

An alternative implementation of the incremental energy/dissipation based arc-length control method

İ. Özdemir

İzmir Institute of Technology, Department of Civil Engineering, Gülbahçe Kampüsü, Urla, İzmir, Turkey

ARTICLE INFO

Keywords:

Arc-length method
Path following
Cohesive fracture
Dissipation

ABSTRACT

A robust solution algorithm is essential to trace the arduous equilibrium paths typically confronted with in cohesive fracture and continuum damage mechanics of quasi-brittle materials. Although robust arc-length type solvers exist suitable for such problems, the use of these methods is hindered by their non-standard implementation requirements. Departing from this fact, in this paper, the recently proposed arc-length solver presented in reference May et al. (2016) is reconsidered within the limitations/capabilities of the commercial software packages and recast in a form which is suitable for implementation through user element formalism. The constraint equation is re-expressed and appended to the system of equations through the internal force column and tangent stiffness matrix of a user element. The effectiveness of the proposed alternative implementation is illustrated by means of two cohesive fracture problems.

1. Introduction

Prediction of fracture induced structural failure can be addressed through different approaches ranging from linear elastic fracture mechanics to computational techniques such as cohesive fracture formulations and the extended finite element method (XFEM).

Continuous (e.g. continuum damage mechanics), discontinuous (e.g. cohesive zones) and combined models of failure all require robust numerical solution algorithms to trace the complete equilibrium path. Due to localization of deformation and bifurcation, the equilibrium path might have limit points and (multiple) snap-backs which cannot be traced by purely force or purely displacement controlled Newton–Raphson solution procedures.

In fact the limitations of purely displacement based or purely force based incremental solution algorithms first appeared in the non-linear analysis of buckling prone elastic structures. To overcome the limit points in aforementioned type analysis, arc-length methods were proposed by references [2,3]. The essential idea was to treat the incremental load step as an additional unknown and augment the system of equilibrium equations by an additional constraint equation. These initial ideas were modified and recast in suitable forms for the finite element based analysis of structural systems by references [4,5].

The additional constraint equation was probably the most decisive factor to be able to trace the complete equilibrium path. In a discrete setting, the typical constraint equations were considering all the degrees of freedom (dofs) with equal weights and this hampered the

success of the method particularly for physically non-linear problems. In such problems, the deformation localizes within a relatively small part of the problem domain and the associated ‘critical’ dofs are masked by the large number of dofs representing the rest of the structure, see reference [6]. To address this deficiency, in reference [7] the author proposed the indirect displacement control method and could analyze crack propagation problems successfully. However, the critical dofs associated with the localization zone should be identified beforehand which might not be an easy task for a large number of localization problems.

The crux of the matter in fact was to use a monotonically increasing function as the constraint equation. Resting on the firm grounds of thermodynamics, reference [8] proposed to use dissipation associated with the structure as the constraint equation for physically non-linear problems. The capacity of the method was successfully demonstrated by selected benchmark problems of continuum damage mechanics. The method was further extended for plasticity and geometrically non-linear damage mechanics problems in reference [9] and effectively used by others in different problems, e.g. composite fracture and delamination problems, see e.g. references [10,11]. However, it has to be noted that the proposed constraint is activated when the dissipation reaches a critical level and therefore requires a switching mechanism between an initial force controlled phase and the dissipation controlled phase. Recently, a more versatile, closely related method was proposed in reference [1], which is based on internal energy and dissipation energy rates such that when there is practically no dissipation or it is lower

E-mail address: izzetozdemir@iyte.edu.tr.

<https://doi.org/10.1016/j.tafmec.2019.01.007>

Received 24 November 2018; Received in revised form 31 December 2018; Accepted 7 January 2019

Available online 08 January 2019

0167-8442/ © 2019 Elsevier Ltd. All rights reserved.

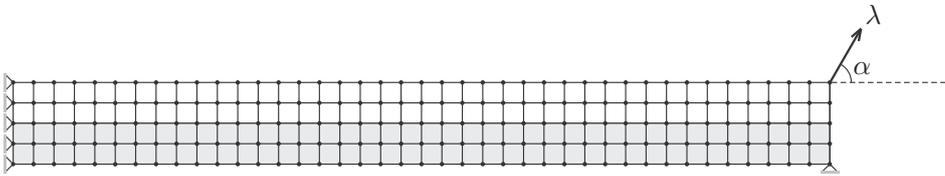


Fig. 1. Discretized two-layer (white and grey) structure under loading at a single node. Zero thickness cohesive elements are inserted along the interface of the two layers.

than a threshold, an internal energy based constraint function is used. Otherwise, the constraint function controls the amount of dissipated energy within the increment.

As far as implementation of arc-length solvers geared towards physically non-linear problems is concerned, it turns out that some manipulation at the solver level is necessary since the system of equations has to be augmented by the additional constraint equation. To achieve this in an efficient and the least costly way (in terms of solution of system of equations), both in references [9,1] the Sherman-Morrison formula is used. This formula is an effective way of computing the inverse of a matrix which can be expressed as the sum of an invertible square matrix and the outer product of two vectors. After re-writing the system of equations (see Appendix 1 of reference [1]), Sherman-Morrison formula can be used to solve them effectively without destroying the sparsity of the stiffness matrix of the structure.

However, these solvers would be accessible to a much wider group of researchers and much easier to implement in commercial software if they could be formulated in an alternative way which does not require access to system level matrix manipulations. This is the typical situation for commercial software. To address this limitation, in this paper, an alternative implementation of the arc-length solver proposed by reference [1] is presented and tested by benchmark problems with interface elements. The paper is organized as follows. In the next section, the general structure of arc-length solution framework is concisely presented. In the following section, a physically nonlinear model problem is introduced and the rate of internal energy and dissipation rate are expressed in concrete forms. Through a suitable numerical integration scheme, the framework is expressed in a discrete setting and the overall algorithmic structure is summarized. In Section 4, the alternative implementation is explained and it is discussed how this new implementation can be realized within the commercial software Abaqus. The next section is reserved for illustrative examples before the paper is closed by some conclusive comments and outlook.

2. Tracing an equilibrium path

In this section, for the sake of completeness, the general strategy to trace the equilibrium path for a physically non-linear problem is reiterated briefly.

The equilibrium of internal \underline{f}_{int} and external \underline{f}_{ext} forces in a discrete setting can be conveniently expressed as,

$$\underline{r} = \underline{f}_{int} - \underline{f}_{ext} \tag{1}$$

where \underline{r} is the out of balance residual force column and has to be zero (within a tolerance) for equilibrium. Arc-length solvers typically expresses the external force as the multiplication of fixed normalized external force column $\hat{\underline{f}}$ and the unknown multiplier λ as,

$$\underline{f}_{ext} = \lambda \hat{\underline{f}} \tag{2}$$

In addition to the structural degrees of freedom of the system, the multiplier λ is also an unknown. Therefore an additional equation has to be specified so that a solvable system emerges. This extra equation is in fact known as the path following constraint and tailored for different problems resulting in different arc-length techniques. Therefore the total set of equations that have to be satisfied can be written as,

$$\begin{bmatrix} \underline{f}_{int} - \lambda \hat{\underline{f}} \\ \phi(\underline{u}, \lambda) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3}$$

in which $\phi(\underline{u}, \lambda) = 0$ is the constraint equation. In an incremental-iterative solution framework, the non-linear equations are linearized around a solution state,

$$\begin{bmatrix} \underline{r} \\ \phi \end{bmatrix} \Big|_k + \frac{\partial \underline{r}}{\partial \underline{u}} \Big|_k \delta \underline{u} + \frac{\partial \underline{r}}{\partial \lambda} \Big|_k \delta \lambda = 0 \quad \begin{bmatrix} \underline{K}_{uu} & \underline{K}_{u\lambda} \\ \underline{K}_{\lambda u} & \underline{K}_{\lambda\lambda} \end{bmatrix} \begin{bmatrix} \delta \underline{u} \\ \delta \lambda \end{bmatrix} = \begin{bmatrix} -\underline{r} \\ -\phi \end{bmatrix} \tag{4}$$

and the updated values of the system unknowns are obtained through $\underline{u}_{k+1} = \underline{u}_k + \delta \underline{u}$ and $\lambda_{k+1} = \lambda_k + \delta \lambda$. In the next section a specific form of the constraint equation will be presented using a model problem.

3. Model problem: Cohesive fracture with interface elements

An efficient finite element based technique suitable for fracture problems with known crack path is to use cohesive interface elements along the crack path in the model. Such a model with two layers of different materials is shown in Fig. 1. Total volume is represented by Ω and Γ depicts the interface area where the cohesive fracture takes place. It is assumed that the material behavior in each layer is linear elastic and the behavior of the interface is described by traction-opening relations, see e.g. reference [12]. In what follows, the tractions (normal and tangential components) associated with the interface are represented by the column \underline{T} and the conjugate openings by $\underline{\Delta}$. For the details of their general definitions, the reader may consult reference [12]. The traction-opening relations could be of any type provided that the unloading branch is linear and there is no permanent opening left upon unloading. A very large fraction of existing traction-opening laws fall within this category.

As stated by the second law of thermodynamics, for non-dissipative systems, the dissipated energy has to be non-negative. Typically at early stages of loading history and in case of unloading, until the inelasticity mechanisms are (re-) activated, the dissipation does not increase. Therefore the path following constraint can not be based solely on dissipation. For such intervals of loading history, the constraint equation shall be based on the rate of internal energy. For the model problem in hand, the rate of internal energy can be expressed as,

$$\dot{U} = \int_{\Omega} \dot{\psi}_B dV + \int_{\Gamma} \dot{\psi}_I dV \tag{5}$$

where ψ_B and ψ_I are the internal energy densities associated with the bulk and the interface, respectively. In a spatially discretized setting, \dot{U} can be put in the following form,

$$\begin{aligned} \dot{U} = & \int_{\Omega} \frac{1}{2} \underline{\dot{u}}^T \underline{B}^T \underline{\sigma} dV + \int_{\Omega} \frac{1}{2} \underline{\dot{u}}^T \underline{B}^T \underline{\hat{\sigma}} dV + \int_{\Gamma} \frac{1}{2} \underline{\Delta}^T \underline{N}^T \underline{t} d\Gamma \\ & + \int_{\Gamma} \frac{1}{2} \underline{\Delta}^T \underline{N}^T \underline{\dot{t}} d\Gamma \end{aligned} \tag{6}$$

$$\dot{U} = \frac{1}{2} \underline{\dot{u}}^T \underline{f}_B + \frac{1}{2} \underline{\dot{u}}^T \underline{\dot{f}}_B + \frac{1}{2} \underline{\dot{u}}^T \underline{f}_{cz} + \frac{1}{2} \underline{\dot{u}}^T \underline{\dot{f}}_{cz} \tag{7}$$

where \underline{f}_B and \underline{f}_{cz} represent the nodal internal and cohesive zone forces, respectively. \underline{B} is the strain-displacement operator resulting from the finite element discretization. Introducing the total internal forces as

$\underline{f}_{int} = \underline{f}_B + \underline{f}_{cz}$ (here summation is an assembly in the usual finite element sense) and using the equilibrium condition $\underline{f}_{int} = \underline{f}_{ext}$ which implies $\underline{f}_{int} = \lambda \hat{\underline{f}}$, the following compact form for \dot{U} ,

$$\dot{U} = \frac{1}{2} \left(\lambda \underline{\dot{u}}^T + \dot{\lambda} \underline{u}^T \right) \hat{\underline{f}} \quad (8)$$

is obtained. Expressing the rate in a discrete form through the mid-point rule (see Appendix of [11]), the following constraint function

$$\phi^U = \frac{1}{2} \left(\lambda_{n+1} \underline{u}_{n+1}^T - \lambda_n \underline{u}_n^T \right) \hat{\underline{f}} - \Delta\tau^U = 0 \quad (9)$$

emerges. Here $\Delta\tau^U$ is the parameter controlling the amount of energy to be stored in the whole structure in an increment. Therefore Eq. (9) can be used as the path following constraint conveniently when there is no increase in dissipation.

Upon the onset of inelasticity, the dissipation of the structure starts to increase. In other words, the difference between the exerted power and the rate of stored energy \dot{E} has to be positive and can be written as,

$$D = P - \dot{E} = \lambda \hat{\underline{f}} \dot{\underline{u}} - \dot{E} \quad (10)$$

The stored energy has two contributions, namely from the bulk and the cohesive interface expressed as,

$$E = \int_{\Omega} \frac{1}{2} \underline{\epsilon}^T \underline{\sigma} dV + \int_{\Omega} \frac{1}{2} \underline{\Delta}^T \underline{t} d\Gamma \quad (11)$$

$$E = \frac{1}{2} \underline{u}^T \left(\underline{f}_B + \underline{f}_{cz} \right) = \frac{1}{2} \underline{u}^T \underline{f}_{int} \quad (12)$$

The time rate of E simply reads,

$$\dot{E} = \frac{1}{2} \underline{\dot{u}}^T \underline{f}_{int} + \frac{1}{2} \underline{u}^T \dot{\underline{f}}_{int} \quad (13)$$

and in turn results in,

$$D = \lambda \hat{\underline{f}} \dot{\underline{u}} - \frac{1}{2} \underline{\dot{u}}^T \dot{\underline{f}}_{int} - \frac{1}{2} \underline{u}^T \dot{\underline{f}}_{int} = \frac{1}{2} \left(\lambda \underline{\dot{u}}^T - \dot{\lambda} \underline{u}^T \right) \hat{\underline{f}} \quad (14)$$

through insertion into Eq. (10). Using the mid-point rule, time discretized form of dissipation can be written as,

$$\phi^D = \frac{1}{2} \left(\lambda_n \underline{u}_{n+1}^T - \lambda_{n+1} \underline{u}_n^T \right) \hat{\underline{f}} - \Delta\tau^D = 0 \quad (15)$$

where $\Delta\tau^D$ is the amount of energy to be dissipated within the increment. In fact, Eq. (15) serves as the constraint equation to be satisfied when the increment under consideration is a dissipation controlled one. In the algorithmic setting, to avoid a singular system in the first increment, the following simple constraint function,

$$\phi_1^F = (\lambda - \Delta\tau_1^F) = 0 \quad (16)$$

is used, where $\Delta\tau_1^F$ is to be specified. Since this constraint is equivalent to $\lambda = \Delta\tau_1^F$, it corresponds to a force controlled increment. The algorithmic structure of a typical time step is given in Table 1.

The critical step is the decision on the type of constraint equation to be used in the next increment. This decision is done at the end of each increment and the algorithmic flow is given in Table 2. It has to be noted that these tables are slightly modified versions of the original algorithm given in a single table in reference [1]. Prior to snap-back and within the snap-back phase of the equilibrium path, incremental energy becomes negative. This case has to be monitored as well which is done here by means of the negative energy flag. Obviously, the ratio $a = \frac{\Delta\tau^D}{\Delta\tau^U}$ which facilitates the switch from incremental energy controlled to dissipation controlled algorithm, has to be supplied by the user.

Table 1

Algorithmic flow for a typical increment. ‘←’ means ‘is assigned to’.

```

▷ Beginning of increment (counter n)
  ▶ If this is the first increment (n = 1)
    φ ← φF
  ▶ Else
    ◊ If incremental energy controlled (if energy flag is on; EF = 1) then
      φ ← φU
    ◊ Else
      φ ← φD
    ◊ End
  ▶ End
▷ Solve equation system (4) iteratively
▷ Decide on the arc-length control type for the next increment (See Table 2)
▷ n ← n + 1
▷ Go to the beginning of increment
    
```

Table 2

Decision algorithmic for the arc-length control type.

```

▷ If this is the first increment (n = 1)
  ▶ Calculate Δτ1U
  ▶ ΔτU ← Δτ1U
  ▶ Set energy flag EF = 1 and negative energy flag NEF = 0
▷ Else
  ▶ If energy flag is on (EF = 1) then
    ◊ Calculate ΔτD
    ◊ If ΔτD > a × Δτ1U then
      □ Switch energy flag EF ← 0
      □ ΔτD ← a × Δτ1U
    ◊ End
  ▶ Else
    ◊ Calculate ΔτU
    ◊ If ΔτU < 0 and NEF = 0 then
      □ Switch negative energy flag NEF ← 1
    ◊ Else If ΔτU > 0 and ΔτU < Δτ1U and EF = 0 then
      □ Switch negative energy flag NEF ← 0
    ◊ Else if ΔτU > Δτ1U and EF = 0 then
      □ Switch energy flag EF ← 1
      □ ΔτU ← Δτ1U
      □ If NEF = 1, switch it; NEF ← 0
    ◊ End
  ▶ End
▷ End
    
```

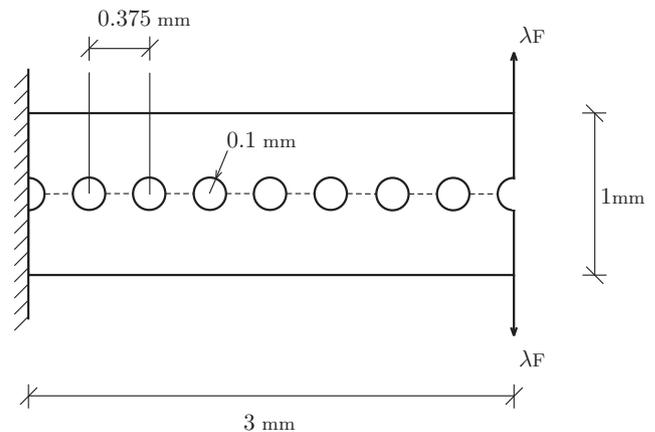


Fig. 2. Geometry and boundary conditions of the perforated beam problem.

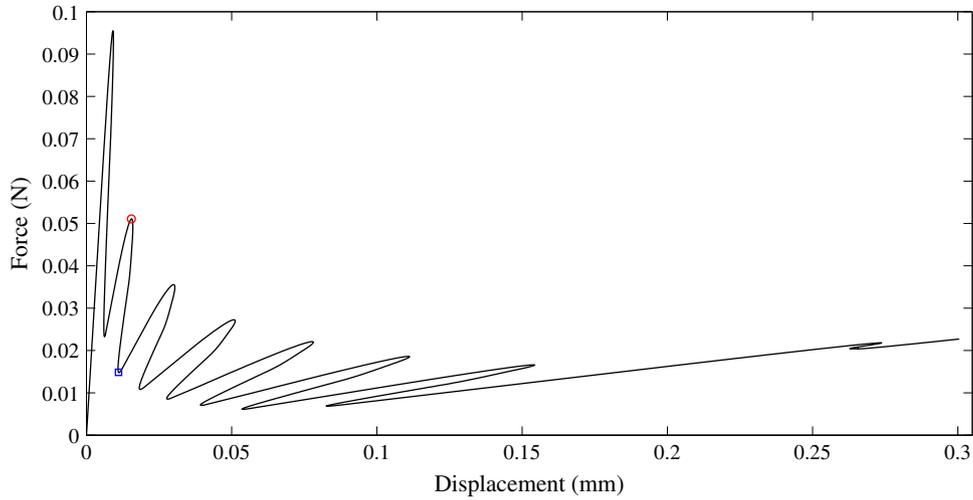


Fig. 3. Force–displacement curve for the perforated beam. The markers correspond to the deformed configurations shown in Fig. 4.

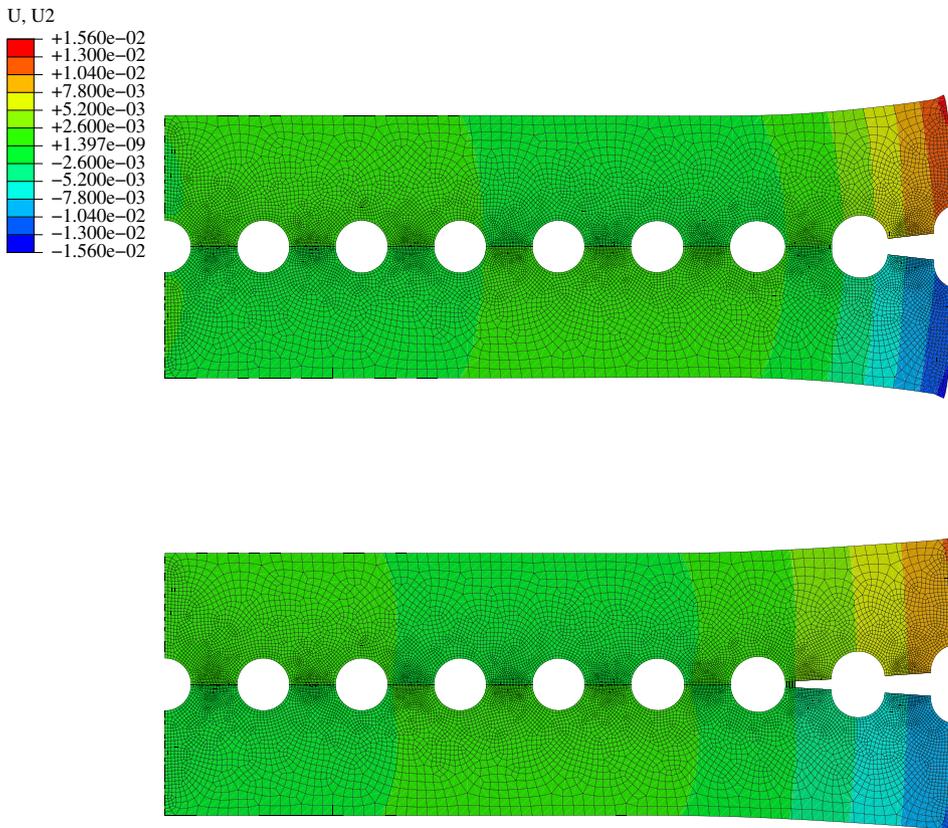


Fig. 4. Perforated beam with interface elements; displacements are magnified by 5. The first configuration corresponds to the first (blue) marker and the second one corresponds to the second (red) marker on force–displacement curve, see Fig. 3. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4. Alternative implementation

Considering a model problem such as given in Fig. 1, the external force column can be expressed as,

$$\underline{f}_{n+1}^{ext} = \mathbf{A} \lambda_{n+1} \begin{bmatrix} \cos \alpha_m \\ \sin \alpha_m \end{bmatrix} = \mathbf{A} \lambda_{n+1} \hat{\underline{f}}_m \quad (17)$$

where \mathbf{A} represents the assembly of nodal forces to the system level external force column in the usual finite element sense and L is the total number of nodes with external loading. It has to be noted that the direction of loading (in other words α_m ; positive in counter-clockwise direction) is fixed throughout the loading history. Within this setting, the explicit forms of ϕ^U and ϕ^D read,

$$\phi^U = \frac{1}{2} \sum_{m=1}^L \left(\underbrace{\lambda_{n+1} \hat{\underline{f}}_m^T \underline{u}_{n+1}^m - \lambda_n \hat{\underline{f}}_m^T \underline{u}_n^m}_{\phi_m^U} - \frac{\Delta \tau^U}{L} \right) = 0 \quad (18)$$

$$\phi^D = \frac{1}{2} \sum_{m=1}^L \left(\underbrace{\lambda_n \hat{\underline{f}}_m^T \underline{u}_{n+1}^m - \lambda_{n+1} \hat{\underline{f}}_m^T \underline{u}_n^m}_{\phi_m^D} - \frac{\Delta \tau^D}{L} \right) = 0 \quad (19)$$

where $\hat{\underline{f}}_m$ is the column of unit external force components $\hat{\underline{f}}_m^T = [\cos \alpha \quad \sin \alpha]$; and \underline{u}_m is the column of displacement components $(\underline{u}^m)^T = [u_m \quad v_m]$ of node m . The last terms of Eqs. (9) and (15) are pulled under the summation operator by re-expressing $\Delta \tau^{U,D}$ as $\frac{\Delta \tau^{U,D}}{L}$.

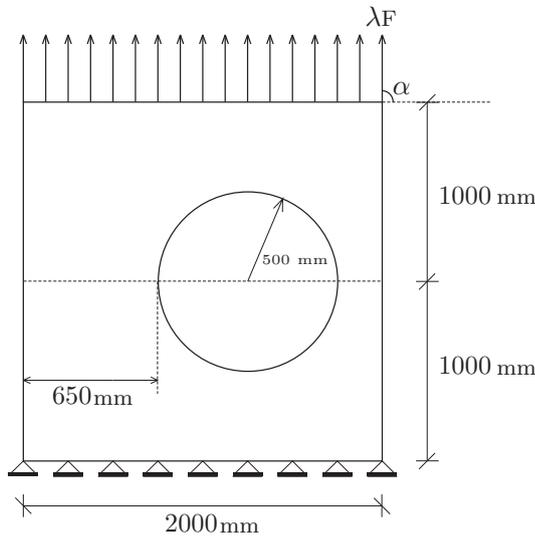


Fig. 5. Geometry and boundary conditions of plate with an eccentric hole.

Since λ is an additional degree of freedom, it is appropriate to introduce an additional node in the model with a single degree of freedom corresponding to λ . In the following parts, this node is going to be labeled as arc-length node. Referring back to the system of Eqs. (3), unknown (due to unknown pre-multiplier λ_{n+1}) external forces and ϕ^U (or ϕ^D) have to be assembled in addition to the internal forces f_{n+1}^{int} which are handled internally by the corresponding element routines in a straight forward manner. To handle assembly of f_{n+1}^{ext} and $\phi_{n+1}^{U,D}$, a new element is introduced which is basically motivated by the structure of Eqs. (17)–(19).

The element is comprised of two nodes; the first node is the arc-length node and the second is node m on which external load acts. Therefore, a separate element has to be added to the model for each node with external loading. The degrees of freedom of this new element type can be written as $\underline{u}^* = [\lambda \quad \underline{u}^m]^T$ where $\underline{u}^m = [u_m \quad v_m]^T$ is the sub-column with horizontal and vertical displacement degrees of freedom of the node m with external node. Since Eqs. (18) and (19) are scalar equations, the summation operator can be interpreted as the assembly operator. Therefore, by constructing the force column of the new element as,

$$\underline{f}^* = \begin{bmatrix} \phi_m^{U,D} - \frac{\Delta t \cdot U,D}{L} \\ -\lambda_{n+1} \cos \alpha_m \\ -\lambda_{n+1} \sin \alpha_m \end{bmatrix} \quad (20)$$

The assembly of external force column (Eq. (17)) and the enforcement of the constraint equation ((18) or (19)) can be embedded into the internal force column assembly procedure. Constructing the corresponding element stiffness matrix through, $K_{ij}^{arc} = \frac{\partial f_i^*}{\partial u_j^*}$ and assembling it to the structural tangent stiffness matrix naturally leads to the system of linearized equations (given by equation system (4)) that have to be solved. In case of energy controlled increment, the element stiffness matrix reads as,

$$\underline{K}^{arc} = \begin{bmatrix} \frac{1}{2} \hat{f}_m^T \underline{u}_{n+1}^m & \frac{1}{2} \lambda_{n+1} \hat{f}_m^T \\ -\cos \alpha & \underline{Q}^T \\ -\sin \alpha & \underline{Q}^T \end{bmatrix} \quad (21)$$

and for the dissipation controlled step, \underline{K}^{arc} takes the following form,

$$\underline{K}^{arc} = \begin{bmatrix} -\frac{1}{2} \hat{f}_m^T \underline{u}_n^m & \frac{1}{2} \lambda_n \hat{f}_m^T \\ -\cos \alpha & \underline{Q}^T \\ -\sin \alpha & \underline{Q}^T \end{bmatrix} \quad (22)$$

where $\underline{Q}^T = [0 \quad 0]$.

Upon completion of the increment, proper arc-length constraint function has to be chosen following the algorithm given in Table 2. A closer look reveals that incremental energy and dissipation need to be calculated. In other words, the values of $\sum_{m=1}^L \phi_m^U$ and $\sum_{m=1}^L \phi_m^D$ have to be calculated. Within the user element subroutine developed, individual element contributions (ϕ_m^U and ϕ_m^D , see Eqs. (18) and (19)) are calculated and updated in each iteration and stored in a user defined common block. Upon convergence, the summation of individual contributions is done at the end of each increment. In case of commercial software Abaqus, this is accomplished by using the uexternaldb.for subroutine with access to the user defined common block, which is called at the end of each increment. Similar user subroutine options exist in other commercial softwares (e.g. MSC Marc-Mentat) therefore the approach presented here is not restricted to a particular software system.

The major drawback of the proposed implementation is the fact that the band-width of the system matrix to be solved is increased. To avoid this, the original implementation suggested the use of Sherman-Morrison formula to ‘invert’ the system matrix and solve for the corrective iterative displacements in an efficient way.

5. Numerical examples

The alternative implementation of the arc-length solver is realized in Abaqus ¹ following the formulation laid out in the previous section. Two benchmark problems are presented for which the bulk behavior is taken as linear elastic. The behavior of the cohesive interface is described by means of traction-opening relations which assume a linear elastic form prior to damage. The uncoupled elastic response of the interface is controlled by the parameters K_{nn} and K_{ss} and can be interpreted as initial stiffness (or penalty parameters) of the traction-opening relations in normal and tangential directions, respectively.

Damage initiation is controlled by a specific criterion which can be expressed in terms of tractions or openings in Abaqus. Here, maximum opening criterion is used such that damage starts to evolve when the normal or tangential opening reaches the user defined critical opening value. The stiffness of the interface is degraded by the pre-multiplier $(1 - D)$ where D is the scalar damage variable varying between 0 and 1. The evolution of D can be described by different functional forms in Abaqus. The evolution of damage under a combination of normal and tangential opening is described by an effective opening $\delta_m = \sqrt{\langle \delta_n \rangle^2 + \delta_s^2}$ where $\langle \rangle$ is the Macaulay bracket. Among various choices, for the following examples, the damage evolution is based on energy dissipated due to interface failure, G^C (fracture energy), and a linear softening law is specified. Therefore Abaqus ensures that the area under the effective opening versus traction curve is equal to the fracture energy, [13]. Furthermore, the effect of mode mixity on fracture energy is taken into account by Benzeggagh-Kenane fracture criterion expressed as,

$$G_n^C + \left(G_s^C - G_n^C \right) \left(\frac{G_s}{G_s + G_n} \right)^\eta = G^C \quad (23)$$

where G_n^C , G_s^C and η are pure mode I fracture energy, mode II fracture energy and mode mixity exponent, respectively. These three quantities are considered to be material properties and specified by the user. G_n and G_s represent the work of normal and tangential tractions over associated work conjugate openings during loading.

5.1. Perforated beam

The perforated beam lay-out shown in Fig. 1 is frequently used to

¹ The implementation is freely available on sourceforge.net. Please use ‘Abaqus arc-length implementation’ as the search key on the website.

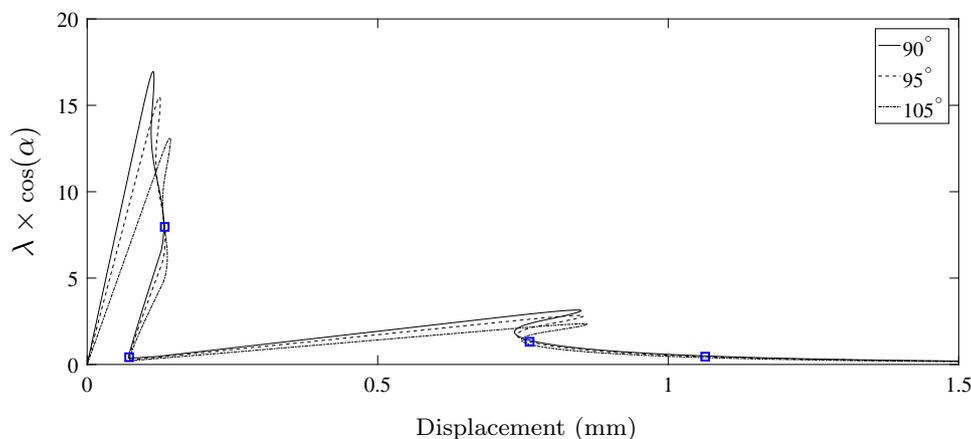


Fig. 6. Force–displacement graphs for different α values; square markers correspond to deformed configurations shown in Fig. 7.

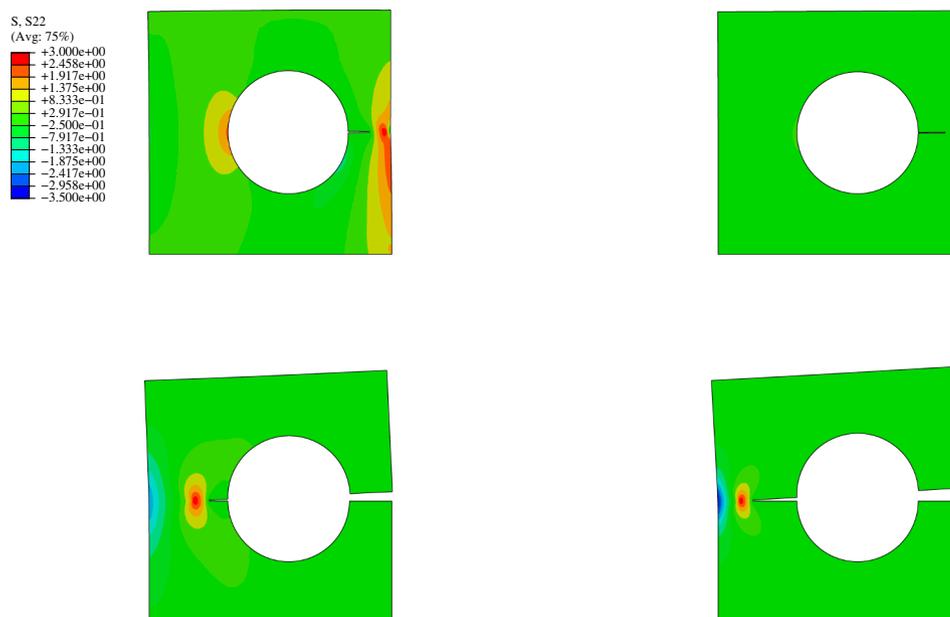


Fig. 7. Crack propagation at four different stages for $\alpha = 95^\circ$ (marked with squares on the force–displacement graph); displacements are magnified by 100.

test the capabilities of arc-length solvers for physically non-linear problems. Different authors adopted different dimensions and the ones presented in Fig. 2 are based on the values given in reference [1].

The bulk is discretized by 4-noded quadrilateral plane strain elements (15104 elements) and 4-noded cohesive interface elements (238 elements) are inserted along the dashed line shown in Fig. 2. The cantilever beam is loaded by two point forces acting on opposite directions as shown in the figure. Modulus of elasticity and Poisson’s ratio for the linear elastic bulk material are $E = 100 \text{ MPa}$ and $\nu = 0.3$, respectively. Fracture energies and the mode-mixity power are taken as $G_n^C = 0.25 \text{ N/mm}$, $G_t^C = 0.25 \text{ N/mm}$ and $\eta = 1$, respectively.

The critical opening values controlling the damage onset are taken to be 0.0001 mm for both normal and tangential openings. In combination with initial stiffnesses of $K_{nn} = 10,000 \text{ N/mm}$ and $K_{tt} = 10,000 \text{ N/mm}$, these critical opening values lead to an ultimate strength of 10 N for mode I opening which matches the value used in reference [1]. Initial incremental energy and the switching parameter are set to $\tau_1^F = 0.05 \text{ N}$; $a = 0.05$.

The resulting force displacement curve is given in Fig. 3 which is almost identical to the one given in [1]. As the crack reaches a hole, a large amount of bulk volume unloads and the amount of incremental released energy from the bulk turns out to be much larger than the

incremental energy dissipated along the failure interface. This results in snap-back type behavior as clearly seen in Fig. 3.

Two representative deformed configurations during loading are shown in Fig. 4. The corresponding points are marked on the force–displacement curve given in Fig. 3. Contour plots reflect the distribution of displacement component in y-direction and the snap-back type behavior can be clearly seen by comparing the two configurations.

5.2. Plate with an eccentric hole

The geometry of the problem is taken from reference [14] and the boundary conditions are shown in Fig. 2. In this example, external load is acting on the nodes along the top edge of the plate as opposed to the previous one in which there was only a pair of loaded nodes.

Furthermore, three different values for the loading angle $\alpha = 90^\circ$, $\alpha = 95^\circ$ and $\alpha = 105^\circ$ are considered to introduce mode-mixity to the problem. 4-noded quadrilateral plane strain elements (9976 elements) are used for the bulk and 4-noded zero thickness interface elements (140 elements) are inserted along the horizontal dashed line shown in Fig. 5. Initial incremental energy and the switching parameter are set to $\tau_1^F = 75 \text{ N}$; $a = 0.2$ for $\alpha = 90^\circ$ and $\tau_1^F = 75 \text{ N}$; $a = 0.1$ for $\alpha = 95^\circ$ and $\alpha = 105^\circ$ cases.

Sticking to the values given in [14], the strength and fracture energy of the interface elements are taken to be 3 MPa and 0.1 N/mm in mode I, 3 MPa and 0.15 N/mm in mode II, respectively. The initial stiffness of the interface elements is set to 10,000 N/mm in all opening modes and damage initiation opening is forced to be 0.0003 mm. Mode-mixity value η is taken to be unity similar to the previous example. Modulus of elasticity and Poisson's ratio of the bulk are set to 20,000 Mpa and 0.2, respectively.

In Fig. 6, the vertical component of external load versus the vertical displacement of upper right corner of the plate is given for three different α values. The markers shown on this graph corresponds to the four different deformed configurations shown in Fig. 7.

For all three α values, initiation and growth of the crack follows similar sequences starting from the left end of the narrower side of the plate and at a certain stage jumping to the free surface on the rightmost and consequently the plate detaches completely on the narrower side, see the first and the second configurations shown in Fig. 7. Afterwards, the structure reloads until the crack propagation starts on the left hand-side of the eccentric hole.

6. Conclusion and outlook

This paper has focused on arc-length techniques for physically non-linear problems and an alternative implementation of a recently proposed arc-length method is elaborated and realized within a commercial finite element software. The effectiveness of the implementation is demonstrated by two cohesive fracture problems.

Since incremental energy/dissipation could be written in terms of global quantities, material point level (integration point) information was not necessary for the treatment of the constraint equation appended to the system of equilibrium equations. Similarly, incremental energy/dissipation for geometrically linear continuum damage mechanics formulations could be cast in the same format which implies that the proposed implementation can be used for both local and non-local damage models as well. However, in case of permanent deformations (e.g. elasto-plasticity) and/or geometrically non-linear problems, element level information is essential. Nevertheless, for a non-negligible portion of localization problems (e.g. failure in quasi-brittle materials) geometrically linear formulations are considered to be satisfactory and the proposed implementation would be instrumental to

trace the complete equilibrium path for such problems.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at <https://doi.org/10.1016/j.tafmec.2019.01.007>.

References

- [1] S. May, J. Vignollet, R. de Borst, A new arc-length control method based on the rates of the incremental and the dissipated energy, *Eng. Comput.* 33 (1) (2016) 100–115, [https://doi.org/10.1016/0003-4916\(63\)90068-X](https://doi.org/10.1016/0003-4916(63)90068-X).
- [2] E. Riks, An incremental approach to the solution of snapping and buckling problems, *Int. J. Solids Struct.* 15 (1979) 529–551, [https://doi.org/10.1016/0003-4916\(63\)90068-X](https://doi.org/10.1016/0003-4916(63)90068-X).
- [3] G.A. Wempner, Discrete approximations related to nonlinear theories of solids and structures, *Int. J. Solids Struct.* 7 (1971) 1581–1599, [https://doi.org/10.1016/0003-4916\(63\)90068-X](https://doi.org/10.1016/0003-4916(63)90068-X).
- [4] M.A. Crisfield, A fast incremental/iterative solution procedure that handles 'snap-through', *Comput. Struct.* 13 (1981) 55–62, [https://doi.org/10.1016/0003-4916\(63\)90068-X](https://doi.org/10.1016/0003-4916(63)90068-X).
- [5] E. Ramm, Strategies for tracing the nonlinear response near limit points, in: W. Wunderlich, E. Stein, K.J. Bathe (Eds.), *Nonlinear Finite Elements in Structural Mechanics*, 1981, pp. 63–89, [https://doi.org/10.1016/0003-4916\(63\)90068-X](https://doi.org/10.1016/0003-4916(63)90068-X).
- [6] M. Geers, Enhanced solution control for physically and geometrically non-linear problems. Part I - the subplane control approach, *Int. J. Numer. Meth. Eng.* 46 (1999) 177–204, [https://doi.org/10.1016/0003-4916\(63\)90068-X](https://doi.org/10.1016/0003-4916(63)90068-X).
- [7] R. de Borst, Computation of post-bifurcation and post-failure behavior of strain-softening solids, *Comput. Struct.* 25 (2) (1987) 211–224, [https://doi.org/10.1016/S0031-8914\(53\)80099-6](https://doi.org/10.1016/S0031-8914(53)80099-6).
- [8] M.A. Guterrez, Energy release control for numerical simulations of failure in quasi-brittle solids, *Commun. Numer. Methods Eng.* 20 (2004) 19–29, [https://doi.org/10.1016/0003-4916\(63\)90068-X](https://doi.org/10.1016/0003-4916(63)90068-X).
- [9] C. Verhoosel, J. Remmers, M.A. Guterrez, A dissipation-based arc-length method for robust simulation of brittle & ductile failure, *Int. J. Numer. Meth. Eng.* 77 (2009) 1290–1321, [https://doi.org/10.1016/0003-4916\(63\)90068-X](https://doi.org/10.1016/0003-4916(63)90068-X).
- [10] V.P. Nguyen, H. Nguyen-Xuan, High order bsplines based finite elements for the delamination analysis of laminated composites, *Compos. Struct.* 102 (2013) 261–275, [https://doi.org/10.1016/0003-4916\(63\)90068-X](https://doi.org/10.1016/0003-4916(63)90068-X).
- [11] F.P. van der Meer, L.J. Sluys, S.R. Hallett, M.R. Wisnom, Computational modeling of complex failure mechanisms in laminates, *J. Compos. Mater.* 46 (5) (2012) 603–623, [https://doi.org/10.1016/0003-4916\(63\)90068-X](https://doi.org/10.1016/0003-4916(63)90068-X).
- [12] R. de Borst, Some recent issues in computational failure mechanics, *Int. J. Numer. Meth. Eng.* 52 (2001) 63–95, [https://doi.org/10.1016/S0031-8914\(53\)80099-6](https://doi.org/10.1016/S0031-8914(53)80099-6).
- [13] Abaqus 6.14 theory guide, Dassault Systemes.
- [14] E. Lorentz, P. Badel, A new path-following constraint for strain softening finite element simulations, *Int. J. Numer. Meth. Eng.* 60 (2004) 499–526, <https://doi.org/10.1002/nme.971>.