = \left(1 - \frac{\sqrt{2}}{2}\right)h$, $h_T = \frac{\sqrt{2}}{2}h$, and $\sigma_T = 2(\sqrt{2} + 1)$.

Definition 6. (Generation). The generation $g(T)$ of a element T is the number of its ancestors in the refinement process, or, equivalently, the number of bisections needed to create T from the initial coarsest mesh element T_0 .

With the definition of the generation number of the element, some related properties of the elements can be obtained directly. For example, the element type of the

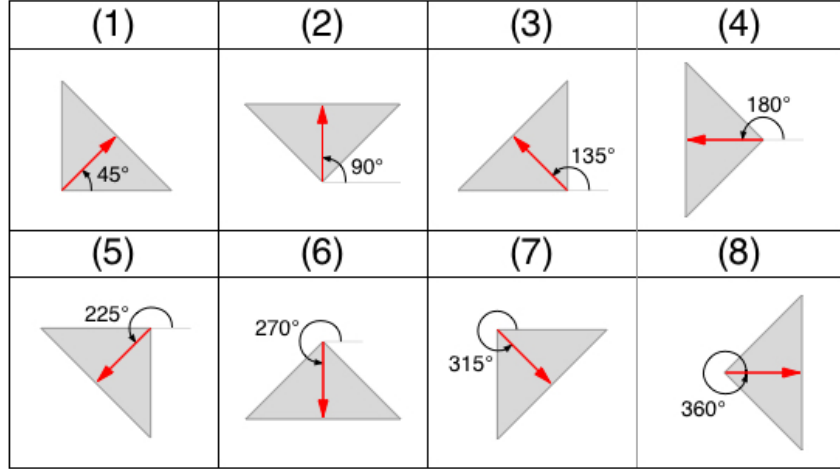


Figure 5.2: All the 2D Kuhn triangles belong to one similarity class with different orientations.

descendants of element T_0 is “ $(g(T) + t_0) \bmod d$ ”, where t_0 is the element type of T_0 , and descendant element size is

$$h_T = 2^{-g(T)/d} h_{T_0}, \quad (5.31)$$

where h_{T_0} is the size of coarsest mesh element T_0 .

Theorem 4. (Shape Regularity for a Kuhn Simplex). All 2^g descendants of generation g of a Kuhn simplex $T_\pi = \{z_0^\pi, \dots, z_d^\pi\}_0$ are mutually congruent with at most d different shapes. In the $d = 2$ case, all descendants of a Kuhn-triangle belong to one similarity class.

Definition 7. (Triangulation). Let $\Omega \subset \mathbb{R}^d$ be a bounded, polyhedral domain. A finite set \mathcal{J} of d -simplices in \mathbb{R}^d with

$$\bar{\Omega} = \bigcup_{T \in \mathcal{J}} T \quad \text{and} \quad |\Omega| = \sum_{T \in \mathcal{J}} |T| \quad (5.32)$$

in a triangulation of Ω . The *Shape coefficient* of a triangulation \mathcal{J} is the quantity $\sigma_{\mathcal{J}} := \max_{T \in \mathcal{J}} \sigma_T$. A triangulation \mathcal{J} is *conforming* if it satisfies the following property: if any two simplices $T_1, T_2 \in \mathcal{J}$ have a nonempty intersection $S = T_1 \cap T_2 \neq \emptyset$, then S is a k -subsimplex of both T_1 and T_2 with $k \in \{0, \dots, d\}$. In another words, conforming mesh means there are no vertices in the interior of edges. A sequence of triangulations $\{\mathcal{J}_k\}_{k \leq 0}$ is *shape regular* if $\sup_{T \in \mathcal{J}_k} \sigma_{\mathcal{J}} \leq C$. It is called *quasi-uniform* if there exists a finite constant C such that, for all k , there holds $\max_{T \in \mathcal{J}_k} \bar{h}_T \leq C \min_{T \in \mathcal{J}_k} \underline{h}_T$.

Finally, we give out the definition of uniform mesh and quasi-uniform mesh to distinguish the uniformly refined mesh used in multigrid method from the adaptive Kuhn triangulation.

Definition 8. (Uniform mesh). Let \mathcal{J}_N be a partition of Ω with N nodes. \mathcal{J}_N is *uniform mesh* if all the elements have exactly the same generation number.

Definition 9. (Quasi-uniform mesh). Let \mathcal{J}_N be a partition of Ω with N nodes. \mathcal{J}_N is *quasi-uniform mesh* if

$$\max_{T \in \mathcal{J}_N} \bar{h}_T^d \lesssim \min_{T \in \mathcal{J}_N} h_T^d, \quad (5.33)$$

where “ \lesssim ” means there can be a hidden constant independent of N .

FE basis functions of 2D Kuhn triangle elements

For the 2D Kuhn triangulation, we use constant strain triangle (CST) elements to solve equation system (5.4-5.5). Assuming the Cartesian coordinates of three nodes of 2D Kuhn triangle element are $\{(x_1, y_1), (x_2, y_2), (x_3, y_3)\}$, shape functions of the element T are

$$\begin{cases} N_1 = \frac{1}{2A} [(x_2 y_3 - x_3 y_2) + (y_2 - y_3)x + (x_3 - x_2)y] \\ N_2 = \frac{1}{2A} [(x_3 y_1 - x_1 y_3) + (y_3 - y_1)x + (x_1 - x_3)y], \\ N_3 = \frac{1}{2A} [(x_1 y_2 - x_2 y_1) + (y_1 - y_2)x + (x_2 - x_1)y] \end{cases}, \quad (5.34)$$

where A is the area of the triangle element.

$$A = \frac{1}{2} \det \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}. \quad (5.35)$$

There holds $\forall (x, y) \in T$,

$$x = \sum_{i=1}^3 N_i x_i; \quad y = \sum_{i=1}^3 N_i y_i \quad \text{and} \quad \mathbf{u} = \sum_{i=1}^3 N_i \mathbf{U}_i. \quad (5.36)$$

The derivatives of shape functions, DN , can be computed directly.

$$\begin{cases} \frac{\partial N_1}{\partial x} = \frac{1}{2A}(y_2 - y_3); & \frac{\partial N_1}{\partial y} = \frac{1}{2A}(x_3 - x_2); \\ \frac{\partial N_2}{\partial x} = \frac{1}{2A}(y_3 - y_1); & \frac{\partial N_2}{\partial y} = \frac{1}{2A}(x_1 - x_3); \\ \frac{\partial N_3}{\partial x} = \frac{1}{2A}(y_1 - y_2); & \frac{\partial N_3}{\partial y} = \frac{1}{2A}(x_2 - x_1); \end{cases} \quad (5.37)$$

$$DN = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 \\ \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} & 0 \\ 0 & \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} \end{bmatrix}. \quad (5.38)$$

Quadtree mesh

A 2D Quadtree (Figure 5.1 second row) is a tree mesh structure in which each parent element has exactly four children elements. They can be used to partition a two-dimensional space by recursively subdividing into four quadrants.

Definition 10. (*d*-cube). A *d*-cube Q , or also called *d* dimensional hypercube, is the convex hull of the points given by all sign permutations of the Cartesian coordinates $(\pm 1, \pm 1, \dots, \pm 1)$. A subset Q' of Q is also a hypercube with lower dimension. Specifically, any subset hypercube of the $(d - 1)$ dimension is called a *side* of the parent *d*-cube.

A hypercube of dimension d has $2d$ sides. A zero-dimensional hypercube is a *vertex* or *point*, and each *d*-cube has 2^d vertices. A one-dimensional cube is a *line*, which has 2 sides also called *endpoints*. A two-dimensional cube is a *square* and has 4 sides or *edges*. A three-dimensional cube is a square *box* with 6 two dimensional *faces*.

Definition 11. (Size measure of *d*-cube). The *d*-cube has volume 2^d . We define the element darameter h_Q as the length of the edge of *d*-cube.

Definition 12. (Generation). The generation $g(Q)$ of an element Q in a Quadtree mesh is the number of its ancestors in the refinement process, or equivalently the number of subdivisions needed to create Q from the inital coarsest mesh element Q_0 .

With the definition of the generation number of the element, the descendants of the generation g of a Quadtree mesh element all have the same shape with the element size:

$$h_Q = 2^{-g(Q)} h_{Q_0}, \quad (5.39)$$

where h_{Q_0} is the size of the initial element Q_0 .

Definition 13. (Shape regularity for d -cube). Each parent d -cube has 2^d or none children $(d - 1)$ -cube elements. All the descendants of generation number g of a d -cube belong to one similarity class.

Definition 14. (d -cube mesh). Let $\Omega \subset \mathbb{R}^d$ be a bounded, polyhedral domain. A finite set \mathcal{K} of d -cubes in \mathbb{R}^d with

$$\bar{\Omega} = \bigcup_{Q \in \mathcal{K}} Q \quad \text{and} \quad |\Omega| = \sum_{Q \in \mathcal{K}} |Q| \quad (5.40)$$

is a d -cube mesh of Ω . When $d = 2$, it is called Quadtree mesh, while when $d = 3$, it is also called Octree mesh. Any two elements Q_1, Q_2 have a nonempty intersection $S = Q_1 \cap Q_2 \neq \emptyset$ are called *neighboring elements*. A d -cube mesh \mathcal{K} is *conforming* if it satisfies the following property: $|g(Q_1) - g(Q_2)| \leq 1$, where $\{Q_1, Q_2\}$ are any two neighboring elements. Specially, a d -cube mesh \mathcal{K} is *uniformly conforming* if $|g(Q_1) - g(Q_2)| = 0$ where $\{Q_1, Q_2\}$ are any two neighboring elements. Elements that have at least one different generation number neighboring element are called *Transition elements*.

Theorem 5. Each d -cube mesh has at most $(2^{2d} - 1)$ kinds of transition elements.

Example 1. (Transition elements in 2D case). There are 15 in total kinds of transition elements, which are shown in Figure 5.3 (2)-(16).

FE shape functions of 2D Quadtree elements

For each 2D Quadtree non-transition element, we use bilinear Q4 isoparametric shape functions, which are defined as

$$\begin{cases} N_1 = \frac{1}{4}(1 - \xi)(1 - \eta); \\ N_2 = \frac{1}{4}(1 + \xi)(1 - \eta); \\ N_3 = \frac{1}{4}(1 + \xi)(1 + \eta); \\ N_4 = \frac{1}{4}(1 - \xi)(1 + \eta), \end{cases} \quad (5.41)$$

where $\{\xi, \eta\}$ is the natural coordinates attached to the element; see Figure 5.4. For any point (x, y) inside the element with four corner vertices $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\}$, there holds

$$x = \sum_{i=1}^4 N_i x_i; \quad y = \sum_{i=1}^4 N_i y_i \quad \text{and} \quad \mathbf{u} = \sum_{i=1}^4 N_i \mathbf{U}_i. \quad (5.42)$$

(1)	(2)	(3)	(4)	(5)	
(6)	(7)	(8)	(9)	(10)	(11)
(12)	(13)	(14)	(15)	(16)	

Figure 5.3: All kinds of elements in the Quadtree mesh, where nodes 1-4 are regular corner nodes from Q4 finite element and nodes 5-8 are hanging nodes to keep C^0 continuity of the deformation field.

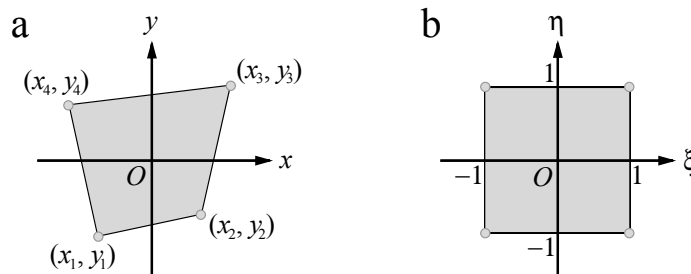


Figure 5.4: Isoparametric mapping in Q4 elements.

If none of the edges' midpoints exist, this element will be the above classic Q4 finite element, as seen in Figure 5.3 (1). However, if at least one edge's midpoints exist, this element will be one of the transition elements in Figure 5.3 (2)-(16). To implement the shape functions of transition elements, we choose Figure 5.3 (16) as the transition element template and extend classic Q4 shape functions using Gupta's method [75], where we define $\delta(\cdot)$ as

$$\delta(P_{ti}) = \begin{cases} 1, & \text{If midpoint of refinement edge, } P_{ti}, \text{ exists;} \\ 0, & \text{If midpoint of refinement edge, } P_{ti}, \text{ doesn't exist.} \end{cases} \quad (5.43)$$

Shape functions of Gupta's conforming transition elements are defined as:

$$\begin{cases} \tilde{N}_1 = N_1 - \frac{1}{2} \left(\tilde{N}_8 + \tilde{N}_5 \right); & \tilde{N}_5 = \frac{1}{2}(1 - \eta)(1 - |\xi|)\delta(\text{Pt5}); \\ \tilde{N}_2 = N_2 - \frac{1}{2} \left(\tilde{N}_5 + \tilde{N}_6 \right); & \tilde{N}_6 = \frac{1}{2}(1 + \xi)(1 - |\eta|)\delta(\text{Pt6}); \\ \tilde{N}_3 = N_3 - \frac{1}{2} \left(\tilde{N}_6 + \tilde{N}_7 \right); & \tilde{N}_7 = \frac{1}{2}(1 + \eta)(1 - |\xi|)\delta(\text{Pt7}); \\ \tilde{N}_4 = N_4 - \frac{1}{2} \left(\tilde{N}_7 + \tilde{N}_8 \right); & \tilde{N}_8 = \frac{1}{2}(1 - \xi)(1 - |\eta|)\delta(\text{Pt8}). \end{cases} \quad (5.44)$$

Displacement field is approximated using the linear combination of the above defined shape functions.

$$x = \sum_{i=1}^8 \tilde{N}_i x_i; \quad y = \sum_{i=1}^8 \tilde{N}_i y_i \quad \text{and} \quad \mathbf{u} = \sum_{i=1}^8 \tilde{N}_i \mathbf{U}_i. \quad (5.45)$$

The derivatives of Gupta's conforming transition elements shape functions can also be easily computed as follows:

$$\begin{cases} \frac{D\tilde{N}_1}{D\xi} = -\frac{1-\eta}{4} - \frac{1}{2} \left(\frac{D\tilde{N}_8}{D\xi} + \frac{D\tilde{N}_5}{D\xi} \right); & \frac{D\tilde{N}_1}{D\eta} = -\frac{1-\xi}{4} - \frac{1}{2} \left(\frac{D\tilde{N}_8}{D\eta} + \frac{D\tilde{N}_5}{D\eta} \right); \\ \frac{D\tilde{N}_2}{D\xi} = \frac{1-\eta}{4} - \frac{1}{2} \left(\frac{D\tilde{N}_5}{D\xi} + \frac{D\tilde{N}_6}{D\xi} \right); & \frac{D\tilde{N}_2}{D\eta} = -\frac{1+\xi}{4} - \frac{1}{2} \left(\frac{D\tilde{N}_5}{D\eta} + \frac{D\tilde{N}_6}{D\eta} \right); \\ \frac{D\tilde{N}_3}{D\xi} = \frac{1+\eta}{4} - \frac{1}{2} \left(\frac{D\tilde{N}_6}{D\xi} + \frac{D\tilde{N}_7}{D\xi} \right); & \frac{D\tilde{N}_3}{D\eta} = \frac{1+\xi}{4} - \frac{1}{2} \left(\frac{D\tilde{N}_6}{D\eta} + \frac{D\tilde{N}_7}{D\eta} \right); \\ \frac{D\tilde{N}_4}{D\xi} = -\frac{1+\eta}{4} - \frac{1}{2} \left(\frac{D\tilde{N}_7}{D\xi} + \frac{D\tilde{N}_8}{D\xi} \right); & \frac{D\tilde{N}_4}{D\eta} = \frac{1-\xi}{4} - \frac{1}{2} \left(\frac{D\tilde{N}_7}{D\eta} + \frac{D\tilde{N}_8}{D\eta} \right); \\ \frac{D\tilde{N}_5}{D\xi} = \frac{1}{2}(1 - \eta)\text{sign}(-\xi)\delta(\text{Pt5}); & \frac{D\tilde{N}_5}{D\eta} = -\frac{1}{2}(1 - |\xi|)\delta(\text{Pt5}); \\ \frac{D\tilde{N}_6}{D\xi} = \frac{1}{2}(1 - |\eta|)\delta(\text{Pt6}); & \frac{D\tilde{N}_6}{D\eta} = \frac{1}{2}(1 + \xi)\text{sign}(-\eta)\delta(\text{Pt6}); \\ \frac{D\tilde{N}_7}{D\xi} = \frac{1}{2}(1 + \eta)\text{sign}(-\xi)\delta(\text{Pt7}); & \frac{D\tilde{N}_7}{D\eta} = \frac{1}{2}(1 - |\xi|)\delta(\text{Pt7}); \\ \frac{D\tilde{N}_8}{D\xi} = -\frac{1}{2}(1 - |\eta|)\delta(\text{Pt8}); & \frac{D\tilde{N}_8}{D\eta} = \frac{1}{2}(1 - \xi)\text{sign}(-\eta)\delta(\text{Pt8}). \end{cases} \quad (5.46)$$

The Jacobian matrix of the isoparametric mapping is

$$J = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix}. \quad (5.47)$$

The DN matrix in the FEM is assembled to be

$$DN = \begin{bmatrix} \text{inv}(J) & 0 & 0 \\ 0 & 0 & \text{inv}(J) \end{bmatrix} \begin{bmatrix} \frac{D\tilde{N}_1}{D\xi} & 0 & \frac{D\tilde{N}_2}{D\xi} & 0 & \dots & \dots & \frac{D\tilde{N}_8}{D\xi} & 0 \\ \frac{D\tilde{N}_1}{D\eta} & 0 & \frac{D\tilde{N}_2}{D\eta} & 0 & \dots & \dots & \frac{D\tilde{N}_8}{D\eta} & 0 \\ 0 & \frac{D\tilde{N}_1}{D\xi} & 0 & \frac{D\tilde{N}_2}{D\xi} & \dots & \dots & 0 & \frac{D\tilde{N}_8}{D\xi} \\ 0 & \frac{D\tilde{N}_1}{D\eta} & 0 & \frac{D\tilde{N}_2}{D\eta} & \dots & \dots & 0 & \frac{D\tilde{N}_8}{D\eta} \end{bmatrix}. \quad (5.48)$$

The H^1 conformity of finite element spaces

Using our above defined conforming Kuhn triangulation \mathcal{J} or Quadtree mesh \mathcal{K} , we define finite-dimensional spaces for our equation system (5.4-5.5) and (5.10-5.11) as

$$\mathbb{V}(\mathcal{J}) := \{v \in C(\overline{\Omega}) \mid v|_T \in \mathbb{P}_1(T), \forall T \in \mathcal{J}\}, \quad (5.49)$$

$$\mathbb{V}(\mathcal{K}) := \{v \in C(\overline{\Omega}) \mid v|_Q \in \mathbb{P}_1(Q), \forall Q \in \mathcal{K}\}, \quad (5.50)$$

where $\mathbb{P}_1(\cdot)$ stands for the space of polynomials with degree not more than 1 over the inside domain. More precisely, we will show that $\mathbb{V}(\mathcal{J}) \subset H^1(\Omega)$ and $\mathbb{V}(\mathcal{K}) \subset H^1(\Omega)$, which possess bases which are locally supported and easy to implement.

Lemma 3. (H^1 -Conformity for Kuhn triangulation mesh). If \mathcal{J} is a conforming triangulation of a bounded polyhedral Lipschitz domain $\Omega \subset \mathbb{R}^d$, then $\mathbb{V}(\mathcal{J}) \subset H^1(\Omega)$.

Proof. Let $v \in \mathbb{V}(\mathcal{J})$. We show that v has a weak derivative. $\forall \eta \in C_0^\infty(\Omega)$ and $i \in \{1, 2\}$, there holds

$$\int_{\Omega} v \partial_i \eta = \sum_{T \in \mathcal{J}} \int_T v \partial_i \eta = \sum_{T \in \mathcal{J}} \int_T (\partial_i v) \eta + \sum_{T \in \mathcal{J}} \sum_{S \in \partial T} \int_S v \eta n_{T,i}, \quad (5.51)$$

where $n_{T,i}$ is the i -th coordinate of the exterior normal to ∂T . In the second term, if $S \subset \partial \Omega$, then there holds $\eta|_S = 0$; otherwise there exists a unique simplex $T' \in \mathcal{J}$ such that $S = T \cap T'$ and $n_{T',i} = -n_{T,i}$, so the second term vanishes. Consequently, there exists $w \in L^\infty(\Omega)$ is the weak derivative of v along the i -direction, where $w|_T = \partial_i v|_T$ for all $T \in \mathcal{J}$. So we have $\mathbb{V}(\mathcal{J}) \subset H^1(\Omega)$. \blacksquare

Analogously, H^1 -conformity also holds for Quadtree meshes.

Lemma 4. (H^1 -Conformity for Quadtree mesh). If \mathcal{K} is a conforming Quadtree mesh of a bounded polyhedral Lipschitz domain $\Omega \subset \mathbb{R}^d$, then $\mathbb{V}(\mathcal{K}) \subset H^1(\Omega)$.

5.6 A priori error estimate

Previously, Corollary 2 shows the error of Galerkin solutions of equation system (5.4-5.5) and (5.10-5.11) using a stable discretization is dictated by the best approximation from the discrete space. In this section, we provide a priori error estimate of the regularized Global DIC method and ALDIC Subproblem 2 method.

Theorem 6. (A Priori Error Estimate). Let $1 \leq s \leq 2, 1 \leq p \leq 2$ and let the exact solution $\mathbf{u}_{\text{Exact}}$ of the regularized Global DIC and ALDIC Subproblem 2 satisfy $\mathbf{u}_{\text{Exact}} \in W_p^s(\Omega)$ with $r := \text{sob}(W_p^s) - \text{sob}(H^1) > 0$. Let $\mathbf{U}_N \in \mathbb{V}(\mathcal{J}_N)$ with Kuhn triangulation mesh or $\mathbb{V}(\mathcal{K})$ with Quadtree mesh be the corresponding discrete solution. If $h : \Omega \rightarrow \mathbb{R}$ denotes the piecewise constant mesh density function that h equals h_T for all $T \in \mathcal{J}_N$, then

$$\|\nabla(\mathbf{u}_{\text{Exact}} - \mathbf{U}_N)\|_{L^2(\Omega)} \lesssim \frac{|\mathfrak{B}|}{C} \|h^r D^s \mathbf{u}_{\text{Exact}}\|_{L^p(\Omega)}, \quad (5.52)$$

where “ \lesssim ” means there can be a hidden constant, which depends on shape coefficient of the mesh and the dimension d . Sobolev number of $W_p^k(\Omega)$ is defined by $\text{sob}(W_p^k) := k - d/p$.

Proof. We refer to [74] Section §5.2 for the proof. ■

Corollary 3. (Quasi-Uniform Meshes). Let $1 \leq s \leq 2$, and let the exact solution $\mathbf{u}_{\text{Exact}}$ of the regularized Global DIC and ALDIC satisfy $\mathbf{u}_{\text{Exact}} \in H^s(\Omega)$. Let \mathcal{J}_N be a quasi-uniform partition of Ω with N nodes and let $\mathbf{U}_N \in \mathbb{V}(\mathcal{J}_N)$ with Kuhn triangulation mesh or $\mathbb{V}(\mathcal{K}_N)$ with Quad/Octree mesh be the corresponding discrete solution. Then,

$$\|\nabla(\mathbf{u}_{\text{Exact}} - \mathbf{U}_N)\|_{L^2(\Omega)} \lesssim \frac{|\mathfrak{B}|}{C} |\mathbf{u}_{\text{Exact}}|_{H^s(\Omega)} N^{-(s-1)/d}. \quad (5.53)$$

Proof. Quasi-uniformity of mesh \mathcal{J}_N implies

$$\max_{T \in \mathcal{J}_N} h_T^d \leq \max_{T \in \mathcal{J}_N} \bar{h}_T^d \lesssim \min_{T \in \mathcal{J}_N} \underline{h}_T^d \leq \frac{1}{N} \sum_{T \in \mathcal{J}_N} h_T^d = \frac{|\Omega|}{N}. \quad (5.54)$$

Since $r = (s - d/2) - (1 - d/2) = s - 1$, plug above upper bound for h_T into (5.52), we can prove (5.53) holds. Since uniform mesh can be viewed as one special case of quasi-uniform mesh, (5.53) also holds for uniform mesh. ■

Example 2. Using quasi-uniform meshes, if $1 \leq s \leq 2$, the decay rate of error $|\mathbf{u}_{\text{Exact}} - \mathbf{U}_N|_{H^1(\Omega)}$ is bounded by $O(N^{-(s-1)/d})$, slower than $O(N^{-1/d})$.

Using uniform meshes, h_T is constant over the whole domain, where H^1 norm of the error of discrete Galerkin solution is upper bounded by the global behaviors of $|\mathbf{u}_{\text{Exact}}|_{H^s(\Omega)}$ and the value of h_T . However, this upper bound can be further decreased to achieve higher decay rate by adaptively choosing element size h_T due to different local $|\mathbf{u}_{\text{Exact}}|_{H^s}$ property.

For example, in the two-dimensional $d = 2$ case, Binev *et al.*[76] proved the following theorem for h-adaptive mesh.

Theorem 7. Let $\mathbf{u}_{\text{Exact}} \in H^1(\Omega)$ and \mathbf{u} has the regularity that $\mathbf{u}_{\text{Exact}} \in C^2(\Omega) \cap W_p^2(\Omega)$ with $1 < p \leq 2$. The number of nodes of initial uniform mesh \mathcal{J}_0 is N_0 . Then there exists refined mesh \mathcal{J}_N with N nodes such that

$$|\mathbf{u}_{\text{Exact}} - \mathbf{U}_N|_{H^1(\Omega)} \lesssim |\Omega|^{1-1/p} \|D^2 \mathbf{u}_{\text{Exact}}\|_{L^p(\Omega)} N^{-1/2}. \quad (5.55)$$

More generally, in $d \geq 2$ dimensional, using element shape function with polynomials degree $n \geq 1$, the maximum decay rate could reach $O(N^{-n/d})$ using suitable adaptive mesh.

In order to suitably refine the mesh adaptively, we will introduce the *principle of error equidistribution* that the optimal mesh refinement should approximately satisfy.

Principle of error equidistribution

Considering the following optimization problem: given a function $\mathbf{u} \in H^1(\Omega)$ and an integer $N > 0$, we want to find a shape regular mesh \mathcal{J}_N to minimize the error $|\mathbf{u}_{\text{Exact}} - \mathbf{U}_N|_{H^1(\Omega)}$ subject to the constraint that the number of degrees of freedom does not exceed N . Especially, here we only discuss $d = 2$ case, and assume the regularity of $\mathbf{u}_{\text{Exact}} \in C^2(\Omega) \cap W_p^2(\Omega)$ with $1 < p \leq 2$. The number of degrees of freedom N is related with the mesh density function h by

$$N = \int_{\Omega} \frac{1}{h} d\Omega. \quad (5.56)$$

Applying Theorem 6 (5.52), the error is bounded from above by

$$\|\nabla(\mathbf{u}_{\text{Exact}} - \mathbf{U}_N)\|_{L^2(\Omega)}^2 \lesssim \left(\int_{\Omega} h^{2(p-1)} |D^2 \mathbf{u}_{\text{Exact}}|^p d\Omega \right)^{\frac{2}{p}}. \quad (5.57)$$

To minimize the above upper bound with a given number of degrees of freedom N , we propose the Lagrangian

$$\mathfrak{L}[h, \lambda] = \int_{\Omega} \left(h^{2(p-1)} |D^2 \mathbf{u}_{\text{Exact}}|^p - \frac{\lambda}{h^2} \right) d\Omega. \quad (5.58)$$

A stationary point of \mathfrak{L} satisfies

$$h^{2(p-1)+2} |D^2 \mathbf{u}_{\text{Exact}}|^p = \text{constant}. \quad (5.59)$$

If we further assume the mesh \mathcal{J} is sufficiently fine so that $D^2\mathbf{u}_{\text{Exact}}$ is essentially constant within each element $T \in \mathcal{J}$, the above stationary condition is equivalent to the equidistribution of error E_T , where

$$\begin{cases} E_T = h_T^{2(p-1)} \int_T |D^2\mathbf{u}_{\text{Exact}}|^p \approx h_T^{2(p-1)+2} |D^2\mathbf{u}_{\text{Exact}}(\mathbf{x}_T)|^p \\ E_T \approx \text{constant} \quad \text{for all } T \in \mathcal{J}. \end{cases} \quad (5.60)$$

The above principle of error equidistribution gives us the idea to refine the elements with larger errors, and Theorem 7 implies there exists certain refined mesh to reach $O(N^{-1/2})$ decay rate in the error. In practice, the exact solution $\mathbf{u}_{\text{Exact}}$ is unknown. We are only able to replace the element error by computable element error indicator, which will be discussed in Section 5.7. We also comment that in practice, the above error's maximum decay rate $O(N^{-1/2})$ is usually hard to reach, since it's very hard to find the optimal mesh to make the error E_T exactly equidistributed.

5.7 A posteriori error estimate

In this section, a posteriori error estimators of adaptive regularized Global DIC and adaptive ALDIC are proposed, providing the indicators of elements with large errors to be adaptively refined.

A posteriori error estimate of adaptive regularized Global DIC method

In regularized Global DIC, we are actually solving the above linearized weak form (5.3), which holds in the integral meaning while strong form doesn't satisfy pointwisely. This inspired us to introduce the pointwise residual of Global DIC method at the $(n+1)$ th step:

$$\mathfrak{R}_G(\mathbf{U}_{N_{n+1}}) := \alpha \Delta \mathbf{U}_{N_{n+1}} + \left[f(\mathbf{X}) - g(\mathbf{X} + \mathbf{U}_{N_n}) - \frac{\partial f(\mathbf{X})}{\partial \mathbf{X}} \cdot (\mathbf{U}_{N_{n+1}} - \mathbf{U}_{N_n}) \right] \frac{\partial f(\mathbf{X})}{\partial \mathbf{X}}. \quad (5.61)$$

Let us denote $\tilde{\mathbf{f}}_G := \left[f(\mathbf{X}) - g(\mathbf{X} + \mathbf{U}_{N_n}) - \frac{\partial f(\mathbf{X})}{\partial \mathbf{X}} \cdot (\mathbf{U}_{N_{n+1}} - \mathbf{U}_{N_n}) \right] \frac{\partial f(\mathbf{X})}{\partial \mathbf{X}}$ and suppose $\tilde{\mathbf{f}}_G \in L^2(\Omega)$. This allows us to write $\langle \mathfrak{R}, \mathbf{w} \rangle$ as integrals over each $T \in \mathcal{J}$ and integrate by parts:

$$\langle \mathfrak{R}_G, \mathbf{w} \rangle = \int_{\Omega} \mathfrak{R}_G \cdot \mathbf{w} d\mathbf{X} = \sum_{T \in \mathcal{J}} \int_T \mathbf{r}_G \cdot \mathbf{w} d\mathbf{X} + \sum_{\Gamma \in \mathcal{J}} \int_{\Gamma} \mathbf{j}_G \cdot \mathbf{w} ds, \quad \mathbf{w} \in H^1(\Omega) \quad (5.62)$$

$$\text{with } \begin{cases} \mathbf{r}_G = \mathfrak{R}_G(\mathbf{U}_{N_{n+1}}), & \text{Interior residual in any simplex } T \in \mathcal{J}, \\ \mathbf{j}_G = \llbracket \alpha \nabla(\mathbf{U}_{N_{n+1}}) \rrbracket, & \text{Jump residual on any element's internal side } \Gamma \in \mathcal{J}, \end{cases} \quad (5.63)$$

where $\llbracket \alpha \nabla(\mathbf{U}_{N_{n+1}}) \rrbracket = \alpha \mathbf{n}^+ \cdot \nabla(\mathbf{U}_{N_{n+1}})|_{T^+} + \alpha \mathbf{n}^- \cdot \nabla(\mathbf{U}_{N_{n+1}})|_{T^-}$ and $\mathbf{n}^+, \mathbf{n}^-$ are unit normals pointing towards $T^+, T^- \in \mathcal{J}$, respectively.

Element local error indicator is defined as

$$\mathcal{E}_G^2(\mathbf{U}_{N_{n+1}}, T) = h_T^2 \|\mathbf{r}_G\|_{L^2(T)}^2 + h_T \|\mathbf{j}_G\|_{L^2(\partial T \setminus \partial \Omega)}^2 \quad (5.64)$$

and total error estimator is summed to be

$$\mathcal{E}_G^2(\mathbf{U}_{N_{n+1}}, \mathcal{J}) = \sum_{T \in \mathcal{J}} \mathcal{E}_G^2(\mathbf{U}_{N_{n+1}}, T). \quad (5.65)$$

Error of the solution of (5.3) is bounded by

$$\frac{1}{C_1} (\mathcal{E}(\mathbf{U}_{N_{n+1}}, \mathcal{J}) - \text{osc}(\mathbf{U}_{N_{n+1}}, \mathcal{J})) \lesssim \|\nabla(\mathbf{U}_{N_{n+1}} - \mathbf{u}_{\text{Exact}})\|_{L^2(\Omega)} \lesssim \frac{\mathcal{E}_G(\mathbf{U}_{N_{n+1}}, \mathcal{J})}{C_1}, \quad (5.66)$$

where $\mathbf{u}_{\text{Exact}}$ is the exact solution, C_1 is the same value in Lemma 1. Term $\text{osc}(\mathbf{U}_{N_{n+1}}, \mathcal{J})$ measures oscillations of both interior residual \mathbf{r}_G and jump residual \mathbf{j}_G .

$$\text{osc}(\mathbf{U}_{N_{n+1}}, \mathcal{J}) = \|h(\mathbf{r}_G - \bar{\mathbf{r}}_G)\|_{L^2(\Omega)} + \|h^{1/2}(\mathbf{j}_G - \bar{\mathbf{j}}_G)\|_{L^2(\Gamma)} \quad (5.67)$$

$$\bar{\mathbf{r}}_G|_T = \frac{\int_T \mathbf{r}_G}{|T|} \quad \text{for all } T \in \mathcal{J} \quad \text{and} \quad \bar{\mathbf{j}}_G|_S = \frac{\int_S \mathbf{j}_G}{|S|} \quad \text{for all } S \in \Gamma. \quad (5.68)$$

A posteriori error estimate of adaptive ALDIC method

We now estimate a posteriori error of adaptive ALDIC method. ALDIC has two types of a posteriori error estimates based on two ADMM subproblems respectively. From the first subproblem local step, a posteriori error estimate at the $(n+1)$ step is the same with the Local Subset DIC method SSD(summation of squared differences) correlation function:

$$\mathcal{E}_{AL1}^2(\mathbf{U}_{N_{n+1}}, \mathbf{F}_{N_{n+1}}, \mathcal{J}) = \sum_i \int_{\Omega} \left(|f(\mathbf{X}) - g(\mathbf{X} + \mathbf{U}_{N_{n+1}} + \mathbf{F}_{N_{n+1}}(\mathbf{X} - \mathbf{X}_{i0}))|^2 \right) d\mathbf{X}. \quad (5.69)$$

From the second subproblem global step, the pointwise residual at the $(n + 1)$ step is defined as

$$\mathfrak{R}_{AL}(\hat{\mathbf{U}}_{N_{n+1}}) = \beta \left(\Delta \hat{\mathbf{U}}_{N_{n+1}} - \operatorname{div}(\mathbf{F}_{n+1} - \mathbf{W}_n) \right) - \mu \left(\hat{\mathbf{U}}_{N_{n+1}} - \mathbf{U}_{N_{n+1}} + \mathbf{v}_n \right). \quad (5.70)$$

We can further denote $\tilde{\mathbf{f}}_{AL} := -\beta \operatorname{div}(\mathbf{F}_{n+1} - \mathbf{W}_n) - \mu \left(\hat{\mathbf{U}}_{N_{n+1}} - \mathbf{U}_{N_{n+1}} + \mathbf{v}_n \right)$ and suppose $\tilde{\mathbf{f}}_{AL} \in L^2(\Omega)$. The local error indicator of each element is computed as

$$\mathcal{E}_{AL2}^2(\hat{\mathbf{U}}_{N_{n+1}}, T) = h_T^2 \|\mathbf{r}_{AL}\|_{L^2(T)}^2 + h_T \|\mathbf{j}_{AL}\|_{L^2(\partial T \setminus \partial \Omega)}^2 \quad (5.71)$$

with

$$\begin{cases} \mathbf{r}_{AL} = \mathfrak{R}_{AL}(\hat{\mathbf{U}}_{N_{n+1}}), & \text{Interior residual in any simplex } T \in \mathcal{J}, \\ \mathbf{j}_{AL} = \llbracket \beta \nabla(\hat{\mathbf{U}}_{N_{n+1}}) \rrbracket, & \text{Jump residual on any element's internal side } \Gamma \in \mathcal{J}, \end{cases} \quad (5.72)$$

where $\llbracket \beta \nabla \hat{\mathbf{U}}_{N_{n+1}} \rrbracket = \beta \mathbf{n}^+ \cdot \nabla \hat{\mathbf{U}}_{N_{n+1}}|_{T^+} + \beta \mathbf{n}^- \cdot \nabla \hat{\mathbf{U}}_{N_{n+1}}|_{T^-}$ and $\mathbf{n}^+, \mathbf{n}^-$ are unit normals pointing towards $T^+, T^- \in \mathcal{J}$, respectively.

The total error estimator of ALDIC Subproblem 2 is summed to be

$$\mathcal{E}_{AL2}^2(\hat{\mathbf{U}}_{N_{n+1}}, \mathcal{J}) = \sum_{T \in \mathcal{J}} \mathcal{E}_{AL2}^2(\hat{\mathbf{U}}_{N_{n+1}}, T). \quad (5.73)$$

Error of the solution of (5.10-5.11) is bounded by

$$\begin{aligned} \frac{1}{C_3} \left(\mathcal{E}_{AL2}(\hat{\mathbf{U}}_{N_{n+1}}, \mathcal{J}) - \operatorname{osc}(\hat{\mathbf{U}}_{N_{n+1}}, \mathcal{J}) \right) &\lesssim \left\| \nabla(\hat{\mathbf{U}}_{N_{n+1}} - \mathbf{u}_{\text{Exact-Sub2}}) \right\|_{L^2(\Omega)} \\ &\lesssim \frac{\mathcal{E}_{AL2}(\hat{\mathbf{U}}_{N_{n+1}}, \mathcal{J})}{C_3}, \end{aligned} \quad (5.74)$$

where $\mathbf{u}_{\text{Exact-Sub2}}$ is the exact solution of ADMM Subproblem 2, C_3 is the same value in Lemma 2.

$$\operatorname{osc}(\hat{\mathbf{U}}_{N_{n+1}}, \mathcal{J}) = \|h(\mathbf{r}_{AL} - \bar{\mathbf{r}}_{AL})\|_{L^2(\Omega)} + \|h^{1/2}(\mathbf{j}_{AL} - \bar{\mathbf{j}}_{AL})\|_{L^2(\Gamma)} \quad (5.75)$$

$$\bar{\mathbf{r}}_{AL}|_T = \frac{\int_T \mathbf{r}_{AL}}{|T|} \quad \text{for all } T \in \mathcal{J} \quad \text{and} \quad \bar{\mathbf{j}}_{AL}|_S = \frac{\int_S \mathbf{j}_{AL}}{|S|} \quad \text{for all } S \in \Gamma \quad (5.76)$$

We can define the overall a posteriori error estimate of adaptive ALDIC method as

$$\mathcal{E}_{AL}^2(\mathbf{U}_{N_{n+1}}, \mathbf{F}_{N_{n+1}}, \hat{\mathbf{U}}_{N_{n+1}}, \mathcal{J}) = \mathcal{E}_{AL1}^2(\mathbf{U}_{N_{n+1}}, \mathbf{F}_{N_{n+1}}, \mathcal{J}) + \mathcal{E}_{AL2}^2(\hat{\mathbf{U}}_{N_{n+1}}, \mathcal{J}). \quad (5.77)$$

Lemma 5. $\mathcal{E}_{AL}(\mathbf{U}_{N_{n+1}}, \mathbf{F}_{N_{n+1}}, \hat{\mathbf{U}}_{N_{n+1}}, \mathcal{J}) \rightarrow 0$ if and only if $\mathcal{E}_{AL1}(\mathbf{U}_{N_{n+1}}, \mathbf{F}_{N_{n+1}}, \mathcal{J}) \rightarrow 0$ and $\mathcal{E}_{AL2}(\hat{\mathbf{U}}_{N_{n+1}}, \mathcal{J}) \rightarrow 0$.

Proof. Since all the three terms are non-negatives, the above Lemma holds automatically. \blacksquare

Theorem 8. Let $\mathbf{u}_{\text{Exact}}$ denotes exact solution of the overall ALDIC problem. If $\mathcal{E}_{AL} \rightarrow 0$, then we have $\hat{\mathbf{U}}_N \rightarrow \mathbf{u}_{\text{Exact}}$.

Proof. First, use the above Lemma 5, $\mathcal{E}_{AL} \rightarrow 0$ implies both $\mathcal{E}_{AL1} \rightarrow 0$ and $\mathcal{E}_{AL2} \rightarrow 0$. Then we have the inequality:

$$\begin{aligned} \left\| \hat{\mathbf{U}}_N - \mathbf{u}_{\text{Exact}} \right\|_{H^1} &\leq \left\| \hat{\mathbf{U}}_N - \mathbf{u}_{\text{Exact-Sub2}} \right\|_{L^2} + \left\| \nabla \hat{\mathbf{U}}_N - \nabla \mathbf{u}_{\text{Exact-Sub2}} \right\|_{L^2} \\ &\quad + \left\| \mathbf{u}_{\text{Exact-Sub2}} - \mathbf{U}_N \right\|_{L^2} + \left\| \nabla \mathbf{u}_{\text{Exact-Sub2}} - \mathbf{F}_N \right\|_{L^2} \\ &\quad + \left\| \mathbf{U}_N - \mathbf{u}_{\text{Exact}} \right\|_{L^2} + \left\| \mathbf{F}_N - \nabla \mathbf{u}_{\text{Exact}} \right\|_{L^2}. \end{aligned} \quad (5.78)$$

On the right hand side, the first two terms are bounded by \mathcal{E}_{AL2} , the last two terms are bounded by \mathcal{E}_{AL1} , and the middle two terms also converge to zeros if $\mathcal{E}_{AL1} \rightarrow 0$. \blacksquare

5.8 Mesh refinement

Inspired by the principle of error equidistribution, we mark and refine elements whose with large a posteriori error estimates, while keep the other elements with small a posteriori error estimates unchanged.

Mark elements with large a posteriori error estimate

We list three popular marking strategies to collect elements with largest a posteriori error estimates into set \mathcal{M} such that $\min_{T \in \mathcal{M}} \mathcal{E}_{\mathcal{J}}(\mathbf{U}_{\mathcal{J}}, T) \geq \max_{T \in \mathcal{J} \setminus \mathcal{M}} \mathcal{E}_{\mathcal{J}}(\mathbf{U}_{\mathcal{J}}, T)$.

(1) **Maximum Strategy:** For given parameter $\theta \in [0, 1]$, we let

$$\mathcal{M} = \{T \in \mathcal{J} \mid \mathcal{E}_{\mathcal{J}}(\mathbf{U}_{\mathcal{J}}, T) \geq \theta \mathcal{E}_{\mathcal{J}, \max}\} \quad \text{with} \quad \mathcal{E}_{\mathcal{J}, \max} = \max_{T \in \mathcal{J}} \mathcal{E}_{\mathcal{J}}(\mathbf{U}_{\mathcal{J}}, T). \quad (5.79)$$

(2) **Equidistribution Strategy:** For given parameter $\theta \in [0, 1]$, we let

$$\mathcal{M} = \{T \in \mathcal{J} \mid \mathcal{E}_{\mathcal{J}}(\mathbf{U}_{\mathcal{J}}, T) \geq \theta \mathcal{E}_{\mathcal{J}}(\mathbf{U}_{\mathcal{J}}, \mathcal{J}) / \sqrt{N}\}. \quad (5.80)$$

(3) **Dörfler's Strategy:** For given parameter $\theta \in [0, 1]$, we let $\mathcal{M} \subset \mathcal{J}$ such that

$$\mathcal{E}_{\mathcal{J}}(\mathbf{U}_{\mathcal{J}}, \mathcal{M}) \geq \theta \mathcal{E}_{\mathcal{J}}(\mathbf{U}_{\mathcal{J}}, \mathcal{J}). \quad (5.81)$$

Next, we come to refine marked elements in set \mathcal{M} . The refinement algorithm using Kuhn triangulation and Quadtree mesh are described below.

2D Kuhn triangle simplex bisection

As for Kuhn simplex, the edge $\overline{z_0^\pi, z_d^\pi}$ of any Kuhn-simplex is always the longest edge which connects the first and last vertex of the simplex, and it is always the refinement edge whose midpoint $\bar{z} = (z_0^\pi + z_d^\pi)/2$ will become the new vertex. Connecting the new vertex \bar{z} with the vertices of T other than z_0, z_d will generate the new edge shared by the two children simplex T_1, T_2 of parent simplex T , whose vertex order is dictated by the bisection rule

$$\begin{aligned} T_1 &:= \{z_0, \bar{z}, \underbrace{z_1, \dots, z_t}_{\rightarrow}, \underbrace{z_{t+1}, \dots, z_{d-1}}_{\rightarrow}\}_{(t+1) \bmod d}, \\ T_2 &:= \{z_d, \bar{z}, \underbrace{z_1, \dots, z_t}_{\rightarrow}, \underbrace{z_{d-1}, \dots, z_{t+1}}_{\leftarrow}\}_{(t+1) \bmod d} \end{aligned} \quad (5.82)$$

with the convention that arrows point in the direction of increasing indices and $\{z_1, \dots, z_0\} = \emptyset, \{z_d, \dots, z_{d-1}\} = \emptyset$.

In $d = 2$ case, the bisection rule does not depend on the element type and for $T = \{z_0, z_1, z_2\}$ the two children are

$$T_1 = \{z_0, \bar{z}, z_1\}, \quad T_2 = \{z_2, \bar{z}, z_1\}. \quad (5.83)$$

Algorithm 4: Bisect compatibly divisible element T into two children elements

function BISECT(T)

Step 1, Find the longest edge E and the midpoint of E .

Step 2, Find the vertex opposite to the longest edge.

Step 3, Connect the vertex found in Step 2 with the midpoint of E found in Step 1, which bisects element E into two children elements $\{T_1, T_2\}$.

Return $\{T_1, T_2\}$.

To assist recursively bisect Kuhn triangle elements, we summarize some useful definitions as follows [74].

Definition 15. (Binary tree, Forest and Refinement). A *binary tree* is a tree data structure in which each node has at most two children. Given initial simplex T_0 , the binary tree $\mathbb{F}(T_0)$ includes all the refined descendant elements that will be generated by recurrent bisection of T_0 . Let \mathcal{J}_0 be an initial conforming triangulation. Then

$$\mathbb{F} = \mathbb{F}(\mathcal{J}_0) := \bigcup_{T_0 \in \mathcal{J}_0} \mathbb{F}(T_0) \quad (5.84)$$

is the associated *master forest* of binary trees. For a node $T \in \mathbb{F}$ so that $T \in \mathbb{F}(T_0)$ with $T_0 \in \mathcal{J}_0$, the generation $g(T)$ is the generation of T within $\mathbb{F}(T_0)$. A subset $\mathcal{F} \subset \mathbb{F}$ is called *forest* iff

- (1) $\mathcal{J}_0 \subset \mathcal{F}$;
- (2) All nodes of $\mathcal{F} \setminus \mathcal{J}_0$ have a predecessor;
- (3) All nodes of \mathcal{F} have either two successors or none.

A forest \mathcal{F} is called *finite*, if $\max_{T \in \mathcal{F}} g(T) < \infty$. The nodes with no successors are called *leaves* of \mathcal{F} . Any finite forest \mathcal{F} is uniquely associated with a triangulation $\mathcal{J} = \mathcal{J}(\mathcal{F})$ of Ω by defining the nodes of \mathcal{J} to be the set of all leaves in \mathcal{F} . Given two finite forests $\mathcal{F}_1, \mathcal{F}_2 \in \mathbb{F}$ with associated triangulations $\mathcal{J}_1, \mathcal{J}_2$ we call \mathcal{J}_2 as refinement of \mathcal{J}_1 iff $\mathcal{F}_1 \subset \mathcal{F}_2$ and we denote this by $\mathcal{J}_1 \leq \mathcal{J}_2$.

Definition 16. (Neighboring Elements and Conforming Refinements). Two elements $T_1, T_2 \in \mathcal{J}$ are called *neighboring elements* if they share a common side, which is a $(d - 1)$ -subsimplex. The class of *Conforming refinements* of initial triangulation \mathcal{J}_0 is defined as

$$\mathbb{T} = \{ \mathcal{J} = \mathcal{J}(\mathcal{F}) \mid \mathcal{F} \subset \mathbb{F} \text{ is finite and } \mathcal{J}(\mathcal{F}) \text{ is conforming} \}. \quad (5.85)$$

In $d = 2$ case, the new vertices are always midpoints of edges. Any two neighboring elements will be bisected at the same edge with the same new midpoint vertex, and the conformity of the refined mesh can be guaranteed automatically.

Definition 17. (Compatible Bisection). Two elements $T, T' \in \mathbb{F}$ are *compatibly divisible* if they have the same refinement edge. If all elements share an edge that compatibly divisible, then they form a *bisection patch*.

Definition 18. (Refinement Patch). Let \mathcal{J} be a given conforming grid and let $T \in \mathcal{J}$ be an element with refinement edge E . We define the *refinement patch* of T to be

$$R(\mathcal{J}; T) := \{ T' \in \mathcal{J} \mid T' \in \mathcal{J} \text{ with } E \subset T' \}. \quad (5.86)$$

Next, we give out the theorem about divisible compatibility in the Kuhn triangle refinement patch.

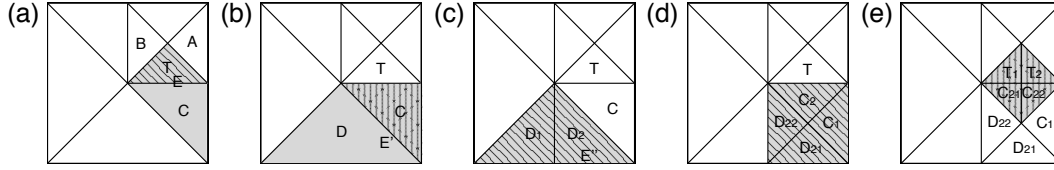


Figure 5.5: Example of recursive refinement of 2D Kuhn triangulation.

Theorem 9. Let \mathcal{J}_0 be a uniform conforming triangulation formed by Kuhn simplices of type 0, and let \mathcal{J} be a conforming refinement. Then any $T \in \mathcal{J}$ is of locally highest generation in $R(\mathcal{J}; T)$, i.e.,

$$g(T) = \max\{g(T') \mid T' \in R(\mathcal{J}; T)\} \quad (5.87)$$

and $T' \in R(\mathcal{J}; T)$ is *compatibly divisible* with T is and only if $g(T') = g(T)$.

Futhermore, $\min\{g(T') \mid T' \in R(\mathcal{J}; T)\} \leq g(T) - d + 1$ and a non-compatibly divisible neighboring element of T has generation $g(T) - 1$.

Definition 19. (Recursive Refinement). If all elements in refinement patch of T , $R(\mathcal{J}, T)$, are compatibly divisible, we can simultaneously refine all the elements. Otherwise, if there are elements in $R(\mathcal{J}, T)$ that are not compatibly divisible with T , the basic idea is to recursively refine these elements first.

In 2D case, if the longest edge of element T , E , is interior, there is only one neighbor element sharing the refinement edge E . Either this neighbor element is compatibly divisible, or the neighboring child is compatibly divisible after bisection of the neighbor. If refinement edge E lies on the boundary, bisection can be executed directly. We summarize the recursive refinement of a single element T in Algorithm 5.

Example 3. (Recursive refinement of a single Kuhn triangle element). As shown in Figure 5.5 (a), T is the marked element to refine in the current conforming Kuhn triangulation \mathcal{J} . The refinement patch of T is $R(\mathcal{J}; T) = \{A, B, C, T\}$, with generation number $g(A) = g(B) = g(T) = 2 > g(C) = 1$. So we next access element C since $g(C) = \min\{g(T'') \mid T'' \in R(\mathcal{J}, T)\} < g(T)$ and we need to compute $\text{REFINE_RECURSIVE}(\mathcal{J}, C)$ first. $R(\mathcal{J}; C) = \{D, C, T\}$ where element D reaches the minimum generation number and we need to compute $\text{REFINE_RECURSIVE}(\mathcal{J}, D)$ (see Figure 5.5 (b)). $R(\mathcal{J}; D) = \{D, C\}$ and

Algorithm 5: Recursive refinement of a single element $T \in \mathcal{J}$

function REFINE_RECURSIVE(\mathcal{J}, T)

Do forever

 Get refinement patch $R(\mathcal{J}, T)$;

 Access $T' \in R(\mathcal{J}, T)$ with $g(T') = \min\{g(T'') \mid T'' \in R(\mathcal{J}, T)\}$;

 If $g(T') < g(T)$ then

$\mathcal{J} := \text{REFINE_RECURSIVE}(\mathcal{J}, T')$;

 Else

 break;

 End if

end do

Get refinement patch $R(\mathcal{J}, T)$;

For all $T' \in R(\mathcal{J}, T)$, do

 If $g(T') = g(T)$ then

$\{T'_0, T'_1\} = \text{BISECT}(T')$;

$\mathcal{J} := \mathcal{J} \setminus \{T'\} \cup \{T'_0, T'_1\}$;

 End if

End for

Return \mathcal{J} .

$g(D) = \min\{g(T'') \mid T'' \in R(\mathcal{J}, D)\}$, so we refine refinement patch $R(\mathcal{J}, D)$. Since $g(C) < g(D)$, applying Theorem 9, element C is not compatibly divisible with element D . Thus only element D is bisected next through Algorithm 4. The Kuhn triangulation mesh in Figure 5.5 (c) is generated as

$$\mathcal{J}_1 := \mathcal{J} \setminus \{D\} \cup \{D_1, D_2\}. \quad (5.88)$$

Recursively, next we compute $\text{REFINE_RECURSIVE}(\mathcal{J}_1, C)$ which updates the mesh to be Figure 5.5 (d).

$$\mathcal{J}_2 := \mathcal{J}_1 \setminus \{D_2, C\} \cup \{D_{21}, D_{22}, C_1, C_2\}. \quad (5.89)$$

Finally, we compute $\text{REFINE_RECURSIVE}(\mathcal{J}_2, T)$ and finish refining marked element T (see Figure 5.5 (e)).

$$\mathcal{J}_3 := \mathcal{J}_2 \setminus \{C_2, T\} \cup \{C_{21}, C_{22}, T_1, T_2\}. \quad (5.90)$$

Theorem 10. (Recursive Refinement of a single element $T \in \mathcal{J}$). Let \mathcal{J}_0 be a uniform conforming Kuhn triangulation and let \mathcal{J} be a conforming refinement. Then, for any $T \in \mathcal{J}$ a call of $\text{REFINE_RECURSIVE}(\mathcal{J}, T)$ terminates and outputs the smallest conforming refinement \mathcal{J}_* of \mathcal{J} where T is bisected. All newly created elements $T' \in \mathcal{J}_* \setminus \mathcal{J}$ satisfy $g(T') \leq g(T) + 1$.

Quadtree mesh refinement

To refine a marked 2D Quadtree element, we need to check whether it is divisible or not. If it is divisible, then we just need to connect the opposite edges' midpoints and cut the square element evenly into 2×2 smaller elements, which is summarized in Algorithm 6. As in Kuhn triangulation mesh, we give relative definitions for Quadtree mesh as follows.

Algorithm 6: 2D Quad cut divisible element Q into 2×2 children elements.

function QUADREF(Q)

Step 1, Find all the edges' midpoints of element Q .

Step 2, Connect the midpoints of opposite edges in element Q found in Step 1, and cut element Q evenly into 2×2 children elements $\{Q_1, \dots, Q_4\}$.

Return $\{Q_1, Q_2, Q_3, Q_4\}$.

Definition 20. (Quadtree forest and refinement). A Quadtree is a tree data structure in which each node has none or exact four children. Given initial square Q_0 , the Quadtree $\mathbb{F}(Q_0)$ holds full information the refined descendant elements that will be generated by recurrent bisection of Q_0 . Let \mathcal{K}_0 be an initial uniform square mesh. Then

$$\mathbb{F} = \mathbb{F}(\mathcal{K}_0) := \bigcup_{Q_0 \in \mathcal{K}_0} \mathbb{F}(Q_0) \quad (5.91)$$

is the associated *master forest* of Quadtrees. For a node $Q \in \mathbb{F}$ so that $Q \in \mathbb{Q}_g$ with $Q_0 \in \mathcal{K}_0$, the generation $g(Q)$ is the generation of Q within $\mathbb{F}(Q_0)$. A subset $\mathcal{F} \subset \mathbb{F}$ is called *forest* iff

- (1) $\mathcal{K}_0 \subset \mathcal{F}$;
- (2) All nodes of $\mathcal{F} \setminus \mathcal{K}_0$ have a predecessor;
- (3) All nodes of \mathcal{F} have either four successors or none for quadtree mesh.

A forest \mathcal{F} is called *finite* if $\max_{Q \in \mathcal{F}} g(Q) < \infty$. The nodes with no successors are called *leaves* in \mathcal{F} . Given two finite forests $\mathcal{F}_1, \mathcal{F}_2 \in \mathbb{F}$ with associated Quadtree meshes $\mathcal{K}_1, \mathcal{K}_2$, we call \mathcal{K}_2 as refinement of \mathcal{K}_1 iff $\mathcal{F}_1 \subset \mathcal{F}_2$ and we denote this by $\mathcal{K}_1 \leq \mathcal{K}_2$.

Definition 21. (Conforming Refinements). The class of *Conforming Refinements* of initial Quadtree mesh \mathcal{K}_0 is defined as

$$\mathbb{Q} = \{\mathcal{K} = \mathcal{K}(\mathcal{F}) \mid \mathcal{F} \subset \mathbb{F} \text{ is finite and } \mathcal{K}(\mathcal{F}) \text{ is conforming}\}. \quad (5.92)$$

Definition 22. (Refinement Patch in Quadtree mesh). Let \mathcal{K} be a given conforming mesh and let $Q \in \mathcal{K}$ be an element marked to be refined. We define the *refinement patch* of Q to be the set of all Q 's neighboring elements.

$$R(\mathcal{K}; Q) := \{Q' \in \mathcal{K} \mid Q' \in \mathcal{K} \text{ with } Q' \cap Q \neq \emptyset\}. \quad (5.93)$$

Next, we give out the theorem about divisibility in the Quadtree refinement patch.

Theorem 11. Let \mathcal{K} be a conforming Quadtree mesh refinement. Then any $Q \in \mathcal{K}$ is of locally highest generation in $R(\mathcal{K}; Q)$, i.e.,

$$g(Q) = \max\{g(Q') \mid Q' \in R(\mathcal{K}; Q)\}. \quad (5.94)$$

Futhermore, if $\min\{g(Q') \mid Q' \in R(\mathcal{K}; Q)\} = g(Q)$, then element Q is *divisible*. If a neighboring element of Q has generation number to be $g(Q) - 1$, then it is a transition element.

Definition 23. (Recursive Quadtree Refinement). If all the marked elements Q are divisible, we can refine them directly. Otherwise, if there are elements that are not divisible, which means $\min\{g(Q') \mid Q' \in R(\mathcal{K}; Q)\} < g(Q)$, the basic idea is to recursively refine these elements (with generation number $\min\{g(Q') \mid Q' \in R(\mathcal{K}; Q)\}$) first.

We summarize the recursive refinement of a single Quadtree element Q in Algorithm 7.

Example 4. (Recursive refinement of a single Quadtree element). As shown in Figure 5.6 (a), Q is the marked element to refine in the current conforming Quadtree mesh \mathcal{K} . The refinement patch of Q is $R(\mathcal{K}; Q) = \{A, B, C, D, Q\}$, with generation number $g(A) = g(D) = 2; g(B) = g(C) = g(T) = 3$. So we next access elements A and D , since $g(A) = g(D) = \min\{g(Q'') \mid Q'' \in R(\mathcal{K}, Q)\} < g(Q)$ and we need to compute $\text{REFINE_RECURSIVE_QUAD}(\mathcal{K}, [A, D])$ first. $R(\mathcal{K}; A) = \{H, F, T, B, A\}$ where element H reaches the minimum generation number and we need to compute

$\text{REFINE_RECURSIVE_QUAD}(\mathcal{K}, H)$; $R(\mathcal{K}; D) = \{G, F, T, C, D\}$ where element G reaches the minimum generation number and we need to compute $\text{REFINE_RECURSIVE_QUAD}(\mathcal{K}, G)$ (see Figure 5.6 (b)). $R(\mathcal{K}; H) = \{I, A, F, H\}$ and element H is divisible since $g(H) = \min\{g(Q'') \mid Q'' \in R(\mathcal{K}, H)\}$, so we refine element H directly to four 2×2 smaller square elements, H_1, H_2, H_3 , and

Algorithm 7: Recursive refinement of a single Quad element $Q \in \mathcal{K}$

function REFINE_RECURSIVE_QUADtree(\mathcal{K}, Q)

Do forever

 Get refinement patch $R(\mathcal{K}, Q)$;

 Access $Q' \in R(\mathcal{K}, Q)$ with $g(Q') = \min\{g(Q'') \mid Q'' \in R(\mathcal{K}, Q)\}$;

 If $g(Q') < g(Q)$ then

$\mathcal{K} := \text{REFINE_RECURSIVE_QUAD}(\mathcal{K}, Q')$;

 Else

 break;

 End if

end do

Get refinement patch $R(\mathcal{K}, Q)$;

For all $Q' \in R(\mathcal{K}, Q)$, do

 If $g(Q') = g(Q)$ then

$\{Q'_0, Q'_1, Q'_2, Q'_3\} = \text{QUADREF}(Q')$;

$\mathcal{K} := \mathcal{K} \setminus \{Q'\} \cup \{Q'_0, Q'_1, Q'_2, Q'_3\}$;

 End if

End for

Return \mathcal{K} .

H_4 . Similarly, element G is also divisible and is refined to four children elements, G_1, G_2, G_3 , and G_4 , as shown in 5.6 (c)).

$$\mathcal{K}_1 := \mathcal{K} \setminus \{H, G\} \cup \{H_1, H_2, H_3, H_4, G_1, G_2, G_3, G_4\} \quad (5.95)$$

Recursively, next we compute $\text{REFINE_RECURSIVE}(\mathcal{K}_1, [A, D])$ which updates the mesh to be Figure 5.6 (d).

$$\mathcal{K}_2 := \mathcal{K}_1 \setminus \{A, D\} \cup \{A_1, A_2, A_3, A_4, D_1, D_2, D_3, D_4\}. \quad (5.96)$$

Finally, we compute $\text{REFINE_RECURSIVE_QUAD}(\mathcal{K}_2, Q)$ and finish refining marked element Q ; see Figure 5.6 (e).

$$\mathcal{K}_3 := \mathcal{K}_2 \setminus \{Q\} \cup \{Q_1, Q_2, Q_3, Q_4\}. \quad (5.97)$$

Theorem 12. (Recursive Refinement of a single Quadtree element $Q \in \mathcal{K}$). Let \mathcal{K}_0 be a uniform conforming Quadtree mesh and let \mathcal{K} be a conforming refinement s.t. $\mathcal{K}_0 \leq \mathcal{K}$. Then, for any $Q \in \mathcal{K}$ a call of $\text{REFINE_RECURSIVE_QUAD}(\mathcal{K}, Q)$ terminates and outputs the smallest conforming refinement \mathcal{K}_* of \mathcal{K} where Q is Quad cut. All newly created elements $Q' \in \mathcal{K}_* \setminus \mathcal{K}$ satisfy $g(Q') = g(Q) + 1$.

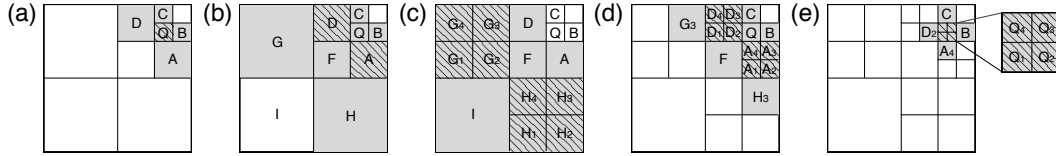


Figure 5.6: Example of recursive refinement of 2D Quadtree mesh.

5.9 Convergence and stopping criterion

The adaptive mesh regularized Global DIC and ALDIC method compute the following iteration until converged; see Algorithm 8.

$$SOLVE \rightarrow ESTIMATE \rightarrow MARK \rightarrow REFINE. \quad (5.98)$$

- In the *SOLVE* part, we solve the Galerkin solution of the above formulated regularized Global DIC (Section 5.2) and ALDIC (Section 5.3) problem with the h-adaptive mesh. In the adaptive regularized Global DIC method, a small number of Gauss points usually are very bad to apply Gaussian quadrature for integral approximations, where we still need to sum all the terms pixelwisely over the whole domain. However, in adaptive ALDIC method, since there are no image grayscale terms in the ALDIC Subproblem 2 global update, nine points Gaussian quadrature works well to approximate all the numerical integrals. So generally speaking, adaptive ALDIC is much faster than the adaptive regularized Global DIC method.
- In the *ESTIMATE* part, we compute a posterior element error estimates of adaptive regularized Global DIC $\mathcal{E}_G(\mathbf{U}, T)^2$ and adaptive ALDIC method (5.69-5.71) $\mathcal{E}_{AL1}(\mathbf{U}, T)^2$, and $\mathcal{E}_{AL2}(\mathbf{U}, T)^2$ as a weighted summation of elements interior error \mathbf{r} and element side jump error \mathbf{j} .
- In the *MARK* part, we mark elements with large a posteriori error estimates, which will further be refined in the next part. There are three popular marking strategies proposed in the literature (5.8), and we use the Dörfler strategy in our numerical demonstrations.
- In the *REFINE* part, we recursively refine all the marked elements to update Kuhn triangulation and Quadtree mesh (5.8).

Since the variational problems of regularized Global DIC and ALDIC Subproblem 2 are both formulated with coercive symmetric bilinear operator \mathfrak{B}_G or \mathfrak{B}_{AL} . Using

the above finite element spaces \mathbb{V} , a posteriori error estimate \mathcal{E} and refinement algorithms, the sequence of discrete Galerkin solution converges to the best solution in \mathbb{V} .

Theorem 13. The sequence of Galerkin solution $\{\mathbf{U}_k\}_{k \geq 0}$ generated by Algorithm 8 satisfies

$$\lim_{k \rightarrow \infty} \|\mathbf{U}_k - \mathbf{u}\|_{\mathbb{V}} = 0 \quad \text{and} \quad \lim_{k \rightarrow \infty} \mathcal{E}_k(\mathbf{U}_k, \mathcal{J}_k) = 0. \quad (5.99)$$

However, since we are dealing with digital images where only grayscale values at integer pixel positions are known exactly without additional interpolation bias errors. Both Kuhn triangulation and Quadtree mesh refinements always generate new nodes at midpoints of refined edges. So we can set the initial mesh element size to be a very large even number with a factor of powers of 2. The adaptive DIC refinements will stop when there exists a refined edge owning an odd length $\{\text{mod}(\min_{T \in \mathcal{J}_k} h_T, 2) = 1\}$ or where the shortest edge length already reaches 1 pixel $\{\min_{T \in \mathcal{J}_k} h_T \leq 1\}$. If our initial uniform mesh has edge length in the powers of 2 (pixels), we only need to consider the stopping condition $\{\min_{T \in \mathcal{J}_k} h_T \leq 1\}$.

The adaptive regularized Global DIC and adaptive ALDIC methods are summarized in Algorithm 8.

Algorithm 8: h-adaptive mesh regularized Global DIC/ALDIC methods

Input: Reference image f , deformed image g ;

Output: Displacement \mathbf{U}_N using RG-DIC or $\hat{\mathbf{U}}_N$ using ALDIC;

Step 1: Initialize displacements \mathbf{U}_0 using FFT method;

Step 2: Set up initial uniform mesh \mathcal{J}_0 and compute a posteriori error estimate $\mathcal{E}_0(\mathbf{U}_0, \mathcal{J}_0)$;

while $\{(\min_{T \in \mathcal{J}_k} h_T) \pmod{2}\}$ and $\{\min_{T \in \mathcal{J}_k} h_T > 1\}$ **do**

Step 3: $\mathbf{U}_k := \text{SOLVE}(\mathbb{V}_k)$ using adaptive regularized Global DIC (Section 5.2) or adaptive ALDIC (Section 5.3) ;

Step 4: $\{\mathcal{E}_k(\mathbf{U}_k, T)\}_{T \in \mathcal{J}_k} := \text{ESTIMATE}(\mathbf{U}_k, \mathcal{J}_k)$ and

$\mathcal{E}_k(\mathbf{U}_k, \mathcal{J}_k) = \sum_{T \in \mathcal{J}_k} \mathcal{E}_k(\mathbf{U}_k, T)$;

if $\mathcal{E}_k(\mathbf{U}_k, \mathcal{J}_k) / \mathcal{E}_0(\mathbf{U}_0, \mathcal{J}_0) < \varepsilon$,

break;

else

Step 5: $\mathcal{M}_k := \text{MARK}(\{\mathcal{E}_k(\mathbf{U}_k, T)\}_{T \in \mathcal{J}_k}, \mathcal{J}_k)$;

Step 6: $\mathcal{J}_{k+1} := \text{REFINE}(\mathcal{J}_k, \mathcal{M}_k)$; $k = k + 1$;

end if

end

Table 5.1: Comparison of the RMS errors in different DIC algorithms solved displacement and strain for the SEM 2D-DIC synthetic images of Sample 14 L5

	Adaptive mesh DIC method				Uniform mesh DIC methods		
	Adaptive Kuhn tri Global DIC	Adaptive Quadtree Global DIC	Adaptive Kuhn tri ALDIC	Adaptive Quadtree ALDIC	Local Subset DIC	Regularized Global DIC	ALDIC
Displacements	0.0349	0.0190	0.0226	0.0154	0.0201	0.0234	0.0141
Strain	2.07×10^{-3}	1.74×10^{-3}	1.56×10^{-3}	9.96×10^{-4}	3.20×10^{-3}	3.10×10^{-3}	1.00×10^{-3}

5.10 Demonstration

We illustrate the proposed adaptive ALDIC method using a series of case studies. These examples demonstrate the superior speed of the proposed algorithm while retaining good accuracy.

Case study I: Synthetic images Sample 14 from the SEM 2D-DIC Challenge

We test our proposed adaptive ALDIC algorithms using synthetic images from SEM 2D-DIC Challenge Sample 14 L5, where the deformations are sinusoidal with varying frequency in the x direction as shown in Figure 3.3.

We set all the local window sizes in Local Subset DIC and ALDIC Subproblem 1 to be 30×30 pixels, and set both the initial local neighboring windows distance uniformly to be 80×80 pixels. This is also the global mesh finite element size in ALDIC Subproblem 2. Every time the mesh is refined, the minimum of element size is halved until 5×5 pixels. We measure the RMS errors of solved displacements and strains in Table 5.1, and the results are also compared with the previous three DIC algorithms with uniform meshes.

First, we find adaptive ALDIC method (Figure 5.8) is more accurate than adaptive Global DIC method (Figure 5.7). In the adaptive ALDIC method, Kuhn triangulation refines the mesh in a more homogeneous way compared with Quadtree mesh (Figure 5.8 first row), whose overall accuracy is worse than the adaptive Quadtree mesh ALDIC, especially in the solved strain field.

The error in strain solved by the adaptive Quadtree mesh ALDIC is very close to the uniform mesh ALDIC but it saves more than 40% of computation time. It is also much faster than the adaptive Global DIC and current uniform Global DIC methods(cf Tables 5.2-5.3).

We plot two types of a posteriori error estimates in Figure 5.9. A posteriori error estimates based on ADMM Subproblem 2 continuously decrease both for adaptive

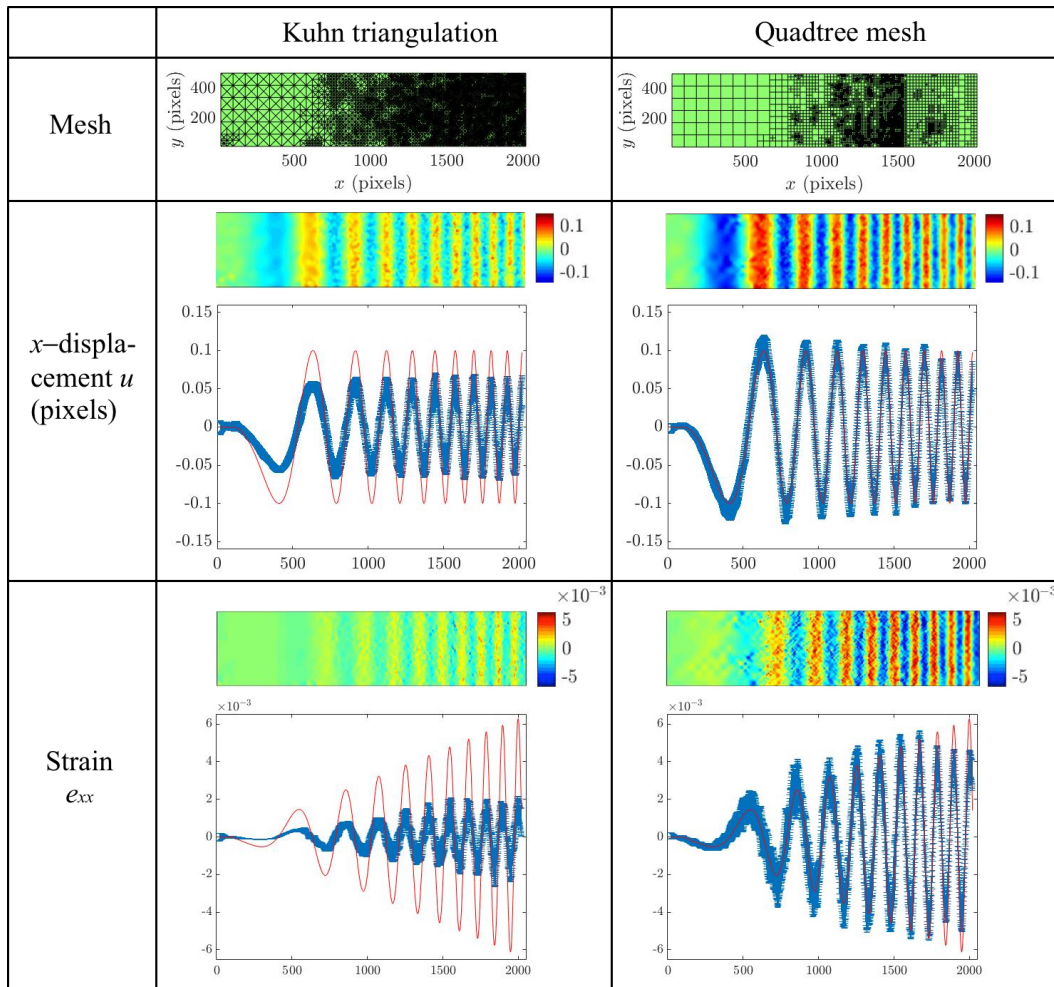


Figure 5.7: Comparison of solved Sample 14 L5 deformation field using adaptive regularized Global DIC with Kuhn triangulation and Quadtree mesh.

Kuhn triangulation and Quadtree ALDIC method, while the a posteriori error estimate of Subproblem 1 (normalized over the number of nodes) is insensitive to the mesh refinements.

To better visualize how a posteriori estimates guide us to mark elements to refine, we plot both two types a posteriori error estimates at the end of “SOLVE” part in the coarsest mesh in Figure 5.10. The a posteriori error estimates of ADMM Subproblem 1 in both Kuhn triangulation and Quadtree mesh have almost the same pattern: they are not sensitive to capture the oscillating strain changes. In contrast, a posteriori error estimates of ADMM Subproblem 2 can more accurately mark local regions where there are large strains and strain rates.

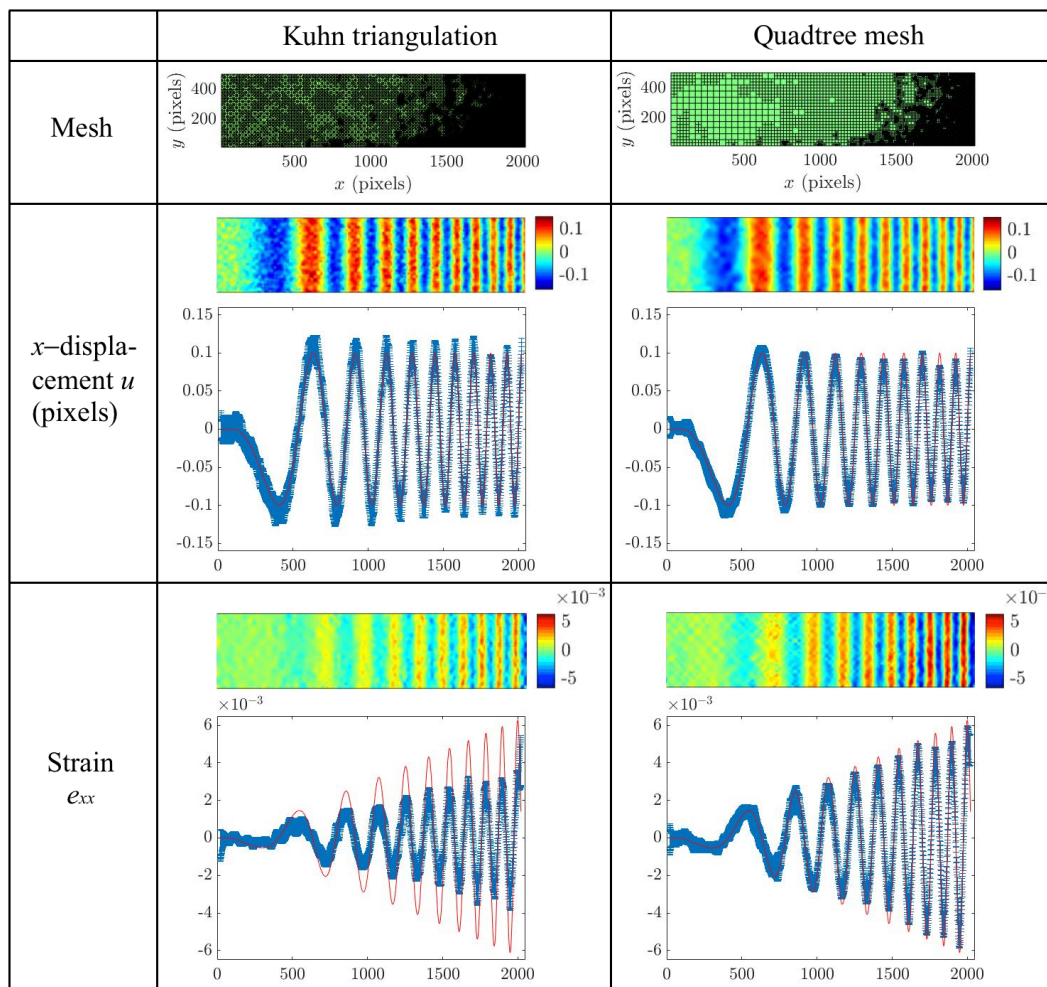


Figure 5.8: Comparison of solved Sample 14 L5 deformation field using adaptive ALDIC with Kuhn triangulation and Quadtree mesh.

Case study II: Collective cell migration

In the second example, we test adaptive ALDIC algorithms on real DIC experimental images. Here a collective group of cells are tracked through 2D-DIC methods to measure the collective cell migration, see Figure 5.11 [77]. All the cells are touching with each other, and there are almost no overlappings or voids among the cells.

We start from uniform finite element mesh (element size: 128×128 pixels) and adaptively refine elements where a posteriori error estimates \mathcal{E}_{AL1} and \mathcal{E}_{AL2} are large. We choose local subset size to be 12×12 in Local Subset DIC and ALDIC ADMM Subproblem 1 processes. Figure 5.12 and Figure 5.13 plot the solved deformation fields using adaptive ALDIC method with Kuhn triangulation and Quadtree meshes. Their solved deformation fields agree.

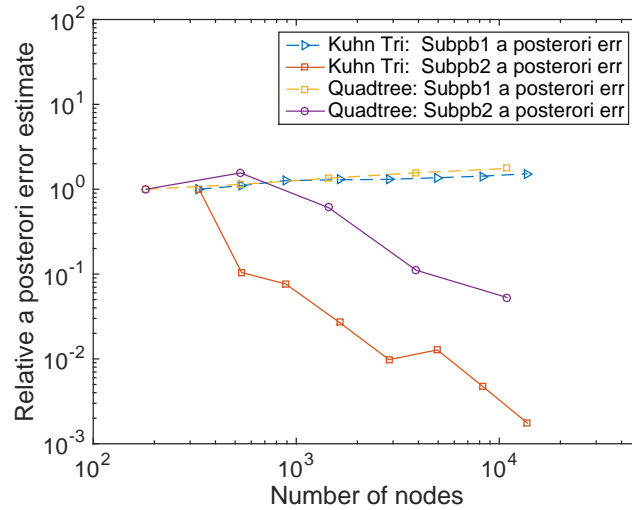


Figure 5.9: Plot of a posteriori error estimates of SEM 2D-DIC synthetic images of Sample 14 L5 based on ADMM Subproblem 1 and Subproblem 2.

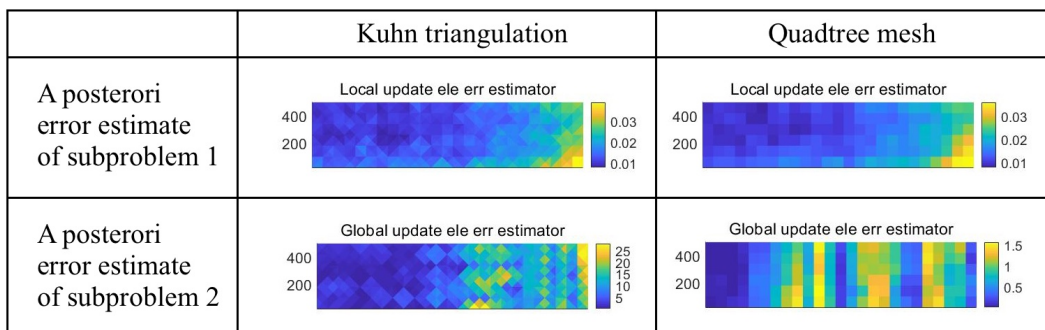


Figure 5.10: Comparison of a posteriori error estimate based on ADMM subproblem 1 local update and subproblem 2 global update.

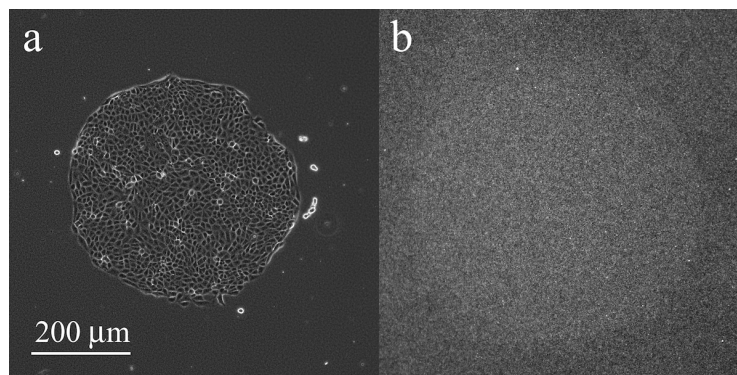


Figure 5.11: (a) Collective cells under microscope. (b) DIC image with speckle pattern to measure collective cell migration. (Images courtesy of Jacob Notbohm.)

Table 5.2: Computation time of adaptive regularized Global DIC using Quadtree mesh to solve SEM-2D synthetic images of Sample 14 L5

Minimum element size	Adaptive regularized Global mehtod						Uniform mesh DIC methods			
	# of nodes	Solve time cost(s)	Estimate time cost(s)	Mark time cost(s)	Refine time cost(s)	Each level adaptive Global DIC time cost(s)	# of nodes	Uniform Local Subset DIC time cost(s)	Uniform Global DIC time cost(s)	Uniform ALDIC time cost(s)
80×80	182	592.64	3.20	0.0048	0.26	596.10	182	6.81	592.64	33.39
40×40	377	504.86	6.61	0.0048	0.72	512.19	663	14.46	567.31	94.78
20×20	976	544.11	7.09	0.0055	1.81	553.02	2525	42.24	812.87	329.48
10×10	2560	692.54	8.01	0.0176	13.89	714.46	9849	156.22	1647.3	1143.7
5×5	7056	967.24	15.21	0.0754	0	982.53	38897	743.36	11967	2960.7
Total time cost(s) with minimum element size equals 5 pixels						3358.3	—	743.36	11967	2960.7

In keeping with the results of case study I, adaptive Kuhn triangulation has relatively more uniform mesh refinement compared with Quadtree adaptive mesh. Kuhn triangulation also smooths the strain field more than the Quadtree adaptive mesh. However, both these two adaptive ALDIC schemes have much less strain field noise compared with uniform Local Subset DIC; see Figure 5.14. Among all these DIC algorithms, adaptive Quadtree mesh ALDIC performs best for both good accuracy and fast computation speed (see Section 5.11).

We also checked a posteriori error estimates in Figure 5.15. The posteriori error estimates based on ADMM Subproblem 2 continuously decrease, while the a posteriori error estimate based on Subproblem 1 — summation of squared differences (SSD) correlation function — is insensitive to the mesh refinements.

5.11 Computation cost

All the computations are performed on the same workstation with Intel (R) Xeon(R) CPU E5-2650 v3 2.30 GHz (2 Processors), RAM 32.0 GB Memory, 64-bit nodes using Matlab. In the Local Subset DIC and ALDIC Subproblem 1 IC-GN iterations, we use 20 clusters to perform parallel computing.

We summarize the computation time of our demonstrations in Tables 5.3-5.7. We find that the adaptive Quadtree mesh ALDIC significantly saves computation time compared with uniform mesh ALDIC and uniform Global DIC methods with just a small loss in accuracy. Adaptive regularized Global DIC method usually are very slow compared with adaptive ALDIC method. And adaptive Kuhn triangulation ALDIC is also slow because it refines mesh more uniformly and needs more refinement time to reach the same smallest element size with adaptive Quadtree mesh.

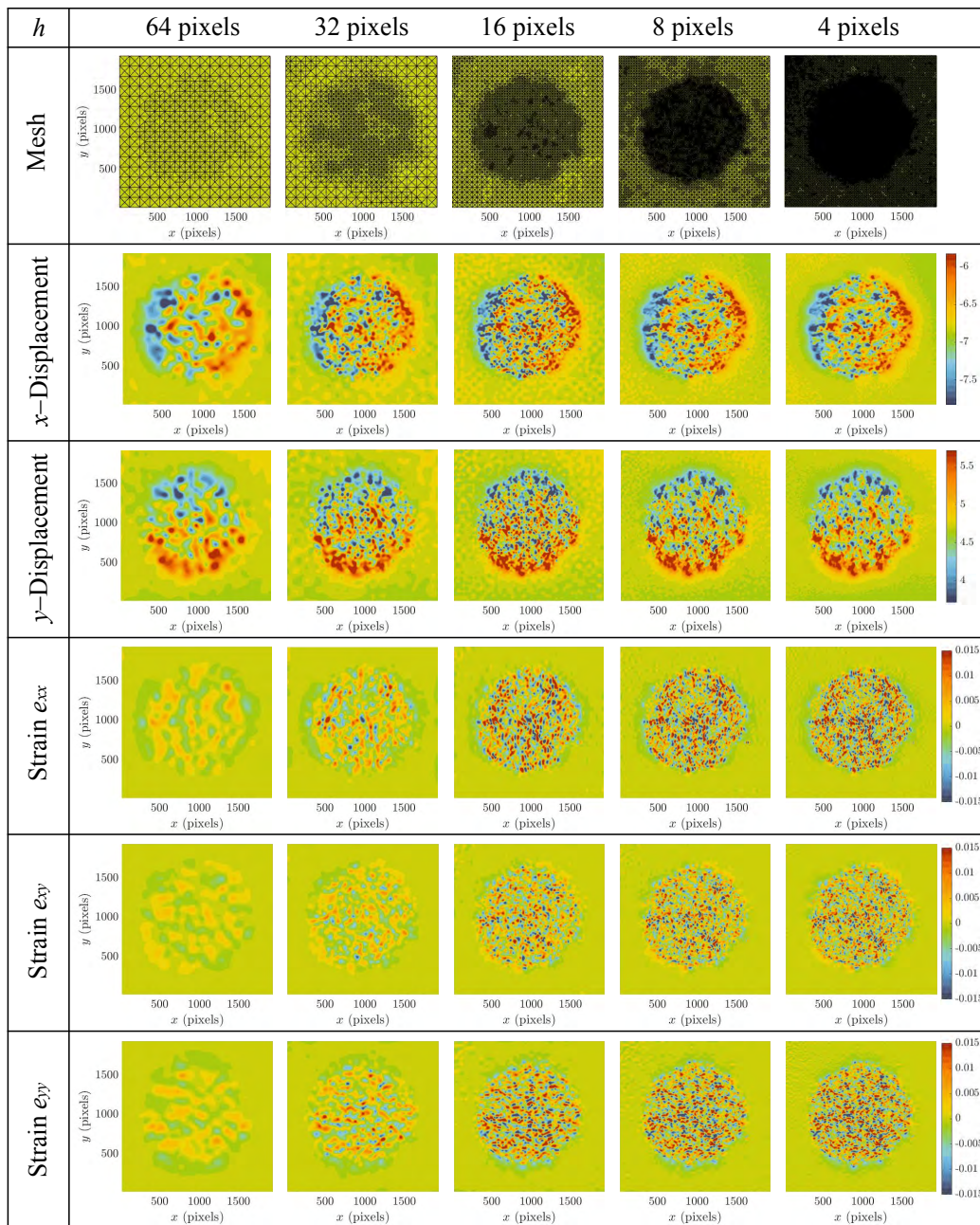


Figure 5.12: Adaptive ALDIC solved collective cell migration using Kuhn adaptive triangulation mesh.

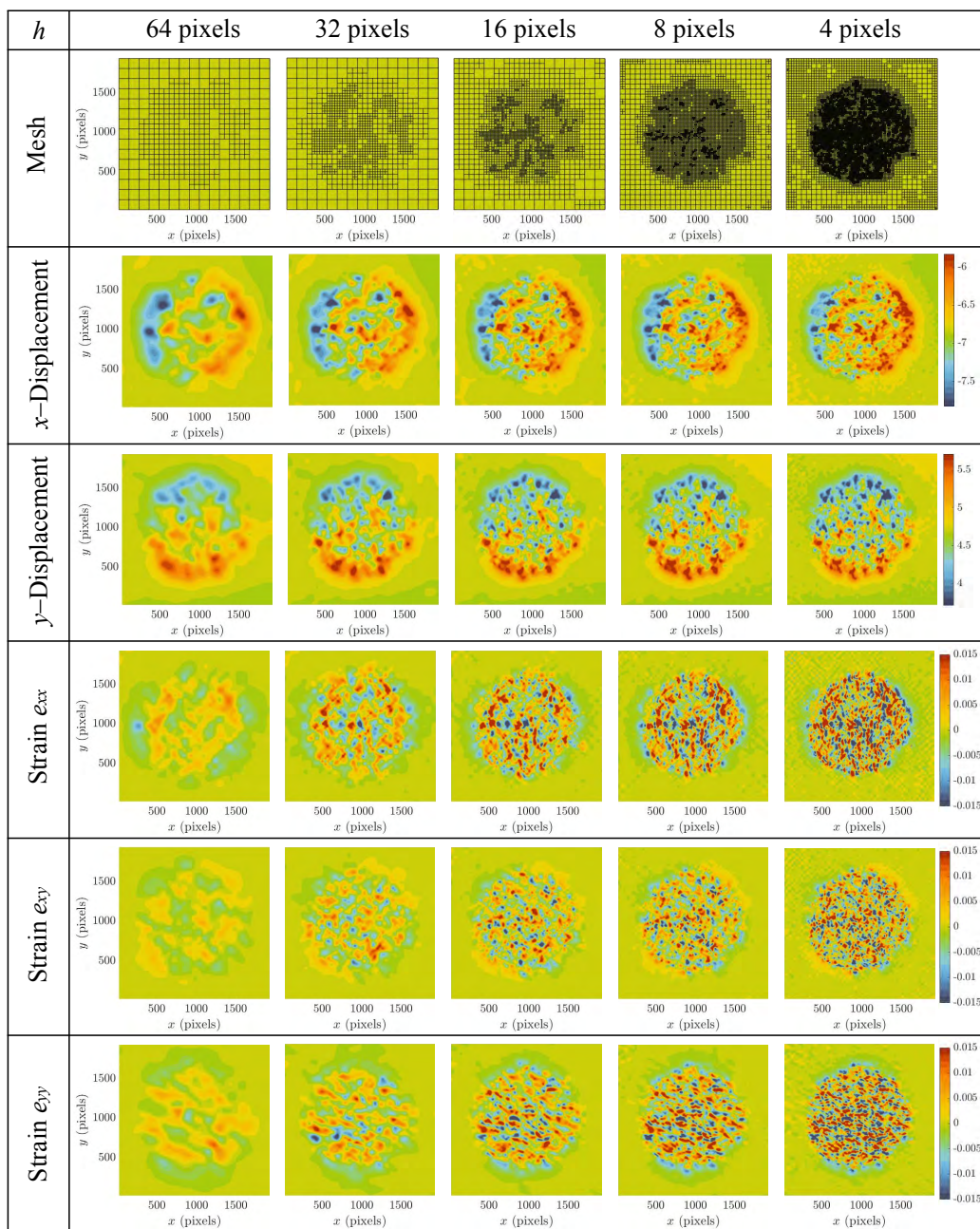


Figure 5.13: Adaptive ALDIC solved collective cell migration using Quadtree adaptive mesh.

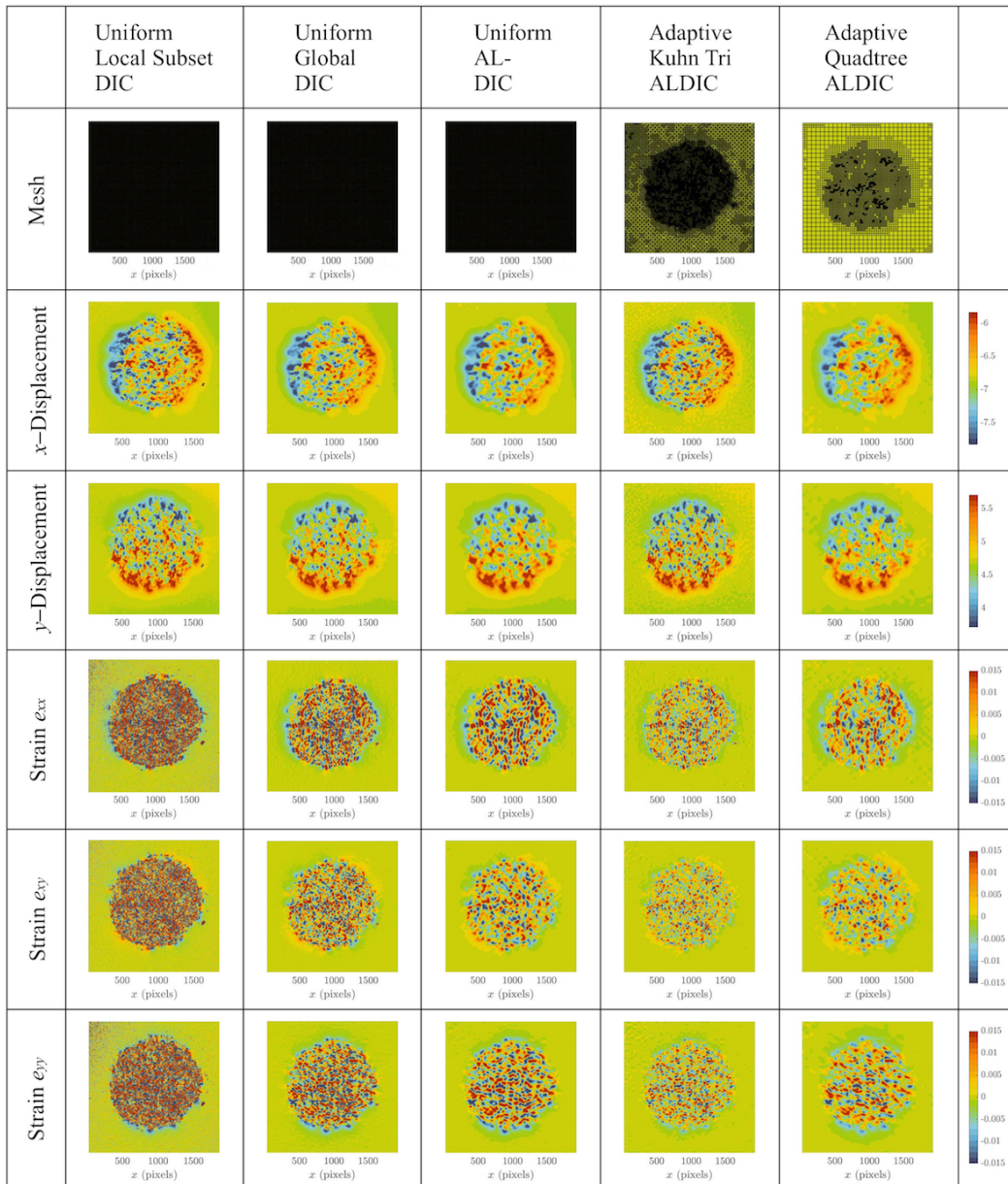


Figure 5.14: Comparison between solved collective cell migration using uniform Local Subset DIC, adaptive Kuhn triangulation ALDIC, and adaptive Quadtree mesh ALDIC.

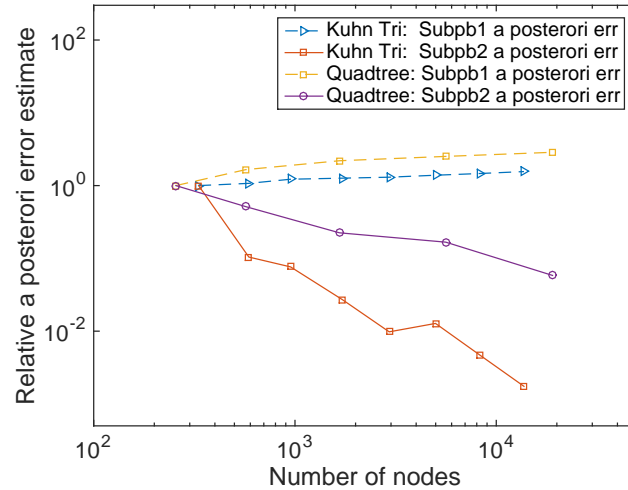


Figure 5.15: Plot of a posteriori error estimates of cell migration based on ADMM Subproblem 1 and Subproblem 2.

Table 5.3: Computation time of adaptive ALDIC using Quadtree mesh to solve SEM-2D synthetic images of Sample 14 L5

Minimum element size	Adaptive ALDIC mehtod						Uniform mesh DIC methods			
	# of nodes	Solve time cost(s)	Estimate time cost(s)	Mark time cost(s)	Refine time cost(s)	Each level adaptive ALDIC time cost(s)	# of nodes	Uniform Subset DIC time cost(s)	Uniform Global DIC time cost(s)	Uniform ALDIC time cost(s)
80×80	182	28.14	3.20	0.0094	0.40	31.75	182	6.81	592.64	33.39
40×40	531	48.09	3.71	0.0062	0.95	52.76	663	14.46	567.31	94.78
20×20	1444	156.50	4.84	0.0105	4.58	165.93	2525	42.24	812.87	329.48
10×10	3898	320.38	12.15	0.0273	45.40	377.96	9849	156.22	1647.3	1143.7
5×5	10907	1070.9	42.31	0.0948	0	1113.3	38897	743.36	11967	2960.7
Total time cost(s) with minimum element size equals 5 pixels						1741.7	—	743.36	11967	2960.7

Table 5.4: Computation time of adaptive ALDIC using Kuhn triangulation mesh to solve SEM-2D synthetic images of Sample 14 L5

Minimum element size	Adaptive ALDIC mehtod						Each level adaptive ALDIC time cost(s)
	# of nodes	Solve time cost(s)	Estimate time cost(s)	Mark time cost(s)	Refine time cost(s)		
$40\sqrt{2} \times 40\sqrt{2}$	332	84.14	0.58	0.0084	0.87	85.60	
40×40	538	87.60	3.79	0.0075	1.43	92.83	
$20\sqrt{2} \times 20\sqrt{2}$	888	135.28	3.60	0.0125	2.61	141.49	
20×20	1634	230.88	4.83	0.0181	4.39	240.10	
$10\sqrt{2} \times 10\sqrt{2}$	2863	398.80	6.37	0.0413	7.68	412.89	
10×10	4952	669.39	10.04	0.0560	13.88	493.37	
$5\sqrt{2} \times 5\sqrt{2}$	8315	1134.53	13.72	0.0871	25.59	1173.9	
5×5	13772	1933.83	23.34	0.1622	0	1957.3	
Total time cost(s) with minimum element size equals 5 pixels							4597.5

Table 5.5: Computation time of adaptive regularized Global DIC using Quadtree mesh to solve collective cell migration

Minimum element size	Adaptive regularized Global mehtod						Uniform mesh DIC methods			
	# of nodes	Solve time cost(s)	Estimate time cost(s)	Mark time cost(s)	Refine time cost(s)	Each level adaptive Global DIC time cost(s)	# of nodes	Uniform Local Subset DIC time cost(s)	Uniform Global DIC time cost(s)	Uniform ALDIC time cost(s)
128×128	256	2226.4	40.17	0.0057	3.62	2272.0	256	14.26	2226.4	41.23
64×64	497	2305.6	40.31	0.0050	3.40	2349.3	961	36.75	3461.9	124.47
32×32	1423	3368.7	45.53	0.0085	5.05	3419.3	3721	22.34	6380.6	155.66
16×16	4151	4375.8	43.12	0.02207	36.36	4455.3	14641	49.12	13350	385.54
8×8	10404	6095.8	67.79	0.05627	642.35	6806.0	58078	201.58	23958	1881.9
Total time cost(s) with minimum element size equals 8 pixels						19302	—	201.58	23958	1881.9

Table 5.6: Computation time of adaptive ALDIC using Quadtree mesh to solve collective cell migration

Minimum element size	Adaptive ALDIC mehtod						Uniform mesh DIC methods			
	# of nodes	Solve time cost(s)	Estimate time cost(s)	Mark time cost(s)	Refine time cost(s)	Each level adaptive ALDIC time cost(s)	# of nodes	Uniform Local Subset DIC time cost(s)	Uniform Global DIC time cost(s)	Uniform ALDIC time cost(s)
128×128	256	41.23	8.42	0.0074	0.37	50.03	256	14.26	2226.4	41.23
64×64	570	75.54	12.33	0.0064	3.17	91.05	961	36.75	3461.9	124.47
32×32	1664	81.20	10.11	0.0092	6.83	98.15	3721	22.34	6380.6	155.66
16×16	5613	238.84	19.33	0.0340	20.92	279.12	14641	49.12	13350	385.54
8×8	19037	780.34	27.97	0.0877	0	808.40	58078	201.58	23958	1881.9
Total time cost(s) with minimum element size equals 8 pixels						1326.8	—	201.58	23958	1881.9

Table 5.7: Computation time of adaptive ALDIC using adaptive Kuhn triangulation mesh to solve collective cell migration

Minimum element size	Adaptive ALDIC mehtod					
	# of nodes	Solve time cost(s)	Estimate time cost(s)	Mark time cost(s)	Refine time cost(s)	Each level adaptive ALDIC time cost(s)
$64\sqrt{2} \times 64\sqrt{2}$	332	50.74	1.28	0.0084	1.05	53.08
64×64	590	126.89	3.79	0.0075	2.34	133.03
$32\sqrt{2} \times 32\sqrt{2}$	951	145.42	3.59	0.0125	5.32	154.34
32×32	1728	246.58	4.83	0.018	13.12	264.55
$16\sqrt{2} \times 16\sqrt{2}$	2938	402.80	6.37	0.041	25.48	434.69
16×16	4991	676.23	10.04	0.056	30.12	716.45
$8\sqrt{2} \times 8\sqrt{2}$	8268	1121.9	13.72	0.087	35.87	1171.6
8×8	13731	1888.1	23.34	0.162	0	1911.6
Total time cost(s) with minimum element size equals 5 pixels						4839.3

5.12 Conclusion

In this chapter, we have developed a fast, efficient adaptive mesh DIC method using our proposed ALDIC and ADMM scheme. We compare it with other adaptive DIC methods in Figure 5.16. Current adaptive Local Subset DIC [70](Figure 5.16 left), and adaptive Global DIC [35](Figure 5.16 middle) use image matching error, e.g., SSD (summation of squared differences) correlation function, as a mesh refinement criterion. We find that this is not efficient or robust. We develop a new h-adaptive technique and apply it to Global DIC method. However, this method is very expensive, since the stiffness matrix and the external force vector have to be rebuilt every time the mesh is changed and there involves the summation of image grayscale value terms pointwisely. In our newly proposed adaptive ALDIC method using ADMM scheme (Figure 5.16 right), both subproblems can be solved very fast using adaptive mesh. Therefore, this has the best overall robustness and efficiency among current adaptive DIC methods.

Dealing with digital images with the unit of pixel in DIC experiments, we apply adaptive Kuhn triangulation and adaptive Quadtree mesh to decrease the number of nodes. We use a posteriori error estimates as mesh refinement criterion which considers both the image matching errors (SSD correlation function, related with ADMM Subproblem 1 local update) and the global kinematic compatibility between the neighboring elements (related with ADMM Subproblem 2 global update). We find a posteriori error estimate that based on the image matching errors (Subproblem 1) is not sensitive to the mesh refinement, while a posteriori error estimate based on the ADMM Subproblem 2 is. Between adaptive Kuhn triangulation and adaptive Quadtree mesh in two-dimensional DIC problems, the latter adaptive mesh strategy refines mesh more non-uniformly, and can significantly save computation cost with little loss in accuracy.

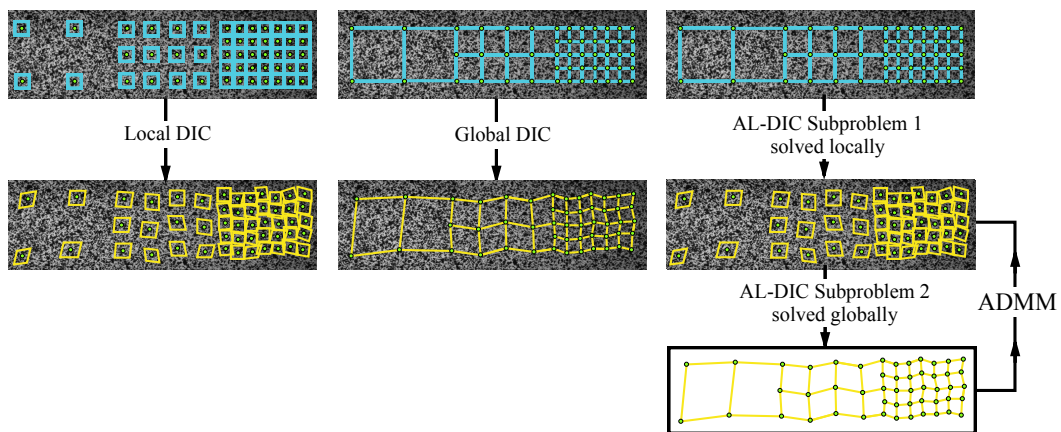


Figure 5.16: Comparison of adaptive Local Subset DIC method, adaptive regularized Global DIC method and adaptive ALDIC method.

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this thesis, we have developed a new digital image correlation (DIC) algorithm, augmented Lagrangian DIC (ALDIC) in Chapter 3. ALDIC combines both advantages of current Local Subset DIC method (fast, in parallel) and Global DIC method (compatible, certain smoothness). Using the alternating direction method of multipliers (ADMM), ALDIC can be solved fast, and its computation time usually is two to four times of Local Subset DIC, and less than the Global DIC method.

DIC requires lots of high resolution images, which imposes significant needs on data storage and transmission. We combine DIC algorithms with the image compression techniques in Chapter 4 and find that it is still possible to obtain accurate deformation fields with only 5% of the original image size. Among the three DIC algorithms, we find that Local Subset DIC leads to the largest errors and ALDIC to the smallest when compressed images are used. We also find that the wavelet image compression technique introduces less error compared to the discrete cosine transform image compression technique.

To further speed up DIC algorithms, especially in the study of complex heterogeneous strain fields at various length scales, we developed a new h-adaptive mesh technique and applied it to ALDIC in Chapter 5. We demonstrated the efficiency of this new adaptive mesh ALDIC through synthetic images and real experimental DIC images. We studied two types of adaptive mesh strategies, Kuhn triangulation and Quadtree mesh in adaptive ALDIC, and found that Quadtree mesh outperforms Kuhn triangulation adaptive mesh both in accuracy and computation cost.

6.2 Future work

3D DVC ALDIC algorithm

All the demonstrations in this thesis are limited to two-dimensional images. However, the presentation of the method and algorithm is valid for two- and three-dimension(or digital volume correlation, DVC). We hope to extend our 2D ALDIC algorithm to 3D DVC case in the future.

As for adaptive ALDIC, we also extend 2D Quadtree adaptive mesh to 3D Octree

adaptive mesh (cf Appendix C), where all the elements keep the shape and is easier to implement compared with 3D Kuhn triangulation mesh.

Phase correlation methods to speed up ALDIC

We use the IC-GN iteration for the ADMM Subproblem 1 local update. This is robust but still slow since it involves pixel-wise summation. We can further improve this by using phase correlation methods and fast Fourier transforms that have been introduced for Local Subset DIC method. In practice, phase correlation methods are not as accurate as IC-GN iterations when the deformations are small. However, it is much faster and our early results show that it still works well with very large strains (could be around 40%) [78].

Dealing with discontinuities

All the demonstrations in this thesis are limited to continuous deformations. However, the adaptive mesh ALDIC algorithm can be extended to discontinuous deformation fields. Specifically, in ADMM Subproblem 1 local step, subsets passed by the crack path need to be treated with subset splitting strategies [79], and in the ADMM Subproblem 2 global update, the finite element basis needs to be extended with discontinuous functions [54]. After each ADMM iteration (Subproblem 1 and Subproblem 2), the level set method [54] can be used to update the crack path.

Hp-adaptive DIC

We focus on h-adaptive mesh DIC method in Chapter 5. Besides h-adaptive mesh DIC, p-adaptive mesh DIC has been shown to work very well for very large elements whose sizes are from 50×50 pixels to 100×100 pixels [71]. We wonder whether we can continue improving the computation efficiency and overall accuracy of DIC algorithm by combining h-adaptive DIC and p-adaptive DIC.

Appendix A

THE OPERATOR **D**

The matrix **D** in Section 3 is the discrete gradient operator. This depends on the choice of discretization. In this paper, we use first order finite difference based on a uniform square mesh. We provide explicit details for this case, but note that ALDIC is compatible with any discretization and would lead to different matrices.

We describe the explicit discrete gradient operator in one dimension for convenience, and the generalization to higher dimensions is obvious. We assume that the domain is discretized uniformly with the distance h between nodes $x_1, x_2 \dots x_N$. Then the equation (3.1) is explicitly written as

$$\underbrace{\begin{pmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \\ \mathbf{F}_3 \\ \vdots \\ \mathbf{F}_{N-1} \\ \mathbf{F}_N \end{pmatrix}}_{\{\mathbf{F}\}} = \frac{1}{2h} \underbrace{\begin{bmatrix} -2 & 2 & & & & \\ -1 & 0 & 1 & & & \\ & -1 & 0 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 0 & 1 \\ & & & & -2 & 2 \end{bmatrix}}_{\mathbf{D}} \underbrace{\begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \\ \vdots \\ \mathbf{u}_{N-1} \\ \mathbf{u}_N \end{pmatrix}}_{\{\mathbf{u}\}}, \quad (\text{A.1})$$

where $\mathbf{u}^i, \mathbf{F}^i$ are the values at node x^i .

In two dimension, the approximation of $\nabla \hat{u}$ using matrix **D** is very similar to the above one dimension case. For example, when the finite element meshes are 4×4 and 10×10 FEM Q4 meshes, with element length $h = 1$, matrix **D**'s are shown in Figure A.1 and they are very sparse, especially when the mesh size is large.

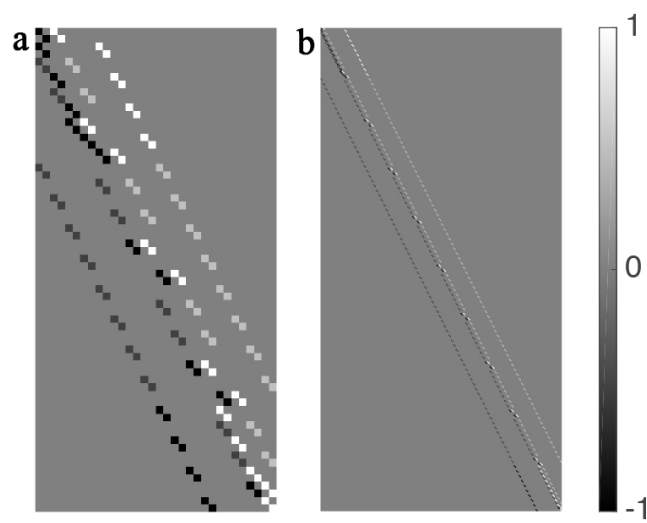


Figure A.1: Matrix D in (a) 4×4 and (b) 10×10 FEM Q4 meshes where both element length $h = 1$.

Appendix B

PROOF OF THEOREM 1

Theorem 1. (Modified Lax-Milgram Theorem) Let $\mathfrak{B}_G : H^1(\Omega) \times H^1(\Omega) \rightarrow \mathbb{R}$ be a continuous bilinear form that is coercive, then the equation system Eqs(5.4-5.5) has a unique solution $\delta \mathbf{u} \in H^1(\Omega)$, which depends continuously on \mathbf{p} .

Proof. We denote by $J : H^1(\Omega) \rightarrow H^{-1}(\Omega)$ the isometric Riesz isomorphism between $H^1(\Omega)$ and $H^{-1}(\Omega)$. Then by Riesz representation theorem, there exists a unique linear operator $\mathbf{B}_G \in L(H^1(\Omega); H^1(\Omega))$ such that

$$\langle \mathbf{B}_G \mathbf{v}, \mathbf{w} \rangle_{H^1(\Omega)} = \mathfrak{B}_G[\mathbf{v}, \mathbf{w}] \quad \text{for all } \mathbf{v}, \mathbf{w} \in H^1(\Omega), \quad (\text{B.1})$$

where $L(H^1(\Omega); H^1(\Omega))$ is the space of all linear and continuous operators from $H^1(\Omega)$ into $H^1(\Omega)$ with operator norm

$$\begin{aligned} |\mathbf{B}_G|_{L(H^1(\Omega); H^1(\Omega))} &= \sup_{\mathbf{v} \in H^1(\Omega)} \frac{|\mathbf{B}\mathbf{v}|_{H^1(\Omega)}}{|\mathbf{v}|_{H^1(\Omega)}} \\ &= \sup_{\mathbf{v} \in H^1(\Omega)} \sup_{\mathbf{w} \in H^1(\Omega)} \frac{\langle \mathbf{B}\mathbf{v}, \mathbf{w} \rangle_{H^1(\Omega)}}{|\mathbf{v}|_{H^1(\Omega)} |\mathbf{w}|_{H^1(\Omega)}} \\ &= \sup_{\mathbf{v} \in H^1(\Omega)} \sup_{\mathbf{w} \in H^1(\Omega)} \frac{\mathfrak{B}_G[\mathbf{v}, \mathbf{w}]}{|\mathbf{v}|_{H^1(\Omega)} |\mathbf{w}|_{H^1(\Omega)}} := |\mathfrak{B}_G|. \end{aligned} \quad (\text{B.2})$$

Then, the original problem (5.4) is equivalent to

$$\delta \mathbf{u} \in H^1(\Omega) : \quad \mathbf{B}_G \delta \mathbf{u} = J^{-1} \mathbf{p}. \quad (\text{B.3})$$

Claim: Coercicity of \mathfrak{B}_G guarantees the above unique linear operator $\mathbf{B}_G : H^1(\Omega) \rightarrow H^1(\Omega)$ is an *isomorphism* and invertible with $|\mathbf{B}_G^{-1}|_{L(H^1(\Omega); H^1(\Omega))} \leq C_1^{-1}$, where C_1 is the same number with (5.7).

Then from (5.5), $\forall \varphi \in H^1(\Omega)$,

$$\begin{aligned} |\langle \mathbf{p}, \varphi \rangle| &\leq \alpha |\mathbf{u}_n|_{H^1(\Omega)} |\varphi|_{H^1(\Omega)} + \sup |f(\mathbf{X}) - g(\mathbf{X} + \mathbf{u}_n)| |\nabla f|_{L^2} |\varphi|_{L^2} \\ &\leq \alpha |\mathbf{u}_n|_{H^1(\Omega)} |\varphi|_{H^1(\Omega)} + C_2 |\Omega|^{1/d} |\nabla f|_{L^2} |\varphi|_{H^1(\Omega)} \\ &\leq \left(\alpha |\mathbf{u}_n|_{H^1(\Omega)} + C_2 |\Omega|^{1/d} |\nabla f|_{L^2} \right) |\varphi|_{H^1(\Omega)}, \end{aligned} \quad (\text{B.4})$$

where C_2 is some real finite number. Thus, we have

$$|\mathbf{p}|_{H^{-1}(\Omega)} \leq \left(\alpha |\mathbf{u}_n|_{H^1(\Omega)} + C_2 |\Omega|^{1/d} |\nabla f|_{L^2} \right). \quad (\text{B.5})$$

Thus, $\mathbf{p} \in H^{-1}(\Omega)$ and there is unique solution $\delta \mathbf{u} \in H^1(\Omega)$ of (B.3) is given by $\delta \mathbf{u} = \mathbf{B}_G^{-1} J^{-1} \mathbf{p}$ and $\delta \mathbf{u}$ depends continuously on \mathbf{p} with

$$\begin{aligned} |\delta \mathbf{u}|_{H^1(\Omega)} &\leq |\mathbf{B}_G^{-1}|_{L(H^1(\Omega); H^1(\Omega))} |J^{-1} \mathbf{p}|_{H^1(\Omega)} = |\mathbf{B}_G^{-1}|_{L(H^1(\Omega); H^1(\Omega))} |\mathbf{p}|_{H^{-1}(\Omega)} \\ &\leq C_1^{-1} |\mathbf{p}|_{H^{-1}(\Omega)} \leq C_1^{-1} \left(\alpha |\mathbf{u}_n|_{H^1(\Omega)} + C_2 |\Omega|^{1/d} |\nabla f|_{L^2} \right). \end{aligned} \quad (\text{B.6})$$

Finally, we prove the above claim. Using coercivity, $\forall \mathbf{u} \in H^1(\Omega)$,

$$C_1 |\mathbf{u}|_{H^1(\Omega)} \leq \frac{\mathfrak{B}_G[\mathbf{u}, \mathbf{u}]}{|\mathbf{u}|_{H^1(\Omega)}} = \frac{\langle \mathbf{B}_G \mathbf{u}, \mathbf{u} \rangle_{H^1(\Omega)}}{|\mathbf{u}|_{H^1(\Omega)}} \leq \sup_{\mathbf{v} \in H^1(\Omega)} \frac{\langle \mathbf{B}_G \mathbf{u}, \mathbf{v} \rangle_{H^1(\Omega)}}{|\mathbf{v}|_{H^1(\Omega)}} = |\mathbf{B}_G \mathbf{u}|_{H^1(\Omega)}. \quad (\text{B.7})$$

If $\mathbf{B}_G \mathbf{v} = 0$, using (B.7),

$$0 \leq C_1 |\mathbf{v}|_{H^1(\Omega)} \leq |\mathbf{B}_G \mathbf{v}|_{H^1(\Omega)} = 0, \quad \text{with } C_1 > 0. \quad (\text{B.8})$$

The above inequalities must hold equality everywhere and $\mathbf{v} = 0$, whence \mathbf{B}_G is *injective*. Let $\mathbf{w}_k = \mathbf{B}_G \mathbf{v}_k$ be a sequence such that $\mathbf{w}_k \rightarrow \mathbf{w} \in H^1(\Omega)$ as $k \rightarrow \infty$. Invoking (B.7), we have

$$C_1 |\mathbf{v}_k - \mathbf{v}_j|_{H^1(\Omega)} \leq |\mathbf{B}_G(\mathbf{w}_k - \mathbf{w}_j)|_{H^1(\Omega)} = |\mathbf{w}_k - \mathbf{w}_j|_{H^1(\Omega)} \rightarrow 0, \quad \text{as } j, k \rightarrow \infty. \quad (\text{B.9})$$

Thus $\{\mathbf{v}_k\}_{k=0}^{\infty}$ is a Cauchy sequence in $H^1(\Omega)$ and it converges $\mathbf{v}_k \rightarrow \mathbf{v} \in H^1(\Omega)$ as $k \rightarrow \infty$. Continuity of \mathbf{B}_G yields that

$$\mathbf{B}_G \mathbf{v} = \lim_{k \rightarrow \infty} \mathbf{B}_G \mathbf{v}_k = \mathbf{w} \in \mathbf{B}_G(H^1(\Omega)), \quad (\text{B.10})$$

which shows that $\mathbf{B}_G(H^1(\Omega))$ is *closed*.

We argue \mathbf{B}_G is also *surjective* by contradiction. Assume $\mathbf{B}_G(H^1(\Omega)) \neq H^1(\Omega)$. Since $\mathbf{B}_G(H^1(\Omega))$ is closed we can decompose $H^1(\Omega) = \mathbf{B}_G(H^1(\Omega)) \oplus \mathbf{B}_G(H^1(\Omega))^\perp$, where $\mathbf{B}_G(H^1(\Omega))^\perp$ is the orthogonal complement of $\mathbf{B}_G(H^1(\Omega))$ in $H^1(\Omega)$. By assumption $\mathbf{B}_G(H^1(\Omega))^\perp$ is non-trivial, there exists $0 \neq \mathbf{w}_0 \in \mathbf{B}_G(H^1(\Omega))^\perp$,

$$\mathbf{w}_0 \neq 0, \quad 0 = \langle \mathbf{B}_G \mathbf{v}, \mathbf{w}_0 \rangle = \mathfrak{B}_G[\mathbf{v}, \mathbf{w}_0], \quad \forall \mathbf{v} \in H^1(\Omega). \quad (\text{B.11})$$

Choose $\mathbf{v} = \mathbf{w}_0$, above equation contradicts (5.7). So we have that $\mathbf{B}_G(H^1(\Omega)) = H^1(\Omega)$. Therefore, we conclude that \mathbf{B}_G is an isomorphism from $H^1(\Omega)$ onto $H^1(\Omega)$. We can further rewrite (B.7) as follows:

$$C_1 \left| \mathbf{B}_G^{-1} \mathbf{w} \right|_{H^1(\Omega)} \leq \left| \mathbf{w} \right|_{H^1(\Omega)}, \quad \forall \mathbf{w} \in H^1(\Omega), \quad (\text{B.12})$$

which means $\left| \mathbf{B}_G^{-1} \right|_{L(H^1(\Omega); H^1(\Omega))} \leq C_1^{-1}$. ■

Appendix C

3D KUHN TRIANGULATION AND OCTREE ADAPTIVE MESH

In 3D, a cube can be subdivided by three planar cuts in 6 similar Kuhn triangulation simplices (cf Figure C.1(a)), however, all the descendants are belong to three similarity classes [80].

As for the 3D Kuhn simplex descendants, there are three shapes in different refinement levels with fixed shape regularities; see Figure C.1. Compared with 3D Kuhn triangulation, 3D Octree adaptive mesh, each parent element has none or exact eight children elements, is easier to implement where all the elements keep the cubic shape, (cf Figure C.2). Similarly as 2D Quadtree mesh, there are hanging nodes and transition elements in Octree mesh which connect different size elements in 3D Octree mesh.

FE shape functions of 3D Octant elements

In an analogous manner with 2D Quadtree, we still apply Gupta's strategy [75] to generate 3D Octree mesh element shape functions. Using isoparametric mapping,

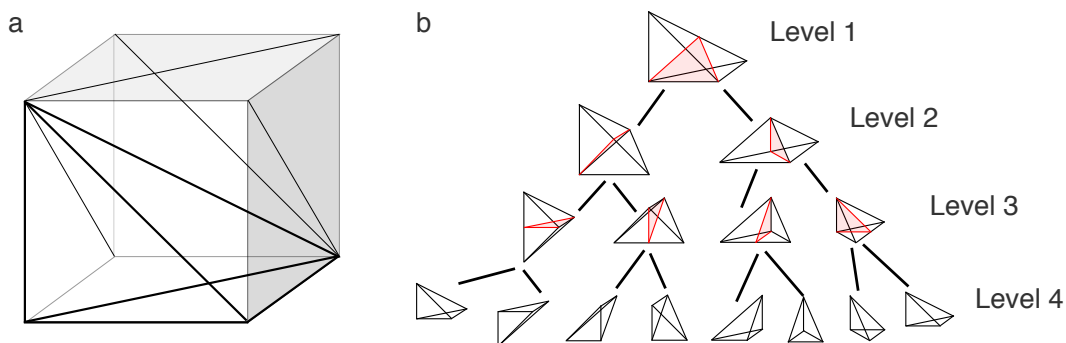


Figure C.1: Kuhn 3D simplex. (a) A 3D cube can be subdivided by three planar cuts in 6 similar Kuhn triangulation simplices. (b) Binary bisection tree of a 3D Kuhn triangulation simplex.

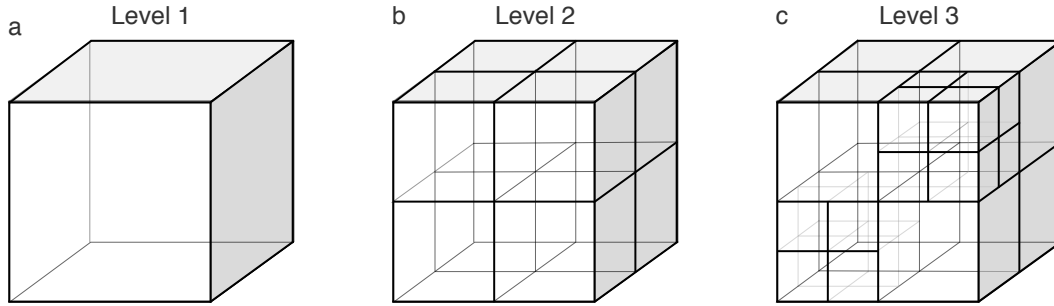


Figure C.2: Adaptive 3D Octree mesh refinement.

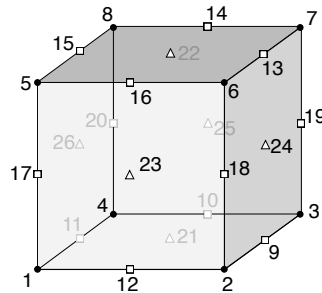


Figure C.3: 3D Octree element template.

classic cube finite element basis shape functions are defined as

$$\left\{ \begin{array}{l} N_1 = \frac{1}{8}(1 - \xi)(1 - \eta)(1 - \zeta); \\ N_2 = \frac{1}{8}(1 + \xi)(1 - \eta)(1 - \zeta); \\ N_3 = \frac{1}{8}(1 + \xi)(1 + \eta)(1 - \zeta); \\ N_4 = \frac{1}{8}(1 - \xi)(1 + \eta)(1 - \zeta); \\ N_5 = \frac{1}{8}(1 - \xi)(1 - \eta)(1 + \zeta); \\ N_6 = \frac{1}{8}(1 + \xi)(1 - \eta)(1 + \zeta); \\ N_7 = \frac{1}{8}(1 + \xi)(1 + \eta)(1 + \zeta); \\ N_8 = \frac{1}{8}(1 - \xi)(1 + \eta)(1 + \zeta), \end{array} \right. \quad (\text{C.1})$$

where $\{\xi, \eta, \zeta\}$ is the natural coordinates attached to the cube element, see Figure C.4. For any point (x, y, z) inside the element with eight corner vertices $\{(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), (x_4, y_4, z_4), (x_5, y_5, z_5), (x_6, y_6, z_6), (x_7, y_7, z_7), (x_8, y_8, z_8)\}$, there holds

$$x = \sum_{i=1}^8 N_i x_i; \quad y = \sum_{i=1}^8 N_i y_i; \quad z = \sum_{i=1}^8 N_i z_i \quad \text{and} \quad \mathbf{u} = \sum_{i=1}^8 N_i \mathbf{U}_i. \quad (\text{C.2})$$

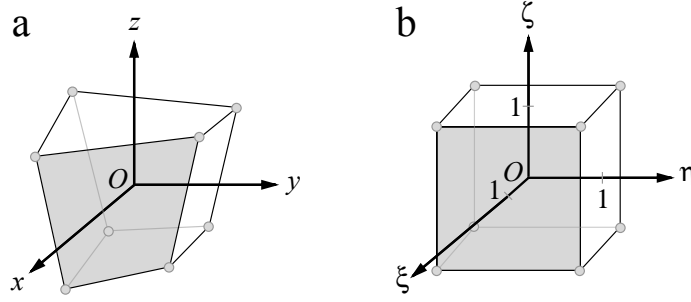


Figure C.4: Isoparametric mapping in Octant elements.

We still use the same symbol $\delta(\cdot)$ to mark whether midpoints due to refinement surface exist or not. The shape functions of 3D transition elements are

$$\left\{ \begin{array}{l}
 \tilde{N}_1 = N_1 - \frac{1}{2}(\hat{N}_{11} + \hat{N}_{12} + \hat{N}_{17}) + \frac{1}{4}(\tilde{N}_{21} + \tilde{N}_{23} + \tilde{N}_{26}); \\
 \tilde{N}_2 = N_2 - \frac{1}{2}(\hat{N}_9 + \hat{N}_{12} + \hat{N}_{18}) + \frac{1}{4}(\tilde{N}_{21} + \tilde{N}_{23} + \tilde{N}_{24}); \\
 \tilde{N}_3 = N_3 - \frac{1}{2}(\hat{N}_9 + \hat{N}_{10} + \hat{N}_{19}) + \frac{1}{4}(\tilde{N}_{21} + \tilde{N}_{24} + \tilde{N}_{25}); \\
 \tilde{N}_4 = N_4 - \frac{1}{2}(\hat{N}_{10} + \hat{N}_{11} + \hat{N}_{20}) + \frac{1}{4}(\tilde{N}_{21} + \tilde{N}_{25} + \tilde{N}_{26}); \\
 \tilde{N}_5 = N_5 - \frac{1}{2}(\hat{N}_{15} + \hat{N}_{16} + \hat{N}_{17}) + \frac{1}{4}(\tilde{N}_{22} + \tilde{N}_{23} + \tilde{N}_{26}); \\
 \tilde{N}_6 = N_6 - \frac{1}{2}(\hat{N}_{13} + \hat{N}_{16} + \hat{N}_{18}) + \frac{1}{4}(\tilde{N}_{22} + \tilde{N}_{23} + \tilde{N}_{24}); \\
 \tilde{N}_7 = N_7 - \frac{1}{2}(\hat{N}_{13} + \hat{N}_{14} + \hat{N}_{19}) + \frac{1}{4}(\tilde{N}_{22} + \tilde{N}_{24} + \tilde{N}_{25}); \\
 \tilde{N}_8 = N_8 - \frac{1}{2}(\hat{N}_{14} + \hat{N}_{15} + \hat{N}_{20}) + \frac{1}{4}(\tilde{N}_{22} + \tilde{N}_{25} + \tilde{N}_{26}); \\
 \tilde{N}_9 = \hat{N}_9 - \frac{1}{2}(\tilde{N}_{21} + \tilde{N}_{24}); \quad \tilde{N}_{15} = \hat{N}_{15} - \frac{1}{2}(\tilde{N}_{22} + \tilde{N}_{26}); \\
 \tilde{N}_{10} = \hat{N}_{10} - \frac{1}{2}(\tilde{N}_{21} + \tilde{N}_{25}); \quad \tilde{N}_{16} = \hat{N}_{16} - \frac{1}{2}(\tilde{N}_{22} + \tilde{N}_{23}); \\
 \tilde{N}_{11} = \hat{N}_{11} - \frac{1}{2}(\tilde{N}_{21} + \tilde{N}_{26}); \quad \tilde{N}_{17} = \hat{N}_{17} - \frac{1}{2}(\tilde{N}_{23} + \tilde{N}_{26}); \\
 \tilde{N}_{12} = \hat{N}_{12} - \frac{1}{2}(\tilde{N}_{21} + \tilde{N}_{23}); \quad \tilde{N}_{18} = \hat{N}_{18} - \frac{1}{2}(\tilde{N}_{23} + \tilde{N}_{24}); \\
 \tilde{N}_{13} = \hat{N}_{13} - \frac{1}{2}(\tilde{N}_{22} + \tilde{N}_{24}); \quad \tilde{N}_{19} = \hat{N}_{19} - \frac{1}{2}(\tilde{N}_{24} + \tilde{N}_{25}); \\
 \tilde{N}_{14} = \hat{N}_{14} - \frac{1}{2}(\tilde{N}_{22} + \tilde{N}_{25}); \quad \tilde{N}_{20} = \hat{N}_{20} - \frac{1}{2}(\tilde{N}_{25} + \tilde{N}_{26}); \\
 \tilde{N}_{21} = \frac{1}{2}(1 - \zeta)(1 - |\xi|)(1 - |\eta|)\delta(\text{Pt21}); \\
 \tilde{N}_{22} = \frac{1}{2}(1 + \zeta)(1 - |\xi|)(1 - |\eta|)\delta(\text{Pt22}); \\
 \tilde{N}_{23} = \frac{1}{2}(1 - \eta)(1 - |\xi|)(1 - |\zeta|)\delta(\text{Pt23}); \\
 \tilde{N}_{24} = \frac{1}{2}(1 + \xi)(1 - |\eta|)(1 - |\zeta|)\delta(\text{Pt24}); \\
 \tilde{N}_{25} = \frac{1}{2}(1 + \eta)(1 - |\xi|)(1 - |\zeta|)\delta(\text{Pt25}); \\
 \tilde{N}_{26} = \frac{1}{2}(1 - |\eta|)(1 - \xi)(1 - |\zeta|)\delta(\text{Pt26}),
 \end{array} \right. \quad (\text{C.3})$$

where $\hat{N}_9 \sim \hat{N}_{20}$ are intermediate functions defined as

$$\left\{ \begin{array}{ll} \hat{N}_9 = \frac{1}{4}(1 + \xi)(1 - |\eta|)(1 - \zeta)\delta(\text{Pt9}); & \hat{N}_{15} = \frac{1}{4}(1 - \xi)(1 - |\eta|)(1 + \zeta)\delta(\text{Pt15}); \\ \hat{N}_{10} = \frac{1}{4}(1 + \eta)(1 - |\xi|)(1 - \zeta)\delta(\text{Pt10}); & \hat{N}_{16} = \frac{1}{4}(1 - \eta)(1 - |\xi|)(1 + \zeta)\delta(\text{Pt16}); \\ \hat{N}_{11} = \frac{1}{4}(1 - \xi)(1 - |\eta|)(1 - \zeta)\delta(\text{Pt11}); & \hat{N}_{17} = \frac{1}{4}(1 - \eta)(1 - |\zeta|)(1 - \xi)\delta(\text{Pt17}); \\ \hat{N}_{12} = \frac{1}{4}(1 - \eta)(1 - |\xi|)(1 - \zeta)\delta(\text{Pt12}); & \hat{N}_{18} = \frac{1}{4}(1 - \eta)(1 - |\zeta|)(1 + \xi)\delta(\text{Pt18}); \\ \hat{N}_{13} = \frac{1}{4}(1 + \xi)(1 - |\eta|)(1 + \zeta)\delta(\text{Pt13}); & \hat{N}_{19} = \frac{1}{4}(1 + \eta)(1 - |\zeta|)(1 + \xi)\delta(\text{Pt19}); \\ \hat{N}_{14} = \frac{1}{4}(1 + \eta)(1 - |\xi|)(1 + \zeta)\delta(\text{Pt14}); & \hat{N}_{20} = \frac{1}{4}(1 + \eta)(1 - |\zeta|)(1 - \xi)\delta(\text{Pt20}). \end{array} \right. \quad (\text{C.4})$$

Displacement field is approximated using the linear combination of the above defined shape functions.

$$x = \sum_{i=1}^{26} \tilde{N}_i x_i; \quad y = \sum_{i=1}^{26} \tilde{N}_i y_i; \quad z = \sum_{i=1}^{26} \tilde{N}_i z_i \quad \text{and} \quad \mathbf{u} = \sum_{i=1}^{26} \tilde{N}_i \mathbf{U}_i \quad (\text{C.5})$$

$$\left\{ \begin{array}{lll} \frac{DN_1}{D\xi} = -\frac{1}{8}(1 - \eta)(1 - \xi); & \frac{DN_1}{D\eta} = -\frac{1}{8}(1 - \xi)(1 - \zeta); & \frac{DN_1}{D\zeta} = -\frac{1}{8}(1 - \xi)(1 - \eta); \\ \frac{DN_2}{D\xi} = \frac{1}{8}(1 - \eta)(1 - \zeta); & \frac{DN_2}{D\eta} = -\frac{1}{8}(1 + \xi)(1 - \zeta); & \frac{DN_2}{D\zeta} = -\frac{1}{8}(1 + \xi)(1 - \eta); \\ \frac{DN_3}{D\xi} = \frac{1}{8}(1 + \eta)(1 - \zeta); & \frac{DN_3}{D\eta} = \frac{1}{8}(1 + \xi)(1 - \zeta); & \frac{DN_3}{D\zeta} = -\frac{1}{8}(1 + \xi)(1 + \eta); \\ \frac{DN_4}{D\xi} = -\frac{1}{8}(1 + \eta)(1 - \zeta); & \frac{DN_4}{D\eta} = \frac{1}{8}(1 - \xi)(1 - \zeta); & \frac{DN_4}{D\zeta} = -\frac{1}{8}(1 - \xi)(1 + \eta); \\ \frac{DN_5}{D\xi} = -\frac{1}{8}(1 - \eta)(1 + \zeta); & \frac{DN_5}{D\eta} = -\frac{1}{8}(1 - \xi)(1 + \zeta); & \frac{DN_5}{D\zeta} = \frac{1}{8}(1 - \xi)(1 - \eta); \\ \frac{DN_6}{D\xi} = \frac{1}{8}(1 - \eta)(1 + \zeta); & \frac{DN_6}{D\eta} = -\frac{1}{8}(1 + \xi)(1 + \zeta); & \frac{DN_6}{D\zeta} = \frac{1}{8}(1 + \xi)(1 - \eta); \\ \frac{DN_7}{D\xi} = \frac{1}{8}(1 + \eta)(1 + \zeta); & \frac{DN_7}{D\eta} = \frac{1}{8}(1 + \xi)(1 + \zeta); & \frac{DN_7}{D\zeta} = \frac{1}{8}(1 + \xi)(1 + \eta); \\ \frac{DN_8}{D\xi} = -\frac{1}{8}(1 + \eta)(1 + \zeta); & \frac{DN_8}{D\eta} = -\frac{1}{8}(1 - \xi)(1 + \zeta); & \frac{DN_8}{D\zeta} = \frac{1}{8}(1 - \xi)(1 + \eta); \end{array} \right. \quad (\text{C.6})$$

$$\left\{ \begin{array}{l}
\frac{D\tilde{N}_1}{D\chi} = \frac{DN_1}{D\chi} - \frac{1}{2} \left(\frac{D\tilde{N}_{11}}{D\chi} + \frac{D\tilde{N}_{12}}{D\chi} + \frac{D\tilde{N}_{17}}{D\chi} \right) + \frac{1}{4} \left(\frac{D\tilde{N}_{21}}{D\chi} + \frac{D\tilde{N}_{23}}{D\chi} + \frac{D\tilde{N}_{26}}{D\chi} \right); \\
\frac{D\tilde{N}_2}{D\chi} = \frac{DN_2}{D\chi} - \frac{1}{2} \left(\frac{D\tilde{N}_9}{D\chi} + \frac{D\tilde{N}_{12}}{D\chi} + \frac{D\tilde{N}_{18}}{D\chi} \right) + \frac{1}{4} \left(\frac{D\tilde{N}_{21}}{D\chi} + \frac{D\tilde{N}_{23}}{D\chi} + \frac{D\tilde{N}_{24}}{D\chi} \right); \\
\frac{D\tilde{N}_3}{D\chi} = \frac{DN_3}{D\chi} - \frac{1}{2} \left(\frac{D\tilde{N}_9}{D\chi} + \frac{D\tilde{N}_{10}}{D\chi} + \frac{D\tilde{N}_{19}}{D\chi} \right) + \frac{1}{4} \left(\frac{D\tilde{N}_{21}}{D\chi} + \frac{D\tilde{N}_{24}}{D\chi} + \frac{D\tilde{N}_{25}}{D\chi} \right); \\
\frac{D\tilde{N}_4}{D\chi} = \frac{DN_4}{D\chi} - \frac{1}{2} \left(\frac{D\tilde{N}_{10}}{D\chi} + \frac{D\tilde{N}_{11}}{D\chi} + \frac{D\tilde{N}_{20}}{D\chi} \right) + \frac{1}{4} \left(\frac{D\tilde{N}_{21}}{D\chi} + \frac{D\tilde{N}_{25}}{D\chi} + \frac{D\tilde{N}_{26}}{D\chi} \right); \\
\frac{D\tilde{N}_5}{D\chi} = \frac{DN_5}{D\chi} - \frac{1}{2} \left(\frac{D\tilde{N}_{15}}{D\chi} + \frac{D\tilde{N}_{16}}{D\chi} + \frac{D\tilde{N}_{17}}{D\chi} \right) + \frac{1}{4} \left(\frac{D\tilde{N}_{22}}{D\chi} + \frac{D\tilde{N}_{23}}{D\chi} + \frac{D\tilde{N}_{26}}{D\chi} \right); \\
\frac{D\tilde{N}_6}{D\chi} = \frac{DN_6}{D\chi} - \frac{1}{2} \left(\frac{D\tilde{N}_{13}}{D\chi} + \frac{D\tilde{N}_{16}}{D\chi} + \frac{D\tilde{N}_{18}}{D\chi} \right) + \frac{1}{4} \left(\frac{D\tilde{N}_{22}}{D\chi} + \frac{D\tilde{N}_{23}}{D\chi} + \frac{D\tilde{N}_{24}}{D\chi} \right); \\
\frac{D\tilde{N}_7}{D\chi} = \frac{DN_7}{D\chi} - \frac{1}{2} \left(\frac{D\tilde{N}_{13}}{D\chi} + \frac{D\tilde{N}_{14}}{D\chi} + \frac{D\tilde{N}_{19}}{D\chi} \right) + \frac{1}{4} \left(\frac{D\tilde{N}_{22}}{D\chi} + \frac{D\tilde{N}_{24}}{D\chi} + \frac{D\tilde{N}_{25}}{D\chi} \right); \\
\frac{D\tilde{N}_8}{D\chi} = \frac{DN_8}{D\chi} - \frac{1}{2} \left(\frac{D\tilde{N}_{14}}{D\chi} + \frac{D\tilde{N}_{15}}{D\chi} + \frac{D\tilde{N}_{20}}{D\chi} \right) + \frac{1}{4} \left(\frac{D\tilde{N}_{22}}{D\chi} + \frac{D\tilde{N}_{25}}{D\chi} + \frac{D\tilde{N}_{26}}{D\chi} \right); \\
\frac{D\tilde{N}_9}{D\chi} = \frac{D\tilde{N}_9}{D\chi} - \frac{1}{2} \left(\frac{D\tilde{N}_{21}}{D\chi} + \frac{D\tilde{N}_{24}}{D\chi} \right); & \frac{D\tilde{N}_{15}}{D\chi} = \frac{D\tilde{N}_{15}}{D\chi} - \frac{1}{2} \left(\frac{D\tilde{N}_{22}}{D\chi} + \frac{D\tilde{N}_{26}}{D\chi} \right); \\
\frac{D\tilde{N}_{10}}{D\chi} = \frac{D\tilde{N}_{10}}{D\chi} - \frac{1}{2} \left(\frac{D\tilde{N}_{21}}{D\chi} + \frac{D\tilde{N}_{25}}{D\chi} \right); & \frac{D\tilde{N}_{16}}{D\chi} = \frac{D\tilde{N}_{16}}{D\chi} - \frac{1}{2} \left(\frac{D\tilde{N}_{22}}{D\chi} + \frac{D\tilde{N}_{23}}{D\chi} \right); \\
\frac{D\tilde{N}_{11}}{D\chi} = \frac{D\tilde{N}_{11}}{D\chi} - \frac{1}{2} \left(\frac{D\tilde{N}_{21}}{D\chi} + \frac{D\tilde{N}_{26}}{D\chi} \right); & \frac{D\tilde{N}_{17}}{D\chi} = \frac{D\tilde{N}_{17}}{D\chi} - \frac{1}{2} \left(\frac{D\tilde{N}_{23}}{D\chi} + \frac{D\tilde{N}_{26}}{D\chi} \right); \\
\frac{D\tilde{N}_{12}}{D\chi} = \frac{D\tilde{N}_{12}}{D\chi} - \frac{1}{2} \left(\frac{D\tilde{N}_{21}}{D\chi} + \frac{D\tilde{N}_{23}}{D\chi} \right); & \frac{D\tilde{N}_{18}}{D\chi} = \frac{D\tilde{N}_{18}}{D\chi} - \frac{1}{2} \left(\frac{D\tilde{N}_{23}}{D\chi} + \frac{D\tilde{N}_{24}}{D\chi} \right); \\
\frac{D\tilde{N}_{13}}{D\chi} = \frac{D\tilde{N}_{13}}{D\chi} - \frac{1}{2} \left(\frac{D\tilde{N}_{22}}{D\chi} + \frac{D\tilde{N}_{24}}{D\chi} \right); & \frac{D\tilde{N}_{19}}{D\chi} = \frac{D\tilde{N}_{19}}{D\chi} - \frac{1}{2} \left(\frac{D\tilde{N}_{24}}{D\chi} + \frac{D\tilde{N}_{25}}{D\chi} \right); \\
\frac{D\tilde{N}_{14}}{D\chi} = \frac{D\tilde{N}_{14}}{D\chi} - \frac{1}{2} \left(\frac{D\tilde{N}_{22}}{D\chi} + \frac{D\tilde{N}_{25}}{D\chi} \right); & \frac{D\tilde{N}_{20}}{D\chi} = \frac{D\tilde{N}_{20}}{D\chi} - \frac{1}{2} \left(\frac{D\tilde{N}_{25}}{D\chi} + \frac{D\tilde{N}_{26}}{D\chi} \right);
\end{array} \right.$$

where $\chi = \xi, \eta,$ and ζ .

$$\left\{ \begin{array}{l}
\frac{D\tilde{N}_{21}}{D\xi} = \frac{1}{2}(1 - |\eta|)(1 - \zeta)\text{sign}(-\xi)\delta(\text{Pt21}); & \frac{D\tilde{N}_{21}}{D\eta} = \frac{1}{2}(1 - |\xi|)(1 - \zeta)\text{sign}(-\eta)\delta(\text{Pt21}); \\
\frac{D\tilde{N}_{22}}{D\xi} = \frac{1}{2}(1 + \zeta)(1 - |\eta|)\text{sign}(-\xi)\delta(\text{Pt22}); & \frac{D\tilde{N}_{22}}{D\eta} = \frac{1}{2}(1 + \zeta)(1 - |\xi|)\text{sign}(-\eta)\delta(\text{Pt22}); \\
\frac{D\tilde{N}_{23}}{D\xi} = \frac{1}{2}(1 - \eta)(1 - |\zeta|)\text{sign}(-\xi)\delta(\text{Pt23}); & \frac{D\tilde{N}_{23}}{D\eta} = -\frac{1}{2}(1 - |\xi|)(1 - |\zeta|)\delta(\text{Pt23}); \\
\frac{D\tilde{N}_{24}}{D\xi} = \frac{1}{2}(1 - |\eta|)(1 - |\zeta|)\delta(\text{Pt24}); & \frac{D\tilde{N}_{24}}{D\eta} = \frac{1}{2}(1 + \xi)(1 - |\zeta|)\text{sign}(-\eta)\delta(\text{Pt24}); \\
\frac{D\tilde{N}_{25}}{D\xi} = \frac{1}{2}(1 + \eta)(1 - |\zeta|)\text{sign}(-\xi)\delta(\text{Pt25}); & \frac{D\tilde{N}_{25}}{D\eta} = \frac{1}{2}(1 - |\xi|)(1 - |\zeta|)\delta(\text{Pt25}); \\
\frac{D\tilde{N}_{26}}{D\xi} = -\frac{1}{2}(1 - |\eta|)(1 - |\zeta|)\delta(\text{Pt26}); & \frac{D\tilde{N}_{26}}{D\eta} = -\frac{1}{2}(1 - |\xi|)(1 - |\zeta|)\text{sign}(-\eta)\delta(\text{Pt26}); \\
\frac{D\tilde{N}_{21}}{D\zeta} = -\frac{1}{2}(1 - |\xi|)(1 - |\eta|)\delta(\text{Pt21}); \\
\frac{D\tilde{N}_{22}}{D\zeta} = \frac{1}{2}(1 - |\xi|)(1 - |\eta|)\delta(\text{Pt22}); \\
\frac{D\tilde{N}_{23}}{D\zeta} = \frac{1}{2}(1 - \eta)(1 - |\xi|)\text{sign}(-\zeta)\delta(\text{Pt23}); \\
\frac{D\tilde{N}_{24}}{D\zeta} = \frac{1}{2}(1 + \xi)(1 - |\eta|)\text{sign}(-\zeta)\delta(\text{Pt24}); \\
\frac{D\tilde{N}_{25}}{D\zeta} = \frac{1}{2}(1 + \eta)(1 - |\xi|)\text{sign}(-\zeta)\delta(\text{Pt25}); \\
\frac{D\tilde{N}_{26}}{D\zeta} = \frac{1}{2}(1 - |\eta|)(1 - \xi)\text{sign}(-\zeta)\delta(\text{Pt26});
\end{array} \right. \tag{C.7}$$

and derivatives of intermediate functions are computed as

$$\left\{ \begin{array}{ll}
 \frac{D\hat{N}_9}{D\xi} = \frac{1}{4}(1 - |\eta|)(1 - \zeta)\delta(\text{Pt9}) & \frac{D\hat{N}_9}{D\eta} = \frac{1}{4}(1 + \xi)(1 - \zeta)\text{sign}(-\eta)\delta(\text{Pt9}) \\
 \frac{D\hat{N}_{10}}{D\xi} = \frac{1}{4}(1 + \eta)(1 - \zeta)\text{sign}(-\xi)\delta(\text{Pt10}) & \frac{D\hat{N}_{10}}{D\eta} = \frac{1}{4}(1 - |\xi|)(1 - \zeta)\delta(\text{Pt10}) \\
 \frac{D\hat{N}_{11}}{D\xi} = -\frac{1}{4}(1 - |\eta|)(1 - \zeta)\delta(\text{Pt11}) & \frac{D\hat{N}_{11}}{D\eta} = \frac{1}{4}(1 - \xi)(1 - \zeta)\text{sign}(-\eta)\delta(\text{Pt11}) \\
 \frac{D\hat{N}_{12}}{D\xi} = \frac{1}{4}(1 - \eta)(1 - \zeta)\text{sign}(-\xi)\delta(\text{Pt12}) & \frac{D\hat{N}_{12}}{D\eta} = -\frac{1}{4}(1 - |\xi|)(1 - \zeta)\delta(\text{Pt12}) \\
 \frac{D\hat{N}_{13}}{D\xi} = \frac{1}{4}(1 - |\eta|)(1 + \zeta)\delta(\text{Pt13}) & \frac{D\hat{N}_{13}}{D\eta} = \frac{1}{4}(1 + \xi)(1 + \zeta)\text{sign}(-\eta)\delta(\text{Pt13}) \\
 \frac{D\hat{N}_{14}}{D\xi} = \frac{1}{4}(1 + \eta)(1 + \zeta)\text{sign}(-\xi)\delta(\text{Pt14}) & \frac{D\hat{N}_{14}}{D\eta} = \frac{1}{4}(1 - |\xi|)(1 + \zeta)\delta(\text{Pt14}) \\
 \frac{D\hat{N}_{15}}{D\xi} = -\frac{1}{4}(1 - |\eta|)(1 + \zeta)\delta(\text{Pt15}) & \frac{D\hat{N}_{15}}{D\eta} = \frac{1}{4}(1 - \xi)(1 + \zeta)\text{sign}(-\eta)\delta(\text{Pt15}) \\
 \frac{D\hat{N}_{16}}{D\xi} = \frac{1}{4}(1 - \eta)(1 + \zeta)\text{sign}(-\xi)\delta(\text{Pt16}) & \frac{D\hat{N}_{16}}{D\eta} = -\frac{1}{4}(1 - |\xi|)(1 + \zeta)\delta(\text{Pt16}) \\
 \frac{D\hat{N}_{17}}{D\xi} = -\frac{1}{4}(1 - \eta)(1 - |\zeta|)\delta(\text{Pt17}) & \frac{D\hat{N}_{17}}{D\eta} = -\frac{1}{4}(1 - |\eta|)(1 - \xi)\delta(\text{Pt17}) \\
 \frac{D\hat{N}_{18}}{D\xi} = \frac{1}{4}(1 - \eta)(1 - |\zeta|)\delta(\text{Pt18}) & \frac{D\hat{N}_{18}}{D\eta} = -\frac{1}{4}(1 - |\zeta|)(1 + \xi)\delta(\text{Pt18}) \\
 \frac{D\hat{N}_{19}}{D\xi} = \frac{1}{4}(1 + \eta)(1 - |\zeta|)\delta(\text{Pt19}) & \frac{D\hat{N}_{19}}{D\eta} = \frac{1}{4}(1 + \xi)(1 - |\zeta|)\delta(\text{Pt19}) \\
 \frac{D\hat{N}_{20}}{D\xi} = -\frac{1}{4}(1 + \eta)(1 - |\zeta|)\delta(\text{Pt20}) & \frac{D\hat{N}_{20}}{D\eta} = \frac{1}{4}(1 - \xi)(1 - |\zeta|)\delta(\text{Pt20}) \\
 \\
 \frac{D\hat{N}_9}{D\zeta} = -\frac{1}{4}(1 + \xi)(1 - |\eta|)\delta(\text{Pt9}) & \\
 \frac{D\hat{N}_{10}}{D\zeta} = -\frac{1}{4}(1 + \eta)(1 - |\xi|)\text{sign}(-\xi)\delta(\text{Pt10}) & \\
 \frac{D\hat{N}_{11}}{D\zeta} = -\frac{1}{4}(1 - \xi)(1 - |\eta|)\delta(\text{Pt11}) & \\
 \frac{D\hat{N}_{12}}{D\zeta} = -\frac{1}{4}(1 - \eta)(1 - |\xi|)\delta(\text{Pt12}) & \\
 \frac{D\hat{N}_{13}}{D\zeta} = \frac{1}{4}(1 + \xi)(1 - |\eta|)\delta(\text{Pt13}) & \\
 \frac{D\hat{N}_{14}}{D\zeta} = \frac{1}{4}(1 - |\xi|)(1 + \eta)\delta(\text{Pt14}) & \\
 \frac{D\hat{N}_{15}}{D\zeta} = \frac{1}{4}(1 - \xi)(1 - |\eta|)\delta(\text{Pt15}) & \\
 \frac{D\hat{N}_{16}}{D\zeta} = \frac{1}{4}(1 - \eta)(1 - |\xi|)\delta(\text{Pt16}) & \\
 \frac{D\hat{N}_{17}}{D\zeta} = \frac{1}{4}(1 - \eta)(1 - \xi)\text{sign}(-\zeta)\delta(\text{Pt17}) & \\
 \frac{D\hat{N}_{18}}{D\zeta} = \frac{1}{4}(1 - \eta)(1 + \xi)\text{sign}(-\zeta)\delta(\text{Pt18}) & \\
 \frac{D\hat{N}_{19}}{D\zeta} = \frac{1}{4}(1 + \eta)(1 + \xi)\text{sign}(-\zeta)\delta(\text{Pt19}) & \\
 \frac{D\hat{N}_{20}}{D\zeta} = \frac{1}{4}(1 + \eta)(1 - \xi)\text{sign}(-\zeta)\delta(\text{Pt20}) &
 \end{array} \right. \tag{C.8}$$

The Jacobian matrix of the 3D isoparametric mapping is defined as

$$J = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{bmatrix}. \tag{C.9}$$

The DN matrix in the FEM is finally assembled to be

$$DN = \begin{bmatrix} & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{inv}(J) & 0 & 0 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & & 0 & 0 & 0 \\ 0 & 0 & 0 & \text{inv}(J) & 0 & 0 & 0 \\ 0 & 0 & 0 & & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & \text{inv}(J) \\ 0 & 0 & 0 & 0 & 0 & 0 & \end{bmatrix} \begin{bmatrix} \frac{D\tilde{N}_1}{D\xi} & 0 & 0 & \frac{D\tilde{N}_2}{D\xi} & 0 & 0 & \dots & \dots & \frac{D\tilde{N}_{26}}{D\xi} & 0 & 0 \\ \frac{D\tilde{N}_1}{D\eta} & 0 & 0 & \frac{D\tilde{N}_2}{D\eta} & 0 & 0 & \dots & \dots & \frac{D\tilde{N}_{26}}{D\eta} & 0 & 0 \\ \frac{D\tilde{N}_1}{D\zeta} & 0 & 0 & \frac{D\tilde{N}_2}{D\zeta} & 0 & 0 & \dots & \dots & \frac{D\tilde{N}_{26}}{D\zeta} & 0 & 0 \\ 0 & \frac{D\tilde{N}_1}{D\xi} & 0 & 0 & \frac{D\tilde{N}_2}{D\xi} & 0 & \dots & \dots & 0 & \frac{D\tilde{N}_{26}}{D\xi} & 0 \\ 0 & \frac{D\tilde{N}_1}{D\eta} & 0 & 0 & \frac{D\tilde{N}_2}{D\eta} & 0 & \dots & \dots & 0 & \frac{D\tilde{N}_{26}}{D\eta} & 0 \\ 0 & \frac{D\tilde{N}_1}{D\zeta} & 0 & 0 & \frac{D\tilde{N}_2}{D\zeta} & 0 & \dots & \dots & 0 & \frac{D\tilde{N}_{26}}{D\zeta} & 0 \\ 0 & 0 & \frac{D\tilde{N}_1}{D\xi} & 0 & 0 & \frac{D\tilde{N}_2}{D\xi} & \dots & \dots & 0 & 0 & \frac{D\tilde{N}_{26}}{D\xi} \\ 0 & 0 & \frac{D\tilde{N}_1}{D\eta} & 0 & 0 & \frac{D\tilde{N}_2}{D\eta} & \dots & \dots & 0 & 0 & \frac{D\tilde{N}_{26}}{D\eta} \\ 0 & 0 & \frac{D\tilde{N}_1}{D\zeta} & 0 & 0 & \frac{D\tilde{N}_2}{D\zeta} & \dots & \dots & 0 & 0 & \frac{D\tilde{N}_{26}}{D\zeta} \end{bmatrix}. \quad (\text{C.10})$$

The H^1 -conformity of Octree mesh finite element spaces

Analogously, H^1 -conformity also holds for Octree meshes.

Lemma 6. (H^1 -conformity for Octree mesh). If \mathcal{K} is a conforming Octree mesh of a bounded polyhedral Lipschitz domain $\Omega \subset \mathbb{R}^d$, then $\mathbb{V}(\mathcal{K}) \subset H^1(\Omega)$.

Octree mesh refinement

To refine a marked 3D Octree element, we need to check whether it's divisible or not. If it's divisible, then we just need to connect the opposite edges' midpoints and cut the cubic element evenly into $2 \times 2 \times 2$ smaller elements, which is summarized in Algorithm 9.

Algorithm 9: 3D Octant cut divisible element Q into eight children elements.

function OCREF(Q)

Step 1, Find all the faces' midpoints of element Q .

Step 2, Connect the opposite faces' midpoints of element Q found in Step 1, and cut element Q evenly into eight children elements $\{Q_1, \dots, Q_8\}$.

Return $\{Q_1, Q_2, Q_3, Q_4, Q_5, Q_6, Q_7, Q_8\}$.

Similarly to 2D Quadtree mesh, we summarize 3D Octree mesh refinement algorithms in Algorithms 10-11.

Algorithm 10: Recursive refinement of a single Octant element $Q \in \mathcal{K}$

function REFINE_RECURSIVE_OC(\mathcal{K}, Q)

Do forever

 Get refinement patch $R(\mathcal{K}, Q)$;

 Access $Q' \in R(\mathcal{K}, Q)$ with $g(Q') = \min\{g(Q'') \mid Q'' \in R(\mathcal{K}, Q)\}$;

 If $g(Q') < g(Q)$ then

$\mathcal{K} := \text{REFINE_RECURSIVE_OC}(\mathcal{K}, Q')$;

 Else

 break;

 End if

end do

Get refinement patch $R(\mathcal{K}, Q)$;

For all $Q' \in R(\mathcal{K}, Q)$, do

 If $g(Q') = g(Q)$ then

$\{Q'_0, Q'_1, Q'_2, Q'_3, Q'_4, Q'_5, Q'_6, Q'_7\} = \text{OCREF}(Q')$;

$\mathcal{K} := \mathcal{K} \setminus \{Q'\} \cup \{Q'_0, Q'_1, Q'_2, Q'_3, Q'_4, Q'_5, Q'_6, Q'_7\}$;

 End if

End for

Return \mathcal{K} .

Algorithm 11: Recursive refinement of all the marked Octree elements \mathcal{M}

function REFINE_OC(\mathcal{K}, \mathcal{M})

For all $Q \in \mathcal{M} \cap \mathcal{K}$, do

$\mathcal{K} := \text{REFINE_RECURSIVE_OC}(\mathcal{K}, Q)$;

end

Return \mathcal{K} .

BIBLIOGRAPHY

- [1] F Hild and S Roux. “Digital image correlation: from displacement measurement to identification of elastic properties-a review”. In: *Strain* 42 (2006), pp. 69–80.
- [2] B Pan et al. “Two-dimensional digital image correlation for in-plane displacement and strain measurement: a review”. In: *Measurement science and technology* 20 (2009), p. 062001.
- [3] MA Sutton, JJ Orteu, and HW Schreier. *Image correlation for shape, motion and deformation measurements: basic concepts, theory and applications*. Springer-Verlag GmbH, 2009.
- [4] MA Sutton et al. “Determination of displacements using an improved digital correlation method”. In: *Image and vision computing* 1 (1983), pp. 133–139.
- [5] DJ Chen et al. “Digital speckle-displacement measurement using a complex spectrum method”. In: *Applied Optics* 32 (1993), pp. 1839–1849.
- [6] SK Dhir and JP Sikora. “An improved method for obtaining the general-displacement field from a holographic interferogram”. In: *Experimental Mechanics* 12 (1972), pp. 323–327.
- [7] T Kreis. “Holographic interferometry: principles and methods”. In: *Simulation and Experiment in Laser Metrology: Proceedings of the International Symposium on Laser Applications in Precision Measurements Held in Balatonfured/Hungary, June 3-6, 1996*. Vol. 2. John Wiley & Sons. 1996, p. 323.
- [8] PK Rastogi. “Principles of holographic interferometry and speckle metrology”. In: *Photomechanics*. Springer, 2000, pp. 103–151.
- [9] AS Dickinson et al. “Experimental validation of a finite element model of the proximal femur using digital image correlation and a composite bone model”. In: *Journal of Biomechanical Engineering* 133 (2011), p. 014504.
- [10] C Franck et al. “Three-dimensional full-field measurements of large deformations in soft materials using confocal microscopy and digital volume correlation”. In: *Experimental Mechanics* 47 (2007), pp. 427–438.
- [11] C Franck et al. “Three-dimensional traction force microscopy: a new tool for quantifying cell-matrix interactions”. In: *PloS one* 6 (2011), e17833.
- [12] C Rehrl et al. “A methodology to study crystal plasticity inside a compression test sample based on image correlation and EBSD”. In: *Materials Characterization* 62 (2011), pp. 793–800.
- [13] S Daly. “Deformation and fracture of thin sheets of nitinol”. PhD thesis. California Institute of Technology, 2007.

- [14] AF Bastawros, H Bart-Smith, and AG Evans. “Experimental analysis of deformation mechanisms in a closed-cell aluminum alloy foam”. In: *Journal of the Mechanics and Physics of Solids* 48 (2000), pp. 301–322.
- [15] M Jerabek, Z Major, and RW Lang. “Strain determination of polymeric materials using digital image correlation”. In: *Polymer Testing* 29 (2010), pp. 407–416.
- [16] Y Wang and AM Cuitiño. “Full-field measurements of heterogeneous deformation patterns on polymeric foams using digital image correlation”. In: *International Journal of Solids and Structures* 39 (2002), pp. 3777–3796.
- [17] J Zdunek et al. “Digital Image Correlation investigation of Portevin–Le Chatelier effect in an aluminium alloy”. In: *Materials Characterization* 59 (2008), pp. 1429–1433.
- [18] J Tracy, A Waas, and S Daly. “Experimental assessment of toughness in ceramic matrix composites using the J-integral with digital image correlation part I: methodology and validation”. In: *Journal of Materials Science* 50 (2015), pp. 4646–4658.
- [19] M Kimiecik, JW Jones, and S Daly. “Quantitative studies of microstructural phase transformation in Nickel-Titanium”. In: *Materials Letters* 95 (2013), pp. 25–29.
- [20] S Chang et al. “Nanoscale in-plane displacement evaluation by AFM scanning and digital image correlation processing”. In: *Nanotechnology* 16 (2005), p. 344.
- [21] V Rubino, N Lapusta, and A Rosakis. “Laboratory earthquake measurements with the high-speed Digital Image Correlation method and applications to super-shear transition”. In: *AGU Fall Meeting Abstracts*. Vol. 1. 2012, p. 06.
- [22] V Rubino et al. “Static laboratory earthquake measurements with the Digital Image Correlation method”. In: *Experimental Mechanics* (2014), pp. 1–18.
- [23] E Erten et al. “Glacier velocity monitoring by maximum likelihood texture tracking”. In: *IEEE Transactions on Geoscience and Remote Sensing* 47 (2009), pp. 394–405.
- [24] JP Avouac et al. “The 2005, Mw 7.6 Kashmir earthquake: Sub-pixel correlation of ASTER images and seismic waveforms analysis”. In: *Earth and Planetary Science Letters* 249 (2006), pp. 514–528.
- [25] G Besnard et al. “Analysis of image series through global digital image correlation”. In: *The Journal of Strain Analysis for Engineering Design* 47 (2012), pp. 214–228.
- [26] J Blaber, B Adair, and A Antoniou. “Ncorr: Open-source 2D digital image correlation Matlab software”. In: *Experimental Mechanics* (2015), pp. 1–18.

- [27] EMC Jones et al. “In situ measurements of strains in composite battery electrodes during electrochemical cycling”. In: *Experimental Mechanics* 54 (2014), pp. 971–985.
- [28] B Pan et al. “Comparison of subset-based local and finite element-based global digital image correlation”. In: *Experimental Mechanics* 55 (2015), pp. 887–901.
- [29] Correlated Solutions. “Vic-2D”. In: *Reference Manual* (2009).
- [30] B Pan et al. “Digital image correlation using iterative least squares and point-wise least squares for displacement field and strain field measurements”. In: *Optics and Lasers in Engineering* 47 (2009), pp. 865–874.
- [31] S Avril et al. “Comparison of two approaches for differentiating full-field data in solid mechanics”. In: *Measurement Science and Technology* 21 (2009), p. 015703.
- [32] JQ Zhao et al. “Improved Hermite finite element smoothing method for full-field strain measurement over arbitrary region of interest in digital image correlation”. In: *Optics and Lasers in Engineering* 50 (2012), pp. 1662–1671.
- [33] BK Horn and BG Schunck. “Determining optical flow”. In: *1981 Technical Symposium East*. International Society for Optics and Photonics. 1981, pp. 319–331.
- [34] J Modersitzki. *Numerical methods for image registration*. Oxford University Press on Demand, 2004.
- [35] A Ronovsk and A Vašatová. “Elastic image registration based on domain decomposition with mesh adaptation”. In: *Mathematical analysis and numerical mathematics* 15 (2017), pp. 322–330.
- [36] R Bouclier and JC Passieux. “A domain coupling method for finite element digital image correlation with mechanical regularization: Application to multiscale measurements and parallel computing”. In: *International Journal for Numerical Methods in Engineering* 111 (2017), pp. 123–143.
- [37] M Merta et al. “Parallel implementation of Total-FETI DDM with application to medical image registration”. In: *Domain Decomposition Methods in Science and Engineering XXI*. Springer, 2014, pp. 917–925.
- [38] JC Passieux, JN Perie, and M Salaun. “A dual domain decomposition method for finite element digital image correlation”. In: *International Journal for Numerical Methods in Engineering* 102 (2015), pp. 1670–1682.
- [39] TY Wang and KM Qian. “Parallel computing in experimental mechanics and optical measurement: A review (II)”. In: *Optics and Lasers in Engineering* (2017).

- [40] J Nocedal and S Wright. *Numerical optimization*. Springer Science & Business Media, 2006.
- [41] AR CONN, NIM GOULD, and PL TOINT. “A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds”. In: *SIAM Journal on Numerical Analysis* 28 (1991), pp. 545–572.
- [42] MV Afonso, JM Bioucas-Dias, and MAT Figueiredo. “An augmented Lagrangian approach to the constrained optimization Formulation of Imaging Inverse Problems”. In: *IEEE Transactions on Image Processing* 20 (2011), pp. 681–695.
- [43] JC SIMO and TA LAURSEN. “An augmented Lagrangian treatment of contact problems involving friction”. In: *Computers & Structures* 42 (1992), pp. 97–116.
- [44] JC Michel, H Moulinec, and P Suquet. “A computational method based on augmented Lagrangians and fast Fourier Transforms for composites with high contrast”. In: *CMES-Computer Modeling in Engineering & Sciences* 1 (2000), pp. 79–88.
- [45] T Goldstein et al. “Fast alternating direction optimization methods”. In: *SIAM Journal on Imaging Sciences* 7 (2014), pp. 1588–1623.
- [46] S Boyd et al. “Distributed optimization and statistical learning via the alternating direction method of multipliers”. In: *Machine Learning* 3 (2010), pp. 1–122.
- [47] JF Yang and Y Zhang. “Alternating direction algorithms for $l(1)$ -problems in compressive sensing”. In: *SIAM Journal on Scientific Computing* 33 (2011), pp. 250–278.
- [48] MV Afonso, JM Bioucas-Dias, and MAT Figueiredo. “Fast image recovery using variable splitting and constrained optimization”. In: *IEEE Transactions on Image Processing* 19 (2010), pp. 2345–2356.
- [49] R Glowinski and P Le Tallec. *Augmented Lagrangian and operator-splitting methods in nonlinear mechanics*. SIAM, 1989.
- [50] J Yang and K Bhattacharya. “Combining image compression with digital image correlation”. In: *Experimental Mechanics. Under review* (2018).
- [51] B Pan, H Xie, and Z Wang. “Equivalence of digital image correlation criteria for pattern matching”. In: *Applied optics* 49 (2010), pp. 5501–5509.
- [52] B Simon and M Iain. “Lucas-kanade 20 years on: A unifying framework”. In: *International journal of computer vision* 56 (2004), pp. 221–255.
- [53] J Réthoré, F Hild, and S Roux. “Shear-band capturing using a multiscale extended digital image correlation technique”. In: *Computer Methods in Applied Mechanics and Engineering* 196 (2007), pp. 5016–5030.

- [54] J Réthoré, F Hild, and S Roux. “Extended digital image correlation with crack shape optimization”. In: *International Journal for Numerical Methods in Engineering* 73 (2008), pp. 248–272.
- [55] R Fedele, L Galantucci, and A Ciani. “Global 2D digital image correlation for motion estimation in a finite element framework: a variational formulation and a regularized, pyramidal, multi-grid implementation”. In: *International Journal for Numerical Methods in Engineering* 96 (2013), pp. 739–762.
- [56] GE Christensen. *Deformable shape models for anatomy*. 1994.
- [57] M Burger, J Modersitzki, and L Ruthotto. “A hyperelastic regularization energy for image registration”. In: *SIAM Journal on Scientific Computing* 35 (2013), B132–B148.
- [58] B Fischer and J Modersitzki. “Combining landmark and intensity driven registrations”. In: *PAMM* 3 (2003), pp. 32–35.
- [59] M Sadati et al. “Smooth full field reconstruction of velocity and its gradients from noisy scattered velocimetry data in a cross-slot flow”. In: *Journal of Rheology* 55 (2011), pp. 353–377.
- [60] PL Reu et al. “DIC Challenge: Developing images and guidelines for evaluating accuracy and resolution of 2D analyses”. In: *Experimental Mechanics* (2017).
- [61] M Bornert et al. “Shortcut in DIC error assessment induced by image interpolation used for subpixel shifting”. In: *Optics and Lasers in Engineering* 91 (2017), pp. 124–133.
- [62] L Avellar and G Ravichandran. “Deformation and fracture of 3D printed heterogeneous materials”. In: *Society for Experimental Mechanics Annual Conference* (2016).
- [63] W. Hackbusch. *Multi-grid method and applications*. Springer Verlag, 2003.
- [64] K Sayood. *Introduction to data compression, 5th edition*. Morgan Kaufman Publishers, 2018.
- [65] C Christopoulos, A Skodras, and T Ebrahimi. “The JPEG2000 still image coding system: an overview”. In: *IEEE transactions on consumer electronics* 46 (2000), pp. 1103–1127.
- [66] GK Wallace. “The JPEG still picture compression standard”. In: *IEEE transactions on consumer electronics* 38 (1992), pp. xviii–xxxiv.
- [67] J Yang and K Bhattacharya. “Augmented Lagrangian digital image correlation”. In: *Experimental Mechanics. Under review* (2018).
- [68] JC Passieux et al. “Multiscale displacement field measurement using Digital Image Correlation: Application to the identification of elastic properties”. In: *Experimental Mechanics* 55 (2015), pp. 121–137.

- [69] E Tadmor, S Nezzar, and L Vese. “A multiscale image representation using hierarchical (BV, L²) decompositions”. In: *Multiscale Modeling & Simulation* 2 (2004), pp. 554–579.
- [70] Y Yuan et al. “A self-adaptive sampling digital image correlation algorithm for accurate displacement measurement”. In: *Optics and Lasers in Engineering* 65 (2015), pp. 57–63.
- [71] L Wittevrongel et al. “A Self Adaptive Global Digital Image Correlation Algorithm”. In: *Experimental Mechanics* 55 (2015), pp. 361–378.
- [72] B Pan et al. “Study on subset size selection in digital image correlation for speckle patterns”. In: *Optics express* 16.10 (2008), pp. 7037–7048.
- [73] Y Yuan et al. “Accurate displacement measurement via a self-adaptive digital image correlation method based on a weighted ZNSSD criterion”. In: *Optics and Lasers in Engineering* 52 (2014), pp. 75–85.
- [74] RH Nochetto, KG Siebert, and A Veese. “Theory of adaptive finite element methods: an introduction”. In: *Multiscale, nonlinear and adaptive approximation*. Springer, 2009, pp. 409–542.
- [75] AK Gupta. “A finite element for transition from a fine to a coarse grid”. In: *International Journal for Numerical Methods in Engineering* 12 (1978), pp. 35–45.
- [76] P Binev et al. “Approximation classes for adaptive methods”. In: *Serdica Mathematical Journal* 28 (2002), pp. 391–416.
- [77] A Saraswathibhatla and J Notbohm. “Applications of DIC in the Mechanics of Collective Cell Migration”. In: *International Digital Imaging Correlation Society*. 2017, pp. 51–53.
- [78] J Yang and K Bhattacharya. “Efficient FFT-based digital image correlation”. In: *To be submitted* (2018).
- [79] J Poissant and F Barthelat. “A novel subset splitting procedure for digital image correlation on discontinuous displacement fields”. In: *Experimental mechanics* 50 (2010), pp. 353–364.
- [80] RM Persiano, JLD Comba, and V Barbalho. “An adaptive triangulation refinement scheme and construction”. In: *Proceedings of the VI Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*. 1993, pp. 259–266.