



# Classic and Inverse Compositional Gauss-Newton in Global DIC

Jean-Charles Passieux, Robin Bouclier

## ► To cite this version:

Jean-Charles Passieux, Robin Bouclier. Classic and Inverse Compositional Gauss-Newton in Global DIC. International Journal for Numerical Methods in Engineering, 2019, 119 (6), pp.453-468. <10.1002/nme.6057>. <hal-02059472>

**HAL Id: hal-02059472**

**<https://hal.science/hal-02059472v1>**

Submitted on 13 Jun 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# Classic and Inverse Compositional Gauss-Newton in Global DIC

J.-C. Passieux<sup>\*1</sup> and R. Bouclier<sup>1,2</sup>

1. Institut Clément Ader, Université de Toulouse, CNRS/INSA/UT3/Mines Albi/ISAE, France

2. Institut de Mathématiques de Toulouse, Université de Toulouse, CNRS/UT1/UT2/UT3/INSA, France

## SUMMARY

Today, effective implementations of Digital Image Correlation (DIC) are based on iterative algorithms with constant linear operators. A relevant idea of the classic Finite Element (or more generally global) DIC (FE-DIC) solver consists in replacing the gradient of the deformed state image with that of the reference image, so as to obtain a constant operator. Different arguments (small strains, small deformations, equality of the two gradients close to the solution...) have been given in the literature to justify this approximation, but none of them are fully accurate. Indeed, the convergence of the optimization algorithm has to be investigated from its ability to produce descent directions. Through such a study, this paper attempts to explain why this approximation works and what is its domain of validity. Then an Inverse Compositional Gauss-Newton (ICGN) implementation of FE-DIC is proposed as a cost effective and mathematically sound alternative to this approximation.

KEY WORDS: global DIC ; inverse compositional Gauss-Newton, forward compositional Gauss-Newton

## 1. INTRODUCTION

Experimental mechanics has been experiencing a veritable revolution for the past twenty years, thanks to numerous electromagnetic (visible, IR, UV, RX, terahertz, etc.) and electronic imaging modalities and derived techniques (stereo, volume, confocal, hyperspectral imaging, etc.). Imaging offers the possibility to measure mechanical fields that could redefine standard mechanical characterization procedures used daily in the industry and significantly reduce design time of structures. As the diversity and the definition of sensors are increasing, the volume of data generated during a mechanical test is growing, which implies developing more and more efficient image analysis algorithms.

In experimental mechanics, Digital Image Correlation (DIC) has become one of the most commonly used full-field measurement methods, because of its simplicity of implementation and its modularity (no intrinsic physical scale). The first work is attributed to Lucas and Kanade [22] and the first application in experimental mechanics is due to Sutton [42]. These pioneering works propose to split the image into subsets and to look for the two components of

---

\*Correspondence to: passieux@insa-toulouse.fr

the rigid displacement for each sub-image. The maximization of the cross-correlation is done effectively in the Fourier domain. To be able to consider a deformation of the subset, iterative algorithms are used. Most of the time, a descent algorithm is used: Steepest Descent [9], Newton-Raphson [6], Gauss-Newton [4], Levenberg-Marquardt [8]. In experimental mechanics, measured kinematic fields are useful for the validation—or the identification of the parameters of—finite element models [12, 21]. The use of a finite element interpolation of the displacement in the DIC algorithm (FE-DIC, [17, 41]) thus appears as a natural choice in this field [3]. To solve the non-linear least-squares problem encountered in FE-DIC, a descent algorithm is used as well.

Most of the descent methods mentioned above lead to the resolution of a succession of linear systems which provide the search direction at each iteration. The most numerically efficient algorithms are those that lead to a constant linear operator (iteration independent), so that it can be assembled and factorized only once in a pre-processing step.

From this point of view, the most mature method in subset-DIC is certainly the Inverse Compositional Gauss-Newton (ICGN) [1, 2, 28, 29, 43, 40]. The elegant idea behind is to perform a specific linearization of the minimization problem in such a way that the use of a Gauss-Newton naturally leads to a constant operator. For global approaches, and thus for FE-DIC, the linear operator is made constant thanks to the following trick: at iteration  $k$ , the gradient  $\nabla g \circ \phi_k$  (that is of the deformed image  $g$  advected by the current approximation of the mapping  $\phi_k$ ) is replaced by the gradient  $\nabla f$  (*i.e.* of the reference image  $f$ ). The mathematical foundations of this approximation are not clearly established. Several arguments are proposed in the literature to justify this approximation (small strains or small deformations, equality of the two gradients close to the solution, etc) but none of them are strictly accurate or, at least, some of them are too restrictive. From here on, it seems important to clarify the terminology used for the standard continuum mechanics assumptions mentioned in this paper. The case of small deformations means that displacements, rotations and strains are small while only the strains are assumed to be small in the small strain or large deformation context. In the same idea, we will use the naming large strains when large strains (and so, including large deformations) are expected. In the article by Neggers *et al.* [24], a first analysis provides interesting results. The authors showed that some of the arguments put forward in the literature are not totally correct. Based on several examples, it was shown that this approximation may reduce the efficiency of the method and sometimes even prevent it from converging. The outcome of this paper was that the only algorithms that always converge effectively are those which require updating of the linear operator.

In this article, a mathematical understanding of the standard FE-DIC algorithm is proposed. The validity domain of the underlying approximation is being studied. A condition on FE-DIC operators is proposed. We try to interpret it as a condition on the nature of the measured displacement field. In a second step, we propose an alternative based on the ICGN algorithm which (1) comes with a constant linear operator; (2) converges even in cases where classical FE-DIC fails and (3) leads to a minimal computing overhead compared to conventional FE-DIC.

The document is organized as follows. In Section 2, the DIC problem is first formulated in an algebraic setting. Then some basics regarding gradient based optimization algorithms are recalled (classical algorithms and related theoretical results). In section 3, we propose an analysis of the classic algorithm used in FE-DIC and an alternative based on ICGN is presented. Finally, in Section 4, we present some examples from synthetic images to numerically confirm

the above theoretical results.

## 2. FE-DIC: RESOLUTION OF A NON-LINEAR LEAST-SQUARES PROBLEM

In this section, FE-DIC is introduced from an optimization point of view. After writing the FE-DIC problem algebraically to exhibit its non-linear least-squares nature in  $\mathbb{R}^m$  ( $m$  being the number of data), the standard descent algorithms that are usually considered to solve such an unconstrained optimization problem are built. This provides a global framework for the resolution and enables in particular to better appreciate, from a mathematical point of view, the foundations of the usual solver performed in FE-DIC.

### 2.1. FE-DIC

*2.1.1. Continuous formulation.* The principle of Digital Image Correlation (DIC) is to experimentally measure a kinematic transformation  $\phi(x, u)$  of any point  $x$  of the region  $\Omega$  at the surface of a specimen by the comparative analysis of graylevel digital images taken before  $f(x)$  and after  $g(x)$  the transformation, see Figure 1. Given  $f(x)$  and  $g(x)$ , DIC thus

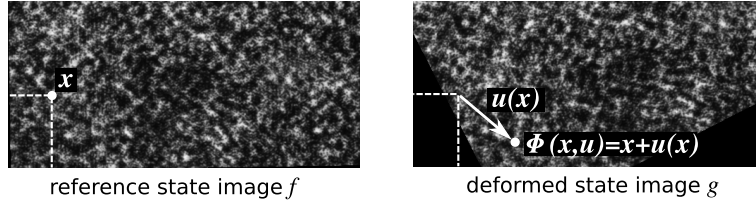


Figure 1. Graylevel conservation problem: Find transformation  $\phi$  (or displacement  $u$ ) that registers the two graylevel digital images  $f(x)$  and  $g(x)$ .

consists in computing the transformation  $\phi(x, u)$  as the solution of the graylevel conservation equation [15]:

$$r(x, u) = f(x) - g \circ \phi(x, u) = 0, \quad \forall x \in \Omega, \quad (1)$$

where a digital image  $f$  maps any sampling point  $x \in \Omega$  to a quantized graylevel value  $f(x)$ , and  $\circ$  refers to the composition operator between two functions. The residual map  $r(x, u)$ , quantifying the non-compliance with the graylevel conservation, can be used to validate or improve the kinematic model [36, 25]. The unknown transformation  $\phi(x, u)$  is of the form:

$$\phi(x, u) = x + u(x), \quad (2)$$

where  $u(x)$  is the unknown displacement field. The latter is usually sought in an approximation subspace  $\mathcal{U}_h$  spanned by a set of chosen basis functions  $(N_i)_{i \in \{1, \dots, n\}}$ :

$$u(x) = \sum_{i=1}^n N_i(x) u_i, \quad (3)$$

in which the  $(u_i)_{i \in \{1, \dots, n\}}$  are the unknown coefficients of the linear combination, the so-called degrees of freedom (DOF). Depending on the choice made for  $N_i$ , the DIC methods are divided

into two main families: subset methods using mostly low-order piecewise polynomials that are discontinuous across the subsets [42] whereas global methods mostly use finite elements that are also low-order piecewise polynomials, but continuous across elements [17].

*Remark:* Written like this, the problem consists in finding the transformation that brings the deformed state image back to the reference image. The associated displacement  $u(x)$  is then expressed in the Lagrangian specification. However, it is possible to formulate this problem reciprocally ( $f \circ \phi(x, \tilde{u}) - g(x) = 0$ ), namely to seek transformation  $\phi(x, \tilde{u})$  which transports the reference image to the distorted image [44, 20, 16]. In this case, the displacement  $\tilde{u}$  is in the Eulerian specification. To move from one formulation to the other, it just has to be noticed that  $\phi(x, \tilde{u}) = \phi(x, u)^{-1}$ .

Problem (1) being ill-posed, it is classically written as a non-linear least-squares problem over domain  $\Omega$ :

$$u^* = \arg \min_{u \in \mathcal{U}_h} \frac{1}{2} \sum_{p=1}^m \omega_p \left( f(x_p) - g \circ \phi(x_p, u) \right)^2, \quad (4)$$

where the  $(x_p)_{p \in \{1, \dots, m\}}$  are the  $m$  integration points and  $(\omega_p)_{p \in \{1, \dots, m\}}$  the corresponding quadrature weights.

*Remark:* One may notice that a more general FE-DIC formulation relies on camera models  $p$ :

$$u^* = \arg \min_{u \in \mathcal{U}_h} \frac{1}{2} \sum_{p=1}^m \omega_p \left( f \circ p(X_p) - g \circ p \circ \phi(X_p, u) \right)^2. \quad (5)$$

A camera model  $p(X)$  maps any point  $X$  in the 3D space  $\Omega^{FE}$  to its corresponding point  $x = p(X)$  in the image frame  $\Omega$ . This formulation is required in stereo FE-DIC (FE-SDIC) [33, 39] but may improve also 2D FE-DIC [34]. This is the method implemented in the Python code used here for the numerical experiments (see section 4). In the following, for the sake of clarity, the camera models are omitted. Their presence does not add any particular difficulty. Only the number, the location and the weight  $\omega_p$  (including quadrature weight and determinant of the jacobian of the transformation) of the integration points would be different. In the following, we will also consider that the quadrature is performed in the image frame using pixels centers as integration points. Like this, the weights are all equal to  $\omega_p = 1$  (rectangle-type rule) which further improves readability without altering generality.

*2.1.2. Algebraic reformulation.* We now reformulate the FE-DIC problem (4) algebraically so as to be closer to the standard writing encountered in an optimization framework. For the sake of clarity, we consider a one-dimensional problem for the presentation in the following. The displacement  $u$  (see Eq. (3)) thus constitutes an unidirectional field and  $x_p$  refers to a one-dimensional coordinate that locates pixel  $p$ . We emphasize that such a choice is adopted only to ease the reading and does not alter generality, the implementation being, however, achieved in 2D (see section 4). To begin with, let us introduce vectors  $\mathbf{u}$  and  $\mathbf{x}$  that collects, respectively, the  $n$  parameters of our model (*i.e.*, the DOF of  $u$ ) and the  $m$  positions of our

data (*i.e.*, the pixels):

$$\mathbf{u} = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} \quad \text{and} \quad \mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}. \quad (6)$$

Regarding notations, we use capital letters in boldface type for matrices, lowercase in boldface type for vectors and letters in normal type for scalar. Let us also define the images as vectors which gather the graylevel values associated to the different pixels:

$$\mathbf{f} : \mathbf{x} \rightarrow \mathbf{f}(\mathbf{x}) = \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_m) \end{pmatrix} \quad \text{and} \quad \mathbf{g} : \mathbf{x} \rightarrow \mathbf{g}(\mathbf{x}) = \begin{pmatrix} g(x_1) \\ \vdots \\ g(x_m) \end{pmatrix}. \quad (7)$$

The kinematic transformation  $\phi$  between reference image  $\mathbf{f}$  and deformed image  $\mathbf{g}$  then reads:

$$\phi : (\mathbf{x}, \mathbf{u}) \rightarrow \phi(\mathbf{x}, \mathbf{u}) = \mathbf{x} + \mathbf{N}(\mathbf{x})\mathbf{u}, \quad (8)$$

where  $\mathbf{N}$  is a  $m \times n$  matrix that collects the value of the basis functions at the pixels:

$$\mathbf{N} = \begin{pmatrix} N_1(x_1) & \dots & N_n(x_1) \\ \vdots & \ddots & \vdots \\ N_1(x_m) & \dots & N_n(x_m) \end{pmatrix}. \quad (9)$$

With above notations, problem (4) consists of solving, for a given  $\mathbf{x}$ , the following non-linear least-squares problem:

$$\mathbf{u}^* = \arg \min_{\mathbf{u} \in \mathbb{R}^n} j(\mathbf{u}), \quad (10)$$

where functional  $j$  reads:

$$j(\mathbf{u}) = \frac{1}{2} \|\mathbf{r}(\mathbf{x}, \mathbf{u})\|_2^2 = \frac{1}{2} \mathbf{r}(\mathbf{x}, \mathbf{u})^T \mathbf{r}(\mathbf{x}, \mathbf{u}), \quad (11)$$

the residual map  $\mathbf{r}(\mathbf{x}, \mathbf{u})$  being defined as:

$$\mathbf{r} : (\mathbf{x}, \mathbf{u}) \rightarrow \mathbf{r}(\mathbf{x}, \mathbf{u}) = \mathbf{f}(\mathbf{x}) - \mathbf{g} \circ \phi(\mathbf{x}, \mathbf{u}). \quad (12)$$

## 2.2. Descent algorithms

**2.2.1. Basics.** Problem (10) formally consists in an unconstrained optimization problem over  $\mathbb{R}^n$ , which is, provided the functional is at least differentiable, classically solved using a descent algorithm. The principle is as follows: starting from an initial guess  $\mathbf{u}^{(0)}$ , a descent algorithm iterates to generate a series  $(\mathbf{u}^{(k)})_{k \in \mathbb{N}}$  such that:

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + s^{(k)} \mathbf{d}^{(k)}. \quad (13)$$

$\mathbf{d}^{(k)}$  is called the descent direction. To ensure that it is a descent direction (*i.e.*, that we are able to minimize the functional along this direction),  $\mathbf{d}^{(k)}$  must satisfy:

$$\nabla j(\mathbf{u}^{(k)})^T \mathbf{d}^{(k)} < 0, \quad (14)$$

where  $\nabla j$  denotes the gradient of  $j$  with respect to  $\mathbf{u}$ , namely:

$$\nabla j = \begin{pmatrix} \frac{\partial j}{\partial u_1} \\ \vdots \\ \frac{\partial j}{\partial u_n} \end{pmatrix}. \quad (15)$$

$s^{(k)} > 0$  is the step. The procedure consisting of determining it is called the line search procedure. Depending on the choice of the descent direction, several algorithms can be constructed. The most standard ones are written in the following for the FE-DIC functional, *i.e.*, for  $j$  (see Eq. (11)). In particular, the developments hereafter will enable to better understand the foundations of the usual FE-DIC solver which merely consists of a modified Gauss-Newton (see section 3.1).

*2.2.2. Gradient method.* As its name suggests, the gradient algorithm relies on the use of the gradient of the functional to define the descent direction. More precisely,  $\mathbf{d}^{(k)} = -\nabla j(\mathbf{u}^{(k)})$  is chosen, so that relation (14) is automatically satisfied.  $\mathbf{x}$  being fixed, we drop it in the following. As a result, we notice that when a differential operator of the residual map  $\mathbf{r}(\mathbf{x}, \mathbf{u})$  is computed, this is the derivation with respect to  $\mathbf{u}$  that is implicitly performed. Let us start by computing the gradient of functional  $j$ . Given its least-squares structure, we obtain:

$$\nabla j(\mathbf{u}) = \mathbf{J}_\mathbf{r}(\mathbf{u})^T \mathbf{r}(\mathbf{u}), \quad (16)$$

where  $\mathbf{J}_\mathbf{r}$  is the Jacobian matrix of  $\mathbf{r}$ , that is, the  $m \times n$  matrix such that:

$$\mathbf{J}_\mathbf{r} = \begin{pmatrix} \frac{\partial r_1}{\partial u_1} & \cdots & \frac{\partial r_1}{\partial u_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial r_m}{\partial u_1} & \cdots & \frac{\partial r_m}{\partial u_n} \end{pmatrix}. \quad (17)$$

Reminding that  $\mathbf{r}$  is a composed application, we can then write the following relation for  $\mathbf{J}_\mathbf{r}$  :

$$\mathbf{J}_\mathbf{r}(\mathbf{u}) = -\mathbf{J}_\mathbf{g}(\phi(\mathbf{u}))\mathbf{J}_\phi(\mathbf{u}), \quad (18)$$

which yields:

$$\mathbf{J}_\mathbf{r}(\mathbf{u}) = -\nabla \mathbf{G}(\mathbf{u})\mathbf{N}, \quad (19)$$

where  $\nabla \mathbf{G}(\mathbf{u})$  is a  $m \times m$  diagonal matrix that collects the values of  $\nabla g \circ \phi(x_p, u)$ , that is, the value of  $\nabla g$  evaluated at the deformed coordinate  $\phi(x_p, u)$  of pixel  $p$  (see Eqs. (2) and (3)). We end up with:

$$\nabla j(\mathbf{u}) = -\mathbf{N}^T \nabla \mathbf{G}(\mathbf{u})\mathbf{r}(\mathbf{u}), \quad (20)$$

and therefore the gradient method series is built as follows:

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + s^{(k)}\mathbf{N}^T \nabla \mathbf{G}(\mathbf{u}^{(k)})\mathbf{r}(\mathbf{u}^{(k)}). \quad (21)$$

Although this method may appear interesting since no linear system resolution at each iteration is required, this procedure is rarely used in DIC (see [9] for instance) because it is known to converge very slowly (only the information regarding the first derivative of the functional is used), even with a suitable line search strategy. Regarding the latter, one may wish to perform the optimal line search strategy, that is, find  $s_k$  solution to:

$$s_k = \arg \min_{s>0} j(\mathbf{u}^{(k)} + s \mathbf{d}^{(k)}) \quad (22)$$

at each iteration. In practice, given the complexity and the cost of problem (22), this is rarely performed. The step is usually chosen more pragmatically so that it is not too small to limit the number of iterations of the algorithm and it is not too large to ensure its convergence. However, the gradient algorithm has the interest to be robust since it intrinsically verifies condition (14) and provided the step is adequately picked, it converges towards a critical point of  $j$  (*i.e.*, such that  $\nabla j(\mathbf{u}) = \mathbf{0}$ ).

*2.2.3. Newton method.* To define the descent direction in the Newton method, it is required to additionally compute the Hessian matrix of  $j$ , that is, the symmetric matrix such that:

$$\mathbf{H}_j = \begin{pmatrix} \frac{\partial^2 j}{\partial u_1^2} & \cdots & \frac{\partial^2 j}{\partial u_1 \partial u_n} \\ \vdots & \ddots & \vdots \\ sym & \cdots & \frac{\partial^2 j}{\partial u_n^2} \end{pmatrix}. \quad (23)$$

From here on, it may be noticed that the second derivatives (*i.e.*, the curvature of the functional) are used for the minimization. As a result, the method will exhibit a higher rate of convergence as compared to that of the gradient method. Exploiting again the least-squares structure of functional  $j$ , one can write:

$$\mathbf{H}_j(\mathbf{u}) = \mathbf{J}_r(\mathbf{u})^T \mathbf{J}_r(\mathbf{u}) + \sum_{p=1}^m \mathbf{H}_{r_p}(\mathbf{u}) r_p(\mathbf{u}), \quad (24)$$

where  $\mathbf{H}_{r_p}$  refers to the Hessian matrix of component  $p$  of  $\mathbf{r}$ , *i.e.*, of  $r_p = r(x_p, u) = f(x_p) - g \circ \phi(x_p, u)$  (see Eq. (1)). Making use of Eq. (19), the first term above is easily estimable. However, the computation of  $\mathbf{H}_{r_p}(\mathbf{u})$  may require further developments. In order to do so, one may first compute  $\nabla r_p(\mathbf{u})$ , and then, the Jacobian matrix associated to this gradient:  $\mathbf{H}_{r_p}(\mathbf{u}) = \mathbf{J}_{\nabla r_p}(\mathbf{u})$ . Once again, it has to be underlined that both of these operations require the derivation of a composed application. Making these precautions, we finally get:

$$\mathbf{H}_j(\mathbf{u}) = \mathbf{N}^T \nabla \mathbf{G}(\mathbf{u}) \nabla \mathbf{G}(\mathbf{u}) \mathbf{N} - \mathbf{N}^T \mathbf{H}_g(\mathbf{u}) \mathbf{R}(\mathbf{u}) \mathbf{N}, \quad (25)$$

where  $\mathbf{H}_g(\mathbf{u})$  and  $\mathbf{R}(\mathbf{u})$  are  $m \times m$  diagonal matrices that collect, respectively,  $H_g \circ \phi(x_p, u)$  and  $r(x_p, u)$ . With the Hessian matrix at hand, the descent direction of the Newton method is obtained through the following linear system resolution:

$$\mathbf{H}_j^{(k)} \mathbf{d}^{(k)} = -\nabla j(\mathbf{u}^{(k)}). \quad (26)$$

$\mathbf{d}^{(k)}$  above does constitute a descent direction (see Eq. (14)) when  $\mathbf{H}_j^{(k)}$  (*i.e.*  $\mathbf{H}_j(\mathbf{u}^{(k)})$ ) is positive definite. Indeed, we can write in this case:

$$-\nabla j(\mathbf{u}^{(k)})^T \mathbf{d}^{(k)} = -\nabla j(\mathbf{u}^{(k)})^T \left[ \mathbf{H}_j^{(k)} \right]^{-1} \nabla j(\mathbf{u}^{(k)}) < 0. \quad (27)$$

With such a descent direction, it can also be shown that a good choice for the step is, for all iteration,  $s^{(k)} = 1$ . We thus end up with the following procedure to build the Newton method series:

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} - \left[ \mathbf{H}_j^{(k)} \right]^{-1} \nabla j(\mathbf{u}^{(k)}), \quad (28)$$



where the Hessian and the gradient operators read as in Eqs. (25) and (20), respectively.

The interest of the Newton method is that it converges quadratically. However, only a local convergence can be proven. In addition, the process requires the assembling and the resolution of a linear system at each iteration of the algorithm, which may not be desirable from a computational cost point of view. Finally, note that the presence of the second derivatives of the image may increase the sensitivity to noise.

*2.2.4. Gauss-Newton method.* The Gauss-Newton method is especially built to address the minimization of a non-linear least-squares functional, which is the case in FE-DIC. The process is particularly appealing in this case since it somehow combines the interests of the gradient method (robustness) and of the Newton strategy (convergence speed). The derivation of the algorithm can be performed in two different ways. The first one, which is almost never carried out in the field of FE-DIC, consists of simplifying the Newton method. Indeed, it may be noticed that once the correlation residual  $\mathbf{r}(\mathbf{u})$  (and so  $\mathbf{R}(\mathbf{u})$ ) is small, that is, when we get close to the intended solution, the second term in the Hessian matrix (25) becomes insignificant. The Gauss-Newton method simply consists of neglecting this term. Using a step-size of 1 ( $s^{(k)}=1$ ), the following iterative procedure is thus obtained:

$$\tilde{\mathbf{H}}_j^{(k)} \mathbf{d}^{(k)} = \mathbf{b}_G^{(k)} \quad \text{with} \quad \mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \mathbf{d}^{(k)}, \quad (29)$$

and

$$\begin{cases} \tilde{\mathbf{H}}_j^{(k)} &= \mathbf{J}_r(\mathbf{u}^{(k)})^T \mathbf{J}_r(\mathbf{u}^{(k)}) &= \mathbf{N}^T \nabla \mathbf{G}(\mathbf{u}^{(k)}) \nabla \mathbf{G}(\mathbf{u}^{(k)}) \mathbf{N}, \\ \mathbf{b}_G^{(k)} &= -\nabla j(\mathbf{u}^{(k)}) &= \mathbf{N}^T \nabla \mathbf{G}(\mathbf{u}^{(k)}) \mathbf{r}(\mathbf{u}^{(k)}), \end{cases} \quad (30)$$

where  $\tilde{\mathbf{H}}_j^{(k)}$  approximates the Hessian matrix  $\mathbf{H}_j^{(k)}$  of  $j$  with only first derivatives. The right hand side remains the same as for Newton method (26). As  $\tilde{\mathbf{H}}_j^{(k)}$  still depends on  $\mathbf{u}^{(k)}$ , the computational cost of the standard Gauss-Newton algorithm may still appear potentially important since one assembling and one factorization need to be performed at each iteration.

*Remark:* The Gauss-Newton algorithm can also be constructed by replacing, at each iteration, the non-linear least-squares problem to be solved by an approximate, linear least-squares problem. This second approach for obtaining the solver is the one that is almost always carried out in the DIC community. It is built as follows: given the current solution  $\mathbf{u}^{(k)}$ , problem (10) is replaced in the neighborhood of  $\mathbf{u}^{(k)}$  by:

$$\mathbf{v}^* = \arg \min_{\mathbf{v} \in \mathbb{R}^n} \left( \frac{1}{2} \left\| \mathbf{r}(\mathbf{u}^{(k)}) + \mathbf{J}_r(\mathbf{u}^{(k)}) \mathbf{d} \right\|_2^2 \right), \quad (31)$$

*i.e.*, where  $\mathbf{r}$  has been linearized around  $\mathbf{u}^{(k)}$ . Problem (31) being a linear least-squares problem: its solution  $\mathbf{u}^{(k+1)}$  satisfies the associated normal equations, namely:

$$\mathbf{J}_r(\mathbf{u}^{(k)})^T \mathbf{J}_r(\mathbf{u}^{(k)}) (\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}) = -\mathbf{J}_r(\mathbf{u}^{(k)})^T \mathbf{r}(\mathbf{u}^{(k)}), \quad (32)$$

which is equivalent to (29).

Applying the same reasoning as the one conducted for the Newton algorithm (see Eq. (27)), it can be shown that we do have a descent direction for the Gauss-Newton algorithm when  $\tilde{\mathbf{H}}_j^{(k)}$  is definite (it is positive by nature). In practice, provided the images possess a regular texture (*i.e.*, the gradient may vanish exclusively over a null measure subset), the condition is

satisfied in FE-DIC [10]. For higher performance, a line search strategy can also be performed to ensure the convergence of the algorithm. Therefore, this algorithm provides a good trade-off between robustness (we always have a descent direction), simplicity (no need to compute the second derivatives) and convergence speed (close to a quadratic convergence when  $\mathbf{r}(\mathbf{u})$  becomes small). Nevertheless, as stated above, we still need to solve a linear system (that is different) at each iteration, that is why the usual FE-DIC solver is not exactly a Gauss-Newton algorithm, as will be shown in the forthcoming section.

### 3. SOLVERS USED IN DIC

In this section, we present the algorithm conventionally used in FE-DIC. Based on the theoretical results recalled in the previous section, we try to understand how and why it works. A new algorithm based on the compositional family of solvers is also proposed. It will be shown that it converges efficiently when the conventional one diverges.

#### 3.1. Classic solver: a modified Gauss-Newton

In the context of FE-DIC, starting from the Gauss-Newton algorithm (29), a common (and quite smart) practice is to replace term  $\nabla \mathbf{G}(\mathbf{u}^{(k)})$  by  $\nabla \mathbf{F}$  (*i.e.* the diagonal matrix collecting  $\nabla f(x_p)$ ) in the construction of the operators of the linear system to be solved [14, 19]. The resulting algorithm is no longer a Gauss-Newton. It will be referred to as modified Gauss-Newton (mod.-GN) herein:

$$\tilde{\mathbf{H}}_j \mathbf{d}^{(k)} = \mathbf{b}_F^{(k)} \quad \text{with} \quad \mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \mathbf{d}^{(k)}, \quad (33)$$

and

$$\begin{cases} \tilde{\mathbf{H}}_j &= \mathbf{N}^T \nabla \mathbf{F} \nabla \mathbf{F} \mathbf{N}, \\ \mathbf{b}_F^{(k)} &= \mathbf{N}^T \nabla \mathbf{F} \mathbf{r}(\mathbf{u}^{(k)}). \end{cases} \quad (34)$$

The interest of making such a modification is obvious since the linear operator  $\tilde{\mathbf{H}}_j$  can be assembled and factorized once before entering the Gauss-Newton iterations. However, as we leave the strict framework of Gauss-Newton, there is no proof that the method is convergent.

This approximation is sometimes justified in the literature (including in some articles published by the authors) by the fact that, by virtue of the graylevel conservation principle, these two quantities would be equal in the neighborhood of the solution [35, 18, 31, 37, 45, 32, 47, 24, 7, 26]. There is actually no reason why these quantities should be equal in general. The relationship between these two gradients can only be written at convergence. Denoting the converged solution by  $\mathbf{u}^{(\infty)}$ , the derivative with respect to  $\mathbf{x}$  of the graylevel conservation gives [24]:

$$\nabla \mathbf{F} = \nabla (\mathbf{g} \circ \phi(\mathbf{x}, \mathbf{u}^{(\infty)})) \quad (35)$$

$$= \nabla \phi(\mathbf{x}, \mathbf{u}^{(\infty)}) \nabla \mathbf{g} \circ \phi(\mathbf{x}, \mathbf{u}^{(\infty)}) \quad (36)$$

$$= (\mathbf{I} + \nabla \mathbf{U}^{(\infty)}) \nabla \mathbf{G}(\mathbf{u}^{(\infty)}) \quad (37)$$

$$= \mathbf{F}^{(\infty)T} \nabla \mathbf{G}(\mathbf{u}^{(\infty)}), \quad (38)$$

where  $\nabla \mathbf{U}$  and  $\mathbf{F}^{(\infty)}$  are  $m \times m$  diagonal matrices that collect the value of the gradient of the displacement and the gradient of the transformation at integration point  $\mathbf{x}_p$ , respectively.  $\mathbf{I}$  is the identity matrix of size  $m$ .

Even under these conditions, it has to be said that relation (38) is only an approximation because, in addition to the experimental disturbances (noise, light variation...), the spatial sampling and quantization of numerical images do not make it possible to have a strict equality. If the algorithm is well initialized (sufficiently close to the optimum), then this relation explains why sometimes the small deformation assumption ( $\nabla \mathbf{U} \ll \mathbf{I}$ ) is used to justify this approximation [11, 24, 30]. But as will be shown below, the framework of small deformations is a sufficient condition (although very restrictive), but it is certainly not a necessary condition.

To know if the algorithm can converge in spite of this important modification, the search direction it provides must satisfy condition (14), that is for the present situation:

$$-\left(\mathbf{b}_G^{(k)}\right)^T \tilde{\mathbf{H}}_j^{-1} \mathbf{b}_F^{(k)} < 0. \quad (39)$$

It may be noticed, by the way, that it would not be excessively expensive (scalar product only) to test this condition, for example, at the first iteration of the modified Gauss-Newton. This convergence condition on the search direction can be interpreted as the following condition on the unknown displacement field  $\mathbf{u}^{(\infty)}$ :

$$-\mathbf{r}^T \nabla \mathbf{G} \mathbf{N} \tilde{\mathbf{H}}_j^{-1} \mathbf{N}^T (\mathbf{I} + \nabla \mathbf{U}^{(\infty)}) \nabla \mathbf{G} \mathbf{r} < 0, \quad (40)$$

where the dependence on  $\mathbf{u}^{(k)}$  of  $\mathbf{r}$  and  $\nabla \mathbf{G}$  has been omitted for clarity. The transformation gradient  $\mathbf{F}^{(\infty)} = (\mathbf{I} + \nabla \mathbf{U}^{(\infty)})^T$  being diagonal, its positivity is a necessary and sufficient condition, in 1D at least, to fulfill condition (40). As a result, the non-compliance of relation (40) can be interpreted mechanically as the following condition on  $u$ :

$$\varepsilon_x = \frac{\partial u}{\partial x} < -1, \quad (41)$$

which is an unlikely event in continuum mechanics (overlap of matter). In any case, it is much less restrictive than the simple framework of small deformations. We conjecture that in the general case (2D), the positivity of the transformation gradient  $\mathbf{F}^{(\infty)}$  is closely related to the convergence of the modified GN algorithm. As will be illustrated in the examples (see section 4), in 2D, very large rotations can make this operator negative, thus preventing the algorithm from converging. However, there is no reason to proscribe general large deformations (or even large strains) with regard to condition (40).

*Remark.* Based on this consideration, a very simple way to build up an unconditionally descendant algorithm consists in replacing  $\nabla \mathbf{G}(u^{(k)})$  by  $\nabla \mathbf{F}$  for the construction of the approximate hessian only and to keep the exact expression of the right hand side:

$$\tilde{\mathbf{H}}_j \mathbf{d}^{(k)} = \mathbf{b}_G^{(k)} \quad (42)$$

As a quasi Gauss-Newton, this method is supposed to provide always a descent direction (14). From the numerical experiments we have done, this method worked well and in any case. We did not further study this solution since its computation cost involving  $\nabla \mathbf{G}(u^{(k)})$  for  $\mathbf{b}_G^{(k)}$  (with our implementation) is not sufficiently reduced as compared to that of the a standard Gauss-Newton method. Still, it may be a interesting alternative since it is safe and very easy to implement.

### 3.2. Inverse Compositional Gauss-Newton

The family of Compositionnal Gauss-Newton algorithms was first introduced in image processing by Baker and Matthews [1]. Many variants (inverse, forward, compositionnal, additive...) were then developed [2, 44] and applied to subset-DIC [44, 28, 43, 40] and subset-DVC [29].

The aim is still to develop a fast and robust solver with a constant operator. Let us recall that the classical algorithm in FE-DIC is a modified Gauss-Newton which, at each iteration, seeks a correction  $\mathbf{d}^{(k)}$  that best solve the following correlation problem, *i.e.* such as:

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{g} \circ \phi(\mathbf{x}, \mathbf{u}^{(k)} + \mathbf{d}^{(k)}). \quad (43)$$

The underlying idea of Inverse Compositional Gauss-Newton (ICGN) is rather to modify the linearization of the correlation problem at each iteration. More precisely, displacement  $\mathbf{u}^{(k)}$  being known, it consists in computing a correction  $-\tilde{\mathbf{d}}^{(k)}$  that transports image  $f$  to image  $\mathbf{g}(\mathbf{x} + \mathbf{u}^{(k)})$ :

$$\mathbf{f} \circ \phi(\mathbf{x}, -\tilde{\mathbf{d}}^{(k)}) \approx \mathbf{g} \circ \phi(\mathbf{x}, \mathbf{u}^{(k)}). \quad (44)$$

The resolution of this non-linear least-squares problem with a standard Gauss-Newton leads exactly to the same linear system (33) as with the modified GN method of the standard FE-DIC problem. It thus has the double advantage of leading to a constant operator while remaining within the strict framework of a Gauss-Newton (condition (14) systematically true). The link between these two algorithms can be established by composing equation (44) by the inverse of  $\phi(\mathbf{x}, -\tilde{\mathbf{d}}^{(k)})$ :

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{g} \circ \phi(\mathbf{x}, \mathbf{u}^{(k)}) \circ \phi^{-1}(\mathbf{x}, -\tilde{\mathbf{d}}^{(k)}). \quad (45)$$

As stated above, the linear system is strictly the same, only the update of the running approximation  $\mathbf{u}^{(k)}$  changes. Comparing Eq. (43) with Eq. (44), we notice that, while the new displacement approximation was simply a sum  $\mathbf{u}^{(k+1)} \leftarrow \mathbf{u}^{(k)} + \mathbf{d}^{(k)}$  with the modified GN algorithm, it consists now of the solution of the following problem:

$$\text{Find } \mathbf{u}^{(k+1)}, \text{ such that } \phi(\mathbf{x}, \mathbf{u}^{(k+1)}) = \phi(\mathbf{x}, \mathbf{u}^{(k)}) \circ \phi^{-1}(\mathbf{x}, -\tilde{\mathbf{d}}^{(k)}). \quad (46)$$

In the literature, the ICGN algorithm was mainly applied to subset DIC methods associated with affine transformations (rigid body translations, linear warps or homographies). In this case, the transformation admitted a writing as a linear operator (at least in homogeneous coordinates). In this context, the inversion and the composition was trivial, which certainly explains why this algorithm was implemented for a relatively long time in subset-DIC [1, 2].

The implementation of these two operations (inversion and composition) requires to estimate the value of the displacement field  $\mathbf{u}^{(k)}$  outside the points of definition (pixels or quadrature points) and often even outside its domain of definition (subset or FE mesh). The algebraic writing of a linear warp in subset based DIC implicitly extrapolate this warp outside the subset. To implement this method in the FE-DIC framework, we propose here to perform the inversion and the composition using a linearization of the transformation  $\phi$  with respect to  $\mathbf{x}$ :

$$\phi(\mathbf{x} + \boldsymbol{\delta}, \mathbf{u}^{(k)}) \approx \phi(\mathbf{x}, \mathbf{u}^{(k)}) + \mathbf{F}^{(k)} \boldsymbol{\delta}, \quad \forall \boldsymbol{\delta} \in \mathbb{R}^m \text{ small enough.} \quad (47)$$

In the remainder of the article, the two following compositional variants will be considered:

- Inverse Compositional Gauss-Newton (ICGN): inversion is performed using the linearization of the displacement according to Eq. (47). The inversion of  $\phi(\mathbf{x}, -\tilde{\mathbf{d}}^{(k)})$  thus becomes simple (resolution of a linear system), the associated computing cost is low. The composition is also performed considering the linearization of the displacement (47).
- Forward Compositional Gauss-Newton (FCGN): to further simplify the implementation, it is possible to consider the first order approximation of the inversion:  $\phi^{-1}(\mathbf{x}, -\tilde{\mathbf{d}}^{(k)}) \approx \phi(\mathbf{x}, \tilde{\mathbf{d}}^{(k)})$ . The resulting family of solvers is generally called *forward* as opposed to *inverse* [1]. The composition remain computed thanks to the linearization of the displacement (47). For example, with this approach, the update step simply reads:

$$\phi(\mathbf{x}, \mathbf{u}^{(k)}) \circ \phi^{-1}(\mathbf{x}, -\tilde{\mathbf{d}}^{(k)}) \approx \phi(\mathbf{x}, \mathbf{u}^{(k)}) \circ \phi(\mathbf{x}, \tilde{\mathbf{d}}^{(k)}) \quad (48)$$

$$\approx \phi(\mathbf{x} + \mathbf{N} \tilde{\mathbf{d}}^{(k)}, \mathbf{u}^{(k)}) \quad (49)$$

$$\approx \mathbf{x} + \mathbf{N} \mathbf{u}^{(k)} + \mathbf{F}^{(k)} \mathbf{N} \tilde{\mathbf{d}}^{(k)}, \quad (50)$$

$$\approx \underbrace{\mathbf{x} + \mathbf{N} \mathbf{u}^{(k)} + \mathbf{N} \tilde{\mathbf{d}}^{(k)} + (\nabla \mathbf{U}^{(k)})^T \mathbf{N} \tilde{\mathbf{d}}^{(k)}}_{\mathbf{N} \mathbf{u}^{(k+1)}}. \quad (51)$$

This modification of the modified GN scheme is similar to the one proposed in [13]. For the computation of  $\mathbf{u}^{(k)}$ , we need to map last term  $(\nabla \mathbf{U}^{(k)})^T \mathbf{N} \tilde{\mathbf{d}}^{(k)}$  from the pixels ( $\mathbb{R}^m$ ) to a vector of nodal values ( $\mathbb{R}^n$ ). This can be simply done by means of a least-squares projection. Finally the displacement is updated as:

$$\mathbf{u}^{(k+1)} \leftarrow \underbrace{\mathbf{u}^{(k)} + \tilde{\mathbf{d}}^{(k)}}_{\text{classic update}} + \underbrace{(\mathbf{N}^T \mathbf{N})^{-1} \mathbf{N}^T \nabla \mathbf{U}^{(k)} \mathbf{N} \tilde{\mathbf{d}}^{(k)}}_{\text{new corrective term}} \quad (52)$$

Most of the operators of this last correction term can be computed in a pre-processing phase and, as shown in the example section, the computational overhead with respect to the classic FE-DIC algorithm is almost insignificant.

Compared to the modified GN method, the only difference lies in the addition of a second correction term (obtained from two  $\frac{n}{2} \times \frac{n}{2}$  vector-matrix products in 2D) to  $\mathbf{u}^{(k)}$  at the end of each iteration.

### 3.3. Stopping criteria

Since all the methods presented above are iterative, it is necessary to define a stopping criterion. We generally consider two ways to stop the algorithm: i) a criterion based on the stagnation of the objective function or ii) a criterion based on the stagnation of the design variable (here the displacement field). In general in DIC, the algorithm is stopped when the criterion of stagnation on the displacement is lower than a certain value  $\epsilon$ :

$$\frac{\|s^{(k)} \mathbf{d}^{(s)}\|}{\|\mathbf{u}^{(k)}\|} < \epsilon. \quad (53)$$

This is what is also performed in the present work. It must be mentioned that, with such a stopping criteria, the accuracy of the solution may depend on the convergence rate of the algorithm.

## 4. EXAMPLES

## 4.1. Example 1: Rotation

This section investigates the rotation testcase proposed by Neggers et al. [24]. Let us consider a set of synthetic images (8bits,  $512 \times 512\text{px}^2$ ) representing rigid body rotations around the center of the image ranging between 0 and  $180^\circ$ . The displacement field is searched for in a finite element approximation subspace. The mesh simply consists of a single square four-noded bilinear element of 300px side as depicted in Figure 2. The reference speckle is synthesized from a Perlin noise [27]. The images are rotated using bicubic spline interpolation. Each image is analyzed in comparison to the reference image, see Figure 2 (left). Each time,

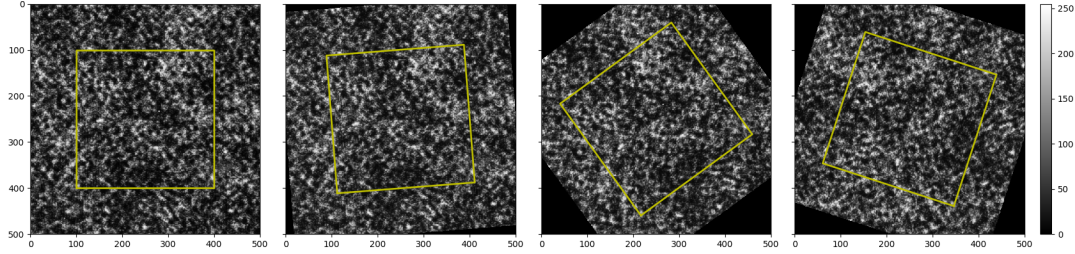


Figure 2. Images and mesh used for the rotation example illustrated with different angles  $\theta$ ; from left to right:  $\theta = 0^\circ$ ,  $\theta = 4.5^\circ$ ,  $\theta = 36^\circ$ ,  $\theta = 72^\circ$

the DIC algorithm is initialized with nodal displacements computed using the rotation angle  $\theta_{exact} - 1.8^\circ$ , such that the initial solution is sufficiently close to the solution to make gradient-based solvers converge to the intended solution, but sufficiently far to compare the efficiency of the considered solvers.

**4.1.1. Gauss-Newton and modified Gauss-Newton algorithms.** The image series is analyzed using standard (exact) Gauss-Newton (29) and the classic FE-DIC solver referred to as modified Gauss-Newton (mod.-GN) (33) as in previous section. Figure 3 presents the number of iterations required for each algorithm to reach a stagnation error of  $\epsilon = 10^{-4}$  as a function of the prescribed rotation angle. From a numerical point of view, the same trends as those obtained by Neggers et al. [24] are observed. Indeed, the standard Gauss-Newton algorithm converges whatever the angle with a number of iterations almost constant (about 10 iterations). The modified GN presents the same efficiency as the standard Gauss-Newton, at least for moderate angles, namely  $\theta \leq 20^\circ$ . For larger angles,  $20^\circ \leq \theta \leq 45^\circ$ , the modified GN still converges, but the number of iteration increases significantly. In this zone, the efficiency of the modified GN algorithm is reduced. Finally, as described in [24], for angles greater than  $45^\circ$ , the modified GN algorithm seems to no longer converge.

In order to consider this example from a theoretical standpoint, the evolution of the descent criterion for the modified GN method (39) is now plotted against the rotation angle in Figure 4. For this rotation test-case, it appears that property (14) is verified by the modified GN algorithm up to an angle of  $90^\circ$ . Although it is difficult to interpret this descent criterion by a condition on the displacement field  $u(x)$  in the general case, it seems here to correspond to the loss of positivity of  $\mathbf{F}^{(\infty)} = (\mathbf{I} + \nabla \mathbf{U}^{(\infty)})^T$ . Indeed, the symmetric part of  $\mathbf{F}^{(\infty)}$  has as double

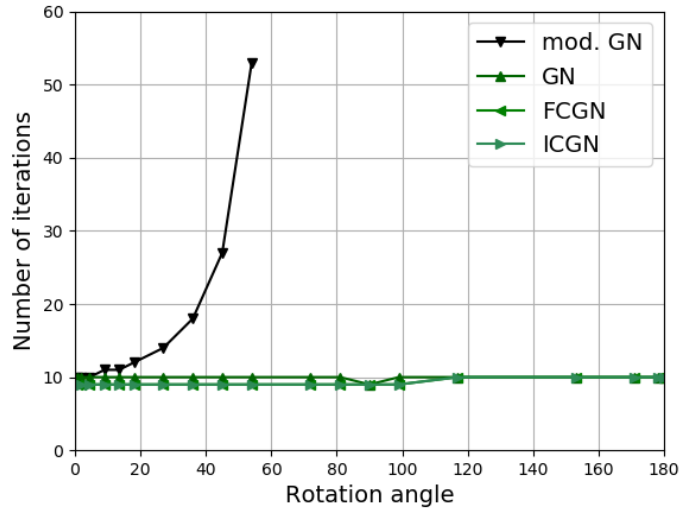


Figure 3. Number of GN iterations required to reach the convergence criterion as a function of the rotation angle  $\theta$  for the four considered Gauss-Newton variants.

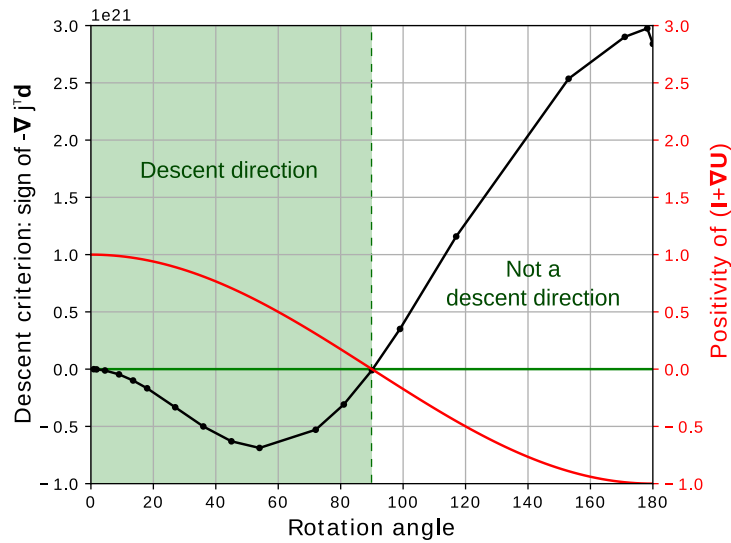


Figure 4. In black, sign of  $\nabla j^T \mathbf{d}$  as a function of the rotation angle. Negative values mean that the search direction  $\mathbf{d}$  provided by modified GN is a descent direction. In red, the eigenvalue of the symmetric part of  $\mathbf{F}^{(\infty)} = (\mathbf{I} + \nabla \mathbf{U}^{(\infty)})^T$ , negative for angles larger than  $90^\circ$ .

eigenvalue equal to  $\cos(\theta)$  which is negative for  $\theta > 90^\circ$ . This test case seems to empirically confirm the relevance of the condition of positivity of the transformation gradient  $\mathbf{F}^{(\infty)}$  as a condition to make the modified GN algorithm converge, as shown in 1D, see (41).

To sum up, from a theoretical point of view, the modified GN algorithm is supposed to converge for any angle less than  $90^\circ$ , whereas numerically, the basic algorithm does not seem to be able to reach convergence beyond a  $45^\circ$  angle. More precisely, the algorithm seems to fail to reach convergence with a step size of  $s^{(k)} = 1$  along the descent direction  $\mathbf{d}^{(k)}$ . Indeed, as explained in section 2.2.4, for a Gauss-Newton type algorithm a step size of 1 is not necessarily a good choice. In practice, a line search strategy may allow the modified GN algorithm to converge well beyond  $45^\circ$ . As long as the solution to linear system (33) is a descent direction, that is, if it complies with property (14), there is always a small enough step size that allows the modified GN algorithm to converge. To illustrate our purpose, we now consider the example of an angle of  $81^\circ$  (chosen significantly beyond  $45^\circ$ ). The displacement is initialized by the nodal displacements computed from the exact rotation angle. On Figure 5, the evolutions of the normalized residual (left) and the stagnation criterion (right) are plotted as a function of the GN iterations for the modified GN method with steps size of 1 (mod.-GN) and with a suitable step-size (Relaxed mod.-GN). On the one hand, the figure confirms that the algorithm with

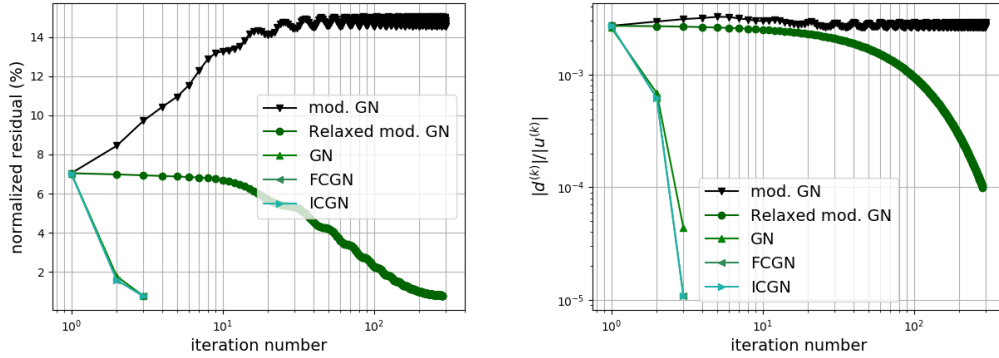


Figure 5. Convergence of the DIC solver for the rotation problem with  $\theta = 81^\circ$ , as a function of the number of GN iterations, for the five considered Gauss-Newton variants: (left) evolution of the standard deviation of the gray level residual map normalized by the reference image dynamic (in %) ; (right) evolution of the norm of the displacement increment  $\mathbf{d}^{(k)}$  normalized by the running approximation of  $\mathbf{u}^{(k)}$ .

step size of 1 does not converge. On the other hand, we can see that, without changing the algorithm and its operators, it is possible to make the classical algorithm converge for angles larger than  $45^\circ$ . Obviously, this was made possible at the expense of a relatively large number of iterations (roughly speaking, the descent nature of the modified GN search direction being less and less pronounced).

We can also notice that, even in this situation, the standard Gauss-Newton algorithm converges effectively. Note, however, that the computational cost of a GN iteration is much greater than that of an iteration of modified GN, since at each iteration, the operator  $\hat{\mathbf{H}}_j^{(k)}$  must be assembled and factorized. As shown later (see Fig. 6), one iteration of the modified GN



is nearly 4.5 times cheaper than that of GN. The user faces a dilemma: the choice between an inexpensive but potentially unstable algorithm and a robust but more expensive algorithm. We will see in the following paragraph that the developed ICGN and FCGN algorithms represent alternatives, as cheap as modified GN and as efficient as GN.

*4.1.2. Forward and inverse compositional Gauss-Newton.* The images are now analyzed using the proposed forward and inverse compositional Gauss-Newton variants to FE-DIC. According to Figure 3 and 5, both algorithms behave in the same way as the Gauss-Newton. Indeed, the number of iterations required to converge is stable (again no more than 10 iterations), whatever the angle of rotation, even up to extreme rotations angles such as  $\theta = 180^\circ$ . Efficiency and robustness are effective, despite the fact that the composition is locally approximated by linearizing the displacement.

Once again, it is worth saying that nothing changes as compared to the modified GN algorithm (same constant operator, same right-hand side). Only the summation step  $\mathbf{u}^{(k+1)} \leftarrow \mathbf{u}^{(k)} + \mathbf{d}^{(k)}$  is replaced by a composition step  $\phi(x, \mathbf{u}^{(k+1)}) = \phi(x, \mathbf{u}^{(k)}) \circ \phi(x, \mathbf{d})$  (with possibly the inversion  $\phi(x, \mathbf{u}^{(k+1)}) = \phi(x, \mathbf{u}^{(k)}) \circ \phi(x, -\mathbf{d})^{-1}$  for ICGN). This very slight modification implies a minor implementation effort.

The extra computing cost is also very modest since there are only two matrix-vector products of size  $n/2$  to be carried out at each iteration. Only the two aforementioned linear operators (see section 3.2) must be computed before the first GN iteration. Figure 6 attempts to quantify this extra cost as a function of the number of iterations. It can be seen that in this case, the

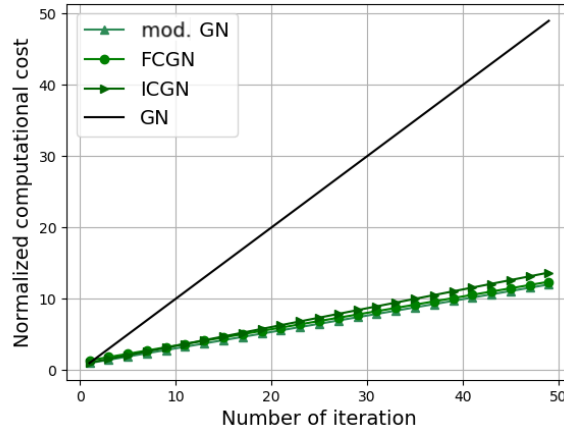


Figure 6. Evolution of the computational time taken by 4 different solvers: modified Gauss-Newton, Gauss-Newton, Forward and Inverse Compositional Gauss-Newton. The time is normalized by the time taken by one iteration of the Gauss-Newton.

modified GN calculation cost is divided by more than 4 compared to GN. The difference between modified GN, ICGN and FCGN is much smaller. Obviously, this result may depend on the test case (on mesh and image sizes for instance) and on the implementation, but it gives an idea of the possible savings. To plot this curve, a mesh of 6240 dofs was used with nearly 2.5 million integration points.

#### 4.2. Example 2: large strains

We have shown from a theoretical point of view, in Section 3.1, that the modified GN method may converge even outside the framework of small deformations to which it is sometimes restricted in the literature. We are therefore interested in this part in the case of large deformations to illustrate numerically that the convergence domain of the modified GN method is quite large. To put more emphasis on that point, we even consider large strains (that, of course, include large deformations following the terminology given in the introduction). We investigate two cases: a case of normal strain and a case of shear strain, both being homogeneous:

$$\underline{\underline{\varepsilon}}^{normal} = \begin{bmatrix} 1.8 & 0 \\ 0 & -0.4 \end{bmatrix} \quad \text{and} \quad \underline{\underline{\varepsilon}}^{shear} = \begin{bmatrix} 0.7 & -1.1 \\ -1.1 & 0.7 \end{bmatrix},$$

which corresponds to a biaxial strain state of 180% in one direction and -40% in the other. The corresponding images are given in Figure 7. These images are analyzed with the 4

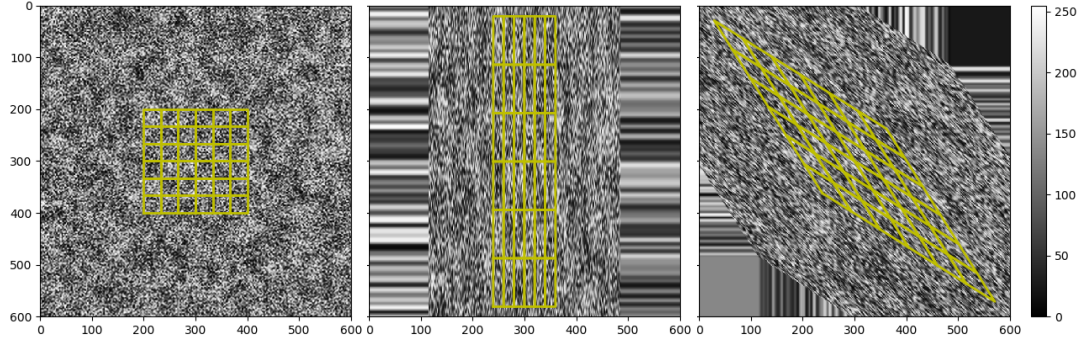


Figure 7. Images and mesh used for the large strain example: (left) reference image and initial mesh ; (middle) deformed state image of the tension case ; (right) deformed state image of the shear case.

discussed algorithms (modified GN, GN, FCGN and ICGN). A structured FE mesh made of 4-nodes bilinear elements of 32 pixels side is used. The displacement vector is initialized by the value of the displacement at the nodes of the mesh calculated for a strain level reduced by 5% than the one used to generate the images. Figures 8 and 9 show convergence, in terms of normalized correlation residual and of stagnation criterion during GN iterations. As theoretically predicted, each algorithm manages to reach convergence, including the modified GN algorithm. The number of iterations can vary from one method to another, but it is difficult to conclude on the superiority of one algorithm over the others. And the conclusion is probably, again, problem dependent. Also here, the chosen step size is always set to 1. The use of a line-search strategy could certainly reverse the observable trend. What is important to conclude is that the modified GN algorithm that is widely used in FE-DIC, remains quite efficient, even outside the framework of small deformations and/or strains. Examples with heterogeneous (large) strain fields were also considered and the same results were observed.

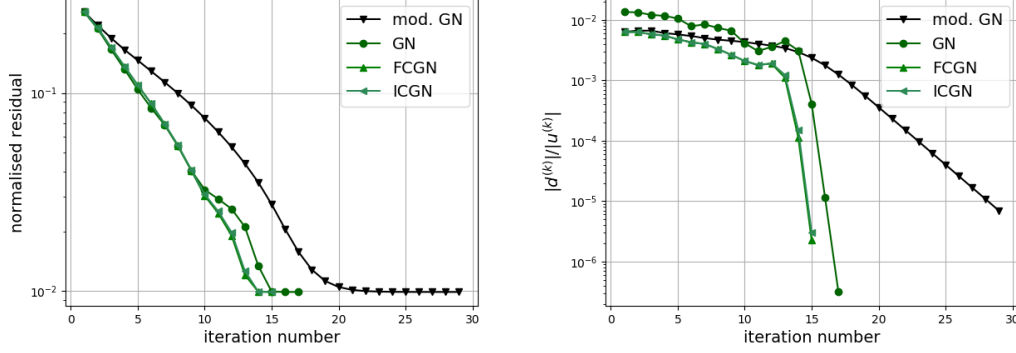


Figure 8. Convergence of the DIC solver for the large strain problem under tension, as a function of the number of GN iterations, for the five considered Gauss-Newton variants: (left) evolution of the standard deviation of the gray level residual map normalized by the reference image dynamic ; (right) evolution of the norm of the displacement increment  $\mathbf{d}^{(k)}$  normalized by the running approximation of  $\mathbf{u}^{(k)}$ .

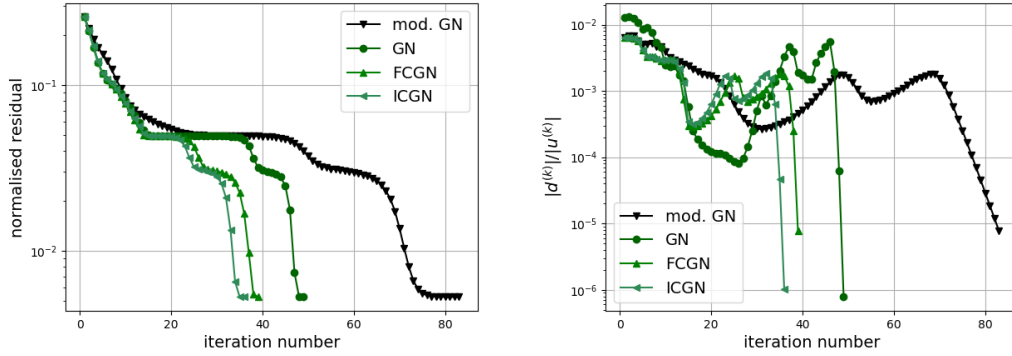


Figure 9. Convergence of the DIC solver for the large strain problem under shear, as a function of the number of GN iterations, for the five considered Gauss-Newton variants: (left) evolution of the standard deviation of the gray level residual map normalized by the reference image dynamic ; (right) evolution of the norm of the displacement increment  $\mathbf{d}^{(k)}$  normalized by the running approximation of  $\mathbf{u}^{(k)}$ .

## 5. CONCLUSION

An analysis of the most commonly used algorithm in FE-DIC was proposed in this paper. We have shown that it simply consists of a modified Gauss-Newton and that it does not always provide a descent direction. Several examples have been presented to show that depending on the case, the convergence speed can be (strongly) slowed down or the algorithm may even no longer converge at all. A condition on the DIC operators has been established to ensure to provide a descent direction. This condition is also interpreted as a condition on

the displacement field. Contrary to what is sometimes stated in the literature, this condition (positivity of the gradient of the transformation  $\mathbf{I} + \nabla \mathbf{U}^{(\infty)}$ ) is much broader than the case of small deformations (or small rotations). It was shown that, thanks to an appropriate choice of the step-size, the standard FE-DIC algorithm can converge up to angles of  $90^\circ$  and for very large strain levels (180% in the last examples).

When the unknown displacement field is such that the classic algorithm can no longer provide a descent direction, it should be stressed that even if the algorithm is initialized very close to the solution (even with the exact solution if it exists) it will diverge. In such a situation, updating the correlation operator at each iteration (like in the exact Gauss-Newton) is not the only way to reach convergence. Several alternatives were proposed in this paper. A first one was based on quasi Gauss-Newton, where the image gradients in the left and right-hand sides were different. Two other alternatives based on the family of Compositional Gauss-Newton algorithms were also proposed. At very little extra cost and using the same operators (and the same image gradients), it was possible to obtain fast algorithms with guaranteed convergence. This tends to confirm the approximation on images gradients in FE-DIC is not the problem.

The proposed new algorithms for Global(or FE)-DIC may prove useful in particular (a) for evolution problems, where one operator is assembled and factorized once for all GN iterations and for each time-step and (b) for stereo DIC where projector gradients may evolve during iterations. To go even further in algorithmic efficiency, these solvers could be coupled with HPC tools [23, 32, 38, 5, 46].

## REFERENCES

1. Simon Baker and Iain Matthews. Equivalence and efficiency of image alignment algorithms. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1090–1097, 2001.
2. Simon Baker and Iain Matthews. Lucas-Kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, Feb 2004.
3. 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, 2006.
4. Michael J. Black and Allan D. Jepson. Eigenttracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26(1):63–84, 1998.
5. R. Bouclier and J.-C. Passieux. A domain coupling method for finite element digital image correlation with mechanical regularization: Application to multiscale measurements and parallel computing. *International Journal for Numerical Methods in Engineering*, 111(2):123–143, july 2017.
6. H.A. Bruck, S.R. McNeill, M.A. Sutton, and W.H. Peters. Digital image correlation using Newton-Raphson method of partial differential correction. *Experimental Mechanics*, 29:261–267, 1989.
7. A. Buljac, C. Jailin, A. Mendoza, J. Neggers, T. Taillandier-Thomas, A. Bouterf, B. Smaniotto, F. Hild, and S. Roux. Digital volume correlation: Review of progress and challenges. *Experimental Mechanics*, 58(5):661–708, 2018.
8. P. Cheng, M. Sutton, H.W. Schreier, and S. R. McNeill. Full-field speckle pattern image correlation with b-spline deformation function. *Experimental mechanics*, 42(3):344–352, 2002.
9. P. Doumalin. *Microextensometrie locale par correlation d’images numeriques application aux etudes micromecaniques par microscopie electronique a balayage*. PhD thesis, Ecole Polytechnique, Palaiseau, France, 2000.
10. R. Fedele, L. Galantucci, and A. Ciani. Global 2D digital image correlation for motion estimation in a finite element framework: a variational formulation and a regularized, pyramidal, multi-grid implementation. *International Journal for Numerical Methods in Engineering*, 96(12):739–762, 2013.
11. R. Gras, H. Leclerc, S. Roux, S. Otin, J. Schneider, and J.-N. Périé. Identification of the out-of-plane shear modulus of a 3D woven composite. *Experimental Mechanics*, 53(5):719–730, 2013.
12. M. Grediac. Principe des travaux virtuels et identification/principe of virtual work and identification. *Comptes Rendus de l’Académie des Sciences*, 302:1–5, 1989.

13. A. Guery, F. Latourte, F. Hild, and S. Roux. Characterization of sem speckle pattern marking and imaging distortion by digital image correlation. *Measurement Science and Technology*, 25(1):015401, 2014.
14. F. Hild and S. Roux. Comparison of local and global approaches to digital image correlation. *Experimental Mechanics*, 52(9):1503–1519, 2012.
15. B.K.P. Horn and G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.
16. Clément Jailin, Ante Buljac, Amine Bouterf, François Hild, and Stéphane Roux. Fast 4D tensile test monitored via X-CT: Single projection based digital volume correlation dedicated to slender samples. *Journal of Strain Analysis for Engineering Design*, page In Press, 2018.
17. H. Kirchner and H. Niemann. Finite element method for determination of optical flow. *Pattern Recognition Letters*, 13(2):131 – 141, 1992.
18. H. Leclerc, J.-N. Périé, F. Hild, and S. Roux. Digital volume correlation: What are the limits to the spatial resolution? *Mechanics & Industry*, 13(6):361–371, november 2012.
19. H. Leclerc, J.-N. Périé, S. Roux, and F. Hild. Voxel-scale digital volume correlation. *Experimental Mechanics*, 51(4):479–490, 2011.
20. H. Leclerc, S. Roux, and F. Hild. Projection savings in CT-based digital volume correlation. *Experimental Mechanics*, 55(1):275–287, 2015.
21. David Lecompte, Arwen Smits, Hugo Sol, John Vantomme, and Danny Van Hemelrijck. Mixed numerical-experimental technique for orthotropic parameter identification using biaxial tensile tests on cruciform specimens. *International Journal of Solids and Structures*, 44(5):1643 – 1656, 2007.
22. B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of Imaging Understanding Workshop*, pages 121–130, 1981.
23. Michal Merta, Alena Vašatová, Václav Hapla, and David Horák. Parallel implementation of total-FETI DDM with application to medical image registration. In Jocelyne Erhel, Martin J. Gander, Laurence Halpern, Géraldine Pichot, Taoufik Sassi, and Olof Widlund, editors, *Domain Decomposition Methods in Science and Engineering XXI*, pages 917–925. Springer International Publishing, 2014.
24. J. Neggers, B. Blaysat, J. P. M. Hoefnagels, and M. G. D. Geers. On image gradients in digital image correlation. *International Journal for Numerical Methods in Engineering*, 105:243–260, 2016.
25. J. Neggers, F. Mathieu, F. Hild, S. Roux, and N. Swiergiel. Improving full-field identification using progressive model enrichments. *International Journal of Solids and Structures*, 118-119:213 – 223, 2017.
26. Jan Neggers, Olivier Allix, François Hild, and Stéphane Roux. Big data in experimental mechanics and model order reduction: Today’s challenges and tomorrow’s opportunities. *Archives of Computational Methods in Engineering*, 25(1):143–164, 2018.
27. J.-J. Orteu, D. Garcia, L. Robert, and F. Bugarin. A speckle-texture image generator. In *Speckle’06 International Conference*, volume 6341, page <http://dx.doi.org/10.1117/12.695280>, 2006.
28. B. Pan, K. Li, and W. Tong. Fast, robust and accurate digital image correlation calculation without redundant computations. *Experimental Mechanics*, 53(7):1277–1289, Sep 2013.
29. Bing Pan, Bo Wang, Dafang Wu, and Gilles Lubineau. An efficient and accurate 3d displacements tracking strategy for digital volume correlation. *Optics and Lasers in Engineering*, 58:126 – 135, 2014.
30. J.-C. Passieux, R. Bouclier, and J.-N. Périé. A space-time PGD-DIC algorithm: Application to 3d mode shapes measurements. *Experimental Mechanics*, 58(7):1195–1206, 2018.
31. J.-C. Passieux and J.-N. Périé. High resolution digital image correlation using Proper Generalized Decomposition: PGD-DIC. *International Journal for Numerical Methods in Engineering*, 92(6):531–550, 2012.
32. J.-C. Passieux, J.-N. Périé, and M. Salaün. A dual domain decomposition method for finite element digital image correlation. *International Journal for Numerical Methods in Engineering*, 102(10):1670–1682, 2015.
33. J.-E. Pierré, J.-C. Passieux, and J.-N. Périé. Finite element stereo digital image correlation: framework and mechanical regularization. *Experimental Mechanics*, 53(7):443–456, may 2017.
34. J.-E. Pierré, J.-C. Passieux, J.-N. Périé, F. Bugarin, and L. Robert. Unstructured finite element-based digital image correlation with enhanced management of quadrature and lens distortions. *Optics and Lasers in Engineering*, 77:44–53, february 2016.
35. J. Réthoré. A fully integrated noise robust strategy for the identification of constitutive laws from digital images. *Int. J. Numer. Meth. Eng.*, 84(6):631–660, 2010.
36. J. Réthoré, F. Hild, and S. Roux. Extended digital image correlation with crack shape optimization. *Int. J. Num. Meth. Eng.*, 73(2):248–272, 2008.
37. J. Réthoré, Muhibullah, T. Elguedj, M. Coret, P. Chaudet, and A. Combescure. Robust identification of elasto-plastic constitutive law parameters from digital image using 3D kinematics. *Int. J. Solids and Structures*, 50:73–85, 2013.
38. A. Ronovsky and A. Vasatova. Elastic image registration based on domain decomposition with mesh adaptation. *Mathematical Analysis and Numerical Mathematics*, 15(2):322–330, 2017.
39. J. Serra, J.-E. Pierré, J.-C. Passieux, J.-N. Périé, C. Bouvet, and B. Castanié. Validation and modeling of aeronautical composite structures subjected to combined loadings: the VERTEX project. part 1:

- Experimental setup, FE-DIC instrumentation and procedures. *Composite Structures*, 179:224–244, october 2017.
40. S.A. Stanier, J. Blaber, W.A. Take, and D.J. White. Improved image-based deformation measurement for geotechnical applications. *Canadian Geotechnical Journal*, 53(5):727–739, 2016.
  41. Y. Sun, J.H.L. Pang, C. Khuen Wong, and F. Su. Finite element formulation for a digital image correlation method. *Applied Optics*, 44(34):7357–7363, 2005.
  42. M.A. Sutton, W.J. Wolters, W.H. Peters, W.F. Ranson, and S.R. McNeill. Determination of displacements using an improved digital correlation method. *Image and Vision Computing*, 1(3):133–139, 1983.
  43. Javier Sánchez. The Inverse Compositional Algorithm for Parametric Registration. *Image Processing On Line*, 6:212–232, 2016.
  44. W. Tong. Formulation of lucas–kanade digital image correlation algorithms for non-contact deformation measurements: A review. *strain*, 49:313–334, 2013.
  45. J. van Beeck, J. Neggers, P. J. G. Schreurs, J. P. M. Hoefnagels, and M. G. D. Geers. Quantification of three-dimensional surface deformation using global digital image correlation. *Experimental Mechanics*, 54(4):557–570, 2014.
  46. Tianyi Wang and Qian Kema. Parallel computing in experimental mechanics and optical measurement: A review (ii). *Optics and Lasers in Engineering*, 104:181 – 191, 2018. Optical Tools for Metrology, Imaging and Diagnostics.
  47. L. Wittevrongel, P. Lava, S.V. Lomov, and D. Debruyne. A self adaptive global digital image correlation algorithm. *Experimental Mechanics*, 55(2):361–378, 2015.