

A posteriori performance-based comparison of three new path-following constraints for damage analysis of quasi-brittle materials

Peer-reviewed author version

FAYEZIOGHANI, Amir; VANDOREN, Bram & Sluys, L.J. (2019) A posteriori performance-based comparison of three new path-following constraints for damage analysis of quasi-brittle materials. In: Computer methods in applied mechanics and engineering, 346, p. 746-768..

DOI: 10.1016/j.cma.2018.09.014

Handle: <http://hdl.handle.net/1942/27175>

A posteriori performance-based comparison of three new path-following constraints for damage analysis of quasi-brittle materials

A. Fayezioghani^a, B. Vandoren^{a,*}, L. J. Sluys^b

^a*Hasselt University, Faculty of Engineering Technology, Hasselt, Belgium*

^b*Delft University of Technology, Faculty of Civil Engineering and Geosciences, Delft, The Netherlands*

Abstract

Using a path-following algorithm to analyze a quasi-static nonlinear structural problem involves selecting an appropriate constraint function. This function should improve the desired performance targets of the path-following algorithm such as robustness, speed, accuracy, and smoothness. In order to be able to draw a fair objective selection of a constraint function, it is necessary to collect adequate constraint equations as well as to define the performance of nonlinear methods. In this paper, three new path-following constraints applicable for damage analysis of quasi-brittle materials are proposed. Additionally, performance criteria and their numerical measures for a posteriori assessment of robustness, smoothness, accuracy, and speed of solving nonlinear problems by a path-following method are proposed. Based on the proposed criteria, the performance of the three new constraints and two existing ones is compared for two example problems. As a result, the performance measures are shown to possess an ability to clearly explore the strengths of each constraint. They establish a firm basis for the assessment of not only path-following methods but also other methods for solving nonlinear structural problems.

Keywords: performance criterion, path-following method, quasi-brittle materials, snap-back

1. Introduction

When solving an under-determined nonlinear system of equations (e.g. a quasi-static nonlinear structural problem) and the path to reach a final solution is of significance, *path-following*

*Corresponding author.

E-mail address: bram.vandoren@uhasselt.be (Bram Vandoren).

methods are powerful and widely used numerical tools. Essentially, they add an extra equation, called a *constraint function*, to the system of equilibrium equations. Different constraint functions have been employed by authors to mainly improve the robustness in convergence of searching a solution. However, few works have focused on the speed and quality of a solution by the path-following method which uses a specific constraint function. Thus, there is a need for defining criteria to measure the performance of a solution method as well as for constructing enhanced constraint functions. In this paper, we therefore propose and discuss performance criteria, their measures, and several objective parametric measures to be applied in the assessment of the performance of nonlinear solvers. Additionally, we suggest three new constraint functions and compare them with two existing ones in two example problems of the damage analysis of quasi-brittle materials.

The use of a constraint function in a structural problem is traced back to the works by Wempner [1] and Riks [2]. They used a constant incremental length of the solution path to control the evolution of the load factor. Shortly afterwards, Crisfield [3] proposed a modification of Riks' procedure which was appropriate for implementation in the finite element method (FEM). However, the task of choosing an appropriate solution root when using a spherical or cylindrical arc-length as a constraint function was still an issue. To address the issue, several solutions have been suggested. For example, Ramm [4] substituted the dot product of the tangent vector to the incremental vector of the solution path for Rik's arc-length equation; Schweizerhof [5] formulated a consistent linearization of the problem and Forde and Stierner [6] utilized it to present a class of constraints that provided only one root; Hellweg and Crisfield [7] showed that a smaller angle between the consecutive tangents of the solution path is a robust criterion for root selection; and, Feng et al. [8] and de Souza Neto and Feng [9] employed a criterion in the predictor phase of the solution procedure that avoided misleading roots.

The constraint functions which are fully or partially composed of the displacements (like those in the outlined works) are effective especially where encountering large displacements. However, for the nonlinearities which are not strongly reflected in the displacements (e.g. those caused by nonlinear material behavior), inclusion of the responses like strains in the construction of the constraint function could be very effective (see [10–12]). Moreover, tracing the solution path by a constraint which controls a critical zone enhances the robustness of a path-following method (see [10–15]). For instance, May and Duan [13] suggested controlling the relative dis-

placements inside the elements which are partially or fully damaged; Lorentz and Badel [15] presumed that the maximum of elastic predictors is an effective choice of a critical region. However, controlling either an adaptive or a fixed region needs carefully pinpointing the region, which is a difficult task when there are complex nonlinearities. Material nonlinearity was also efficiently dealt with by directly (see [16–19]) or indirectly (see [12, 15]) following a path with the dissipation of energy. For example, Gutiérrez [16] proposed that the rate of change of the dissipated energy should control the solution. However, this constraint function is not able to follow non-dissipative paths. To overcome this shortcoming, for instance, Pohl et al. [12] suggested employing a parameter of a damage model which could direct the solution to both dissipative and non-dissipative paths and May et al. [18] used a switch between two constraint functions controlling the rate of dissipated energy and internal energy. A recent review on the path-following methods applied in computational mechanics can be found in the works by Ritto-Corrêa and Camotim [20] and Stanić et al. [21].

Constraint functions, such as the ones proposed in Section 3, should be designed in a way that strengthens the method’s *performance criteria* such as the speed of finding a complete solution path, the accuracy of computations, the smoothness of the obtained path, or the convergence sensitivity to the parameters of a method. For quantitative comparison purposes, we must define the numerical measure of each criterion as well as objective combinations of measures, as outlined in Section 4.

2. Basics of the problem and the nonlinear solution method

In this section, we explain the basics of our problem and the path-following method which are used throughout the text.

2.1. Path-following method

In a quasi-static structural problem which is discretized by finite elements and parametrized for external loads by a load factor λ , we get a nonlinear algebraic system of equations

$$\mathbf{r}(\mathbf{a}, \lambda) = \mathbf{f}_{\text{int}}(\mathbf{a}, \lambda) - \mathbf{f}_{\text{ext}}(\mathbf{a}, \lambda) = 0 \quad (1)$$

where \mathbf{r} is the vector of unbalanced or residual forces, \mathbf{f}_{int} and \mathbf{f}_{ext} are the vectors of internal and external forces respectively, \mathbf{a} is the displacement vector, and λ is the load factor. To solve (1),

a constraint function is needed in order to control the change of the load factor. It usually takes the form

$$g(\mathbf{a}, \lambda, \eta) = h(\mathbf{a}, \lambda) - \eta = 0 \quad (2)$$

where g is the (total) constraint function, h is the (total) constraint kernel, and η is the total independent scalar step-length parameter (or tolerance). The function h is called the total constraint kernel because its increase by η controls finding the solution points. In a sequential process of searching for responses in an incremental-iterative method, we presumed (3) for the decomposition of \mathbf{a}_{n+1} and λ_{n+1} :

$$\mathbf{a}_{n+1} = \mathbf{a}_n + \delta\mathbf{a}_{n+1}, \quad \lambda_{n+1} = \lambda_n + \delta\lambda_{n+1} \quad (3)$$

where $\delta\mathbf{a}_{n+1}$ and $\delta\lambda_{n+1}$ are the variations of \mathbf{a} and λ in step (or increment) $n + 1$, respectively. In addition, the system composed of equations (1) and (2) is reformulated to have functions of $\delta\mathbf{a}_{n+1}$ and $\delta\lambda_{n+1}$ (instead of functions of \mathbf{a}_{n+1} and λ_{n+1}) for two reasons: a) it is more convenient to explicitly show the evolution of the system which is going to be solved in each step and b) (incremental) constraint functions which are functions of the variation of the displacements and the load factor are usually proposed and utilized:

$$\begin{bmatrix} \mathbf{r}_{n+1}(\delta\mathbf{a}_{n+1}, \delta\lambda_{n+1}) \\ g_{n+1}(\delta\mathbf{a}_{n+1}, \delta\lambda_{n+1}, \delta\eta_{n+1}) \end{bmatrix} = 0 \quad (4)$$

where $\mathbf{r}_{n+1}(\delta\mathbf{a}_{n+1}, \delta\lambda_{n+1}) = \mathbf{r}(\mathbf{a}_{n+1}, \lambda_{n+1})$ and $g_{n+1}(\delta\mathbf{a}_{n+1}, \delta\lambda_{n+1}, \delta\eta_{n+1}) = g(\mathbf{a}_{n+1}, \lambda_{n+1}, \eta_{n+1})$ are the residual force vector and the (incremental) constraint function in step $n + 1$, respectively. The (incremental) constraint function is equal to $h_{n+1}(\delta\mathbf{a}_{n+1}, \delta\lambda_{n+1}) - \delta\eta_{n+1}$ where h_{n+1} and $\delta\eta_{n+1}$ are the (incremental) constraint kernel and step-length, respectively. It should be noted that \mathbf{r}_{n+1} , g_{n+1} , and h_{n+1} are different functions from \mathbf{r} , g , and h . The Newton-Raphson method is a good choice to solve (4) in step $n + 1$:

$$\begin{bmatrix} \mathbf{K}_t^k & \mathbf{V}^k \\ \mathbf{v}^k & w^k \end{bmatrix} \begin{bmatrix} \Delta\mathbf{a}_{n+1}^{k+1} \\ \Delta\lambda_{n+1}^{k+1} \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_{n+1}^k \\ g_{n+1}^k \end{bmatrix} \quad (5)$$

where the components of the Jacobian matrix of step $n + 1$ are defined by

$$\mathbf{K}_t^k = \frac{\partial \mathbf{r}_{n+1}}{\partial (\delta \mathbf{a}_{n+1}^k)^T}, \quad (6)$$

$$\mathbf{V}^k = \frac{\partial \mathbf{r}_{n+1}}{\partial (\delta \lambda_{n+1}^k)}, \quad (7)$$

$$\mathbf{v}^k = \frac{\partial g_{n+1}}{\partial (\delta \mathbf{a}_{n+1}^k)^T}, \quad (8)$$

$$w^k = \frac{\partial g_{n+1}}{\partial (\delta \lambda_{n+1}^k)}, \quad (9)$$

the difference vector of the solution is composed of

$$\Delta \mathbf{a}_{n+1}^{k+1} = \delta \mathbf{a}_{n+1}^{k+1} - \delta \mathbf{a}_{n+1}^k, \quad (10)$$

$$\Delta \lambda_{n+1}^{k+1} = \delta \lambda_{n+1}^{k+1} - \delta \lambda_{n+1}^k, \quad (11)$$

and the components of the right-hand side of (5) are equal to

$$\mathbf{r}_{n+1}^k = \mathbf{r}_{n+1}(\delta \mathbf{a}_{n+1}^k, \delta \lambda_{n+1}^k), \quad (12)$$

$$g_{n+1}^k = g_{n+1}(\delta \mathbf{a}_{n+1}^k, \delta \lambda_{n+1}^k), \quad (13)$$

in which superscript \square^k indicates the k^{th} iteration, and \mathbf{x}^T is the transpose of vector \mathbf{x} . The Jacobian matrix in (5) is unsymmetric in general. Hence, in practice, a bordering algorithm (e.g. see [22]) is employed to benefit from the symmetry of the tangential stiffness matrix \mathbf{K}_t by the following decomposition:

$$\Delta \mathbf{a}_{n+1}^{k+1} = \bar{\Delta} \mathbf{a}_{n+1}^{k+1} - \Delta \lambda_{n+1}^{k+1} \tilde{\Delta} \mathbf{a}_{n+1}^{k+1} \quad (14)$$

where $\bar{\Delta} \mathbf{a}_{n+1}^{k+1}$ and $\tilde{\Delta} \mathbf{a}_{n+1}^{k+1}$ are obtained by solving

$$\mathbf{K}_t^k \bar{\Delta} \mathbf{a}_{n+1}^{k+1} = -\mathbf{r}_{n+1}^k, \quad \mathbf{K}_t^k \tilde{\Delta} \mathbf{a}_{n+1}^{k+1} = \mathbf{V}^k \quad (15)$$

and $\Delta \lambda_{n+1}^{k+1}$ is calculated by substituting equation (14) in the second part of (5):

$$\Delta \lambda_{n+1}^{k+1} = -\frac{g_{n+1}^k + \mathbf{v}^k \bar{\Delta} \mathbf{a}_{n+1}^{k+1}}{w^k - \mathbf{v}^k \tilde{\Delta} \mathbf{a}_{n+1}^{k+1}} \quad (16)$$

2.2. Two existing constraint functions

As mentioned in the introduction, we have chosen two of the most common existing constraint functions to be compared with three new ones (proposed in Section 3). The first one is the well-known cylindrical arc-length control proposed by Crisfield [3]:

$$h_{n+1}(\delta \mathbf{a}_{n+1}, \delta \lambda_{n+1}) = \delta \mathbf{a}_{n+1}^T \delta \mathbf{a}_{n+1} \quad (17)$$

This constraint function is interpreted as a cylinder in the hyperspace of loading factor and displacements which is centered at the displacements of step n . Using this constraint function means intersecting the surface of the responses of (1) by the mentioned cylinder. This function could be used in problems with deformation dependent external forces and with any kind of nonlinearity. The second one is the energy release control function proposed by Gutiérrez [16]:

$$h_{n+1}(\delta \mathbf{a}_{n+1}, \delta \lambda_{n+1}) = \frac{1}{2}(\lambda_n \delta \mathbf{a}_{n+1}^T - \delta \lambda_{n+1} \mathbf{a}_n^T) \hat{\mathbf{f}}_{\text{ext}} \quad (18)$$

where h_{n+1} is the variation of the energy release in step $n + 1$, $\hat{\mathbf{f}}_{\text{ext}}$ is the unit vector of external forces when the external force is independent of the deformations (i.e. $\mathbf{f}_{\text{ext}} = \lambda \hat{\mathbf{f}}_{\text{ext}}$). This constraint is based on the second law of thermodynamics. It controls the problem solution by the energy release rate which is discretized by the forward Euler method. This means that it imposes the variation of the energy release of a step to be equal to the incremental step-length of the same step. Moreover, in its formulation, it is assumed that the external force only depends on the load factor, the problem is geometrically linear, and the behavior is linear elastic with progressive damage.

2.3. Damage model

An isotropic damage model is adopted to model the nonlinear behavior of interface elements in the example problems. The constitutive relation follows

$$\mathbf{t}_{\text{int}} = (1 - \omega) \mathbf{C}_{\text{int}} \mathbf{u}_{\text{int}} \quad (19)$$

where \mathbf{t}_{int} and \mathbf{u}_{int} are the traction vector and the relative displacement vector of the interface, respectively, \mathbf{C}_{int} is the constant stiffness matrix of an undamaged state of the interface, and ω is the scalar damage parameter of the isotropic damage model which takes a value between 0 (no damage) and 1 (fully damaged). An irreversible damage model requires ω to be defined as a function of a parameter which stores the largest damage state occurred in each material point. The damage history parameter κ plays the role of this parameter. Its value increases with an increase in the equivalent relative displacement u_{eq} , but stays constant as the equivalent relative displacement decreases; this is referred by the Karush-Kuhn-Tucker conditions:

$$u_{\text{eq}} - \kappa \leq 0, \quad \dot{\kappa} \geq 0, \quad (u_{\text{eq}} - \kappa) \dot{\kappa} = 0 \quad (20)$$

where $\dot{\kappa}$ is the rate of change of damage history parameter and $u_{\text{eq}} = u_{\text{eq}}(\mathbf{u}_{\text{int}})$ is an invariant measure of the total relative displacement. It should be noted that two other assumptions are

made for modeling damage in this paper: a) healing of the damaged points does not occur and b) the damage history parameter is a continuous function of time.

3. Three new constraint functions

In this section, three new constraint function are proposed and discussed. All of them possess the advantage of leading the solution path to a dissipative one and preventing global artificial unloading.

3.1. Angle control constraint

This function exploits an augmented angle which consists of elastic unloading angles in the hyperplane of load factor and displacements. It assumes that the external load is deformation independent and is applicable for any kinds of nonlinearity:

$$\theta(\mathbf{a}, \lambda) = \arctan \left(\sqrt{\frac{1}{m} \sum_{j=1}^m \left(\frac{\gamma \lambda}{\mathbf{a}_{d_j}} \right)^2} \right) \quad (21)$$

where \mathbf{a}_{d_j} is the vector of the displacements at the degrees of freedom with external loads, γ is a normalizing factor, and m is the number of degrees of freedom with external loads. By analogy with the reformulation in (4), the augmented angle reforms to

$$\theta_{n+1}(\delta \mathbf{a}_{n+1}, \delta \lambda_{n+1}) = \arctan \left(\sqrt{\frac{1}{m} \sum_{j=1}^m \left(\frac{\gamma(\lambda_n + \delta \lambda_{n+1})}{(\mathbf{a}_n)_{d_j} + (\delta \mathbf{a}_{n+1})_{d_j}} \right)^2} \right) \quad (22)$$

and the proposed constraint kernel is defined by

$$h_{n+1}(\delta \mathbf{a}_{n+1}, \delta \lambda_{n+1}) = \theta_n - \theta_{n+1}(\delta \mathbf{a}_{n+1}, \delta \lambda_{n+1}) = -\delta \theta_{n+1}(\delta \mathbf{a}_{n+1}, \delta \lambda_{n+1}) \quad (23)$$

where θ_n is the converged value of the angle in step n .

The idea of proposing this constraint function is to provide a simple robust solution which is easy to understand from a graphical point of view. Figure 1 illustrates the angle control constraint for a single point load. The angle starts from θ_0 for the initial elastic behavior and decreases in each step according to the constraint kernel (23). The angle control constraint prevents global artificial unloading in the material models with elastic unloading by imposing at least one degree of freedom to follow the progressive damage path. In other words, if we assume that all of the degrees of freedom with external loads follow an artificial unloading path in step $n + 1$, θ_{n+1} will be equal to θ_n and the angle control constraint will not vanish, which is contradictory to (2) for nonzero incremental step-lengths.

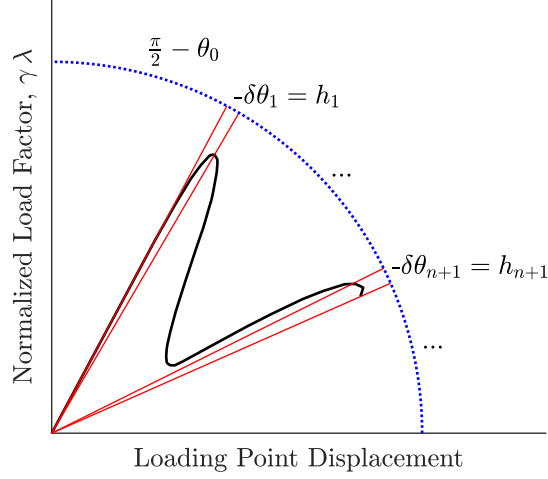


Figure 1: The decrease of the augmented angle in each step is shown by moving the red line clockwise.

3.2. κ control constraint

According to the definition of the damage history parameter and the assumptions made in Section 2 for modeling damage, there are two possibilities: a) δu_{eq} is non-positive which implies that $\delta\kappa$ must be zero and b) δu_{eq} is positive and it necessitates $\delta\kappa$ to be either positive or zero depending on the state of κ . Table 1 summarizes possible situations with respect to the variation in the damage history parameter, $\delta\kappa$. Based on Table 1, if at least one integration point has an increasing κ , global unloading is avoided. Thus, the integration of $\delta\kappa$ over the whole domain of the problem is proposed as a constraint kernel in (24):

$$h_{n+1}(\delta\mathbf{a}_{n+1}) = \int_{\Omega_n} \delta\kappa_{n+1} d\Omega_n \quad (24)$$

where $\delta\kappa_{n+1} = \kappa_{n+1} - \kappa_n$ is the variation in the history parameter in step $n+1$, $\kappa_{n+1} = \max(\kappa_n, u_{\text{eq}}(\mathbf{a}_n + \delta\mathbf{a}_{n+1}))$, κ_n is the converged value of damage history in step n , and Ω_n is the domain of the body at the end of step n which is normally assumed to remain constant for problems with small displacements.

There are two advantages of the proposed kernel (24) over similar existing ones like the adaptive strain control suggested by Pohl et al. [12] which controls the equivalent strain of an active integration point: 1) it does not require an additional procedure to find the active point in each step and 2) it does not impose an increase to the equivalent relative displacement of a specific integration point in each step, which is more accurate especially for large incremental

Table 1: The possible situations of loading/unloading during an analysis shows that $\delta\kappa > 0$ will avoid unloading.

Damage state	$\delta u_{\text{eq}} \leq 0 \Rightarrow \delta\kappa = 0$	$\delta u_{\text{eq}} > 0 \Rightarrow \delta\kappa > 0 \text{ OR } \delta\kappa = 0$
no damage $\kappa \in \{\kappa \omega(\kappa) = 0\}$	elastic unloading of undamaged material	elastic loading OR elastic loading
partially damaged $\kappa \in \{\kappa 0 < \omega(\kappa) < 1\}$	elastic unloading of partially damaged material	progressive damage loading OR elastic loading
fully damaged $\kappa \in \{\kappa \omega(\kappa) = 1\}$	elastic unloading of fully damaged material	fully damaged separating OR fully damaged separating

step-lengths.

3.3. κ - ω control constraint

This constraint follows the same idea as in κ control except that the contribution of the variation of the damage history parameter is emphasized less for damaged points. That is, the more damage in a point of material the less it is assumed to be influential for the overall damage progression. This constraint function reads

$$h_{n+1}(\delta \mathbf{a}_{n+1}) = \int_{\Omega_n} (1 - \omega_{n+1}) \delta \kappa_{n+1} d\Omega_n \quad (25)$$

where $\omega_{n+1} = \omega(\kappa_n + \delta \kappa_{n+1})$ is the damage parameter.

The constraint kernel (25) behaves more effectively than (24) when a fixed incremental step-length is implemented and the number of fully damaged points is growing. In the mentioned case, large magnitudes of κ usually happen in the fully damaged points, which causes slow damage progression in the rest of the points when using a fixed value of incremental step-length in (24). Therefore, in contrast to the mentioned constraint in [12], utilizing (25) automatically prevents contributions of points which are fully damaged. The parameters κ and ω are embedded entities of a damage model which offer no extra computations as they are calculated during the assembly of the system stiffness matrix.

The constraint functions of κ , κ - ω , and the energy release control conceptually express the second law of thermodynamics in accordance with the entropy rate being non-negative. While the energy release control causes the solver to follow only dissipative paths, by utilizing κ and κ - ω control, both dissipative and non-dissipative paths of a loading situation are gained. For

instance, κ and κ - ω control are able to effectively find points on the elastic paths while the energy release control jumps from a point before to a point after the elastic paths. Furthermore, these two constraint functions have the advantage of being directly utilized for the problems with large displacements, large deformations, and deformation dependent external forces.

4. Performance criteria

In this section, we propose four performance criteria (robustness, speed, accuracy, and smoothness) in order to assess the solutions found by path-following algorithms. Moreover, we propose their measures which provide a posteriori information useful for fair comparisons made in the numerical examples in Section 5.

4.1. Robustness

Robustness is a generic term for the demonstration of the stability of a system with regard to the changes of its parameters. The term ‘system’, here, refers to the union of the physical system under analysis and the method to solve it. In the context of nonlinear solvers, it may be defined as the ability to reach to a desired state of the problem. For an incremental-iterative method employed in a structural engineering problem, we suggest three *robustness* measures. The first and the most important issue is the convergence in finding the solution in all incremental steps. It is common that if a method cannot find a finite solution in an increment, the searching process of that increment will restart with different parameters (e.g. different step-length, initial guess of the solution vector, or searching method). This, hereafter, is called the ‘*restarting strategy*’. The total number of restartings needed by using a method is calculated as follows:

$$N^r = \sum_n N_n^r \quad (26)$$

where N^r is the total number of restartings in an analysis and N_n^r is the number of restartings needed in step n . Finally, the quantitative robustness measure, R , is suggested to be:

$$R = \frac{1}{N^r + 1} \quad (27)$$

According to (27), the robustness measure of a method with no need for a restarting is equal to 1, with one restarting is equal to 1/2 and so on. In the example problems of Section 5, if a method is not able to converge in an increment by a limited number of restartings, the method is

assumed to be unacceptable. Otherwise, the method is acceptable and measure (27) is considered as one of the quantitative measures of the performance assessment. The restarting strategy and its parameters are specified in Section 5. The second measure is the solution path with no global artificial unloading, which is usually desired in the damage analysis of structures. In case there is any global artificial unloading, the solution, again, is assumed to be unacceptable. And, being able to deal with problems when their behavior changes from an elastic to an inelastic state, or vice versa is the third measure of robustness.

4.2. *Speed*

The inverse of the computational time which is specifically spent when employing a method to find a complete solution path represents a helpful measure of speed. However, the computational time would be a correct measure if one can calculate the actual CPU time in a way that is independent of background processes (e.g. processes necessary for an operating system) on a computer and effectively considers the architecture of the computer units. To keep the measure as effective and, at the same time, as simple as possible, we suggest the inverse of the total number of iterations needed for solving the problem by an incremental-iterative method as the *speed* measure.

$$N^i = \sum_n N_n^i \quad (28)$$

where N^i is the total number of iterations and N_n^i is the number of iterations needed in step n . As mentioned before, we only accept the obtained solution paths which reach a desired state of the problem. If we name it as a complete solution path, the unit of the speed measure will be ‘complete solution path per iteration’.

4.3. *Accuracy*

In general, accuracy is defined as the closeness of an entity to a reference one. For example, in a structural analysis, the law of balance of forces or conservation of energy is accepted to provide a measure of closeness. In other words, they provide equations which should vanish in a perfectly accurate case. In a numerical analysis, the equations have residuals that might be defined as a measure of error (or a measure of not being accurate). It should be noted that, in the context of incremental-iterative methods of solving a finite element discretized system of equations, the error is calculated from the residual of the system of equations and not from

the residual of the governing differential equation(s). However, no accuracy measure is taken into account as a performance measure in the example problems of Section 5, because we have utilized inequalities (42) and (43) as the convergence criteria in each increment of the analysis to be sure that enough accuracy is obtained for the analysis.

4.4. Smoothness

In mathematics, an m^{th} order smoothness property of a function is defined as the quality of having continuous derivatives up to the order m . This property is of interest in a numerical analysis because it assures a better linear approximation between every two consecutive solution points. For instance, it brings about substantial improvements in tracing the solution regions with sharp variations. In fact, the more smooth, the more accurate is a numerical curve. Since a numerical analysis searches for points on the exact solution curve and commonly assumes linear variation between the points, we classify the smoothness as a separate criterion rather than accuracy.

The concept of smoothness in (artificial) time discretization of quasi-static problems is analogous to that in spatial discretization. The latter is a well-known issue of finite element methods which is typically called meshing. In general, a finite element mesh should provide as uniformly smooth responses of the analysis steps as possible.

Although it is observed by numerical experience that achieving more smoothness requires more time (and effort), the exact relation between smoothness and speed varies from method to method. Thus, it is necessary to define a measure of smoothness in order to more systematically assess the obtained solution performance which is normally desired to fall within a reasonable performance range away from extremes (e.g. extremely fast nonsmooth or extremely slow smooth solution curves). Figure 2 shows an example of a fast nonsmooth solution curve which has jumped over curved parts of the exact solution curve.

The sum of norms of the displacement vector variations of all analysis steps is suggested as a measure of distance:

$$D = \sum_n \|\delta \mathbf{a}_n\| \quad (29)$$

where $\|\mathbf{x}\|$ is the L_2 norm of vector \mathbf{x} . Therefore, we can conclude that if a calculated D of an analysis is equal to a reference distance (e.g. a distance calculated by a very comprehensive analysis with very small incremental step-lengths), the solution will be perfectly smooth. We

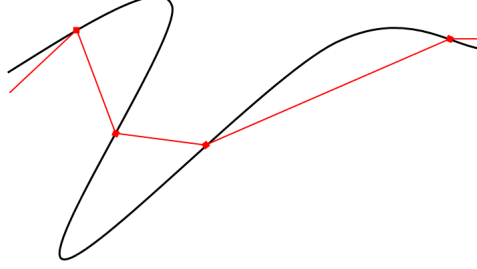


Figure 2: A nonsmooth numerical solution found by a very fast method (red line) may jump over the curved parts of the exact solution curve (black line).

suggest the *smoothness* measure S such that it has a peak at the reference distance, D_{ref} , and decreases as D moves away from D_{ref} :

$$S = \begin{cases} \frac{D}{D_{\text{ref}}} & , 0 \leq D < D_{\text{ref}} \\ \frac{D_{\text{ref}}}{D} & , D \geq D_{\text{ref}} \end{cases} \quad (30)$$

where D_{ref} is the value of the reference distance. The reason for defining the smoothness measure (30) by two functions is that, depending on what method to use, D goes to D_{ref} either from above or from below when using smaller time discretization intervals. An example is the ‘sawtooth’ curve obtained by conducting a typical “sequentially linear analysis” which approaches to the exact solution from above by using smaller intervals in the definition of stress-strain constitutive law. However, according to the definition of the smoothness measure, its value cannot be greater than 1.0 regardless of the method and its parameters.

4.5. Objective performance measures

In order to perform a fair comparison between the performances of different nonlinear solvers, we recommend satisfaction of the qualitative measures of robustness and then combining numerical performance measures in a single measure, called an *objective performance measure* P , in a way that it accords with the following conditions:

- (a) To be able to effectively combine the measures in a single P , *normalized measures* X_i , should be utilized. That is, all of the measures are firstly scaled to become dimensionless and to be in the same range of values (say between 0 and 1).
- (b) P should monotonically increase as each X_i increases.

- (c) In order to explicitly specify the relative importance of the measures, each X_i should be accompanied by an *importance factor*, α_i .
- (d) The influence of each X_i in P should become greater when its corresponding importance factor increases.

In accordance with the mentioned conditions, four objective performance measures are proposed in equations (31) to (34). These objective measures are composed of basic mathematical operators and functions. The first one has the simplest form which is a linear combinations of the normalized measures:

$$P^I = \sum_{i=1}^m Y_i, \quad Y_i = \alpha_i X_i \quad (31)$$

The second one is a linear combination of the exponentials of the normalized measures which gives more emphasis on larger values of X_i :

$$P^{II} = \sum_{i=1}^m Y_i, \quad Y_i = \alpha_i \exp X_i \quad (32)$$

By assuming the importance on the powers of X_i , we construct the third one:

$$P^{III} = \prod_{i=1}^m Y_i, \quad Y_i = X_i^{\alpha_i} \quad (33)$$

Finally, the fourth one proposes a function of X_i which gives a sharp drop for very small values of X_i and assumes the importance on the powers of this function:

$$P^{IV} = \prod_{i=1}^m Y_i, \quad Y_i = \left| \frac{1}{-1 + \ln X_i} \right|^{\alpha_i} \quad (34)$$

In the equations of the objective measures, X_i is the i^{th} normalized measure which is assumed to be scaled between 0 and 1, Y_i is the i^{th} constituent of P which solely relates to X_i , α_i is the importance factor for X_i and should be positive, m is the total number of performance measures, $\exp x$ and $\ln x$ are the exponential and natural logarithm of x , respectively, $|x|$ is the absolute value of x , and the operators \sum and \prod indicate the sum and scalar product of their operands, respectively.

Now, we check the above conditions (a) to (d). By the imposed restriction to X_i (i.e. $X_i \in (0, 1)$), the condition (a) is satisfied for all P . According to the condition (b), the derivative of P

with respect to X_p should be positive for all p :

$$\frac{\partial P^I}{\partial X_p} = \alpha_p > 0 \quad (35)$$

$$\frac{\partial P^{II}}{\partial X_p} = \alpha_p \exp X_p > 0 \quad (36)$$

$$\frac{\partial P^{III}}{\partial X_p} = \frac{\alpha_p}{X_p} \prod_{i=1}^m X_i^{\alpha_i} > 0 \quad (37)$$

$$\frac{\partial P^{IV}}{\partial X_p} = -\frac{\alpha_p}{X_p} \frac{-1 + \ln X_p}{|-1 + \ln X_p|^4} \prod_{i=1}^m \left| \frac{1}{-1 + \ln X_i} \right|^{\alpha_i} > 0, \quad (38)$$

which is shown to be satisfied for all of the objective performance measures. The condition (c) is also embedded in the definition of Y_i of each P . The mathematical description of the condition (d) for P^I and P^{II} differs from P^{III} and P^{IV} as they use different operators. The condition is formulated in the following:

- For all p , Y_p should be monotonic with respect to α_p . And, its absolute value should go to infinity when α_p goes to infinity and to zero when α_p goes to zero:

$$\lim_{\alpha_p \rightarrow +\infty} |Y_p| = +\infty, \quad \lim_{\alpha_p \rightarrow 0} |Y_p| = 0 \quad (39)$$

- For all p , Y_p should be monotonic with respect to α_p . And, its absolute value should go either to infinity or to zero when α_p goes to infinity and to 1 when α_p goes to zero:

$$\lim_{\alpha_p \rightarrow +\infty} |Y_p| = +\infty \text{ or } 0, \quad \lim_{\alpha_p \rightarrow 0} |Y_p| = 1 \quad (40)$$

All of the objective performance measures obviously satisfy the above conditions.

It is now possible to sort different sets of X_i (which could be results of employing different numerical methods) based on each objective performance measure. However, it is more efficient to firstly identify both the number of and the form of *effective importance factors* and then draw a comparison. The number of effective importance factors is defined as the number of importance factors which could affect the sequence of sorted measure sets. For instance, assume that the performance of set s is greater than set r , i.e. $P_s > P_r$. Thus, for P^I ,

$$P_s^I > P_r^I \Rightarrow \sum_{i=1}^m \alpha_i \cdot X_i^s > \sum_{i=1}^m \alpha_i \cdot X_i^r \quad (41)$$

After dividing both sides of (41) by the importance factor α_j , P_s^I remains greater than P_r^I while the *importance ratios* α_i/α_j ($i \neq j$) affect the performance values. Thus, there exist $m - 1$ effective importance factors in the form of α_i/α_j for P^I and for the mentioned comparison. Similar

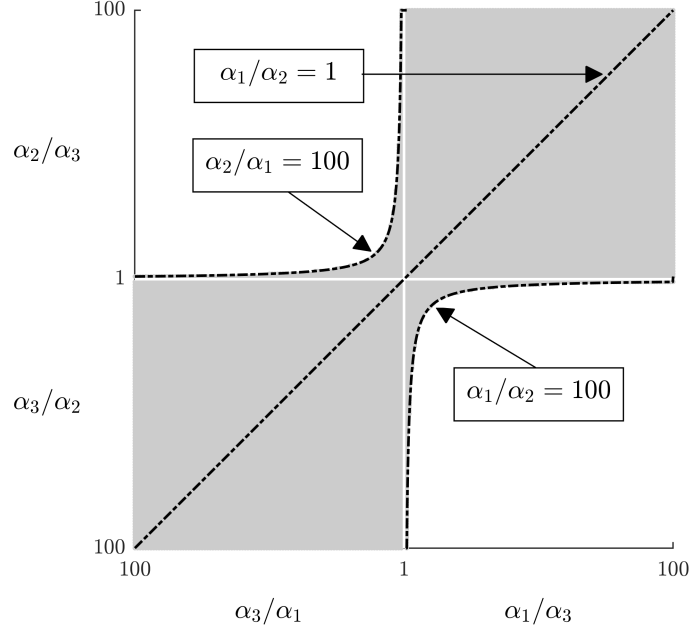


Figure 3: The gray area is the domain for which each objective measure is computed. The vertical and horizontal white lines inside of the gray area indicate 1 for α_i/α_j where $i, j \in \{1, 3\}$ and where $i, j \in \{2, 3\}$, respectively. The dashed-line hyperbolas restrict the domain between 1 and 100 for α_i/α_j where $i, j \in \{1, 2\}$.

procedures are adopted to find the number of and the form of effective importance factors for the other objective measures. They all give the same number and form as for P^I .

In the performance comparisons performed in Section 5, we use three numerical performance measures (i.e. those for robustness, speed, and smoothness) which lead to two effective importance ratios. To explore the space of each objective performance measure withing a practical range, the importance ratios α_i/α_j are kept between 1 and 100. Figure 3 illustrates a uniform domain of exploration of an objective performance which is utilized in Section 5. It is designed in Figure 3 that α_1/α_3 decreases from right to left up to 1. Going further to left also decreases α_1/α_3 , but for the sake of uniformity of illustration, the increase in α_3/α_1 is presented. The figure is designed in the same way for α_2/α_3 and for moving downward. The last ratio, α_1/α_2 , decreases from the lower hyperbola (as it is indicated in the figure) up to the bisector which

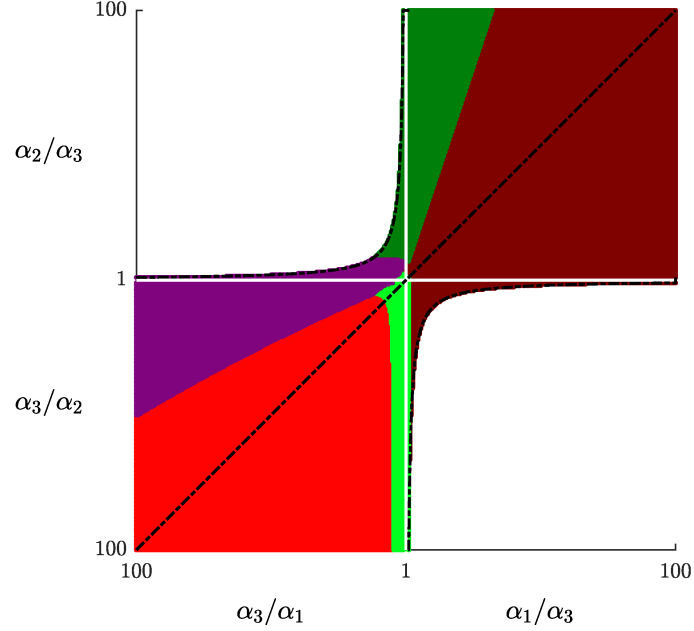


Figure 4: The hypothetical example shows the areas of the best performing methods by different colors.

presents $\alpha_1/\alpha_2 = 1$. Apparently, further movement will increase α_2/α_1 from 1 at the bisector to 100 at the upper hyperbola.

To clarify how Figure 3 works, a hypothetical example is presented in this paragraph. Assume that three numerical measures (X_1 to X_3) are considered to compare performances of six numerical methods of solving a problem. The best performing method will be the method with the largest objective performance measure, P . By calculating the objective measures of the methods in the domain of effective importance ratios (which is presented in Figure 3 in gray), we end up with Figure 4. In the figure, each area filled by a specific color represents that its corresponding method has been the best among the others in that area of importance ratios. For instance, if we desire to assign the importance of X_1 and X_2 to be 50 and 20 times of the importance of X_3 , respectively, the method with brown color will perform best. Note that there are only five colors shown in the figure, which means that one of the methods has not been the best in any of the areas. It should be noted that the values of the effective importance ratios are selected in accordance with the significance that a user of a nonlinear solver assigns to each of the performance measures.

5. Numerical examples

The performance measures mentioned in Section 4 are influenced by three major factors: 1) the problem definition, 2) the employed analysis procedure and its parameters, and 3) the hardware, software, and programming style used for the analysis. To assess the influence of the factors on the performance of an analysis, it is simpler to compare the results of variations of only one factor at a time and keep the others fixed. In this section, we draw comparisons between the constraint functions of a portfolio composed of the proposed functions and two existing ones (described in Section 2) in two example problems by means of the proposed objective performance measures. A constraint function is the keystone of the path-following method which itself is the core block of a quasi-static analysis procedure. The other contributing factors are determined to be fixed for each example problem as in the following:

- It is assumed that the external force is deformation independent. In other words, $\mathbf{f}_{\text{ext}} = \lambda \hat{\mathbf{f}}_{\text{ext}}$, where $\hat{\mathbf{f}}_{\text{ext}}$ is the unit vector of external loads that remains constant during an analysis.
- All analyses for each example are carried out by the path-following method together with a Newton-Raphson scheme. The analyses differ over the employed constraint function and the optimal value of iterations per step, N_{opt}^i (see Table 2).
- The analyses are initially started by the force control on the body at rest. They switch to one of the mentioned constraint functions when passing a threshold for the dissipated energy increment (defined in [16]) and continue to reach a desired state of the body. Both specifying the desired state and assigning a value for the threshold are reported for each example separately.
- Convergence of the responses in each step is sought by satisfying both of the following stopping criteria:

$$\frac{\max |\mathbf{r}_{n+1}^k|}{\max |\lambda_{n+1}^k \hat{\mathbf{f}}_{\text{ext}}|} \leq \text{TOL} \quad (42)$$

$$\frac{|g_{n+1}^k|}{\delta \eta_{n+1}} \leq \text{TOL} \quad (43)$$

where $\max |\mathbf{x}|$ is the maximum absolute value of vector \mathbf{x} and TOL is the error tolerance chosen to be equal to 0.001. Practically, one of the above inequalities will be satisfied with

more number of iterations than the other one. In other words, one of them controls the stop of the iterative loop of the increment. There is no prior knowledge at the start of each increment that which one controls the stop of that increment. If, in an increment, inequality (43) controls the increment, it means that inequality (42) has been satisfied in previous iterations. In this case, $\max |\mathbf{r}_{n+1}^k|$ becomes smaller than when (42) is satisfied for the first time and this difference directly depends on the constraint function. And, if inequality (42) controls the increment, $\max |\mathbf{r}_{n+1}^k|$ directly depends on the last iteration. Additionally, both of the convergence criteria are relative ones, which means that the absolute tolerance is equal to TOL multiplied by a value related to the incremental step-length (i.e. the increment of η) and the constraint function. Therefore, even in case (42) controls the increment, $\max |\mathbf{r}_{n+1}^k|$ will relate to TOL, the step-length, and the constraint function.

- To keep the incremental step-length in an appropriate range, we use the adaptation rule proposed by Crisfield [3] and Bellini [23]:

$$\delta\eta_{n+1} = \delta\eta_n \left(\frac{N_{\text{opt}}^i}{N_n^i} \right)^\beta \quad (44)$$

where β typically ranges between 0.25 and 1.0 [24] and is chosen to be 1.0 in the examples. It should be noted that selecting tighter tolerances (i.e. smaller values of TOL) and/or faster convergence parameters (i.e. smaller values of N_{opt}^i) yields smaller incremental step-lengths which usually lead to more analysis time and more smoothness.

- A simple adaptation rule is also used for the initial guess of each step after calculation of the incremental step-length:

$$\begin{bmatrix} \delta\mathbf{a}_{n+1}^0 \\ \delta\lambda_{n+1}^0 \end{bmatrix} = \frac{\delta\eta_{n+1}}{\delta\eta_n} \begin{bmatrix} \delta\mathbf{a}_n \\ \delta\lambda_n \end{bmatrix} \quad (45)$$

where $\delta\mathbf{a}_n$ and $\delta\lambda_n$ are the corresponding converged responses of step n .

- In case of non-convergence after 15 iterations in an analysis step, a restarting strategy is adopted. Each time a non-convergence occurs in an analysis step, the strategy adds 15 to N_n^i and restarts the analysis of the step from adapting the step-length and the initial guess. For example, assume that a method converges by 3 iterations in step n ; thus, $N_n^i = 3$ and the method goes to the next step; then, it starts step $n + 1$ with $N_n^i = 3$; if it cannot converge

within 15 iterations for the first time, the method restarts the step with $N_n^i = 15 \times 1 = 15$; if it, again, is not able to converge by less than or equal to 15 iterations for the second time, it restarts the step with $N_n^i = 15 \times 2 = 30$; assuming that the method converges by 5 iterations in the third round, $N_{n+1}^i = 15 \times 2 + 5 = 35$ and it goes to the next step. Furthermore, it is assumed that, the acceptable number of restartings in one step is set to 34 in order to reduce the incremental step-length up to almost one-hundredth of its value at the start of each step. Consequently, if a method does not manage to find a converged solution in a step after 34 restarting rounds, it is labeled as an unacceptable method.

- The analyses are implemented by programming in Matlab R2017a installed on Windows 8 64-bit on a PC with Intel®Core™i5-4210M CPU 2.60 GHz and 8.0 GB DDR3 RAM.
- The robustness of each analysis is checked according to the corresponding measures. If it satisfies crucial measures (i.e. the ability to reach a desired state without any global artificial unloading), its result is considered as an acceptable one for further assessment.
- The measure of robustness, speed, and smoothness of the acceptable analyses are calculated and scaled to fall between 0 and 1:

$$X_1 = \frac{R}{R_{\max}}, \quad X_2 = \frac{N_{\min}^i}{N^i}, \quad X_3 = \frac{S}{S_{\max}} \quad (46)$$

where X_1, X_2, X_3 are the normalized measures mentioned in Section 4 and R_{\max}, N_{\min}^i , and S_{\max} are the maximum of robustness, the minimum of total number of iterations, and the maximum of smoothness measures among all analyses, respectively. It should be noted that only the parts of the solution path which is obtained by employing the mentioned constraints are considered in the calculation of R, N^i , and S . In other words, the path initially found by force control is omitted from the calculations.

- The objective performance measures are calculated for each set of measures, $[X_1, X_2, X_3]$, and the set which has the largest objective measure is shown with a specific color on the space of importance ratios (e.g. see Figure 8 or 12).

5.1. Perforated beam

This example problem is presented in [25] and is a modified version of the numerical example in [16]). The problem considers fracture of a perforated beam (see Figure 5) by using

Table 2: The constraint functions and the values of N_{opt}^i employed for the example problems.

Constraint function	N_{opt}^i	Constraint symbol
Cylindrical arc-length control (equation (17))	4	$Cy4$
Cylindrical arc-length control (equation (17))	5	$Cy5$
Energy release control (equation (18))	4	$En4$
Energy release control (equation (18))	5	$En5$
κ control (equation (24))	4	$\kappa4$
κ control (equation (24))	5	$\kappa5$
κ - ω control (equation (25))	4	$\kappa\omega4$
κ - ω control (equation (25))	5	$\kappa\omega5$
Angle control (equation (23))	4	$An4$
Angle control (equation (23))	5	$An5$

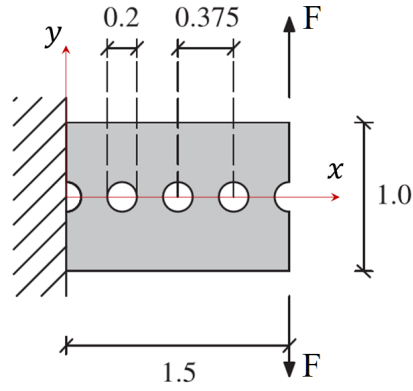


Figure 5: Perforated beam problem set-up and its dimensions in mm.

cohesive zones which are modeled by predefined planes in the beam. Pure mode-I fracture is assumed to occur along the x-axis because of the positions of holes and symmetry of the problem. Characteristics of the continuum and cohesive zones are shown in Table 3. For the continuum part, 6-node triangular elements with a 7-point Gauss integration scheme and for the cohesive zones, 3-node quadratic interface elements with a 3-point Newton-Côtes integration scheme (as suggested in [26]) are considered. A fixed 0.00156 N incremental step-length is selected for the force control together with 10^{-11} J for the dissipated energy threshold. The desired state of the perforated beam is defined to be where the vertical displacement of the top loading point is equal to 0.075 mm. All analyses are terminated at the first step which has passed the desired state.

Table 4 presents the performance measures when employing the acceptable constraint functions. As mentioned before, the acceptable constraints are those which have made the solver able to pass the desired state without any global artificial unloadings (see Figure 6). Moreover, it should be noted that by using dissipated energy control or the angle control constraint, the method will skip the parts of the path in which the body behaves elastically. This is because of the non-dissipative nature of the elastic behavior as well as the elastic unloading of the damage model to the origin, respectively. Figure 6 illustrates the force-displacement curve in y direction of top loading point of the perforated beam. In Figure 7, deformation of and crack propagation in the perforated beam are drawn for some steps of the analysis.

To select the best performing constraint, we may simply take the best of each single measure into account. For instance, according to the values of the speed and the smoothness measures, constraints $En5$ and $\kappa\omega4$ are the best among the others, respectively; additionally, according to the robustness measure, the constraints $En4$, $En5$, $\kappa4$, and $\kappa5$ are the best and equally perform without any restartings. Unless tracing snap-backs is not of significance, the solution path obtained by $En5$, the fastest constraint function in this example, is normally considered as a weak one since it does not find enough points on the exact curve (see Figure 6). On the other hand, the solution by $\kappa\omega4$, the smoothest constraint function in this example, might be considered as slow. Nevertheless, the constraint functions are put in the portfolio of acceptable ones of this example problem because the proposed performance measures are value-free. In other words, the analyst of the problem is the one who should give a judgment about the magnitudes of the measures and then decide on including (or excluding) numerical results of a method in (or from) the portfolio of results.

Table 3: Characteristics of the perforated beam problem.

Continuum area	
• Modulus of elasticity: $E = 100 \text{ N/mm}^2$	
• Poisson's ratio: $\nu = 0.30$	
• Thickness = 1 mm	
• Constitutive law: $\boldsymbol{\sigma} = \mathbf{C} \boldsymbol{\varepsilon}$	
• Plain strain condition: $\mathbf{C} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)}$	$\begin{bmatrix} 1 & \frac{\nu}{1-\nu} & 0 \\ \frac{\nu}{1-\nu} & 1 & 0 \\ 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix}$
Cohesive zone	
• Uniaxial tensile strength: $f_t = 1 \text{ N/mm}^2$	
• Mode-I fracture energy per unit area: $G_{f,I} = 2.5 \times 10^{-3} \text{ N/mm}$	
• Damage law: $\mathbf{t}_{\text{int}} = (1 - \omega)\mathbf{C}_{\text{int}}\mathbf{u}_{\text{int}}$	
with $\mathbf{C}_{\text{int}} = \begin{bmatrix} d_s & 0 \\ 0 & d_n \end{bmatrix}$ and $\omega = \begin{cases} 0 & \kappa < \kappa_0 \\ 1 - \frac{\kappa_0}{\kappa} \frac{\kappa_u - \kappa}{\kappa_u - \kappa_0} & \kappa_0 \leq \kappa < \kappa_u \\ 1 & \kappa \geq \kappa_u \end{cases}$	
where $\mathbf{t}_{\text{int}}^T = \begin{bmatrix} t_s & t_n \end{bmatrix}$ and $\mathbf{u}_{\text{int}}^T = \begin{bmatrix} u_s & u_n \end{bmatrix}$: traction and relative displacement vector of the interface in 2D respectively, subscripts s and n indicate direction tangent and normal to the interface surface respectively, $d_s = 5 \times 10^3 \text{ N/mm}^3$, and $d_n = 10^4 \text{ N/mm}^3$.	
• $\kappa_0 = f_t/d_n$, $\kappa_u = 2 G_{f,I}/f_t$, and κ is equal to maximum u_n occurred until current state of the body under analysis.	

Table 4: The performance measures calculated from the results of the perforated beam problem controlled by the acceptable constraint functions.

Constraint symbol	R	$1/N^i$	S
$En4$	1.00	17.241e-3	0.8479
$En5$	1.00	23.810e-3	0.7706
$\kappa4$	1.00	9.615e-3	0.8399
$\kappa5$	1.00	14.085e-3	0.8210
$\kappa\omega4$	0.25	2.833e-3	0.9810
$An4$	0.25	4.525e-3	0.9792

The combination of the measures, specially with desired importance factors, gives more freedom of choice for the best constraint. Furthermore, by the proposed objective measures, the advantages of each constraint could be more thoroughly explored. Figure 8 illustrates the areas of the importance ratios (described in Section 4) in which each acceptable constraint function has the largest objective performance measure based on equations (31) to (34).

It is observed from Figure 8 that $En5$ performs best where $\alpha_2/\alpha_3 > 1$; $En4$ is almost the best performing constraint function where $\alpha_3/\alpha_2 > 1$ and α_3/α_1 is between 0.2 and 5; $\kappa4$ and $\kappa5$ could not outperform in any region; the rest of the constraints perform differently based on different objective performance measures where α_3/α_1 and α_3/α_2 are approximately greater than 5: a) $An4$ has almost the best performance by P^{III} and P^{IV} except in the region of $\alpha_3/\alpha_2 > 90$ by P^{IV} where $\kappa\omega4$ performs better. b) based on P^I , $An4$ or $\kappa\omega4$ outperform where $\alpha_3/\alpha_2 < 45$ or > 45 , respectively; c) based on P^{II} , $An4$ or $\kappa\omega4$ has shown to have a better performance in the areas where $\alpha_3/\alpha_2 < 20$ or > 20 .

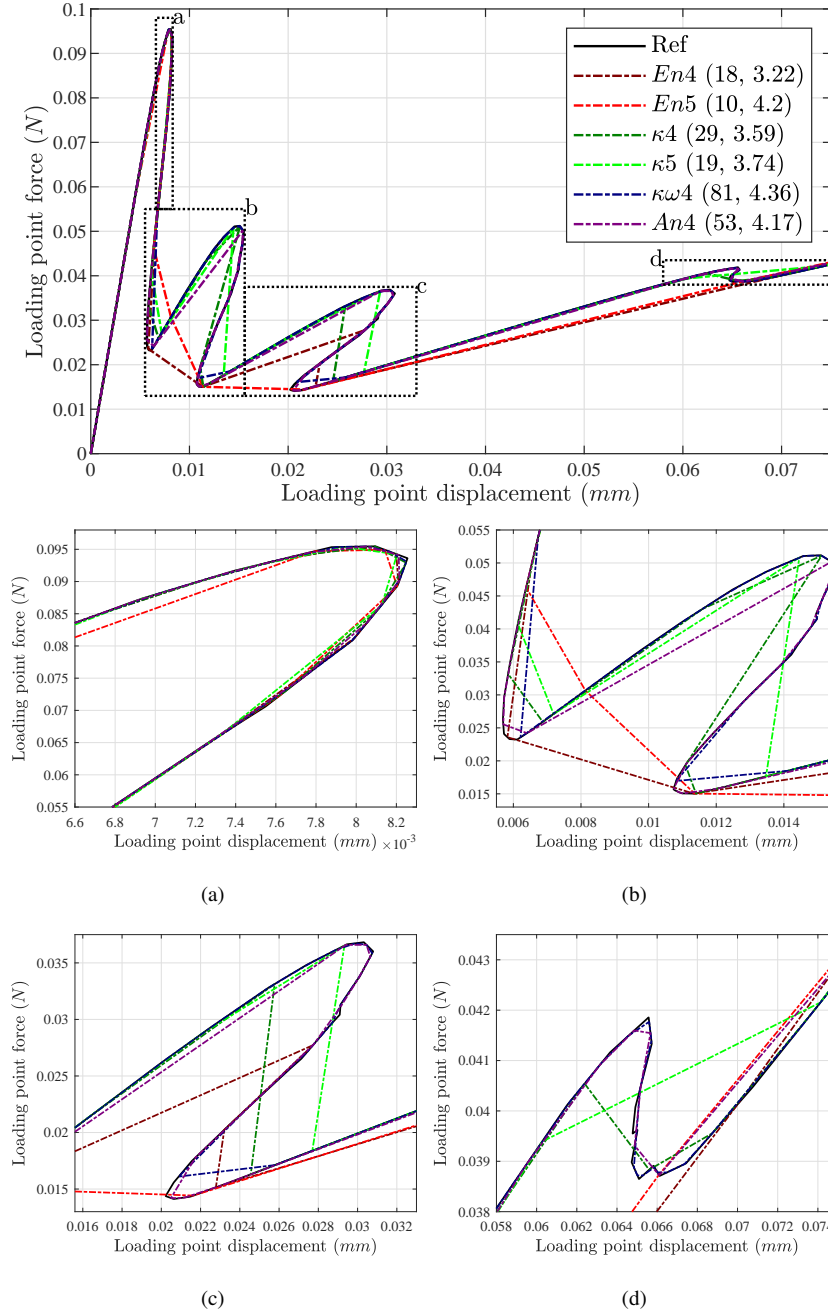


Figure 6: The presented constraints make the method proceed to the desired state although there are parts of the reference path which are skipped in the perforated beam problem. The black solid line represents the reference solution curve, the first number inside of the parentheses in front of the names of the constraint function shown in the legend indicates the total number of steps and the second one shows the average number of iterations per step. Zoom boxes 6(a) to 6(d) more clearly show the distance from the reference curve from the first to the fourth snap-back, respectively.

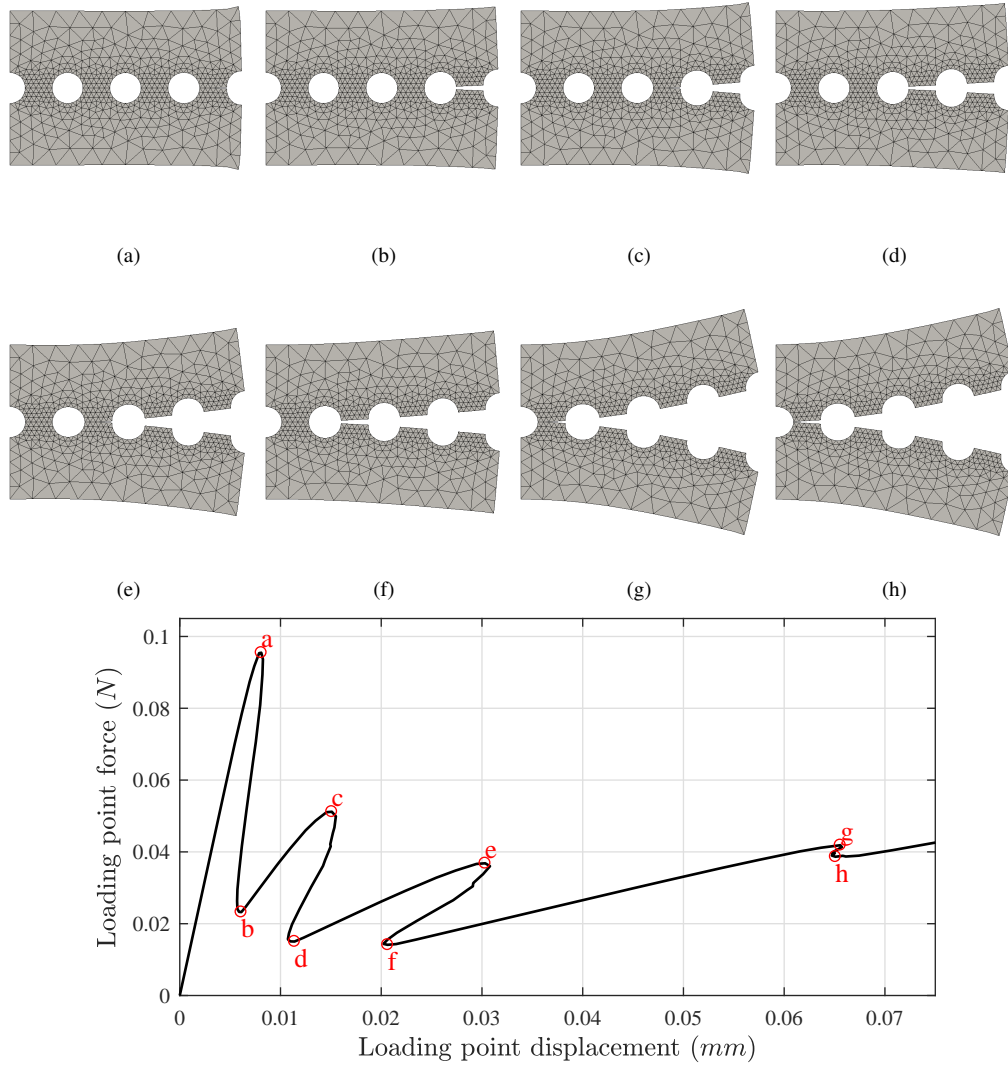


Figure 7: Crack propagation along symmetry axis of the perforated beam in some configurations of the body during analysis.

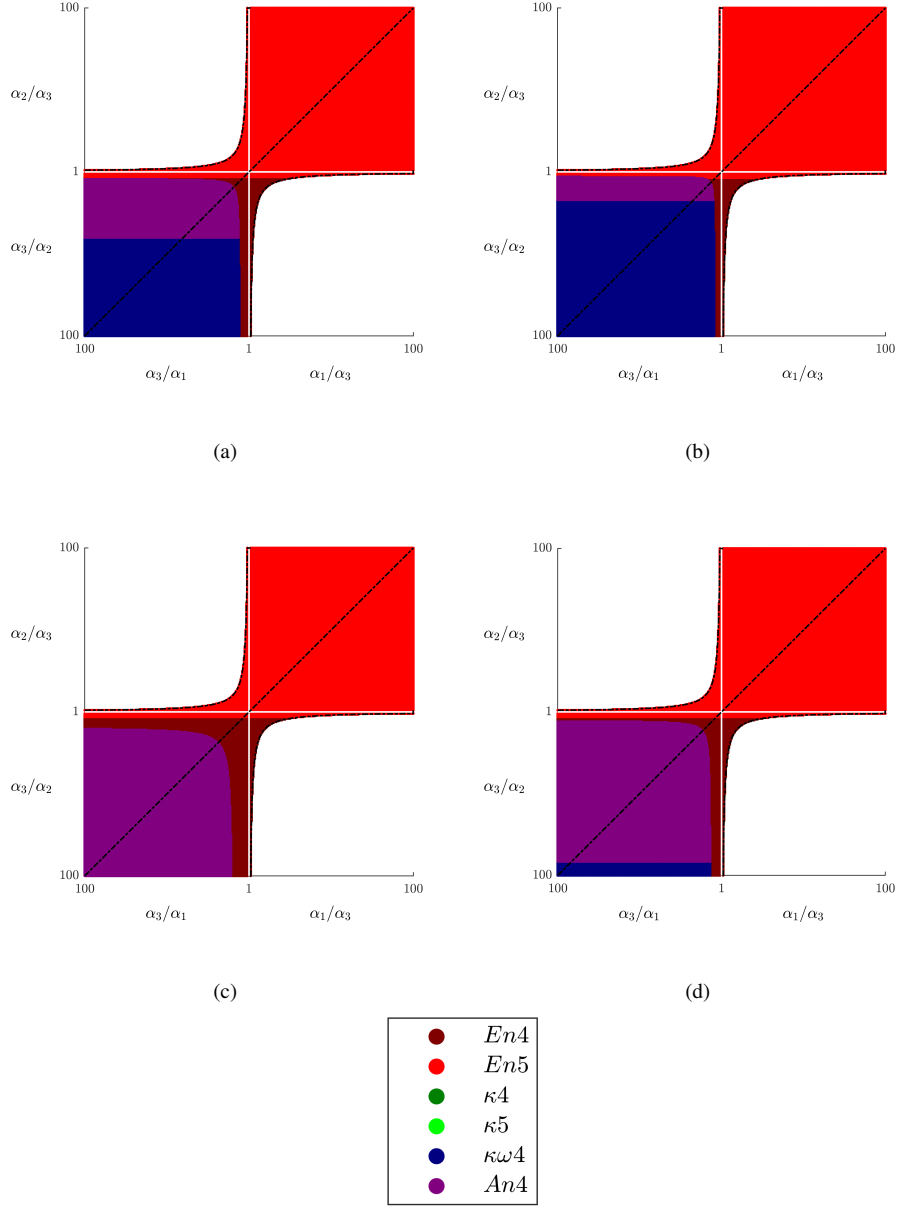


Figure 8: The space of objective performance measures based on (a) P^I , (b) P^{II} , (c) P^{III} , and (d) P^{IV} for the perforated beam problem.

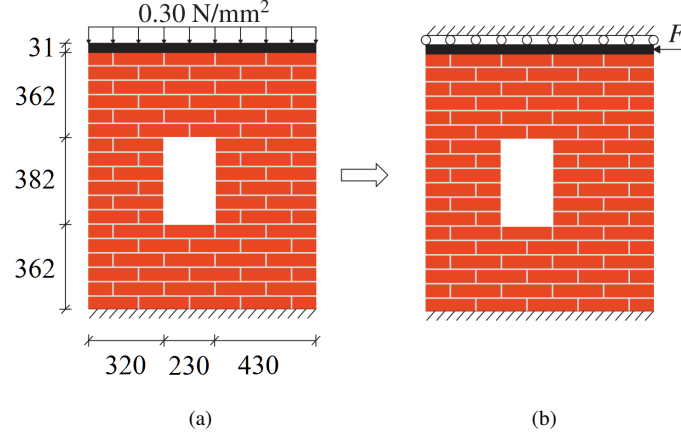


Figure 9: Confined masonry wall problem set-up and its dimensions in mm. (a) phase I: confining by a compressive load; (b) phase II: imposing a shear load.

5.2. Confined masonry wall

In this example, which is a slightly changed version of that explained in [25], we discuss the quasi-static response of a masonry wall preconfined with a vertical compressive load and subjected to a lateral in-plane force at its top (see Figure 9). The wall set-up is made up of bricks (full: $210 \text{ mm} \times 52 \text{ mm}$ and half: $100 \text{ mm} \times 52 \text{ mm}$) and 10 mm-wide mortar joints with 100 mm overall thickness. Unlike the wall in [25], kinematics of both bricks and joints are defined by conventional finite elements. In addition, bricks and joints are modeled exactly with their mentioned dimensions and geometry (i.e. bricks are not expanded to fill the gap introduced by zero-thickness joints). Damage is assumed to only occur within the mortar material by adopting mode-I fracture along the centerline of mortar joints while bricks remain elastic. Consequently, non-zero-width joints are modeled by interface elements located in their centerlines. Table 5 summarizes the properties of brick and mortar joint. Bricks are meshed by 4-node rectangular elements with a 4-point Gauss integration scheme and joints are modeled by 2-point quadratic interface elements with a 2-point Newton-Côtes integration scheme. For simulation purposes, a very stiff bar is added on top of the wall.

The force control started the analysis by a fixed incrementation of 297.24 N and a switching threshold of 0.001 J. After it switched to one of the mentioned control constraints, the analysis proceeded to pass the desired state which was defined by the horizontal displacement of the top right node to be equal to 1 cm. Table 6 shows the performance measure values of the results

Table 5: Characteristics of the confined masonry wall problem.

Bricks	
• Full brick dimensions: 210 mm × 52 mm × 100 mm	
• Half brick dimensions: 100 mm × 52 mm × 100 mm	
• Modulus of elasticity: $E = 16700 \text{ N/mm}^2$	
• Poisson's ratio: $\nu = 0.15$	
• Constitutive law: $\sigma = \mathbf{C} \epsilon$	
• Plain stress condition: $\mathbf{C} = \frac{E}{1-\nu} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}$	
Mortar joints	
• Dimensions: width = 10 mm and thickness = 100 mm	
• Uniaxial tensile strength: $f_t = 0.25 \text{ N/mm}^2$	
• Uniaxial compressive strength: $f_c = 10.5 \text{ N/mm}^2$	
• Poisson's ration: $\nu_{\text{int}} = 0.14$	
• Mode-I fracture energy per unit volume: $G_{f,I} = 0.018 \text{ N/mm}^2$	
• Damage law: $\mathbf{t}_{\text{int}} = (1 - \omega)\mathbf{C}_{\text{int}}\mathbf{u}_{\text{int}}$	
with $\mathbf{C}_{\text{int}} = \begin{bmatrix} d_s & 0 \\ 0 & d_n \end{bmatrix}$ and $\omega = \begin{cases} 0 & \kappa < \kappa_0 \\ 1 - \frac{\kappa_0}{\kappa} \exp\left(-\frac{\kappa - \kappa_0}{\beta_{\text{int}}}\right) & \kappa_0 \leq \kappa \end{cases}$	
where $\mathbf{t}_{\text{int}}^T = [t_s \quad t_n]$ and $\mathbf{u}_{\text{int}}^T = [u_s \quad u_n]$: traction and relative displacement vector of the interface in 2D respectively, subscripts \square_s and \square_n indicate direction tangent and normal to the interface surface respectively, $d_s = 36 \text{ N/mm}^3$, and $d_n = 82 \text{ N/mm}^3$.	
• $\kappa_0 = f_t/d_n$, $\beta_{\text{int}} = \frac{hG_{f,I}}{f_t} - \frac{1}{2}\kappa_0$, and κ is equal to maximum equivalent relative displacement, u_{eq} , occurred until current state of the body under analysis.	
• Equivalent relative displacement, u_{eq} , defined by a degenerated capped Drucker-Prager model [27]:	
$u_{eq} = \begin{cases} A I_u + B J_u & J_u \geq \frac{A-C}{D-B} I_u \\ C I_u + D J_u & J_u \leq \frac{A-C}{D-B} I_u \end{cases}$	
where $I_u = (1 + \nu_{\text{int}})u_n$,	
$J_u = \frac{1}{1+\nu_{\text{int}}} \sqrt{\frac{1}{3}(1 + \nu_{\text{int}} + \nu_{\text{int}}^3 + \nu_{\text{int}}^4)u_n^2 + \frac{1}{4}u_s^2}$,	
$A = \frac{1}{2} \frac{f_c - f_t}{f_c}$, $B = \frac{\sqrt{3}}{2} \frac{f_c + f_t}{f_c}$, $C = \frac{(f_b - f_c)f_t}{f_b f_c}$, and $D = \sqrt{3} \frac{(2f_b + f_c)f_t}{f_b f_c}$.	
• Biaxial compressive strength: $f_b = 1.2 \times f_c$	

Table 6: The performance measures calculated from the results of the confined masonry wall problem controlled by the acceptable constraint functions.

Constraint symbol	R	$1/N^i$	S
$En4$	0.333	2.762e-3	0.9863
$En5$	0.143	2.653e-3	0.9758
$\kappa4$	0.333	2.747e-3	0.9926
$\kappa5$	0.200	3.175e-3	0.9945
$\kappa\omega4$	0.143	2.079e-3	0.9969
$\kappa\omega5$	0.071	1.859e-3	0.9910
$An4$	0.200	2.463e-3	0.9902
$An5$	0.143	2.755e-3	0.9954

of employing acceptable constraints. In this example, $\kappa\omega5$ and $An5$ have also been capable of passing the desired state, in contrast with the previous example. Figure 10 illustrates the total lateral force versus the absolute displacement of the top right node related to the phase II analysis of the wall by utilizing the acceptable constraints. Deformed configurations of the masonry wall in some steps are illustrated in Figure 11. It shows that the damage propagation in the mortar joints mainly occurred from the top right to the bottom left of the wall due to direction of the lateral force.

The constraints $\kappa5$ and $\kappa\omega4$ have the first rank among the others when only considering the speed and the smoothness measures, respectively and the constraints $\kappa4$ and $En4$ are the best according to the robustness measure. Main observations from Figure 12 which explores the space of importance ratios based on the objective performance measures are that $\kappa4$ and $\kappa5$ perform best where $\alpha_3/\alpha_1 > 1$ or $\alpha_3/\alpha_2 > 1$; in the area of $\alpha_1/\alpha_3 > 1$ and $\alpha_2/\alpha_3 > 1$, $\kappa5$ and $En4$ have the best performance where $\alpha_2/\alpha_1 > 3$ and < 3 , respectively; the other acceptable constraints could not perform better than the best performing ones in any areas of the space.

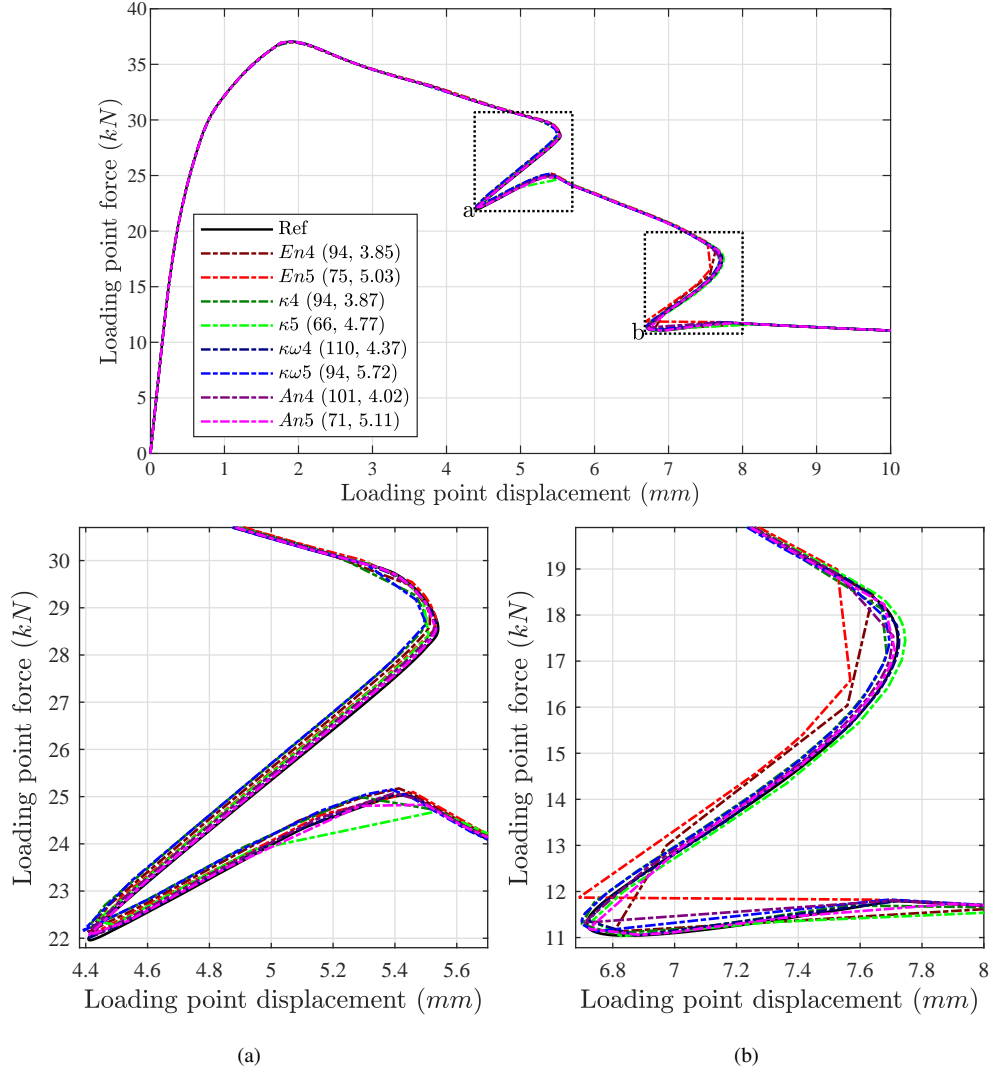


Figure 10: The presented constraints make the method proceed to the desired state although the obtained solution points are not exactly on the reference path in the confined masonry wall problem. The black solid line represents the reference solution curve. The first number inside of the parentheses in front of the names of the constraint function shown in the legend indicates the total number of steps and the second one shows the average number of iterations per step. Zoom boxes 10(a) and 10(b) more clearly show the distance from the reference curve for the first and the second snap-backs, respectively.

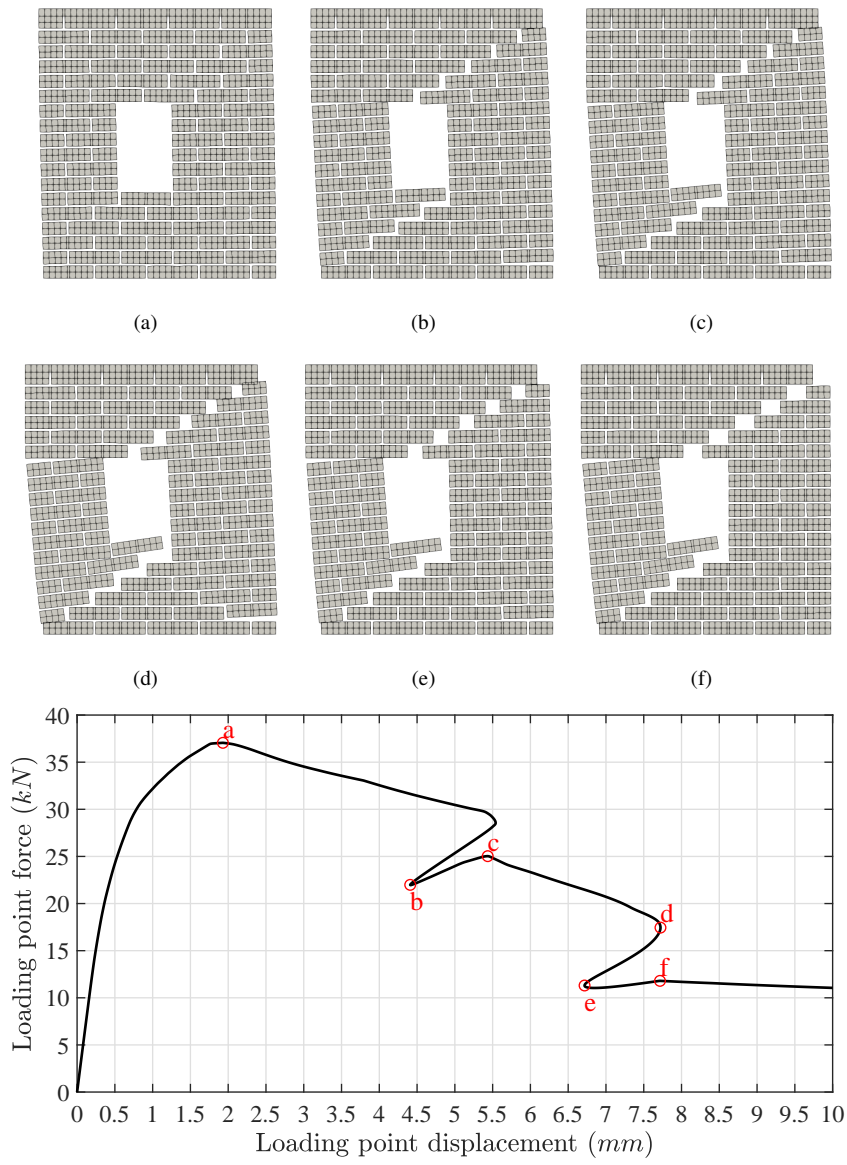


Figure 11: Damage propagation in the mortar joints in some configurations of the wall during analysis.

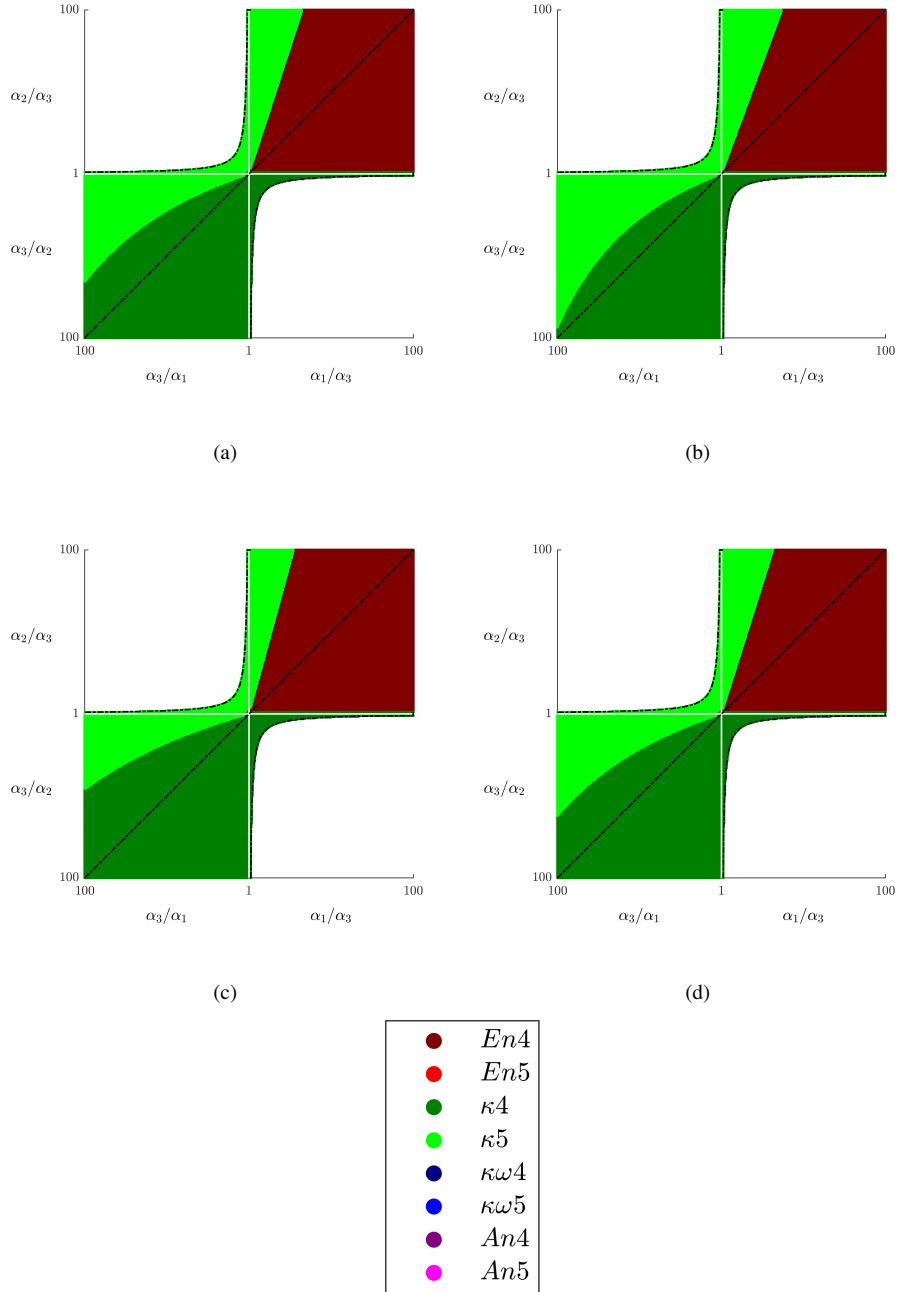


Figure 12: The space of objective performance measures based on (a) P^I , (b) P^{II} , (c) P^{III} , and (d) P^{IV} for the confined masonry wall problem.

6. Conclusions

We proposed four performance criteria and their measures for the a posteriori performance assessment of numerical results obtained by solving a quasi-static structural problem. Although the measure of each criterion is suggested, they could be defined in a different way which is specific for the problem and the solution method under assessment. Additionally, the conditions to construct objective performance measures as well as a framework for performance comparison according to them is proposed. These objective measures give the overall strength of a method in finding a numerical solution. Although, they are not absolute measures and should be used for relative comparisons between the performance of methods of a portfolio.

Furthermore, three new constraints functions applicable to the path-following method of solving quasi-static structural damage problems have been proposed. Based on the results of two numerical examples, it is concluded that κ control performs better when there are more non-linear interfaces present in the problems. In detail, by a larger ratio of the number of interface nodes over the number of elastic material nodes, the nonlinearity of the problem is more affected by the damage in the interfaces and is better dealt with the κ control constraint. Regarding the angle and κ - ω control constraints, the path-following method becomes more efficient than the dissipated energy control when the significance of the smoothness is higher than almost five times of that of speed and robustness for relatively small number of nonlinear interfaces.

It should be noted that, in general, if an analyst is able to directly or indirectly restrict the value of a performance measure to a certain range which is entirely labeled as ‘good’, that measure can be excluded from the set of performance measures. Furthermore, the example problems have been chosen to give a sense of the efficiency areas of the studied and proposed constraint functions by a typical set of influencing factors which are commonly employed (such as the step-length adaptation law, initial guess formula, etc.). The above conclusions can be generalized to problems with similar features such as finite element size, type of elements, underlying phenomena, etc. and may change with the variation of the problem features.

References

- [1] G. A. Wempner, Discrete approximations related to nonlinear theories of solids, *International Journal of Solids and Structures* 7 (11) (1971) 1581–1599. doi:10.1016/0020-7683(71)90038-2.
URL <http://www.sciencedirect.com/science/article/pii/0020768371900382>

- [2] E. Riks, An incremental approach to the solution of snapping and buckling problems, *International Journal of Solids and Structures* 15 (7) (1979) 529–551. doi:10.1016/0020-7683(79)90081-7.
URL <http://www.sciencedirect.com/science/article/pii/0020768379900817>
- [3] M. A. Crisfield, A fast incremental-iterative solution procedure that handles "snap-through", *Computers and Structures* 13 (1-3) (1981) 55–62. doi:10.1016/0045-7949(81)90108-5.
URL <GotoISI>://WOS:A1981LS27000007
- [4] E. Ramm, Strategies for tracing the nonlinear response near limit points, in: *Nonlinear Finite Element Analysis in Structural Mechanics*, Springer, Berlin, Heidelberg, 1981, pp. 63–89.
- [5] K. H. Schweizerhof, P. Wriggers, Consistent linearization for path following methods in nonlinear fe analysis, *Computer Methods in Applied Mechanics and Engineering* 59 (3) (1986) 261–279. doi:10.1016/0045-7825(86)90001-0.
URL <GotoISI>://WOS:A1986F097800001
- [6] B. W. R. Forde, S. F. Stiemer, Improved arc length orthogonality methods for nonlinear finite element analysis, *Computers and Structures* 27 (5) (1987) 625–630. doi:[http://dx.doi.org/10.1016/0045-7949\(87\)90078-2](http://dx.doi.org/10.1016/0045-7949(87)90078-2).
URL <http://www.sciencedirect.com/science/article/pii/0045794987900782>
- [7] H. B. Hellweg, M. A. Crisfield, A new arc-length method for handling sharp snap-backs, *Computers and Structures* 66 (5) (1998) 705–709.
URL <GotoISI>://WOS:000073956900014
- [8] Y. T. Feng, D. Peri, D. R. J. Owen, Determination of travel directions in path-following methods, *Mathematical and Computer Modelling* 21 (7) (1995) 43–59. doi:10.1016/0895-7177(95)00030-6.
- [9] E. A. de Souza Neto, Y. T. Feng, On the determination of the path direction for arc-length methods in the presence of bifurcations and 'snap-backs', *Computer Methods in Applied Mechanics and Engineering* 179 (1-2) (1999) 81–89. doi:10.1016/S0045-7825(99)00042-0.
URL http://ac.els-cdn.com/S0045782599000420/1-s2.0-S0045782599000420-main.pdf?_tid=1fbe0c5a-79af-11e6-80c1-00000aacb35e&acdnat=1473770554_97d3023559cc94d649c433179006b8cd
- [10] Z. Chen, H. L. Schreyer, A numerical-solution scheme for softening problems involving total strain control, *Computers and Structures* 37 (6) (1990) 1043–1050. doi:10.1016/0045-7949(90)90016-U.
URL <GotoISI>://WOS:A1990EP01400016
- [11] M. G. D. Geers, Enhanced solution control for physically and geometrically non-linear problems. part ii - comparative performance analysis, *International Journal for Numerical Methods in Engineering* 46 (2) (1999) 205–230. doi:10.1002/(Sici)1097-0207(19990920)46:2<205::Aid-Nme669>3.0.Co;2-S.
URL <GotoISI>://WOS:000082114300003
- [12] T. Pohl, E. Ramm, M. Bischoff, Adaptive path following schemes for problems with softening, *Finite Elements in Analysis and Design* 86 (2014) 12–22. doi:10.1016/j.finel.2014.02.005.
- [13] I. M. May, Y. Duan, A local arc-length procedure for strain softening, *Computers and Structures* 64 (1) (1997) 297–303. doi:10.1016/S0045-7949(96)00172-1.
- [14] G. Alfano, M. A. Crisfield, Solution strategies for the delamination analysis based on a combination of local-control

- arc-length and line searches, *International Journal for Numerical Methods in Engineering* 58 (7) (2003) 999–1048. doi:10.1002/nme.806.
URL <GotoISI>://WOS:000185796000002
- [15] E. Lorentz, P. Badel, A new path-following constraint for strain-softening finite element simulations, *International Journal for Numerical Methods in Engineering* 60 (2) (2004) 499–526. doi:10.1002/nme.971.
- [16] M. A. Gutiérrez, Energy release control for numerical simulations of failure in quasi-brittle solids, *Communications in Numerical Methods in Engineering* 20 (1) (2004) 19–29. doi:10.1002/cnm.649.
URL <GotoISI>://WOS:000188168200002
- [17] C. V. Verhoosel, J. J. C. Remmers, M. A. Gutiérrez, A dissipation-based arc-length method for robust simulation of brittle and ductile failure, *International Journal for Numerical Methods in Engineering* 77 (9) (2009) 1290–1321. doi:10.1002/nme.2447.
URL <GotoISI>://WOS:000263574500004
- [18] S. May, J. Vignollet, R. de Borst, A new arc-length control method based on the rates of the internal and the dissipated energy, *Engineering Computations* 33 (1) (2016) 100–115. doi:10.1108/ec-02-2015-0044.
- [19] A. Stanić, B. Brank, A path-following method for elasto-plastic solids and structures based on control of plastic dissipation and plastic work, *Finite Elements in Analysis and Design* 123 (2017) 1–8. doi:10.1016/j.finel.2016.09.005.
URL <GotoISI>://WOS:000390621700001
- [20] M. Ritto-Corrêa, D. Camotim, On the arc-length and other quadratic control methods: Established, less known and new implementation procedures, *Computers and Structures* 86 (11-12) (2008) 1353–1368. doi:10.1016/j.compstruc.2007.08.003.
URL <GotoISI>://WOS:000256285900018
- [21] A. Stanić, B. Brank, J. Korelc, On path-following methods for structural failure problems, *Computational Mechanics* 58 (2) (2016) 281–306. doi:10.1007/s00466-016-1294-y.
URL <GotoISI>://WOS:000379322400006
- [22] H. Keller, The bordering algorithm and path following near singular points of higher nullity, *SIAM Journal on Scientific and Statistical Computing* 4 (4) (1983) 573–582. doi:https://doi.org/10.1137/0904039.
- [23] P. X. Bellini, A. Chulya, An improved automatic incremental algorithm for the efficient solution of nonlinear finite element equations, *Computers and Structures* 26 (1) (1987) 99–110. doi:http://dx.doi.org/10.1016/0045-7949(87)90240-9.
URL http://www.sciencedirect.com/science/article/pii/0045794987902409
- [24] M. G. D. Geers, Enhanced solution control for physically and geometrically non-linear problems. part i - the subplane control approach, *International Journal for Numerical Methods in Engineering* 46 (2) (1999) 177–204. doi:10.1002/(Sici)1097-0207(19990920)46:2<177::Aid-Nme668>3.3.Co;2-C.
URL <GotoISI>://WOS:000082114300002
- [25] B. Vandoren, K. De Proft, A. Simone, L. J. Sluys, A novel constrained large time increment method for modelling quasi-brittle failure, *Computer Methods in Applied Mechanics and Engineering* 265 (2013) 148–162. doi:10.1016/j.cma.2013.06.005.
URL <GotoISI>://WOS:000324005600010

- [26] J. C. J. Schellekens, R. de Borst, On the numerical integration of interface elements, *International Journal for Numerical Methods in Engineering* 36 (1) (1993) 43–66. doi:DOI10.1002/nme.1620360104.
URL <GotoISI>://WOS:A1993KE09900003
- [27] B. Vandoren, K. De Proft, A. Simone, L. J. Sluys, Mesoscopic modelling of masonry using weak and strong discontinuities, *Computer Methods in Applied Mechanics and Engineering* 255 (2013) 167–182. doi:10.1016/j.cma.2012.11.005.