Politecnico di Torino

Politecnico di Torino

Corso di Laurea Magistrale

in Ingegneria Biomedica

Finite Element Model of a composite human femur and validation through Digital Image Correlation

Relatrice:
Dr. Mara Terzini
Correlatori:
Dr. Andrea Tancredi Lugas
Ing. Federico Andrea Bologna

Candidata:
Sara Negri
S303305

A.A. 2023/2024

Summary

Abstract 5

List of Figures & Tables 7

List of Acronyms & Abbreviations 10

1 Aim & Design of the study 11

1.1 Aim of the study 11

1.2 Design of the study 11

2 Background 12

2.1 Characteristics of the bone tissue 12

2.2 Mechanical properties of the bone tissue 13

2.3 Anatomy of the human femur 16

2.4 Osteolysis and femur fractures 17

2.5 Characteristics of the femur implants 20

2.6 Finite Element Method 22

2.6.1 Introduction to FEM basis 22

2.6.2 Continuum-Level FE and Femur model 23

2.7 Digital Image Correlation 24

2.7.1 Local DIC Approach 26

2.7.2 Global DIC Approach 27

2.7.3 Subset Shape Functions 28

3 Preliminary Set-up: Cemented Femur 30

3.1 Materials and Methods 30

3.1.1 Preliminary test on a simil-sawbone cylinder 30

3.1.2 Femur set-up preparation and DIC acquisition 31

3.1.3 VIC-3D for image post-processing 33

3.1.4 Finite Element Model and Analysis in Abaqus 35

3.1.5 MATLAB implementation for FEM validation through DIC 37

3.2 Results 43

3.2.1 FEM analysis in Abaqus 43

3.2.2 MATLAB implementation for FEM validation through DIC 45

4 Design of a new experimental set-up 52

4.1 Materials and Methods 52

4.2 Results 54

5 Conclusions and future directions 56

6 Appendix 58

6.1 Point Clouds alignment 58

6.2 FEM validation through DIC 61

Bibliography 66

Abstract

In the field of osteosynthesis, modelling of medical devices in silico is essential to predict and simulate their post-implantation behavior. The purpose of these procedures is to develop and facilitate an increasingly patient-specific approach that prevents the occurrence of complications due to errors in the placement of the implant.

The term osteosynthesis refers to a reconstructive surgical technique whose purpose is to join two or more bone fragments after their realignment, using screws, plates and other devices, involving materials with both mechanical and biological properties appropriate to the implant. In this regard, it is of utmost importance to adopt a patient-specific approach, as implant failure or success depends on both technical choices and patient-related factors. Into this scenario, there are several studies that propose finite element modeling of the human femur, with the aim of obtaining fracture risk prediction and assessment.

The study tries to define an approach focused on proper bone modeling of the femur that takes into account its mechanical and physiological characteristics. As such, starting from tomographic images of the patient, it would be desired to develop a numerical model that reproduces the biomechanics of the femur in the loading condition. In this case, the load was simplified and considered as a uniaxial compression force concentrated on the femoral head, hypothetically representing the load of the coxo-femoral joint.

It was chosen to use the finite element approach, discretizing bone geometry with tetrahedral elements and assigning the mechanical properties of Sawbone, the synthetic material chosen to mimic the femur. The model obtained was loaded uniaxially and bound in the region of the condyles using an encastre, simplifying the real case.

Since the purpose was the validation of the finite element model, a comparison of displacements and strains obtained with Finite Element Analysis (FEA) was made with those obtained through an experimental procedure. This involved the use of Digital Image Correlation (DIC) which allowed the recording of the displacements and strains of a Sawbone femur bound to the test machine. The comparison procedure was automated using a MATLAB script, calculating the errors and accuracy of the Finite Element Model (FEM) estimate compared to the real experimental case with DIC.

The results obtained showed a satisfactory prediction of displacements along the longitudinal axis and the transverse axis. The deformations, however, did not show a good degree of prediction, resulting in high errors and poor accuracy. The reasons for this could be the lack of further experimental evidence and acquisitions, together with the need to perform new FE simulations. In the latter case, changing the test parameters and the physical-mechanical characteristics of the bone could lead to improvements in the predicted results.

Knowing the limitations of the previous test set-up, a new structure is presented in the second part of the work, designed considering the modularity of the applications and the anatomical position of the femur as accurately as possible. The new set-up consists of six elements, three of which are made of steel and three to be made of ABS with 3D printing to mimic the shape of the femur.

The new experimental set-up needs to be tested using the DIC technique to validate a FE model to be built from scratch based on the geometry created. The optimization of the model could make up for the shortcomings highlighted by the preliminary analysis, reproducing more accurately the mechanical properties of the femur.

List of Figures & Tables

Fig. 2.1: Human bone.

Fig. 2.2: Morphological distinction between cortical and trabecular bone.

Fig. 2.3: Explanation of bone structure.

Fig. 2.4: Stress-strain, tensile and compression curves of cortical bone in the directions longitudinal and transverse; Compression tests on sample of cortical bone and on sample of trabecular bone with different apparent densities.

Fig. 2.5: Schematic representation of internal force applied normally.

Fig. 2.6: Front and back view of the femur.

Fig. 2.7: Anatomical regions in which osteolysis can occur proximally.

Fig. 2.8: Diaphyseal femur fractures classification.

Fig. 2.9: Examples of distal femur fractures classification.

Fig. 2.10: a: LCP locking and compression hole; b: locking head screw; c: example of LCP.

Fig. 2.11: Antero-posterior (C) and lateral (D) vision of healed fracture by using a longer plate with multiple nonlocking screws proximal to the fracture and a longer working length.

Fig. 2.12: Flowchart of model-based simulation (MBS) by computer.

Fig. 2.13: Workflow of DIC displacement and strain measurement.

Fig. 2.14: Example of step, subset, subset centre, pixel and strain window.

Fig. 3.1: MATLAB script for automation of DIC images rotation and cropping; cylinder with speckle pattern and dummy reference system mounted onto Instron, acquired by DIC 0 camera.

Fig. 3.2: Left Sawbone femur with speckle pattern used for the test.

Fig. 3.3: Relationship between mechanical axis and anatomical axis in normal lower extremity.

Fig. 3.4: Schematic view of the set-up used for DIC acquisition.

Fig. 3.5: DIC cameras positioning with respect to femur onto Instron.

Fig. 3.6: Femur diaphysis with DIC camera 1; ROI definition using VIC-3D.

Fig. 3.7: Displacement U for the first and last acquisition.

Fig. 3.8: Strain ε_1 for the first and last acquisition.

Fig. 3.9: Strain ε_2 for the first and last acquisition.

Fig. 3.10: Solid femur imported in Abaqus.

Fig. 3.11: Groups of nodes for condyles and head regions chosen from the mesh.

Fig. 3.12: Reference points used for load and boundary condition application.

Fig. 3.13: F0: initial FEM coordinates; F1: final FEM coordinates; D0: initial DIC coordinates; D130: final DIC coordinates.

Fig. 3.14: Rigid rotation and translation, ROI definition for FEM data.

Fig. 3.15: MATLAB script that displays the application of ICP algorithm using DIC data as moving point cloud and FEM data as fixed point cloud.

Fig. 3.16: Computation of displacements and modules; comparison between mod_0, mod_R and mod_T.

Fig. 3.17: Interpolation of displacements and NaN elimination.

Fig. 3.18: Calculation of errors.

Fig. 3.19: SVD application, strain calculation and interpolation of DIC to FEM strain.

Fig. 3.20: RMSE and RMSE % calculation for interpolated strains.

Fig. 3.21: Definition of a FEM surface connectivity matrix and construction of a new one by correspondence with DIC id nodes.

Fig. 3.22: U displacement (magnitude) for the deformed configuration.

Fig. 3.23: Max. principal strain and Min. principal strain for the deformed configuration.

Fig. 3.24: Principal strains with force and constraint vectors.

Fig. 3.25: Point clouds of FEM and DIC, before and after rigid transformation (initial coordinates).

Fig. 3.26: ROI defined for FEM data.

Fig. 3.27: ICP algorithm result of alignment between FEM and DIC point clouds.

Fig. 3.28: Comparison between DIC displacements before and after rigid3D, and FEM displacements.

Fig. 3.29: Result of displacements interpolation.

Fig. 3.30: Coordinates comparison after interpolation for verification.

Fig. 3.31: Scatter plot of displacements for DIC and FEM comparison.

Fig. 3.32: Contour plot of the differences for displacements DIC-FEM.

Fig. 3.33: Contour plot of the difference in percentage for displacements DIC-FEM.

Fig. 4.1: Design of the wedge realized with SOLIDWORKS®.

Fig. 4.2: Design of the L-shaped solid realized with SOLIDWORKS®.

Fig. 4.3: Design of the vice realized with SOLIDWORKS®.

Fig. 4.4: Three steel elements realized with Ansys SpaceClaim.

Fig. 4.5: ABS elements and femur realized with Ansys SpaceClaim.

Fig. 4.6: Representation of the complete structure.

Tab. 3.1: Errors calculated on displacements for x, y, z directions and modules.

Tab. 3.2: RMSE % for principal strains.

List of Acronyms & Abbreviations

FEM – Finite Element Model.

DIC – Digital Image Correlation.

FEA – Finite Element Analysis.

AO/OTA – Arbeitsgemeinschaft für Osteosynthesefragen/Orthopaedic Trauma Association.

LCP – Locking Compression Plate.

DCP – Dynamic Compression Plate.

MIPO – Minimally Invasive Percutaneous Osteosynthesis.

QCT – Quantitative Computed Tomography.

MBS – Model Based-Simulation.

ZNSSD – Zero-mean Normalised Sum of Squared Differences.

SSD – Sum of Squared Differences.

ROI – Region Of Interest.

ICP – Iterative Closest Point.

RMSE – Root Mean Square Error.

IDW – Inverse Distance Weighting.

SVD – Singular Value Decomposition.

ABS – Acrylonitrile Butadiene Styrene.

PLA – PolyLactic Acid.

1 Aim & Design of the study

1.1 Aim of the study

This project aims to investigate and predict the mechanisms of femoral rupture that may occur as a result of osteolysis. The study wants to propose a patient-specific approach that allows the definition of the correct positioning and configuration of the implant to avoid related complications and failure. Furthermore, the project aims to define and implement a Finite Element Method (FEM) model of the femur and validate it using experimental data from DIC analysis.

1.2 Design of the study

The study can be divided into two main sections.

1. *Previous experimental set-up: potted femur.*
   - 1.1 Uniaxial compressive test on a potted femur and DIC acquisition of the specimen with VIC-Snap.
   - 1.2 Post-processing of images with VIC-3D and output data definition.
   - 1.3 Creation of a FE model of the femur starting from its geometry and performing FE analysis to obtain displacements and strains.
   - 1.4 Definition of a MATLAB algorithm to align DIC and FEM point clouds and validation of FE model with experimental DIC outputs.

2. *New experimental set-up.*
   - 2.1. Definition of the drawbacks and the inaccuracies of the previous potted femur set-up.
   - 2.2. Design with Ansys SpaceClaim of a modular structure for housing the femur, considering its features and anatomical inclination.

2 Background

This section is dedicated to the presentation of the characteristics of the bone tissue, the mechanical properties of the bone and the anatomy of the human femur, to which is added a description of the osteolysis and the characteristics of the femur implant. The second part of the chapter focuses on the theoretical and applicative basis of FEM and DIC analysis.

2.1 Characteristics of the bone tissue

Bone tissue is a type of connective tissue characterized by an extracellular mineralized matrix, which makes it able to perform support and protection functions. This characteristic is conferred by inorganic molecules of calcium phosphate in the form of hydroxyapatite crystals in the formula: $[Ca_{10}(PO_4)_6(OH)_2]$. The bone also acts as a deposit of calcium and phosphate ions that, when released into the bloodstream, allow the maintenance of adequate levels in the body. The bone matrix contains both collagen type I and V, and proteins of a non-collagenic nature that maintain the homeostasis of the tissue.

The mature bone tissue consists of structural units called osteons, formed by concentric lamellae surrounding the central canal of Havers, provided with vessels and nerve fibers (Fig. 2.1). Inside the bone matrix there are lacunae in which there are osteocytes that, together with the lacunae, have a fundamental role in the maintenance of the mineralized tissue. Along with osteocytes, there are osteoprogenitor cells that produce osteoblasts, osteoblasts that are responsible for osteogenesis and evolve into osteocytes, and osteoclasts, the cells responsible for bone resorption.

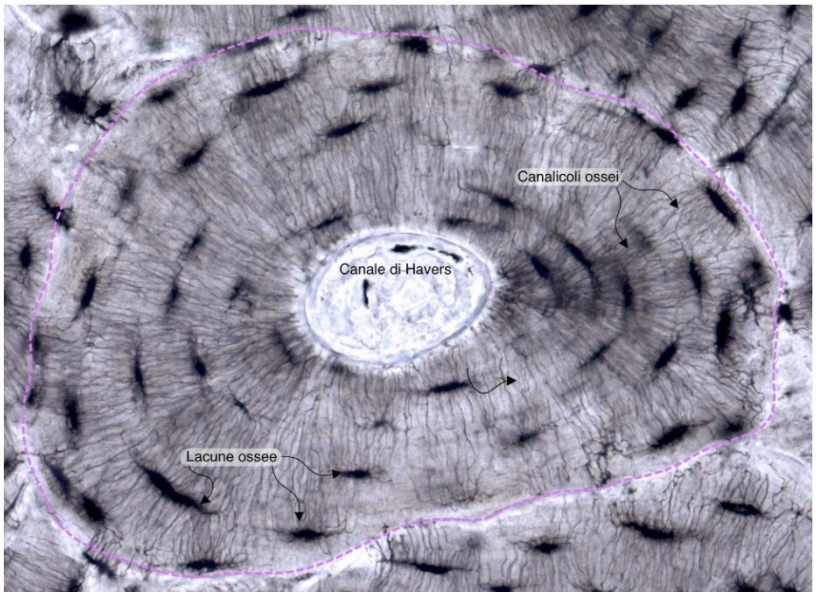


Figure 2.1: Human bone. Single osteon (outlined in pink), the concentric lamellar system surrounding the Havers canal. The cavities occupied by the osteocytes (bone gaps) and the canalicles where the cytoplasmic extensions of the osteocytes (bone canalicles) pass (Atlas of histology, Universià di Genova).

In particular, an imbalance between osteoblasts and osteoclasts, with a prevalence of action of osteoclasts, promotes the development of certain diseases such as osteoporosis, one of the main causes of osteolysis.

Bone consists of bone tissue and other connective tissues such as hematopoietic tissue, adipose tissue, blood vessels and nerve fibers, and can be divided into cortical (or compact) bone and spongious (or trabecular) bone (Fig. 2.2). The first consists of a dense and compact lamina that forms the outer part of the bone, while the second has a spongiform texture formed by trabeculae that form the inner part of the bone. The two types are distributed differently. In details, the compact bone is found mainly in the diaphysis (in the case of long bones such as the femur) and the spongious bone is localized in the epiphyses. Externally, the bones are covered with a dense lamina of connective tissue containing the osteoprogenitor cells, called periosteum, while the internal cavity of the bones is covered by the endosteum, composed of connective cells.

The bone cavity is called medullary cavity, inside which there is the bone marrow: the red one is formed by blood cells and it is designed to decrease in adulthood to be replaced by the yellow bone marrow, composed mainly of fat cells.[1]

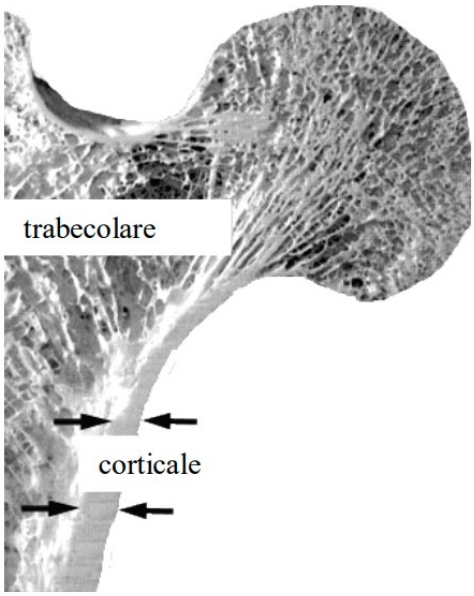


Figure 2.2: Morphological distinction between cortical and trabecular bone.[2]

2.2 Mechanical properties of the bone tissue

Bone is a dynamic tissue that has the function of mechanical support for muscle activity and protection of internal organs and soft tissues. Wolff's law expresses the relationship between bone and a mechanical load that allows the description of the skeletal system behavior as a dynamic structure. Wolff's law states that functional and morphological changes in bones result in structural

alterations of bone substance and changes in the external conformation of bones.[3] In particular, Wolff observed that bone trabeculae are designed not only to perform their specific structural function, but also to respond to the loads applied by varying their configuration; alterations are intended to modify the skeletal system to adapt its functional capacity during mechanical stimuli.[4]

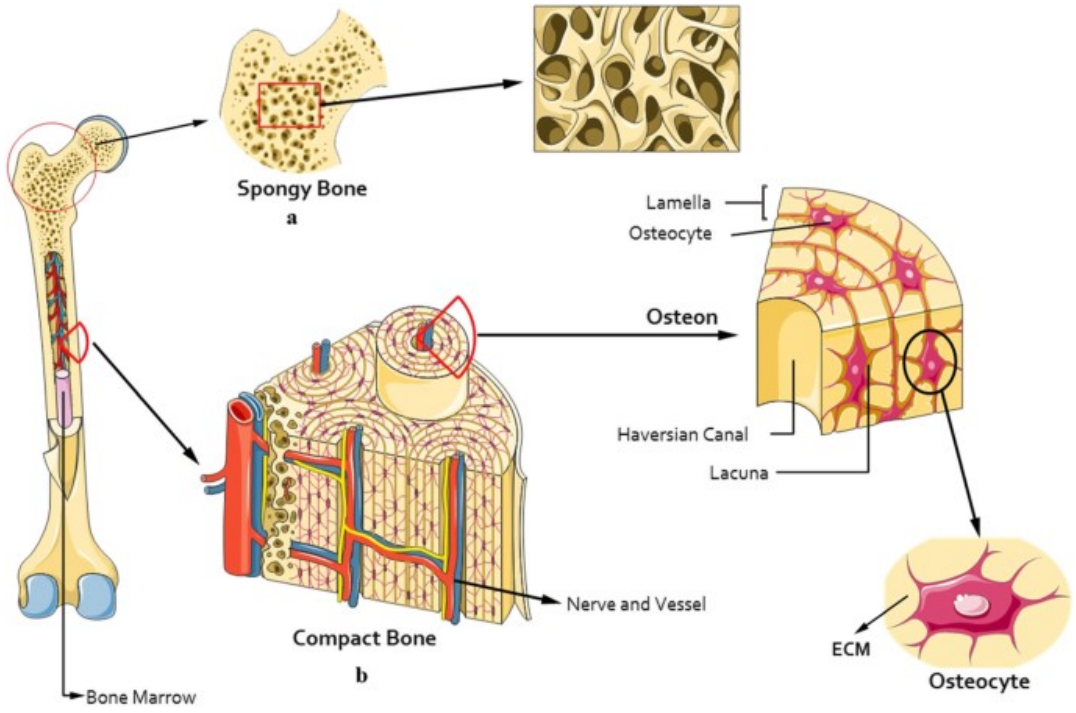


Figure 2.3: Explanation of bone structure. Two main types of bone include spongy (trabecular or cancellous) and compact (cortical) bone. a. The spongy bone is highly porous. b. The compact bone consists of osteon or haversian system, surrounded by the lamella which is made up of nerve and blood vessels.[5]

The mechanical behavior of bone can be assessed from the material properties or structural properties points of view, which together constitute the response of bone in vivo to the forces applied. In the first case, observations can be made on sections of bones, in the second case on whole bones. The bone is an anisotropic material, which means that its mechanical properties vary not only according to the magnitude of the applied force, but also depending on its rate and direction of application (Fig. 2.4, up). With regard to the distinction between trabecular and cortical bone, the former has a lower elastic modulus than the latter because it shows greater porosity. Furthermore, trabecular bone can withstand greater strains than cortical bone.

Our interest in this study relies on the femur and it's possible to consider the stresses and strains generated in its diaphysis: the stress (δ) is defined as the ratio between the force and the area onto which the force is applied; the strain (ε) is defined as the ratio between the difference between the deformed length and the original length over the original length.

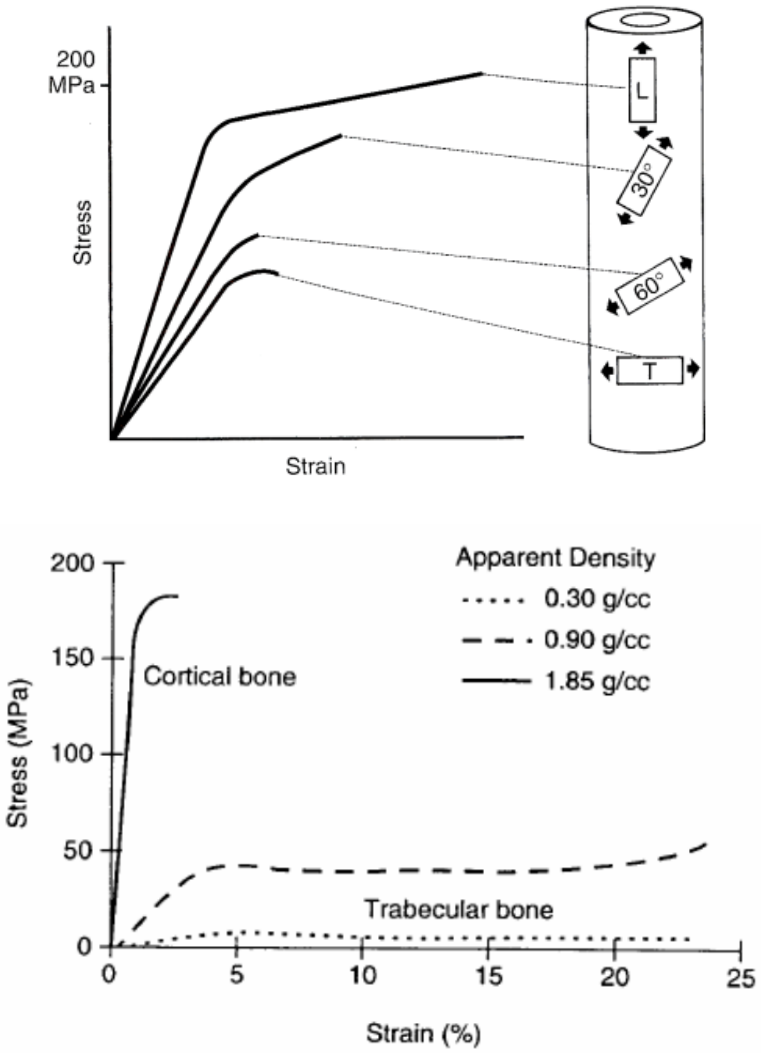


Figure 2.4: up: Stress-strain, tensile and compression curves of cortical bone in the directions longitudinal and transverse[6]*; down: Compression tests on sample of cortical bone and on sample of trabecular bone with different apparent densities.*[7]

It is possible to have the application of two types of internal forces, a perpendicular force, that produces a normal stress, and a shear force, that produces a shear stress (Fig. 2.5). We can assume that a thin portion of femur diaphysis may be considered as a small cube (or cylinder) onto which the forces are applied.

$$\delta = \frac{force}{area} \qquad \textit{Equation 1}$$

$$\varepsilon = \frac{deformed\ length - original\ length}{original\ length} \qquad \textit{Equation 2}$$

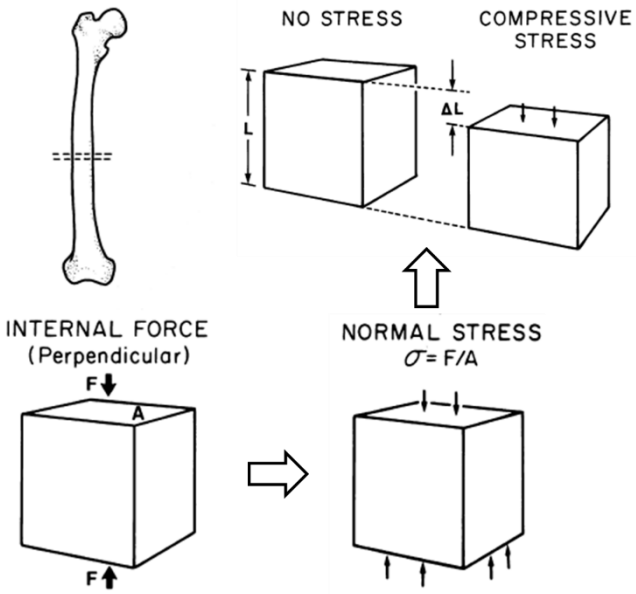


Figure 2.5:Schematic representation of internal force applied normally to the surface of a cube that causes the development of a normal stress; this results in a compressive local deformation.[4]

In this work, the focus is on the compressive stress, defined as a shortening by a distance L and for which the normal strain is defined as:

$$Normal\ strain = \frac{\Delta L}{L} = \varepsilon_N \qquad \text{Equation 3}$$

The application of great loads and the generation of large stresses may lead to a fracture in the bone.[4]

2.3 Anatomy of the human femur

The femur is the longest bone of the human body and it is fundamental in the locomotor system; it is articulated proximal to the hip through the coxo-femoral joint and distally to the tibia through the knee joint. The femur is composed of a central body, which forms the long part of the bone called the diaphysis, and two ends, a proximal and a distal, called epiphysis. In the standing position, the femur is tilted down and medially to align the knee to the center of gravity of the body forming the angle of anti-rotation. The diaphysis of the femur has a smooth and uniform surface, convex at the front and with a triangular prism section in the back.[8]

At the proximal end of the femur, there is the head of the femur covered with hyaline cartilage that articulates with the acetabulum in the hip joint. On the head, there is the so-called fovea capitis that allows the insertion of the head ligament. This ends with the anatomical neck of the femur; in particular, the axis of the head and neck forms an angle of 130° with the diaphysis and has a front torsion angle of 12°. At the base of the femoral neck, there are two reliefs, the large trochanter and the small trochanter, respectively inferiorly and medially.[8]

The great trochanter forms the bone prominence of the hip and determines the upper limit of the femoral diaphysis; medially to the great trochanter lies the trochanteric fossa which continues downwards as an intertrochanteric crest. The small trochanter is located in the junction between the lower margin of the femoral neck and the diaphysis and is connected to the large trochanter posteriorly by the intertrochanteric crest while anteriorly by the intertrochanteric line.[9]

The distal end of the femur is composed of two oval-shaped convex masses called lateral and medial femoral condyles, they are coated with cartilage and articulate with the tibia and the patellar face. On the back, the condyles are separated by a depression called the intercondiloid fossa. Above the condyles, there are two reliefs, the lateral and medial epicondyles that allow the insertion of ligaments. Above the medial epicondyle, there is the tubercle of the adductor on which the large adductor muscle is inserted, while between the condyle and the lateral epicondyle there is the popliteal groove on which the tendon of the homonymous muscle is inserted.[8]

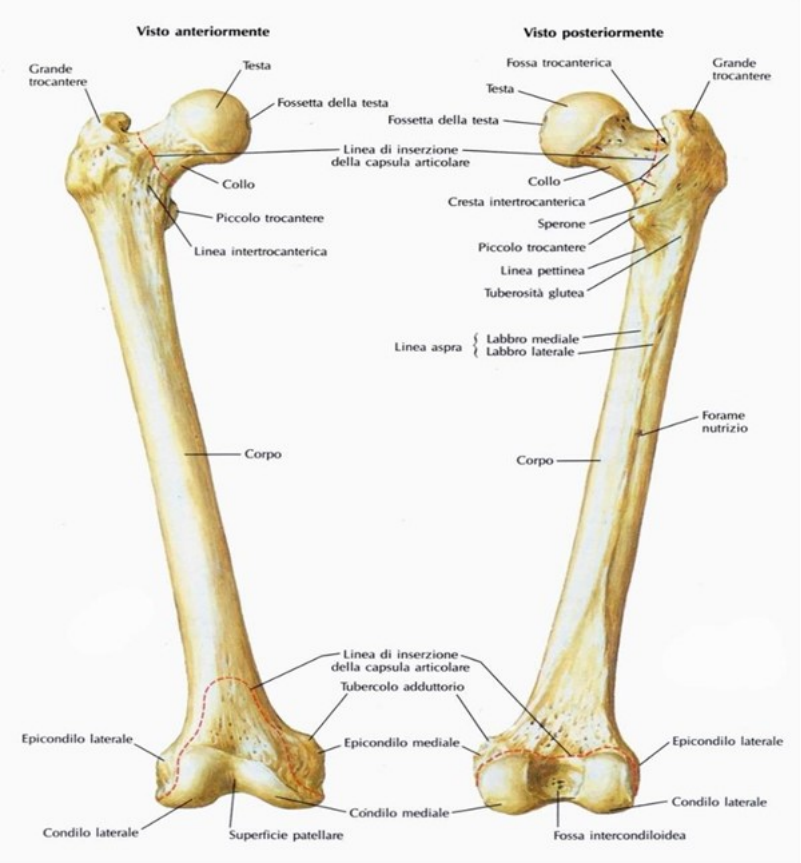


Figure 2.6: Front (left) and back (right) view of the femur.[8]

2.4 Osteolysis and femur fractures

Osteolysis refers to a process induced by wear particles that develop at the bone-metal or bone-cement interface of a prosthetic implant; osteolysis results in the loosening of implants and periprosthetic fractures caused by progressive destruction of bone tissue around the implant. Other

causes of osteolysis include intra-articular pressure, which increases during movement, micromotion at the bone-prosthesis interface, which causes bone resorption, a possible infection, developed during implantation, and host factors, which refer to the immune response to wear particles and the genetic predisposition to osteolysis.[10]

In this case study, the femur is treated, whose fractures can be divided into proximal femur fractures, diaphysis fractures and distal femur fractures, depending on their location. Regarding the fractures of the proximal femur, there are several anatomical regions in which osteolysis can take place, such as the subtrochanteric region, the intertrochanteric region, the great trochanter and the femoral neck region (Fig. 2.7). Intertrochanteric fractures occur in the transitional bone, composed of cortical and trabecular bone, between the femoral neck and the femoral shaft and may involve both the large and the small trochanter; they are classified as stable fractures, standard oblique fractures, and unstable, reverse oblique fracture.[11]

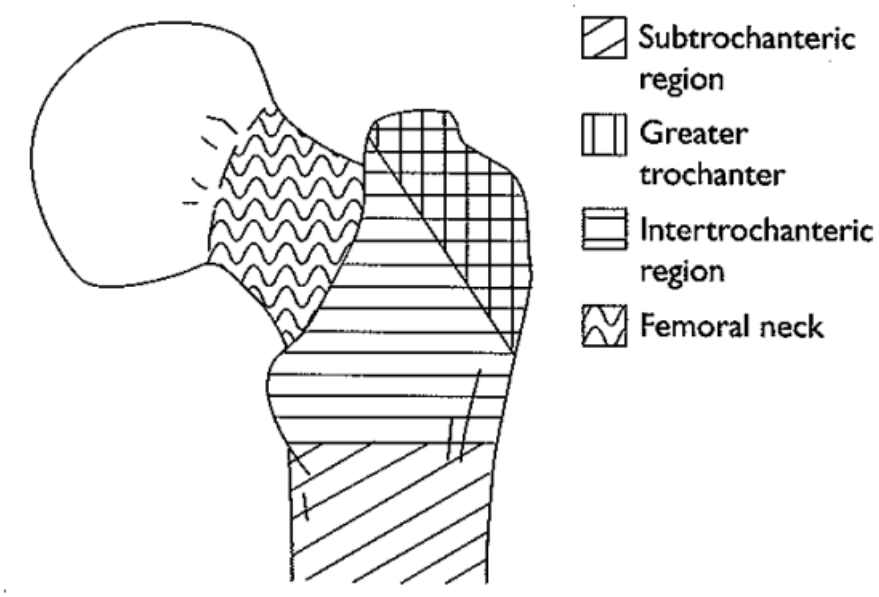


Figure 2.7: Anatomical regions in which osteolysis can occur proximally.[12]

Fractures of the femoral neck are intracapsular fractures and develop between the extremity of the articular surface of the femoral head and the intertrochanteric region, and are classified with the Garden system based on fracture displacement in type I, II, III, IV.

Subtrochanteric fractures develop between the small trochanter and the femoral diaphysis, the Russell-Taylor classification categorizes them into type II or type I, respectively, whether there is or not an extension of fracture posteriorly with involvement of the great trochanter and the piriform pit.

Finally, the fractures of the great trochanter can be divided into two types. The first and most common is the avulsion fracture of the great trochanter. It occurs in skeletally immature patients and it is

minimally decomposed. The second occurs in elderly patients with osteoporosis as a result of a direct trauma that causes the formation of a broken fracture.[12]

Concerning the fractures of the femoral diaphysis, there is a system of universal classification Arbeitsgemeinschaft für Osteosynthesefragen/Orthopaedic Trauma Association (AO/OTA), that is based on X-ray images and classifies the fractures identifying their position, morphological characteristics, fracture level and damage to the surrounding soft tissue. In Fig. 2.8 the universal classification system AO/OTA is represented; depending on the type of fracture, there are several osteosynthesis procedures.[13]

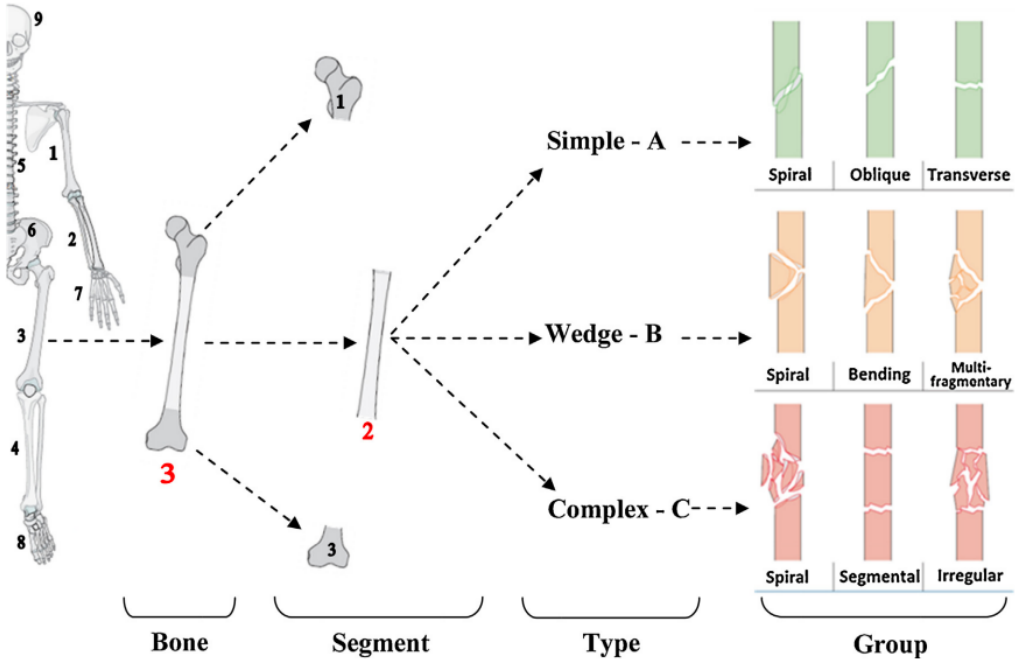


Figure 2.8: Diaphyseal femur fractures classification.[13]

About the distal femur fracture, also in this case the reference is the system of universal classification AO/OTA: hence we can have extraarticular fracture, partial articular fracture and complete articular fracture, involving the condyles and the epicondyles, each one with relative subgroups.[14] In Fig. 2.9 there are some examples.

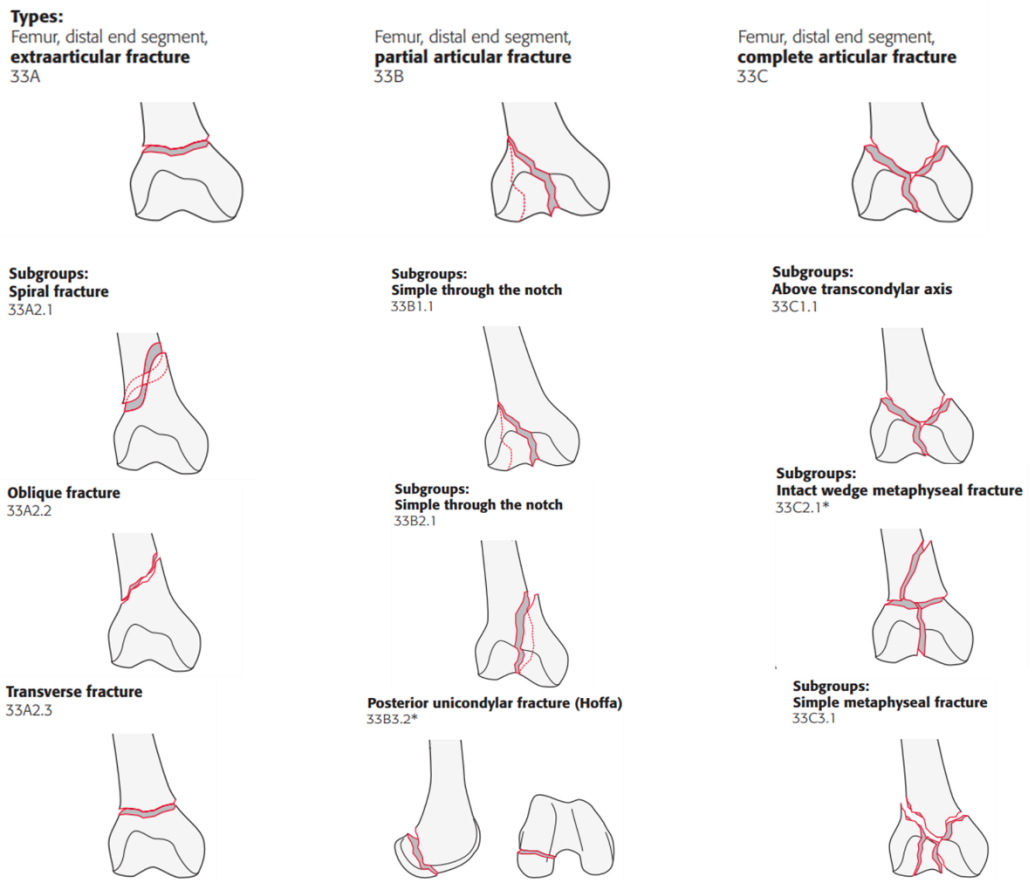


Figure 2.9: Examples of distal femur fractures classification.[14]

2.5 Characteristics of the femur implants

Femoral implants are used for example because of osteolysis, and differ from one another according to the type of fracture and its localization. In the case of proximal femur fractures, hip implants are the main solution, the components of which may be cemented or non-cemented femoral and cemented or non-cemented acetabular; in the case of diaphyseal fractures, the choice relies on locking plate compression (LCP) or dynamic compression plate (DCP), which showed a good result even in the case of fractures in the distal femur at the level of condyles and epicondyles. These systems allow to perform minimally invasive percutaneous osteosynthesis (MIPO) by using screw-plate devices.[15]

In our specific case, the focus is on diaphyseal fractures: DCP is one of the most used implants in internal fixings in the case of fracture, whose screws are pressed inside holes and compress the plate on the surface of the bone. The LCP has been proposed to exploit the advantages of DCP while avoiding vascular damage caused by the force of compression between bone and plate. The LCP device uses combined holes: one portion of them allows the use of compression screws as in the DCP, while the second one allows the use of locking head screws (Fig. 2.10).

Thus, the load is transferred without causing compression on the bone, allowing a proper blood supply and facilitating the healing of the fracture.[16]

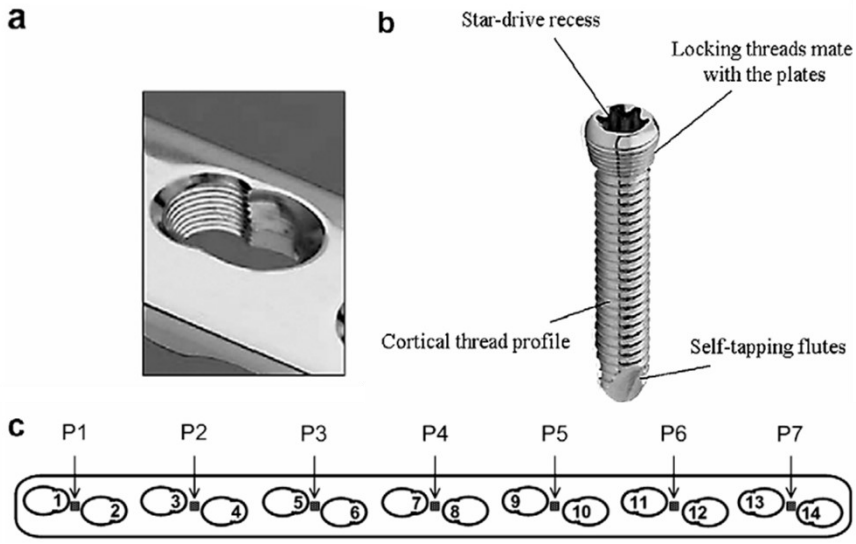


Figure 2.10: a: LCP locking and compression hole; b: locking head screw; c: example of LCP.[16]

Some factors need to be taken into account about the influence they have on the outcome of the use of LCP:

- the most recommended material is titanium which shows an elastic modulus more similar to that of bone compared to other metals;
- the length of the plaque and the density of the screws, which must be both non-locking and locking type to ensure the healing of the fracture;
- the most distal screw proximal to the fracture, which sets the working length of the system.

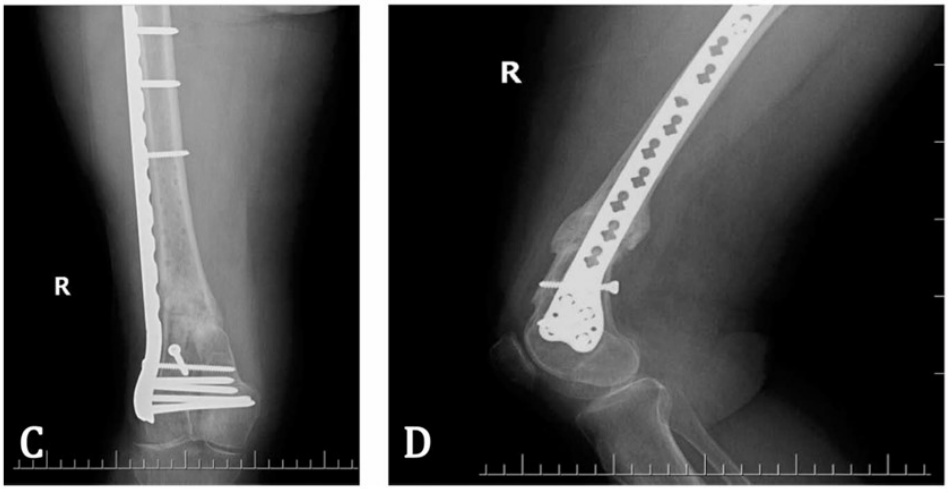


Figure 2.11:Antero-posterior (C) and lateral (D) vision of healed fracture by using a longer plate with multiple nonlocking screws proximal to the fracture and a longer working length.[15]

The optimal working length is currently unknown but it's crucial to fully understand how the plates affect fractures and how they lead to bone healing.[15]

2.6 Finite Element Method

2.6.1 Introduction to FEM basis

With the term Finite Element Method, we refer to a numerical analysis technique that allows solving engineering problems through equations and is used for the evaluation of stresses, strains and deformations. The application of external loads leads to the occurrence of internal forces and deformations that are directly influenced by the load conditions, the morphology and the architecture of the structure.[17]

The main idea of the method is to discretize complex geometries with simpler elements joined together in several ways. A flowchart of model-based simulation has been defined: starting from a physical system, a mathematical model is built up, from which it's possible to perform discretization to obtain a discrete model and then a discrete solution. As always, a solution needs to be verified and validated to be referred to the initial physical system (Fig. 2.12). Thus, the Finite Element Analysis can be used as a validation method for experimental approaches, by simulating with computer software the behavior of a complex structure also under the application of loadings.

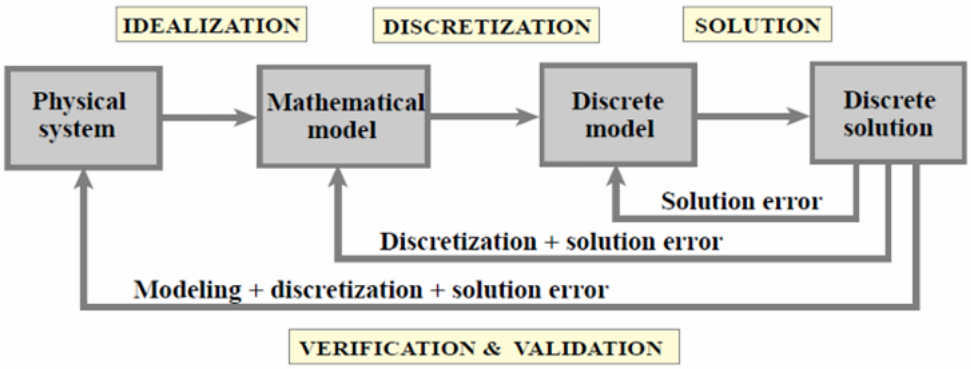


Figure 2.12: Flowchart of model-based simulation (MBS) by computer.[18]

Discretization is made possible by the definition and use of approximating functions in terms of the values of field variables at the nodes, which can be categorized into nodes that connect adjacent elements and interior nodes. In addition, a complex problem is thus divided into a multitude of simplified problems that are then assembled, following a step-by-step process.[18]

Starting from continuum discretization, it's possible to choose among a variety of different shapes, and, according to the type of components inside an element that needs to be discretized, is highly recommended to use different geometries inside the same region. The next step is to assign nodes to elements and choose the interpolation function with which to represent the variation of the field

variable on the element; it's then possible to obtain the matrix equations defining the element's properties, that are assembled to find the properties of the total system. After the imposition of boundary conditions is thus possible to solve system equations; the solution requires to find the displacement and stress distribution when a load is imposed on the structure, that in this study case is an axial compression of the femur.[18]

2.6.2 Continuum-Level FE and Femur model

Bones have different morphologies and shapes, related to their functions that are based on the conditions in which the bones are loaded. There are also very complicated loading scenarios that require complete descriptions and simple modelling is not immediate; for this reason, we rely upon the use of FE models.

The FE method allows obtaining a prevision of the fracture risk associated with the mechanical behavior of bone when a pathology like osteoporosis is present; it's possible to have two main approaches, continuum-level FE and microstructural-level FE. The case proposed for this study relies on the first method where the models are based on QCT imaging (Quantitative CT): this technique estimates the BMD (Bone mineral density) from which elastic moduli and bone strength in vivo of elements can be derived. However, Continuum-Level FE models are not able to represent the microarchitecture of bones that is necessary to better understand deformation mechanisms and failure. It is therefore possible to consider and introduce material degrees of complexity (e.g. inhomogeneity, anisotropy and nonlinearity) by increasing the computational time.[17]

Femur is an extremely important bone of the skeleton where the most fractures occur and because of this interest, it's widely studied as a whole FEA model. The aim of those studies is to assess the mechanical properties, the failure characteristics and stress and strain of the bone. Continuum femur models allow the study of fracture but don't consider the internal architecture of the bone: the geometries and properties of material are generated directly from QCT scans and, even if cortical and trabecular bone are recognised as different materials with their stiffness and density, the trabecular architecture that leads to bone anisotropy can't be considered.[19]

The trabecular bone has aroused particular interest since it has important implications for bone loss and the development of fractures, especially in the neck of the femur or at the level of the trochanter. Because it has different internal mechanical properties that depend on both anatomical localization and bulk density, internal microarchitecture and tissue properties, in our study that uses continuum-level FE models, trabecular bone is not taken into account.[17]

2.7 Digital Image Correlation

Digital Image Correlation (DIC) is a widely used optical technique in the biomechanical field; the main strengths of the method are its flexibility and scalability in a wide size range, the robustness of the operating principle and its ease of use.[20] The DIC technique allows full-field measurements, starting from the distributions of the grey levels of two images it is possible to calculate the field of the displacements point by point, from which to derive the field of the deformations.[21] For this reason, a set of images of the surface of a specimen in two configurations is used , one of reference not deformed and one deformed.[20]

The initial problem that the technique proposes to solve is the non-univocal correspondence between two points belonging to two different images, which is even more compromising in the case of non-rigid surfaces. The goal is therefore to use Image Matching methods to create a point match between two surface patterns representing different states of stress of the test piece, correspondence from which a full-range information of displacements and strain of each point of the surface considered is obtained.[22]

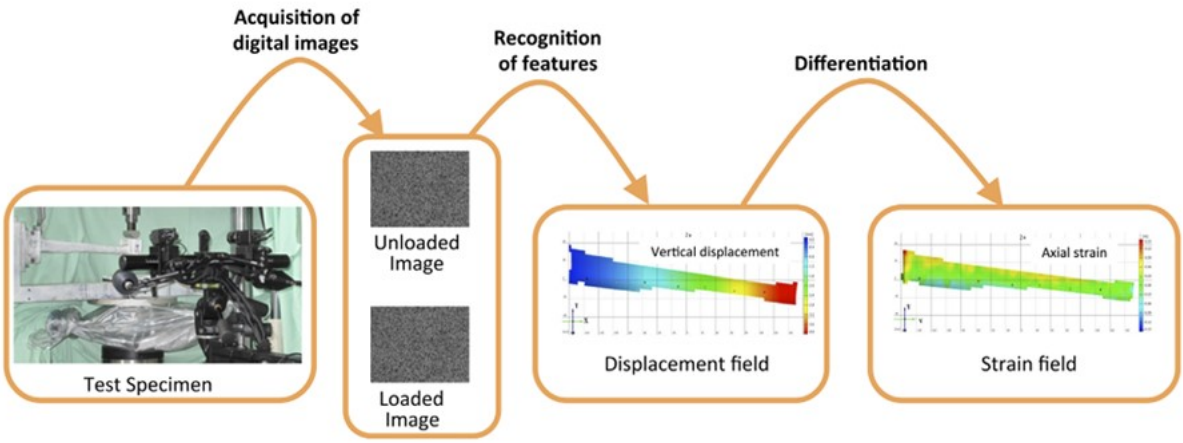


Figure 2.13: Workflow of DIC displacement and strain measurement.[20]

Consequently, the surface of the object to be analyzed must have specific properties: first, it should be isotropic with a non-periodic texture and, for this reason, the choice of the surface pattern to be applied to a specimen falls on the speckle, that adheres to the surface and is deformed with it.[22] The realization of the speckle pattern consists in the application of a spotted pattern on the sample, usually black on a white background, and must meet some requirements:

- the distribution of the pattern must be random, so that each area of the surface is uniquely identifiable;
- high contrast and minimal roughness is required;
- the optimization of the size of the single point of the pattern must be performed (usually corresponds to 3-5 pixels) in relation to the measurement window considered.[20]

The DIC technique includes a series of parameters to be optimized to determine the accuracy, precision and spatial resolution of the results:

- Facet size;
- Grid spacing;
- Strain computation window;
- Validity quote;
- Filtering. (Fig. 3.14)

Facet size

Facets are sub-images of M x N pixels into which digital images are divided and a single facet is represented by a grey-level distribution. This distribution must be interpolated by a bi-cubic spline to obtain an approximation of grey-scale between adjacent pixels. There are some specifications to be respected in the choice of the facet size:

- The specimen size;
- The size of the speckles;
- The strain gradients depending on loading conditions and anatomy.

Grid spacing

Grid spacing is the distance between two consecutive facets and it refers to the density of the facets inside the measurement window. In terms of influence on the precision and accuracy of the displacement field, grid spacing has minimal influence while overlaps bring advantages.[23]

Strain computation window

The choice of a suitable strain computation window is essential for managing noise attenuation and strain gradients. Thus, a larger window reduces the noise in the strain distribution but it can cause also the attenuation of strain gradients with consequent damage for specimens characterized by irregular geometries.[22]

Validity quote

The validity quote is the tolerance for computing/ignoring the strain in a certain region. This depends on the number of valid neighbouring points: if they are under a threshold, the strain in that region is excluded. If the value of the validity quote is high, the strains are considered and computed when redundant information is available. Also, if the geometry of the specimen is complex, the threshold is necessary.

Filtering

Strain distributions are obtained by derivation of displacement fields and derivation enhances the noise in the latter, acting as the opposite of a filter. In this way, the strain fields result affected by

large random errors. That is the main reason why filtering is applied to digital images, to DIC-computed displacement field and/or DIC-computed strain field. On the other hand, the application of filters results in a smoothing of gradient or stress/strain concentration and consequent loss of information. To be considered is also the continuum assumption which acts as a low-pass filter and it is intrinsic to FE modelling.

The comparison between grey level distribution and the calculation of displacement and strain field can be done using two main approaches: a Local DIC Approach and a Global DIC Approach.[21]

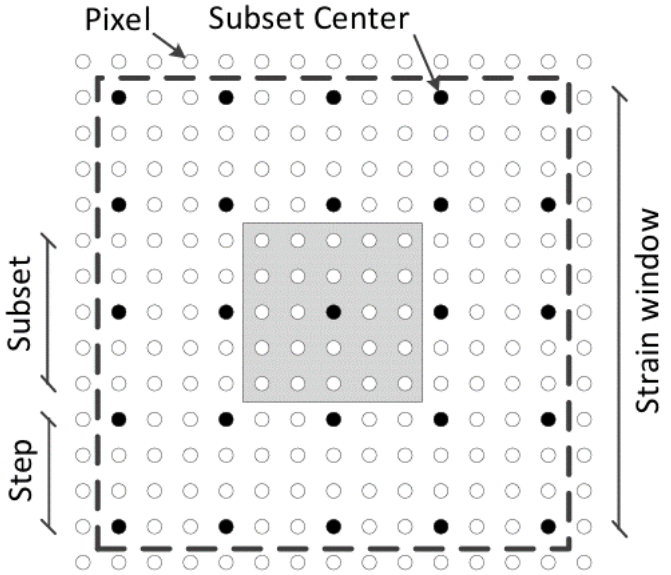


Figure 2.14: Example of step, subset, subset centre, pixel and strain window.[21]

2.7.1 Local DIC Approach

The local method is called also subset method and allows locating the same pixel on the reference image and the deformed one; the information to do this are located in the subset, a group of pixels around the pixel considered at each iteration.[21] The idea is to optimize a correlation function that represents the difference between the subset of the reference image and that of the deformed image; another possibility is to use a minimization criterion as in the case of ZNSSD or SSD.[24]

ZNSSD stands for Zero-mean Normalised Sum of Squared Differences and it is a matching algorithm that uses a correlation value to represent the similarity between two images. This can be done considering the relationship between the grey levels of a target image and of the image to be matched. The relationship allows to obtain a correlation description that is very sensitive to illumination change and image noise.[25]

SSD stands for Sum of Squared Differences and it is a template matching method in which the measurement of compatibility between a template image and an input image is based on the degree of difference in their pixel intensity.[26]

We define *f* as the reference image and *g* as the deformed one, *(x,y)* is a coordinate system and *(xsc,ysc)* are the central pixel coordinates of a dimension subset (2N+1)•(2N+1) in the image *f(x,y)*. Using r_{ZNSSD} as a correlation coefficient:

$$\sum_x \sum_y \left[\frac{f[x,y] - \bar{f}}{\sqrt{\sum_x \sum_y \left(f[x,y] - \bar{f}\right)^2}} - \frac{g[x,y;\boldsymbol{s}] - \bar{g}}{\sqrt{\sum_x \sum_y (g[x,y;\boldsymbol{s}] - \bar{g})^2}} \right]^2 \qquad \text{Equation 4}$$

$$\Sigma_y = \sum_{y=y_{SC}-N}^{y_{SC}+N} \, , \Sigma_x = \sum_{x=x_{SC}-N}^{x_{SC}+N}$$

The vector of the unknown parameters *s* creates a relationship between the coordinates of the reference image *f* with those of the deformed image *g* through the so-called shape functions of the subset; the polynomial order of these functions and the number of parameters of the unknown vector *s* define how the subset deforms. The value of the unknown parameters is derived by minimizing *Eq. 4* with an iterative process that evaluates the values of grey at the non-integer positions of pixels; in this way you get the displacement field for each pixel when taken as the centre of the subset. From the displacement field by derivation, we can calculate the strain field: displacements are approximated as bilinear Lagrange polynomials (*Eq. 5*) multiplied by the unknown sets of parameters a_{ij}^u and a_{ij}^v, from which we derive the deformation gradient **F** (*Eq. 6*). From this, choosing an appropriate strain tensor yields the strain field.[24]

$$u(x_0, y_0) = a_{ij}^u x_0^i y_0^j$$
$$v(x_0, y_0) = a_{ij}^v x_0^i y_0^j \qquad \text{Equation 5}$$

$$F^{Q4} = \begin{bmatrix} 1 + a_{10}^u + a_{11}^u y_0 & a_{01}^u + a_{11}^u x_0 \\ a_{10}^v + a_{11}^v y_0 & 1 + a_{01}^v + a_{11}^v x_0 \end{bmatrix} \qquad \text{Equation 6}$$

2.7.2 Global DIC Approach

Another approach for DIC is the global method where, unlike the local approach, it tracks all pixels within the analysed element for which a complete mesh tracking is obtained. As with the previous approach, there are the functions *f*(**x**) and *g*(**x**) that represent the grey values in the position with coordinates **x** = *(x,y)* respectively in the reference image and in the deformed one. The method is based on the conservation of the optical flux whereby **d** is defined as the displacement field not known for an element Ω_e so that for each point the relation applies: *f*(**x**-**d**) = *g*(**x**). As in the local approach, there is the definition of functions, such as SSD or ZNSSD, to be used to minimize the difference between *f*(**x**-**d**) and *g*(**x**) (*Eq. 7*).

$$\epsilon^2 = \iint_{\Omega_e} ([f(\boldsymbol{x} - \boldsymbol{d}) - g(\boldsymbol{x})]^2) d\boldsymbol{x} \qquad \textit{Equation 7}$$

$$\epsilon^2 = \iint_{\Omega_e} ([-\boldsymbol{d} \cdot \nabla f(\boldsymbol{x}) + f(\boldsymbol{x}) - g(\boldsymbol{x})]^2) d\boldsymbol{x} \qquad \textit{Equation 8}$$

$$\boldsymbol{d} = \sum_{i,a} \phi_i \delta_{ia} \boldsymbol{e}_a \qquad \textit{Equation 9}$$

The displacement field is defined by equation *Eq. 9*, as the default function:[27]

- *i*: number of the shape function;
- *a*: system directions;
- δ_{ia}: displacement parameters
- ϕ_i: arbitrary basis.

Minimizing *Eq. 8*, which is the first order Taylor expansion of *f*(**x**-**d**) (*Eq. 8*), approximate displacements are obtained by applying an iterative algorithm to the Taylor expansion. To obviate this criticality, an additional shift **d'** is applied (*Eq. 10*); thus, knowing the relationship defined in *Eq. 11*, knowing $[K_{ijab}]^e$ and $[F_{jb}]^e$ (*Eq. 12* and *Eq.13*), δ_{ia} is obtained and so **d**, which is the complete displacement field within the element Ω_e that can be used for next iterations.

$$f(\boldsymbol{x} - \boldsymbol{d}) = g(\boldsymbol{x} + \boldsymbol{d}') \qquad \textit{Equazione 10}$$

$$[K_{ijab}]^e [\delta_{ia}]^e = [F_{jb}]^e \qquad \textit{Equazione 11}$$

$$[K_{ijab}]^e = \iint \phi_i [\nabla_a f] \phi_j [\nabla_b f] d\boldsymbol{x} \qquad \textit{Equazione 12}$$

$$[F_{jb}]^e = \iint [g(\boldsymbol{x} + \boldsymbol{d}') - f(\boldsymbol{x})] \phi_j [\nabla_b f] d\boldsymbol{x} \qquad \textit{Equazione 13}$$

- *e*: element number;
- $\nabla_{a,b}$: derivative function of a and b ($a,b \in \{x,y\}$);
- [K] and [F]: matrixes based on grey values of reference and deformed images.

The solution of the system for all mesh elements results in a displacement field for each element of the mesh without interconnectivity between them; from the displacement field by derivation is then possible to obtain the strain field.[21]

2.7.3 Subset Shape Functions

The method used for this thesis is the Local DIC Approach which is limited by a square pixel subset between the reference image and the deformed one; since the DIC technique is used here for the analysis of non-elementary geometries (the human femur) in compression, the calculation of

complex displacement fields is required. The iterativity of the matching method allows, however, to be extended to deformations and not to remain limited to translations, through the introduction of the shape functions $\xi(\boldsymbol{x},\boldsymbol{p})$. The purpose of a shape function is to transform the coordinates of a pixel in the reference subset into the coordinates of the deformed image; it is therefore possible to define an SSD cost function to be minimized with respect to **p** vector of the parameters and in which $\chi^2(\boldsymbol{p})$ defines residues squared between the original subset and the deformed subset (*Eq. 11*).

$$\chi^2(\boldsymbol{p}) = \sum (G(\xi(\boldsymbol{x},\boldsymbol{p})) - F(\boldsymbol{x}))^2 \qquad \textit{Equazione 14}$$

It has to be taken into account that the use of high-order shape functions leads to complication in the correspondence between reference and deformed images and thus two main considerations can be pointed out. First of all, the shape function must be chosen to represent accurately the displacement field over the size of the subset; then, the shape function has also low pass filtering effect, that can be limited by minimizing the subset size, since it should be avoided.

To summarised, DIC algorithm is able to work for complex deformation fields by introducing subset shape functions that have to be chosen carefully for their low pass filtering effect that limits the spatial resolution of the technique.[22]

3 Preliminary Set-up: Cemented Femur

The first part of the work was focused on performing DIC acquisition of a cemented femur previously used for other researches with an experimental set-up that was already defined. After the application of an axial compressive load on the femur head and the acquisition and post-processing of images, the next step was to create a FEM model of the same femur using Abaqus. This part of the work consisted of the load application and definition of constraints and boundary conditions, starting from a .stp file of the same femur tested with DIC. The analysis has been performed and, using MATLAB for post-processing, it was possible to test the validation process of FEM model on DIC results, in terms of displacements and strains.

3.1 Materials and Methods

3.1.1 Preliminary test on a simil-sawbone cylinder

Before performing a test on a sawbone femur, a preliminary test was done on a small white cylinder with a diameter similar to that of the femoral diaphysis. The cylinder was sprayed with black paint to create the characteristic speckle for DIC acquisition and it was then fixed onto the servo hydraulic testing system ElectroPuls® E3000 Linear-Torsion (Instron, UK) with 5kN load cell using a grip. A local reference system was created using an L-shaped rubber template on which three white dots as markers were placed to identify the axes and the origin of the frontal plane (Fig 3.1). The use of this reference system has allowed in the post-processing on MATLAB to create an algorithm with which to align the acquired images with the global reference system.

```
img=imread('img-cal-0000_0.tif'); % image loading
figure()
imshow(img);
[x,y]=ginput(2); % manual selection of 2 points to define the rotation axis
alpha=rad2deg(atan2((y(2)-y(1)),(x(2)-x(1))));
img_rot=imrotate(img,alpha); % image rotation
clear x y
imshow(img_rot);

[x,y]=ginput(2); % definition of a rectangular cropping area
rect=[x(1),y(1),x(2)-x(1),y(2)-y(1)];
img_crop=imcrop(img_rot,rect); % image cropping
imshow(img_crop);
clear x y

images.alpha=alpha;
images.rect=rect;

% Automation of the process for all images
for i = 1:5
    name = sprintf('img-cal-000%d_0.tif',i);
    img = imread(name);
    figure(i)
    imshow(img)
    img_rot = imrotate(img,images(1).alpha);
    imshow(img_rot)
    img_crop = imcrop(img_rot,images(1).rect);
    imshow(img_crop);
    images(i).crop = img_crop;
end
```

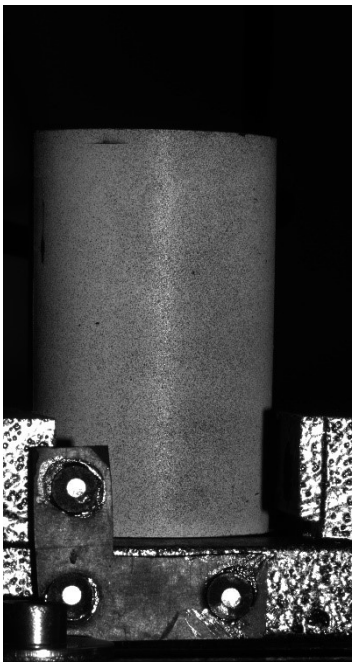


Figure 3.1: MATLAB script for automation of DIC images rotation and cropping; cylinder with speckle pattern and local reference system mounted onto Instron, acquired by DIC 0 camera.

3.1.2 Femur set-up preparation and DIC acquisition

For the first part of this work of thesis, a Sawbone left femur was used and fixed inside a rigid support. In particular, the femur was sprayed with white paint and then the speckle pattern was made using a spray black paint (Fig. 3.2).

Figure 3.2: Left Sawbone femur with speckle pattern used for the test.

The femur was oriented at an angle of 9° with respect to the body's vertical axis and at an angle of 6° with respect to femur mechanical axis on the frontal plane (Fig. 3.3).

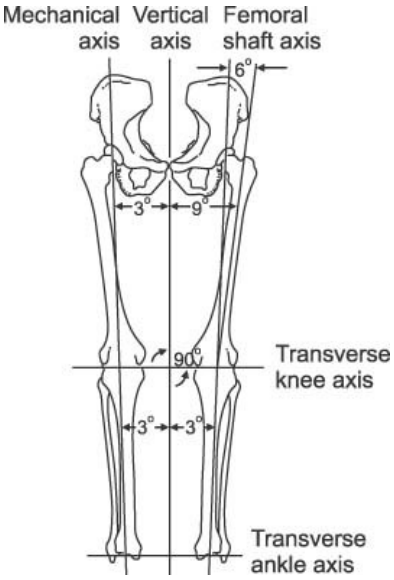


Figure 3.3: Relationship between mechanical axis and anatomical axis in normal lower extremity.[28]

The femur was then inserted and fixed using cement inside a steel cub bracket. The steel cup bracket was then fixed on the ElectroPuls® E3000 Linear-Torsion (Instron, UK) using a mobile guide and DIC cameras were positioned according to distances and configurations that allowed the best image resolution achievable together with the most accurate calibration possible (Fig 3.4). The DIC system used for this work consists of two cameras placed at the same distance from the specimen (Fig. 3.5) and, adjusting the light, it was possible to enlight properly the ROI on the femural diaphisys for the analysis.

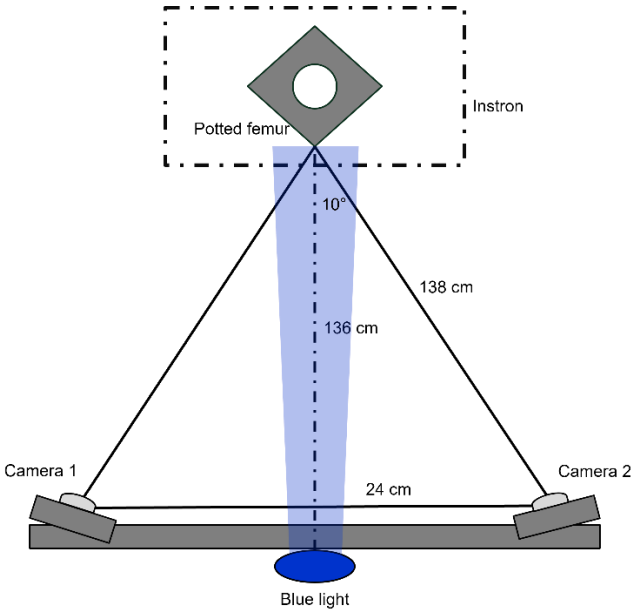


Figure 3.4: Schematic view of the set-up used for DIC acquisition.

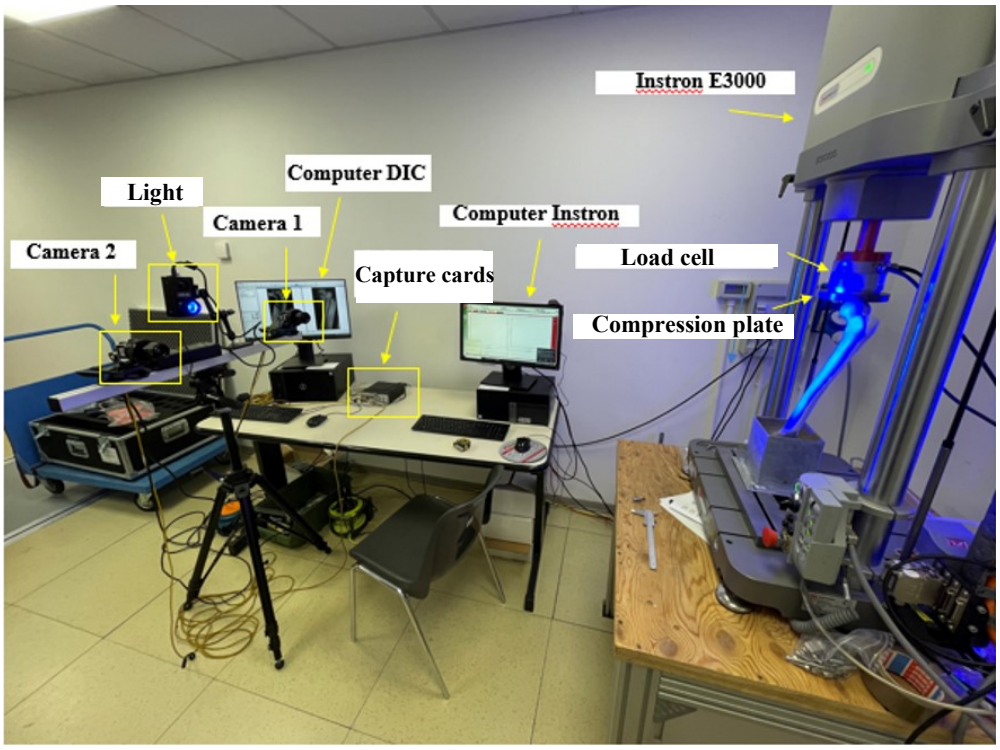


Figure 3.5: DIC cameras positioning with respect to femur onto Instron.

The acquisition of images, both calibration and test ones, has been done using VIC-Snap on ElectroPuls® E3000 Linear-Torsion (Instron, UK) computer. After the procedure of calibration for both the cameras, a zero-load test has been performed to obtain an undeformed configuration of images: this allows to measure the uncertainties of the acquisition. Systematic and casual error are evaluated respectively as mean and standard deviation of principal strains (inserire valori). Actually, for a better evaluation and understanding of background noise, the femur has been subjected to two other tests, one with ElectroPuls® E3000 on but inactive and another with ElectroPuls® E3000 moving without interfacing itself with the femur.

For the acquisition of test images, the chosen load was an axial compression load of 1600kPa, applied on the head of femur. After the acquisition images were obtained as .TIF files and for each one a .csv file was reporting coordinates, displacements and strains for each point of the specimen, using VIC-3D for ROI identification and post-processing. In Fig 3.6 the femur diaphisys is displayed in VIC-3D for both cameras.

3.1.3 VIC-3D for image post-processing

VIC-3D has been used after DIC acquisition with VIC-Snap to process images and to obtain a .csv file for each image, containing all the necessary information for MATLAB importation and elaboration.

The first step using the software was to import and identify calibration images, as well as test images, and to validate the calibration for both cameras, named as 0 and 1. After the calibration, since the

projection error was high, the autocorrection was performed. When running the calibration, in the Camera 1 image it's possible to choose a point and predict a line along which the same point must lie in the Camera 2 image. The distance in pixel of the point away from the prediction line, is called projection error and it should be in 0.02-0.05 rage of values. A high projection error may be caused by motion blur, synchronization issues, a poor calibration of cameras or cameras that are not rigidly mounted.[29]

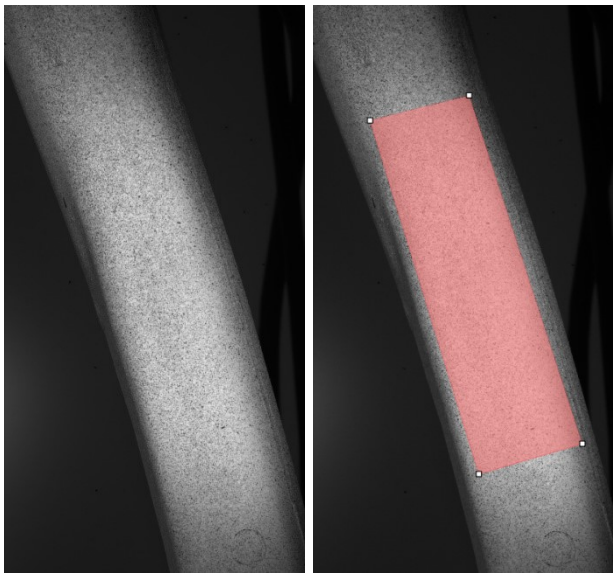


Figure 3.6: Femur diaphysis acquired with DIC camera 1; ROI definition using VIC-3D.

Next it was necessary to define a ROI in correspondence with femur diaphysis, where the illumination and the speckle pattern appeared to be better. The analysis has been performed calculating as outputs displacements and strains, so the gradient was displayed as grow ranges only. As setting, the subset size was 100, the step 30 and the projection error in pixel was 0.015 after the autocalibration correction.

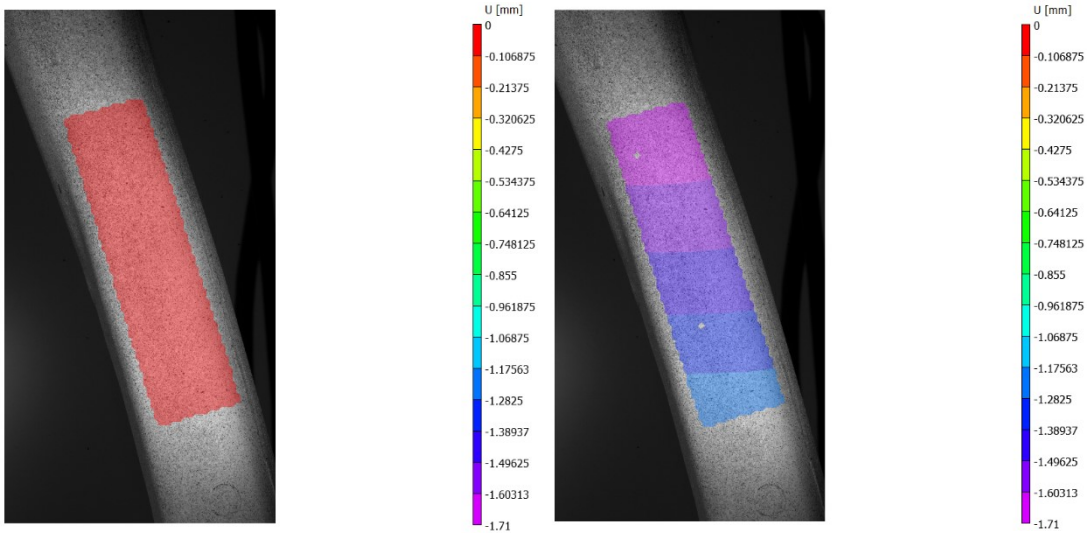


Figure 3.7: Displacement U for the first and last acquisition.

The main interest was U [mm] displacement and ε_1 [με] and ε_2 [με]; in Fig. 3.7, 3.8 and 3.9 displacement and principal strains for the first and the last image acquired are depicted together. For what concerns strains, negative values indicate a compression, positive value stand for a traction.

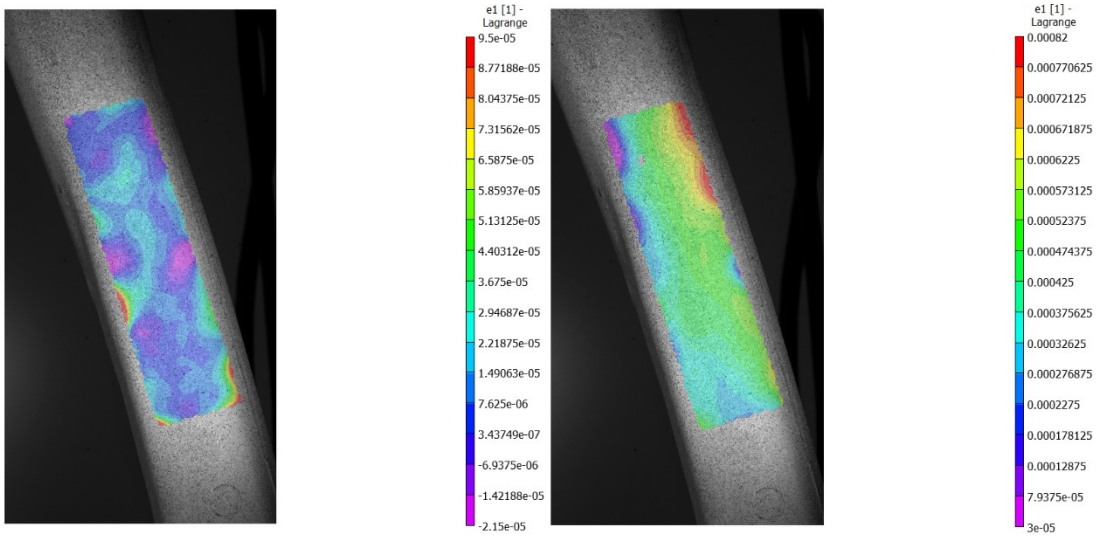


Figure 3.8: Strain ε_1 for the first and last acquisition.

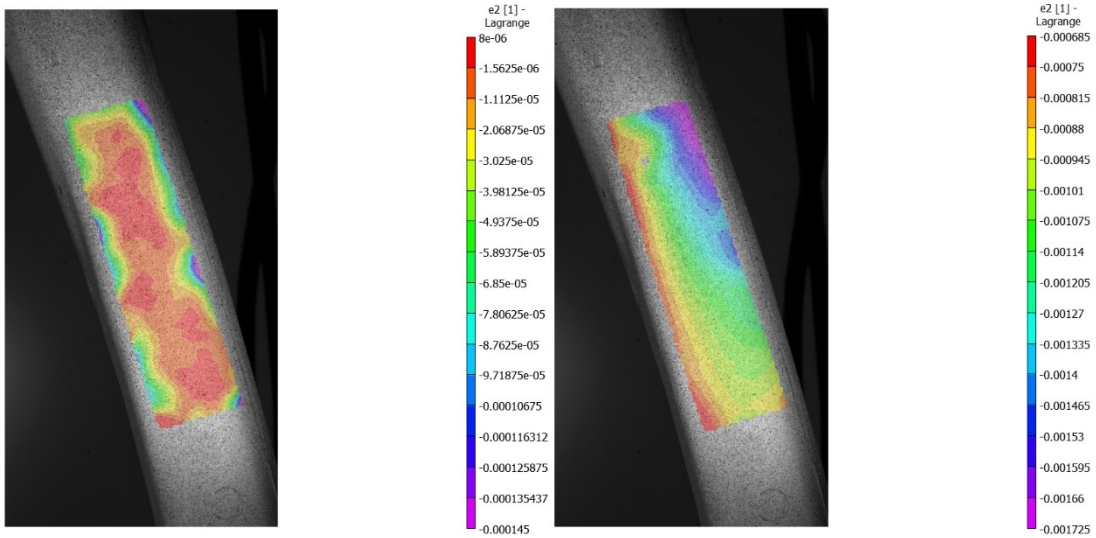


Figure 3.9: Strain ε_2 for the first and last acquisition.

3.1.4 Finite Element Model and Analysis in Abaqus

To validate FEM model with DIC acquisition it was necessary to build a model starting from the .step file of a left human femur that has been imported at first in Ansys SpaceClaim. In this step the femur has been turned into a solid to be used as a model for Finite Element Analysis.

The solid femur has been imported in Abaqus as a part and considered as a solid section (Fig. 3.10) for which the definition and assignment of material has been done: the elastic properties were set with reference to Sawbone as 16,7 MPa for Young modulus and 0.3 for Poisson's ratio.

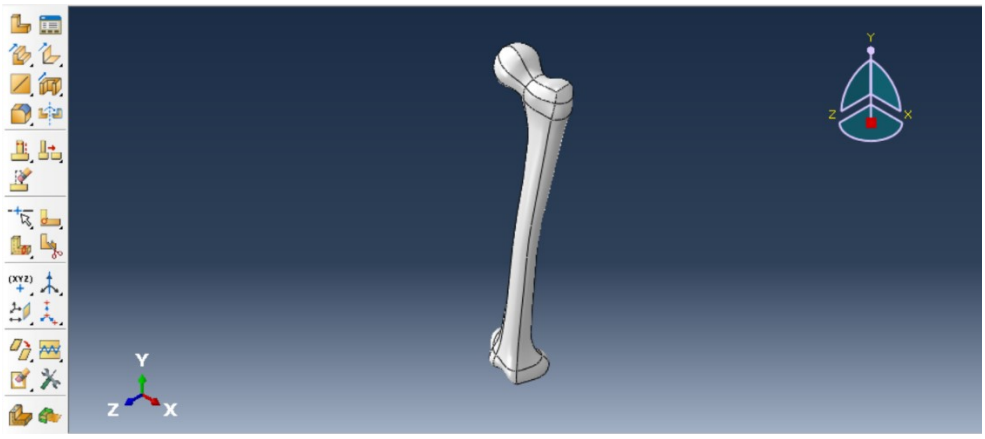


Figure 3.10: Solid femur imported in Abaqus.

From the geometry, three sets of nodes were defined as condyles nodes, head nodes and nodes of all the femur. This step allowed to consider in a more specific way just condyles and head regions for the application of the load and the boundary condition (Fig. 3.11).

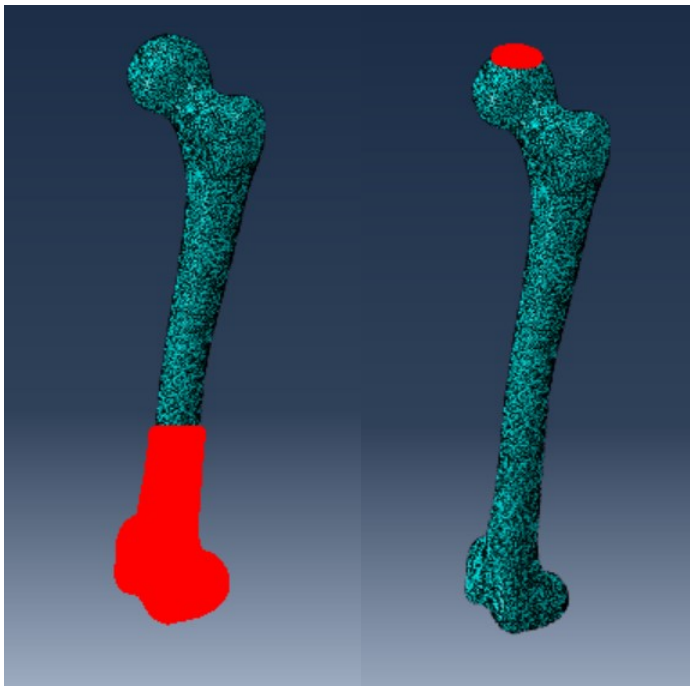


Figure 3.11: Groups of nodes for condyles and head regions chosen from the mesh.

The mesh chosen for the femur was a tetrahedra mesh (with a size of 2mm) from which an orphan mesh has been extracted and used to define the instance as part of assembly. Two reference points were also defined: one on the femur head for the application of the load and the other on the condylar

region for the definition of the boundary condition. The load was applied as a concentrated force of 1600N(-ŷ), according to the experimental load performed during DIC acquisition, in correspondence of the reference point on the femur head. The boundary condition was applied as an encastre in the reference point on the condylar area (Fig. 3.12).

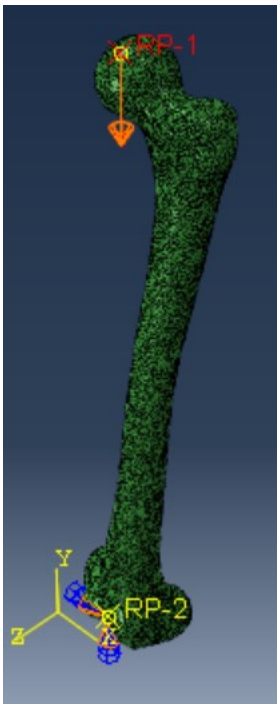


Figure 3.12: Reference points used for load and boundary condition application.

For the application of load, it was necessary to define an initial step of analysis in which the load was not applied and a load step in which the application of the concentrated force has been considered as a ramp with a normalized amplitude of 1 and a step time of 1s. The field output request was checked to obtain stresses, strains, displacements, forces and contact as outputs.

After the creation of a job, it was possible to launch the analysis and obtain the requested output as .inp file from which extract initial and final coordinates of the nodes. It was also important to consider the skin of the femur model from Abaqus, since the DIC ROI has been defined on the surface of the femur: for this reason, the extraction of the nodal superficial coordinates has also been performed.

3.1.5 MATLAB implementation for FEM validation through DIC

The first part of the work done in MATLAB consists of the definition of a set of FEM initial and final coordinates extracted from the solid femur for correspondence with femur skin. This step was necessary to have a range of z values that matched with the one from DIC ROI, considering just the surface of femur. At first, the remotion of outliers was performed on DIC data to perform a better alignment of DIC and FEM point clouds. Since FEM and DIC referred to two different reference

systems, the calculation of centroids for both datasets has been done: a centroid is considered as the mean of all position of a set of coordinates (Fig. 3.13).

```
53  % centroids
54
55  pfem = mean(F0);    % mean of the positions of the datasets
56  qfem = mean(F1);
57  F0c = F0-pfem;   % Initial position centered into the origin
58  F1c = F1-qfem;   % Final position centered into the origin
59  displFEM_c = F1c - F0c; % FEM displacements
60
61  pdic = mean(D0);
62  qdic = mean(D130);
63  D0c = D0-pdic;
64  D130c = D130-qdic;
65  displDIC_c = D130c - D0c; % DIC displacements
```

Figure 3.13: F0: initial FEM coordinates; F1: final FEM coordinates; D0: initial DIC coordinates; D130: final DIC coordinates.

To align FEM and DIC point clouds, it was necessary to perform a rigid rotation and a rigid translation on DIC point cloud; by defining maximum and minimum in x, y, z directions of DIC data, it was possible to create a ROI also for FEM (Fig. 3.14).

```
82  % Rigid rotation and translation
83
84  alfa = deg2rad(153);
85  Az = [cos(alfa) sin(alfa) 0 ;
86        -sin(alfa) cos(alfa) 0;
87        0 0 1];
88
89  T = [11,0,20];
90
91  rot = rigid3d(Az,T);
92  pc_dic_rot = pctransform(pc_dic0,rot);
93
94  % Definition of FEM ROI
95
96  minx = min(pc_dic_rot.Location(:,1));
97  miny = min(pc_dic_rot.Location(:,2));
98  minz = min(pc_dic_rot.Location(:,3));
99
100 maxx = max(pc_dic_rot.Location(:,1));
101 maxy = max(pc_dic_rot.Location(:,2));
102 maxz = max(pc_dic_rot.Location(:,3));
103
104 roi = [minx maxx miny maxy minz maxz];
105 index = findPointsInROI(pc_fem0,roi);
106 pc_femroi = select(pc_fem0,index);
107 nodi_indx = nodi(index,:);
```

Figure 3.14: Rigid rotation and translation, ROI definition for FEM data.

The algorithm chosen for perform the alignment of DIC and FEM point clouds was ICP, Iterative Closest Point (Fig. 3.15): in MATLAB the definition of a 3D object is required, as result of a rigid transformation, which contains all the information in translation and rotation for the alignment (line 120). Then, this transformation is applied to DIC data by considering FEM data as the reference set. ICP algorithm was first invented by Besl et al.[30], which states that a set of data is moved to be

align to a reference set of data that is fixed and considered as a model. The theorem states that “the ICP algorithm always converges monotonically to a local minimum with respect to the mean-square distance objective function”. The rigid transformation applied before is useful for the algorithm to minimize the distance between a point in the moving data set and a point in the reference data set. As displayed in line 120, when ICP is applied with “Verbose” set as “true”, the function gives the RMSE value for the point-to-point relation between the two data sets.

```
118    % ICP Algorithm
119
120    tform = pcregistericp(pc_dic_rot,pc_femroi,'MaxIterations',700,'Tolerance',[1e-12 1e-7],'Verbose',true);
121    mskReg = pctransform(pc_dic_rot,tform);
```

Figure 3.15: MATLAB script that displays the application of ICP algorithm using DIC data as moving point cloud and FEM data as fixed point cloud.

For comparison, displacements for coordinates after the rigid transformation and after the ICP algorithm are calculated, as well as displacements for FEM coordinates in ROI region. It was also important to check for modules correspondence: in particular, after rotation and translation, the modules did not change, while performing the ICP algorithm changes the mutual distances between nodes in the point cloud (Fig. 3.16).

- *Mod_0*: modules of DIC displacements in coordinates centred in the origin;
- *Mod_R*: modules of DIC displacements in coordinates after the rotation and translation;
- *Mod_T*: modules of DIC displacements in coordinates after ICP algorithm.

$$mod = \sqrt{\sum displacements^2} \qquad \textit{Equation 15}$$

```
225    % Displacements and modules
226
227    displR = DR130-DR;
228    displ_T = D130_T-D0_T;
229    displFEM = FEM_ROI1-FEM_ROI0;
230
231    mod_0 = sqrt(displDIC_c(:,1).^2+displDIC_c(:,2).^2+displDIC_c(:,3).^2);
232    mod_R = sqrt(displR(:,1).^2+displR(:,2).^2+displR(:,3).^2);
233    mod_T = sqrt(displ_T(:,1).^2+displ_T(:,2).^2+displ_T(:,3).^2);
```

mod_0 — 2051x1 double

	1
1	0.2634
2	0.2611
3	0.2585
4	0.2558
5	0.2534
6	0.2667
7	0.2646
8	0.2627
9	0.2607
10	0.2583

mod_R — 2051x1 double

	1
1	0.2634
2	0.2611
3	0.2585
4	0.2558
5	0.2534
6	0.2667
7	0.2646
8	0.2627
9	0.2607
10	0.2583

mod_T — 2051x1 double

	1
20	0.6632
21	0.6623
22	0.6606
23	0.6592
24	0.6584
25	0.6574
26	0.6563
27	0.6551
28	0.6692
29	0.6689

Figure 3.16: Computation of displacements and modules; comparison between mod_0, mod_R and mod_T.

The next part of the work on MATLAB is focused on the validation of FEM model through DIC data; for this purpose, the first step was to calculate interpolated displacements of DIC to FEM ones, using the *Inverse Distance Weighting* function. *Idw* needs DIC initial nodes coordinates, it uses DIC displacements as input values and FEM initial coordinates for the interpolation, while *IntD_dic* is the vector with interpolated values. The NaN elimination was crucial because when the distance used for interpolation (1 in this case) is smaller than the actual distance between DIC and FEM points, the *idw* function returns a NaN value (Fig 3.17).

```
% Interpolation of DIC to FEM displacements

IntD_dic(:,1) = idw(D0,displDIC(:,1),F0R,2,1);
IntD_dic(:,2) = idw(D0,displDIC(:,2),F0R,2,1);
IntD_dic(:,3) = idw(D0,displDIC(:,3),F0R,2,1);

% NaN elimination

ind1 = find(isnan(IntD_dic(:,1)) == 0);
IntD_dic_new = IntD_dic(ind1,:);
F0c_new = F0R(ind1,:);
displFEM_c_new = displFEM(ind1,:);
nodi_new = nodi(ind1,:);

MatriceDef = [nodi_new F0c_new displFEM_c_new IntD_dic_new];
```

Figure 3.17: Interpolation of displacements and NaN elimination.

To perform the scatter plots of DIC and FEM displacements, it was necessary to calculate the modules to obtain the total displacements; next the removal of outliers is performed. The scatter plots were done for x, y, z directions and for the total displacements, using a reference line calculated as bisector of the first and third quadrants.

To evaluate the goodness of the interpolation results with respect to the predictions of the FEM model, the absolute error, the relative error, the RMSE, the percentage RMSE and the value of R-squared were calculated (Fig. 3.18).

```
% Calculation of absolute value
DiffDispl = abs(MatriceDef_out(:,8:10) - MatriceDef_out(:,5:7));
ErrAv = mean(abs(DiffDispl));
DiffDispl_mod = sqrt(DiffDispl(:,1).^2+DiffDispl(:,2).^2+DiffDispl(:,3).^2);
ErrAv_mod = mean(DiffDispl_mod);

% Calculation of relative error
DiffDisplPerc = (abs(MatriceDef_out(:,8:10) - MatriceDef_out(:,5:7)) ./ abs(MatriceDef_out(:,8:10))) * 100;
Perc_mod = (abs(DIC_mod-FEM_mod) ./ DIC_mod) * 100;
ErrPerc = mean(abs(DiffDisplPerc));
ErrPerc_mod = mean(abs(Perc_mod));

% Calculation of RMSE
rmseX = sqrt(sum((MatriceDef_out(:,5)-MatriceDef_out(:,8)).^2)/size(MatriceDef_out,1));
rmseY = sqrt(sum((MatriceDef_out(:,6)-MatriceDef_out(:,9)).^2)/size(MatriceDef_out,1));
rmseZ = sqrt(sum((MatriceDef_out(:,7)-MatriceDef_out(:,10)).^2)/size(MatriceDef_out,1));
rmse_mod = sqrt(sum((FEM_mod-DIC_mod).^2)/size(MatriceDef_out,1));

% Calculation of RMSE%
rmsePX = (rmseX/abs(max(MatriceDef_out(:,8))))*100;
rmsePY = (rmseY/abs(max(MatriceDef_out(:,9))))*100;
rmsePZ = (rmseZ/abs(max(MatriceDef_out(:,10))))*100;
rmseP_mod = rmse_mod/abs(max(DIC_mod))*100;
```

```
% Calculation R2
[signFit_x, GoF_x, ~] = fit(MatriceDef_out(:,8),MatriceDef_out(:,5), 'poly1');
[signFit_y, GoF_y, ~] = fit(MatriceDef_out(:,9),MatriceDef_out(:,6), 'poly1');
[signFit_z, GoF_z, ~] = fit(MatriceDef_out(:,10),MatriceDef_out(:,7), 'poly1');
[signFit_mod, GoF_mod, ~] = fit( DIC_mod,FEM_mod, 'poly1');
R2Y = GoF_y.rsquare;
R2X = GoF_x.rsquare;
R2Z = GoF_z.rsquare;
R2mod = GoF_mod.rsquare;

X = [R2X; rmseX; rmsePX; ErrAv(1); ErrPerc(1)];
Y = [R2Y; rmseY; rmsePY; ErrAv(2); ErrPerc(2)];
Z = [R2Z; rmseZ; rmsePZ; ErrAv(3); ErrPerc(3)];
TOT = [R2mod; rmse_mod; rmseP_mod; ErrAv_mod; ErrPerc_mod];

MAT2 = table({'R^2';'RMSE';'RMSE%';'errore assouto medio';'errore % medio'}, X, Y, Z, TOT);
```

Figure 3.18: Calculation of errors.

Below are the formulas used for the calculations:

$$Absolute\ error = |U_{FEM} - U_{DIC}| \qquad \textit{Equation 16}$$

$$Relative\ error = \frac{Absolute\ error}{|U_{DIC}|} * 100 \qquad \textit{Equation 17}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(U_{FEM} - U_{DIC})^2}{N}} \qquad \textit{Equation 18}$$

$$RMSE\ \% = \frac{RMSE}{\max(U_{DIC})} \qquad \textit{Equation 19}$$

For the interpolation of strain, it was first necessary to calculate centroids for DIC and FEM and report the coordinates and displacements to them. Then, the *SVD* (Singular Value Decomposition) has been applied to FEM nodes: *SVD* is a technique to decompose a matrix into several component matrices, as can be seen in line 220 (Fig. 3.19), and it allows to build a rotation matrix and a translation vector.

MyCrustOpen is a function that allows to triangulate an un-organized set of 3D points and that is what it does to FEM initial coordinates of nodes; as output the connectivity matrix is obtained as it contains triangulated points id.

TriStrain and *triToNodes* are two functions that calculate the strains ε_1 and ε_3 for FEM data set. In this case the interpolation of DIC to FEM strains was done using *idw* function with DIC and FEM initial nodes coordinates and *S130*'s first and second columns which are respectively ε_1 and ε_2 strains for the DIC final nodes coordinates.

```
% SVD to FEM points
E = (F0c)'*(F1c);
[U,S,V] = svd(E);
R1 = V*U';  % rotation matrix
T1 = mean(F1c)' - R1*mean(F0c)'; % translation vector
F1_rot_trasl = (R1*(F0c)')';
rigidFEM = F1_rot_trasl - (F0c);
defFEM = displFEM - rigidFEM;
UtotFEM = rigidFEM + defFEM;

trisFEM = MyCrustOpen(F0c);

strainFEM_def = triStrain(trisFEM,F0c,defFEM); % eps3(def min) - eps1 (def max)
strainFEM_n_def = triToNodes(strainFEM_def,trisFEM); % Strains calculated in nodes
strainFEM_n_def = [strainFEM_n_def(:,2),strainFEM_n_def(:,1)];

% Interpolation of DIC to FEM strains
IntS_dic(:,1) = idw(D0c,S130(:,1),F0c,2,1);
IntS_dic(:,2) = idw(D0c,S130(:,2),F0c,2,1);

ind1 = find(isnan(IntS_dic(:,1)) == 0);
IntS_dic_new = IntS_dic(ind1,:);
strainFEM_n_new = strainFEM_n_def(ind1,:);

strainDef = [strainFEM_n_new, IntS_dic_new];
```

Figure 3.19: SVD application, strain calculation and interpolation of DIC to FEM strain.

To quantify the accuracy of the interpolation of strains, RMSE and percentage RMSE are calculated (Fig. 3.20).

```
% Calculation of RMSE for strains

rmseEps2 = sqrt(sum((strainDef(:,2)-strainDef(:,4)).^2)/size(strainDef,1));
rmseEps1 = sqrt(sum((strainDef(:,1)-strainDef(:,3)).^2)/size(strainDef,1));

% Calculation of RMSE% for strains
rmsePEps2 = (rmseEps2/abs(max(strainDef(:,4))))*100;
rmsePEps1 = (rmseEps1/abs(max(strainDef(:,3))))*100;

Eps2 = [rmseEps2; rmsePEps2];
Eps1 = [rmseEps1; rmsePEps1];

MAT_deform = table({'RMSE';'RMSE%'}, Eps1, Eps2);
```

Figure 3.20: RMSE and RMSE % calculation for interpolated strains.

The last part of the MATLAB post-processing regards the connectivity matrix: in this case, since the femur mesh chosen in Abaqus was a volumetric tetrahedral mesh, each node is in correlation with other three. All the considerations and calculations have been done considering just the femur skin

both for FEM and for DIC: in this way, it was necessary to modify the connectivity matrix extracted from FEM analysis. At this purpose, *list_faces* and *extract_unique_surfaces* are two functions which allow to obtain a triangular connectivity matrix that matches the DIC ROI; a new connectivity matrix is built by correspondence between FEM connectivity matrix and DIC id nodes (Fig. 3.21).

```
roi = MatriceDef;
mat_def2=[[1:length(roi)]' roi];

% Surface faces extraction
F_all = list_faces(connectivity);
F_surf = extract_unique_triangles(F_all);
connectivity = F_surf;


connectivitynew = zeros(size(connectivity,1),size(connectivity,2));
L = ismember(connectivity,mat_def2(:,2));

for i=1:size(connectivity,1)
    if sum(L(i,:)) == 3
        for j=1:size(connectivity,2)
            aa = find(mat_def2(:,2) == connectivity(i,j));
            connectivitynew(i,j) = mat_def2(aa,1);
        end
    end
end

connectivitynew( ~any(connectivitynew,2), : ) = [];
```

Figure 3.21: Definition of a FEM surface connectivity matrix and construction of a new one by correspondence with DIC id nodes.

3.2 Results

3.2.1 FEM analysis in Abaqus

After FEM analysis, results were visualized for the deformed femur configuration; in particular, the U displacement (magnitude), the maximum and minimum principal strains and the direction vectors of force and boundary condition were displayed. As it can be seen in Fig. 3.22, U displacement in mm shows a descendent gradient, from the head region where the load was applied and there is a maximum, until the condylar region that has been bound by an encastre where the displacement is 0 mm.

The gradient reflects the direction of force application and respects the imposed boundary condition. The magnitude of maximum principal strain and minimum principal strain are represented (Fig. 3.23). To complete the representation of the results, the graphs of the principal strains with the force and constraint vectors imposed are also reported (Fig. 3.24).

The first graph of Fig. 3.23 represents the numerically largest principal strain, as can be seen in the red region along the diaphysis. The fact that the maximum value is positive means that in the region the femur undergoes a traction, while a negative value of strain represents a compression.

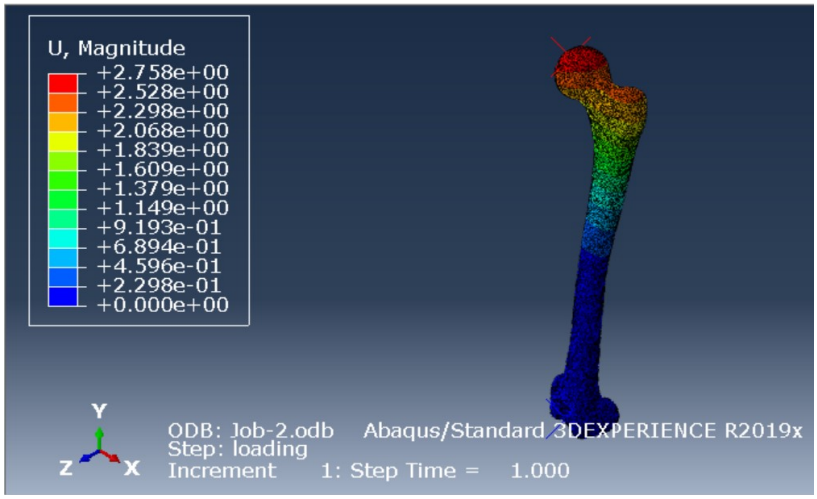
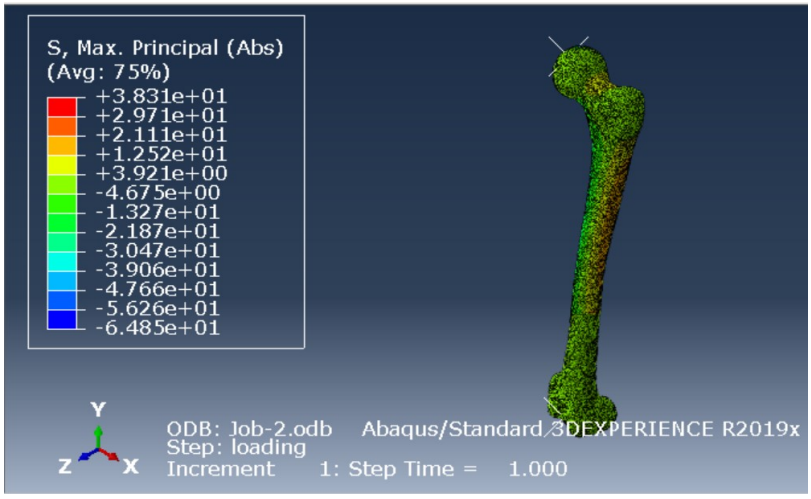


Figure 3.22: U displacement (magnitude) for the deformed configuration.



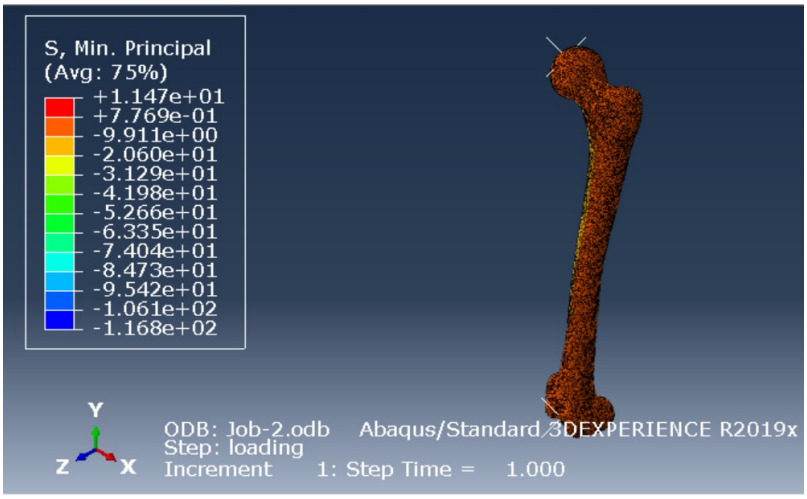


Figure 3.23: Max. principal strain and Min. principal strain for the deformed configuration.

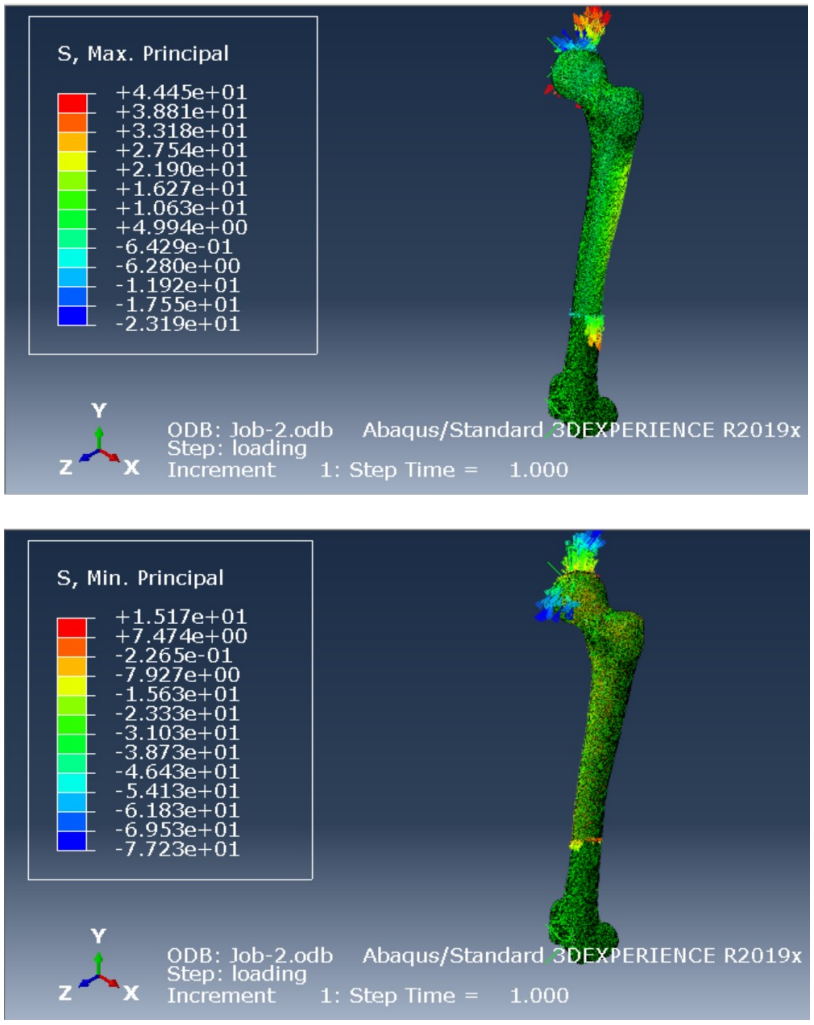
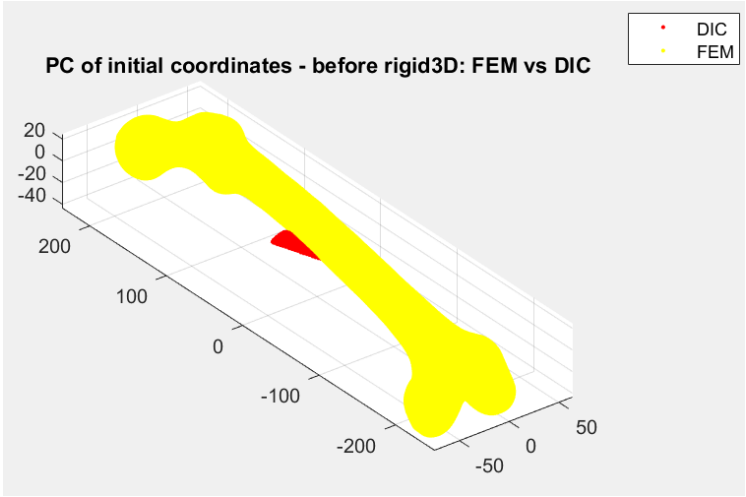


Figure 3.24: Principal strains with force and constraint vectors.

3.2.2 MATLAB implementation for FEM validation through DIC

The first step addressed for the FEM validation was the alignment of the DIC point cloud on FEM point cloud considering just the surface of the femur. With reference to the script described in Fig. 3.14, Fig. 3.25 represents the FEM femur in yellow extracted from Abaqus and the ROI extracted from DIC in red. The first figure refers to the point clouds before the rigid transformation of rotation and translation. The second picture represents the point clouds after the transformation and the ROI in which it is possible to notice the correspondence of the ROI on the femur.

Fig. 3.26 displays the ROI defined for FEM dataset using minimum and maximum of DIC data. In Fig. 3.27 the result of ICP algorithm is represented: it is possible to notice the goodness of the alignment performance. It is important to consider that the point clouds should be as aligned as possible to minimize the distance between FEM reference point and DIC moving point. Otherwise, the algorithm does not recognize the previously applied rigid transformation and is unable to perform alignment.



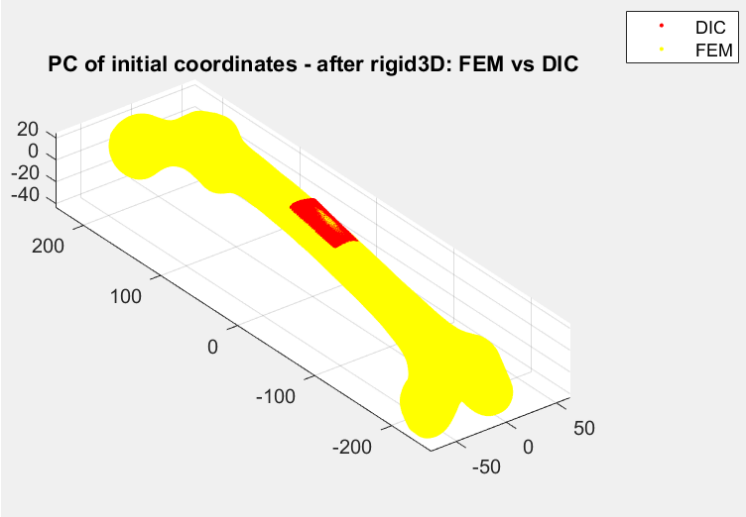


Figure 3.25: Point clouds of FEM and DIC, before and after rigid transformation (initial coordinates).

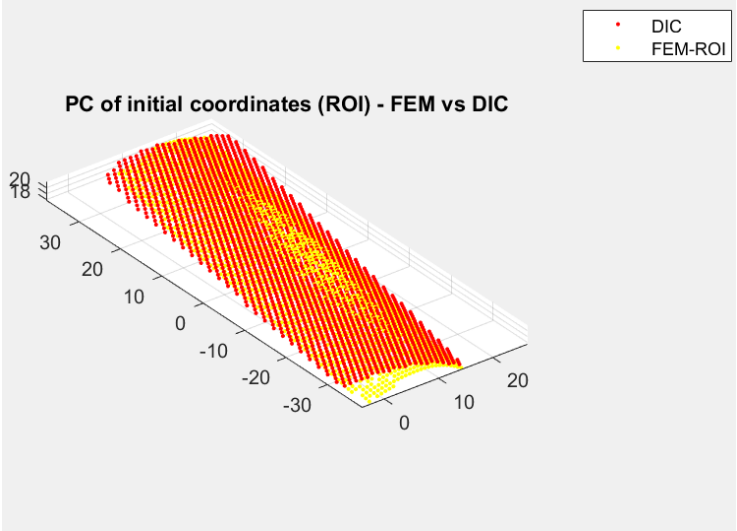


Figure 3.26: ROI defined for FEM data.

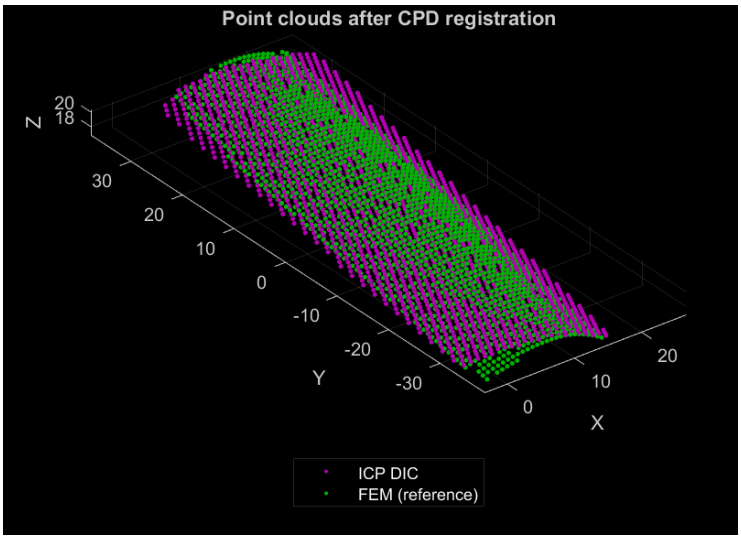


Figure 3.27: ICP algorithm result of alignment between FEM and DIC point clouds.

The same process has been done for final DIC and FEM coordinates, thus it was possible to calculate displacements and modules of displacements.

The displacements of DIC coordinates before the transformation, the ones after the ICP algorithm and FEM displacements have been depicted together for comparison. In Fig. 3.28 is possible to notice that DIC displacements before *pcregistericp* function are coarser than the other two (pink dots). This is the main reason why modules calculated before the algorithm are greater than the ones after ICP. The goodness of the alignment is represented by the graphic similarity between FEM and DIC displacements after the transformation (respectively blue and red dots); they are in the same reference system and easily comparable. The blank spaces that can be seen in DIC ROI are eliminated as outliers after the interpolation of DIC displacements and before performing the scatter plot.

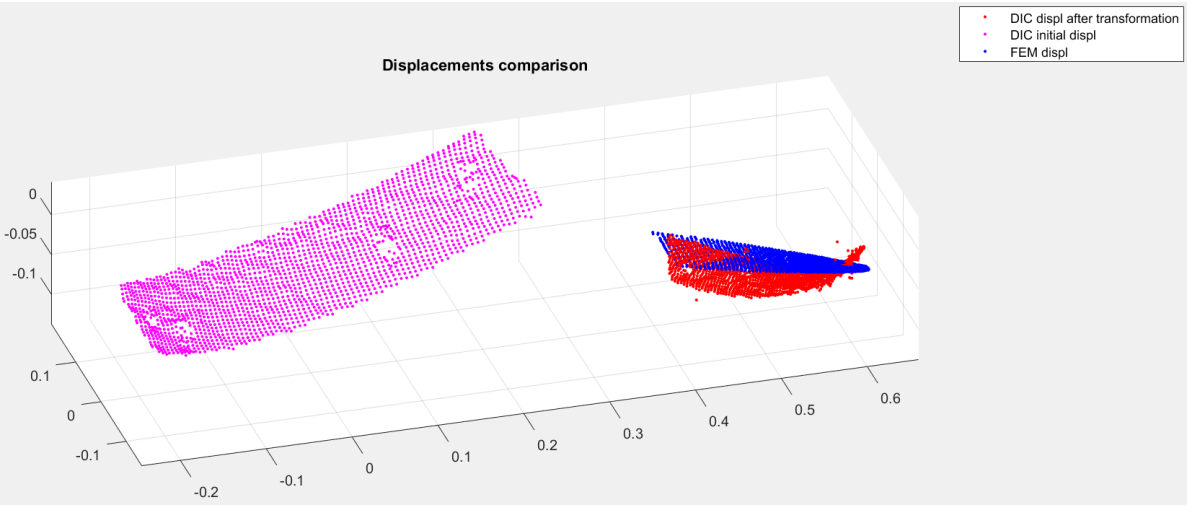


Figure 3.28: Comparison between DIC displacements before and after rigid3D, and FEM displacements.

In Fig. 3.29 the result of interpolation of DIC displacements on FEM ones is displayed together with the other two mentioned before. The choice of the Euclidean distance was important for the correct interpolation point-to-point and, as it can be seen, DIC displacements match with interpolated ones.

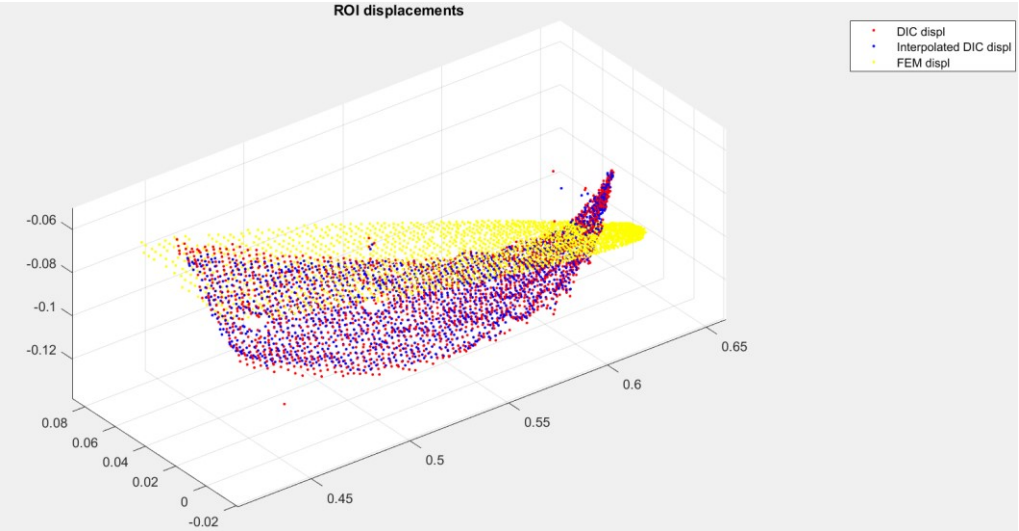


Figure 3.29: Result of displacements interpolation.

In addition, in Fig. 3.30 a verification of the interpolation has been done and it is depicted in terms of coordinates.

The next step was the scatter plot of displacements, comparing DIC and FEM for x, y, z directions together with modules; Fig. 3.31 represents the results. The red line has been considered as the reference behavior of DIC displacements and it has been calculated referring to minimum and maximum of FEM data. As it can be seen, U_x and U_{tot} have a very similar slope to the reference line, while U_y and U_z are coarser but more distributed on the red line. Even if the modules are calculated considering all the three directions, the fact that they have a close behavior to U_x it means that the x direction has more influence than the other two.

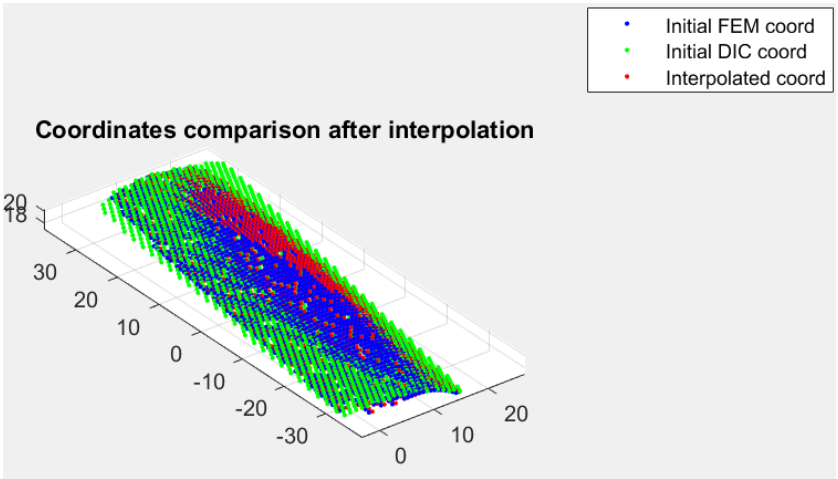


Figure 3.30: Coordinates comparison after interpolation for verification.

The dispersion throughout the line is quantified with R-squared and errors calculation: as it can be seen in Fig. 3.31 for U_x and U_{tot}, FEM underestimates DIC. Tab. 3.1 sums up all the calculated errors to evaluate if computational model can predict the experimental behavior.

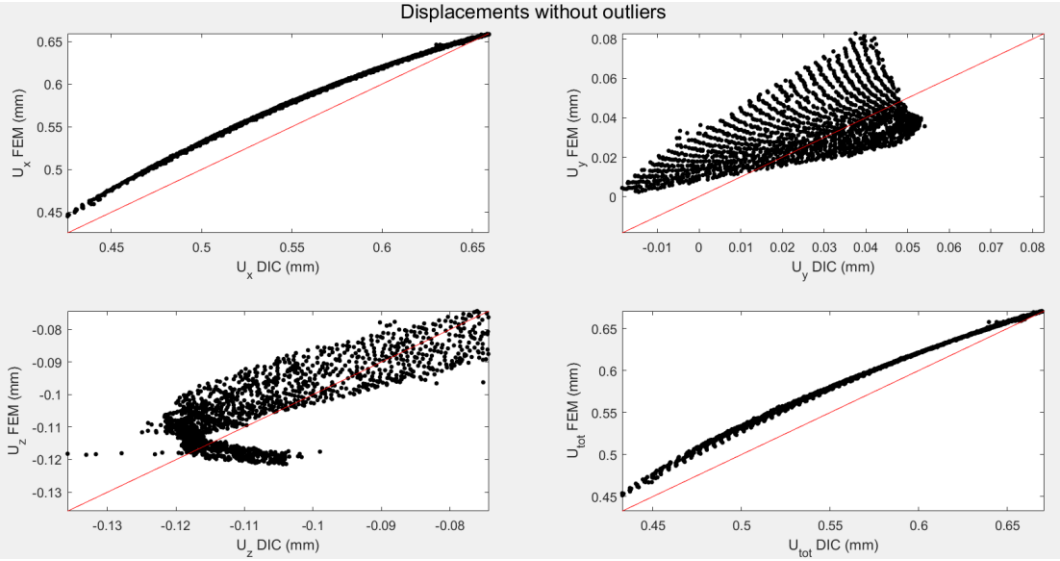


Figure 3.31: Scatter plot of displacements for DIC and FEM comparison.

A high value of R-squared indicates that a great percentage of the variation in FEM values has a correspondence in DIC values. Since y direction has the lower R-squared value, only x and z directions are representative of results; this can be explained by the fact that DIC in one direction is hyper inaccurate.

	X	Y	Z	Tot
R-squared	0.9907	0.3641	0.6186	0.9922
RMSE	0.0236	0.0180	0.0084	0.0232
RMSE %	3.5781	33.2370	11.3164	3.4617
Mean Absolute Error	0.0214	0.0150	0.0071	0.0288
Mean % Error	4.0028	365.8123	6.6234	3.8531

Table 3.1: Errors calculated on displacements for x, y, z directions and modules.

The strains have been interpolated as well as displacements, for which also RMSE % has been calculated for ε_1 and ε_2 (Tab. 3.2). RMSE % value indicates the accuracy of FEM model with respect

to DIC results; having high values as for strains, shows poor accuracy in the prediction of deformations.

	Eps1	Eps2
RMSE %	30.1807	198.6733

Table 3.2: RMSE % for principal strains.

The last step of MATLAB implementation was to consider the connectivity matrix extracted from FEM analysis and convert it into a matrix compatible with the DIC ROI. This process allowed to have a comparison of displacements between DIC and FEM, to obtain a contour plot of their difference also in percentage (Fig. 3.32 and Fig. 3.33). It is noticeable that the connectivity matrix has some blank spots: the reason may be because this new matrix has been built for correspondence of nodes and elements with DIC ones, using an optimization process. One of the further solutions for this issue should be to perform another FEM analysis using Abaqus by changing the size of the mesh and femur properties like Young modulus and Poisson's ratio.

For what concerns strains, it has not been possible to perform a contour plot due to the mismatch between DIC and FEM data in the connectivity matrix; a future solution could be a new analysis for both. Regarding DIC, by observing the contour plot it must be taken into account that it has been performed just one test on the femur, without trying other configurations that better mimic its physiology in vivo.

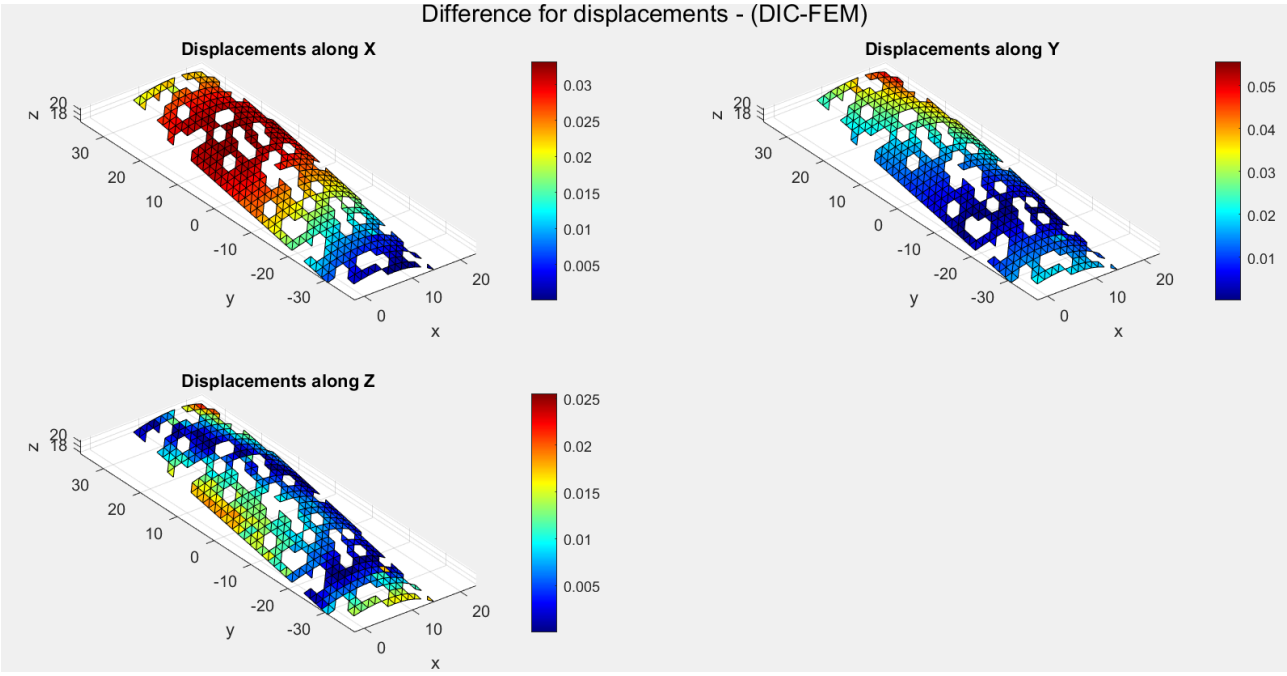


Figure 3.32: Contour plot of the differences for displacements DIC-FEM.

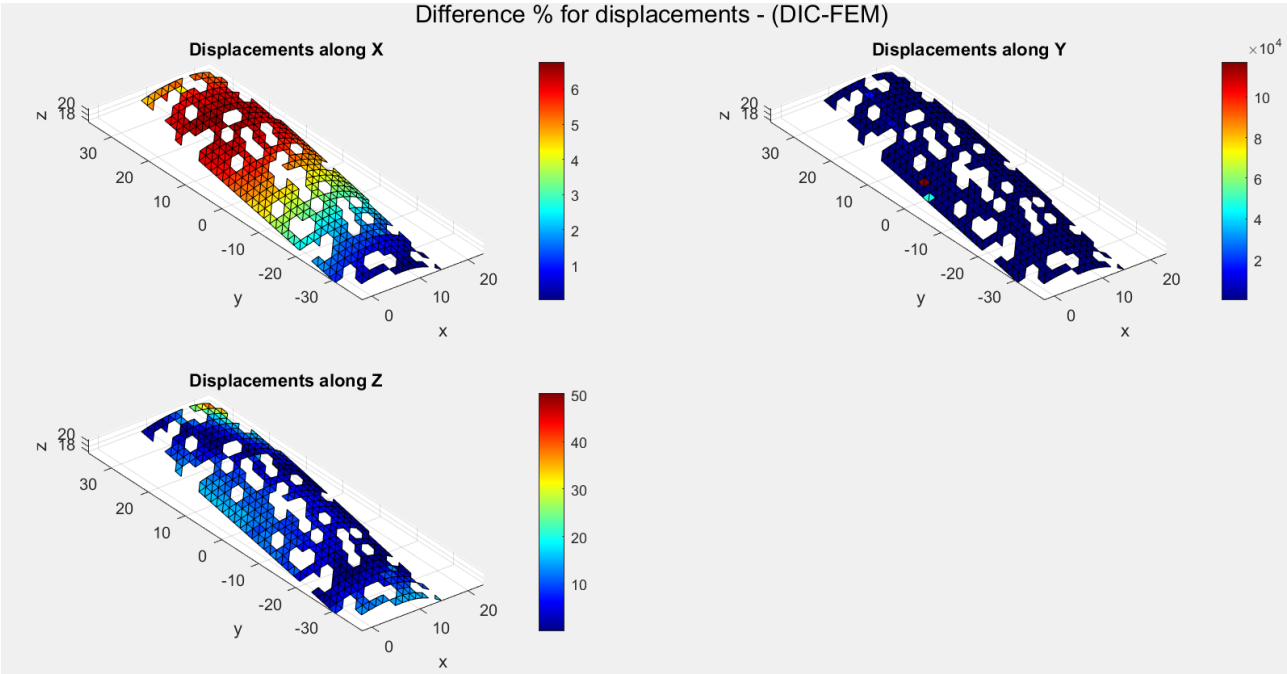


Figure 3.33: Contour plot of the difference in percentage for displacements DIC-FEM.

4 Design of a new experimental set-up

The second part of the work was focused on the design and realization of a new experimental set-up for the femur. The purpose was to create a modular structure, made up of different components that could be assembled and then fixed on the ElectroPuls® E3000 for the compression test. This idea came from the need to reproduce more truthfully the physiological positioning of the femur in the hip joint.

In this section has been treated only the CAD design of the structure, explaining the choice of materials for the realization and the reasons of the conformations chosen for each element.

4.1 Materials and Methods

The whole section is based on the use of Ansys SpaceClaim, starting from the import of the same .step file used for section 3.1.4 (Fig. 3.10) and turning it into a solid femur. The design process started from the definition and realization of a wedge as the basis of the structure, a solid L-shaped and a vice, all made of steel.

The wedge was designed with a 150 mm square side base and extruded vertically to create a parallelepiped. Since it was necessary that the femur was positioned respecting the physiological orientation with reference to Fig. 3.3, the parallelepiped was cut diagonally to obtain an upper base inclined 9° to the horizontal plane (Fig. 4.1). In addition, four holes were created to allow the insertion of the wedge on the test machine guide through threaded screws.

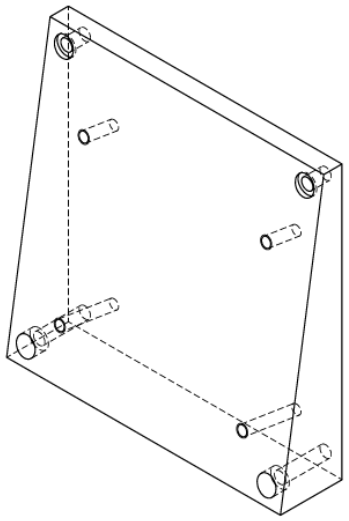


Figure 4.1: Design of the wedge realized with SOLIDWORKS®.

The second element was designed as an L-shaped solid (Fig. 4.2) to be fixed to the basal wedge below. In particular, on the horizontal and vertical surface of the L were realized two rectangular recesses for the parts that interface directly with the femur. The solid L has been equipped with eight holes, four on the base to be fixed to the inclined wedge and then to the guide of the test machine, and four on the vertical face to allow the fixation of the femur through the vice.

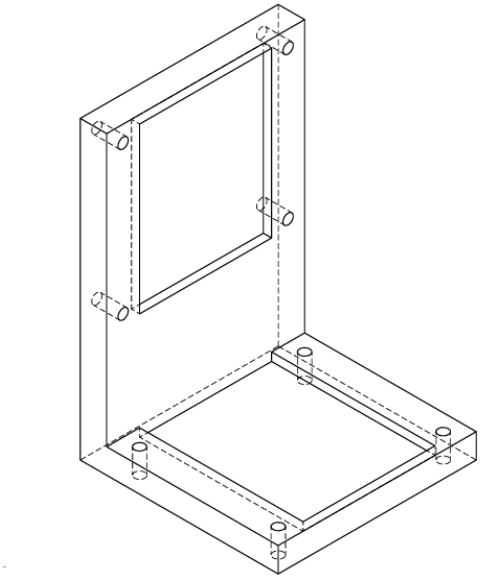


Figure 4.2: Design of the L-shaped solid realized with SOLIDWORKS®.

The vice (Fig. 4.3) was realized as a parallelepiped with a rectangular recess as well as the L-shaped solid. It was equipped with four holes to allow the femur to be held in place and attached to the L-shaped solid by compression.

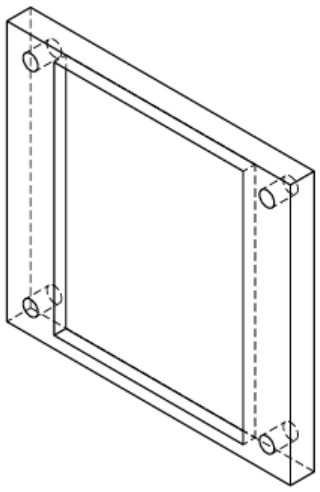


Figure 4.3: Design of the vice realized with SOLIDWORKS®.

The structure parts that interface directly with the femur have been designed to be made using the 3D printer with a plastic material, such as ABS or PLA. The femur was bound in the condylar and diaphyseal distal region, recalling the rigid constraint imposed in the previous model of the potted femur. To create the interface element with the femoral condyles, a cast of the geometry of the condyles was created and then extruded as a solid. The dimensions of the parallelepiped were chosen considering the recess created at the base of the steel L-shaped solid.

The diaphyseal region was similarly extruded in the direction of the L-shaped solid and of the vice, resulting in two parallelepipeds compatible with the recesses.

Since the femur had to be easily housed in the structure, overcoming the conformation of the condyles, the vice and the associated cast, were made only for a limited portion of diaphysis. The design thus realized allowed the insertion of the femur resting on the L-shaped solid and the insertion of the vice in the opposite direction for an additional fixing.

Since the inclined wedge at the base is removable, the structure can also be used for tests without inclination. Similarly, the creation of ABS parts with 3D printing, independent of the steel structure, allows the use of the set-up also for other bones.

4.2 Results

The output of the CAD design of the set-up described above is reported as a result. In Fig. 4.4, the three steel elements are represented from different angles.

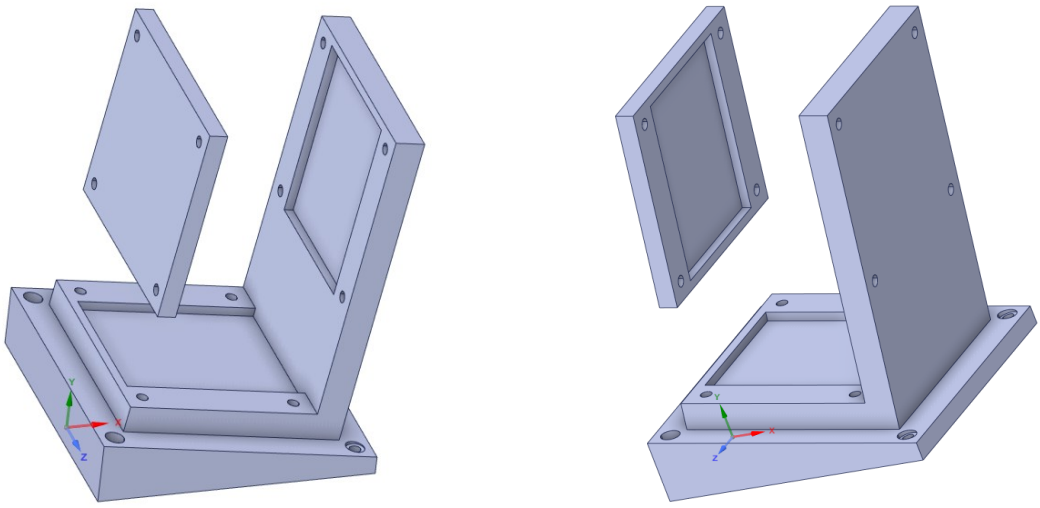


Figure 4.4: Three steel elements realized with Ansys SpaceClaim.

The ABS elements are displayed together with the femur in Fig. 4.5.

As can be seen from Fig. 4.5, the two ABS elements at the diaphysis were modelled without perfect contact. This choice was made considering a margin of acceptable error between the femur and the

elements. In addition, the solid femur created with the software was obtained by smoothing the geometry of the .step file, to prevent future errors in mesh definition.

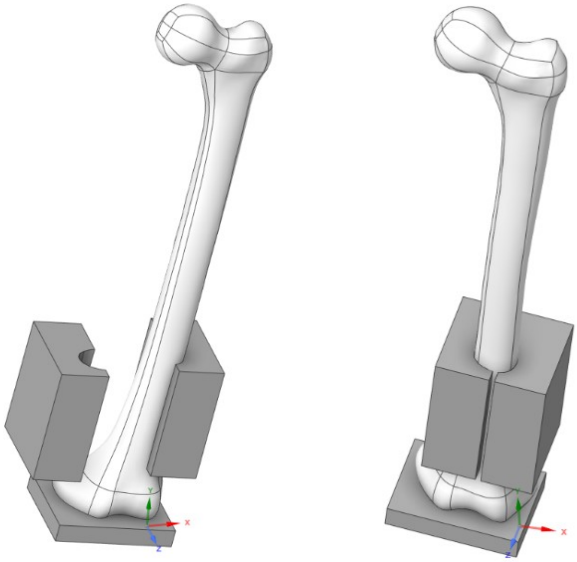


Figure 4.5: ABS elements and femur realized with Ansys SpaceClaim.

In Fig. 4.6 the whole model is represented.

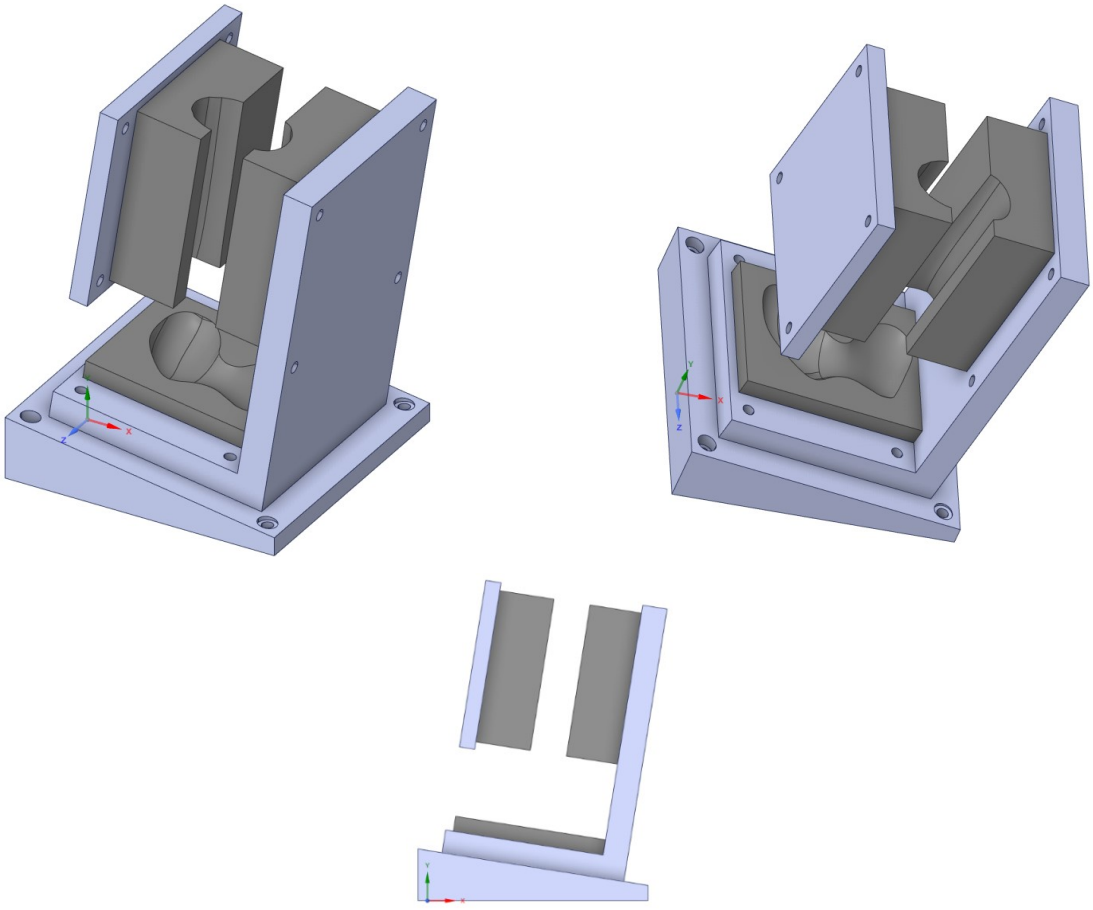


Figure 4.6: Representation of the complete structure.

5 Conclusions and future directions

The purpose of the presented study can be divided into two sub-purposes. First, starting from the geometry of a Sawbone femur, the aim was to evaluate the possibility of constructing a patient-specific FE model. Second, a more accurate experimental set-up was developed and modeled, to be physically validated with a testing machine.

The construction of the FE model involved the use of a .step file of a femur identical to that tested with DIC to maintain a consistency between experimental and numerical results. In this regard, the mechanical characteristics of the Sawbone, modeled as an anisotropic homogeneous solid material, were considered as the correct approximation that would allow data agreement between the DIC and FEM sets. The results obtained from the first validation performed were satisfactory, particularly for displacements along the x- and z-directions. In these two cases the value of R-square was good, while for the y-direction some improvements to the interpolation algorithm might be needed. The connectivity matrix constructed from the two data sets represented the ROI with good accuracy despite missing some elements, allowing a positive evaluation of the displacements.

The interpolation of the strains was more complex and, in order to decrease the error rate, new DIC tests could be conducted along with a modification of the parameters of the interpolation algorithm. To increase the accuracy of the predictions, varying the mesh refinement could improve the results in terms of strains. However, placing the restraint on the condyles and distal diaphysis as an encastre was a valid choice and agreed with the DIC results.

Regarding the new experimental set-up, the primary future direction is definitely the fabrication of the elements in the two different materials. This implies the possibility of testing the compressive strength of the created structure and its modularity, coupled with the ability to firmly accommodate a femur and to be able to be properly attached to the testing machine. Since in the case of the cemented femur, a FE model had been created that constrained with an encastre all the nodes belonging to the distal region of the diaphysis and condyles, the same considerations should be made for the new set-up. Specifically, on Abaqus , the idea would be to make a tetrahedral mesh for the femur and hexahedral mesh for each element of the structure, considering the presence of holes and possible screws, and then proceed with material assignment. The application of constraints should consider both the contacts of the structure with the test machine guide and the mutual contacts between femur and elements of the structure and between element and element. In this case an interlocking condition would no longer be representative of experimental reality, but possible roto-translations between surfaces should be considered. By aligning the point clouds and interpolating the DIC displacements and strains to the FEM ones, considerations could be made about the reliability of the new set-up compared to the previous one.

Post-processing on MATLAB proved to be the most complex part of the entire work, requiring more ongoing adaptations and several changes based on the data acquired. The definition of a semi-automated process for the management of DIC and FEM data in two different reference systems and the validation of the latter, is a positive result that can be used for new tests.

At the same time, the design of the new experimental set-up took into account several aspects that were improvable in the potted structure. The fabrication of the same and its test with the femur for which it was designed will be completed work already done.

In conclusion, the FE model of sawbone femur realized in this work represents a good starting point for the prosecution of the study at an experimental and numerical level, improving already existing methods.

6 Appendix

6.1 Point Clouds alignment

```
load("job2.mat");
load("displx.mat");
load("fempelle.mat");

F0 = Job2(:,2:4); % initial coordinates FEM
F1 = displx(:,2:4); % final coordinates FEM
nodi = displx(:,1); % nodes FEM

F0 = round(F0,5);
F1 = round(F1,5);
fempelle = round(fempelle,5);

ind = find(ismember(F0,fempelle,'rows'));

F0 = F0(ind,:);
F1 = F1(ind,:);
nodi = nodi(ind,:);

P0 = readmatrix(strcat(folder,'img-0001_0.csv')); % initial set DIC
P130 = readmatrix(strcat(folder,'img-0130_0.csv')); % final set DIC
D0 = P0(:,1:3); % initial coordinates DIC
D130 = P130(:,1:3)+P130(:,4:6); % final coordinates DIC
S0 = P0(:,10:11);
S130 = P130(:,10:11);

% Outliers removal
ind1 = find(D0(:,1) == 109.5460);
ind2 = find(D130(:,1) == 109.5460);

D0(ind1, :) = [];
D130(ind2, :) = [];
S0(ind1,:) = [];
S130(ind2,:) = [];

displ0 = D130-D0;
i = any(displ0==0,2);
displ0(i,:) = [];

% Centroids
pfem = mean(F0); % mean of the positions of the datasets
qfem = mean(F1);
F0c = F0-pfem; % initial position centred into the origin
F1c = F1-qfem; % final position centred into the origin
displFEM_c = F1c - F0c; % FEM displacements

pdic = mean(D0);
qdic = mean(D130);
D0c = D0-pdic;
D130c = D130-qdic;
displDIC_c = D130c - D0c; % DIC displacements

% Initial coordinates - DIC and FEM (D0, F0)
pc_dic0 = pointCloud(D0c);
pc_fem0 = pointCloud(F0c);

figure()
plot3(pc_dic0.Location(:,1),pc_dic0.Location(:,2),pc_dic0.Location(:,3),'r.','DisplayName','DIC')
axis equal
hold on
plot3(pc_fem0.Location(:,1),pc_fem0.Location(:,2),pc_fem0.Location(:,3),'y.','DisplayName','FEM')
```

```
legend('show')
grid on
title('PC of initial coordinates - before rigid3D: FEM vs DIC')

% Rigid rotation and translation
alfa = deg2rad(153);
Az = [cos(alfa) sin(alfa) 0 ;
      -sin(alfa) cos(alfa) 0;
      0 0 1];

T = [11,0,20];

rot = rigid3d(Az,T);
pc_dic_rot = pctransform(pc_dic0,rot);

DR = pc_dic_rot.Location;

figure()
plot3(pc_dic_rot.Location(:,1),pc_dic_rot.Location(:,2),pc_dic_rot.Location(:,3),'r.','DisplayNam
e','DIC')
axis equal
hold on
plot3(pc_fem0.Location(:,1),pc_fem0.Location(:,2),pc_fem0.Location(:,3),'y.','DisplayName','FEM')
legend('show')
grid on
title('PC of initial coordinates - after rigid3D: FEM vs DIC')

% Definition of FEM ROI
minx = min(pc_dic_rot.Location(:,1));
miny = min(pc_dic_rot.Location(:,2));
minz = min(pc_dic_rot.Location(:,3));

maxx = max(pc_dic_rot.Location(:,1));
maxy = max(pc_dic_rot.Location(:,2));
maxz = max(pc_dic_rot.Location(:,3));

roi = [minx maxx miny maxy minz maxz];
index = findPointsInROI(pc_fem0,roi);
pc_femroi = select(pc_fem0,index);
nodi_indx = nodi(index,:);

figure()
plot3(pc_dic_rot.Location(:,1),pc_dic_rot.Location(:,2),pc_dic_rot.Location(:,3),'r.','DisplayNam
e','DIC')
axis equal
hold on
plot3(pc_femroi.Location(:,1),pc_femroi.Location(:,2),pc_femroi.Location(:,3),'y.','DisplayName',
'FEM-ROI')
legend('show')
grid on
title('PC of initial coordinates (ROI) - FEM vs DIC')

% ICP Algorithm
tform = pcregistericp(pc_dic_rot,pc_femroi,'MaxIterations',700,'Tolerance',[1e-12 1e-
7],'Verbose',true);
mskReg = pctransform(pc_dic_rot,tform);

figure()
pcshowpair(mskReg,pc_femroi,'MarkerSize',50)
axis equal
xlabel('X'),ylabel('Y'),zlabel('Z')
title('Point clouds after CPD registration')
legend({'ICP DIC','FEM (reference)'},'TextColor','w')
legend('Location','southoutside')

D0_T=mskReg.Location;
```

```
figure()
plot3(pc_femroi.Location(:,1),pc_femroi.Location(:,2),pc_femroi.Location(:,3),'y.','DisplayName',
'FEM')
legend('show')
grid on
axis equal
hold on
plot3(D0_T(:,1),D0_T(:,2),D0_T(:,3),'r.','DisplayName','DIC')
axis equal
legend('show')

% Final coordinates - DIC and FEM (D130, F1)
pc_dic1 = pointCloud(D130c);
pc_fem1 = pointCloud(F1c);

figure()
plot3(pc_dic1.Location(:,1),pc_dic1.Location(:,2),pc_dic1.Location(:,3),'r.','DisplayName','DIC')
axis equal
hold on
plot3(pc_fem1.Location(:,1),pc_fem1.Location(:,2),pc_fem1.Location(:,3),'y.','DisplayName','FEM')
legend('show')
grid on
title('PC of final coordinates - before rigid3D: FEM vs DIC')

% Rigid rotation and translation
alfa = deg2rad(153);
Az = [cos(alfa) sin(alfa) 0 ;
      -sin(alfa) cos(alfa) 0;
      0 0 1];

T = [11,0,20];

rot = rigid3d(Az,T);
pc_dic_rot = pctransform(pc_dic1,rot);

figure()
plot3(pc_dic_rot.Location(:,1),pc_dic_rot.Location(:,2),pc_dic_rot.Location(:,3),'r.','DisplayNam
e','DIC')
axis equal
hold on
plot3(pc_fem1.Location(:,1),pc_fem1.Location(:,2),pc_fem1.Location(:,3),'y.','DisplayName','FEM')
legend('show')
grid on
title('PC of final coordinates - after rigid3D: FEM vs DIC')

DR130 = pc_dic_rot.Location;

% Definition of FEM ROI
pc_femroi2 = select(pc_fem1,index);

figure()
plot3(pc_dic_rot.Location(:,1),pc_dic_rot.Location(:,2),pc_dic_rot.Location(:,3),'r.','DisplayNam
e','DIC')
axis equal
hold on
plot3(pc_femroi2.Location(:,1),pc_femroi2.Location(:,2),pc_femroi2.Location(:,3),'y.','DisplayNam
e','FEM-ROI')
legend('show')
grid on
title('PC of final coordinates (ROI) - FEM vs DIC')

% ICP Algorithm
tform = pcregistericp(pc_dic_rot,pc_femroi2,'MaxIterations',700,'Tolerance',[1e-12 1e-
7],'Verbose',true);
mskReg = pctransform(pc_dic_rot,tform);

figure()
```

```
pcshowpair(mskReg,pc_femroi2,'MarkerSize',50)
axis equal
xlabel('X'),ylabel('Y'),zlabel('Z')
title('Point clouds after CPD registration')
legend({'ICP DIC','FEM (reference)'},'TextColor','w')
legend('Location','southoutside')

D130_T=mskReg.Location;

figure()
plot3(pc_femroi2.Location(:,1),pc_femroi2.Location(:,2),pc_femroi2.Location(:,3),'y.','DisplayNam
e','FEM')
legend('show')
grid on
axis equal
hold on
plot3(D130_T(:,1),D130_T(:,2),D130_T(:,3),'r.','DisplayName','DIC')
axis equal
legend('show')

FEM_ROI0 = pc_femroi.Location;
FEM_ROI1 = pc_femroi2.Location;

% Displacements and modules
displR = DR130-DR;
displ_T = D130_T-D0_T;
displFEM = FEM_ROI1-FEM_ROI0;

mod_0 = sqrt(displDIC_c(:,1).^2+displDIC_c(:,2).^2+displDIC_c(:,3).^2);
mod_R = sqrt(displR(:,1).^2+displR(:,2).^2+displR(:,3).^2);
mod_T = sqrt(displ_T(:,1).^2+displ_T(:,2).^2+displ_T(:,3).^2);

figure()
plot3(displ_T(:,1),displ_T(:,2),displ_T(:,3),'.r','DisplayName','DIC displ after transformation')
hold on
plot3(displDIC_c(:,1),displDIC_c(:,2),displDIC_c(:,3),'.m','DisplayName','DIC initial displ')
hold on
plot3(displFEM(:,1),displFEM(:,2),displFEM(:,3),'.b','DisplayName','FEM displ')
grid on
axis equal
legend show
title('Displacements comparison')
```

6.2 FEM validation through DIC

```
% Displacements
displDIC = D130-D0; % DIC
displFEM = F1R-F0R; % FEM

% Interpolation of DIC to FEM displacements
IntD_dic(:,1) = idw(D0,displDIC(:,1),F0R,2,1);
IntD_dic(:,2) = idw(D0,displDIC(:,2),F0R,2,1);
IntD_dic(:,3) = idw(D0,displDIC(:,3),F0R,2,1);

% NaN elimination
ind1 = find(isnan(IntD_dic(:,1)) == 0);
IntD_dic_new = IntD_dic(ind1,:);
F0c_new = F0R(ind1,:);
displFEM_c_new = displFEM(ind1,:);
nodi_new = nodi(ind1,:);

MatriceDef = [nodi_new F0c_new displFEM_c_new IntD_dic_new];

figure()
plot3(displDIC(:,1),displDIC(:,2),displDIC(:,3),'r.','DisplayName','DIC displ')
hold on
```

```
plot3(IntD_dic_new(:,1),IntD_dic_new(:,2),IntD_dic_new(:,3),'b.','DisplayName','Interpolated DIC
displ')
hold on
plot3(displFEM_c_new(:,1),displFEM_c_new(:,2),displFEM_c_new(:,3),'y.','DisplayName','FEM displ')
axis equal
grid on
legend show
title('ROI displacements')

% Displacements Scatter Plot

% Modules calculation
FEM_mod = sqrt(MatriceDef(:,5).^2+MatriceDef(:,6).^2+MatriceDef(:,7).^2);
DIC_mod = sqrt(MatriceDef(:,8).^2+MatriceDef(:,9).^2+MatriceDef(:,10).^2);
MatriceDef = [MatriceDef, FEM_mod, DIC_mod];

% Removal of outliers
[MatriceDef_out, TF] = rmoutliers(MatriceDef(:,5:end));
FEM_mod = FEM_mod(~TF);
DIC_mod = DIC_mod(~TF);

% x direction
figure()
subplot(2,2,1)
plot(MatriceDef_out(:,4), MatriceDef_out(:,1), 'k.','MarkerSize',10)
x1 = [MatriceDef_out(:,4);MatriceDef_out(:,1)];
y1 = [min(x1), max(x1)];
xlim(y1)
ylim(y1)
hold on
plot(y1,y1,'r-')
xlabel('U_{x} DIC (mm)'), ylabel('U_{x} FEM (mm)')
% y direction
subplot(2,2,2)
plot(MatriceDef_out(:,5), MatriceDef_out(:,2), 'k.','MarkerSize',10)
x2 = [MatriceDef_out(:,5);MatriceDef_out(:,2)];
y2 = [min(x2), max(x2)];
xlim(y2)
ylim(y2)
hold on
plot(y2,y2,'r-')
xlabel('U_{y} DIC (mm)'), ylabel('U_{y} FEM (mm)')
% z direction
subplot(2,2,3)
plot(MatriceDef_out(:,6), MatriceDef_out(:,3), 'k.','MarkerSize',10)
x3 = [MatriceDef_out(:,6);MatriceDef_out(:,3)];
y3 = [min(x3), max(x3)];
xlim(y3)
ylim(y3)
hold on
plot(y3,y3,'r-')
xlabel('U_{z} DIC (mm)'), ylabel('U_{z} FEM (mm)')
% total
subplot(2,2,4)
plot(MatriceDef_out(:,8),MatriceDef_out(:,7),'k.','MarkerSize',10)
x4 = [DIC_mod;FEM_mod];
y4 = [min(x4), max(x4)];
xlim(y4)
ylim(y4)
hold on
plot(y4,y4,'r-')
xlabel('U_{tot} DIC (mm)'), ylabel('U_{tot} FEM (mm)')
sgtitle("Displacements without outliers")

% Displacements errors
MatriceDef_out = [MatriceDef(~TF,1:4), MatriceDef_out];
```

```
% Calculation of absolute value
DiffDispl = abs(MatriceDef_out(:,8:10) - MatriceDef_out(:,5:7));
ErrAv = mean(abs(DiffDispl));
DiffDispl_mod = sqrt(DiffDispl(:,1).^2+DiffDispl(:,2).^2+DiffDispl(:,3).^2);
ErrAv_mod = mean(DiffDispl_mod);

% Calculation of relative error
DiffDisplPerc = (abs(MatriceDef_out(:,8:10) - MatriceDef_out(:,5:7)) ./
abs(MatriceDef_out(:,8:10))) * 100;
Perc_mod = (abs(DIC_mod-FEM_mod) ./ DIC_mod) * 100;
ErrPerc = mean(abs(DiffDisplPerc));
ErrPerc_mod = mean(abs(Perc_mod));

% Calculation of RMSE
rmseX = sqrt(sum((MatriceDef_out(:,5)-MatriceDef_out(:,8)).^2)/size(MatriceDef_out,1));
rmseY = sqrt(sum((MatriceDef_out(:,6)-MatriceDef_out(:,9)).^2)/size(MatriceDef_out,1));
rmseZ = sqrt(sum((MatriceDef_out(:,7)-MatriceDef_out(:,10)).^2)/size(MatriceDef_out,1));
rmse_mod = sqrt(sum((FEM_mod-DIC_mod).^2)/size(MatriceDef_out,1));

% Calculation of RMSE%
rmsePX = (rmseX/abs(max(MatriceDef_out(:,8))))*100;
rmsePY = (rmseY/abs(max(MatriceDef_out(:,9))))*100;
rmsePZ = (rmseZ/abs(max(MatriceDef_out(:,10))))*100;
rmseP_mod = rmse_mod/abs(max(DIC_mod))*100;

% Calculation of R-squared
[signFit_x, GoF_x, ~] = fit(MatriceDef_out(:,8),MatriceDef_out(:,5), 'poly1');
[signFit_y, GoF_y, ~] = fit(MatriceDef_out(:,9),MatriceDef_out(:,6), 'poly1');
[signFit_z, GoF_z, ~] = fit(MatriceDef_out(:,10),MatriceDef_out(:,7), 'poly1');
[signFit_mod, GoF_mod, ~] = fit( DIC_mod,FEM_mod, 'poly1');
R2Y = GoF_y.rsquare;
R2X = GoF_x.rsquare;
R2Z = GoF_z.rsquare;
R2mod = GoF_mod.rsquare;

X = [R2X; rmseX; rmsePX; ErrAv(1); ErrPerc(1)];
Y = [R2Y; rmseY; rmsePY; ErrAv(2); ErrPerc(2)];
Z = [R2Z; rmseZ; rmsePZ; ErrAv(3); ErrPerc(3)];
TOT = [R2mod; rmse_mod; rmseP_mod; ErrAv_mod; ErrPerc_mod];

MAT2 = table({'R^2';'RMSE';'RMSE%';'errore assouto medio';'errore % medio'}, X, Y, Z, TOT);

% Strains
% Centroids
pfem = mean(F0R);
qfem = mean(F1R);
F0c = F0R-pfem;
F1c = F1R-qfem;
displFEM_c = F1c-F0c;

pdic = mean(D0);
qdic = mean(D130);
D0c = D0-pdic;
D130c = D130-qdic;
displDIC_c = D130c-D0c;

% SVD to FEM points
E = (F0c)'*(F1c);
[U,S,V] = svd(E);
R1 = V*U'; % rotation matrix
T1 = mean(F1c)' - R1*mean(F0c)'; % translation vector
F1_rot_trasl = (R1*(F0c)')';
rigidFEM = F1_rot_trasl - (F0c);
defFEM = displFEM - rigidFEM;
UtotFEM = rigidFEM + defFEM;

trisFEM = MyCrustOpen(F0c);
```

```
strainFEM_def = triStrain(trisFEM,F0c,defFEM);
strainFEM_n_def = triToNodes(strainFEM_def,trisFEM);
strainFEM_n_def = [strainFEM_n_def(:,2),strainFEM_n_def(:,1)];

% Interpolation of DIC to FEM strains
IntS_dic(:,1) = idw(D0c,S130(:,1),F0c,2,1);
IntS_dic(:,2) = idw(D0c,S130(:,2),F0c,2,1);

ind1 = find(isnan(IntS_dic(:,1)) == 0);
IntS_dic_new = IntS_dic(ind1,:);
strainFEM_n_new = strainFEM_n_def(ind1,:);
strainDef = [strainFEM_n_new, IntS_dic_new];

% Calculation of RMSE for strains
rmseEps2 = sqrt(sum((strainDef(:,2)-strainDef(:,4)).^2)/size(strainDef,1));
rmseEps1 = sqrt(sum((strainDef(:,1)-strainDef(:,3)).^2)/size(strainDef,1));

% Calculation of RMSE% for strains
rmsePEps2 = (rmseEps2/abs(max(strainDef(:,4))))*100;
rmsePEps1 = (rmseEps1/abs(max(strainDef(:,3))))*100;

Eps2 = [rmseEps2; rmsePEps2];
Eps1 = [rmseEps1; rmsePEps1];

MAT_deform = table({'RMSE';'RMSE%'}, Eps1, Eps2);

% Connectivity matrix
load("connectivity.mat");

roi = MatriceDef;
mat_def2=[[1:length(roi)]' roi];

% Surface faces extraction
F_all = list_faces(connectivity);
F_surf = extract_unique_triangles(F_all);
connectivity = F_surf;

connectivitynew = zeros(size(connectivity,1),size(connectivity,2));
L = ismember(connectivity,mat_def2(:,2));

for i=1:size(connectivity,1)
    if sum(L(i,:)) == 3
        for j=1:size(connectivity,2)
            aa = find(mat_def2(:,2) == connectivity(i,j));
            connectivitynew(i,j) = mat_def2(aa,1);
        end
    end
end

connectivitynew( ~any(connectivitynew,2), : ) = [];

figure()
trisurf(connectivitynew,roi(:,2),roi(:,3),roi(:,4))
axis equal

% Connectivity matrix built on initial coord of FEM
connectivity0 = MyCrustOpen(MatriceDef(:,2:4));

figure()
trisurf(connectivitynew,roi(:,2),roi(:,3),roi(:,4))
hold on
trisurf(connectivity0,MatriceDef(:,2),MatriceDef(:,3),MatriceDef(:,4),'FaceColor',[0 1 1])
axis equal
legend('Connettività estratta dal fem','Connettività costruita con MyCrustOpen')

% Difference between DIC and FEM displacements
```

```
DiffRoi = abs(MatriceDef(:,8:10) - MatriceDef(:,5:7));
DiffPercRoi = (abs(MatriceDef(:,8:10) - MatriceDef(:,5:7)) ./ abs(MatriceDef(:,8:10))) * 100;

figure()
subplot(2,2,1)
plotField(connectivitynew,MatriceDef(:,2:4),DiffRoi(:,1))
axis equal
colorbar
xlabel('x')
ylabel('y')
zlabel('z')
title('Displacements along X')
subplot(2,2,2)
plotField(connectivitynew,MatriceDef(:,2:4),DiffRoi(:,2))
axis equal
colorbar
xlabel('x')
ylabel('y')
zlabel('z')
title('Displacements along Y')
subplot(2,2,3)
plotField(connectivitynew,MatriceDef(:,2:4),DiffRoi(:,3))
axis equal
colorbar
xlabel('x')
ylabel('y')
zlabel('z')
title('Displacements along Z')
sgtitle('Difference for displacements - (DIC-FEM)')

figure()
subplot(2,2,1)
plotField(connectivitynew,MatriceDef(:,2:4),DiffPercRoi(:,1))
axis equal
colorbar
xlabel('x')
ylabel('y')
zlabel('z')
title('Displacements along X')
subplot(2,2,2)
plotField(connectivitynew,MatriceDef(:,2:4),DiffPercRoi(:,2))
axis equal
colorbar
xlabel('x')
ylabel('y')
zlabel('z')
title('Displacements along Y')
subplot(2,2,3)
plotField(connectivitynew,MatriceDef(:,2:4),DiffPercRoi(:,3))
axis equal
colorbar
xlabel('x')
ylabel('y')
zlabel('z')
title('Displacements along Z')
sgtitle('Difference % for displacements - (DIC-FEM)')
```

Bibliography

[1] P. Wojciech and R. M. H, *Istologia. Testo e Atlante : Con elementi di biologia cellulare e molecolare*. Zanichelli, 2021.

[2] S. C. (Stephen C. Cowin, *Bone mechanics handbook / edited by Stephen C. Cowin.*, 2nd ed. Boca Raton [etc.]: Boca Raton [etc.] : CRC Press, c2001, 2001.

[3] J. Wolff, *The Law of Bone Remodelling*, 1st ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1986. doi: 10.1007/978-3-642-71031-5.

[4] D. M. Cullinane and T. A. Einhorn, “Chapter 2 - Biomechanics of Bone,” in *Principles of Bone Biology (Second Edition)*, J. P. Bilezikian, L. G. Raisz, and G. A. Rodan, Eds., San Diego: Academic Press, 2002, pp. 17–32. doi: https://doi.org/10.1016/B978-012098652-1.50104-9.

[5] H. Amirazad, M. Dadashpour, and N. Zarghami, “Application of decellularized bone matrix as a bioscaffold in bone tissue engineering,” *J Biol Eng*, vol. 16, no. 1, p. 1, 2022, doi: 10.1186/s13036-021-00282-5.

[6] G. L. Lucas, F. W. Cooke, and E. Friis, “A Primer of Biomechanics,” in *Springer: New York*, 1998. [Online]. Available: https://api.semanticscholar.org/CorpusID:7431212

[7] T. M. Keaveny and W. C. Hayes, “Mechanical properties of cortical and trabecular bone,” *Bone*, vol. 7, pp. 285–344, 1993.

[8] G. Anastasi, G. Anastasi, and G. Anastasi, *Trattato di anatomia umana / Giuseppe Anastasi ... [et al.]*, 4. ed. Milano: Edi-Ermes, 2010.

[9] B. E. Hirsch, “Gray’s Anatomy: The Anatomical Basis of Clinical Practice,” *JAMA*, vol. 301, no. 17, pp. 1825–1831, May 2009, doi: 10.1001/jama.2009.594.

[10] S. Agarwal, “Osteolysis - Basic science, incidence and diagnosis,” *Curr Orthop*, vol. 18, no. 3, pp. 220–231, 2004, doi: 10.1016/j.cuor.2004.03.002.

[11] E. Em, “The treatment of trochanteric fractures of the femur.,” *J Bone Joint Surg Br*, vol. 31B 2, pp. 190–203, 1949, [Online]. Available: https://api.semanticscholar.org/CorpusID:209165911

[12] B. J. Mcgrory, “Fractures of the proximal femur,” 2017. [Online]. Available: https://www.researchgate.net/publication/285778427

[13] F. Bayram and M. Çakirołlu, “DIFFRACT: DIaphyseal Femur FRActure Classifier SysTem,” *Biocybern Biomed Eng*, vol. 36, no. 1, pp. 157–171, 2016, doi: 10.1016/j.bbe.2015.10.003.

[14] E. G. Meinberg, J. Agel, C. S. Roberts, M. D. Karam, and J. F. Kellam, “Fracture and Dislocation Classification Compendium-2018,” *J Orthop Trauma*, vol. 32, pp. S1–S170, Jan. 2018, doi: 10.1097/BOT.0000000000001063.

[15] M. E. Hake, M. E. Davis, A. M. Perdue, and J. A. Goulet, “Modern Implant Options for the Treatment of Distal Femur Fractures,” *Journal of the American Academy of Orthopaedic Surgeons*, vol. 27, no. 19. Lippincott Williams and Wilkins, pp. E867–E875, Oct. 01, 2019. doi: 10.5435/JAAOS-D-17-00706.

[16] C. Kanchanomai, V. Phiphobmongkol, and P. Muanjan, “Fatigue failure of an orthopedic implant - A locking compression plate,” *Eng Fail Anal*, vol. 15, no. 5, pp. 521–530, Jul. 2008, doi: 10.1016/j.engfailanal.2007.04.001.

[17] D. Ruffoni, “3.307. Finite Element Analysis in Bone Research: A Computational Method Relating Structure to Mechanical Function.”

[18] V. Jagota, A. Preet, S. Sethi, and K. Kumar, “Finite Element Method: An Overview,” 2013. [Online]. Available: http://wjst.wu.ac.th

[19] E. Verhulp, B. van Rietbergen, and R. Huiskes, “Comparison of micro-level and continuum-level voxel models of the proximal femur,” *J Biomech*, vol. 39, no. 16, pp. 2951–2957, 2006, doi: 10.1016/j.jbiomech.2005.10.027.

[20] M. Palanca, G. Tozzi, and L. Cristofolini, “The use of digital image correlation in the biomechanical area: A review,” *International Biomechanics*, vol. 3, no. 1. Taylor and Francis Ltd., pp. 1–21, 2016. doi: 10.1080/23335432.2015.1117395.

[21] L. Wittevrongel, “A self adaptive algorithm for accurate strain measurements using global digital image correlation,” 2015.

[22] H. Schreier, J. J. Orteu, and M. A. Sutton, *Image correlation for shape, motion and deformation measurements: Basic concepts, theory and applications*. Springer US, 2009. doi: 10.1007/978-0-387-78747-3.

[23] P. Lava, S. Cooreman, S. Coppieters, M. De Strycker, and D. Debruyne, “Assessment of measuring errors in DIC using deformation fields generated by plastic FEA,” *Opt Lasers Eng*, vol. 47, no. 7, pp. 747–753, 2009, doi: https://doi.org/10.1016/j.optlaseng.2009.03.007.

[24] M. Rossi, P. Lava, F. Pierron, D. Debruyne, and M. Sasso, “Effect of DIC spatial resolution, noise and interpolation error on identification results with the VFM.”

[25] J. Luo, B. Liu, P. Yang, and X. Fan, “High-speed vision measurement of vibration based on an improved ZNSSD template matching algorithm,” *Systems Science and Control Engineering*, vol. 10, no. 1, pp. 43–54, 2022, doi: 10.1080/21642583.2021.2024099.

[26] J. Elektronik, I. K. Udayana, W. D. Sidi, I. Gede, and A. Wibawa, “Sum of Squared Difference (SSD) Template Matching Testing on Writing Learning Application,” vol. 8, no. 4, pp. 2654–5101, 2020.

[27] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, “The Finite Element Method: Its Basis and Fundamentals Sixth edition.” [Online]. Available: www.cimne.upc.es

[28] D. H. Ro, S. Lee, S. C. Seong, and M. C. Lee, “Lateral Closing Wedge Osteotomy of Tibia for Degenerative Arthritis,” *J Korean Orthop Assoc*, vol. 49, no. 2, pp. 95–106, Apr. 2014, [Online]. Available: https://doi.org/10.4055/jkoa.2014.49.2.95

[29] “Vic-3d Software Manual.”

[30] P. J. Besl and N. D. McKay, “A method for registration of 3-D shapes,” *IEEE Trans Pattern Anal Mach Intell*, vol. 14, no. 2, pp. 239–256, 1992, doi: 10.1109/34.121791.