Analysis of data identification and generation methods for data-driven approaches in solid mechanics

Nour Hachem

MÉMOIRE DE DOCTORAT DE

L'ÉCOLE CENTRALE DE NANTES

ÉCOLE DOCTORALE N° 602
Sciences de l'Ingénierie et des Systèmes
Spécialité : *Mécanique des Solides, des Matériaux, des structures et des surfaces*

Par
Nour HACHEM

Analysis of data identification and generation methods for data-driven approaches in solid mechanics

Thèse présentée et soutenue à l'École Centrale de Nantes, le 18 novembre 2025
Unité de recherche : Institut de Recherche en Génie Civil et Mécanique GeM, UMR 6183

Rapporteurs avant soutenance :

Ludovic CHAMOIN	Professeur des universités, ENS Paris-Saclay
Maya DE BUHAN	Ingénieure de recherche HDR, Safran Tech

Composition du Jury :

Président :	Grégoire ALLAIRE	Professeur des universités, École Polytechnique
Examinateurs :	Marc BONNET	Directeur de recherche, ENSTA Paris
	Robin BOUCLIER	Professeur des universités, INSA Toulouse
	Ludovic CHAMOIN	Professeur des universités, ENS Paris-Saclay
	Maya DE BUHAN	Ingénieure de recherche HDR, Safran Tech
	Laura DE LORENZIS	Full professor, ETH Zürich
Directeur de recherches doctorale :	Laurent STAINIER	Professeur des universités, École Centrale de Nantes
Co-enc. de recherches doctorale :	Adrien LEYGUE	Chargé de recherche, École Centrale de Nantes

Analysis of data identification and generation methods for data-driven approaches in solid mechanics

Nour Hachem

18 november 2025

"It is through logic that we prove, but through intuition that we discover."
-Henri Poincaré

Remerciements

Écrire ces dernières lignes constitue pour moi la partie la plus difficile de ce manuscrit, en ce qu'elles s'accompagnent d'un au revoir. Cela me fait penser à cette phrase : *"How lucky I am to have something that makes saying goodbye so hard"*. Ces mots me rappellent combien j'ai été chanceuse tout au long de cette thèse. Les lignes qui suivent alors s'adressent à l'ensemble de la communauté académique et personnelle qui m'a accompagnée.

Mes premiers remerciements vont naturellement à mes encadrants de thèse, Laurent Stainier et Adrien Leygue. Je leur suis reconnaissante pour leur gentillesse, leur patience et leur disponibilité, en particulier au début de la thèse, lorsqu'ils m'ont initiée à la mécanique. Je les remercie également pour la confiance qu'ils m'ont accordée, et qui a renforcé la mienne, pour leur écoute attentive ainsi que pour leur maturité scientifique. Ce fut un réel plaisir de travailler à leurs côtés tout au long de ces trois années. J'ai beaucoup appris, tant sur le plan scientifique que personnel, au fil de nos échanges et de vos conseils. Je vous remercie sincèrement pour l'accompagnement que vous m'avez offert et pour m'avoir permis de grandir et de me former en tant que chercheuse.

Je souhaite grandement remercier l'ensemble des membres du jury pour avoir accepté de participer à ma soutenance. La qualité scientifique de ce jury, tant en mathématiques qu'en mécanique, me rend fière. J'exprime ma profonde gratitude à chacun d'entre eux pour la rigueur scientifique de leurs évaluations et la richesse de leurs échanges. Je remercie chaleureusement Ludovic Chamoin et Maya De Buhan pour avoir accepté de rapporter cette thèse, pour la qualité de leur travail et la relecture attentive de mon manuscrit. Je remercie Grégoire Allaire d'avoir accepté de présidé ce jury, Marc Bonnet et Laura De Lorenzis pour leur rigueur scientifique, ainsi que Robin Bouclier, qui a également contribué aux CSI.

Je remercie également les gestionnaires du GeM, et tout particulièrement Sophie Schmidt, pour leur efficacité, ainsi que l'ensemble des personnes du laboratoire avec lesquelles j'ai eu l'occasion d'échanger au cours de ces années.

Bien évidement, je n'oublierai pas mes camarades doctorant.e.s et post-doctorant.e.s qui ont partagé cette aventure avec moi : Ghita, Lamia, Lilou, Saba, Annie, Lara, Loïc, Julien, Mathis, Vasu, Raphaël, Héloïse, Camille, Fabien, ... J'espère rester votre collègue préférée. Merci à tous les habitants

du bâtiment F et T, qui ont contribué à créer cette belle ambiance au laboratoire. Je n'oublierai pas les gâteaux du lundi, les présentations ESPRIT et science infuse. Merci pour tous ces souvenirs. Je remercie les amis que j'ai rencontrées à Nantes, et en particulier Mayssam et Walaa, pour leur écoute et leur soutien précieux. Je remercie également mes amis du Liban, et tout spécialement Soukayna, qui m'a soutenue malgré la distance.

Finalement, je souhaite remercier ma famille. Toute ma gratitude va à ma mère, mon père, ma sœur Nagham, ainsi qu'à mes frères Hassan et Hadi, pour leur soutien continu et leurs encouragements. Je leur dois ce chemin. شكرًا.

Contents

List of Figures

List of Tables

General introduction

Identifying the mechanical responses of materials is a fundamental challenge in solid mechanics. The constitutive equations close the boundary value problems of mechanics by establishing a relationship between kinematics (strain) and equilibrium (stress). This relationship can be identified through mechanical tests: simple tests or complex tests. Simple tests on the material allow for direct calculation of the stress field from a force measurement, since the resulting fields are homogeneous. These tests are limited to simple deformation modes and often require multiple tests to obtain sufficient data for identifying the stress field. In contrast, complex tests involving heterogeneous strain fields provide more information in a single test, although the stress field cannot be determined directly using traditional methods.

This leads to the formulation of an inverse problem: given experimental measurements (forces, displacements, strain fields), we want to reconstruct the underlying stress field or constitutive relation. This problem, however, is ill-posed in the sense of Hadamard, as it is sensitive to noise and admits multiple solutions. This ill-posedness arises because multiple constitutive relations could explain the same observed displacements, especially in the presence of noise or incomplete information.

Classically, the stress field is estimated using a constitutive model with parameters adjusted through identification methods such as the finite element model updating (FEMU) [54], the virtual fields method (VFM) [42, 51], the constitutive equation gap method (CEGM) [39, 71], and the modified error in constitutive relation (MECR) [19, 35], among others. These methods require the choice of *a priori* constitutive models, and their parameters are adjusted by minimizing a distance between experimental measurements and numerical results. The initial choice of the model strongly influences the results, potentially introducing bias.

Other identification methods have been proposed, such as the unsupervised discovery of interpretable hyperelastic constitutive laws (EUCLID) [38] that proposed the use of sparse regression to identify relevant components from a library of (potential) models. In addition, machine learning based methods

are being used to build surrogate models that learn material behavior directly from data [7, 11, 87]. While these methods offer greater flexibility, they remain parametric: they assume a functional form (model) and require the adjustment of their parameters. Although adding more parameters to a model has been shown to improve the accuracy of identification [72], selecting the right parameters to include remains a nontrivial and open problem.

At the same time, advances in experimental mechanics have opened new opportunities. Full-field measurement techniques [43], such as Digital Image Correlation (DIC) provide richer strain and displacement data. When combined with force measurements, these methods deliver a far more detailed description of material behavior than pointwise (simple) tests. This raises a fundamental question: can we move beyond predefined models and instead let the data speak directly?

Inspired by the Data-Driven Computational Mechanics (DDCM) method introduced by Kirchdoerfer and Ortiz [55], Data-Driven Identification (DDI) was introduced by Leygue *et al.* [61] as a model-free method that aims to identify the material mechanical response of samples without postulating a constitutive model. DDI solves the inverse problem relying on the principle that similar strain should yield similar stress, which is achieved by minimizing the distance between the mechanical state and its nearest representative in a database while ensuring equilibrium is satisfied.

DDI has been tested and applied to synthetic and real data, including studies on: small-strain elasticity [61], finite-strain elasticity [28, 30], viscoelasticity [90], and elastoplasticity [59, 91]. It has also been applied to the case of bi-material samples [90]. Despite these promising applications, a theoretical study of the method to validate the capacity of DDI to accurately estimate and converge to the true mechanical stress was still missing. Consequently, the parameters of the method have usually been selected empirically, which may affect its robustness. Their influence was evaluated using synthetic data as detailed in [31]. As these parameters are pushed to gain accuracy, the DDI problem becomes ill-posed.

The objective of this thesis is to advance the theoretical and practical development of the Data-Driven Identification (DDI) method. Recently, Leygue [60] introduced a new formulation that offers a criterion to determine the well-posedness of the problem and an error estimate for the identified stress field. In this work, we investigate numerically and algorithmically the preliminary work of Leygue [60] to deepen the mathematical understanding of DDI, with the aim of providing a more rigorous basis for its reliability and practical application.

To guide the reader, the manuscript is organized into five chapters, each addressing a specific aspect of the problem:

- **Chapter 1:** revisits the mathematical formulation of mechanical problems in continuum mechanics, introduces the concept of inverse problems, and reviews existing parametric and non-parametric identification methods. This chapter highlights the limitations of traditional approaches and motivates the introduction of DDI as a model-free alternative.

- **Chapitre 2:** presents the DDI method in detail. The formulation and solution strategy are described, and the method is illustrated with a simple example. A review of its evolution in the literature demonstrates its applicability to various classes of materials and highlights open challenges.

- **Chapter 3:** focuses on the well-posed case of the DDI problem, and presents an analysis and builds on the first mathematical result of Leygue (2025) [60], which introduced a criterion to assess well-posedness. This chapter presents three methods to evaluate that criterion and develops a new implementation of the DDI algorithm based on alternating projections. The effectiveness of these approaches is illustrated on synthetic data, and the convergence of the proposed algorithm is rigorously proven.

- **Chapter 4:** addresses the case of an ill-posed DDI problem and the following questions: What should be done in this situation, and how can the solution be corrected? We introduce three corrective methods: naïve elimination, database pruning, and regularization, and compare their performance, discussing their relative advantages and limitations.

- **Chapter 5:** translates the theoretical and numerical findings into practical guidance. It offers structured, step-by-step recommendations guidelines for implementing DDI.

Chapter 1

State of the art

Summary

1.1 Introduction

This chapter revisits the formulation of mechanical problems in continuum mechanics. It begins with the fundamental equations: kinematic relations and conservation laws, which are universal and independent of the material under consideration. However, these equations are not sufficient to fully define a mechanical problem or to describe the corresponding mechanical behavior. To close the problem, constitutive equations are introduced, which characterize the material response by establishing the relation between kinematics and equilibrium.
The focus of this work is on the cases where the constitutive relation is not known a priori. Such problems are referred to as inverse problems, where the objective is to deduce the material behavior from experimental data, typically derived from displacement, strain, or load cell measurements.
The chapter then reviews the main methodologies for addressing these problems, from parametric to non-parametric (model-free) approaches, highlighting their advantages, limitations, and scopes of applicability.

1.2 Notations

In this work, two types of notations are used: tensor notation and matrix notation. To avoid ambiguity, we adopt the following conventions throughout the manuscript:

- `Bold symbols` are used in tensor notation, e.g., displacement vector $\mathbf{u}$, stress tensor $\boldsymbol{\sigma}$, strain tensor $\boldsymbol{\varepsilon}$, ...
- `Plain symbols` are used in matrix notation, typically in the discretized finite element form, e.g., global displacement vector U, discrete gradient operator B, strain vector ε, ...

1.3 Mechanical problem

A mechanical problem, in the context of continuum mechanics, refers to the mathematical and physical formulation that characterizes the response of materials and structures to external loads, prescribed displacements, or environmental effects. In this section, we focus on the formulation of such problems. Throughout this work, all fields are assumed to be sufficiently regular. For analytical simplicity, we adopt the linearized kinematic measure, and the equilibrium equation is expressed in the reference configuration for the quasi-static case.
These equations are expressed in two formulations: a continuous representation involving partial differential equations (Sections 1.3.1 and 1.3.2), and a discrete

representation obtained via numerical discretization, resulting in a system of algebraic equations (Section 1.3.3).

This section is based on the references [16, 63, 82, 83].

1.3.1 Definition of a mechanical problem in continuum mechanics

Let $\Omega \subset \mathbb{R}^3$ denote a continuous solid body with boundary $\Gamma = \Gamma_D \sqcup \Gamma_N$[1], as shown in Figure 1.1. The solid body is subjected to a volume force $\mathbf{f}$ in Ω, a surface force $\mathbf{T}$ on Γ_N, and an imposed displacement $\mathbf{u}_D$ on $\Gamma_D = \Gamma \backslash \Gamma_N$. The deformation induced by external loads gives rise to stress and strain fields

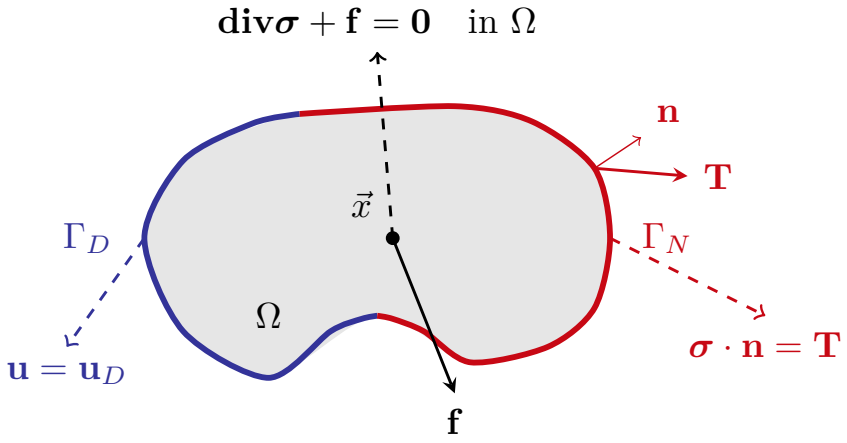


Figure 1.1: Schematic representation of a continuous solid body Ω with boundary conditions [83].

within the material. The objective of the mechanical problem is to determine the unknown field variables (displacement $\mathbf{u}$, strain $\boldsymbol{\varepsilon}$, and stress $\boldsymbol{\sigma}$) based on the prescribed boundary conditions, applied loads, and the material response. These unknown fields are governed by three types of equations: the equilibrium equations, which ensure force balance, the kinematic equations, which describe deformation, and the constitutive equations, which define material behavior and describe the relation between stress and strain.
The mechanical problem is then given by the following equations:

- Kinematic equations:
  - *Compatibility equation:*

$$\boldsymbol{\varepsilon} = \mathrm{sym}(\nabla \mathbf{u}) = \frac{1}{2}\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\top}\right), \qquad \text{in } \Omega\ . \tag{1.1}$$

[1]The symbol $\sqcup$ denotes the disjoint union, meaning that $\Gamma_D \cap \Gamma_N = \emptyset$.

  - *Dirichlet boundary condition:*

$$\mathbf{u} = \mathbf{u}_D, \qquad \text{on } \Gamma_D \ . \tag{1.2}$$

  A displacement field $\mathbf{u}$ satisfying these equations is called *kinematically admissible*.

- Equilibrium (conservation) laws:
  - *Balance of linear momentum (static equilibrium):*

$$\mathbf{div}(\boldsymbol{\sigma}) + \mathbf{f} = \mathbf{0}, \qquad \text{in } \Omega \ . \tag{1.3}$$

  - *Balance of angular momentum (symmetry of the stress tensor):*

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^\top, \qquad \text{in } \Omega \ . \tag{1.4}$$

  - *Neumann boundary condition:*

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{T}, \qquad \text{on } \Gamma_N \ . \tag{1.5}$$

  A stress field $\boldsymbol{\sigma}$ satisfying these equations is called *statically admissible*.

Here,

- $\mathbf{u}$ is the displacement vector field,
- $\mathbf{f}, \mathbf{T}$ are the external (body) force and surface traction vectors, respectively,
- $\boldsymbol{\varepsilon}, \boldsymbol{\sigma}$ are the strain and stress second-order tensors, respectively,
- $\nabla\mathbf{u}$ denotes the gradient of the displacement field, and it is a second-order tensor with components $\dfrac{\partial u_i}{\partial x_j}\mathbf{e}_i \otimes \mathbf{e}_j$,
- $\mathbf{div}(\mathbf{A})$ denotes the divergence of a second-order tensor $\mathbf{A}$ and yields a vector field with components $\dfrac{\partial A_{ij}}{\partial x_j}\mathbf{e}_i$,
- $\mathrm{sym}(\cdot)$ denotes the symmetric part of a tensor, defined as $\mathrm{sym}(\mathbf{A}) = \dfrac{1}{2}(\mathbf{A} + \mathbf{A}^\top)$.

For detailed definitions of tensors, vector calculus operators (e.g., gradient, divergence), and tensor notation, see [50].

All the equations written so far are *universal*, that is, they are independent of the material considered, provided that the same fundamental assumptions hold (e.g., the hypothesis of infinitesimal strain). These equations remain the same whether the body is composed of steel, aluminum, plastic, wood, concrete, or any other solid material. The mechanical behavior of each material is encoded in its constitutive model, which completes the mathematical description of the problem.

1.3.2 Constitutive model

Let us enumerate the number of unknowns on Ω, as well as the number of equations. The total number of unknowns is:

- three components of the displacement vector $\mathbf{u}$,
- six components of the strain tensor $\boldsymbol{\varepsilon}$,
- six components of the stress tensor $\boldsymbol{\sigma}$,

which results in a total of 15 scalar unknowns. For the number of equations on Ω, we have:

- six scalar equations arising from the compatibility equation (Eq. (1.1)) and considering the symmetry of the stress tensors derived from Eq. (1.4),
- three scalar equations corresponding to the equilibrium of forces (Eq. (1.3)),

yielding a total of 9 scalar equations. The equivalent of six scalar equations are missing to close the problem. These six equations are usually provided by the constitutive models.
The constitutive equations close the boundary value problems of mechanics by establishing a relationship between kinematics (strain) and equilibrium (stress). This relationship characterizes the material's behavior and is specific to each material. It may be influenced by various factors such as temperature, loading rate, aging, and other environmental or mechanical conditions. Some materials, like linear elastic material, have simple mechanical behavior that can be described with just a few parameters. However, many other materials (such as viscoelastic or plastic materials) show responses that depend not only on the current state but also on the history of strain. Note that, the history can be replaced by internal variables. The constitutive equations are typically expressed as a functional relation expressing the stress tensor as a function of the history of the strain tensor (i.e. on the value of the strain tensor at the current time as well as at all previous times).
These relations are complex in nature as they aim to describe internal material mechanisms and properties that are not directly measurable. Consequently, the identification of the constitutive equations cannot usually be derived theoretically and must instead be determined through experiments.
If this relation is known, then this problem is referred as the *direct* or *forward problem*. In this case, in the presence of a constitutive relation, the strong form of the static direct mechanical problem is expressed as:

The fields $\mathbf{u}(\mathbf{x})$, $\boldsymbol{\varepsilon}(\mathbf{x})$ and $\boldsymbol{\sigma}(\mathbf{x})$ satisfy:

$$\begin{cases} \boldsymbol{\varepsilon} = \frac{1}{2}\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^\top\right), & \text{in } \Omega \ , \\ \mathbf{u} = \mathbf{u}_D, & \text{on } \Gamma_D \ , \\ \mathbf{div}\boldsymbol{\sigma} + \mathbf{f} = \mathbf{0}, & \text{in } \Omega \ , \\ \boldsymbol{\sigma} = \boldsymbol{\sigma}^\top, & \text{in } \Omega \ , \\ \boldsymbol{\sigma}\cdot\mathbf{n} = \mathbf{T}, & \text{on } \Gamma_N \ , \\ +\text{constitutive relation.} \end{cases} \tag{1.6}$$

Problem (1.6) usually does not admit a simple analytical solution and approximate solutions have to be sought.

1.3.3 Discrete form of the mechanical problem

To obtain an approximate solution to the boundary value problem (1.6), a spatial discretization approach is often employed. In this work, we use the Finite Element Method (FEM), which discretizes the domain into a finite set of elements and reformulates the continuous problem as a system of algebraic equations that can be solved numerically.

The starting point is the weak (variational) form of the equilibrium equation. Let $\mathbf{u}^*$ be a test function such that $\mathbf{u}^* = 0$ on Γ_D. Multiplying the equilibrium equation by the virtual displacement $\mathbf{u}^*$, and integrating over the domain Ω yields:

$$\int_\Omega \mathbf{div}\,\boldsymbol{\sigma}\cdot\mathbf{u}^* d\Omega + \int_\Omega \mathbf{f}\cdot\mathbf{u}^* d\Omega = \mathbf{0}, \qquad \forall\mathbf{u}^* \ .$$

By applying the divergence theorem (Green-Ostrogradsky theorem) [53] and enforcing the boundary conditions, the weak form becomes:

$$-\int_\Omega \boldsymbol{\sigma} : \boldsymbol{\varepsilon}^* d\Omega + \int_\Omega \mathbf{f}\cdot\mathbf{u}^* d\Omega + \int_{\Gamma_N} \mathbf{T}\cdot\mathbf{u}^* d\Gamma = \mathbf{0}, \qquad \forall\mathbf{u}^* \ ,$$

where $\boldsymbol{\varepsilon}^* = \mathrm{sym}(\nabla\mathbf{u}^*)$ is the symmetric gradient of the test function.
Let Ω be discretized using a finite element mesh comprising m quadrature points and n nodes. To evaluate the integrals, numerical quadrature is applied, which approximates the integral as a weighted sum over the quadrature points, expressed as

$$\int_\Omega f(x) d\Omega \approx \sum_{e=1}^{m} f(x_e) w_e^{\mathrm{FE}} \ ,$$

where w_e^{FE} and x_e denote the quadrature weight and point, respectively.
The continuous fields are approximated using a linear combination of basis

(shape) functions. In particular, the displacement field is approximated as

$$\mathbf{u}(\mathbf{x}) \approx \sum_{k=1}^{n} \phi_k(\mathbf{x})\mathbf{u}_k \ , \qquad \forall \mathbf{x} \ , \tag{1.7}$$

where $\mathbf{u}_k$ denotes the displacement at node k and ϕ_k is the k-th shape function. The virtual displacement $\mathbf{u}^*$ is approximated using the same set of shape functions, which corresponds to the Galerkin method.
Substituting Eq. (1.7) in Eq. (1.1), the strain field is approximated as:

$$\boldsymbol{\varepsilon}(\mathbf{x}) \approx \frac{1}{2} \sum_{k=1}^{n} [\nabla \phi_k(\mathbf{x}) + \nabla \phi_k^{\top}(\mathbf{x})]\mathbf{u}_k \ , \qquad \forall \mathbf{x} \ . \tag{1.8}$$

Using Mandel notation[2] [65, 66], this relation is expressed in matrix form at the quadrature points as:

$$\varepsilon = BU \ , \tag{1.9}$$

where $B \in \mathbb{R}^{\ell m \times dn}$ is the discrete gradient operator, $U \in \mathbb{R}^{dn}$ is the global nodal displacement vector, and $\varepsilon \in \mathbb{R}^{\ell m}$ is the strain vector. The number of components ℓ depends on the spatial dimension d, with $\ell = 3$ for two dimensions $d = 2$ and $\ell = 6$ for three dimensions $d = 3$.

Based on this approximation and the discretized fields, the discrete equilibrium equation is expressed in matrix form as:

$$D\sigma = F \ , \tag{1.10}$$

where $D = B^{\top} W^{\mathrm{FE}} \in \mathbb{R}^{dn \times \ell m}$ is the discrete divergence operator, with W^{FE} a diagonal matrix containing the quadrature weights, $\sigma \in \mathbb{R}^{\ell m}$ is the stress vector assembled using Mandel notation and $F \in \mathbb{R}^{dn}$ represents the discrete external force.

For a detailed discussion of the mathematical formulation of FEM, the reader is referred to [1, 9, 26].

1.4 Inverse problem

The *inverse problem* consists of identifying the causes based on the observed effects. In the context of solid mechanics, the observable effects correspond to displacement and/or force fields that can be experimentally measured, for instance using digital image correlation for displacements or load cells for boundary forces. The hidden causes are the strain and stress fields inside the body, which govern the mechanical behavior but are not directly accessible to measurement.

[2]Mandel notation is described in detail in Chapter 5.

Notably, the strain fields can be computed from the measured displacement fields through the compatibility equations. In this way, strain can be regarded as known quantity. The central difficulty then shifts to the identification of the stress field, which cannot be measured directly and must instead be inferred from measured data that are often incomplete and affected by experimental noise.
Based on this, the inverse problem can be formulated as follows:

Given $\mathbf{f}$, $\mathbf{T}$, $\mathbf{u}$ and $\boldsymbol{\varepsilon}$, find the stress field $\boldsymbol{\sigma}$ such that:

$$\begin{cases} \mathbf{div}\,\boldsymbol{\sigma} + \mathbf{f} = \mathbf{0}, & \text{in } \Omega\ , \\ \boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{T}, & \text{on } \Gamma_N\ , \\ \text{there exists a constitutive relation between } \boldsymbol{\sigma} \text{ and } \boldsymbol{\varepsilon}. \end{cases} \tag{1.11}$$

Remark. *In this formulation, the existence of a constitutive relation between the stress field $\boldsymbol{\sigma}$ and the strain field $\boldsymbol{\varepsilon}$ is postulated, but its explicit form is not known.*

This problem is underdetermined meaning that it has more unknowns than equations. That leads to the existence of multiple possible solutions, in other words, it is an ill-posed problem in the sense of Hadamard. According to Hadamard's definition [46], a well-posed problem must satisfy the following conditions:

1. a solution exists,
2. the solution is unique,
3. the solution depends continuously on the input data.

The third condition is commonly referred to as stability. It requires that small perturbations in the input data (e.g., due to measurement noise or discretization errors) lead only to small perturbations in the solution. When stability is not satisfied, the solution becomes highly sensitive to errors, and the problem cannot be solved in a physically or numerically meaningful way. If any of these three conditions fail, the problem is considered ill-posed.
To ensure uniqueness and stability of Problem 1.11, it is necessary to incorporate additional information, typically in the form of regularization. These regularization methods will be discussed in the next section.
The key differences between the direct and inverse problems are summarized in Table 1.1.

Relevant literature includes references on inverse problems [17, 23, 49], inverse problems in mechanics [18, 32, 86], and regularization methods [12, 24].

[3] $\hat{\sigma}(\cdot)$ denotes the constitutive model.

Problem	Known	Unknowns	Direction	
Direct	$\mathbf{f},\ \mathbf{T},\ \hat{\sigma}(\cdot)$ [3]	$\mathbf{u},\ \boldsymbol{\varepsilon},\ \boldsymbol{\sigma}$	cause $\longrightarrow$ effect	Well-posed
Inverse	$\mathbf{f},\ \mathbf{T},\ \mathbf{u},\ \boldsymbol{\varepsilon}$	$\hat{\sigma}(\cdot),\ \boldsymbol{\sigma}$	effect $\longrightarrow$ cause	Ill-posed

Table 1.1: Comparison between direct and inverse problems.

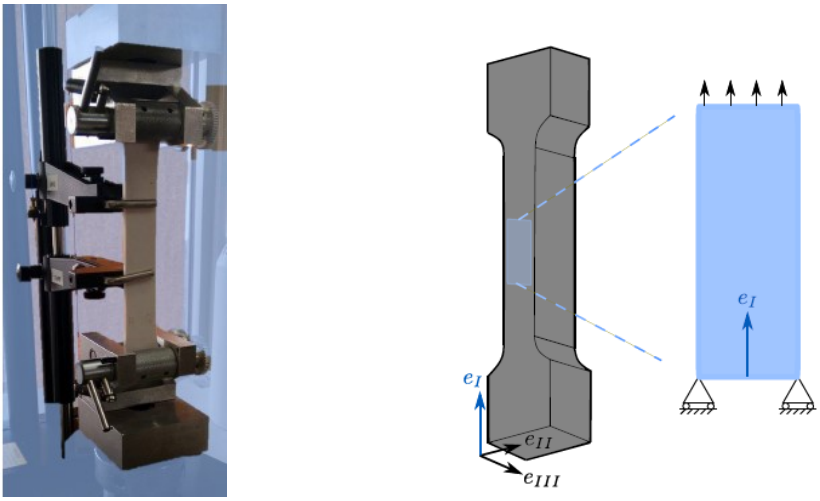
1.5 Behavior identification

In this section, we review the existing methods for identifying the mechanical behavior of materials, starting with the simplest (using homogeneous isostatic tests) and gradually progressing to more advanced ones. Their limitations will be presented progressively, and a summary table will be provided at the end of this section.

1.5.1 Identification from a homogeneous isostatic tests

The first way to solve this a priori ill-posed problem (Eq. (1.11)) consists in using a specimen with a geometry such that the problem becomes well-posed, that is, the matrix D in Eq. (1.10) is square and invertible. We refer to this method as *isostatic regularization*. This is the case for *isostatic homogeneous tests*, also known as *simple tests* or *statically determinate tests*.
This strategy relies mainly on the assumption that the stress and strain fields remain homogeneous in the region of interest (the gauge length) during the entire test.
In statically determinate tests, since the fields are uniform (homogeneous) in the gauge length, quantities such as loads, displacements, and temperatures can be directly obtained using pointwise measurement techniques such as load cells, extensometers, strain gauges and thermocouples.
Thanks to the geometric simplicity and homogeneous loading, the stress field (which is not measurable) can be determined directly from the static equilibrium equations, while the strain field can be obtained from displacement data.
Classic examples of such tests include uniaxial tension, compression, and bending tests (see [48]). Let us consider the uniaxial tension test, as illustrated in Figure 1.2b. The specimen has a dog-bone geometry and is subjected to an axial tensile force F along the axial direction e_I. The axial displacement in the gauge section can be measured directly using, for example, an extensometer (Figure 1.2a), which measures the variation in length during loading.
The gauge section of the specimen is designed to ensure a homogeneous state of stress and strain. Under this assumption:

- The axial strain $\boldsymbol{\varepsilon}_I$ is constant throughout the gauge length and can be



(a) Extensometer. (b) Uniaxial tension test.

Figure 1.2: Example of a simple test [29].

computed directly from the measured axial elongation ΔL as:

$$\boldsymbol{\varepsilon}_I = \frac{\Delta L}{L_0} \ ,$$

where L_0 is the initial gauge length.

- The corresponding stress tensor is also homogeneous and can be determined from the resultant force F, measured using a load cell.
  The resulting stress tensor in the orthonormal basis is given by:

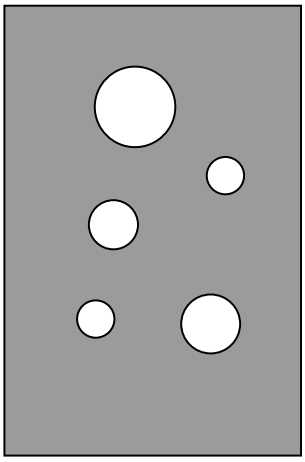
$$\boldsymbol{\sigma} = \begin{pmatrix} F/S & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}_{(e_I, e_{II}, e_{III})}$$

  where S denotes the current cross-sectional area of the gauge section during the test.

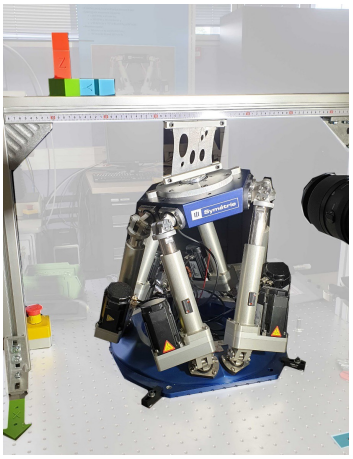
From such tests, it is sometimes possible to adjust a constitutive model, which may take various forms, such as a scatter plot, an explicit relation, or another representation. However, these tests only give a scalar measurement, as they rely on the assumption of field homogeneity. As a result, they are limited by the lack of representative data that fully capturing the material's physical behavior. This, in turn, necessitates performing a large number of experimental tests to ensure an accurate modeling of the material response. Additionally, isostatic tests may fail to capture complex phenomena, such as those occurring under high strain rates, material anisotropy, heterogeneous materials, or local effects.
More advanced experimental techniques are often used to identify and characterize more complex aspects of material behavior.

1.5.2 Identification from full-field measurement

1.5.2.1 Statically undetermined tests and full-field measurement



(a) Sample with different holes.



(b) Experimental test using Hexapod [27].

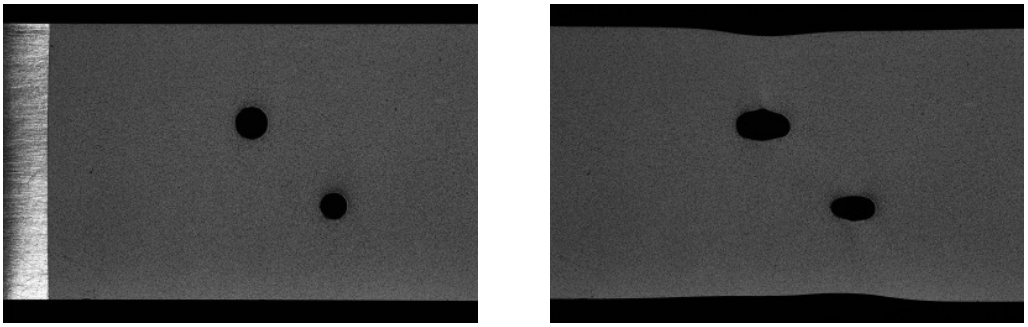
Figure 1.3: Examples of a complex test.

The *statically undetermined tests*, also called *complex tests* are defined as non-simple tests. In such cases, the problem remain ill-posed and the matrix D (Eq. 1.10) is not invertible. Their complexity can arise from various factors, including:

- complex geometry, such as samples with holes as shown in Figure 1.3a,
- complex loading, for instance, using a Hexapod (also known as a Stewart platform), which allows motion along six degrees of freedom (three translations and three rotations, see Figure 1.3b) [29].
- A combination of complex geometry and loading, as illustrated in Figure 1.3b.

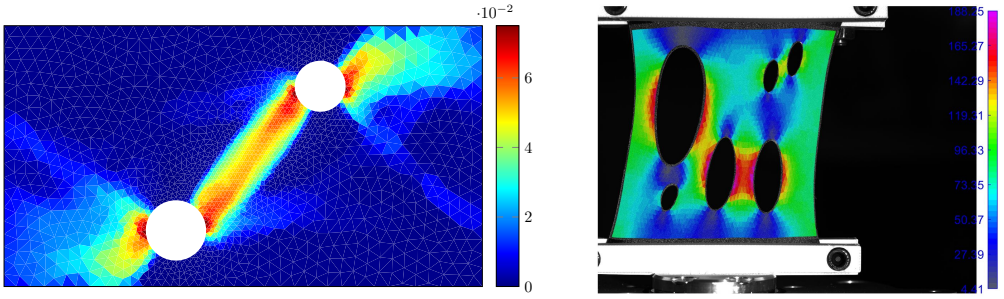
As a result, the strain and stress fields are heterogeneous.

In this test, full-field measurements [43] are used to determine the kinematic fields throughout the experiment. A commonly used technique is Digital Image Correlation (DIC) [69, 75], in which a speckle pattern is applied to the surface of the sample (see Figure 1.4). During the experiment, a series of images is captured, and the displacement of surface points is tracked by comparing these images. Using a mesh, often derived from a finite element grid, the displacement fields are then reconstructed from the observed point movements at various loading stages. Unlike pointwise measurements, the displacement field in this case is represented as a vector field, capturing the spatially heterogeneous nature of the deformation.



(a) Image of the reference sample. (b) Image of the deformed sample.

Figure 1.4: Example of configurations of the same object at two different snapshot [58].



(a) Example from Langlois thesis [58]. (b) Example from Costecalde thesis [27].

Figure 1.5: Example of the strain field using DIC.

Figure 1.5 presents DIC results that highlight two different types of complexity. In Figure 1.5a, the sample contains two holes and is subjected to a uniaxial tension test. In contrast, Figure 1.5b shows a sample with multiple holes, loaded using a Hexapod. In both cases, these complex tests produce heterogeneous strain fields, offering a richer and more detailed range of strain information. However, this advantage comes with a cost. For such tests, the stress field cannot be directly determined from the static equilibrium equations, as is possible in simpler, statically determinate tests.
The differences between simple and complex tests are summarized in Table 1.2.
In the following, various methodologies for identifying material behavior from

Test type	Data measurement	Fields	Identification
Simple	Pointwise	Homogeneous	Isostatic regularization
Complex	Full-field	Heterogeneous	Parametric & non-parametric methods

Table 1.2: Comparison between simple and complex tests.

these complex experiments will be presented.

1.5.2.2 Parametric methods

Parametric methods regularize the problem by postulating a (parametric) model *a priori* and adjusting its parameters to fit the measured data. In general, parametric approaches rely on two main ingredients:

1. **Choice of model**, that can be grouped into three main classes:
    - *Classical methods:* a constitutive model is proposed and its parameters is adjusted.
    - *Sparse regression based method:* a library of constitutive model is proposed and the sparse regression is used to identify the dominant terms and their associated parameters.
    - *Neural network approaches:* the proposed model is a surrogate, typically a neural network, that learns the material behavior directly from data.
2. **Choice of loss function**, which may be defined in terms of a displacement misfit, an equilibrium (force) misfit, or by using the *Constitutive Relation Error* (CRE) [56], which measures the error on the material behavior (stress).

A brief overview of these methods is provided in this section. We begin with the classical methods class, where three loss functions are detailed: (i) the

finite element model updating, (ii) the modified error in constitutive relation, and (iii) the virtual fields method. The main differences between them are the objective function used to find the material parameters. For global overview of this class of methods, the interested reader is referred to Avril *et al.* [8] and Martins *et al.* [67].
The two other model classes are then presented. Importantly, the choice of model and the choice of loss function are independent: the loss functions first applied to classical methods can also be extended and adapted to the other model classes.
The limitations of each class will be discussed following their respective presentations.

Finite Element Model Updating (FEMU)

The *Finite Element Model Updating* (FEMU) [54, 74] is a classical inverse identification technique used to estimate material parameters of the a priori assumed constitutive model by comparing numerical predictions to experimental measurements.
This approach requires the construction of a finite element model of the mechanical test in order to generate numerical predictions of the response. By comparing the numerically simulated fields with the corresponding experimental data, and employing an optimization algorithm, the material parameters are iteratively adjusted until convergence is achieved, i.e., when the simulated mechanical quantities best approximate the measured data.
Two main variants of FEMU exist, depending on the experimental data and the loss function used:
Given the constitutive model $\hat{\sigma}(\varepsilon, p)$ assumed a priori, where p is the set of unknown material parameters that must be identified.

- FEMU-U (displacement method) minimizes the residual between the measured displacement field U_M and the simulated displacement $U_S(p)$, defined as:
$$p^* = \arg\min_p \|U_M - U_S(p)\|_2^2 \ . \tag{1.12}$$
- FEMU-F (force balance method) minimizes the residual between the measured force vector F_M and the simulated force vector obtained from the stress field associated with the measured strain ε_M. The identification problem becomes:
$$p^* = \arg\min_p \|F_M - B^\top W \hat{\sigma}(\varepsilon_M, p)\|_2^2 \ . \tag{1.13}$$

Other variations of this method exist. For example, one can combine the displacement and force-based methods and perform FEMU-U-F [40]. Another example is FEMU-TU, which incorporates thermal data along with displacement measurements to identify thermomechanical properties [5].

Modified Error in Constitutive Relation (MECR)

The *Modified Error in Constitutive Relation* (MECR or mCRE) [19, 35, 57] is based on the minimization of the Constitutive Relation Error (CRE) [56], augmented by a data misfit term. The CRE quantifies the distance between two stress fields: one computed from the measured displacement field via the constitutive model, and another statically admissible stress field that satisfies equilibrium. The mCRE functional is defined as

$$\mathcal{F}(\mathbf{u}, \boldsymbol{\sigma}; \mathbf{p}) = \int_\Omega \left[\boldsymbol{\psi}(\boldsymbol{\varepsilon}(\mathbf{u}); \mathbf{p}) + \boldsymbol{\psi}^*(\boldsymbol{\sigma}; \mathbf{p}) - \boldsymbol{\sigma} : \boldsymbol{\varepsilon}(\mathbf{u})\right] d\Omega + \alpha \left\|\Pi\mathbf{u} - \mathbf{u}_M\right\|^2, \tag{1.14}$$

where $\mathbf{p}$ denotes the constitutive parameters, α is a penalty scaling factor, Π is a projector, and $\boldsymbol{\psi}$ and $\boldsymbol{\psi}^*$ are the strain-energy potential and its dual, respectively. The term $\alpha\|\Pi\mathbf{u} - \mathbf{u}_M\|^2$ enforces agreement with the measured displacement data $\mathbf{u}_M$. The minimization is carried out over kinematically and statically admissible pairs $(\mathbf{u}, \boldsymbol{\sigma})$. In the case of linear elasticity, we have

$$\boldsymbol{\psi}(\boldsymbol{\varepsilon}; \mathbf{p}) = \frac{1}{2}\,\boldsymbol{\varepsilon} : \mathbb{H}(\mathbf{p}) : \boldsymbol{\varepsilon}\ , \qquad \boldsymbol{\psi}^*(\boldsymbol{\sigma}; \mathbf{p}) = \frac{1}{2}\,\boldsymbol{\sigma} : \mathbb{H}^{-1}(\mathbf{p}) : \boldsymbol{\sigma}\ , \tag{1.15}$$

where $\mathbb{H}$ is the fourth-order elasticity tensor that depends on material parameters $\mathbf{p}$.

When the kinematic constraint is strongly enforced, the formulation reduces to the *Constitutive Equation Gap Method* (CEGM) [39, 71], which minimizes the CRE term only, without the data misfit penalty.

Virtual Fields Method (VFM)

The *Virtual Fields Method* (VFM) [42, 51, 79] is based on the principle of virtual work (variational formulation), where a set of carefully chosen virtual displacement fields (test functions) is used to transform the weak form of equilibrium into a system of equations that allows for the identification of unknown material parameters in an assumed constitutive model.
This principle is written as follows:

$$-\int_\Omega \boldsymbol{\sigma} : \boldsymbol{\varepsilon}^* + \int_\Omega \mathbf{f} \cdot \mathbf{u}^* + \int_{\Gamma_N} \mathbf{T} \cdot \mathbf{u}^* = \mathbf{0}\ ,$$

where $\mathbf{u}^*$ denotes a kinematically admissible virtual displacement field.
Each choice of virtual displacement field $\mathbf{u}^*$ generates one scalar equation. By selecting a sufficient number of independent virtual fields, the resulting system of equations can be solved for the unknown material parameters. The accuracy, robustness, and conditioning of this identification process depend strongly on the choice of virtual fields.

The classical methods presented above (FEMU, VFM, ...) require the choice of *a priori* constitutive models, and their parameters are adjusted by minimizing the distance between experimental measurements and numerical results. This initial choice of model has a strong influence on the results and may introduce significant bias.

Efficient Unsupervised Constitutive Law Identification and Discovery (EUCLID)

The *Efficient Unsupervised Constitutive Law Identification and Discovery* (EUCLID) [38] introduces a different approach compared to classical identification methods by postulating, a priori, a library of candidate (potential) models. It uses a sparse regression techniques (e.g., LASSO) to select the most relevant building blocks. The method simultaneously identifies both the underlying structure of the constitutive law and its associated material parameters, using full-field experimental data.

However, both classical methods and EUCLID (a sparse regression-based approach) still depend on predefined, parameterized models, requiring the adjustment of their parameters. Although adding more parameters to a model has been shown to improve the accuracy of identification [72], determining the right parameters to include remains a nontrivial and open problem.

Neural Network approaches (NN)

A very active line of research in material response identification involves machine learning-based approaches [7, 11, 15, 87]. These approaches replace the constitutive model by a phenomenological surrogate model, typically a neural network trained to approximate the underlying material response. These methods remain fundamentally parametric, but they differ from classical ones by some aspects: in particular, instead of identifying a small set of material parameters, they optimize a high-dimensional parameter space. This surrogate model can adapt to the data while remaining compatible with the principles of continuum mechanics.
These models are designed to respect physical principles (such as equilibrium, compatibility, and thermodynamic consistency) either through the architecture itself (e.g., TANN [68]) or via physics-informed loss functions. Such loss functions can be chosen from those detailed earlier for classical methods, as in NN-mCRE [11] and PiNN [80], or from sparse regression based methods, as in NN-EUCLID [87].

However, the large number of parameters increases computational cost. The resulting optimization problems can be ill-posed or overparameterized, leading to sensitivity to noise and a risk of overfitting.

1.5.2.3 Non-parametric methods

To address the limitations of parametric methods, alternative identification methods have been proposed that aim to estimate the stress fields without relying on a predefined model. These are referred to as *non-parametric methods*. Below, we briefly describe the non-parametric approaches that have been proposed in the literature.

- The *alignment method* introduces assumptions on the behavior, as it is based on the assumption that the principal directions (eigenvectors) of stress and strain are aligned. This method was introduced by Cameron and Tasan [22], and by Liu [62]. It is applicable only to isotropic materials, where this assumption holds true. However, its use is limited and not suitable for general use with anisotropic materials.

- The *dynamic equilibrium* method [2, 78], proposed for dynamic problems. It is based on the use of full-field acceleration data within the weak formulation, such as in the virtual fields method. This method is not general, and requires specific loading conditions, such as inertial impact tests, to compute the acceleration.

	Method(s)	Limitations
Parametric	Classical methods	Model bias, choice of number of parameters
	EUCLID	Depends on library design, choice of number of parameters
	NN approaches	Overfitting, high data requirement
Non-parametric	Alignment method	Only valid for isotropic materials
	Dynamic equilibrium	Not general, requires specific dynamic loading

Table 1.3: Comparison of parametric and non-parametric identification methods and their limitations.

A more general alternative method, called *Data-Driven Identification* (DDI) [60, 61], proposed by Leygue *et al.*, aims to identify the material mechanical response of samples without postulating a constitutive model from full-field kinematic measurements and load cell measurements.
The next chapter presents the DDI framework in detail, which forms the main focus of this work.

1.6 Conclusion

In this chapter, we revised the mathematical formulation of mechanical problem, and introduced the associated inverse problem that arise when attempting to identify internal fields or material behavior from measurements.
We examined the challenges posed by the ill-posed nature of inverse problems and discussed how behavior identification can be approached through either parametric or non-parametric methods based on full-field measurements obtained from complex mechanical tests. Parametric methods rely on predefined models and parameter fitting. While effective, they are limited by model bias and the difficulty of selecting an appropriate level of model complexity (e.g., the number of parameters). On the other hand, non-parametric methods aim to bypass constitutive assumptions entirely by directly estimating stress fields or mechanical behavior from experimental data. However, we noted that the two existing non-parametric approaches in the literature introduce additional assumptions that limits their applicability to certain material behaviors.
A model-free method for estimating the mechanical response of materials and heterogeneous stress fields, known as Data-Driven Identification (DDI), was proposed by Leygue and his co-authors [61]. This approach is inspired by Data-Driven Computational Mechanics (DDCM) [55], in which the material behavior is represented by a database of admissible strain-stress pairs that sample the mechanical response of the material.
The following chapter will present the DDI method in detail, along with its applications and developments over the years.

Chapter 2

Data-Driven Identification

Summary

2.1 Introduction

The Data-Driven Identification (DDI) method offers a framework for estimating the mechanical stress directly from experimental or numerical data, without the need to assume an explicit constitutive model a priori. Initially introduced by Leygue *et al.* (2018) [61], the method has since evolved through multiple extensions and applications, ranging from simple elastic cases to history-dependent and viscoelastic materials, and from homogeneous samples to heterogeneous samples (bi-materials).
This chapter presents the formulation and solution strategy of the DDI method, illustrates its application through a simple numerical example, and reviews its evolution over time, highlighting key contributions in the literature. Current challenges and open questions are discussed, providing motivation for the developments proposed in this work.

2.2 The Data-Driven Identification (DDI) Method

In this section, we present the DDI in a discretized form on an FE mesh with n nodes and m integration points. Instead of a constitutive model, DDI uses a database of size $N_{\mathcal{D}}$, to regularize the problem (1.11).

Index notations

The following index notation is used:

- $\bullet_e$ or $\bullet_a$ denotes a quantity evaluated at quadrature point, with $e, a \in \{1, \dots, m\}$.
- $\bullet_k$ denote a quantity at node k, with $k \in \{1, \dots, n\}$.
- $\bullet_i$ refers to a quantity associated with material state i, with $i \in \{1, \dots, N_{\mathcal{D}}\}$.
- Primed indices (e.g., e', a', k', i') are used as dummy summation index.

2.2.1 Definitions

The terms associated with the DDI method and used throughout this work are introduced below:

- The *constitutive space* is a multi-dimensional space in which the material behavior is approximated. In this work, we restrict ourselves to the elastic case, where the constitutive space is taken as $(\boldsymbol{\varepsilon}, \boldsymbol{\sigma})$. This space can be extended for inelastic problems by including additional variables, such as history-dependent variables.

- A *mechanical state* $(\boldsymbol{\varepsilon}_e, \boldsymbol{\sigma}_e)$ is an element in the constitutive space that represents the strain-stress state at a point in the sample, typically corresponding to an integration point.
- The *material database* $\mathcal{D}$ samples the material behavior into $N_{\mathcal{D}}$ representative strain-stress pairs.
- A *material state* $(\boldsymbol{\mathcal{E}}_i, \boldsymbol{\mathcal{S}}_i)$ is an element in the material database $\mathcal{D}$.
- The binary matrix pairing $P \in \{0,1\}^{m \times N_{\mathcal{D}}}$ connects each mechanical state to a material state.
- The weights w_e, associated with the m quadrature points.
- The $\mathbb{C}$-norm is introduced to define a distance between strain and stress tensors in the constitutive space and is given by

$$\|(\boldsymbol{\varepsilon}, \boldsymbol{\sigma})\|_{\mathbb{C}}^2 = \boldsymbol{\varepsilon} : \mathbb{C} : \boldsymbol{\varepsilon} + \boldsymbol{\sigma} : \mathbb{C}^{-1} : \boldsymbol{\sigma} \ . \tag{2.1}$$

  For more details regarding the norm see [55, 61].

The inputs, outputs, and parameters of the DDI problem are summarized in Table 2.1.

Inputs	Parameters	Outputs
Geometry, F, u, $\boldsymbol{\varepsilon}$	$\mathbb{C}$, w_e, $N_{\mathcal{D}}$	$\boldsymbol{\sigma}$, $\boldsymbol{\mathcal{E}}$, $\boldsymbol{\mathcal{S}}$, P

Table 2.1: Main ingredients of the DDI method.

In the following, the DDI method is presented in detail, and some of the definitions introduced here will be revisited and further clarified.

2.2.2 Formulation

The method relies on the following input data:

- the discrete divergence operator $\mathbf{D}_{ke}$, where the matrix $\mathbf{D}$ encodes the geometry of the samples,
- the external force vector $\mathbf{f}_k$,
- the strain at all quadrature points $\boldsymbol{\varepsilon}_e$, computed from the measured displacement field.

The static admissibility is therefore expressed as:

$$\sum_{e'=1}^{m} \mathbf{D}_{ke'} \cdot \boldsymbol{\sigma}_{e'} = \mathbf{f}_k \quad \forall k \in \{1, \dots, n\} \ . \tag{2.2}$$

The DDI method is based on the idea that similar strain fields should lead to similar stress fields while satisfying equilibrium. This similarity is captured by clustering the strain and stress fields into a set of $N_{\mathcal{D}}$ representative strain-stress pairs, referred to as material states, denoted by $(\boldsymbol{\mathcal{E}}_i, \boldsymbol{\mathcal{S}}_i)$. The collection of these representative pairs defines the material database:

$$\mathcal{D} = \{(\boldsymbol{\mathcal{E}}_i, \boldsymbol{\mathcal{S}}_i)\}_{i=1,\dots,N_{\mathcal{D}}} \ . \tag{2.3}$$

Each mechanical state $(\boldsymbol{\varepsilon}_e, \boldsymbol{\sigma}_e)$ is connected to a material state $(\boldsymbol{\mathcal{E}}_i, \boldsymbol{\mathcal{S}}_i)$ through a binary matrix pairing $P \in \{0,1\}^{m \times N_{\mathcal{D}}}$. The matrix component P_{ei} takes the value 1 if the mechanical state e is paired with the material state i, and 0 otherwise:

$$P_{ei} \in \{0,1\} \quad \text{for } 1 \le e \le m \text{ and } 1 \le i \le N_{\mathcal{D}} \ , \tag{2.4}$$

$$\sum_{i'=1}^{N_{\mathcal{D}}} P_{ei'} = 1 \quad \forall e \ . \tag{2.5}$$

The columns of P partition the set of mechanical states into $N_{\mathcal{D}}$ clusters:

$$\Omega_i = \{e \,|\, P_{ei} = 1\} \quad \forall i \ . \tag{2.6}$$

The number of elements in cluster i is then given by

$$N_i = |\Omega_i| = \sum_{e'=1}^{m} P_{e'i} \quad \forall i \ . \tag{2.7}$$

To illustrate the definition of matrix P, consider the example shown in Figure 2.1, where the data is partitioned into $N_{\mathcal{D}} = 3$ clusters. Let us examine the first mechanical state $(\boldsymbol{\varepsilon}_1, \boldsymbol{\sigma}_1)$, we observe that this pair is closest to the second material cluster. Therefore, it is assigned to cluster $i = 2$. This assignment is encoded in the binary matrix P by setting $P_{12} = 1$, and the remaining entries in the first row to zero.

The material database and the pairing matrix are used to build the material strain field:

$$\boldsymbol{\varepsilon}_e^* = \sum_{i'=1}^{N_{\mathcal{D}}} P_{ei'} \boldsymbol{\mathcal{E}}_{i'} \quad \forall e \ , \tag{2.8}$$

and the material stress field:

$$\boldsymbol{\sigma}_e^* = \sum_{i'=1}^{N_{\mathcal{D}}} P_{ei'} \boldsymbol{\mathcal{S}}_{i'} \quad \forall e \ . \tag{2.9}$$

By construction, these material fields are constant over each cluster Ω_i.

DDI is governed by the following principles:

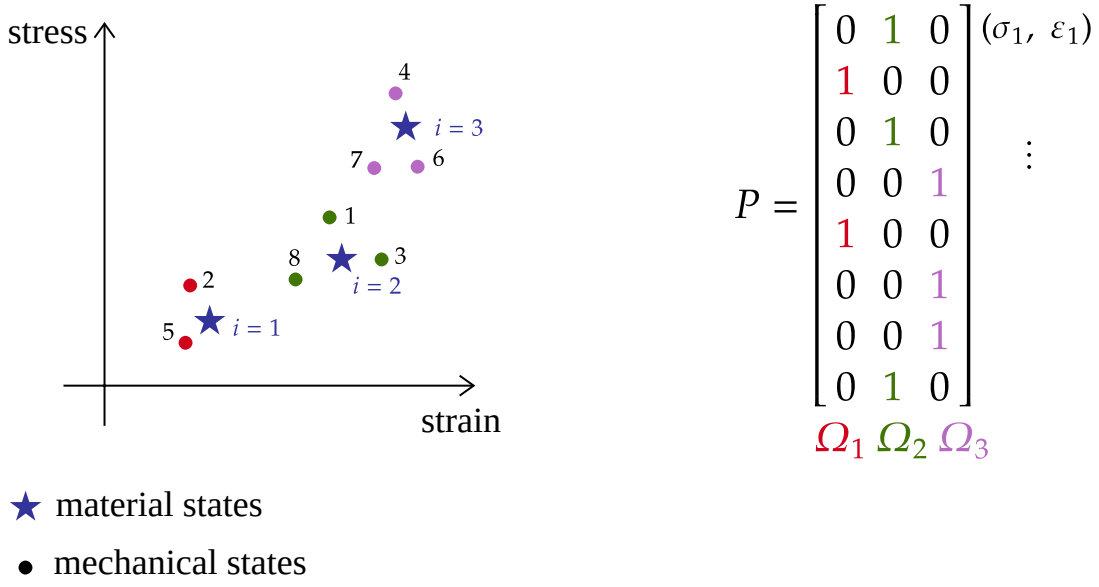


Figure 2.1: Example of the binary matrix P, where each row corresponds to an element and each column to a cluster Ω_i.

- At each element e is assigned a material state $(\boldsymbol{\mathcal{E}}_i, \boldsymbol{\mathcal{S}}_i)$ that approximates its corresponding mechanical state $(\boldsymbol{\varepsilon}_e, \boldsymbol{\sigma}_e)$ by minimizing the distance between them. This distance is measured using the $\mathbb{C}$-norm defined in Eq. 2.1.

- The mechanical stress field $\boldsymbol{\sigma}_e$ must satisfy the equilibrium given in Eq. (2.2),

This leads to the following loss function $\mathcal{F}$:

$$\begin{aligned}\mathcal{F}(\boldsymbol{\sigma}_e, \boldsymbol{\mathcal{E}}_i, \boldsymbol{\mathcal{S}}_i, P_{ei}) =& \frac{1}{2}\sum_{e'=1}^{m} w_{e'}\,(\boldsymbol{\varepsilon}_{e'} - \boldsymbol{\varepsilon}^*_{e'}) : \mathbb{C} : (\boldsymbol{\varepsilon}_{e'} - \boldsymbol{\varepsilon}^*_{e'}) \\ &+ \frac{1}{2}\sum_{e'=1}^{m} w_{e'}\,(\boldsymbol{\sigma}_{e'} - \boldsymbol{\sigma}^*_{e'}) : \mathbb{C}^{-1} : (\boldsymbol{\sigma}_{e'} - \boldsymbol{\sigma}^*_{e'}) \\ =& \frac{1}{2}\sum_{e'} w_{e'} \left\| (\boldsymbol{\varepsilon}_{e'}, \boldsymbol{\sigma}_{e'}) - \sum_{i'} P_{e'i'} (\boldsymbol{\mathcal{E}}_{i'}, \boldsymbol{\mathcal{S}}_{i'}) \right\|^2_{\mathbb{C}} . \end{aligned} \tag{2.10}$$

Thus, the DDI method consists of minimizing this loss function over the mechanical stress field $\boldsymbol{\sigma}_e$, the material database $\mathcal{D}$, and the matrix P, subject to the equilibrium constraint:

$$\{\boldsymbol{\sigma}_e, \boldsymbol{\mathcal{E}}_i, \boldsymbol{\mathcal{S}}_i, P_{ei}\} = \arg\min \mathcal{F}(\boldsymbol{\sigma}_{e'}, \boldsymbol{\mathcal{E}}_{i'}, \boldsymbol{\mathcal{S}}_{i'}, P_{e'i'}) \tag{2.11}$$
$$\text{subject to}$$
$$\sum_{e'=1}^{m} \mathbf{D}_{ke'} \cdot \boldsymbol{\sigma}_{e'} - \mathbf{f}_k = 0 \quad \forall k \ .$$

The only parameters of this method are: the scalar $N_{\mathcal{D}}$ referring to the number of clusters, the tensor $\mathbb{C}$ related to the norm, and the scalar weights w_e, which are not necessarily equal to the finite element quadrature weights w_e^{FE}.

This formulation corresponds to the original method introduced by Leygue *et al.* (2018) [61], with notations inspired by [10, 52, 60]. It was proposed for multiple loading steps (common in Digital Image Correlation data) by including a sum over snapshots -loading steps- in the objective function. In the present work, the equations are written for a single snapshot, but the generalization to multiple snapshots is straightforward.

2.2.3 Solution

Following [61], the DDI problem is solved using an alternating minimization approach, in which the discrete variables P_{ei} and the continuous variables $\{\boldsymbol{\sigma}_e, \boldsymbol{\mathcal{E}}_i, \boldsymbol{\mathcal{S}}_i\}$ are updated alternately until a fixed point is reached. The procedure is given by:

- The first minimization over P_{ei} with a fixed continuous variables $\{\boldsymbol{\sigma}_e, \boldsymbol{\mathcal{E}}_i, \boldsymbol{\mathcal{S}}_i\}$ is performed independently for each e and is given by:

$$P_{ei} = \delta_{ii^*} \qquad \forall e \ , \tag{2.12}$$
$$\text{with,}$$
$$i^* = \arg\min_{i'} (\boldsymbol{\varepsilon}_e - \boldsymbol{\mathcal{E}}_{i'}) : \mathbb{C} : (\boldsymbol{\varepsilon}_e - \boldsymbol{\mathcal{E}}_{i'}) + (\boldsymbol{\sigma}_e - \boldsymbol{\mathcal{S}}_{i'}) : \mathbb{C}^{-1} : (\boldsymbol{\sigma}_e - \boldsymbol{\mathcal{S}}_{i'}) \ , \tag{2.13}$$

  where δ_{ii^*} denotes the Kronecker delta. This equation assigns each mechanical state to the closest database point according to the $\mathbb{C}$-norm.

- For the second minimization with a fixed P_{ei} the problem is solved as a constraint minimization problem using a set of Lagrange multipliers $\boldsymbol{\eta}_k$ leading to the following stationarity problem:

$$\boldsymbol{\delta}(\mathcal{L}(\boldsymbol{\sigma}_e, \boldsymbol{\mathcal{E}}_i, \boldsymbol{\mathcal{S}}_i; \boldsymbol{\eta}_k)) = 0 \ , \tag{2.14}$$

where the Lagrangian is given by

$$\begin{aligned}\mathcal{L}(\boldsymbol{\sigma}_e, \boldsymbol{\mathcal{E}}_i, \boldsymbol{\mathcal{S}}_i; \boldsymbol{\eta}_k) =& \frac{1}{2}\sum_{e'} w_{e'}\Big(\boldsymbol{\varepsilon}_{e'} - \sum_{i'} P_{e'i'}\boldsymbol{\mathcal{E}}_{i'}\Big) : \mathbb{C} : \Big(\boldsymbol{\varepsilon}_{e'} - \sum_{i'} P_{e'i'}\boldsymbol{\mathcal{E}}_{i'}\Big) \\ &+ \frac{1}{2}\sum_{e'} w_{e'}\Big(\boldsymbol{\sigma}_{e'} - \sum_{i'} P_{e'i'}\boldsymbol{\mathcal{S}}_{i'}\Big) : \mathbb{C}^{-1} : \Big(\boldsymbol{\sigma}_{e'} - \sum_{i'} P_{e'i'}\boldsymbol{\mathcal{S}}_{i'}\Big) \\ &- \sum_{k'} \boldsymbol{\eta}_{k'} \cdot \Big(\sum_{e'} \mathbf{D}_{k'e'} \cdot \boldsymbol{\sigma}_{e'} - \mathbf{f}_{k'}\Big) \ , \end{aligned} \tag{2.15}$$

and $\boldsymbol{\delta}(\cdot)$ denotes the first variation. Taking variations with respect to each variable yields:

$$\boldsymbol{\delta}\boldsymbol{\mathcal{E}}_i \implies \sum_{e'} w_{e'}\mathbb{C} : \Big(\boldsymbol{\varepsilon}_{e'} - P_{e'i}\boldsymbol{\mathcal{E}}_i\Big) = 0 \qquad \forall i \ , \tag{2.16a}$$

$$\boldsymbol{\delta}\boldsymbol{\mathcal{S}}i \implies \sum_{e'} w_{e'}\mathbb{C}^{-1} : \Big(\boldsymbol{\sigma}_{e'} - P_{e'i}\boldsymbol{\mathcal{S}}_i\Big) = 0 \qquad \forall i \ , \tag{2.16b}$$

$$\boldsymbol{\delta}\boldsymbol{\sigma}_e \implies w_e\mathbb{C}^{-1} : \Big(\boldsymbol{\sigma}_e - \sum_{i'} P_{ei'}\boldsymbol{\mathcal{S}}_{i'}\Big) - \sum_{k'} \mathbf{D}^{\top}_{k'e} \cdot \boldsymbol{\eta}_{k'} = 0 \quad \forall e \ , \tag{2.16c}$$

$$\boldsymbol{\delta}\boldsymbol{\eta}_k \implies \sum_{e'} \mathbf{D}_{ke'} \cdot \boldsymbol{\sigma}_{e'} - \mathbf{f}_k = 0 \qquad \forall k \ . \tag{2.16d}$$

Equations (2.16a) and (2.16b) correspond to the w_e-weighted average of the mechanical state in elements paired with this specific material state:

$$\boldsymbol{\mathcal{E}}_i = \frac{1}{W_i}\sum_{e'} P_{e'i} w_{e'} \boldsymbol{\varepsilon}_{e'} \quad \forall i \ , \tag{2.17}$$

and,

$$\boldsymbol{\mathcal{S}}_i = \frac{1}{W_i}\sum_{e'} P_{e'i} w_{e'} \boldsymbol{\sigma}_{e'} \quad \forall i \ , \tag{2.18}$$

with cluster weight

$$W_i = \sum_{e'} P_{e'i} w_{e'} = \sum_{e' \in \Omega_i} w_{e'} \quad \forall i \ . \tag{2.19}$$

Combining Eqs. (2.16c) and (2.16d), the above equations can be simplified to:

$$\sum_{e'} \mathbf{D}_{ke'} \left[\sum_{i'} P_{e'i'}\boldsymbol{\mathcal{S}}_{i'} + \frac{1}{w_{e'}}\sum_{j} \mathbb{C} : \mathbf{D}^{\top}_{ke'} \cdot \boldsymbol{\eta}_j\right] - \mathbf{f}_k = 0 \quad \forall k \ . \tag{2.20}$$

Finally, the algorithm to solve DDI problem is given by the following steps:

1. Initialize P_{ei} and $\boldsymbol{\mathcal{E}}_i$ using a clustering method on $\boldsymbol{\varepsilon}_e$.
2. Initialize $\boldsymbol{\sigma}_e$ in any reasonable way.

3. Compute $\boldsymbol{\mathcal{S}}_i$ and $\boldsymbol{\eta}_k$ from Eqs. (2.18) and (2.20).
4. Update $\boldsymbol{\sigma}_e$ using Eq. (2.16c).
5. Update $\boldsymbol{\mathcal{E}}_i$ using Eq. (2.17).
6. Update the pairing P_{ei} using Eq. (2.12).
7. Iterate 3-6 until convergence.

In the first step, the clustering method used throughout this work will be the k-means clustering method [70], described in detail in Appendix A. An illustrative example with three clusters, visualized both in the constitutive space and on the mesh, is presented in Appendix B.

2.2.4 Simple example: nonlinear elastic truss

To facilitate a clearer understanding of DDI, we consider a simple truss example. DDI is applied using synthetic (numerical) data generated by a finite element simulation. For this simulation, a nonlinear elastic behavior of the form

$$\sigma = E(\varepsilon + 50\varepsilon^3) \ ,$$

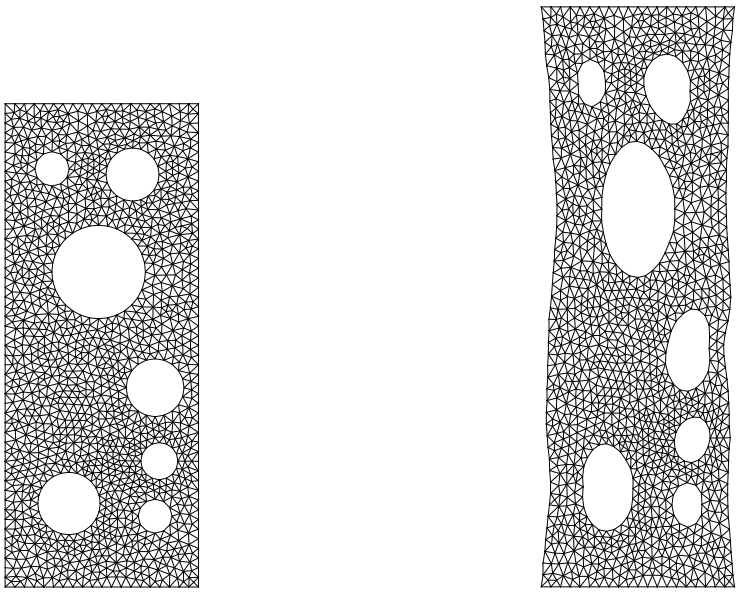
where E denotes the elastic modulus. For convenience, we introduce the normalized stress

$$\bar{\sigma} := \frac{\sigma}{E} \ .$$

The structure comprises 1340 nodes and 2416 bar elements (Figure 2.2) and is subjected to a 20% strain in the vertical direction as shown in Figure 2.2b. These simulations provide the required inputs for the algorithm: the geometry, applied loads, displacement fields, and strain fields.

The effectiveness of the method depends on the availability of heterogeneous and rich data. Figure 2.3a shows the FEM generated reference response that DDI seeks to recover. The color scale indicates the logarithm of the local sampling density of mechanical states. The strain samples are concentrated in the interval $[-0.1, 0.2]$, as indicated by this density figure and by the $\boldsymbol{\varepsilon}$ histogram in Figure 2.3b. Accordingly, DDI is expected to identify more accurate stress fields in this range, where data coverage is highest. In addition, DDI cannot estimate stress values outside the interval $[-0.5, 1.5]$, as shown in Figure 2.3a.

Figure 2.4 shows the DDI results obtained using a database size of $N_{\mathcal{D}} = 50$ clusters. The blue points represent the material responses estimated by DDI. As expected, mechanical states with lower data availability (such as strains in the range $[0.2, 0.3]$) exhibit slightly less alignment with the reference curve, but remain well identified.



(a) (b)

Figure 2.2: Truss structure used to generate the synthetic data. (a) Undeformed mesh of the membrane. (b) Traction deformation 20 % in the vertical direction.

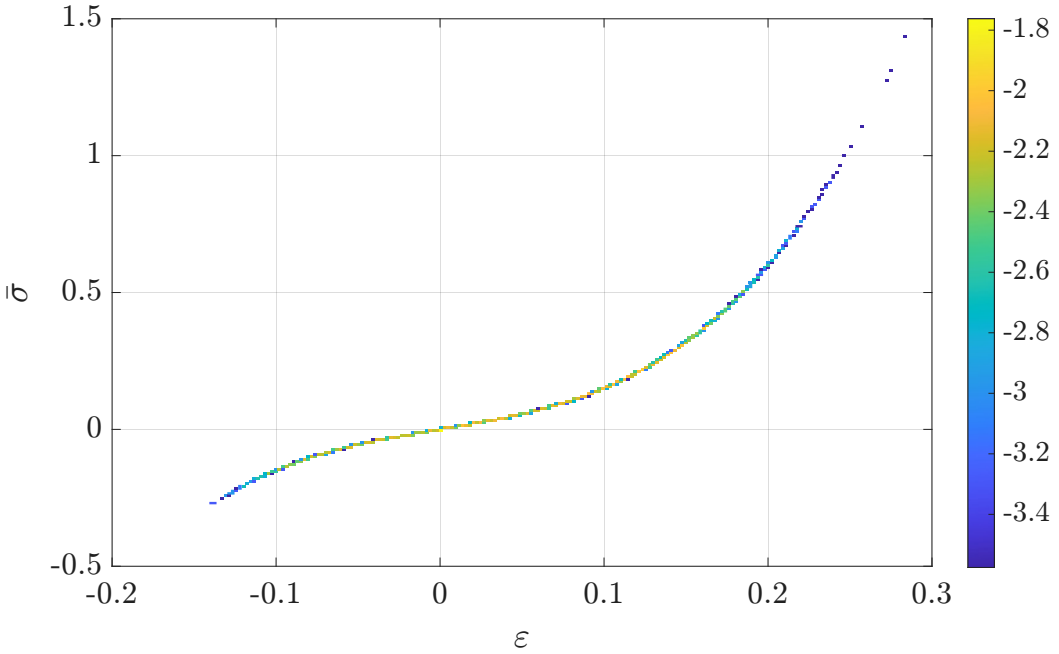
Figure 2.5 provides a quantitative assessment of the performance of DDI. The relative error is defined as

$$\text{relative error} = \frac{\|\sigma^{\text{FE}} - \sigma^{\text{DDI}}\|_2}{\|\sigma^{\text{FE}}\|_2} \,, \tag{2.21}$$

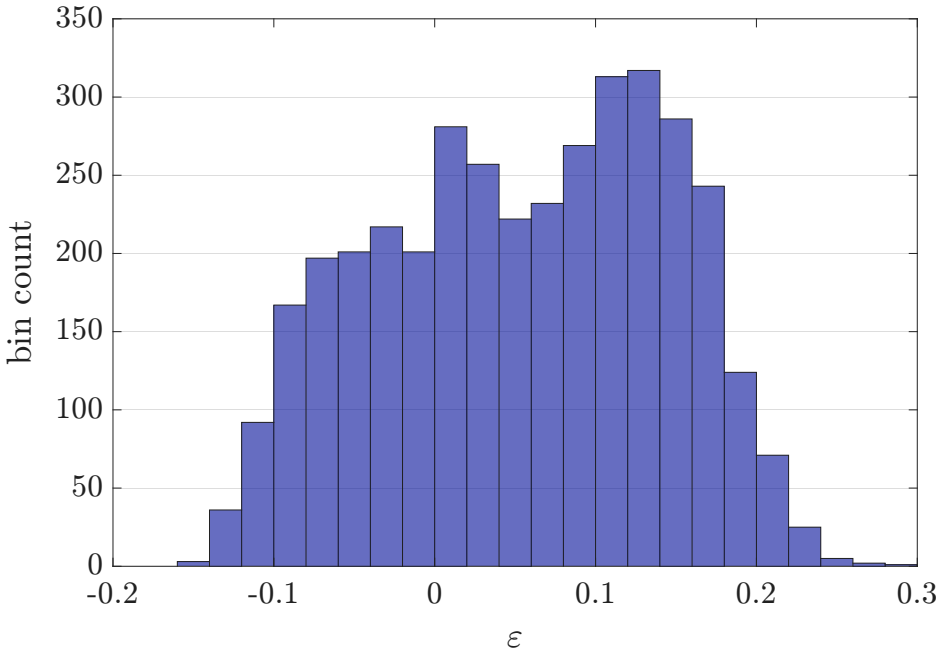
where σ^{FE} and σ^{DDI} denote the stresses obtained from finite element simulations and the DDI method, respectively. We consider 10 different $N_{\mathcal{D}}$ values ranging from 10 to 1000 clusters. As the number of clusters increases, the accuracy of DDI improves, with the relative error decreasing to values close to 0.1%.

2.3 Evolution and development of the DDI method

The evolution of the DDI method is summarized in Table 2.2, which presents selected work that contributed to its development in chronological order. The table highlights the main DDI-related contributions of each work, not the full scope of each study. Additional studies applying the DDI method can be found, for example, in [64, 84, 91].



(a) Density plot of the sampled response behavior obtained by FEM.



(b) Histogram of the distribution of $\boldsymbol{\varepsilon}$ values.

Figure 2.3: Illustration of sampled response behavior and the resulting synthetic data distribution.

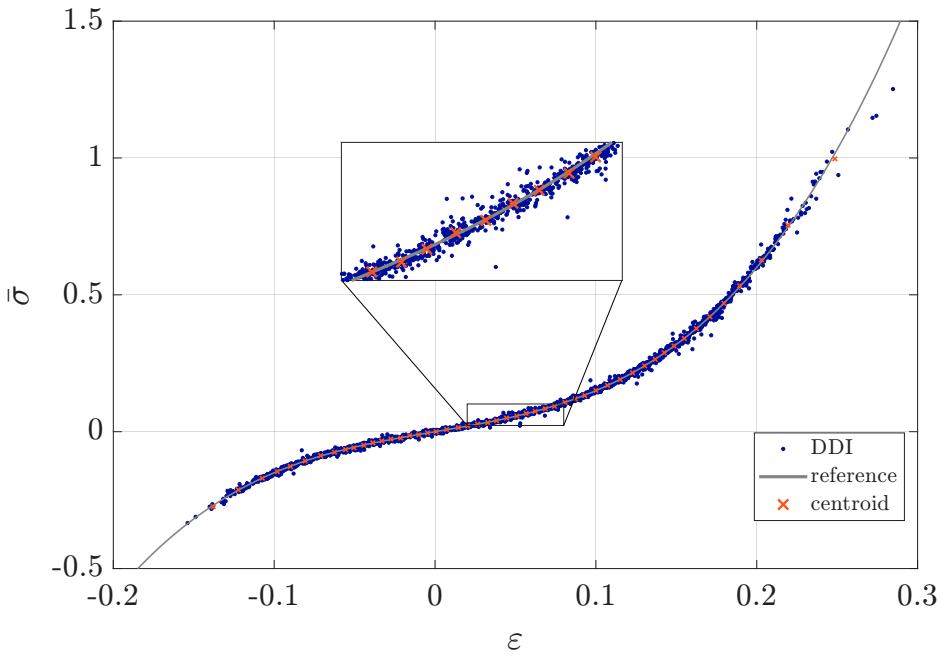


Figure 2.4: DDI result computed for $N_{\mathcal{D}} = 50$ clusters.

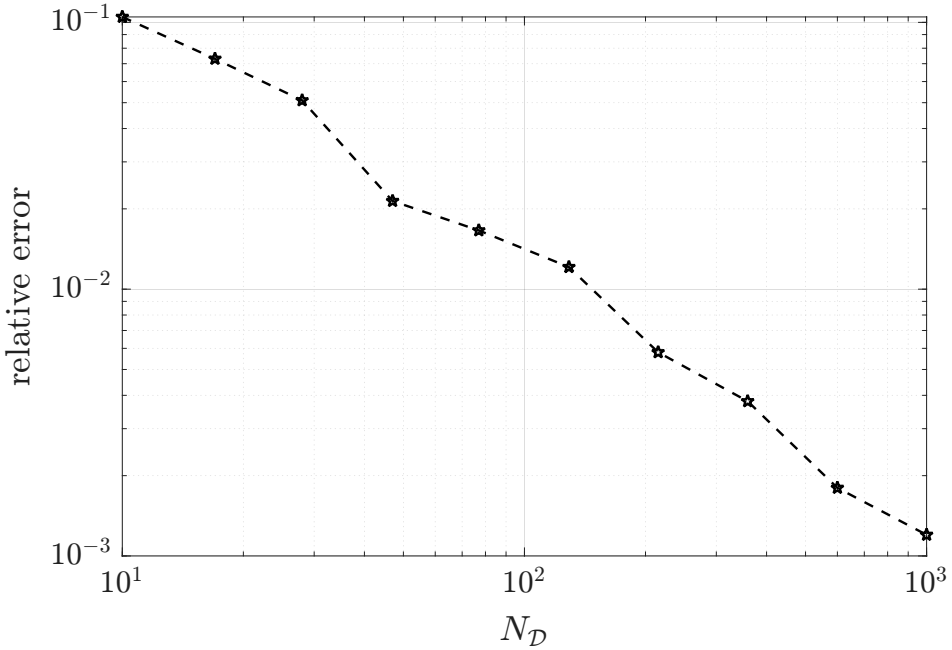


Figure 2.5: Relative error for 10 different values of $N_{\mathcal{D}}$.

Table 2.2: Summary of key contributions to the evolution and development of the DDI method.

Author(s) (Year)	**Main contributions**
Leygue *et al.* (2018) [61]	• Introduced the DDI method to identify material response without postulating a model, first for truss structures and then extended to elastic materials (see Section 2.2). • Applied the method to synthetic data for both cases and conducted a preliminary noise sensitivity study showing that DDI remains largely robust to displacement field noise thanks to the averaging effect inherent in the clustering process.
Dalémat *et al.* (2019) [30]	• First application of the DDI method to real experimental data. Stress fields were computed from DIC displacement measurements taken during a uniaxial test on an elastomer membrane with holes. • Extended the DDI framework to finite-strain elasticity by replacing the linearized strain tensor (unsuitable for large strains) with the Hencky strain tensor [47].
Valdés-Alonzo *et al.* (2021) [90]	• Applied DDI to heterogeneous (bi-material) samples, and employed Correspondence Analysis (CA) [44] to visualize the state mapping matrix P and its evolution, enabling the identification of material phases and the separation of the behaviors of each material identified by DDI.

Continued on next page

Table 2.2 – continued from previous page

Author(s) (Year)	**Main contributions**
Langlois *et al.* (2022) [59]	• The DDI method was extended to history-dependent materials using a differential representation with first-order time derivatives. The mapping becomes stress-strain, stress-rate, strain-rate mapping. • The core algorithm remains unchanged in this extension, preserving the intuition that similar stresses should produce similar strains, while adding the same intuition for their history. This extension was applied to elastoplastic behavior with localization.
Valdés-Alonzo (2022) [89]	• Extended the method to viscoelastic behavior by considering the strain history in the database space rather than using only the instantaneous strain.
Costecalde *et al.* (2023) [28]	• Applied the DDI approach for hyperelastic models, as presented by Dalémat *et al.* [30], which determines the strain energy density and enables complete material characterization. • Introduced the Data-Driven Model Identification method, which consists of selecting a hyperelastic model and fitting its parameters using experimental data (either with the DDI stress field or the strain energy density field). New graphical tools were also developed to facilitate the visualization of the strain and strain energy fields.
Dalémat *et al.* (2023) [31]	• Studied the influence of incomplete experimental data and analyzed the robustness of the solution. They explained the causes of data incompleteness and proposed a strategy to handle it without making any assumptions beyond the balance of linear momentum. • Investigated the influence of method parameters: the number of clusters $N_{\mathcal{D}}$ and the fourth-order tensor $\mathbb{C}$.

2.4 Discussion

In this section, we discuss some challenges that have been found in the literature regarding the method.

Error increase after a threshold number of clusters

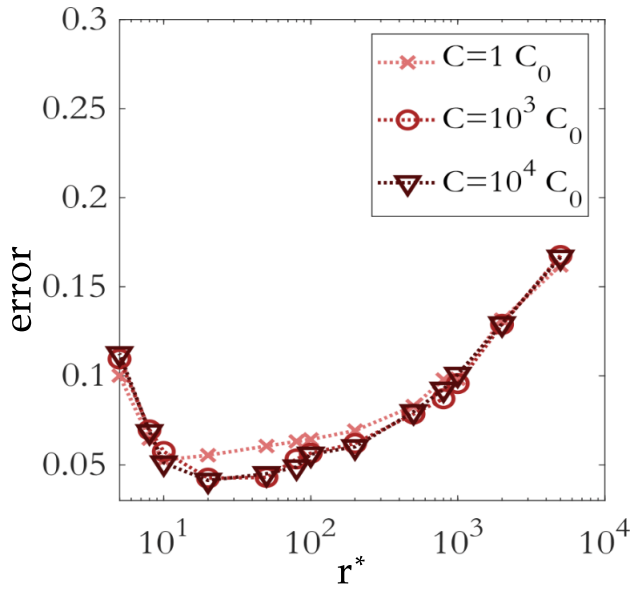


Figure 2.6: Influence of DDI parameters as a function of a measure constraint identification error. Figure from Dalémat (2019) [29].

Figure 2.6 is from the work of Dalémat [29]. In this figure, e denotes the relative error as defined earlier in Eq. (2.21), r^* is inversely proportional to the size of the database through

$$r^* = \frac{m}{N_{\mathcal{D}}} ,$$

and C is the parameter related to the norm in Eq. (2.1) with $\mathbb{C} = C\mathbb{I}$.

As in Figure 2.5, a decrease in r^* (an increase in $N_{\mathcal{D}}$), leads to a reduction in error. However, below a certain value of r^*, the error unexpectedly increases. This effect is analogous to overfitting in regression: the behavior is no longer sufficiently regularized, resulting in a larger error.
This detrimental effect of having too many material states does not appear in Figure 2.5 because that earlier analysis did not explore this range of parameter values.

Presence of limited number of points with large strain error

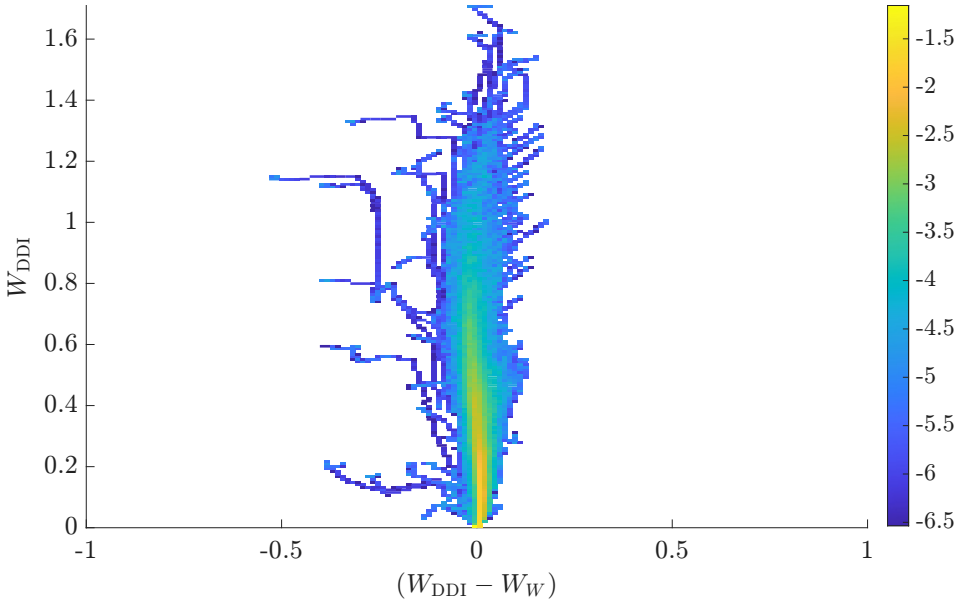


Figure 2.7: Comparison between fitted Ogden model strain field and DDI strain field. Figure from Costecalde (2023) [27].

Figure 2.7 is from the thesis of Costecalde [27] and compares the fitted Ogden model strain energy density W_W and DDI strain energy density W_{DDI}, showing the absolute errors.
In this figure, the data forms a vertical point cloud centered around zero error, with majority of the clusters along the ordinate axis, indicating low absolute error for most points. A small number of points (around 4% [27]) deviate significantly from this cluster, creating the blue slopes visible in both the absolute error plot.
A similar observation was reported by Madeira *et al.* [64], who noted in Section 3.1.2. that *"low occurrence values are filtered by a* $10^{-4.5}$ *threshold"*, referring to the filtering of a small number of similar outliers points.

This challenge was not clearly understood in the previous study, giving rise to the following questions:

- What causes the error to increase when the number of clusters becomes larger, and can we go beyond this limit to achieve higher accuracy?
- If convergence occurs, can we be confident that the solution corresponds to the true mechanical stress?
- What are these outliers points, and can we detect them and propose a correction to obtain a more stable solution?

These questions highlight the need for a rigorous theoretical study of the method to validate the ability of DDI to accurately estimate and converge to the true mechanical stress. This forms the principal motivation of the present work and aims to address these questions.

2.5 Conclusion

In this chapter, we recall the original DDI formulation, presenting the algorithm alongside a simple illustrative example on an elastic truss sample. We then outline the evolution of the associated problems. The DDI method has since been successfully applied to both synthetic and experimental data, capturing a wide spectrum of material behaviors: small-strain elasticity, finite-strain elasticity, viscoelasticity, and elastoplasticity. Its key strengths include its model-free formulation, robustness to noise, and capacity to handle heterogeneous full-field measurements and complex behaviors.
Nevertheless, challenges remain, notably the accuracy loss observed beyond a certain number of clusters and the occurrence of stress outliers that affect stability. Overcoming these limitations calls for a deeper theoretical analysis to guarantee convergence to the true mechanical response and to improve the method reliability in practical applications.
The insights gained in this chapter thus provide the foundation for the methodological, mathematical, and numerical developments presented in the following chapters.

Chapter 3

Well-posedness analysis of the Data-Driven Identification problem

Summary

3.1 Introduction

The previous chapter highlighted the need for a theoretical analysis of the DDI problem. As the DDI parameters are refined to gain accuracy, the problem becomes ill-posed. Recently, Leygue [60] introduced a new formulation that offers a criterion to determine the well-posedness of the problem and an error estimate for the identified stress field under elastic behavior.
This chapter investigates numerically and algorithmically the preliminary work of Leygue [60]. Building on that foundation, this chapter aims to deepen the mathematical understanding of DDI, with particular emphasis on the choice of database size and the convergence of the associated algorithm.
The recent formulation of the DDI method is first recalled, together with the criterion for the uniqueness of the obtained solution. Three methods are then presented to evaluate this well-posedness criterion, supported by a theoretical analysis of their efficiency. A new implementation strategy for the DDI algorithm is subsequently introduced, inspired by the third method, along with a proof of its convergence. The effectiveness of these methods is systematically tested using synthetic data, with a detailed numerical analysis and discussion of their performance.

This chapter is adapted from Hachem *et al.* (2025) [45], with minor modifications for clarity and integration into the present manuscript.

3.2 Data-Driven Identification

This section examines the recent DDI formulation presented in [60] and revisits the uniqueness criterion established by this formulation.

3.2.1 Formulation

Recently, DDI has been formulated by a new loss function that no longer explicitly includes the material states, but implicitly through the matrix $\tilde{H}$. The recent formulation [60] is given by:

$$(\boldsymbol{\sigma}_e, P_{ei}) = \arg\min \frac{1}{2} \sum_{e',a'} \tilde{H}_{e'a'} \boldsymbol{\varepsilon}_{e'} : \mathbb{C} : \boldsymbol{\varepsilon}_{a'} + \frac{1}{2} \sum_{e',a'} \tilde{H}_{e'a'} \boldsymbol{\sigma}_{e'} : \mathbb{C}^{-1} : \boldsymbol{\sigma}_{a'} \ , \tag{3.1}$$

$$\text{subject to} \quad \sum_{e'=1}^{m} \mathbf{D}_{ke'} \cdot \boldsymbol{\sigma}_{e'} = \mathbf{f}_k \quad \forall k \ .$$

with the matrix $\tilde{H}$ defined as:

$$\tilde{H}_{ea} = \sum_{e'} (I_{ee'} - \tilde{M}_{ee'}) w_{e'} (I_{e'a} - \tilde{M}_{e'a}) \quad \forall e, a \ . \tag{3.2}$$

where

$$\tilde{M}_{ea} = \sum_{i} \frac{1}{W_i} P_{ei} P_{ai} w_a \quad \forall e, a \ , \tag{3.3}$$

and

$$W_i = \sum_{e'} P_{e'i} w_{e'} \quad \forall i \ . \tag{3.4}$$

The matrix $\tilde{H}$ in (3.1) depends implicitly on the matrix P, i.e. $\tilde{H} := \tilde{H}(P)$. As discussed in [60], the matrix $\tilde{M}$ computes the mean of each cluster and $\tilde{M}$ is a projection onto the nullspace of $\tilde{H}$. The columns of P form a basis of $\ker H$ and P is thus orthogonal by construction. For a more detailed discussion on the properties of $\tilde{M}$ and $\tilde{H}$, see [60].
The unknown material states of the DDI problem (2.11) can be found by this new formulation. They are simply the w_e-weighted average of the mechanical states over each cluster. Mathematically, this can be expressed as follows:

$$\boldsymbol{\mathcal{E}}_i = \frac{1}{W_i} \sum_{e'} P_{e'i} w_{e'} \boldsymbol{\varepsilon}_{e'} \quad \forall i \ , \tag{3.5}$$

and

$$\boldsymbol{\mathcal{S}}_i = \frac{1}{W_i} \sum_{e'} P_{e'i} w_{e'} \boldsymbol{\sigma}_{e'} \quad \forall i \ . \tag{3.6}$$

For algebraic convenience, the problem will be expressed in matrix notation instead of index notation. The problem (3.1) can be rewritten using Mandel notation [65] as follows:

$$(\sigma, P) = \arg\min \frac{1}{2} \varepsilon^\top H_\varepsilon \varepsilon + \frac{1}{2} \sigma^\top H_\sigma \sigma \qquad \text{subject to} \quad D\sigma = F \ . \tag{3.7}$$

where,

- $\varepsilon = [\varepsilon_e^M]$ denotes the collection vector corresponding to the second-order tensors $\boldsymbol{\varepsilon}$, in Mandel notation,
- $\sigma = [\sigma_e^M]$ denotes the collection vector corresponding to the second-order tensors $\boldsymbol{\sigma}$, in Mandel notation,
- $H_\varepsilon = \tilde{H} \otimes C$ represents the Kronecker product of the matrix $\tilde{H}$ and C,
- $H_\sigma = \tilde{H} \otimes C^{-1}$ represents the Kronecker product of the matrix $\tilde{H}$ and C^{-1},

- C refers to the matrix representation of the fourth-order tensor $\mathbb{C}$, in Mandel notation,
- $M = \tilde{M} \otimes I$ represents the Kronecker product of the matrix $\tilde{M}$ and identity matrix I,
- D and F denote the discrete divergence operator $\mathbf{D}$ and the external force vector $\mathbf{f}$, in Mandel notation, respectively.

3.2.2 Uniqueness of the DDI solution

Theorem 3.2.1. *[60] The solution of DDI problem is unique if and only if there does not exist a non-zero vector $\tilde{\sigma}$ such that $H_\sigma \tilde{\sigma} = 0$ and $D\tilde{\sigma} = 0$, that is, if and only if* $\ker D \cap \ker H_\sigma = \{0\}$.

Remark. *To relax the notation, we denote H_σ as H, unless otherwise specified.*

We consider a given clustering of deformations, defined through P_{ei}, i.e. the matrices H_σ and H_ε in Equation (3.7) are fixed. Then, the mechanical stresses $\boldsymbol{\sigma}_e$ are the solution of the following problem:

$$\sigma = \arg\min \frac{1}{2}\|\sigma\|_H^2 \qquad \text{with} \qquad D\sigma = F \ . \tag{3.8}$$

Here, $\|\cdot\|_H$ denotes a semi-norm, as H is positive semi-definite [60], and defined as $\|\sigma\|_H^2 := \sigma^\top H \sigma$.

Proof. We will prove only the implication of interest to us.
($\Longleftarrow$) Suppose that $\ker D \cap \ker H = \{0\}$. Our goal is to prove that the solution is unique under this assumption. Let σ_1 and σ_2 be two different solutions to the problem. Thus, we have:

1. Since $D\sigma_1 = F$ and $D\sigma_2 = F$, it follows that $\sigma_1 - \sigma_2 \in \ker D$.
2. As both σ_1 and σ_2 minimize the same quadratic functional, we have $\frac{1}{2}\sigma_1^\top H \sigma_1 = \frac{1}{2}\sigma_2^\top H \sigma_2$. This implies that $(\sigma_1 - \sigma_2)^\top H(\sigma_1 - \sigma_2) = 0$. Since H is positive semi-definite, it follows that $\sigma_1 - \sigma_2 \in \ker H$.

From points 1 and 2, we can deduce that $\sigma_1 - \sigma_2 \in \ker D \cap \ker H = \{0\}$. As a result, we have $\sigma_1 = \sigma_2$. □

Corollary 3.2.2. *If problem (3.8) admits multiple solutions, then any two solutions differ by an element of* $\ker H \cap \ker D$.

Proof. This follows immediately from the proof of Theorem 3.2.1. □

3.3 How to characterize the intersection of the two kernels in practice?

This section presents three methods for characterizing the intersection of two kernels: (3.3.1) explicit calculation of the basis, (3.3.2) examination of the angles between the subspaces, and (3.3.3) alternating projection onto the two kernels.
The proofs of the theorems will not be detailed in the following subsections, with references provided for interested readers.

3.3.1 Compute an intersection basis explicitly

Proposition 3.3.1. *[85] The intersection of the kernel of H ($\ker H$) and the kernel of D ($\ker D$) is the kernel of a larger matrix G, where $G = \begin{bmatrix} H \\ D \end{bmatrix}$.*

In other words,

$$\ker D \cap \ker H = \ker G \ . \tag{3.9}$$

Proof. $G\sigma = 0$ implies both $H\sigma = 0$ and $D\sigma = 0$. So σ is in the intersection of $\ker D$ and $\ker H$. □

The matrices H and D are sparse, therefore, the matrix G is also sparse. Numerically, the null space of a matrix is computed using singular value decomposition (SVD), which involves non-sparse linear algebra calculations. The computational complexity of SVD is $O(ab^2)$ where a and b are the dimensions of G, with $a \geq b$. For example, in the bidimensional case, we have $a = 3m + 2n$ and $b = 3m$, showing that this complexity is independent of the database size $N_{\mathcal{D}}$. Nevertheless, it provides an exact basis for the intersection.

3.3.2 Angles between subspaces

Let $A \in \mathbb{R}^{d \times p}$ and $B \in \mathbb{R}^{d \times q}$ be two matrices containing the base of two given subspaces $\mathcal{A}, \mathcal{B} \subset \mathbb{R}^d$, respectively, and assume that $p = \dim \mathcal{A} \geq \dim \mathcal{B} = q \geq 1$.

Definition 3.3.1. *[14] The principal angles $\theta_r \in [0, \pi/2]$ between $\mathcal{A}$ and $\mathcal{B}$ are defined for $r = 1, 2, \ldots, q$ by:*

$$\cos\theta_r = \max_{v \in \mathcal{A}} \max_{u \in \mathcal{B}} u^\top v, \qquad \|u\|_2 = 1, \|v\|_2 = 1 \ , \tag{3.10}$$

subject to

$$u_j^\top u_r = 0, \quad v_j^\top v_r = 0, \qquad j = 1, 2, \ldots, r-1 \ . \tag{3.11}$$

The vectors $(u_1, u_2, \ldots, u_q)$ and $(v_1, v_2, \ldots, v_q)$ are called the principal vectors associated to $\{\theta_r\}_{r=1}^q$.

The following theorem illustrates the relation between singular values and the principal angles.

Theorem 3.3.2. *[14] Let Q_A and Q_B be two matrices such that their columns form a unitary basis of $\mathcal{A}$ and $\mathcal{B}$ respectively. Let $N = Q_A^\top Q_B$ and let the singular values decomposition SVD of this $p \times q$ matrix be*

$$N = Y \Lambda Z^\top, \qquad \Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_q) \ . \tag{3.12}$$

Then, the principal angles and the principal vectors associated are given by,

$$\cos(\theta_r) = \lambda_r(N), \qquad U = Q_A Y, \qquad V = Q_B Z \ , \tag{3.13}$$

where λ_r are the singular values of the matrix N.

Proof. The proof of this theorem is provided in Section 3 of [14]. □

If a principal angle θ_r equals zero, or equivalently, a singular value equals one, the solution is non-unique. The number of principal angles equal to zero corresponds to the dimension of the kernel intersection.
The theorem outlined above can also be used to compute an orthonormal basis for the intersection, as given by the following theorem.

Theorem 3.3.3. *[41] Let $\{\cos(\theta_r)\}_{r=1}^q$ and $\{u_r, v_r\}_{r=1}^q$ be defined by Theorem 3.3.2. If the index s is defined by $1 = \cos(\theta_1) = \cdots = \cos(\theta_s) > \cos(\theta_{s+1})$, then*

$$\operatorname{span}\{u_1, \ldots, u_s\} = \operatorname{span}\{v_1, \ldots, v_s\} = \mathcal{A} \cap \mathcal{B} \ . \tag{3.14}$$

Proof. The proof is based on the observation that when $\cos(\theta_r) = 1$, it implies that $u_r = v_r$. □

To practically determine the intersection dimension s, a numerical tolerance must be defined to decide when a computed singular value is considered equal to 1.

We apply these theorems by setting $\mathcal{A} = \ker D$ and $\mathcal{B} = \ker H$. The procedure follows the following steps:

0. According to [60], the null space of H is spanned by the columns of P,
1. Compute the kernel of D, numerically, using SVD decomposition,
2. Compute unit basis of $\ker D$ and $\ker H$, Q_A and Q_B, respectively,
3. Perform SVD on matrix N, the singular values represent the principal angles between the subspaces according to Theorem 3.3.2. And from the principal vectors we can compute an intersection basis according to Theorem 3.3.3.

This method provides insight into the conditioning of our problem, as the presence of very small angles near zero indicates poor conditioning.
Moreover, this approach is computationally less expensive than the previous one. In the first step, we only need to compute the null space of D, which is less costly than computing the null space of matrix G. Although this method allows us to determine the basis for the intersection, it is based on a chosen tolerance. However, it still involves processing full matrices, which does not meet the sparsity criterion. To address this need, we propose an algorithm based on alternating projection onto the two subspaces and preserve the sparse matrix property of the matrices H and D.

3.3.3 Sampling the intersection though alternating projection

The *alternating projection* is a simple algorithm to calculate a point at the intersection of two sets by using a sequence of projections on these sets.
Suppose $\mathcal{X}$ and $\mathcal{Y}$ are two closed convex sets in $\mathbb{R}^d$, and let $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ denote projection onto $\mathcal{X}$ and $\mathcal{Y}$, respectively. The algorithm begins with any point $x_0 \in \mathbb{R}^d$ and then subsequently alternates projections onto $\mathcal{X}$ and $\mathcal{Y}$,

$$y_j = P_{\mathcal{Y}}(x_j), \quad x_{j+1} = P_{\mathcal{X}}(y_j), \qquad \text{for } j \geq 0 \ . \tag{3.15}$$

Theorem 3.3.4. *[21]*

1. *If $\mathcal{X} \cap \mathcal{Y} \neq \varnothing$, then the two sequences x_j and y_j converge to $x^* \in \mathcal{X} \cap \mathcal{Y}$, meaning the alternating projection finds a point in the intersection.*

2. *If $\mathcal{X} \cap \mathcal{Y} = \varnothing$, then $x_j \longrightarrow x^* \in \mathcal{X}$ and $y_j \longrightarrow y^* \in \mathcal{Y}$, where $\|x^* - y^*\|_2 = \operatorname{dist}(\mathcal{X}, \mathcal{Y})$.*

Proof. The convergence proof is given in Section 2 of [21]. □

The null space of a matrix is convex by definition, as it forms a subspace where all linear combinations are invariant. Moreover, this subspace is closed because any linear operator in finite-dimensional space is continuous, and the kernel is the preimage of a closed set, the singleton zero $\{0\}$.
As both kernels are subspaces, their intersection necessarily contains the zero vector. Therefore, our analysis focuses on the first part of the theorem, where we define $\mathcal{X} = \ker H$ and $\mathcal{Y} = \ker D$.

The algorithm is given by:

$$\sigma_0 \xrightarrow{P_H} \sigma_1^H \xrightarrow{P_D} \sigma_1^D \xrightarrow{P_H} \sigma_2^H \xrightarrow{P_D} \sigma_2^D \xrightarrow{P_H} \cdots \to \sigma_f \ , \tag{3.16}$$

where the projections, for $j \geq 1$, are defined as follows:

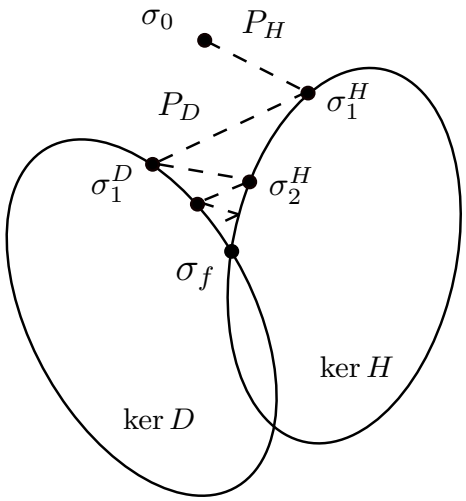


Figure 3.1: Alternating projection onto $\ker D$ and $\ker H$.

- The projection P_H onto $\ker H$ is achieved by multiplying σ by M [60]:

$$P_H(\sigma_j) := \sigma_j^H = M\sigma_{j-1} \ , \tag{3.17}$$

- The projection P_D onto $\ker D$ is computed by solving the following minimization problem:

$$P_D(\sigma_j^H) := \sigma_j^D = \arg\min_{\sigma_j}(\sigma_j - \sigma_j^H)^\top w(\sigma_j - \sigma_j^H) \quad \text{with} \quad D\sigma_j = 0 \ . \tag{3.18}$$

  where w is the weight.

To understand how this algorithm works, consider that the initial point is composed of $\sigma_0 = \sigma_0^I + \sigma_0^\perp$, where σ_0^I lies in the intersection $\ker D \cap \ker H$ and $\sigma_0^\perp$ represents the remainder. The iterative process is defined by:

$$(P_D P_H)^j(\sigma_0) = (P_D P_H)^j(\sigma_0^I) + (P_D P_H)^j(\sigma_0^\perp) \ , \tag{3.19}$$

as $j \to \infty$, the convergence results hold:

- the sequence $(P_D P_H)^j \sigma_0$ converges to an intersection, denoted σ_f,
- since σ_0^I is invariant under these projections, the sequence $(P_D P_H)^j \sigma_0^I$ remains unchanged and converges to σ_0^I,
- the term $(P_D P_H)^j \sigma_0^\perp$, representing the component orthogonal to the intersection, decreases over iterations and tends to zero.

In the case where $\sigma_0^I = 0$, or equivalently, $\sigma_0 \in (\ker H \cap \ker D)^\perp$, the algorithm can converge to $\sigma_f = 0$, even though the solution is not necessarily unique.

Given that σ_0 in (3.16) is chosen randomly, it is highly improbable that $\sigma_0^I = 0$. This condition is similar to the requirement in the power method: the initial vector should not be orthogonal to the dominant eigenvector to ensure proper convergence. Bindel [13] and Demmel [33] discuss that the probability of randomly selecting an initial vector that satisfies this condition is zero. Based on this, we can conclude:

- if $\sigma_f \neq 0$, the solution is non-unique,
- if $\sigma_f = 0$, the solution is unique, with very high probability - assuming that σ_0 is chosen randomly.

In addition, this method allows us to compute a basis by performing multiple initializations and then using a method to check the linear independence of the resulting vectors, such as Gram-Schmidt.
The convergence rate of the alternating projection method is related to the angle between the two subspaces through the following theorem:

Theorem 3.3.5. *[34] For all $j \in \mathbb{N}$,*

$$\|(P_{\mathcal{X}} P_{\mathcal{Y}})^j - P_{\mathcal{X}\cap\mathcal{Y}}\| = \cos^{2j-1}(\theta_{s+1}) \ , \tag{3.20}$$

where θ_{s+1} is the smallest positive principal angle.

Proof. The proof of the rate of convergence theorem is in Chapter 9, Theorem 9.31, of [34]. □

Based on this theorem, the convergence is slow if the smallest positive angle θ_{s+1} is small.
This algorithm is computationally less expensive as it takes advantage of the sparsity of the matrix. As shown in Equation (3.18), there is no need to compute $\ker D$ explicitly, thus, making the algorithm faster and easier to implement.

3.4 Solution

As decribed earlier in Chapter 2 in Section 2.2.3, the algorithm follows the same principle and solve the DDI problem using an alternating minimization approach regarding the discrete variable P and the continuous variable σ [60]. The first minimization over P with a fixed continuous variable σ is given by applying w_e-weighted k-means clustering in the $\mathbb{C}$-norm on the mechanical states. For the second minimization with a fixed P, the problem (3.7) is simplified to:

$$\sigma = \arg\min \frac{1}{2}\|\sigma\|_H^2 \qquad \text{with} \qquad D\sigma = F \ . \tag{3.21}$$

In the following section, we focus on this iteration where P is fixed, and the goal is to solve the minimization problem given in Equation (3.21).

3.4.1 Limitations

The Lagrangian for the minimization problem (3.21) is given by:

$$\mathcal{L}(\sigma;\eta) = \tfrac{1}{2}\|\sigma\|_H^2 + \eta^\top(D\sigma - F) \ , \tag{3.22}$$

The corresponding system obtained from the stationarity equations is

$$\begin{bmatrix} H & D^\top \\ D & 0 \end{bmatrix} \begin{bmatrix} \sigma \\ \eta \end{bmatrix} = \begin{bmatrix} 0 \\ F \end{bmatrix} \ , \tag{3.23}$$

where η denotes the associated Lagrange multiplier, and we define

$$T := \begin{bmatrix} H & D^\top \\ D & 0 \end{bmatrix} \ . \tag{3.24}$$

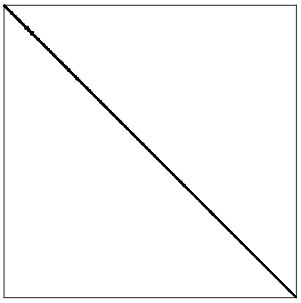
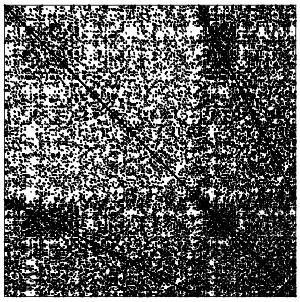
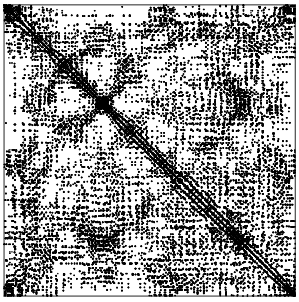
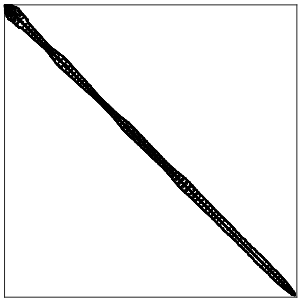
Numbering	$\boldsymbol{H}$	$\boldsymbol{D^\top D}$
Cluster-based		
Mesh-based		

Table 3.1: Comparison of connectivity patterns using the Reverse Cuthill-McKee permutation, with connectivity based on clustering (top row) and mesh (bottom row). The mesh used is the one shown in Figure 3.2, and for the clustering we take $N_{\mathcal{D}} = 300$.

The matrix H encodes the connectivity between mechanical states based on clustering, while the matrix $D^\top D$ encodes the connectivity within the

mesh. By applying the reverse Cuthill-McKee algorithm [41] to either of these matrices we can compute a renumbering of the degrees of freedom, either clustering based or mesh based. In table 3.1, we apply these renumbering to both matrices and observe that their sparsity patterns are incompatible and that the difference in the non-zero structures might result in significant fill-in during standard linear algebra operations.
In direct methods such as LU decomposition, this fill-in appears and increases memory usage and computational cost by introducing additional non-zero elements into the matrix during the factorization process. For example, in the example presented in table 3.1, the matrix T (Eq. (3.24)) has only 0.19% non-zero elements. However, after LU decomposition, the percentage of non-zero elements in $L + U$ rises to 13.14%.
To address these limitations, we propose an iterative approach based on the alternating projection described in Section 3.3.3.

3.4.2 New algorithm based on alternating projection

We recall from [60] that M is a projection onto $\ker H$, obtained by computing the mean of each cluster.
The solution of Equation (3.21) is obtained by alternately projecting σ onto $\ker H$ and $E := \{\sigma \quad \text{s.t.} \quad D\sigma = F\}$. The algorithm for solving (3.21) is given by the following steps:

1. Initialization of σ_0,
2. Projection onto $\ker H$:

$$\sigma_j^H = M\sigma_{j-1} \qquad \text{for } j \geq 1, \tag{3.25}$$

3. Projection onto E [55]:

$$\sigma_j = \sigma_j^H + D^\top \eta \quad \text{with} \quad DD^\top \eta = F - D\sigma_j^H \qquad \text{for } j \geq 1. \tag{3.26}$$

Proof of convergence. Since both $\ker H$ and E are closed convex sets, we can safely apply Theorem 3.3.4, ensuring the convergence of the algorithm. According to this theorem, the algorithm will converge either to a distance or a point within their intersection. In both cases, the algorithm converges to the minimum of our loss function. □

We recommend initializing the algorithm with $\sigma_0 = 0$. In this case, the first projection σ_1^H yields to:

$$\sigma_1^H = M\sigma_0 = 0 \ ,$$

and the projection onto the set E then gives:

$$\sigma_1 = D^\top (DD^\top)^{-1} F = D^\dagger F \ ,$$

Algorithm 1 DDI algorithm based on alternating projection

Require: $\boldsymbol{\varepsilon}$, D and F
1: Choose DDI parameters: w_e, $N_{\mathcal{D}}$ and $\mathbb{C}$
2: Choose algorithm parameters: IterMax, tolerance
3: Initialize σ_0
4: Compute P by performing a w_e-weighted k-means clustering of $\boldsymbol{\varepsilon}$ in the $\mathbb{C}$-norm
5: **for** $j = 1$ to IterMax **do**
6: $\quad \sigma_j^H = M\sigma_{j-1}$ ▷ Proj. onto $\ker H$
7: $\quad$ Solve for η: $DD^\top \eta = F - D\sigma_j^H$
8: $\quad$ Update: $\sigma_j = \sigma_j^H + D^\top \eta$ ▷ Proj. onto E
9: $\quad$ **if** $\|\sigma_{j-1} - \sigma_j\| \leq$ tolerance **then**
10: $\quad\quad$ **break** ▷ Stagnation reached
11: $\quad$ **end if**
12: **end for**
Ensure: $\boldsymbol{\sigma}$, P

where $D^\dagger$ is the pseudo-inverse of matrix D.
This means that the first iterate σ_1 is the pseudo-inverse solution (the minimum norm solution) of the equilibrium equation $D\sigma = F$. This provides a meaningful and stable starting point for the iterative projection process.

This algorithm can still be applied even if the matrix T (Eq. (3.24)) is singular, unlike direct methods, and it will converge to one of the possible solutions.
Similarly to the third method, the rate of convergence depends on the smallest positive principal angle according to Theorem 3.3.5. In the same way, the conditioning of the problem is related to the presence of very small angles close to zero, as discussed in Section 3.3.2.

3.5 Application and discussion

In this section, we apply the DDI method to synthetic data generated from a finite element simulation. The 2D finite element mesh includes 1340 nodes and 2416 quadrature points, as shown in Figure 3.2. The model is subjected to a 10% strain in the vertical direction, assuming plane stress. The material is assumed incompressible and is characterized by the following strain-stress relation:

$$\hat{\boldsymbol{\sigma}}(\boldsymbol{\varepsilon}) = -p\mathbf{I} + 2G(1 + \alpha\,\mathbf{dev}(\boldsymbol{\varepsilon}) : \mathbf{dev}(\boldsymbol{\varepsilon}))\mathbf{dev}(\boldsymbol{\varepsilon}) \ , \tag{3.27}$$

where $\mathbf{dev}(\boldsymbol{\varepsilon})$ represents the deviatoric part of $\boldsymbol{\varepsilon}$, and p is determined by the plane stress condition. The parameter α is chosen to be $\alpha = 3$, the parameter

$\mathbb{C}$ is selected as:

$$\mathbb{C} = \left.\frac{\partial \hat{\boldsymbol{\sigma}}}{\partial \boldsymbol{\varepsilon}}\right|_{\boldsymbol{\varepsilon}=\mathbf{0}} ,$$

and w_e are set to the mesh quadrature weights. For all the results presented in this section, the DDI is carried out for different values of $N_{\mathcal{D}}$ ranging from 10 to 630, more significantly corresponding on average to 4 to 242 material points per cluster. For each $N_{\mathcal{D}}$, the pairing matrix P_{ei} is computed from a w_e-weighted k-means clustering performed on the strain field only.
The stress field generated by the model $\hat{\boldsymbol{\sigma}}$ is denoted as $\sigma_e^{FE} = \hat{\boldsymbol{\sigma}}(\boldsymbol{\varepsilon})$. The mechanical stress resulting from the DDI will be denoted as σ_e^{DDI}. The DDI solution presented in this section is computed using the new algorithm described in Section 3.4.2. For each application of the DDI, the error is calculated as follows:

$$\text{error} = \frac{1}{m}\sum_e \left\|\sigma_e^{\mathrm{DDI}} - \sigma_e^{FE}\right\|_2^2 . \tag{3.28}$$

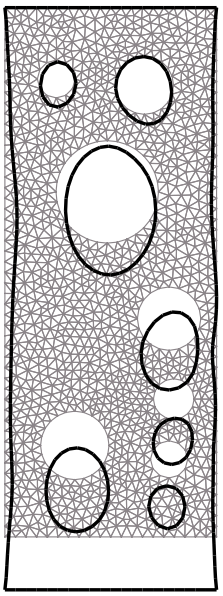
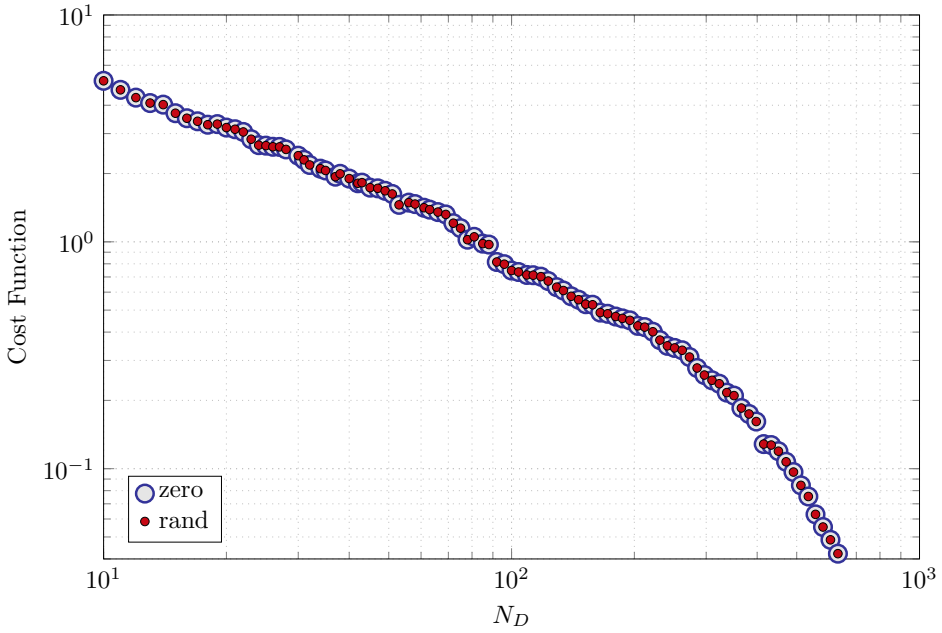
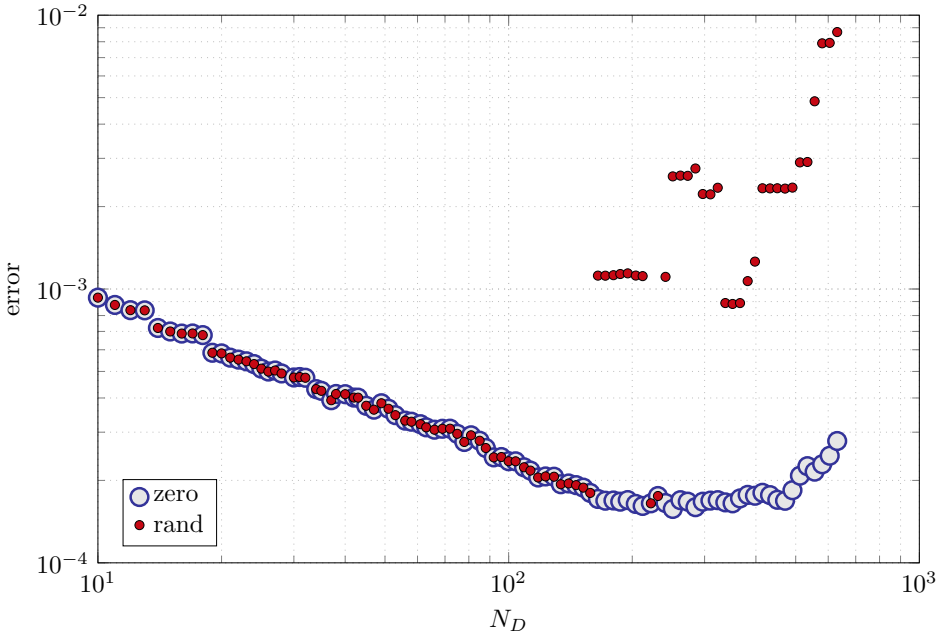


Figure 3.2: Finite element mesh used to generate the synthetic data. The black line indicates the deformation contour.

Figure 3.3 illustrates the loss of well-posedeness of DDI problem. The upper Figure 3.3a shows the cost function of DDI across different database size $N_{\mathcal{D}}$ for two stress initialization σ_0 schemes: random initialization following a normal distribution and zero initialization. The results indicate that the cost function reaches the same minimum value regardless of the initialization, confirming that convergence is achieved. The lower Figure 3.3b shows the error as a function for the same $N_{\mathcal{D}}$ and same two initialization schemes. The error decreases as $N_{\mathcal{D}}$ increases, indicating that DDI converges toward the true stress



(a) Cost function for different database sizes $N_{\mathcal{D}}$ with two initializations.



(b) Comparison of the error using the finite element solution and DDI results for the same two initializations.

Figure 3.3: Illustration of the loss of uniqueness.

field. However, above a certain $N_{\mathcal{D}}$, the DDI converges to different results depending on the initialization. This highlights the loss of well-posedness of the problem: for this $N_{\mathcal{D}}$, it is possible to find a non-trivial stress field that is constant within each cluster (kernel of H) and self-balanced (kernel of D). This stress field corresponds to the difference between the two computed solutions. This can be demonstrated through one of the methods described earlier in Section 3.3.

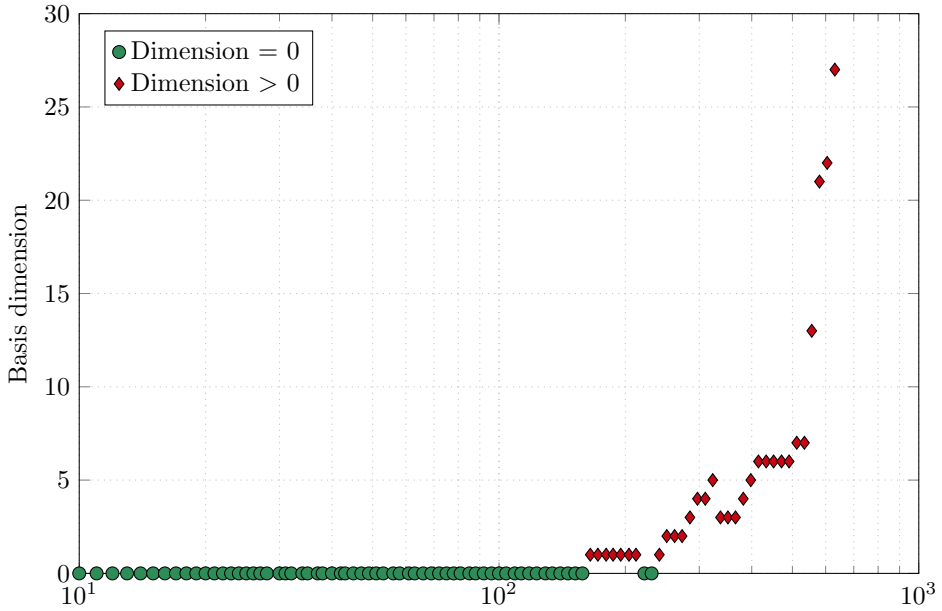


Figure 3.4: Dimension of the intersection basis for various values of $N_{\mathcal{D}}$, with green points indicating dimension zero and red points for dimensions greater than zero.

In Figure 3.4, we apply the first method, where the basis is explicitly computed. Figure 3.4 shows the dimension of the computed basis as a function of $N_{\mathcal{D}}$. Instances with a zero basis dimension, indicating unique solutions, are marked in green, while red points represent cases with a non-zero basis dimension, corresponding to a loss of uniqueness. Notably, the loss of uniqueness observed in Figure 3.3b aligns with these red points, as expected. Although the first method is used here, the same result would be obtained using any of the three methods detailed earlier, as supported by mathematical proof.
Table 3.2 compares the computational costs of the three methods outlined in Section 3.3 for determining the uniqueness of solutions and computing the intersection basis.
The first method explicitly computes the intersection basis, requiring 5763 seconds for both calculations. It relies on dense matrix operations, which demand substantial time and memory. The second method reduces the computation time to 456.51 seconds for the two tasks using principal angles, with

Time (s)	Method 1	Method 2	Method 3
Uniqueness	5763.00	456.51	41.85
Basis	5763.00	456.51	294.77

Table 3.2: Comparison of computational costs for verifying the well-posedness and computing the full intersection basis for different database values $N_{\mathcal{D}}$. The CPU times were measured using the *tic/toc* functions in MATLAB® on an Intel Core i7-12800H processor (2.40 GHz) with 32 GB of RAM.

a tolerance of 10^{-6} to identify zero angles. It also relies on dense matrices and lacks scalability for large-scale problems. The third method uses an iterative approach for computing a vector in the intersection of $\ker D$ and $\ker H$ using alternating projection. If the vector is zero, the solution is unique. Otherwise, there are multiple solutions. The time for uniqueness verification corresponds to the computation of this single vector. To compute the full intersection basis, the method is run repeatedly with different initializations, and the Gram-Schmidt process is applied to ensure linear independence of the resulting vectors. The total time to compute a basis of dimension s is the time required to compute one vector (i.e. a single algorithm run), multiplied by $s+1$, for all $s \geq 0$. By using sparse matrix structures, this third method significantly reduces computational time to 41.85 seconds for uniqueness verification and 294.77 seconds for basis computation.

All three methods produced the same results, although, for large $N_{\mathcal{D}}$, slight differences in the basis were observed with method 3. These differences come from the choice of tolerance, the iteration limit, and the presence of very small positive principal angles, which are nearly zero. Using Theorem 3.3.5, we know:

$$j = \frac{1}{2}\frac{\ln(\text{tol})}{\ln(\cos(\theta_{s+1}))} + \frac{1}{2}\ , \tag{3.29}$$

where $\text{tol} = \|(P_{\mathcal{X}}P_{\mathcal{Y}})^j - P_{\mathcal{X}\cap\mathcal{Y}}\|$. The convergence becomes slower as the smallest positive angle decreases, requiring more iterations to achieve convergence. For example, when the smallest positive angle is nearly zero $\theta_{s+1} \to 0$, the above equation shows that we need infinite iteration $j \to \infty$ to achieve convergence. In this analysis, we applied a tolerance of 10^{-5} for the Gram-Schmidt process. For the alternating projection method, the stopping criteria were either a maximum of 5000 iterations or stagnation indicating convergence.

The last method (alternating projection) is the most effective and efficient with resect to time and memory usage. For example, in larger meshes such as in [28] and [84], methods 1 and 2 cannot be applied due to their high memory requirements. The same discussion on the efficiency of the third method also applies to the new algorithm given in Section 3.4.2.

Leygue *et al.* (2018) [61] conducted a preliminary study on noise influence and showed that DDI is largely robust to displacement field noise due to the

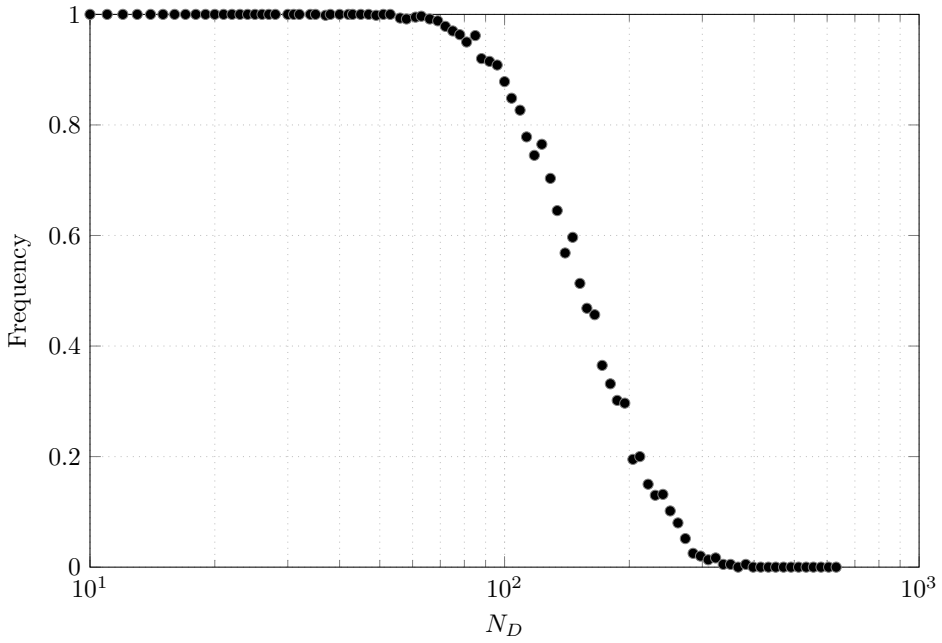


Figure 3.5: Probability estimator of uniqueness (true/false) for different $N_{\mathcal{D}}$ values across 600 different k-means initializations.

averaging effect inherent in clustering measurement. However, it is important to consider the uncertainty that arises from k-means clustering initialization. k-means is usually initialized with k-means++ [6], which provides a good heuristic for initialization. Since k-means++ relies on a random number generator to sample the data distribution, it introduces a natural variability in DDI clustering.

Figure 3.5 illustrates the influence of k-means initializations on the frequency of obtaining a unique solution for different numbers of clusters $N_{\mathcal{D}}$. We use the last method, where the algorithm output is a vector in the intersection basis. A boolean value is defined to indicate "true" if uniqueness is achieved, equivalently when $\sigma^* = 0$, and "false" otherwise. This process is repeated across 600 different clustering for each $N_{\mathcal{D}}$ to calculate the frequency of obtaining a unique solution for various $N_{\mathcal{D}}$. The results reveal three distinct zones. In the low $N_{\mathcal{D}}$ zone, the frequency of uniqueness is equal to one, representing a confidence zone where the problem is well-posed regardless of k-means initializations. In the high $N_{\mathcal{D}}$ zone, the frequency is zero, indicating that the problem has multiple solutions. In between lies a long transition zone, where the chances of finding a unique solution are getting lower and depend on the clustering.

In Figure 3.6, the standard deviation of the basis dimension is analyzed as a function of $N_{\mathcal{D}}$, computed across 100 different k-means initializations. The second method, based on the angles between subspaces, is employed to determine the dimension of the intersection space. For small values of $N_{\mathcal{D}}$, the

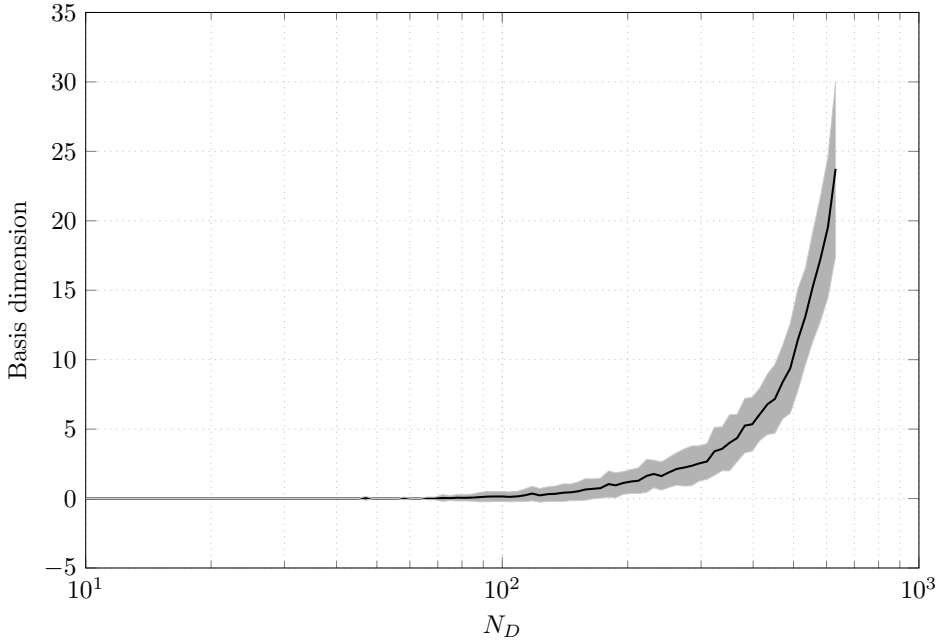


Figure 3.6: Evolution of the average intersection basis dimension as a function of the database size. The shaded area corresponds to the standard deviation of said size.

standard deviation of the basis dimension remains close to zero, indicating minimal variability across initializations and confirming that the problem is well-posed in this range. After a certain threshold, as $N_{\mathcal{D}}$ increases, the standard deviation rises slowly and, the computed basis dimension remains globally small.
As shown in Figure 3.3b, the error decreases as the size of the database increases, up to the point where uniqueness is lost. This leads to the question of whether it is beneficial to correct the solutions beyond this point to gain more accuracy. The goal of Figures 3.5 and 3.6 is to examine the consequences of losing the uniqueness of our solution, how severe the problem is, and determine whether it can be resolved. The result of both figures reveals that the transition zone is wide, and within this region, the dimension is generally small. Moreover, since the difference between two distinct solutions lies in the kernel intersection (as shown in the proof of Proposition 3.2.1), this suggests that the loss of well-posedness can be effectively corrected, for example, by adding a regularization term.

3.6 Conclusion

In this chapter, we present a mathematical and numerical analysis of the robustness of the Data-Driven Identification (DDI) method based on the

preliminary work of Leygue [60], which introduced a criterion for the uniqueness of solutions. The DDI solution is unique if and only if there does not exist a non-zero stress field that is constant per cluster (in $\ker H$) and self-balanced (in $\ker D$) simultaneously.

We evaluate three methods to numerically characterize this criterion: explicit calculation of the basis, examination of the angles between the subspaces, and alternating projection onto the two kernels, along with comprehensive theoretical evaluation of their efficiency. Furthermore, we address the limitations of the current DDI algorithm by introducing a new version based on alternating projection, with a proof of convergence and improved initialization. The numerical conditioning of the problem and the convergence rate have also been discussed.

These methods were tested on synthetic data, and their effectiveness was analyzed, including a comparison of the computational costs. Although all three methods gave similar results, the third method, based on alternating projection, demonstrated the highest computational efficiency in terms of time and memory usage. The DDI method can now be applied more effectively while determining whether the problem has a unique solution and assessing its convergence. This analysis provides a way for selecting the parameter $N_{\mathcal{D}}$, representing the database size We also examined the implications of losing the well-posedness of the problem, which can be identified using the proposed approaches.

Table 3.3: Continuation of Table 2.2: Summary of key contributions to the evolution and development of the DDI method.

Author(s) (Year)	**Main contributions**
Leygue (2025) [60]	• Introduced a novel formulation that depends implicitly on the material states, reformulating the problem as an equilibrium-constrained minimization problem that is easier to analyze theoretically. • Provided the first mathematical results for DDI, including a criterion for the uniqueness of the solution and an error estimate for the identified stress field.

Continued on next page

Table 3.3 – continued from previous page

Author(s) (Year)	**Main Contributions**
Hachem *et al.* (2025) [45]	• Proposed three methods for evaluating the well-posedness criterion, evaluated theoretically and numerically. • Advanced the mathematical understanding of the DDI method through a new implementation strategy, convergence proof, improved initialization, and analysis of numerical conditioning and convergence rate.

Chapter 4

Addressing the ill-posedness in Data-Driven Identification problem

Summary

4.1 Introduction

In this chapter, we turn to the more challenging situation where the DDI problem becomes ill-posed. In this case, the intersection basis is non-trivial, and the solution is no longer unique. This raises key questions: is the ill-posedness fatal, requiring the results to be discarded entirely, or can a meaningful solution still be recovered from the available information?
To address these questions, we first examine the consequences of ill-posedness and show how it can be detected through the structure of the intersection basis. Building on this understanding, we then introduce three correction strategies designed to recover a reliable DDI solution. The effectiveness of these approaches is assessed through comparative numerical evaluations.
The aim of this chapter is therefore to move beyond the well-posed case and provide a systematic framework for handling ill-posedness in DDI, ensuring that meaningful and robust solutions can still be obtained.

We recall the DDI formulation for a given clustering of strains:

$$\sigma = \arg\min \frac{1}{2}\|\sigma\|_H^2 \qquad \text{with} \qquad D\sigma = F \ . \tag{4.1}$$

In the ill-posed case, this problem admits multiple solutions. Any two such solutions differ by a vector belonging to $\ker H \cap \ker D$. In what follows, we aim to characterize this space and we examine how to select a "good" solution among the possible solutions.

4.2 Definitions and notations

Definitions

Definition 4.2.1 (Support of a matrix)**.** *Let $A = [a_{uv}] \in \mathbb{R}^{p\times q}$ be a matrix. The* support *of A, denoted by* $\mathrm{supp}(A)$*, is the set of indices of its nonzero entries:*

$$\mathrm{supp}(A) = \{(u,v) \in \{1,\dots,p\} \times \{1,\dots,q\} \ \mid \ a_{uv} \neq 0\}\,.$$

Thus, the cardinality $|\mathrm{supp}(A)|$ equals the number of nonzero entries of A.

Definition 4.2.2 (Sparse ratio)**.** *Let $A \in \mathbb{R}^{p\times q}$ be a sparse matrix. The* sparse ratio *of A is defined as*

$$\text{sparse ratio}(A) = \frac{\#\{(u,v) \mid a_{uv} = 0\}}{pq} = 1 - \frac{|\,\mathrm{supp}(A)|}{pq}.$$

That is, the proportion of zero entries in the matrix A.

Notations

We consider that $\ker D \cap \ker H$ is computed using the alternating projection algorithm described in Section 3.3.3.

- We denote $\mathcal{K} := \{k_j, \quad j = 1, \ldots, s\}$ a basis of the subspace $\ker D \cap \ker H$, where $s = \dim(\ker D \cap \ker H)$.
  Each vector k_j belongs to $\mathbb{R}^{\ell m}$, with $\ell = 3$ in 2D and $\ell = 6$ in 3D.

- Let $K \in \mathbb{R}^{\ell m \times s}$ be the matrix whose columns are the basis vectors k_j, i.e.,
$$K = [k_1 \mid \cdots \mid k_s] \ .$$

- Let $\boldsymbol{\alpha} \in \mathbb{R}^s$ be the vector collecting the coefficients α_j, i.e.,
$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 & \cdots & \alpha_s \end{bmatrix}^\top \ .$$

- Any vector $\sigma \in \ker D \cap \ker H$ can be written as a linear combination of the basis vectors:
$$\sigma = \sum_{j=1}^{s} \alpha_j k_j = K\boldsymbol{\alpha} \ .$$

4.3 Understanding the intersection basis

"Understanding your problem is half the solution."
In this section, we aim to gain deeper insight into the structure of the intersection basis. We begin with an interpretation that clarifies why it is essential to understand this space and highlights its role in the problem. This is followed by an observation supported by an illustrative example, which further develops our intuition for the basis.

4.3.1 Interpretation

Building on Theorem 3.2.1, we provide a more precise interpretation of the uniqueness property of problem (4.1). The possible non-uniqueness of solutions is confined to the subspace $\mathcal{K}$, while the component in the orthogonal complement $\mathcal{K}^\perp$ is uniquely determined. In other words, the solution of problem (4.1) is unique when restricted to $\mathcal{K}^\perp$.
Let σ_1 and σ_2 be two distinct solutions of problem (4.1). They can be decomposed as

$$\sigma_1 = \sigma_1^{\mathcal{K}} + \sigma_1^{\perp} \ , \qquad \sigma_1^{\mathcal{K}} \in \mathcal{K},\ \sigma_1^{\perp} \in \mathcal{K}^\perp \ , \tag{4.2a}$$

$$\sigma_2 = \sigma_2^{\mathcal{K}} + \sigma_2^{\perp} \ , \qquad \sigma_2^{\mathcal{K}} \in \mathcal{K},\ \sigma_2^{\perp} \in \mathcal{K}^\perp \ . \tag{4.2b}$$

Subtracting (4.2b) from (4.2a), we obtain

$$\sigma_1 - \sigma_2 = (\sigma_1^{\mathcal{K}} + \sigma_1^{\perp}) - (\sigma_2^{\mathcal{K}} + \sigma_2^{\perp}) \ . \tag{4.3}$$

According to Corollary 3.2.2, the difference $\sigma_1 - \sigma_2$ belongs to $\mathcal{K}$, leading to

$$\underbrace{\sigma_1 - \sigma_2}_{\in \mathcal{K}} = \underbrace{(\sigma_1^{\mathcal{K}} - \sigma_2^{\mathcal{K}})}_{\in \mathcal{K}} + \underbrace{(\sigma_1^{\perp} - \sigma_2^{\perp})}_{\in \mathcal{K}^{\perp}} \ . \tag{4.4}$$

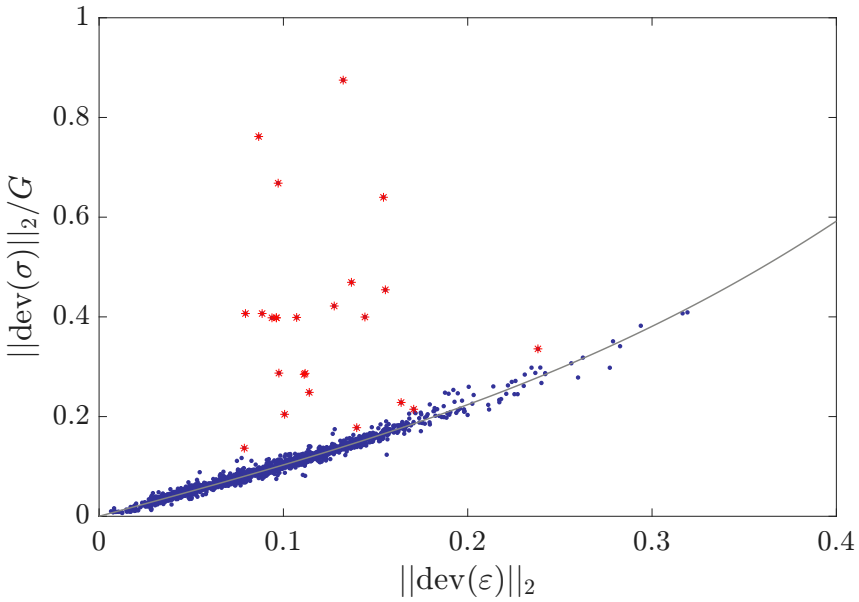
Since the difference lies in $\mathcal{K}$ and the two terms on the right-hand side are orthogonal, we must have $\sigma_1^{\perp} = \sigma_2^{\perp}$.
Therefore, the solution is unique in $\mathcal{K}^{\perp}$, and any modification or correction of the solution must be carried out within the space $\mathcal{K}$.

4.3.2 Observation and discussion

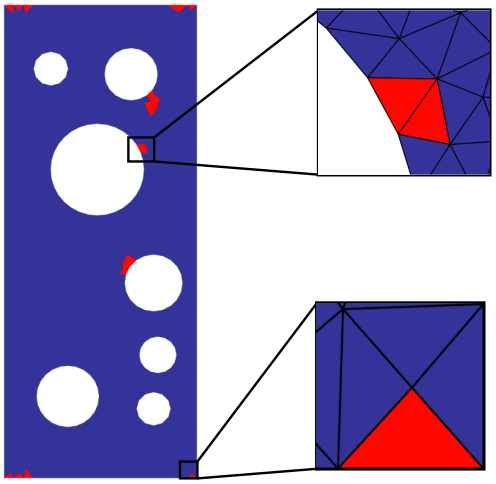
To better understand the subspace $\mathcal{K}$, consider the case $\dim(\mathcal{K}) = 13$, obtained with $N_{\mathcal{D}} = 500$ clusters. The DDI method is applied to the synthetic data presented in Chapter 3, Section 3.5. The intersection basis $\mathcal{K}$ is visualized on the constitutive response in Figure 4.1a, and the corresponding mesh is shown in Figure 4.1b.

In Figure 4.1a, the red stars indicate elements whose indices belong to $\text{supp}(K)$, blue points represent the remaining elements, and the gray curve corresponds to the reference finite element solution. It shows that most elements follow the FE reference response, with only a few elements exhibiting significant deviations. These outliers correspond to the indices in the support of K, i.e., the elements associated with the nonzero entries of K.
This is similar to that observed in Figure 2.7, the present result clarifies that the outliers correspond to the elements associated with the nonzero entries of K.
When mapped back onto the mesh in Figure 4.1b, these elements appear highly localized, predominantly near the upper and lower boundaries or around the holes, and are highlighted in red. The same color convention is used here: red denotes indices in $\text{supp}(K)$, while blue corresponds to the remaining elements. This localization can be attributed to the reduced number of constraints associated with these regions. Specifically, elements located near the boundaries are connected to nodes with fewer contributing stress components, due to the absence of neighboring elements on one or more sides (as illustrated in the zoom-in of Figure 4.1b). As a result, the corresponding equations are less constrained compared to those in the interior.

This observation indicates that the matrix K is sparse, in this example: it exhibits a sparsity ratio of approximately 0.99 at the element level and 0.96 at the cluster level. Although $\dim(\mathcal{K}) = 13$, the effect is not distributed over all elements only 24 out of 2416 elements, equivalently, only 18 out of 500 clusters



(a)



(b)

Figure 4.1: Illustration of the nonzero entries of the intersection basis: (a) representation in the constitutive space and (b) corresponding representation in the mesh.

are involved.
These few elements are responsible for the significant error, as they show the largest deviations from the reference solution. Importantly, they can be explicitly and reliably identified.

We refer to the elements with indices in $\mathrm{supp}(K)$ as *problematic elements*, and to their associated clusters as *problematic clusters*.

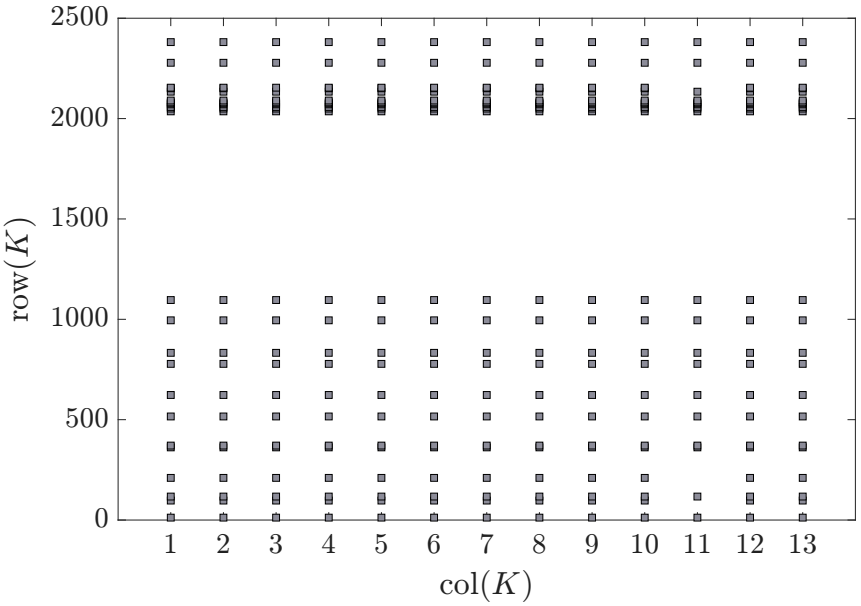


Figure 4.2: Visualization of the matrix K. Grey squares indicate the nonzero entries of K. The horizontal and vertical axes correspond to the columns $\mathrm{col}(K)$ and rows $\mathrm{row}(K)$, respectively. This representation highlights the common support of the vectors $\{k_j\}_{j=1}^{13}$.

Notably, when the alternating projection algorithm is applied starting from random initial points (or any full-vector initialization), the resulting vectors in $\mathcal{K}$ exhibit the same sparsity pattern, as illustrated in Figure 4.2 and previously discussed in Section 3.3.3. This can be explained by noting that the components belonging to $\mathcal{K}^{\perp}$ converge to zero under the iterations, while the remaining components persist and form a set of linearly independent vectors. Consequently, the support of the nonzero entries is preserved across all vectors generated by the alternating projection algorithm. More precisely, the support of the set of linearly independent vectors $\{k_j\}_{j=1}^{s}$ satisfies

$$S_{\mathrm{supp}} := \mathrm{supp}(k_r) = \mathrm{supp}(k_h) \quad \forall r, h \ .$$

As a result, to identify the active support, it is sufficient to compute a single non-zero vector in the intersection basis. There is no need to construct the

full basis in order to detect the problematic elements and clusters.
It can be concluded that there exists an inverse relationship between the sparsity ratio and the dimension of $\mathcal{K}$: higher-dimensional intersection spaces tend to involve a greater number of problematic elements/clusters (see also discussion in Appendix C). In particular, increasing the number of clusters $N_{\mathcal{D}}$ typically leads to an increase in the dimension of the intersection space $\mathcal{K}$. When $N_{\mathcal{D}}$ approaches the number of elements m, this growth can lead to a significant loss of sparsity in the vectors spanning $\mathcal{K}$, as more elements become involved in the support.

Now that the problem has been identified and the structure of the intersection basis has been characterized, we can proceed to formulate a correction to the solution.

4.4 How to correct DDI solution?

In this section, we present three approaches for improving the DDI solution: (4.4.1) the naive approach, (4.4.2) the database pruning approach, and (4.4.3) adding regularization using the basis $\mathcal{K}$.
Other methods can be considered, but we focus on these three.

4.4.1 Naïve approach: Eliminate the problematic elements

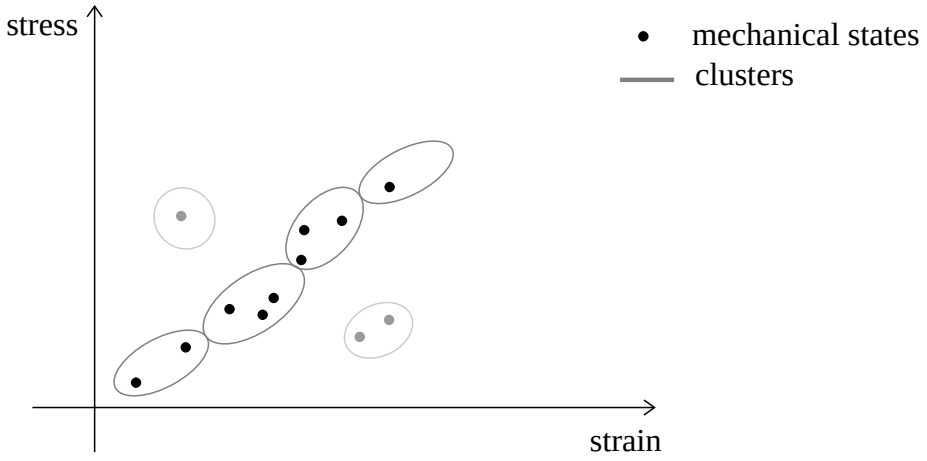


Figure 4.3: Visualization of the method. Points with higher opacity (darker markers) are considered to represent the material response.

In the case where the intersection space $\mathcal{K}$ has small dimension, the associated vectors are typically sparse. A straightforward correction strategy is

to eliminate the problematic elements entirely, which yields uniqueness in the -reduced- new space, in other words, leading to $\sigma_1^{\perp} = \sigma_2^{\perp}$ (see Section 4.3.1). Consequently, the elimination of problematic elements implies the removal of their associated clusters, i.e. the problematic clusters. An illustration of this procedure is provided in Figure 4.3.
This approach is easy to implement and can be effective when the number of problematic elements is small i.e. when a sparsity criterion is satisfied on the vectors in $\mathcal{K}$. For example, one may require the sparsity ratio to exceed 0.8, meaning at least 80% of the entries of a vector in $\mathcal{K}$ are zero.
The main advantages of this approach are its simplicity and low computational cost. However, despite its practicality, this method is not always optimal, as it modifies the original solution space and eliminates data close to boundaries that might be of interest. Moreover, as the dimension of $\mathcal{K}$ increases, the number of problematic elements typically grows. In such cases, this approach becomes less suitable, as it may lead to excessive information loss and a degradation of solution accuracy.

4.4.2 Database pruning: Eliminate the problematic clusters

Instead of working directly with the degrees of freedom, as in the naïve approach, the centroids used in the k-means clustering can be modified. The degrees of freedom that have nonzero entries in the vector basis are associated with certain clusters, which are referred to here as problematic clusters.
The idea is that for the elements belonging to these clusters, the assignment is redirected from the nearest centroid to the second-nearest centroid, thereby reducing the overall number of clusters. A difficulty arises when the second-nearest centroid also corresponds to a problematic cluster. In such cases, the element would need to be reassigned to the third-nearest centroid, and so forth. To avoid this iterative reassignment process, a more systematic strategy consists of eliminating all the problematic clusters altogether. The elements originally associated with these clusters are then naturally reassigned to their next nearest available centroids within the updated clustering, while the assignments of all remaining elements are preserved from the original k-means clustering. This procedure is illustrated in Figure 4.4.
The *database pruning* procedure can be summarized as follows:

1. Eliminate all problematic database entries (problematic clusters).
2. Reassign the elements that belong to a removed cluster to their nearest remaining centroid (via k-NN search).
3. Update the pairing matrix P accordingly.
4. Perform DDI using the pruned database (with the updated P).

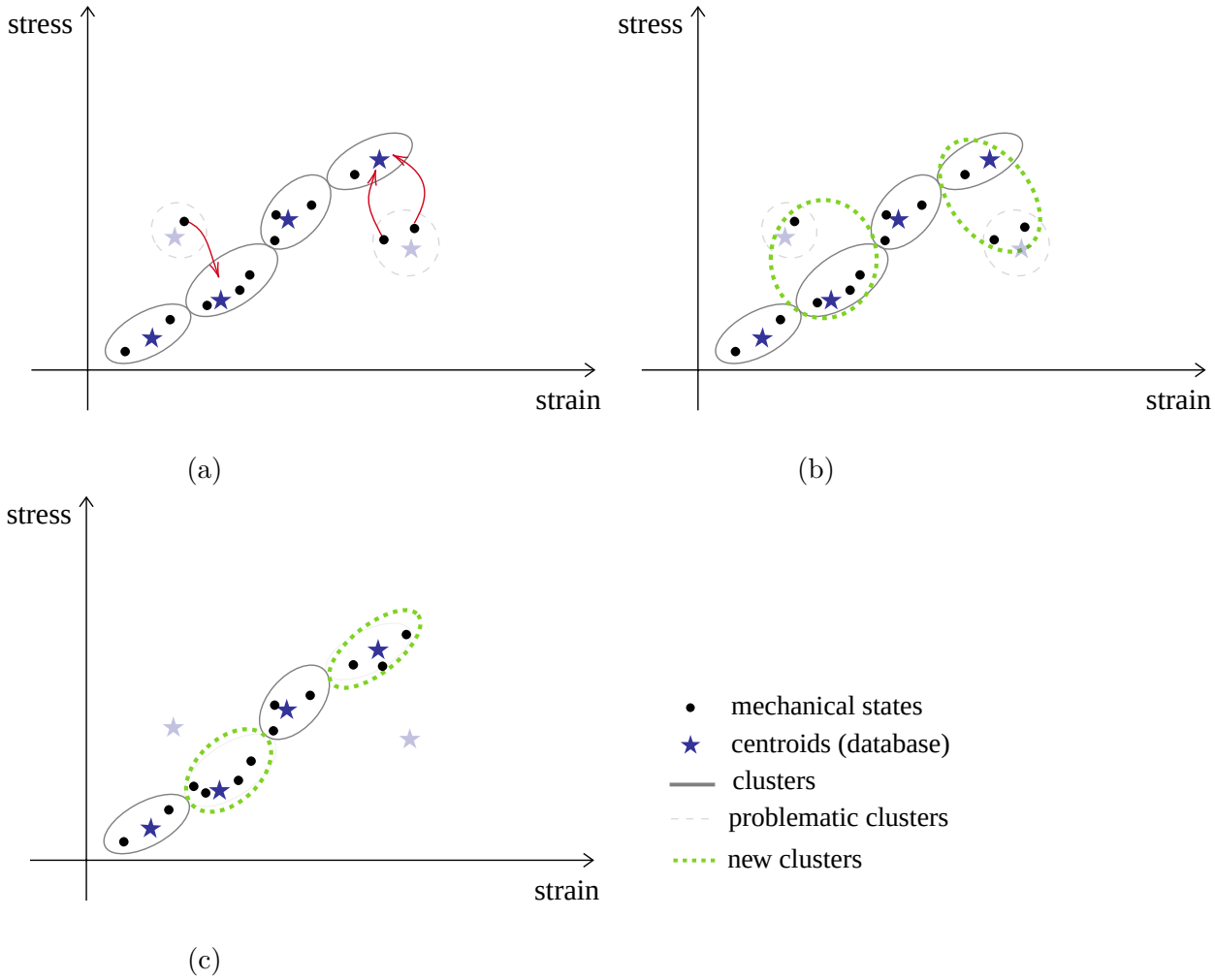


Figure 4.4: Visualization of the database pruning method. The grey dashed clusters represent the problematic clusters that are eliminated, the red arrows indicate the reassignment of their elements to the next nearest centroid and the green dotted line denote the new clusters. (a) Database pruning. (b) Reassignment to the nearest centroid. (c) DDI with the updated database.

The second step can be regarded as an iteration of the k-means algorithm. No re-initialization of centroids is required, only the distances are recomputed in order to update the assignments.

4.4.3 Additional regularization

In cases where the problem admits multiple solutions, it is natural to introduce a regularization term. Following the framework of the pseudo-inverse [77, 93] and the weighted pseudo-inverse [37], we select the solution with the minimum (semi-)norm among all possible solutions. Mathematically, this can be expressed as

$$\begin{aligned} \text{solution} = \arg\min_{\sigma} \ & \frac{1}{2}\|\sigma\|_N^2 \\ \text{subject to} \quad & \sigma \text{ is a solution of problem (4.1)} , \end{aligned} \tag{4.5}$$

where $\|\sigma\|_N^2 := \sigma^T N \sigma$, and N to be specified later.

The solution set can be explicitly determined and is given by the following proposition.

Proposition 4.4.1. *Let σ_s be one of the possible solutions of problem (4.1). Then, the solution set S of the DDI problem (4.1) is given by:*

$$S := \{\sigma \mid \sigma \text{ is a solution of problem (4.1)}\} \tag{4.6a}$$
$$= \{\sigma \mid \sigma = \sigma_s + K\boldsymbol{\alpha}\} . \tag{4.6b}$$

Proof. The proof is relatively simple, but we will present it in detail. The equivalence is established by proving both inclusions.

1. (4.6a) $\subseteq$ (4.6b): We want to proof that any solution σ to problem (4.1) can be written as $\sigma = \sigma_s + K\boldsymbol{\alpha} = \sigma_s + \sum_{j=1}^{s} \alpha_j k_j$.
   Let σ be another solution to the problem. Then, the difference

   $$v := \sigma - \sigma_s$$

   lies in $\ker D \cap \ker H$. Consequently, $v \in \operatorname{span}\{k_1, \ldots, k_s\}$, and $\sigma = \sigma_s + \underbrace{K\boldsymbol{\alpha}}_{v}$ for some $\boldsymbol{\alpha} \in \mathbb{R}^s$.

2. (4.6a) $\supseteq$ (4.6b): Consider $\sigma = \sigma_s + K\boldsymbol{\alpha}$ for some $\boldsymbol{\alpha} \in \mathbb{R}^s$. Since $k_j \in \ker D$ for all j,

   $$D\sigma = D\sigma_s + DK\boldsymbol{\alpha} = F + 0 = F.$$

Moreover, as $k_j \in \ker H$,

$$\sigma^\top H\sigma = (\sigma_s + K\boldsymbol{\alpha})^\top H(\sigma_s + K\boldsymbol{\alpha}) = \sigma_s^\top H\sigma_s.$$

Thus, σ attains the same objective value as σ_s, which is minimal by assumption, and therefore belongs to the solution set.

□

Using the Proposition 4.4.1, the regularization problem (4.5) can be reformulated as:

$$\arg\min_{\sigma\in S} \frac{1}{2}\|\sigma\|_N^2 \;=\; \arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^s} \frac{1}{2}\,\|\sigma_s + K\boldsymbol{\alpha}\|_N^2 \;. \tag{4.7}$$

This reduced formulation shows that, rather than optimizing over the entire solution set S, it suffices to determine the coefficients $\boldsymbol{\alpha}$ associated with the basis vectors of $\ker D \cap \ker H$. Consequently, the effective dimension of the optimization problem is therefore $s = \dim(\ker D \cap \ker H)$, which considerably simplifies its resolution.
The solution to problem (4.7) is given by:

$$K^\top NK\,\boldsymbol{\alpha} = -\,K^\top N\sigma_s \;. \tag{4.8}$$

For the choice of metric $\|\cdot\|_N$:

- $N = I$: this corresponds to the standard Euclidean norm $\|\cdot\|_2$, yielding the classical pseudo-inverse solution [93]. By construction, this pseudo-inverse solution is unique.
  This formulation is also known in the literature as the *minimum-norm least-squares* problem [20, 41].

- $N = H_1$: here $\|\cdot\|_{H_1}$ denotes the DDI semi-norm. The matrix H_1 encodes a coarser clustering into N_{new} clusters, chosen such that $N_{\text{new}} < N_{\mathcal{D}}$.
  Since H_1 is positive semi-definite, the matrix $K^\top H_1 K$ is not necessarily invertible. To ensure existence and uniqueness of the solution, the following condition is imposed:

  $$\ker(H_1) \cap \mathrm{Im}(K) = \{0\} \;. \tag{4.9}$$

  This condition requires that the two clusterings encoded by H_1 and H be incompatible. Equivalently, no vector representing a cluster in H may also belong to the clustering structure defined by H_1.
  Under this condition, $K^\top H_1 K$ is positive definite on the relevant subspace, guaranteeing a unique solution.

- Hybrid regularization: to enforce the condition above in Eq. (4.9), a novel approach referred to as *hybrid regularization* is introduced.
  This strategy replaces the original semi-norm induced by the positive

semi-definite matrix H_1 with a modified parameter-dependent norm defined by the matrix $H_1 + \gamma I$, where I is the identity matrix and $\gamma \geq 0$. The corresponding optimization problem is

$$\arg\min_{\boldsymbol{\alpha},\,\gamma} \ \frac{1}{2} \left\| \sigma_s + K\boldsymbol{\alpha} \right\|^2_{H_1+\gamma I} \ . \tag{4.10}$$

For $\gamma = 0$, the formulation reduces to the H_1 semi-norm problem, while for large γ, the contribution of I dominates, recovering a formulation close to the Euclidean norm. This hybrid construction ensures positive definiteness of $H_1 + \gamma I$ whenever $\gamma > 0$, thereby guaranteeing the existence and uniqueness of the solution.

4.5 Results and discussion

In this section, we apply the DDI method to the same synthetic data described in Chapter 3, Section 3.5.

Our objective is to assess the ill-posedness inherent in the DDI formulation and to evaluate three correction strategies introduced in Section 4.4. These strategies are tested for both small and large dimensions of the intersection basis, with performance assessed through visual and quantitative comparisons. To quantify the error between the stress field obtained by DDI or its corrected variants and the reference solution, we employ the Mean Squared Error (MSE). For a set of m elements, the MSE is defined as:

$$\text{error} = \frac{1}{m} \sum_{e=1}^{m} \|\sigma_e^{\text{FE}} - \sigma_e^{N}\|_2^2 \ , \tag{4.11}$$

where, σ_e^{FE} denotes the reference solution at element e, and σ_e^{N} is the identified stress at element e, obtained either from the initial DDI solution or from one of the corrected formulations. The MSE is reported at two levels:

- globally: over the entire domain, and
- locally: restricted to elements whose indices lie within supp(K).

For all applications, method 3 is applied with $N_{\text{new}} = 100$ and $\gamma = 10^{-3}$. The choice of $N_{\text{new}} = 100$ for the regularization ensures the condition $N_{\text{new}} < N_{\mathcal{D}}$ is satisfied. Since the loss of uniqueness was first observed to occur after $N_{\mathcal{D}} \approx 100$, this represents a safe and efficient choice.

4.5.1 Evaluation for low dimension basis

In figure 4.5, we take the same example presented in Section 4.3.2 where the DDI solution was shown to be non-unique and the basis dimension is small,

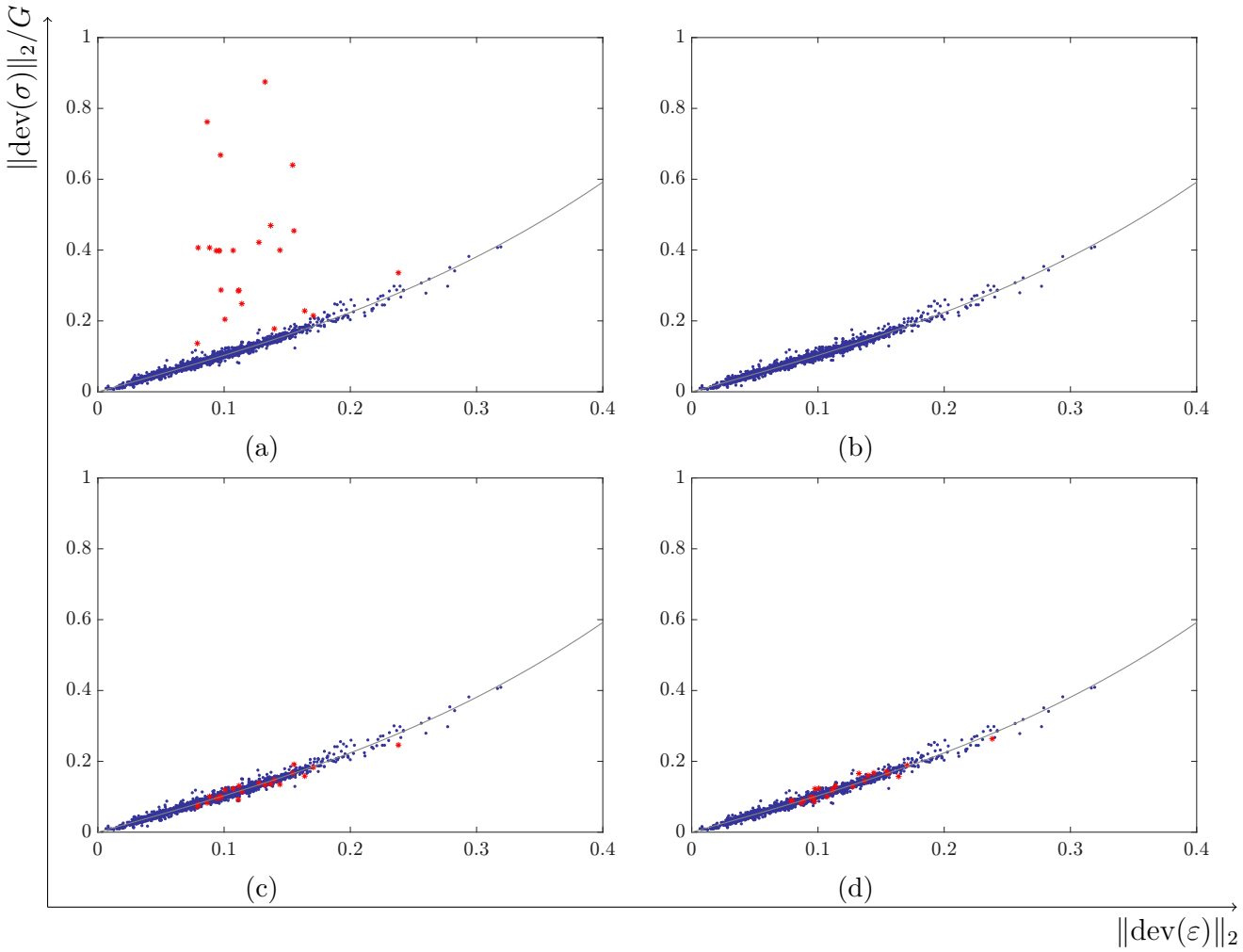


Figure 4.5: Comparison of correction methods for $\dim(\ker D \cap \ker H) = 13$ with $N_{\mathcal{D}} = 500$: (a) uncorrected DDI solution, (b) naïve approach, (c) database pruning approach, (d) proposed regularization method.

equal to 13. In all these figures, red stars indicate elements whose indices belong to supp(K), blue points represent the remaining elements, and the gray curve corresponds to the reference finite element solution. The three correction methods presented in Section 4.4 are evaluated. We revisit the result of the uncorrected DDI solution in Figure 4.5a.
In Figure 4.5b, the naive approach is applied, which consists of excluding elements that disturb the intersection of the two kernels (i.e., the red stars). Given that the sparsity ratio is high (0.99), it is safe to use this method. More precisely, only 24 out of 2416 was filtered. The figure illustrates the theoretical result established in Section 4.3.1: the DDI solution remains unique in $\mathcal{K}^{\perp}$.
In figure 4.5c, the database pruning method is employed, where it eliminates problematic clusters and their associated elements are merged with their new nearest cluster. In this case, 18 problematic clusters were removed, yielding a new database $\mathcal{D}$ of size 482. Thus, 482 material states are retained (instead of the original 500) to sample the mechanical response. The result shows improved alignment of the problematic elements (red stars) with the reference solution.
In Figure 4.5d, we apply the third method, and apply the hybrid regularization using $N_{\text{new}} = 100$ clusters and a regularization parameter of $\gamma = 10^{-3}$. The corrected solution aligns closely with the reference solution, despite using a relatively small number of clusters.

Error	**DDI**	**Method 1**	**Method 2**	**Method 3**
Global MSE	3.52×10^{-3}	2.11×10^{-4}	2.26×10^{-4}	2.23×10^{-4}
Local MSE	3.34×10^{-1}	–	1.49×10^{-3}	1.46×10^{-3}

Table 4.1: Error comparison of the corrected solution using MSE.

Despite varying levels of complexity and computational cost, all three correction strategies lead to significant improvements over the initial DDI result. These improvements are shown quantitatively in Table 4.1, which provides a comparison of the corrected solutions using mean squared error (MSE). Both the global error and the error restricted to elements $e \in \text{supp}(\mathcal{K})$ are reported.

The uncorrected DDI solution shows a relatively high global error (3.52×10^{-3}) and localized error (3.34×10^{-1}), consistent with the expectation that ill-posedness primarily affects elements that disturb the intersection of the two kernels, as established in Section 4.3.1. This error is expected and can be observed in the distance of the red stars from the reference solution in Figure 4.5a. Although the naïve method (Method 1) reports a lower global error (2.11×10^{-4}), this method excludes problematic elements from the initial space and is therefore not directly comparable; instead, it confirms that the ill-posedness only affects a small number of elements. Both the database

pruning method (Method 2) and the regularization approach (Method 3) achieve significant reductions in global error (2.26×10^{-4} and 2.23×10^{-4}, respectively) and localized error (1.49×10^{-3} and 1.46×10^{-3}, respectively), confirming their effectiveness in improving the precision of the DDI solution.

4.5.2 Evaluation for high dimension basis

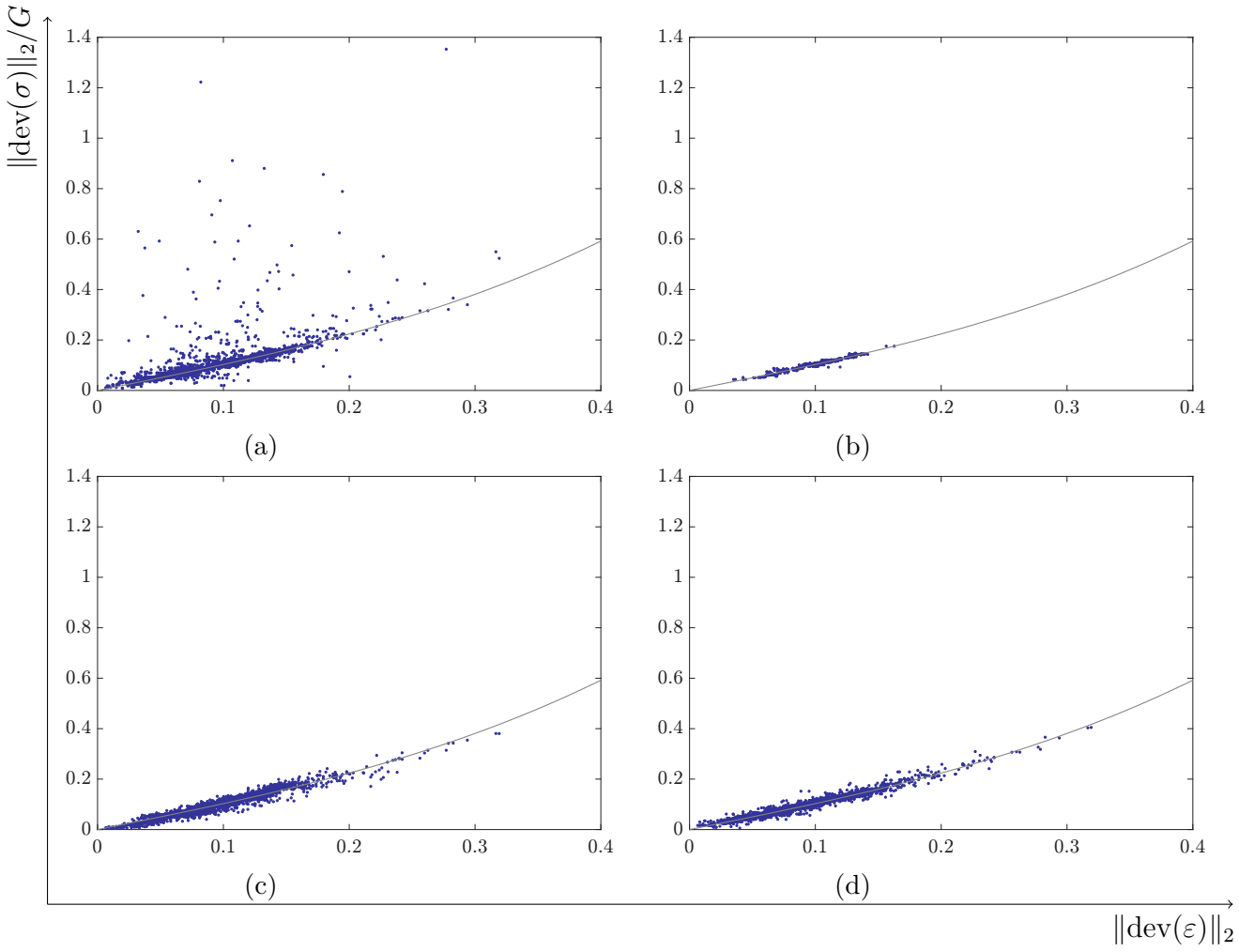


Figure 4.6: Comparison of correction methods for $\dim(\ker D \cap \ker H) = 127$ with $N_{\mathcal{D}} = 800$: (a) uncorrected DDI solution, (b) naïve approach, (c) database pruning approach, (d) proposed regularization method.

Figure 4.6 presents the results of the correction methods applied to a larger basis dimension. Specifically, the DDI solution is computed with $N_{\mathcal{D}} = 800$, yielding a basis $\mathcal{K}$ of dimension 127. In this case, the number of problematic elements increases substantially to 1980 out of 2416, which are grouped into 708 clusters out of 800.
As illustrated in Figure 4.6a, the DDI solution deviates significantly, producing

large deviations from the reference curve. As discussed earlier in Section 4.3.2, increasing $N_{\mathcal{D}}$ leads to a higher number of problematic elements and clusters. In Section 4.4.1, we noted that the naïve approach is recommended when the sparsity ratio of $\mathcal{K}$ is greater than or equal to 0.8. In the present case, the sparsity ratio is approximately 0.18, meaning that 82% of the elements are removed. This indicates that the method is not suitable. Nevertheless, for the sake of comparison and to illustrate its behavior, the corresponding result is included in Figure 4.6b.
In contrast, the database pruning method (Figure 4.6c) continues to provide meaningful corrections. The regularization approach (Figure 4.6d), applied with $N_{\text{new}} = 100$ and $\gamma = 10^{-3}$, also remains highly accurate, maintaining close alignment with the reference solution.

Error	**DDI**	**Method 1**	**Method 2**	**Method 3**
Global MSE	1.35×10^{-2}	1.54×10^{-4}	6.44×10^{-4}	4.33×10^{-4}
Local MSE	1.56×10^{-2}	–	6.92×10^{-4}	4.75×10^{-4}

Table 4.2: Error comparison of the corrected solution using MSE.

The quantitative performance of the correction methods in this case is summarized in Table 4.2, providing localized error metrics corresponding to the results shown in Figure 4.6.
For the uncorrected DDI solution, the global MSE and local are relatively large, confirming the significant deviation observed in Figure 4.6a. The naive method formally yields a low global MSE, as it is unique in this restricted space. However, it is not suitable in this case due to the very low sparsity ratio, approximately 0.18, far below the recommended threshold of 0.8. Both the database pruning (Method 2) and the regularization (Method 3) methods yield substantial improvements. Method 2 achieves a global MSE of 6.44×10^{-4} and a local MSE of 6.92×10^{-4}, while Method 3 provides the lowest errors, with a global MSE of 4.33×10^{-4} and a local MSE of 4.75×10^{-4}. These results confirm that both the database pruning and regularization methods are effective in addressing ill-posedness, even for high-dimensional bases.

4.5.3 Performance for different intersection basis dimensions

Figure 4.7 presents the evolution of the mean squared error as a function of the database size $N_{\mathcal{D}}$. For comparison, we reuse the same values presented in Chapter 3, Figure 3.3 to assess whether the methods can successfully surpass the threshold. We recall that, for this range of database sizes, the highest intersection-base dimension is 32, which yields 133 problematic elements out

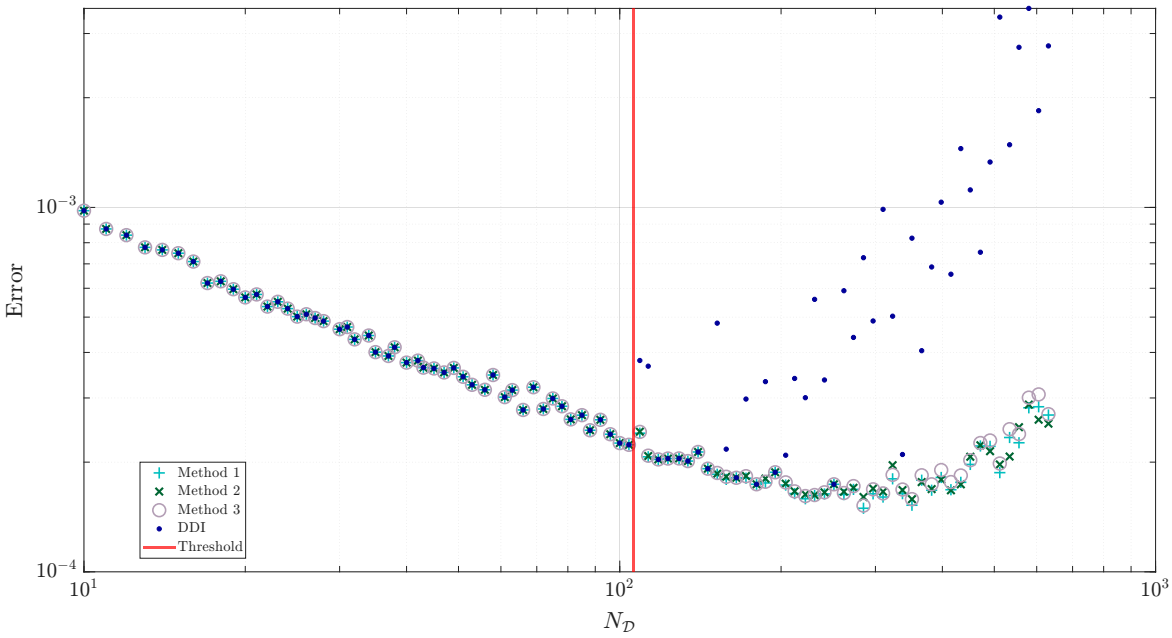


Figure 4.7: Comparison of the error for different values of $N_{\mathcal{D}}$. The vertical red line indicates the threshold beyond which uniqueness is lost.

of 2416, corresponding to a sparsity ratio of 0.94. Therefore, it is safe to apply the first method for all values of $N_{\mathcal{D}}$.

It can be observed that all three correction strategies significantly improve upon the uncorrected DDI solution obtained with random initialization. Notably, all three approaches successfully extend beyond the threshold, with the error continuing to decrease as $N_{\mathcal{D}}$ increases. This behavior demonstrates that these methods recover a solution in the region where the original DDI approach fails. However, a new threshold emerges around $N_{\mathcal{D}} \approx 200$, beyond which the error begins to increase again, indicating that the DDI method has reached its intrinsic limitations.

Since the error curve initially decreases, reaches a minimum, and then rises again, this suggests that adding more data can continue to reduce the error and allows to add more clusters in the database. Such additional data can come from experiments, for example by introducing more measurement points, using more heterogeneous samples or capturing additional snapshots. Alternatively, it can be generated through numerical improvements, such as refining the mesh in simulations or through data augmentation, such as applying rigid rotations to the data.

To illustrate this, in Figure 4.8, we repeat the analysis using only correction method 2, with a refined mesh in which the element size is reduced by a factor of four (9204 elements, 4866 nodes). The results clearly show that with this richer data, the error continues to decrease beyond the threshold observed in the first mesh. Furthermore, under correction method 2, the error reduction

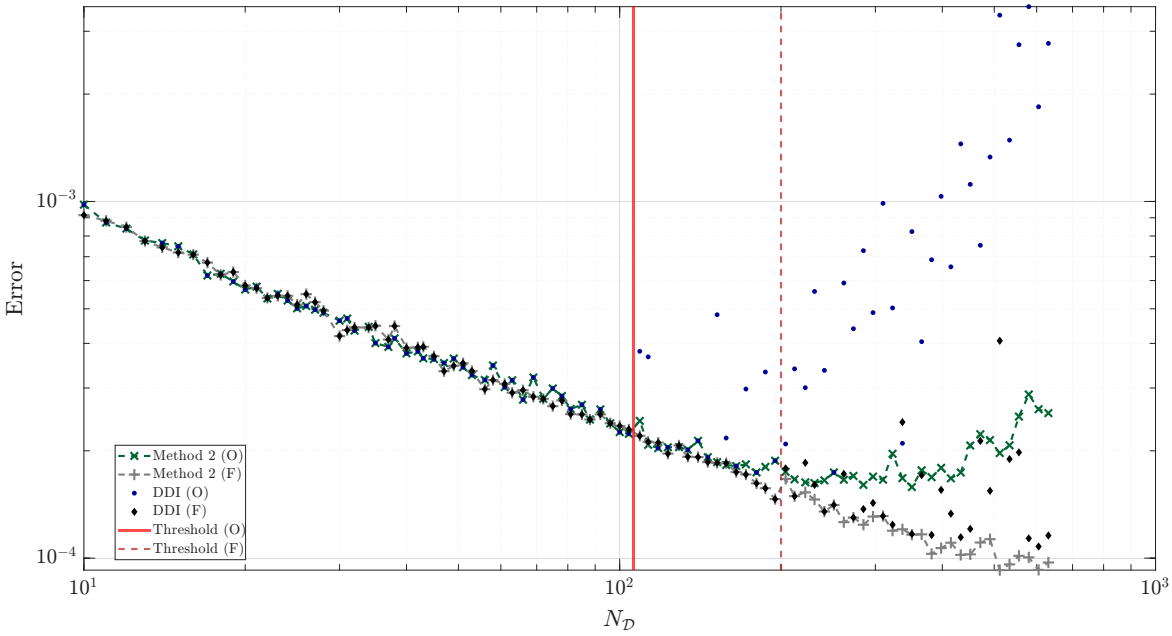


Figure 4.8: Comparison of the error with respect to different values of $N_{\mathcal{D}}$. Results labeled with (O) correspond to the original mesh, while those labeled with (F) correspond to the fine mesh.

persists well past the previous limit, confirming the benefit of expanding the data quality and size.

4.5.4 Discussion

The naive approach requires only a vector in $\mathcal{K}$ to detect and filter problematic elements, making it computationally inexpensive and straightforward to implement. It is particularly effective when the basis dimension is small, i.e., when the sparsity ratio is high.
This approach has already been applied empirically in previous DDI applications, for example:

- In Madeira *et al.* [64]: *"low occurrence values are filtered by a* $10^{-4.5}$ *threshold"*.
- In Costecalde [27]: *"The plane section representing a number of points corresponding to a hundredth of the point number of the densest area or less are removed. Only 4% of the data is removed."*

However, in these studies, the approach was used without fully understanding why these points appear in some cases and not in others, what they represent, or how to systematically detect them.

The database pruning method is slightly more computationally expensive, as it requires solving an additional DDI problem with the updated clustering database, as well as another one to verify uniqueness. Despite this added cost, the approach remains practical and does not require any modifications to the original mesh. Moreover, it can still be applied even if the sparsity of the vectors is lost, as long as there are enough clusters to perform the new DDI problem with the updated clustering.

The last method, the additional regularization based on the pseudo-inverse framework, is the most computationally intensive. The computational cost is controlled by the size of the intersection basis $\ker D \cap \ker H$.
Specifically, it requires $s+1$ DDI solves to compute the basis, with $s = \dim(\mathcal{K})$ (see Chapter 3, Section 3.5), followed by solving Equation (4.8), whose cost also depends on s. Alternatively, the basis can be obtained from a single non-trivial vector, as discussed in Section 4.3.2, but this approach becomes increasingly expensive as the number of unstable clusters grows, i.e. as $N_{\mathcal{D}}$ increase. Despite its higher computational demand, this method provides the strongest theoretical robustness.

Since the dimension of the basis $\mathcal{K}$ is not known in advance and the sparsity of the elements or clusters can be found from one vector, the second method stands out as a practical and computationally efficient strategy. It requires relatively low computational effort, preserves the original element space, and does not introduce any additional parameters.
The results also confirmed that utilizing richer data set is an effective strategy for improving accuracy, as expected (previously discussed in Chapter 2, Figure 2.3). The effectiveness of the DDI method, therefore, critically depends on the availability of heterogeneous and sufficiently rich data.

4.6 Conclusion

The chapter addressed the ill-posedness of the Data-Driven Identification problem, which arises when $\ker D \cap \ker H$ is non-trivial. We first showed that the problem can be detected and analyzed through the study of the intersection basis, providing a deeper understanding of its structure. Based on this analysis, three correction strategies were proposed:

- a naïve elimination of problematic elements, effective when the intersection basis is sparse,

- a database pruning approach, which eliminates problematic clusters and reassigns their elements, proving to be both practical and efficient,

- a regularization strategy based on pseudo-inverse formulations, the most robust but also the most computationally demanding.

Through numerical experiments, it was shown that all three methods enhance accuracy of the solution compared to the uncorrected DDI. Among them, the database pruning approach proved to be the most effective, combining moderate computational cost with the ability to recover a solution without the need for additional parameters. Furthermore, enriching the data set, whether by collecting more experimental data or using higher-resolution simulations, was shown to improve solution robustness and precision. However, determining the optimal conditions for obtaining such efficient data remains an open research question.

This chapter also addressed the questions formulated in Chapter 2, advancing the understanding of the DDI problem, clarifying the outlier points, while also going beyond the threshold, into the regime characterized by the loss of uniqueness.

Chapter 5

From theory to practice: Guidelines for effective DDI implementation

Summary

5.1 Introduction

The present chapter turns from theoretical analysis to the practical implementation of the Data-Driven Identification (DDI) method. Building on the mathematical analysis and algorithmic contributions established previously, we now aim to formalize the insights gained into a coherent framework for effective and reproducible application.
The chapter begins with a presentation of Mandel notation, which offers a convenient representation of tensorial quantities in linear algebra form while preserving their fundamental properties. Particular emphasis is placed on the formulation of recommended practices and the integration of both existing and newly developed components into a computational approach. To make the methodology accessible, the chapter also presents a step-by-step guide covering initialization, clustering, projectors, convergence acceleration, and efficient basis computation.
In addition to methodological details, the chapter includes an application study that illustrates the role of convergence acceleration, particularly through Anderson acceleration, and discusses computational aspects such as clustering and database construction. The results highlight the efficiency gains achievable with appropriate acceleration schemes.
Through this combination of theoretical clarification and algorithmic guidance, the chapter aims to provide a comprehensive reference for the implementation of DDI in realistic computational settings, and provides a foundation for future research and applications of DDI.

5.2 Mandel notation

Mandel notation [65] (also known as *Kelvin notation*) is a convention that represents tensors in matrix or vector form by appropriately rescaling their components. This allows tensor operations to be expressed using linear algebra. Alternative notations exist, such as *Voigt notation* [66]; however, Mandel notation is preferred here because it preserves the duality of inner products (tensor inner products coincide with standard Euclidean inner products for any combination of strain and stress tensors).
In three dimensions, the index mapping between tensor indices ij and the Mandel representation index I is given by

$$\begin{array}{rccccccc} ij = & 11 & 22 & 33 & 23,32 & 13,31 & 12,21 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ I = & 1 & 2 & 3 & 4 & 5 & 6 \end{array}$$

and the scaling factor α is defined as:

$$\alpha(i,j) = \begin{cases} 1 & i = j \ , \\ \sqrt{2} & i \neq j \ . \end{cases} \tag{5.1}$$

Based on this, the second order tensor and fourth order tensor are given as follows.

Second-order tensor

Let $\boldsymbol{\sigma} \in \mathbb{R}^{3\times3}_{\text{sym}}$ be a symmetric second-order tensor. Its Mandel vector representation is

$$\sigma_I^M = \alpha(i,j)\sigma_{ij} \ , \tag{5.2}$$

with the component ordering indicated by the red arrows in the mapping above: the three diagonal entries appear first, followed by the three shear components (scaled by $\sqrt{2}$).

$$\boldsymbol{\sigma} \ = \ \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ & \sigma_{22} & \sigma_{23} \\ & & \sigma_{33} \end{bmatrix} \in \mathbb{R}^{3\times3} \ .$$

Equivalently, the resulting Mandel representation is:

$$\sigma^M = \begin{bmatrix} \sigma_{11} & \sigma_{22} & \sigma_{33} & \sqrt{2}\sigma_{23} & \sqrt{2}\sigma_{31} & \sqrt{2}\sigma_{12} \end{bmatrix}^\top \in \mathbb{R}^{6\times1} \ .$$

Forth-order tensor

Let $\mathbb{C} \in \mathbb{R}^{3\times3\times3\times3}$ satisfy the minor symmetries $C_{ijkl} = C_{jikl} = C_{ijlk}$. Its Mandel representation is the matrix $C^M \in \mathbb{R}^{6\times6}$,

$$C_{IJ}^M = \alpha(i,j)\alpha(k,l)C_{ijkl} \ , \tag{5.3}$$

where $(ij) \mapsto I$, $(kl) \mapsto J$ follow the prescribed ordering. Equivalently,

$$C^M = \begin{bmatrix} C_{1111} & C_{1122} & C_{1133} & \sqrt{2}\,C_{1123} & \sqrt{2}\,C_{1131} & \sqrt{2}\,C_{1112} \\ & C_{2222} & C_{2233} & \sqrt{2}\,C_{2223} & \sqrt{2}\,C_{2231} & \sqrt{2}\,C_{2212} \\ & & C_{3333} & \sqrt{2}\,C_{3323} & \sqrt{2}\,C_{3331} & \sqrt{2}\,C_{3312} \\ & & & 2\,C_{2323} & 2\,C_{2331} & 2\,C_{2312} \\ & & & & 2\,C_{3131} & 2\,C_{3112} \\ & & & & & 2\,C_{1212} \end{bmatrix} \in \mathbb{R}^{6\times6} \ .$$

In 2D the definition is the same, except indices are restricted to $\{1, 2\}$. The index mapping is

$$ij = 11,\ 22,\ 12 \text{ or } 21 \quad \longmapsto \quad I = 1,\ 2,\ 3\ .$$

Thus,

$$\sigma^M = \begin{bmatrix}\sigma_{11} & \sigma_{22} & \sqrt{2}\,\sigma_{12}\end{bmatrix}^\top \in \mathbb{R}^{3\times 1}, \qquad C^M = \begin{bmatrix} C_{1111} & C_{1122} & \sqrt{2}\,C_{1112} \\ & C_{2222} & \sqrt{2}\,C_{2212} \\ & & 2\,C_{1212} \end{bmatrix} \in \mathbb{R}^{3\times 3}\ .$$

Then, the double contractions become

$$\boldsymbol{\varepsilon} : \mathbb{C} : \boldsymbol{\varepsilon} = (\varepsilon^M)^\top C^M \varepsilon^M\ , \tag{5.4a}$$

$$\boldsymbol{\sigma} : \mathbb{C}^{-1} : \boldsymbol{\sigma} = (\sigma^M)^\top (C^M)^{-1} \sigma^M\ , \tag{5.4b}$$

with $\boldsymbol{\varepsilon}$ and $\boldsymbol{\sigma}$ denoting second-order tensors, $\mathbb{C}$ a fourth-order tensor, and ε^M, σ^M, C^M their Mandel representations, respectively.

For DDI, we use:

- $\varepsilon = [\varepsilon_e^M] \in \mathbb{R}^{\ell m}$ denotes the collection vector corresponding to the second-order tensors $\boldsymbol{\varepsilon}$,
- $\sigma = [\sigma_e^M] \in \mathbb{R}^{\ell m}$ denotes the collection vector corresponding to the second-order tensors $\boldsymbol{\sigma}$,
- $C = I_m \otimes C^M \in \mathbb{R}^{\ell m \times \ell m}$ refers to the matrix representation of the fourth-order tensor $\mathbb{C}$.

Here, m denotes the number of quadrature points and ℓ is the number of Mandel components ($\ell = 3$ in 2D, $\ell = 6$ in 3D).

The collections ε and σ can be regarded either as $m \times \ell$ matrices (one row per tensor in Mandel form) or equivalently as vectors in $\mathbb{R}^{\ell m}$ obtained by stacking all rows.

5.3 Recommended practice for DDI implementation

The implementation method based on alternating projections, presented in Chapter 3, Algorithm 1, for a fixed pairing P, constitutes the recommended implementation for the DDI method. The procedure relies on two projections:

- P_H, the projection onto $\ker H$, and

- P_E, the projection onto the affine space $E = \{\sigma \quad \text{s.t.} \quad D\sigma = F\}$.

Given an initial approximation σ_0, the algorithm alternates between these projections,

$$\sigma_0 \xrightarrow{P_H} \sigma_1^H \xrightarrow{P_E} \sigma_1 \xrightarrow{P_H} \sigma_2^H \xrightarrow{P_E} \cdots \to \sigma^{\text{DDI}} \ . \tag{5.5}$$

This procedure can be interpreted as a fixed-point algorithm: σ^{DDI} is a fixed point of the operator $P_E \circ P_H$.

For all $j \geq 1$, the iterates σ_j^H are constant by cluster, in other words, all entries corresponding to a given cluster take the same value, namely the cluster average of the corresponding elements. Consequently, the relevant information can be fully represented by a reduced matrix S, with $S = \mathcal{S}^M$ where $\mathcal{S}$ is defined in Eq. 3.6. The matrix S stores only the cluster-wise averages and its dimension depends on $N_{\mathcal{D}}$, the size of the database.
Accordingly, the fixed-point iteration is formulated in terms of S rather than σ. This requires the introduction of a preliminary projection step, which will be described in detail later. The alternating projection scheme can then be expressed in terms of S as

$$S_0 \xrightarrow{\Pi_S} \sigma_1^H \xrightarrow{P_E} \sigma_1 \xrightarrow{\Pi_\sigma} S_1 \xrightarrow{\Pi_S} \sigma_2^H \xrightarrow{P_E} \cdots \longrightarrow \sigma^{\text{DDI}} \ . \tag{5.6}$$

Here,

- $\Pi_\sigma : \mathbb{R}^{\ell m} \to \mathbb{R}^{\ell N_{\mathcal{D}}} \ , \quad \sigma \mapsto S$ (reduction to clusters),
- $\Pi_S : \mathbb{R}^{\ell N_{\mathcal{D}}} \to \mathbb{R}^{\ell m} \ , \quad S \mapsto \sigma^H$ (expansion into σ^H).

Within this framework, the projection onto $\ker H$ can be written as the composition $P_H = \Pi_\sigma \circ \Pi_S$.
From a computational perspective, the reformulation is advantageous: the cost of the implementation scales with the database size $N_{\mathcal{D}}$, whereas an iteration carried out directly on σ results in a cost that invariably depends on m.

Then, the recommended practice for implementing the DDI method can be summarized as follows:

1. Compute a DDI solution by performing a fixed-point iteration on S.
2. Verify the uniqueness of the obtained solution using the third method based on alternating projections, as described in Chapter 3.
   (a) If the solution is unique, terminate the procedure.
   (b) Otherwise, proceed to the next step.
3. Correct the solution by applying one of the three methods introduced in Chapter 4, selected according to the sparsity ratio of the vectors.

In the following, a set of guidelines and practical tools for effective implementation will be presented. The ill-posed case will not be revisited here, as it has already been addressed in detail in Chapter 4.

5.4 Step-by-step guide to the use of tools

5.4.1 Initialization

As discussed earlier, we recommend initializing the algorithm with $S_0 = 0$, equivalently with $\sigma_0 = 0$. With this choice, the first iterate σ_1 corresponds to the minimum $\mathbb{C}$-norm solution of the equilibrium equation $D\sigma = F$. This provides a meaningful and stable starting point for the iterative projection process, which in turn reduces the number of iterations required to reach convergence.

The initialization only influences the result when the solution is not unique, in which case it selects one among the possible solutions.

5.4.2 Clustering

The clustering method used for DDI is the k-means clustering algorithm [70], detailed in Appendix A. It is an unsupervised learning method that partitions the data into k groups (clusters).
The algorithm is typically initialized randomly: each data point is assigned to the cluster with the closest centroid by solving a minimization problem, and the centroids are then updated iteratively.
In practice, it is usually recommended to use the k-means++ initialization [6] strategy instead of purely random initialization, as it provides better starting centroids and reduces computational cost. In this approach, the first centroid is chosen uniformly at random, while each subsequent centroid is selected from the remaining data points with probability proportional to the squared distance to the nearest existing centroid (see Appendix A for details).
To accelerate the assignment step, the centroids can be stored in a kd-tree, so that the nearest centroid for each data point is obtained by a nearest-neighbor query instead of computing all k distances explicitly [76].
However, k-means algorithm is based on the Euclidean norm, whereas the DDI formulation requires a weighted $\mathbb{C}$-norm. To address this, we introduce the transformed variables

$$x_e = \begin{bmatrix} \sqrt{C^M}\,\varepsilon_e^M \\ \sqrt{(C^M)^{-1}}\,\sigma_e^M \end{bmatrix},$$

where $\sqrt{\cdot}$ denotes the matrix square root. With this transformation, the $\mathbb{C}$-norm reduces to the standard Euclidean norm.

The weighted centroid of cluster i in the transformed space is

$$\bar{x}_i = \begin{bmatrix} \sqrt{C^M}\,\mathcal{E}_i^M \\ \sqrt{(C^M)^{-1}}\,\mathcal{S}_i^M \end{bmatrix},$$

with

$$\mathcal{E}_i = \frac{1}{W_i}\sum_{e'} P_{e'i}\, w_{e'}\, \varepsilon_{e'}, \qquad \mathcal{S}_i = \frac{1}{W_i}\sum_{e'} P_{e'i}\, w_{e'}\, \sigma_{e'},$$

and

$$W_i = \sum_{e'} P_{e'i}\, w_{e'}.$$

5.4.3 Projectors

This section describes the projectors used in the DDI algorithm and their practical computation. We recall from Section 5.3 that the algorithm proceeds as follows:

$$S_0 \xrightarrow{\Pi_S} \sigma_1^H \xrightarrow{P_E} \sigma_1 \xrightarrow{\Pi_\sigma} S_1 \xrightarrow{\Pi_S} \sigma_2^H \xrightarrow{P_E} \cdots \longrightarrow \sigma^{\mathrm{DDI}}\ .$$

Projection onto E.

For all $j \geq 1$, the projection is given by

$$DC_wD^\top\eta = F - D\sigma_j^H\ , \tag{5.7a}$$

$$\sigma_j = \sigma_j^H + C_wD^\top\eta\ . \tag{5.7b}$$

where the weighting operator is given by

$$C_w = CW^{-1} \in \mathbb{R}^{\ell m\times \ell m},$$

with

$$W = \mathrm{diag}(w) \in \mathbb{R}^{\ell m\times \ell m}, \qquad w = (w_1,\ldots,w_m)^\top \otimes \mathbf{1}_\ell \in \mathbb{R}^{\ell m},$$

and $\mathbf{1}_\ell$ is the vector of ones in $\mathbb{R}^\ell$.

In other words, P_E projects (with respect to the weighted $\mathbb{C}$-norm) an unbalanced stress field σ_j^H that satisfies equilibrium.
Equation (5.7a) is solved by performing a Cholesky factorization [85] of the symmetric positive definite matrix $DC_wD^\top$, using a symmetric approximate minimum degree permutation [3] to reduce fill-in. Other permutations are possible, we use this one in our implementation. The resulting factors can be computed once and reused throughout the iterations.

Projection onto $\ker H$**.**

The projection onto the kernel of H is given by the composition

$$P_H = \Pi_\sigma \circ \Pi_S \ . \tag{5.8}$$

It relies on two operators:

$$\Pi_\sigma(\sigma_j) = W_P^{-1}\ P_{\text{kron}}^\top\ W\,\sigma_j := S_j, \tag{5.9}$$

$$\Pi_S(S_j) = P_{\text{kron}}\ S_j := \sigma_j^H. \tag{5.10}$$

The matrices are given as:

$P_{\text{kron}} = I_\ell \otimes P$	$\in \mathbb{R}^{\ell m \times \ell N_\mathcal{D}}$	binary pairing matrix,
$W = \text{diag}(w)$	$\in \mathbb{R}^{\ell m \times \ell m}$	weight matrix,
$W_P = \text{diag}(P_{\text{kron}}^\top\ w)$	$\in \mathbb{R}^{\ell N_\mathcal{D} \times \ell N_\mathcal{D}}$	projected weight matrix.

The operator Π_σ computes the weighted average of the mechanical states over each cluster, while Π_S assigns each cluster average back to its associated stress field using P.

5.4.4 Convergence

The alternating projection algorithm is a fixed-point iteration method. As discussed in Chapter 3, its convergence rate depends on the smallest positive principal angle between the sets: the smaller this angle, the slower the convergence, requiring more iterations to achieve convergence.
To improve convergence, we employ the *Anderson acceleration algorithm*, originally introduced in [4]. Anderson acceleration enhances the convergence of fixed-point iterations by incorporating information from several past iterates. The key idea is to exploit this memory to construct an improved approximation for the next iterate, thereby achieving faster convergence.
The algorithm is given by:

Algorithm 2 Anderson Acceleration

Require: x_0, fixed-point mapping $f : \mathbb{R}^n \to \mathbb{R}^n$

1: Choose memory parameter $m_A \in \mathbb{N}_+$
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Set $m_k = \min\{m_A, k\}$
4: Compute weights β_j^k based on the last m_k iterations, satisfying

$$\sum_{j=0}^{m_k} \beta_j^k = 1$$

5: Update the iterate:

$$x_{k+1} = \sum_{j=0}^{m_k} \beta_j^k f(x^{k-m_k+j})$$

6: **end for**

Here, f denotes the fixed-point mapping associated with the alternating projection algorithm.
By definition, when $m_A = 0$ the method reduces to the standard fixed-point iteration. The integer m_k is called memory in iteration k, indicating how many past iterates are used. The next iterate is then formed as a linear combination of the images of these last $m_k + 1$ iterates under the mapping f. Based on how the weights β_j^k in Algorithm 2 are chosen, Anderson acceleration is classified into two sub-classes [25, 88, 92]: type-I and type-II.
In this work, we focus on type-II Anderson. Let $g : \mathbb{R}^n \to \mathbb{R}^n$ the residual of f, defined as

$$g(x) = x - f(x) \ .$$

At each iteration k, the coefficients β^k are obtained by solving the optimization problem:

$$\min \left\| \sum_{j=0}^{m_k} \beta_j g(x^{k-m_k+j}) \right\|_2^2 \quad \text{subject to} \quad \sum_{j=0}^{m_k} \beta_j = 1 \ , \tag{5.11}$$

where $\beta = (\beta_0, \cdots, \beta_{m_k}) \in \mathbb{R}^{m_k+1}$. The optimal solution β^k provides the weights used in the update step of Algorithm 2.

For DDI, Anderson acceleration is applied to the fixed-point iteration restricted to S, where f is given by:

$$f = \Pi_\sigma \circ P_E \circ \Pi_S \ ,$$

and the fixed-point is $f(S) = S$.
It is worth emphasizing that performing the iteration on S is computationally beneficial, as S contains the information of σ^H in a reduced dimension, since σ is constant within each cluster. The dimension of S depends on $N_\mathcal{D}$ (the database size), and Anderson acceleration requires the storage of m_k past iterates, memorizing and updating these reduced iterates in S is therefore significantly less costly than working on σ.

5.4.5 Uniqueness

We have already discussed three methods to characterize the uniqueness criterion (cf. Chap. 3), among which the alternating projection proved the most effective for computing the intersection of $\ker H$ and $\ker D$. We now introduce an additional, straightforward procedure to test whether the problem is well-posed.
For a fixed H,

1. Solve the DDI problem using the alternating projection algorithm 1, starting from two distinct initializations.

2. Subtract the two resulting solutions to determine whether the problem admits a unique solution:

    - If they coincide (up to a numerical tolerance), the solution is unique with high probability, and the intersection subspace is trivial.
    - If the solutions differ, this indicates a loss of uniqueness, implying the existence of a non-trivial vector in the intersection subspace.

In practice, the two initializations could be chosen as: (i) zero and random, or (ii) a perturbed version of the first solution obtained from the zero initialization, by adding random noise to all its components.
This works because applying alternating projection onto $\ker H$ and $\ker D$ amounts to solving a DDI problem with homogeneous boundary conditions. Therefore, subtracting two DDI solutions is equivalent to solving the DDI problem with homogeneous boundary conditions.

Hence, two paths can be used to test uniqueness:

1. check uniqueness by alternating projection $\ker H$ and $\ker D$, as described in Chapter 3,

2. compute two DDI solutions with two different initializations.

Each path determines well-posedness and, when relevant, produces an explicit vector in the intersection. The first achieves this with a single alternating projection run, while the second requires two runs yet, in the ill-posed case, also delivers two distinct possible DDI solutions.
From a computational perspective, if the problem is well-posed, both approaches have the same cost with two runs in total, one to verify uniqueness and one to compute the solution in the first case, or two runs that simultaneously provide uniqueness verification and solution in the second.
In the ill-posed case, with the first path, we start by a first run that produces a non-trivial solution, but one may need to change $N_{\mathcal{D}}$ and repeat, or alternatively refine the obtained DDI solution using one of the proposed correction strategies. With the second path, two distinct solutions are obtained directly from two runs, after which the continuation procedure can proceed and the correction step can be applied.
The choice between these approaches depends on whether one prefers to continue with the initially chosen $N_{\mathcal{D}}$ or to work directly with multiple candidate solutions.

5.4.6 Efficient calculation of the basis

Since the mechanical stresses are constant within each cluster, linear independence of the mechanical stresses σ and of the corresponding material stresses (on S) are equivalent. In other words, the mechanical stresses are linearly independent if and only if the material stresses are linearly independent.

In the ill-posed case, inspired by this observation and by the second path to test well-posedness (Section 5.4.5), the construction of the basis can be carried out as follows (if needed):

1. Solve the DDI problem using the alternating projection algorithm, starting with $\sigma_0 = 0$. Denote the resulting solution by σ_1^{DDI}.

2. For $j \geq 2$, compute another DDI solution starting from a perturbed version of the previous solution by adding random noise to all components, i.e.
$$\sigma_0 = \sigma_{j-1}^{\mathrm{DDI}} + \text{noise}.$$

3. Compute the corresponding material stresses
$$S_j = \Pi_\sigma(\sigma_j^{\mathrm{DDI}}), \qquad S_{j-1} = \Pi_\sigma(\sigma_{j-1}^{\mathrm{DDI}}).$$

4. Compute the difference $\Delta S_j := S_j - S_{j-1}$. The corresponding mechanical stress vectors is then
$$\Pi_S(\Delta S_j) \in \ker D \cap \ker H \ .$$

5. Test the linear independence of the resulting vector ΔS_j with respect to the current basis set (the check is on S-level). This can be done, for example, using the Gram-Schmidt orthogonalization algorithm. If ΔS_j is linearly independent, include $\Pi_S(\Delta S_j)$ in the basis.

6. Repeat the process until no additional linearly independent vectors can be identified. The resulting set spans the space $\mathcal{K}$.

The idea of the algorithm follows the same idea to the one presented in Chapter 3, with two distinctions:

- we use Corollary 3.2.2 to guarantee that the two distinct solutions lie in the intersection basis, and

- the computation is carried out on the reduced vector S, exploiting its equivalence with the linear independence of σ^H.

5.4.7 Multiple snapshots

Up to this point, the presentation of DDI equations and algorithms has been restricted to a single snapshot. A snapshot corresponds to a mesh at a given loading step. In practical settings, however, data is typically available across multiple snapshots, which may include different loading steps or scenarios applied to one or several samples. The above analysis remains valid, provided that the indices associated with quadrature points and nodes are extended accordingly. These indices should no longer be restricted to a single computational mesh, but instead span all meshes.

For multiple snapshots, all matrices and vectors are assembled in block form. For instance, with two snapshots one has

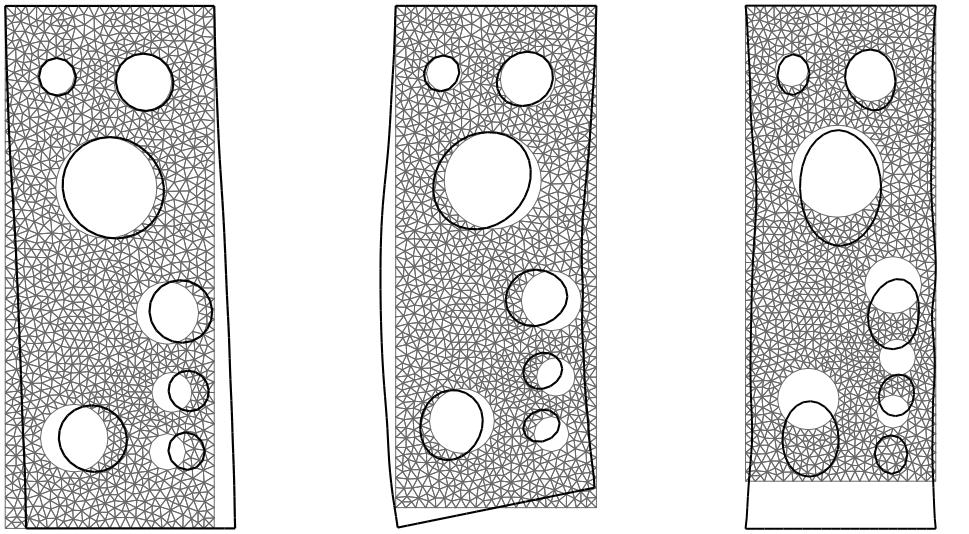
$$D = \begin{bmatrix} D^1 & 0 \\ 0 & D^2 \end{bmatrix}, \qquad \varepsilon = \begin{bmatrix} \varepsilon^1 \\ \varepsilon^2 \end{bmatrix}, \qquad \sigma = \begin{bmatrix} \sigma^1 \\ \sigma^2 \end{bmatrix}, \qquad F = \begin{bmatrix} F^1 \\ F^2 \end{bmatrix}, \qquad P = \begin{bmatrix} P^1 \\ P^2 \end{bmatrix},$$

with $\cdot^1$ and $\cdot^2$ denoting the contributions from snapshots 1 and 2. The extension to any larger number of snapshots follows directly.

In the algorithm, all equations are solved independently by blocks (locally), except for the projection operator Π_σ, which is applied globally. This block-wise calculation enables the method to efficiently handle large amounts of data.

5.5 Application and discussion

In this application, we focus on the approach not addressed in the previous chapters, namely the use of snapshots and Anderson acceleration for convergence. The other computational tools have already been examined in detail. A dataset comprising 21 snapshots is considered, which will also serve for time comparison when analyzing convergence with larger amounts of data.
Specifically, the DDI method is applied to the nonlinear model previously introduced in Chapter 3, Section 3.5, using the same two-dimensional finite element mesh. The bottom boudary of the sample is subjected to a combined loading consisting of a vertical displacement of 10%, a horizontal displacement of 2%, and a rigid-body rotation of 10°, as illustrated in Figure 5.1. The dataset of 21 snapshots corresponds to a total of 50736 quadrature points (=21 × 2416).
DDI is applied for $N_{\mathcal{D}} = 4000$, which corresponds to approximately 12 elements per cluster. This represents a well-posed case, where the uniqueness of the solution was verified, as recommended, by comparing two different initializations.
It should be noted that the fixed-point iteration (alternating projection) was here performed on the reduced vector S, in accordance with the recommended practice for DDI. In this case, the dimension of the problem decreases from



(a) Horizontal displacement. (b) Rotation. (c) Vertical displacement.

Figure 5.1: Finite element mesh used to generate the synthetic dataset. The black line denotes the boundary of the deformed configuration under each loading condition.

50736 to 4000, i.e., by a factor of about 12. The same reduction applies to the memory parameter m_k in Anderson acceleration, which leads to a significant decrease in both memory requirements and computational cost.

Figure 5.2 shows the rate of convergence of the Anderson acceleration for

Anderson	$m_A = 0$	$m_A = 1$	$m_A = 5$	$m_A = 10$	$m_A = 50$	$m_A = 100$
Time (s)	18.92	9.37	6.22	5.46	5.64	5.64

Table 5.1: Computation times for fixed-point iteration with Anderson acceleration for different memory parameters m_A.

different memory parameters m_A, using convergence criteria of 10^{-10} on the residual norm. The corresponding computational times, obtained using the MATLAB® "run and time" function, are presented in Table 5.1.

When $m_A = 0$, corresponding to the original fixed-point method without acceleration, the convergence is very slow, requiring 1969 iterations and a total run time of 18.92 seconds. Introducing even a small memory, $m_A = 1$, substantially improves convergence, reducing the iterations to 644 and the run time to 9.37 seconds. For intermediate memory values, such as $m_A = 5$, the efficiency continues to increase; however, the most substantial gain occurs for $m_A \geq 10$, where the number of iterations decreases by almost an order of magnitude, from roughly 2000 down to about 100. This reduction in iterations results in a corresponding decrease in computational cost, with run times stabilizing between 5 and 6 seconds. Increasing the memory parameter further,

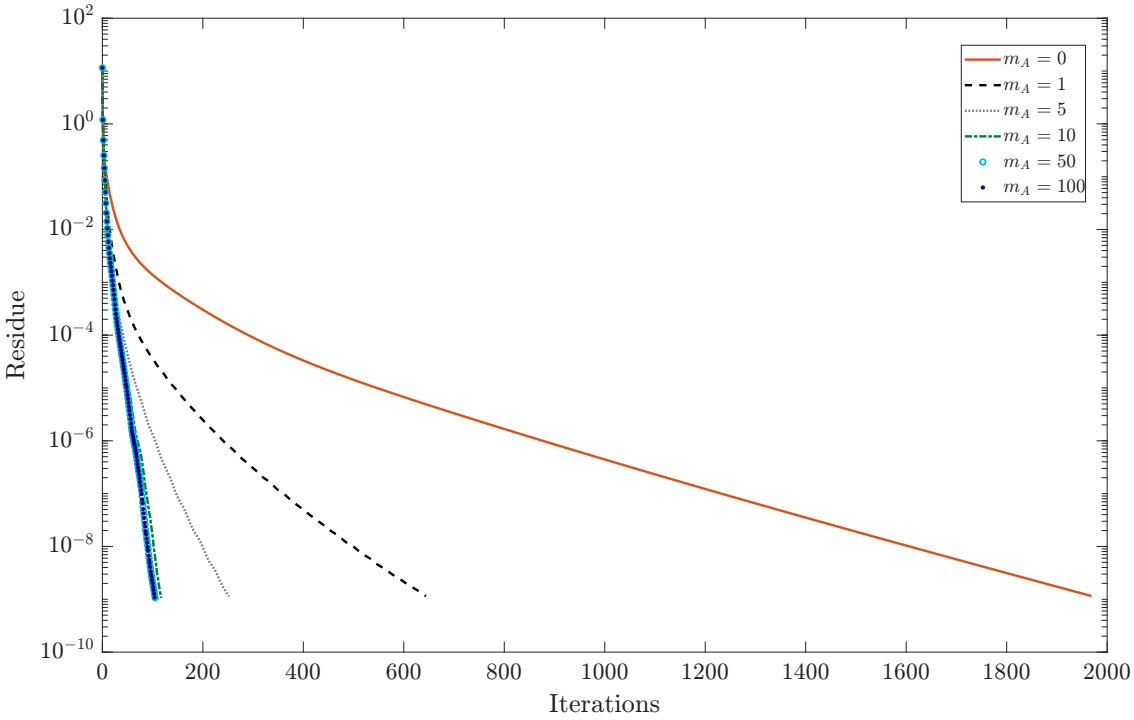
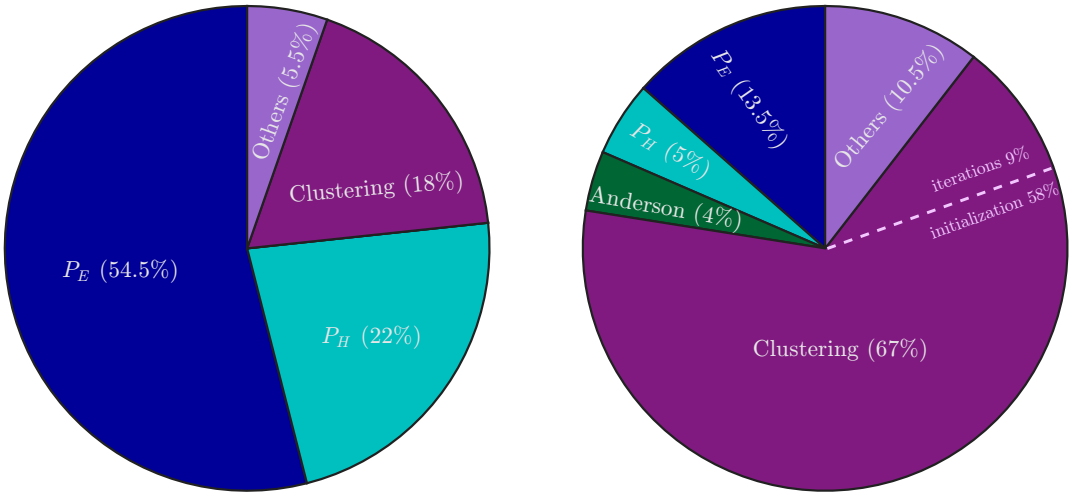


Figure 5.2: Convergence of fixed-point iteration with Anderson acceleration for various memory parameters m_A.

beyond $m_A = 50$, no additional gains are observed, suggesting that the method has reached its practical efficiency limit.
Since $m_A = 10$ offers the best trade-off between iteration count, memory usage, and computational time, we focus the comparison on $m_A = 0$ and $m_A = 10$ in the analysis of run time distribution given in Table 5.1. Although $m_A = 50$ achieves a similar performance, the additional memory cost brings limited benefit.

The time distribution of the algorithm, measured using MATLAB® "run and time" function, is shown in Figure 5.3. For both cases we display the percentage of run time spent on the projections onto E and H (the fixed-point steps), the Anderson update (only present for $m_A = 10$), and the clustering, the remaining operations are grouped as "Others".
For $m_A = 0$, most of the computational time is devoted to the fixed-point iterations, with 76.5% of the total time consumed by the projections (P_E and P_H), while clustering accounts for 18%. In contrast, for $m_A = 10$ the situation reverses: only 22.5% of the run time is spent on the convergence related steps (including P_E, P_H, and the Anderson acceleration), whereas clustering becomes the dominant cost, taking 67% of the total, most of which (58%) arises from the k-means++ initialization. This reflects the efficiency gained by acceleration, where the reduced cost of convergence shifts the computational burden toward



(a) (b)

Figure 5.3: Comparison of the algorithm time distribution for (a) $m_A = 0$ and (b) $m_A = 10$.

the database construction.
It is important to note that the absolute time spent on clustering remains the same in all cases, since the database size $N_{\mathcal{D}}$ is fixed, with 87% of the clustering time due to the k-means++ initialization.

Several optimizations have already been applied. Anderson acceleration was introduced to reduce the number of fixed-point iterations, and the iterations were carried out on the reduced vector S. Clustering was also improved by replacing random initialization with k-means++ and by accelerating the assignment stage through the use of k-d trees and k-NN search. After these improvements, the clustering initialization now constitutes the principal computational bottleneck.

5.6 Conclusion

This chapter has presented a structured framework for the practical implementation of DDI, covering initialization, clustering, projectors, convergence acceleration, uniqueness verification, and efficient computation strategies.
The application example highlights that convergence can be significantly accelerated by Anderson acceleration in the reduced formulation on S, and uniqueness can be effectively addressed, both with computational costs that remain manageable. At the same time, the analysis revealed the shifting role of clustering as the primary computational bottleneck once convergence acceleration is employed. Despite the use of k-means++ initialization and assignment via k-d trees and k-NN search, the initialization remains dominant. Importantly, this cost originates from the clustering algorithm itself and is

independent of DDI-specific tools.
These observations establish clear guidelines for efficient DDI implementation and provide a reference for practical application of DDI and a foundation for future developments.

Conclusions and perspectives

This thesis presents the development and theoretical validation of the Data-Driven Identification (DDI) method, which aims to identify the mechanical response of materials from full-field strain measurements and load cell data, without assuming a constitutive model. DDI solves an inverse problem based on the principle that clustered strain should yield similar stress, while the latter should at the same time verify mechanical balance equilibrium.

The formulation, evolution, and refinement of the DDI method are examined in detail, along with the challenges that have emerged in its application. Despite the promising potential of DDI for various material behaviors, a rigorous theoretical study capable of assessing its accuracy, stability, and convergence toward the true mechanical stress has remained lacking, motivating the developments presented in this work.

We can outline the contributions of this work into three main parts:

- **Theoretical foundations and well-posedness analysis:**
  We provide a mathematical and numerical analysis of the robustness of the Data-Driven Identification (DDI) method, building upon the preliminary work of Leygue [60], who introduced a criterion for the well-posedness. The DDI solution is unique if and only if there does not exist a non-zero stress field that is simultaneously constant within each cluster ($\in \ker H$) and self-balanced ($\in \ker D$).

  - Three methods were developed to characterize the uniqueness criterion: (i) the explicit computation of the intersection basis, which provides exact results but is computationally costly, (ii) the angle-based analysis of subspaces, where the smallest principal angles determine the conditioning of the problem and indicate how close it is to losing uniqueness and, (iii) the alternating projection method, an iterative approach that efficiently samples the intersection space and scales to larger problems.
  - A new implementation strategy based on alternating projection was proposed, with a proof of convergence.

  - The numerical conditioning and convergence rate of the DDI problem were thoroughly analyzed, revealing that the presence of very small principal angles between $\ker H$ and $\ker D$ leads to ill-conditioning, slower convergence, and increased sensitivity to noise.
  - Even in noise-free settings, the clustering initialization remained random. This randomness was analyzed to investigate its impact on uniqueness, to quantify the severity of the problem, and to assess whether it can be resolved.

  These advances ensure that the DDI method can now be applied more effectively, with the ability to determine in practice whether the problem admits a unique solution and to assess its convergence properties.

- **Treatment of ill-posed DDI problem:**
  Building on the uniqueness analysis, the second part of the thesis focuses on situations where the DDI problem becomes ill-posed. Instead of considering such cases as invalid and discarding the results, the study examines how meaningful solutions may still be recovered from the available information.

  - The basis responsible for the loss of uniqueness was examined in detail, and the structure of the resulting solution space was analyzed.
  - Three corrective strategies were proposed and systematically evaluated: (i) naïve elimination of the problematic elements responsible for the loss of uniqueness, (ii) database pruning through elimination problematic clusters, and (iii) additional regularization inspired by the pseudo-inverse framework. Each approach was examined with respect to its efficiency, robustness, and its capacity to recover a meaningful stress field.
  - Numerical experiments were performed across both low- and high-dimensional bases, as well as for different database values, to understand their efficiency and limitations. In addition, the results demonstrate that enriching the database with more diverse data improves the accuracy of the method.

  This contribution shows that, even in an ill-posed case (regardless of the dimension of the intersection basis) there is no need to discard the results. Instead, one can recover and select among the possible solutions, ensuring that meaningful stress fields can still be obtained.

- **Guidelines for effective DDI implementation:**
  The final contribution of this work is the unification of theoretical and numerical developments into a set of practical guidelines for the effective implementation of the DDI method.

  - A step-by-step framework for the algorithm was proposed, including recommendations for initialization, clustering, fixed-point operators, convergence acceleration, ...
  - The benefits of Anderson acceleration were shown to significantly improve convergence, while the initialization of the clustering is now (after this improvements and recommendations) to be the principal computational bottleneck of the method.

  These guidelines provide a systematic framework that improves efficiency and enables the reliable application of DDI in real-world contexts, establishing a reproducible basis for its use on experimental data with greater confidence. Through these developments, the DDI method can now be applied more safely, supported by essential tools and a clearer understanding of its theoretical foundations and practical limitations.

Finally, this theoretical analysis of the DDI framework opens promising perspectives of several aspects of the method.

- Real-world applications:
  - The current results were obtained using synthetic, noise-free data. Since real experimental data are typically noisy and large-scale, an important next step is to adapt and validate the framework on real experimental data, using the guidelines introduced in Chapter 5.
  - Although the applications presented focused on nonlinear elastic materials, the theoretical results remain valid for more general cases. Future work would explore applying the framework to more complex behaviors, such as bi-materials samples and history-dependent responses.
- Improving clustering:
  - The matrix H, associated with clustering, and the matrix D, associated with the mesh, have incompatible sparsity patterns (see Table 3.1). In addition, the k-means initialization constitutes the main computational bottleneck of the DDI algorithm (see Figure 5.3). To overcome this limitations, alternative clustering methods can be applied to establish a stronger connection between clustering structures and mesh structures.
  - The k-means clustering method is geometric and does not account for the underlying physics. A potential avenue is therefore to introduce physics-informed constraints into the clustering step.
  - Another possible approach is nested clustering [36, 73]. Instead of initializing with a large number of clusters $N_{\mathcal{D}}$ and correct the

solution afterward (Chapter 4), one could begin with a smaller $N_{\mathcal{D}}$ and progressively subdivide clusters using a constraint related to kernel intersections.

- Data acquisition:
  - While adding more data has been shown to improve accuracy (Figure 4.8), the optimal strategy for data enrichment remains an open question. Promising directions include incorporating additional experimental data and designing topology-optimized samples through variations in geometry, loading conditions, or their combinations (see Figures 1.3 and 1.5).
  - Alternatively, additional information can be generated through mesh refinement or data augmentation techniques [81] based on geometric transformations (e.g., rotation, flipping).
  - Best effective way to incorporate results from simple experiments into the algorithm of DDI, for example by including them as a snapshot.

Appendix A

k-means clustering method

In this appendix, we revisit the k-means clustering algorithm, originally introduced by MacQueen [70].

Let $(x_1, \ldots, x_n)$ be the dataset. The n data points are partitioned into k ($k \leq n$) sets denoted $C = \{C_1, \ldots, C_k\}$ by minimizing the within-cluster sum of squared distances between the points and their centroids:

$$\arg\min_{C} \sum_{i=1}^{k} \sum_{j=1}^{n} p_{ij} \|x_j - \mu_i\|_2^2 \ , \tag{A.1}$$

where μ_i is the centroid of the points in C_i and

$$p_{ij} = \begin{cases} 1 & \text{if } x_j \in C_i \ , \\ 0 & \text{otherwise} \ . \end{cases}$$

The algorithm proceeds by first initializing the k centroids randomly and then repeating the following steps until convergence:

- assign each data point to the cluster with the closest centroid by updating p_{ij},
- recompute each centroid as the mean of the points assigned to it:

$$\mu_i = \frac{\sum_{j=1}^{n} p_{ij}\, x_j}{\sum_{j=1}^{n} p_{ij}} \ . \tag{A.2}$$

An illustration of the procedure is shown in Figure A.1.

[1] Source: Wikipedia, translated to English.

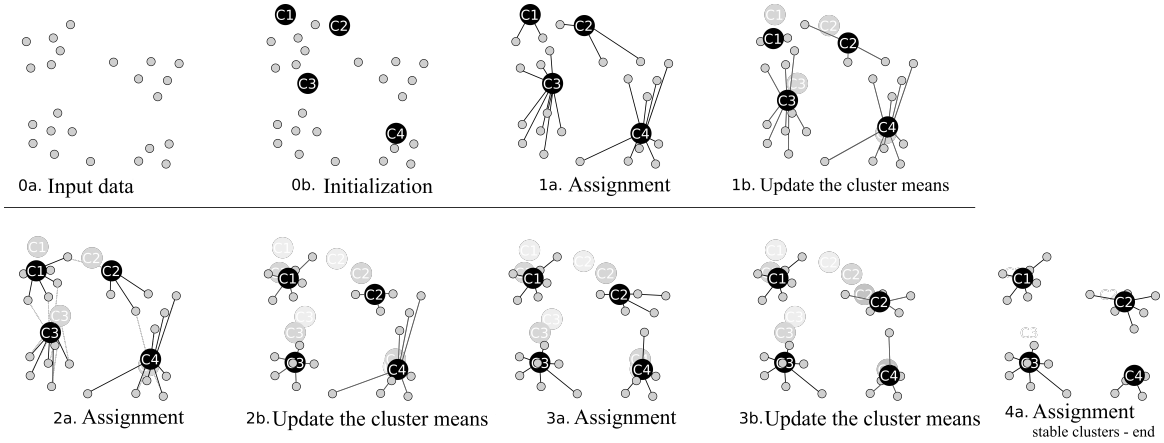


Figure A.1: Illustration of the k-means clustering.[1]

k-means++ initialization

Instead of randomly initialize the centroids, it is recommanded to use k-means++ initialization [6] which selects initial centers according to a data-dependent probability distribution.

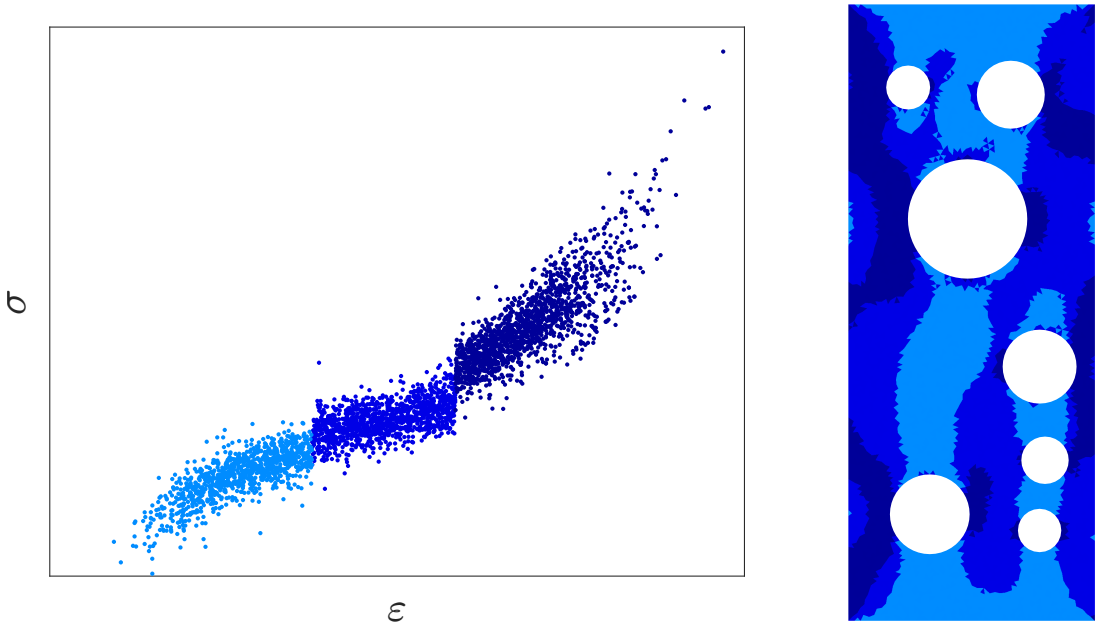
Let $D(x;\mu)$ denote the shortest distance from a data point x to the closest center we have already chosen. The k-means++ initialization proceeds as follows:

1. Choose the first centroid μ_1 uniformly at random over the dataset $\{x_1,\ldots,x_n\}$.
2. Select the next centroid μ_i , such that $\mu_i = x' \in \{x_1,\ldots,x_n\}$ with probability
$$\frac{D(x';\mu_{i-1})^2}{\sum_{j=1}^{n} D(x_j;\mu_{i-1})^2}\,.$$
3. Repeat Step 2 until k centroids are chosen.

This initialization spreads out the centroids across the dataset, typically resulting in faster convergence compared to random initialization.

Appendix B

Visualization of clustering in DDI results



(a) (b)

Figure B.1: Visualization of the clustering, where each color corresponds to a cluster. Here, three clusters are shown: in (a) the constitutive space, and in (b) the mesh.

Appendix C

An idea for constructing a sparse basis of $\ker H \cap \ker D$ from a single element

We recall the notations introduced in Section 4.2.

A sparse basis of $\ker D \cap \ker H$ can be generated from a single nonzero vector in $\mathcal{K}$, obtained via the alternating projection algorithm.
Since every element of $\mathcal{K}$ lies in both $\ker H$ and $\ker D$, and the subspace $\ker H$ is known (spanned by the columns of a matrix P) any vector $k \in \mathcal{K}$ can be expressed as a linear combination of the columns of P:

$$k = \sum_j \beta_j P_j,$$

where $\{P_j\}$ are the columns of P.

Remark. *The indices corresponding to the coefficients $\beta_j \neq 0$ identify the problematic clusters.*

To determine which components also belong to $\ker D$, each column P_j is tested by evaluating DP_j. Those satisfying $DP_j = 0$ are retained, since they lie in $\ker D \cap \ker H$.
The remaining components, which do not individually satisfy this condition, may nevertheless combine linearly to produce vectors in $\ker D$. Such combinations are identified incrementally, starting with pairs of columns and extending to larger subsets as needed.
The procedure is summarized as follows:

1. Compute a vector in $\ker H \cap \ker D$ using alternating projection described in Chapter 3, we denote this vector k.

2. Decompose k as a linear combination of vectors belonging to $\ker H$, which is spanned by the columns of P:

$$k = \sum_j \beta_j P_j \ .$$

3. For each column P_j with $\beta_j \neq 0$, compute DP_j.
   - If $DP_j = 0$, retain P_j as part of the basis of $\ker D \cap \ker H$.
   - Otherwise, proceed to the next step.
4. For the remaining columns that do not individually satisfy $DP_j = 0$, form linear combinations (starting with pairs, then progressively larger subsets if needed). Retain any combination Q such that $DQ = 0$.

Remark. *Since the columns of P encode disjoint clusters, any subset of retained columns (or their linear combinations) is linearly independent by construction.*

Through this procedure, a complete sparse basis of $\ker D \cap \ker H$ can be constructed from a single vector produced by the alternating projection algorithm, together with the known basis of $\ker H$. The resulting basis is equivalent to the basis of $\mathcal{K}$.

References

[1] G. Allaire, *Numerical analysis and optimization: an introduction to mathematical modelling and numerical simulation*. OUP Oxford, 2007 (Cited on page 11).

[2] S. Aloui, R. Othman, A. Poitou, P. Guégan, and S. El-Borgi, "Non-parametric identification of the non-homogeneous stress in high strain-rate uni-axial experiments," *Mechanics Research Communications*, vol. 35, no. 6, pp. 392–397, 2008 (Cited on page 21).

[3] P. R. Amestoy, T. A. Davis, and I. S. Duff, "An approximate minimum degree ordering algorithm," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 4, pp. 886–905, 1996 (Cited on page 85).

[4] D. G. Anderson, "Iterative procedures for nonlinear integral equations," *Journal of the ACM (JACM)*, vol. 12, no. 4, pp. 547–560, 1965 (Cited on page 86).

[5] T. Archer, M. Berny, P. Beauchêne, and F. Hild, "Creep behavior identification of an environmental barrier coating using full-field measurements," *Journal of the European Ceramic Society*, vol. 40, no. 15, pp. 5704–5718, 2020 (Cited on page 18).

[6] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," Stanford, Tech. Rep., 2006 (Cited on pages 55, 84, 100).

[7] F. As' ad, P. Avery, and C. Farhat, "A mechanics-informed artificial neural network approach in data-driven constitutive modeling," *International Journal for Numerical Methods in Engineering*, vol. 123, no. 12, pp. 2738–2759, 2022 (Cited on pages 2, 20).

[8] S. Avril et al., "Overview of identification methods of mechanical parameters based on full-field measurements," *Experimental Mechanics*, vol. 48, pp. 381–402, 2008 (Cited on page 18).

[9] I. Babuška and T. Strouboulis, *The finite element method and its reliability*. Oxford university press, 2001 (Cited on page 11).

[10] C. Bauckhage, "K-means clustering is matrix factorization," *arXiv preprint arXiv:1512.07548*, 2015 (Cited on page 28).

[11] A. Benady, E. Baranger, and L. Chamoin, "Nn-mcre: A modified constitutive relation error framework for unsupervised learning of nonlinear state laws with physics-augmented neural networks," *International Journal for Numerical Methods in Engineering*, vol. 125, no. 8, e7439, 2024 (Cited on pages 2, 20).

[12] M. Benning and M. Burger, "Modern regularization methods for inverse problems," *Acta numerica*, vol. 27, pp. 1–111, 2018 (Cited on page 12).

[13] D. Bindel, *Lecture notes for CS 6210: Matrix computations*, `https://www.cs.cornell.edu/~bindel/class/cs6210-f16/lec/2016-10-17.pdf`, 2016 (Cited on page 47).

[14] Å. Björck and G. H. Golub, "Numerical methods for computing angles between linear subspaces," *Mathematics of computation*, vol. 27, no. 123, pp. 579–594, 1973 (Cited on pages 43, 44).

[15] J. Bleyer, "Learning elastoplasticity with implicit layers," *Computer Methods in Applied Mechanics and Engineering*, vol. 444, p. 118 145, 2025 (Cited on page 20).

[16] J. Bonet and R. D. Wood, *Nonlinear continuum mechanics for finite element analysis*. Cambridge university press, 1997 (Cited on page 7).

[17] M. Bonnet, *Modélisation en mécanique des milieux continus*. Cours École Polytechnique, 2008 (Cited on page 12).

[18] M. Bonnet and A. Constantinescu, "Inverse problems in elasticity," *Inverse problems*, vol. 21, no. 2, R1, 2005 (Cited on page 12).

[19] M. Bonnet, P. Salasiya, and B. B. Guzina, "Modified error-in-constitutive-relation (mecr) framework for the characterization of linear viscoelastic solids," *Journal of the Mechanics and Physics of Solids*, vol. 190, p. 105 746, 2024 (Cited on pages 1, 19).

[20] S. Boyd, "Least-norm solutions of undetermined equations," *Lecture*, vol. 8, p. 8, 2007 (Cited on page 69).

[21] S. Boyd and J. Dattorro, "Alternating projections," *EE392o, Stanford University*, 2003 (Cited on page 45).

[22] B. C. Cameron and C. C. Tasan, "Full-field stress computation from measured deformation fields: A hyperbolic formulation," *Journal of the Mechanics and Physics of Solids*, vol. 147, p. 104 186, 2021 (Cited on page 21).

[23] C. Charles, "Introduction aux problèmes inverses," *Notes de Statistique et d'Informatique*, 2014 (Cited on page 12).

[24] G. Chavent, *Nonlinear least squares for inverse problems: theoretical foundations and step-by-step guide for applications*. Springer Science & Business Media, 2010 (Cited on page 12).

[25] Y. Chen, L. Gélébart, C. Chateau, M. Bornert, C. Sauder, and A. King, "Analysis of the damage initiation in a sic/sic composite tube from a direct comparison between large-scale numerical simulation and synchrotron x-ray micro-computed tomography," *International Journal of Solids and Structures*, vol. 161, pp. 111–126, 2019 (Cited on page 87).

[26] P. G. Ciarlet, *The finite element method for elliptic problems*. SIAM, 2002 (Cited on page 11).

[27] L. Costecalde, "Data-driven model identification for hyperelasticity: Mapping the strain energy throughout multiaxial experiments," Ph.D. dissertation, École centrale de Nantes, 2023 (Cited on pages 15, 16, 37, 76).

[28] L. Costecalde, A. Leygue, M. Coret, and E. Verron, "Data-driven identification of hyperelastic models by measuring the strain energy density field," *Rubber Chemistry and Technology*, vol. 96, no. 4, pp. 443–454, 2023 (Cited on pages 2, 35, 54).

[29] M. Dalémat, "Une expérimentation réussie pour l'identification de la réponse mécanique sans loi de comportement: Approche data-driven appliquée aux membranes élastomères," PhD thesis, École Centrale de Nantes, 2019 (Cited on pages 14, 15, 36).

[30] M. Dalémat, M. Coret, A. Leygue, and E. Verron, "Measuring stress field without constitutive equation," *Mechanics of Materials*, vol. 136, p. 103 087, 2019 (Cited on pages 2, 34, 35).

[31] M. Dalémat, M. Coret, A. Leygue, and E. Verron, "Robustness of the data-driven identification algorithm with incomplete input data," *Journal of Theoretical, Computational and Applied Mechanics*, 2023 (Cited on pages 2, 35).

[32] M. De Buhan, "Problemes inverses et simulations numériques en viscoélasticité 3d.," Ph.D. dissertation, Université Pierre et Marie Curie-Paris VI, 2010 (Cited on page 12).

[33] J. W. Demmel, *Applied numerical linear algebra*. SIAM, 1997 (Cited on page 47).

[34] F. Deutsch, *Best approximation in inner product spaces*. Springer, 2001, vol. 7 (Cited on page 47).

[35] M. I. Diaz, W. Aquino, and M. Bonnet, "A modified error in constitutive equation approach for frequency-domain viscoelasticity imaging using interior data," *Computer methods in applied mechanics and engineering*, vol. 296, pp. 129–149, 2015 (Cited on pages 1, 19).

[36] R. Eggersmann, L. Stainier, M. Ortiz, and S. Reese, "Efficient data structures for model-free data-driven computational mechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 382, p. 113 855, 2021 (Cited on page 97).

[37] L. Eldén, "A weighted pseudoinverse, generalized singular values, and constrained least squares problems," *BIT Numerical Mathematics*, vol. 22, pp. 487–502, 1982 (Cited on page 68).

[38] M. Flaschel, S. Kumar, and L. De Lorenzis, "Unsupervised discovery of interpretable hyperelastic constitutive laws," *Computer Methods in Applied Mechanics and Engineering*, vol. 381, p. 113 852, 2021 (Cited on pages 1, 20).

[39] E. Florentin and G. Lubineau, "Identification of the parameters of an elastic material model using the constitutive equation gap method," *Computational Mechanics*, vol. 46, pp. 521–531, 2010 (Cited on pages 1, 19).

[40] M. Giton, A. Caro-Bretelle, and P. Ienny, "Hyperelastic behaviour identification by a forward problem resolution: Application to a tear test of a silicone-rubber," *Strain*, vol. 42, no. 4, 2006 (Cited on page 18).

[41] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU press, 2013 (Cited on pages 44, 49, 69).

[42] M. Grédiac, "Principe des travaux virtuels et identification," *Comptes rendus de l'Académie des sciences. Série 2, Mécanique, Physique, Chimie, Sciences de l'univers, Sciences de la Terre*, vol. 309, no. 1, pp. 1–5, 1989 (Cited on pages 1, 19).

[43] M. Grédiac and F. Hild, *Full-field measurements and identification in solid mechanics*. John Wiley & Sons, 2012 (Cited on pages 2, 15).

[44] M. Greenacre, *Correspondence analysis in practice*. chapman and hall/crc, 2017 (Cited on page 34).

[45] N. Hachem, A. Leygue, and L. Stainier, "Mathematical and numerical assessment of data-driven identification method applied to nonlinear elasticity," *Computer Methods in Applied Mechanics and Engineering*, vol. 446, p. 118 273, 2025 (Cited on pages 40, 58).

[46] J. Hadamard, "Sur les problèmes aux dérivées partielles et leur signification physique," *Princeton university bulletin*, pp. 49–52, 1902 (Cited on page 12).

[47] H. Hencky, "The elastic behavior of vulcanized rubber," *Journal of Applied Mechanics*, 1933 (Cited on page 34).

[48] W. F. Hosford, *Mechanical behavior of materials*. Cambridge university press, 2010 (Cited on page 13).

[49] V. Isakov, *Inverse problems for partial differential equations*. Springer, 2006 (Cited on page 12).

[50] M. Itskov et al., *Tensor algebra and tensor analysis for engineers*. Springer, 2009, vol. 2 (Cited on page 8).

[51] E. Jones et al., "Parameter covariance and non-uniqueness in material model calibration using the virtual fields method," *Computational Materials Science*, vol. 152, pp. 268–290, 2018 (Cited on pages 1, 19).

[52] Y. Kanno, "Mixed-integer programming formulation of a data-driven solver in computational elasticity," *Optimization Letters*, vol. 13, no. 7, pp. 1505–1514, 2019 (Cited on page 28).

[53] V. J. Katz, "The history of stokes' theorem," *Mathematics Magazine*, vol. 52, no. 3, pp. 146–156, 1979 (Cited on page 10).

[54] K. T. Kavanagh and R. W. Clough, "Finite element applications in the characterization of elastic solids," *International Journal of Solids and Structures*, vol. 7, no. 1, pp. 11–23, 1971 (Cited on pages 1, 18).

[55] T. Kirchdoerfer and M. Ortiz, "Data-driven computational mechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 304, pp. 81–101, 2016 (Cited on pages 2, 22, 25, 49).

[56] P. Ladevèze and D. Leguillon, "Error estimate procedure in the finite element method and applications," *SIAM Journal on Numerical Analysis*, vol. 20, no. 3, pp. 485–509, 1983 (Cited on pages 17, 19).

[57] P. Ladevèze, D. Nedjar, and M. Reynier, "Updating of finite element models using vibration tests," *AIAA journal*, vol. 32, no. 7, pp. 1485–1491, 1994 (Cited on page 19).

[58] R. Langlois, "Non-parametric evaluation of stress fields for history-dependent materials: Formulation, validation and appications," Ph.D. dissertation, École centrale de Nantes, 2023 (Cited on page 16).

[59] R. Langlois, M. Coret, and J. Réthoré, "Non-parametric stress field estimation for history-dependent materials: Application to ductile material exhibiting piobert–lüders localization bands," *Strain*, vol. 58, no. 4, e12410, 2022 (Cited on pages 2, 35).

[60] A. Leygue, "On the formulation and convergence of data-driven identification," *Comptes Rendus. Mécanique*, vol. 353, no. G1, pp. 761–773, 2025 (Cited on pages 2, 3, 21, 28, 40, 41, 42, 44, 46, 47, 49, 57, 95).

[61] A. Leygue, M. Coret, J. Réthoré, L. Stainier, and E. Verron, "Data-based derivation of material response," *Computer Methods in Applied Mechanics and Engineering*, vol. 331, pp. 184–196, 2018 (Cited on pages 2, 21, 22, 24, 25, 28, 34, 54).

[62] C. Liu, "Nonuniform stress field determination based on deformation measurement," *Journal of Applied Mechanics*, vol. 88, no. 7, p. 071 005, 2021 (Cited on page 21).

[63] I.-S. Liu et al., *Continuum mechanics*. Springer, 2002, vol. 5 (Cited on page 7).

[64] H. Madeira et al., "Data-driven identification unravels multiaxial mechanical response of a carbon-black filled elastomer during ageing," *Polymer Testing*, vol. 137, p. 108 521, 2024 (Cited on pages 31, 37, 76).

[65] J. Mandel, "Généralisation de la théorie de plasticité de wt koiter," *International Journal of Solids and structures*, vol. 1, no. 3, pp. 273–295, 1965 (Cited on pages 11, 41, 80).

[66] T. Mánik, "A natural vector/matrix notation applied in an efficient and robust return-mapping algorithm for advanced yield functions," *European Journal of Mechanics-A/Solids*, vol. 90, p. 104 357, 2021 (Cited on pages 11, 80).

[67] J. Martins, A. Andrade-Campos, and S. Thuillier, "Comparison of inverse identification strategies for constitutive mechanical models using full-field measurements," *International Journal of Mechanical Sciences*, vol. 145, pp. 330–345, 2018 (Cited on page 18).

[68] F. Masi and I. Stefanou, "Multiscale modeling of inelastic materials with thermodynamics-based artificial neural networks (tann)," *Computer Methods in Applied Mechanics and Engineering*, vol. 398, p. 115 190, 2022 (Cited on page 20).

[69] N. McCormick and J. Lord, "Digital image correlation," *Materials today*, vol. 13, no. 12, pp. 52–54, 2010 (Cited on page 15).

[70] J. B. McQueen, "Some methods of classification and analysis of multivariate observations," in *Proc. of 5th Berkeley Symposium on Math. Stat. and Prob.*, 1967, pp. 281–297 (Cited on pages 30, 84, 99).

[71] A. Moussawi, G. Lubineau, E. Florentin, and B. Blaysat, "The constitutive compatibility method for identification of material parameters based on full-field measurements," *Computer methods in applied mechanics and engineering*, vol. 265, pp. 1–14, 2013 (Cited on pages 1, 19).

[72] J. Neggers, F. Mathieu, F. Hild, S. Roux, and N. Swiergiel, "Improving full-field identification using progressive model enrichments," *International Journal of Solids and Structures*, vol. 118, pp. 213–223, 2017 (Cited on pages 2, 20).

[73] J. Newling and F. Fleuret, "Nested mini-batch k-means," *Advances in neural information processing systems*, vol. 29, 2016 (Cited on page 97).

[74] E. Pagnacco, D. Lemosse, F. Hild, and F. Amiot, "Inverse strategy from displacement field measurement and distributed forces using fea," in *2005 SEM annual conference and exposition on experimental and applied mechanics, Portland*, 2005, pp. 7–9 (Cited on page 18).

[75] J.-C. Passieux and R. Bouclier, "Classic and inverse compositional gauss-newton in global dic," *International Journal for Numerical Methods in Engineering*, vol. 119, no. 6, pp. 453–468, 2019 (Cited on page 15).

[76] D. Pelleg and A. Moore, "Accelerating exact k-means algorithms with geometric reasoning," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, 1999, pp. 277–281 (Cited on page 84).

[77] R. Penrose, "A generalized inverse for matrices," in *Mathematical proceedings of the Cambridge philosophical society*, Cambridge University Press, vol. 51, 1955, pp. 406–413 (Cited on page 68).

[78] F. Pierron, H. Zhu, and C. Siviour, "Beyond hopkinson's bar," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 372, no. 2023, p. 20 130 195, 2014 (Cited on page 21).

[79] F. Pierron and M. Grédiac, *The virtual fields method: extracting constitutive mechanical parameters from full-field deformation measurements*. Springer Science & Business Media, 2012 (Cited on page 19).

[80] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational physics*, vol. 378, pp. 686–707, 2019 (Cited on page 20).

[81] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019 (Cited on page 98).

[82] F. Sidoroff, *Mécanique des milieux continus*. Éditions de l'École Polytechnique, 1984 (Cited on page 7).

[83] L. Stainier, *Modélisation en mécanique des milieux continus*. Cours. École centrale de Nantes, 2022 (Cited on page 7).

[84] L. Stainier, A. Leygue, and M. Ortiz, "Model-free data-driven methods in mechanics: Material data identification and solvers," *Computational Mechanics*, vol. 64, no. 2, pp. 381–393, 2019 (Cited on pages 31, 54).

[85] G. Strang, *Linear Algebra and Its Applications*, 4th ed. Cengage Learning, 2012 (Cited on pages 43, 85).

[86] M. Tanaka and G. S. Dulikravich, *Inverse problems in engineering mechanics*. Elsevier, 1998 (Cited on page 12).

[87] P. Thakolkaran, A. Joshi, Y. Zheng, M. Flaschel, L. De Lorenzis, and S. Kumar, "Nn-euclid: Deep-learning hyperelasticity without stress data," *Journal of the Mechanics and Physics of Solids*, vol. 169, p. 105 076, 2022 (Cited on pages 2, 20).

[88] A. Toth and C. T. Kelley, "Convergence analysis for anderson acceleration," *SIAM Journal on Numerical Analysis*, vol. 53, no. 2, pp. 805–819, 2015 (Cited on page 87).

[89] G. Valdés-Alonzo, "Identification of material properties and phase distribution of heterogeneous materials through data-driven computational methods: Towards an enhaced constitutive space," Ph.D. dissertation, École centrale de Nantes; Universitat politècnica de Catalunya-BarcelonaTech, 2022 (Cited on page 35).

[90] G. Valdés-Alonzo, C. Binetruy, B. Eck, A. García-González, and A. Leygue, "Phase distribution and properties identification of heterogeneous materials: A data-driven approach," *Computer Methods in Applied Mechanics and Engineering*, vol. 390, p. 114 354, 2022 (Cited on pages 2, 34).

[91] A. Vinel, R. Seghir, J. Berthe, G. Portemont, and J. Réthoré, "Experimental characterization of material strain-rate dependence based on full-field data-driven identification," *International Journal of Impact Engineering*, vol. 194, p. 105 083, 2024 (Cited on pages 2, 31).

[92] H. F. Walker and P. Ni, "Anderson acceleration for fixed-point iterations," *SIAM Journal on Numerical Analysis*, vol. 49, no. 4, pp. 1715–1735, 2011 (Cited on page 87).

[93] P.-Å. Wedin, "Perturbation theory for pseudo-inverses," *BIT Numerical Mathematics*, vol. 13, pp. 217–232, 1973 (Cited on pages 68, 69).

Titre : Analyse des méthodes d'identification et de génération de données pour les approches data-driven en mécanique des solides

Mot clés : Méthodes Data-Driven, Identification, Problème inverse, Mesure de champs, Projection alternée, Régularisation

Résumé : La méthode *Data-Driven Identification* (DDI) permet d'estimer les champs de contrainte à partir des mesures de champs de déformation et des résultantes d'efforts appliqués, sans postuler a priori un modèle de comportement. Elle résout un problème inverse fondé sur le principe selon lequel des déformations similaires doivent correspondre à des contraintes similaires, tout en satisfaisant l'équilibre mécanique.
Cette thèse se concentre sur le développement et la validation théorique de la méthode. Tout d'abord, une analyse mathématique et numérique rigoureuse est réalisée, comprenant l'étude du critère d'unicité, une nouvelle implémentation accompagnée d'une preuve de convergence, et une évaluation du conditionnement et de la vitesse de convergence. Ensuite, le cas des problèmes DDI mal posés est examiné, et des approches correctives sont proposées pour garantir des solutions valides. Enfin, des outils pratiques et des recommandations pour une mise en œuvre robuste sont développés, constituant une référence pour de futures recherches et applications. Ces contributions établissent une base théorique solide pour la méthode DDI et en renforcent la robustesse ainsi que l'applicabilité, grâce aux outils proposés et à une compréhension approfondie de ses fondements théoriques et de ses limites pratiques.

Title: Analysis of data identification and generation methods for data-driven approaches in solid mechanics

Keywords: Data-Driven methods, Material identification, Inverse problem, Full-field measurement, Alternating projection, Regularization

Abstract: The Data-Driven Identification (DDI) method estimates the mechanical response of material samples without assuming a predefined constitutive model. DDI solves an inverse problem based on the principle that similar strains should correspond to similar stresses, while ensuring mechanical equilibrium.
This thesis focuses on the development and theoretical validation of the DDI method. First, a rigorous mathematical and numerical analysis is provided, including the study of the uniqueness criterion, a new implementation with a proof of convergence, and an assessment of conditioning and convergence rates. Second, the case of ill-posed DDI problems is addressed, and corrective strategies are introduced to obtain valid solutions. Finally, essential tools and practical guidelines for reliable implementation are developed, providing a reference for future research and applications. These contributions establish a rigorous theoretical foundation for the DDI method, enhancing its robustness and applicability while enabling safer use through essential tools and a clearer understanding of its practical limitations.