Performance-based step-length adaptation laws for path-following methods

Fayezioghani, A.; Vandoren, B.; Sluys, L. J.

DOI
10.1016/j.compstruc.2019.07.009

Publication date
2019

Document Version
Final published version

Published in
Computers and Structures

Computers and Structures 223 (2019) 106100

Contents lists available at ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc

Performance-based step-length adaptation laws for path-following methods

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

ARTICLE INFO

Article history:
Received 13 March 2019
Accepted 10 July 2019
Available online 9 August 2019

Keywords:
Step-length adaptation law
Path-following method
Performance measure
Quasi-brittle materials
Snap-back

ABSTRACT

A quasi-static problem is usually formulated by the equations of static equilibrium and a load parameter which shows the level of loading. Path-following methods are widely used to analyze these kinds of problems. These methods add a constraint function to the equilibrium equations in order to determine the loading evolution. There is a parameter in the constraint function, the step-length, which should be positive in each analysis step and which is determined by a step-length adaptation law. Different adaptation laws control the step-length growth differently, and thus, they influence the performance of the solution. We propose two novel types of adaptation laws based on (a) a local degree of smoothness and (b) global performance measures of the solution. The former uses the angle between the linearized solution path and the tangent to the analytical solution curve while the latter employs simple prediction models for the future evolution of two performance measures. Moreover, appropriate constraint functions for the latter are suggested. Example problems of structural damage are solved by path-following methods utilizing the proposed adaptation laws as well as a conventional one. Results show that the new laws raise distinct possibilities to have solutions with an improved performance.

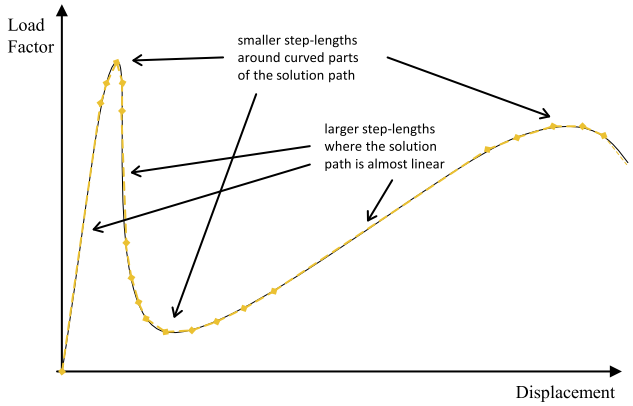
1. Introduction

Solving a quasi-static nonlinear structural problem usually requires determining the evolution of a load parameter. Path-following methods are effectively used to find the load parameter in each analysis step. The method, essentially, adds another equation, called a *constraint function*, to the system of equilibrium equations. The characteristic of this function is that it always increases a response of the structure (e.g. the arc-length [1], dissipated energy [2], etc.) by a positive valued step-length. The amount of step-length increase in each analysis step influences the quality of the numerical solution. Fig. 1 emphasizes the importance of using appropriate step-lengths by showing the solution points obtained from two different step-lengths in the analysis steps. The step-length either is a constant during the analysis or is determined in each analysis step according to a *step-length adaptation law*. The former is not a practical choice because it usually faces convergence issues for sharp nonlinear parts of the solution path or for rather considerable changes in the displacement vector. It may also lead to too many solution points which provide negligible progress in the displacement vector. The latter usually utilizes current and/or previous solution points so as to modify the step-length of the next step.

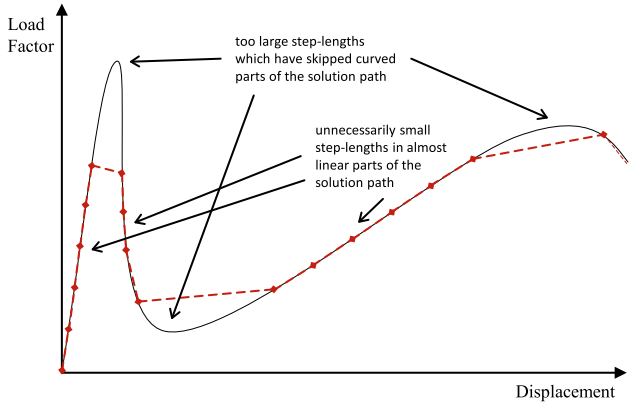
For example, Crisfield [1] and Ramm [3] used the ratio between an optimal number of iterations and the number of iterations needed to converge to a solution point in each step as a correction to the current step-length. Bergan et al. [4] introduced a different nonlinearity measure called 'current stiffness parameter'. Bergan [5] also recommended a step-length correction ratio that maintained the variation of this measure approximately constant. Widjaja [6] used this measure to adapt the step-length but in a different way than Bergan. And, Chan [7] adopted a simpler version than Bergan's current stiffness parameter. Like Bergan, Eriksson [8] derived two stiffness measures which provided some information about the iterative behavior of solution method, limit points of load, bifurcations, etc. He used different iteration constraining equations together with searching critical solution points [9]. He also utilized derivatives of the tangential stiffness matrix and several higher-order methods in order to improve predictions in the incremental solution procedure and identification of critical points [10,11]. With emphasis on the development of criteria utilized during the corrector phase, Eriksson and Kouhia [12] developed two step-length adaptation procedures. Schmidt [13] proposed an adaptive step-length procedure which imposed the drifting error to be within a contractive boundary at each analysis step. He, then,

* Corresponding author.
E-mail address: bram.vandoren@uhasselt.be (B. Vandoren).

https://doi.org/10.1016/j.compstruc.2019.07.009
0045-7949/



(a) Solution points found by method A



(b) Solution points found by method B

Fig. 1. Two methods have found different solution points on the exact solution curve (solid black line). Method A has used better values for the step-lengths than method B.

used it together with a set of conditions for the detection of impending divergence of the corrector phase. From the view of corrector phase, Georg [14] sketched a step-length adaptation law based on elementary asymptotic error considerations while Den Heijer and Rheinboldt [15] proposed an error model for the corrector iterations. By the latter, one could predict an approximation of the step-length value needed for a specified error. In contrast, Schwetlick [16] investigated on the predictor phase. He proposed a Newton-type path-following method by using a model based on the Kantorovich theorem and provided a condition for the maximal step-length. Szyszkowski and Husband [17] proposed a path-following technique that monitored and used the curvature of the equilibrium path to predict both the step-length and forward direction. Main categories and aspects of using step-length adaptation laws are discussed in [18,19], respectively.

The mentioned step-length adaptation laws focus on providing relative robustness by keeping a simple measure of local degree of nonlinearity almost constant in each analysis step. Although, it may prove to be beneficial to consider other objectives (such as smoothness and speed of the solution process explained in Section 3.1) in forming the adaptation law. We, therefore, propose a first law in Section 4 which modifies a local degree of nonlinearity based on the angle between the linearized solution path and the tangent to the analytical solution curve. As a second law, we propose an adaptation based on an objective combination of a speed measure and a cosine measure in Section 5. An issue of using global performance measures in the adaptation process is that it is needed to know the complete solution path to be able to calculate them, while only a part of the solution path is known when step-length adaptations of intermediate steps are needed! This contradiction could be dealt with via the *prediction* of the (unknown) future part of the solution path. For this, we employ simple and effective mathematical models to solely predict the future values of basic performance measures in each step. By using these prediction models, a step-length which maximizes an objective performance measure is determined and used as the step-length of the next step. As mentioned before, path-following methods could be improved in two aspects: the constraint functions and the step-length. In [20], we focused on improvements of the quality of path-following methods by focusing on new mathematical formulations of constraint functions, while we, in this paper, propose step-length adaptation laws to enhance the overall performance of path-following methods.

2. Damage model and the path-following method

In this section, we present the damage model of our problem as well as the path-following methods which are used in the next sections.

2.1. Damage model

In the numerical examples of Section 6, an isotropic damage model is adopted to model the nonlinear behavior of interface elements. The constitutive relation follows

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

where $\boldsymbol{t}_{\text{int}}$ and $\boldsymbol{u}_{\text{int}}$ are the traction and the relative displacement field (or jump) of the interface, respectively, $\mathbf{C}_{\text{int}}$ 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). In case of irreversible damage, ω is defined as a function of a parameter which stores the largest damage state occurred in each material point. The parameter is usually called the damage history parameter, κ. The relation between κ and u_{eq} is depicted by the Karush-Kuhn-Tucker conditions:

$$u_{\text{eq}} - \kappa \leqslant 0, \quad \dot{\kappa} \geqslant 0, \quad (u_{\text{eq}} - \kappa)\,\dot{\kappa} = 0 \tag{2}$$

where $\dot{\kappa}$ is the rate of change of damage history parameter and $u_{\text{eq}} = u_{\text{eq}}(\boldsymbol{u}_{\text{int}})$ is an invariant measure of the total relative displacement. It should be noted that two other assumptions are made for modelling damage in this paper: (a) healing of the damaged points does not occur and (b) the damage history parameter is a continuous function of (artificial) time. These assumptions together with (2) guarantee the monotonic increase of κ. Typical examples of damage parameter functions are *linear damage*

$$\omega = \begin{cases} 0, & \kappa < \kappa_0 \\ 1 - \frac{\kappa_0}{\kappa}\frac{\kappa_{\text{u}}-\kappa}{\kappa_{\text{u}}-\kappa_0}, & \kappa_0 \leqslant \kappa < \kappa_{\text{u}} \\ 1, & \kappa_{\text{u}} \leqslant \kappa \end{cases} \tag{3}$$

and *exponential damage*

$$\omega = \begin{cases} 0, & \kappa < \kappa_0 \\ 1 - \frac{\kappa_0}{\kappa}\exp\left(-\frac{\kappa-\kappa_0}{\beta_{\text{int}}}\right), & \kappa_0 \leqslant \kappa \end{cases} \tag{4}$$

where κ_0 and κ_{u} are the values of the history parameter indicating the initiation and end of the damage growth, respectively and β_{int} is a parameter that scales the exponential growth related to the fracture energy.

2.2. Path-following method

A quasi-static structural finite element problem which is parametrized for external loads by a load factor normally takes the following form of a nonlinear algebraic system of equations

$$\boldsymbol{r}(\boldsymbol{a},\lambda) = \boldsymbol{f}_{\mathrm{IN}}(\boldsymbol{a},\lambda) - \boldsymbol{f}_{\mathrm{EX}}(\boldsymbol{a},\lambda) = \boldsymbol{0} \tag{5}$$

where $\boldsymbol{r}$ is the vector of unbalanced or residual forces, $\boldsymbol{f}_{\mathrm{IN}}$ and $\boldsymbol{f}_{\mathrm{EX}}$ are the vectors of internal and external forces, respectively, $\boldsymbol{a}$ is the displacement vector, and λ is the load factor. A path-following method utilizes an extra equation, called a constraint function, in order to solve (5) and find the evolution of the displacements and the load factor. In other words, the method searches for a solution with respect to an artificial (or pseudo) time. A typical constraint function can be formulated as

$$g(\boldsymbol{a},\lambda,\eta) = h(\boldsymbol{a},\lambda) - \eta = 0 \tag{6}$$

where g is the (total) constraint function, h is the (total) constraint kernel, and η is the independent scalar total step-length parameter which must monotonically increase. The function h is called the (total) constraint kernel because its increase by the magnitude of η controls finding the solution points. In a sequential process of searching for responses by an incremental-iterative method, it is presumed for the decomposition of $\boldsymbol{a}_{n+1}, \lambda_{n+1}$, and η_{n+1} that

$$\boldsymbol{a}_{n+1} = \boldsymbol{a}_n + \Delta\boldsymbol{a}_{n+1}, \quad \lambda_{n+1} = \lambda_n + \Delta\lambda_{n+1}, \quad \eta_{n+1} = \eta_n + \Delta\eta_{n+1} \tag{7}$$

where $\Delta\boldsymbol{a}_{n+1}, \Delta\lambda_{n+1}$, and $\Delta\eta_{n+1}$ are the increments of $\boldsymbol{a}, \lambda$, and η in step (or increment) $n+1$, respectively. In addition, the system composed of Eqs. (5) and (6) is reformulated to have functions of $\Delta\boldsymbol{a}_{n+1}, \Delta\lambda_{n+1}$, and $\Delta\eta_{n+1}$ (instead of functions of $\boldsymbol{a}_{n+1}, \lambda_{n+1}$, and η_{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 increment of the displacements and the load factor are usually proposed and utilized:

$$\begin{bmatrix} \boldsymbol{r}_{n+1}(\Delta\boldsymbol{a}_{n+1}, \Delta\lambda_{n+1}) \\ g_{n+1}(\Delta\boldsymbol{a}_{n+1}, \Delta\lambda_{n+1}, \Delta\eta_{n+1}) \end{bmatrix} = \boldsymbol{0} \tag{8}$$

where $\boldsymbol{r}_{n+1}(\Delta\boldsymbol{a}_{n+1}, \Delta\lambda_{n+1}) = \boldsymbol{r}(\boldsymbol{a}_{n+1}, \lambda_{n+1})$ and $g_{n+1}(\Delta\boldsymbol{a}_{n+1}, \Delta\lambda_{n+1}, \Delta\eta_{n+1}) = g(\boldsymbol{a}_{n+1}, \lambda_{n+1}, \eta_{n+1})$ are the residual force vector and the (incremental) constraint function in step $n+1$, respectively. The (incremental) constraint function is equal to $h_{n+1}(\Delta\boldsymbol{a}_{n+1}, \Delta\lambda_{n+1}) - \Delta\eta_{n+1}$ where h_{n+1} and $\Delta\eta_{n+1}$ are the (incremental) constraint kernel and step-length, respectively. It should be noted that $\boldsymbol{r}_{n+1}, g_{n+1}$, and h_{n+1} are different functions from $\boldsymbol{r}, g$, and h. We have employed the Newton-Raphson method to solve (8) in step $n+1$:

$$\begin{bmatrix} \boldsymbol{K}_{\mathrm{t}}^k & \boldsymbol{V}^k \\ \boldsymbol{z}^k & w^k \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{a}_{n+1}^{k+1} \\ \delta\lambda_{n+1}^{k+1} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{r}_{n+1}^k \\ g_{n+1}^k \end{bmatrix} \tag{9}$$

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

$$\boldsymbol{K}_{\mathrm{t}}^k = \frac{\partial \boldsymbol{r}_{n+1}}{\partial \left(\Delta\boldsymbol{a}_{n+1}^k\right)^{\mathrm{T}}}, \tag{10}$$

$$\boldsymbol{V}^k = \frac{\partial \boldsymbol{r}_{n+1}}{\partial \left(\Delta\lambda_{n+1}^k\right)}, \tag{11}$$

$$\boldsymbol{z}^k = \frac{\partial g_{n+1}}{\partial \left(\Delta\boldsymbol{a}_{n+1}^k\right)^{\mathrm{T}}}, \tag{12}$$

$$w^k = \frac{\partial g_{n+1}}{\partial \left(\Delta\lambda_{n+1}^k\right)}, \tag{13}$$

the difference vector of the solution is composed of

$$\delta\boldsymbol{a}_{n+1}^{k+1} = \Delta\boldsymbol{a}_{n+1}^{k+1} - \Delta\boldsymbol{a}_{n+1}^k, \tag{14}$$

$$\delta\lambda_{n+1}^{k+1} = \Delta\lambda_{n+1}^{k+1} - \Delta\lambda_{n+1}^k, \tag{15}$$

and the components of the right-hand side of (9) are equal to

$$\boldsymbol{r}_{n+1}^k = \boldsymbol{r}_{n+1}\left(\Delta\boldsymbol{a}_{n+1}^k, \Delta\lambda_{n+1}^k\right), \tag{16}$$

$$g_{n+1}^k = g_{n+1}\left(\Delta\boldsymbol{a}_{n+1}^k, \Delta\lambda_{n+1}^k\right), \tag{17}$$

in which superscript k indicates the kth iteration, and $\boldsymbol{x}^{\mathrm{T}}$ is the transpose of vector $\boldsymbol{x}$. The Jacobian matrix in (9) is unsymmetric in general. Therefore the bordering algorithm (e.g. see [21]) is employed to benefit from the symmetry of the tangential stiffness matrix $\boldsymbol{K}_{\mathrm{t}}$ by the following decomposition:

$$\delta\boldsymbol{a}_{n+1}^{k+1} = \bar{\delta}\boldsymbol{a}_{n+1}^{k+1} - \delta\lambda_{n+1}^{k+1}\,\tilde{\delta}\boldsymbol{a}_{n+1}^{k+1} \tag{18}$$

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

$$\boldsymbol{K}_{\mathrm{t}}^k\,\bar{\delta}\boldsymbol{a}_{n+1}^{k+1} = -\boldsymbol{r}_{n+1}^k, \boldsymbol{K}_{\mathrm{t}}^k\,\tilde{\delta}\boldsymbol{a}_{n+1}^{k+1} = \boldsymbol{V}^k \tag{19}$$

and $\delta\lambda_{n+1}^{k+1}$ is calculated by substituting Eq. (18) in the second part of (9):

$$\delta\lambda_{n+1}^{k+1} = -\frac{g_{n+1}^k + \boldsymbol{z}^k\,\bar{\delta}\boldsymbol{a}_{n+1}^{k+1}}{w^k - \boldsymbol{z}^k\,\tilde{\delta}\boldsymbol{a}_{n+1}^{k+1}} \tag{20}$$

2.3. Constraint functions

The constraint functions used in solving the example problems are the energy release control [2] and slightly different versions of κ and κ–ω control proposed by the authors in [20]. The *energy release control* constraint is

$$\begin{aligned} g_{n+1} &= h_{n+1}(\Delta\boldsymbol{a}_{n+1}, \Delta\lambda_{n+1}) - \Delta\eta_{n+1} \\ &= \frac{1}{2}\left(\lambda_n\,\Delta\boldsymbol{a}_{n+1}^{\mathrm{T}} - \Delta\lambda_{n+1}\,\boldsymbol{a}_n^{\mathrm{T}}\right)\hat{\boldsymbol{f}}_{\mathrm{EX}} - \Delta\eta_{n+1} \end{aligned} \tag{21}$$

where h_{n+1} is defined as the variation of the energy release in step $n+1$, $\hat{\boldsymbol{f}}_{\mathrm{EX}}$ is the unit vector of external forces when the external force is independent of the deformations (i.e. $\boldsymbol{f}_{\mathrm{EX}} = \lambda\hat{\boldsymbol{f}}_{\mathrm{EX}}$). The second constraint function is called *modifiedκcontrol*:

$$g_{n+1}\left(\Delta\boldsymbol{a}_{n+1}, \Delta\eta_{n+1}\right) = \int_{\Omega_0} \Delta\kappa_{n+1}\,\mathrm{d}\Omega_0 - \Delta\eta_{n+1} \tag{22}$$

where $\Delta\kappa_{n+1} = \kappa_{n+1} - \kappa_n$ is the increment of the history parameter in step $n+1$, κ_{n+1} is equal to $\max(\kappa_n, u_{\mathrm{eq}}(\boldsymbol{a}_n + \Delta\boldsymbol{a}_{n+1}))$, κ_n is the converged value of the damage history in step n, and Ω_0 and $\mathrm{d}\Omega_0$ are the domain of and the volume element of the undeformed body under analysis. The only difference between this constraint function and its original formulation in [20] is that it integrates over the undeformed domain of the body instead of over last deformed configuration. For the formation of a third constraint function, the same idea of κ–ω control in [20] is adopted except that $\dot{\kappa}$ (instead of $\Delta\kappa$) is weakened for the points with more damage. To do this, firstly, we suggest the following function

$$\varpi(\kappa) = \int_0^t (1 - c_{\mathrm{L}}\omega_{\mathrm{L}})\,\dot{\kappa}\,\mathrm{d}t = \int_0^{\kappa} (1 - c_{\mathrm{L}}\omega_{\mathrm{L}})\,\mathrm{d}\kappa \tag{23}$$

which, after integration, becomes

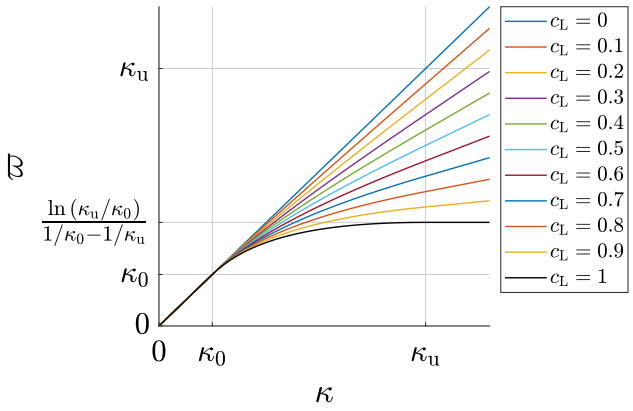


Fig. 2. The curve of ϖ versus κ for different values of c_L.

$$\varpi(\kappa) = \begin{cases} \kappa, & 0 \leqslant \kappa < \kappa_0 \\ \kappa + c_L \dfrac{\ln\left(\frac{\kappa}{\kappa_0}\right) - \frac{\kappa}{\kappa_0} + 1}{\frac{1}{\kappa_0} - \frac{1}{\kappa_u}}, & \kappa_0 \leqslant \kappa < \kappa_u \\ \kappa + c_L \left(\dfrac{\ln\left(\frac{\kappa_u}{\kappa_0}\right)}{\frac{1}{\kappa_0} - \frac{1}{\kappa_u}} - \kappa \right), & \kappa_u \leqslant \kappa \end{cases} \tag{24}$$

where ϖ is an augmented damage parameter, ω_L is the linear damage function of (3), and c_L is a coefficient between 0 and 1 to modify the contribution of damaged points for the solution progress. By setting smaller values of c_L, the contribution of fully damaged points will increase and vice versa. Fig. 2 shows the curve of ϖ versus κ for different values of c_L. The figure obviously illustrates that (a) ϖ is equal to κ from $\kappa = 0$ to κ_0 regardless of the magnitude of c_L, (b) the curve with $c_L = 0$ is $\varpi = \kappa$, and (c) ϖ remains as a constant value ($= \ln(\kappa_u/\kappa_0)/(1/\kappa_0 - 1/\kappa_u)$) after κ_u when $c_L = 1$. Finally, the integration of the increment of ϖ is proposed as the *modified*κ–ω *control*:

$$g_{n+1}(\Delta \boldsymbol{a}_{n+1}, \Delta\eta_{n+1}) = \int_{\Omega_0} (\Delta\varpi_{n+1} - \Delta\eta_{n+1})\, d\Omega_0 \tag{25}$$

where $\Delta\varpi_{n+1} = \varpi_{n+1} - \varpi_n$ is the increment in the augmented damage parameter. It should be noted that the function of ω in (1) can differ from ω_L. The difference of the above constraint function from the original κ–ω control is that (a) it integrates over the undeformed domain of the body and (b) it uses the augmented damage parameter instead of directly multiplying $1 - \omega$ to the increment of κ.

The constraint functions (22) and (25), like their original forms, possess the advantage of leading the solution path to a dissipative one and preventing global artificial unloading. In addition, the increments of the constituent parameters of modified κ and κ–ω control can be precisely calculated according to the values obtained in the analysis steps without knowing how those values are evolved between the steps. This is a significantly required property of a constraint function when utilizing the global adaptation law as will be discussed in Section 5.

3. Objective performance assessment

A method can increase its *speed* by jumping over curved parts of the analytical solution, which decrease the *smoothness* of the numerical curve; or, a method could be very fast for a specific problem while it does not converge well in most of other problems, which shows a weak *robustness*. To assess the performance of the numerical solutions found by using the adaptation laws, it is required to use performance measures as well as appropriate combinations of them.

3.1. Performance measures

We utilize the same *speed* and *robustness* measures proposed by Fayezioghani et al. [20]

$$P_{sp} = \frac{1}{N^i} \tag{26}$$

$$P_{rob} = \frac{1}{N^r + 1} \tag{27}$$

where P_{sp} is the speed measure, $N^i = \sum_{n=1}^{m} N_n^i$ is the total number of iterations of the analysis steps, N_n^i is the number of iterations needed to find the solution point in step n, m is the total number of steps (increments), P_{rob} is the robustness measure, $N^r = \sum_{n=1}^{m} N_n^r$ is the total number of restartings of a complete analysis, and N_n^r is the number of restartings needed in step n. It is common that if a method cannot find a 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 search method), which is often called the *restarting strategy*. In contrast to these two measures, we use a different *smoothness* measure based on the distance followed on a piecewise linear solution curve:

$$P_{sm} = \begin{cases} -\frac{D/D_{ref}}{D - D_{ref}} & , D < D_{ref} \\ \frac{1}{D - D_{ref}} & , D > D_{ref} \end{cases} \tag{28}$$

where P_{sm} is the smoothness measure, $D = \sum_{n=1}^{m} D_n$ is the sum of lengths of the lines connecting consecutive solution points, $D_n = \|\Delta \boldsymbol{v}_n\|$ is the length of that line, $\Delta \boldsymbol{v}_n^T = [\Delta \boldsymbol{a}_n^T, \gamma_0 \Delta\lambda_n]$ is the generalized solution vector, γ_0 is a scale factor between the space of nodal displacements and the load parameter, $\|\boldsymbol{x}\|$ is the Euclidean norm of vector $\boldsymbol{x}$, and subscript $\square_{ref}$ indicates the reference value. The solution curve obtained from a very comprehensive analysis is referred as a *reference solution* here. Practically, a reference solution is obtained by setting very small step-lengths which provide a very smooth numerical curve. Moreover, a measure which shows the relative smoothness of a numerical curve and is appropriate to be used in the procedure of step-length adaptation in Section 5 is introduced:

$$P_{cos} = \frac{\frac{1}{\cos\Theta - 1} + \frac{1}{2}}{\frac{1}{\cos\Theta_{ref} - 1} + \frac{1}{2}} \tag{29}$$

where P_{cos} is the *cosine* measure and

$$\cos\Theta = \frac{\sum_{n=1}^{m} \frac{1}{2} \Delta\eta_n (\cos\Theta_{n-1} + \cos\Theta_n)}{\sum_{n=1}^{m} \Delta\eta_n} \tag{30}$$

is the weighted average of the hyper cosines of consecutive lines connecting consecutive solution points, and $\cos\Theta_n = \frac{\Delta \boldsymbol{v}_n^T}{\|\Delta \boldsymbol{v}_n\|} \frac{\Delta \boldsymbol{v}_{n+1}}{\|\Delta \boldsymbol{v}_{n+1}\|}$ is the hyper cosine in step n. Fig. 3 graphically shows the linearized solution curve and the relative location and numbering of the hyper angles. It should be noted that (a) the same piecewise linear curve as that in the smoothness measure is assumed for the calculation of the hyper cosine of each step, (b) the reference values are suggested for each example problem of Section 6, and c) P_{cos} is specifically defined for the global adaptation law because it is more straightforward and more effective to construct a prediction model for it.

3.2. Objective combinations of the performance measures

The objective performance measure, P_{obj}, is composed of the normalized performance measures, and their corresponding

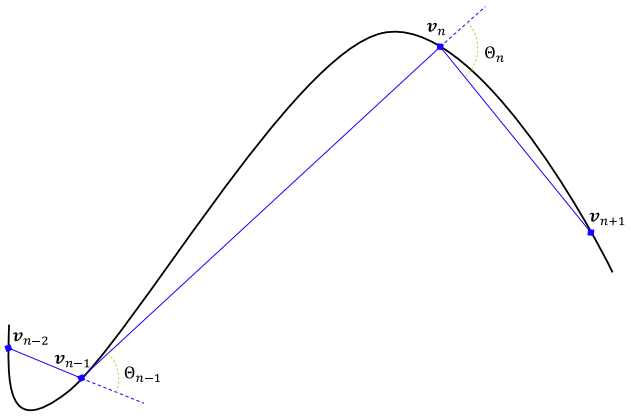


Fig. 3. Smaller angles (Θ) between two consecutive linearized solution paths (solid blue lines) makes the numerical curve get closer to the analytical one (black line), which means having a more smooth numerical solution. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

importance factors. In the example problems, we utilize two of the objective measures recommended in [20]:

$$P_{\text{obj.I}} = \sum_i \alpha_i X_i \tag{31}$$

$$P_{\text{obj.II}} = \prod_i X_i^{\alpha_i} \tag{32}$$

where X_i is the normalized measure, α_i is the positive importance factor of X_i relative to the other ones, and $\sum$ and $\prod$ are sum and product operators. A normalized measure is a performance measure which is scaled to fall between a certain interval, say 0 and 1. We assess the performance of the numerical solutions by three measures for smoothness, speed, and robustness. Therefore, two importance ratios, α_i/α_j, will affect the comparison results based on (31) or (32). We use the space of importance ratios proposed in [20] to uniformly visualize the areas in which each solution has the best performance (i.e. has the largest objective performance measure).

4. The local adaptation law based on local responses

A well-known and mostly used *conventional adaptation law*[1] is

$$\Delta\eta_{n+1} = \left(\frac{N^{\text{i}}_{\text{opt}}}{N^{\text{i}}_{n}}\right)\Delta\eta_n \tag{33}$$

where $N^{\text{i}}_{\text{opt}}$ is the optimal number of iterations per increment. This law is designed to approximately maintain the number of iterations needed for convergence in each step constant for the whole system of problem and solution method. To also consider the smoothness of the numerical curve in a new step-length adaptation law, we first define a measure of *local degree of smoothness* and include it in the formulation of the conventional law.

A numerical curve is usually drawn by connecting consecutive solution points by lines. The larger the number of solution points on a complete solution curve, the smoother the linearized numerical curve. In a perfect case of an infinite number of points, the direction of any line connecting two consecutive points is approximately the same as the tangent to the analytical curve. Thus, we suggest utilizing a *hyper angle* between the line connecting two solution points and the tangent to the analytical curve at the second point. The cosine of the hyper angle is calculated by

$$\cos(\xi_n) = \frac{\Delta \boldsymbol{v}_n^{\text{T}}}{\|\Delta \boldsymbol{v}_n\|} \frac{\frac{\partial \Delta \boldsymbol{v}_n}{\partial \Delta \eta_n}}{\|\frac{\partial \Delta \boldsymbol{v}_n}{\partial \Delta \eta_n}\|} \tag{34}$$

where ξ_n is the hyper angle of step n and

$$\frac{\partial \Delta \boldsymbol{v}_n}{\partial \Delta \eta_n} = \begin{bmatrix} \frac{\partial \Delta \boldsymbol{a}_n}{\partial \Delta \eta_n} \\ \gamma_0 \frac{\partial \Delta \lambda_n}{\partial \Delta \eta_n} \end{bmatrix}$$

is the generalized tangent to the analytical curve at the end of step n. The tangent vector $\partial \Delta \boldsymbol{a}_n/\partial \Delta \eta_n$ and the tangent value $\partial \Delta \lambda_n/\partial \Delta \eta_n$ are determined from solving the following algebraic equation

$$\begin{bmatrix} \boldsymbol{K}_{\text{t}}^k & \boldsymbol{V}^k \\ \boldsymbol{z}^k & w^k \end{bmatrix}_n \begin{bmatrix} \frac{\partial \Delta \boldsymbol{a}_n}{\partial \Delta \eta_n} \\ \frac{\partial \Delta \lambda_n}{\partial \Delta \eta_n} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ 1 \end{bmatrix} \tag{35}$$

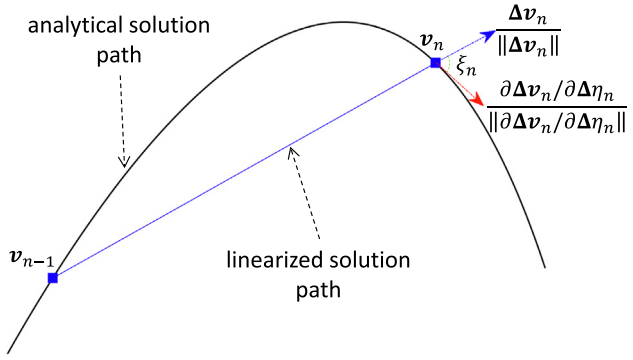
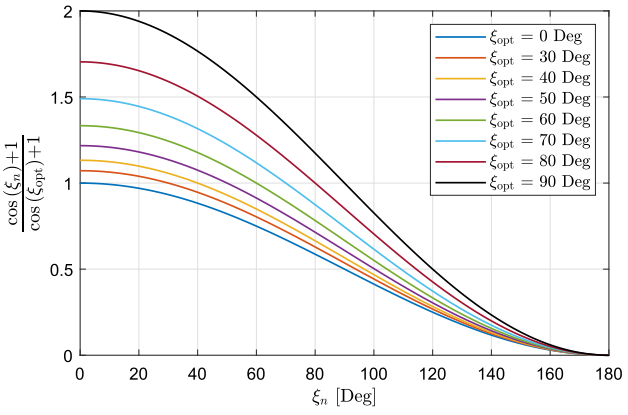
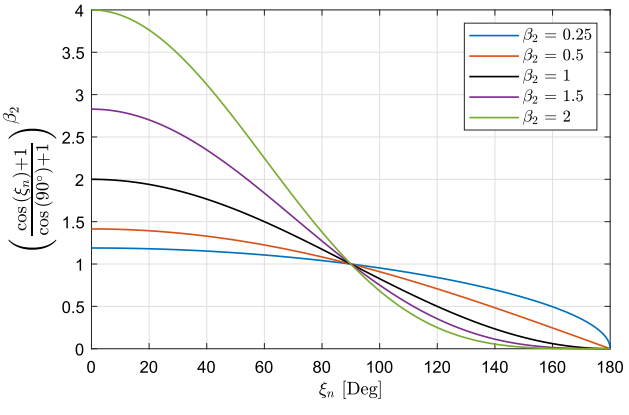


Fig. 4. If the linearized solution curve (the solid blue line) between two consecutive solution points, $\boldsymbol{v}_{n-1}$ and $\boldsymbol{v}_n$, has a smaller inclination with the tangent to the curve (i.e. it has a smaller ξ_n), a smoother curve will be obtained. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a)



(b)

Fig. 5. The local modification factor in Eq. (36) versus the hyper angle, ξ_n, is drawn for different values of ξ_{opt} and β_2 in (a) and (b), respectively.

where the Jacobian matrix is calculated according to the converged values at the end of step n. By the same analogy of the conventional law, we define a desired optimum value of the hyper angle, ξ_{opt}, and then try to keep ξ_n as close to it in every step as possible. The idea is to use the cosine distance between the linearized vector of solution and the tangent to the curve in order to obtain a smoother solution. Thus, we propose a modified version of the conventional adaptation law and call it *local adaptation law*:

$$\Delta\eta_{n+1} = \left(\frac{N^{i}_{opt}}{N^{i}_{n}}\right)^{\beta_1} \left(\frac{\cos(\xi_n)+1}{\cos(\xi_{opt})+1}\right)^{\beta_2} \Delta\eta_n \tag{36}$$

where β_1 and β_2 are the parameters to adjust magnification of each ratio in (36). Fig. 4 depicts a schematic visualization of the vectors and the hyper angle. The conventional adaptation law is a special case of the adaptation law (36) when β_1 and β_2 are set to 1 and 0, respectively. Fig. 5 shows the variation of the local modification factor with respect to ξ_{opt} and β_2. It is illustrated in the figure that (a) the modification factor decreases the step-length for the hyper angles larger than ξ_{opt} and vice versa and (b) for larger β_2, the modification is sharper and vice versa. For the analysts who want to use CPU time in the definition of the speed measure, the calculation of the tangent vector needed by the local adaptation law should be considered as a surplus to the computations of each increment by a conventional law.

5. The global adaptation law based on prediction of performance measures

The conventional as well as the local adaptation law use previous converged values to adapt the step-length in each analysis step. In other words, they adjust the step-length according to the *past* information. These kinds of adaptation laws are usually simple and easy to implement while they do not consider any information related to the global performance of the solution (i.e. the performance of the solution up to the final increment). The global performance is assessed by an objective combination of performance measures like in Eqs. (31) or (32). To be able to use these global measures in the global adaptation law in each step, the *future* responses should be known. Obviously, the future values are unknown in each step and, thus, an approximation or prediction of the future responses should be adopted, where it should be noted that the prediction of desired responses does not mean prediction of the whole solution path. In this section, we propose a global adaptation law based on the prediction of the speed and cosine measures defined in Section 3.1.

5.1. Progress indicator

For our global adaptation law, it is necessary to define the progress of a solution. In order to determine the progress of a problem with a progress indicator, we suggest using the sum of the step-lengths up to the current increment, q:

$$s_q = \sum_{n=1}^{q} \Delta\eta_n \tag{37}$$

where s_q is the indicator of the progress of the problem along the solution path up to increment q. To consider the ability of the method to reach to a *desired stage* of an analysis, the progress indicator is selected to be equal to a reference value at the final step, m. Thus,

$$s_m = \sum_{n=1}^{m} \Delta\eta_n = s_{ref} \tag{38}$$

where s_{ref} is a reference value determined according to our desired stage and will be specified for each example problem in Section 6.

5.2. Prediction of two basic measures

Prediction, here, means giving an approximation to a future response of a problem during an analysis based on relevant information of the problem or simplifying assumptions. In this subsection, we propose simple models to predict two basic measures which are applicable in step q (the current step) during an analysis. The basic measures are the total number of iterations for a complete solution, N^{i}, and the weighted average of cosines of linear hyper angles, $\cos\Theta$, which are used to calculate the speed and the cosine measure, respectively. While having a larger P_{sp} needs fewer solution points, a larger P_{cos} imposes more points and thus a smoother numerical solution curve. Therefore, there will theoretically be an optimum set of solution points for each objective performance function composed of the two performance measures. However, finding such set of points requires conducting full analyses of a problem several times, which is rather time consuming. If finding the optimum set of points is not of interest but obtaining a full analysis which fairly balances the solution against the performance measures is desired, a *predict-and-reorient* strategy can be employed. For instance, we predict the future part of the mentioned basic measures in each step, calculate P_{obj}, and find an optimum value of the step-length and use it at the start of the next step.

The prediction model, y_i, is a simple function which allows the calculation of the future part of each basic measure up to the final step. For the future trend/evolution of the step-length used in the constraint function, it is assumed that

$$\Delta\eta_n = \widetilde{\Delta\eta} \quad \text{for} \quad n > q \tag{39}$$

where $\widetilde{\Delta\eta}$ is a fixed step-length for the rest of the steps to the end. With the assumption (39) for s_m (the progress indicator of the final step), the following equation is obtained

$$\widetilde{s_m} = \sum_{n=1}^{q} \Delta\eta_n + y_0 \tag{40}$$

where

$$y_0 = m_r\, \widetilde{\Delta\eta} \tag{41}$$

and $\widetilde{s_m}$ is the prediction of the progress indicator of the final step, y_0 is a prediction of the future part of s_m which is $\sum_{n=q+1}^{m} \Delta\eta_n$, and $m_r = m - q$ is the remaining number of steps to the final analysis step. The prediction of the total number of iterations, $\widetilde{N^{i}}$, is written as

$$\widetilde{N^{i}} = \sum_{n=1}^{q} N^{i}_{n} + y_1 \tag{42}$$

where y_1 is a prediction of the future part of N^{i} which is $\sum_{n=q+1}^{m} N^{i}_{n}$. To propose a prediction model for y_1, it is needed to assume some minimal information: (a) the restarting strategy should be considered in the model, (b) if a determined maximum number of iterations considering a maximum number of restartings (i.e. N^{i}_{ref}) occurs in an increment of an analysis, the solution of that increment is labeled as non-convergent, (c) the mentioned number is proportional to the rest of the progress, and (d) if the total number of iterations of a complete path goes to infinity, the step-length goes to zero, and consequently the increments converge with one iteration. Therefore, every function with a minimum number of parameters which satisfies these assumptions is a candidate for y_1. Here, we propose the following equation

$$y_1 = a_1 m_r + b_1 \exp(-c_1 m_r) \tag{43}$$

where a_1, b_1, and c_1 are the parameters of y_1 and should be updated in each step. To determine these parameters in step q, three pairs of (m_r, y_1) are needed. These pairs are determined according to the minimal assumptions of y_1 : $\left(0, N^i_{ref}\right), \left(1, \left(1 - \frac{s_q}{s_{ref}}\right)N^i_{ref}\right)$, and $(+\infty, m_r)$, respectively, which leads to

$$a_1 = 1 \tag{44}$$

$$b_1 = N^i_{ref} \tag{45}$$

$$c_1 = -\ln\left(\frac{d_1}{b_1}\right) \tag{46}$$

where ln is the natural logarithm. In order to prevent a logarithm with a negative value in the very last analysis steps, we suggest d_1 to be equal to $\left(1 - \frac{s_q}{s_{ref}}\right)N^i_{ref} - a_1$ if it is positive; otherwise, a very small value such as $0.05a_1$ is used for d_1. Fig. 6 shows a schematic curve of y_1. A decrease of y_1 is seen from the figure, which confirms consideration of the restarting strategy in the formulation of y_1. As assumed, $m_r = 0$ means non-convergence which happens when the path-following method is not able to find a solution within the maximum number of iterations per step as well as within the maximum number of restartings. In general, the method converges with fewer restartings and consequently with a smaller total number of iterations as m_r increases from 0. This trend continues until around the point where no restarting occurs (i.e. the minimum point of the curve indicated by $m^*_r = -\frac{1}{c_1}\ln\left(\frac{a_1}{c_1 b_1}\right)$ and $y^*_1 = \frac{a_1}{c_1}\left(1 - \ln\left(\frac{a_1}{c_1 b_1}\right)\right)$. Afterwards, y_1 starts to increase simply because the final number of steps is increased without any restartings. Adopting the assumption (38) for $\cos\Theta$ leads to

$$\widetilde{\cos\Theta} = \frac{1}{s_{ref}}\left(\sum_{n=1}^{q-1}\frac{1}{2}\Delta\eta_n(\cos\Theta_{n-1} + \cos\Theta_n) + \frac{1}{2}\Delta\eta_q\cos\Theta_{q-1} + y_2\right) \tag{47}$$

where y_2 is a prediction of the future part of the weighted sum of cosines of linear hyper angles which is $\sum_{n=q}^{m}\frac{1}{2}\Delta\eta_n(\cos\Theta_{n-1} + \cos\Theta_n) - \frac{1}{2}\Delta\eta_q\cos\Theta_{q-1}$. By the same analogy of proposing y_1, we minimally assume that (a) the minimum value of $\cos\Theta_q$ (which is equal to -1) occurs in case of non-convergence, (b) the linearized solution curve approaches to a linearized curve with $\cos\Theta = \cos\Theta_{ref}$ (which is not equal to its ideal value, +1, because of the machine precision) when the number of increments goes to infinity, and (c) the prediction function should be monotonic with respect to m_r. We propose the following candidate for y_2:

$$y_2 = (s_{ref} - s_q)\left(a_2\cos\left(\left(1 - \frac{s_q}{s_{ref}}\right)\frac{2\pi}{m_r + 2}\right) + c_2\right) \tag{48}$$

where a_2 and c_2 are the parameters of y_2 and are updated in each step. The minimal assumptions of y_2 are interpreted as two pairs of $(m_r, y_2) : (0, -(s_{ref} - s_q))$ and $(+\infty, \cos\Theta_{ref}\cdot(s_{ref} - s_q))$, respectively, which gives us

$$a_2 = \frac{1 + \cos\Theta_{ref}}{1 - \cos\left(\left(1 - \frac{s_q}{s_{ref}}\right)\pi\right)} \tag{49}$$

$$c_2 = \cos\Theta_{ref} - a_2. \tag{50}$$

Fig. 7 draws a schematic illustration of y_2 versus m_r. The curve in the figure shows to possess a value of -1 in case of non-convergence. In addition, approaching to $\cos\Theta_{ref}$ which is not equal to $+1$ expresses that the machine precision as well as the numerical deviation from the exact analytical solution curve are considered in the formulation of y_2.

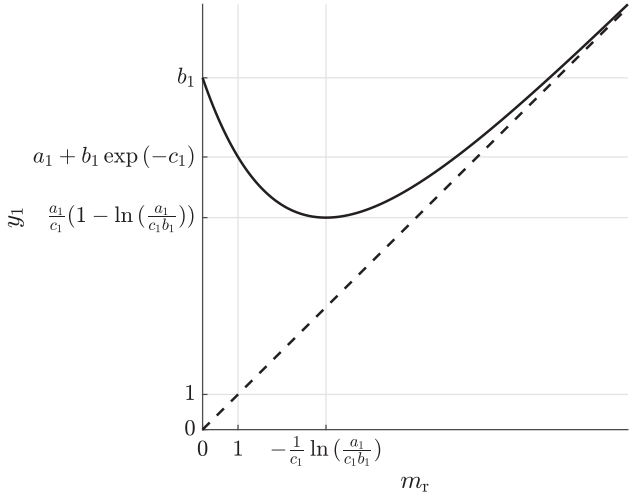


Fig. 6. The prediction value of y_1 starts from N^i_{ref} and decreases to a minimum as m_r increases; then, the curve approaches to the bisector as defined in the minimal assumptions.

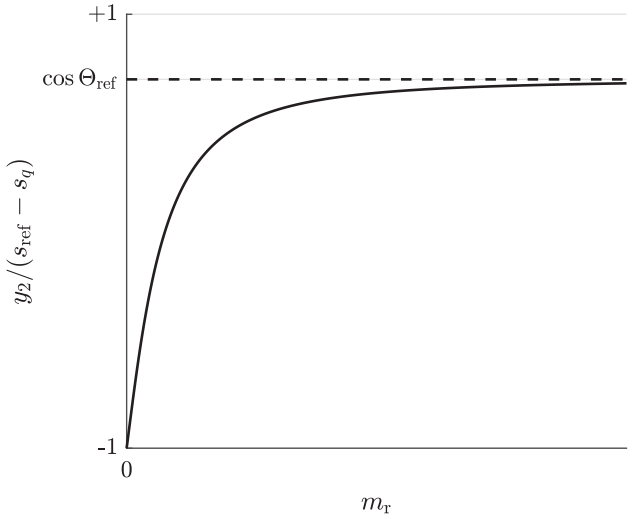


Fig. 7. The prediction value of $y_2/(s_{ref} - s_q)$ starts from its lowest possible value, -1, and monotonically increases to approach to $\cos\Theta_{ref}$ when m_r goes to infinity.

5.3. The global adaption law

The value of the prediction of the objective performance measure, $\widetilde{P_{obj}}$, is calculated according to the prediction of the speed and cosine measures. In other words,

$$\widetilde{P_{obj}} = P_{obj}\left(\widetilde{X_j}; \widetilde{\alpha_j}\right) \tag{51}$$

where

$$\widetilde{X_1} = \frac{N^i_{ref}}{\widetilde{N^i}}, \tag{52}$$

$$\widetilde{X_2} = \frac{\frac{1}{\widetilde{\cos\Theta} - 1} + \frac{1}{2}}{\frac{1}{\cos\Theta_{ref} - 1} + \frac{1}{2}}, \tag{53}$$

and $\widetilde{\alpha_1}$ and $\widetilde{\alpha_2}$ are their corresponding importance factors. To find a suitable step-length for the next step (i.e. step $q+1$), the value of m_r which maximizes $\widetilde{P_{obj}}$ should be found first, i.e. $m^*_r = \arg\left(\max_{m_r}\widetilde{P_{obj}}\right)$. Then, by substituting Eqs. (37), (38), and

(41) into Eq. (40) and reordering, an equation is found by which a step-length corresponding to m_r^* could be calculated:

$$\Delta\eta^* = \frac{s_{ref} - s_q}{m_r^*} \tag{54}$$

where $\Delta\eta^*$ is the optimal value of $\widetilde{\Delta\eta}$ according to the maximization of $\widetilde{P_{obj}}$. Therefore, the *global adaptation law* will be

$$\Delta\eta_{q+1} = \Delta\eta^*. \tag{55}$$

It is significant for this adaptation law that the employed constraint functions to be derivative-free (i.e. the incremental constraint functions can be precisely calculated from the total constraint functions without any need for discretization in artificial time). Otherwise, the progress indicator, Eq. (38), cannot be employed as an equation of finding the optimal step-length in (54). For example, the constraint functions of the modified κ and κ–ω control are derivative-free while the original κ–ω control is not. Again, for the analysts who define the speed measure as a function of CPU time, the calculations solely assigned to the global adaptation law should be considered as a surplus to the computations of the conventional law.

6. Example problems

In the following example problems, the proposed adaptation laws with different sets of parameters are used and their results are compared to the conventional adaptation law (Eq. (33)) according to the objective performance measures. The example problems share the following assumptions:

- It is assumed that the external force is deformation-independent.
- The problems are analyzed for each constraint function and for different sets of adaptation parameters. The constraint functions are energy release (En), modified κ ($M\kappa$), and modified κ–ω ($M\kappa\omega$). For the local adaptation law, the sets of parameters are determined $N_{opt}^{i} = 5, \cos(\xi_{opt}) = 1.0$, and $\beta_1 = 1$ together with $\beta_2 \in \{0, 1, 2, 3\}$ to simulate the influence of the modification factor to the conventional law and $N_{opt}^{i} = 5, \beta_1 = 0$, and $\beta_2 = 1$ together with $\cos(\xi_{opt}) \in \{0.70, 0.80, 0.95\}$ in order to focus on the effectiveness of the modification factor itself. In

Table 1
The parameter sets of the global adaptation law used in the example problems.

Global adaptation law parameters					
P-Type	$\widetilde{\alpha_1}$	$\widetilde{\alpha_2}$	P-Type	$\widetilde{\alpha_1}$	$\widetilde{\alpha_2}$
I	1	1	*II*	1	1
I	1	10	*II*	1	10
I	1	25	*II*	1	25
I	1	50	*II*	1	50
I	1	100	*II*	1	100
I	10	1	*II*	10	1
I	25	1	*II*	25	1
I	50	1	*II*	50	1
I	100	1	*II*	100	1

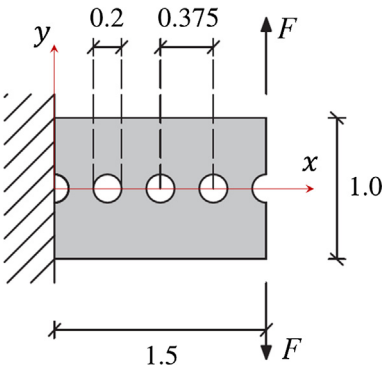


Fig. 8. Perforated beam problem set-up and its dimensions in mm.

Table 2
Characteristics of the perforated beam problem.

Continuum region
- Modulus of elasticity: $E = 100\ \mathrm{N/mm^2}$
- Poisson's ratio: $\nu = 0.30$
- Thickness = 1 mm
- Constitutive law: $\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\varepsilon}$
- Plain strain condition: $\mathbf{C} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)}\begin{bmatrix} 1 & \frac{\nu}{1-\nu} & 0 \\ \frac{\nu}{1-\nu} & 1 & 0 \\ 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix}$
- Element type: 6-node triangular
- Integration scheme: 7-point Gauss

Cohesive zone
- Uniaxial tensile strength: $f_t = 1\ \mathrm{N/mm^2}$
- Mode-I fracture energy per unit area: $G_{f,I} = 2.5 \times 10^{-3}\ \mathrm{N/mm}$
- Damage law: $\boldsymbol{t}_{int} = (1-\omega)\mathbf{C}_{int}\boldsymbol{u}_{int}$
- Damage parameter function: linear damage function of Eq. (3)
- $\boldsymbol{t}_{int}^{T} = [t_s, t_n]$, $\boldsymbol{u}_{int}^{T} = [u_s, u_n]$, and $\mathbf{C}_{int} = \begin{bmatrix} d_s & 0 \\ 0 & d_n \end{bmatrix}$ where subscripts $\square_s$ and $\square_n$ indicate directions tangent and normal to the interface surface in a 2D space, respectively.
- $d_s = 5 \times 10^3\ \mathrm{N/mm^3}$ and $d_n = 10^4\ \mathrm{N/mm^3}$.
- $\kappa_0 = f_t/d_n$ and $\kappa_u = 2G_{f,I}/f_t$
- Element type: 3-node quadratic interface
- Integration scheme: 3-point Newton-Cotes

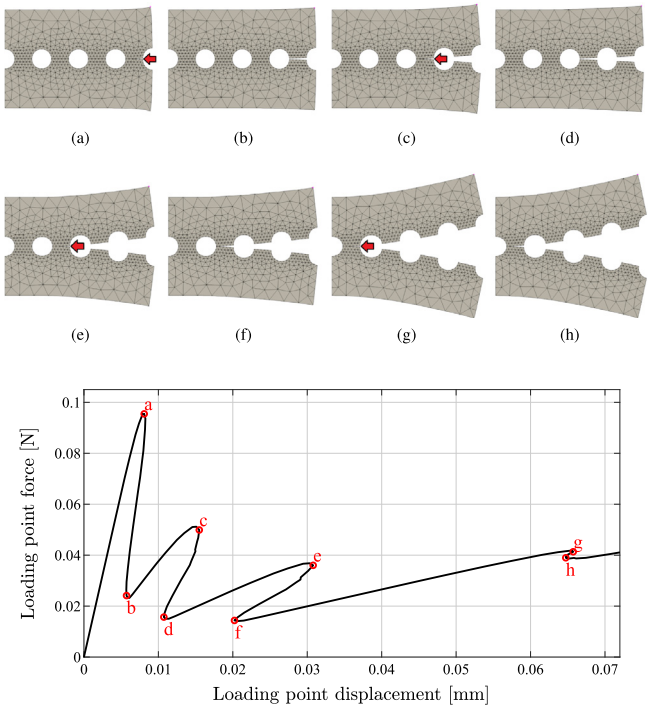
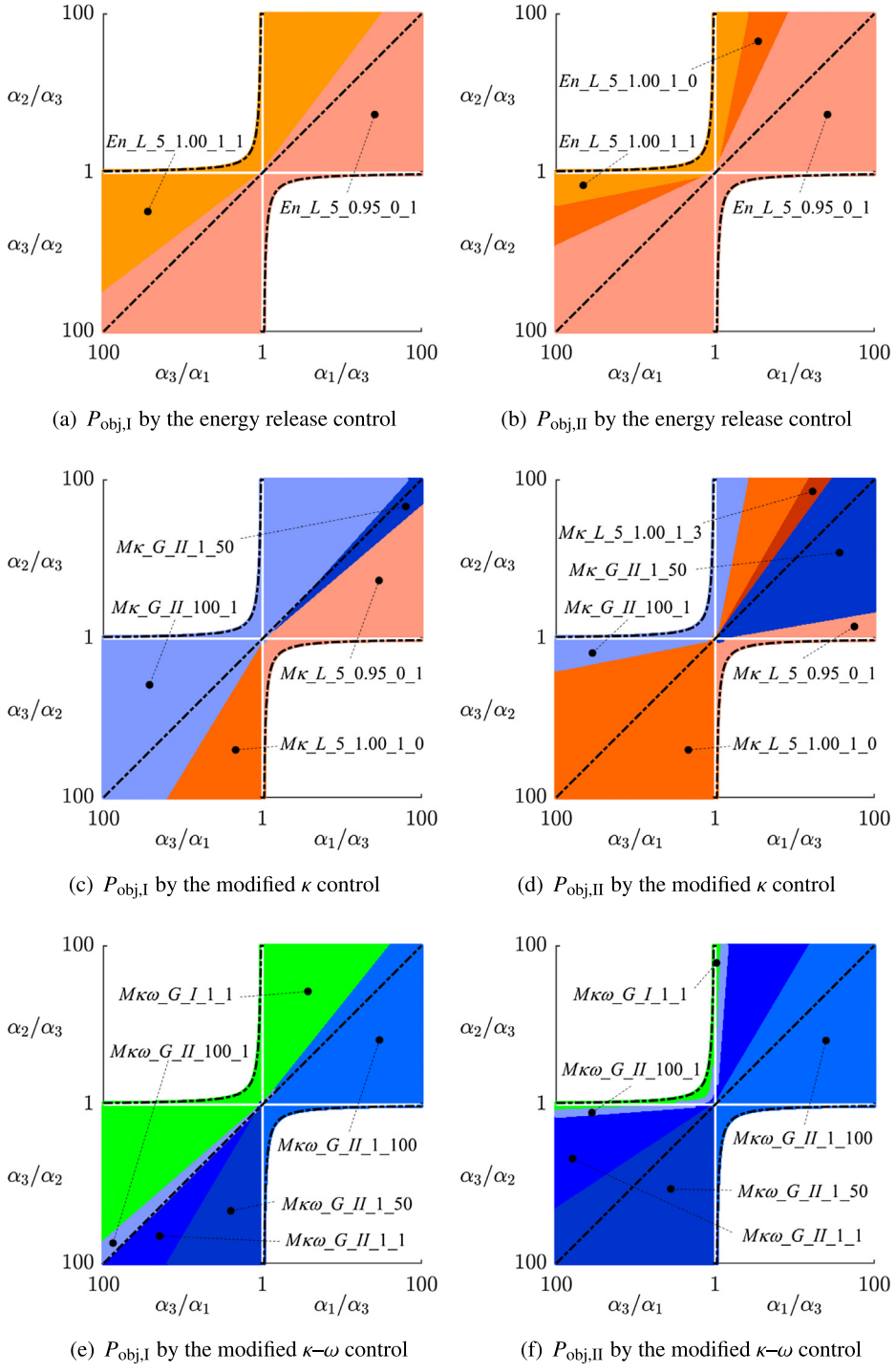


Fig. 9. The interface elements which are responsible for the start of snap-backs are indicated by red arrows in (a), (c), (e), and (f). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

addition, Table 1 lists the sets of parameters employed for the global adaptation law.

- We adopted an abbreviated name for each simulation: '*Symbol*_*L*_N^{i}_{opt}_$\cos(\xi_{\mathrm{opt}})$_β_1_β_2' for the local adaptation law and '*Symbol*_*G*_P – Type_$\widetilde{\alpha_1}$_$\widetilde{\alpha_2}$' for the global adaptation law. It should be emphasized that the energy release constraint function is not derivative-free and, therefore, is not used together with the global adaptation law.
- The analyses are initially started by 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 [2]) and continue to reach the desired stage. Both the desired stage and the threshold value are reported for each example separately.
- A simple adaptation rule is also used for the initial guess of each step after adaptation of the incremental step-length:



(a) $P_{\mathrm{obj,I}}$ by the energy release control

(b) $P_{\mathrm{obj,II}}$ by the energy release control

(c) $P_{\mathrm{obj,I}}$ by the modified κ control

(d) $P_{\mathrm{obj,II}}$ by the modified κ control

(e) $P_{\mathrm{obj,I}}$ by the modified κ–ω control

(f) $P_{\mathrm{obj,II}}$ by the modified κ–ω control

Fig. 10. The space of objective performance measures of the perforated beam example.

$$\begin{bmatrix} \Delta \boldsymbol{a}^0_{n+1} \\ \Delta \lambda^0_{n+1} \end{bmatrix} = \frac{\Delta \eta_{n+1}}{\Delta \eta_n} \begin{bmatrix} \Delta \boldsymbol{a}_n \\ \Delta \lambda_n \end{bmatrix} \tag{56}$$

- Each time non-convergence occurs in an analysis step, the restarting strategy adds 15 to N^i_n and restarts the analysis of the step from adapting the step-length and the initial guess. In case of non-convergence after a predefined maximum number of restartings, the method is labeled 'unacceptable'. This maximum number is a value which scales the incremental step-length down to almost one-hundredth of its value at the start of each step. For example, it will be 27 or 34 for $N^i_{opt} = 4$ or 5.

- If each of the simulations is able to reach a desired stage of the analysis, its result is considered as an acceptable one for further assessment; otherwise, it is assumed as an unacceptable simulation and is excluded from the portfolio of simulations.
- The normalized measures mentioned in Section 3.2 are

$$X_1 = \frac{P_{sm}}{P_{sm\max}}, \quad X_2 = \frac{P_{sp}}{P_{sp\max}}, \quad X_3 = \frac{P_{rob}}{P_{rob\max}} \tag{57}$$

  where $P_{sm\max}, P_{sp_{max}}$, and $P_{rob\max}$ are the maximum of smoothness, speed, and robustness measures among all acceptable simulations, respectively. It should be noted that only the parts of the solution path which are obtained by employing the constraint functions are considered in these calculations with excluding the path initially found by force control.
- 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 Fig. 10 or 14).

6.1. Perforated beam

This example problem is presented in [22] and is a modified version of the numerical example in [2]. The problem considers fracture of a perforated beam (see Fig. 8) by using 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 listed in Table 2. For a smooth transition from force control to the control by the employed constraint functions, a fixed 0.00156 N incremental step-length (which is equal to $\frac{1}{60}$ of the maximum force) is selected for the force control together with 10^{-11} J for the dissipated energy threshold. The desired stage of the perforated beam is defined to be where the vertical displacement of the top loading point is equal to 0.075 mm. It is worth noting that the desired stage might be defined in another way. For instance, by determining a value for the progress indicator of a derivative-free constraint function. This is not recommended for a constraint function which is not derivative-free because the discretization adopted to obtain its incremental form causes a difference between the cumulative sum of step-lengths up to the current stage of the problem and the current state of the constraint kernel. The reference distance D_{ref} and the reference total number of iterations N^i_{ref} are equal to 2.063 mm and 510, respectively. Representative deformed meshes of the perforated beam problem are shown in Fig. 9. In a quasi-static analysis, the dissipation of energy (in the interfaces) requires a decrease of the external load at the snap-backs (i.e. from a, c, e, and g, to b, d, f, and h, respectively).

The performance measure values resulted from employing the adaptation laws are listed in Appendix A. Fig. 10 shows the regions of dominance of the best sets of adaptation parameters for the perforated beam example. It is observed from the figure that

- By using energy release control, the local adaptation law is always better based on $P_{obj,I}$. It also performs better based on $P_{obj,II}$ except in the region $2.14 \leqslant \alpha_2/\alpha_1 \leqslant 5.06$ in which the conventional law is the best.
- By using the κ control, the global adaptation law has been the best based on $P_{obj,I}$ where α_2/α_1 is greater than 1. The local adaptation has performed better in the region $1.17 \leqslant \alpha_1/\alpha_2 \cap 1.0 \leqslant \alpha_1/\alpha_3$; the conventional law has been dominant in $1.65 \leqslant \alpha_1/\alpha_2 \cap 1.0 \leqslant \alpha_3/\alpha_1$. Based on $P_{obj,II}$, the global adaptation has been better for $5.06 \leqslant \alpha_2/\alpha_1$; and, the conventional law has the highest rank for $\alpha_2/\alpha_1 < 5.06 \cap 1.0 \leqslant \alpha_3/\alpha_1$ and for $1.69 \leqslant \alpha_2/\alpha_1 \leqslant 5.06 \cap 1.0 \leqslant \alpha_1/\alpha_3$. In the region of $1.0 \leqslant \alpha_1/\alpha_3$, the local adaptation law has outperformed for $1.36 \leqslant \alpha_2/\alpha_1 \leqslant 1.69$ and for $5.80 \leqslant \alpha_1/\alpha_2$. In this region, the global adaptation law has owned another dominance subregion of $0.20 \leqslant \alpha_1/\alpha_2 \leqslant 5.80$.

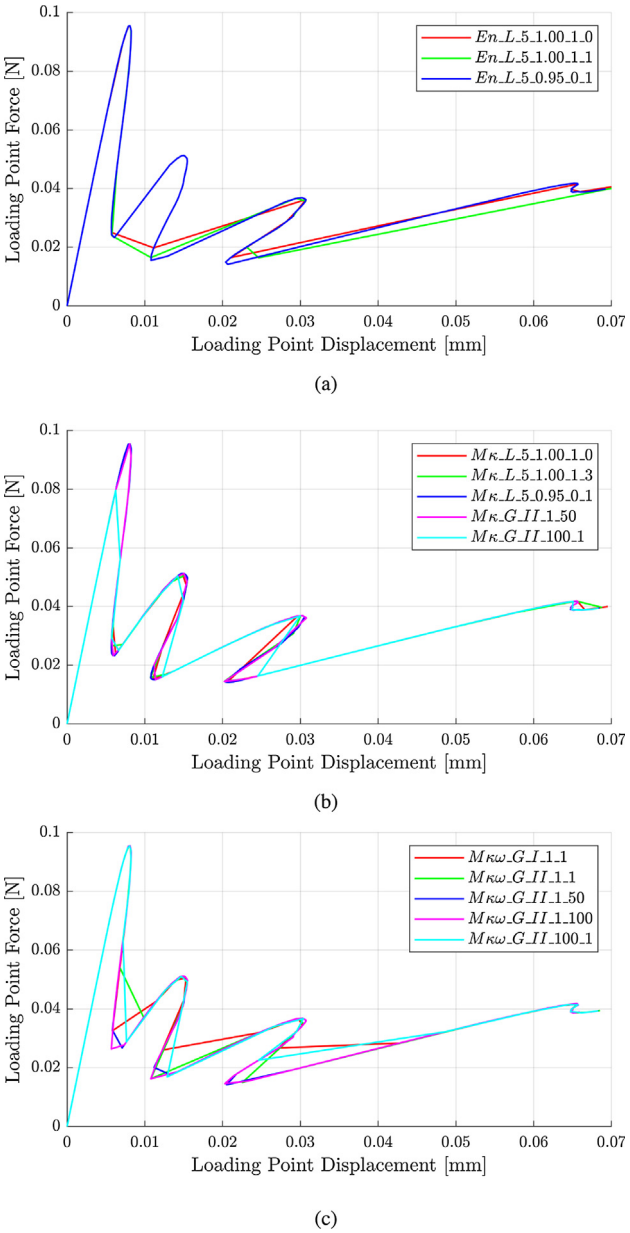


Fig. 11. The loading point displacement-force curve of the dominant solutions found by using (a) the energy release, (b) the modified κ, and (c) the modified κ-ω control for the perforated beam problem.

- By using the modified κ–ω control, the global adaptation has been the only dominant law. However, the dominance regions are very different based on $P_{obj,I}$ and $P_{obj,II}$.
- The global adaptation law with $P_{obj,II}$ has been more effective than $P_{obj,I}$. By using $P_{obj,II}$, the step-length is adapted to keep the normalized speed and cosine measures almost equal; in contrast, by using $P_{obj,I}$, the step-length tends to a very large or small value in order to maximize $P_{obj,I}$.
- The proposed adaptation laws have been really competitive to the conventional law, specially for the modified κ–ω control.

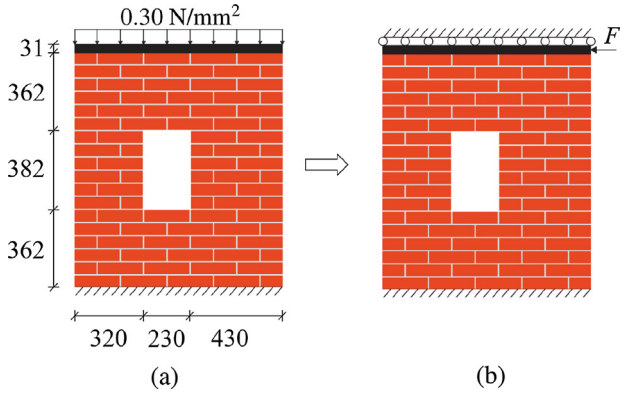


Fig. 12. Confined masonry wall problem set-up and its dimensions in mm. (a) phase I: confining by a compressive load and (b) phase II: imposing a lateral load.

Table 3
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: $E = 16700\ \mathrm{N/mm^2}$
- Poisson's ratio: $\nu = 0.15$
- Constitutive law: $\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\varepsilon}$
- Plain stress condition: $\mathbf{C} = \frac{E}{1-\nu}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}$
- Element type: 4-node rectangular
- Integration scheme: 4-point Gauss

Mortar joints
- Dimensions: h_{int} = 10 mm and thickness = 100 mm
- Uniaxial tensile strength: $f_t = 0.25\ \mathrm{N/mm^2}$
- Uniaxial compressive strength: $f_c = 10.5\ \mathrm{N/mm^2}$
- Poisson's ration: $\nu_{int} = 0.14$
- Mode-I fracture energy per unit volume: $G_{f,I} = 0.018\ \mathrm{N/mm^2}$
- Damage law: $\boldsymbol{t}_{int} = (1-\omega)\mathbf{C}_{int}\boldsymbol{u}_{int}$
- Damage parameter function: exponential damage function of Eq. (4)
- $\boldsymbol{t}_{int}^{T} = [t_s, t_n]$, $\boldsymbol{u}_{int}^{T} = [u_s, u_n]$, and $\mathbf{C}_{int} = \begin{bmatrix} d_s & 0 \\ 0 & d_n \end{bmatrix}$ where subscripts $\square_s$ and $\square_n$ indicate directions tangent and normal to the interface surface in a 2D space, respectively.
- $d_s = 36\ \mathrm{N/mm^3}$ and $d_n = 82\ \mathrm{N/mm^3}$
- $\kappa_0 = f_t/d_n$ and $\beta_{int} = \frac{h_{int}G_{f,I}}{f_t} - \frac{1}{2}\kappa_0$
- Equivalent relative displacement, u_{eq}, is defined by a degenerated capped Drucker-Prager model [23]:

$$u_{eq} = \begin{cases} AI_u + BJ_u, & J_u \geqslant \frac{A-C}{D-B}I_u \\ CI_u + DJ_u, & J_u \leqslant \frac{A-C}{D-B}I_u \end{cases}$$

where $I_u = (1+\nu_{int})u_n$,

$$J_u = \frac{1}{1+\nu_{int}}\sqrt{\tfrac{1}{3}(1+\nu_{int}+\nu_{int}^3+\nu_{int}^4)u_n^2 + \tfrac{1}{4}u_s^2},$$

$A = \frac{1}{2}\frac{f_c-f_t}{f_c}$, $B = \frac{\sqrt{3}}{2}\frac{f_c+f_t}{f_c}$, $C = \frac{(f_b-f_c)f_t}{f_b f_c}$, and $D = \sqrt{3}\frac{(2f_b+f_c)f_t}{f_b f_c}$

- Biaxial compressive strength: $f_b = 1.2f_c$
- Element type: 2-node linear interface
- Integration scheme: 2-point Newton-Cotes

Fig. 11 depicts the curve of the loading point displacement versus force for the perforated beam example. Note that the colors of the curves are chosen for a clear distinction between them and should not be confused with the consistent colors used for the space of importance ratios in Fig. 10. Solely, the results which have performed best in a region of the space of importance ratios are shown in the figure. It is seen from the curves that (a) only the smoothest local adaptation law has been able to trace the second snap-back when using the energy release control, (b) the fastest global adaptation law has skipped a part of the first, second, and third snap-backs by the modified κ control, and (c) the smoothest path belongs to the global adaptation by the modified κ–ω control which has followed the end parts of the snap-backs quite well.

6.2. Masonry wall

In this example, which is a slightly changed version of that explained in [23], we discuss the quasi-static response of a masonry wall preconfined with a vertical compressive load in phase I and subjected to a lateral in-plane force at its top in phase II (see Fig. 12). Unlike the wall in [23], 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. Damage is assumed to only occur within the mortar material by adopting mode-I fracture along the centerline

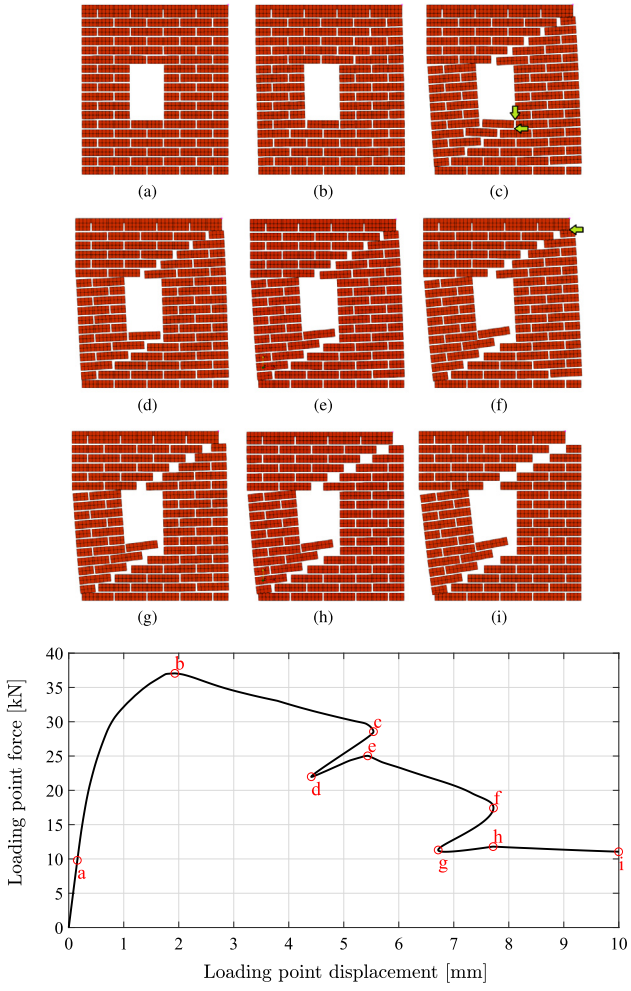
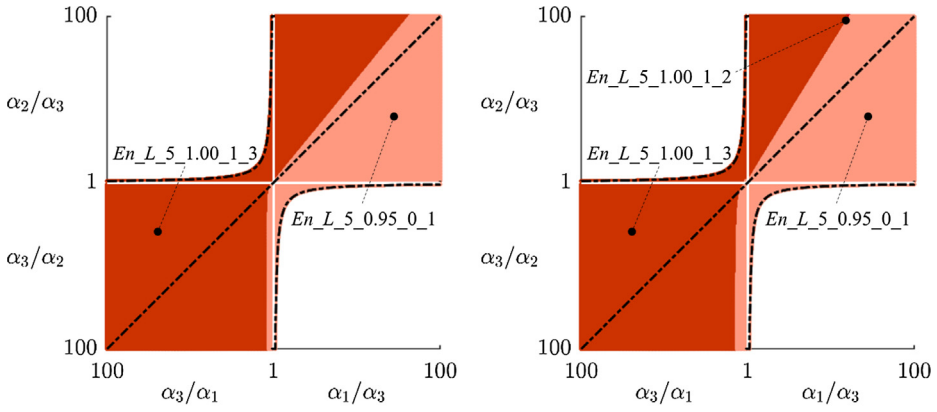


Fig. 13. The interface elements which are responsible for the start of snap-backs are indicated by green arrows in (c) and (f). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

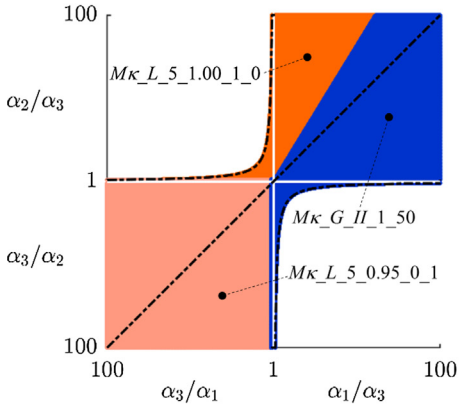
of mortar joints while the bricks remain elastic. Consequently, non-zero-width joints are modeled by interface elements located in their centerlines [20]. Table 3 summarizes the properties of bricks and mortar joints. For simulation purposes, a very stiff bar is added on top of the wall so that the bar transfers the concentrated load to the top nodes of the wall.

The analysis is started by using force control with a fixed incrementation of 297.24 N (which is equal to $\frac{1}{125}$ of the maximum force) and a switching threshold of 0.001 J in order to smoothly switch to the employed constraint functions. After it switched to one of the mentioned control constraints, the analysis proceeded to pass the desired stage which was defined by the absolute value of the horizontal displacement of the top right node to be equal to 1 cm. The reference distance and the reference total number of iterations are equal to 456.1 mm and 510, respectively. Deformed meshes of the masonry wall problem at representative steps are shown in Fig. 13. Like previous example, a decrease of the external load in the snap-backs (i.e. from c and f, to d and g, respectively) is required for energy dissipation (in the interfaces) in a quasi-static analysis.

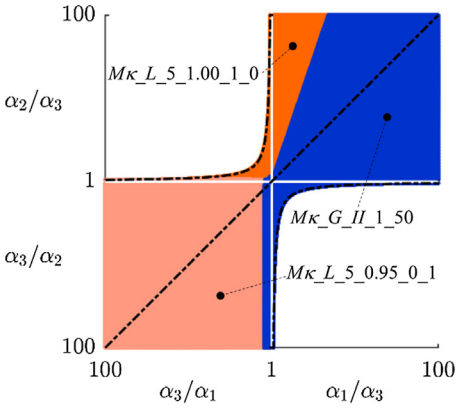


(a) $P_{obj,I}$ by the energy release control

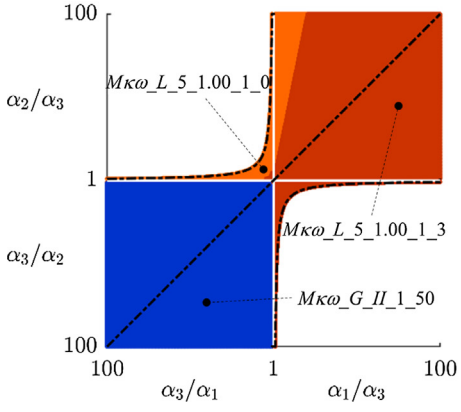
(b) $P_{obj,II}$ by the energy release control



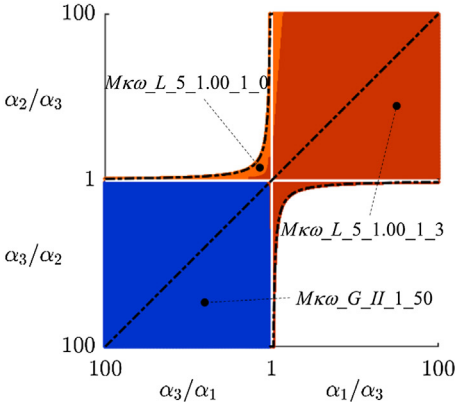
(c) $P_{obj,I}$ by the modified κ control



(d) $P_{obj,II}$ by the modified κ control



(e) $P_{obj,I}$ by the modified κ–ω control



(f) $P_{obj,II}$ by the modified κ–ω control

Fig. 14. The space of objective performance measures of the masonry wall example.

Results of employing the adaptation laws are summarized by the performance measures in Appendix A. It is observed from Fig. 14 (which shows the regions of dominance of the best sets of adaptation parameters), that

- The local adaptation law is the best when using the energy release control.
- By utilizing the κ control, the global adaptation law has outperformed in the region of (a) $1.0 \leqslant \alpha_1/\alpha_3 \cap \alpha_2/\alpha_1 \leqslant 1.63$ based on $P_{\text{obj,I}}$ and (b) $1.0 \leqslant \alpha_1/\alpha_3 \cap \alpha_2/\alpha_1 \leqslant 2.96$ based on $P_{\text{obj,II}}$; the conventional law is better in the region of (a) $1.0 \leqslant \alpha_2/\alpha_3 \cap 1.63 \leqslant \alpha_2/\alpha_1$ based on $P_{\text{obj,I}}$ and (b) $1.0 \leqslant \alpha_2/\alpha_3 \cap 2.96 \leqslant \alpha_2/\alpha_1$ based on $P_{\text{obj,II}}$; and the local adaptation law is almost dominant in the region of $1.0 \leqslant \alpha_3/\alpha_1 \cap 1.0 \leqslant \alpha_3/\alpha_2$.

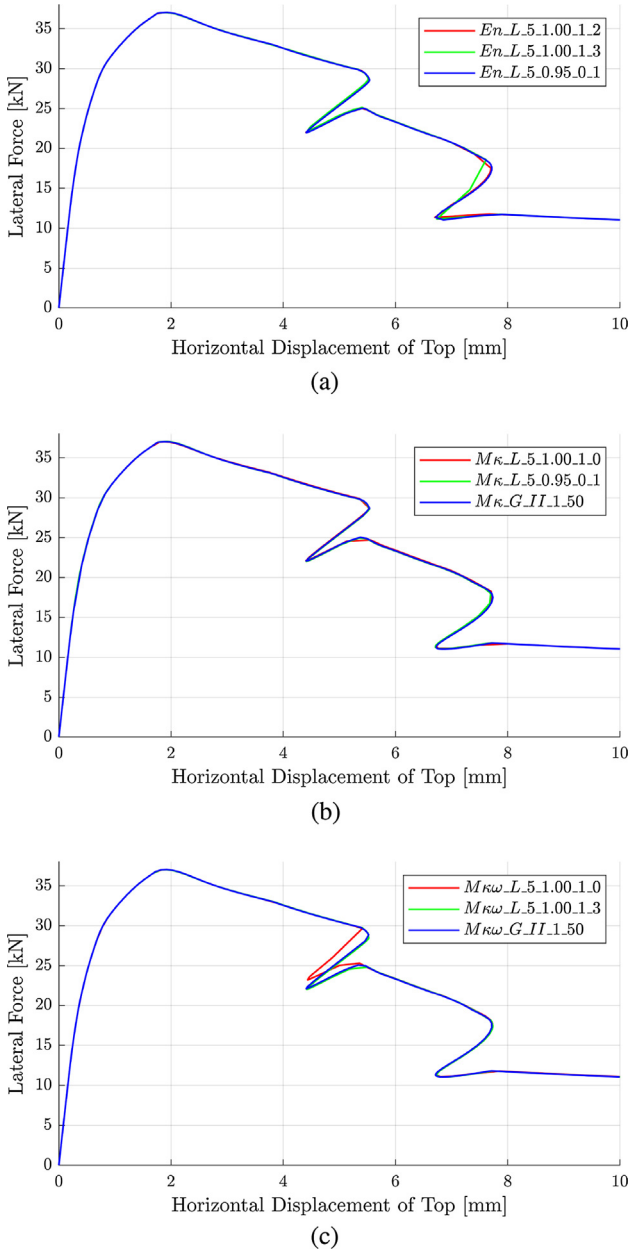


Fig. 15. The loading point displacement-force curve of the dominant solutions found by using (a) the energy release, (b) the modified κ, and (c) the modified κ-ω control for the masonry wall problem.

- In the region of $1.0 \leqslant \alpha_3/\alpha_1 \cap 1.0 \leqslant \alpha_3/\alpha_2$, the global adaptation law has the first rank among the others by the modified κ–ω control. While, in the rest of the regions, the local and conventional laws are the best based on (a) $P_{\text{obj,I}}$ where $\alpha_2/\alpha_1 \leqslant$ and $\geqslant 4.94$ and (b) $P_{\text{obj,II}}$ where $\alpha_2/\alpha_1 \leqslant$ and $\geqslant 12.90$, respectively.
- The outperforming parameter sets does not vary by the objective performance measure except the analysis *En_L_5_1.00_1_2* which is added in the very limited region of $1.0 \leqslant \alpha_1/\alpha_3 \cap 1.62 \leqslant \alpha_2/\alpha_1 \leqslant 1.70$ based on $P_{\text{obj,II}}$.
- The proposed adaptation laws are functioning quite well compared to the conventional law.

Like the previous example problem, the loading point displacement-force curves of the best performing results of the masonry wall are shown in Fig. 15. Different colors from the space of importance ratios are chosen for the sake of visual clarity. One can see from the curves that (a) one of the results by the local adaptation has skipped a small part of the second snap-back by utilizing the energy release control, (b) all of the dominant results have smoothly traced the snap-backs by the modified κ control, and c) the conventional law has started the snap-back a little before the other ones by the modified κ-ω control.

7. Summary and conclusions

In a quasi-static analysis of structures using a path-following method, determination of the step-length in each analysis increment is necessary to improve the performance of the method. We proposed two new step-length adaptation laws: a local and a global law. The former locally considers the adaptation and adds a multiplier as a modification to a conventional law in order to enhance the smoothness of the solution curve. The latter is based on the global performance of the numerical solution and adopts prediction models to be able to determine the step-length in each analysis step. The results of comparisons between the new and the conventional adaptation laws based on objective performance measures show that:

- As intended, the local adaptation modification factor has increased the smoothness of solution when it is solely used. However, if the modification factor is multiplied to the conventional law, the step-length increases in some steps instead of decreasing. The reason for this is that if the modification factor significantly decreases the step-length calculated by the conventional law in an analysis step, that step converges with a significantly smaller number of iterations which sharply increases the step-length of the next step. Thus, in problems with sharp variations of the modification factor, a decrease in the smoothness might be observed when multiplying the modification factor to the step-length calculated by the conventional law.
- The global adaptation law has controlled the step-length well. By assigning more importance to smoothness or speed, the adaptation law has managed to increase the smoothness or speed measure of the solution, respectively.
- The lower and upper bounds which are imposed to restrict the step-length in practice as well as the restarting strategy affect the results of the performance measures of a solution. They, thus, may slightly change the overall performances which are expected from the adaptation laws.

It should be noted that the example problems have been chosen to explore the efficiency areas of the studied and proposed constraint functions and step-length adaptation laws. 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 variations of the problem features.

Appendix A. Performance measures resulted from the example problems

The values of the performance measures of the acceptable results by employing the adaptation laws are listed in Tables A.1, A.2, and A.3 for the perforated beam as well as in Tables A.4, A.5, and A.6 for the masonry wall example problem. In the tables, the numbers of each column should be multiplied to the number inside of bracket to obtain the value of objective performance measure of that column. For example, if the header of a column shows $P_{sm}\left[\times 10^{+3}\right]$, and x is a number in that column, the value of P_{sm} is equal to $x \times 10^{+3}$.

Table A.1
The performance measures are calculated for the acceptable results of the perforated beam example which is controlled by the energy release constraint.

Analysis name	$P_{sm}\left[\times 10^{+3}\right]$	$P_{sp}\left[\times 10^{-3}\right]$	P_{rob}
En_L_5_1.00_1_0	5.635	9.091	1.00
En_L_5_1.00_1_1	3.322	10.101	1.00
En_L_5_1.00_1_2	5.667	8.264	0.50
En_L_5_1.00_1_3	7.988	5.988	1.00
En_L_5_0.70_0_1	10.698	6.098	1.00
En_L_5_0.80_0_1	22.234	3.344	0.33
En_L_5_0.95_0_1	70.689	2.793	1.00

Table A.2
The performance measures are calculated for the acceptable results of the perforated beam example which is controlled by the modified κ control.

Analysis name	$P_{sm}\left[\times 10^{+3}\right]$	$P_{sp}\left[\times 10^{-3}\right]$	P_{rob}
Mκ_L_5_1.00_1_0	7.734	6.135	1.00
Mκ_L_5_1.00_1_1	5.917	4.274	0.50
Mκ_L_5_1.00_1_2	8.636	4.717	0.50
Mκ_L_5_1.00_1_3	14.363	4.292	0.33
Mκ_L_5_0.70_0_1	4.511	3.268	0.25
Mκ_L_5_0.80_0_1	7.516	3.802	1.00
Mκ_L_5_0.95_0_1	44.967	0.916	0.14
Mκ_G_I_1_25	3.350	5.917	0.50
Mκ_G_I_1_50	4.054	5.208	0.33
Mκ_G_I_1_100	4.095	4.926	0.33
Mκ_G_I_10_1	2.970	6.849	1.00
Mκ_G_I_25_1	3.292	6.250	0.50
Mκ_G_I_50_1	3.292	6.250	0.50
Mκ_G_I_100_1	2.858	7.463	1.00
Mκ_G_II_1_1	3.552	5.128	0.33
Mκ_G_II_1_10	12.924	1.916	0.06
Mκ_G_II_1_25	7.568	1.742	0.06
Mκ_G_II_1_50	39.037	2.070	0.13
Mκ_G_II_1_100	29.639	1.675	0.09
Mκ_G_II_10_1	3.126	5.025	0.25
Mκ_G_II_25_1	3.274	6.211	0.50
Mκ_G_II_50_1	3.282	6.250	0.50
Mκ_G_II_100_1	2.858	7.463	1.00

Table A.3
The performance measures are calculated for the acceptable results of the perforated beam example which is controlled by the modified κ–ω control.

Analysis name	$P_{sm}\left[\times 10^{+3}\right]$	$P_{sp}\left[\times 10^{-3}\right]$	P_{rob}
Mκω_L_5_1.00_1_0	7.243	4.016	1.00
Mκω_L_5_1.00_1_1	7.804	3.497	1.00
Mκω_L_5_1.00_1_2	10.264	3.345	0.50
Mκω_L_5_1.00_1_3	7.877	3.436	0.50
Mκω_L_5_0.70_0_1	3.057	4.926	1.00
Mκω_L_5_0.80_0_1	3.230	4.149	1.00
Mκω_L_5_0.95_0_1	37.262	1.675	0.50
Mκω_G_I_1_1	2.770	5.682	1.00
Mκω_G_I_1_10	3.505	5.376	1.00
Mκω_G_I_1_25	5.199	5.465	1.00
Mκω_G_I_1_50	3.695	5.076	1.00
Mκω_G_I_1_100	2.687	5.000	1.00
Mκω_G_I_10_1	3.620	5.618	1.00
Mκω_G_I_25_1	3.620	5.618	1.00
Mκω_G_I_50_1	3.620	5.618	1.00
Mκω_G_I_100_1	3.620	5.618	1.00
Mκω_G_II_1_1	6.092	5.405	1.00
Mκω_G_II_1_10	9.826	3.311	1.00
Mκω_G_II_1_25	15.200	2.381	1.00
Mκω_G_II_1_50	30.332	1.866	1.00
Mκω_G_II_1_100	65.587	1.323	0.33
Mκω_G_II_10_1	3.652	5.556	1.00
Mκω_G_II_25_1	3.820	5.525	1.00
Mκω_G_II_50_1	3.526	5.618	1.00
Mκω_G_II_100_1	3.620	5.618	1.00

Table A.4
The performance measures are calculated for the acceptable results of the masonry wall example which is controlled by the energy release constraint.

Analysis name	$P_{sm}\left[\times 10^{+3}\right]$	$P_{sp}\left[\times 10^{-3}\right]$	P_{rob}
En_L_5_1.00_1_0	0.089	2.370	0.17
En_L_5_1.00_1_1	0.151	2.577	0.20
En_L_5_1.00_1_2	0.208	2.294	0.20
En_L_5_1.00_1_3	0.138	2.907	0.33
En_L_5_0.70_0_1	0.081	2.169	0.17
En_L_5_0.80_0_1	0.103	1.949	0.20
En_L_5_0.95_0_1	1.032	0.849	0.25

Table A.5
The performance measures are calculated for the acceptable results of the masonry wall example which is controlled by the modified κ control.

Analysis name	$P_{sm}\left[\times 10^{+3}\right]$	$P_{sp}\left[\times 10^{-3}\right]$	P_{rob}
Mκ_L_5_1.00_1_0	2.871	2.710	0.33
Mκ_L_5_1.00_1_1	0.536	2.632	0.33
Mκ_L_5_0.80_0_1	0.176	2.370	0.25
Mκ_L_5_0.95_0_1	1.083	1.370	1.0
Mκ_G_I_1_10	0.193	1.992	0.06
Mκ_G_I_1_100	0.397	2.359	0.17
Mκ_G_I_10_1	0.199	1.984	0.05
Mκ_G_II_1_1	0.442	1.812	0.05
Mκ_G_II_1_10	1.167	2.037	0.25
Mκ_G_II_1_25	1.049	1.493	0.20
Mκ_G_II_1_50	31.128	1.203	0.50
Mκ_G_II_1_100	5.228	0.001	0.50
Mκ_G_II_10_1	0.717	1.764	0.04
Mκ_G_II_25_1	0.071	2.155	0.05
Mκ_G_II_100_1	0.051	2.174	0.05

Table A.6
The performance measures are calculated for the acceptable results of the masonry wall example which is controlled by the modified κ–ω control.

Analysis name	P_{sm} $\left[\times 10^{+3}\right]$	P_{sp} $\left[\times 10^{-3}\right]$	P_{rob}
Mκω_L_5_1.00_1_0	0.104	2.833	0.50
Mκω_L_5_1.00_1_1	0.150	2.591	0.50
Mκω_L_5_1.00_1_2	0.391	2.433	0.50
Mκω_L_5_1.00_1_3	1.666	2.294	0.50
Mκω_L_5_0.80_0_1	0.388	2.141	0.20
Mκω_L_5_0.95_0_1	0.590	1.072	0.50
Mκω_G_I_1_1	0.160	1.502	0.03
Mκω_G_I_1_10	0.324	1.866	0.05
Mκω_G_I_1_50	0.311	2.105	0.10
Mκω_G_I_1_100	0.227	1.894	0.17
Mκω_G_I_10_1	0.084	1.376	0.03
Mκω_G_I_50_1	0.050	1.815	0.05
Mκω_G_II_1_1	0.338	2.037	0.07
Mκω_G_II_1_10	0.155	1.980	0.33
Mκω_G_II_1_25	0.656	1.300	0.25
Mκω_G_II_1_50	0.435	1.212	1.00
Mκω_G_II_1_100	0.838	0.001	0.50
Mκω_G_II_10_1	0.066	1.439	0.03
Mκω_G_II_25_1	0.063	1.764	0.04

References

[1] Crisfield MA. A fast incremental-iterative solution procedure that handles snap-through. Comput Struct 1981;13(1–3):55–62. https://doi.org/10.1016/0045-7949(81)90108-5.

[2] Gutiérrez MA. Energy release control for numerical simulations of failure in quasi-brittle solids. Commun Numer Methods Eng 2004;20(1):19–29. https://doi.org/10.1002/cnm.649.

[3] Ramm E. Strategies for tracing the nonlinear response near limit points. In: Nonlinear finite element analysis in structural mechanics. Berlin, Heidelberg: Springer; 1981. p. 63–89.

[4] Bergan PG, Horrigmoe G, Bråkeland B, Søreide TH. Solution techniques for non-linear finite element problems. Int J Numer Meth Eng 2005;12(11):1677–96. https://doi.org/10.1002/nme.1620121106.

[5] Bergan PG. Solution algorithms for nonlinear structural problems. Comput Struct 1980;12(4):497–509. https://doi.org/10.1016/0045-7949(80)90125-X.

[6] Widjaja BR. Path-following technique based on residual energy suppression for nonlinear finite element analysis. Comput Struct 1998;66(2):201–9. https://doi.org/10.1016/S0045-7949(97)00071-0.

[7] Chan Siu L. Geometric and material non-linear analysis of beam-columns and frames using the minimum residual displacement method. Int J Numer Meth Eng 2005;26(12):2657–69. https://doi.org/10.1002/nme.1620261206.

[8] Eriksson A. On some path-related measures for nonlinear structural F.E. problems. Int J Numer Meth Eng 1988;26(8):1791–803. https://doi.org/10.1002/nme.1620260808.

[9] Eriksson A. On linear constraints for Newton-Raphson corrections and critical point searches in structural F.E. problems. Int J Numer Meth Eng 1989;28(6):1317–34. https://doi.org/10.1002/nme.1620280607.

[10] Eriksson A. Derivatives of tangential stiffness matrices for equilibrium path descriptions. Int J Numer Meth Eng 1991;32(5):1093–113. https://doi.org/10.1002/nme.1620320511.

[11] Eriksson A. On improved predictions for structural equilibrium path evaluations. Int J Numer Meth Eng 1993;36(2):201–20. https://doi.org/10.1002/nme.1620360203.

[12] Eriksson A, Kouhia R. On step size adjustments in structural continuation problems. Comput Struct 1995;55(3):495–506. https://doi.org/10.1016/0045-7949(95)98875-q.

[13] Schmidt WF. Adaptive step size selection for use with the continuation method. Int J Numer Meth Eng 1978;12(4):677–94. https://doi.org/10.1002/nme.1620120411.

[14] Georg K. A note on stepsize control for numerical curve following. US, Boston, MA: Springer; 1983. https://doi.org/10.1007/978-1-4613-3572-6_8. pp. 145–154.

[15] Den Heijer C, Rheinboldt WC. On steplength algorithms for a class of continuation methods. SIAM J Numer Anal 1981;18(5):925–48. , http://www.jstor.org/stable/2157137.

[16] Schwetlick H. On the choice of steplength in path following methods. ZAMM - J Appl Math Mech/Zeitschrift für Angewandte Mathematik und Mechanik 1984;64(9):391–6. https://doi.org/10.1002/zamm.19840640903.

[17] Szyszkowski W, Husband JB. Curvature controlled arc-length method. Comput Mech 1999;24(4):245–57. https://doi.org/10.1007/s004660050513.

[18] Allgower EL, Georg K. Numerical continuation methods: an introduction. Berlin: Springer; 1990.

[19] Kouhia R, Mikkola M. Some aspects on efficient path-following. Comput Struct 1999;72(4):509–24. https://doi.org/10.1016/S0045-7949(98)00336-8.

[20] Fayezioghani A, Vandoren B, Sluys LJ. A posteriori performance-based comparison of three new path-following constraints for damage analysis of quasi-brittle materials. Comput Methods Appl Mech Eng 2019;346:746–68. https://doi.org/10.1016/j.cma.2018.09.014. , http://www.sciencedirect.com/science/article/pii/S0045782518304584.

[21] Keller H. The bordering algorithm and path following near singular points of higher nullity. SIAM J Scient Stat Comput 1983;4(4):573–82. https://doi.org/10.1137/0904039.

[22] Vandoren B, De Proft K, Simone A, Sluys LJ. A novel constrained large time increment method for modelling quasi-brittle failure. Comput Methods Appl Mech Eng 2013;265:148–62. https://doi.org/10.1016/j.cma.2013.06.005.

[23] Vandoren B, De Proft K, Simone A, Sluys LJ. Mesoscopic modelling of masonry using weak and strong discontinuities. Comput Methods Appl Mech Eng 2013;255:167–82. https://doi.org/10.1016/j.cma.2012.11.005.