UNIVERSITA' DEGLI STUDI DI PARMA

Facoltà di Ingegneria

Dottorato di ricerca in Ingegneria Industriale

XXII Ciclo

DEVELOPMENT OF A DIGITAL IMAGE CORRELATION SOFTWARE FOR FULL-FIELD STRAIN ANALYSIS OF CFRP

Coordinatore:
Chiar.mo Prof. Ing. Marco SPIGA

Dottorando:
Giancarlo ANZELOTTI

Tutor:
Ing. Enrica RIVA

This work is licensed under the *Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported* License. You are free:

- **to share:** to copy, distribute, and transmit the work
- **to remix:** to adapt the work

Under the following conditions:

- **Attribution:** you must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- **Noncommercial:** you may not use this work for commercial purposes.
- **Share Alike:** if you alter, transform, or build upon this work, you may distribute the resulting work only under a license identical to this one.

With the understanding that:

- **Waiver:** any of the above conditions can be waived if you get permission from the copyright holder.
- **Public Domain:** where the work or any of its elements is in the public domain under applicable law, that status is in no way affected by the license.
- **Other Rights:** in no way are any of the following rights affected by the license:
  - your fair dealing or fair use rights, or other applicable copyright exceptions and limitations;
  - the author's moral rights;
  - rights other persons may have either in the work itself or in how the work is used, such as publicity or privacy rights.

For any reuse or distribution, you must make clear to others the license terms of this work.

To view a full copy of this license, visit `http://creativecommons.org/licenses/by-nc-sa/3.0/` or send a letter to Creative Commons, 171 2nd Street, Suite 300, San Francisco, California, 94105, USA.

Giancarlo Anzelotti 🅭 2010

To my wife, شقایق *, for her kind help*
and her endless support.

Contents

Chapter

1

Introduction

Composite materials have become an attractive alternative to standard materials (e.g. plastics and metals) in recent years. These materials have many applications, especially in high performance applications, such as aerospace, marine, sport cars, etc. Their wide use is reducing drastically the production costs and so they started to be used also for their esthetic appearance and fashion, as they became symbol of high-tech and performances.

Since Charles Lindbergh made the first nonstop solo flight across the Atlantic in May 1927, airplanes have been made mainly of riveted metal. In last years this paradigm has been changed drastically. For example Boeing's 787 Dreamliner, Figure 1.1(a), that made its first flight in December 15^{th} 2009, consists of [1] 50% of composites, 20% of aluminum, 15% of titanium, 10% of steel and 5% of other materials. This is very important considering that the Boeing 777 (presented in August 2009) had just 12% of composite materials [1].

In Boeing 787 carbon laminates with epoxy resin form most of the fuselage exterior. In some areas also sandwiches of carbon fibers reinforced plastic has been used. The plane nose and the junctions of the wings with the fuselage are made mainly of glass fibers. The need of composites materials is mainly due to the high strength-to-weight ratio that is very important in applications like airplanes where the weight is strictly related to the costs, while the strength is related to the speed and number of passengers (the payload) that can be hosted and also the available pressure inside the cabin and so the comfort of the passengers.

Another important progress in the airplane industries is the new Airbus A380. It is made with about 22% of composite materials, mainly carbon fiber reinforced plastic but also glass and quartz fiber reinforced plastic and it is also highly using glass fiber reinforced aluminum (GLARE), an hybrid honeycomb material, that is 25% stronger than conventional airframe aluminum but 20% lighter (source `http://www.newscientist.com`). Its main advantage is the fatigue resistance. Composed of alternating layers of four or more 0.38 mm aluminum sheets and glass fiber reinforced bond film, the material thickness can be varied by adding sheets to match local property requirements. The glass fibers both control the load path and work as cracks stopper. Corrosion resistance is also increased relative to aluminum.

Carbon fiber reinforced plastic is used for the central box of the wings, the horizontal sta-

(a) Boeing 787 during its first flight, in December 2009

(b) Airbus A380 from Singapore Airlines

(c) Visby class corvettes, designed with stealth technology

(d) Lamborghini Murciélago LP 650-4 Roadster

Figure 1.1: Example of composite materials applications

bilizers, the fin, the rear fuselage section and for ceiling beams. Glare parts are used in the construction of the panels for the upper fuselage.

Another interesting example of composites construction is the Visby corvette stealth ship (see Figure 1.1(c)). As in last years most ships are switching over to stealth technology to enhance their first strike capability in case of conflicts, also the Royal Swedish Navy (RSV) developed its own stealth solution. These kind of ships have non-magnetic hull (mainly carbon fibers reinforced plastic) that features large, flat surfaces and sharp edges to lower its radar cross section, hydro acoustic, magnetic, infra red, optical signatures.

The Visby design started in 1996 and the first ships were ready around 2000–2002. Even if it is still a relatively old project, it is still a state-of-art design of stealth technology and of use of composite materials.

Lamborghini Murciélago LP 650-4 Roadster (Figure 1.1(d)) is certainly a good example of state-of-art composite structure in the sport cars field. The body is completely of carbon fiber reinforced plastic. The result is low weight, exemplary energy absorption and therefore a high level of occupant safety, plus a torsional rigidity that is unprecedented for an open-top car.

Many other examples can be used to show the high use of composites where the need of performance is critical. Roughly speaking, any material consisting of at least two components with different properties and distinct boundaries between themselves can be considered as a composite material, consequently there are many different kinds of composite materials, with different classifications according to different parameters. An interesting classification is shown in Figure 3.2.

Figure 3.2 is mainly focusing on the fiber reinforced composites, namely a matrix reinforced by dispersed phase in form of fibers (see Section 3.1.2), because are generally more used and shows more interesting mechanical properties. On the other hands they are more complex to study as they are made of a more complex architecture. It is obvious that the mechanical properties of the composite material depend strictly on the interactions between the components. These interactions can lead to complex behaviors, especially when the mechanical properties of the components are very different.

Looking at the microscopic scale, there are different damage modes: fiber breaking, transverse cracking and shear cracking [2]. Considering a simple twill-weave architecture (see Section 3.4.2) as an example, the damage evolution can be quite complex to understand and describe. The transverse cracks occurs in weft yarns, where there is maximum out-of-plane deformation. Then these cracks evolve and grow connecting with each other along the weft yarns. These are also propagating along the matrix till reaching the surface of the specimen where they are more visible. This behavior keeps going on along the transverse direction (i.e. perpendicular to the loading direction) till the final catastrophic collapse [2, 3].

It is obvious that such complex structures and damage behaviors must be studied carefully and at different magnification levels. At the micro-level the macroscopic mechanical properties must fulfill the static and dynamic minimum characteristics required. At the meso-level it is more important to study the damage evolution as design approaches as damage tolerant or similar need to know the properties of the composite material after a damage occurs. At the micro-level the interaction between the constituents leads to unpredictable failures if not considered.

Considering the complexity of studying the damage behavior, the traditional measuring methods are not satisfactory and therefore many researchers focused their work on novel approaches. An example is the numerical simulation of the composite materials. Most of the effects influencing the mechanical properties, start from internal interrelations between different parts of the components. It is almost impossible to observe or to measure them using just laboratory experiments. In such study a numerical simulation may help.

Although the numerical simulation is a powerful method, it may mislead the researchers because of the lake of measurable information, therefore it has to be used carefully. The numerical simulation has to be validated using a real calibrating measurement. For example a simple characteristic that is also measurable by traditional methods such as tensile properties. On the other hands we need some mesoscopic information and the numerical model has to respect all the measurable parameters. There are many approaches to follow and to check the validation of a numerical model.

In addition to the traditional methods for measuring the mechanical properties, that are mainly contact methods, there are some non-contact methods. The most important non-contact methods are optical techniques. Digital image correlation (DIC) technique is a non-contact and non-destructive tool to measure 2D or even 3D strain map of specimens under loading. DIC is an easy, affordable and encouraging method that is growing in the field of experimental mechanics.

Considering the importance of having a correct knowledge of the mechanical behavior of composite materials and their applications in industries, military, aerospace, etc., this study is an investigation on the strain behavior as one of the most important mechanical responses of every kind of composite materials.

The main focus of this study is the development of a software to capture the 2D strain mapping of specimens. This software, called *Phoenix*©, is based on numerical approaches applying DIC fundamentals. This study continues with presenting several examples of tests on different materials such as carbon fiber reinforced plastic, nodular cast iron and aluminum alloy. Regarding to the remarkable power of numerical analyses, this study is trying to connect two potent numerical methods, DIC and FEM, together. Therefore the second part of the study is starting some researches on the comparison between FEM and DIC analyses. There is also an example of the comparison between the the results of DIC analyses and FEM analyses of the extension behavior applied on carbon fiber reinforced plastic material with twill weave architecture. The study ends in introducing an anisotropic non-linear FEM analysis applied on non-crimp fabric carbon composite materials, implemented in Abaqus.

The following paragraphs are giving the summary of this thesis.

1.1 Summary of the thesis

Chapter 2 is an investigation on the strain measurement methods. In this chapter the most important traditional strain measurement methods are introduced. This investigation has made mainly on extensometers and strain-gages. It has also a brief review about full-field techniques.

Chapter 3 contains a brief summary on composite materials. In this chapter different categories and different classifications of composite materials are introduced. The chapter is mainly focused on fiber reinforced composite materials and specially on carbon fiber reinforced polymers. Woven fabrics and their producing methods have been investigated as well.

Chapter 4 presents the main issue of this study. It starts with a study on digital image correlation. It continues with the development of a software for the strain measurements by DIC method. In order to understand better the available outputs and the reliability of the presented software, some applications are presented. Some tests have been designed for carbon fiber reinforced plastic materials, nodular cast iron and aluminum alloy. The results of applying this software have been compared with the results of a commercial software (Vic-2D) and with some artificial strain behavior in order to check the reliability of the software predictions.

Regarding to the strong relation that can be established between DIC and finite element modeling, Chapter 5 is a focus on the development of a numerical model for the study of the anisotropic non-linear behavior of non-crimp fabric composites. The chapter starts with an introduction on the simulation approach and on the theory used for the anisotropic non-linear description of the studied material. Consequently the model is described and finally the results presented.

Chapter

2

Strains measurement methods

IN ENGINEERING it is common to consider the bodies as rigid-bodies. Namely "objects" that may only move in the space, but that cannot have any internal deformation. Taken any couple of points in the volume of the body, after displacement the distance between these two points remains fixed.

This simplification is very good in many engineering fields, but not when we are dealing with the effects due to external loading[1]. These loads generally induce deformations to the bodies, that can be classified in rotations, translations and strains.

The first two can be collected in one unique concept: rigid-body displacement. The last is somehow the measurement of how far is the complete displacement of our body from a rigid-body one.

It is better to say that *the strain is the dimensional change in shape or volume of a body as result of an applied stress*[2] *or stresses* [5]. The stresses are strictly related to the loads and are the cause of the deformation. The relation between stresses and strains is expressed by the *constitutive equations*, that differ according to the material behavior.

A simple way to express this relation is using the stress and strain tensors:

$$\underline{\sigma} = \underline{\underline{E}} \cdot \underline{\epsilon} \tag{2.1}$$

where $\underline{\sigma}$ is the stress tensor, $\underline{\epsilon}$ is the strain tensor and $\underline{\underline{E}}$ is a 4th order tensor called *stiffness tensor*.

A simple situation is the one of one-dimensional approach, where are considered only the stresses and strains along one unique direction. In this case there will be just one stiffness constant that it is called *elastic modulus*:

$$\sigma = E \cdot \epsilon \tag{2.2}$$

---

[1] Here "load" is used as general term for any external agent influencing the system, as forces, temperatures, etc.

[2] The *stress* is the internal normal force a material exerts per unit of its area resulting from externally applied loading [4] thus it will be normally N/mm^2 or more commonly MPa

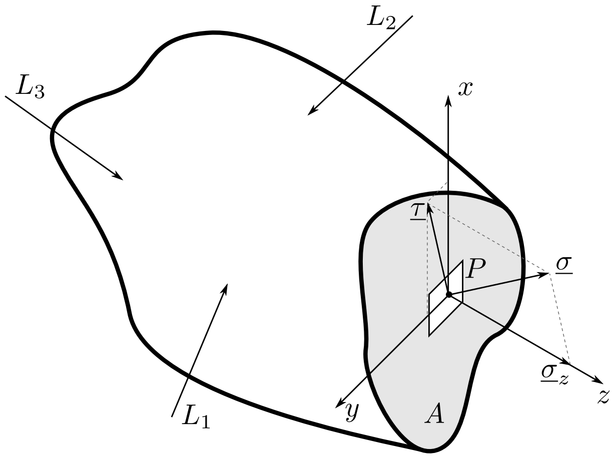


Figure 2.1: Stress distribution on a generic deformable body

It follows that a very simple equation to find the strain will be obtained just dividing the dimensional change by the original dimension:

$$\epsilon = \frac{L - L_0}{L_0} = \frac{\Delta L}{L_0} \tag{2.3}$$

Even if this equation is enough valid for most of the engineering requirements, actually it is only valid to achieve general information averaged over an area or a volume. In fact using the original dimension to calculate the strain according to Equation (2.3), leads to an average value of strain over the all dimensional change, and not an instant strain value.

To obtain this last value, the instant one, it is needed a different equation, that is mainly similar to the previous one, but where ΔL is brought close to zero:

$$\epsilon = \frac{dL}{L} \tag{2.4}$$

In a two-dimensional space one unique value of strain or stress is not enough, but applying an external load causes a reaction in terms of strain along the load and also a strain perpendicular to this load. This strain, studied mainly by Poisson, is proportional to the longitudinal strain by the following equation:

$$\epsilon_T = -\nu \cdot \epsilon_L \tag{2.5}$$

where ν is a material constant called *Poisson's ratio*. Generally speaking stresses and strains are three-dimensional vectors. The Figure 2.1 shows a general deformable body, cut by a plane lying on the point P. Because of the applied external loads L_i, there will be internal stresses reacting to these loads. Considering an infinitesimal area around the point P, it can be found the stress vector $\underline{\sigma}$ that depends on the point P and the orientation of the cutting plane. The stress $\underline{\sigma}$ can be thought as sum of a stress normal to the cutting plane, σ_z (considered just as intensity of a

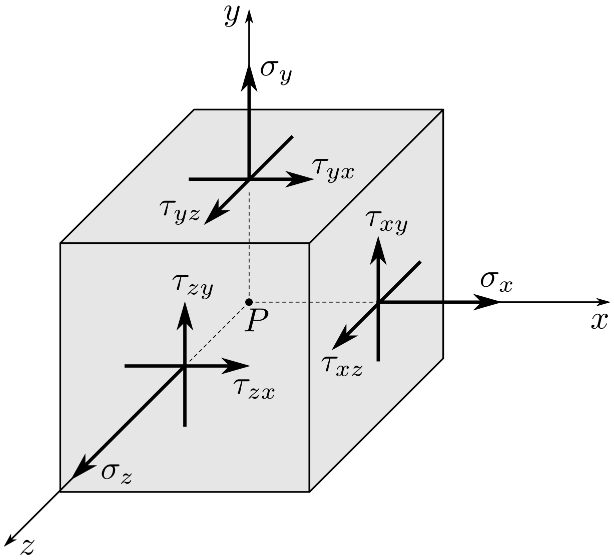


Figure 2.2: Three-dimensional stress distribution in a generic point P

vector with direction z), and a shear stress lying on the plane, $\underline{\tau}$. This last stress can be divided in its component along x, τ_{zx}, and its component along y, τ_{zy}. In order to know completely the stress state at the point P it is needed to cut the body with three planes perpendicular to the three axes of the coordinate system. Thanks to these cuts the components $\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{yz}$ and τ_{xz} will be known (for equilibrium $\tau_{ij} = \tau_{ji}$). The relation between the cutting plane and the stresses is well known and can be found for example in [6].

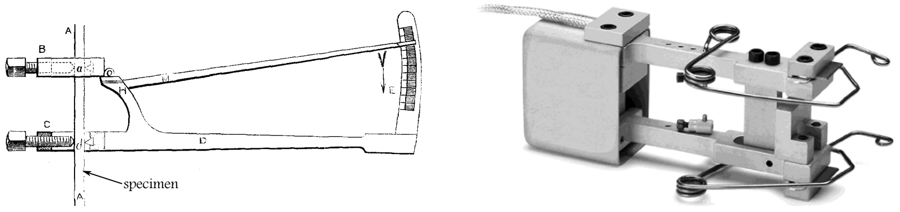
Similar remarks can be outlined for the strains, where it is possible to define $\epsilon_x, \epsilon_y, \epsilon_z, \gamma_{xy}, \gamma_{yz}$ and γ_{xz}. In a generic three-dimensional space the Equation (2.1) can be written also in matrix form as follows:

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{xz} \end{bmatrix} = \begin{bmatrix} e_{1,1} & e_{1,2} & e_{1,3} & e_{1,4} & e_{1,5} & e_{1,6} \\ & e_{2,2} & e_{2,3} & e_{2,4} & e_{2,5} & e_{2,6} \\ & & e_{3,3} & e_{3,4} & e_{3,5} & e_{3,6} \\ & & & e_{4,4} & e_{4,5} & e_{4,6} \\ & sym. & & & e_{5,5} & e_{5,6} \\ & & & & & e_{6,6} \end{bmatrix} \cdot \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{bmatrix} \tag{2.6}$$

2.1 Extensometers

The *extensometer* is an instrument for measuring changes in linear dimensions [7]. It was invented by Charles Huston who described it in an article in the *Journal of the Franklin Institute* in 1879 which scheme can be seen in Figure 2.3(a) [8]. There are different ways to categorize the extensometers, but they are mainly divided in contact and non-contact devices.

The former is able to measure the overall strain of a part of the specimen by keeping the



(a) Original extensometer by Dr. Charles Huston [8]

(b) Example of modern contact extensometer

Figure 2.3: Evolution of contact extensometers along the history

device in contact with it. The latter devices are based on different systems, most of them optical.

2.1.1 Contact extensometers

The contact extensometers are based on the measurement of the displacement of two "knifes" separated by a certain gage-length. Thus the strain will be calculated as the measured displacement, ΔL, divided by the gage length L_0. These knifes must follow the deformation of the specimen and so they are tightly in contact with it, resulting in a possible damage for the specimen (if it is particularly small and/or weak compared to the extensometer) or influence the results of the test.

As it is shown in Figure 2.3(b), the extensometer presents two knifes and two springs to keep the device in touch with the specimen during the test. The extensometer is designed to consider a particular gage-length (e.g. 20 mm) but it is possible to change the real gage-length by adding some extensions. Usually the gage-length range may even go from 10 mm till 50 mm. It is obvious that as the device is designed for one particular value of the gage-length, the output of the extensometer must be corrected if a different value is used.

Modern contact extensometers are made of an LVDT device, namely a *linear variable differential transformer*. The LVDT is a type of electrical transformer used for measuring linear displacement. It works by giving a different voltage as the output of the device according to the displacement of a metallic bar moving linearly. It is composed of a primary coil and two secondary coils, all around a sliding ferromagnetic core. An alternating current is induced on each secondary coil by an input alternating current on the primary coil, due to the mutual inductance between the primary and the secondaries. Connecting the two secondary coils in order to subtract the two voltages, the results will be zero voltage in central position of the sliding bar, and increasing voltage in all the other positions. The direction of sliding is given by the phase of output respect to input (i.e. the same phase for one direction and the opposite phase for opposite direction). This configuration of the circuit gives the name of *differential* to the device.

A correct calibration of the device makes it giving an output proportional to the strain, that is actually displacement per unit of gage-length.

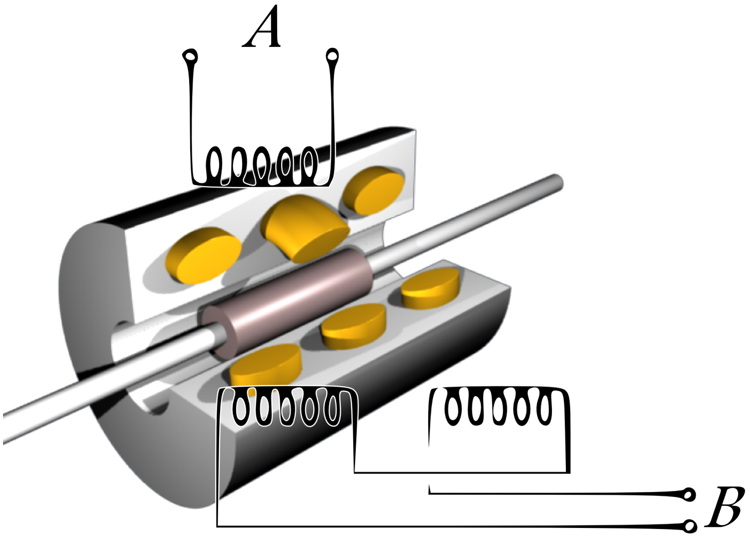


Figure 2.4: Scheme of an LVDT device

2.2 Strain-gages

The strain-gage is a device used to measure the strain of an object. Invented in early 1980s by Edward E. Simmons and Arthur C. Ruge, in recent years it is developed by several companies and in hundreds of models. The strain-gage is mainly made of a thin flexible plastic backing supporting a metallic foil pattern. Attaching the device to a deformable body will cause the deformation of the foil too and so a variation in its electrical resistance, measurable for example by a Wheatstone bridge.

Considering for simplicity the metallic pattern as made of a small metallic wire with circular cross section, then the electrical resistance R of the electrical conductor will be:

$$R = \rho \cdot \frac{L}{S} \tag{2.7}$$

where L is the length of the wire, S is the cross section and ρ is the electrical resistivity of the material. Expressing the natural logarithm of Equation (2.7) and differentiating leads to:

$$\frac{dR}{R} = \frac{d\rho}{\rho} + \frac{dL}{L} - 2 \cdot \frac{dr}{r} \tag{2.8}$$

where r is the radius of the wire (as it is $S = \pi \cdot r^2$). Using Equations (2.4) and (2.5) it is possible to obtain:

$$\frac{dR}{R} = \epsilon \left[\frac{1}{\epsilon} \cdot \frac{d\rho}{\rho} + (1 + 2 \cdot \nu_{sg}) \right] = \epsilon \cdot S_g \tag{2.9}$$

where ν_{sg} is the Poisson's ratio of the material of the wire and S_g is the *gage-factor* of the strain-gage, namely the proportional coefficient between the resistance variation and the strain variation. This leads to the most important equation in extensimetry:

$$\epsilon = \frac{1}{S_g} \cdot \frac{\Delta R}{R} \tag{2.10}$$

2.2.1 Gage-factor

The gage-factor is the most important parameter characterizing a strain-gage. It is considered constant according to the material, shape of the strain-gage and for a particular temperature range. The gage-factor is formed by the influence of two effects: the geometrical effect and the piezoelectrical effect. The former, $(1 + 2 \cdot \nu_{sg})$, is generally constant, because the Poisson's ratio is usually constant. The latter, $d\rho / (\rho \cdot \epsilon)$, is due to the fact that the deformation causes a variation of the electrical deformation as the volume remains constant and so length and cross section change.

Considering an elastic deformation, it is seen that usually the part due to piezoelectrical effect remains relatively constant and for some materials such as *constantan* and *karma* it is around 0.5, while the part due to geometrical effect is 1.6. Thus the gage-factor is 2.1. For plastic deformations the volume is constant and so the calculation of the gage-factor is as follows:

$$S_g = \frac{1}{\epsilon} \cdot \frac{\Delta R}{R} = \frac{1}{\epsilon} \cdot \frac{\Delta L^2 + 2 \cdot \Delta L \cdot L}{L^2} = \frac{1}{\epsilon} \cdot \left[\left(\frac{\Delta L}{L} \right)^2 + 2 \cdot \frac{\Delta L}{L} \right] = \frac{1}{\epsilon} \cdot \epsilon \left(\epsilon + 2 \right) \approx 2 \tag{2.11}$$

since

$$R + \Delta R = \rho \cdot \frac{L + \Delta L}{S} = \rho \cdot \frac{\left(L + \Delta L \right)^2}{V} = \frac{\rho}{V} \cdot L^2 + \underbrace{\frac{\rho}{V} \cdot \left(\Delta L^2 + 2 \cdot \Delta L \cdot L \right)}_{\Delta R} \tag{2.12}$$

2.2.2 Transverse sensitivity

Transverse sensitivity refers to the behavior of strain-gages in responding to strains perpendicular to the main axis of the strain-gage due to the shape of the metallic pattern of the strain-gage and due to the Poisson's ratio of it [9]. However the highest contribution is due to the portions of grid that is in the transverse direction. This effect is always present. However, in case of an uniaxial stress state with base material Poisson's ratio of 0.285 (that is the calibration experimental set-up) the effect is included in the calculation of the gage-factor. Therefore the user must take into account the correction because of transverse sensitivity in any of the following situations:

- the strain-gage is installed on a material with Poisson's ratio different from the one used for calibration (i.e. with $\nu = 0.285$)
- the strain-gage is installed on a material with Poisson's ratio of 0.285 but subjected to

other than uniaxial stress state

- the strain-gage is installed on a material with Poisson's ratio of 0.285 and subjected to uniaxial stress state but aligned along a direction other than the maximum principal stress

In general it can be considered that a strain-gage has two different gage-factors: one for a uniaxial strain state (thus biaxial stress state) along the main direction and one for uniaxial strain state along the transverse direction. For a generic strain field the output of the strain-gage will be:

$$\frac{\Delta R}{R} = S_a \cdot \epsilon_a + S_t \cdot \epsilon_t = S_a \cdot (\epsilon_a + K_t \cdot \epsilon_t) \tag{2.13}$$

where S_a is the axial gage-factor, S_t is the transverse gage-factor and K_t is the transverse sensitivity. Considering a material with Poisson's ration equal to ν_0 that is the one of the calibration material, the Equation (2.13) will be:

$$\frac{\Delta R}{R} = S_a \cdot (\epsilon_a - \nu_0 \cdot K_t \cdot \epsilon_a) = \underbrace{S_a \cdot (1 - \nu_0 \cdot K_t)}_{S_{sg}} \cdot \epsilon \tag{2.14}$$

that shows the relation between the gage-factor of the strain-gage and the effect of the transverse sensitivity.

Neglecting this effect will lead to an error, that can be calculated by the Equation (2.15):

$$\begin{aligned} e_{\%} &= \frac{\left.\frac{\Delta R}{R}\right|_{true} - \left.\frac{\Delta R}{R}\right|_{wrong}}{\left.\frac{\Delta R}{R}\right|_{true}} \cdot 100 = \frac{S_a \cdot \epsilon_a \cdot (1 - \nu_0 \cdot K_t) - S_a \cdot (\epsilon_a + \epsilon_t \cdot K_t)}{S_a \cdot \epsilon_a \cdot (1 - \nu_0 \cdot K_t)} \cdot 100 \\ &= \frac{K_t \cdot \left(\frac{\epsilon_t}{\epsilon_a} + \nu_0\right)}{1 - \nu_0 \cdot K_t} \cdot 100 \end{aligned} \tag{2.15}$$

Plotting this equation for different values of the transverse strain–axial strain ratio will lead to the curves of Figure 2.5 that show very well how the effect of transverse sensitivity is relatively negligible. Especially for standard values of transverse sensitivity. For example a strain-gage *Vishay CEA-06-250UN-350* has a transverse sensitivity of $(+0.2 \pm 0.2)\,\%$ at room temperature. In this case, even with a high transverse strain–axial strain ratio (e.g. equal to 4), the error will be 0.86% that can be acceptable for many situations.

Even if it is negligible, when precise measurements are required, this correction should be considered. This effect is especially of interest in the case of rosettes (see Section 2.2.3), where the strain state is usually not uniaxial (or however not aligned with all the grids of the rosette).

2.2.3 Single strain-gages and rosettes

Normally a strain-gage is made of a single plastic backing with a single metallic grid, as shown in Figure 2.6(a). This configuration is able to measure the strain along the main direction (longitudinal direction) of the strain-gage and in case to take into account the effect of strains

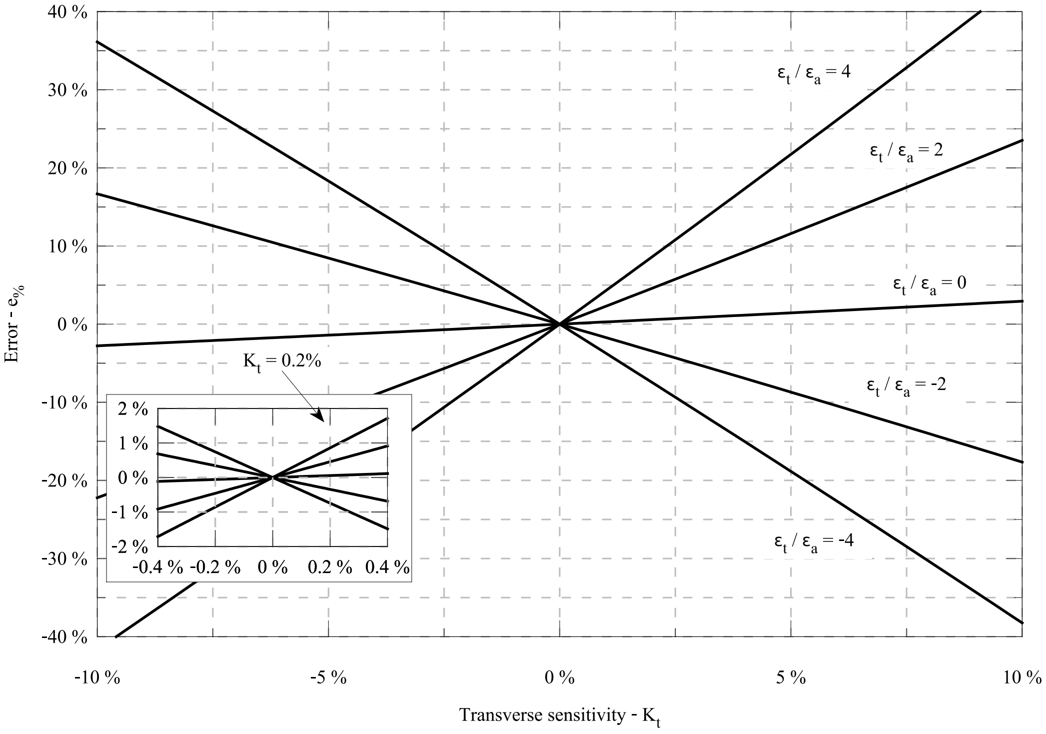
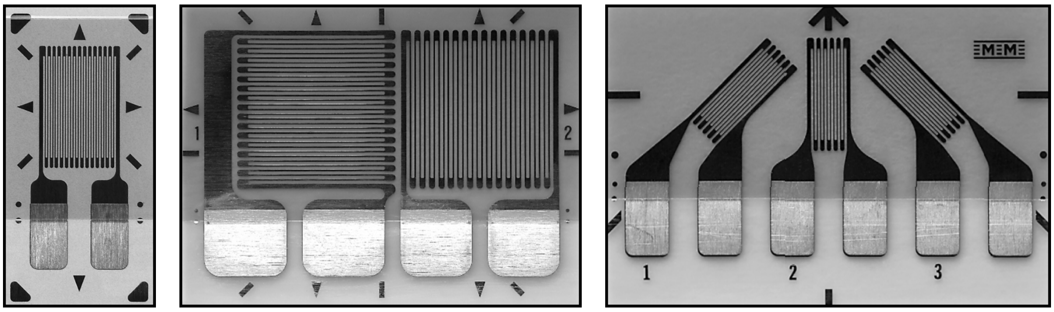


Figure 2.5: Error due to the transverse senstivity of strain-gages

in other directions thanks to the correction made by transverse sensitivity (see Section 2.2.2). If the strain state is complex (biaxial) or not known in advance, one unique strain-gage cannot give a correct and meaningful value. In these situations more than one strain-gage is needed in order to measure the strain along different directions. In order to have an easy installation and a precise measurement, it is possible to buy a group of strain-gages bound to a unique plastic backing. This solution is called *rosette.*

A biaxial rosette, as shown in Figure 2.6(b), is sufficient if the directions of the principal strains are known, but not their values. If the gage-factor suggested by the manufacturer is used, the measured strains will be:

$$\begin{cases} \tilde{\epsilon}_1 = \dfrac{S_a \cdot \epsilon_1 + S_t \cdot \epsilon_2}{S_g} \\ \tilde{\epsilon}_2 = \dfrac{S_a \cdot \epsilon_2 + S_t \cdot \epsilon_1}{S_g} \end{cases} \tag{2.16}$$



(a) Strain-gage (b) biaxial 90 degrees rosette (c) triaxial 45 degrees rosette

Figure 2.6: Different configurations of rosettes [10–12]

Solving the system of two equations in ϵ_1 and ϵ_2 will lead to the following:

$$\begin{cases} \epsilon_1 = \dfrac{(1-\nu_0 \cdot K_t) \cdot (\tilde{\epsilon}_1 - K_t \cdot \tilde{\epsilon}_2)}{1-K_t^2} \\ \epsilon_2 = \dfrac{(1-\nu_0 \cdot K_t) \cdot (\tilde{\epsilon}_2 - K_t \cdot \tilde{\epsilon}_1)}{1-K_t^2} \end{cases} \tag{2.17}$$

Considering (as commonly possible) $1 - K_t^2 \approx 1$ as a common possibility (coming back to the example of the strain-gage *Vishay CEA-06-250UN-350* this means that $0.999996 \approx 1$), it is possible to obtain:

$$\begin{cases} \epsilon_1 = \hat{\epsilon}_1 - K_t \cdot \hat{\epsilon}_2 \\ \epsilon_2 = \hat{\epsilon}_2 - K_t \cdot \hat{\epsilon}_1 \end{cases} \tag{2.18}$$

where $\hat{\epsilon}$ is the strain measured from the strain-gage if the gage-factor has been set as following:

$$\hat{S}_g = \frac{S_g}{1-\nu_0 \cdot K_t} \tag{2.19}$$

If the principal strain directions are not known in advance, three strain-gages as shown in Figure 2.6(c) are needed. Even in this situation is possible to link the measured strains to the real strains along the axes 1 and 2 by the following:

$$\begin{cases} \epsilon_1 = \dfrac{(1-\nu_0 \cdot K_t) \cdot (\tilde{\epsilon}_1 - K_t \cdot \tilde{\epsilon}_3)}{1-K_t^2} \\ \epsilon_2 = \dfrac{(1-\nu_0 \cdot K_t) \cdot [\tilde{\epsilon}_2 - K_t \cdot (\tilde{\epsilon}_1 + \tilde{\epsilon}_3 - \tilde{\epsilon}_2)]}{1-K_t^2} \\ \epsilon_3 = \dfrac{(1-\nu_0 \cdot K_t) \cdot (\tilde{\epsilon}_3 - K_t \cdot \tilde{\epsilon}_1)}{1-K_t^2} \end{cases} \tag{2.20}$$

where the transverse strain of the strain-gage number 2 can be obtained by the Mohr circle of the strains inside the rosette, as shown in Figure 2.7. Correcting the gage-factor as in Equation (2.19)

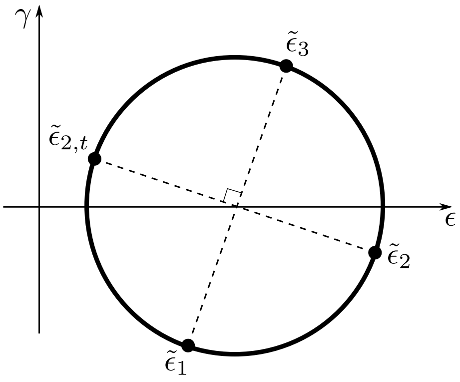


Figure 2.7: Mohr circle of the triaxial rosette of Figure 2.6(c)

will lead to:

$$\begin{cases} \epsilon_1 = \hat{\epsilon}_1 - K_t \cdot \hat{\epsilon}_3 \\ \epsilon_2 = \hat{\epsilon}_2 - K_t \cdot (\hat{\epsilon}_1 + \hat{\epsilon}_3 - \hat{\epsilon}_2) \\ \epsilon_3 = \hat{\epsilon}_3 - K_t \cdot \hat{\epsilon}_1 \end{cases} \tag{2.21}$$

2.2.4 Misalignments

The high sensitivity of the strain-gages leads to the fact that if the strain-gage is bonded with a misalignment, the measured value can be quite far from the real value. For example considering a uniaxial stress state (and so a biaxial strain state), the measured stress along a direction angled ϕ respect to the main principal direction will be:

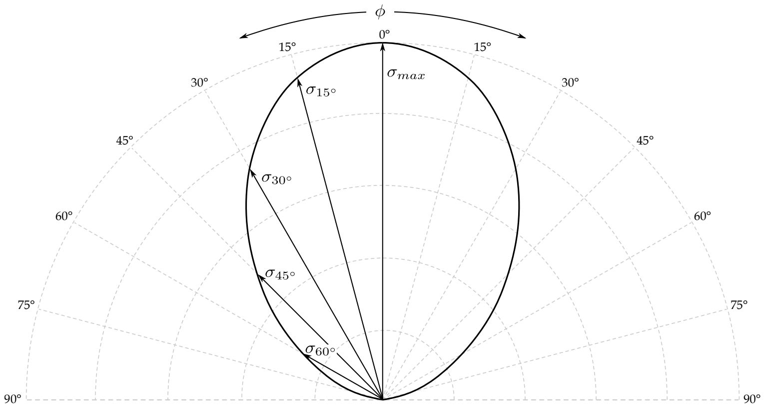
$$\sigma(\phi) = \frac{\sigma_{max}}{2} \cdot [1 + cos(2 \cdot \phi)] \tag{2.22}$$

The polar distribution of these stresses according the the angle ϕ is shown in Figure 2.8(a). Similar conclusions can be made considering the strains. As there is a uniaxial stress state, there will be a biaxial strain state due to the Poisson's effect:

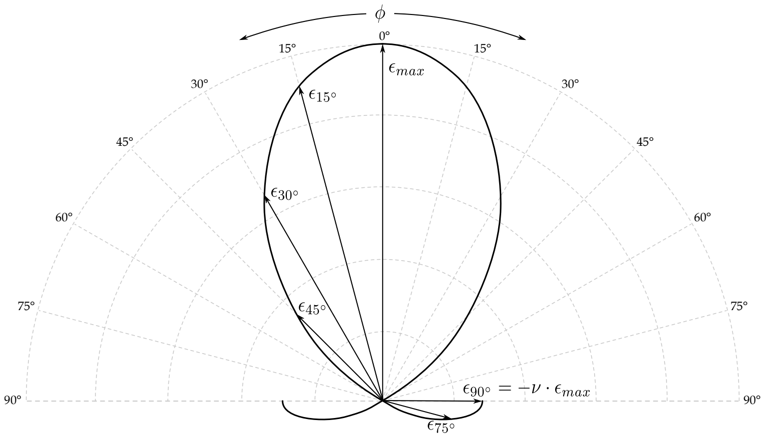
$$\epsilon(\phi) = \frac{\epsilon_{max}}{2} \cdot [1 - \nu + (1 + \nu) \cdot cos(2 \cdot \phi)] \tag{2.23}$$

which its distribution can be seen in Figure 2.8(b). Figure 2.8(b) shows that from an angle $\phi \approx 62°$ the strain gets zero and then negative till the value of $-\nu \cdot \epsilon_{max}$ for an angle of 90°.

The error due to the misalignment is introduced when the strain-gage is bonded with an angle of $\phi \pm \beta$ instead of the required ϕ angle respect to axial loading direction. This caused error is smaller when ϕ is around to 0° or 90° but it grows when ϕ gets far from these values. Setting the second derivative of Equation (2.23) equal to zero and solving for the angle ϕ, it is possible to see that the angle at which the error is higher is $\phi = 45°$. The loweset will be for angles of



(a) Stress distribution



(b) Strain distribution

Figure 2.8: Polar graph of the stress and strain distributions according to the angle ϕ

0° and 90° as aforementioned, apart from the case of constant strain field (as in a pressurized sphere), where the direction of bonding is not changing the measured value [13].

Considering now the percentage error respect to the correct alignment, there will be:

$$e_{\%} = \frac{\epsilon\left(\phi \pm \beta\right) - \epsilon\left(\phi\right)}{\epsilon\left(\phi\right)} \cdot 100 = \frac{cos\left[2 \cdot \left(\phi \pm \beta\right)\right] - cos\left(2 \cdot \phi\right)}{\dfrac{\epsilon_p + \epsilon_q}{\epsilon_p - \epsilon_q} + cos\left(2 \cdot \phi\right)} \cdot 100 \tag{2.24}$$

where ϵ_p and ϵ_q are the maximum and minimum principal strains respectively. The results for different values of β can be found in Figure 2.9.

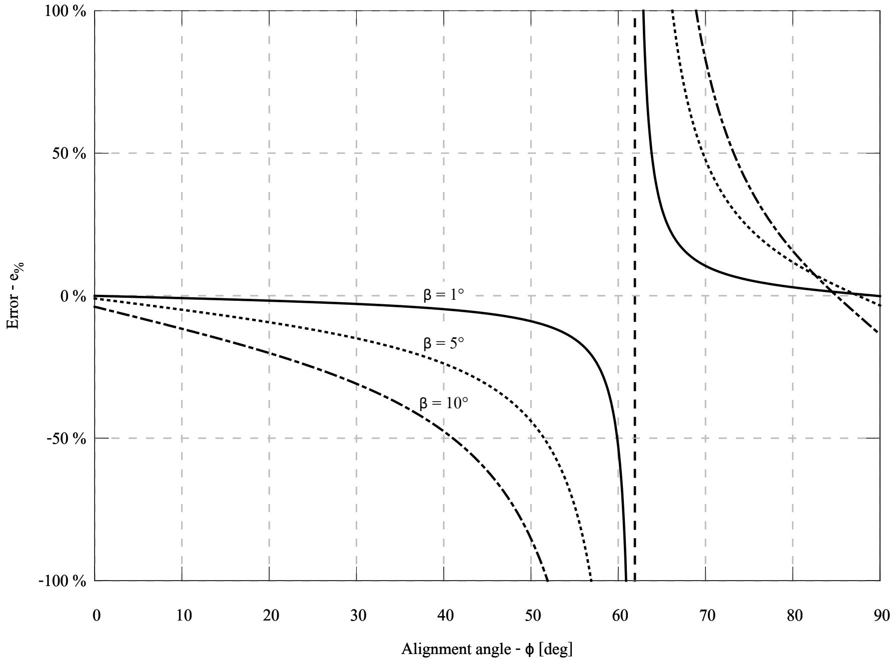


Figure 2.9: Errors due to misalignment of strain-gages

2.2.5 Temperature effects

The temperature is an important parameter that has to be taken into account when the strain is being measured using the strain-gages because the behavior of strain-gages is influenced by the differences between the calibration temperature and the working temperature. The effects are different and hereafter some of them will be explained. The main problem is that the electrical resistance of the the strain-gages material and the gage-factor are temperature dependent therefore they change when the temperature changes [14].

Usually the strain-gages are calibrated and built to work in specific temperature ranges because inside these ranges the unwanted effects due to temperature are negligible. Out of these ranges some approaches must be performed to avoid measurement errors.

Thermal output

Bonding a strain-gage to a mechanical part and connecting it to a strain indicator will show that, even without applying any strain, the strain-gage will show a value for the strain which is due to the changes of the temperature. This is apparent strain and must be taken into account during the tests. This apparent strain is called *thermal output* [14].

The source of the error is twofold: the first is that the resistivity of the material is temperature dependent and the second is the differential thermal expansion coefficient between the strain-gage and the material to which the strain-gage is bonded. The first contribution can be expressed by a linear dependence on the temperature variation respect to the calibration temperature:

$$\left(\frac{\Delta R}{R}\right)_{\beta} = \beta_R \cdot \Delta T \tag{2.25}$$

where β_R is the temperature coefficient of resistance of the grid conductor and $\Delta R/R$ is the variation of resistance due to the variation of temperature. The second contribution depends on the thermal expansion coefficients of strain-gage and testing material and can be expressed as follows:

$$\left(\frac{\Delta R}{R}\right)_{\alpha} = S_a \cdot (\epsilon_a + K_t \cdot \epsilon_t) \tag{2.26}$$

A variation of temperature of the material will cause a strain $\epsilon_a = \epsilon_t = \epsilon$ that is dependent on the temperature by the following:

$$\epsilon = (\alpha_s - \alpha_e) \cdot \Delta T \tag{2.27}$$

where α_s and α_e are respectively the thermal expansion coefficient of the strain-gage and of the test material. Adding all together the thermal output will be:

$$\left(\frac{\Delta R}{R}\right)_{T} = \left[\beta_R + S_g \cdot \frac{1 + K_t}{1 - \nu_0 \cdot K_t} (\alpha_s - \alpha_e)\right] \cdot \Delta T = \beta_e \cdot \Delta T \tag{2.28}$$

where β_e can be considered as the thermal coefficient of the entire system composed of the strain-gage and the testing part.

Compensating the error can be done in different ways. A simple approach is to use a dummy strain-gage that is subdue to the same thermal loading but not to any applied load to the test. A special configuration of the Wheatstone bridge can be used to have an electrical and automatic compensation, subtracting the thermal output measured by the dummy one (see Section 2.2.6) from the measured strain by the active strain-gage. By this way it is possible to obtain an output directly from the strain indicator which is just dependent no the real applied strain.

Another approach is to use self-temperature-compensated strain-gages [14]. These strain gages have made of some modified material which show negligible thermal output for a wide range of temperatures. The material of such strain-gages and the applied modification have to be according to the interesting material.

In order to identify the material the strain-gage is designed for, the manufacturers usually use a so called *S-T-C number*. It is common to use the thermal expansion coefficient expressed in ppm/°F. This S-T-C number is for example 06 for steel or cast iron, 09 for pure copper, 13 for aluminum alloys, etc.

Another way to correct the effect of the thermal output is to calculate it according to the measured temperature and then manually subtract it from the measured strain from the strain-gage. The manufacturers usually provide a thermal output graph (example of which is shown in Figure 2.10) together with a polynomial expression to facilitate the calculation of thermal

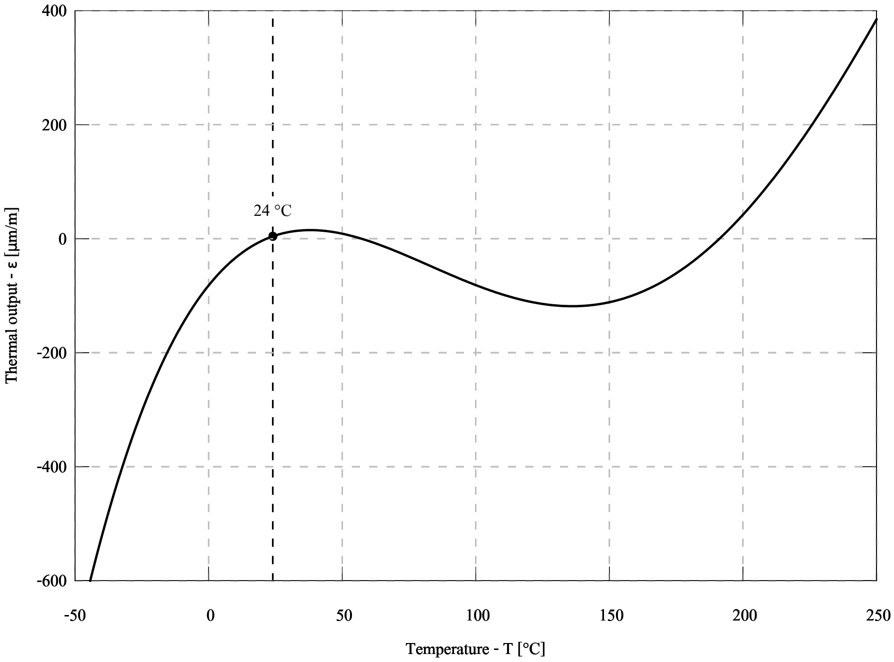


Figure 2.10: Thermal output of a self-temperature-compensated strain-gage *Vishay CEA-06-250UN-350*

output. The graph of Figure 2.10 is plotted using Equation (2.29).

$$\epsilon\left(\Delta T\right) = -81.9 + 5.82 \cdot \Delta T - 0.106 \cdot \Delta T^2 + 5.57 \cdot 10^{-4} \cdot \Delta T^3 - 7.85 \cdot 10^{-7} \cdot \Delta T^4 \tag{2.29}$$

where the temperature difference $\Delta T = T - 24°C$. 24°C is the base temperature and the resulting strain is in μm/m.

Influence of temperature on the gage-factor

The influence of the temperature on the gage-factor can be defined using a simple linear equation [14]:

$$S_g\left(\Delta T\right) = S_g \cdot \left(1 + \beta_{S_g} \cdot \Delta T\right) \tag{2.30}$$

where β_{S_g} is the temperature coefficient of the gage-factor, that usually is just 2% every 100°C.

The procedure to correct the measured strain to take into account this effect is very easy. Considering a measured strain of $\tilde{\epsilon}$ with a gage-factor of $\tilde{S}_g$ and with a temperature $T \neq 24°C$, the real strain will be:

$$\epsilon = \tilde{\epsilon} \cdot \frac{\tilde{S}_g}{S_g\left(T - 24\right)} \tag{2.31}$$

In order to correct simultaneously for both thermal output and gage-factor the following

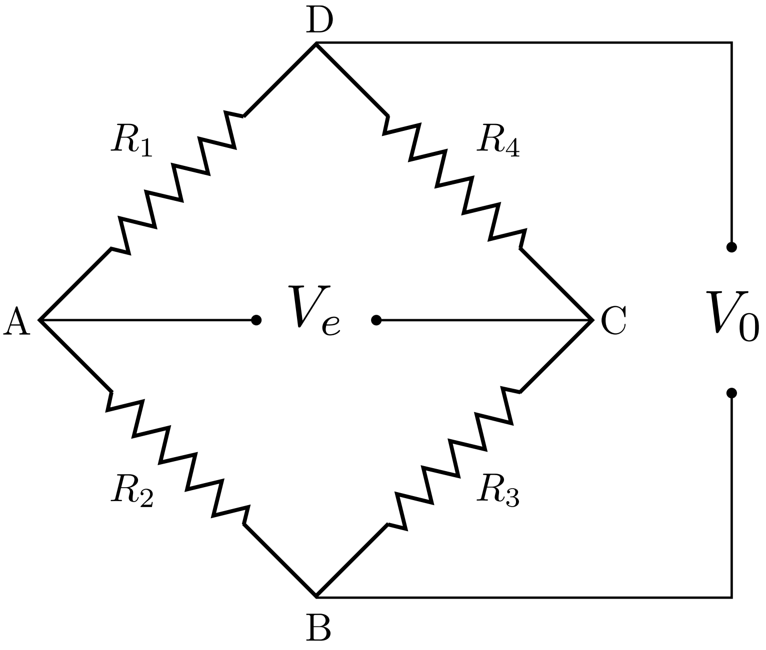


Figure 2.11: Scheme of the Wheatstone bridge

expression can be used:

$$\epsilon = [\tilde{\epsilon} - \epsilon\,(T - 24)] \cdot \frac{\tilde{S}_g}{S_g\,(T - 24)} \tag{2.32}$$

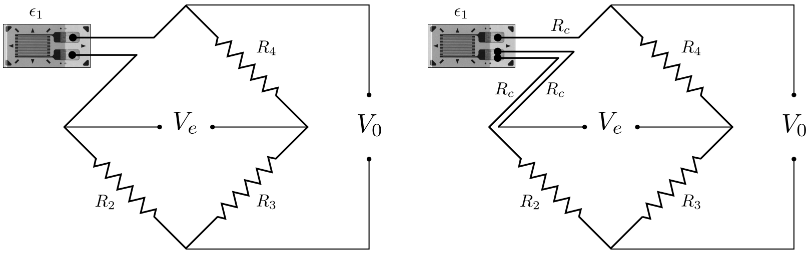
2.2.6 Wheatstone bridge

Strain-gage is an electrical resistance changing proportional to the strain. Joining the strain-gage to a device which is able to calculate and reveal the resistance changes is the interested point. This would be possible with using Wheatstone bridge. Wheatstone bridge is a circuit able to give an electrical output proportional to the changes of resistance and therefore in case of strain-gage proportional to the strain. So that, it gives zero as output when there is no applied strain. Wheatstone bridge is a measured circuit invented by Samuel Hunter Christie in 1833 and improved by Charles Wheatstone in 1843 [15]. It is used to measure an unknown electrical resistance by balancing the legs of the bridge, or as in this case, to identify the difference from a known value.

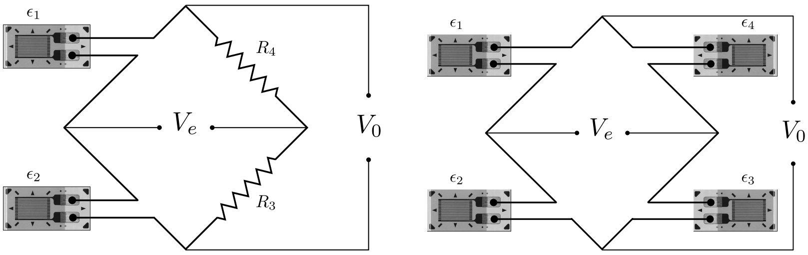
Solving the Wheatstone bridge, shown in Figure 2.11, means finding the electrical tension V_e that is due to the values of the resistances and to the electrical input V_0:

$$V_e = V_C - V_A = V_0 \cdot \left(\frac{R_3}{R_3 + R_4} - \frac{R_2}{R_1 + R_2} \right) \tag{2.33}$$

where all the four resistances have been considered constant. In this situation it is easy to prove that if $R_1 \cdot R_3 = R_2 \cdot R_4$ than the electrical output V_e is always zero. This special case is called



(a) Quarter-bridge (b) Three wires quarter-bridge



(c) Half-bridge (d) Full-bridge

Figure 2.12: Different configurations of the Wheatstone bridge

balanced bridge. For non balanced bridge circuits errors due to non-linearity may occur (refer to [16] for more details).

If a small variation to one or more resistances occurs the new value of the electrical output will be:

$$V_e = V_0 \cdot \left(\frac{R_3 + \Delta R_3}{R_3 + \Delta R_3 + R_4 + \Delta R_4} - \frac{R_2 + \Delta R_2}{R_1 + \Delta R_1 + R_2 + \Delta R_2} \right) \tag{2.34}$$

where expanding the expression and neglecting the contributions of second order in ΔR brings to:

$$V_e = V_0 \cdot \frac{R_1 \cdot R_2}{(R_1 + R_2)^2} \cdot \left(\frac{\Delta R_1}{R_1} - \frac{\Delta R_2}{R_2} + \frac{\Delta R_3}{R_3} - \frac{\Delta R_4}{R_4} \right) \tag{2.35}$$

In extensimetry it is common the case where the starting values of all the electrical resistances are equal to one unique value, dependent on the electrical resistance of the strain-gage (i.e. $R_1 = R_2 = R_3 = R_4 = R_0$). Introducing also the Equation (2.10) will lead to:

$$V_e = \frac{V_0}{4} \cdot S_g \cdot (\epsilon_1 - \epsilon_2 + \epsilon_3 - \epsilon_4) \tag{2.36}$$

where $\epsilon_i, i \in [1, 4]$ is the strain measured from a generic strain-gage in the i^{th} leg of the bridge.

Strain-gage can be placed instead of one, two or all four constant electrical resistance legs of a Wheatstone bridge, leading to quarter-bridge configuration, half-bridge configuration or full-bridge configuration, revealed in Figure 2.12.

Quarter-bridge

The quarter-bridge is the simplest and most commonly used configuration. Here one of the four resistances are substituted by a strain-gage as shown in Figure 2.12(a). If no strain is applied on the testing object and the bridge is balanced the output will be zero. If the bridge is set as a quarter-bridge configuration, when a strain is applied the output will be proportional to it and of the same sign. But this configuration has one problem. The resistances of the connection wires are unbalancing the bridge even where there is no applied strain, as the other legs of the bridge are set of having a constant resistance. For example 6 m of copper wires with a diameter 0.48 mm cause an output as generated by a strain of 5000 $\mu\epsilon$ if a strain-gage with gage-factor of 2.0 and electrical resistance of 120 Ω.

In order to solve this problem a different configuration has been proposed: the three wires quarter-bridge, Figure 2.12(b). In this situation the resistance of cables R_c is not important as it is present in both the upper and lower part of the bridge and their effects will be neutralized with each other. Of course all the cables must be of the same material and length.

The equation of the quarter-bridge is:

$$\frac{V_e}{V_0} = \frac{S_g}{4} \cdot \epsilon_1 \tag{2.37}$$

Half-bridge

The half-bridge is a common configuration when amplifying of the measured strain is the point of interest. The amplification is implemented to increase the sensibility and to reduce the noise-to-signal ratio. For example, in case of a cantilever beam, bonded from its two faces (upper and lower) to such configuration of bridge, the output strain will be two times of the real strain (the real strain would be detected on each face with opposite signs).

Another common application is the thermal compensation. This is easily possible by implementing a half- bridge configuration, connecting a strain-gage to the testing object and another strain-gage to a similar object (with the same shape and of the same material) subdue to the same temperature but not loaded. The difference between the two measured strain will be the real applied strain (as both strain-gages are measuring an equal apparent strain due to thermal effects).

The equation of the half-bridge is:

$$\frac{V_e}{V_0} = \frac{S_g}{4} \cdot (\epsilon_1 - \epsilon_2) \tag{2.38}$$

Full-bridge

The full-bridge, Figure 2.12(d), is the most complex to manage and for this reason is mostly used for transducers construction. The thermal output is compensated and usually the measured strain is amplified.

The equation of the full-bridge is:

$$\frac{V_e}{V_0} = \frac{S_g}{4} \cdot (\epsilon_1 - \epsilon_2 + \epsilon_3 - \epsilon_4) \tag{2.39}$$

2.2.7 Balance and shunt calibration

The Wheatstone bridge is a very good circuit to measure the variation of resistance of strain-gages but must be carefully set at the beginning of every test. In fact the peculiarities of the installation may change the actual value of the gage-factor (that must consider not only the strain-gage but also how this is connected to the measuring device) or it may happen that even without any applied strain the strain-gage is giving some not zero values because of the tolerance of the manufacturing process of making strain-gages.

All these contributions must be taken into account at every test and any source of error must be eliminated. As the electrical output of the Wheatstone bridge is proportional to the strain to be measured, the problem can be considered divided in two parts: scale factor and offset. The former is related to the gage-factor (see Equation (2.10)) while the second is related to the balance of the bridge. As they are not correlated they can be corrected separately.

To correct the output offset it is enough to introduce an electrical voltage equal to the value presented by the bridge when the strain-gage is not loaded. This electrical voltage has to be subtracted from the electrical output of the bridge.

To avoid errors due to a wrong scale factor a calibration must be used. In particular what usually it is performed is a *shunt calibration* [17]. In this kind of calibration an electrical resistance is connected parallel to the strain-gage in order to induce a resistance change in the bridge and to produce an electrical output of the bridge. This output depends on the strain-gage starting resistance and on the added resistance. This setting is applied to have a higher accuracy which is not achievable by adding a resistance in series with the strain-gage [17].

The electrical resistance variation due to an shunt resistance R_s will be:

$$\Delta R = \frac{R_s \cdot R_1}{R_s + R_1} - R_1 \tag{2.40}$$

where R_1 is the electrical resistance of the strain-gage unloaded. This new value of resistance produce an output as follows:

$$\epsilon_s = \frac{1}{S_g} \cdot \frac{-R_s}{R_s + R_1} \tag{2.41}$$

Normally the strain-gage manufacturers sell also strain indicators able to select the shunt resistance in order to induce an apparent strain known *a priori*. The procedure is then very simple. The user have to activate the shunt calibration and change the gage-factor till the value indicated by the strain indicator suggested by the manual. A common value is 5000 $\mu\epsilon$.

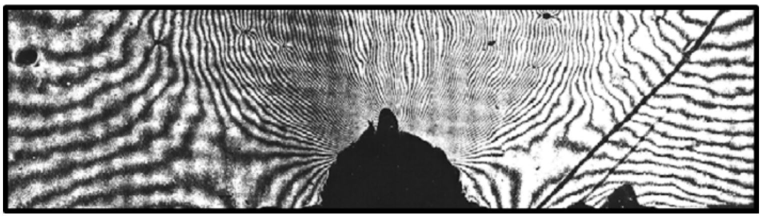


Figure 2.13: Example of moiré interferometry patterns [18]

2.3 Full-field techniques

All the techniques and the devices described up to now are based on the measurement of one unique value of strain. This value is obtained as an integral average over an area or a volume that can be very big as well as small. In any case it was possible to gain just one value. If more information is required, we must apply several devices (e.g. several strain-gages) with difficulties in the preparation of the specimen, sometimes even too delicate to accept invasive devices. Another big problem is that the acquisition will get complex when the number of devices will grow more than 4. In fact let's imagine 4 strain-gages (the simplest and smallest single-value device) over a thin polymeric specimen. First of all it can be difficult to locate accurately the strain-gages and then it can be difficult to acquire several channels, requiring a complex and expensive acquisition system with high speed. And all these efforts are made just for 4 values. If the load condition is quite complex or the material under investigation is highly anisotropic this approach is useless.

In order to have high number of data over an area or volume, a full-field approach must be studied. These kind of techniques allow the tester to get several local information at the same time, as hundreds or thousands strain-gages applied to the specimen. There are different kinds of full-field techniques. One of the most used is the *moiré interferometry*, Section 2.3.1, but it requires qualified technicians and advanced equipment. Other techniques like *phase shifting* and *temporal speckle pattern interferometry* are still based on the same principles and so quite complex and not affordable.

A breakthrough in the world of full-field techniques has been made by *digital image correlation*, Chapter 4. In the following paragraphs some information about full-field methods will be presented.

2.3.1 Moiré interferometry

Moiré interferometry combines the concepts and techniques of geometrical moiré and optical interferometry [19]. It is proved that all moiré phenomena can be treated as cases of optical interference, although some moiré phenomena can be treated also as mechanical interference. The moiré interferometry is capable of measuring displacement with high sensitivity (even thousands lines per millimeter), providing full-field patterns of high spatial resolution and excellent clarity.

The relation between fringe order and displacement occurred is based on the theory of *geometric moiré*, namely:

$$\begin{aligned} U(x,y) &= \frac{1}{f} N_x(x,y) \\ V(x,y) &= \frac{1}{f} N_y(x,y) \end{aligned} \tag{2.42}$$

where f is the frequency[3] of the virtual reference grating and N_x and N_y are the fringe orders[4].

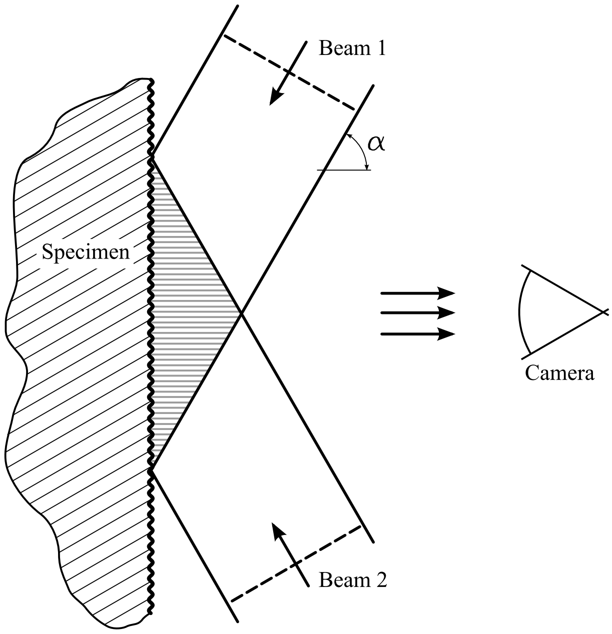


Figure 2.14: Scheme of moiré interferometry. Specimen and virtual reference grating

The virtual grating is created by the interference of two beams of coherent light that illuminate the specimen grating obliquely at angles of $+\alpha$ and $-\alpha$. The two beams generate walls of constructive and destructive interference in the zone of their intersection. As the specimen is exactly inside this zone, an array of closely spaced bright and dark lines will appear on the specimen surface. This artificial grating has a frequency dependent on the beam properties:

$$f = \frac{2}{\lambda} sin\alpha \tag{2.43}$$

where λ is the wave-length of the beams.

A very important characteristic has to be know before making any test, is the *sensitivity* of the technique. The moiré interferometry has a sensitivity equal to the number of fringes generated per unit of displacement, so equal to the frequency of the reference grating. This means that it

---

[3] Number of lines per unit of distance, or reciprocal of the pitch between adjacent lines

[4] Subtractive moiré fringe order, namely the moiré pattern representation by parametric curves, representative of the displacement occurred

can be easily few fringes/μmand so the displacement measured at each fringe order can reach even some tenth of micrometer.

The real limit is even bigger. From equation Equation (2.43) is obvious that the limit is reached when α is zero, and so the maximum frequency is $2/\lambda$. For example, with a coherent beam light with $\lambda = 488$ nm the limit is $f = 4098$ lines/mm. Weissman and Post [20] reached the 97.6% of this limit in 1982.

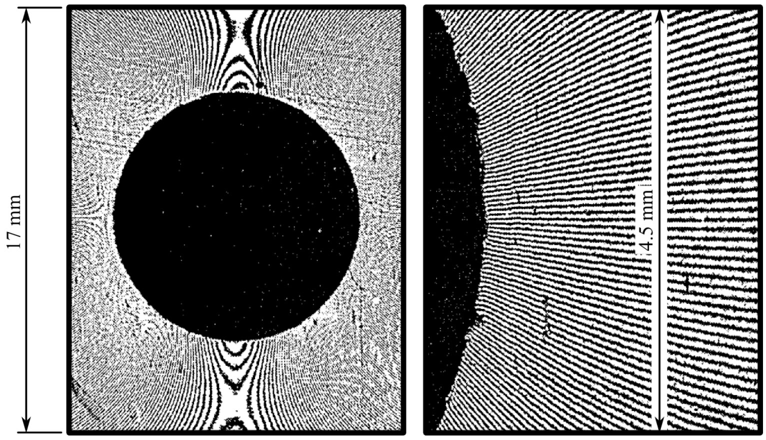


Figure 2.15: Moiré patterns near to the theoretical limit [20]

Another important parameter is the *displacement resolution*. It reflects the reliability of a displacement measurement [19]. Usually a visual interpolation of 1/5 or 1/10 of a fringe order is enough reliable. Then the displacement can be resolved to $1/(5 \cdot f)$ or $1/(10 \cdot f)$. Again it is easily possible to reach tenth or even hundreds of micrometer.

2.3.2 Digital image correlation

This technique, which will be explained completely in Chapter 4, has been introduced by Peters and Ranson [21] in 1982. However the appearance of faster computer in recent decade has made this technique developing faster and becoming one the most interesting multi-purpose techniques not only in research areas but also in applied material science and industries.

Chapter

3

Composite materials

TECHNOLOGICAL PROGRESS is associated with continuous improvement of existing material properties as well as with expansion of structural material classes and types. New materials provide new opportunities to develop updated structures and technology. Introduced in the middle of the 20th century, composite materials are the masterpieces of associating material, structural design and technology, presented by material science. Composite materials possess a great number of physical, mechanical and chemical properties. Generally speaking any material consisting of two or more components with different properties and distinct boundaries between the components can be referred as a composite material. We have taken the idea of composite materials from “mother nature”. Wood, shells, bones, muscles, any kind of tissue and natural fibers like silk, jute, sisal and cotton are all examples of natural composites.

Figure 3.1: Spider silk fibers composite [22]

Composite materials are generally composed of two phases [23]:

- **Matrix**: it is the continuous phase surrounding the reinforcement. This phase may be made of even metal, polymer or ceramic. The ductility is desired for this phase to some extent. The functions of the matrix are:

  - binding the reinforcement (fibers or particulates) together,
  - mechanically support the reinforcement,
  - transfer the load to the reinforcement,
  - protect the reinforcement from surface damage due to abrasion or chemical attacks.

- **Reinforcement**: it is the phase which is dispersed in matrix. This is the strong and stiff integral component of the composite that usually bears the majority of the applied stresses on composite (in filled material the reinforcement is replaced by filler particles. In this case, the filler is not considered as the component which has to bear the major stresses). Therefore the main role of reinforcement is the strengthening of the composite, and making it stiff and mechanical resistant.

The interaction between these two phases determines the properties to a significant extent. Bonding strength on the interface between the dispersed phase and matrix is the main interaction between matrix and reinforcement. Good bonding (adhesion) between matrix phase and dispersed phase provides transfer of load, applied to the material to the dispersed phase via the interface. Adhesion is necessary for achieving high level of mechanical properties of the composite. There are three forms of interface between the two phases:

1. direct bonding with no intermediate layer. In this case adhesion (“wetting”) is provided by either covalent bonding or Van der Waals forces,
2. intermediate layer (inter-phase) is in form of solid solution of the matrix and dispersed phases constituents,
3. intermediate layer is in form of a third bonding phase (adhesive).

Composite materials can be classified according to their structural components, structural design or matrix material. The classification according to structural components divides the composites in to two groups [24, 25]:

1. **Particulate composites or filled materials**: main feature of these materials is the existence of some basic or matrix material whose properties are improved by filling it with some particles. Usually the matrix volume fraction is more than 50% in such materials, and material properties, being naturally modified by the fillers, are governed mainly by the matrix. As a rule, filled materials can be treated as homogeneous and isotropic, i.e., traditional models of mechanics of materials developed for metals and other conventional materials can be used to describe their behavior.
2. **Reinforced composites or reinforced materials**: The basic components of these materials (sometimes referred to as “advanced composites”) are long and thin fibers possessing

high strength and stiffness. The fibers are bound with a matrix material whose volume fraction in a composite is usually less than 50%. The main properties of advanced composites due to which these materials find a wide application in engineering are governed by fibers. In this research we are mainly concentrated on this category of the composites respect to particulate composite.

3.1 Composite structure

The structural design of composites divides them in to three groups: particulate composites, fibrous composites and laminate composites.

3.1.1 Particulate composites

Particulate composites in turn can be divided into two sub-groups:

- composites with random orientation of particles (mainly refers to spherical particles),
- composites with preferred orientation of particles. Dispersed phase of these materials consists of two-dimensional flat platelets (flakes), laid parallel to each other

3.1.2 Fibrous Composites

Dispersed phase in form of fibers (fibrous composites) improves strength, stiffness and fracture toughness of the material, impeding crack growth in the directions normal to the fiber. Effect of the strength increase becomes much more significant when the fibers are arranged in a particular direction (preferred orientation) and a stress is applied along the same direction. The strengthening effect is higher in long-fiber (continuous-fiber) reinforced composites than in short-fiber (discontinuous-fiber) reinforced composites.

Short-fiber reinforced composites, consisting of a matrix reinforced with a dispersed phase in form of discontinuous fibers (length $< 100 \cdot$ diameter), has a limited ability to share load. Load, applied to a long-fiber reinforced composite, is carried mostly by the dispersed phase-fibers. Matrix in such materials serves only as a binder of the fibers, keeping them in a desired shape and protecting them from mechanical or chemical damages. The characteristics of fibrous composite materials are dictated by:

- volume fraction of the reinforcement,
- size of the reinforcement,
- shape of the reinforcement (particles, flakes, fibers, laminates),
- distribution of the reinforcement,
- orientation of the reinforcement (random or oriented).

3.1.3 Laminate composites

Laminate composites consist of layers with different anisotropic orientations or of a matrix reinforced with a dispersed phase in form of sheets. When a fiber reinforced composite consists of several layers with different fiber orientations, it is called multilayer (angle-ply) composite. Laminate composites provide increased mechanical strength in two directions and only in one direction, perpendicular to the preferred orientations of the fibers or sheet, mechanical properties of the material are low.

However the general classification of composites according to their structures is figured in Figure 3.2 [26]. Figure 3.3 shows some of the reinforcements used in reinforced composites.

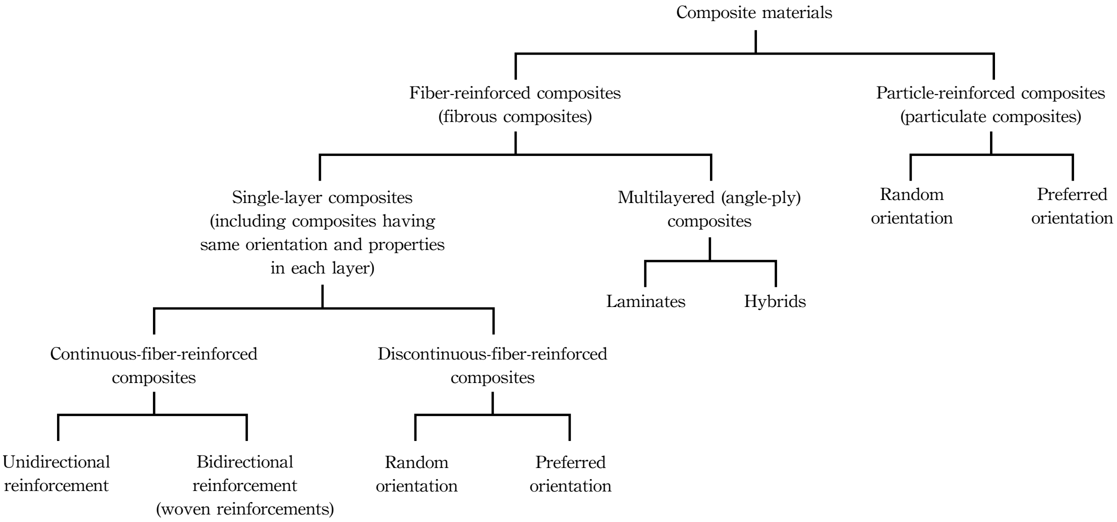


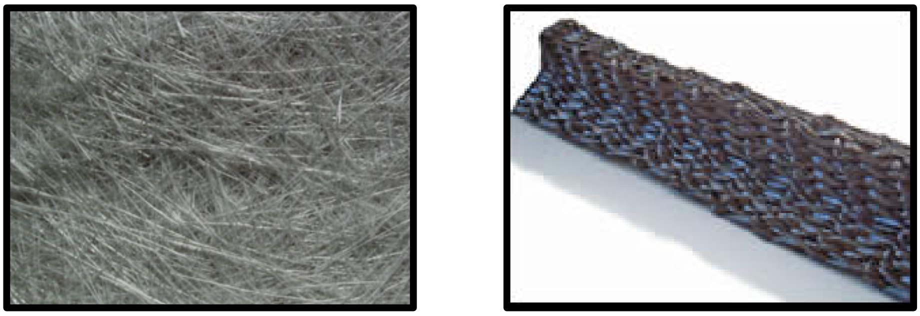
Figure 3.2: Classification of composite materials

3.2 Composite matrix material

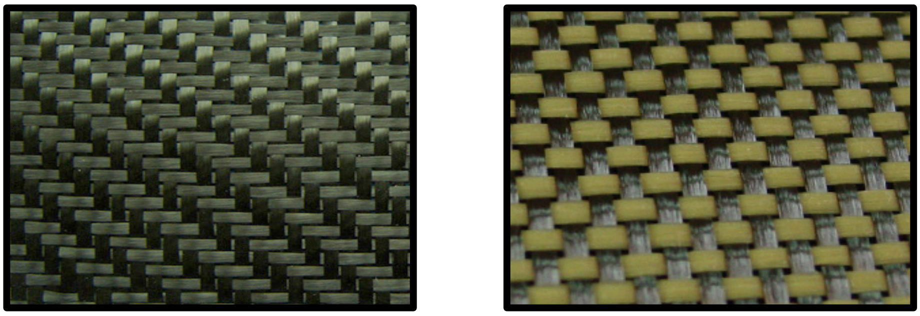
There are three classes of composites according to matrix material [25]: metal matrix composites, ceramic matrix composites and polymer matrix composites.

3.2.1 Metal matrix composites

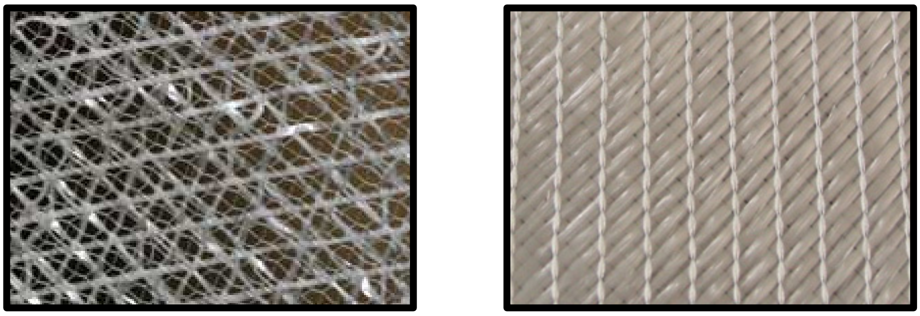
Metal matrix composites (MMC) are composed of a metallic matrix (aluminum, magnesium, iron, cobalt, copper) and a dispersed ceramic (oxides, carbides) or metallic (lead, tungsten, molybdenum) phase. Figure 3.4 shows three states of MMC for long-fibers, short-fibers and particulate MMCs (see `http://mmc-assess.tuwien.ac.at/mmc/cat/Glossary.html`).



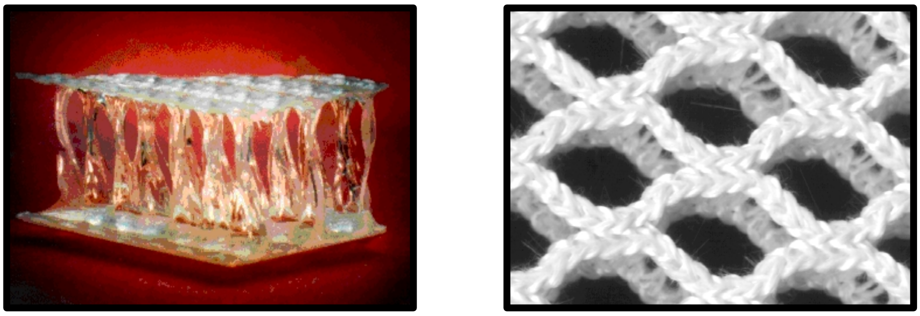
(a) Fiberglass chopped-strand mat (b) 3D braided carbon preform



(c) Carbon fiber fabric, twill weave 2x2 (d) Carbon fiber/kevlar fabric plain weave

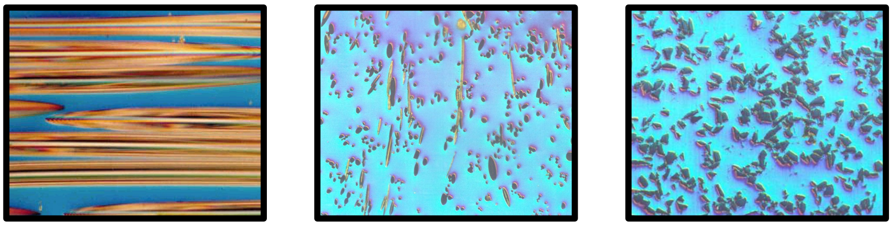


(e) Triaxial fiberglass fabric (f) Fiber glass fabric, biaxial 45°



(g) 3D woven sandwich composite (h) Open 3D knitted sandwich fabric

Figure 3.3: Reinforcements used in laminate composites



(a) Long-fibers MMC (b) Short-fibers MMC (c) Particulate MMC

Figure 3.4: Metal matrix composites (MMC)

3.2.2 Ceramic matrix composites

Ceramic matrix composites (CMC) are composed of a ceramic matrix and embedded fibers of other ceramic materials (dispersed phase). Figure 3.5 shows an example (see `http://www.ultramet.com/fiber_interface_scanning.html`).

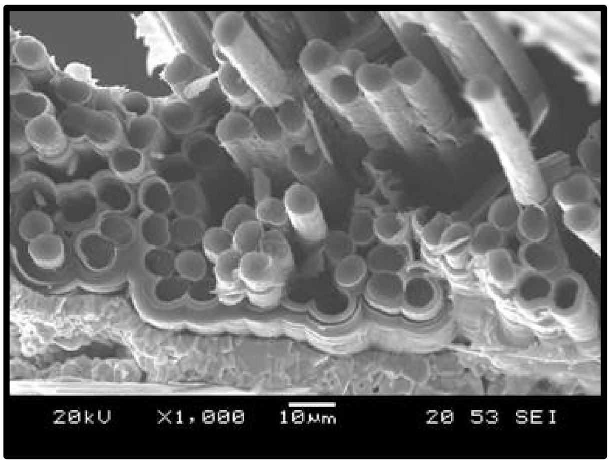


Figure 3.5: SEM image showing fiber pullout in ceramic matrix composite with zirconium oxide interface coating

3.2.3 Polymer matrix composites

Polymer matrix composites (PMC) are composed of a matrix from thermoset (unsaturated polyester – UP, epoxy – EP) or thermoplastic (polycarbonate – PC, polyvinylchloride, nylon, polysterene) and embedded glass, carbon, steel, kevlar fibers and many other fibrous material as dispersed phase.

Polymer matrix composites are very popular due to their low cost and simple fabrication methods. Use of non-reinforced polymers as structure materials is limited by low level of their mechanical properties: tensile strength of one of the strongest polymers–epoxy resin is 140 MPa. In addition to relatively low strength, polymer materials possess low impact resistance.

Reinforcement of polymers by strong fibrous network permits fabrication of polymer matrix composites characterized by the following properties [25]:

- high tensile strength,
- high stiffness,
- high fracture toughness,
- good abrasion resistance,
- good puncture resistance,
- good corrosion resistance,
- low cost.

Glass fibers, carbon fibers, aramid fibers are the main remarkable fibers used in PMCs revealing enormous high performance advantages to the composite material. However there are many other fibers used as reinforcement in PMCs. The fibers used in PMCs are briefly presented in the following paragraphs.

Glass fibers reinforced polymer composites (GFRP)

Continuous glass fibers (the first type of fibers used in advanced composites) are made by pulling molten glass (at a temperature about 1300°C) through $0.8 \div 3.0$ mm diameter dies and further high-speed stretching to a diameter of $3 \div 19$ μm. Usually glass fibers have solid circular cross sections. However there exist fibers with rectangular (square or plane), triangular and hexagonal cross sections, as well as hollow circular fibers. Typical mechanical characteristics and density of glass fibers are listed in Table 3.1.

Important properties of glass fibers as components of advanced composites for engineering applications are their high strength which is maintained in humid environments but degrades under high temperatures, relatively low stiffness (about 40% of the stiffness of steel), high chemical and biological resistance, and low cost. Being actually elements of monolithic glass, the fibers do not absorb water and change their dimensions in water. For the same reason they are brittle and sensitive to surface damage [24]. Figure 3.6 shows the novel application of GRPC for self-healing purposes. The advantages of glass fibers (fiber glass) are:

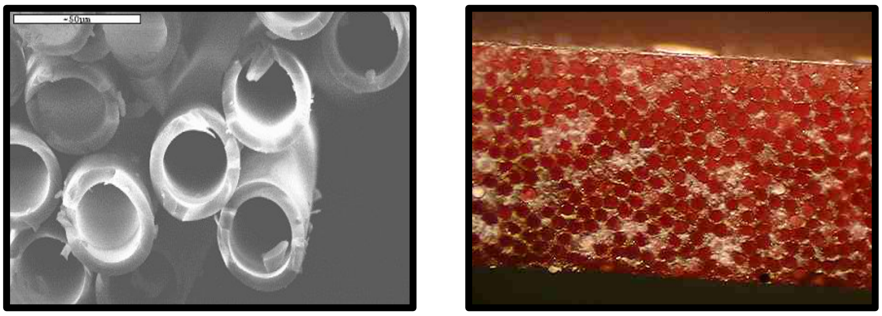
- easy to manufacture,
- chemical resistant,
- it can be surface protected by sizing post-processing with a thin layer of polymer,
- remarkable mechanical properties.

Table 3.1: Mechanical properties of structural materials and fibers

Material	Ultimate tensile stress [MPa]	Elastic modulus [GPa]	Specific gravity	Maximum specific strength [mm]	Maximum specific modulus [mm]
Metal alloys					
Steel	770-2200	180-210	7.8-7.85	28.8	2750
Aluminum	260-700	69-72	2.7-2.85	26.5	2670
Titanium	1000-1200	110	4.5	26.7	2440
Magnesium	260	40	1.8	14.4	2220
Beryllium	620	320	1.85	33.5	17300
Nickel	400-500	200	8.9	5.6	2250
Metal wires (diameter, μm)					
Steel (20-1500)	1500-4400	180-200	7.8	56.4	2560
Aluminum (150)	290	69	2.7	10.7	2550
Titanium (100-800)	1400-1500	120	4.5	33.3	2670
Beryllium (50-500)	1100-1450	240-310	1.8-1.85	80.5	17200
Tungsten (20-50)	3300-4000	410	19-19.3	21.1	2160
Molybdenum (25-250)	1800-2200	360	10.2	21.5	3500
Thermoset polymeric resins					
Epoxy	60-90	2.4-4.2	1.2-1.3	7.5	350
Polyester	30-70	2.8-3.8	1.2-1.35	5.8	310
Phenol-formaldehyde	40-70	7-11	1.2-1.3	5.8	910
Organosilicone	25-50	6.8-10	1.35-1.4	3.7	740
Polymide	55-110	3.2	1.3-1.43	8.5	240
Bismaleimide	80	4.2	1.2	6.7	350
Thermoplastic polymers					
Polyethylene	20-45	6-8.5	0.95	4.7	890
Polystyrene	35-45	30	1.05	4.3	2860
Teflon	15-35	3.5	2.3	1.5	150
Nylon	80	2.8	1.14	7	240
Polyester (PC)	60	2.5	1.32	4.5	190
Polysulfone (PSU)	70	2.7	1.24	5.6	220
Polyamide-imide (PAI)	90-190	2.8-4.4	1.42	13.4	360
Polyetheretherketone (PEEK)	90-100	3.1-3.8	1.3	7.7	300
Polyphenylenesulfide (PPS)	80	3.5	1.36	5.9	250
Synthetic fibers					
Capron	680-780	4.4	1.1	70	400
Dacron	390-880	4.9-15.7	1.4	60	1430
Teflon	340-440	2.9	2.3	190	130
Nitron	390-880	4.9-8.8	1.2	70	730
Polypropylene	730-930	4.4	0.9	100	480
Viscose	930	20	1.52	60	1300
Fibers for advanced composites (diameter, μm)					
Glass (3-19)	3100-5000	72-95	2.4-2.6	200	3960
Quartz (10)	6000	74	2.2	270	3360
Basalt (9-13)	3000-3500	90	2.7-3.0	130	3300
Aramid (12-15)	3500-5500	140-180	1.4-1.47	390	12800
Polyethylene (20-40)	2600-3300	120-170	0.97	310	17500
Carbon (5-11)					
High-strength	7000	300	1.75	400	17100
High-modulus	2700	850	1.78	150	47700

continue...

Material	Ultimate tensile stress [MPa]	Elastic modulus [GPa]	Specific gravity	Maximum specific strength [mm]	Maximum specific modulus [mm]
Boron (100-200)	2500-3700	390-420	2.5-2.6	150	16800
Alumina – Al_2O_3 (20-500)	2400-4100	470-530	3.96	100	13300
Silicon carbide – SiC (10-15)	2700	185	2.4-2.7	110	7700
Titanium carbide – TiC (280)	1500	450	4.9	30	9100
Boron carbide – B_4C (50)	2100-2500	480	2.5	100	10000
Boron nitride – BN (7)	1400	90	1.9	70	4700



(a) Hollow glass fibers (35 μm)

(b) Hollow glass fiber/epoxy laminate

Figure 3.6: Fibers used for self-healing composites

The same process that is used for glass fibers can be employed to manufacture mineral fibers, e.g., basalt fibers made of molten basalt rocks. Having relatively low strength and high density (see Table 3.1) basalt fibers are not used for high performance, e.g. aerospace structures, but are promising reinforcing elements for pre-stressed reinforced concrete structures in civil engineering.

Carbon fiber reinforced polymer (CFRP)

Carbon fibers are the stiffest and strongest reinforcing fibers for polymer composites, the most used after glass fibers. Made of pure carbon in form of graphite, they have low density and a negative coefficient of longitudinal thermal expansion. Carbon fibers are very expensive and can give galvanic corrosion in contact with metals. They are generally used together with epoxy, where high strength and stiffness are required, i.e. race cars, automotive and space applications, sport equipments. Carbon fibers are produced by the PAN or the pitch methods. The PAN method separates a chain of carbon atoms from polyacrylnitrile (PAN) through heating and oxidation.

For PAN-based fibers the process consists of three stages: stabilization, carbonization and graphitization. In the first step (stabilization) a system of polyacrylonitrile (PAN) filaments stretched and heated up to about 400°C in the oxidation furnace, while in the subsequent step (carbonization under 900°C in an inert gas media) most elements of the filaments other than carbon are removed or converted into carbon.

During the successive heat treatment at temperature reaching 2800°C (graphitization) crystalline carbon structure oriented along the fibers length is formed resulting in PAN-based carbon fibers. The same process is used for rayon organic filaments (instead of PAN), but results in carbon fibers with lower characteristics because rayon contains less carbon than PAN. For pitch-based carbon fibers, initial organic filaments are made in approximately the same manner as for glass fibers from molten petroleum or coal pitch and pass through carbonization and graphitization processes. Because pyrolysis is accompanied with a loss of material, carbon fibers have a porous structure and their specific gravity (about 1.8) is less than that of graphite (2.26). The properties of carbon fibers are affected with the crystallite size, crystalline orientation, porosity and purity of carbon structure. Figure 3.7 presents the complete PAN process of carbon fibers [24, 27].

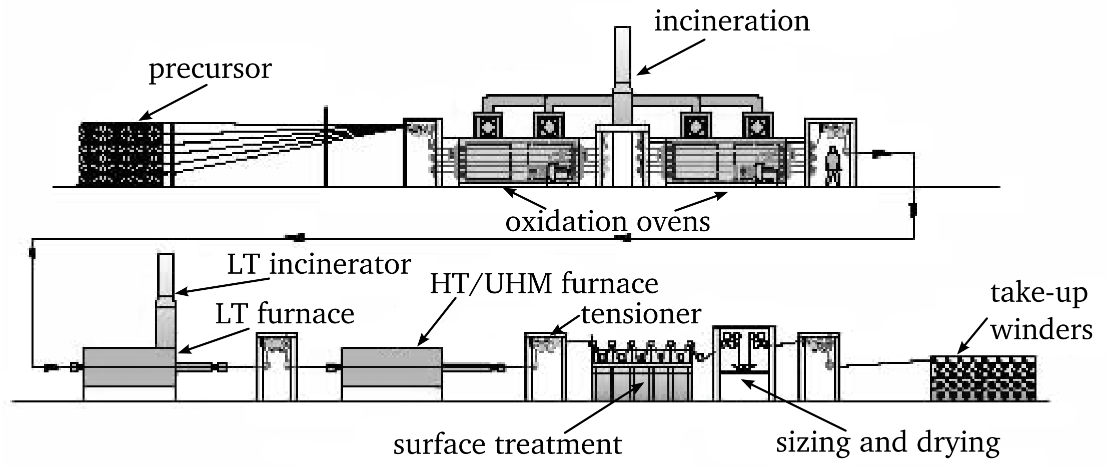
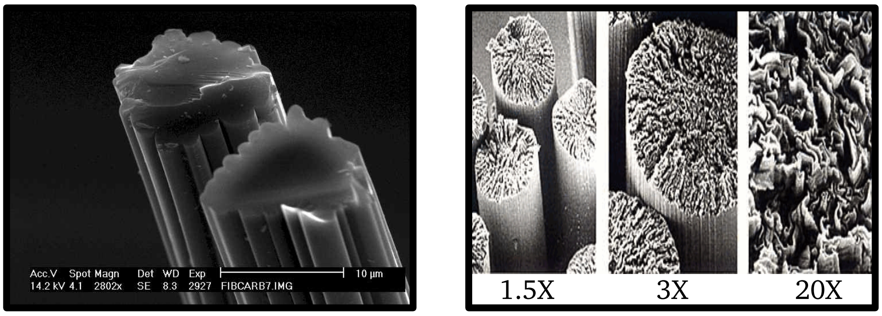


Figure 3.7: Complete PAN process of carbon fibers [27]

All commercial carbon fibers are manufactured with the same basic goal: to create an aligned graphitic structure and the mechanical, electrical and thermal properties associated with that structure. To this end, mesophase pitch (MP) production processes produce large aromatic molecules and concentrate them into mesophase (an ordered liquid crystalline form that will graphitize at high temperatures) with a viscosity suitable for spinning at temperatures below the point of decomposition. However based on precursor fiber materials, carbon fibers are classified into [28, 25]:

- PAN-based carbon fibers,
- pitch-based carbon fibers,
- mesophase pitch-based carbon fibers,
- isotropic pitch-based carbon fibers,



(a) PAN-based Carbon fiber (b) Pitch-based Carbon fiber

Figure 3.8: Cross section of carbon fiber

- rayon-based carbon fibers,
- gas-phase-grown carbon fibers,

Figure 3.8 presents cross section of PAN-based and pitch-based carbon fibers. Development of carbon (or graphite) fibers was a natural step aiming at a rise of fiber's stiffness the proper level of which was not exhibited by glass fibers. Modern high-modulus carbon fibers demonstrate modulus that is by the factor of about four higher than the modulus of steel, while the fiber density is by the same factor lower. Though first carbon fibers had lower strength than glass fibers, modern high strength fibers demonstrate tensile strength that is 40% higher than the strength of the best glass fibers, while the density of carbon fibers is 30% less. According to their properties, carbon fibers can be grouped into [24, 28]:

- ultra-high-modulus, type UHM (modulus $>$ 450 GPa),
- high-modulus, type HM (modulus between 350$\div$450 GPa),
- intermediate-modulus, type IM (modulus between 200$\div$350 GPa),
- low modulus and high-tensile, type HT (modulus $<$ 100 GPa, tensile strength $>$ 3.0 GPa),
- super high-tensile, type SHT (tensile strength $>$ 4.5 GPa).

Based on final heat treatment temperature, carbon fibers are classified into:

- **type I**: high-heat-treatment carbon fibers (HTT), where final heat treatment temperature should be above 2000°C and can be associated with high-modulus type fiber,
- **type II**: intermediate-heat-treatment carbon fibers (IHT), where final heat treatment temperature should be around or above 1500°C and can be associated with high-strength type fiber,
- **type III**: low-heat-treatment carbon fibers, where final heat treatment temperatures not greater than 1000°C. These are low modulus and low strength materials.

Table 3.2: Characteristics and applications of carbon fibers

Characteristics	Application
Physical strength, specific toughness, light weight	Aerospace, road and marine transport, sporting goods
High dimensional stability, low coefficient of thermal expansion, and low abrasion	Missiles, aircraft brakes, aerospace antenna and support structure, large telescopes, optical benches, waveguides for stable high-frequency (GHz) precision measurement frames
Good vibration damping, strength, and toughness	Audio equipment, loudspeakers for Hi-Fi equipment, pickup arms, robot arms
Electrical conductivity	Automobile hoods, novel tooling, casings and bases for electronic equipments, EMI and RF shielding, brushes
Biological inertness and x-ray permeability	Medical applications in prostheses, surgery and x-ray equipment, implants, tendon/ligament repair
Fatigue resistance, self-lubrication, high damping	Textile machinery, genera engineering
Chemical inertness, high corrosion resistance	Chemical industry, nuclear field, valves, seals, and pump components in process plants
Electromagnetic properties	Large generator retaining rings, radiological equipment

Table 3.2 provides the characteristics and applications of carbon fiber [28].

Boron fiber reinforced composites

Boron fibers were developed to increase the stiffness of composite materials while glass fibers were mainly used to reinforce composites of the day. Being followed by high-modulus carbon fibers with higher stiffness and lower cost, boron fibers have now rather limited applications. Boron fibers are manufactured by chemical vapor deposition (CVD) of boron trichloride (BCl_3) onto about 12 μmdiameter tungsten or carbon fiber (core). Because of this technology, boron fibers have relatively large diameter, $10\div200\mu$m. They are extremely brittle and sensitive to surface damage. Mechanical properties of boron fibers are presented in Table 3.1.

Aramid fiber reinforced composite

Various types of polymer fibers (nylon, rayon, polyester) have been in use for many years as reinforcement in automobile tires, large balloons and dirigibles, body armor and rubber coated fabrics. Organic fibers commonly encountered in textile applications can be employed as reinforcing elements of advanced composites. Naturally, only high performance fibers, i.e. fibers possessing high stiffness and strength, can be used for this purpose. The most widely used organic fibers that satisfy these requirements are known as aramid (aromatic polyamide) fibers. They are extruded from a liquid crystalline solution of the corresponding polymer in sulfuric acid with subsequent washing in a cold water bath and stretching under heating. Properties of typical aramid fibers are listed in Table 3.1. Kevlar, as one of the most important aramid fibers, was first introduced in 1971 by dry jet spinning method. Fiber properties can be alternated by using solvent additives, varying the spinning condition and using post-spinning heat treatment. Kevlar fibers possess unique properties. Their tensile strength and modulus are substantially higher while their elongation is significantly lower respect to other organic fibers. However kevlar fibers have poor characteristics in compression, results from their anisotropic structure, which permits rather easy local yielding, buckling and kicking of the fibers in compression. They are not as brittle as glass or graphite fibers and can be readily woven on conventional fabric loom [26, 24].

3.3 Fabrication of composites

3.3.1 Fabrication of thermosetting resin matrix composites

Thermosetting resin systems become hard by chemical reactions during curing and rather heating can not soften them. The hardening is irreversible. During curing they undergo a chemical change or reaction called cross-linked polymerization to form network polymers. Fabrication processes of thermosetting resin matrix composites can be broadly classified as wet forming processes and processes using premix or pre-preg. The coming paragraphs are different methods of Fabrication of thermosetting resin matrix composites [26].

Hand lay-up technique

Hand lay-up technique is the oldest, simplest and most used method for manufacturing of both small and large reinforced products. A flat surface, a cavity or a positive-shaped mold, made from wood, metal, plastic, reinforced plastic or a combination of this materials may be used. Fiber reinforcement and resin are placed manually against the mold surface. Thickness is controlled by the layer of materials placed against the mold. This technique is also called contact lay-up. The used materials are usually unsaturated polyester and epoxy resins in association with fibers (usually glass fiber, mats, fabrics or non-crimp fabrics). A chemical reaction initiated by a catalyst causes hardening to a finished part. Figure 3.9 shows a scheme of this procedure.

Spray-up

This is a partially automated form of hand lay-up. Chopped glass fibers and resin are simultaneously deposited on an open mold.

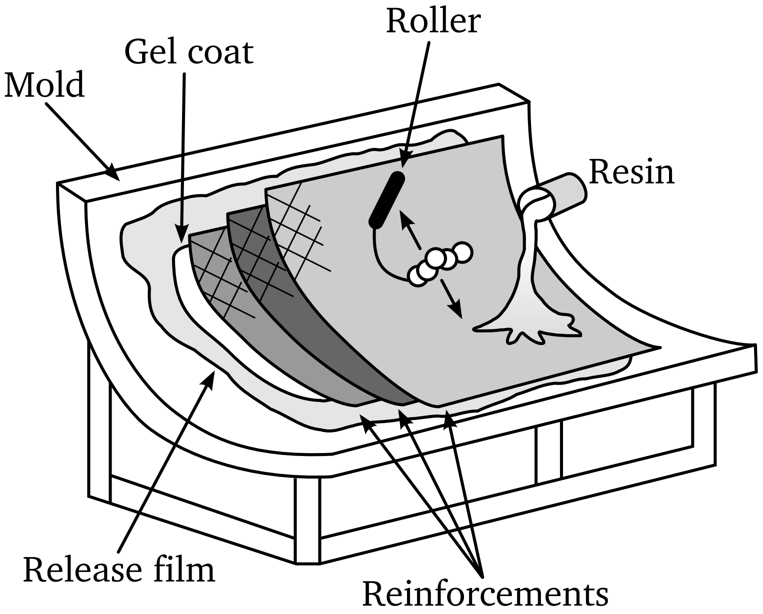


Figure 3.9: Hand lay-up technique

Bag molding process

Bag molding is one of the oldest and most versatile of the processes used in manufacturing composite parts. The laminate is laid up in a mold and resin is spread or coated, covered with a flexible diaphragm or bag and cured with heat and pressure. After the required curing cycle, the materials become an integrated molded part shaped to the desired configuration.

Filament winding

Filament winding is a technique used for the manufacturing of surface of revolution such as pipes, tubes, cylinders and spheres. This method is widely used for the construction of big tanks and pipe works for the chemical industries. In this method, continues reinforcement in the form of rovings are fed from a multiplicity of creels. The roving reinforcement goes from the creel to the resin bath and then perhaps gathered into a band of given width and wound over a rotating male mandrel.

Pultrusion

Pultrusion is an automated process for manufacturing composite materials into continuous and constant cross section profiles. In pultrusion, the product is pulled from the die rather than forced out by pressure. A large number of profiles such as rods, tubes and various structural shapes can be produced using appropriate dies. Pultrusion method generally consists of pulling continuous rovings and/or continuous glass mats through a resin bath or impregnator and then into performing fixtures where the section is partially shaped and excess resin and/or air are removed. Then it goes into a heat die where the section is cured continuously.

Preformed molding compounds

A large number of reinforced thermosetting resin products are made by matched die molding processes such as hot-press compression molding, injection molding and transfer molding. Matched die molding can be a wet process, but it is most convenient to use a preformed molding compound or premix to which all necessary ingredients have been added. This enables faster production rates to be achieved. Molding compounds can be divided into three broad categories:

- bulk or dough molding compounds (BMC, DMC),
- sheet molding compound (SMC),
- pre-preg.

3.3.2 Fabrication of thermoplastic resin matrix composites

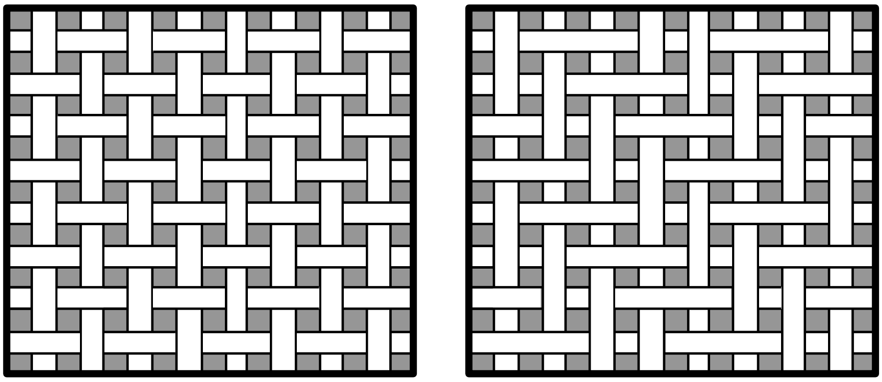
The principal method used for the production of parts with short fiber reinforced thermoplastics is injection molding. The choice of compounding technique depends on the requirements of fiber length, volume fraction, and degree of dispersion of fibers throughout the matrix. The two most common compounding methods are extruder compounding and strand coating.

3.4 Woven fabrics reinforcements

Weaves are generally referred and defined by a notation such as: 2x2, 4x4, and 3x1 for example. The first number in this set, for example the 3 in 3x1, refers to how many strands are crossed “over” before going “under” the perpendicular strands (in a 90° weave). The second number refers to how many strands are crossed “under” before going back “over” the perpendicular strands (again in a 90° weave). That is, a 3x1 weave would run: over, over, over, under, over, over, over, under, over, over, over, etc. A 1x1 weave would run: over, under, over, under, over, under, etc. [29].

3.4.1 Plain weave

A plain weave is defined as a 1x1 weave. In this most simple weave pattern, warp and fill yarns are interlaced over and under each other in alternating fashion. The plain weave provides good stability, porosity and the least yarn slippage for a given yarn count. For high strength panel and linear angle part fabrication plain pattern (see Figure 3.10(a)) is used when high strength parts are desired. It is ideal for construction, composites reinforcement, mold making, aircraft and auto parts tooling, marine and other composite lightweight applications.

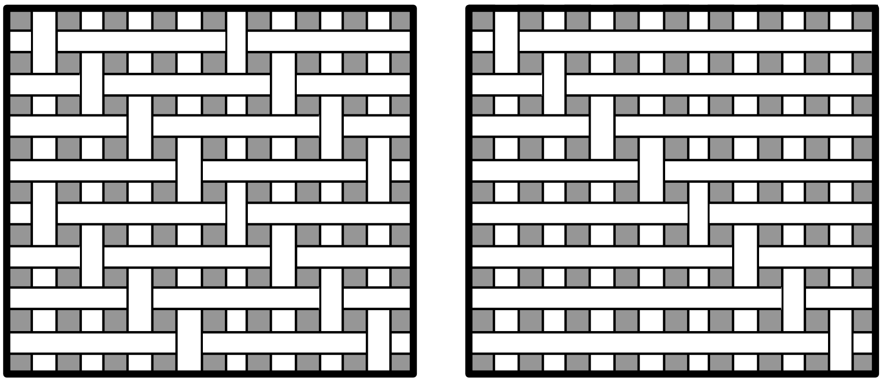


(a) Plain weave structure (b) Twill weave structure

Figure 3.10: Plain and twill weave structures

3.4.2 Twill weave

Twill weave is more pliable than the plain weave and has better drapability while maintaining more fabric stability than a four or eight harness satin weave. Twill weave also wets out better than plain weave. The weave pattern is characterized by a diagonal rib created by one warp yarn floating over at least two filling yarns. This enables a greater number of yarns per unit area than a plain weave, while losing little fabric stability. This type of fabric looks different on one side than on the other. A twill weave has that 45°, or 3D, "look" to it, which is so often desired. It is much easier to bend around complex curves than a plain weave, because its weave is looser. Therefore, a 4x4 twill will bend around curves better than a 2x2 twill weave. The looser the fabric, the more likely the fabric will fray at the ends and create spaces in the fabric when bent around complex curves. But a loose fabric will fit around complex curves much better than a tighter weave fabric [29].



(a) 4 harness satin weave (b) 8 harness satin weave

Figure 3.11: Satin weave structures

3.4.3 Satin Weave

This style fabric is one of the easiest fabrics to use and it is ideal for laying up cowls, fuselages, ducts and other contoured surfaces with minimal distortions. The fabric is more pliable and can comply with complex contours and spherical shapes. Because of its tight weave style, satin weaves are typically used as the surface ply for heavier and courser weaves. This technique helps reduce fabric print through and requires less gel coat to create a smoother surface. A harness-satin, sometimes referred to as: harness, satin or crowfoot, is defined as any number larger than 1, followed by x, and a larger number. That is, a 3x1 harness-satin is referred to as a harness-satin 4, H4, or 4HS. A 4x1 harness-satin is referred to as a harness-satin 5 (5HS or H5) and a 7x1 harness-satin is referred to as a harness-satin 8, H8, or 8HS. In the H4 weave picture below the weave runs: over, over, over, under, over, over, over, under, etc.

Four harness satin (crowfoot)

The four harness satin weave is more pliable than the plain weave and is easier to conform to curved surfaces typical in reinforced plastics. In this weave pattern there is a three by one

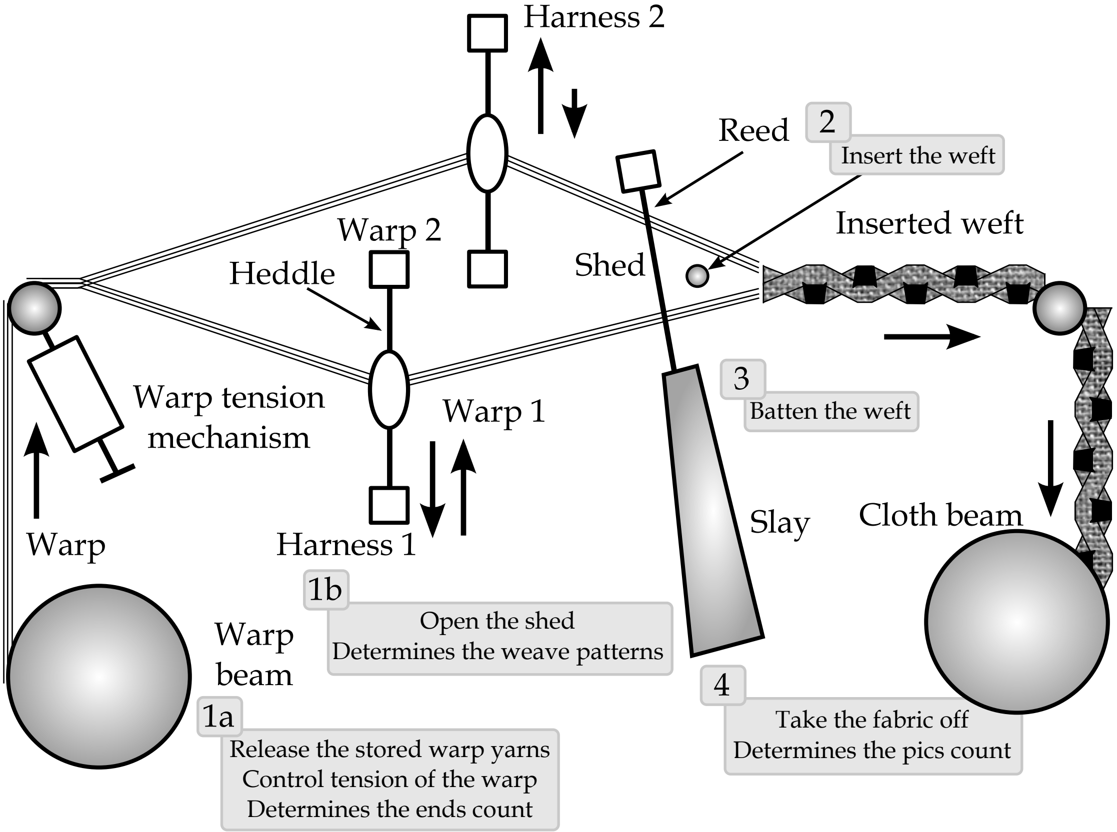


Figure 3.12: Schematic diagram of a weaving loom

interfacing where a filling yarn floats over three warp yarns and under one (see Figure 3.11(a)).

Eight harness satin

The eight harness satin (see Figure 3.11(b)) is similar to the four harness satin except that one filling yarn floats over seven warp yarns and under one. This is a very pliable weave and is used for forming over curved surfaces.

Which one for what?

A harness-satin bends over complex curves better than either a plain or twill weave. A harness-satin almost always has more weaves per inch than a plain or twill weave (defined as "pics"). So a 17pic x 17pic has more weaves per inch than a 12pic x 12pic (the difference between the weave definition, such as 2x2, with a pic definition, such as 12pic x 12pic should not be confused!). So, because a harness-satin has a higher pic count, it will hold together a bit better than a twill weave, when handled carefully.

The best 90° woven fabric for going around complex curves is a harness-satin 8 (8HS). A fabric that easy pivots, or can easily change fiber angle will also fit very complex curves. To allow a fabric to change fiber angle easily, it may be necessary to cut or remove the selvage. Most flat fabric weaves, such as plain weave, twills, harness-satins, etc. are woven on a machine called loom (see Figure 3.12). This machine weave at a 90° angle. That is, fiber going lengthwise and widthwise [29].

In general, if there are no complex curves and we do not care about aesthetics, a plain weave is the best option. If aesthetics are very important, generally a twill weave is selected, but for a

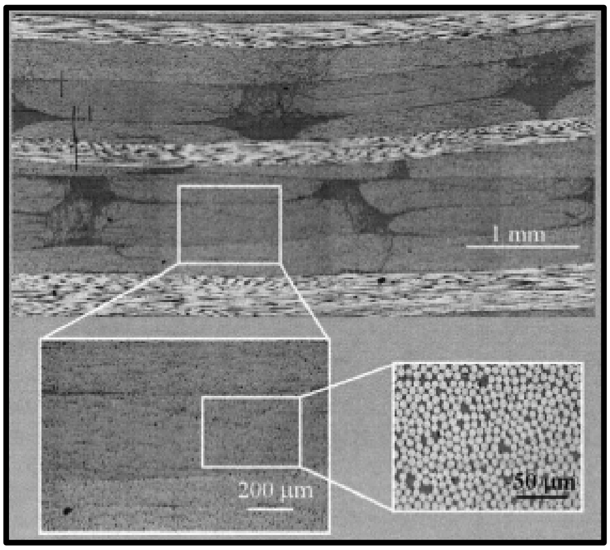


Figure 3.13: Hierarchical structure of the NCF composites

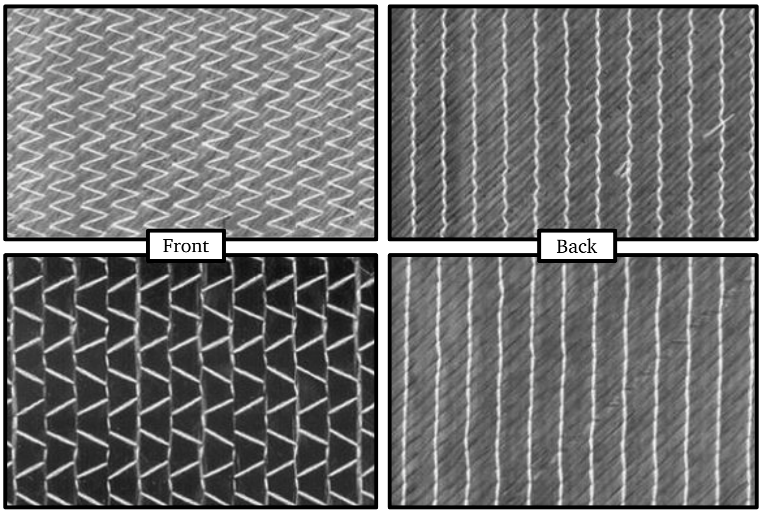


Figure 3.14: Two examples of carbon fibre non-crimp fabrics with different stitching patterns

sophisticated look a harness-stain H7 or H8 is often used. If there are very complex curves, an 8 harness-satin is the best choice.

3.5 Non-crimp fabrics (NCF)

Non-crimp fabric (NCF) composites due to their excellent performance and relatively low manufacturing costs have become an attractive alternative for aerospace and marine applications. They have reinforcement in form of blankets each of them consisting of several layers stitched together by stitching yarns (Figure 3.13). Each layer consists of straight unidirectional fiber bundles. NCF composites are produced using techniques like resin transfer molding (RTM), resin film infusion, etc. Additionally to substantial production cost reductions compared to pre-preg tape based materials also improvements in damage tolerance and out-of-plane fracture toughness of these materials have been reported [30, 31].

One of the features of the NCF composite is that it is an inherently multiscale material. On the micro-scale each bundle is a unidirectional (UD) fiber composite with a certain fiber content V_f and the homogenized bundle properties may be calculated using micro-mechanics expressions for long fiber composites. On the meso-scale bundle is considered as homogeneous transversely isotropic material surrounded by matrix and other bundles of the same or different orientation. A meso-scale characteristic of the composite is bundle content V_b in a layer. The geometrical shape of bundles (cross-section and axial alignment) is complex and depends on bundle orientation in the blanket, surface compression during production, resin pockets etc. The described meso-scale configuration determines the NCF composite properties on macro-scale and is typically used in simulations of macro behavior [31]. Some examples of non-crimp fabrics are shown in Figure 3.14. Non-crimp fabrics are built up of several layers of UD fibers, which are then stitched together, as shown in Figure 3.15. Because the different layers of UD fibers are held together by the stitching yarns, the fabric's handleability is much better than that of the pre-preg material. The yarns themselves stay straight, so there is virtually no crimp present. When using a material for applications like automobiles, it is very important to know how the material will behave under load conditions that are relevant for that application. For cars, apart from the stiffness, strength and strain of composite, properties like the fatigue and impact behavior are important factors.

It is necessary to characterize this behavior thorough mechanical testing and, simultaneously, to develop a fundamental understanding of the damage mechanisms involved. This will allow for a material model to be built, which will enable users to estimate the behavior of the material [32].

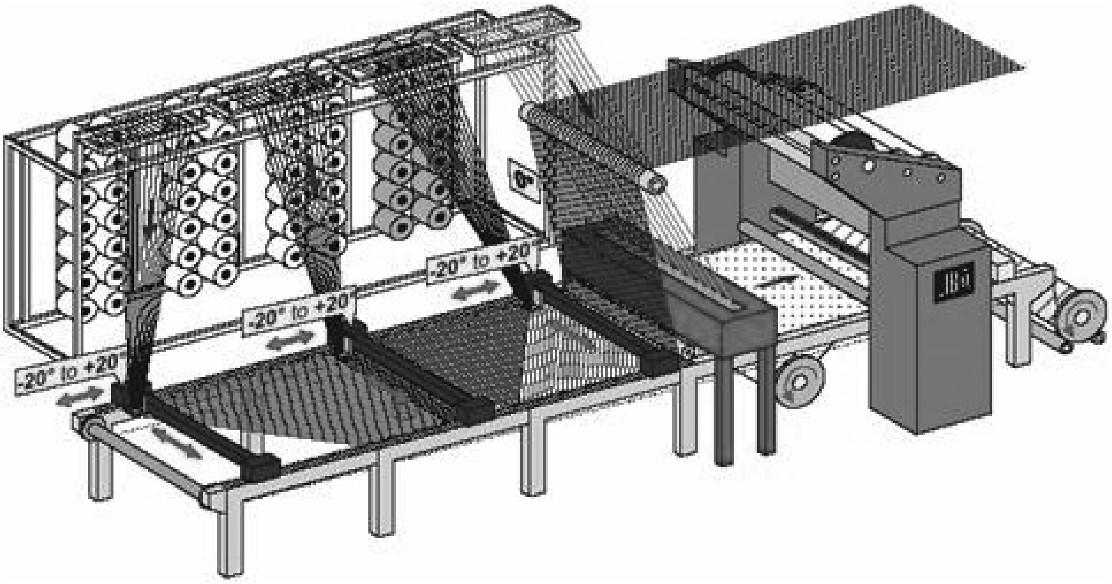


Figure 3.15: The LIBA-machine, a large stitching machine with many parallel needles, used for the production of non-crimp fabrics

Chapter

4

Digital Image Correlation

The digital image correlation (DIC) is a non-contact optical technique used to obtain full-field information of displacements and strains over an observed surface [21, 33]. There are two different approaches: two-dimensional approach and three-dimensional approach. The former is more used as it is less expensive and easier to implement, while the latter is more for special cases where the displacements cannot be considered lying on a plane (i.e. the out-of-plane movements are not negligible). The present Chapter will mainly discussed about the two-dimensional digital image correlation technique, apart from where specified.

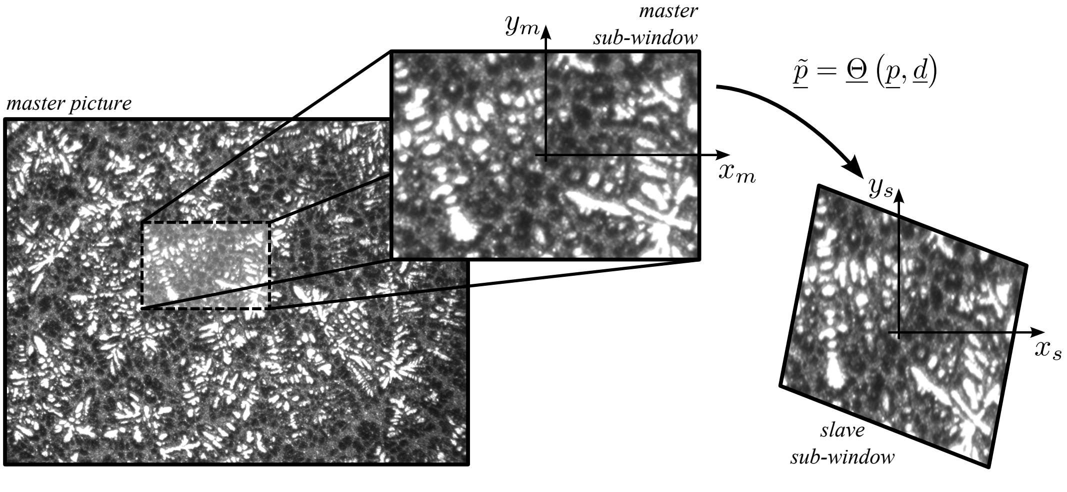


Figure 4.1: Basic scheme of the digital image correlation technique

The technique is based on optical recording of the specimen surface during the experiment. The image of not loaded specimen is considered as reference and it is called *master* image. And all the other images of the specimen when it is under load are called *slave* image. These are representatives of the deformed specimen. Subsequently the master image is divided in sub-windows (according to a user defined grid) and each sub-window is compared with a similar area in the slave image. Moving and deforming the master sub-window using a guessed displacement field allow a comparison between the master sub-window and the deformed slave sub-window (see Figure 4.1). Once the guessed displacement field is good enough, this is considered as the displacement field of the analyzed sub-window.

The correspondence of the master sub-window and the slave sub-window is calculated by a correlation coefficient (see Appendix A.2). This coefficient is indicating the similarity of the two sets of pixels. Digital image correlation algorithm (which is shown in Algorithm 4.1) is based on minimization of this correlation coefficient (see Appendix B.3). This method modifies the guess of the displacement field and leads to the real displacement that is the goal of the digital image correltion.

Algorithm 4.1 Digital Image Correlation Algorithm

```
Require: 1 master image and $N$ slave images
Ensure: $N$ displacement fields $\underline{u}^{(k)}, k \in [1, N]$

for $k = 1$ to $N$ do
    define a grid of $m \times n$ points
    for $i = 1$ to $m$ do
        for $j = 1$ to $n$ do
            define a sub-window centered in $\underline{p}_0^T \equiv (x_i, y_j)$
            $\underline{u}_{i,j}^{(k)} \leftarrow$ starting guess of the displacement field
            $C \leftarrow C\left(\text{master sub-window pixels, slave sub-window pixels}, \underline{u}_{i,j}^{(k)}\right)$
            minimize $C$ and get a better value for $\underline{u}_{i,j}^{(k)}$
        end for
    end for
end for
```

Considering a uniform and homogeneous material subdue to deformations, thanks to the continuum mechanics it is possible to express the displacement field $\underline{u}$ point-by-point as following:

$$\underline{u}\left(\underline{p}\right) = \underline{\tilde{p}} - \underline{p} \tag{4.1}$$

where $\underline{p}^T \equiv (x, y)$ is a generic point in the undeformed state, $\underline{\tilde{p}}^T \equiv (\tilde{x}, \tilde{y})$ represents the same point after deformation (so generally will be $\underline{\tilde{p}} \neq \underline{p}$) and $\underline{u}^T \equiv (u, v)$.

Considering this expression as the representative of the generic field inside one sub-window and choosing the center $\underline{p}_0$ of each sub-window as reference point, it is possible to evaluate the Taylor series of the displacement field inside the sub-window as following:

$$\underline{u}\left(\underline{p}\right) = \sum_{n \geq 0} \frac{\nabla^{(n)} \underline{u}\big|_{\underline{p}=\underline{p}_0}}{n!} \left(\underline{p} - \underline{p}_0\right)^n \tag{4.2}$$

where $\nabla^{(n)}$ is the derivative of order n. This expression can be cut after the first order:

$$\underline{u}\left(\underline{p}\right) = \underline{u}\left(\underline{p}_0\right) + \nabla\underline{u}\big|_{\underline{p}=\underline{p}_0} \cdot \left(\underline{p} - \underline{p}_0\right) + O\left(\left\|\underline{p} - \underline{p}_0\right\|^2\right) \tag{4.3}$$

$$\approx \underline{u}\left(\underline{p}_0\right) + \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix}_{\underline{p}=\underline{p}_0} \cdot \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} \tag{4.4}$$

where $\nabla\underline{u}$ is the Jacobian of the displacement vector.

Applying Equation (4.1) on Equation (4.4) leads to a more useful equation:

$$\underline{\tilde{p}} \approx \underline{p} + \underline{u}\left(\underline{p}_0\right) + \nabla\underline{u}\big|_{\underline{p}=\underline{p}_0} \cdot \left(\underline{p} - \underline{p}_0\right) \tag{4.5}$$

Equation (4.5) shows the relation between the points on the master sub-window (before deformation) and the coordinates of the same points on the slave sub-window (i.e. $\underline{\tilde{p}}$). By introducing a normalized sum of square differences between the gray-levels of the pixels in the master image and in the slave image it is possible to discriminate between a good guess and a bad guess of $\underline{u}$.

As the representation of $\underline{u}$ is only discrete, the calculation of its derivatives is not possible. Therefore they have to be in form of parameters too. Using again a first order linearization of the displacement field, Equation (4.5) will change into:

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} + \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix}_{\underline{p}=\underline{p}_0} \cdot \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} \tag{4.6}$$

that can be easily written as

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_0 \\ d_1 \end{bmatrix} + \begin{bmatrix} d_2 & d_4 \\ d_3 & d_5 \end{bmatrix} \cdot \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} \tag{4.7}$$

where $\underline{d}$ is the parameters vector. Finally it is possible to summarize all the equations defining a function $\underline{\Theta} : \mathbb{R}^2 \times \mathbb{R}^6 \to \mathbb{R}^2$ that is using $\underline{p}$ and $\underline{d}$ and it is giving $\underline{\tilde{p}}$ (see Figure 4.1).

Master image and slave image are both defined in a 256 gray-scale distribution. Function ϕ moves the pixels of master image over the pixels of slave images. When the color of master image pixel at point $\underline{p}$ and slave image pixel at point $\underline{\tilde{p}}$ are equal, the guessed displacement field is correct. In the other word, the comparison between the colors of master and slave images at each pixel gives the aforementioned correlation coefficient.

After a brief introduction to digital image correlation, this chapter will continue describing the development of a software for the strain measurements by DIC method. The algorithm of the present software, called *Phoenix*©, will be described in details here and some results will be outlined.

4.1 Literature review

One of the first scientific articles dealing with digital image correlation in the form known nowadays, is the article of Peters and Ranson in 1982 [21]. After that article many others followed trying to improve the mathematical approach and the numerical methods on the base of the technique, especially by the researchers of the University of South Carolina (Department of Mechanical Engineering) under the supervision of Sutton. In 1989 Brucks, McNeil, Sutton, and Peters presented a paper applying Newton-Raphson method to perform the minimization of the correlation coefficient [33]. This approach was a breakthrough respect to the past when a coarse-fine search algorithm was used. The coarse-fine search approach was performed calculating the correlation algorithm for several combinations of the parameters $\underline{d}$ describing the displacement field. If in a normal case at least 6 parameters are used, this approach consists of moving a point in a seven-dimensional space trying all the combinations of the six independent variables, in a range $\underline{d} \in [\underline{d}^{min}, \underline{d}^{max}]$ considering a step vector $\underline{d}^{step}$. The total number of evaluations of the correlation coefficient will be:

$$N = \prod_{i=1}^{6} \left\lceil \frac{d_i^{max} - d_i^{min}}{d_i^{step}} \right\rceil \tag{4.8}$$

that can be easily equal to 10^6. It means that if one single evaluation takes 1 ms all the evaluations of each sub-window will take more than 16 minutes. Considering all the sub-windows (maybe thousands) and all the slave images, the calculation time will grow very fast.

A Newton-Raphson approach is able to move towards the minimum of the seven-dimensional function of the correlation coefficient faster as it is using the gradient of the function in order to find the direction of the minimum. The only problem is that when the starting guess is not good, a relative minimum will be found instead of an absolute minimum. The same approach has been suggested by Vendroux and Knauss in 1998 [34] where an approximated but highly accurate Hessian matrix has been proposed in order to simplify the Newton-Raphson method. The authors say that thanks to this they got an improvement in terms of speed and convergence robustness of 25%.

But the evolution has not stopped since those years. In fact several approaches have been developed in parallel. In resent years a new coarse-fine approach has been proposed by Zhang et al. [35]. This new method is mainly a nested coarse-fine search algorithm, where if a precision of 0.01 pixel is needed for an area around the sub-window center of ±5 × ±5 pixels, it is not needed to perform one million evaluations of the correlation coefficient. The idea is to start looking for a precision of 1 pixel and so perform 100 evaluations. Once the best position (i.e. the coordinate with the highest correlation coefficient) has been found a second set of 100 evaluations will be performed around that point, with a 10 times smaller step (i.e. of 0.1 pixels). When the best position has been found another set of 100 evaluations will be performed with a step of 0.01 pixels. The approach is guaranteed to achieve the same precision of a normal coarse-fine search but with much less evaluations. For the same precision of the outlined example, $100 \cdot 3 = 300$

evaluations would be enough instead of 10^6.

A similar approach is based on the *peak-finding* approach. This approach is based on the evaluation of the correlation coefficient at the pixel level on the neighborhood of the sub-window center and fit these values by a continuos function using a least-squares method. Then the maximum of the function is found analytically. Wattrisse et al. [36] in 2001 used a quadratic polynomial function after the evaluation of an intercorrelation function ψ in a neighborhood of the sub-window center:

$$\underset{k,l\in[-RZ,RZ]}{\overline{\psi}(k,l)} = \frac{\sum\limits_{i,j=-CZ/2}^{CZ/2} I_1\left(i_0+i, j_0+j\right)\cdot I_2\left(i_0+i+k, j_0+j+l\right)}{\sqrt{\left[\sum\limits_{i,j=-CZ/2}^{CZ/2} I_1\left(i_0+i, j_0+j\right)^2\right]\cdot\left[\sum\limits_{i,j=-CZ/2}^{CZ/2} I_2\left(i_0+i+k, j_0+j+l\right)^2\right]}} \tag{4.9}$$

where CZ is the dimension of the correlation zone, RZ is the dimension of the research zone, I_1, I_2 are the pixel intensity (gray level) in the master and slave image respectively and (i_0, j_0) are the coordinates of the sub-window center.

A similar approach has been used by Hung and Voloshin [37] in 2003 using a least-squares fit of a quadratic function over the eight neighbor pixels around the center of the sub-window. The same approach has been used recently in 2007 by Hua et al. [38] who have used digital image correlation as strain measurement technique for a novel tensile machine for low-dimensional materials like films and fibers. Again Sun et al. [39] in 2007 presented a paper about a new approach based on a local coordinate system where the best values of the parameters were found by a peak finding method based on a bi-dimensional quadratic function.

Paepegem et al. [40] in 2009 presented an interested work where digital image correlation measured the notch sensitivity and crack bridging of window security film. In that paper they used a slightly different approach. To find the peak, they calculated the mass center of the signal peak:

$$(x_{cm}, y_{cm}) = \frac{\sum\limits_{x,y} C\left(x,y\right)\cdot\left(x,y\right)}{C\left(x,y\right)} \tag{4.10}$$

This approach reduce drastically the number of calculations respect to a complete least-squares fitting.

Wang et al. [41] presented recently (March 2009) a weighted function that is giving an importance to each value of the correlation coefficient in each pixel in a neighborhood of the sub-window center according to how far this pixel is from the center. By this function they developed a new correlation coefficient, starting from a common cross-correlation that it is unique when the master and slave sub-window match each other and otherwise it is less than one. Comparison with the traditional methods showed a better response respect to this technique.

A different approach has been proposed in 2001 by Zhou and Goodson [42]: the so called *spacial-gradient* algorithm. It is a least-squares approach for subpixel displacement estimation. The idea behind it is to start from the equations relating f with g as follows:

$$\begin{cases} f\left(\underline{p}\right) = g\left(\underline{p} + \underline{u}\right) \\ g\left(\underline{p}\right) = f\left(\underline{p} - \underline{u}\right) \end{cases} \tag{4.11}$$

and calculate their first-order Taylor series:

$$\begin{cases} f\left(\underline{p}\right) = g\left(\underline{p}\right) + \nabla g\left(\underline{p}\right) \cdot \underline{u} \\ g\left(\underline{p}\right) = f\left(\underline{p}\right) - \nabla f\left(\underline{p}\right) \cdot \underline{u} \end{cases} \tag{4.12}$$

Subtracting the first equation of the system 4.12 from the second leads to:

$$\left[\nabla f\left(\underline{p}\right) + \nabla g\left(\underline{p}\right)\right] \cdot \underline{u} = 2 \cdot \left[f\left(\underline{p}\right) - g\left(\underline{p}\right)\right] \tag{4.13}$$

Equation (4.13) can be solved in a least-squares meaning and the displacement $\underline{u}$ can be easily and quickly found.

Again in 2003 Zhang et al. [43] presented a work dealing with this method. They stated that intensity interpolation approaches are computationally too heavy while correlation coefficient interpolation approaches are very good and fast but dependent on the computation step. On the other hands differential iterative methods (like the one presented by Brucks et al. [33]) are able to find several parameters simultaneously but they are computationally complex because they need the calculation of second-order spatial derivations. Fourier approaches are fast but anyway need still some interpolation approach for subpixel precision and so they experience still the same problems. With all these considerations a least-squares approach is still faster and easier.

The main idea is a peak-finding-like approach divided in two step. A coarse peak finding method, performed as usual at pixel level. Then a subpixel approach done by defining an approximation of the function of the gray-scale levels of the pixels in the slave image $g\left(\underline{p}\right)$ by a Taylor expansion (usually stopped at the first or second order). Than the subpixel component of the displacement field is calculated by differentiating the correlation coefficient by the displacement vector $u \equiv (u, v)$ and setting it to zero:

$$\frac{\partial C}{\partial \underline{u}} = \frac{\partial}{\partial \underline{u}} \frac{\sum\limits_{\underline{p} \in S} \left[f\left(\underline{p}\right) - g\left(\underline{p}\right) - \nabla g\left(\underline{p}\right) \cdot \underline{u}\right]^2}{\sum\limits_{\underline{p} \in S} f^2\left(\underline{p}\right)} = 0 \tag{4.14}$$

where $\underline{p}$ is the generic point in the sub-window S and f, g are the master and slave image functions respectively.

The result of this expression is the following system of linear equations:

$$\begin{cases} u \cdot \sum\limits_{\underline{p} \in S} \left(\dfrac{\partial g\left(\underline{p}\right)}{\partial x} \right)^2 + v \cdot \sum\limits_{\underline{p} \in S} \dfrac{\partial g\left(\underline{p}\right)}{\partial x} \cdot \dfrac{\partial g\left(\underline{p}\right)}{\partial y} = \sum\limits_{\underline{p} \in S} \left[f\left(\underline{p}\right) - g\left(\underline{p}\right) \right] \cdot \dfrac{\partial g\left(\underline{p}\right)}{\partial x} \\ u \cdot \sum\limits_{\underline{p} \in S} \dfrac{\partial g\left(\underline{p}\right)}{\partial x} \cdot \dfrac{\partial g\left(\underline{p}\right)}{\partial y} + v \cdot \sum\limits_{\underline{p} \in S} \left(\dfrac{\partial g\left(\underline{p}\right)}{\partial y} \right)^2 = \sum\limits_{\underline{p} \in S} \left[f\left(\underline{p}\right) - g\left(\underline{p}\right) \right] \cdot \dfrac{\partial g\left(\underline{p}\right)}{\partial y} \end{cases} \tag{4.15}$$

In 2009 Pan et al. [44] proved that the least-squares method is usually faster and simpler but it is not taking into account the variations of intensity of the pixel colors because of the problems related to the acquisition device, such as:

- overexposure/underexposure,
- non-uniform or unstable illumination
- gain and offset of the imaging device in the digital images acquired at different time

Thus they proposed a *linear intensity change model*. The approach adds a linear model to the function f in order to cope with the aforementioned problems. The resulting equation is as follows:

$$a \cdot f\left(\underline{p}\right) + b = g\left(\underline{\tilde{p}}\right) \tag{4.16}$$

where a and b are two optical parameters (i.e. the scale and offset parameter respectively). This equation is differentiated and solved by iterative least-squares approach considering also a bi-cubic spline subpixel interpolation function. Finally a *pointwise least-squares strain estimator* technique was proposed and implemented to reduce the noise-to-signal ratio. The method is based on the fitting of the displacement values of the points in a neighborhood of a point by a continuous function and then evaluating analytically the derivatives of the continuous function instead of the discrete differentiation over the raw displacement data.

Other approaches try to find more innovative ways to get the displacement field. Pitter et al. [45] in 2001 tried with a software implementing neural networks to analyze microscopic deformations. Genetic algorithm is another used technique. Genetic algorithm is a global search heuristic method to approximate solution of minimization problems. It is based on the evolutionary theory of Charles Darwin ("the origin of species", 1859) about the evolution of species. The idea is to define a set of parameters (genes) of a problem and evaluate sevral solutions based on different values of these parameters. The best solutions will generate offspring and their genes will survive and evolve. After many generations weak entries will evolve in very strong ones [46].

Pilch et al. [47] in 2004 presented a work where a genetic algorithm made of chromosomes of six genes (two displacements and four derivatives of the displacements) and the cross-correlation coefficient is used as objective function. Preliminary results with a bilinear subpixel interpolation function indicate that 1/1000 pixel accuracy can be achieved. Each new generation is made by crossing and blending of the parents and by mutation. Some examples with an initial population of 200 chromosomes and a mutation rate of 5% converged in about 34 iterations with a percentage of error about 2%. An hybrid genetic algorithm implementing crossover, mutation, adaptive mechanisms to improve the hill-climbing ability of optimum solutions, elite strategy and roulette criteria selection has been presented by Hwang et al. [48] in 2008 to measure the strain during the micro tensile testing of SU-8 photoresist.

In all these approaches, the solution path that is the path that the algorithm is following sub-window by sub-window analyzing all the sub-windows defined in the master image is important. Usually starting from the first sub-window, called seed, the neighbors are solved using as starting guess the solution of the close sub-window according to the continuos deformation assumption. This can speed up the calculations (as it can help the iterative methods to start from a value close to the solution) but it will also spread potential errors, maybe due to discontinuities, crack, bad illuminated areas, etc. In order to solve this problem a *reliability-guided digital image correlation* (RGDIC) method has been proposed by Pan in 2009 [49]. His method builds a map of correlation coefficients on the analysis grid and it uses this map to create a path along which solve the entire grid, following the points with highest correlation coefficient value. In this way the points with low correlation will be analyzed as last. In Figure 4.2(a) there is an example developed using *Phoenix*© where it is possible to see the growing path of the analyzed area. The Figure 4.2(b) shows very well the percentage of improvement of the RGDIC respect to traditional approaches. Apart from a small area with no difference, the differences may easily reach up to 80%.

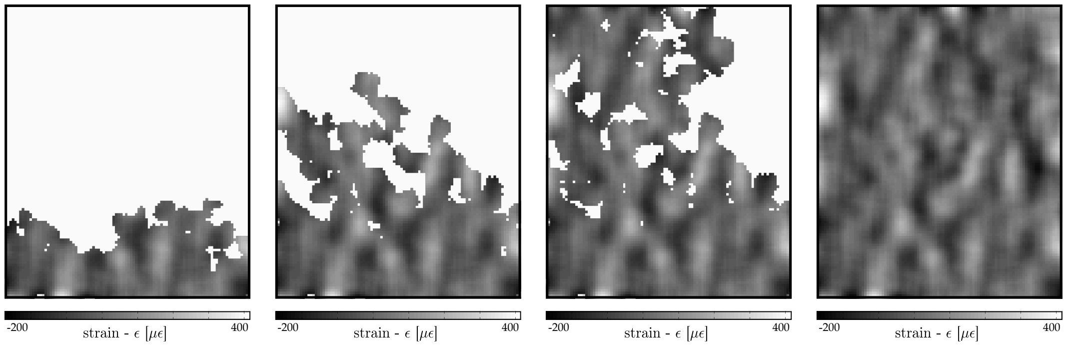
In most of the mentioned studies a first-order displacement field has been implemented. This is mainly because of simplicity and to reduce the computational cost of the analysis. A zero-order displacement field means a pure translation that it would be not interesting in many cases and a second-order displacement field may introduce too many parameters to found in the minimization algorithm. Here some examples of displacement fields:

$$\begin{cases} u^{(0)} = u_0 \\ v^{(0)} = v_0 \end{cases} \tag{4.17}$$

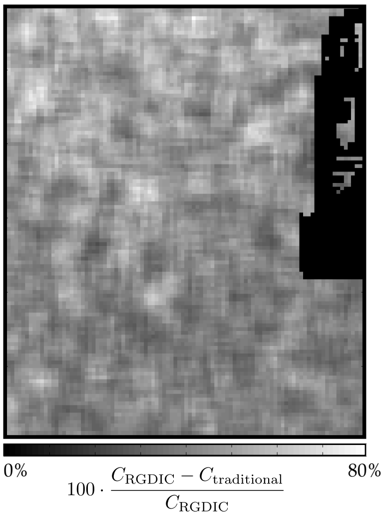
$$\begin{cases} u^{(1)} = u_0 + \dfrac{\partial u}{\partial x} \cdot (x - x_0) + \dfrac{\partial u}{\partial y} \cdot (y - y_0) \\ v^{(1)} = v_0 + \dfrac{\partial v}{\partial x} \cdot (x - x_0) + \dfrac{\partial v}{\partial y} \cdot (y - y_0) \end{cases} \tag{4.18}$$

$$\begin{cases} u^{(2)} = u_0 + \dfrac{\partial u}{\partial x} \cdot (x - x_0) + \dfrac{\partial u}{\partial y} \cdot (y - y_0) + \dfrac{\partial^2 u}{\partial x^2} \cdot (x - x_0)^2 \\ \qquad + \dfrac{\partial^2 u}{\partial y^2} \cdot (y - y_0)^2 + \dfrac{\partial^2 u}{\partial x \partial y} \cdot (x - x_0) \cdot (y - y_0) \\ v^{(2)} = v_0 + \dfrac{\partial v}{\partial x} \cdot (x - x_0) + \dfrac{\partial v}{\partial y} \cdot (y - y_0) + \dfrac{\partial^2 v}{\partial x^2} \cdot (x - x_0)^2 \\ \qquad + \dfrac{\partial^2 v}{\partial y^2} \cdot (y - y_0)^2 + \dfrac{\partial^2 v}{\partial x \partial y} \cdot (x - x_0) \cdot (y - y_0) \end{cases} \tag{4.19}$$

Many authors tried also a second-order shape function (as in Equation (4.19)) but they



(a) Example of RGDIC results. From left to right the analysis is covering all the area



(b) Improvement due to the reliability-guided approach

Figure 4.2: Results from a reliability-guided digital image analysis

achieved different results. Lu and Cary [50] in 2000 stated that accounting for higher order gradients allows a more accurate determination of the deformation parameters directly related to the strain components and so the first-order gradients can be measured more precisely.

Schreier and Sutton [51] in 2002 showed how this stating is wrong as it starts from an incorrect hypothesis because *the shape functions must be capable of closely approximating the displacement field encoded in the images* (constraint four in [51]). This assumption suggests that the smallest sub-window size should be chosen. It is also true that a big sub-window gives less standard deviation because of less noise-to-data ratio. Thus a trade-off must be used: the smallest sub-window considering the lowest noise effect must be used. Moreover a second-order shape function matches a generic displacement field better than a first-order shape function, but it is causing higher standard deviation. Thus generally a second-order shape function requires a bigger sub-window size. It is also true that a second-order displacement field is not so commonly applied in usual situations. Another important point is the performance of the method. An iterative cross-correlation algorithm with first-order shape function can perform correlations per second at least twice respect to an algorithm with second-order shape function[1] [51].

Once the displacement field has been found, the strain field must be calculated. It can be defined starting from the displacement field and by discrete differentiation, but it is also true that in some cases (mainly in Newton-Raphson like approaches) the strains are already counted inside the displacement field equation (see Equation (4.4)). In fact the method needs the information of different derivatives of the displacement field, but the problem is that these information is full of noise and it is not precise [33]. An easy approach to encounter with this problem is discrete differentiation of the displacement field as described better in Appendix C.3. In this case the value of strain in one point is just equal to the difference of the values of displacement between the subsequent and the previous point divided by their distance:

$$\left.\frac{\partial u}{\partial x}\right|_i = \frac{u_{i+1} - u_{i-1}}{x_{i+1} - x_{i-1}} \tag{4.20}$$

It is obvious that this method is as noisy as simple. In fact even small variations in displacement may lead to big irregularities in strain as shown in Figure 4.3.

A common approach is then to analyze each sub-window one by one and evaluate a least-squares approximation of the values of displacement around the studied point and so that define a continuous function instead of a cloud of points. This approach allows easily to calculate analytically the value of strain (differentiation the displacement). Common methods are based on moving least-squares [52] or Savitzky-Golay filter [53, 54]. The former is based on a least-squares regression scheme executed over a moving area, defined around the studied point. The latter is a similar approach presented in 1964 by Abraham Savitzky and Marcel J. E. Golay where a least-squares fit is performed considering that in a local coordinate system (centered in the studied point) the equation of the fitting surface is equal to the constant term. In a simple one-dimensional case this is expressed by:

$$f(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + \cdots + a_n \cdot x^n \tag{4.21}$$

where at the origin of the local coordinate system $f(x) = f(0) = a_0$. At the same way the first derivative will be $\frac{\partial f(0)}{\partial x} = a_1$. The second derivative will be $\frac{\partial^2 f(0)}{\partial x^2} = 2 \cdot a_2$. And so on for the next derivatives. Considering the least-squares method, described in Appendix D.2, the best fit can be found using Equation (D.5), where $\underline{\underline{G}} = \left(\underline{\underline{X}}^T \cdot \underline{\underline{X}}\right)^{-1} \cdot \underline{\underline{X}}^T$ is a constant matrix once the polynomial is decided and meanwhile a local coordinate system on the center of the sub-window is assigned and a least-squares fit over $(2 \cdot m + 1) \times (2 \cdot m + 1)\,, m > 0$ sub-windows is calculated. The fact that just the coefficient a_0 is needed, simplifies again the calculations as it can be easily

---

[1] Test executed on a Intel Pentium III processor with a clock-speed of 700 MHz

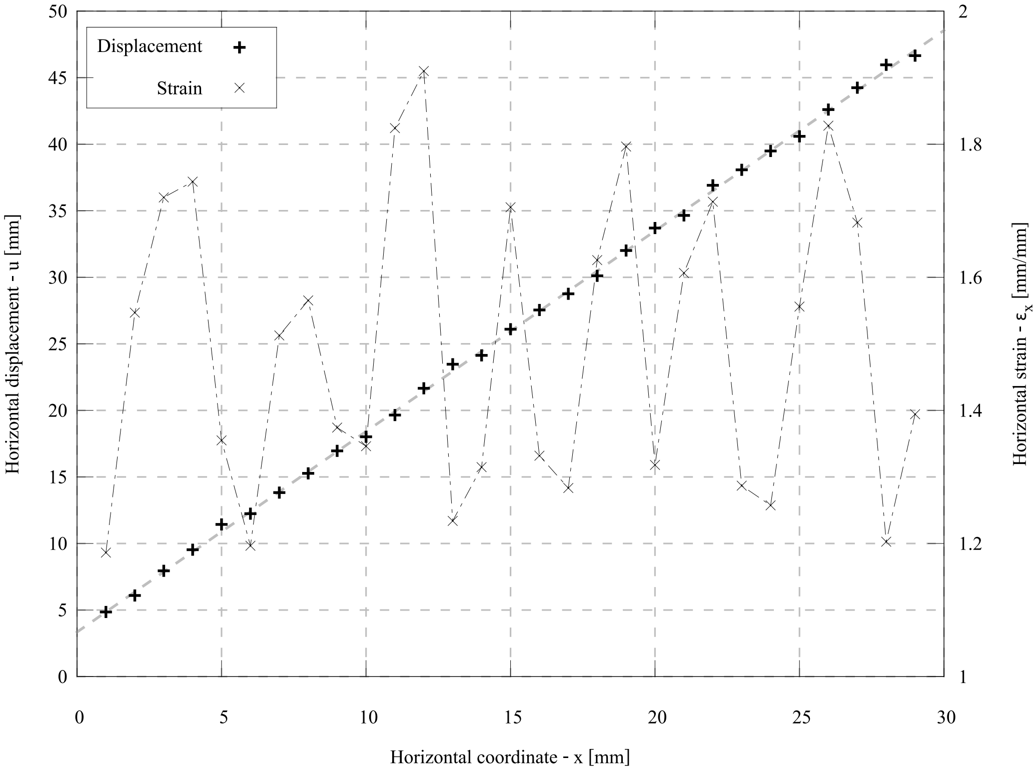


Figure 4.3: Example of discrete differentiation of a linear displacement field

found by the following:

$$a_0 = \left\langle \underline{g}_1, \underline{y} \right\rangle \tag{4.22}$$

where $\underline{y}$ is the vector of raw data and $\underline{g}_1$ is the first row of the matrix $\underline{\underline{G}}$. Tables with the entries of $\underline{g}_1$ for different polynomials and with different number of points can be found in reference [53].

This smoothing procedure, as the more general least-squares method, are point-wise methods as they don't take into account a more general relation between points. Approaches similar to finite elements method can consider a continuity in displacement or strain fields and define a net between the sub-windows center in order to induce a smoothing in the fields. In fact these approaches satisfy the requirements of displacement continuity and derivative continuity on images [55–57]. For example Sun et al. [55] describes the displacement field along the x axis as:

$$\begin{cases} u\left(x, y, \underline{d}\right) = \sum_{i=1}^{m} N_i\left(x, y, \underline{d}\right) \cdot u_i \\ v\left(x, y, \underline{d}\right) = \sum_{i=1}^{m} N_i\left(x, y, \underline{d}\right) \cdot v_i \end{cases} \tag{4.23}$$

where m is the number of nodes in the element where the point is located, $N_i\left(x, y\right)$ are the shape functions of the element and u_i, v_i are the components of the displacement at the i^{th} node. The approach to find the displacement field is the same afore discussed, just in this case there will be a matrix of sub-windows information instead of solving all the sub-windows one by one. The matrix is built by the *nodal assembly matrix* as in traditional finite elements methods.

4.2 Description of the algorithm

Defining a good digital image correlation algorithm is quite difficult considering the amount of contributions that since 1981 have been reported. All of the authors are obviously stating the good quality of their approaches among the others, both in terms of speed and of precision. However precision is usually considered more important as it is not a real-time problem[2]. It comes that deciding for a brand new software can be complicated. A good review of the scientific literature anyway helped the choice trying to compare the different approaches. The need of a better understanding of the big amount of not sorted papers about digital image correlation is so strong that recently Pan et al. [58] presented a review on *Measurement Science & Technology.* Moreover the authors stated that "*[...] although numerous literature regarding 2D DIC can be found, there still lacks a review paper which focuses on the technical details and accuracy analysis of the simple and widely used optical method. A more detailed review of this method therefore seems to be necessary. [...]*".

Leaving out some special approaches that seam to be not suitable for a general purpose software, the algorithm presented in this study is based mainly on the following characteristics:

- first-order displacement function,
- minimization approach done by Levenberg-Marquardt,
- correlation coefficient calculated as normalized sum of squared differences,
- implementing the reliability-guided digital image correlation method,
- subpixel interpolation approach, based on bilinear and bicubic-spline functions,
- calculating the strains from displacements with Cauchy or Lagrange formulation.

4.2.1 First-order displacement function

Many authors have discussed on the need to have a high order displacement function or just accept a first order one. It is clear that a first-order displacement function leads to a high variety of real situations as rarely the displacement field, locally, it is well approximated just with at least a second order function. Some authors stated that anyway accounting for a second-order displacement function allows a more accurate determination of the deformation

---

[2] The analysis doesn't need to be solved meanwhile the test is running, so the influence of requested time is less important than the quality of the results

parameters directly related to the strain components and so the first-order gradients can be measured more precisely. In Figure 4.4 some examoples of displacement functions applied to a bidimensional array of dots show very well how a second order function may achieve more complex dots distributions and so more heterogeneous displacement fields.

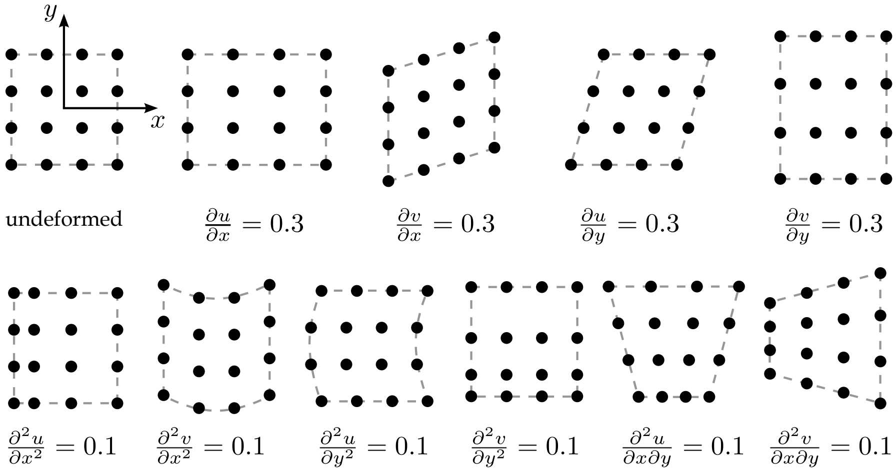


Figure 4.4: Examples of the effect of the displacement function order

Schreier and Sutton [51] in a more recent paper proved as that statement was made upon a series of wrong hypothesis. Moreover a second-order displacement function is introducing an higher standard deviations in the calculated displacements and so a bigger sub-window must be chosen respect to a simpler first-order displacement function.

Moreover a second-order displacement function is introducing higher number of unknowns in the minimzation cycle. In fact as a first-order function has usually about 6 unknowns, a second-order function has about 12 unknowns and so twice. This it means at least a surface in 13 dimensions, which global minimum should be found.

Considering all these contributions, finally the software here presented used a first-order displacement function written as follows:

$$\begin{cases} u = u_0 + \dfrac{\partial u}{\partial x} \cdot (x - x_0) + \dfrac{\partial u}{\partial y} \cdot (y - y_0) \\ v = v_0 + \dfrac{\partial v}{\partial x} \cdot (x - x_0) + \dfrac{\partial v}{\partial y} \cdot (y - y_0) \end{cases} \tag{4.24}$$

As no information are known about the real displacement functions, even not the shape of it, it is not possible to determine the derivatives of it. Thus the Equation (4.24) cannot really account for the first derivatives of u and v, but it is better represented by the following:

$$\underline{u} = \begin{bmatrix} u_0 & u_x & u_y \\ v_0 & v_x & v_y \end{bmatrix} \cdot \begin{bmatrix} 1 \\ x - x_0 \\ y - y_0 \end{bmatrix} \tag{4.25}$$

where u_x represents the value assumed by $\frac{\partial u}{\partial x}$ calculated in $x = x_0$ and the same for the other derivatives. $u_0, u_x, u_y, v_0, v_x, v_y$ are the 6 unknowns to be found to characterize completely the displacement field.

4.2.2 Sub-pixel interpolation

Applying a certain displacement field $\underline{u}$ to the set of points $\underline{p}$ related to a specific sub-window S leads to a set of points $\underline{\tilde{p}}$ that generally will not have whole numbers coordinates as required in a digital image. In fact after deformation a pixel at coordinate $\underline{p} \equiv (12, 78)$ may be placed at the coordinate $\underline{\tilde{p}} \equiv (13.5387, 76.1549)$. Of course there is no clear color information at this point and therefore some approaches must be developed to select a suitable color.

A simple method can be defining each pixel as a square inside a bi-dimensional space with size 1×1 and so every point $\underline{\tilde{p}}$ inside a square will get the color of that square. This approach is very simple but introduces high errors. For example two points with coordinates $\underline{\tilde{p}}_1 \equiv (12.501, 14)$ and $\underline{\tilde{p}}_2 \equiv (12.499, 14)$, have same y-coordinate and very close x-coordinate. Difference of 0.002 pixels can be considered a good precision for DIC method in many cases. This means that the colors at those two points must be very close as the difference in displacement field between $\underline{u}_1$ and $\underline{u}_2$ – the two displacement fields that brought $\underline{p}$ in $\underline{\tilde{p}}_1$ and $\underline{\tilde{p}}_2$ – has to be very small. However, as they will fall in two different pixels domain than the two colors can be even very different. In the example of Figure 4.5 the two colors have gray-scale level 230 for the pixel centered in (12, 14) and gray-scale level 140 for the pixel centered in (13, 14). The difference then will be 110 over a scale of 256 levels of gray.

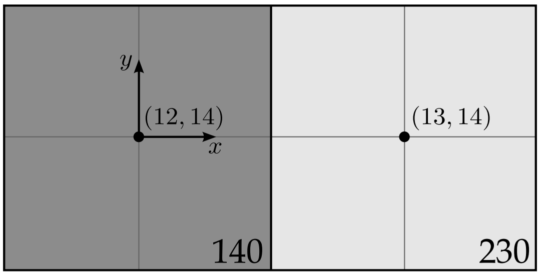


Figure 4.5: Simple sub-pixel approach

A more accurate procedure may consider a linear change of gray-scale levels between pixels. In this approach for a bi-dimensional space, a bi-linear interpolating function is used.

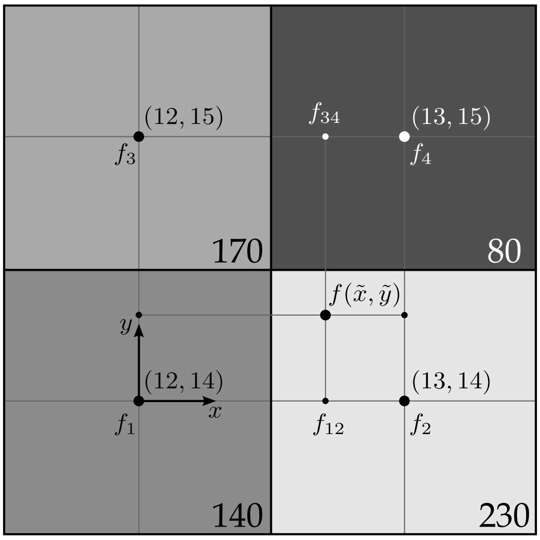


Figure 4.6: Bi-linear sub-pixel interpolation

Figure 4.6 shows a scheme of bi-linear interpolation. In order to understand the color of a general point $\underline{\tilde{p}} \equiv (\tilde{x}, \tilde{y})$ a two-step linear interpolation must be followed. As first a linear interpolation to calculate the gray-scale color f_{12} and f_{34} in two points with the same coordinate $\tilde{x}$ of the point $\underline{\tilde{p}}$ but with coordinte $\tilde{y}$ of the real pixels before and after it. Then a linear interpolation between these two points to calculate the final gray-scale value f at the point $\underline{\tilde{p}}$. The calculation of this interpolation has been described in Appendix C.2.1. The final results is presented here:

$$f(\tilde{x}, \tilde{y}) = \begin{bmatrix} 1 - \tilde{x} + x_1 & \tilde{x} - x_1 \end{bmatrix} \cdot \begin{bmatrix} f_1 & f_3 \\ f_2 & f_4 \end{bmatrix} \cdot \begin{bmatrix} 1 - \tilde{y} + y_1 \\ \tilde{y} - y_1 \end{bmatrix} \tag{4.26}$$

where $\underline{p}_1 \equiv (x_1, y_1)$ is the bottom-left pixel with gray-scale color f_1. It is obvious that even if starting from two linear interpolation functions, the resulting interpolation scheme is non-linear. Even more simple formulation can be introduced if a new coordinate system with center in x_1, y_1 is used. In that case, using $\hat{x} = \tilde{x} - x_1$ and $\hat{y} = \tilde{y} - y_1$ it is possible to write:

$$\begin{aligned} f(\hat{x}, \hat{y}) &= f_1 + \hat{x} \cdot (f_2 - f_1) + \hat{y} \cdot (f_3 - f_1) + \hat{x} \cdot \hat{y} \cdot (f_1 - f_2 - f_3 + f_4) \\ &= a_0 + a_1 \cdot \hat{x} + a_2 \cdot \hat{y} + a_3 \cdot \hat{x} \cdot \hat{y} \end{aligned} \tag{4.27}$$

that is a quadratic function with 4 coefficients.

Using again the example aforementioned, the difference in gray-scale color between two points 0.002 pixels far will be 0.22 on a scale of 256 levels of gray. Thus now two close points will have also close colors.

The problem of this approach is that there is no good continuity of the solution as the bilinear interpolation function doesn't have continuity of the second derivative. A more complex approach is based on the bi-cubic spline interpolation (Appendix C.2.2). This is made by a function described by a bi-cubic function, exactely as the bi-linear function works. In order to increase the smoothness of the interpolation scheme (passing from bi-linear to bi-cubic) a cost in terms of complexity must be fulfilled. In fact while a bi-linear interpolation needs 4 parameters to be defined (related to the 4 colors at the 4 closer pixels), a bi-cubic spline needs 16 parameters to be defined, which depends on 4 color values, 8 first order derivatives (4 respect to x and 4 respect to y) and 4 second-order derivatives (respect to x and y).

The equation which describes the bi-cubic spline interpolation can be written as follows:

$$f(x,y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{i,j} \cdot \hat{x}^i \cdot \hat{y}^j \tag{4.28}$$

where $a_{i,j}$ are the 16 parameters to be found.

This software can consider both bi-linear and bi-cubic spline approach but the latter is suggested as it is generally giving better results even if it is a bit slower.

4.2.3 Minimization approach

As it is mentioned earlier, the approach used in the presented software in this study is based on Newton-Raphson like minimization of the correlation coefficient, by modifying the parameters which describe the displacement field. As explained in detail in Appendix B, the Levenberg-Marquard algorithm is faster than traditional Newton-Raphson techniques and it is also less dependent on the starting guess. This last point is the most important characteristic of the algorithm as a slow algorithm is anyway an algorithm that is moving towards the solution, but an algorithm that is particularly dependent on the starting guess is an algorithm that probably would not necessarily converge to the solution.

The details of the algorithm are described in Algorithm B.1. Here the particular considerations due to the special application will be outlined, especially the needed calculations of the Jacobian vector $\underline{J}$ and of the Hessian matrix $\underline{\underline{H}}$. In fact the iterative method is based on the fact that starting from a value $\underline{u}^{(i)}$ at the iteration i, the next value will be $\underline{u}^{(i+1)} = \underline{u}^{(i)} + \underline{\delta}$ where $\underline{\delta}$ can be calculated as follows:

$$\underline{\delta} = \left[\underline{\underline{H}} + \lambda \cdot diag\left(\underline{\underline{H}}\right)\right]^{-1} \cdot \underline{J} \tag{4.29}$$

where λ is a scalar number that must be calculated by linear search (see Appendix B.3).

The Jacobian vector and the Hessian matrix are used to minimize the correlation coefficient C for each sub-window defined by the analysis grid, and so they are vector and matrix of first and second derivatives of it. The correlation coefficient considered is as follows:

$$C = \frac{\sum\limits_{\underline{p}\in S} \left\{ f\left(\underline{p}\right) - \left[(1-c)\cdot \tilde{g}\left(\underline{p},\underline{d}\right) + b\right] \right\}}{\sum\limits_{\underline{p}\in S} f^2\left(\underline{p}\right)} \tag{4.30}$$

where $\underline{p}$ are the $m \times n$ pixels inside the sub-window S while c and b are parameters able to take into account the contrast and brightness changes during the test to make the calculation of the displacement field independent from these unwanted effects. Considering these 8 parameters, the Jacobian vector and Hessian matrix are as follows:

$$\underline{J} = \begin{bmatrix} \frac{\partial C}{\partial d_1} & \frac{\partial C}{\partial d_2} & \frac{\partial C}{\partial d_3} & \frac{\partial C}{\partial d_4} & \frac{\partial C}{\partial d_5} & \frac{\partial C}{\partial d_6} & \frac{\partial C}{\partial d_7} & \frac{\partial C}{\partial d_8} \end{bmatrix} \tag{4.31a}$$

$$\underline{\underline{H}} = \begin{bmatrix} \frac{\partial^2 C}{\partial d_1^2} & \frac{\partial^2 C}{\partial d_1 \partial d_2} & \frac{\partial^2 C}{\partial d_1 \partial d_3} & \frac{\partial^2 C}{\partial d_1 \partial d_4} & \frac{\partial^2 C}{\partial d_1 \partial d_5} & \frac{\partial^2 C}{\partial d_1 \partial d_6} & \frac{\partial^2 C}{\partial d_1 \partial d_7} & \frac{\partial^2 C}{\partial d_1 \partial d_8} \\ & \frac{\partial^2 C}{\partial d_2^2} & \frac{\partial^2 C}{\partial d_2 \partial d_3} & \frac{\partial^2 C}{\partial d_2 \partial d_4} & \frac{\partial^2 C}{\partial d_2 \partial d_5} & \frac{\partial^2 C}{\partial d_2 \partial d_6} & \frac{\partial^2 C}{\partial d_2 \partial d_7} & \frac{\partial^2 C}{\partial d_2 \partial d_8} \\ & & \frac{\partial^2 C}{\partial d_3^2} & \frac{\partial^2 C}{\partial d_3 \partial d_4} & \frac{\partial^2 C}{\partial d_3 \partial d_5} & \frac{\partial^2 C}{\partial d_3 \partial d_6} & \frac{\partial^2 C}{\partial d_3 \partial d_7} & \frac{\partial^2 C}{\partial d_3 \partial d_8} \\ & & & \frac{\partial^2 C}{\partial d_4^2} & \frac{\partial^2 C}{\partial d_4 \partial d_5} & \frac{\partial^2 C}{\partial d_4 \partial d_6} & \frac{\partial^2 C}{\partial d_4 \partial d_7} & \frac{\partial^2 C}{\partial d_4 \partial d_8} \\ & & & & \frac{\partial^2 C}{\partial d_5^2} & \frac{\partial^2 C}{\partial d_5 \partial d_6} & \frac{\partial^2 C}{\partial d_5 \partial d_7} & \frac{\partial^2 C}{\partial d_5 \partial d_8} \\ & & sym. & & & \frac{\partial^2 C}{\partial d_6^2} & \frac{\partial^2 C}{\partial d_6 \partial d_7} & \frac{\partial^2 C}{\partial d_6 \partial d_8} \\ & & & & & & \frac{\partial^2 C}{\partial d_7^2} & \frac{\partial^2 C}{\partial d_7 \partial d_8} \\ & & & & & & & \frac{\partial^2 C}{\partial d_8^2} \end{bmatrix} \tag{4.31b}$$

where $\underline{d} = \{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8\} = \{u_0, v_0, u_x, v_x, u_y, v_y, c, b\}$ is the vector of parameters to be found (see Section 4.2.1). The Jacobian vector will be defined as follows:

$$\underline{J} = \frac{-2}{\sum\limits_{\underline{p}\in S} f^2\left(\underline{p}\right)} \cdot \begin{bmatrix} \sum\limits_{\underline{p}\in S} \left[f\left(\underline{p}\right) - \tilde{g}\left(\underline{p},\underline{d}\right)\right] \cdot \frac{\partial \tilde{g}\left(\underline{p},\underline{d}\right)}{\partial d_1} \\ \sum\limits_{\underline{p}\in S} \left[f\left(\underline{p}\right) - \tilde{g}\left(\underline{p},\underline{d}\right)\right] \cdot \frac{\partial \tilde{g}\left(\underline{p},\underline{d}\right)}{\partial d_2} \\ \vdots \\ \sum\limits_{\underline{p}\in S} \left[f\left(\underline{p}\right) - \tilde{g}\left(\underline{p},\underline{d}\right)\right] \cdot \frac{\partial \tilde{g}\left(\underline{p},\underline{d}\right)}{\partial d_8} \end{bmatrix} \tag{4.32}$$

Considering now the Hessian matrix, the complexity of calculating all the second order partial derivatives is generally too high and may lead to a considerably time consuming calculation. Thus Vendroux and Knauss [34] proposed a simplified approach that anyway is still applicable and it gives very good results in short time. The exact calculation of a general partial derivative is as follows:

$$\frac{\partial^2 C}{\partial d_i \partial d_j} = -\frac{2}{\sum\limits_{\underline{p}\in S} f^2\left(\underline{p}\right)} \cdot \left[\sum_{\underline{p}\in S} \left[f\left(\underline{p}\right) - \tilde{g}\left(\underline{p}, \underline{d}\right)\right] \cdot \frac{\partial^2 \tilde{g}\left(\underline{p}, \underline{d}\right)}{\partial d_i \partial d_j} + \sum_{\underline{p}\in S} \frac{\tilde{g}\left(\underline{p}, \underline{d}\right)}{\partial d_i} \cdot \frac{\tilde{g}\left(\underline{p}, \underline{d}\right)}{\partial d_j}\right] \tag{4.33}$$

When the set of parameters $\underline{d}$ is close to the solution then $\tilde{g}\left(\underline{p}, \underline{d}\right) \approx f\left(\underline{p}\right)$ and so the first part of the equation can be neglected. Thus it is possible to approximate the second order partial derivative of the correlation coefficient as follows:

$$\frac{\partial^2 C}{\partial d_i \partial d_j} \approx \frac{2}{\sum\limits_{\underline{p}\in S} f^2\left(\underline{p}\right)} \cdot \sum_{\underline{p}\in S} \frac{\tilde{g}\left(\underline{p}, \underline{d}\right)}{\partial d_i} \cdot \frac{\tilde{g}\left(\underline{p}, \underline{d}\right)}{\partial d_j} \tag{4.34}$$

Now just terms as $\frac{\tilde{g}\left(\underline{p}, \underline{d}\right)}{\partial d_i}$ should be calculated. These terms are simpler and so the whole calculation will be not only simpler to be implemented but also faster to run.

Using the chain rule the partial derivative can be calculated as follows:

$$\frac{\tilde{g}\left(\underline{p}, \underline{d}\right)}{\partial d_i} = \frac{\tilde{g}\left(\underline{p}, \underline{d}\right)}{\partial \tilde{x}} \cdot \frac{\partial \tilde{x}}{\partial d_i} + \frac{\tilde{g}\left(\underline{p}, \underline{d}\right)}{\partial \tilde{y}} \cdot \frac{\partial \tilde{y}}{\partial d_i} \tag{4.35}$$

Once the sub-pixel interpolation scheme has been selected, $\frac{\tilde{g}\left(\underline{p}, \underline{d}\right)}{\partial \tilde{x}}$ or $\frac{\tilde{g}\left(\underline{p}, \underline{d}\right)}{\partial \tilde{y}}$ can be calculated and $\frac{\partial \tilde{x}}{\partial d_i}$ or $\frac{\partial \tilde{y}}{\partial d_i}$ can be determined according to the adopted displacement function.

Considering a bi-linear interpolation function there will be:

$$\begin{cases} \frac{\tilde{g}\left(\underline{p}, \underline{d}\right)}{\partial \tilde{x}} & = \frac{\partial}{\partial \tilde{x}} \left[a_0 + a_1 \cdot (\tilde{x} - x_1) + a_2 \cdot (\tilde{y} - y_1) + a_3 \cdot (\tilde{x} - x_1) \cdot (\tilde{y} - y_1)\right] = \\ & = a_1 + a_3 \cdot \hat{y} \\ \frac{\tilde{g}\left(\underline{p}, \underline{d}\right)}{\partial \tilde{y}} & = \frac{\partial}{\partial \tilde{y}} \left[a_0 + a_1 \cdot (\tilde{x} - x_1) + a_2 \cdot (\tilde{y} - y_1) + a_3 \cdot (\tilde{x} - x_1) \cdot (\tilde{y} - y_1)\right] = \\ & = a_2 + a_3 \cdot \hat{x} \end{cases} \tag{4.36}$$

and for bi-cubic spline interpolation function:

$$\begin{cases} \dfrac{\tilde{g}\left(\underline{p},\underline{d}\right)}{\partial \tilde{x}} = \dfrac{\partial}{\partial \tilde{x}}\left[\displaystyle\sum_{i=0}^{3}\sum_{j=0}^{3} a_{i,j}\cdot\left(\tilde{x}-x_1\right)^i\cdot\left(\tilde{y}-y_1\right)^j\right] = \\ \quad = a_{1,0}+a_{1,1}\cdot\hat{y}+a_{1,2}\cdot\hat{y}^2+a_{1,3}\cdot\hat{y}^3+ \\ \quad +2\cdot\hat{x}\cdot\left(a_{2,0}+a_{2,1}\cdot\hat{y}+a_{2,2}\cdot\hat{y}^2+a_{2,3}\cdot\hat{y}^3\right)+ \\ \quad +3\cdot\hat{x}^2\cdot\left(a_{3,0}+a_{3,1}\cdot\hat{y}+a_{3,2}\cdot\hat{y}^2+a_{3,3}\cdot\hat{y}^3\right) \\ \dfrac{\tilde{g}\left(\underline{p},\underline{d}\right)}{\partial \tilde{y}} = \dfrac{\partial}{\partial \tilde{y}}\left[\displaystyle\sum_{i=0}^{3}\sum_{j=0}^{3} a_{i,j}\cdot\left(\tilde{x}-x_1\right)^i\cdot\left(\tilde{y}-y_1\right)^j\right] = \\ \quad = a_{0,1}+a_{1,1}\cdot\hat{x}+a_{2,1}\cdot\hat{x}^2+a_{3,1}\cdot\hat{x}^3+ \\ \quad +2\cdot\hat{y}\cdot\left(a_{0,2}+a_{1,2}\cdot\hat{x}+a_{2,2}\cdot\hat{x}^2+a_{3,2}\cdot\hat{x}^3\right)+ \\ \quad +3\cdot\hat{y}^2\cdot\left(a_{0,3}+a_{1,3}\cdot\hat{x}+a_{2,3}\cdot\hat{x}^2+a_{3,3}\cdot\hat{x}^3\right) \end{cases} \tag{4.37}$$

Considering the first-order displacement function, the partial derivatives of the functions $\tilde{x}$ and $\tilde{y}$ are as follows:

$$\begin{cases} \dfrac{\partial \tilde{x}\left(x,y,\underline{d}\right)}{\partial d_i} = \dfrac{\partial}{\partial d_i}\left[d_1+d_3\cdot\left(x-x_0\right)+d_5\cdot\left(y-y_0\right)\right] = \\ \quad = \left\{1,0,x-x_0,0,y-y_0,0\right\} \\ \dfrac{\partial \tilde{y}\left(x,y,\underline{d}\right)}{\partial d_i} = \dfrac{\partial}{\partial d_i}\left[d_2+d_4\cdot\left(x-x_0\right)+d_6\cdot\left(y-y_0\right)\right] = \\ \quad = \left\{0,1,0,x-x_0,0,y-y_0\right\} \end{cases} \tag{4.38}$$

4.3 The software *Phoenix*©

The main work of this thesis is the development of the software *Phoenix*© to calculate the 2D strain map of a object under load, based on DIC method. There are several papers reporting DIC home-made software as well as some commercial packages. The idea of developing *Phoenix*© DIC software started after trying several experiences with other home-made and professional software. The technique is very powerful and the number of its possible applications is without limit. Depending on the quality and characteristics of the camera and the testing apparatus, the precision can be very high.

Digital image correlation technique is very powerful and delicate to set according to the particular kind of different tests. In order to have reliable results, each experiment needs several attentions. Regarding to the importance of the quality of the results and also the speed of the execution, development of DIC software is a very difficult task, particularly because of the matter of speed. The existing commercial DIC software is enough good for non-programmer technicians and researchers, but they are very expensive to be employed in every research laboratory. Another fact is that the reliability of the results of a DIC software has to be evaluated according to some scientific approaches by engineers and experts in experimental mechanics techniques. However the commercial developed software follow the idea of "a solution must be given to the user", so that sometimes the user may lose the reliability of the results, even unconsciously. Therefore the lack of powerful DIC software is felt as well as many other numerical applying software.

Considering all these, this research is mostly focused on presenting a home-made software able to produce good results in a reasonable time, focusing on the understanding of the influence of several reported techniques available in this field. *Phoenix*© has developed according to algorithm explained in Section 4.2. The present section focuses on the internal structure of the software. As first the main task is to chose a programming language. The most used programming languages are C (or its object-oriented version C++), Visual Basic©, Python™, Java, etc. Considering that the application to develop is quite heavy, full of calculations, and skipping the user friendly interface that can be neglected in the beginning, C++ sounds the preferred language. It is a compiled language (not interpreted) and can be used easily for cross-platform software. To make it even more cross-platform and in order to be sure that all the requirements of the ANSI standard ISO 9899 are fulfilled, GCC (GNU Compiler Collection, `http://gcc.gnu.org/`) compiler has been selected. In particular the object-oriented C++ "dialect" has been selected. The GCC includes front ends for C, C++, Objective-C, Fortran, Java, and Ada, as well as libraries for these languages.

As the GCC is mainly a GNU/Linux tool but with porting on all the main operating systems, an application that will compile in Microsoft© Windows© will easily compile also in GNU/Linux and Mac OS X, just with changing the few libraries that are operating system dependent. The software is command line and it works as follows:

```
Dic -f list_images -size left bottom right top [options]
```

where `list_images` is an ASCII file containing the list of the path of the files containing all the images to be analyzed (master and slaves). While `left`, `bottom`, `right` and `top` are the coordinates of the rectangle where the analysis has to be focused. Inside this area the grid defining the sub-windows centers will be created. The options can be as follows:

- `-g`: defines the grid spacing (the default value is 25 pixels)
- `-w`: defines the sub-window size (the default value is 30 pixels)
- `-o`: defines the name of the folder where the results will be saved (the default value is simply "output")
- `-d`: defines the path of an ASCII file containing the x and y coordinates of the points of the grid that should not be analyzed (in order to remove areas of the previously defined rectangle)
- `-s`: defines the algorithm used to calculate the strains (possible values are the default `cauchy` or `lagrange`)

- `-i`: defines the algorithm used for the sub-pixel interpolation (possible values are `bilinear` or the default `bicubcspline`)
- `-m`: defines the minimization approach (possible values are `newtonraphson` or the default `levenbergmarquardt`)
- `-df`: defines the displacement field degree (possible values are `rigid`, the default `affine` or `quadratic`)
- `-ms`: defines the maximum number of iterations to be used in the minimization algorithm (the default value is 50)
- `-t`: defines the value of the correlation coefficient that stops the minimization iterations
- `-n`: defines the number of points in an area of $(2 \cdot n + 1)^2$ points for the smoothing of strain data
- `-h`: shows the help page

When the software starts, it checks the availability of the license and then it checks the existence of all the listed images (master and slaves). Then it creates the sub-windows grid as defined at the command-line and it starts loading the master image and the first slave image. Now the real DIC algorithm starts and at the end the displacements associated with each sub-window is calculated. Then the strains are calculated and the cycle starts again with a new slave image.

4.3.1 Image loader

In order to read the pixel color of many kinds of image formats, some library have been developed and some others have been download from the open source community and joined inside the software. The image formats accepted are: BMP, JPEG (or JPG), PNG and TIFF (or TIF).

BMP file format

This file format is directly implemented using GDI+ APIs. The file especially is opened by the object `CBitmap` and then the pixels data are extracted by the use of `GetBits` method. Finally the data are converted in 8-bit gray-scale.

JPEG file format

JPEG standard is widely used in many applications and even if is not the best choice for digital image to be used in DIC analyses, it is considered by this software. The implementation is done by the use of the open source and free to use library by the Independent JPEG Group (IJG, `http://www.ijg.org/`). The process is mainly consisting in the following steps:

- open the file
- create a “decompressor”
- read the header of the file
- decompress the header
- read the file and convert it in pixel colors
- destroy the “decompressor”
- close the file

At the end the colors are converted in 8-bit gray-scale.

PNG file format

The PNG is another widely used standard. It is a lossless compression image format with 8-bit alpha channel (see `http://www.libpng.org/pub/png/`). This is a very good format as usually it needs much less storage space respect to formats like TIFF, but it keeps still the same quality as it is a lossless format. The library used to manage the file is an open source free to use library developed by the PNG team. The process is mainly consisting in the following steps:

- open the file
- read the header
- read the file and convert it in pixel colors
- close the file

As the file can be saved in different formats (like RGB, RGBA, 8-bit gray, 8-bit gray plus 8-nit alpha channel, etc.), at the end the colors are converted in 8-bit gray-scale.

TIFF file format

The TIFF is a relatively old standard, developed by Aldus and now under the control of Adobe Systems. Even with being old, as it can store more than 8 bits per channel or floating point color information, it is still quite used in scientific applications. Moreover more than one image can be stored in a unique file. The library used to manage the file is an open source free to use library (see `http://www.remotesensing.org/libtiff/`). The process is mainly consisting in the following steps:

- open the file
- read the header

- read the file and convert it in pixel colors
- close the file

At the end the colors are converted in 8-bit gray-scale.

4.4 Applications

The digital image correlation technique is used in many situations, especially when high heterogeneity is observed or when small cracks must be followed during propagations [59]. Thus it is mainly used when traditional techniques are not able of enough precision or sensibility or when traditional techniques are too expensive or too complex to be implemented.

In many cases it is possible to see digital image correlation as just a simple tool for normal measurements. In fact as a DIC based device is easy to sue, a laboratory that adopted such apparatus and is equipped by a working digital image correlation software, may uses it for many purposes. A common digital image correlation equipment is made of a normal computer plus a good quality camera or camcorder. It is also possible to make a movie and then extract the frames at the required speed. Of course a lossless algorithm helps in storing data without losing any information. It is important to notice that while the wide use of JPEG [60] compression format is good for most of purposes as it gives a typical compression ratio of about 10:1, in this case the details are more important than the compression ratio. In fact JPEG compression is lossy, meaning that some original image information is lost and cannot be restored. It is known that human vision is much more sensitive to small variations in color or brightness over large areas than to the strength of high-frequency brightness variations. Thus in JPEG the color information are converted in the spatial frequency domain and the colors information are stored with different accuracy according to the frequency. There are some lossless version of JPEG but are rare and not standardized. Considering the most known image formats, it is better to use something like BMP [61], PNG [62] or TIFF [63], that are generally lossless.

In next sections some direct applications of the digital image correlation techniques performed at the laboratory of the Industrial Engineering Department (see `http://ied.unipr.it`) of the University of Parma (see `http://www.unipr.it`) will be presented. In particular two different kind of materials have been studied:

- metallic materials: nodular cast iron (NCI) and silumin (AlSi),
- composite materials: carbon fibers reinforced plastic (CFRP).

The first set is characterized by generally macroscopically homogeneous and isotropic materials behavior of which is highly dependent on the local plastic deformations at micro-scale. The second set of materials is made of even macroscopically heterogeneous materials, characteristic of which is still dependent on local deformations but at meso-scale.

4.4.1 Metallic materials

Knowledge of the micro-structure/deformation relationship in materials is a crucial issue in mechanics because micro-structure strain heterogeneity and damage accumulation control fundamental phenomena of the mechanical behavior of materials such as fracture, fatigue and creep. The theoretical activity in this area has been developing at a faster pace than the experimental activity, although the detailed information required is often not readily available and is supplanted by assumptions. Optical full-field measurement techniques have been providing significant inputs about material mechanics, although their application to micro-structural scales remains a serious challenge. Moiré interferometry has been used to determine heterogeneous plastic strains [18, 65], but it requires a diffraction grating to be deposed and so it obscures the underlying microstructure and it makes the local strain/microstructure correlation quite difficult. On the other hand the DIC technique is able to use directly the micro-structure as random gray speckle [64].

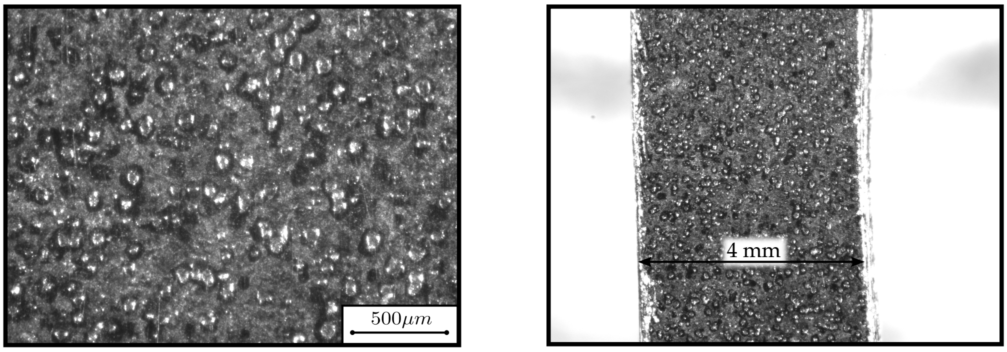
Two metallic materials have been used in this study: nodular cast iron (NCI) and silumin (AlSi). The first one is a GJS400 with a ferritic/pearlitic matrix characterized by a distribution of spherical graphite nodules with a nodule density of 104 nodules per mm^2 and an average diameter of 30 μm. Ferrite is a high-ductility low-yield material while pearlite is a high-strength low-ductility material. Consequently the heterogeneous microstructure deforms in a complex way, even if macroscopically it seams homogeneous with the characteristics described in Table 4.1.

In order to use the micro-structure as suitable speckle, it must be carefully prepared. After an adequate polishing the surface has been treated with Nital 3%. In Figure 4.7 some images of a NCI specimen are shown. The heterogeneous nature of the nodular cast iron is clearly visible in this figure. The light gray spheres are graphite nodules, while the black rings around them are made of ferrite. The dark gray material is pearlite.

The second studied material is an aluminum alloy AlSi7Mg, mechanical properties of which are presented in Table 4.2. This alloy (also called A356-T6) is mainly used as cast material for automotive, especially for the crankcase and the cylinder head, because of its good castability, hydrostatic loads resistance and corrosion resistance, due to the presence of particles of Mg$_2$Si that has an electrochemical potential close to the one of the aluminum matrix. Moreover a thermal treatment of precipitation hardening helps to increase the mechanical resistance of the material.

Table 4.1: Nodular cast iron GJS400 mechanical properties [64]

Property	Symbol	Value	Unit
Ultimate strength	R_m	559	MPa
Yield stress	$R_{p_{0.2}}$	345	MPa
Elastic modulus	E	170	GPa
Elongation	A_5	14	%



(a) 4.5X image of a GJS 400 NCI (b) 2X image of a NCI specimen

Figure 4.7: Nodular cast iron images

Table 4.2: Silumin AlSi7Mg mechanical properties [66]

Property	Symbol	Value	Unit
Ultimate strength	R_m	234	MPa
Yield stress	$R_{p_{0.2}}$	166	MPa
Elastic modulus	E	72	GPa
Elongation	A_5	3.5	%

Also in this case an important preparation of the specimen surface must be executed before the testing procedure. In order to reveal the dendritic structure at the micro-level the *tucker* hatcher agent is used. The chemical composition of this hatcher is showed in Table 4.3. The preparation is based on the following steps:

- submerge the specimen in tucker for 10÷15 seconds
- an optional heating may give a better hatching effect
- wash the specimen with hot water and alcohol
- dry carefully the specimen
- wait roughly one minute before using it

Table 4.3: Tucker chemical composition

Components	Quantity	Unit
HCl	45	ml
HNO_3	15	ml
HF	15	ml
H_2O	25	ml

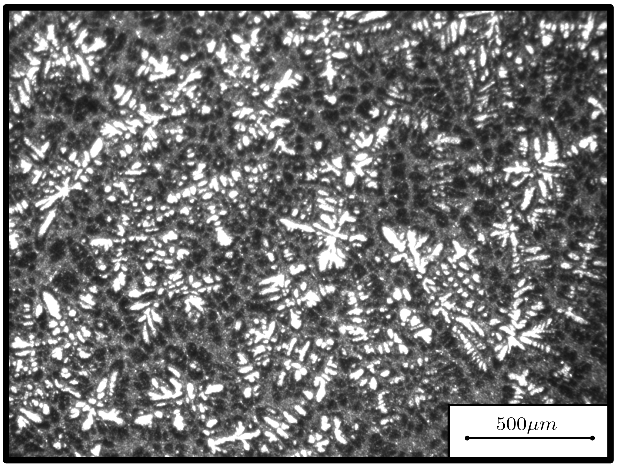


Figure 4.8: Image of the AlSi7Mg specimen after hatching with tucker

Figure 4.8 shows the results after hatching. The image has been taken with a magnitude of 4X.

In both cases, as the heterogeneity is clear just at micro-level, a high-magnification approach must be followed. According to this idea a mini-testing machine has been designed and built at the Department of Industrial Engineering of the University of Parma [67].

Experimental set-up

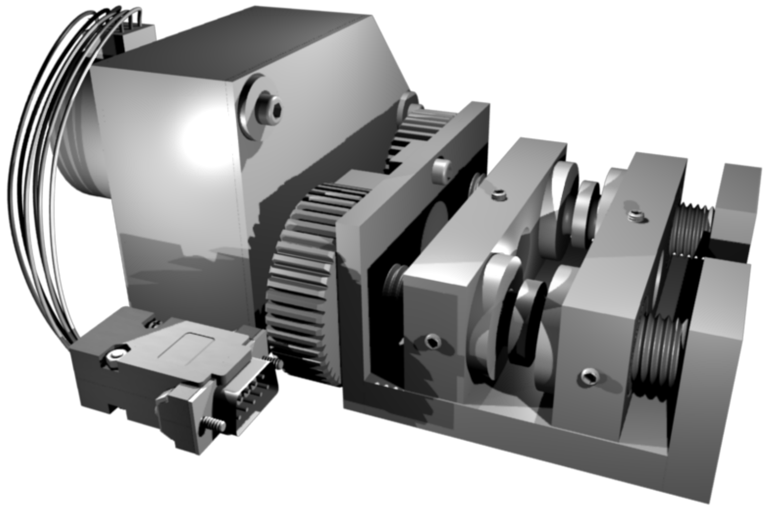


Figure 4.9: 3D drawing of the mini-machine MM1

Studying materials at high-magnification introduces several problems related to the stability of the testing process. Normal testing machines for macroscopic analyses are not suitable. The problem is that even the precision of these machines are high, it is not high enough for small specimens of few millimeters are less to measure the f displacements, but mainly the loads. In fact the frictions of the big actuators are even order of magnitude higher than the required load to be applied to the specimen. Moreover the unavoidable vibrations of the chassis may lead to erroneous measurements.

To solve all these problems an *ad-hoc* testing machine has been designed and built. Figure 4.9 shows the testing machine. Designed to be inserted inside the scanning electron microscope (SEM), it presents a compact shape and a small size. The main idea is to clamp the specimen in the center of the machine and pull it symmetrically by two moving crossheads connected to synchronized screws. These are in turn connected to a gear reducer and finally to a stepper motor.

The stepper motor is electronically connected to a computer in order to be able to control the movements of the crossheads. Apart from the actuators, also a load transducer is used. Figure 4.10 shows the drawing of the load cell. It is made of a cylindrical sheet of metal with four strain-gages glued to it. The strain-gages are connected to each other as in a full-bridge configuration (see Figure 2.12(d)). When the crossheads are moving away from each other they pull the specimen and compress the load cell. The compression along the cylinder axis causes the deformation of the strain-gages. The output from the Wheatstone bridge is proportional to the load, and related to it by an accurate calibration procedure. Before sending the digital data to the computer by the parallel port, the measured electrical tension is converted by a ADC[3].

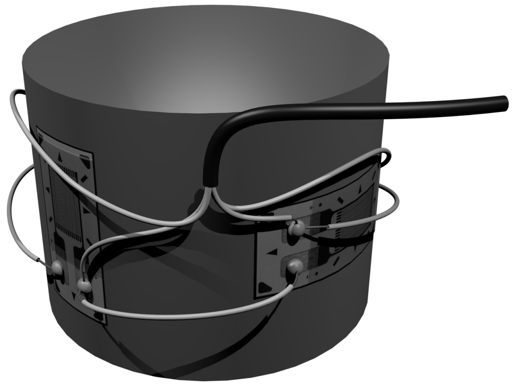


Figure 4.10: Load cell designed and built for the MM1 testing machine

The used specimen is a small sheet of metal, usually with a thickness of 1 mm, cut with a shape as shown in Figure 4.11. The two holes are used to mount it inside the testing machine using two pins. This area is very important as the load is passing by these pins to the specimen through the holes. Finite element models of the testing machine and of the specimen proved that for common materials the deformations of the clamps are negligibles and the specimen holes are able to resist against the load, allowing a fracture surface in the central part of the specimen.

In order to acquire digital images of the specimen surface, a digital device which is a Basler

---

[3] Analog-to-Digital Converter, a device which converts continuous signals to discrete digital numbers.

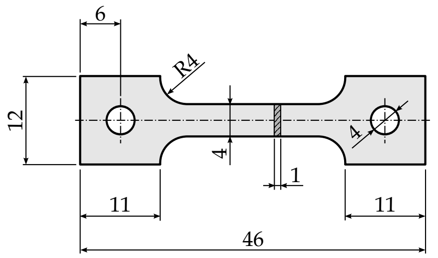


Figure 4.11: Drawing of the specimen designed for the MM1 testing machine

A102f with CCD sensor ICX285AL with a diagonal of 11 mm and a resolution of 1392x1040 pixel has been used. Each pixel has a size of 6.45 μm. The transfer rate can reach up to 15 frame/second through a firewire 1394a connection cable (see `http://www.baslerweb.com`).

The utilized lens is a TV macro lens TVML/Z4.5 (with a numerical aperture from 0.035 to 0.05 for a magnitude from 0.75X to 4.5X) from Goyo Optical Inc. (see `http://www.goyooptical.com`). The complete experimental set-up can be seen in Figure 4.12 where just the computer and the connecting cables are omitted. The micrometric movements allowed by the vertical shaft provides a good focus.

Silumin results

The tests performed on the silumin specimens are focused on the understanding of this material at the micro-level. The typical applications of this material needs high performance and thus good understanding of the plasticization and damage evolution. A stress-strain curve is obtained using the testing machine as afore described and using the strain maps applying the DIC method. This curve, shown in Figure 4.13, presents an elastic modulus of 68.9 GPa and so relatively similar to what expected by standard tests on the same material (see Table 4.2). Since the beginning of the test up to reaching to the final strain of 6%, some parts of the specimen start to reveal the higher values of the strain. This behavior is probably because of the sliding of the crystal lattices or because of the effects of softer parts of the material.

The hatching technique used here has made the direct comparison between the strain map and the micro-structure possible. Figure 4.14 shows this method well. In fact the central colored area is the area analyzied with DIC and the strain map superimposed is the longitudinal strain ϵ_x (the loading is along the horizontal direction) at the time instant when the average strain is about 6.0%, with a transparent effect added and placed at the correct coordinates in the image space. In fact here the map is showing the strain at a frame close to the final failure.

Nodular cast iron results

The specimen was stretched quasi-statically up to a total strain of about 4% with simultaneous digital image acquisition. The field of view of each image depends on the microscope lens used. Displacement fields were obtained after processing and compared with the micro-structure.

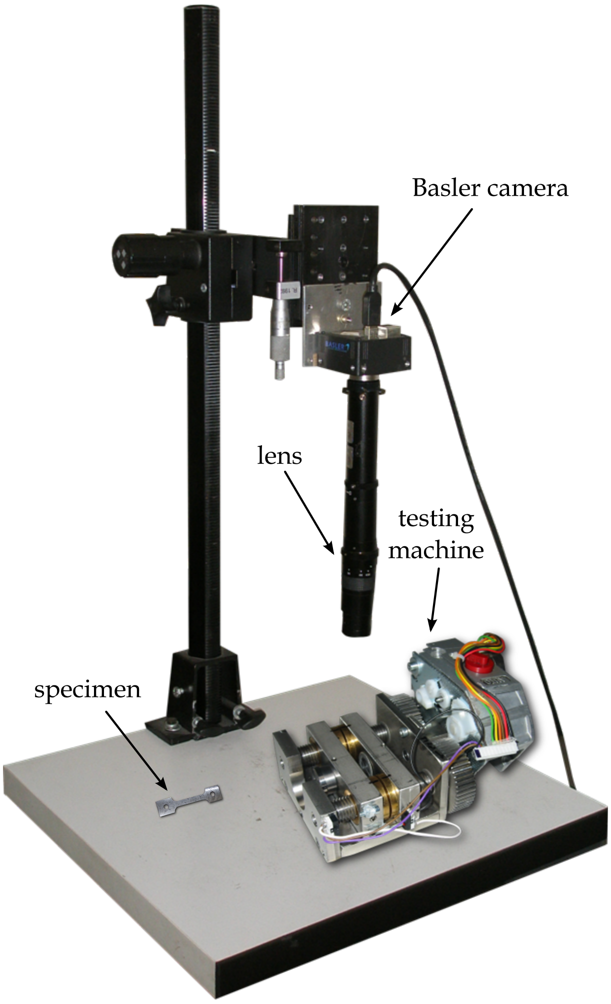


Figure 4.12: Experimental set-up of the mini testing machine MM1 with Basler camera

The heterogeneity in the displacement field becomes a local strain concentration. With increasing load, such heterogeneities modify in intensity but not in position. A qualitative correlation between strain and graphite nodule distribution is shown in Figure 4.15(a). Strong strain gradients are determined in the vicinity of the graphite nodules which can be assimilated to a circular discontinuity in a plate with an associated elastic strain concentration factor of about 3. When the elastic limit is exceeded, the plastic strain concentration factor – defined as the ratio between the peak strain and the averaged strain – is expected to increase beyond the elastic value of 3 and the present measurements confirm it.

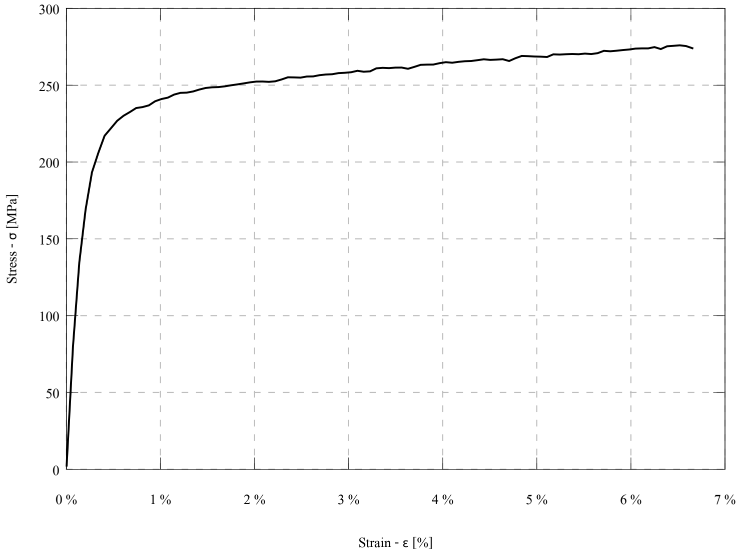


Figure 4.13: Stress-strain curve of AlSi7Mg specimen. Stresses are taken from mini tensile machine and strains from DIC results

4.4.2 Composite materials

Woven fabric composites are increasingly used to obtain superior dimensional stability and shapeability and balanced in-plane properties. A structural laminate is obtained by laying up a number of laminas, each with its own specific orientation in a mold to obtain a desired 3D shape. Due to their anisotropic nature it is difficult to fully predict the mechanical characteristics and behavior of textiles composites, especially if they are produced with complicated architectures. Full-field strain mapping methods with high spatial resolution and strain sensitivity can play a significant role in the understanding of how much changes occur on the surface of textile composites. DIC method is finding increasing use in recent years because of the relative simplicity, low cost and adequate performance in many applications dealing with the mechanics of materials.

In this study a 2x2 twill-weave carbon fiber reinforced epoxy lamina was monitored during in-plane tensile loading in the global linear elastic regime and local strains are determined using DIC technique. The experimental data are then directly compared to the strain distribution at the unit cell level obtained computationally using the finite element method.

A single lamina has been used in order to reduce the three dimensional effect due to several plies connected together. As DIC technique is only able to monitor the behavior of the material at the surface, the surface of a specimen made by a single ply will give a better understanding of the free movement of yarns respect to the constrained movement of the yarns in multiplies composites. Several specimens were obtained out of a pre-preg mat sheet, according to ASTM D3039 standard [68]. According to this standard the specimen is rectangular shaped with the dimensions of 250 mm x 25 mm in 0.3 mm. Considering that each yarn is 2 mm wide, about 12

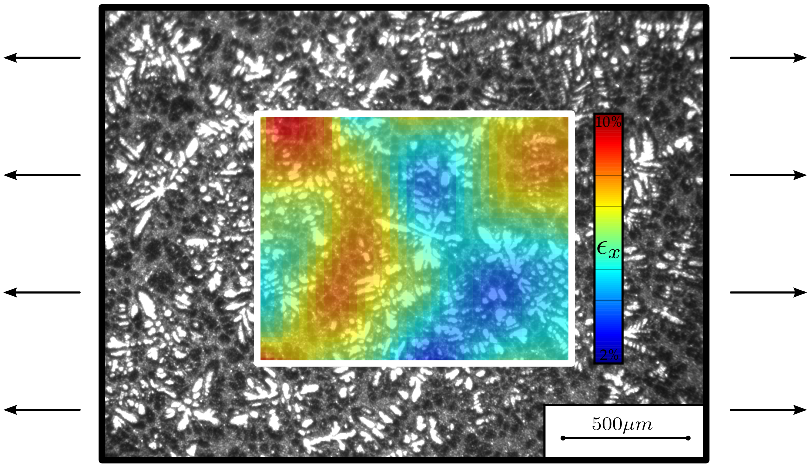
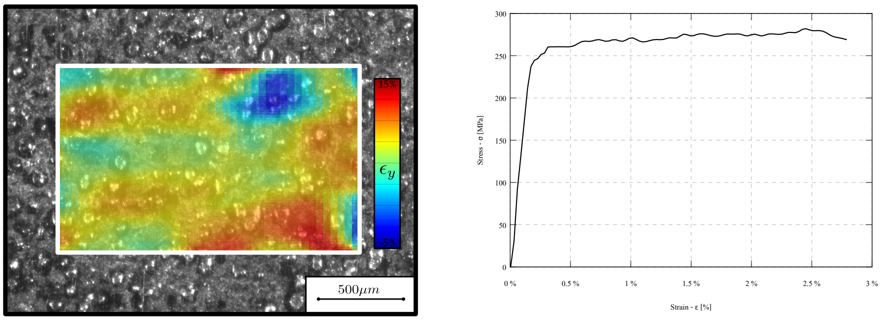


Figure 4.14: Direct superposition of strain map on the micro-structure of AlSi7Mg

yarns are present along the width. Aluminum tabs were glued at the ends of the specimens to allow reliable machine grip without breaking the specimen.



(a) Direct superposition of strain map on the micro-structure of NCI

(b) Stress-strain curve of NCI specimen

Figure 4.15: Examples of results obtained from tests on NCI specimens

Experimental set-up

Considering the ultimate elongation of 1.2%, the ultimate stress of 600 MPa and the elastic modulus of 45 GPa for the specimen of carbon fiber reinforced plastic composite with the mentioned dimensions, an MTS 810 (Material Testing System, `http://www.mts.com`) servo hydraulic machine has been used. The selected load cell is 5 kN. The set-up of the machine is constant rate of elongation (CRE). According to ASTM D3039 the time to failure has to be within 1 to 10 minutes. Therefore for these specimens the crosshead speed has been set on 1 mm/min.

In addition to the loading machine, a digital camera, Basler A102f with CCD sensor ICX285AL with a diagonal of 11 mm and a resolution of 1392x1040 pixel, has been used to capture the im-

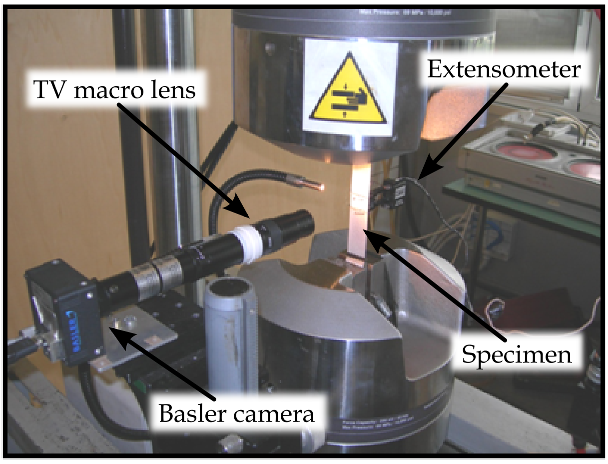


Figure 4.16: Experimental set-up of the MTS 810 with digital camera

ages of the specimen surface during the extension test. Each pixel of the image has a size of 6.45 μm. The transfer rate can reach to 15 frames/second using a firewire 1394a connection cable (see `http://www.baslerweb.com`).

The two employed lenses are respectively TV macro lens TVML/Z4.5 (with a numerical aperture from 0.035 to 0.05 for a magnitude from 0.75X to 4.5X) and manual zoom lens GMZ135108 (with a focal length from 13.5 to 108 mm), both from Goyo Optical Inc. (see `http://www.goyooptical.com`).

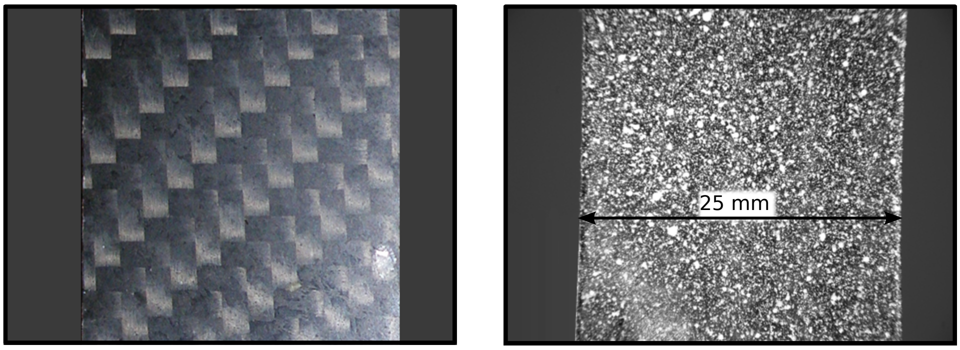
Specimen preparation

The digital image correlation needs a random distribution of grays in order to be effective. In fact a big uniformly colored area can lead to a misunderstanding caused by this method. It is obvious that an accurate preparation must precede the testing phase.

In detail, as just the gray scale level will be captured by the camera, therefore the specimen should be painted in back and white. There are several ways to make black and white painted specimens:

- a black background with white dots over it,
- a white background with black dots over it,
- randomly placed black and white dots.

The first and the second are qualitatively similar, just the background color is switched with the fore color. Anyway in both cases it is possible to be sure that the specimen is completely covered by painting. Moreover the high contrast between black and white helps to evaluate the quality of the painted specimen. This is referring to when the specimen has the agglomeration of black (or white) dots or, in the other side, when it has areas that are poorly covered. The third option is more confusing if there is not enough contrast between the color of the specimen and the color of the paintings.



(a) Not painted specimen (b) Painted specimen

Figure 4.17: Image of a CFRP specimen before and after preparation for being used by DIC

The drawback of the first two cases is that what the DIC will follow will be the painting and not the surface deformations. Thus even the painting has to follow well the specimen movements or DIC method will give bad results. In order to avoid this important problem the third option is preferable but just if the contrast between the paintings and the color of the specimen is high enough. As the CFRP has a natural black color itself, using the white dots would be enough, that is also easier to prepare the specimen.

Figure 4.17(b) shows an example of the specimen with white dots. The used paint is an acrylic enamel gloss white (RAL 9010, `http://www.ral.de`). Spraying far from the specimen leads to small randomly distributed dots (homogeneously over the whole area).

4.5 DIC results on CFRP specimens

In order to study different problems related to DIC different kind of tests have been performed. Hereafter the main results of those tests will be presented and discussed in detail. In particular the analyses will follow three steps:

- DIC vs. extensometer (macroscopic);
- DIC vs. FEM (mesoscopic);
- off-axis tension of CFRP.

4.5.1 DIC vs. extensometer

The extensometer (see Section 2.1) is an electronic device able to measure the strain occurred between two defined points of the specimen with a high precision. As the distance between these two points, called gage-length, is usually of the order of few millimeters or few centimeters, it is suitable just for macroscopic measurement of strains or for homogeneous materials. In this study a first comparison of the DIC with extensometer is used in order to check the precision of the method even at strain levels of few percents.

In order to compare a macroscopic strain information with *Phoenix*© output at the meso-level, an integration method must be used. In this case a simple approach was chosen. As all sub-windows had the same size once the displacements at each grid-point was calculated, the strains was derived and then averaged. At this point the average value of the strain, calculated at each slave image, was compared with the stress-strain curve of the extensometer.

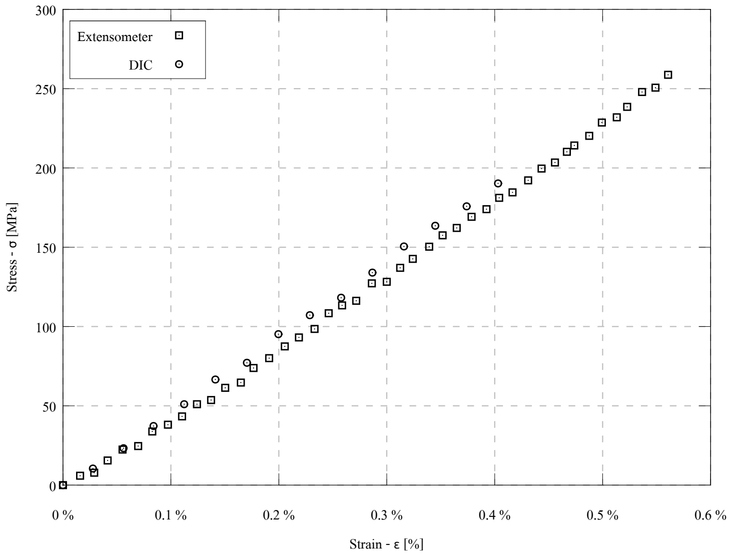


Figure 4.18: DIC vs. extensometer at the early stage of the tensile test

Figure 4.18 shows a comparison of the data from an extensometer and the average values calculated by DIC at the early stages of a tensile test. This comparison, made with a young version of the developed software, shows a good agreement between DIC and the extensometer. It is important to notice that the small thickness of specimen leads to an explosive failure and so it is important to unmount the extensometer at low load.

The elastic modulus calculated from the data derived from the extensometer is 44.7 GPa while the one which is calculated by linear regression of the DIC data is 46.8 GPa, with a difference of just 4.7%. Better results is achievable by using a more accurate set of parameters. This means that DIC method can be a considerable approach to "virtual extensometer". In fact in many cases the use of the extensometer can be difficult or even impossible. For example in high or low temperature conditions or when the geometry of the specimen is too complex to let the extensometer to be mounted or finally when the mechanical properties of the material will easily changes with connecting the extensometer to the device. In all these cases DIC is a good choice, as affordable and easy to use non-contact technique.

4.5.2 DIC vs. FEM

After a first and rough validation of the method, a more important comparison was performed: a comparison between the DIC strain fields and the strains calculated by finite element models of CFRP laminas. This comparison has a twofold meaning: validate DIC using low strains data obtained by FE model and set the material parameters of the FE model using high strains data (non-linear behavior).

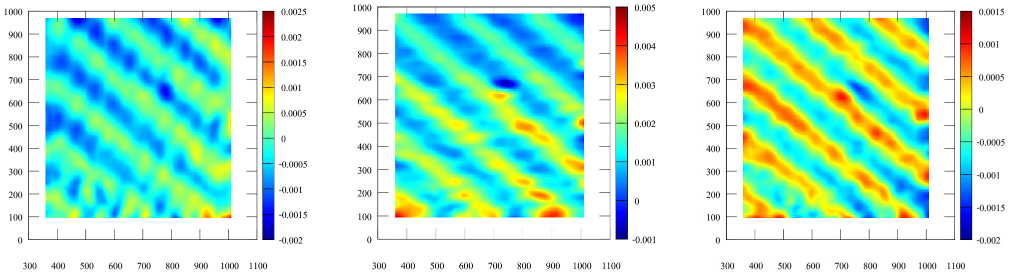
This two-way setting is possible because DIC is able only to evaluate the strains. Then considering that at the linear regime, according to an average strain value in the unit-cell, the strain field on the whole cell is not strongly dependent on the mechanical parameters of the materials used to build the model. Thus the FEM can be used to compare the strain results with DIC data. When the stresses are used, the approach is different and so that mechanical properties are much more important as in this case there are different stresses for the equal strains. Moreover the damage criteria implemented in the numerical model are mainly working at high strains and so the non-linear behavior of the material is emphasized.

The tests were performed mainly with two different lenses. One at the magnitude of 0.75X with the lens TVML/Z4.5 and so a field of view of about 13x9.8 mm^2 that is 2.2 times bigger than the area of the unit-cell (that is about 7.7x7.7 mm^2). The dimension of each pixel is about 9.45 μm. The second one with the lens GMZ135108 and with a field of view of about 48.3x36.1 mm^2 that is 29.4 times bigger than the area of the unit-cell. The dimension of each pixel is about 34.7μm. As the warp yarns are along the loading direction and the weft are perpendicular to it, the axis y will be referred as "longitudinal" direction (the direction of loading) and the axis x as "transverse" direction. Thus the ϵ_y refers to a longitudinal strain, ϵ_x refers to a transverse strain and γ_{xy} refers to shear on the plane xy.

Some strain maps will be analyzed and discussed in the following pages. The maps are usually presented with the $x-y$ coordinates in pixels. Moreover the dimension of the studied area depends on several afore discussed considerations, and so will not be directly related to the dimensions of the images presented here.

Strain components

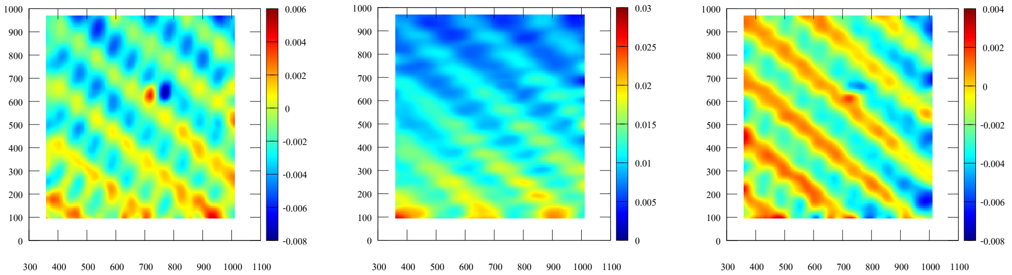
An important aspect of the DIC method is that it is easily able to calculate all the components of the strain tensor on the surface of the specimen. This means that it gives directly the two normal strains ϵ_x and ϵ_y as well as the shear strain γ_{xy}. Whereas DIC has an almost constant precision over the displacement and not proportional to the absolute values, lower values have higher errors. An example is shear strain which is usually at least one order of magnitude less than the longitudinal mean strain.



(a) ϵ_x (b) ϵ_y (c) γ_{xy}

Figure 4.19: Strain maps of a CFRP specimen under tension, at an average longitudianl strain of 0.15%

Figure 4.19 shows the strain maps related to a specimen under tensile loading with an average strain along the loading direction of 0.15%. It is possible to see that in the areas of positive longitudinal strains, there are negative transverse strains, due to the Poisson effect. It is also obvious that the complex architecture of the material leads to a complex distribution of the Poisson's ratio. Despite of what expected the longitudinal strain distribution has negative values that can be probably due to numerical approximations, errors of the DIC calculations or local defects on the images. On the other hand the shear distribution is almost symmetric and with low values (as it is expected from this kind of test). The problem aforementioned is also due to the low measured values (the average longitudinal strain is just 0.15%). In fact considering higher average strain condition would give better results.



(a) ϵ_x (b) ϵ_y (c) γ_{xy}

Figure 4.20: Strain maps of a CFRP specimen under tension, at an average longitudianl strain of 1.12%

Figure 4.20 shows a situation where the average longitudinal strain is 1.12% (close to the final failure). In this situation the longitudinal strain is never negative as it is expected.

Strain evolution

Another important information achieved by DIC method is the capability of obtaining the complete strain evolution of the specimen surface during the entire test[4], for each component of the strain tensor. Figure 4.21 shows the strain evolution of the longitudinal strain ϵ_y. Four different time instants are selected and plotted, starting from early stages till the final failure. Each map has been selected with an average strain about 0.3% higher than the previous map.

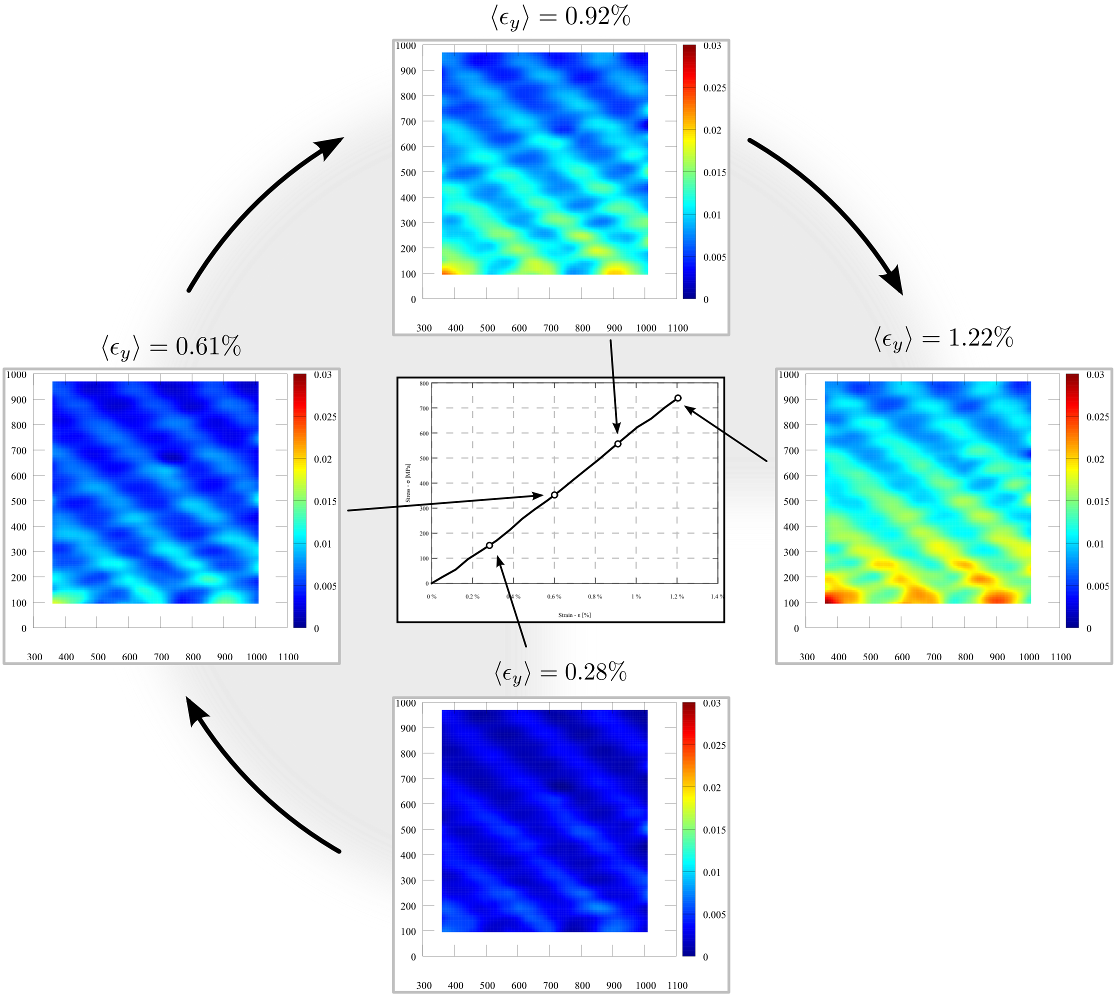


Figure 4.21: Strain evolution of a CFRP specimen under tension, from early stages till final failure

In order to reveal better the differences and the localization of strains the color range is selected equal in all the maps. It is clear that the same process can be exerted for all the components of the strain tensor. At the same way it is possible to select one point of the

---

[4] This statement is true till the defined analysis grid is not moving outside the observed area.

analyzed grid, or average the values of an area and follow its evolution.

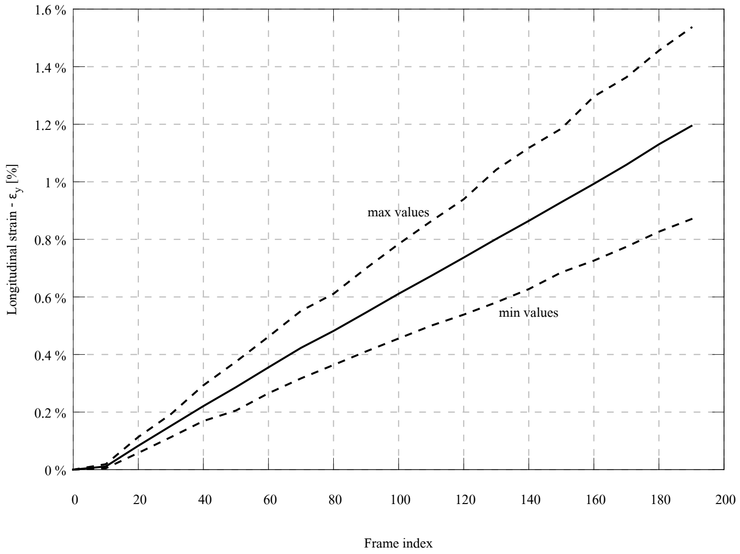
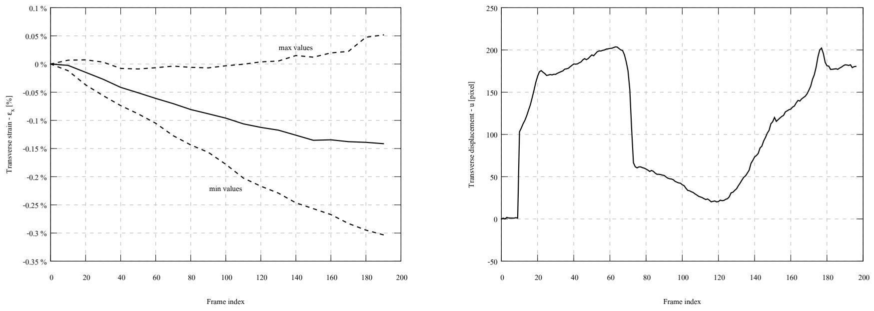


Figure 4.22: Longitudinal strain evolution of a CFRP specimen under tension with comparison between minimum values, maximum values and mean values

Figure 4.22 shows the evolution of the longitudinal mean strain during the test in a 0° oriented specimen (continuos line). The two dashed lines are representing the curves of minimum and maximum strains of each frame during the test on all the analyzed area.



(a) Transverse strain evolution of the entire analyzed area

(b) Transverse displacement evolution of one point

Figure 4.23: Exmaples of strain and displacement evolution along the transverse direction

More complex strain evolution curves can be found using the other two components of the strain tensor. In Figure 4.23(a) the curves representing the minimum, maximum and mean transverse strains are presented. Figure 4.23(b) represents the evolution along the time of the displacement of a single point of the analyzed grid. This curve well represents the complex behavior of the material with complex yarns movements on the plane of the ply and also out-of-plane movements.

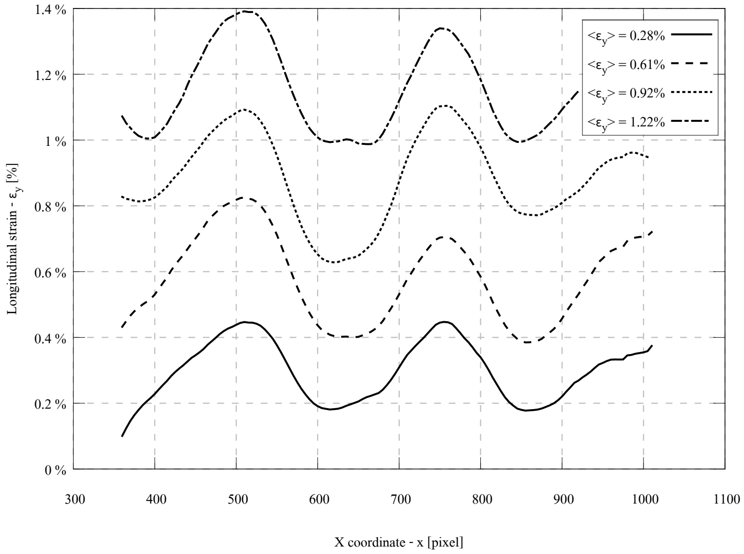


Figure 4.24: Longitudinal strain evolution of a CFRP specimen under tension. Data along x axis, at the middle of the specimen

Another interesting output of DIC method is shown in Figure 4.24. The figure shows four plots of longitudinal strain ϵ_y measured along the axis x, at a y coordinate almost at the middle of the specimen. Different plots, taken at different time instants during the test, show the evolution of the strain which is mainly keeping the shape, but increasing the values according to the increasing damage of the specimen.

Architecture superposition

The ability of the technique to observe the surface directly provides the direct superposition of the strain map over the material architecture or micro-structure. In fact as the method is defined by the observation of the surface of a specimen, it is easily possible to define all the moving sub-windows in the bi-dimensional space of the image, in pixel coordinates. This is very powerful as it allows the user to understand *what* is the cause of the presented behavior.

In Figure 4.25 the architecture reveals that the transverse yarns have higher deformations. The reason of this behavior will be explained better later.

The approach encounters with some difficulties when the specimen needs to be painted as the paint hides the material structure. This is the case of CFRP specimens as described in Section 4.4.2. Furthermore if the painting technique is based just on the deposition of white dots, there is still one possibility. Considering the specimen already mounted in the testing machine and ready to start the test, an accurate selection of the illumination can reveal the material below the painting. This single image can be used to find the position of the yarns.

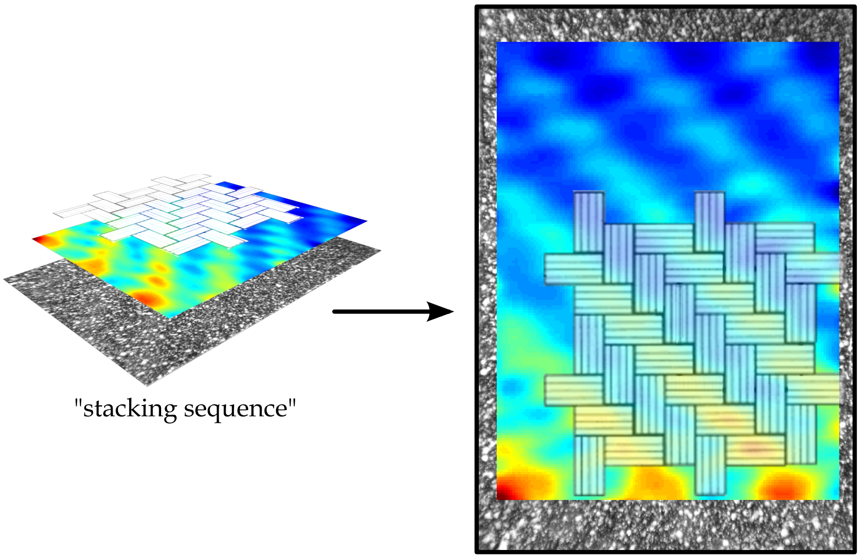


Figure 4.25: Architecture superposition on the strain maps of a CFRP specimen

Comparison with FEM results

Figure 4.26 shows a FE model of a twill weave 2x2 unit-cell made of CFRP where the matrix has been removed to make the yarns more visible. According to the model described in detail later (see Chapter 5) the model has been built and loaded to simulate a tensile test on a infinite media (made by a continuous repetition of unit-cells in x and y directions, which are both lying on the plane of the ply).

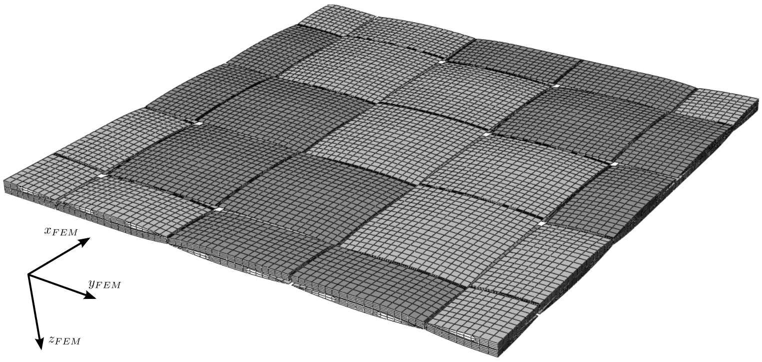


Figure 4.26: FE model of a twill weave 2x2 unit-cell made of CFRP

In order to compare just the strains, localized comparisons were performed using configurations where the average strain on the three-dimensional unit-cell of FE model equals the average strain on the bi-dimensional DIC grid (or as close as possible). Once the overall strain is the same, the local comparison assumes greater significance.

Figure 4.27 shows the average strain evolution along the time obtained by DIC analysis (as

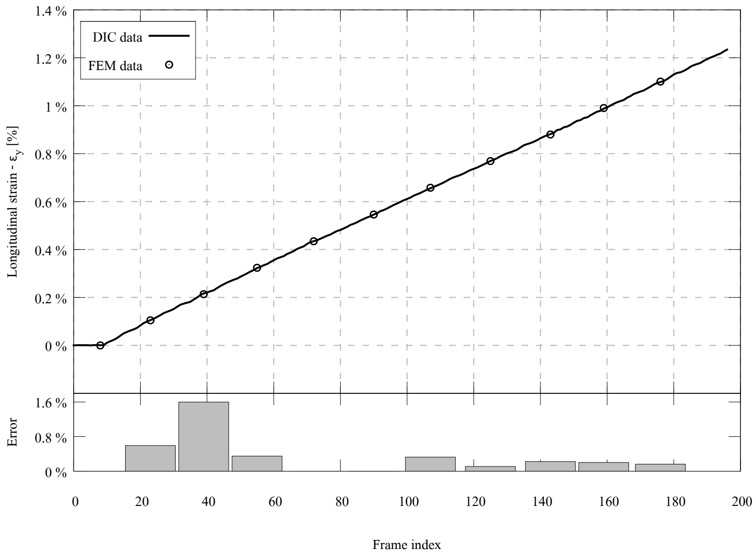


Figure 4.27: DIC vs. FEM average strain data

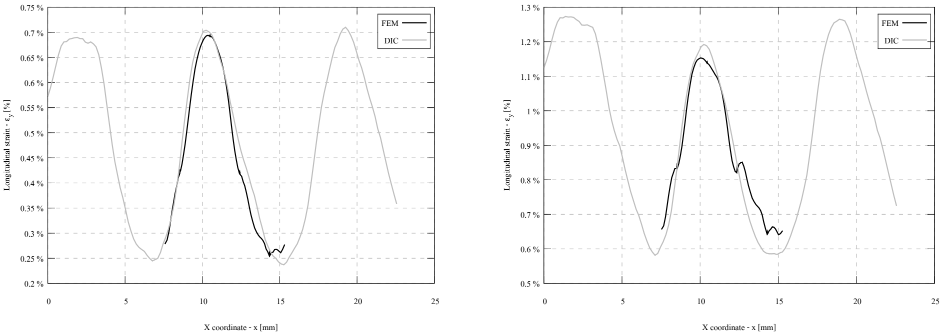
for example shown also in Figure 4.22). In the same plot the homogenized value of longitudinal strain obtained from the finite element model has been added (it is important to notice that the *frame index* axis is meaningless for FE model). The x-axis position of the FE data reveals the DIC frame with closer strain values. The error between FE results and DIC results at the selected time instants (that cannot be exactly the same) are shown in the same graph.

Table 4.4: DIC – FEM matching

FEM increment	DIC frame	Difference [%]
0	8	0.00%
1	23	0.59%
2	39	1.60%
3	55	0.35%
4	72	0.00%
5	90	0.00%
6	107	0.33%
7	125	0.11%
8	143	0.22%
9	159	0.20%
10	176	0.16%

The Table 4.4 shows matching between DIC frames and FEM increments coupled also with the percentage difference between the average strains. The values are quite good (the highest is just 1.6%) because of the high number of DIC frames (195 frames for 1.2% of final failure strain) and so the maximum error is due to a strain difference of half of the distance between two consequent frames that is $\frac{1}{2} \cdot \frac{1.2\%}{195} \approx 0.00308\%$. This leads to an error of roughly 1.6% as here obtained for the third value (FEM increment 2).

A comparison between FE and DIC results can be done by selecting some time instants and define a transverse line at a specified y coordinate and compare the results from FE model and from DIC calculations. According to the coordinate system of Figure 4.26 the x axis of the DIC images corresponds to the y_{FEM} axis of the FE model. Moreover, according to how the model has been built and the specimen has been mounted on the testing machine, the observed surface is the one at negative values of z_{FEM}.



(a) $\langle\epsilon_y\rangle = 0.32\%$ (b) $\langle\epsilon_y\rangle = 0.66\%$

Figure 4.28: DIC vs. FEM comparison of the values along the x axis

Figure 4.28 shows the comparison between DIC and FE at two different strain levels. As the field of view was enough big to observe all the specimen width and the analyzed area was just few millimeters smaller (to avoid border effects), DIC results are able to complete almost three different unit-cells and so it is possible to see three times the behavior of the single unit-cell that actually is shown by FE results (black line in the graph). The well matched results demonstrate the ability of the method to reveal local strains with high precision.

4.5.3 Off-axis tension of CFRP

The complex architecture of woven composites leads to a number of studies related to the damage initiation and propagation, even when the testing conditions are clear and simple. When there is an angle θ between the loading direction and the longitudinal yarns direction the deformation map of those materials is even more complex.

For an angle θ of zero, namely loading conditions like in Section 4.5.2, the load is mainly borne by the longitudinal yarns. For low loads these yarns are crimped according to the woven architecture, but the more the load is increasing, the less is the crimp ratio as the load tends to unbend and to straighten them. The crimp of transversal yarns increase in order to let the longitudinal yarns to deform. This process is redistributing the load, giving even more load to the longitudinal yarns. This behavior is shown in Figure 4.18, where the presented curves have sub-linear trend.

When the angle θ is not zero (where a negative value will have the same effects of a positive value due to the obvious symmetric behavior), the behavior of the material is quite different. First of all as the long warp yarns are not anymore parallel to the side edges of the specimen. It means that the warp starting from the lower tab may end to the right or left side of the specimen before reaching to the upper tab. This important geometric characteristic cause high strains inside the material even if just tensile loads are applied. In fact the effect of yarns that cannot completely bear the load tends to make them rotating and shearing one over the other. This effects lead to a more complex strain distribution and a much weaker response of the material, that needs to be investigated. For this purpose specimens with three different angles θ were cut and tested. The three angles are 0°, 30° and 45° respect to the loading direction. The experimental set-up is the same in all test configurations, as depicted in Figure 4.16.

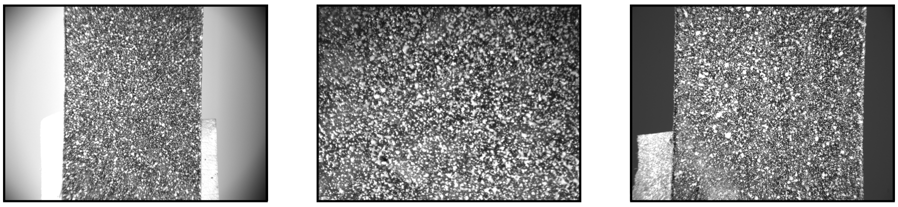

(a) 0° specimen (b) 30° specimen (c) 45° specimen

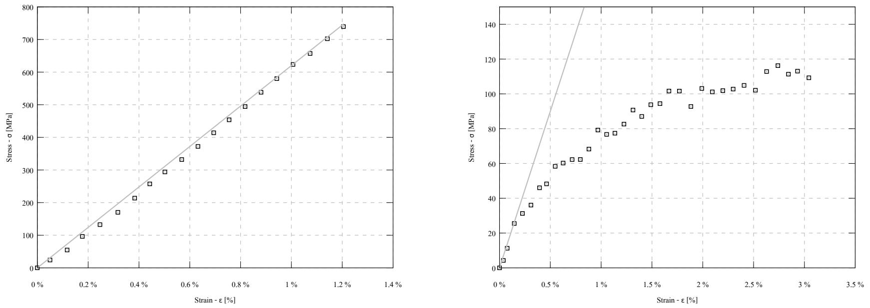
Figure 4.29: Field of view of the three specimen configurations for off-axis tensile tests

The DIC set-up is mainly the same in all tests, except that for the specimens of 30° the field of view is smaller respect to the other two configurations. In order to reduce the effects of small aperiodicity of the material the bigger area provided by 0° and 45° configurations are more preferable as these two configurations present a bigger field of view (see Figures 4.29(a) and 4.29(c)). For 30° the area of interest is smaller. Therefore it is possible to have more localized strain information (see Figure 4.29(b)).

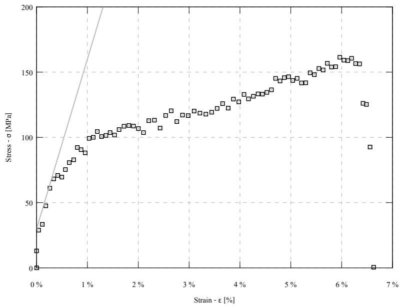
A first result can be a macroscopic information on the strain evolution of the specimen, done by averaging the strain data over the area of interest. This can be an important result as it can give useful information about the effect of off-axis loading of CFRP laminates, especially the elastic modulus along the loading direction. An observation of the specimen surface followed by DIC strain analysis and synchronization of these data with stress information coming from tensile testing machine lead to the graph of Figure 4.30.

It is obvious that the stress-strain curve of the specimen with 0° configuration is almost a straight line (or sub-linear). This has been discussed in Section 4.5.2.

The specimen with 45° configuration has a similar field of view and therefore it has similar strain distribution. However the stress-strain curve is different from the 0° configuration. All the warp yarns starting from one tab cannot reach to the other tab. Because the distance between the tabs is 180 mm and the width of the specimen is 25 mm and this means that the minimum



(a) 0° specimen (b) 30° specimen



(c) 45° specimen

Figure 4.30: Stress-strain curves of tensile tests on three different oriented specimens

angle to have a complete disjointing is about 8°. Moreover in central areas, enough far from the borders, the 45° configuration under tension leads to a localized pure shear loading condition. In fact the shear stresses and strains are preponderant. The behavior of the material is mainly due to the mechanical properties of the polymeric matrix and the elongation before final failure is very high. Actually in some cases it is even difficult to define a final failure, as after several centimeters of displacement of the crossheads the material looks non damaged (but it has lost its strength and it is highly damaged).

As shown in Figure 4.30(c), at mean strain value of about 2% the strength of the specimen has reached to the value of 100÷110 MPa with a clear non-linear behavior. After this value the material is mainly damaged and just the matrix plus the pullout resistances[5] are bearing the load applied to the specimen. A sub-linear stress-strain curve with a slope of 1455 MPa is continuing till 160 MPa and a strain of 6.3% which is the final failure point.

The specimen with 30° configuration was observed by a higher magnification to reveal better the intra-yarns heterogeneities. The field of view has a size of about 9x6.7 mm^2. Thus each pixel has a size of 6.45 μm. Such magnification leads to a smaller observable elongation by the digital camera that is obviously fixed during the test. Therefore the achieved results last till an average

---

[5] Pullout effect is mainly related to the friction between yarns and the ability of the fabric to absorb energy. This is especially important in ballistic applications [69]

strain of 3% (see Figure 4.30(b)). The same as the previous specimen (45° configuration), the warp yarns starting from a tab cannot reach the other tab. Here the elongation before final failure is again very high and the maximum force is considerably lower than the case of 0°. The strength of the material is slightly higher then the specimen with 45° configuration. However it is still quite lower than the specimen with 0° configuration.

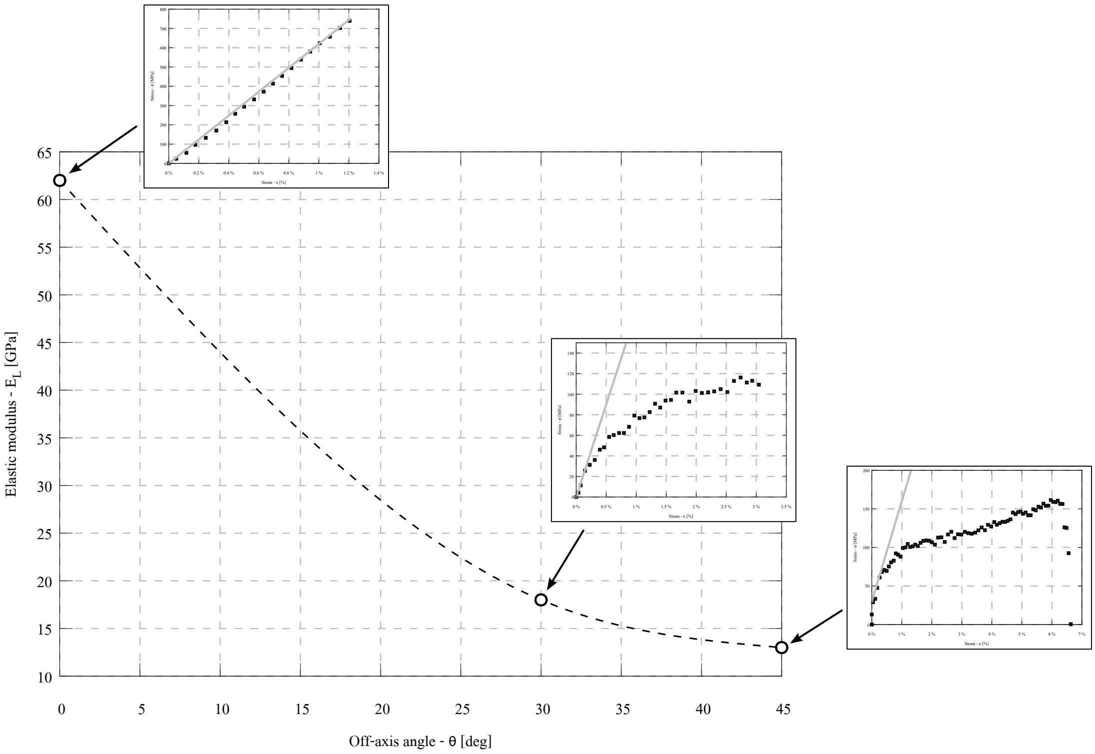


Figure 4.31: Elastic modulus vs. off-axis angle curve

Figure 4.31 shows the trend of the elastic modulus respect to the off-axis angle and the important effects of the off-axis angle on mechanical behavior of the composite material. Even for few degrees the elastic modulus is quite lower for example in 20° configuration the elastic modulus is almost half of 0° configuration. This macroscopic result is very important during the designing process as it gives useful information about the tolerances on geometrical characteristics of the component to the designer.

Skipping the results of 0° configuration because it has already been discussed in Section 4.5.2, hereafter some considerations about the strain maps of the other two configurations will be discussed. In particular the capabilities of DIC method will be considered.

In Figure 4.32 the strain map evolution of the specimen with an off-axis angle of 45° is shown. The graph in the middle shows the stress-strain curve from the beginning of the experiment up to the time that the area analyzed by DIC moves outside the camera view. Here again four time

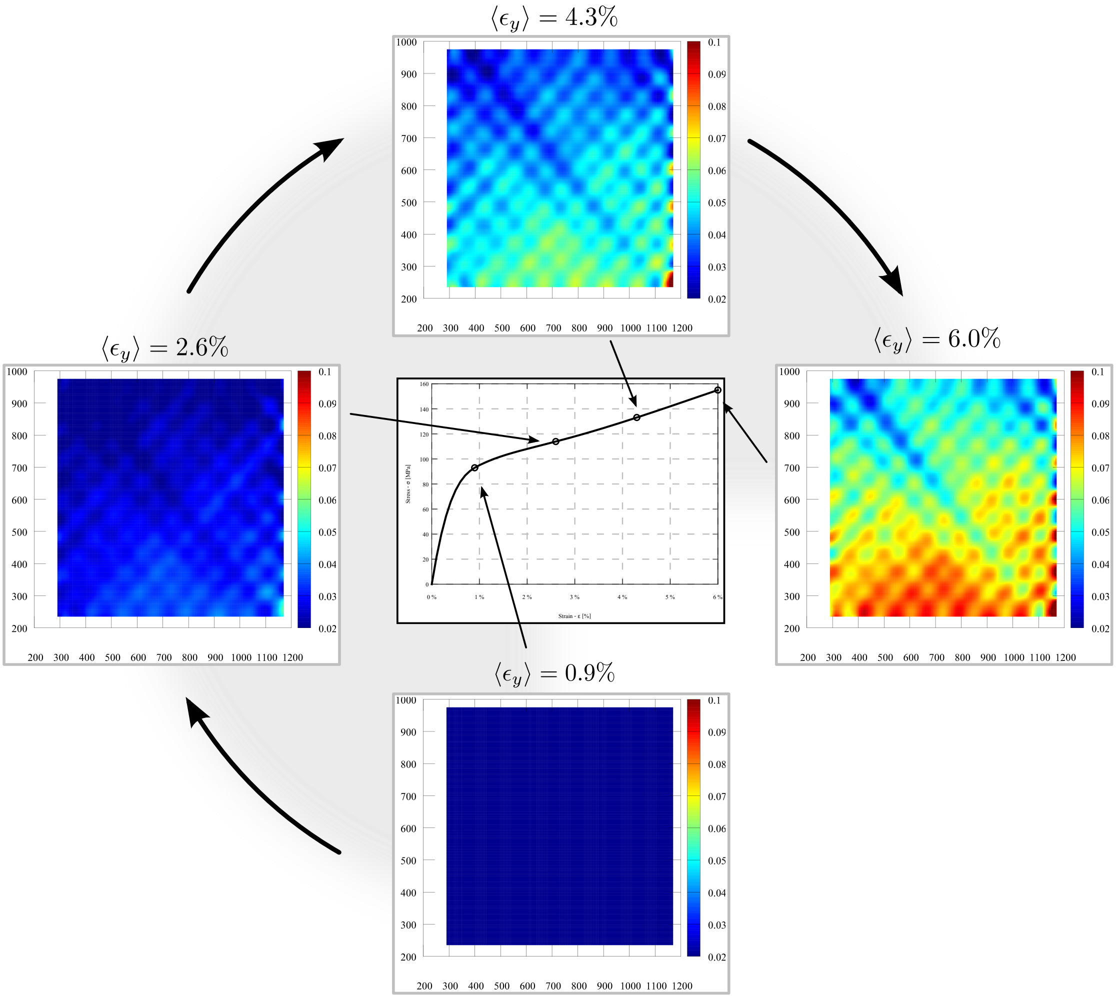


Figure 4.32: Strain evolution of a 45° oriented CFRP specimen, from early stages till final failure

instants have been selected and the strain maps have been calculated. It is possible to see the time evolution of the strains (that are increasing during the test) from the four strain maps. Moreover due to the particular configuration of the yarns and the loading conditions, the strains are also localized in specific areas of the specimen. Keeping the same strain limits on the color scale of these four images, it is clear where these strains occur.

4.6 Evaluation of the software

After presentation of many results dealing with different materials, here the capabilities of the software will be verified with some more technical comparisons are using some objective criteria to prove the reliability of the software.

Therefore two groups of comparisons are presented:

- a comparison with artificial created strain field,
- a comparison with a commercial software.

4.6.1 Artificial strain fields

In order to test the capabilities of the proposed software, the creation of artificial images with an imposed strain field would help. This can be done mainly in two ways: deforming the real images or creating an artificial images. The first approach can be difficult as it could be difficult do really deform according to user-defined strain field. Moreover the quality of the image would play an important role in the capability of the software to understand the field correctly.

The second approach seams to be more reliable and powerful as can easily create different kind of deformations or strain fields and any kind of images. In order to create images with an imposed deformation and also with random distribution of gray-scale values (to simulate white and black dots on the specimen surface) a *procedural texture*[6] should be used. In particular a pseudo-random appearance noise distribution is preferable because it allows the application of strain fields without losing the correspondence of the colors. In fact a pseudo-random distribution just looks like random. However at a given coordinate, even with many runs of the pseudo-random generator, when there is no applied strain field the generated color for one pixel will be always the same.

The Perlin noise function fulfills all the aforementioned constrains. It was developed in 1983 by Ken Perlin to give a natural looking to computer effects in the film Tron by Walt Disney Pictures. Since that time it has been used in many occasions. Skipping the mathematical description, it is just important to know that Perlin noise is based on the sum of functions describing the color of the image, at different frequencies (i.e. at different level of detail as big frequency means coarse description and low frequency means many details). The final result appears in Figure 4.33.

Deforming Figure 4.33 according to the following function

$$f(x, y) = x + a \cdot p \cdot sin\left(2 \cdot \pi \cdot \frac{x}{p}\right) \tag{4.39}$$

will lead to a image with a sinusoidal strain field along x (and constant along y). Figure 4.34 shows the comparison of the artificially created image with *Phoenix*© results. As it is obvious from this figure, the correspondence between these two series of data is very high. In the particular proposed example the result has been a series of values with the mean of $-1.4 \cdot 10^{-4}$ pixels and a standard deviation of $67 \cdot 10^{-4}$ pixels, this is a good evidence for the precision of *Phoenix*©

---

[6] A procedural texture is an artificial image created using an algorithm aimed to obtain a realistic representation of natural elements.

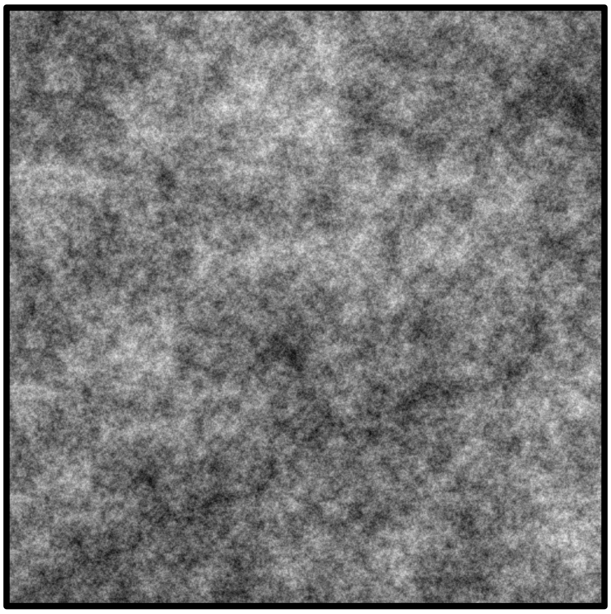


Figure 4.33: Example of image created by Perlin noise algorithm

software, as one ten-thousandth pixel is much lower than the smaller movement measurable by this technique.

In an other attempt and in order to understand the limit of the software, a set of images created using a translational function (in the form of $f(x, y) = x + k$) have been analyzed. The analysis always starts from the same original image and not from any deformed image (i.e. Figure 4.33). The function moves the field only along the x axis of a certain amount of pixels. The mean value and the standard deviations are listed in Table 4.5.

Table 4.5: Mean and standard deviation of different test at small translations

Translation [pixel]	Mean value [pixel]	Standard deviation [pixel]
1	1.000	0.00042
0.1	0.120	0.0031
0.01	0.0135	0.0012
0.001	0.0014	0.00046
0.0001	0.00014	0.00015

The results of Table 4.5 are obtained with a completely flat field, i.e. a constant value. In a more realistic case the displacements are very small and also changing from point to point. In order to understand how the software may react to this scenario different sinusoidally deformed images have been created and analyzed (i.e. functions like in Equation (4.39) but where a assumes different values). The results are shown in Figure 4.35.

Comparing Figure 4.35(a) and Figure 4.34, these two figures are very similar to each other and

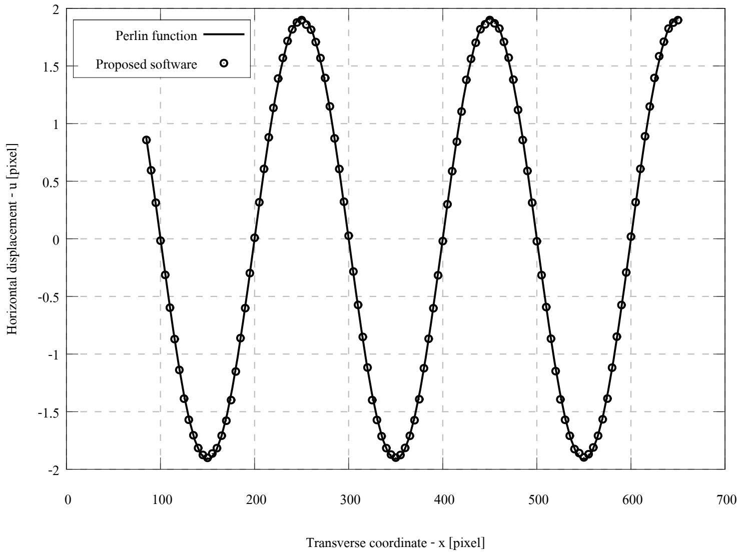


Figure 4.34: DIC analysis on images created by Perlin noise algorithm

just the amplitude is one half in graph of Figure 4.35(a) respect to Figure 4.34. In Figure 4.34 the field is well followed, apart a small difference at the peaks of the sinusoidal curve. Figure 4.35(b) starts to show a higher difference, again at the function peaks. Anyway the shape is still followed and the errors are still relatively low. This behavior is even more visible in Figure 4.35(c) where an irregular amplitude appears as well. The last graph, Figure 4.35(d), shows an extremely irregular distribution of the values that are following the sinusoidal shape. Considering very small applied amplitude, even this result is a good evidence for *Phoenix*© abilities.

Introducing an error parameter Ψ that is calculated as follows:

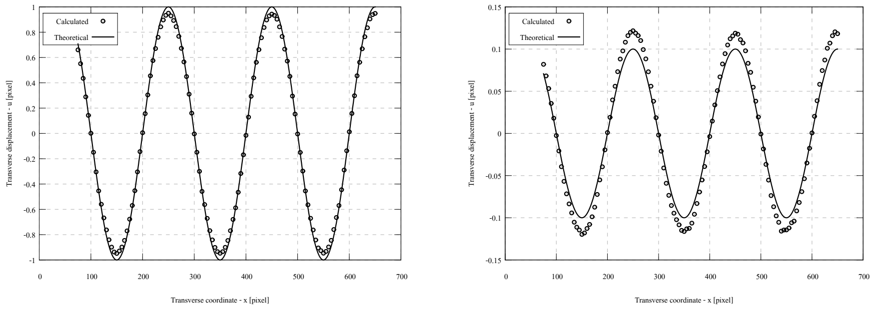
$$\Psi = \sum \frac{1}{a^2} \cdot \left[u_{DIC}(x) - a \cdot sin \left(2 \cdot \pi \cdot \frac{x}{p} \right) \right]^2 \tag{4.40}$$

It is possible to define the error for each of the four analyzed cases (a is the amplitude and p is the period). The resulting values are presented in Table 4.6.

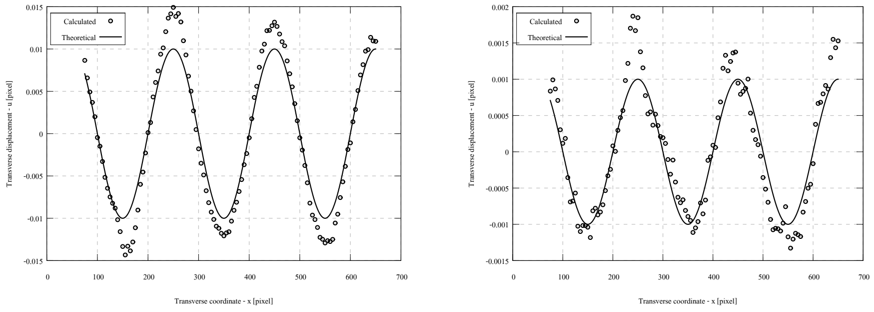
Table 4.6: Error parameter Ψ for different amplitude values

Amplitude a [pixel]	Ψ
1	0.16
0.1	2.2
0.01	5.4
0.001	10.4

Similar results can be achieved by a bi-dimensional displacement function.



(a) Amplitude of 1 pixel (b) Amplitude of 0.1 pixels



(c) Amplitude of 0.01 pixels (d) Amplitude of 0.001 pixels

Figure 4.35: Tests of the reliability of the software at very small displacement fields

4.6.2 Vic-2D

Artificial images are very good as they allow the user to create easily even very complex displacement and strain fields and check the quality of the algorithm or the proposed software. However in this case errors in the illumination, specimen mounting, not good painting, etc. are not considered. This may lead to erroneous results. In order to test the software in a more irregular environment, i.e. where the images are not as perfect as when they are created by a function, the authors decided to compare the results of *Phoenix*© with a commercial software. The software here used is Vic-2D from Correlated Solutions Inc., a company from South Carolina (USA) leader in the development of non-contacting measurement solutions (see `http://www.correlatedsolutions.com`).

The company produces software in the field of laser shearography, videostroboscope systems and also digital image correlation. For this last technology the company offers a solution for bi-dimensional measurements, Vic-2D, and a solution also for three-dimensional measurements, Vic-3D. This is a tool that uses two cameras to measure the object shape, displacements and full-field strains in three dimensions.

The proposed software for two-dimensional solution is Vic-2D. On the website of the company it is stated that this software is able to measure strains from 500 microstrains till 500% and from a specimen size of 1 mm to 10 m. No special illumination or lasers are required.

The comparison is based on few analyses of a CFRP specimen as tested also in Section 4.4.2 but here the optics used is the one of Correlated Solutions Inc. In fact the acquisition is made by Vic-2D and than the same images are used by *Phoenix*©. A image is shown in Figure 4.36 in order to test the algorithm and not the best parameters set used, the analysis has been performed trying to keep as much parameters as possible equal between the two software. The number of sub-windows was more than 3000, with a size of 29x29 pixels.

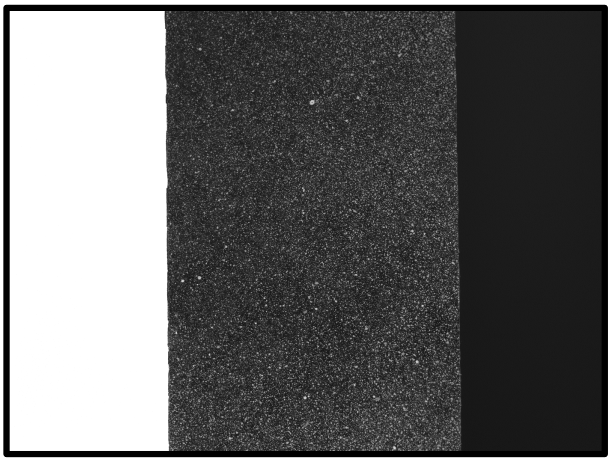


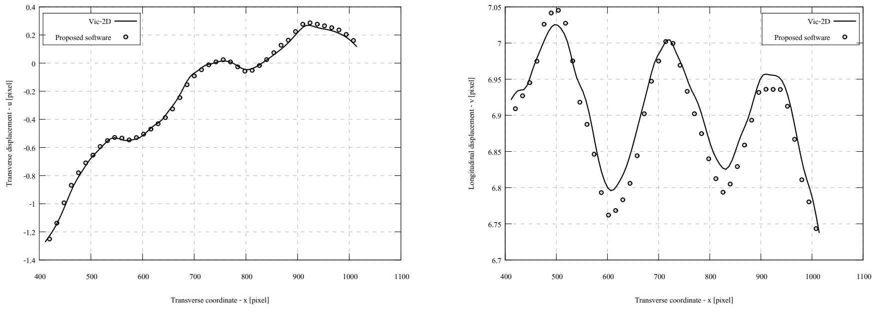
Figure 4.36: Pictuyre taken by Vic-2D on a CFRP specimen

As the distribution of strain is more or less regular and repetitive, a line almost at the middle of the specimen was selected and the data of displacements and strains are evaluated and plotted as shown in Figure 4.37. Figure 4.37(a) shows the displacement along the x axis. The values from Vic-2D or *Phoenix*© software are considerably consistent with each other. Figure 4.37(b) which shows the displacement along loading axis y reveals a higher difference respect to Figure 4.37(a).

After the calculation of the strains (Figure 4.37(c) shows the transverse strains and Figure 4.37(d) shows the longitudinal strains), in both cases using the Lagrangian approach and by smoothing, the differences are more noticeable but still small, apart from the left and the right borders which are revealing a higher difference maybe due to the smoothing process.

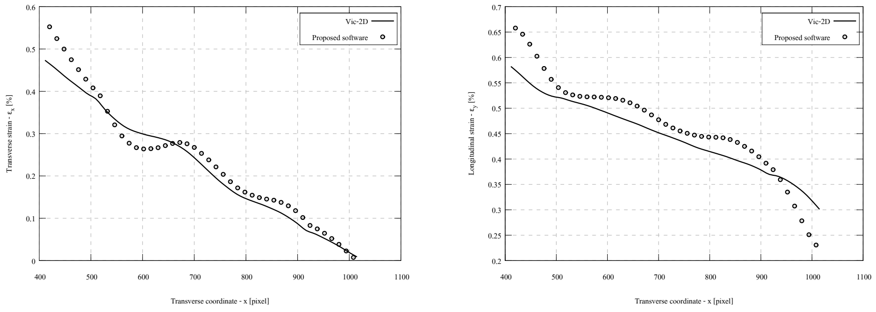
4.7 Digital image correlation as validation tool

In common design applications the best solution is when performance meets simplicity and low cost. In these situations the designer uses frequently numerical simulations to find a good solution, but using his experience in guiding the final result. Usually the experiments are reduced as much as possible for the high cost. When the performance is the key task of the design, the cost will get less important. Sportive cars, aircrafts, medical devices and many other products are mainly just performance defined. In such applications the cost of the experiments is excused



(a) Transverse displacement

(b) Longitudinal displacement



(c) Transverse strain

(d) Longitudinal strain

Figure 4.37: Comparisons between Vic-2D and the proposed software

by high performance goal. However the advantages of numerical simulations are also proved in these areas.

When a testing campaign is designed, several specimens must be prepared and tested. Different costs such as the costs of material, the preparation of the specimens, time to set the testing machine, time needed to perform the test, etc. must be considered. In order to reduce these costs it is possible to carefully decide the number and kind of performed tests. To do so a well known approach is based on the so called *design of experiments* (DOE) [70–72]. The DOE is an experimental technique that helps to investigate the best combinations of parameters, changing quantities, levels and combinations to obtain results statistically reliable, but even with reducing the number of experiments the costs remain usually very high.

On the other hands the numerical simulations may allow the designer more effective design approaches. In fact it is common to implement some sort of automatic optimization using some “clever” software able to modify the starting assumptions (e.g. geometry, material properties, etc.) according to the results of the simulations [73, 74]. Automate the design process there is the risk to lose a good understanding of the process and achieving bad results.

Moreover when the numerical model gets complex the number of parameters and their relations usually get uncontrollable. In such situation the model must be experimentally validated. When macroscopic information are enough a normal tensile test or the use of a extensometer can be effective (see Section 2.1). A more localized value of stress (or strain) can be measured by the use of strain-gages (see Section 2.2). Other techniques are needed if full-field information must be measured because of the high heterogeneity of the tested material or because of the high value of the deformation gradients (see Section 2.3). In this last category can be found the digital image correlation. DIC can be an effective tool for validation purposes of numerical analysis. This technique has been used to validate finite element models of textile composite materials [75–78]: It is also used to validate complex biological materials [79], to define mechanical properties (even in dynamic situations) [80, 81] and to validate analytical approaches to the solution of practical situations like as the study of drape-ability of textile composites [82–85], etc.

Chapter

5

Numerical models of composite materials

Since early eighties the scientists have tried to focus more on composites as new but widely applicable materials. To evaluate the mechanical properties of composites different analytical approaches have been proposed since the beginning, using two-dimensional [86–88] or three-dimensional [89, 90] descriptions. Since 1996 the improvements in computer hardware and numerical simulation software made the study on small and simple (but effective) finite element models of composite materials possible [91–93].

In 1996, at the University of Parma (Italy), Riva [94] started to work on the numerical modeling of woven composite materials, as they were getting more interesting and applicable than other kinds of composite materials and have received less attention up to that time [95]. The studied model was on a plain-weave textile composite built using three-dimensional bricks and wedge elements. The materials used were epoxy matrix and graphite-epoxy plain woven fabric reinforcment. The former has been considered elasto-plastic with literature characteristics (see Blackketter et al. [91]). A micro-mechanical finite element model was performed to get the mechanical properties of the reinforcement starting from the properties of its constituents (with square array scheme). Optical microscope observations helped in the definition of the architecture of the composite, see Figure 5.1.

The global stiffness and the stress-strain curve fitted well the experiments. In fact Guagliano and Riva [95] stated that "*[...] the FE approach appears a useful tool for predicting the mechanical response of the material once the elastic constants and the strength of the constituent materials are known [...]*". Subsequently Nicoletto and Riva [3] presented a work on twill-weave architecture. Again they presented some numerical results compared with experimental results, focusing mainly on the damage initiation and evolution, describing where and how it starts and grows. In fact in those years many papers started to deal with interesting behavior of composite materials. Contributions such as the work of Zako et al. [2] started to become a standard in the definition of damage in finite element approaches (see Section 5.2.4). Later the geometric structure of textile or knitted composite materials started to be studied more and some more

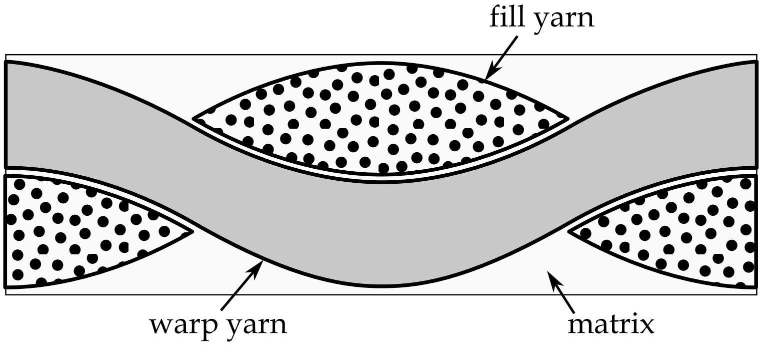


Figure 5.1: Scheme of the cross section of a plain-weave textile composite

or less simplified structures have been proposed. Verpoest and Lomov [96] proposed a software (called *WiseTex*) to define the geometry of complex knitted structures and then with the possibility of convert it in a finite element mesh (with *FETex* or *MeshTex*, developed in collaboration with Politecnico di Milano and Osaka University). Some structures that can be calculated by *WiseTex* are shown in Figure 5.2 [96].

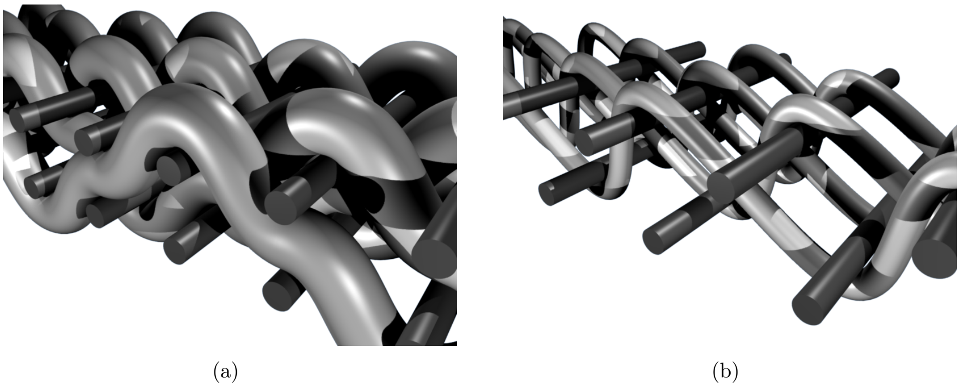

(a) (b)

Figure 5.2: Examples of reinforcement fabrics (realised with Blender)

5.1 Numerical simulation of non-crimp fabrics

Nowadays composite materials are getting a standard in fields like aeronautics, marine, automotive and many other fields thanks to their high performances and particularly their great strength-to-weight ratio. The increasing demand of these materials helps to reduce the cost of design and manufacturing. However the costs are still high for most of industrial interests. In order to produce a unidirectional composite the traditional way is to lay-up pre-impregnated unidirectional tape according to a required stacking sequence of plies in matrix (for example a polymeric resin) and then cure it. This process has several disadvantages, which the most important is less control on directing the reinforcement material inside the ply. To overcome the disadvantages woven composites, which have a easier manufacturing process and so a lower cost are purposed. The drawback of these materials is the lower mechanical properties respect to unidirectional reinforced composites, due to effect of the crimp of the fibers, caused by the weave architecture. However this lost is negligible and besides a woven composite material shows very high mechanical properties in two directions instead of just one direction (the direction of the warp and weft yarns).

A way to solve most of the problems of both kinds of these composites is to use a unidirectional layer made by a stitching process [97]. These materials, called non-crimp fabrics (NCF) as they are not crimped as woven materials and can be easily made in different layers and different angles [98]. This it decreases drastically the cost of ply producing. Unlike what is expressed by the name of these materials, a small crimp cause by the knitting needles (on the plane of plies) deviates the yarns from their perfect UD placement [99]. However this effect is considered negligible. The stitching process produces resin rich zones (also called pockets) that can form channels inside the composite. As outlined by Truong et al. [100] the stitching pattern in the fabric is well correlated with damage evolution pattern, as resin rich pockets may initiate cracks. Subsequently the cracks can follow the stitching pattern. As outlined by Mouritz et al. [101] the stitching process is degrading the in-plane properties but it is giving better out-of-plane properties and more hole-tension and compression-after-impact strength, both properties are very important for the aerospace industry. Similar examples can be made for automotive components. Mattsson et al. [102] proposed a method to study the parameters governing the NCF performances. He showed how the fiber volume fraction is not enough to characterize the material and how the study at the meso-scale is important to characterize the behavior of the composite. Again Mikhaluk et al. [103] revealed the effect of stitching, quantifying it through FE models of quadriaxial NCF. In this study the meso-level models showed a good agreement with experiments.

Regarding to previous studies, modeling of NCF at the meso-scale has become a key task in prediction of the behavior of this material, as it provides the best trade-off between a macro-mechanical model. It is not able to account for the high complexity of the geometry and the heterogeneity of the material, but also it is able to provide a micro-mechanical model that meanwhile needs heavy computational efforts.

Concerning to all the interests and importances of FE modeling of composite materials and the potential of FE numerical modeling in characterization of mechanical properties of material and regarding to the fact that FE modeling can join to DIC analysis to present the mechanical properties of material, this chapter focuses on FE modeling of NCF carbon-epoxy laminate composite as an introduction to join this study to DIC analysis. This study is made in collaboration with the Department of Metallurgy and Materials Engineering (*Metaalkunde en Toegepaste Materiaalkunde*, MTM) of the Catholic University of Leuven (*Katholieke Universiteit Leuven*, KUL) in Belgium.

The present work deals with the modeling of a unit-cell of a non-crimp fabric by the use of

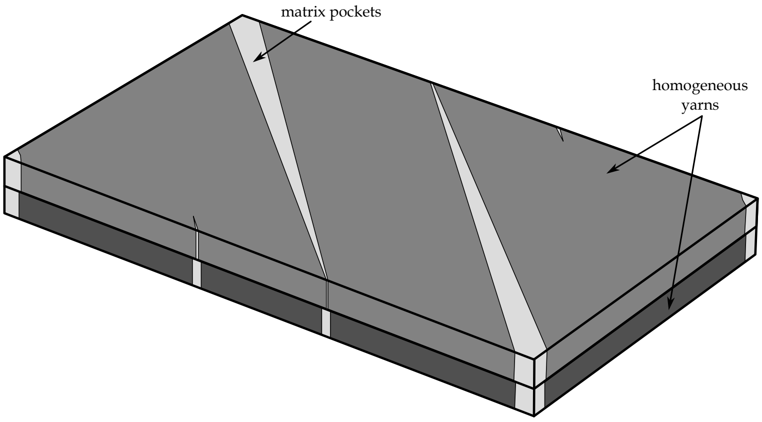


Figure 5.3: Two materials model of a unit-cell of a non-crimp fabric (dark gray the yarns and light gray the pure matrix)

a two-materials model:

- non-linear isotropic and homogeneous matrix,
- non-linear anisotropic with homogeneous yarns and matrix (see Figure 5.3 where dark gray is used for the "homogeneous" yarns and light gray is used for the pure matrix pockets).

In fact in this study the complex architecture of the yarns (namely a bunch of fibers) has been treated as homogeneous material and a special material constitutive law has been developed to take the interactions between yarns and matrix into account.

In order to define the correct behavior of the yarns a special approach has been used according to an interesting paper by Sun and Lau [104] (see Section 5.2). Inelastic anisotropic constitutive law proposed by Sun and Lau needs several parameters to suit the particular material under study. Therefore, in order to obtain a homogeneous behavior at meso-level by studying the heterogeneous behavior at micro-level, several micro-mechanical numerical models have been built and studied (see Section 5.1.1). Mechanical tests have been used to define the mechanical properties of constituent materials of the micro-mechanical models. The resulting flow-chart is shown in Figure 5.4.

5.1.1 Micro-, meso- and macro-mechanical models

To develop a mechanical numerical model the key point is deciding for the required level of detail and precision. Macro-mechanical models for simulations of simple mechanical tests are able to predict the global specimen response. The parameters to build the model are usually the stacking sequence, the geometrical dimensions and the components mechanical properties (e.g. elastic modulus, yield stress, etc.). The validity of such approach is limited to the global behavior of the material. For example usually it ca not take into account localized phenomena like damage evolution or the phenomena which are dependent on the architecture of the composite material.

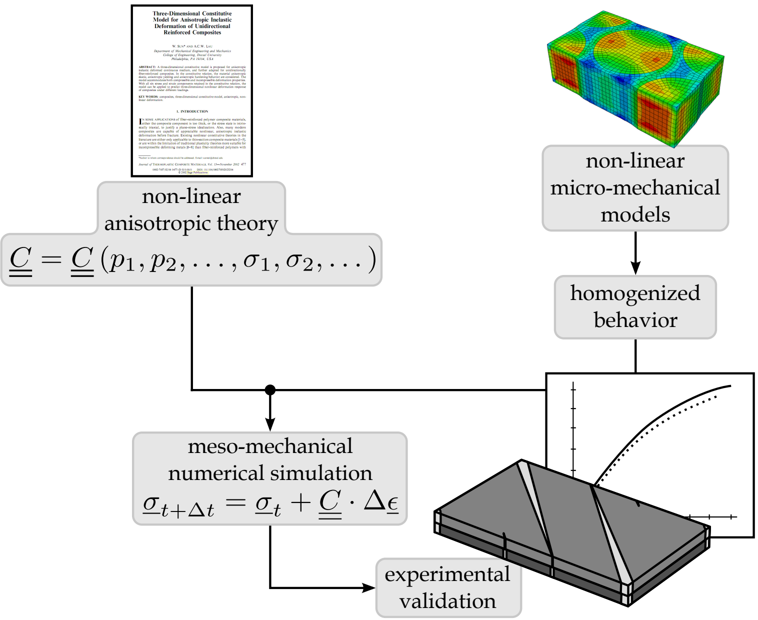


Figure 5.4: Non-crimp fabrics numerical simulation flow-chart

A micro-mechanical model is considered dealing with fibers. Thus usually it works on few micrometers. This scale of magnitude is used to understand the relationship between fibers and between fibers and matrix [105] or to develop finite element micro-mechanical failure criteria [106].

The meso-mechanical model has an intermediate level respect to micro-models and macro-models. It has dimensions smaller than the macro-mechanical model but with less details than the micro-mechanical model. The dimensions are typically around few millimeters or centimeters. The model can present the architecture of the composites (especially for woven composite materials) but it is described by homogeneous yarns and not by single fibers. Therefore the mechanical behavior of the yarns must be known. Lomov et al. [107] presented a detailed road map on the multi-level modeling process, starting from the micro-mechanical model towards the macro-mechanical model. The process needs a homogenization procedure at each step, explained in detail in [108, 109]. The idea is to build a numerical model in a scale and then calculate the global behavior of the model and use it as characteristic of the next level model. In fact if a woven fabric composite has regular distribution of the fibers in the yarns and regular arrangement of the reinforcing yarns in the composites then the problem can be studied by the micro-mechanical theory of heterogeneous periodic materials.

5.2 Basic theory

In most cases non-linear approaches used in literature are accompanied with the limitation of traditional plasticity theory. These non-linear models are mainly suitable for incompressible deforming metals than for fiber-reinforced polymers that have an anisotropic non-linear behavior before fracture [104]. To overcome the problems and limits due to a plasticity theory or the use of a single isotropic hardening function and in order to develop a simple but effective framework, Sun and Lau [104] proposed a mathematical approach to non-linear anisotropic problems of continuous materials. The proposed model accommodates compressible or incompressible deformation properties and retains all possible stress and strain components. Moreover the model can predict the global response in the global-local analysis for the problems involving three-dimensional non-linear deformations.

The theory starts from the definition of the stress-strain relationship in a generic material point:

$$\underline{\epsilon} = \underline{\underline{S}} \cdot \underline{\sigma} \tag{5.1}$$

where $\underline{\underline{S}}$ is the compliance matrix and $\underline{\epsilon}$ and $\underline{\sigma}$ represent the true strain and physical stress respectively.

For simplicity it is possible to split the strain in its elastic and its inelastic parts as follows:

$$\underline{\epsilon}^e + \underline{\epsilon}^p = \left(\underline{\underline{S}}^e + \underline{\underline{S}}^p\right) \cdot \underline{\sigma} \tag{5.2}$$

where superscript e and superscript p stand for elastic and plastic respectively. The elastic part of the compliance tensor is constant as is is just a function of the elastic moduli (longitudinal and transverse) and Poisson's ratio while the plastic part is function of the stress tensor. Furthermore it is possible to assume that for each tensor component there is a yield function describing the plasticity behavior of that component. Considering the following:

$$\begin{aligned} \underline{\sigma} &= \{\sigma_1, \sigma_2, \sigma_3, \sigma_4 = \tau_{12}, \sigma_5 = \tau_{13}, \sigma_6 = \tau_{23}\}^T \\ \underline{\epsilon} &= \{\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4 = \gamma_{12}, \epsilon_5 = \gamma_{13}, \epsilon_6 = \gamma_{23}\}^T \end{aligned} \tag{5.3}$$

then six yield functions are required:

$$f_i = \Lambda^{q_i} \qquad i = 1, \ldots, 6 \tag{5.4}$$

where q_i are exponents and Λ is a quadratic function of current stresses, define as:

$$\begin{aligned} \Lambda =& A_{11}\sigma_1^2 + A_{22}\sigma_2^2 + A_{33}\sigma_3^2 + A_{44}\sigma_4^2 + A_{55}\sigma_5^2 + A_{66}\sigma_6^2 + \\ &+ 2A_{12}\sigma_1\sigma_2 + 2A_{13}\sigma_1\sigma_3 + 2A_{23}\sigma_2\sigma_3 \end{aligned} \tag{5.5}$$

where A_{ij} are numerical coefficients. It must be noticed that the terms with odd power of shear stresses are omitted since the material responses equally to positive and negative shear.

Considering the flow rule it is possible to write the plastic part of the i^{th} component of the strain tensor:

$$\epsilon_i^p = \lambda_i \cdot \frac{\partial f_i}{\partial \sigma_i} \tag{5.6}$$

and introducing the yield function as in Equation (5.4):

$$\epsilon_i^p = \lambda_i \cdot \frac{\partial f_i}{\partial \sigma_i} = \lambda_i \cdot \frac{\partial f_i}{\partial \Lambda} \cdot \frac{\partial \Lambda}{\partial \sigma_i} = \lambda_i \cdot q_i \cdot \Lambda^{q_i - 1} \cdot \frac{\partial \Lambda}{\partial \sigma_i} = \sum_{j=1}^{6} S_{ij}^p \cdot \sigma_j \tag{5.7}$$

where it is possible to extract a suitable definition for S_{ij}^p considering also that must be satisfied that $S_{ij} = S_{ji}$. Thus it follows:

$$S_{ij}^p = \left(\lambda_i \cdot q_i \cdot \Lambda^{q_i - 1} + \lambda_j \cdot q_j \cdot \Lambda^{q_j - 1}\right) \cdot A_{ij} \tag{5.8}$$

In order to define the values of the coefficients, an experimental approach must be defined.

A Ramberg-Osgood [110] law can be a suitable tool to build a relationship between strain and stresses as it can take into account all the components and with inelastic behavior. The common Ramberg-Osgood equation is as follows:

$$\epsilon_i = \frac{\sigma_i}{E_i} + \frac{1}{E_i}\left(\frac{|\sigma_i|}{\sigma_i^Y}\right)^{n_i - 1} \sigma_i \tag{5.9}$$

where σ_i^Y are the yield stresses, E_i are the elastic moduli and n_i are work hardening parameters.

Equaling the Equation (5.9) with the Equation (5.7) it is possible to define:

$$q_i = \frac{n_i + 1}{2} \tag{5.10a}$$

$$\lambda_i = \frac{\left(\sigma_i^Y\right)^2}{E_i \cdot (n_i + 1)} \tag{5.10b}$$

$$A_{ij} = \frac{\alpha_{ij}}{\sigma_i^Y \cdot \sigma_j^Y} \tag{5.10c}$$

where α_{ij} are interaction parameters with $\alpha_{ii} = 1$.

Considering an application of fiber composite materials and considering an almost linear behavior along the direction of fibers (that here is considered the direction 1), thus $A_{1j} \approx 0$.

Finally the compliance matrix can be written as follows:

$$\underline{\underline{S}} = \begin{bmatrix} \frac{1}{E_1} & -\frac{\nu_{21}}{E_2} & -\frac{\nu_{31}}{E_3} & 0 & 0 & 0 \\ & \frac{1}{E_2} + S^p_{22} & -\frac{\nu_{32}}{E_3} + S^p_{23} & 0 & 0 & 0 \\ & & \frac{1}{E_3} + S^p_{33} & 0 & 0 & 0 \\ & & & \frac{1}{G_{12}} + S^p_{44} & 0 & 0 \\ & sym. & & & \frac{1}{G_{13}} + S^p_{55} & 0 \\ & & & & & \frac{1}{G_{23}} + S^p_{66} \end{bmatrix} \tag{5.11}$$

where

$$S^p_{22} = 2 \cdot \frac{\left(\sigma^Y_2\right)^2}{E_2 \cdot (n_2+1)} \cdot \frac{n_2+1}{2} \cdot \Lambda^{\frac{n_2-1}{2}} \cdot \frac{1}{\left(\sigma^Y_2\right)^2} \tag{5.12a}$$

$$S^p_{23} = \left[\frac{\left(\sigma^Y_2\right)^2}{E_2 \cdot (n_2+1)} \cdot \frac{n_2+1}{2} \cdot \Lambda^{\frac{n_2-1}{2}} + \right. \tag{5.12b}$$

$$\left. + \frac{\left(\sigma^Y_3\right)^2}{E_3 \cdot (n_3+1)} \cdot \frac{n_3+1}{2} \cdot \Lambda^{\frac{n_3-1}{2}} \right] \cdot \frac{\alpha_{23}}{\sigma^Y_2 \cdot \sigma^Y_3}$$

$$S^p_{33} = 2 \cdot \frac{\left(\sigma^Y_3\right)^2}{E_3 \cdot (n_3+1)} \cdot \frac{n_3+1}{2} \cdot \Lambda^{\frac{n_3-1}{2}} \cdot \frac{1}{\left(\sigma^Y_3\right)^2} \tag{5.12c}$$

$$S^p_{44} = 2 \cdot \frac{\left(\sigma^Y_4\right)^2}{E_4 \cdot (n_4+1)} \cdot \frac{n_4+1}{2} \cdot \Lambda^{\frac{n_4-1}{2}} \cdot \frac{1}{\left(\sigma^Y_4\right)^2} \tag{5.12d}$$

$$S^p_{55} = 2 \cdot \frac{\left(\sigma^Y_5\right)^2}{E_5 \cdot (n_5+1)} \cdot \frac{n_5+1}{2} \cdot \Lambda^{\frac{n_5-1}{2}} \cdot \frac{1}{\left(\sigma^Y_5\right)^2} \tag{5.12e}$$

$$S^p_{66} = 2 \cdot \frac{\left(\sigma^Y_6\right)^2}{E_6 \cdot (n_6+1)} \cdot \frac{n_6+1}{2} \cdot \Lambda^{\frac{n_6-1}{2}} \cdot \frac{1}{\left(\sigma^Y_6\right)^2} \tag{5.12f}$$

and Λ can be now easily written as:

$$\Lambda = \left(\frac{\sigma_2}{\sigma^Y_2}\right)^2 + \left(\frac{\sigma_3}{\sigma^Y_3}\right)^2 + \left(\frac{\sigma_4}{\sigma^Y_4}\right)^2 + \left(\frac{\sigma_5}{\sigma^Y_5}\right)^2 + \left(\frac{\sigma_6}{\sigma^Y_6}\right)^2 + 2 \cdot \alpha_{23} \cdot \frac{\sigma_2 \cdot \sigma_3}{\sigma^Y_2 \cdot \sigma^Y_3} \tag{5.13}$$

The compliance tensor of Equation (5.11) describes well the behavior of an anisotropic non-linear material but it needs several parameters to be defined: E_1, E_2, E_3, ν_{12}, ν_{13}, ν_{23}, G_{12}, G_{13}, G_{23}, α_{23}, n_2, n_3, n_4, n_5, n_6, σ^Y_2, σ^Y_3, σ^Y_4, σ^Y_5, σ^Y_6. These 20 parameters are generally not independent. In fact considering the special case of this study, namely yarns made of fiber reinforced plastic composites, the material is transversely isotropic and so one of the directions is perpendicular to a plane of isotropy. Thus it is possible to define a local coordinate system $\{O_{1,2,3}\}$ where for example the direction 1 is the main direction (i.e. longitudinal) and the directions 2 and 3 (i.e. the two transverse directions) are on the plane of isotropy. It can be then defined a simpler coordinate system where $1 \equiv L$ and $2, 3 \equiv T$ and so every parameter along 2 is equivalent to the correspondent parameter along 3. For transversely isotropic materials the Equation (5.13) becomes:

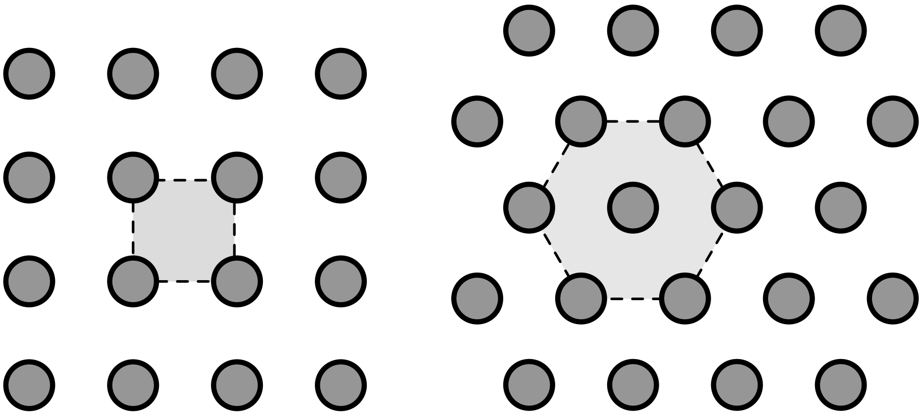
$$\Lambda = \frac{\sigma_2^2 + \sigma_3^2 + 2 \cdot \alpha_{23} \cdot \sigma_2 \cdot \sigma_3}{\left(\sigma_T^Y\right)^2} + \frac{\sigma_4^2 + \sigma_5^2}{\left(\tau_L^Y\right)^2} + \left(\frac{\sigma_6}{\tau_T^Y}\right)^2 \tag{5.14}$$

and the number of independent parameters is 13: E_L, E_T, ν_{12}, ν_{23}, G_T, G_L, α_{23}, n_2, n_4, n_6, σ_T^Y, τ_L^Y, τ_T^Y.

5.2.1 Description of the geometry

The geometry of a micro-mechanical model must comply with the complex micro-structure of the material. A yarn made of fibers and matrix can be very complex considering the number of fibers (few thousands) and the irregular distribution inside the yarns. Thus the difficulties of the modeling could be unsurmountable. Considering a random distribution anyway can be simplified as a media with statistical periodicity of the behavior at the micro-level and so from a statistical point of view can be considered as made of a repetition of a representative volume, a micro-mechanical unit-cell.

There are different ways to make a unit-cell. In Figure 5.5 are shown the two kinds mostly used. Considering a square array of fibers in a continuous matrix is a good choice to get orthotropic materials. In fact a square array has three mutually perpendicular planes of symmetry. A better choice can be an hexagonal array of fibers as it is closer to a random distribution of fibers and it has an axis of symmetry along the fiber direction. This is like a transversely isotropic formulation [111].



(a) Square array (orthotropic materials)

(b) Hexagonal array (transversely isotropic materials)

Figure 5.5: Examples of square and hexagonal fibers arrays

A simple geometry to study the hexagonal array is made of a rectangle with the ratio of $1 : \sqrt{3}$ between the edges (see Figure 5.6). Rectangular shape is preferable as makes easy to apply the boundary conditions. Starting from this shape is possible to vary the volume fraction V_f of fibers inside the yarns by varying the radius r of the fibers respect to the dimensions of the unit-cell. From Figure 5.6 it is possible to write:

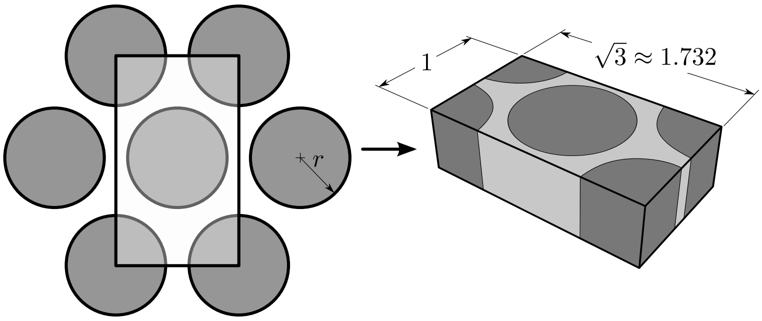


Figure 5.6: Extraction of a rectangular scheme by an hexagonal array

$$r = \sqrt{\frac{\sqrt{3}}{2 \cdot \pi} \cdot V_f} \tag{5.15}$$

where r is directly proportional to the real dimension of the short edge of the rectangle. If for example the rectangle has dimensions 4×6.93 mm^2 and the material has a volume fraction of 70%, then the radius of the fibers must be 1.76 mm.

5.2.2 Unit-cell approach

Building a finite element model of heterogeneous media can be quite difficult if no semplificative hypothesis have been considered or can be developed. In the case of textile composites it can be considered that:

- there is regular distribution of fibers inside the yarns,
- there is regular arrangement of yarns inside the composites.

In this situation they can be considered heterogeneous periodic materials, and so as they are periodic, just one period can be studied. The unit cell in fact is the smallest volume (or area if in bi-dimensional space) that represents, if thought repeated in all the directions in a three dimensional space, the behavior of the entire material. The unit-cell can be easily found by observing the periodic architecture of textile composite materials and finding the smallest repetitive scheme, as shown in Figure 5.7 for a twill-weave woven composite architecture.

Applying appropriate periodic boundary conditions to the unit cell, the numerical model will behave as surrounded by infinite number of cells. An interesting approach has been proposed by

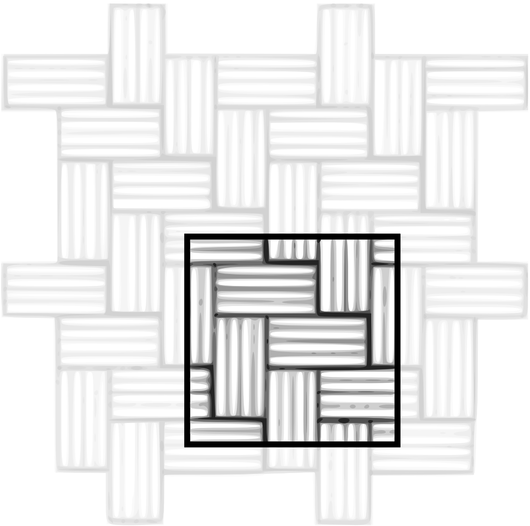


Figure 5.7: Scheme of the periodic unit-cell of a twill-weave woven composites

Carvelli and Taliercio [108], Carvelli and Poggi [109] where the displacement field due to a small strain is as follows:

$$\underline{u}\left(\underline{x}\right) = \underline{u}_0 + \left(\underline{\underline{\Omega}} + \underline{\underline{E}}\right)\cdot\underline{x} + \underline{\tilde{u}}\left(\underline{x}\right) \tag{5.16}$$

where $\underline{\underline{\Omega}}$ is the anti-symmetric tensor related to small rigid body rotations, $\underline{u}_0$ is the rigid body translation of the unit-cell, $\underline{\underline{E}}$ is the macroscopic strain tensor and $\underline{\tilde{u}}$ is the microscopic strain with periodicity equal to the volume of the unit-cell.

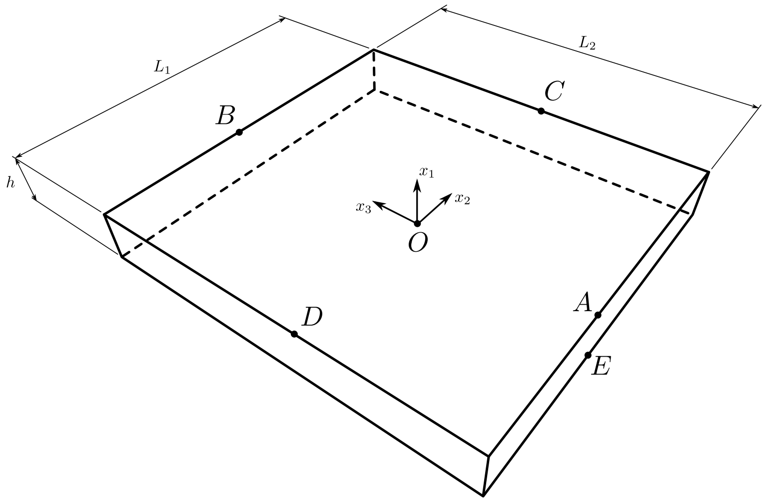


Figure 5.8: Periodic boundary conditions of a generic unit-cell

Considering a scheme as in Figure 5.8 it is possible to ensure periodic conditions applying the followings:

- zero rigid body translations: $\underline{u}_O = 0$
- zero rigid body rotations: $\underline{\underline{\Omega}} = 0$
- periodicity along x_1: $\underline{u}_A - \underline{u}_E = \underline{u}_p - \underline{u}_e$
- periodicity along x_2: $\underline{u}_C - \underline{u}_D = \underline{u}_c - \underline{u}_d$
- periodicity along x_3: $\underline{u}_B - \underline{u}_A = \underline{u}_b - \underline{u}_a$

where a is the generic point on the surface $x_3 = -L_2/2$, b is the generic point on the surface $x_3 = L_2/2$, c is the generic point on the surface $x_2 = L_1/2$, d is the generic point on the surface $x_2 = -L_1/2$, p is the generic point on the surface $x_1 = 0$ and e is the generic point on the surface $x_1 = -h$.

To ensure the second point (zero rigid body rotations) it is possible to define the tensor of strains and rigid displacements ($\underline{\underline{\Omega}} + \underline{\underline{E}}$) and set the anti-symmetric part of it[1] to zero. Therefore the tensor of strains and displacement, $\underline{\underline{\Psi}}$, is defined as follows:

$$\underline{\underline{\Psi}} = \underline{\underline{\Omega}} + \underline{\underline{E}} = \left.\frac{\partial u_i}{\partial x_j}\right|_{i,j=1,2,3} = \begin{bmatrix} \dfrac{u_1^A - u_1^E}{h} & \dfrac{u_1^C - u_1^D}{L_1} & \dfrac{u_1^B - u_1^A}{L_2} \\ \dfrac{u_2^A - u_2^E}{h} & \dfrac{u_2^C - u_2^D}{L_1} & \dfrac{u_2^B - u_2^A}{L_2} \\ \dfrac{u_3^A - u_3^E}{h} & \dfrac{u_3^C - u_3^D}{L_1} & \dfrac{u_3^B - u_3^A}{L_2} \end{bmatrix} \tag{5.17}$$

then to avoid rigid rotations it is possible to write:

$$\frac{\underline{\underline{\Psi}} - \underline{\underline{\Psi}}^T}{2} = \frac{1}{2} \cdot \begin{bmatrix} 0 & \dfrac{u_1^C - u_1^D}{L_1} - \dfrac{u_2^A - u_2^E}{h} & \dfrac{u_1^B - u_1^A}{L_2} - \dfrac{u_3^A - u_3^E}{h} \\ -\dfrac{u_1^C - u_1^D}{L_1} + \dfrac{u_2^A - u_2^E}{h} & 0 & \dfrac{u_2^B - u_2^A}{L_2} - \dfrac{u_3^C - u_3^D}{L_1} \\ -\dfrac{u_1^B - u_1^A}{L_2} + \dfrac{u_3^A - u_3^E}{h} & -\dfrac{u_2^B - u_2^A}{L_2} + \dfrac{u_3^C - u_3^D}{L_1} & 0 \end{bmatrix} = 0 \tag{5.18}$$

that is equivalent to the following system of equations:

$$\begin{cases} \dfrac{u_1^C - u_1^D}{L_1} = \dfrac{u_2^A - u_2^E}{h} \\ \dfrac{u_1^B - u_1^A}{L_2} = \dfrac{u_3^A - u_3^E}{h} \\ \dfrac{u_2^B - u_2^A}{L_2} = \dfrac{u_3^C - u_3^D}{L_1} \end{cases} \tag{5.19}$$

---

[1] It is always possible to extract from a square matrix an anti-symmetric and a symmetric part: $A = \frac{1}{2} \cdot \left(A + A^T\right) + \frac{1}{2} \cdot \left(A - A^T\right)$

5.2.3 Homogenization procedure

As explained in Section 5.1.1, a model in a certain level (micro or meso) must be homogenized to get global behavior and to be used at higher level (meso or macro respectively). This process makes obtaining one stress-strain value for each time step possible. A simple but effective way to average the behavior of the entire model is using the first mean value theorem for integration:

Definition 1. *If* $f : [a, b] \to \mathbb{R}$ *is a continuous function then there exists a number* $x \in [a, b]$ *such that*

$$f(x) = \frac{1}{b-a} \int_a^b f(t)\,dt$$

$f(x)$ *represents the mean value of* f *in an integral meaning.*

Considering the previous definition, the homogenized value of stress and strain for a generic volume V are:

$$\langle \sigma \rangle = \frac{1}{V} \int_V \sigma\left(x, y, z\right)\, dV \tag{5.20}$$

$$\langle \epsilon \rangle = \frac{1}{V} \int_V \epsilon\left(x, y, z\right)\, dV \tag{5.21}$$

5.2.4 Damage evolution criteria

In order to model the behavior of woven fabric composites or non-crimp fabrics, considering linear elastic behavior for material properties is not confident and enough. A more detailed characterization must be performed, due to the complex three-dimensional architectures of such materials. A non-linear analysis with anisotropic damage constitutive equations should be considered.

Some classical failure criteria can predict the occurrence of damage inside the composite (like Hoffman's failure criterion [112]) but generally they cannot indicate the type of damage mode that actually affects strongly the mechanical behavior of the material. A good approach to damage mode identification is presented by Zako et al. [2]. They presented four different modes according to the loading condition caused the damage. Considering an orthotropic coordinte system $\{O_{L,T,Z}\}$ where L is along the fiber axis, T and Z, are perpendicular to the fiber axis (see Figure 5.9), it is possible to define the following damage modes:

- mode L: fiber breaking due to excessive tensile or compressive stress;
- mode T and LT: matrix breaking along direction T or due to the shear LT;
- mode Z and ZL: matrix breaking along direction Z or due to the shear ZL;
- mode TZ: matrix breaking due to the shear TZ;

which schematic damage mode is outlined in Figure 5.9.

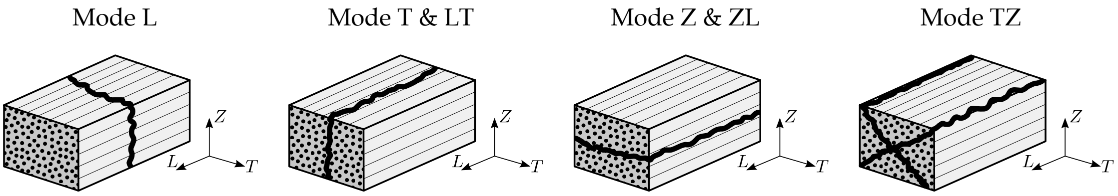


Figure 5.9: Anisotropic damage model for yarns

Furthermore it is possible to identify the beginning of the damage according to the mode with the highest stress-to-strength value. In particular the criteria to identify the starting of the damage are as follows:

$$\begin{cases} \dfrac{\sigma_L^2}{R_L^t \cdot R_L^c} & \text{Mode L} \\ \dfrac{\sigma_T^2}{R_T^t \cdot R_T^c} & \text{Mode T} \\ \dfrac{\tau_{LT}^2}{R_{LT}^2} & \text{Mode LT} \\ \dfrac{\sigma_Z^2}{R_Z^t \cdot R_Z^c} & \text{Mode Z} \\ \dfrac{\tau_{ZL}^2}{R_{ZL}^2} & \text{Mode ZL} \\ \dfrac{\tau_{TZ}^2}{R_{TZ}^2} & \text{Mode TZ} \end{cases} \tag{5.22}$$

In order to characterize the damage modes in a mathematical way, Zako et al. [2] adopted the Murakami's damage tensor that in matrix form is expressed as follows:

$$\underline{\underline{D}} = \begin{bmatrix} D_L & 0 & 0 \\ 0 & D_T & 0 \\ 0 & 0 & D_Z \end{bmatrix} \tag{5.23}$$

where $D_{L,T,Z}$, called damage factors, define the area reduction due to cracks and voids. They can easily be defined as follows:

$$D_i = \frac{A_i^{(0)} - A_i}{A_i^{(0)}} \tag{5.24}$$

where A_i indicates the cross-sectional area in actual damaged configuration and $A_i^{(0)}$ indicates the cross-sectional area in undamaged configuration, both are on planes perpendicular to the direction i. It is obvious that a value of zero means undamaged conditions (the area of damaged condition is equal to the one of undamaged condition as there is no damage at all) and the value of one means complete damage (the residual area of actual damage condition is zero).

The stress-strain constitutive law can slightly change introducing a stiffness matrix $\underline{\underline{Q}}$ function of the damage $\underline{\underline{D}}$:

$$\underline{\sigma} = \underline{\underline{Q}} \left(\underline{\underline{D}} \right) \cdot \underline{\epsilon} \tag{5.25}$$

where $\underline{\underline{Q}}$ can be written as follows:

$$\underline{\underline{Q}} \left(\underline{\underline{D}} \right) = \begin{bmatrix} d_L^2 \cdot Q_{11} & d_L \cdot d_T \cdot Q_{12} & d_Z \cdot d_T \cdot Q_{13} & 0 & 0 & 0 \\ & d_T^2 \cdot Q_{22} & d_T \cdot d_Z \cdot Q_{23} & 0 & 0 & 0 \\ & & d_Z^2 \cdot Q_{33} & 0 & 0 & 0 \\ & & & d_{TZ} \cdot Q_{44} & 0 & 0 \\ & sym. & & & d_{ZL} \cdot Q_{55} & 0 \\ & & & & & d_{TZ} \cdot Q_{66} \end{bmatrix} \tag{5.26}$$

The parameters d_i are lined to D_i as follows:

$$\begin{cases} d_L = 1 - D_L \\ d_T = 1 - D_T \\ d_Z = 1 - D_Z \\ d_{TZ} = \left(\dfrac{2 \cdot d_T \cdot d_Z}{d_T + d_Z} \right)^2 \\ d_{ZL} = \left(\dfrac{2 \cdot d_Z \cdot d_L}{d_Z + d_L} \right)^2 \\ d_{LT} = \left(\dfrac{2 \cdot d_L \cdot d_T}{d_L + d_T} \right)^2 \end{cases} \tag{5.27}$$

where the damage factors D_i can change from 0 (initial undamaged state) till 1 (full damaged state). For the numerical simulation purposes it is better to set the maximum value of $D_i = 1 - \delta$ where δ can be very low but not zero. In this way the instability problems would not occur. In fact too many zeros in matrix $\underline{\underline{Q}}$ may lead to ill conditioned inverse matrix (the compliance matrix, needed in finite element softwares to solve the structure).

Thus a value close to zero but not equal to zero is enough to give stability to the solution. In the case of $\delta << 1$, for example 0.01, it will be $d_L = \delta = 0.01$ and $d_{LT} = \delta^2 = 0.0001$. Both values are practically equal to zero but numerically different and so they will give more stability to the solution.

The process, applied to finite element software, will be as follows:

- solve the equilibrium equations and find the stress at each gauss point
- calculate the stress-to-strength ratio and find damaged elements
- modify the stiffness matrix $\underline{\underline{Q}}$ of the damaged elements
- solve again the equilibrium according to the new redistribution of stresses due to the presence of damaged elements

The cycle is repeated until no more damaged elements are going to modify the equilibrium.

The present method, was proven by Zako et al. [2] that is "*[...] useful for the evaluation and the design of structures using fiber composite materials*". In fact the comparison with experiments showed good agreement and moreover can be easily used to study the damage mechanism of composite materials.

5.3 Materials properties

In order to build the numerical models of the yarns, the material properties of the constituent (i.e. fibers and matrix) are required. Specimens made of these two materials can be subdue to different loading conditions in order to understand all the properties.

Table 5.1: Mechanical properties of an epoxy-carbon fibers UD ply

Property	Symbol	Value	Unit
Longitudinal strenght	R_L	2459±95	MPa
Longitudinal stiffness	E_L	156±3	GPa
Transverse strenght	R_T	42±7	MPa
Transverse stiffness	E_T	8.2±0.2	GPa

The material tested is an epoxy matrix reinforced by unidirectional carbon fibers with a volume fraction of fibers V_f of 70%. The overall properties of the UD ply are given in Table 5.1. These properties have been measured over several tests carried out at the laboratory of the Metallurgy and Material Engineering of the University of Leuven.

The fibers used are standard STS carbon fibers properties of which (measured by the manufacturer or reported in other papers [113]) are tabled in Tables 5.2 and 5.3.

Table 5.2: Carbon fibers manufacturer's data

Property	Symbol	Value	Unit
Type		STS carbon fibers	
Number of fibers per yarn	N_f	24K	
Weight	W_f	1600	tex
Tensile strength	R_f	4000	MPa
Tensile modulus	E_f	240	GPa
Elongation	ΔL_f	1.7	%
Fiber diameter	d_f	7.0	μm
Density	ρ_f	1.75	g/cm^3

The epoxy matrix has been tested building few specimens of homogeneous material and performing tensile tests. As it is isotropic as well, performing unidirectional tests are enough. After the tests, carried out at the laboratory of the Metallurgy and Material Engineering of the University of Leuven, the stress-strain data are fitted by a suitable constitutive law. Due to its

Table 5.3: Mechanical properties of carbon fibers

Property	Symbol	Value	Unit
Longitudinal elastic modulus	E_1	240	GPa
Transversal elastic modulus	E_2	14	GPa
Longitudinal shear modulus	G_{12}	14	GPa
Longitudinal Poisson's ratio	ν_{12}	0.2	
Transversal Poisson's ratio	ν_{23}	0.25	

general expression, able to catch very well non-linear behavior of materials, a Ramberg-Osgood [110] constitutive law has been used.

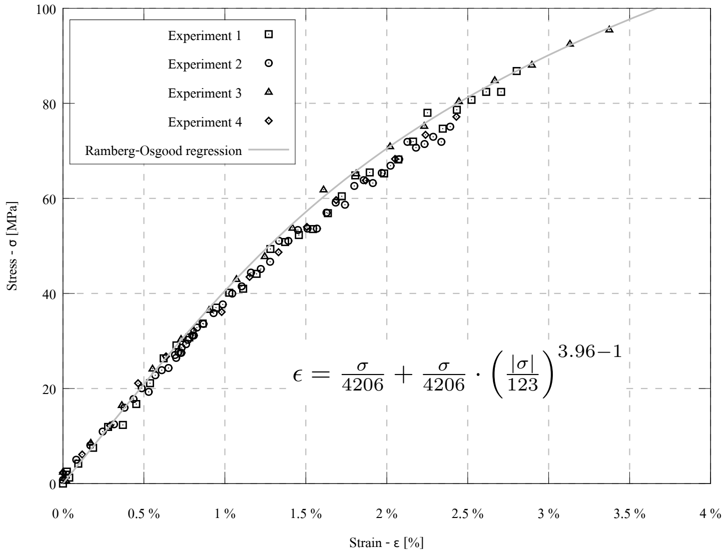


Figure 5.10: Ramberg-Osgood regression of eopxy matrix tensile tests

Figure 5.10 shows the data taken from four experiments and the Ramberg-Osgood regression curve. The fitting data are listed in Table 5.4.

5.4 Micro-mechanical models

Here the models built to obtain the 13 parameters necessary to describe the complete behavior of the non-crimp fabric yarns are presented.

Using a model like in Figure 5.11, several analyses can be performed. In particular the following analysis is presented:

- tension along 1 in order to obtain E_L and ν_{12}
- tension along 2 in order to obtain E_T, n_2, σ_T^Y and ν_{23}

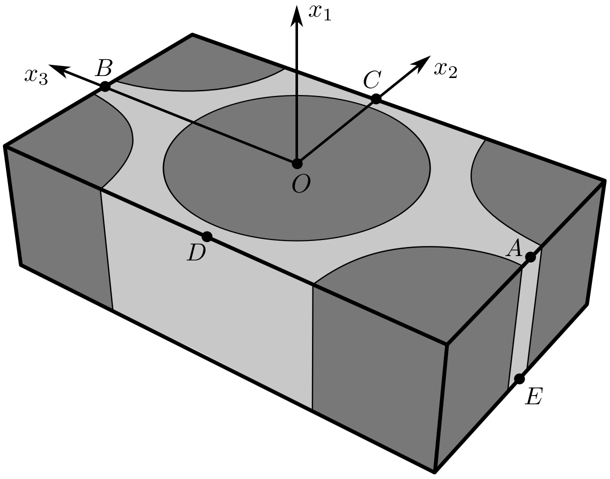


Figure 5.11: Micro-mechanical model of fiber reinforced plastic composites

- shear along 2-3 in order to obtain G_T, τ_T^Y, n_6 and α_{23}
- shear along 1-3 in order to obtain G_L, τ_L^Y and n_4

5.4.1 Tension along 1

It is possible to calculate the values of the elastic modulus E_L and the Poisson's ratio ν_{12} from a micro-mechanical model simulating a tension along direction 1. The prevalent contribution of the fibers makes the behavior of the model linear, with an elastic modulus mainly dependent on the volume fraction of fibers. This is also predictable from the Chamis formula:

$$E_L^{\text{Chamis}} = V_f \cdot E_{11_{\text{fibers}}} + (1 - V_f) \cdot E_{\text{matrix}} \tag{5.28}$$

For a volume fraction of 70% and an elastic modulus of fibers and matrix of 240 GPa and 4.2 GPa respectively the longitudinal elastic modulus of the yarns will be 169.3 GPa. From the analysis, values of which are listed in Table 5.5, the value calculated by linear regression is 166.8 GPa (the difference is 1.5%).

Measuring the deformation along 2 due to the tension along 1 (transverse contraction) it is

Table 5.4: Mechanical properties of epoxy matrix

Property	Symbol	Value	Unit
Tensile elastic modulus	E_m	4206	MPa
Yield stress	σ_m^Y	123	MPa
Hardening exponent	n_m	3.96	

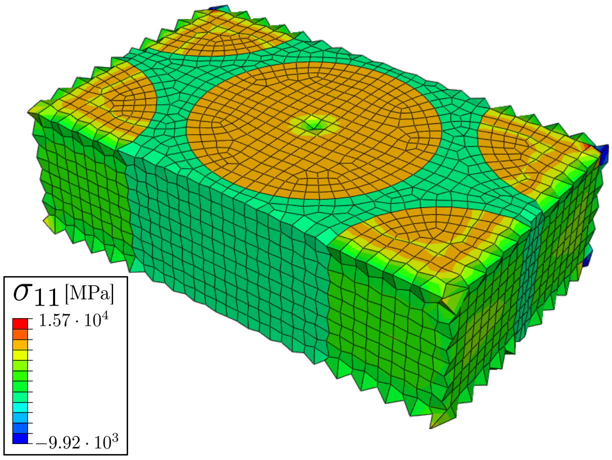


Figure 5.12: Micro-mechanical model $\sigma_{11} - \epsilon_{11}$

Table 5.5: Results from the micro-mechanical model “tension along 1”

frame	$\langle\sigma_{11}\rangle$ [MPa]	$\langle\epsilon_{11}\rangle$ [%]	$\langle\epsilon_{22}\rangle$ [%]	$\langle\epsilon_{33}\rangle$ [%]
0	0	0.00	0.00	0.00
1	620	0.37	-0.09	-0.09
2	1239	0.74	-0.18	-0.18
3	1858	1.11	-0.27	-0.28
4	2476	1.48	-0.36	-0.37
5	3093	1.85	-0.46	-0.47
6	3710	2.22	-0.55	-0.57
7	4327	2.59	-0.65	-0.68
8	4943	2.96	-0.75	-0.78
9	5559	3.33	-0.85	-0.88
10	6175	3.71	-0.94	-0.99

possible to calculate the Poisson’s ratio as follows:

$$\nu_{12} = -\frac{\langle\epsilon_{22}\rangle}{\langle\epsilon_{11}\rangle} \tag{5.29}$$

In the same way it is possible to calculate ν_{13} but as the material is transversely isotropic must be $\nu_{13} = \nu_{12}$. From Table 5.5 and using again Chamis formula the following parameters can be calculated:

$$\nu_{12}^{\text{Chamis}} = V_f \cdot \nu_{12_{\text{fibers}}} + (1 - V_f) \cdot \nu_{\text{matrix}} = 0.315 \tag{5.30}$$

The value calculated from the finite element results is not so perfect due to numerical approximations. In fact it is $\nu_{12} = 0.252$ and $\nu_{13} = 0.263$ with a mean value of $\nu = 0.258$ with a difference of 22% respect to analytically achieved results.

5.4.2 Tension along 2

It is possible to calculate the values of the elastic modulus E_T, the Poisson's ratio ν_{23}, the yield stress σ_T^Y and the hardening coefficient n_2 from a micro-mechanical model simulating a tension along direction 2 (or 3 as well). In fact along this direction the non-linear behavior of the matrix becomes important and also the yarns behave non-linearly. Thus a Ramberg-Osgood regression must be used.

The behavior of the micro-mechanical model loaded along 2 or along 3 is slightly different. When loaded along direction 2 raises to the following regression parameters:

- $E_2 = 8711$ MPa
- $\sigma_2^Y = 304$ MPa

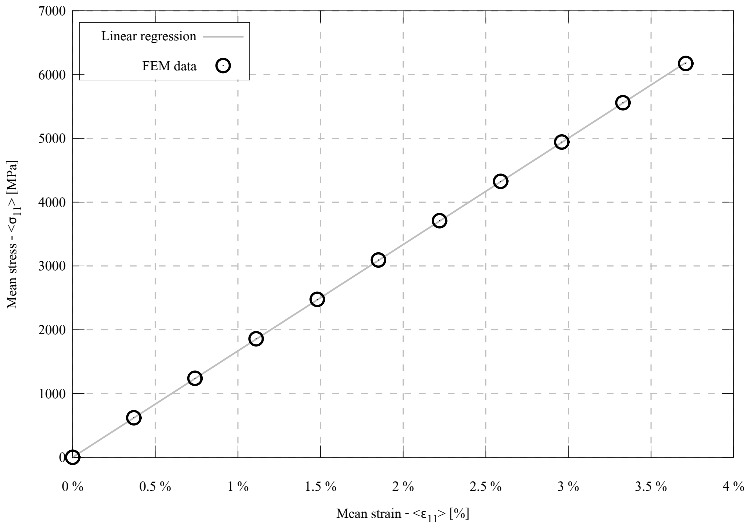
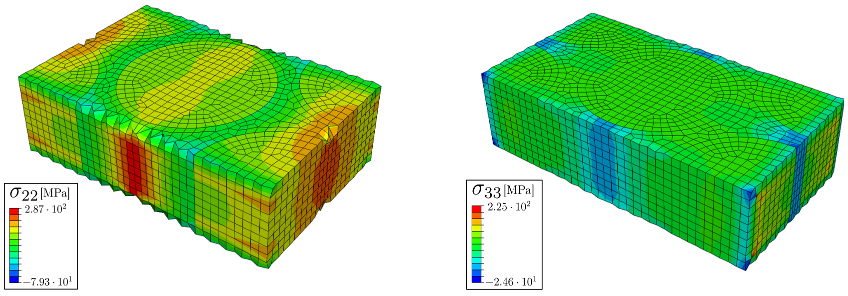


Figure 5.13: Linear regression of $\langle\sigma_{11}\rangle$–$\langle\epsilon_{11}\rangle$ data



(a) Micro-mechanical model $\sigma_{22} - \epsilon_{22}$

(b) Micro-mechanical model $\sigma_{33} - \epsilon_{33}$

Figure 5.14: Micro-mechanical models along direction 2 and 3

Table 5.6: Results from the micro-mechanical model “tension along 2 and 3”

(a) Tension along direction 2

frame	$\langle\sigma_{22}\rangle$ [MPa]	$\langle\epsilon_{22}\rangle$ [%]
0	0	0.00
1	17	0.20
2	34	0.39
3	50	0.59
4	66	0.78
5	81	0.98
6	96	1.17
7	109	1.36
8	122	1.56
9	134	1.75
10	145	1.95

(b) Tensione along direction 3

frame	$\langle\sigma_{33}\rangle$ [MPa]	$\langle\epsilon_{33}\rangle$ [%]
0	0	0.00
1	10	0.11
2	20	0.23
3	30	0.34
4	39	0.46
5	49	0.57
6	58	0.68
7	68	0.80
8	77	0.91
9	85	1.02
10	94	1.14

- $n_2 = 3.37$

while if a tension along direction 3 is performed the models gives the following results:

- $E_3 = 8668$ MPa
- $\sigma_3^Y = 285$ MPa
- $n_3 = 3.67$

The small difference is even clearer in Figure 5.15 where both data from a tension along direction 2 and a tension along direction 3 have been plotted together. A mean behavior can be calculated using both set of data. The result is plotted in Figure 5.15. The parameters of the Ramberg-Osgood expression are:

- $E_T = 8703$ MPa
- $\sigma_3^Y = 296$ MPa
- $n_3 = 3.46$

Taking the advantage of the previous analyses, it is easily possible to calculate the Poisson's ration ν_{23}. In fact using the homogenized strain data along direction 2 and along direction 3 from an analysis performed loading along direction 2 it is possible to calculate ν_{23} and at the same way from an analysis performed loading along direction 3 is possible to calculate ν_{32} using the homogenized strain data along direction 3 and along direction 2. These two values must be theoretically the same. However because of the numerical approximations of the finite element approach the two values will be different but should be still very close to each other.

By linear regression of $\langle\epsilon_{33}\rangle$ vs. $\langle\epsilon_{22}\rangle$ the Poisson's ratio is 0.361 and by linear regression of $\langle\epsilon_{22}\rangle$ vs. $\langle\epsilon_{33}\rangle$ the Poisson's ratio is 0.374 with a difference of 4%. The used value will be the mean value that is $\nu_{23} = 0.368$.

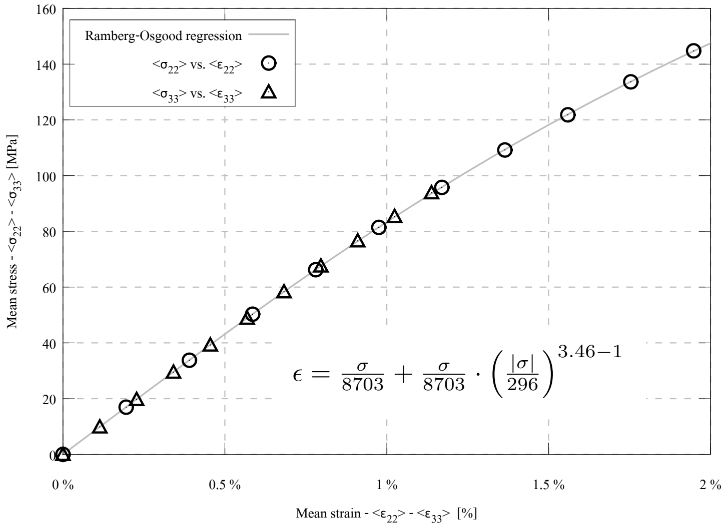


Figure 5.15: Ramberg-Osgood regression of $\langle\sigma_{22}\rangle$–$\langle\epsilon_{22}\rangle$ and $\langle\sigma_{33}\rangle$–$\langle\epsilon_{33}\rangle$ data

5.4.3 Shear along 2-3

The shear 2-3 is the one in the plane of symmetry, perpendicular to the direction of fibers. This loading condition leads obviously to a non-linear behavior of the micro-mechanical model and so consequently of the yarns. Again a Ramberg-Osgood regression must be used and fitting parameters must be calculated.

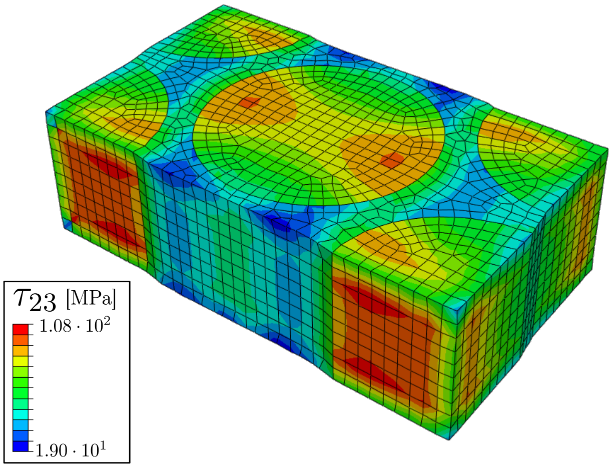


Figure 5.16: Micro-mechanical model $\sigma_{23} - \epsilon_{23}$

From Table 5.7 it is possible to plot the data, showed in Figure 5.17. The Ramberg-Osgood regression leaded to the following results:

Table 5.7: Results from the micro-mechanical model “shear along 2-3”

frame	$\langle\tau_{23}\rangle$ [MPa]	$\langle\gamma_{23}\rangle$ [%]
0	0.00	0.00
1	7.27	0.23
2	14.51	0.45
3	21.65	0.68
4	28.60	0.91
5	35.28	1.14
6	41.61	1.36
7	47.59	1.59
8	53.21	1.82
9	58.49	2.05
10	63.46	2.28

- $G_T = 3205$ MPa
- $\tau_T^Y = 135$ MPa
- $n_6 = 3.52$

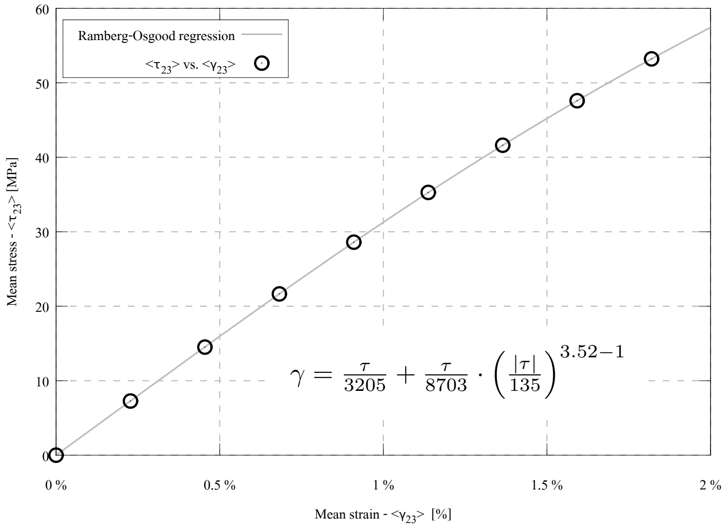


Figure 5.17: Ramberg-Osgood regression of $\langle\tau_{23}\rangle$–$\langle\gamma_{23}\rangle$ data

5.4.4 Shear along 1-3

The shear along 1-3 is the one considering a plane that contains the axis of the fibers. The influence of the matrix anyway is introducing non-linear behavior in the shear response of the yarns.

Table 5.8: Results from the micro-mechanical model “shear along 1-2 and 1-3”

(a) Shear along direction 1-2

frame	$\langle\tau_{12}\rangle$ [MPa]	$\langle\gamma_{12}\rangle$ [%]
0	0	0.00
1	21	0.39
2	39	0.78
3	53	1.18
4	64	1.57
5	72	1.96
6	79	2.35
7	85	2.75
8	90	3.14
9	95	3.54
10	99	3.93

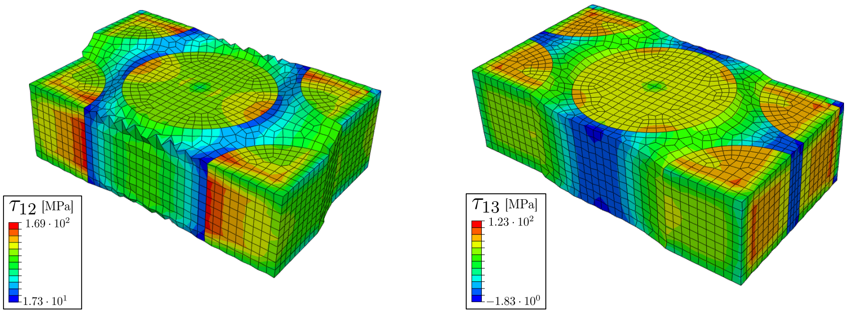
(b) Shear along direction 1-3

frame	$\langle\tau_{13}\rangle$ [MPa]	$\langle\gamma_{13}\rangle$ [%]
0	0	0.00
1	13	0.23
2	25	0.46
3	35	0.68
4	44	0.91
5	52	1.14
6	58	1.37
7	63	1.60
8	68	1.83
9	72	2.05
10	75	2.28

Also in this case, considering the transversely isotropic nature of the material it is possible to define two numerical models, one with a loading condition along 1-2 and one with a loading condition along 1-3 (Figure 5.18). The comparison of the two models should give similar results, while the differences should remain related to numerical approximations due to the finite element approach. The analysis performed with the loading along direction 1 and 2 can be seen in Figure 5.18(a). The results, listed in Table 5.8(a), lead to the following regression parameters:

- $E_4 = 5461$ MPa
- $\sigma_4^Y = 94$ MPa
- $n_4 = 3.85$

The analysis performed with the loading along direction 1 and 3 can be seen in Figure 5.18(b). The results, listed in Table 5.8(b), lead to the following regression parameters:



(a) Micro-mechanical model $\tau_{12} - \gamma_{12}$

(b) Micro-mechanical model $\tau_{13} - \gamma_{13}$

Figure 5.18: Micro-mechanical models along direction 1-2 and 1-3

- $E_5 = 5552$ MPa
- $\sigma_5^Y = 86$ MPa
- $n_5 = 3.90$

As it was expected the values obtained from the two analyses have comparable results. The difference between E_4 and E_5 is of 1.7%, the difference between σ_5^Y and σ_6^Y is of 1.3% and the difference between n_5 and n_6 is of 8.4%.

5.5 Implementation in Abaqus/Standard

The model described up to now can be easily implemented in any available finite element software. The present work has been developed inside the suite Abaqus, using the implicit solver Abaqus/Standard.

In order to analyze the described models, few steps must be followed carefully:

1. creating the geometry as shown in Figure 5.6 and according to the dimension of the fibers as calculated by the Equation (5.15)
2. defining the material properties of matrix and fibers according to the data of Table 5.4 and Table 5.3
3. building the three-dimensional mesh describing the geometry
4. removing the rigid-body translation by suitable boundary conditions
5. removing the rigid-body rotations and applying the periodic boundary conditions by adding suitable equations
6. running the analysis

Most of the listed options can be defined using Abaqus/CAE, namely the front-end and pre-processor of Abaqus/Standard. The option 5 needs the definition of equations. This procedure can be done inside Abaqus/CAE but linking sets of nodes according to an explicit defined equation, like:

$$\sum_{i=1}^{N} A_i \cdot u_{j_i}^{r_i} = 0 \tag{5.31}$$

which is a sum of i products of coefficients A_i and displacement components $u_j, j = 1, 2, 3$ (where $\underline{u} = (u_1, u_2, u_3)$ is the displacement vector) of the node number r.

Considering writing the equation between two sets of nodes (on which the equation can be applied as well) the equation will be applied practically on each couple of nodes (one each set) according to their order inside the set. As there is no way to push Abaqus for a particular sorting strategy, a software has been developed to create the sets of nodes, sort them and then create the appropriate equations.

5.5.1 Describtion of Abaqus terminology

Before starting the description of the software developed to interact with Abaqus input files, it is better to describe how Abaqus creates these files and how it is possible to modify them and to use them for the purpose of defining proper boundary conditions.

The input file of Abaqus

The *input file* is an ASCII file which describes the model to be solved by Abaqus. This file contains the description of the mesh (nodes and elements), the material properties, the boundary conditions (constraints and loads), equations, sets, fields to be calculated during te solution. etc. thanks to a proper syntax.

Considering the ASCII format of the file, it can be easily modified (or created) by a user developed software before sending it to the solver. An example of a very simple input file is the following:

Example of Abaqus input file

```
*Heading
** Job created as example
*Preprint, echo=NO, model=NO, history=NO, contact=NO
*Node
        1,           0.,           0.,          20.
        2,         -14.,           0.,          20.
        3,         -28.,           0.,          20.
        4,         -42.,           0.,          20.
        5,           0.,           0.,          10.
        6,         -14.,           0.,          10.
        7,         -28.,           0.,          10.
        8,         -42.,           0.,          10.
        9,           0.,           0.,           0.
       10,         -14.,           0.,           0.
       11,         -28.,           0.,           0.
       12,         -42.,           0.,           0.
  *Element, type=S4R
  1,  1,  2,  6,  5
  2,  2,  3,  7,  6
  3,  3,  4,  8,  7
  4,  5,  6, 10,  9
  5,  6,  7, 11, 10
  6,  7,  8, 12, 11
  *Nset, nset=AllNodes, generate
   1,  12,   1
  *Elset, elset=AllElements, generate
   1,  6,  1
  *Shell Section, elset=AllElements, material=Steel
   10., 5
  *System
```

```
*Nset, nset=BlockedNodes, generate
 4,  12,   4
*Elset, elset=BlockedElements
 3, 6
*Nset, nset=LoadedNode
 1,
*Material, name=Steel
*Elastic
 206000., 0.3
*Step, name=Step-1
*Static
 1., 1., 1e-05, 1.
*Boundary
BlockedElements, ENCASTRE
*Cload
LoadedNode, 1, 1.
*Output, field, variable=PRESELECT
*Output, history, variable=PRESELECT
*End Step
```

At the beginning `*Heading` defines just a title for the analysis and `*Preprint` selects the printout that will be obtained from the analysis input file processor. Then the file continues with the definition of nodes by `*Node` and the elements by `*Element`. `*Nset` and `*Elset` define sets of nodes or elements respectively. Other important parts are the material definition, using the key word `*Material`, and the boundary conditions definition, thanks to `*Boundary` and `*Cload`.

This file can be easily managed by user developed software. In particular it is important to analyze the nodes and apply the correct boundary conditions to the model.

Nodes sets

In Abaqus the definition of nodes sets is done by the use of the special word `*Nset` followed by:

- `nset=` plus the name of the set (e.g. `nset=BlockedNodes`)
- `unsorted`. If this parameter is included, the nodes in this node set will be assigned to the set in the order in which they are given. If this parameter is omitted, the nodes in the set are sorted into ascending order of their node numbers, with duplicates eliminated.
- `generate` can be used in order to generate a nodes set by the definition of one or more rules to select which nodes must be included. The following lines should give a first node n_1, a last node n_2 and the increment in node numbers between these nodes, i. Then, all nodes going from n_1 to n_2 in steps of i will be added to the set. i must be an integer such that $\frac{n_2-n_1}{i}$ is a whole number.

After this line a series of lines are needed to assign the nodes number to add to the set (apart from when `generate` is used). Each number is separated from another by a comma (,) and the maximum number of nodes in a line is 16. Also a label set is allowed, but only if it has been set before.

An example of a definition of a node set of 20 nodes is:

Example of nodes set definition

```
*Nset, nset=Set-Example, unsorted
 187, 570, 114, 804, 113, 236, 170, 168, 269, 271, 475, 477, 935, 937, 434, 432
 802, 930, 548, 823
```

Equations in Abaqus

The equations are used to define linear multi-point constraints in the form of an equation. In Abaqus the definition of equations is made by the use of the key word `*Equation` followed by:

- a line that contains the number of the terms in the equation
- the equation as sum of coefficients multiplied by a DOF of a certain node or nodes set

The equation finally can be written as follow

$$\sum_{i=1}^{N} A_i \cdot u_{j_i}^{r_i} = 0 \tag{5.32}$$

where i is the term's index, j stands for the j^{th} degree of freedom of the node (or nodes set label) pointed by r. Each term is thus composed of three parts:

- node number or node set's label, r_i
- degree of freedom involved in the equation, j_i
- coefficient of the term, A_i

This has to be repeated as many times as needed to define all terms of the equation, but more than four terms can not be defined in a line. The first term can be defined either as a node number or a nodes set label. However if the first term is defined as a node, then the others have to be a single node (or a set made by a single node).

An example of a definition of such an equation is:

Example of equation definition

```
*Equation
2
Set-up, 3, 1.0, Set-down, 2, -2.0
```

that has to be seen as

$$1.0 \cdot u_3^{\text{Set-up}} - 2.0 \cdot u_2^{\text{Set-down}} = 0$$

The way the equations link to the nodes inside node sets depends on the position of each node on the set. So a simple equation like the afore reported links the first node of set *Set-up* with the first node of set *Set-down*, the second node of *Set-up* with the second node of set *Set-down* and so on.

Boundary conditions in Abaqus

In Abaqus the definition of boundary conditions is done by the use of the special word `*Boundary` followed by:

- node number or nodes set's label
- first degree of freedom constrained
- last degree of freedom constrained (this can be left blank if only one degree of freedom is being constrained)
- magnitude of the variable (displacement, velocity, etc.). If this magnitude is a rotation has to be given in radians

All parameters has to be separated by comma (,). An example of a definition of a equation is:

Example of boundary condition definition

```
*Boundary
Set-Point-A, 3, 3, -0.01
```

that has to be seen as

$$u_3^{\text{Set-Point-A}} = -0.01$$

in the unit of measurement of the model, that is defined by the user.

5.5.2 Micro-mechanical generator

It is possible to obtain the boundary conditions inside Abaqus/CAE creating proper equations able to link nodes displacement. Abaqus/CAE allows the user to manage equations but is not so powerful and also it is not so easy, or even impossible, to define the correct sets of nodes. Moreover the same degree of freedom of one node cannot be present in more than one equation because Abaqus is not able to manage the multi-presence of degree of freedom in different equations.

Furthermore the sorting technique of Abaqus doesn't let the user to select precisely which couple of nodes are interacting thanks to a special equation defined between sets of nodes.

As it is not possible to perform this procedure from Abaqus/CAE, a software in C++ has been developed to create sets and equations automatically. For each pair of surfaces, one of the surfaces will be sorted according to the coordinates of the nodes on that surface (in the order x, y and than z), while the other one will be sorted just looking through the nodes list of the first surface and looking for the node of the second surface that is as close as possible to it (in order to avoid numerical approximation mistakes).

Thus the procedure for creating a micro-mechanical model will be:

- create the geometry with Abaqus/CAE
- set material properties to the model
- define the mesh
- impose $\underline{u}_O = 0$ by fixing the node at coordinates $O \equiv (0, 0, 0)$
- save the input file (.INP)
- update the input file using the software here discussed
- run the analysis from the *Abaqus command prompt*

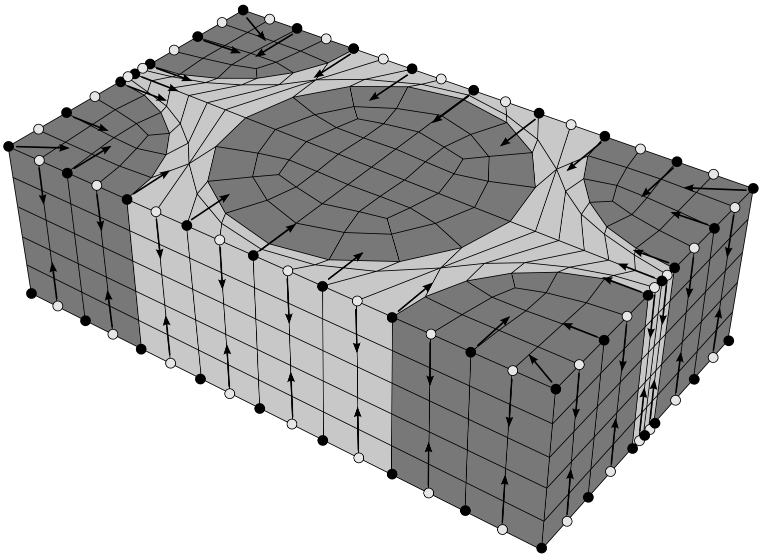


Figure 5.19: Assignation scheme for nodes sets

The software is made for managing a prismatic model as depicted in Figure 5.19. It reads all nodes coordinates and creates 11 sets of nodes (6 surfaces plus 5 single nodes) starting from the input file. The sets are made in a way that each node is owned by only a set of 11 possible sets.

This is mandatory because otherwise Abaqus will not be able to manage a degree of freedom of one node on multiple equations.

Furthermore the software will create sets on surfaces and sets of single nodes after counting the dimension of the cell so that the software can work fine on any prismatic model, not depending on the dimensions or number of nodes. The only requirement is the existence of nodes on the intersections of planes 12, 23 and 31, otherwise the 5 sets of single nodes will not be created, and a warning will be shown. The power of the software comes mainly from the capability of creating sets to have as more uniform behavior of the model as possible. For this reason the nodes lying on the edges of the upper surface will belong alternatively to the upper set and to one of the four front sets. As it is shown in Figure 5.19, black nodes will belong to top and bottom surfaces meanwhile gray nodes will belong to front and lateral surfaces.

Finally the software will define also the proper loading conditions according to the specified micro-mechanical test to be analyzed. Thus to summarize the present software works as follows:

- it opens the Abaqus input file ".inp"
- it reads the nodes coordinates
- it calculates the dimensions of the unit-cell
- it defines the 11 necessary node sets for applying equations and boundary conditions in general (sorting the nodes)
- it writes a new file with the following:
  - 6 node sets for the 6 surfaces of the unit-cell
  - 5 node sets for the 5 main nodes governing the equations
  - equations relating the nodes sets according to Equation (5.19)
  - boundary condition of zero rigid-body translation by fixing the node at coordinates $O \equiv (0, 0, 0)$
  - boundary condition of imposed displacement to impose the proper loading condition (tension along 1, tension along 2, shear 1-2, etc.)
- it closes the original Abaqus input file
- it closes the new Abaqus input file

5.5.3 Homogenization script

Once a micro-mechanical model has been analyzed, the model has to be homogenized according to what has been explained in Section 5.2.3. In this study to have a faster solution, a user developed Python$^{\text{TM}}$ script has been written.

Python$^{\mathrm{TM}}$ is a scripting language, namely a programming language that allows control of one or more software applications. Scripts are distinct from the core code of the application and are often interpreted from source code (while the applications they control are traditionally compiled to native machine code). Scripting languages are not intended for writing applications from scratch but they are intended primarily for plugging the components to each other. In fact they are often used to extend the features of components but they are rarely used for complex algorithms and data structures. The power and ease of use it makes Python$^{\mathrm{TM}}$ one of the most used scripting languages.

The presented script has the following algorithm:

Algorithm 5.1 Homogenization Python$^{\mathrm{TM}}$ script

```
for all steps do
    for all frames do
        $\text{value}^{(\text{frame})} \leftarrow \begin{bmatrix} \sum_{i=1}^{N_{\text{gauss points}}} V^{(i)} \cdot \sigma^{(i)} \\ \sum_{i=1}^{N_{\text{gauss points}}} V^{(i)} \cdot \epsilon^{(i)} \\ \sum_{i=1}^{N_{\text{gauss points}}} V^{(i)} \end{bmatrix}$
        write $\langle \sigma \rangle = \frac{\text{value}_0^{(\text{frame})}}{\text{value}_2^{(\text{frame})}}$
        write $\langle \epsilon \rangle = \frac{\text{value}_1^{(\text{frame})}}{\text{value}_2^{(\text{frame})}}$
    end for
end for
```

The algorithm can be performed by the Python$^{\mathrm{TM}}$ script described in detailed in the following:

Homogenization Python$^{\mathrm{TM}}$ script

```
from abaqus import mdb, session
from abaqus import *
from abaqusConstants import *
import os, nt

def homogenizationProcedure(stress, strain):
    vpName    = session.currentViewportName
    modelName = session.sessionState[vpName]['modelName']
    model     = mdb.models[modelName]
    filename  = 'homogenization.txt'

    currentOdb = session.viewports[session.currentViewportName].odbDisplay.name

    if os.path.exists(filename):
```

```
    nt.remove(filename)
file = open(filename, 'w')

thisOdb = session.odbs[currentOdb]

for sStep in thisOdb.steps.values():
    file.write('\n\nSTEP:␣%s\n'%(sStep.name))
    value = [[0 for x in range (3)] for frame in sStep.frames]
    for frame in sStep.frames:
        field = frame.fieldOutputs
        tmpField = field['IVOL'] * field['S'].
                                    getScalarField(componentLabel=stress)
        for v in tmpField.values:
            value[frame.frameId][0] += v.data
        tmpField = field['IVOL'] * field['E'].
                                    getScalarField(componentLabel=strain)
        for v in tmpField.values:
            value[frame.frameId][1] += v.data
        for v in field['IVOL'].values:
            value[frame.frameId][2] += v.data
    file.write('FRAME\t%s\t\t%s\t\tTOTAL␣IVOL\n'%(stress, strain))
    i = 0
    for v in value:
        file.write('%i\t%f\t%f\t%f\n'%(i, v[0] / v[2], v[1] / v[2], v[2]))
        i = i + 1

file.close()
```

5.5.4 Meso-mechanical model

After the definition of the micro-mechanical models, the meso-mechanical model must be built. The studied material is a non-crimp fabric made of carbon fibers in epoxy matrix with a stacking sequence of the UD plies of $(-45, +45)_{2s}$. A careful examination of the material revealed also a small difference in volume fraction between the -45 and the $+45$ degree plies. The -45 degree ply has a volume fraction of fibers $V_f = 62.6\%$ while the $+45$ degree ply has a volume fraction of fibers $V_f = 63.2\%$.

The matrix has been considered homogeneous, isotropic and with non-linear stress-strain behavior, Section 5.3. The characteristic stress-strain relationship of the yarns has been defined by the micro-mechanical models aforementioned (see Section 5.2). As the plies have different volume fraction, the micro-mechanical models have been built and analyzed according to this difference (see Section 5.3 for the characteristics of the pure fibers).

The Figure 5.20 shows the unit-cell of such a composite. The light gray defines the homogeneous matrix. The two levels of dark gray show the two ply of non-crimp UDs, where the upper one has lower volume content of fibers. The volumes of pure matrix are units called “pockets”, namely channels inside the composite formed by the stitching process, as described in Section 5.1.

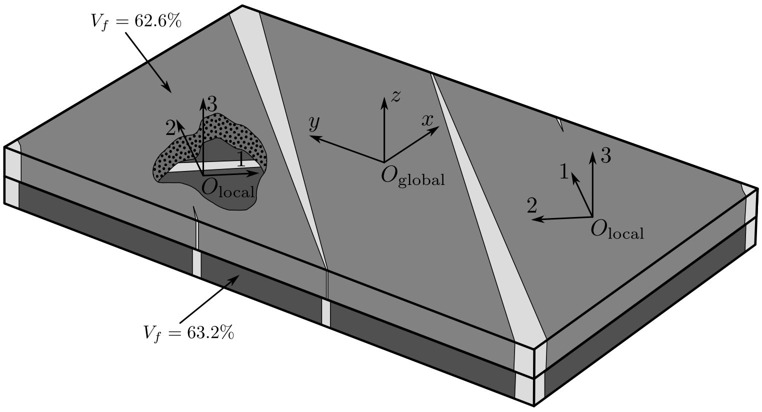


Figure 5.20: Meso-mechanical model of $(-45, +45)_{2\text{s}}$ non-crimp fabric

The unit-cell of the meso-mechanical model has been obtained by the study of the particular spatial distribution of these resin rich zones.

In Figure 5.21 a series of pockets with the central unit-cell highlighted are shown. A similar scheme is for both upper and lower ply. The unit-cell is made of just two plies thanks to the special symmetry of the stacking sequence. Proper boundary conditions will impose a behavior to the model in order to simulate the presence of the not modeled plies.

The model has a global coordinate system $O_{\{x,y,z\}}$ that is used to build the geometry and the mesh. Locally each ply has a different coordinate system $O_{\{1,2,3\}}$ as each ply has a different fiber orientation. In fact as the used material is anisotropic, the special selection of the coordinate system is fundamental for the solution of the finite element model. The upper ply and the lower ply have local coordinate systems oriented with the third axis parallel and perpendicular one to each other (the axis 1 of the first system is parallel to the axis 2 of the second system), as described by the stacking sequence. The coordinate system of the resin rich zones is irrelevant as the matrix is isotropic.

Once the unit-cell is built, it has been constrained as shown in Figure 5.8, thus according to what is explained in Section 5.2.2. The main boundary conditions are related to periodicity, while part of them are related to avoid rigid-body motions as translations or rotations. A user developed software as the one described in Section 5.5.2 has been used to apply the correct boundary conditions.

5.5.5 User material in Abaqus/Standard

User-defined mechanical material behavior in Abaqus is provided by means of an interface whereby any mechanical constitutive model can be added to the library [114]. This requires a constitutive model programmed in user subroutine UMAT.

Any material constants that are needed in user subroutine UMAT must be specified as part of a user-defined material behavior definition.

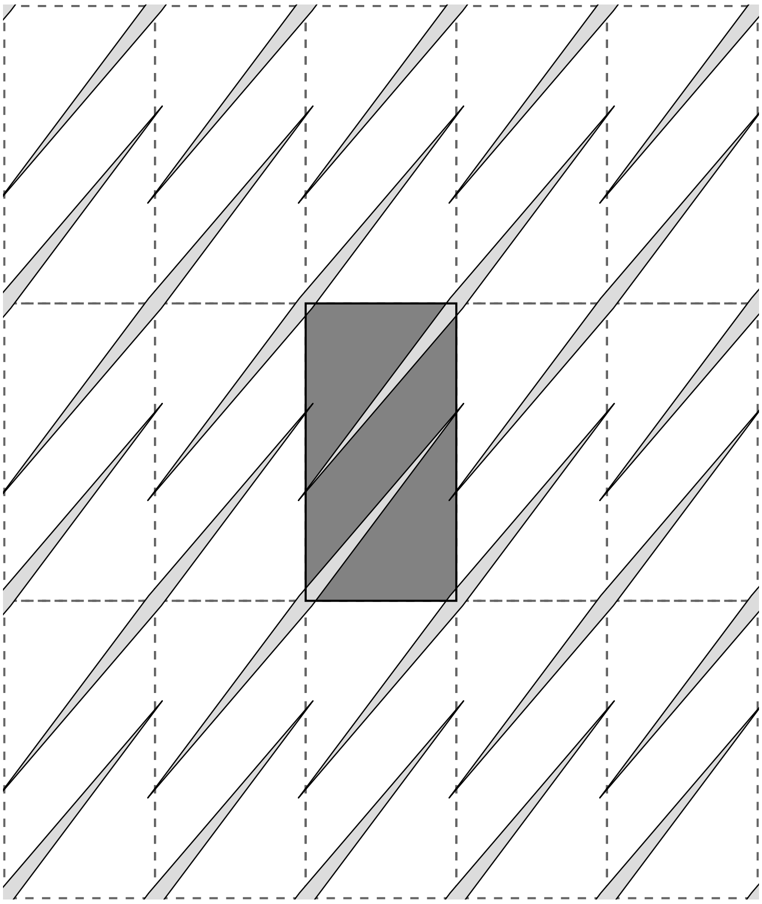


Figure 5.21: Non-crimp fabric +45 degree unit cell indentification

The user subroutine is called for each material point at each iteration of every increment. It is provided with the material state at the start of the increment (stress, solution-dependent state variables, temperature, and any predefined field variables) and with the increments in temperature, predefined state variables, strain, and time. It must return values for the new stresses and the new internal state variables.

In addition to updating the stresses and the solution-dependent state variables to their values at the end of the increment, subroutine UMAT must also provide the material Jacobian matrix, $\partial\Delta\sigma/\partial\Delta\epsilon$, for the mechanical constitutive model. When the constitutive model is in rate form and is integrated numerically in the subroutine, the material Jacobian matrix will be also dependent on the applied integration scheme. For any nontrivial constitutive model these will be a challenging tasks. For example, the accuracy with which the Jacobian matrix is defined is a major determinant of the convergence rate of the solution and therefore it has a strong influence on computational efficiency.

An example of user subroutine UMAT is the following:

UMAT example

```
      SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
     1 RPL,DDSDDT,DRPLDE,DRPLDT,
     2 STRAN,DSTRAN,TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,CMNAME,
     3 NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,DROT,PNEWDT,
     4 CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,KSPT,KSTEP,KINC)
C
      INCLUDE 'ABA_PARAM.INC'
C
      CHARACTER*80 CMNAME
      DIMENSION STRESS(NTENS),STATEV(NSTATV),
     1 DDSDDE(NTENS,NTENS),DDSDDT(NTENS),DRPLDE(NTENS),
     2 STRAN(NTENS),DSTRAN(NTENS),TIME(2),PREDEF(1),DPRED(1),
     3 PROPS(NPROPS),COORDS(3),DROT(3,3),DFGRD0(3,3),DFGRD1(3,3)

       user coding defining DDSDDE, STRESS, STATEV, SSE, SPD, SCD
       and, optionally, RPL, DDSDDT, DRPLDE, DRPLDT, PNEWDT

      RETURN
      END
```

Some variables must be set inside a UMAT subroutine, while others are optional. The variables passed in are the following:

- `STRAN(NTENS)`: an array containing the total strains at the beginning of the increment. These strains are available for output as “elastic” strains. In finite-strain problems the strain components have been rotated to account for rigid body motion in the increment before `UMAT` is called and are approximations of the logarithmic strain.
- `DSTRAN(NTENS)`: array of strain increments. If thermal expansion is included in the same material definition, these are the mechanical strain increments (the total strain increments minus the thermal strain increments)
- `TIME(1)`: value of *step* time at the beginning of the current increment
- `TIME(2)`: value of *total* time at the beginning of the current increment
- `DTIME`: time increment
- `TEMP`: temperature at the start of the increment
- `DTEMP`: increment of temperature
- `PREDEF`: array of interpolated values of predefined field variables at this point at the start of the increment, based on the values read in at the nodes
- `DPRED`: array of increments of predefined field variables

- `CMNAME`: user-defined material name, left justified. Some internal material models are given names starting with the "`ABQ_`"character string. To avoid conflict "`ABQ_`" should not be used as the leading string for `CMNAME`
- `NDI`: number of direct stress components at this point (i.e. $\sigma_1, \sigma_2, \sigma_3$)
- `NSHR`: number of engineering shear stress components at this point (i.e. $\tau_{12}, \tau_{13}, \tau_{23}$)
- `NTENS`: size of the stress or strain component array (`NDI` + `NSHR`)
- `NSTATV`: number of solution-dependent state variables that are associated with this material type
- `PROPS(NPROPS)`: user-specified array of material constants associated with this user material
- `NPROPS`: user-defined number of material constants associated with this user material
- `COORDS`: an array containing the coordinates of this point. These are the current coordinates if geometric nonlinearity is accounted for during the step, otherwise the array contains the original coordinates of the point
- `DROT(3,3)`: rotation increment matrix. This matrix represents the increment of rigid body rotation of the basis system in which the components of stress (`STRESS`) and strain (`STRAN`) are stored. It is provided so that vector- or tensor-valued state variables can be rotated appropriately in this subroutine: stress and strain components are already rotated by this amount before `UMAT` is called. This matrix is passed in as a unit matrix for small-displacement analysis and for large-displacement analysis if the basis system for the material point rotates with the material (as in a shell element or when a local orientation is used)
- `CELENT`: characteristic element length. It depends on the shape and the order of the element
- `DFGRD0(3,3)`: array containing the deformation gradient at the beginning of the increment
- `DFGRD1(3,3)`: array containing the deformation gradient at the end of the increment. The components of this array are set to zero when the nonlinear geometric effects are not included in the step definition associated with this increment
- `NOEL`: element number
- `NPT`: integration point number
- `LAYER`: layer number (for composite shells and layered solids)
- `KSPT`: section point number within the current layer
- `KSTEP`: step number

- `KINC`: increment number

In this work a user-defined material with 13 parameters has been used and so `NPROPS` will be 13 and the array `PROPS` is according the following convention:

- `PROPS(1)` $= E_L$
- `PROPS(2)` $= E_T$
- `PROPS(3)` $= G_L$
- `PROPS(4)` $= G_T$
- `PROPS(5)` $= \nu_{12}$
- `PROPS(6)` $= \nu_{23}$
- `PROPS(7)` $= n_2$
- `PROPS(8)` $= n_4$
- `PROPS(9)` $= n_6$
- `PROPS(10)` $= \sigma_T^Y$
- `PROPS(11)` $= \tau_T^Y$
- `PROPS(12)` $= \tau_L^Y$
- `PROPS(13)` $= \alpha_{23}$

Furthermore in all situations the user should define several variables, described in the following paragraphs.

DDSDDE(NTENS,NTENS)

Jacobian matrix of the constitutive model. `DDSDDE(I,J)` defines the change in the `I`$^{\text{th}}$ stress component at the end of the time increment caused by an infinitesimal perturbation of the `J`$^{\text{th}}$ component of the strain increment array. In most cases the Jacobian is symmetric and so Abaqus will consider this as default and will calculate the real Jacobian as one half the sum of the matrix and its transpose. If the Jacobian needs to be unsymmetrical the unsymmetrical equation solution capability of Abaqus must be called by the user.

STRESS(NTENS)

This array is passed in as the stress tensor at the beginning of the increment and must be updated in this routine to be the stress tensor at the end of the increment. This array will also contain the initial stresses at the start of the analysis (if any). In finite-strain problems the stress tensor has already been rotated to account for rigid body motion in the increment before UMAT is called. So that only the corotational part of the stress integration should be considered in UMAT. The value of the applied stress is the true (Cauchy) stress, in the order $\sigma_1, \sigma_2, \sigma_3, \sigma_{12}, \sigma_{13}, \sigma_{23}$. The relationship between actual stresses and future stresses is as follows:

$$\underline{\sigma}^{(t+\Delta t)} = \underline{\sigma}^{(t)} + \underline{\underline{J}} \cdot \underline{\Delta\sigma} \tag{5.33}$$

A possible way to write this expression in Fortran90 language is as follows:

Stress update example

```
DO row = 1, NTENS
    DO column = 1, NTENS
        STRESS(row) = STRESS(row) + DDSDDE(row, column) * DSTRAN(column)
    END DO
END DO
```

STATEV(NSTATV)

This is an array containing the solution-dependent state variables. These are passed in as the values at the beginning of the increment unless they are updated in other user subroutines (i.e. `USDFLD` or `UEXPAN`). In this case the updated values are passed in. In all cases `STATEV` must be returned as the values at the end of the increment.

In finite-strain problems every vector-valued or tensor-valued state variables must be rotated to account for rigid body motion of the material, in addition to any update in the values associated with constitutive behavior. The rotation increment matrix, `DROT`, is provided for this purpose.

SSE, SPD, SCD

Specific elastic strain energy, plastic dissipation, and “creep” dissipation, respectively. These are passed in as the values at the start of the increment and should be updated to the corresponding specific energy values at the end of the increment. They have no effect on the solution, except being for energy output.

Fully coupled thermal-stress analysis

In a fully coupled thermal-stress analysis the following variables must be updated by the user subroutine `UMAT`:

- `RPL`: volumetric heat generation per unit time at the end of the increment caused by mechanical work of the material
- `DDSDDT(NTENS)`: variation of the stress increments respect to the temperature
- `DRPLDE(NTENS)`: variation of `RPL` respect to the strain increments
- `DRPLDT`: variation of `RPL` respect to the temperature

Optional variables

`PNEWDT` is another variable that can be updated optionally by the user subroutine `UMAT`. It is the ratio of suggested new time increment to the time increment being used (`DTIME`). This variable allows providing input to the automatic time incrementation algorithms in Abaqus/Standard (if automatic time incrementation is chosen). For a quasi-static procedure the automatic time stepping that Abaqus/Standard uses, which is based on techniques for integrating standard creep laws ca not be controlled by `UMAT` subroutine.

`PNEWDT` is set to a large value before each call to `UMAT`. If `PNEWDT` is redefined to be less than 1.0, Abaqus/Standard must abandon the time increment and attempt to calculate it again with a smaller time increment. The suggested new time increment provided to the automatic time integration algorithms is `PNEWDT * DTIME`, where the `PNEWDT` used is the minimum value for all calls to user subroutines that allow redefinition of `PNEWDT` for this iteration.

If `PNEWDT` is given a value that is greater than 1.0 for all calls to user subroutines for this iteration and the increment converges in this iteration, Abaqus/Standard may increase the time increment. The suggested new time increment provided to the automatic time integration algorithms is `PNEWDT * DTIME`, where the `PNEWDT` used is the minimum value for all calls to user subroutines for this iteration.

If automatic time incrementation is not selected in the analysis procedure, values of `PNEWDT` that are greater than 1.0 will be ignored and values of `PNEWDT` that are less than 1.0 will cause the job to terminate.

5.6 Results

FEM analyses must be followed with a validation comparison between numerical results and experiments. In this study experiments done on non-crimp fabrics ply exactly as considered in this work are compared with numerical results with different levels of complexity.

At the beginning just linear elastic material model has been used because in low strains the material behaves almost linearly as there is no visible plastic contribution or any damage of fibers or matrix. From this comparison it is possible to compare one of the most important macroscopic parameters: the elastic modulus. If this value is very far from the experimentally measured elastic modulus, the performed numerical FE model would not have any significant meaning.

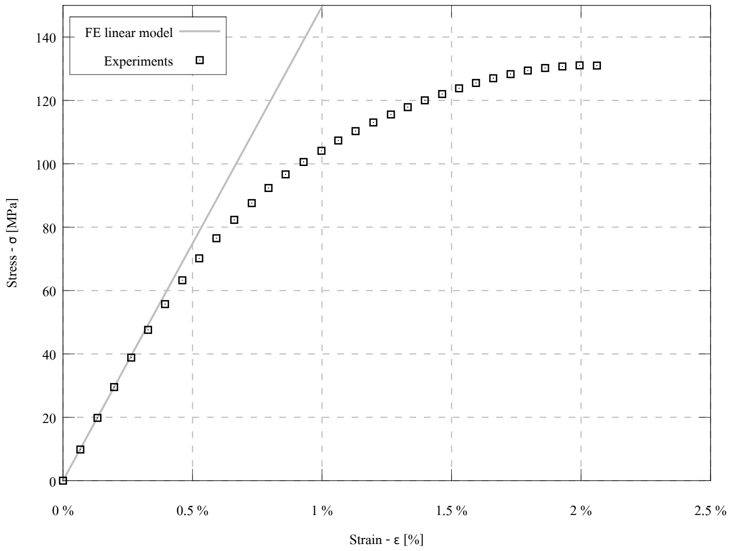


Figure 5.22: Non-crimp fabric experiments vs. FE linear modeling

In Figure 5.22 the experiments are compared directly with a linear numerical model analyzed with Abaqus/Standard introducing just elastic linear material models in both pure matrix and yarns. Up to an averaged strain level of 0.2% the difference between the numerical model and the experiments is negligible. The elastic modulus can be calculated as 14964 MPa.

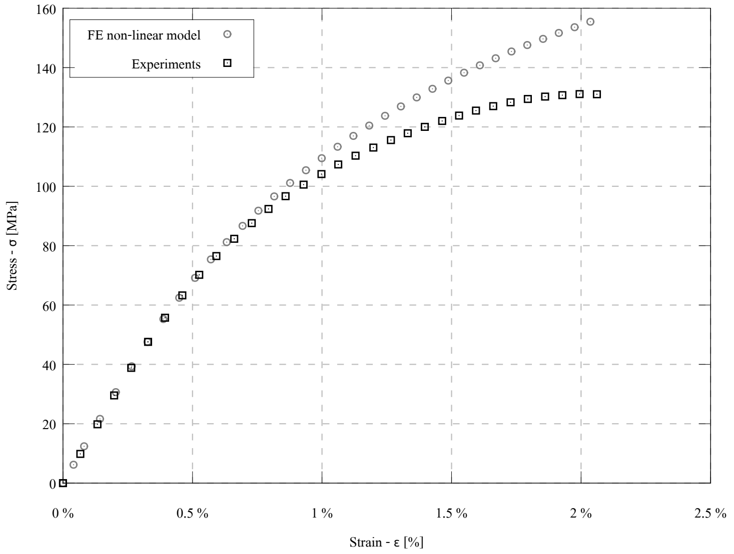


Figure 5.23: Non-crimp fabric experiments vs. FE non-linear modeling

In Figure 5.23 the non-linear behavior has been introduced. Now the numerical model is

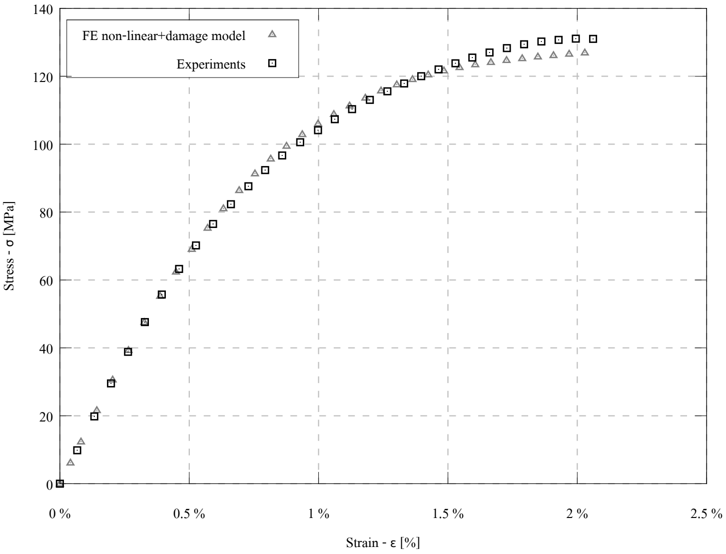


Figure 5.24: Non-crimp fabric experiments vs. FE non-linear with damage modeling

approximating the experiments better and the difference between the two curves is negligible up to 0.8% of averaged strain on the entire specimen. The introduced non-linearity gave a better inelastic prediction to the entire model, especially before the occurrence of damage. In fact since that time damage contribution starts to be relevant and so to make the difference between prediction and experiments.

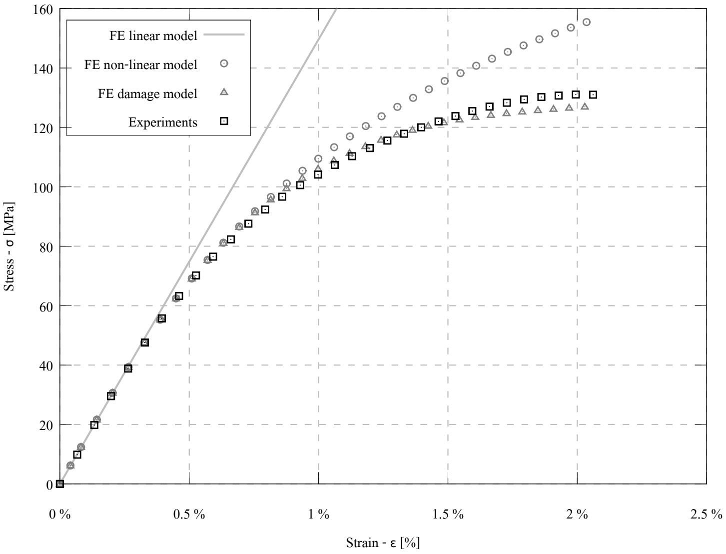


Figure 5.25: Non-crimp fabric experiments vs. different modeling approaches

Introducing the damage – as described in Section 5.2.4 – will lead to a more compliance of the material, once the damage occurs the material loses locally its capability to bear the load.

The introduction of damage allows a lower compliance of the material, especially since about 0.9%-1.0%. In fact since this point, the curve is more in agreement with the experiments respect to the one without damage. Moreover the modeled damage is introducing an excessive damage inside the numerical simulation. Therefore since about 1.5%, in an equal applied strain, the bearable load is lower. At the averaged strain level of 2% the difference between experiments and numerical simulation is about 3% that is a quite good result.

To have a comparison between the different approaches, Figure 5.25 is plotting all the curve herein described. It is now easier to see the behavior of the material after the introduction of the anisotropic non-linear behavior. And moreover how the damage evolution leads to even better results.

Chapter

6

Conclusions and comments

The present work deals with the study of digital image correlation technique. This is an attempt to find a better understanding of the meso-mechanical behavior of non-homogeneous materials that are progressively used in different industries, materials such as aluminum alloys, nodular cast iron, short fiber composites, woven composites are just few examples of which. Emphasizing on the importance of having a deep knowledge of the mechanical behavior of the employed materials in designing and producing different products, specially when we are interested in high performance materials, this study concentrates on investigations on strain as the most important response of the product against the applied loads. The focus of the study is on strain measurement and the most effective tools and means to measure it.

Though it starts with a literature review on different methods of strain measurement, explaining the most applicable methods such as extensometers, strain gages and moiré interferometry as former and traditional methods of strain measuring and continuing with digital image methods which are the novel methods of 2D and 3D strain measurement in few past decades.

There are many researches on developing digital image correlation method as a non-contact method to measure strain. However there is still the lack of enough fast, reliable and precise software to measure the strain especially when 2D or 3D strain map is required.

Concerning to this fact the main aim of this study is to develop a digital image correlation software able to be applied on images captured by a digital camera. This software which is called *Phoenix*© is based on a correlation coefficient minimization algorithm proposing a complete control on DIC method and the parameters influencing the results. The algorithm considers the speed of the analysis but mainly the precision of the results and robustness of the method respect to several parameters influencing the final results. A Lagrange formulation is used to calculate the strains from the displacement fields. *Phoenix*© introduces a smoothing technique. Applying such a technique is indispensable when numerical calculations deal with noisy digital images. The smoothing process is however well controllable by the user and it has been designed to alter the results as less as possible.

In order to see how it reacts respect to different kinds of digital images and to show how it can be useful in different situations, *Phoenix*© has been tested with different materials. Different magnifications were used to understand the sensibility of the method and the software.

Nodular cast iron specimens were tested at high magnification to reveal the plastic distribution at meso-level, namely around the graphite spherical nodules. These nodules, with an average diameter of about 30μm, react as stress intensification factors. Considering the small dimensions of the nodules, a high magnification optics and a small specimen should be used. Thanks to the fact that DIC method is able to work even by observing directly the surface of the specimen, it is easy to compare the strain map with the microstructure of the material directly. The results of the tensile tests of this material has been measured by DIC method and the 2D strain map of it has been presented by *Phoenix*©.

The same analysis has been done on silumin. This material, which is especially used in sport cars, is mainly an aluminum alloy with silicon particles inside. Even if macroscopically homogeneous, it is highly heterogeneous at microscopic observation. Thus again a high magnification optics should be applied. Again the digital images were taken directly from the specimen surface and the results, in terms of strain maps, could be compared directly with the microstructure. Here the dendritic nature of the material is compared with the strains localizations.

The study considers also the widespread application of composite materials. Even the macroscopic behavior of them is mainly due to the damage initiation and propagation inside the material. This is caused by the complex strain distribution inside the complex structure of the material. A direct observation of the phenomenon is very difficult and so numerical simulation approaches are preferable. However there is the problem of lack of reliability. Thus the digital image correlation can help to set the parameters of the numerical models in order to obtain more reliable results.

Comparisons between the simulated behavior of the composite and the measured behavior by the digital image correlation technique have been presented in this work. The employed composite material is a twill fabric carbon reinforced plastic under simple tensile test. The strain has been measured by both extensometer and DIC method. Of course the extensometer is just able to report the average strain, which were well compatible with DIC measured average strain. Here different magnifications and different experimental set-up have been used in order to have a wide range of situations.

In order to evaluate the reliability of *Phoenix*© software, the results of this software have been compared with the results of Vic-2D which is a commercial software to calculate the strain map. In this part of the study the captured images of a simple tensile test on twill fabric carbon reinforced plastic have been feed to both Vic-2D and *Phoenix*© software. The comparison of Vic-2D and *Phoenix*© shows a very high consistence between the results of these two software.

To perform an even more objective validation of *Phoenix*© software, a set of artificially created images have been used. Creating images with artificially applied displacement fields, the calculations performed by *Phoenix*© were tested. These comparisons showed a good agreement between the artificially imposed displacements and the calculated displacement of *Phoenix*© which is a successful evaluation of the precision and reliability of the software.

At the end of this study an FE numerical model of non-crimp fabric composites has been developed using anisotropic non-linear material description, considering the fact that DIC method can be used as a validation tool for numerical analyses of composite materials models. This FE model has been developed for a non-crimp fabric carbon reinforced plastic composite. The material parameters were set using traditional laboratory experiments. Damage criteria were used to define the initiation and subsequent evolution of the damage inside the material. Numerical analyses performed using models with different complexities were compared with macroscopic stress-strain curve obtained from experiments on specimens of the same materials. The comparison between the numerical simulation and the experimental results shows a good consistency which means the validity of the presented FE model.

6.1 Comments

In order to continue this study in future, the following recommends are suggested:

- designing the user interface of *Phoenix*© software for a more handy and easier approach to it
- evaluating *Phoenix*© software with more practical experiments with different material such as woven reinforced composites (e.g. plain, satin, etc.), short fibers reinforced composites, plastics, etc.
- evaluating *Phoenix*© software with different loading cases such as bending, pure shear, etc.
- evaluating the reliability of *Phoenix*© software applying dynamic loads
- comparing *Phoenix*© software with more commercial software such as Aramis from Trilion optical test systems (`http://www.trilion.com/Products/aramis.html`), Image correlation system from Dantec dynamics (`http://www.dantecdynamics.com`), etc.
- meso-scale experimental validation of the non-crimp fabric anisotropic non-linear numerical model presented in this study using DIC method

Bibliography

[1] Boeing. Boeing 787 dreamliner – program fact sheet. `http://www.boeing.com/commercial/787family/programfacts.html`, 2009.

[2] Masaru Zako, Yasutomo Uetsuji, and Tetsusei Kurashiki. Finite element analysis of damaged woven fabric composite materials. *Composites Science and Technology*, 63(3-4):507–516, February-March 2003.

[3] G. Nicoletto and E. Riva. Failure mechanisms in twill-weave laminates: Fem predictions vs. experiments. *Composites Part A: Applied Science and Manufacturing*, 35(7-8):787–795, July 2004.

[4] D.W.A. Rees. *Mechanics of Solids and Structures*. McGraw-Hill, 1990.

[5] Ailsa Allaby and Michael Allaby. A Dictionary of Earth Sciences. 1999. Encyclopedia.com. `http://www.encyclopedia.com/doc/1O13-strain.html`, 2009.

[6] Richard G. Budyans. *Advanced Strength and Applied Stress Analysis*. McGraw-Hill, 2$^{\text{nd}}$ edition, 1999.

[7] Illinois Tool Works Inc. Glossary of materials testing terms. `http://www.instron.us`, 2009.

[8] Charles Huston. The effect of continued and progressively increasing strain upon iron. *Journal of the Franklin Institute*, 107(1):41–45, January 1879.

[9] Vishay Micro-Measurements. *Errors Due to Transverse Sensitivity in Strain Gages*, 2007.

[10] Vishay Micro-Measurements. General Purpose Strain Gages – Linear Pattern – 125UN. `http://www.vishay.com/docs/11224/125un.pdf`, June 2009.

[11] Vishay Micro-Measurements. General Purpose Strain Gages – Tee Rosette – 250UT. `http://www.vishay.com/docs/11300/250ut.pdf`, June 2009.

[12] Vishay Micro-Measurements. General Purpose Strain Gages – Rectangular Rosette – 125UR. `http://www.vishay.com/docs/11225/125ur.pdf`, June 2009.

[13] Vishay Micro-Measurements. *Erros Due to Misalignment of Strain Gages*, 2007.

[14] Vishay Micro-Measurements. *Strain Gage Thermal Output and Gage Factor Variation with Temperature*, 2007.

[15] Brian Bowers. *Sir Charles Wheatstone.* The Institution of Engineering and Technology, 2nd edition, December 2001.

[16] Vishay Micro-Measurements. *Errors Due to Wheatstone Bridge Nonlinearity*, 2007.

[17] Vishay Micro-Measurements. *Shunt Calibration of Strain Gage Instrumentation*, 2007.

[18] Gianni Nicoletto. On the visualization of heterogeneous plastic strains by moiré interferometry. *Optics and Lasers in Engineering*, 37(4):433–442, April 2002.

[19] Daniel Post, Bongtae Han, and Peter Ifju. *High Sensitivity Moiré - Experimental Analysis for Mechanics and Materials.* Springer, 1994.

[20] Eric M. Weissman and Daniel Post. Moiré interferometry near the theoretical limit. *Applied Optics*, 21(9):1621–1623, May 1982.

[21] W.H. Peters and W.F. Ranson. Digital imaging techniques in experimental stress analysis. *Optical Engineering*, 21(3):427–431, May 1982.

[22] I. Amato. A new view of nature. `http://www.genomenewsnetwork.org/articles/2004/04/02/supervision.php`, 2003.

[23] J. Kim. Introduction to composite material. `http://notes.ump.edu.my/fkksa/FKKSA/Archive/2007&2008%20SEM%20I/DEGREE/BKC3363%20Science%20&%20Engineering%20Materials/Composites%20-%20Extra%20Reading%2002.pdf`, 2009.

[24] V. V. Vasiliev and E. V. Morozov. *Mechanics and Analysis of Composite Materials.* Elsevier Science Ltd. Elsevier Science Ltd. Elsevier Science Ltd., Amsterdam, 2001.

[25] D. Kopeliovich. Composites. `http://www.substech.com/dokuwiki/doku.php`, 2007.

[26] B. D. Agarwal and L. Broutman. *Analysis and Performance of Fiber Composites.* John Wiely & Sons, INC., USA, 1990.

[27] Harper. Carbon fiber systems. `http://www.harperintl.eu/technology/carbon-fiber-lines.html`, 2009.

[28] R. R. Hedge, A. Dahiya, and M. G. Kamath. Carbon fibers. `http://web.utk.edu/~mse/pages/Textiles/CARBON%20FIBERS.htm`, 2004.

[29] J. Soller. Weave definition. `http://www.solarcomposites.com/fabricchoice.html`, 2004.

[30] D. Mattsson, R. Joffe, and J. Varna. Methodology for characterization of internal structure parameters governing performance in ncf composites. *Composite Part B*, 38(1):44–57, 2007.

[31] R. Joffe, D. Mattsson, J. Modniks, and J. Varna. Compressive failure analysis of non-crimp fabric composites with large out-of-plane misalignment of fiber bundles. *Composites Part A: Applied Science and Manufacturing*, 36(8):1030–1046, 2005.

[32] K. Vallons. Carbon fibre non-crimp fabric composites for automotive applications. `http://www.mtm.kuleuven.be/Research/C2/poly/phds/kav/phd_kav.html`, 2005.

[33] H.A. Brucks, S.R. McNeil, M.A. Sutton, and W.H. Peters III. Digital imaging correlation using newton-raphson method of partial differential correction. *Experimental Mechanics*, 29(3):261–267, September 1989.

[34] G. Vendroux and W.G. Knauss. Submicron deformation field measurements: Part 2. improved digital image correlation. *Experimental Mechanics*, 38(2):86–92, June 1998.

[35] Zhi-Feng Zhang, Yi-Lan Kang, Huai-Wen Wang, Qing-Hua Qin, Yu Qiu, and Xiao-Qi Li. A novel coarse-fine search scheme for digital image correlation method. *Measurement*, 39 (8):710–718, October 2006.

[36] B. Wattrisse, A. Chrysochoos, J. M. Muracciole, and M. Némoz-Gaillard. Analysis of strain localization during tensile tests by digital image correlation. *Experimental Mechanics*, 41 (1):29–39, March 2001.

[37] Po-Chih Hung and A. S. Voloshin. In-plane strain measurement by digital image correlation. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 25(3): 215–221, September 2003.

[38] Tao Hua, Huimin Xie, Bing Pan, Xinlin Qing, Fulong Dai, and Xiqiao Feng. A new micro-tensile system for measuring the mechanical properties of low-dimensional materials–fibers and films. *Polymer Testing*, 26(4):513–518, June 2007.

[39] Wei Sun, Chenggen Quan, Cho Jui Tay, and Xiaoyouan He. Global and local coordinates in digital image correlation. *Applied Optics*, 46(7):1050–1056, March 2007.

[40] Wim Van Paepegem, Assen A. Shulev, Ilia R. Roussev, Stijn De Pauw, Joris Degrieck, and Ventseslav C. Sainov. Study of the deformation characteristics of window security film by digital image correlation techniques. *Optics and Lasers in Engineering*, 47(3-4):390–397, March-April 2009.

[41] Min Wang, Yuwan Cen, Xiaofang Hu, Xiaoliu Yu, Nenggang Xie, Yuguo Wu, Peimin Xu, and Deyi Xu. A weighting window applied to the digital image correlation method. *Optics & Laser Technology*, 41(2):154–158, March 2009.

[42] Peng Zhou and Kenneth E. Goodson. Subpixel displacement and deformation gradient measurement using digital image/speckle correlation (DISC). *Society of Photo-Optical Instrumentation Engineers*, 40(8):1613–1620, 2001.

[43] Jun Zhang, Guanchang Jin, Shaopeng Ma, and Libo Meng. Application of an improved subpixel registration algorithm on digital speckle correlation measurement. *Optics & Laser Technology*, 35(7):533–542, October 2003.

[44] Bing Pan, Anand Asundi, Huimin Xie, and Jianxin Gao. Digital image correlation using iterative least squares and pointwise least squares for displacement field and strain field measurements. *Optics and Lasers in Engineering*, 47(7-8):865–874, July-August 2009.

[45] Mark C. Pitter, Chung W. Seeand, and Michael G. Somekh. Subpixel microscopic deformation analysis using correlation and artificial neural networks. *Optic Express*, 8(6): 322–327, March 2001.

[46] Giancarlo Anzelotti. Bachelor thesis, 2004. University of Parma – Faculty of Engineering – Department of Industrial Engineering (in Italian).

[47] Alan Pilch, Aiay Mahajan, and Tsuchin Chu. Measurement of whole-field surface displacements and strain using a genetic algorithm based intelligent image correlation method. *Journal of Dynamic Systems, Measurement and Control*, 126(3):479–488, September 2004.

[48] Shun-Fa Hwang, Jhih-Te Horn, and Hou-Jiun Wang. Strain measurement of su-8 photoresist by a digital image correlation method with a hybrid genetic algorithm. *Optics and Lasers in Engineering*, 46(3):281–289, March 2008.

[49] Bing Pan. Reliability-guided digital image correlation for image deformation measurement. *Applied Optics*, 48(8):1535–1542, March 2009. URL `http://ao.osa.org/abstract.cfm?URI=ao-48-8-1535`.

[50] H. Lu and P. D. Cary. Deformation measurements by digital image correlation: Implementation of a second-order displacement gradient. *Experimental Mechanics*, 40(4):393–400, December 2000.

[51] Hubert W. Schreier and Michael A. Sutton. Systematic errors in digital image correlation due to undermatched subset shape functions. *Experimental Mechanics*, 42(3):303–310, September 2002.

[52] P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of computation*, 37(155):141–158, July 1981.

[53] Abraham Savitzky and Marcel J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639, July 1964.

[54] Bing Pan, Huimin Xe, Tao Hua, Weining Wang, and Yan Fang. Displacement smoothing and strain estimation using savitzky-golay filters in digital image correlation. *Key Engineering Materials*, 326-328:135–138, December 2006.

[55] Yaofeng Sun, John H. L. Pang, Chee Khuen Wong, and Fei Su. Finite element formulation for a digital image correlation method. *Applied Optics*, 44(34):7375–7363, December 2005.

[56] L. B. Meng, G. C. Jin, and X. F. Yao. Application of iteration and finite element smoothing technique for displacement and strain measurement of digital speckle correlation. *Optics and Lasers in Engineering*, 45(1):57–63, January 2007.

[57] G. Besnard, F. Hild, and S. Roux. Finite-element displacement fields analysis from digital images: application to Portevin-Le Châtelier bands. *Experimental Mechanics*, 46(6):789–803, December 2006.

[58] Bing Pan, Kemao Qian, Huiminin Xie, and Anand Asundi. Two-dimensional digital image correlation for in-plane displacement and strain measurement: a review. *Measurement Science & Technology*, 20(6):1–17, June 2009.

[59] R. Haman, F. Hild, and S. Roux. Stress intensity factor gauging by digital image correlation: application in cyclic fatigue. *Strain*, 43(3):181–192, August 2007.

[60] Joint Photographic Experts Group. Official website. `http://www.jpeg.org/`, 2009.

[61] Wikipedia. BMP file format — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/wiki/BMP_file_format`, 2009. [Online; accessed 30 November 2009].

[62] Portable Network Graphics. An open, extensible image format with lossless compression. `http://www.libpng.org/pub/png/`, 2009.

[63] Adobe. TIFF – Tagged Image File Format. `http://partners.adobe.com/public/developer/tiff/index.html`, 2009.

[64] Gianni Nicoletto, Tito Marin, Giancarlo Anzelotti, and Riccardo Roncella. Application of high magnification digital image correlation technique to micromechanical strain analysis. *Strain*, December 2008. In Press.

[65] Tito Marin and Gianni Nicoletto. Heterogeneous plastic deformation of polycrystals: visualization and modeling. In *International Conference on Full-Field Measurement Techniques and Their Applications in Experimental Solid Mechanics, Photomechanics*, Clermont-Ferrand, France, July 2006.

[66] ASM International The Materials Information Company. *ASM Handbook 9th edition – Volume 15 – Casting*. ASM International, 1998.

[67] Alberto Cabrio. Sviluppo di un apparato sperimentale per la misura in-situ delle deformazioni con il metodo della correlazione delle immagini digitali. Master's thesis, Università degli Studi di Parma – Facoltà di Ingegneria – Corso di studi in Ingegneria Meccanica, 2005–2006.

[68] ASTM International. Standard test method for tensile properties of polymer matrix composite materials. In *D3039/D3039M – 08*. ASTM, 2008.

[69] M. Valizadeh, S. A. H. Ravandi, M. Salimi, and M. Sheikhzadeh. Determination of internal mechanical characteristics of woven fabrics using the force-balance analysis of yarn pullout test. *Journal of the Textile Institute*, 99(1):47–55, 2008.

[70] Yves Beauchamp and Youssef A. Youssef. An effective approach to teach design of experiments (DOE) using calculation-and-analysis worksheets and computerized spreadsheets. *Computers & Industrial Engineering*, 35(3-4):643–646, December 1998.

[71] G.O. Verran, R.P.K. Mendes, and L.V.O. Dalla Valentina. Doe applied to optimization of aluminum alloy die castings. *Journal of Materials Processing Technology*, 200(1-3):120–125, May 2008.

[72] Hasan Sofuoglu. A technical note on the role of process parameters in predicting flow behavior of plasticine using design of experiment. *Journal of Materials Processing Technology*, 178(1-3):148–153, September 2006.

[73] W. Annicchiarico and M. Cerrolaza. Structural shape optimization 3d finite-element models based on genetic algorithms and geometric modeling. *Finite Elements in Analysis and Design*, 37(5):403–415, May 2001.

[74] J. Hilmann, M. Paas, A. Haenschke, and T. Vietor. Automatic concept model generation for optimisation and robust design of passenger cars. *Advances in Engineering Software*, 38(11-12):795–801, November-December 2007.

[75] Ebrahim Lamkanfi, Wim Van Paepegem, Joris Degrieck, Carla Ramault, Andreas Makris, and Danny Van Hemelrijck. Strain distribution in cruciform specimens subjected to biaxial loading conditions. part 1: Two-dimensional versus three-dimensional finite element model. *Polymer Testing*, 2009. In Press.

[76] Stepan V. Lomov, Dmitry S. Ivanov, Ignaas Verpoest, Masaru Zako, Tetsusei Kurashiki, Hiroaki Nakai, Jerome Molimard, and Alain Vautrin. Full-field strain measurements for validation of meso-fe analysis of textile composites. *Composites Part A: Applied Science and Manufacturing*, 39(8):1218–1231, August 2008.

[77] Gianni Nicoletto, Giancarlo Anzelotti, and Enrica Riva. Mesoscopic strain fields in woven composites: Experiments vs. finite element modeling. *Optics and Lasers in Engineering*, 47(3-4):352–359, March-April 2009.

[78] Giancarlo Anzelotti, Gianni Nicoletto, and Enrica Riva. Mesomechanic strain analysis of twill-weave composite lamina under unidirectional in-plane tension. *Composites Part A: Applied Science and Manufacturing*, 39(8):1294–1301, August 2008.

[79] Kevin M. Moerman, Cathy A. Holt, Sam L. Evans, and Ciaran K.Simms. Digital image correlation and finite element modelling as a method to determine mechanical properties of human soft tissue *in vivo*. *Journal of Biomechanics*, 42(8):1150–1153, May 2009.

[80] S.H. Ju, S.H. Liu, and K.W. Liu. Measurement of stress intensity factors by digital camera. *International Journal of Solids and Structures*, 43(5):1009–1022, March 2006.

[81] Jean Noël Périé, Hugo Leclerc, Stéphane Roux, and François Hild. Digital image correlation and biaxial test on composite material for anisotropic damage law identification. *International Journal of Solids and Structures*, 46(11-12):2388–2396, June 2009.

[82] A. Willems, S.V. Lomov, I. Verpoest, and D. Vandepitte. Drape-ability characterization of textile composite reinforcements using digital image correlation. *Optics and Lasers in Engineering*, 47(3-4):343–351, March-April 2009.

[83] A. Willems, S.V. Lomov, I. Verpoest, and D. Vandepitte. Optical strain fields in shear and tensile testing of textile reinforcements. *Composites Science and Technology*, 68(3-4): 807–819, March 2008.

[84] K. Vanclooster and S.V. Lomov andI. Verpoest. Experimental validation of forming simulations of fabric reinforced polymers using an unsymmetrical mould configuration. *Composites Part A: Applied Science and Manufacturing*, 40(4):530–539, April 2009.

[85] S.V. Lomov, Ph. Boisse, E. Deluycker, F. Morestin, K. Vanclooster, D. Vandepitte, I. Verpoest, and A. Willems. Full-field strain measurements in textile deformability studies. *Composites Part A: Applied Science and Manufacturing*, 39(8):1232–1244, August 2008.

[86] N. K. Naik and P. S. Shembekar. Elastic behavior of woven fabric composites: I-lamina analysis. *Journal of Composite Materials*, 26(15):2196–2225, January 1992.

[87] N. K. Naik and P. S. Shembekar. Elastic behavior of woven fabric composites: Ii-laminate analysis. *Journal of Composite Materials*, 26(15):2226–2246, January 1992.

[88] T. Ishikawa and T. W. Chou. Stiffness and strength behaviour of woven fabric composites. *Journal of Materials Science*, 17(11):3211–3220, November 1982.

[89] Ph. Vandeurzen, J. Ivens, and I. Verpoest. A three-dimensional micromechanical analysis of woven-fabric composites: I. geometric analysis. *Composites Science and Technology*, 56 (11):1303–1315, 1996.

[90] Ph. Vandeurzen, J. Ivens, and I. Verpoest. A three-dimensional micromechanical analysis of woven-fabric composites: Ii. elastic analysis. *Composites Science and Technology*, 56 (11):1317–1327, 1996.

[91] D. Blackketter, D. Walrath, and A. Hansen. Modeling damage in a plain weave fabric-reinforced composite material. *Journal of Composites Technology and Research*, 15(2): 136–142, June 1993.

[92] Kyeongsik Woo and John Whitcomb. Global/local finite element analysis for textile composites. *Journal of Composite Materials*, 28(14):1305–1321, August 1994.

[93] Kyeongsik Woo and John Whitcomb. Three-dimensional failure analysis of plain weave textile composites using a global/local finite element method. *Journal of Composite Materials*, 30(9):984–1003, June 1996.

[94] E. Riva. *Mechanics of woven fabric composites.* PhD thesis, University of Pisa, 1999.

[95] M. Guagliano and E. Riva. Mechanical behaviour prediction in plain weave composites. *Journal of strain analysis*, 36(2):153–162, 2001.

[96] Ignaas Verpoest and Stepan V. Lomov. Virtual textile composites software WiseTex: Integration with micro-mechanical, permeability and structural analysis. *Composites Science and Technology*, 65(15-16):2563–2574, December 2005.

[97] G. A. Bibo, P. J. Hogg, and M. Kemp. Mechanical characterisation of glass- and carbon-fibre-reinforced composites made with non-crimp fabrics. *Composites Science and Technology*, 57(9-10):1221–1241, 1997.

[98] K. H. Leong, S. Ramakrishna, Z. M. Huang, and G. A. Bibo. The potential of knitting for engineering composites – a review. *Composites Part A: Applied Science and Manufacturing*, 31(3):197–220, March 2000.

[99] S. V. Lomov, E. B. Belov, T. Bischoff, S. B. Ghosh, T. Truong Chi, and I. Verpoest. Carbon composites based on multiaxial multiply stitched preforms. Part 1. Geometry of the preform. *Composites Part A: Applied Science and Manufacturing*, 33(9):1171–1183, September 2002.

[100] Thanh Chi Truong, Matteo Vettori, Stepan Lomov, and Ignaas Verpoest. Carbon composites based on multi-axial multi-ply stitched preforms. Part 4. Mechanical properties of composites and damage observation. *Composites Part A: Applied Science and Manufacturing*, 36(9):1207–1221, September 2005.

[101] A. P. Mouritz, K. H. Leong, and I. Herszberg. A review of the effect of stitching on the in-plane mechanical properties of fibre-reinforced polymer composites. *Composites Part A: Applied Science and Manufacturing*, 28(12):979–991, 1997.

[102] David Mattsson, Roberts Joffe, and Janis Varna. Methodology for characterization of internal structure parameters governing performance in NCF composites. *Composites Part B: Engineering*, 37(1):44–57, January 2007.

[103] D.S. Mikhaluk, T.C. Truong, A.I. Borovkov, S.V. Lomov, and I. Verpoest. Experimental observations and finite element modelling of damage initiation and evolution in carbon/epoxy non-crimp fabric composites. *Engineering Fracture Mechanics*, 75(9):2751–2766, June 2008.

[104] W. Sun and A.C.W. Lau. Three-dimensional constitutive model for anisotropic inelastic deformation of unidirectional reinforced composites. *Journal of Thermoplastic Composite Materials*, 15(6):477–496, November 2002.

[105] M. M. Aghdam, D. J. Smith, and M. J. Pavier. Finite element micromechanical modelling of yield and collapse behaviour of metal matrix composites. *Journal of the Mechanics and Physics of Solids*, 48(3):499–528, March 2000.

[106] Ala Tabiei and Ivelin Ivanov. Materially and geometrically non-linear woven composite micro-mechanical model with failure for finite element simulations. *International Journal of non-linear mechanics*, 39(2):175–188, March 2004.

[107] Stepan V. Lomov, Dmitry S. Ivanov, Ignaas Verpoest, Masaru Zako, Tetsusei Kurashiki, Hiroaki Nakai, and Satoru Hirosawa. Meso-fe modelling of textile composites: Road map, data flow and algorithms. *Composites Science and Technology*, 67(9):1870–1891, July 2007.

[108] V. Carvelli and A. Taliercio. A micromechanical model for the analysis of unidirectional elastoplastic composites subjected to 3d stresses. *Mechanics Research Communications*, 26(5):547–553, September-October 1999.

[109] V. Carvelli and C. Poggi. A homogenization procedure for the numerical analysis of woven fabric composites. *Composites Part A: Applied Science and Manufacturing*, 32(10):1425–1432, October 2001.

[110] Walter Ramberg and William R. Osgood. Description of stress-strain curves by three parameters – technical note no. 902. Technical report, National Advisory Committee For Aeronautics, Washington DC., July 1943.

[111] Krishan Kumar Chawla. *Composite materials: science and engineering*. Springer, 1998.

[112] Oscar Hoffman. The brittle strength of orthotropic materials. *Journal of Composite Materials*, 1(2):200–206, 1967.

[113] K. Searles, G. Odegard, and M. Kumosa. Micro- and mesomechanics of 8-harness satin woven fabric composites: I - evaluation of elastic bahavior. *Composites Part A: Applied Science and Manufacturing*, 32(11):1627–1655, November 2001.

[114] ©ABAQUS Inc. *ABAQUS Analysis User's Manual - Version 6.7*, 2007.

[115] K.Madesn, H.B.Nielsen, and O.Tingleff. *Methods for non-linear least squares problems.* Technical University of Denmark, 2004.

[116] William S. Russell. Polynomial interpolation schemes for internal derivative distributions on structured grids. *Applied numerical mathematics*, 17(2):129–171, May 1995.

[117] Gilbert Strang. *Introduction to Linear Algebra.* Wellesley Cambridge Press, 4th edition, February 2009.

[118] Rogeer A. Horn and Charles R. Johnson. *Matrix analysis.* Cambridge University Press, 1985.

[119] Gene H. Golub and Charles F. Van Loan. *Matrix computations.* The Johns Hopkins University Press, 3rd edition, October 1996.

List of Figures

List of Tables

Appendix

A

Correlation coefficient

Generally speaking, the correlation coefficient is a parameter indicating the strength and direction of a linear relationship between two random variables. There are several coefficients able to measure the degree of correlation. A common way to evaluate the correlation is the *Pearson product-moment correlation coefficient*

$$C_{X,Y} = \frac{\mu(X \cdot Y) - \mu(X) \cdot \mu(Y)}{\sqrt{\mu(X^2) - \mu^2(X)}\sqrt{\mu(Y^2) - \mu^2(Y)}} \tag{A.1}$$

The correlation is 1 in case of an increasing linear relationship and -1 in case of a decreasing linear relationship. In case the variables are independent, the correlation will be zero.

A.1 Cross-correlation

The cross-correlation is a measure of similarity of two waveforms as a function of a "shift" to apply to one of them. For continuous functions $f = f(t)$ and $g = g(t)$, the cross-correlation is defined as

$$(f * g)(t) = \int_{-\infty}^{\infty} f^*(\tau) \cdot g(t + \tau) \, \mathrm{d}\tau \tag{A.2}$$

where f^* denotes the complex conjugate of f.

For image processing is usually necessary a normalization of the images, in order to avoid small changes in brightness. This can be done by subtracting the mean and dividing by the standard deviation

$$C_{f,t} = \frac{1}{m \cdot n - 1} \sum_{x=x_1}^{x_2} \sum_{y=y_1}^{y_2} \frac{\big(f(x,y) - \bar{f}\big) \cdot \big(t(x,y) - \bar{t}\big)}{\sigma_f \cdot \sigma_t} \tag{A.3}$$

where t represents the template picture and f the sub-image of dimension $m \times n$ pixels. $\bar{f}$ and $\bar{t}$ are the mean values and σ_f, σ_t are the standard deviations.

A.2 Sum of squared differences

In image processing is usually defined a parameter of correlation slightly different from the cross-correlation aforementioned. Is mainly used the *sum of squared differences* (SSD)

$$SSD(f,t) = \|t-f\|^2 = \|t\|^2 + \|f\|^2 - 2 \cdot f \cdot t \tag{A.4}$$

In order to reduce the influence of the contrast[1] is better to use a normalized version of this coefficient, the *normalized sum of squared differences* (NSSD)

$$NSSD(f,t) = \frac{\|t-f\|^2}{\|t\| \cdot \|f\|} \tag{A.5}$$

If also the brightness[2] can be a problem, is possible to use another difinition of SSD that uses zero mean functions, the *zero mean normalized sum of squared differences* (ZNSSD)

$$ZNSSD(f,t) = \frac{\left\|(t-\bar{t}) - \left(f-\bar{f}\right)\right\|^2}{\|t-\bar{t}\| \cdot \left\|f-\bar{f}\right\|} \tag{A.6}$$

All of these coefficients are in the range $[0,\infty[$ where zero means a perfect matching between f and t, so is the value that is necessary to reach (usually by a minimization approach).

---

[1] *Contrast* is the difference in visual properties that makes an object distinguishable from other objects and the background. A common definition is based on the ratio between a difference of luminances (like the maximum and the minimum) and the average luminance

[2] *Brightness* is an attribute of visual perception in which a source appears to be radiating or reflecting light. In the RGB color space, brightness can be thought of as the arithmetic mean of the red, green, and blue color coordinates

Appendix B

Minimization of a function

Minimization means seeking to minimize a function choosing the best set of values of the variables. This lead to a couple of important definitions

Definition 2. *A global minimum is the value of the variables x that* ***globally*** *minimize the function $f : \mathbb{R}^n \rightarrow \mathbb{R}$*

$$x^* = argmin_x \left\{ f(x) \right\}$$

Definition 3. *A local minimum is the value of the variables x that* ***locally*** *minimize the function $f : \mathbb{R}^n \rightarrow \mathbb{R}$*

$$f(x^*) \leq f(x) \qquad \mathit{for} \qquad \|x - x^*\| < \delta$$

where δ is a positive number that defines a small region around x^*.

The global minimum is usually the best result, but also quite hard to obtain, if not impossible [115]. Trying to find out the minimum (always local in this dissertation) we can start from the Taylor expansion of the function f, which is differentiable for hypothesis

$$f(\underline{x} + \underline{h}) = f(\underline{x}) + \underline{h}^T\, \underline{J} + \frac{1}{2} \underline{h}^T\, \underline{\underline{H}}\, \underline{h} + O(\|\underline{h}\|^3) \tag{B.1}$$

where $\|\underline{h}\|$ is the L2-norm[1] of $\underline{h}$.

---

[1] In real vector space $\mathbb{R}^n$, the L2-norm correspond to the usual Euclidean norm: $\|\underline{v}\| = \sqrt{\sum_{i=1}^{n} v_i^n}$

The $\underline{J}$ is the gradient of f (also called Jacobian[2]) and $\underline{\underline{H}}$ is the Hessian matrix

$$\underline{J} = \nabla f = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix} \tag{B.2}$$

$$\underline{\underline{H}} = \nabla\nabla f = \left[\frac{\partial^2 f(x)}{\partial x_i \partial x_j}\right]_{i,j} \tag{B.3}$$

If $\underline{x}^*$ is a local minimum for the function f a necessary condition for a local minimum is that $\underline{J}(\underline{x}^*) = 0$ and this point will be called *stationary point* of f. This is only a necessary but not sufficient condition. In fact also a local maximizer is a stationary point[3]. But expanding f around $\underline{x}^*$ we have $\underline{J} = 0$ and so the Equation (B.1) will be dominated by the fourth term that actually is dominated by the Hessian matrix. This leads to the following sufficient condition

Theorem 1. *Assuming that $\underline{x}^*$ is a stationary point than is a sufficient condition for a local minimum that $\underline{\underline{H}}$ is positive defined.*

In the following paragraphs will be explained some iterative methods to find the local minimum.

B.1 Steepest descent method

The steepest descent method (also called gradient method) is based on the observation that steepest direction toward the minimum is given by the gradient vector. Is possible to demonstrate this assumption by considering the first order Taylor expansion of a function f around $\underline{x}$

$$f(\underline{x} + \alpha\, \underline{\delta}) \approx f(\underline{x}) + \alpha\, \underline{\delta}^T\, \underline{J} \tag{B.4}$$

where is possible to find the gain in going toward the minimum using α

$$\lim_{\alpha \to 0} \frac{f(\underline{x}) - f(\underline{x} + \alpha\, \underline{\delta})}{\alpha\, \|\delta\|} = -\frac{1}{\|\delta\|} \cdot \underline{\delta}^T\, \underline{J} = -\, \|\underline{J}\| \cdot cos(\theta) \tag{B.5}$$

where θ is the angle between the vector $\underline{\delta}$ and the Jacobian. Is then obvious that the maximum gain is when $\theta = \pi$.

This leads to the conclusion that a good value of the increment $\underline{\delta}$ is the following

$$\underline{\delta} = -\underline{J} \tag{B.6}$$

---

[2] The Jacobian matrix is the matrix of all first-order partial derivatives of a vector-valued function. In this way it generalized the gradient of a scalar valued function of multiple variables

[3] A stationary point that is neither local minimum nor a local maximizer is called *saddle point* [115]

This method is locally very good and has a good convergence speed at initial steps. But is very slow when closer to the minimum.

B.2 Newton's method

Starting from the idea that for every stationary point the Jacobian is zero, we can expand it by Taylor series

$$\begin{aligned} \underline{J}(\underline{x}+\underline{\delta}) &= \underline{J}(\underline{x}) + \underline{\underline{H}}(\underline{x}) \cdot \underline{\delta} + O(\|\underline{\delta}\|) \\ &\approx \underline{J}(\underline{x}) + \underline{\underline{H}}(\underline{x}) \cdot \underline{\delta} \end{aligned} \tag{B.7}$$

Solving for $\underline{J}(\underline{x}^*) = 0$ we derive the Newton's method

$$\underline{\underline{H}} \cdot \underline{\delta} = -\underline{J} \tag{B.8}$$

Newton's method is very good in the final stage of the iteration, where $\underline{x}$ is close to $\underline{x}^*$. Is possible to demonstrate that Newton's method has quadratic convergence close to the solution.

B.3 Levenberg-Marquardt method

This method, part of the general field of nonlinear minimization[4], is mainly similar to the Newton's method described in Appendix B.2. The improvement derives from the understanding that $\underline{\delta}$ in both Equation (B.6) and Equation (B.8) gives just the direction toward the minimum, not how much move in that direction.

Introducing a scale parameter α, the same equations will become

$$\underline{\delta}_{\text{Newton}} = -\alpha \cdot \underline{\underline{H}}^{-1} \cdot \underline{J} \tag{B.9}$$

$$\underline{\delta}_{\text{Steepest descent}} = -\alpha \cdot \underline{\underline{I}} \cdot \underline{J} \tag{B.10}$$

The approach proposed by Marquardt is to introduce a parameter on the diagonal of the Hessian matrix (like α) in order to be able to switch easily between the Newton's method and the Steepest descent method:

$$\underline{\delta} = \left[\underline{\underline{H}} + \lambda \cdot diag\left(\underline{\underline{H}}\right)\right]^{-1} \cdot \underline{J} \tag{B.11}$$

where

$$diag\left(\underline{\underline{H}}\right)_{ij} = \begin{cases} 0 & i \neq j \\ \underline{\underline{H}}_{ij} & i = j \end{cases} \tag{B.12}$$

---

[4] In least squares minimization, a problem is defined linear or nonlinear depending on the relation between the function to minimize and the parameters from which it depends

Is easy to see that if λ is small then the equation will be close to the one of Newton's method, and if λ is big then the matrix multiplying the Jacobian will be diagonally dominant, as in the Steepest descent method. A linear search algorithm is used to decide which λ minimize the function. The present method can be described in pseudocode as follows:

Algorithm B.1 Levenberg-Marquardt minimization algorithm

```
Require: matrix $\underline{\underline{H}}$ and vector $\underline{J}$
Ensure: best guess of $\underline{x}$

$\underline{x} \leftarrow$ initial guess of $\underline{x}$
$\lambda \leftarrow 0.001$
while $f(\underline{x}) \geq tolerance$ do
    $\underline{\delta} \leftarrow \left[\underline{\underline{H}} + \lambda \cdot diag\left(\underline{\underline{H}}\right)\right]^{-1} \cdot \underline{J}$
    if $f(\underline{x} + \underline{\delta}) \geq f(\underline{x})$ then
        $\lambda \leftarrow \lambda \cdot 10$
    else
        $\lambda \leftarrow \lambda/10$
        $\underline{x} \leftarrow \underline{x} + \underline{\delta}$
    end if
end while
```

Appendix

C

Interpolation

In the field of numerical analysis, interpolation is a mathematical method of constructing new data points within the range of a discrete set of known data points. There are many methods for doing this, many of which involve fitting some sort of function (called *interpolant function*) to the data and evaluating that function at the desired point, or getting them by some statistical method.

Considering n distinct couples of numbers $P_i \equiv (x_i, y_i)$, looking for the interpolant function f, it means looking for

$$f(x_i) = y_i, \forall i \in [1, n] \tag{C.1}$$

The interpolation process means getting the value of f at a generic value of the abscissa (i.e. the coordinate x) within x_1 and x_n.

C.1 1D interpolation

The first interpolation scheme is of course the mono-dimensional one. In this case we consider an interpolant function $f : \mathbb{R} \to \mathbb{R}$, which must fit points in the bi-dimensional space, $P \equiv (x_P, y_P)$.

C.1.1 Linear interpolation

The simplest method of interpolation is the linear interpolation. It builds several interpolant functions as straight lines between each couple of data points. If we consider the value of f calculated at $x \in [x_i, x_{i+1}]$, we'll get

$$f(x) = f(x_i) + \frac{x - x_i}{x_{i+1} - x_i} \cdot (y_{i+1} - y_i) \tag{C.2}$$

where is easy to see that $f(x_i) = y_i$ and $f(x_{i+1}) = y_{i+1}$.

This is an easy calculation, but also quite raft and especially it has the problem that the interpolant function is not differentiable at any x_i.

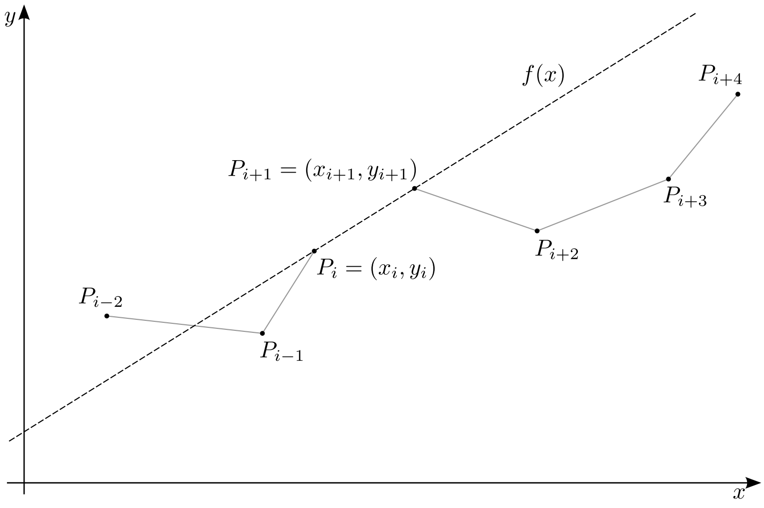


Figure C.1: Linear interpolation of a one-dimensional function

C.1.2 Polynomial interpolation

In order to achieve a better inerpolation of the data points, we can consider to increase the degree of the interpolant function, in the form

$$f(x) = \sum_{j=0}^{j=N} a_j \cdot x^j \tag{C.3}$$

where N is the degree of the function and a_j are coefficients.

Generally from n data points there is a polynomial of at most degree $n-1$ that exactely goes throw all data points (as there are n coefficients). Moreover, as a polynomial is infinitely differentiable, solving one of the most important problems of linear interpolation.

A simple approach is solving n linear equations (one each point) in the form

$$\begin{cases} a_0 + a_1 \cdot x_1 + a_2 \cdot x_1^2 + \cdots + a_{n-1} \cdot x_1^{n-1} = y_1 \\ a_0 + a_1 \cdot x_2 + a_2 \cdot x_2^2 + \cdots + a_{n-1} \cdot x_2^{n-1} = y_2 \\ \vdots \\ a_0 + a_1 \cdot x_n + a_2 \cdot x_n^2 + \cdots + a_{n-1} \cdot x_n^{n-1} = y_n \end{cases} \tag{C.4}$$

This system of equations can be written also in a matricial way, as follows

$$\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^{n-1} \end{bmatrix}^{-1} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \tag{C.5}$$

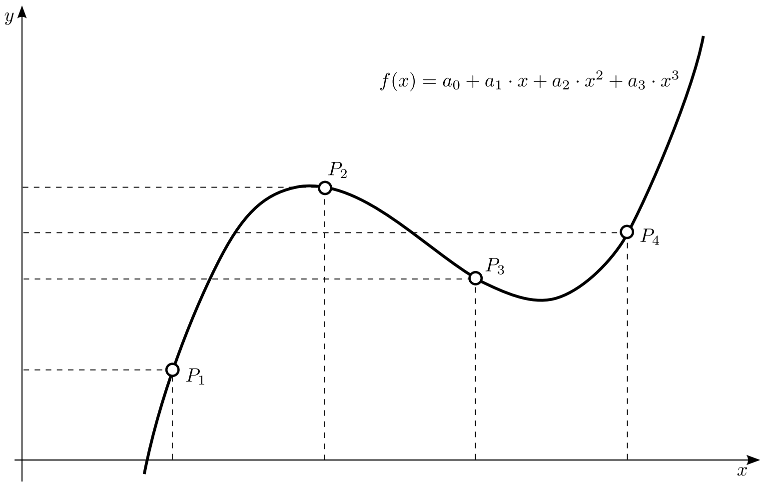


Figure C.2: Polynomial interpolation of a one-dimensional function defined by 4 points

Anyway the polynomial interpolation can be quite complex to calculate and can rise to the Runge's phenomenon (see Appendix C.4).

C.1.3 Spline interpolation

In order to avoid the Runge's phenomenon we consider an approach similar to linear interpolation, where several interpolant functions are built between each couple of data points. But in this case low degree polynomial functions will be designed to fit smoothly together.

A typical function is the *natural cubic spline*, which is a piecewise cubic function (thus it is two times differentiable)

$$f(x) = \sum_{j=0}^{3} a_j \cdot x^j \tag{C.6}$$

where a_j are coefficients, to get from the given data points.

In particular, starting from n points (and so $n-1$ intervals), there will be $2 \cdot (n-1)$ equations from the passage of f throw the n points, n equations involving the first derivative of f and $n-2$ equations involving the second derivative of f (in total there will be $4 \cdot (n-1)$ equations, as the number of unknowns).

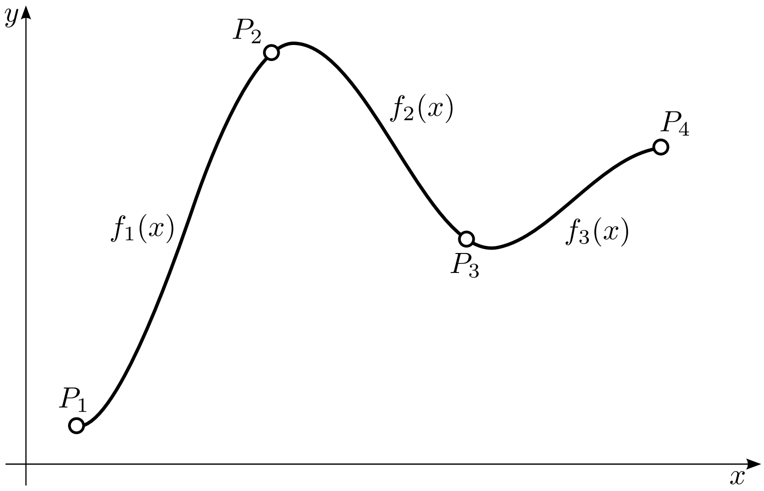


Figure C.3: Cubic spline interpolation of a one-dimensional function defined by 4 points

For example considering zero the first derivatives of f at the ending points, we'll obtain

$$f_i(x_j) = y_j \qquad i = 1, \dots, n-1 \qquad j = 1, \dots, n \tag{C.7}$$

$$\left.\frac{\partial f_1(x)}{\partial x}\right|_{x=x_1} = 0 \tag{C.8}$$

$$\left.\frac{\partial f_{n-1}(x)}{\partial x}\right|_{x=x_n} = 0 \tag{C.9}$$

$$\left.\frac{\partial f_i(x)}{\partial x}\right|_{x=x_{i+1}} = \left.\frac{\partial f_{i+1}(x)}{\partial x}\right|_{x=x_{i+1}} \qquad i = 1, \dots, n-2 \tag{C.10}$$

$$\left.\frac{\partial^2 f_i(x)}{\partial x^2}\right|_{x=x_{i+1}} = \left.\frac{\partial^2 f_{i+1}(x)}{\partial x^2}\right|_{x=x_{i+1}} \qquad i = 1, \dots, n-2 \tag{C.11}$$

C.2 Multivariate interpolation

The multivariate interpolation (also called spatial interpolation) is a kind of interpolation made by functions of more than one variable. As in the mono-dimensional version (see Appendix C.1), we know the value of a not given function at some points $P_i \equiv (x_i, y_i, z_i, \dots)$ and we want to evaluate the function at arbitrary points $P \equiv (x, y, z, \dots)$.

In this chapter it will be considered only a two-dimensional *regular grid*[1].

C.2.1 Bilinear interpolation

This is an extension of linear interpolation (see Appendix C.1.1) for interpolating functions of two variables on a regular grid.

---

[1] A regular grid is a tessellation of the Euclidean plane by congruent rectangles

Mainly is a linear interpolation before in one dimension and then in the second one. Let's consider four points on a regular grid, so that $P_{1,1} \equiv (x_1, y_1)$, $P_{1,2} \equiv (x_1, y_2)$, $P_{2,2} \equiv (x_2, y_2)$ and $P_{2,1} \equiv (x_2, y_1)$, with $x_2 > x_1$ and $y_2 > y_1$. First of all consider the two linear interpolations along x:

$$f(x, y_1) = \frac{x_2 - x}{x_2 - x_1} \cdot f(x_1, y_1) + \frac{x - x_1}{x_2 - x_1} \cdot f(x_2, y_1) \tag{C.12}$$

$$f(x, y_2) = \frac{x_2 - x}{x_2 - x_1} \cdot f(x_1, y_2) + \frac{x - x_1}{x_2 - x_1} \cdot f(x_2, y_2) \tag{C.13}$$

and then finally

$$f(x, y) = \frac{y_2 - y}{y_2 - y_1} \cdot f(x, y_1) + \frac{y - y_1}{y_2 - y_1} \cdot f(x, y_2) \tag{C.14}$$

Considering the unit square (i.e. the one where $x_1 = y_1 = 0$ and $x_2 = y_2 = 1$ is possible to write

$$f(\tilde{x}, \tilde{y}) = \begin{vmatrix} 1 - \tilde{x} & \tilde{x} \end{vmatrix} \cdot \begin{vmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{vmatrix} \cdot \begin{vmatrix} 1 - \tilde{y} \\ \tilde{y} \end{vmatrix} \tag{C.15}$$

easily achieved by the following substitutions

$$\tilde{x} = \frac{x - x_1}{x_2 - x_1} \tag{C.16}$$

$$\tilde{y} = \frac{y - y_1}{y_2 - y_1} \tag{C.17}$$

As it is possible to see from Equation (C.15), the bilinear interpolation rises to a non-linear interpolant function. Especially is linear along straight line parallel to the x or y axis, and quadratic otherwise. This is obvious considering that it is an interpolating surface passing by four points. Considering that a linear function in a three-dimensional space is a plane and it cannot pass by four points, thus it must be at least a quadratic function.

C.2.2 Bicubic spline interpolation

This is an extension of cubic spline interpolation (see Appendix C.1.3) for interpolating functions of two variables on a regular grid. Similarly as in Appendix C.1.3, the bi-dimensional spline function is

$$f(x, y) = \sum_{j=0}^{3} \sum_{k=0}^{3} a_{j,k} \cdot x^j \cdot y^k \tag{C.18}$$

where $a_{j,k}$ are 16 coefficients. In order to find out these coefficients, let's consider the unit square, so is possible to write equations based on the information given us by the data points:

$$f(0,0) = a_{0,0} \tag{C.19a}$$

$$f(1,0) = \sum_{j=0}^{3} a_{j,0} \tag{C.19b}$$

$$f(0,1) = \sum_{k=0}^{3} a_{0,k} \tag{C.19c}$$

$$f(1,1) = \sum_{j=0}^{3}\sum_{k=0}^{3} a_{j,k} \tag{C.19d}$$

$$f_x(0,0) = a_{1,0} \tag{C.19e}$$

$$f_x(1,0) = a_{1,0} + 2a_{2,0} + 3a_{3,0} \tag{C.19f}$$

$$f_x(0,1) = \sum_{k=0}^{3} a_{1,k} \tag{C.19g}$$

$$f_x(1,1) = \sum_{j=1}^{3}\sum_{k=0}^{3} j \cdot a_{j,k} \tag{C.19h}$$

$$f_y(0,0) = a_{0,1} \tag{C.19i}$$

$$f_y(1,0) = \sum_{j=0}^{3} a_{j,1} \tag{C.19j}$$

$$f_y(0,1) = a_{0,1} + 2a_{0,2} + 3a_{0,3} \tag{C.19k}$$

$$f_y(1,1) = \sum_{j=0}^{3}\sum_{k=1}^{3} k \cdot a_{j,k} \tag{C.19l}$$

$$f_{xy}(0,0) = a_{1,1} \tag{C.19m}$$

$$f_{xy}(1,0) = a_{1,1} + 2a_{2,1} + 3a_{3,1} \tag{C.19n}$$

$$f_{xy}(0,1) = a_{1,1} + 2a_{1,2} + 3a_{1,3} \tag{C.19o}$$

$$f_{xy}(1,1) = \sum_{j=1}^{3}\sum_{k=1}^{3} j \cdot k \cdot a_{j,k} \tag{C.19p}$$

where

$$f_x = \frac{\partial f}{\partial x} \tag{C.20}$$

$$f_y = \frac{\partial f}{\partial y} \tag{C.21}$$

$$f_{xy} = \frac{\partial^2 f}{\partial x \partial y} \tag{C.22}$$

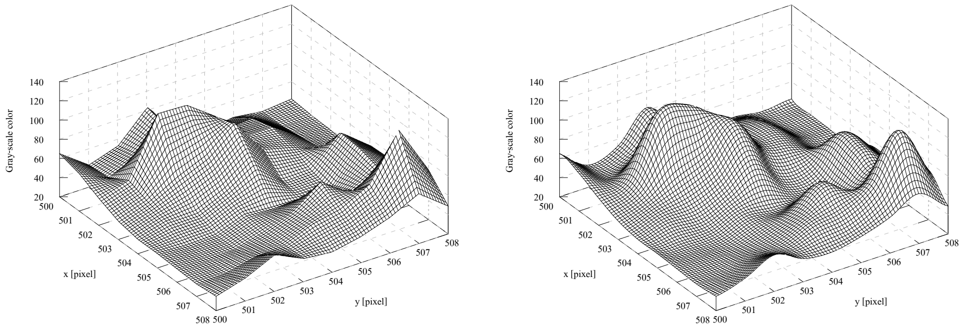
Finding the roots of this system of 16 unknowns in 16 equations is an easy task, due to the linear nature of the system:

$$
\begin{align}
a_{0,0} &= f(0,0) \tag{C.23a}\\
a_{0,1} &= f_y(0,0) \tag{C.23b}\\
a_{0,2} &= 3\,[f(0,1)-f(0,0)]-2f_y(0,0)-f_y(0,1) \tag{C.23c}\\
a_{0,3} &= 2\,[f(0,0)-f(0,1)]+f_y(0,0)+f_y(0,1) \tag{C.23d}\\
a_{1,0} &= f_x(0,0) \tag{C.23e}\\
a_{1,1} &= f_{xy}(0,0) \tag{C.23f}\\
a_{1,2} &= 3\,[f_x(0,1)-f_x(0,0)]-2f_{xy}(0,0)-f_{xy}(0,1) \tag{C.23g}\\
a_{1,3} &= 2\,[f_x(0,0)-f_x(0,1)]+f_{xy}(0,0)+f_{xy}(0,1) \tag{C.23h}\\
a_{2,0} &= 3\,[f(1,0)-f(0,0)]-2f_x(0,0)-f_x(1,0) \tag{C.23i}\\
a_{2,1} &= 3\,[f_y(1,0)-f_y(0,0)]-2f_{xy}(0,0)-f_{xy}(1,0) \tag{C.23j}\\
a_{2,2} &= 9\,[f(0,0)-f(0,1)-f(1,0)+f(1,1)]+ \tag{C.23k}\\
&\quad +6\,[f_x(0,0)-f_x(0,1)+f_y(0,0)-f_y(1,0)]+ \notag\\
&\quad +3\,[f_x(1,0)-f_x(1,1)+f_y(0,1)-f_y(1,1)]+ \notag\\
&\quad +4f_{xy}(0,0)+2\,[f_{xy}(0,1)+f_{xy}(1,0)]+f_{xy}(1,1) \notag\\
a_{2,3} &= 6\,[f(0,1)-f(0,0)+f(1,0)-f(1,1)]+ \tag{C.23l}\\
&\quad -3\,[f_y(0,0)+f_y(0,1)-f_y(1,0)-f_y(1,1)]+ \notag\\
&\quad -4\,[f_x(0,0)-f_x(0,1)]+ \notag\\
&\quad -2\,[f_x(1,0)-f_x(1,1)+f_{xy}(0,0)+f_{xy}(0,1)]+ \notag\\
&\quad -f_{xy}(1,0)-f_{xy}(1,1) \tag{C.23m}\\
a_{3,0} &= 2\,[f(0,0)-f(1,0)]+f_x(0,0)+f_x(1,0) \tag{C.23n}\\
a_{3,1} &= f_{xy}(0,0)+f_{xy}(1,0)+2\,[f_y(0,0)-f_y(1,0)] \tag{C.23o}\\
a_{3,2} &= 6\,[f(0,1)-f(0,0)+f(1,0)-f(1,1)]+4\,[f_y(1,0)-f_y(0,0)]+ \tag{C.23p}\\
&\quad +3\,[f_x(0,1)-f_x(0,0)-f_x(1,0)+f_x(1,1)]+ \notag\\
&\quad +2\,[f_y(1,1)-f_{xy}(0,0)-f_{xy}(1,0)-f_y(0,1)]+ \notag\\
&\quad -f_{xy}(0,1)-f_{xy}(1,1) \tag{C.23q}\\
a_{3,3} &= 4\,[f(0,0)-f(0,1)-f(1,0)+f(1,1)]+ \tag{C.23r}\\
&\quad +2\,[f_x(0,0)-f_x(0,1)+f_x(1,0)-f_x(1,1)+f_y(0,0)+f_y(0,1)-f_y(1,0)-f_y(1,1)]+ \notag\\
&\quad +f_{xy}(0,0)+f_{xy}(0,1)+f_{xy}(1,0)+f_{xy}(1,1) \notag
\end{align}
$$

Writing this solution as a vector of 16 entries, is possible to write the previous system of equations as the following

$$\underline{\underline{C}}\cdot\underline{a}=\underline{f} \tag{C.24}$$

where $\underline{f}$ is the vector of function values and derivatives and $\underline{a}$ is the vector of 16 coefficients of the interpolant function. The matrix $\underline{\underline{C}}$ is the matrix of coefficients of the equations. Solving the system is very easy as is possible to find the inverse of $\underline{\underline{C}}$ as a matrix of integer values. See [116] for more details.



(a) Bilinear interpolation

(b) Bicubic spline interpolation

Figure C.4: Comparison between bilinear and bicubic spline interpolation

C.3 Derivative approximations

In the previous sections, different interpolation scheme have been proposed and shown. They are quite different but most of them have an important common part: to calculate derivatives of functions defined by data points. Their equations will be here presented, in order to make it simpler (see [116]). In order to present a simple approach (anyway without losing generality) herein it will be considered a uniform grid where the grid spacing along x axis and along y axis are the same and equal to g.

Under these hypotheses we'll get

$$\left.\frac{\partial f}{\partial x}\right|_{i,j} = \frac{1}{2\cdot h}\left(f_{i+1,j} - f_{i-1,j}\right) + O(g^2) \tag{C.25}$$

$$\left.\frac{\partial f}{\partial x}\right|_{i+0.5,j} = \frac{1}{h}\left(f_{i+1,j} - f_{i,j}\right) + O(g^2) \tag{C.26}$$

$$\left.\frac{\partial^2 f}{\partial x^2}\right|_{i,j} = \frac{1}{h^2}\left(f_{i+1,j} + f_{i-1,j} - 2\cdot f_{i,j}\right) + O(g^2) \tag{C.27}$$

$$\left.\frac{\partial^2 f}{\partial x \partial y}\right|_{i,j} = \frac{1}{4\cdot h^2}\left(f_{i+1,j+1} + f_{i-1,j-1} - f_{i-1,j+1} - f_{i+1,j-1}\right) + O(g^2) \tag{C.28}$$

and also

$$\left.\frac{\partial f}{\partial x}\right|_{i,j} = \frac{1}{12\cdot h}\left(f_{i-2,j} - 8\cdot f_{i-1,j} + 8\cdot f_{i+1,j} - f_{i+2,j}\right) + O(g^4) \tag{C.29}$$

$$\left.\frac{\partial^2 f}{\partial x^2}\right|_{i,j} = \frac{1}{12\cdot h^2}\left(-f_{i-2,j} + 16\cdot f_{i-1,j} - 30\cdot f_{i,j} + 16\cdot f_{i+1,j} - f_{i+2,j}\right) + O(g^4) \tag{C.30}$$

C.4 Runge's phenomenon

This phenomenon is an important drawback of polynomial interpolation, especially with polynomials of high degree. The name comes from *Carl David Tolmé Runge.* He stated that considering an interpolating function of some given data points, it is possible to demonstrate that if the polynomial is oh high degree, the error defined as difference between the interpolating function and the real function, can go to infinity

$$\lim_{n\to\infty}\left[\max_{-1\le x\le 1}|f(x) - P_n(x)|\right] = \infty \tag{C.31}$$

where $P_n(x)$ is the interpolant polynomial of degree n and f is the Runge's function

$$f(x) = \frac{1}{1+25\cdot x^2} \tag{C.32}$$

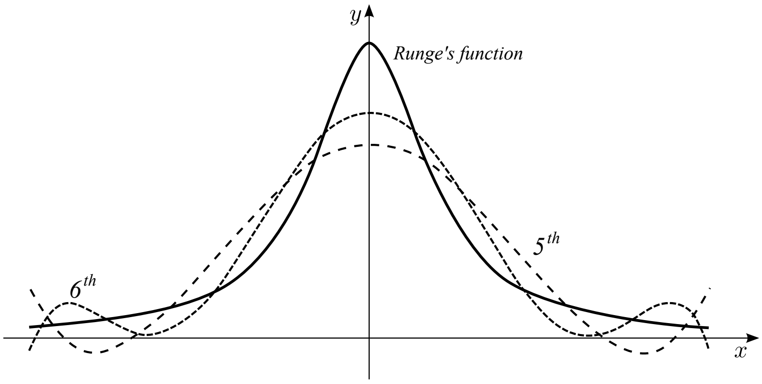


Figure C.5: Runge's function interpolated by a 5^{th} and a 6^{th} degrees functions

Appendix

D

Regression analysis

D.1 Introduction

REGRESSSION ANALYSIS is a statistical technique focused on the modeling of the relationship between a dependent variable and one or more independent variables. The results is commonly a function of the independent variables, called *regression function*. In this chapter it will be discussed a series of parametric[1] regression function and how to evaluate them.

The performance of the regression analysis depends on the assumptions about the form of the data. These assumptions are sometimes testable if there is a big amount of data. This fact must be clear in mind in order to interpret correctly the results of the regression process.

D.2 Least square approach

CONSIDERING a series of data as results from an experiment, the regression function is that function that is representing as best as possible the measured data. The shape of the function can depend on a series of parameters $\underline{\beta} \in \mathbb{R}^n$ that are relating the independent variables to the dependent variable. The relation between the independent variables $\underline{x} \in \mathbb{R}^n$ and the dependent one y is as follows:

$$y = f(\underline{x}, \underline{\beta}) = \underline{x}^T \cdot \underline{\beta} = \langle \underline{x}, \underline{\beta} \rangle \tag{D.1}$$

Consider a set of m experiments and so a set of m values of $\underline{x}$. If the dimension of $\underline{\beta}$ is n it may be possible one of these three options:

- $m < n$: in this case the regression analysis cannot be performed as there are less equations than unknown parameters
- $m = n$: the regression analysis is just a solution of a system of m equations and so it is not really a statistical approach

---

[1] It is called parametric that regression function that is defined by a finite number of unknown parameters

- $m > n$: the problem is so called overdetermined (as it brings to an overdetermined system of equations) and so the regression analysis can be performed

In the last case, even if the system has a higher number of constraints (the number of equations) respect to the number of unknowns, a solution can be found following the *least squares approach*, minimizing the distance between the measured values and the predicted ones (called *square residuals*). This is performed usually by minimizing the sum of square residuals over the data. Considering $y_i, i \in [1, m]$ the generic measured data and f_i the value of the regression function calculated at the i^{th} measured data, the sum of squared residuals will be:

$$\chi^2 = \sum_{i=1}^{m} (y_i - f_i)^2 \tag{D.2}$$

This equation is used in linear regression models[2] to find the values of the unknown parameters $\underline{\beta}$. The minimum is determined by setting to zero the partial derivatives of χ^2 respect to each unknown $\beta_j, j \in [1, n]$. Thus, considering the vector of unknowns $\underline{\beta}$, the vector of measured data $\underline{y}$ and the matrix of independent variables $\underline{\underline{X}} \in \mathbb{R}^{m \times n}$, the problem to be solved is the following:

$$\underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{m} \left(y_i - \sum_{j=1}^{n} X_{ij} \cdot \beta_j \right)^2 = \underset{\beta}{\operatorname{argmin}} \left\| \underline{Y} - \underline{\underline{X}} \cdot \underline{\beta} \right\|^2 \tag{D.3}$$

where $\underline{Y} \in \mathbb{R}^m$ is the vector of measure data.

Considering $\underline{\tilde{\beta}}$ the solution of the previous problem, if the experimental errors are uncorrelated and have zero mean and constant variance, thus the Gauss-Markov theorem states that the least-square estimator $\underline{\tilde{\beta}}$ is the best linear estimator of the parameters.

The general problem can be solved by the following equation:

$$\underline{\underline{X}} \cdot \underline{\tilde{\beta}} = \underline{Y} \tag{D.4}$$

$$\underline{\tilde{\beta}} = \left(\underline{\underline{X}}^T \underline{\underline{X}} \right)^{-1} \underline{\underline{X}}^T \underline{Y} \tag{D.5}$$

The solution of this system of equations can be different according to the kind of equations to be managed. In some cases it is enough to perform the following substitution

$$\underline{\underline{X}}^{MP} = \left(\underline{\underline{X}}^T \underline{\underline{X}} \right)^{-1} \underline{\underline{X}}^T \tag{D.6}$$

where $\underline{\underline{X}}^{MP}$ is the *Moore-Penrose pseudoinverse* of $\underline{\underline{X}}$, that is a generalization of the concept of inverse, applied to non-square matrixes.

If the matrix $\underline{\underline{X}}^T \underline{\underline{X}}$ is well-conditioned and positive defined, the system can be solved by *Cholesky decomposition.* In most of applications the matrix $\underline{\underline{X}}$ is not so well defined and so a more

---

[2] The adjective linear is considering the linear dependence of the regression function to the parameters, not to the data

stable approach must be performed. An example is the *QR decomposition* (see Appendix E.3).

D.3 Linear regression

LINEAR REGRESSION is a special case of regression analysis where the equation to be found is the equation of a straight line in a two-dimensional space. Thus the number of unknown parameters are two: the *slope* and the *y-intercept* and the number of independent variables is one. The equation to find is usually expressed as the common straight line equation, in the explicit form, thus where the dependent variable y is explicited respect to the independent variable x:

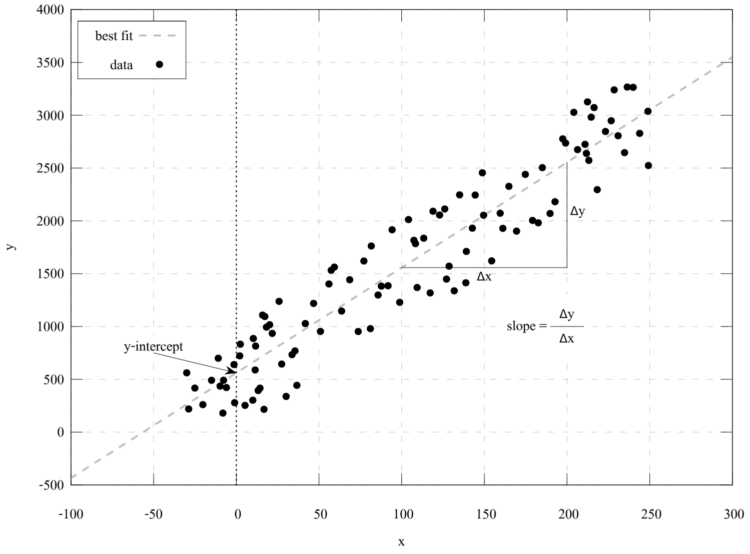


Figure D.1: Example of linear regression

$$y = \beta_1 \cdot x + \beta_2 \tag{D.7}$$

where β_1 is the slope and β_2 is the y-intercept.

A solution to this problem can be found by using the approach aforementioned and so by using matrix algebra and QR factorization but also by applying a simpler method. In fact by partial derivation of the Equation (D.2) it is possible to obtain:

$$\begin{cases} \dfrac{\partial \chi^2}{\partial \beta_1} = 2 \cdot \displaystyle\sum_{i=1}^{m} (y_i - \beta_1 x_i - \beta_2)(-x_i) & = 0 \\ \dfrac{\partial \chi^2}{\partial \beta_2} = 2 \cdot \displaystyle\sum_{i=1}^{m} (y_i - \beta_1 x_i - \beta_2)(-1) & = 0 \end{cases} \tag{D.8}$$

and then, solving the system of two equations in two unknowns:

$$\begin{cases} \beta_1 = \dfrac{\displaystyle\sum_{i=1}^{m} x_i y_i - \overline{y} \sum_{i=1}^{m} x_i}{\displaystyle\sum_{i=1}^{m} x_i^2 - \overline{x} \sum_{i=1}^{m} x_i} \\ \beta_2 = \overline{y} - \beta_1 \overline{x} \end{cases} \tag{D.9}$$

where $\overline{x}$ and $\overline{y}$ represent the mean value of x and of y respectively.

D.4 Bilinear regression

Bilinear regression stands for the regression analysis approach applied to a bilinear function, a function where a dependent variable z depends over two different independent parameters (e.g. x and y) through a non-linear equation. The function it is linear along the x axis and along the y axis but generally it is not linear otherwise. The equation is as follows:

$$z(x, y) = \beta_1 x + \beta_2 y + \beta_3 xy + \beta_4 \tag{D.10}$$

In order to define the four parameters $\beta_j, j \in [1, 4]$ there are two major ways. One is by the use of Gauss-Jordan elimination (see Appendix E.2.1) and the other is by the use of QR factorization (see Appendix E.3). Considering a solution based on Gauss-Jordan elimination it is needed again to calculate the partial derivatives of χ^2 according to the four parameters, considering all m data points:

$$\begin{cases} \dfrac{\partial \chi^2}{\partial \beta_1} = 2 \cdot \displaystyle\sum_{i=1}^{m} (z_i - \beta_1 x_i - \beta_2 y_i - \beta_3 x_i y_i - \beta_4)(-x_i) & = 0 \\ \dfrac{\partial \chi^2}{\partial \beta_2} = 2 \cdot \displaystyle\sum_{i=1}^{m} (z_i - \beta_1 x_i - \beta_2 y_i - \beta_3 x_i y_i - \beta_4)(-y_i) & = 0 \\ \dfrac{\partial \chi^2}{\partial \beta_3} = 2 \cdot \displaystyle\sum_{i=1}^{m} (z_i - \beta_1 x_i - \beta_2 y_i - \beta_3 x_i y_i - \beta_4)(-x_i y_i) & = 0 \\ \dfrac{\partial \chi^2}{\partial \beta_4} = 2 \cdot \displaystyle\sum_{i=1}^{m} (z_i - \beta_1 x_i - \beta_2 y_i - \beta_3 x_i y_i - \beta_4)(-1) & = 0 \end{cases} \tag{D.11}$$

In this situation the algebraical solution of the system is quite complicated and so practically inconvenient to get. Thus it is better to define the system of Equation (D.11) by a matrix form, namely:

$$\underline{\underline{A}} \cdot \underline{\beta} = \underline{a} \tag{D.12}$$

where $\underline{\beta} \in \mathbb{R}^{4\times1}$ is the vector of the four unknowns parameters while $\underline{\underline{A}} \in \mathbb{R}^{4\times4}$ and $\underline{a} \in \mathbb{R}^{4\times1}$

are defined as follows:

$$\begin{bmatrix} \sum_{i=1}^{m} x_i^2 & \sum_{i=1}^{m} x_i y_i & \sum_{i=1}^{m} x_i^2 y_i & \sum_{i=1}^{m} x_i \\ & \sum_{i=1}^{m} y_i^2 & \sum_{i=1}^{m} x_i y_i^2 & \sum_{i=1}^{m} y_i \\ & & \sum_{i=1}^{m} x_i^2 y_i^2 & \sum_{i=1}^{m} x_i y_i \\ sym. & & & m \end{bmatrix} \cdot \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{m} x_i z_i \\ \sum_{i=1}^{m} y_i z_i \\ \sum_{i=1}^{m} x_i y_i z_i \\ \sum_{i=1}^{m} z_i \end{bmatrix} \tag{D.13}$$

Now an approach as described in Appendix E.2.1 will bring to the calculation of the unknowns.

On the other hands, for a solution based on the QR factorization it is needed a system of equations defining each constrain defined by each point to be considered:

$$\begin{bmatrix} x_1 & y_1 & x_1 y_1 & 1 \\ x_2 & y_2 & x_2 y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_i & y_i & x_i y_i & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_m & y_m & x_m y_m & 1 \end{bmatrix} \cdot \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_i \\ \vdots \\ z_m \end{bmatrix} \tag{D.14}$$

Equation (D.14). where the first matrix (i.e. $\underline{\underline{B}}$) is $m \times 4$ and the vector of constant terms (i.e. $\underline{Z}$) has m entries, where usually $m > 4$. Thus in this situation the matrix of coefficients is not a square matrix, and in particular has more rows than columns. In this situation the equivalent system is overdetermined as has more constrain than degree of freedom (unknowns). The solution in general doesn't exist, but can be found a "best solution" in a statistical meaning, namely the best set of values for β_j to have the bilinear function closer to the cloud of points $(x_i, y_i, z_i), i \in [1, m]$.

To perform this task is necessary to invert the matrix $\underline{\underline{B}}$ but as aforementioned this cannot be done if not by a generalized concept of inverse matrix and especially thanks to some factorization approach, that are more numerically stable than a normal pseudoinverse matrix (see Appendix D.2). After factorization of $\underline{\underline{B}}$ it will be possible to find easily the solution of Equation (D.14).

An example of bilinear regression can be seen in Figure D.2 where 49 data points are fitted by a bilinear function. To perform that regression a simple Octave function has been written in order to find the best fit of the points. The function creates the matrix $\underline{\underline{B}}$ and the vector $\underline{Z}$, filling by random generated numbers. At the end the pseudoinverse[3] of the matrix $\underline{\underline{B}}$ was used

---

[3] This approach has been used for simplicity and because is easy to find a stable matrix, but it cannot be normally used

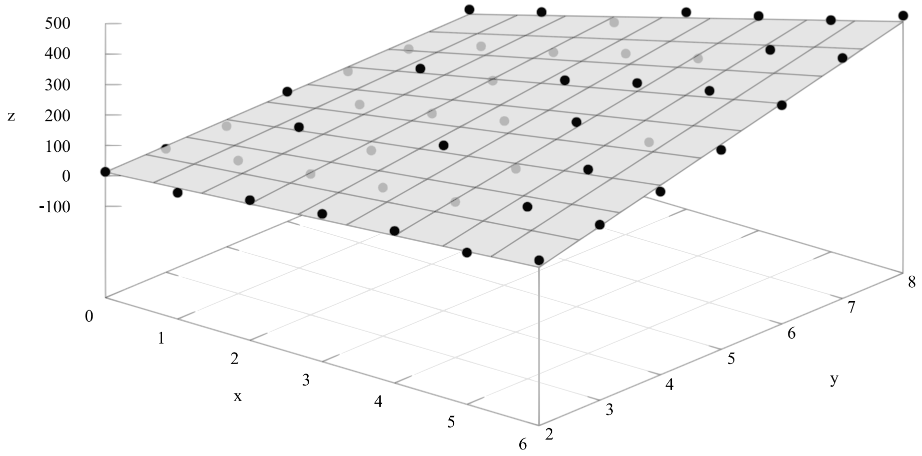


Figure D.2: Example of bilinear regression of data points

to calculate the solution of the regression:

```
function b = bilinearRegression(numberPoints)
        B = ones(numberPoints, 4);
        B(:, 1) = rand();                    % x coordinates generation
        B(:, 2) = rand();                    % y coordinates generation
        B(:, 3) = B(:, 1) .* B(:, 2);        % x * y
        z = rand(numberPoints, 1);           % z coordinates generation
        b = pinv(B) * z;                     % B * b = z
```

Appendix

E

Matrix Algebra

E.1 Introduction

In mathematics a *matrix* it is a rectangular array of numbers [117] organized in rows and columns thus each entry of the array is denoted by a double index subscript: $a_{i,j}$ is the entry of the matrix $\underline{\underline{A}}$ at the i^{th} row and the j^{th} column. A matrix where one of the dimensions (rows or columns) is equal to unity is normally called *vector*. Generally a vector in a $m \times 1$ space is called column vector and a vector in a $1 \times n$ space is called row vector. Generally speaking, a *scalar* can be thought as a 1×1 matrix. A matrix where the number of rows equals the number of columns is called *square matrix*. In this book the matrices will be represented by upper-case (or sometimes lower-case) letter and a double underline, while vectors will be represented by lower-case letters and single underline. The single entry of matrix or vector will be always represented by lower-case letter and no underline.

Matrices can be subdue to different operations (called *matrix operations*) like *matrix addition*, *scalar multiplication* and *transposition*.

Definition 4. *The operation of addition over two matrices* $\underline{\underline{A}}, \underline{\underline{B}} \in \mathbb{R}^{m \times n}$ *gives a matrix* $\underline{\underline{C}} \in \mathbb{R}^{m \times n}$ *where:*

$$c_{i,j} = a_{i,j} + b_{i,j}, \quad i \in [1, m], j \in [1, n]$$

Definition 5. *The operation of scalar multiplication of a matrix* $\underline{\underline{A}} \in \mathbb{R}^{m \times n}$ *and a scalar value* k *gives a matrix* $\underline{\underline{C}} \in \mathbb{R}^{m \times n}$ *where:*

$$c_{i,j} = k \cdot a_{i,j}, \quad i \in [1, m], j \in [1, n]$$

Definition 6. *The operation of transposition of a matrix* $\underline{\underline{A}} \in \mathbb{R}^{m \times n}$ *(denoted often by* $\underline{\underline{A}}^T$*) gives a matrix* $\underline{\underline{B}} \in \mathbb{R}^{n \times m}$ *where:*

$$b_{j,i} = a_{i,j}, \quad i \in [1, m], j \in [1, n]$$

It is also possible to define other operations with matrices, like the *matrix multiplication*.

Definition 7. *The operation of multiplication of a matrix $\underline{\underline{A}} \in \mathbb{R}^{m\times n}$ with a matrix $\underline{\underline{B}} \in \mathbb{R}^{n\times s}$ gives a matrix $\underline{\underline{C}} \in \mathbb{R}^{m\times s}$ where:*

$$c_{i,j} = \sum_{k=1}^{n} a_{i,k} \cdot b_{k,j}, \quad i \in [1,m], j \in [1,s]$$

Matrix multiplication is has associativity and distributivity but is not commutative (i.e. $\underline{\underline{A}} \cdot \underline{\underline{B}} \neq \underline{\underline{B}} \cdot \underline{\underline{A}}$) except when at least one of the two matrices is the *identity matrix*.

A matrix is called *symmetric* if it is true that $\underline{\underline{A}}^T = \underline{\underline{A}}$ and so it follows that only square matrices can be symmetric. A symmetric matrix $\underline{\underline{A}} \in \mathbb{R}^{m\times m}$ is called *positive defined* if for all non-zero vectors $\underline{v} \in \mathbb{R}^m$ the following expression is true:

$$p = \underline{v}^T \cdot \underline{\underline{A}} \cdot \underline{v} > 0 \tag{E.1}$$

For square matrices $\underline{\underline{A}} \in \mathbb{R}^{m\times m}$ may exist a square matrix $\underline{\underline{B}} \in \mathbb{R}^{m\times m}$ that multiplied to the previous one gives the identity matrix, namely:

$$\underline{\underline{A}} \cdot \underline{\underline{B}} = \underline{\underline{I}} \tag{E.2}$$

This special matrix $\underline{\underline{B}}$ is commonly simply indicated by $\underline{\underline{A}}^{-1}$. The existence of this matrix is not assured and it depends on special values of its entries. In particular a matrix has an inverse if and only if it is *not singular* and so if its *determinant* is not zero. To calculate it, it is possible to use the *Leibniz formula*, defined by the following:

$$det\left(\underline{\underline{A}}\right) = \sum_{\sigma\in S_m} \left[sgn\left(\sigma\right) \cdot \prod_{i=1}^{m} a_{i,\sigma(i)} \right] \tag{E.3}$$

where σ is one of the $m!$ permutations of the numbers in the set $[1,\ldots,m]$, that are all grouped in the set of permutations S_m. The sign of σ depends on the number of switches between numbers must be applied to the permutation to have the original ordered list of numbers. If this number of switches is odd the sign will be negative, if the number of switches is even the sign will be positive.

It is also possible to define the determinant by expansion of a row (or column) by the *Laplace formula*. Selecting the i^{th} row the formula will be:

$$det\left(\underline{\underline{A}}\right) = \sum_{j=1}^{m} \left(-1\right)^{i+j} \cdot a_{i,j} \cdot d_{i,j} \tag{E.4}$$

where $d_{i,j}$ is the determinant of the matrix obtained from $\underline{\underline{A}}$ by removing the i^{th} row and the j^{th} column. This formula is very easy and so often used in simple applications or where the matrices are relatively small. It is anyway proved that the time needed for its calculation gets very high with the dimension of the matrix. For matrices bigger than 3×4 it is faster to express (if possible) the matrix by a multiplication of lower triangular matrix ($\underline{\underline{L}}$) and an upper triangular matrix ($\underline{\underline{U}}$) by *LU factorization* (see Appendix E.4). In fact in that case the determinant will be:

$$det\left(\underline{\underline{A}}\right) = det\left(\underline{\underline{L}}\right) \cdot det\left(\underline{\underline{U}}\right) = det\left(\underline{\underline{U}}\right) \tag{E.5}$$

as $det\left(\underline{\underline{L}}\right) = 1$. Computing the determinant of $\underline{\underline{U}}$ is fast as the determinant of a triangular matrix is equalt to the product of the entries on the main diagonal.

For small matrices here is presented the direct expressions of the determinant:

$$det\left(\begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix}\right) = a_{1,1} \cdot a_{2,2} - a_{1,2} \cdot a_{2,1} \tag{E.6}$$

$$det\left(\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}\right) = a_{1,1} \cdot a_{2,2} \cdot a_{3,3} + a_{1,2} \cdot a_{2,3} \cdot a_{3,1} + a_{1,3} \cdot a_{3,2} \cdot a_{2,1} + \tag{E.7}$$

$$- a_{1,1} \cdot a_{2,3} \cdot a_{3,2} - a_{1,2} \cdot a_{2,1} \cdot a_{3,3} - a_{1,3} \cdot a_{2,2} \cdot a_{3,1} \tag{E.8}$$

It is obvious from Equation (E.3) that the number of monomials must be added (and subtracted) to get the determinant is exactly $m!$ and so it is growing very fast with the dimension of the matrix.

For example few tests performed on a computer *AMD Athlon^{TM}64 3000+* with 1.81 GHz processor and 2 GByte of *RAM* gave these results shown in Table E.1.

Table E.1: Speed test on two different algorithm to calculate the determinant of a square matrix

matrix dimension	time [s]	
	Laplace	*LU factorization*
1	0.0000022	0.0000042
2	0.0000017	0.0000064
3	0.000010	0.000011
4	0.000023	0.000017
5	0.00011	0.000025
6	0.00069	0.000035
7	0.0052	0.000048
8	0.038	0.000064
9	0.35	0.00011
10	3.5	0.00013
11	38.7	0.00018
12	465.0	0.00022

The results are only shown to present the big difference between the two approaches. The Laplace algorithm is obviously getting slower much faster than the determinant calculated by LU factorization. In fact when the dimension of the matrix is 7 the ration between the time needed by the Laplace algorithm is roughly 108 times higher. With only a dimension of 9 the ration is already 3181. In both cases the small C++ functions developed are not well optimized and so better results can be achieved, but it is anyway important to notice the big difference between them.

Another important parameter used in matrix calculus and linear algebra is the *rank* of a matrix. The rank is the maximum number of linearly independent rows (or columns) of a matrix [117]. One of the possible ways to calculate it is to row reduce the matrix thanks to Gauss elimination (see Appendix E.2) or by LU decomposition (see Appendix E.4) and count the number of non-zero rows. A better choice is a QR decomposition (see Appendix E.3) with pivoting, that is more numerically robust and less computationally expensive.

Performing the Gauss elimination gives a matrix in a *row echelon form* [117] namely a matrix with the following properties:

- all non-zero rows are above any rows of all zeros
- the *pivot* (the first non-zero entry in each row) of a non-zero row is always strictly to the right of the pivot of the row above it
- the pivot of each non-zero row is 1 (this is not always required)

Adding that "*every pivot is 1 and is the only non-zero entry in its column*" brings to the definition of *reduced row echelon form* [117].

E.2 Gauss elimination

GAUSS ELIMINATION is a common algorithm in linear algebra for solving system of linear equations. In this approach, *elementary row operations*[1] are performed on the matrix representation of a system in order to reduce it to a row echelon form. The algorithm is divided in two parts: forward elimination and back substitution. The former part reduces the system to a triangular or echelon form (or degenerate indicating that the system doesn't allow any solution). The latter part gives directly the solution (if any) of the system.

---

[1] The elementary row operations are special operations that do not change the solution set of a system of linear equations. There are three types of operations: row switching, row multiplication by scalar and row addition by another row. For each elementary operation exists an *elementary matrix* that can perform that operation just by multiplying it with the matrix representing the system.

Let's consider a system of equations as follows:

$$\begin{cases} \begin{array}{cccccc} a_{1,1} \cdot x_1 + & a_{1,2} \cdot x_2 + & \ldots + & a_{1,j} \cdot x_j + & \ldots + & a_{1,n} \cdot x_n = b_1 \\ a_{2,1} \cdot x_1 + & a_{2,2} \cdot x_2 + & \ldots + & a_{2,j} \cdot x_j + & \ldots + & a_{2,n} \cdot x_n = b_2 \\ & & & \vdots & & \\ a_{i,1} \cdot x_1 + & a_{i,2} \cdot x_2 + & \ldots + & a_{i,j} \cdot x_j + & \ldots + & a_{i,n} \cdot x_n = b_i \\ & & & \vdots & & \\ a_{m,1} \cdot x_1 + & a_{m,2} \cdot x_2 + & \ldots + & a_{m,j} \cdot x_j + & \ldots + & a_{m,n} \cdot x_n = b_m \end{array} \end{cases} \tag{E.9}$$

Describing the set of coefficients $a_{i,j}$ by a matrix $\underline{\underline{A}} \in \mathbb{R}^{m \times n}$, the set of unknowns by a column vector $\underline{x} \in \mathbb{R}^n$ and the set of constant terms by a column vector $\underline{b} \in \mathbb{R}^m$ it is easily possible to represent the same system in its matrix form:

$$\underline{\underline{A}} \cdot \underline{x} = \underline{b} \tag{E.10}$$

Gauss elimination is commonly used also to find the determinant of a squared matrix or its inverse, because of its good numerical stability, especially if some sort of *partial pivot* (i.e. the entry with highest absolute value of the analyzed column is selected to be the pivot. This is generally sufficient to reduce round-off errors) approach is applied, and speed. But in any case cannot be applied to higher order tensors (a matrix is a tensor of order 2).

Hereafter is presented the Gauss elimination algorithm with partial pivoting in pseudocode:

Algorithm E.1 Gauss elimination with partial pivoting

```
Require: matrix $\underline{\underline{A}} \in \mathbb{R}^{m \times m}$ and vector $\underline{b} \in \mathbb{R}^m$
Ensure: upper triangular matrix $\underline{\underline{A}}$ and vector $\underline{b}$
$\underline{\underline{C}} \leftarrow \left[\underline{\underline{A}} | \underline{b}\right]$
$rows \leftarrow$ rows of $\underline{\underline{A}}$
$columns \leftarrow$ columns of $\underline{\underline{A}} + 1$
$row, column \leftarrow 1$
while $(row \leq rows)$ and $(column \leq columns)$ do
  $maxrow = row$
  for $i = row + 1$ to $rows$ do
    if $|c_{i,column}| > |c_{maxrow,column}|$ then
      $maxrow \leftarrow i$
    end if
  end for
  if $c_{maxrow,column} \neq 0$ then
    swap $row^{th}$ row with $maxrow^{th}$ row
    for $i = column$ to $columns$ do
```

```
15:             $c_{row,i} \leftarrow c_{row,i} / c_{row,column}$
16:         end for
17:         for $i = row + 1$ to $rows$ do
18:             for $j = column$ to $column$ do
19:                 $c_{i,j} \leftarrow c_{i,j} - c_{row,j} \cdot c_{i,column}$
20:             end for
21:         end for
22:         $row \leftarrow row + 1$
23:     end if
24:     $column \leftarrow column + 1$
25: end while
26: $a_{i,j} \leftarrow c_{i,j}$ for $i \in [1, rows], j \in [1, columns - 1]$
27: $b_i \leftarrow c_{i,columns}$ for $i \in [1, rows]$
```

Once this algorithm is applied to the system, the matrix $\underline{\underline{C}}$ will be an upper triangular matrix (and so also the matrix $\underline{\underline{A}}$ thanks to what it is written in line 26). According to the rank of the matrix and its comparison with the number of unknowns n is possible to understand the number of solutions of the system:

- $rank(A) < n$: the system is underdetermined and so has infinite solutions
- $rank(A) = n$: the system has one unique solution
- $rank(A) > n$: the system is overdetermined (more equations than unknowns) and so has no solution. This is possible only if the matrix $\underline{\underline{A}}$ is not squared and has more rows than columns

Finally a back substitution will calculate the solution of the system:

$$x_i = \frac{b_i - \sum_{j=1}^{i-1} a_{i,j} \cdot x_j}{a_{i,i}} \qquad i = 1, \ldots, m \tag{E.11}$$

E.2.1 Gauss-Jordan elimination

Gauss-Jordan elimination is an algorithm of linear algebra that brings a matrix to its reduced echelon form. Every matrix has a reduced echelon form and so this algorithm is guaranteed to build it [117]. Gauss-Jordan elimination is definitely less efficient than Gauss elimination with back substitution but still used, especially for calculating the inverse of a matrix.

E.3 QR factorization

QR factorization is an algorithm aimed to the decomposition of a matrix in a product of an *orthogonal matrix* and a *right triangular matrix* [118]. Considering a square matrix $\underline{\underline{A}}$, it will be decomposed in $\underline{\underline{A}} = \underline{\underline{Q}} \cdot \underline{\underline{R}}$ where $\underline{\underline{Q}}$ is orthogonal (i.e. its inverse is equal to its transpose, and so $\underline{\underline{Q}}^T \underline{\underline{Q}} = \underline{\underline{I}}$) and $\underline{\underline{R}}$ is a right triangular matrix (also called upper triangular matrix). More generally it is possible to decompose a rectangular matrix $\underline{\underline{A}} \in \mathbb{R}^{m \times n}$ in a square matrix $\underline{\underline{Q}} \in \mathbb{R}^{m \times m}$ and a right triangular matrix $\underline{\underline{R}} \in \mathbb{R}^{m \times n}$ where all the rows $\underline{r}_{i>n}$ are zero. Thanks to this the QR factorization can be expressed also as follows:

$$\underline{\underline{A}} = \underline{\underline{Q}} \cdot \underline{\underline{R}} = \left[\underline{\underline{Q}}_1 | \underline{\underline{Q}}_2\right] \cdot \begin{bmatrix} \underline{\underline{R}}_1 \\ 0 \end{bmatrix} = \underline{\underline{Q}}_1 \cdot \underline{\underline{R}}_1 \tag{E.12}$$

where $\underline{\underline{Q}}_1 \in \mathbb{R}^{m \times n}$ is still an orthogonal matrix and $\underline{\underline{R}}_1 \in \mathbb{R}^{n \times n}$ is still a right triangular matrix. Here the algorithm to calculate $\underline{\underline{Q}}_1$ and $\underline{\underline{R}}_1$ that will be simply considered as the final $\underline{\underline{Q}}$ and $\underline{\underline{R}}$:

Algorithm E.2 Gram-Smidth algorithm for QR factorization

```
Require: matrix $\underline{\underline{A}} \in \mathbb{R}^{m \times n}$
Ensure: orthogonal matrix $\underline{\underline{Q}}$ and right triangular matrix $\underline{\underline{R}}$, with $\underline{\underline{Q}} \cdot \underline{\underline{R}} = \underline{\underline{A}}$
$\underline{u}_1 \leftarrow \underline{a}_1$        ▷ $\underline{a}_i$ is the $i^{th}$ column of $\underline{\underline{A}}$
$\underline{e}_1 \leftarrow \underline{u}_1 / \|\underline{u}_1\|$
for $i = 2$ to $n$ do
    $\underline{u}_i \leftarrow \underline{a}_i - \sum_{j=1}^{i-1} \frac{\langle \underline{e}_j, \underline{a}_i \rangle}{\langle \underline{e}_j, \underline{e}_j \rangle} \cdot \underline{e}_j$
    $\underline{e}_i \leftarrow \underline{u}_i / \|\underline{u}_i\|$
end for
for $i = 1$ to $n$ do
    $\underline{a}_i \leftarrow \sum_{j=1}^{i} \langle \underline{e}_j, \underline{a}_i \rangle \cdot \underline{e}_j$
end for
$\underline{q}_i \leftarrow \underline{e}_i$
$\underline{\underline{R}} \leftarrow 0$
for $i = 1$ to $n$ do
    for $j = i$ to $n$ do
        $r_{i,j} \leftarrow \langle \underline{e}_i, \underline{a}_j \rangle$
    end for
end for
```

E.4 LU factorization

LU FACTORIZATION is an algorithm aimed to the decomposition of a matrix in a product of a lower triangular matrix and an upper triangular matrix [119]. Considering a square matrix $\underline{\underline{A}} \in \mathbb{R}^{m \times m}$, it will be decomposed in $\underline{\underline{A}} = \underline{\underline{L}} \cdot \underline{\underline{U}}$. If the matrix $\underline{\underline{A}}$ is *Hermitian*[2] and positive definite than it is possible to find a matrix $\underline{\underline{U}}$ equals to the conjugate transpose of $\underline{\underline{L}}$. This decomposition is called *Cholesky decomposition* (see Appendix E.4.1). Computing the LU decomposition is possible by the following algorithm:

Algorithm E.3 Doolittle algorithm for LU factorization

```
Require: square matrix $\underline{\underline{A}} \in \mathbb{R}^{m \times m}$
Ensure: lower triangular matrix $\underline{\underline{L}}$ and upper triangular matrix $\underline{\underline{U}}$, with $\underline{\underline{L}} \cdot \underline{\underline{U}} = \underline{\underline{A}}$
$\underline{\underline{B}}^{(0)} \leftarrow \underline{\underline{A}}$
for $i = 1$ to $m - 1$ do
    $\underline{\underline{L}}^{(i)} \leftarrow \underline{\underline{I}}$
    for $j = i + 1$ to $m$ do
        $l_{j,i}^{(i)} \leftarrow -b_{j,i}^{(i-1)} / b_{i,i}^{(i-1)}$
    end for
    $\underline{\underline{B}}^{(i)} \leftarrow \underline{\underline{L}}^{(i)} \cdot \underline{\underline{B}}^{(i-1)}$
end for
$\underline{\underline{U}} \leftarrow \underline{\underline{B}}^{(m-1)}$
$\underline{\underline{L}} \leftarrow \underline{\underline{L}}^{(1)^{-1}} \cdot \underline{\underline{L}}^{(2)^{-1}} \cdot \ldots \cdot \underline{\underline{L}}^{(m-1)^{-1}}$
```

E.4.1 Cholesky decomposition

Cholesky decomposition is an algorithm aimed to the decomposition of a square symmetric positive definite matrix in a product of a lower triangular matrix and its conjugate transpose [118]. Considering a matrix $\underline{\underline{A}} \in \mathbb{R}^{m \times m}$, it will be decomposed in $\underline{\underline{A}} = \underline{\underline{L}} \cdot \underline{\underline{L}}^*$.

It is commonly used in solving system of linear equations in the form $\underline{\underline{A}} \cdot \underline{x} = \underline{b}$. This can be easily solved, if the matrix $\underline{\underline{A}}$ is symmetric and positive definite, by solving as first $\underline{\underline{L}} \cdot \underline{y} = \underline{b}$ for $\underline{y}$ and then $\underline{\underline{L}}^T \cdot \underline{x} = \underline{y}$ for $\underline{x}$. Here the algorithm to compute the Cholesy decomposition:

Algorithm E.4 Cholesky-Banachiewicz algorithm for Cholesky decomposition

```
Require: symmetric and positive definite matrix $\underline{\underline{A}} \in \mathbb{R}^{m \times m}$
Ensure: lower triangular matrix $\underline{\underline{L}}$, with $\underline{\underline{L}} \cdot \underline{\underline{L}}^T = \underline{\underline{A}}$
$\underline{\underline{L}} \leftarrow 0$
for $j = 1$ to $m$ do
    $l_{j,j} \leftarrow \sqrt{a_{j,j} - \sum_{c=1}^{j-1} l_{j,c}^2}$
```

---

[2] A Hermitian matrix is a square matrix equal to its conjugate transpose [117]

```
    for $i = j + 1$ to $m$ do
        $l_{i,j} \leftarrow \frac{1}{l_{j,j}} \cdot \left( a_{i,j} - \sum_{c=1}^{j-1} l_{i,c} \cdot l_{j,c} \right)$
    end for
end for
```

Appendix

F

Software acknowledgment

During my professional life I experienced different kind of software, from very simple one till professional suites. Many software-houses developed very amazing and powerful applications able to fill easily the gap between the idea and the product.

If several years ago professional applications were for companies and simple applications were for home-made products, now it is not true anymore. It is easy to find any kind of software in any house. Moreover the growing of free-ware and open-source applications, especially with the growing of *GNU/Linux* as operating system symbol of open-source projects, set the beginning of a new era, where comunity of programmers join together to create a suitable solution for a problem up to now just solved by professional applications.

My interests in programming and in open-source projects made me closer to the use of these softwares. They usually are far from a professional application, but very powerful considering that they survive just with donations.

In this thesis, as for many of my professional or personal works.

Typesetting

At the beginning of my computer experience, I started with MS-DOS[1] and with Microsoft ©Works[2]. It was my first experience in typewriting without the classical typewriter. After some years Microsoft©Windows arrived almost in any house of the planet and I started to use it too. My experience switched to Microsoft©Office, still the most famous and used suite for the office.

But the interests for free software and open-source projects made me closer to suites like OpenOffice.org. After years of using both Microsoft©and open-source solutions, I met LaTeX. As first I started with my master thesis and then I used it for many technical reports and for this thesis too. The level of quality of a product made by LaTeX is absolutely higher then Microsoft©Office or OpenOffice.org. Moreover the ease in managing the figures, bibliography or references to them is astonished.

---

[1] Acronim for *MicroSoft Disk Operating System*, in use from 1982 till 2001

[2] Under MS-DOS from version 1.05 till version 3.0b

From `http://www.latex-project.org/` "*LaTeX is a high-quality typesetting system; it includes features designed for the production of technical and scientific documentation. LaTeX is the* de facto *standard for the communication and publication of scientific documents. LaTeX is available as free software*".

Here some websites where to find informations or softwares related to LaTeX:

- **LaTeX** official website: `http://www.latex-project.org`
- **MikTeX**, Microsoft©Windows implementation of TeX Tex: urlhttp://www.miktex.org/
- **TeXnicCenter**, Integrated environment for creating LaTeX documents on the Windows platform: `http://www.texniccenter.org/`
- **TeXShop**, free TeX editor and previewer for Mac OS X: `http://www.uoregon.edu/~koch/texshop/`
- **TeXworks**, simple TeX front-end program deliberately modeled on Dick Koch's award-winning TeXShop for Mac OS X: `http://www.tug.org/texworks/`

Figures

In order to prepare high quality figures I tried to use vectorial graphics and softwares may help me in this purpose. The main format I used is SVG, then converted in PDF to be used in LaTeX still as vectorial graphics. In case of raster pictures I preferred to use PNG file format. A lossless format that has a 8-bit alpha channel apart from 24-bit RGB color information. I used different applications from the open-source community and I would like to mention:

- **Blender**, free open source 3D content creation suite: `http://www.blender.org`
- **Inkscape**, open-source vector graphics editor using the W3C standard Scalable Vector Graphics (SVG) file format: `http://www.inkscape.org`
- **The GIMP**, versatile graphics manipulation package: `http://www.gimp.org`
- **Gnuplot**, portable command-line driven interactive data and function plotting utility: `http://www.gnuplot.info/`

Programming

I started using computers early in my childhood. When I was 9 my father bought an Olivetti PC-286 with 2 MB of RAM and 27 MB of hard disk. The operating system was MS-DOS e the main software I was using was Microsoft©Works, in order to write for him the monthly invoices. After few years Microsoft©Windows 3.1 arrived to the market and I bought it as it seamed at that time a great advantage respect to MS-DOS. At the beginning was mostly a "game" than a real working station for me, but then I discovered a small hidden icon in the workspace of Microsoft©Windows: GW-BASIC[3], a dialect of BASIC developed by Microsoft©from BASICA[4].

I started to get interested in the meaning of programming and the power of it. I started with very simple things, even quite useless, but very attractive for me. During the next years I continued passing to QBasic[5] and then to more powerful languages like Pascal (imperative and structured language) that allowed me to start a better and more professional programming. My first useful product was a complete management system for my father for the writing, printing and cataloging of waybills (made when I was about 12 years old). The user interface was easy and clean (made mainly directly in *Assembly*[6]) and it was designed to help the user and to cope with every mistake of him/her.

During the years of high school I started to get familiar with Visual©Basic that became my common language for several years, before starting with C++. Now, apart web-based language, I usually use even C++ or Visual©Basic. The former for its powerful structure and high performance; the latter for its easy-to-use, especially when complex user interface are needed.

During the years of University and moreover during my PhD I also experienced sporadic use of Fortran[7] and Python[8], as common languages for scripting inside engineering softwares as Finite Element Softwares.

Here I present some of the tools I use generally and that I used in this thesis too:

- **Microsoft©Visual Studio©Express**: `http://www.microsoft.com/express/`
- **Cygwin**, Unix-like environment and command-line interface for Microsoft Windows: `http://www.cygwin.com/`
- **wxWidgets** lets developers create applications for Win32, Mac OS X, GTK+, X11, Motif, WinCE, and more using one codebase: `http://www.wxwidgets.org/`
- **MinGW**, port of the GNU Compiler Collection (GCC), and GNU Binutils, for use in the development of native Microsoft Windows applications: `http://www.mingw.org/`
- **MSYS**: `http://www.mingw.org/wiki/msys`
- **GNU GCC**, Compiler Collection including front ends for C, C++, Objective-C, Fortran, Java and Ada, as well as libraries for these languages: `http://gcc.gnu.org/`
- **OpenMP**, support for multi-platform shared-memory parallel programming in C/C++ and Fortran: `http://openmp.org/wp/`

---

[3] Acronim for *Graphics and Windows Beginner's All-purpose Symbolic Instruction Code*

[4] IBM BASICA (short for "Advanced BASIC") is a simple disk-based BASIC interpreter written by Microsoft for PC-DOS

[5] IDE and interpreter for a variant of the BASIC programming language which is based on QuickBasic

[6] Low level language composed of a series of instructions-mnemonics that correspond to a stream of executable instructions

[7] general-purpose, procedural, imperative programming language that is especially suited to numeric computation and scientific computing

[8] general-purpose high-level programming language which design philosophy emphasizes code readability

- **Boost**, free peer-reviewed portable and powerful C++ source libraries: `http://www.boost.org/`
- **Python$^{\mathrm{TM}}$**: `http://www.python.org/`

Most of these are described and presented in the chapters related to the description of software.