Pontifícia Universidade Católica
do Rio de Janeiro

Rafael Otávio Alves Abreu

Advances in implicit integration algorithms for multisurface plasticity

Tese de Doutorado

Thesis presented to the Programa de Pós-graduação em Engenharia Civil of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Engenharia Civil.

Advisor: Profa. Deane de Mesquita Roehl

Co-advisor: Dr. Eleazar Cristian Mejía Sanchez

Rio de Janeiro

September 2023

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO

Rafael Otávio Alves Abreu

Advances in implicit integration algorithms for multisurface plasticity

Thesis presented to the Programa de Pós-graduação em Engenharia Civil of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Engenharia Civil. Approved by the Examination Committee.

Profa. Deane de Mesquita Roehl
Advisor
Departamento de Engenharia Civil e Ambiental – PUC-Rio

Dr. Eleazar Cristian Mejia Sanchez
Co-advisor
Instituto Tecgraf/PUC-Rio

Prof. Alessandro Cirone
Departamento de Engenharia Civil e Ambiental – PUC-Rio

Prof. Ivan Fabio Mota de Menezes
Departamento de Engenharia Mecânica – PUC-Rio

Prof. Eduardo Alberto de Souza Neto
Swansea University

Prof. Roque Luiz da Silva Pitangueira
UFMG

Rio de Janeiro, September 22nd, 2023

Rafael Otávio Alves Abreu

The author holds a Bachelor degree in civil engineering from Universidade Estadual do Maranhão (2016) and a Master degree in civil engineering with emphasis on structures from Pontifical Catholic University of Rio de Janeiro (2019). In 2019, he enrolled in the doctoral program at PUC-Rio with an emphasis on structures. Currently, he is a researcher at the Tecgraf Institute/PUC-Rio in the group of Multiphysics Modeling and Simulation, developing research in geomechanics. He has experience on constitutive models, artificial intelligence, and concrete structures, working in the following subjects: elastoplasticity, continuum damage mechanics, parameter identification, machine learning, optimization algorithms, and structural concrete.

Bibliographic data

Abreu, Rafael Otávio Alves

Advances in implicit integration algorithms for multisurface plasticity / Rafael Otávio Alves Abreu ; advisor: Deane de Mesquita Roehl ; co-advisor: Eleazar Cristian Mejía Sanchez. – 2023.
212 f. : il. color. ; 29.7 cm

Tese (doutorado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Civil e Ambiental, 2023.
Inclui bibliografia

1. Engenharia Civil e Ambiental - Teses. 2. Modelos com múltiplas superfícies de plastificação. 3. Integração implícita. 4. Método de Newton-Raphson. 5. Método de Newton-Krylov. 6. Modelo elastoplástico com dano acoplado. I. Roehl, Deane de Mesquita. II. Mejia Sanchez, Eleazar Cristian. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Civil e Ambiental. IV. Título.

CDD: 624

Acknowledgements

I sincerely thank all the people and institutions who played a vital role in this journey of research and development.

To my parents, Maria Rita and Julio Cesar, and my sister, Paula Renata, who encouraged during challenging times and provided me with the necessary attention and support to achieve my goals.

To my advisor, Deane Roehl, for all support, trust, encouragement, and the priceless knowledge shared throughout the development of this research.

To my co-advisor, Cristian Mejia, for all the guidance, suggestions, and support fundamental to the development of this work.

To Gilcyvania Castro, who won my heart and accompanied me during all this journey, even far from home. We supported each other along this time, mainly in difficult moments. We overcame these challenges together, and we will continue advancing together to reach our objectives.

To all my family and friends who, in various ways, contributed to my success in this academic journey.

To Karla Gonçalves for all the care and invaluable assistance in refining the text of my thesis, enhancing the quality and clarity of my texts throughout this journey.

To my colleagues and former colleagues at Tecgraf Institute/PUC-Rio, who offered their full support and friendship. Special thanks to Francisco Dias, Francisco de Assis, Juliema Fronczak, Juan Carlos, Julio Rueda, Kaio César, Karin Kauss, Karoline Oliveira, Karl Martins, Lourdes de Souza, Luis Felipe, Luis Fernando,

Marcello Congro, Marcelo Texeira, Nilthson Noreña, Paul Ortega, Pedro Firme, Renan Costa, Roberto Quevedo, and Ronald Mejia.

To my colleagues at PUC-Rio, whose company and friendship made this journey more enriching.

To the professors at Universidade Estadual do Maranhão, who played a fundamental role in my academic development. Special thanks to Professors Eduardo Aguiar, João Celso, and Carlos César.

To the professors and staff at PUC-Rio, for their dedication and ongoing support.

To CNPq and PUC-Rio for all the grants provided, which enabled the development of this work. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

To Tecgraf Institute/PUC-Rio, for the opportunity to conduct challenging and highly relevant research.

Thank you all for being part of this journey and for contributing to my academic and personal growth.

Abstract

Abreu, Rafael Otávio Alves; Roehl, Deane de Mesquita (Advisor); Mejia Sanchez, Eleazar Cristian (Co-advisor). **Advances in implicit integration algorithms for multisurface plasticity.** Rio de Janeiro, 2023. 212p. Tese de Doutorado – Departamento de Engenharia Civil e Ambiental, Pontifícia Universidade Católica do Rio de Janeiro.

The mathematical representation of complex material behavior requires a sophisticated constitutive formulation, as it is the case of multisurface plasticity. Hence, a complex elastoplastic model demands a robust integration procedure for the plastic evolution equations. Developing integration schemes for plasticity models is an important research topic because these schemes are directly related to the accuracy and efficiency of numerical simulations for materials such as metals, concrete, soils and rocks. The performance of the finite element solution is directly influenced by the convergence characteristics of the state-update procedure. Therefore, this work explores the implementation of complex constitutive models, focusing on generic multisurface plasticity models. This study formulates and evaluates state-update algorithms that form a robust framework for simulating materials governed by multisurface plasticity. Implicit integration algorithms are developed with an emphasis on achieving robustness, comprehensiveness and flexibility to handle cumbersome plasticity applications effectively. The state-update algorithms, based on the backward Euler method and the Newton-Raphson and Newton-Krylov methods, are formulated using line search strategies to improve their convergence characteristics. Additionally, a substepping scheme is implemented to provide further robustness to the state-update procedure. The flexibility of the algorithms is explored, considering various stress conditions such as plane stress and plane strain states, within a single, versatile integration scheme. In this scenario, the robustness and performance of the algorithms are assessed through classical finite element applications. Furthermore, the developed multisurface plasticity background is applied to formulate a coupled elastoplastic-damage model, which is evaluated using experimental tests in concrete structures. The achieved results highlight the effectiveness of the proposed state-update algorithms in integrating multisurface plasticity equations and their ability to handle challenging finite element problems.

Keywords

Multisurface plasticity; implicit integration; Newton-Raphson method; Newton-Krylov method; coupled elastoplastic-damage model

Resumo

Abreu, Rafael Otávio Alves; Roehl, Deane de Mesquita (Orientadora); Mejia Sanchez, Eleazar Cristian (Coorientador). **Avanços em algoritmos de integração implícita para plasticidade com múltiplas superfícies.** Rio de Janeiro, 2023. 212p. Tese de Doutorado – Departamento de Engenharia Civil e Ambiental, Pontifícia Universidade Católica do Rio de Janeiro.

A representação matemática de comportamentos complexos em materiais exige formulações constitutivas sofisticada, como é o caso de modelos com múltiplas superfícies de plastificação. Assim, um modelo elastoplástico complexo demanda um procedimento robusto de integração das equações de evolução plástica. O desenvolvimento de esquemas de integração para modelos de plasticidade é um tópico de pesquisa importante, já que estes estão diretamente ligados à acurácia e eficiência de simulações numéricas de materiais como metais, concretos, solos e rochas. O desempenho da solução de elementos finitos é diretamente afetado pelas características de convergência do procedimento de atualização de estados. Dessa forma, este trabalho explora a implementação de modelos constitutivos complexos, focando em modelos genéricos com múltiplas superfícies de plastificação. Este estudo formula e avalia algoritmos de atualização de estado que formam uma estrutura robusta para a simulação de materiais regidos por múltiplas superfícies de plastificação. Algoritmos de integração implícita são desenvolvidos com ênfase na obtenção de robustez, abrangência e flexibilidade para lidar eficazmente com aplicações complexas de plasticidade. Os algoritmos de atualização de estado, baseados no método de Euler implícito e nos métodos de Newton-Raphson e Newton-Krylov, são formulados utilizando estratégias de busca unidimensional para melhorar suas características de convergência. Além disso, é implementado um esquema de subincrementação para proporcionar mais robustez ao procedimento de atualização de estado. A flexibilidade dos algoritmos é explorada, considerando várias condições de tensão, como os estados plano de tensões e plano de deformações, num esquema de integração único e versátil. Neste cenário, a robustez e o desempenho dos algoritmos são avaliados através de aplicações clássicas de elementos finitos. Além disso, o cenário desenvolvido no contexto de modelos com múltiplas superfícies de plastificação é aplicado para formular um modelo elastoplástico com dano acoplado, que é avaliado através de ensaios experimentais em estruturas de concreto. Os resultados obtidos evidenciam

a eficácia dos algoritmos de atualização de estado propostos na integração de equações de modelos com múltiplas superfícies de plastificação e a sua capacidade para lidar com problemas desafiadores de elementos finitos.

Palavras-chave

Modelos com múltiplas superfícies de plastificação; integração implícita; método de Newton-Raphson; método de Newton-Krylov; modelo elastoplástico com dano acoplado

Contents

List of figures

List of tables

List of algorithms

“All models are wrong, but some are useful”

George Box

1
Introduction

1.1.
Context and motivation

The mechanical behavior of engineering materials, such as concrete, soils and rocks, is defined by several multiscale phenomena. In the macromechanical level, material heterogeneities are neglected in the development of phenomenological constitutive models for engineering applications. The plasticity theory offers mathematical support to describe the phenomenological behavior of several materials. This theory describes solids under permanent strains during cyclic loading. In the plasticity theory context, these strains are independent of the loading rate.

Yield surfaces are essential components of elastoplastic models since they define the boundary of the elastic domain. Often, those models are defined by a unique yield surface (Foster et al., 2005; Seifert and Schmidt, 2008). On the other hand, multiple yield surfaces can also be adopted to define the elastoplastic model, primarily for complex constitutive behavior. This type of plasticity model is frequent in various engineering applications, as presented by Lourenço and Rots (1997), Mackenzie-Helnwein, Eberhardsteiner and Mang (2003), Lukacevic, Lederer and Füssl (2017), and Xie et al. (2021).

Implementing an elastoplastic constitutive model in a finite element simulator requires the integration of the constitutive evolution equations. Although analytical and semi-analytical solutions for that integration process exist, they are limited to certain conditions (Szabó, 2009; Becker, 2011). Numerical methods are therefore generally used to integrate these equations. While explicit integration algorithms based on the forward Euler method are straightforward to implement, they may not offer the required accuracy for large strain increments (Sloan, 1987; Abbo and Sloan, 1996; Jakobsen and Lade, 2002). Thus, implicit integration schemes based on the backward Euler method are vastly considered (Simo and Taylor, 1986;

Fuschi et al., 1994; Lourenço, de Borst and Rots, 1997; Huang and Griffiths, 2009; Geng et al., 2021; Nie et al., 2022; Santos and de Souza Neto, 2022) due to their excellent accuracy and stability characteristics (Ortiz and Popov, 1985). Importantly, implicit schemes for multisurface plasticity involve a sophisticated integration algorithm capable of identifying which surface contains the updated state.

The pioneering work of Simo, Kennedy and Govindjee (1988) introduced an implicit integration algorithm for generic multisurface plasticity models, including an extension for viscoplasticity. Since then, numerous other authors dedicated effort to enhance the algorithms to deal with multisurface plasticity, even in recent years (Zheng, Zhang and Wang, 2020; Li, Li and Zheng, 2021; Pech, Lukacevic and Füssl, 2021; Zhao et al., 2022), as complex models can lead to severe convergence problems (Adhikary et al., 2017). A robust algorithm for multisurface plasticity effectively handles complex stress trajectories and large strain increments. Overcoming these issues requires specific investigations into appropriate numerical strategies.

In the numerical context, an implicit integration scheme is generally associated with solving a system of nonlinear equations. The classical Newton-Raphson method is a common choice due to its convergence characteristics. However, this method does not guarantee convergence, mainly in complex constitutive models, as reported by Armero and Pérez-Foguet (2002). They emphasized that yield surfaces with high curvature may show convergence difficulties. Similar problems are reported by Bićanić and Pearce (1996) and Lester and Scherzinger (2017). As verified by Scherzinger (2017), the implicit algorithm known as closest point projection may start an infinite iteration loop, which reduces the convergence region of the method.

Besides robustness, efficiency and accuracy of integration schemes for elastoplastic models, versatility is an important characteristic of these algorithms. However, many existing algorithms are designed for specific constitutive models (Aravas, 1987; Hofstetter, Simo and Taylor, 1993; Feenstra and De Borst, 1996), which limit the applicability of those algorithms. Modeling single and multisurface plasticity requires different schemes, since the latter uses a specific process to identify the active surfaces of the problem (Simo, Kennedy and Govindjee, 1988;

Pech, Lukacevic and Füssl, 2021). Similarly, the distinctive nature of plane stress plasticity typically requires an appropriate algorithm (Simo and Taylor, 1986; Lee and Fenves, 2001; Chatziioannou, Karamanos and Huang, 2021). To address these challenges, proposing an implicit integration scheme that can handle single and multisurface plasticity and accommodate plane stress and three-dimensional stress conditions is an appealing task.

It is worth highlighting that elastoplastic models can effectively capture complex material nonlinearity, such as hardening and softening behavior. Nonetheless, these models may not respond satisfactory when the material experiences stiffness degradation, particularly observed during the material unloading step. In these scenarios, coupling elastoplasticity and damage models becomes valuable for accurately modeling the evolution of unloading stiffness. Simo and Ju (1987a, 1987b) and Ju (1989b) coupled multisurface plasticity with a damage model. Nonetheless, most models available in the literature work with single surface plasticity.

In this context, this work proposes the development of novel algorithms to address the challenges posed by complex plasticity models, with a particular focus on multisurface plasticity. The study explores and evaluates state-update algorithms that provide a robust framework to simulate the behavior of complex engineering materials.

1.2. Objectives

This work aims to contribute to the research field of computational plasticity by studying the implementation of complex constitutive models, focusing on multisurface plasticity. Single-surface models are also assessed since they are special cases of multisurface ones. In this scenario, implicit integration schemes are developed, with particular emphasis on robustness, comprehensiveness, and flexibility of the algorithms. Additionally, this research explores coupled elastoplastic-damage modeling based on the developed multisurface plasticity background and the continuum damage mechanics. A constitutive model is then formulated to accurately represent the multiaxial behavior of conventional concrete.

The specific objectives of this study are as follows:

- Develop strategies to improve robustness and comprehensiveness of implicit integration scheme for generic multisurface plasticity models;
- Develop a Newton-Raphson-based strategy to integrate generic multisurface plasticity, specifically addressing the plane stress condition;
- Investigate a non-conventional algorithm to solve systems of nonlinear equations in the context of integrating generic multisurface plasticity for different stress conditions;
- Propose a coupled elastoplastic-damage model to effectively handle the multiaxial behavior of conventional concrete.

1.3. Structure of the thesis

This work comprises 11 chapters that discuss novel developments in multisurface plasticity by formulating algorithms and constitutive models. This introductory chapter serves as Chapter 1 of the work.

Chapter 2 describes the fundamentals of plasticity theory, presenting the plastic evolution equations in the context of multisurface plasticity. These constitutive equations are essential to the definition of the proposed state-update algorithms.

Chapter 3 presents a state-update algorithm for multisurface plasticity in which optimization methods are employed: Newton-Rapson method and line search strategy. To improve the robustness of the proposed algorithm, a substepping scheme is adopted. Additionally, a tangent modulus consistent with multisurface plasticity and the substepping scheme is formulated.

Chapter 4 evaluates the implicit substepping scheme, conducting analyses at the local (Gauss point) and global levels (solving finite element problems). A multisurface plasticity model is here formulated and used to assess the robustness of the proposed integration scheme.

Chapter 5 formulates a novel Newton-Raphson-based implicit integration algorithm capable of handling plane stress, plane strain, axisymmetric, and three-

dimensional stress conditions. This algorithm applies complementary functions to properly consider multisurface plasticity.

Chapter 6 formulates a novel Newton-Krylov-based implicit integration algorithm that eliminates the need for some derivative computations related to the constitutive equations. Moreover, a finite difference-based scheme to compute the consistent tangent modulus is presented.

Chapter 7 evaluates the proposed Newton-Krylov-based scheme by studying finite element problems that exhibit both hardening and softening behavior. The efficiency of the proposed schemes is assessed in terms of computational time.

Chapter 8 proposes a novel coupled elastoplastic-damage model to simulate the multiaxial behavior of conventional concrete. A regularization technique based on the crack band approach is adopted to mitigate mesh dependency. Furthermore, it is presented an appropriate state-update algorithm, along with specific computational aspects to ensure accuracy. The applicability of the proposed coupled elastoplastic-damage model in simulating concrete structures is then assessed. Experimental tests available in the literature are studied to evaluate the accuracy of the model. The parameter identification procedure proposed during the development of this thesis is applied to compute unknown material parameters.

Chapter 9 concludes this thesis by summarizing the main achievements obtained during the research. Additionally, five appendices are included. Appendix A presents the equations related to the stress invariants and their derivatives. Appendix B discusses the constitutive equations required to implement the multisurface plasticity model formulated in this work. Appendix C presents the multisurface plasticity formulation of the Cap Model. Appendix D presents relevant equations related to the implementation and formulation of the proposed damage model. Finally, Appendix E presents a novel parameter identification procedure that utilizes artificial intelligence techniques to obtain unknown material parameters of the proposed coupled elastoplastic-damage model. Important topics related to machine learning and optimization methods are discussed.

2
Fundamentals of plasticity theory

The classical small strain plasticity theory is considered in this work, following the formalism presented by Simo and Hughes (1998) and de Souza Neto, Perić and Owen (2008). Thus, an additive decomposition of the total strain tensor as the sum of the elastic strain tensor $\boldsymbol{\varepsilon}^e$ and the plastic strain tensor $\boldsymbol{\varepsilon}^p$ is assumed:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^p \tag{2.1}$$

For the linear elastic material, the stress-strain relationship derives from the quadratic stored energy density function. Thus, the stress state for the linear elastic law is

$$\boldsymbol{\sigma} = \boldsymbol{D} : \boldsymbol{\varepsilon}^e \tag{2.2}$$

where $\boldsymbol{\sigma}$ is a stress tensor and $\boldsymbol{D}$ is the elastic constitutive modulus. Additionally, the inverse relation is defined as

$$\boldsymbol{\varepsilon}^e = \boldsymbol{C} : \boldsymbol{\sigma} \tag{2.3}$$

It is essential to establish the elastic domain wherein the total strain increment is equal to the elastic strain increment. The hypersurface that defines the boundary of the elastic domain is named yield surface. In general, an elastoplastic constitutive model may be formulated by multiple yield functions $f_i(\boldsymbol{\sigma}, \boldsymbol{q}) = 0$, where $i \in \{1,2,\dots,N\}$ is the index of each yield surface and $\boldsymbol{q}$ is an internal variable tensor. Therefore, the set of states $(\boldsymbol{\sigma}, \boldsymbol{q})$ that defines the elastic domain of an elastoplastic problem is given by

$$\mathbb{E} = \left\{ (\boldsymbol{\sigma}, \boldsymbol{q}) \;\middle|\; f_i(\boldsymbol{\sigma}, \boldsymbol{q}) < 0,\ \forall i \in \{1,2,\dots,N\} \right\} \tag{2.4}$$

Mathematically, the yield envelope is defined by

$$\mathbb{Y} = \left\{ (\boldsymbol{\sigma}, \boldsymbol{q}) \;\middle|\; f_i(\boldsymbol{\sigma}, \boldsymbol{q}) = 0,\ \exists i \in \{1,2,\dots,N\} \right\} \tag{2.5}$$

The set of states $(\boldsymbol{\sigma}, \boldsymbol{q})$ that describes the plastically admissible domain is given by

$$\overline{\mathbb{E}} = \mathbb{E} \cup \mathbb{Y} \tag{2.6}$$

Figure 1 illustrates a schematic representation of the elastic domain for a multisurface plasticity model. Notice the presence of singular points on the boundary $\mathbb{Y}$, also called corners. Nonsmooth plasticity models are typical in several engineering applications, as presented by Lourenço and Rots (1997) and Huang and Griffiths (2008).

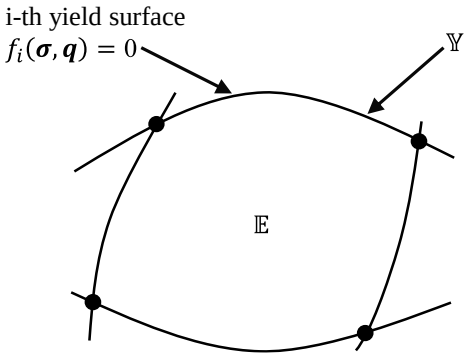

Figure 1 Schematic representation of the elastic domain for a multisurface plasticity model

A plastic model requires the formalization of evolution equations regarding the plastic variables. These definitions are indispensable to characterize the irreversibility of plastic flow. A multisurface plasticity problem requires an adequate characterization of these equations. In this work, Koiter's rule (Koiter, 1953) is considered. Thus, these differential equations are the so-called flow rule and hardening/softening law, respectively

$$\dot{\boldsymbol{\varepsilon}}^p = \sum_{i=1}^{N} \gamma_i \boldsymbol{n}_i(\boldsymbol{\sigma}, \boldsymbol{q}) \tag{2.7}$$

$$\dot{\boldsymbol{q}} = \sum_{i=1}^{N} \gamma_i \boldsymbol{h}_i(\boldsymbol{\sigma}, \boldsymbol{q}) \tag{2.8}$$

Here γ_i are the nonnegative parameters named plastic multipliers, $\boldsymbol{n}_i(\boldsymbol{\sigma}, \boldsymbol{q})$ represent the plastic flow directions, and $\boldsymbol{h}_i(\boldsymbol{\sigma}, \boldsymbol{q})$ characterize the hardening/softening behavior of the material. These predefined functions $\boldsymbol{n}_i$ and $\boldsymbol{h}_i$ depend on material behavior. Equations (2.7) and (2.8) must agree with

complementary conditions to establish the set of plastically admissible states, which are defined by

$$f_i(\boldsymbol{\sigma}, \boldsymbol{q}) \leq 0, \quad \gamma_i \geq 0, \quad \gamma_i f_i(\boldsymbol{\sigma}, \boldsymbol{q}) = 0 \tag{2.9}$$

This set of equations is also called Karush-Kuhn-Tucker conditions. Moreover, the consistency condition must be satisfied:

$$\gamma_i \dot{f}_i(\boldsymbol{\sigma}, \boldsymbol{q}) = 0 \tag{2.10}$$

The plastic multipliers presented in (2.7) and (2.8) can be calculated using the chain rule in Equation (2.10). Therefore, the plastic multipliers γ_i take the form

$$\gamma_i = \sum_{j=1}^{N} d^{ij} \langle \partial_{\boldsymbol{\sigma}} f_j : \boldsymbol{D} : \dot{\boldsymbol{\varepsilon}} \rangle \tag{2.11}$$

in which $\langle \cdot \rangle$ denotes the Macaulay brackets and d^{ij} takes the following form:

$$d^{ij} = \left(\partial_{\boldsymbol{\sigma}} f_i : \boldsymbol{D} : \boldsymbol{n}_j - \partial_{\boldsymbol{q}} f_i . \boldsymbol{h}_j \right)^{-1} \tag{2.12}$$

To define the elastoplastic tangent modulus, it is fundamental to understand the concept of active surface. From an optimization point of view, Equations (2.9) and (2.10) determine active constraints. Thus, the set $\mathbb{J}$ describes the active yield surfaces of the processes.

$$\mathbb{J} = \left\{ i \in \{1,2, \dots, N\} \mid f_i(\boldsymbol{\sigma}, \boldsymbol{q}) = 0 \text{ and } \dot{f}_i(\boldsymbol{\sigma}, \boldsymbol{q}) = 0 \right\} \tag{2.13}$$

The elastoplastic tangent modulus $\boldsymbol{D}^{ep}$ relates the stress rate $\dot{\boldsymbol{\sigma}}$ and the total strain rate $\dot{\boldsymbol{\varepsilon}}$ according to

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{D}^{ep} : \dot{\boldsymbol{\varepsilon}} \tag{2.14}$$

Considering the flow rule (2.7), Equation (2.14) becomes

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{D} : (\dot{\boldsymbol{\varepsilon}} - \dot{\boldsymbol{\varepsilon}}^p) = \boldsymbol{D} : \left(\dot{\boldsymbol{\varepsilon}} - \sum_{i=1}^{N} \gamma_i \boldsymbol{n}_i \right) \tag{2.15}$$

Finally, substituting Equation (2.11) in (2.15), the elastoplastic tangent modulus is

$$\boldsymbol{D}^{ep} = \begin{cases} \boldsymbol{D}, \text{ if } \mathbb{J} = \emptyset \\ \boldsymbol{D} - \sum_{i \in \mathbb{J}} \sum_{j \in \mathbb{J}} d^{ij} (\boldsymbol{D} : \boldsymbol{n}_i) \otimes \left(\boldsymbol{D} : \partial_{\boldsymbol{\sigma}} f_j \right), \text{ if } \mathbb{J} \neq \emptyset \end{cases} \tag{2.16}$$

Note that $\boldsymbol{D}^{ep}$ is symmetric if $\boldsymbol{n}_i = \partial_{\boldsymbol{\sigma}} f_i$. This particular case is called associated flow rule.

3
Implicit substepping integration algorithm

An integration scheme of the elastoplastic evolution equations must be defined to implement a plastic constitutive model in a numerical simulator. Ortiz and Popov (1985) introduced two numerical algorithms in the elastoplasticity context: the generalized trapezoidal and midpoint integrations. Forward and backward Euler schemes (also known as explicit and implicit Euler methods, respectively) are particular cases of these methods. They also studied the numerical stability of the integration algorithms for elastoplastic constitutive equations. In summary, Ortiz and Popov (1985) concluded that the fully implicit integration algorithm is unconditionally stable, mainly for large strain increments. These observations were also made by Gratacos, Montmitonnet and Chenot (1992) and Chaboche and Cailletaud (1996). Therefore, due to their stability and accuracy, implicit integration algorithms are widely adopted by multisurface plasticity applications.

In the past years, several researchers have proposed integration algorithms for multisurface plasticity considering specific constitutive models. Sandler and Rubin (1979), Simo et al. (1988) and Hofstetter, Simo and Taylor (1993) presented state-update algorithms for a well-known three-surface plastic model, Cap Model. Lourenço and Rots (1997) evaluated an interface elastoplastic constitutive model to analyze masonry structures. They also adopted the Cap Model and applied a two-surface plasticity algorithm to treat the nonsmooth transitions between the surfaces. Lourenço, de Borst and Rots (1997) developed an algorithm for plane stress condition to model orthotropic material behavior. They adopted a two-surface plastic model considering a Rankine-type criterion for tension and a Hill-type criterion for compression.

Regarding nonsmooth multisurface plasticity, Simo, Kennedy and Govindjee (1988) presented a one-step return mapping algorithm considering a generic elastoplastic constitutive model. Mackenzie-Helnwein, Eberhardsteiner and Mang

(2003) used this sort of model in a clear wood structural application. Clausen, Damkilde and Andersen (2006, 2007) devoted some attention to studying plasticity problems with multiple yield planes and presented an efficient algorithm to treat classic models such as Tresca and Mohr-Coulomb. Godio et al. (2016) investigated a multisurface numerical integration procedure in a Cosserat finite element formulation. Zheng, Zhang and Wang (2020) solved the constitutive integration of multisurface plasticity equations based on the theory of finite-dimensional variational inequalities. The authors declared that their procedure was globally convergent. Further relevant contributions to the improvement of implicit integration schemes for generic multisurface plasticity models are presented by Karaoulanis (2013), Adhikary et al. (2017) and Pech, Lukacevic and Füssl (2021).

As the classical Newton-Raphson method is a typical choice for solving systems of nonlinear equations, implicit integration algorithms usually undergo convergence problems related to this method. The line search strategy is an efficient approach to improve the convergence properties of the Newton-Raphson method. Several researchers, such as Dutko, Perić and Owen (1993), Pérez-Foguet and Armero (2002), Seifert and Schmidt (2008), and Li and Crouch (2010) adopted this approach in their state-update procedures for single-surface elastoplastic models. Scherzinger (2017) reported that the line search algorithm was able to expand the convergence region of the Newton-Raphson method. Adhikary et al. (2017) also adopted a line search strategy to improve their multisurface return mapping algorithm. All these cited works applied inexact line search methods in their implicit algorithms: quadratic and cubic interpolations.

Another widespread approach to improve state-update algorithms for plasticity is the substepping scheme. This strategy is usual in explicit integration algorithms (commonly based on the forward Euler method) for elastoplastic models, as presented by Sloan (1987), Abbo and Sloan (1996), Sloan, Abbo and Sheng (2001), Jakobsen and Lade (2002), and Lloret-Cabot et al. (2016). These studies demonstrated that the substepping process affords more accuracy to numerical integration. Moreover, the size of each substep is automatically computed using the error measure adopted in the process. In general, forward Euler integration schemes perform accurately only for small increments (Sloan, 1987; Abbo and Sloan, 1996; Jakobsen and Lade, 2002). Even with this limitation,

explicit substepping schemes are widely utilized to simulate complex constitutive models since these procedures are more straightforward in terms of formulation and implementation than implicit integration. From that perspective, Quevedo, Firme and Roehl (2019) described an explicit substepping method to model complex creep behaviors in geomaterials considering subincrements of time, strain and temperature.

Regarding single-surface implicit integration, some researchers considered the substepping scheme to improve the robustness of their implementations. Pérez-Foguet, Rodríguez-Ferran and Huerta (2001) showed a general formulation of the consistent tangent modulus for substepping time-integration schemes. This formulation leads to quadratic convergence in nonlinear finite element problems. Saritas and Filippou (2009) and Ma and Chen (2016) presented an implicit integration scheme for modeling concrete using a plastic-damage constitutive model. In both works, the plastic-corrector phase uses a substepping strategy. Li and Crouch (2010) proposed a single-surface constitutive model for concrete. They adopted implicit numerical integration with line search and substepping techniques with constant substep size. Nevertheless, substepping numerical integration algorithms for multisurface plasticity are barely discussed in the literature. Adhikary et al. (2017) suggested adopting a substepping scheme in multisurface constitutive models with hardening/softening behavior. According to these authors, the return mapping process may have a nonunique solution and may compute a physically incorrect solution. It should be highlighted that these authors did not clearly present the process of implementation.

In this context, this study proposes a numerical integration procedure for multisurface plasticity models. Single-surface models are also studied since they are special cases of multisurface ones. The backward Euler scheme is adopted due to its accuracy and stability. A general framework for 2D and 3D problems is presented to integrate generic plasticity models with nonassociated hardening/softening behavior. Computational aspects of the proposed algorithm are also discussed. The line search strategy is implemented to expand the convergence region of the Newton-Raphson method employed in the solution of the system of nonlinear equations. To enhance the algorithm robustness and deal with huge strain increments, the substepping scheme completes the algorithm. This solution has

been developed in the work of Abreu, Mejia and Roehl (2022a) and is described in this chapter. Moreover, the one-step return mapping formulated by Abreu (2019) serves as basis for each substep calculation.

In opposition to the available works in the literature, the proposed algorithm adopts an exact line search: the golden section method. The presented procedure solved a Drucker-Prager return mapping, which could not be solved satisfactorily by either the traditional closest point projection algorithm or the Newton-Raphson method with typical inexact line search strategies. Therefore, the proposed method is an appropriate alternative to integration schemes for the Drucker-Prager criterion. This study also contributes to the substepping integration algorithms for multisurface plasticity, presenting an adaptive substep size scheme and the consistent tangent modulus formulation. In this context, the proposed algorithm becomes a powerful tool that can be used to assess the mathematical formulation of new plastic models straightforwardly. A novel multisurface plasticity model is utilized to test the robustness of the proposed integration algorithm. This constitutive model, named Capped Hyperbolic Generalized Classical (CHGC), consists of combining the generalized yield surface proposed by Lester and Sloan (2018), and a cap surface with hardening behavior, initially described by DiMaggio and Sandler (1971), Sandler, DiMaggio and Baladi (1976) and Sandler (1976). This model simulates the nonlinear behavior of engineering materials, such as concrete, soils and rock. Nonetheless, the complex model formulation requires a comprehensive state-update procedure.

3.1.Elastic predictor-plastic corrector algorithm

The numerical solution of plasticity problems requires a numerical scheme for the integration of elastoplastic evolution equations (Simo and Hughes, 1998; de Souza Neto, Perić and Owen, 2008). In this context, the backward Euler method provides the incremental form of Equations (2.7) and (2.8). This method updates the plastic variables from the previous global step n to the current global step $n+1$. Thus, the algorithm presented in this work is based on the elastic predictor-plastic corrector scheme presented by Simo, Kennedy and Govindjee (1988) in which the plastic correction is performed by solving the following system of equations:

$$\begin{cases} \boldsymbol{\varepsilon}_{n+1}^{p} = \boldsymbol{\varepsilon}_{n}^{p} + \sum_{i \in \mathbb{J}_{n+1}} \Delta\gamma_{i,n+1} \boldsymbol{n}_{i,n+1} \\ \boldsymbol{q}_{n+1} = \boldsymbol{q}_{n} + \sum_{i \in \mathbb{J}_{n+1}} \Delta\gamma_{i,n+1} \boldsymbol{h}_{i,n+1} \\ f_{i,n+1} = 0, \forall i \in \mathbb{J}_{n+1} \end{cases} \tag{3.1}$$

This work proposes a state-update algorithm following a substepping scheme, illustrated in Figure 2. In this sense, the strain increment $\Delta\varepsilon_{n+1}$ is split in M parts. Hence, the update from substep m to $m+1$ follows through pseudo-time subincrements $\Delta t_m, m \in \{1,2,\dots,M\}$, denoting the fraction of $\Delta\varepsilon_{n+1}$ of each substep. In a one-step return mapping, M and Δt_1 are equal to one.

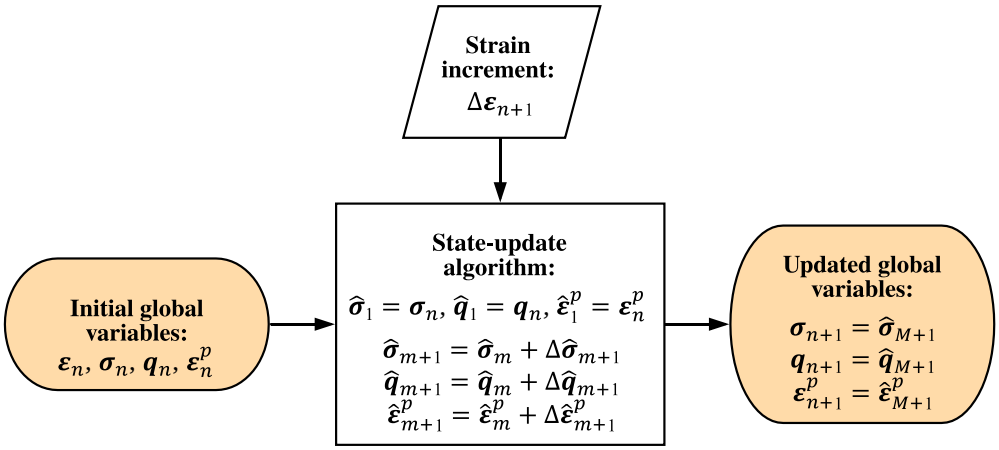

Figure 2 Substepping update procedure for the plastic variables

For the substepping scheme, the system of equations (3.1) is reformulated in terms of local variables $\hat{\boldsymbol{\sigma}}_{m+1}$, $\hat{\boldsymbol{q}}_{m+1}$, $\hat{\boldsymbol{\varepsilon}}_{m+1}^p$, and $\Delta\hat{\gamma}_{i,m+1}$. A new system of equations (3.2) is defined to integrate a generic substep m restricted to the active yield surfaces:

$$\begin{cases} \hat{\boldsymbol{\varepsilon}}_{m+1}^{p} = \hat{\boldsymbol{\varepsilon}}_{m}^{p} + \sum_{i \in \hat{\mathbb{J}}_{m+1}} \Delta\hat{\gamma}_{i,m+1} \hat{\boldsymbol{n}}_{i,m+1} \\ \hat{\boldsymbol{q}}_{m+1} = \hat{\boldsymbol{q}}_{m} + \sum_{i \in \hat{\mathbb{J}}_{m+1}} \Delta\hat{\gamma}_{i,m+1} \hat{\boldsymbol{h}}_{i,m+1} \\ \hat{f}_{i,m+1} = 0, \forall i \in \hat{\mathbb{J}}_{m+1} \end{cases} \tag{3.2}$$

The system of equations (3.2) is nonlinear in terms of the internal variables and requires an adequate numerical solution method. In this sense, the following

residual vectors are defined for the flow rule, hardening/softening law, and yield function. Notice that Equation (3.3) is determined using Equations (2.1) and (2.3).

$$\boldsymbol{r}_{1,m+1} = \boldsymbol{C}\hat{\boldsymbol{\sigma}}_{m+1} - \hat{\boldsymbol{\varepsilon}}_{m+1} + \hat{\boldsymbol{\varepsilon}}_{m}^{p} + \sum_{i \in \hat{\mathbb{J}}_{m+1}} \Delta\hat{\gamma}_{i,m+1}\hat{\boldsymbol{n}}_{i,m+1} \tag{3.3}$$

$$\boldsymbol{r}_{2,m+1} = -\hat{\boldsymbol{q}}_{m+1} + \hat{\boldsymbol{q}}_{m} + \sum_{i \in \hat{\mathbb{J}}_{m+1}} \Delta\hat{\gamma}_{i,m+1}\hat{\boldsymbol{h}}_{i,m+1} \tag{3.4}$$

$$\boldsymbol{r}_{3,m+1} = \hat{\boldsymbol{f}}_{m+1} = \begin{bmatrix} \hat{f}_{i_1,m+1} \\ \hat{f}_{i_2,m+1} \\ \vdots \end{bmatrix} = \begin{bmatrix} f_{i_1}(\hat{\boldsymbol{\sigma}}_{m+1}, \hat{\boldsymbol{q}}_{m+1}) \\ f_{i_2}(\hat{\boldsymbol{\sigma}}_{m+1}, \hat{\boldsymbol{q}}_{m+1}) \\ \vdots \end{bmatrix},\ i_1, i_2, \ldots \in \hat{\mathbb{J}}_{m+1} \tag{3.5}$$

Hence, the system of equations (3.2) becomes

$$\boldsymbol{r}_{m+1}(\hat{\boldsymbol{\sigma}}_{m+1}, \hat{\boldsymbol{q}}_{m+1}, \Delta\hat{\boldsymbol{\gamma}}_{m+1}) = \begin{bmatrix} \boldsymbol{r}_{1,m+1} \\ \boldsymbol{r}_{2,m+1} \\ \boldsymbol{r}_{3,m+1} \end{bmatrix} = \boldsymbol{0} \tag{3.6}$$

An unconstrained optimization algorithm can also be adopted to solve the system of equations (3.6). The optimization problem to be solved is defined as

$$\min_{\boldsymbol{x}_{m+1}} \psi(\boldsymbol{x}_{m+1}) \tag{3.7}$$

Thus, the objective function $\psi(\boldsymbol{x}_{m+1})$ can be defined as (Nocedal and Wright, 2006)

$$\psi(\boldsymbol{x}_{m+1}) = \frac{1}{2}\left(\boldsymbol{r}_{m+1}(\boldsymbol{x}_{m+1})\right)^{T}\boldsymbol{r}_{m+1}(\boldsymbol{x}_{m+1}) \tag{3.8}$$

For the multisurface plasticity problem, vector $\boldsymbol{x}_{m+1}$ takes the form

$$\boldsymbol{x}_{m+1} = \begin{bmatrix} \hat{\boldsymbol{\sigma}}_{m+1} \\ \hat{\boldsymbol{q}}_{m+1} \\ \Delta\hat{\boldsymbol{\gamma}}_{m+1} \end{bmatrix} \tag{3.9}$$

in which

$$\Delta\hat{\boldsymbol{\gamma}}_{m+1} = \begin{bmatrix} \Delta\hat{\gamma}_{1,m+1} \\ \Delta\hat{\gamma}_{2,m+1} \\ \vdots \\ \Delta\hat{\gamma}_{N,m+1} \end{bmatrix} \tag{3.10}$$

In this work, the Newton-Raphson method is adopted to solve the optimization problem (3.7). This method establishes a descent direction, which is calculated by a second-order Taylor expansion of ψ around $\boldsymbol{x}_{m+1}$. Consequently, the Newton-Raphson direction is given by (Nocedal and Wright, 2006):

$$\boldsymbol{d}_{m+1}^{(k)} = -\left(\boldsymbol{\mathcal{H}}_{m+1}^{(k)}\right)^{-1} \nabla\psi_{m+1}^{(k)} \tag{3.11}$$

where $\boldsymbol{\mathcal{H}}_{m+1}^{(k)}$ and $\nabla\psi_{m+1}^{(k)}$ are, respectively, the Hessian matrix and gradient vector of $\psi\left(\boldsymbol{x}_{m+1}^{(k)}\right) = \psi_{m+1}^{(k)}$ in the current iteration k. The terms $\boldsymbol{\mathcal{H}}_{m+1}^{(k)}$ and $\nabla\psi_{m+1}^{(k)}$ can be computed by the residual vector $\boldsymbol{r}_{m+1}^{(k)}$ and its Jacobian matrix $\boldsymbol{\mathcal{J}}_{m+1}^{(k)}$:

$$\nabla\psi_{m+1}^{(k)} = \nabla\psi\left(\boldsymbol{x}_{m+1}^{(k)}\right) = \left(\boldsymbol{\mathcal{J}}_{m+1}^{(k)}\right)^{T} \boldsymbol{r}_{m+1}^{(k)} \tag{3.12}$$

$$\boldsymbol{\mathcal{H}}_{m+1}^{(k)} = \left(\boldsymbol{\mathcal{J}}_{m+1}^{(k)}\right)^{T} \boldsymbol{\mathcal{J}}_{m+1}^{(k)} + \sum_{i=1}^{3} \boldsymbol{r}_{i,m+1}^{(k)} \nabla^2 \boldsymbol{r}_{i,m+1}^{(k)} \tag{3.13}$$

Nocedal and Wright (2006) discuss that the first term of Equation (3.13) usually has the major contribution to the matrix $\boldsymbol{\mathcal{H}}_{m+1}^{(k)}$. Thus, the following approximation for the calculation of the Hessian matrix is adopted:

$$\boldsymbol{\mathcal{H}}_{m+1}^{(k)} \approx \left(\boldsymbol{\mathcal{J}}_{m+1}^{(k)}\right)^{T} \boldsymbol{\mathcal{J}}_{m+1}^{(k)} \tag{3.14}$$

Therefore, the expression for the descent direction of the Newton-Raphson method is computed by Equations (3.11), (3.12) and (3.14):

$$\boldsymbol{d}_{m+1}^{(k)} = -\left(\boldsymbol{\mathcal{J}}_{m+1}^{(k)}\right)^{-1} \boldsymbol{r}_{m+1}^{(k)} \tag{3.15}$$

Note that the Newton-Raphson direction is a descent direction if the matrix $\boldsymbol{\mathcal{H}}_{m+1}^{(k)}$ is positive definite. The Newton-Raphson method fits a quadratic function around the solution $\boldsymbol{x}_{m+1}^{(k)}$ of each iteration. The quadratic approximation has a local maximum or a saddle point when $\boldsymbol{\mathcal{H}}_{m+1}^{(k)}$ is either a negative definite or an indefinite matrix, respectively. These situations are undesirable. Equation (3.15) is a descent direction if the following inequation holds (Nocedal and Wright, 2006):

$$\left(\nabla\psi_{m+1}^{(k)}\right)^T \boldsymbol{d}_{m+1}^{(k)} < 0 \tag{3.16}$$

From Equations (3.12) and (3.15), Equation (3.16) gives:

$$\left(\nabla\psi_{m+1}^{(k)}\right)^T \boldsymbol{d}_{m+1}^{(k)} = -\left(\boldsymbol{r}_{m+1}^{(k)}\right)^T \boldsymbol{r}_{m+1}^{(k)} = -2\psi_{m+1}^{(k)} < 0 \tag{3.17}$$

Therefore, the direction calculated by Equation (3.15) proves to be a descent direction. In order to avoid the complete inversion of the Jacobian matrix, the descent direction can be obtained as described below. Initially, the following system of equations is solved:

$$\boldsymbol{\mathcal{J}}_{m+1}^{(k)} \boldsymbol{d}_{m+1}^{(k)} = -\boldsymbol{r}_{m+1}^{(k)} \tag{3.18}$$

Each element of the descent direction $\boldsymbol{d}_{m+1}^{(k)}$ is related to the variables of the plasticity problem: stresses, internal variables, and plastic multipliers. In this scenario, the descent direction is defined as follows:

$$\boldsymbol{d}_{m+1}^{(k)} = \begin{bmatrix} \boldsymbol{d}_{m+1}^{\boldsymbol{\sigma}\,(k)} \\ \boldsymbol{d}_{m+1}^{\boldsymbol{q}\,(k)} \\ \boldsymbol{d}_{m+1}^{\Delta\boldsymbol{\gamma}\,(k)} \end{bmatrix} \tag{3.19}$$

considering that $\boldsymbol{d}_{m+1}^{\Delta\boldsymbol{\gamma}\,(k)}$ is defined by

$$\boldsymbol{d}_{m+1}^{\Delta\boldsymbol{\gamma}\,(k)} = \begin{bmatrix} d_{i_1,m+1}^{\Delta\boldsymbol{\gamma}\,(k)} \\ d_{i_2,m+1}^{\Delta\boldsymbol{\gamma}\,(k)} \\ \vdots \end{bmatrix},\ i_1, i_2, \ldots \in \hat{\mathbb{J}}_{m+1}^{(k)} \tag{3.20}$$

The determination of the set of active surfaces $\hat{\mathbb{J}}_{m+1}^{(k)}$ in this work is based on the proposition of Simo, Kennedy and Govindjee (1988). The starting set $\hat{\mathbb{J}}_{m+1}^{trial}$ is given by

$$\hat{\mathbb{J}}_{m+1}^{trial} = \hat{\mathbb{J}}_{m+1}^{(0)} = \left\{ i \in \{1,2,\ldots,N\} \mid \hat{f}_{i,m+1}^{(0)} > 0 \right\} \tag{3.21}$$

This set is updated during the Newton-Raphson iterations. Here, a slightly different procedure from the one described by Simo, Kennedy and Govindjee (1988) is adopted. After several numerical experiments, it was verified that a newly

computed state could accuse a negative yield function. For that reason, the proposed algorithm always checks if the yield function is positive (a tolerance may be considered) for the updated set $\hat{\mathbb{J}}_{m+1}^{(k)}$. Furthermore, Simo, Kennedy and Govindjee (1988) suggested checking the signal of $\Delta\hat{\gamma}_{i,m+1}^{(k)}$ to define this set. If the iterative process computes $\Delta\hat{\gamma}_{i,m+1}^{(k)} < 0$, the i-th yield surface is removed from set $\hat{\mathbb{J}}_{m+1}^{(k)}$. Therefore, the set of active surfaces $\hat{\mathbb{J}}_{m+1}^{(k)}$ is computed as follows:

$$\hat{\mathbb{J}}_{m+1}^{(k)} = \left\{ i \in \{1,2,\dots,N\} \,|\hat{f}_{i,m+1}^{(k)} > 0 \text{ and } \Delta\hat{\gamma}_{i,m+1}^{(k)} > 0 \right\} \tag{3.22}$$

Another problem arises when a calculated state enters the elastic domain $\mathbb{E}$ without satisfying (3.6). In this case, the set of active surfaces is redefined by $\hat{\mathbb{J}}_{m+1}^{(k)} = \hat{\mathbb{J}}_{m+1}^{trial}$.

The expression for the Jacobian matrix of $\boldsymbol{r}_{m+1}^{(k)}$ takes the form

$$\boldsymbol{\mathcal{J}}_{m+1}^{(k)} = \begin{bmatrix} \left(\boldsymbol{\Psi}_{m+1}^{(k)}\right)^{-1} & \boldsymbol{E}_{m+1}^{(k)} \\ \left(\boldsymbol{G}_{m+1}^{(k)}\right)^{T} & \boldsymbol{0} \end{bmatrix} \tag{3.23}$$

in which

$$\boldsymbol{\Psi}_{m+1}^{(k)} = \begin{bmatrix} \boldsymbol{C} + \sum_{i\in\hat{\mathbb{J}}_{m+1}^{(k)}} \Delta\hat{\gamma}_{i,m+1}^{(k)} \partial_{\boldsymbol{\sigma}} \hat{\boldsymbol{n}}_{i,m+1}^{(k)} & \sum_{i\in\hat{\mathbb{J}}_{m+1}^{(k)}} \Delta\hat{\gamma}_{i,m+1}^{(k)} \partial_{\boldsymbol{q}} \hat{\boldsymbol{n}}_{i,m+1}^{(k)} \\ \left(\sum_{i\in\hat{\mathbb{J}}_{m+1}^{(k)}} \Delta\hat{\gamma}_{i,m+1}^{(k)} \partial_{\boldsymbol{\sigma}} \hat{\boldsymbol{h}}_{i,m+1}^{(k)} \right)^{T} & -\boldsymbol{I} + \sum_{i\in\hat{\mathbb{J}}_{m+1}^{(k)}} \Delta\hat{\gamma}_{i,m+1}^{(k)} \partial_{\boldsymbol{q}} \hat{\boldsymbol{h}}_{i,m+1}^{(k)} \end{bmatrix}^{-1} \tag{3.24}$$

$$\boldsymbol{G}_{m+1}^{(k)} = \begin{bmatrix} \partial_{\boldsymbol{\sigma}} \hat{\boldsymbol{f}}_{m+1}^{(k)} \\ \partial_{\boldsymbol{q}} \hat{\boldsymbol{f}}_{m+1}^{(k)} \end{bmatrix} \tag{3.25}$$

$$\boldsymbol{E}_{m+1}^{(k)} = \begin{bmatrix} \boldsymbol{N}_{m+1}^{(k)} \\ \boldsymbol{H}_{m+1}^{(k)} \end{bmatrix} \tag{3.26}$$

where $\boldsymbol{I}$ is the identity matrix. Also,

$$\partial_{\boldsymbol{\sigma}} \hat{\boldsymbol{f}}_{m+1}^{(k)} = \begin{bmatrix} \partial_{\boldsymbol{\sigma}} \hat{f}_{i_1,m+1}^{(k)} & \partial_{\boldsymbol{\sigma}} \hat{f}_{i_2,m+1}^{(k)} & \dots \end{bmatrix}, \; i_1, i_2, \dots \in \hat{\mathbb{J}}_{m+1}^{(k)} \tag{3.27}$$

$$\partial_{\boldsymbol{q}}\hat{\boldsymbol{f}}_{m+1}^{(k)} = \left[\partial_{\boldsymbol{q}}\hat{f}_{i_1,m+1}^{(k)} \quad \partial_{\boldsymbol{q}}\hat{f}_{i_2,m+1}^{(k)} \quad \ldots\right],\ i_1, i_2, \ldots \in \hat{\mathbb{J}}_{m+1}^{(k)} \tag{3.28}$$

$$\boldsymbol{N}_{m+1}^{(k)} = \left[\hat{\boldsymbol{n}}_{i_1,m+1}^{(k)} \quad \hat{\boldsymbol{n}}_{i_2,m+1}^{(k)} \quad \ldots\right],\ i_1, i_2, \ldots \in \hat{\mathbb{J}}_{m+1}^{(k)} \tag{3.29}$$

$$\boldsymbol{H}_{m+1}^{(k)} = \left[\hat{\boldsymbol{h}}_{i_1,m+1}^{(k)} \quad \hat{\boldsymbol{h}}_{i_2,m+1}^{(k)} \quad \ldots\right],\ i_1, i_2, \ldots \in \hat{\mathbb{J}}_{m+1}^{(k)} \tag{3.30}$$

Hence, solving the system of equations in (3.18), we obtain the following solutions:

$$\boldsymbol{d}_{m+1}^{\boldsymbol{\sigma},\boldsymbol{q}\,(k)} = \begin{bmatrix}\boldsymbol{d}_{m+1}^{\boldsymbol{\sigma}\,(k)} \\ \boldsymbol{d}_{m+1}^{\boldsymbol{q}\,(k)}\end{bmatrix} = -\boldsymbol{\Psi}_{m+1}^{(k)}\left(\boldsymbol{\rho}_{m+1}^{(k)} + \boldsymbol{E}_{m+1}^{(k)}\boldsymbol{d}_{m+1}^{\Delta\boldsymbol{\gamma}\,(k)}\right) \tag{3.31}$$

$$\boldsymbol{d}_{m+1}^{\Delta\boldsymbol{\gamma}\,(k)} = \boldsymbol{\Lambda}_{m+1}^{(k)}\left(\hat{\boldsymbol{f}}_{m+1}^{(k)} - \left(\boldsymbol{G}_{m+1}^{(k)}\right)^T\boldsymbol{\Psi}_{m+1}^{(k)}\boldsymbol{\rho}_{m+1}^{(k)}\right) \tag{3.32}$$

considering that

$$\boldsymbol{\rho}_{m+1}^{(k)} = \begin{bmatrix}\boldsymbol{r}_{1,m+1}^{(k)} \\ \boldsymbol{r}_{2,m+1}^{(k)}\end{bmatrix} = \begin{bmatrix}-\hat{\boldsymbol{\varepsilon}}_{m+1}^{p\,(k)} + \hat{\boldsymbol{\varepsilon}}_m^p \\ -\hat{\boldsymbol{q}}_{m+1}^{(k)} + \hat{\boldsymbol{q}}_m\end{bmatrix} + \boldsymbol{E}_{m+1}^{(k)}\Delta\hat{\boldsymbol{\gamma}}_{m+1}^{(k)} \tag{3.33}$$

$$\boldsymbol{\Lambda}_{m+1}^{(k)} = \left(\left(\boldsymbol{G}_{m+1}^{(k)}\right)^T\boldsymbol{\Psi}_{m+1}^{(k)}\boldsymbol{E}_{m+1}^{(k)}\right)^{-1} \tag{3.34}$$

$$\hat{\boldsymbol{f}}_{m+1}^{(k)} = \begin{bmatrix}\hat{f}_{i_1,m+1}^{(k)} \\ \hat{f}_{i_2,m+1}^{(k)} \\ \vdots\end{bmatrix},\ i_1, i_2, \ldots \in \hat{\mathbb{J}}_{m+1}^{(k)} \tag{3.35}$$

A quasi-singular matrix may emerge in Equations (3.24) and (3.34). As described by Godio et al. (2016) and Adhikary et al. (2017), provided that the flow directions $\hat{\boldsymbol{n}}_{i,m+1}^{(k)}$ are linearly dependent, then the Jacobian matrix $\boldsymbol{\mathcal{J}}_{m+1}^{(k)}$ becomes a singular matrix. It should be pointed out that the hardening function adopted in this work may produce a quasi-singular Jacobian matrix with a large condition number, deteriorating the globally convergent characteristics of the Newton-Raphson method (Nocedal and Wright, 2006). To deal with matrix singularity, inverse matrices are computed by singular value decomposition, as implemented by Press et al. (2007) and Godio et al. (2016).

Next, it is fundamental to obtain the step size to update $\boldsymbol{x}_{m+1}^{(k)}$. The update uses the following equation:

$$\boldsymbol{x}_{m+1}^{(k+1)} = \boldsymbol{x}_{m+1}^{(k)} + \alpha_{m+1}^{(k)} \boldsymbol{d}_{m+1}^{(k)} \tag{3.36}$$

The step size (or step length) $\alpha_{m+1}^{(k)}$ is computed by a line search strategy by solving the optimization problem (3.37).

$$\min_{\alpha_{m+1}^{(k)}>0} \psi\left(\boldsymbol{x}_{m+1}^{(k)} + \alpha_{m+1}^{(k)} \boldsymbol{d}_{m+1}^{(k)}\right) \tag{3.37}$$

In the Newton-Raphson method, it is usual to adopt $0 < \alpha_{m+1}^{(k)} \leq 1$ (Nocedal and Wright, 2006; Press et al., 2007; Scherzinger, 2017). Finally, the variables are updated according to Equations (3.38) to (3.41). These corrections are made until residuals (3.3), (3.4) and (3.5) become negligible.

$$\widehat{\boldsymbol{\sigma}}_{m+1}^{(k+1)} = \widehat{\boldsymbol{\sigma}}_{m+1}^{(k)} + \alpha_{m+1}^{(k)} \boldsymbol{d}_{m+1}^{\boldsymbol{\sigma}\,(k)} \tag{3.38}$$

$$\widehat{\boldsymbol{q}}_{m+1}^{(k+1)} = \widehat{\boldsymbol{q}}_{m+1}^{(k)} + \alpha_{m+1}^{(k)} \boldsymbol{d}_{m+1}^{\boldsymbol{q}\,(k)} \tag{3.39}$$

$$\Delta\hat{\gamma}_{i,m+1}^{(k+1)} = \Delta\hat{\gamma}_{i,m+1}^{(k)} + \alpha_{m+1}^{(k)} d_{i,m+1}^{\Delta\gamma\,(k)},\ \forall i \in \hat{\mathbb{J}}_{m+1}^{(k+1)} \tag{3.40}$$

$$\hat{\boldsymbol{\varepsilon}}_{m+1}^{p\,(k+1)} = \hat{\boldsymbol{\varepsilon}}_{m+1}^{p\,(k)} - \alpha_{m+1}^{(k)} \boldsymbol{C} \boldsymbol{d}_{m+1}^{\boldsymbol{\sigma}\,(k)} \tag{3.41}$$

Algorithm 1 summarizes the procedure presented in this section for multisurface plasticity. It is noted that for associative plasticity and $\alpha_{m+1}^{(k)} = 1$, the proposed procedure closely reproduces the multisurface plasticity algorithm initially presented by Simo, Kennedy and Govindjee (1988).

Algorithm 1 Elastic predictor-plastic corrector algorithm for multisurface plasticity

1) Initialize: $k = 0$, Δt_m, $\boldsymbol{\varepsilon}_{n+1}$, $\boldsymbol{\varepsilon}_n$, $\hat{\boldsymbol{\varepsilon}}_m^p$, $\hat{\boldsymbol{\sigma}}_m$, $\hat{\boldsymbol{q}}_m$, $\boldsymbol{D}$, $\boldsymbol{C}$, $f_i(\boldsymbol{\sigma}, \boldsymbol{q})$, $\boldsymbol{n}_i(\boldsymbol{\sigma}, \boldsymbol{q})$, $\boldsymbol{h}_i(\boldsymbol{\sigma}, \boldsymbol{q})$, N, tol_1, tol_2, tol_3, and k_{max}.

2) Compute the elastic predictor state:

$\Delta\hat{\boldsymbol{\varepsilon}}_{m+1} = \Delta t_m(\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_n)$

$\hat{\boldsymbol{\sigma}}_{m+1}^{(k)} = \hat{\boldsymbol{\sigma}}_m + \boldsymbol{D}\Delta\hat{\boldsymbol{\varepsilon}}_{m+1}, \quad \hat{\boldsymbol{q}}_{m+1}^{(k)} = \hat{\boldsymbol{q}}_m, \quad \hat{\boldsymbol{\varepsilon}}_{m+1}^{p\,(k)} = \hat{\boldsymbol{\varepsilon}}_m^p, \quad \Delta\hat{\boldsymbol{\gamma}}_{m+1}^{(k)} = \boldsymbol{0}$

3) Define the active yield surfaces:

$$\hat{\boldsymbol{f}}_{m+1}^{trial} = \hat{\boldsymbol{f}}_{m+1}^{(k)} = \begin{bmatrix} f_1\left(\hat{\boldsymbol{\sigma}}_{m+1}^{(k)}, \hat{\boldsymbol{q}}_{m+1}^{(k)}\right) \\ f_2\left(\hat{\boldsymbol{\sigma}}_{m+1}^{(k)}, \hat{\boldsymbol{q}}_{m+1}^{(k)}\right) \\ \vdots \\ f_N\left(\hat{\boldsymbol{\sigma}}_{m+1}^{(k)}, \hat{\boldsymbol{q}}_{m+1}^{(k)}\right) \end{bmatrix} = \begin{bmatrix} \hat{f}_{1,m+1}^{trial} \\ \hat{f}_{2,m+1}^{trial} \\ \vdots \\ \hat{f}_{N,m+1}^{trial} \end{bmatrix}$$

$\hat{\mathbb{J}}_{m+1}^{trial} = \hat{\mathbb{J}}_{m+1}^{(k)} = \left\{i \in \{1,2,\dots,N\} \mid \hat{f}_{i,m+1}^{trial} > tol_1\right\}$

4) Check plastic condition:

If $\hat{\mathbb{J}}_{m+1}^{trial} = \emptyset$:

$\hat{\boldsymbol{\sigma}}_{m+1} = \hat{\boldsymbol{\sigma}}_{m+1}^{(k)}, \quad \hat{\boldsymbol{q}}_{m+1} = \hat{\boldsymbol{q}}_{m+1}^{(k)}, \quad \hat{\boldsymbol{\varepsilon}}_{m+1}^p = \hat{\boldsymbol{\varepsilon}}_{m+1}^{p\,(k)}, \quad \Delta\hat{\boldsymbol{\gamma}}_{m+1} = \Delta\hat{\boldsymbol{\gamma}}_{m+1}^{(k)}$

Return $\hat{\boldsymbol{\sigma}}_{m+1}$, $\hat{\boldsymbol{q}}_{m+1}$, $\hat{\boldsymbol{\varepsilon}}_{m+1}^p$, $\Delta\hat{\boldsymbol{\gamma}}_{m+1}$, k, $fail = false$, and EXIT.

End

5) Compute $\boldsymbol{N}_{m+1}^{(k)}$, $\boldsymbol{H}_{m+1}^{(k)}$ and $\boldsymbol{E}_{m+1}^{(k)}$ using Equations (3.29), (3.30) and (3.26), respectively.

6) Check the stopping criteria:

Compute $\boldsymbol{\rho}_{m+1}^{(k)}$, $\boldsymbol{r}_{1,m+1}^{(k)}$ and $\boldsymbol{r}_{2,m+1}^{(k)}$ using Equation (3.33).

If $\hat{f}_{i,m+1}^{(k)} \leq tol_1$ such that $\forall i \in \{1,2,\dots,N\}$, $\left\|\boldsymbol{r}_{1,m+1}^{(k)}\right\| \leq tol_2$, and $\left\|\boldsymbol{r}_{2,m+1}^{(k)}\right\| \leq tol_3$:

$$\hat{\boldsymbol{\sigma}}_{m+1} = \hat{\boldsymbol{\sigma}}_{m+1}^{(k)}, \quad \hat{\boldsymbol{q}}_{m+1} = \hat{\boldsymbol{q}}_{m+1}^{(k)}, \quad \hat{\boldsymbol{\varepsilon}}_{m+1}^p = \hat{\boldsymbol{\varepsilon}}_{m+1}^{p\,(k)}, \quad \Delta\hat{\gamma}_{i,m+1} = \begin{cases} \Delta\hat{\gamma}_{i,m+1}^{(k)}, & \forall i \in \hat{\mathbb{J}}_{m+1}^{(k)} \\ 0, & \forall i \notin \hat{\mathbb{J}}_{m+1}^{(k)} \end{cases}$$

Return $\hat{\boldsymbol{\sigma}}_{m+1}$, $\hat{\boldsymbol{q}}_{m+1}$, $\hat{\boldsymbol{\varepsilon}}_{m+1}^p$, $\Delta\hat{\boldsymbol{\gamma}}_{m+1}$, k, $fail = false$, and EXIT.

End

If $k = k_{max}$:

Return $fail = true$ and EXIT.

End

7) Compute the element $\boldsymbol{d}_{m+1}^{\Delta\boldsymbol{\gamma}\,(k)}$ of the descent direction:

Redefine $\hat{\boldsymbol{f}}_{m+1}^{(k)}$ as presented in Equation (3.35) and compute $\partial_{\boldsymbol{\sigma}}\hat{\boldsymbol{f}}_{m+1}^{(k)}$, $\partial_{\boldsymbol{q}}\hat{\boldsymbol{f}}_{m+1}^{(k)}$, $\boldsymbol{G}_{m+1}^{(k)}$, $\boldsymbol{\Psi}_{m+1}^{(k)}$, $\boldsymbol{\Lambda}_{m+1}^{(k)}$ and $\boldsymbol{d}_{m+1}^{\Delta\boldsymbol{\gamma}\,(k)}$ using Equations (3.27), (3.28), (3.25), (3.24), (3.34), and (3.32). In a general context, compute $\boldsymbol{\Psi}_{m+1}^{(k)}$ and $\boldsymbol{\Lambda}_{m+1}^{(k)}$using Singular Value Decomposition.

8) Check Karush-Kuhn-Tucker condition:

$\Delta\bar{\gamma}_{i,m+1}^{(k+1)} = \Delta\hat{\gamma}_{i,m+1}^{(k)} + d_{i,m+1}^{\Delta\boldsymbol{\gamma}\,(k)}, \ \forall i \in \hat{\mathbb{J}}_{m+1}^{(k)}$

If $\exists i \in \{1,2,\dots,N\}$ such that $\Delta\bar{\gamma}_{i,m+1}^{(k+1)} < 0$:

Go to step 12).

End

Algorithm 1 Elastic predictor-plastic corrector algorithm for multisurface plasticity (continuation)

9) Compute the elements $\boldsymbol{d}_{m+1}^{\boldsymbol{\sigma}\,(k)}$ and $\boldsymbol{d}_{m+1}^{\boldsymbol{q}\,(k)}$ of the descent direction using Equation (3.31).

10) Perform the line search algorithm to calculate $\alpha_{m+1}^{(k)}$ (Algorithm 3).

11) Update stresses, internal variables, plastic multipliers, and plastic strains:

Compute $\widehat{\boldsymbol{\sigma}}_{m+1}^{(k+1)}$, $\widehat{\boldsymbol{q}}_{m+1}^{(k+1)}$, $\Delta\hat{\gamma}_{i,m+1}^{(k+1)}$, and $\hat{\boldsymbol{\varepsilon}}_{m+1}^{p\,(k+1)}$ using Equations (3.38) to (3.41).

12) Update counter variable:

Set $k \coloneqq k+1$

13) Specify the active yield surfaces:

Redefine $\hat{\boldsymbol{f}}_{m+1}^{(k)}$ as an array consisting of N elements: the plastic functions $\hat{f}_{i,m+1}^{(k)}$ for all surfaces.

If $\exists i \in \{1,2,\dots,N\}$ such that $\Delta\bar{\gamma}_{i,m+1}^{(k)} < 0$:

$$\hat{\mathbb{J}}_{m+1}^{(k)} = \left\{ i \in \hat{\mathbb{J}}_{m+1}^{(k)} \mid \Delta\bar{\gamma}_{i,m+1}^{(k)} > 0 \right\}$$

Else

$$\hat{\mathbb{J}}_{m+1}^{(k)} = \left\{ i \in \{1,2,\dots,N\} \mid \hat{f}_{i,m+1}^{(k)} > -tol_1 \right\}$$

End

If $\hat{\mathbb{J}}_{m+1}^{(k)} = \emptyset$:

$$\hat{\mathbb{J}}_{m+1}^{(k)} = \hat{\mathbb{J}}_{m+1}^{trial}$$

End

Go to step 5).

3.2. Line search

The line search approach is a valuable strategy since it provides the Newton-Raphson method with globally convergent characteristics. For that, the step length must comply with the Wolfe conditions, and the Hessian matrix (3.14) must be positive definite and have a bounded condition number (Nocedal and Wright, 2006). In this work, two line search strategies are combined to compute the optimal value for the step size $\alpha_{m+1}^{(k)}$: the Armijo rule (Armijo, 1966) and the golden section method (Arora, 2012). The Armijo rule is a popular inexact line search strategy, applied in this work to determine if $\alpha_{m+1}^{(k)} = 1$ sufficiently reduces the objective function. This condition occurs when $\alpha_{m+1}^{(k)}$ satisfies the following inequation:

$$\psi\left(\boldsymbol{x}_{m+1}^{(k)} + \alpha_{m+1}^{(k)}\boldsymbol{d}_{m+1}^{(k)}\right) \leq \psi_{m+1}^{(k)} + c_1\alpha_{m+1}^{(k)}\left(\nabla\psi_{m+1}^{(k)}\right)^T\boldsymbol{d}_{m+1}^{(k)} \tag{3.42}$$

The condition (3.42), which is also the first Wolfe condition (Nocedal and Wright, 2006), is defined by a slope control parameter c_1. Authors such as Nocedal and Wright (2006) and Press et al. (2007) suggested $c_1 = 10^{-4}$. For simplicity purposes, inequation (3.42) is reformulated using Equation (3.17):

$$\psi\left(\boldsymbol{x}_{m+1}^{(k)} + \alpha_{m+1}^{(k)}\boldsymbol{d}_{m+1}^{(k)}\right) \leq \psi_{m+1}^{(k)} + s_{m+1}^{(k)}\alpha_{m+1}^{(k)} \tag{3.43}$$

in which $s_{m+1}^{(k)}$ is defined by

$$s_{m+1}^{(k)} = -2c_1\psi_{m+1}^{(k)} \tag{3.44}$$

Inequation (3.43) has the graphic interpretation shown in Figure 3. The right-hand side of this inequation is a linear function with a negative slope $s_{m+1}^{(k)}$. If c_1 has a small value, this function lies above the objective function for small values of $\alpha_{m+1}^{(k)}$. Hence, this region has an acceptable step length. Moreover, as Figure 3 presents, the algorithm makes reasonable progress for large values of $\alpha_{m+1}^{(k)}$ because in this region the objective function lies below the linear function. In this context, large values of $\alpha_{m+1}^{(k)}$ sufficiently decrease the objective function.

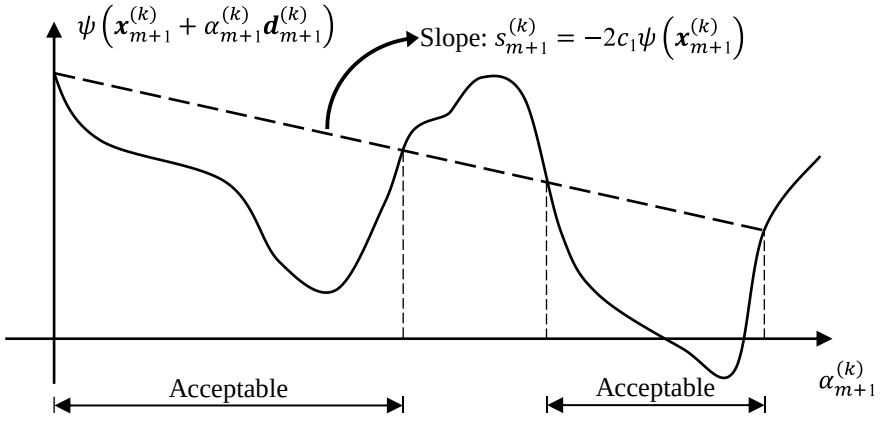

Figure 3 Graphic interpretation of the Armijo rule (adapted from Nocedal and Wright (2006))

According to the adopted line search approach, if inequation (3.43) is satisfied with $\alpha_{m+1}^{(k)} = 1$, the objective function decreases sufficiently. Thus, the full-Newton step is applied in the current iteration. However, if the rule is not satisfied, the

golden section method is adopted to solve problem (3.37). The golden section is an exact line search and an interval-reducing method; it finds an optimal solution while the objective function is evaluated at points determined by the golden ratio. In this work, the golden section method is applied in the interval $0 < \alpha_{m+1}^{(k)} < 1$.

The golden section method is an attractive exact line search since it does not require derivatives in arbitrary directions, as is the case of the Newton-Raphson and the secant methods. In addition, the golden section method uses fewer objective function evaluations per iteration than the bisection method. Algorithm 2 summarizes the Armijo rule and Algorithm 3 shows the golden section method.

Algorithm 2 Armijo rule

1) Initialize: $\psi(\boldsymbol{x}_{m+1}^{(k)})$, $\boldsymbol{x}_{m+1}^{(k)}$, $\boldsymbol{d}_{m+1}^{(k)}$, $\alpha_{m+1}^{(k)}$, and c_1.

2) Compute the slope $s_{m+1}^{(k)}$ according to Equation (3.44).

3) Perform the Armijo rule:

If $\psi(\boldsymbol{x}_{m+1}^{(k)} + \alpha_{m+1}^{(k)}\boldsymbol{d}_{m+1}^{(k)}) \leq \psi(\boldsymbol{x}_{m+1}^{(k)}) + s_{m+1}^{(k)}\alpha_{m+1}^{(k)}$:

Return $accept = true$ and EXIT.

Else:

Return $accept = false$ and EXIT.

End

Algorithm 3 Golden section method

1) Initialize: $i = 1$, $\psi\left(\boldsymbol{x}_{m+1}^{(k)}\right)$, $\boldsymbol{x}_{m+1}^{(k)}$, $\boldsymbol{d}_{m+1}^{(k)}$, $\alpha_{0,m+1}^{o\,(k)} = 0$, $\alpha_{0,m+1}^{f\,(k)} = 1$, $\varphi = \frac{\sqrt{5}-1}{2}$, and tol.

2) Perform the Armijo rule considering $\alpha_{m+1}^{(k)} = \alpha_{0,m+1}^{f\,(k)}$ (Algorithm 2):

If $accept = true$:

Return $\alpha_{m+1}^{(k)} = \alpha_{0,m+1}^{f\,(k)}$ and EXIT.

End

3) Divide the interval using the golden ratio:

$$\alpha_{i,m+1}^{r\,(k)} = \varphi\alpha_{0,m+1}^{f\,(k)}, \quad \psi_{i,m+1}^{r\,(k)} = \psi\left(\boldsymbol{x}_{m+1}^{(k)} + \alpha_{i,m+1}^{r\,(k)}\boldsymbol{d}_{m+1}^{(k)}\right)$$

$$\alpha_{i,m+1}^{l\,(k)} = (1-\varphi)\alpha_{0,m+1}^{f\,(k)}, \quad \psi_{i,m+1}^{l\,(k)} = \psi\left(\boldsymbol{x}_{m+1}^{(k)} + \alpha_{i,m+1}^{l\,(k)}\boldsymbol{d}_{m+1}^{(k)}\right)$$

4) Choose the correct interval and divide it:

If $\psi_{i,m+1}^{r\,(k)} > \psi_{i,m+1}^{l\,(k)}$:

$$\alpha_{i,m+1}^{f\,(k)} = \alpha_{i,m+1}^{r\,(k)}, \quad \alpha_{i,m+1}^{r\,(k)} = \alpha_{i,m+1}^{l\,(k)}$$

$$\theta_{i,m+1}^{(k)} = \alpha_{i,m+1}^{f\,(k)} - \alpha_{i,m+1}^{o\,(k)}, \quad \alpha_{i,m+1}^{l\,(k)} = \alpha_{i,m+1}^{o\,(k)} + (1-\varphi)\theta_{i,m+1}^{(k)}$$

$$\psi_{i+1,m+1}^{r\,(k)} = \psi_{i,m+1}^{l\,(k)}, \quad \psi_{i+1,m+1}^{l\,(k)} = \psi\left(\boldsymbol{x}_{m+1}^{(k)} + \alpha_{i,m+1}^{l\,(k)}\boldsymbol{d}_{m+1}^{(k)}\right)$$

Else:

$$\alpha_{i,m+1}^{o\,(k)} = \alpha_{i,m+1}^{l\,(k)}, \quad \alpha_{i,m+1}^{l\,(k)} = \alpha_{i,m+1}^{r\,(k)}$$

$$\theta_{i,m+1}^{(k)} = \alpha_{i,m+1}^{f\,(k)} - \alpha_{i,m+1}^{o\,(k)}, \quad \alpha_{i,m+1}^{r\,(k)} = \alpha_{i,m+1}^{o\,(k)} + \varphi\theta_{i,m+1}^{(k)}$$

$$\psi_{i+1,m+1}^{l\,(k)} = \psi_{i,m+1}^{r\,(k)}, \quad \psi_{i+1,m+1}^{r\,(k)} = \psi\left(\boldsymbol{x}_{m+1}^{(k)} + \alpha_{i,m+1}^{r\,(k)}\boldsymbol{d}_{m+1}^{(k)}\right)$$

End

5) Check the stopping criterion:

If $\theta_{i,m+1}^{(k)} < tol$:

$$\alpha_{m+1}^{(k)} = \frac{\alpha_{i,m+1}^{o\,(k)} + \alpha_{i,m+1}^{f\,(k)}}{2}$$

Return $\alpha_{m+1}^{(k)}$ and EXIT.

Else:

Set $i \coloneqq i + 1$ and go to step 4).

End

3.3. Substepping scheme

Hardening and softening behavior in plasticity models and huge strain increments can lead to severe convergence problems by the solution of the system of equations (3.1). In order to overcome this problem, a multistep scheme is adopted. In this technique, the strain increment used in the state-update procedure

is divided into subincrements. In the finite element context, the plastic variables for all Gauss points are computed from a fraction of the total strain increment for each substep.

In the substepping technique, the summation of all pseudo-time subincrements Δt_m is equal to one. An adaptive substepping scheme is applied to compute adequate pseudo-times that control the fraction of the total strain increment. The adopted strategy in this work is based on the scheme presented by Ramm (1981). Therefore, the factor δ_m adjusts the current pseudo-time Δt_m to the next pseudo-time $\delta_m \Delta t_m$,

$$\delta_m = \max\left(\min\left(\sqrt{k_d/k}\,, \sqrt{k_d}\right), 0.5\right) \tag{3.45}$$

where k is the number of iterations in the Newton-Raphson process, k_d represents the desired number of iterations, and Δt_{min} defines the minimum pseudo-time subincrement. If the plastic correction is not necessary, then $\delta_m = \sqrt{k_d}$.

In this work, the first attempt of pseudo-time subincrement is $\Delta t_1 = 1$. If the algorithm fails to update stresses, strains and internal variables, a new attempt is performed considering a subincrement of half the size. Hence, global state variables are not updated. On the other hand, when the algorithm is successful, these variables are updated to compute the following substep.

3.4. Consistent tangent modulus

The consistent elastoplastic tangent modulus must be defined to preserve the quadratic rate of convergence of the Newton-Raphson method in the context of nonlinear finite element analyses. Using numerical examples, Simo and Taylor (1985) presented that the consistency between the integration scheme and the tangent modulus is fundamental to reach quadratic convergence in the global Newton-Raphson method. Thus, the consistent tangent modulus $\boldsymbol{D}_{n+1}^{ep}$ is computed as follows:

$$\boldsymbol{D}_{n+1}^{ep} = \frac{d\boldsymbol{\sigma}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} \tag{3.46}$$

Initially, considering the return mapping of a generic substep, tensor $\widehat{\boldsymbol{D}}_{m+1}^{ep}$ is obtained from the system of equations (3.2). For this purpose, each equation of this system is differentiated. Thus, the following equations are obtained using the chain rule:

$$\frac{d\hat{\boldsymbol{\varepsilon}}_{m+1}^{p}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} = \sum_{i\in\hat{\mathbb{J}}_{m+1}} \Delta\hat{\gamma}_{i,m+1}\partial_{\boldsymbol{\sigma}}\widehat{\boldsymbol{n}}_{i,m+1}\frac{d\widehat{\boldsymbol{\sigma}}_{m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} + \sum_{i\in\hat{\mathbb{J}}_{m+1}} \Delta\hat{\gamma}_{i,m+1}\partial_{\boldsymbol{q}}\widehat{\boldsymbol{n}}_{i,m+1}\frac{d\widehat{\boldsymbol{q}}_{m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} + \sum_{i\in\hat{\mathbb{J}}_{m+1}} \widehat{\boldsymbol{n}}_{i,m+1}\frac{d\Delta\hat{\gamma}_{i,m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} \tag{3.47}$$

$$\frac{d\widehat{\boldsymbol{q}}_{m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} = \sum_{i\in\hat{\mathbb{J}}_{m+1}} \Delta\hat{\gamma}_{i,m+1}\partial_{\boldsymbol{\sigma}}\widehat{\boldsymbol{h}}_{i,m+1}\frac{d\widehat{\boldsymbol{\sigma}}_{m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} + \sum_{i\in\hat{\mathbb{J}}_{m+1}} \Delta\hat{\gamma}_{i,m+1}\partial_{\boldsymbol{q}}\widehat{\boldsymbol{h}}_{i,m+1}\frac{d\widehat{\boldsymbol{q}}_{m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} + \sum_{i\in\hat{\mathbb{J}}_{m+1}} \widehat{\boldsymbol{h}}_{i,m+1}\frac{d\Delta\hat{\gamma}_{i,m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} \tag{3.48}$$

$$\partial_{\boldsymbol{\sigma}}\hat{f}_{i,m+1}\frac{d\widehat{\boldsymbol{\sigma}}_{m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} + \partial_{\boldsymbol{q}}\hat{f}_{i,m+1}\frac{d\widehat{\boldsymbol{q}}_{m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} = 0,\ \forall i \in \hat{\mathbb{J}}_{m+1} \tag{3.49}$$

Furthermore, Equation (2.1) is differentiated to support the consistent tangent modulus computation.

$$\frac{d\hat{\boldsymbol{\varepsilon}}_{m+1}^{p}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} = \frac{d\hat{\boldsymbol{\varepsilon}}_{m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} - \frac{d\hat{\boldsymbol{\varepsilon}}_{m+1}^{e}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} = \boldsymbol{I} - \frac{d\hat{\boldsymbol{\varepsilon}}_{m+1}^{e}}{d\widehat{\boldsymbol{\sigma}}_{m+1}}\frac{d\widehat{\boldsymbol{\sigma}}_{m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} = \boldsymbol{I} - \boldsymbol{C}\frac{d\widehat{\boldsymbol{\sigma}}_{m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} \tag{3.50}$$

Therefore, the following system of equations in matrix form is computed, utilizing Equations (3.47) to (3.50):

$$\boldsymbol{\mathcal{J}}_{m+1}\begin{bmatrix} \dfrac{d\widehat{\boldsymbol{\sigma}}_{m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} \\ \dfrac{d\widehat{\boldsymbol{q}}_{m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} \\ \dfrac{d\Delta\widehat{\boldsymbol{\gamma}}_{m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} \end{bmatrix} = \begin{bmatrix} (\boldsymbol{\Psi}_{m+1})^{-1} & \boldsymbol{E}_{m+1} \\ (\boldsymbol{G}_{m+1})^{T} & \boldsymbol{0} \end{bmatrix}\begin{bmatrix} \dfrac{d\widehat{\boldsymbol{\sigma}}_{m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} \\ \dfrac{d\widehat{\boldsymbol{q}}_{m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} \\ \dfrac{d\Delta\widehat{\boldsymbol{\gamma}}_{m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \tag{3.51}$$

where $\boldsymbol{\mathcal{J}}_{m+1}$ is the Jacobian matrix of ψ_{m+1}. The inverse matrix of $\boldsymbol{\mathcal{J}}_{m+1}$ is also defined here.

$$(\boldsymbol{\mathcal{J}}_{m+1})^{-1} = \begin{bmatrix} \boldsymbol{D}_{11,m+1} & \boldsymbol{D}_{12,m+1} & \boldsymbol{D}_{13,m+1} \\ \boldsymbol{D}_{21,m+1} & \boldsymbol{D}_{22,m+1} & \boldsymbol{D}_{23,m+1} \\ \boldsymbol{D}_{31,m+1} & \boldsymbol{D}_{32,m+1} & \boldsymbol{D}_{33,m+1} \end{bmatrix} \tag{3.52}$$

Using Equation (3.52) and the system of equations (3.51), the consistent tangent modulus $\widehat{\boldsymbol{D}}_{m+1}^{ep}$ takes the following form:

$$\widehat{\boldsymbol{D}}_{m+1}^{ep} = \frac{d\widehat{\boldsymbol{\sigma}}_{m+1}}{d\widehat{\boldsymbol{\varepsilon}}_{m+1}} = \boldsymbol{D}_{11,m+1} \tag{3.53}$$

Note that $\boldsymbol{D}_{11,m+1}$ has dimension equal to the number of stress (and strain) components. The Banachiewicz-Schur form for block matrix inversion is applied to obtain that matrix, as described by Baksalary and Styan (2002). Then, a block matrix $\boldsymbol{M}_0$ is given by

$$\boldsymbol{M}_0 = \begin{bmatrix} \boldsymbol{A}_0 & \boldsymbol{B}_0 \\ \boldsymbol{C}_0 & \boldsymbol{D}_0 \end{bmatrix} \tag{3.54}$$

Given $\boldsymbol{S}_0 = \boldsymbol{D}_0 - \boldsymbol{C}_0\boldsymbol{A}_0^{-1}\boldsymbol{B}_0$, matrix $\boldsymbol{M}_0$ can be inverted blockwise as follows:

$$\boldsymbol{M}_0^{-1} = \begin{bmatrix} \boldsymbol{A}_0^{-1} + \boldsymbol{A}_0^{-1}\boldsymbol{B}_0\boldsymbol{S}_0^{-1}\boldsymbol{C}_0\boldsymbol{A}_0^{-1} & -\boldsymbol{A}_0^{-1}\boldsymbol{B}_0\boldsymbol{S}_0^{-1} \\ -\boldsymbol{S}_0^{-1}\boldsymbol{C}_0\boldsymbol{A}_0^{-1} & \boldsymbol{S}_0^{-1} \end{bmatrix} \tag{3.55}$$

Thus, the square matrix (3.55) is utilized to compute the inverse matrix of $\boldsymbol{\mathcal{J}}_{m+1}$ according to Equation (3.52). Based on Equation (3.23), matrices $\boldsymbol{A}_0$, $\boldsymbol{B}_0$, $\boldsymbol{C}_0$ and $\boldsymbol{D}_0$ are defined by

$$\boldsymbol{A}_0 = (\boldsymbol{\Psi}_{m+1})^{-1} \tag{3.56}$$

$$\boldsymbol{B}_0 = \boldsymbol{E}_{m+1} \tag{3.57}$$

$$\boldsymbol{C}_0 = (\boldsymbol{G}_{m+1})^T \tag{3.58}$$

$$\boldsymbol{D}_0 = \boldsymbol{0} \tag{3.59}$$

$$\boldsymbol{S}_0 = -(\boldsymbol{G}_{m+1})^T\boldsymbol{\Psi}_{m+1}\boldsymbol{E}_{m+1} = -(\boldsymbol{\Lambda}_{m+1})^{-1} \tag{3.60}$$

Then, the first term of the matrix $\boldsymbol{M}_0^{-1}$ defines

$$\begin{bmatrix} \boldsymbol{D}_{11,m+1} & \boldsymbol{D}_{12,m+1} \\ \boldsymbol{D}_{21,m+1} & \boldsymbol{D}_{22,m+1} \end{bmatrix} = \boldsymbol{\Psi}_{m+1}(\boldsymbol{I} - \boldsymbol{E}_{m+1}\boldsymbol{\Lambda}_{m+1}(\boldsymbol{G}_{m+1})^T\boldsymbol{\Psi}_{m+1}) \tag{3.61}$$

The consistent tangent modulus for the return mapping of a generic substep is represented by the term $\boldsymbol{D}_{11,m+1}$ of Equation (3.61), according to Equation (3.53). Then, considering a substepping technique, an expression of consistent tangent modulus can be derived for multisurface plasticity. The proposed formulation of the tangent modulus is based on the work by Pérez-Foguet, Rodríguez-Ferran and Huerta (2001), in which the authors treated a single-surface plasticity problem. Initially, a fraction of the strain increment is computed from the total strain increment in each substep as given:

$$\Delta\hat{\boldsymbol{\varepsilon}}_{m+1} = \Delta t_m(\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_n) = \Delta t_m\left(\hat{\boldsymbol{\varepsilon}}_{m+1} + \left(1 - \sum_{i=1}^{m} \Delta t_i\right)\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_n\right) \tag{3.62}$$

The consistent tangent modulus $\widehat{\boldsymbol{D}}^{ep}_{m+1}$ for the substepping scheme requires some modifications in (3.2). The first equation of (3.2) is reformulated as follows:

$$\boldsymbol{C}\hat{\boldsymbol{\sigma}}_{m+1} + \sum_{i\in\hat{\mathbb{J}}_{m+1}} \Delta\hat{\gamma}_{i,m+1}\hat{\boldsymbol{n}}_{i,m+1} = \boldsymbol{C}\hat{\boldsymbol{\sigma}}_m + \Delta\hat{\boldsymbol{\varepsilon}}_{m+1} \tag{3.63}$$

Thus, considering (3.63), the system of equations (3.2) becomes

$$\begin{cases} \boldsymbol{C}\hat{\boldsymbol{\sigma}}_{m+1} + \sum_{i\in\hat{\mathbb{J}}_{m+1}} \Delta\hat{\gamma}_{i,m+1}\hat{\boldsymbol{n}}_{i,m+1} = \boldsymbol{C}\hat{\boldsymbol{\sigma}}_m + \Delta\hat{\boldsymbol{\varepsilon}}_{m+1} \\ -\hat{\boldsymbol{q}}_{m+1} + \sum_{i\in\hat{\mathbb{J}}_{m+1}} \Delta\hat{\gamma}_{i,m+1}\hat{\boldsymbol{h}}_{i,m+1} = -\hat{\boldsymbol{q}}_m \\ \hat{f}_{i,m+1} = 0,\ \forall i \in \hat{\mathbb{J}}_{m+1} \end{cases} \tag{3.64}$$

All equations of system (3.64) are differentiated as in Equations (3.47) to (3.49). Therefore, the following equations are obtained using the chain rule in order to replace Equations (3.47) and (3.48) since Equation (3.49) is still valid.

$$\begin{aligned} \boldsymbol{C}\frac{d\hat{\boldsymbol{\sigma}}_{m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} &+ \sum_{i\in\hat{\mathbb{J}}_{m+1}} \Delta\hat{\gamma}_{i,m+1}\partial_{\boldsymbol{\sigma}}\hat{\boldsymbol{n}}_{i,m+1}\frac{d\hat{\boldsymbol{\sigma}}_{m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} \\ &+ \sum_{i\in\hat{\mathbb{J}}_{m+1}} \Delta\hat{\gamma}_{i,m+1}\partial_{\boldsymbol{q}}\hat{\boldsymbol{n}}_{i,m+1}\frac{d\hat{\boldsymbol{q}}_{m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} \\ &+ \sum_{i\in\hat{\mathbb{J}}_{m+1}} \hat{\boldsymbol{n}}_{i,m+1}\frac{d\Delta\hat{\gamma}_{i,m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} = \boldsymbol{C}\frac{d\hat{\boldsymbol{\sigma}}_m}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} + \frac{d\Delta\hat{\boldsymbol{\varepsilon}}_{m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}} \end{aligned} \tag{3.65}$$

$$-\frac{d\widehat{\boldsymbol{q}}_{m+1}}{d\widehat{\boldsymbol{\varepsilon}}_{m+1}}+\sum_{i\in\mathbb{J}_{m+1}}\Delta\hat{\gamma}_{i,m+1}\partial_{\boldsymbol{\sigma}}\widehat{\boldsymbol{h}}_{i,m+1}\frac{d\widehat{\boldsymbol{\sigma}}_{m+1}}{d\widehat{\boldsymbol{\varepsilon}}_{m+1}}+\sum_{i\in\mathbb{J}_{m+1}}\Delta\hat{\gamma}_{i,m+1}\partial_{\boldsymbol{q}}\widehat{\boldsymbol{h}}_{i,m+1}\frac{d\widehat{\boldsymbol{q}}_{m+1}}{d\widehat{\boldsymbol{\varepsilon}}_{m+1}}+\sum_{i\in\mathbb{J}_{m+1}}\widehat{\boldsymbol{h}}_{i,m+1}\frac{d\Delta\hat{\gamma}_{i,m+1}}{d\widehat{\boldsymbol{\varepsilon}}_{m+1}}=-\frac{d\widehat{\boldsymbol{q}}_{m}}{d\widehat{\boldsymbol{\varepsilon}}_{m+1}} \tag{3.66}$$

Hence, the following system of equations in matrix form is computed utilizing Equations (3.49), (3.65) and (3.66):

$$\boldsymbol{\mathcal{J}}_{m+1}\begin{bmatrix}\frac{d\widehat{\boldsymbol{\sigma}}_{m+1}}{d\widehat{\boldsymbol{\varepsilon}}_{m+1}}\\ \frac{d\widehat{\boldsymbol{q}}_{m+1}}{d\widehat{\boldsymbol{\varepsilon}}_{m+1}}\\ \frac{d\Delta\widehat{\boldsymbol{\gamma}}_{m+1}}{d\widehat{\boldsymbol{\varepsilon}}_{m+1}}\end{bmatrix}=\begin{bmatrix}\boldsymbol{C}&\boldsymbol{0}&\boldsymbol{0}\\ \boldsymbol{0}&-\boldsymbol{I}&\boldsymbol{0}\\ \boldsymbol{0}&\boldsymbol{0}&\boldsymbol{0}\end{bmatrix}\begin{bmatrix}\frac{d\widehat{\boldsymbol{\sigma}}_{m}}{d\widehat{\boldsymbol{\varepsilon}}_{m}}\\ \frac{d\widehat{\boldsymbol{q}}_{m}}{d\widehat{\boldsymbol{\varepsilon}}_{m}}\\ \frac{d\Delta\widehat{\boldsymbol{\gamma}}_{m}}{d\widehat{\boldsymbol{\varepsilon}}_{m}}\end{bmatrix}\frac{d\widehat{\boldsymbol{\varepsilon}}_{m}}{d\widehat{\boldsymbol{\varepsilon}}_{m+1}}+\begin{bmatrix}\frac{d\Delta\widehat{\boldsymbol{\varepsilon}}_{m+1}}{d\widehat{\boldsymbol{\varepsilon}}_{m+1}}\\ \boldsymbol{0}\\ \boldsymbol{0}\end{bmatrix} \tag{3.67}$$

Considering (3.52) and the derivatives of $\Delta\widehat{\boldsymbol{\varepsilon}}_{m+1}$ and $\widehat{\boldsymbol{\varepsilon}}_{m}$ with respect to $\widehat{\boldsymbol{\varepsilon}}_{m+1}$, the solution of the system of equations (3.67) is given by the following recursive equations:

$$\widehat{\boldsymbol{D}}^{ep}_{m+1}=\frac{d\widehat{\boldsymbol{\sigma}}_{m+1}}{d\widehat{\boldsymbol{\varepsilon}}_{m+1}}=\boldsymbol{D}_{11,m+1}\left(\boldsymbol{C}\frac{d\widehat{\boldsymbol{\sigma}}_{m}}{d\widehat{\boldsymbol{\varepsilon}}_{m}}+\Delta t_m\boldsymbol{I}\right)-\boldsymbol{D}_{12,m+1}\frac{d\widehat{\boldsymbol{q}}_{m}}{d\widehat{\boldsymbol{\varepsilon}}_{m}} \tag{3.68}$$

$$\frac{d\widehat{\boldsymbol{q}}_{m+1}}{d\widehat{\boldsymbol{\varepsilon}}_{m+1}}=\boldsymbol{D}_{21,m+1}\left(\boldsymbol{C}\frac{d\widehat{\boldsymbol{\sigma}}_{m}}{d\widehat{\boldsymbol{\varepsilon}}_{m}}+\Delta t_m\boldsymbol{I}\right)-\boldsymbol{D}_{22,m+1}\frac{d\widehat{\boldsymbol{q}}_{m}}{d\widehat{\boldsymbol{\varepsilon}}_{m}} \tag{3.69}$$

where $\boldsymbol{D}_{11,m+1}$, $\boldsymbol{D}_{12,m+1}$, $\boldsymbol{D}_{21,m+1}$ and $\boldsymbol{D}_{22,m+1}$ are defined in Equation (3.61). Thus, using Equations (3.68) and (3.69), the tangent modulus can be computed consistently with the presented implicit substepping integration. Notice that these equations provide nonsymmetric matrices, even for associated plasticity (Ding et al., 2015).

The proposed substepping scheme is summarized in Algorithm 4, considering the computation of the consistent tangent modulus for multisurface plasticity and the substepping scheme. If the substepping scheme leads to Δt_m smaller than a predefined limit Δt_{min}, the procedure is interrupted. Thus, the finite element program returns to the previous converged global step and attempts a smaller global

increment. Figure 4 shows the flowchart of the proposed implicit substepping integration algorithm.

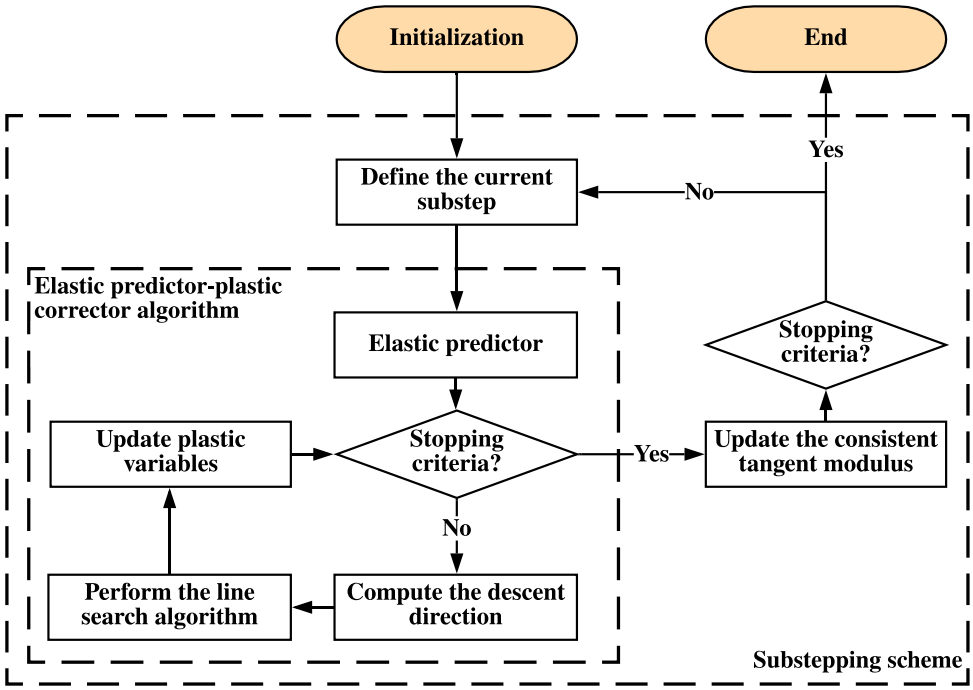

Figure 4 Flowchart of the proposed state-update algorithm

Algorithm 4 Substepping scheme

1) Initialize: $m = 1$, $t = 0$, $\Delta t_1 = 1$, Δt_{min}, k_d, $\boldsymbol{\varepsilon}_n$, $\boldsymbol{\varepsilon}_n^p$, $\boldsymbol{\sigma}_n$, and $\boldsymbol{q}_n$.

2) Initialize the local plastic variables $\hat{\boldsymbol{\sigma}}_1 = \boldsymbol{\sigma}_n$, $\hat{\boldsymbol{q}}_1 = \boldsymbol{q}_n$, $\hat{\boldsymbol{\varepsilon}}_1^p = \boldsymbol{\varepsilon}_n^p$, $\frac{d\hat{\boldsymbol{\sigma}}_1}{d\hat{\boldsymbol{\varepsilon}}_1} = \mathbf{0}$, and $\frac{d\hat{\boldsymbol{q}}_1}{d\hat{\boldsymbol{\varepsilon}}_1} = \mathbf{0}$.

3) Update state for the current Δt_m:

Set $t := t + \Delta t_m$. Perform the elastic predictor-plastic corrector algorithm to compute $\hat{\boldsymbol{\sigma}}_{m+1}$, $\hat{\boldsymbol{q}}_{m+1}$, $\hat{\boldsymbol{\varepsilon}}_{m+1}^p$, $\Delta\hat{\boldsymbol{\gamma}}_{m+1}$, k, and $fail$ according to Algorithm 1.

If $fail = true$:

Set $t := t - \Delta t_m$

Set $\Delta t_m := 0.5\Delta t_m$

Go to step 5).

Else:

If $k \neq 0$:

$$\delta_m = \max\left(\sqrt{k_d/k}, 0.5\right)$$

Else:

$$\delta_m = \sqrt{k_d}$$

End

$\Delta t_{m+1} = \min(\delta_m \Delta t_m, 1 - t)$

End

4) Update the consistent tangent modulus:

Update $\boldsymbol{\Psi}_{m+1}$, $\boldsymbol{G}_{m+1}$, $\boldsymbol{E}_{m+1}$, and $\boldsymbol{\Lambda}_{m+1}$ according to Equations (3.24), (3.25), (3.26), and (3.34), respectively. Compute $\boldsymbol{D}_{11,m+1}$, $\boldsymbol{D}_{12,m+1}$, $\boldsymbol{D}_{21,m+1}$, and $\boldsymbol{D}_{22,m+1}$ using Equation (3.61). Compute $\frac{d\hat{\boldsymbol{\sigma}}_{m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}}$ and $\frac{d\hat{\boldsymbol{q}}_{m+1}}{d\hat{\boldsymbol{\varepsilon}}_{m+1}}$ using Equations (3.68) and (3.69), respectively.

5) Check the stopping criteria:

If $\Delta t_m < \Delta t_{min}$ and $\delta_m < 1$:

The algorithm fails and the procedure is interrupted. Then, EXIT so that the finite element program returns to the previous converged global step and attempts a smaller global increment.

End

If $t = 1$:

Define M as the number of required substeps ($M = m$).

$$\boldsymbol{\sigma}_{n+1} = \hat{\boldsymbol{\sigma}}_{M+1}, \quad \boldsymbol{q}_{n+1} = \hat{\boldsymbol{q}}_{M+1}, \quad \boldsymbol{\varepsilon}_{n+1}^p = \hat{\boldsymbol{\varepsilon}}_{M+1}^p, \quad \boldsymbol{D}_{n+1}^{ep} = \frac{d\hat{\boldsymbol{\sigma}}_{M+1}}{d\hat{\boldsymbol{\varepsilon}}_{M+1}}$$

Return $\boldsymbol{\sigma}_{n+1}$, $\boldsymbol{q}_{n+1}$, $\boldsymbol{\varepsilon}_{n+1}^p$ and $\boldsymbol{D}_{n+1}^{ep}$, and EXIT.

End

If $fail = false$:

Set $m := m + 1$

End

Go to step 3).

4
Applications of the implicit substepping integration algorithm

In this chapter, several problems are used to assess the proposed implicit integration scheme. Linear elasticity was adopted while single and multisurface plasticity models were studied at local and global levels.

4.1.
Elastoplastic model formulation

A new multisurface plastic model is applied to evaluate the state-update procedure. Also known as Hyperbolic Generalized Classical yield surface, the model developed by Lester and Sloan (2018) is the main surface of the proposed model. It requires shape parameters for the definition of the deviatoric and meridional sections of the surface, as presented by Lester and Sloan (2018) and Lagioia and Panteghini (2016). Perfect plastic behavior is adopted for this yield surface. A secondary yield function, a cap surface, is adapted from the works by Hofstetter, Simo and Taylor (1993) and Schwer and Murray (1994). In general, it is used in engineering applications to control the volumetric plastic strain evolution under hydrostatic compression. For this purpose, a nonassociated hardening law is adopted. The proposed constitutive relation, named Capped Hyperbolic Generalized Classical (CHGC) model, is sufficiently complex to require a line search strategy and the substepping scheme simultaneously. The formulation of the CHGC model and some auxiliary equations are fully presented in Appendices A and B. The former surface is described in Equation (4.1):

$$f_1(\boldsymbol{\sigma}) = F + \sqrt{a^2 M^2 + J^2 \Pi^2} \tag{4.1}$$

where a is a user-defined parameter that defines the hyperbolic approximation and

$$F = Mp - K \tag{4.2}$$

$$M = \frac{6\sin\phi}{\sqrt{3}(3-\sin\phi)} \tag{4.3}$$

$$K = \frac{6\cos\phi}{\sqrt{3}(3-\sin\phi)}c \tag{4.4}$$

$$\Pi = \alpha\cos\left(\frac{\pi}{6}(2-\gamma) - \frac{1}{3}\arccos(\beta\sin 3\theta)\right) \tag{4.5}$$

Parameters c and ϕ respectively represent the cohesion and friction angle. In order to define the nonassociated flow rule $\boldsymbol{n}_1(\boldsymbol{\sigma}, \boldsymbol{q})$, the friction angle must be replaced by the dilation angle ψ in Equation (4.1). In addition, α, β and γ are shape parameters presented by Lagioia and Panteghini (2016) and Lester and Sloan (2018). Each parameter value depends on the desired surface. Importantly, the following stress invariants were adopted in this work: $p = I_1/3, J = \sqrt{J_2}, q = \sqrt{3J_2}$, and the Lode angle θ. In this context, I_1 represents the first invariant of the stress tensor and J_2 represents the second invariant of the deviatoric stress tensor.

Following the traditional mechanics convention, tension is assumed to be positive in this formulation. The presented model can simulate several perfect elastoplastic criteria, as well as their rounded versions. These rounded models are defined according to parameters a and β (Lagioia and Panteghini, 2016; Lester and Sloan, 2018). They can eliminate "corners" (nonsmooth transitions) of this surface, generating regions of high curvature.

The proposed cap surface is described in Equation (4.6). There are distinct formulations of this yield surface in the literature. In this work, the form presented by Hofstetter, Simo and Taylor (1993) is combined with Equation (4.1). Thus, the cap surface is given by

$$f_2(\boldsymbol{\sigma}, \boldsymbol{q}) = F_L + \sqrt{a^2M^2 + J^2\Pi^2 + \left(\frac{I_1 - L(\kappa)}{R}\right)^2}, \text{ for } X(\kappa) \leq I_1 \leq L(\kappa) \tag{4.6}$$

considering

$$F_L = \frac{1}{3}ML(\kappa) - K \tag{4.7}$$

The parameter $L(\kappa)$ defines the center of the cap. Additionally, the parameter κ is the internal hardening variable which defines the tensor $\boldsymbol{q}$. The definition of $L(\kappa)$ depends on an initial internal variable value κ_0 that represents the initial position of the cap, according to the following equation:

$$L(\kappa) = \min(\kappa, \kappa_0) \tag{4.8}$$

The hardening behavior of this surface is described by a relation between volumetric plastic strain ε_v^p and the cap limit position $X(\kappa)$. Then, this hardening rule is defined by

$$\varepsilon_v^p = W\left(\exp\left(D\big(X(\kappa) - X(\kappa_0)\big)\right) - 1\right) \tag{4.9}$$

where W and D are material parameters, and

$$X(\kappa) = L(\kappa) - RF_L\big(L(\kappa)\big) \tag{4.10}$$

Figure 5 shows a graphical representation of the CHGC model in the I_1-J plane. The CHGC model has some high curvature regions, introducing further difficulties in the classical closest point projection algorithm. The biaxial strength envelope of the CHGC model, which considers the third principal stress equal to zero, is presented in Figure 6. The Mohr-Coulomb shape parameters are adopted to generate this envelope. A value less than the unity for shape parameter β provides a rounded version of the Mohr-Coulomb surface. Notice that the closer the parameter β is to unity, the higher the curvatures presented in Figure 6. Details about the smoothing of the deviatoric and meridional sections of this failure envelope are given by Lagioia and Panteghini (2016) and Lester and Sloan (2018), respectively.

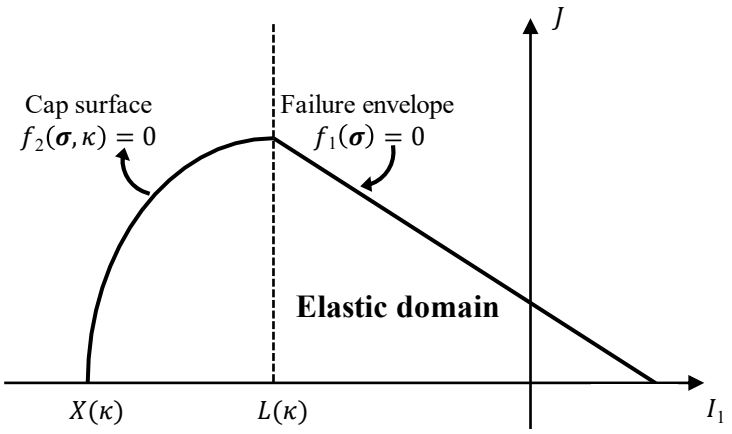

Figure 5 Capped Hyperbolic Generalized Classical (CHGC) model in meridional plane

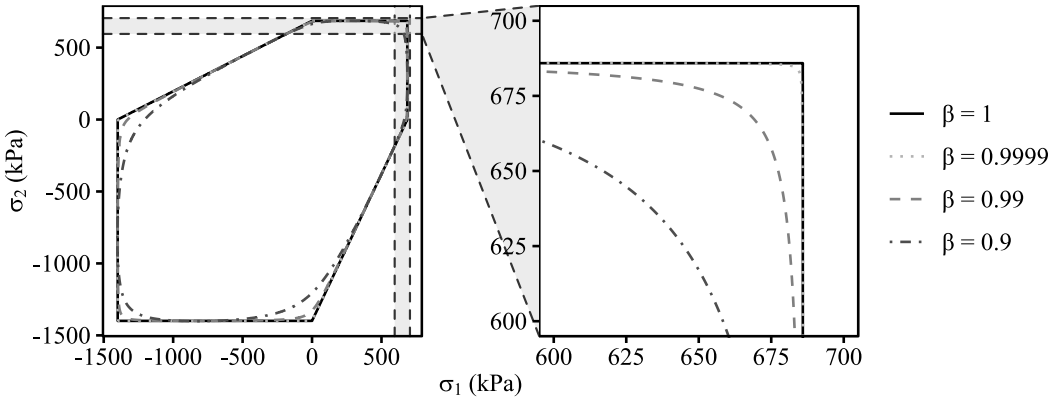

Figure 6 Biaxial strength envelope (σ_1, σ_2) of the CHGC model, as well as a zoom-in at the tensile corner, for $\sigma_3 = 0$, cohesion $c = 490$ kPa, and friction angle $\phi = 20°$

Initially, problems at the material point (Gauss point) were analyzed. The stress path was investigated in detail in each iteration from different trial stresses. The study adopts the Drucker-Prager, Mohr-Coulomb and CHGC models as described in Appendix B. Next, the footing collapse problems presented by de Souza Neto, Perić and Owen (2008) were analyzed considering the finite element method and the single and multisurface models. For the Mohr-Coulomb criterion, the well-known Prandtl solution (Chen, 1975) and the one obtained by Cox, Eason and Hopkins (1961) compute the limit pressure of strip- and circular-footings, respectively.

The proposed state-update procedure and the constitutive models were implemented in the nonlinear finite element in-house framework named GeMA (Geo Modeling Analysis). This framework has been developed at Tecgraf Institute/PUC-Rio and supports the development of new multiphysics simulators (Mendes, Gattass and Roehl, 2016). The framework provides a Newton-Raphson

procedure to prescribe displacements at particular nodes. This process is fundamental to correctly simulate the footings since they are assumed rigid and without friction at the interface between footing and soil. For the return mapping process, unless otherwise specified, it considers $k_d = 8$, $k_{max} = 50$ and $\Delta t_{min} = 10^{-4}$. In addition, the tolerance adopted for the golden section method is equal to 10^{-6}. Lastly, the global and local residual error tolerances are equal to 10^{-9}.

The main material properties in all problems are: Young's modulus $E = 10$ GPa, Poisson's ratio $\nu = 0.48$, cohesion $c = 490$ kPa, friction angle $\phi = 20°$, and dilation angle $\psi = 20°$. The parameters that define the cap surface are $R = 3$, $\kappa_0 = -7$ MPa, $W = 0.1$, and $D = 10^{-4}$ MPa^{-1}. As the Prandtl solution predicts the limit load of strip-footings using the Mohr-Coulomb parameters c and ϕ, the Drucker-Prager parameters must be adjusted so that problems modeled using the Drucker-Prager and Mohr-Coulomb materials produce the same limit load. Chen and Mizuno (1990) proposed a Drucker-Prager approximation to the Mohr-Coulomb surface so that both models produce identical limit loads for plane strain conditions. Adapting the expressions presented by these authors to the model presented by Lester and Sloan (2018), the Drucker-Prager parameters are defined as

$$c = \frac{3 - \sin\phi}{2\cos\phi\sqrt{3 + 4\tan^2\phi_{mc}}}\, c_{mc} \tag{4.11}$$

$$\phi = \arcsin\left(\frac{3\tan\phi_{mc}}{\tan\phi_{mc} + 2\sqrt{3 + 4\tan^2\phi_{mc}}}\right) \tag{4.12}$$

in which c_{mc} and ϕ_{mc} are respectively the cohesion and the friction angle adopted for the Mohr-Coulomb material. In this work, when the plane strain case is required, the Drucker-Prager parameters are computed using Equations (4.11) and (4.12) for $c_{mc} = 490$ kPa and $\phi_{mc} = 20°$. Furthermore, the shape parameters in Table 1 define the deviatoric and meridional section of the Drucker-Prager, Mohr-Coulomb and CHGC surfaces, according to Lagioia and Panteghini (2016) and Lester and Sloan (2018). The selected Mohr-Coulomb shape parameters eliminate the discontinuities in deviatoric and meridional planes, providing a rounded criterion. Although this model is not exactly the classical criterion, it is treated herein as such

since the differences are insignificant. The Drucker-Prager shape parameters produce the original criterion (without the hyperbolic rounding).

Table 1 Plastic model shape parameters

Plastic model	Shape parameter			
	$\boldsymbol{\alpha}$	$\boldsymbol{\beta}$	$\boldsymbol{\gamma}$	$\boldsymbol{a}$
Drucker-Prager	1	0	1	0
Mohr-Coulomb and CHGC	$\sec\left(\frac{\pi}{6}(\bar{\gamma}+1)\right)$	0.9999	$1-\bar{\gamma}$	$0.025c\cot\phi$

The parameter $\bar{\gamma}$ is defined as

$$\bar{\gamma}=\frac{6}{\pi}\arctan\left(\frac{\sin\phi}{\sqrt{3}}\right) \tag{4.13}$$

4.2. Return mapping analysis

The performance of the proposed state-update algorithm is examined in detail for single and multisurface plasticity. As the presented procedure is strain-driven, updated stresses can be computed from known strain increments and material parameters. Additionally, it was considered $\boldsymbol{\sigma}_n=\boldsymbol{q}_n=\boldsymbol{\varepsilon}_n^p=\mathbf{0}$. Stress paths for the strain increments and stress states shown in Table 2 are analyzed in the p-q plane. These increments were selected to produce challenging return mapping paths.

Table 2 Strain increments for the return mapping analysis

Case	Stress state	Strain increments					
		$\Delta\boldsymbol{\varepsilon}_{xx}$	$\Delta\boldsymbol{\varepsilon}_{yy}$	$\Delta\boldsymbol{\varepsilon}_{zz}$	$\Delta\boldsymbol{\varepsilon}_{xy}$	$\Delta\boldsymbol{\varepsilon}_{xz}$	$\Delta\boldsymbol{\varepsilon}_{yz}$
1	Plane strain	5.00E−05	1.00E−05	0	0	-	-
2	General	−1.00E−03	2.00E−03	−1.00E−3	1.00E−03	0	1.00E−3
3	Plane strain	−6.44E−04	4.91E−04	0	−8.83E−04	-	-
4	Plane strain	−6.40E−04	4.90E−04	0	−8.80E−04	-	-

Initially, strain increment case 1 in Table 2 was analyzed for the Drucker-Prager and Mohr-Coulomb criteria, as depicted in Figure 7. The iterative method was initialized at the trial stress according to the elastic predictor and must compute

an updated state at the apex of the surface. The adaptive substepping scheme was not required.

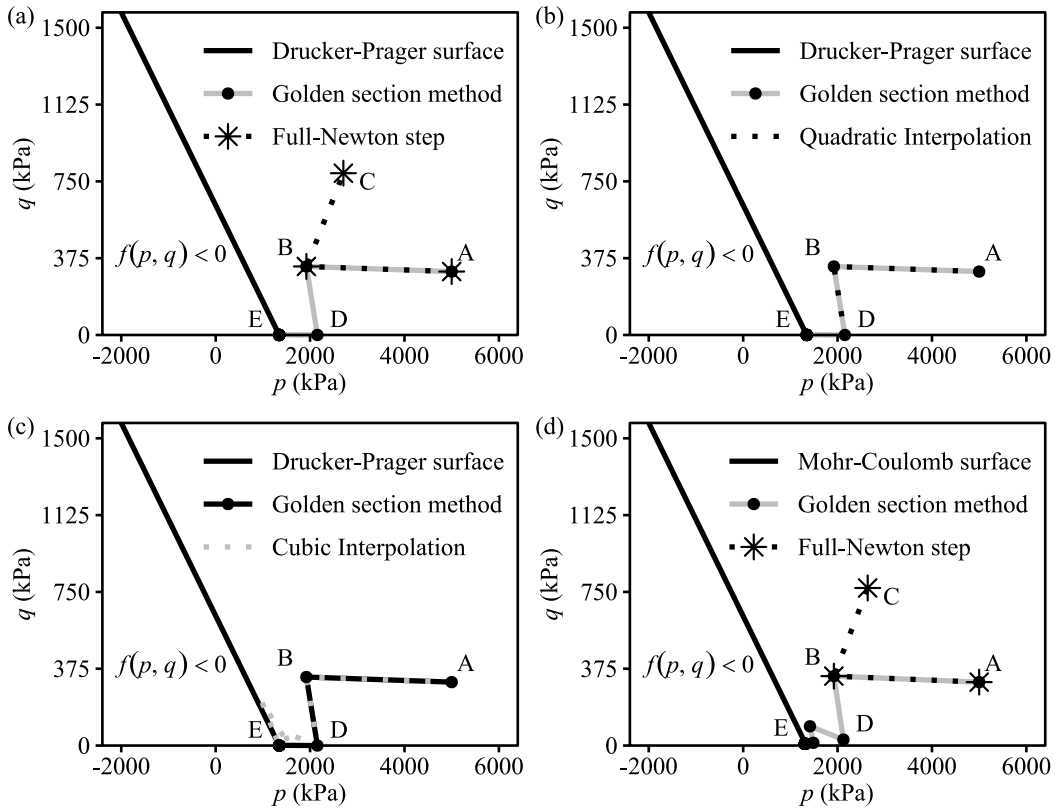

Figure 7 Return mapping paths to the apexes of the surfaces for case 1: (a) proposed algorithm versus classical return mapping for the Drucker-Prager criterion, (b) state-update procedure with quadratic interpolation for the Drucker-Prager criterion, (c) state-update procedure with cubic interpolation for the Drucker-Prager criterion and (d) proposed algorithm versus classical return mapping for the Mohr-Coulomb criterion

The full-Newton step ($\alpha_{m+1}^{(k)} = 1$) was compared with the proposed algorithm in terms of stress paths. Note that the classical return mapping algorithm could not reach the solution, as presented in Figure 7a. This algorithm started at the trial state A and oscillated between states B and C. In opposition, the proposed algorithm with the Armijo rule and golden section method was successful in returning the stress state to the surface, state E, after 12 iterations.

Besides the golden section method, two inexact line searches are assessed: quadratic and cubic interpolations. They were implemented based on Pérez-Foguet and Armero (2002) and Seifert and Schmidt (2008) and compared with the golden section method. The quadratic interpolation could not reach the solution as the algorithm stalls at stress state D in the hydrostatic axis due to minor stress updates, as shown in Figure 7b. In contrast, the cubic interpolation solved the problem for

the prescribed tolerance, as depicted in Figure 7c. However, a large number of iterations were required (90 iterations). Thus, the proposed algorithm, which considers the golden section method, is a suitable method to integrate the Drucker-Prager constitutive equations. The method emerges as an excellent alternative to the studies presented by Hjiaj, Fortin and de Saxcé (2003) and Rezaiee-Pajand and Nasirai (2008).

Figure 7d shows the return to the apex for the Mohr-Coulomb problem. The proposed method could correctly compute the updated stress E. On the other hand, the traditional closest point projection algorithm failed to solve the problem since it entered an infinite iteration cycle at points B and C.

These infinite loops also happen in compression regions of the Mohr-Coulomb surface, as depicted in Figure 8. For case 2 in Table 2, the algorithm that considers the golden section method reached the solution, and the one without line search entered an infinite loop at points B and C, as presented in the stress path in Figure 8. Lode angle for the updated stress is $\theta \approx -30°$, introducing an additional difficulty as this point is in a high curvature region. The line search strategy emerges as a suitable procedure to overcome convergence problems for surfaces with high curvature regions.

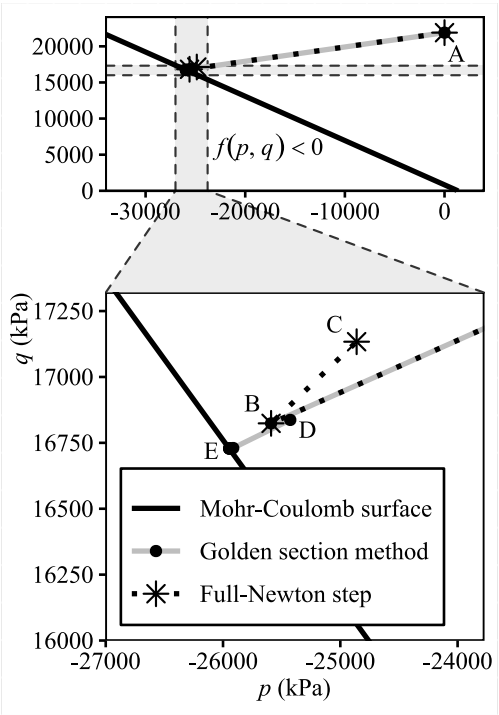

Figure 8 Return mapping path to the Mohr-Coulomb pyramid for case 2: proposed algorithm versus classical return mapping and a zoom-in at the updated stress point

Figure 9 shows the return mapping process considering the CHGC model. For this analysis, it was assumed $k_d = 5$. Moreover, the tolerance for the golden section method equals 10^{-10}. For case 3 in Table 2, the substepping technique was used since the one-step return mapping could not solve the problem. The algorithm stalled after the first update due to successive calculations of negative plastic multipliers. Then, the algorithm automatically divided the strain increment into three subincrements and successfully updated the state.

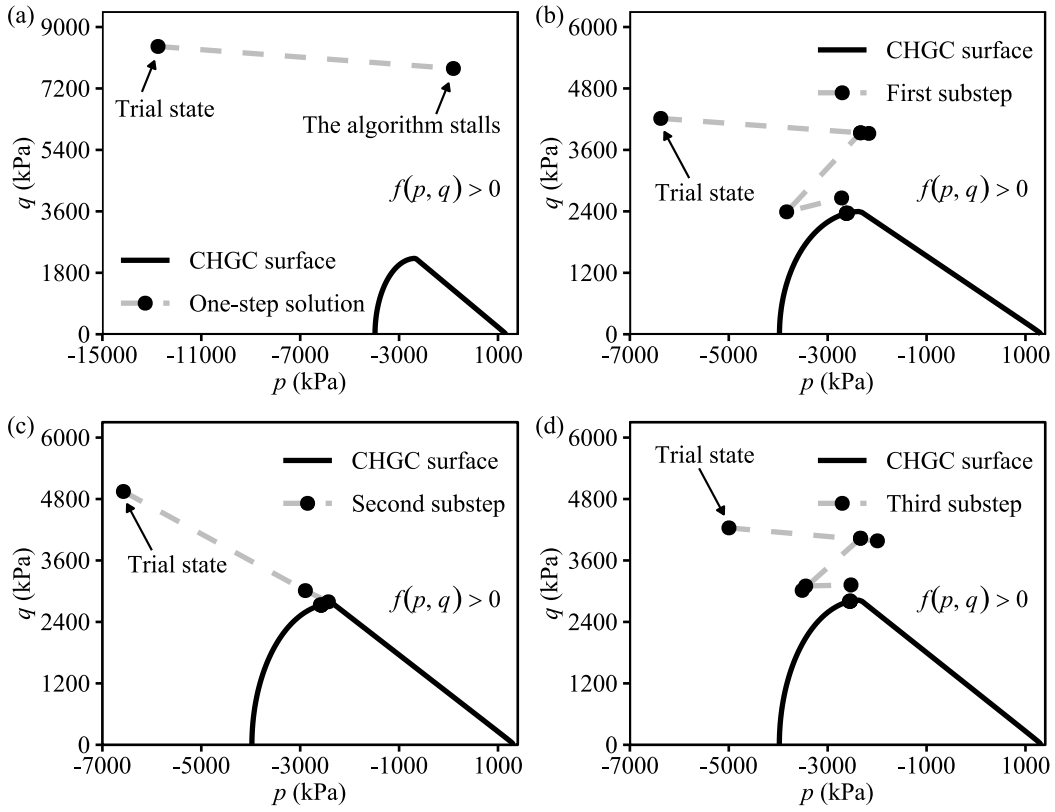

Figure 9 CHGC model return mapping paths for case 3: (a) one-step solution, (b) first substep, (c) second substep, and (d) last substep

The last case in Table 2 considers slightly different strain increments concerning case 3. This difference allowed the numerical method to solve the problem in one-step, as presented in Figure 10a. However, the algorithm was forced to divide the strain increments to compare the final solutions. A priori, this slight difference between the strain increments in cases 3 and 4 should not offer substantial modifications in the iterative process. Nevertheless, the substepping scheme is essential to solve case 3 in Figure 9. This peculiarity may be explained by the convexity properties of the CHGC model.

Franchi, Genna and Paterlini (1990) pointed out that the term convexity is frequently used vaguely in the literature. They also indicate that yield surface convexity does not entail the convexity of the yield function. The latter is essential to provide good convergence characteristics in implicit numerical integration schemes (Panteghini and Lagioia, 2018). Even though the limit of the elastic domain $\mathbb{Y}$ is a convex surface, it is crucial to conserve the convexity when the surface expands beyond stress states outside the current plastically admissible domain $\overline{\mathbb{E}}$.

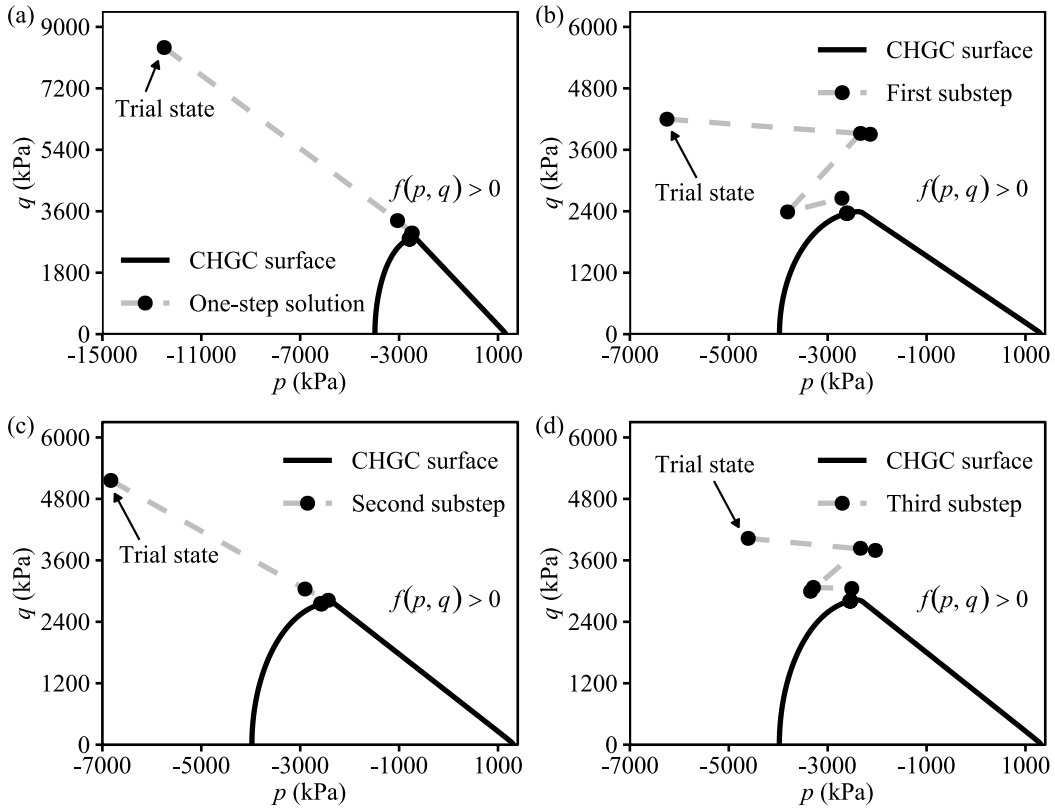

Figure 10 CHGC model return mapping paths for case 4: (a) one-step solution, (b) first substep, (c) second substep, and (d) last substep

In this context, the CHGC surface convexity was checked according to Abbo et al. (2011) and Lester and Sloan (2018). Nonetheless, the CHGC formulation does not consider any technique to provide quasi-convexity to the yield functions to preserve convexity when the surface expands outside the up-to-date $\overline{\mathbb{E}}$, as presented by Panteghini and Lagioia (2018). Therefore, this nonconvexity may explain the convergence problem in a one-step return mapping in case 3, considering the slight difference between strain increments in cases 3 and 4.

Lastly, the final solutions presented in Figure 10a and Figure 10d are compared in Table 3. Observe that the one- and multistep schemes provided different updated states. In the differential equation context, a multistep process provides better accuracy. Hence, the obtained solutions are generally distinct.

Table 3 Comparison between the updated states considering one and multistep schemes

Integration scheme	Updated state	
	p	q
One-step	−2575.94	2785.83
Multistep	−2539.50	2805.59

4.3. Finite element analysis: single-surface plasticity

The proposed state-update algorithm is now evaluated in finite element problems. The collapse of strip- and circular-footings is investigated. The limit pressures of the footings were computed using the mesh composed of 135 quadratic quadrilateral elements with four integration points described in Figure 11. Uniform displacements u were prescribed to the nodes below the footing in order to model a rigid mode settlement. Plane strain and axisymmetric conditions were used to simulate the strip- and circular-footings, respectively.

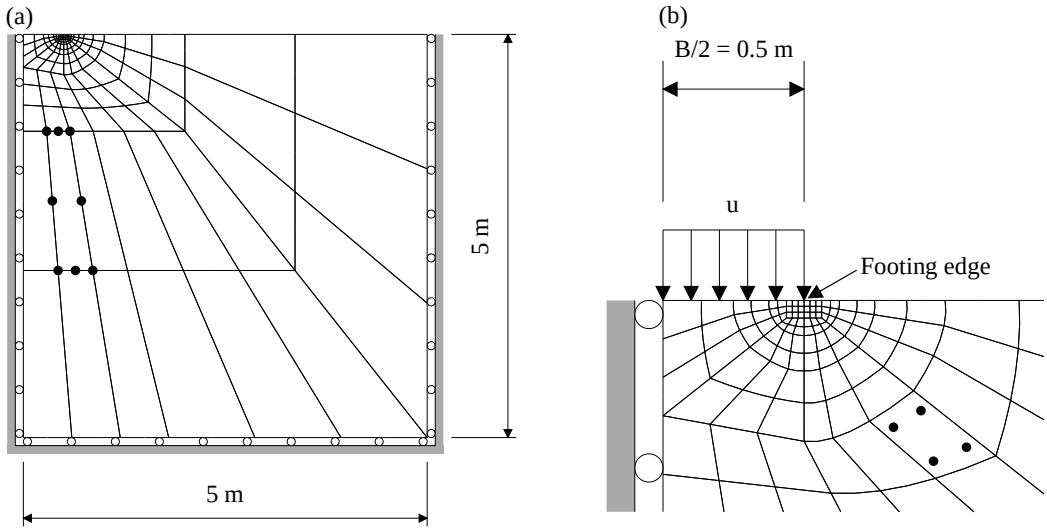

Figure 11 Rigid footing problem: (a) finite element mesh and (b) details about the mesh and the uniform prescribed displacements (adapted from de Souza Neto, Perić and Owen (2008))

The computed pressure-displacement responses are presented in Figure 12. The analytical limit pressure is taken as a reference for model accuracy. For the

strip-footing problem, the Drucker-Prager and Mohr-Coulomb criteria were adopted. The circular-footing analysis considered only the Mohr-Coulomb model, as no analytical solutions are available for the Drucker-Prager model.

In order to assess the algorithm robustness, large displacement increments are prescribed. Initially, considering the strip-footing problem for the Mohr-Coulomb criterion, the first global step is the most challenging. If only the line search strategy is adopted ($\Delta t_1 = 1$), a Gauss point close to the footing edge requires the golden section execution four times during the global iterations. On the other hand, if the substepping scheme alone is employed, the same point requires 108 substeps.

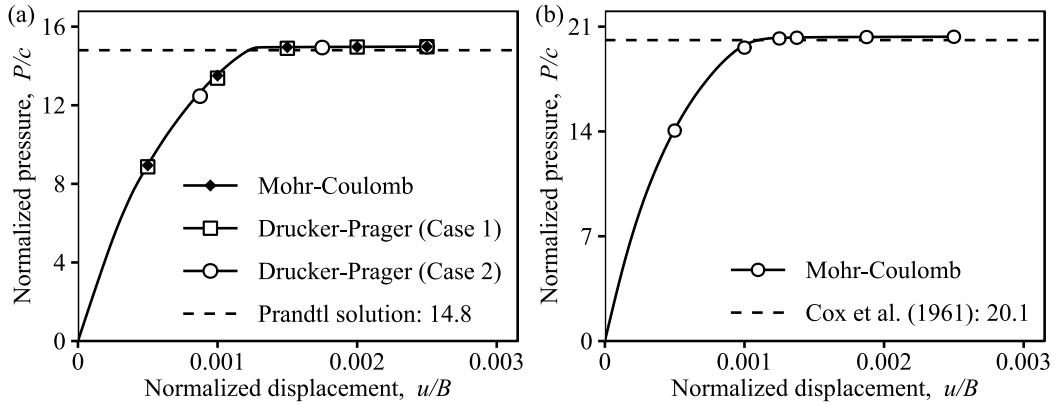

Figure 12 Normalized pressure-displacement relation considering the Mohr-Coulomb and Drucker-Prager criteria for the (a) strip- and (b) circular-footing problems

Additionally, two distinct solutions that consider the Drucker-Prager criterion were developed with different prescribed displacement increments. The first case reproduced the same displacement increments of the Mohr-Coulomb problem, as depicted in Figure 12a. This solution did not require line search and substepping strategies since all updated stresses were on the smooth portion of the Drucker-Prager cone. In the second solution, three huge global increments were adopted. In this case, the substepping scheme was not needed. Nevertheless, the line search method is required in global steps one and two, respectively, 39 and 36 times, to solve the return mapping to the apex. Importantly, the solid lines in Figure 12 represent the solution for very small steps. Observe that the proposed scheme provides accurate responses even for large increments.

Figure 13 presents the maximum and the cumulative number of golden section executions and substeps required in the circular-footing problem throughout the global iterations. In all presented results, the number of substeps does not

include the first attempt to apply the full increment ($\Delta t_1 = 1$). In addition, it presents the number of Gauss points that required these schemes in all iterations. According to Figure 13b, one Gauss point demands five substeps in the first global step; the circular-footing problem cannot be solved without the substepping process for the adopted prescribed increments. In summary, the golden section method was demanded 42746 times while the substepping scheme was needed 35 times. The high curvature regions in the adopted surface and the large displacement increments make this problem more challenging.

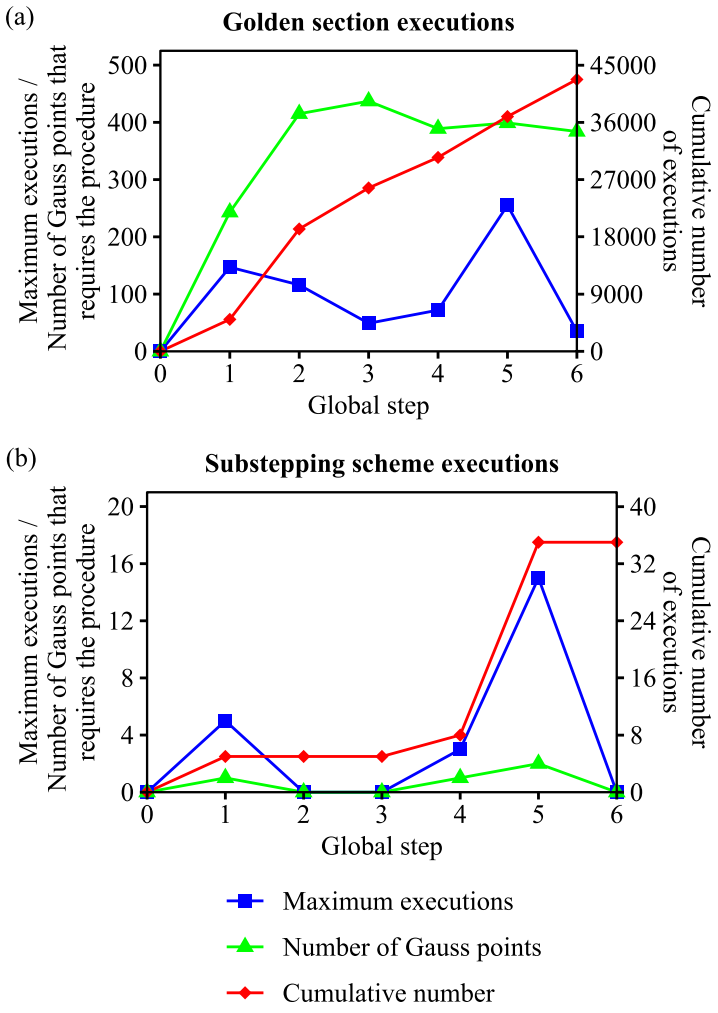

Figure 13 Maximum and cumulative number of (a) golden section executions and (b) substeps in the circular-footing analysis, as well as the number of Gauss points that required these specific strategies

4.4. Finite element analysis: multisurface plasticity

The strip- and circular-footing problems in Figure 11 were run considering the CHGC model to evaluate the performance of the finite element program. Since

the complex plastic model requires a robust state-update algorithm, this study aims to show how robustness directly affects the finite element solution. In the footing problems, the global increments were automatically calculated according to Ramm (1981). In this context, the desired number of iterations was defined as eight. If the global Newton-Raphson procedure did not reach a solution, a new attempt was run for an increment half the size.

The numerical pressure-displacement curves for the strip- and circular-footing analyses are shown in Figure 14 and Figure 15, respectively. These problems were analyzed with the state-update procedure using the line search with and without the substepping process. It should be highlighted that the algorithm with the substepping scheme required roughly three times less global steps than the one without this scheme.

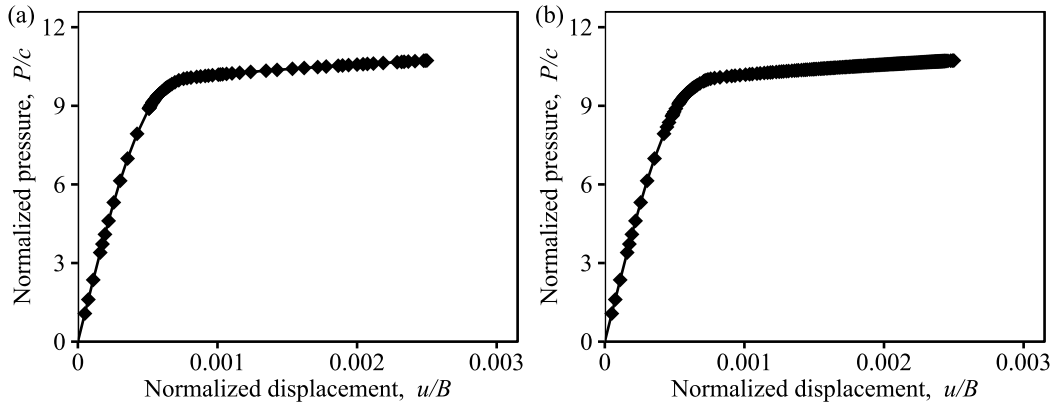

Figure 14 Normalized pressure-displacement relation for the strip-footing problem, the CHGC model, and the proposed state-update algorithm (a) with substepping and (b) without substepping

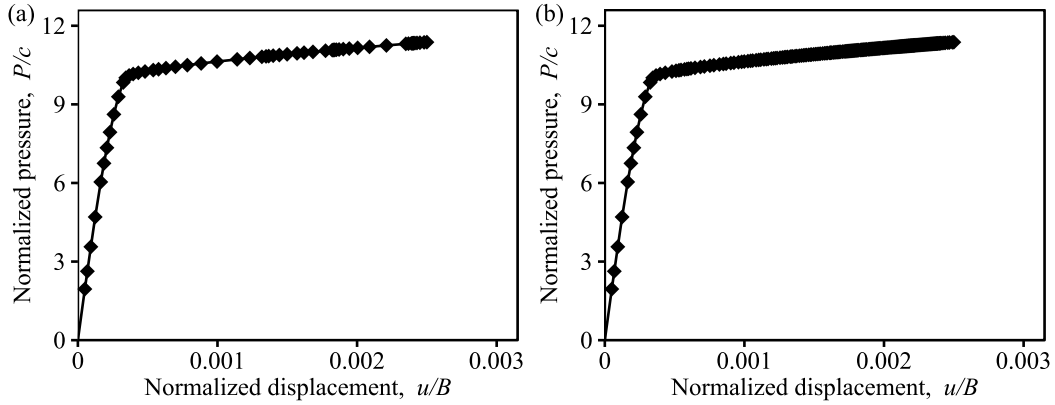

Figure 15 Normalized pressure-displacement relation for the circular-footing problem, the CHGC model, and the proposed state-update algorithm (a) with substepping and (b) without substepping

Some global residual norms computed using the substepping procedure are shown in Table 4 and Table 5. As expected, the convergence rate is quadratic since the consistent tangent modulus (3.68) is employed in the finite element equations.

Table 4 Global residual norms for the strip-footing problem and the CHGC model

Iteration	Step 61		Step 62		Step 63	
	Absolute	**Relative**	**Absolute**	**Relative**	**Absolute**	**Relative**
1	4.87E+00	5.21E−05	1.34E+00	1.44E−04	1.22E+00	1.29E−05
2	1.60E−01	1.71E−06	2.05E−01	2.21E−06	4.32E−02	4.58E−07
3	1.49E−03	1.60E−08	1.20E−03	1.29E−08	2.22E−04	2.36E−09
4	1.01E−06	1.08E−11	1.98E−08	2.13E−13	2.94E−09	3.12E−14

Table 5 Global residual norms for the circular-footing problem and the CHGC model

Iteration	Step 36		Step 37		Step 38	
	Absolute	**Relative**	**Absolute**	**Relative**	**Absolute**	**Relative**
1	4.38E+00	1.71E−05	1.61E+00	6.20E−06	2.70E−01	9.61E−07
2	9.06E−02	3.54E−07	4.91E−02	1.89E−07	3.84E+00	1.37E−05
3	2.18E−04	8.52E−10	7.29E−05	2.80E−10	5.51E−02	1.97E−07
4	-	-	-	-	3.31E−05	1.18E−10

Figure 16 and Figure 17 show the maximum and the cumulative number of golden section executions and substeps in the footing problems, as well as the number of Gauss points that needed these strategies, throughout the global Newton-Raphson iterations (including the failed global steps). Due to the plastic strain evolution directly below the footing, the substepping and line search approaches were extremely demanded. In particular, the substepping technique was mainly needed in the region near the edge of the footing. The Gauss points denoted in Figure 16c and Figure 17c mostly represent this complex region.

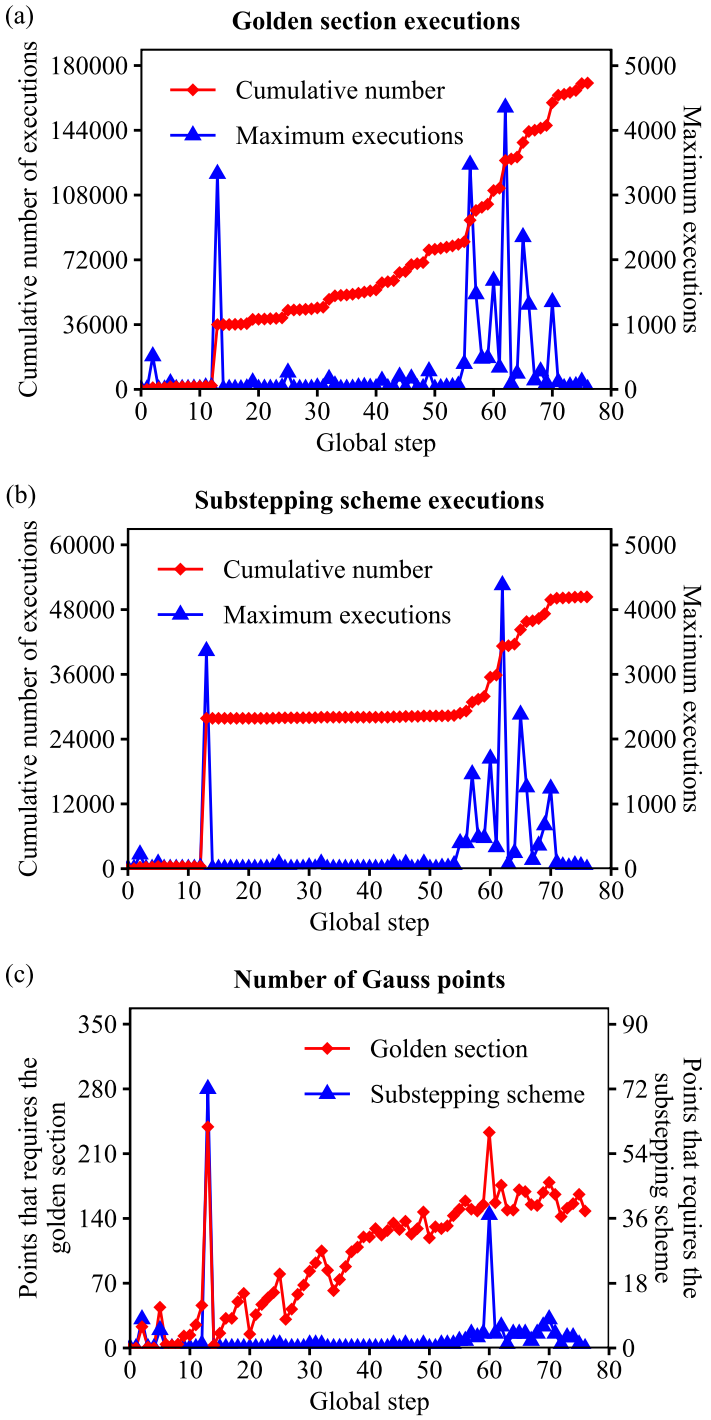

Figure 16 Evolution of (a) the golden section, (b) the substepping scheme executions, and (c) the number of Gauss points that required these strategies throughout the numerical solution for the strip-footing considering the CHGC model

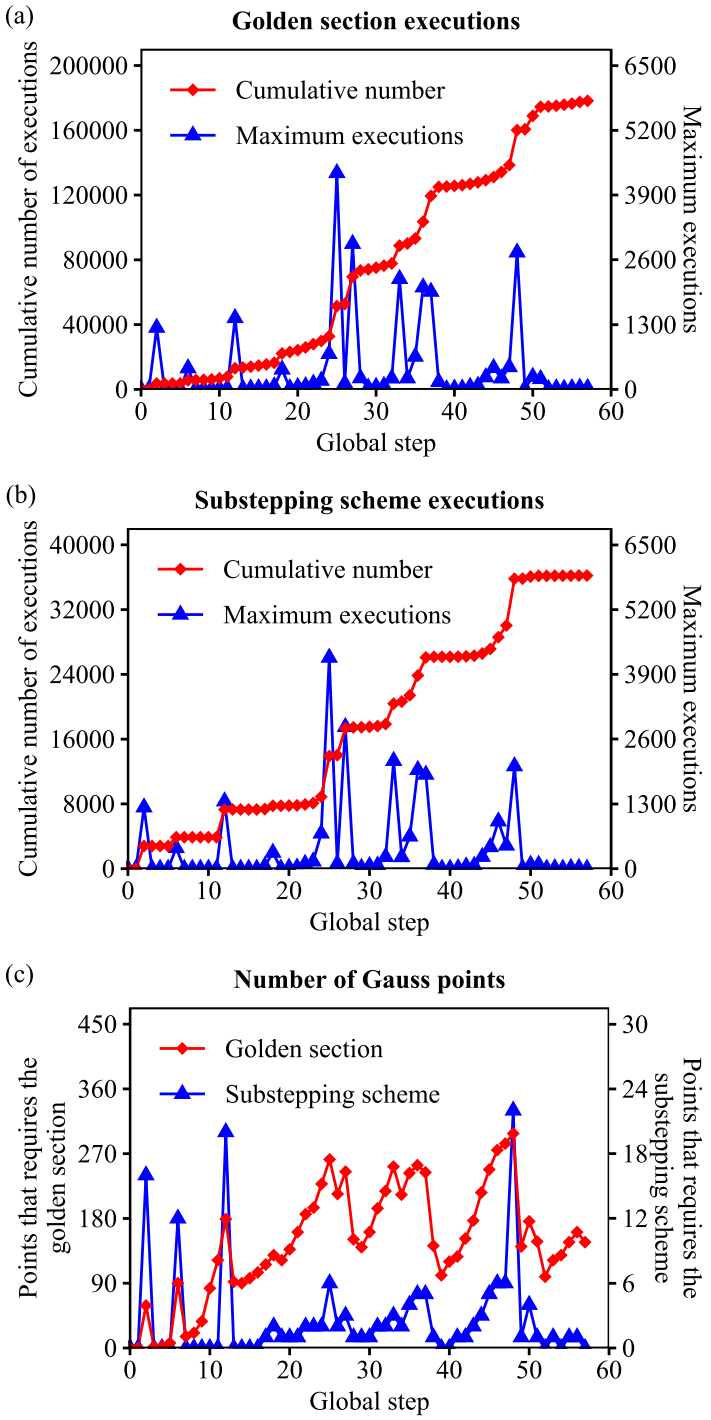

Figure 17 Evolution of (a) the golden section, (b) the substepping scheme executions, and (c) the number of Gauss points that required these strategies throughout the numerical solution for the circular-footing considering the CHGC model

The spatial distribution of substeps in the region near the footing can be observed in Figure 18. The map in this figure presents the total number of substeps required throughout the strip- and circular-footing collapse analysis. Most of the integration points do not require pseudo-time subincrements $\Delta t_m < 1$, including the regions beyond the represented map. Nonetheless, several integration points are

subjected to complex stress states. The substepping technique plays an essential role in updating the states and, consequently, reducing the number of failed global steps. It should be highlighted that one single element (located near the footing edge) requires roughly 42% and 58% of the total substeps of the strip- and circular footing simulations, respectively.

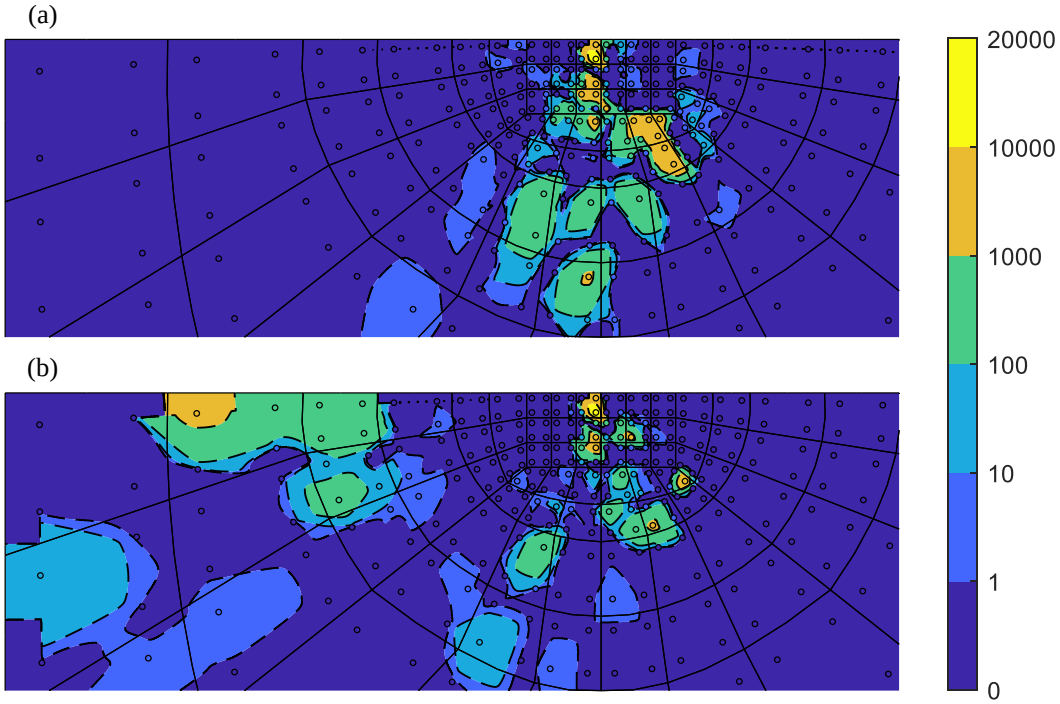

Figure 18 Spatial distribution of the total number of substeps in the (a) strip- and (b) circular-footing analysis

Some numerical experiments were developed in GeMA, regarding the quadratic and the cubic interpolations as inexact line search methods presented by Pérez-Foguet and Armero (2002), and Seifert and Schmidt (2008). In general, exact line search methods, such as the golden section method, require more computational effort. However, an insignificant difference between the computation time of inexact and exact approaches was observed in the numerical tests performed during the research.

5
Newton-Raphson-based implicit integration algorithm for both plane stress and three-dimensional stress conditions

Aiming to contribute to the computational methods for multisurface plasticity, a novel Newton-Raphson-based implicit algorithm is formulated in this chapter, presenting two key characteristics. Firstly, it considers multisurface plasticity by using complementary functions, avoiding the need for an algorithm to define active surfaces. These functions are worthy for elastoplastic applications with nonlinear elastic behavior (Zhou, Lu, Su, et al., 2022; Zhou, Lu, Zhang, et al., 2022; Lu et al., 2023) and multisurface plasticity problems (Li, Li and Zheng, 2021). Secondly, this algorithm is able to deal with the plane stress plasticity based on the computation for the plane strain condition. Hence, the proposed algorithm is able to consider distinct stress conditions.

In the literature, plane stress plasticity is typically addressed by adopting specific algorithms tailored for this state, while a separate algorithm is employed for plane strain, axisymmetric, and three-dimensional conditions. However, some researchers have dedicated efforts to provide generalized plasticity implementations. For instance, de Borst (1991) introduced an implicit algorithm at the structural level to enforce the plane stress condition in elastoplastic problems. This approach eliminates the need for specialized subroutines for plane stress plasticity, allowing the application of a standard three-dimensional algorithm. Dodds (1987) studied a von Mises model with a mixed isotropic-kinematic hardening based on the radial return mapping. At the Gauss point level, the author proposed a scheme, utilizing an outer Newton-Raphson loop, to enforce the plane stress constraint within a standard three-dimensional algorithm. In each outer loop, the radial return mapping is run, considering the updated out-of-plane total strain. Ohno, Tsuda and Kamei (2013) and Ohno et al. (2016) presented implicit integration algorithms for a von Mises-like model. The plane stress constraint was directly incorporated into the constitutive equations, enabling the definition of a unified algorithm for both plane stress and three-dimensional stress conditions,

requiring a static condensation of the consistent tangent modulus for plane stress cases. Grange (2014) developed a strategy based on multisurface plasticity to enforce the plane stress constraints. The author implemented a von Mises model with a mixed isotropic-kinematic hardening and imposed zero-stress conditions by utilizing extra yield surfaces (hyperplanes in the stress space). Santana, Keo and Hjiaj (2022) formulated an efficient and low-memory return mapping algorithm for generic single-surface models, capable of handling general stress conditions, including the plane stress condition. The author's formulation heavily relies on static condensation.

In this context, this work also aims to contribute to the study of multisurface plasticity under plane stress conditions. The literature generally lacks sufficient attention given to this particular aspect, as most authors tend to focus on models based on simple single-surfaces.

5.1.Formulation of the state-update scheme

In this state-update algorithm for multisurface plasticity, the backward Euler method is adopted to provide the incremental form of the flow rule (2.7) and the hardening/softening law (2.8). Additionally, the elastic predictor-plastic corrector scheme is also applied, assuming no update in the plastic strains as an initial hypothesis (Simo, Kennedy and Govindjee, 1988). If the Karush-Kuhn-Tucker conditions are not satisfied under this hypothesis, the plastic correction is then performed by solving the following system of nonlinear equations (Simo, Kennedy and Govindjee, 1988; Pérez-Foguet, Rodríguez-Ferran and Huerta, 2001):

$$\begin{cases} \boldsymbol{C}(\boldsymbol{\sigma}_{n+1} - \boldsymbol{\sigma}_n) + \sum_{i=1}^{N} \Delta\gamma_{i,n+1}\boldsymbol{n}_{i,n+1} = \Delta\boldsymbol{\varepsilon}_{n+1} \\ \boldsymbol{q}_{n+1} = \boldsymbol{q}_n + \sum_{i=1}^{N} \Delta\gamma_{i,n+1}\boldsymbol{h}_{i,n+1} \\ \boldsymbol{\phi}_{n+1} = \boldsymbol{0} \end{cases} \tag{5.1}$$

where $\Delta\boldsymbol{\varepsilon}_{n+1}$ is the current strain increment and $\boldsymbol{\phi}_{n+1}$ is classically defined by the elements $\phi_{i,n+1}$ as

$$\phi_{i,n+1} = f_{i,n+1}, \forall i \in \mathbb{J}_{n+1} \tag{5.2}$$

in which $\mathbb{J}_{n+1}$ represents the set of the active yield surfaces.

Several authors have studied a different approach to define $\phi_{i,n+1}$ in solving the system (5.1) (Scalet and Auricchio, 2018; Cui et al., 2021). This approach involves defining the complementary functions $\phi_{i,n+1} = 0,\ i \in \{1,2,\dots,N\}$ that enforce all Karush-Kuhn-Tucker conditions, including their inequalities. This work aims to contribute to the latter by further exploring these functions since no scheme for identifying the set of active surfaces is required, such as the one presented by Simo, Kennedy and Govindjee (1988). Therefore, all surfaces are considered during the iterative process.

5.2. Complementary function-based Newton-Raphson formulation

As presented by Scalet and Auricchio (2018), these functions can be smooth or nonsmooth. The present work focuses on studying smooth complementary functions to avoid singular points in their derivatives. Specifically, this work examines the application of smooth versions of the Fischer-Burmeister (FB) function (Fischer, 1992), the sigmoid-based function (Chen and Mangasarian, 1995), and Chen-Harker-Kanzow-Smale (CHKS) function (Chen and Mangasarian, 1996). Scalet and Auricchio (2018) referred to the sigmoid-based function as neural network function. However, this term is here considered vague as it does not accurately reflect the relationship with the sigmoid function.

The smooth FB complementary function and its derivatives are presented in Equations (5.3) to (5.5). They consider the smoothing parameter δ and the parameter c^{γ}. The former has a near-zero value and eliminates singular points in the derivatives (5.4) and (5.5), while the latter generates an approximation of the original smooth complementary function that is easier to solve. The parameter c^{γ} is frequently referred to as dimensional parameter by several authors (Areias et al., 2015; Areias, Msekh and Rabczuk, 2016; Zhou, Lu, Su, et al., 2022). In the plasticity context, Scalet and Auricchio (2018) considered the smoothing parameter in their state-update algorithms. Schmidt-Baldassari (2003) and Akpama, Bettaieb

and Abed-Meraim (2016) applied this function without dimensional and smoothing parameters, while Zhou, Lu and Zhang (2022) and Zhou, Lu and Su (2022) adopted both parameters.

$$\phi_{i,n+1} = \sqrt{\left(c^{\gamma}\Delta\gamma_{i,n+1}\right)^2 + \left(f_{i,n+1}\right)^2 + 2\delta} - c^{\gamma}\Delta\gamma_{i,n+1} + f_{i,n+1} \tag{5.3}$$

$$\partial_f \phi_{i,n+1}^{(k)} = \frac{\phi_{i,n+1}^{(k)} + c^{\gamma}\Delta\gamma_{i,n+1}^{(k)}}{\phi_{i,n+1}^{(k)} + c^{\gamma}\Delta\gamma_{i,n+1}^{(k)} - f_{i,n+1}^{(k)}} \tag{5.4}$$

$$\partial_{\Delta\gamma} \phi_{i,n+1}^{(k)} = c^{\gamma} \frac{f_{i,n+1}^{(k)} - \phi_{i,n+1}^{(k)}}{\phi_{i,n+1}^{(k)} + c^{\gamma}\Delta\gamma_{i,n+1}^{(k)} - f_{i,n+1}^{(k)}} \tag{5.5}$$

The sigmoid-based complementary function and its derivatives are presented in Equations (5.6) to (5.8). This was inspired by the sigmoid activation function commonly used in neural networks (Chen and Mangasarian, 1995; Scalet and Auricchio, 2018). Note that, unlike other functions, δ cannot be set to zero to recover a nonsmooth function. In the plasticity context, Areias et al. (2012, 2015) and Areias, Msekh and Rabczuk (2016) studied plasticity problems by considering the dimensional parameter.

$$\phi_{i,n+1} = -f_{i,n+1} - \delta \ln\left(1 + \exp\left(-\frac{c^{\gamma}\Delta\gamma_{i,n+1} + f_{i,n+1}}{\delta}\right)\right) \tag{5.6}$$

$$\partial_f \phi_{i,n+1}^{(k)} = -\frac{1}{\exp\left(-\frac{c^{\gamma}\Delta\gamma_{i,n+1} + f_{i,n+1}}{\delta}\right) + 1} \tag{5.7}$$

$$\partial_{\Delta\gamma} \phi_{i,n+1}^{(k)} = \frac{c^{\gamma} \exp\left(-\frac{c^{\gamma}\Delta\gamma_{i,n+1} + f_{i,n+1}}{\delta}\right)}{\exp\left(-\frac{c^{\gamma}\Delta\gamma_{i,n+1} + f_{i,n+1}}{\delta}\right) + 1} \tag{5.8}$$

Finally, the smooth CHKS complementary function and its derivatives are presented in Equations (5.9) to (5.11). This function is closely related to smooth versions of the max and min functions (Scalet and Auricchio, 2018). To the best of the author's knowledge, this smooth complementary function has not been considered in numerical integration schemes for plasticity applications in the literature. Therefore, this work explores the potential of using this function in such applications.

$$\phi_{i,n+1} = \frac{c^{\gamma}\Delta\gamma_{i,n+1} - f_{i,n+1}}{2} - \frac{\sqrt{\left(c^{\gamma}\Delta\gamma_{i,n+1} + f_{i,n+1}\right)^2 + 4\delta^2}}{2} \tag{5.9}$$

$$\partial_f \phi_{i,n+1}^{(k)} = \frac{c^{\gamma}\Delta\gamma_{i,n+1} - \phi_{i,n+1}}{2\phi_{i,n+1} - c^{\gamma}\Delta\gamma_{i,n+1} + f_{i,n+1}} \tag{5.10}$$

$$\partial_{\Delta\gamma} \phi_{i,n+1}^{(k)} = c^{\gamma} \frac{f_{i,n+1} + \phi_{i,n+1}}{2\phi_{i,n+1} - c^{\gamma}\Delta\gamma_{i,n+1} + f_{i,n+1}} \tag{5.11}$$

To solve the system (5.1), an unconstrained optimization process can be employed. The optimization problem is defined as

$$\min_{\boldsymbol{x}_{n+1}} \psi(\boldsymbol{x}_{n+1}) \tag{5.12}$$

in which $\psi(\boldsymbol{x}_{n+1})$ defines the objective function of the problem, which must be minimized:

$$\psi(\boldsymbol{x}_{n+1}) = \frac{1}{2}\left(\boldsymbol{r}_{n+1}(\boldsymbol{x}_{n+1})\right)^T \boldsymbol{r}_{n+1}(\boldsymbol{x}_{n+1}) \tag{5.13}$$

Considering multisurface plasticity, the vector $\boldsymbol{x}_{n+1}$ is given by

$$\boldsymbol{x}_{n+1} = \begin{bmatrix} \boldsymbol{\sigma}_{n+1} \\ \boldsymbol{q}_{n+1} \\ \Delta\boldsymbol{\gamma}_{n+1} \end{bmatrix} \tag{5.14}$$

with

$$\Delta\boldsymbol{\gamma}_{n+1} = \begin{bmatrix} \Delta\gamma_{1,n+1} \\ \Delta\gamma_{2,n+1} \\ \vdots \\ \Delta\gamma_{N,n+1} \end{bmatrix} \tag{5.15}$$

The vector function $\boldsymbol{r}_{n+1}(\boldsymbol{x}_{n+1})$ is the representation of the system (5.1) in vector form, as defined in Equation (5.16). Observe that the scaling parameters c^q and c^{ϕ} are included since the system (5.1) has multiple units of measure, with $\boldsymbol{r}_{1,n+1}$ being dimensionless, $\boldsymbol{r}_{2,n+1}$ being or not dimensionless, and $\boldsymbol{r}_{3,n+1}$ generally having unit of stress. Dennis and Schnabel (1996) demonstrated that the Newton-Raphson method is not affected by this linear transformation resulting from scaling parameters. However, these parameters may be useful in other approaches, such as

the trust-region method (Lester and Scherzinger, 2017; Zhou, Lu, Zhang, et al., 2022).

$$\boldsymbol{r}_{n+1}(\boldsymbol{x}_{n+1}) = \begin{bmatrix} \boldsymbol{r}_{1,n+1}(\boldsymbol{\sigma}_{n+1}, \boldsymbol{q}_{n+1}, \Delta\boldsymbol{\gamma}_{n+1}) \\ c^{q}\boldsymbol{r}_{2,n+1}(\boldsymbol{\sigma}_{n+1}, \boldsymbol{q}_{n+1}, \Delta\boldsymbol{\gamma}_{n+1}) \\ c^{\phi}\boldsymbol{r}_{3,n+1}(\boldsymbol{\sigma}_{n+1}, \boldsymbol{q}_{n+1}, \Delta\boldsymbol{\gamma}_{n+1}) \end{bmatrix} \tag{5.16}$$

where

$$\boldsymbol{r}_{1,n+1} = \boldsymbol{C}(\boldsymbol{\sigma}_{n+1} - \boldsymbol{\sigma}_n) - \Delta\boldsymbol{\varepsilon}_{n+1} + \sum_{i=1}^{N} \Delta\gamma_{i,n+1}\boldsymbol{n}_{i,n+1} \tag{5.17}$$

$$\boldsymbol{r}_{2,n+1} = -\boldsymbol{q}_{n+1} + \boldsymbol{q}_n + \sum_{i=1}^{N} \Delta\gamma_{i,n+1}\boldsymbol{h}_{i,n+1} \tag{5.18}$$

$$\boldsymbol{r}_{3,n+1} = \boldsymbol{\phi}_{n+1} = \begin{bmatrix} \phi_{1,n+1} \\ \phi_{2,n+1} \\ \vdots \\ \phi_{N,n+1} \end{bmatrix} \tag{5.19}$$

Unconstrained optimization typically requires defining descent directions. Assuming the objective function in (5.13), the descent direction related to the Newton-Raphson method takes the form

$$\boldsymbol{d}_{n+1}^{(k)} = \begin{bmatrix} \boldsymbol{d}_{n+1}^{\boldsymbol{\sigma}\,(k)} \\ \boldsymbol{d}_{n+1}^{\boldsymbol{q}\,(k)} \\ \boldsymbol{d}_{n+1}^{\Delta\boldsymbol{\gamma}\,(k)} \end{bmatrix} = -\left(\boldsymbol{\mathcal{J}}_{n+1}^{(k)}\right)^{-1} \boldsymbol{r}_{n+1}^{(k)} \tag{5.20}$$

with

$$\boldsymbol{d}_{n+1}^{\Delta\boldsymbol{\gamma}\,(k)} = \begin{bmatrix} d_{1,n+1}^{\Delta\boldsymbol{\gamma}\,(k)} \\ d_{2,n+1}^{\Delta\boldsymbol{\gamma}\,(k)} \\ \vdots \\ d_{N,n+1}^{\Delta\boldsymbol{\gamma}\,(k)} \end{bmatrix} \tag{5.21}$$

$$\boldsymbol{\mathcal{J}}_{n+1}^{(k)} = \begin{bmatrix} \left(\boldsymbol{\Psi}_{n+1}^{(k)}\right)^{-1} & \boldsymbol{E}_{n+1}^{(k)} \\ \left(\boldsymbol{G}_{n+1}^{(k)}\right)^{T} & \boldsymbol{\Gamma}_{n+1}^{(k)} \end{bmatrix} \tag{5.22}$$

To compute the Jacobian matrix $\boldsymbol{\mathcal{J}}_{n+1}^{(k)}$, the auxiliary expressions (5.23) to (5.26) are defined. In particular, Equation (5.23) depends on derivatives of $\boldsymbol{n}_i(\boldsymbol{\sigma}, \boldsymbol{q})$

and $\boldsymbol{h}_i(\boldsymbol{\sigma}, \boldsymbol{q})$. In the case of associated plasticity, the former is in fact the derivative of the yield function, and numerical differentiation is an attractive strategy to compute it.

$$\left(\boldsymbol{\Psi}_{n+1}^{(k)}\right)^{-1} = \begin{bmatrix} \boldsymbol{C} + \sum_{i=1}^{N} \Delta\gamma_{i,n+1}^{(k)} \partial_{\boldsymbol{\sigma}} \boldsymbol{n}_{i,n+1}^{(k)} & \sum_{i=1}^{N} \Delta\gamma_{i,n+1}^{(k)} \partial_{\boldsymbol{q}} \boldsymbol{n}_{i,n+1}^{(k)} \\ c^q \left(\sum_{i=1}^{N} \Delta\gamma_{i,n+1}^{(k)} \partial_{\boldsymbol{\sigma}} \boldsymbol{h}_{i,n+1}^{(k)} \right)^T & c^q \left(-\boldsymbol{I} + \sum_{i=1}^{N} \Delta\gamma_{i,n+1}^{(k)} \partial_{\boldsymbol{q}} \boldsymbol{h}_{i,n+1}^{(k)} \right) \end{bmatrix} \tag{5.23}$$

$$\boldsymbol{E}_{n+1}^{(k)} = \begin{bmatrix} \boldsymbol{N}_{n+1}^{(k)} \\ c^q \boldsymbol{H}_{n+1}^{(k)} \end{bmatrix} \tag{5.24}$$

$$\boldsymbol{G}_{n+1}^{(k)} = c^{\phi} \begin{bmatrix} \partial_{\boldsymbol{\sigma}} \boldsymbol{f}_{n+1}^{(k)} \\ \partial_{\boldsymbol{q}} \boldsymbol{f}_{n+1}^{(k)} \end{bmatrix} \begin{bmatrix} \partial_f \phi_{1,n+1}^{(k)} & 0 & \cdots & 0 \\ 0 & \partial_f \phi_{2,n+1}^{(k)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \partial_f \phi_{N,n+1}^{(k)} \end{bmatrix} \tag{5.25}$$

$$\boldsymbol{\Gamma}_{n+1}^{(k)} = c^{\phi} \begin{bmatrix} \partial_{\Delta\gamma} \phi_{1,n+1}^{(k)} & 0 & \cdots & 0 \\ 0 & \partial_{\Delta\gamma} \phi_{2,n+1}^{(k)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \partial_{\Delta\gamma} \phi_{N,n+1}^{(k)} \end{bmatrix} \tag{5.26}$$

where

$$\partial_{\boldsymbol{\sigma}} \boldsymbol{f}_{n+1}^{(k)} = \begin{bmatrix} \partial_{\boldsymbol{\sigma}} f_{1,n+1}^{(k)} & \partial_{\boldsymbol{\sigma}} f_{2,n+1}^{(k)} & \dots & \partial_{\boldsymbol{\sigma}} f_{N,n+1}^{(k)} \end{bmatrix} \tag{5.27}$$

$$\partial_{\boldsymbol{q}} \boldsymbol{f}_{n+1}^{(k)} = \begin{bmatrix} \partial_{\boldsymbol{q}} f_{1,n+1}^{(k)} & \partial_{\boldsymbol{q}} f_{2,n+1}^{(k)} & \dots & \partial_{\boldsymbol{q}} f_{N,n+1}^{(k)} \end{bmatrix} \tag{5.28}$$

$$\boldsymbol{N}_{n+1}^{(k)} = \begin{bmatrix} \boldsymbol{n}_{1,n+1}^{(k)} & \boldsymbol{n}_{2,n+1}^{(k)} & \dots & \boldsymbol{n}_{N,n+1}^{(k)} \end{bmatrix} \tag{5.29}$$

$$\boldsymbol{H}_{n+1}^{(k)} = \begin{bmatrix} \boldsymbol{h}_{1,n+1}^{(k)} & \boldsymbol{h}_{2,n+1}^{(k)} & \dots & \boldsymbol{h}_{N,n+1}^{(k)} \end{bmatrix} \tag{5.30}$$

To analytically solve the system (5.1), the Banachiewicz-Schur form for block matrix inversion is here applied, as defined in Equation (3.55). Therefore, the inverse of the Jacobian matrix $\boldsymbol{\mathcal{J}}_{n+1}^{(k)}$ in Equation (5.22) is given by

$$\left(\boldsymbol{\mathcal{J}}_{n+1}^{(k)}\right)^{-1}=\begin{bmatrix}\boldsymbol{\Psi}_{n+1}^{(k)}\left(\boldsymbol{I}-\boldsymbol{E}_{n+1}^{(k)}\boldsymbol{\Lambda}_{n+1}^{(k)}\left(\boldsymbol{G}_{n+1}^{(k)}\right)^{T}\boldsymbol{\Psi}_{n+1}^{(k)}\right) & \boldsymbol{\Psi}_{n+1}^{(k)}\boldsymbol{E}_{n+1}^{(k)}\boldsymbol{\Lambda}_{n+1}^{(k)}\\ \boldsymbol{\Lambda}_{n+1}^{(k)}\left(\boldsymbol{G}_{n+1}^{(k)}\right)^{T}\boldsymbol{\Psi}_{n+1}^{(k)} & -\boldsymbol{\Lambda}_{n+1}^{(k)}\end{bmatrix} \tag{5.31}$$

in which

$$\boldsymbol{\Lambda}_{n+1}^{(k)}=\left(\left(\boldsymbol{G}_{n+1}^{(k)}\right)^{T}\boldsymbol{\Psi}_{n+1}^{(k)}\boldsymbol{E}_{n+1}^{(k)}-\boldsymbol{\Gamma}_{n+1}^{(k)}\right)^{-1} \tag{5.32}$$

It is worth noting that computing the pseudo-inverse of these matrices is a viable option, particularly when they are quasi-singular (Godio et al., 2016). Considering Equations (5.16), (5.20) and (5.31), the elements of the descent direction related to stresses, internal variables and plastic multipliers are defined as follows:

$$\boldsymbol{d}_{n+1}^{\boldsymbol{\sigma},\boldsymbol{q}\,(k)}=\begin{bmatrix}\boldsymbol{d}_{n+1}^{\boldsymbol{\sigma}\,(k)}\\ \boldsymbol{d}_{n+1}^{\boldsymbol{q}\,(k)}\end{bmatrix}=-\boldsymbol{\Psi}_{n+1}^{(k)}\left(\boldsymbol{\rho}_{n+1}^{(k)}+\boldsymbol{E}_{n+1}^{(k)}\boldsymbol{d}_{n+1}^{\Delta\boldsymbol{\gamma}\,(k)}\right) \tag{5.33}$$

$$\boldsymbol{d}_{n+1}^{\Delta\boldsymbol{\gamma}\,(k)}=\boldsymbol{\Lambda}_{n+1}^{(k)}\left(c^{\phi}\boldsymbol{\phi}_{n+1}^{(k)}-\left(\boldsymbol{G}_{n+1}^{(k)}\right)^{T}\boldsymbol{\Psi}_{n+1}^{(k)}\boldsymbol{\rho}_{n+1}^{(k)}\right) \tag{5.34}$$

where

$$\boldsymbol{\rho}_{n+1}^{(k)}=\begin{bmatrix}\boldsymbol{r}_{1,n+1}^{(k)}\\ c^{q}\boldsymbol{r}_{2,n+1}^{(k)}\end{bmatrix}=\begin{bmatrix}\boldsymbol{C}\left(\boldsymbol{\sigma}_{n+1}^{(k)}-\boldsymbol{\sigma}_{n}\right)-\Delta\boldsymbol{\varepsilon}_{n+1}\\ c^{q}\left(\boldsymbol{q}_{n}-\boldsymbol{q}_{n+1}^{(k)}\right)\end{bmatrix}+\boldsymbol{E}_{n+1}^{(k)}\Delta\boldsymbol{\gamma}_{n+1}^{(k)} \tag{5.35}$$

With the descent direction computed, the next phase is to calculate the step size $\alpha_{n+1}^{(k)}$ using a line search strategy, such as the strategies discussed in Chapters 3 and 4. After determining the step size, the plastic variables can be updated using the following expressions:

$$\boldsymbol{\sigma}_{n+1}^{(k+1)}=\boldsymbol{\sigma}_{n+1}^{(k)}+\alpha_{n+1}^{(k)}\boldsymbol{d}_{n+1}^{\boldsymbol{\sigma}\,(k)} \tag{5.36}$$

$$\boldsymbol{q}_{n+1}^{(k+1)}=\boldsymbol{q}_{n+1}^{(k)}+\alpha_{n+1}^{(k)}\boldsymbol{d}_{n+1}^{\boldsymbol{q}\,(k)} \tag{5.37}$$

$$\Delta\boldsymbol{\gamma}_{n+1}^{(k+1)}=\Delta\boldsymbol{\gamma}_{n+1}^{(k)}+\alpha_{n+1}^{(k)}\boldsymbol{d}_{n+1}^{\Delta\boldsymbol{\gamma}\,(k)} \tag{5.38}$$

At the end of the iterative process, the plastic strains can be computed as follows:

$$\boldsymbol{\varepsilon}_{n+1}^{p} = \boldsymbol{\varepsilon}_{n}^{p} + \Delta\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{C}\left(\boldsymbol{\sigma}_{n+1}^{(k+1)} - \boldsymbol{\sigma}_{n}\right) \tag{5.39}$$

Notably, the consistent tangent modulus $\boldsymbol{D}_{n+1}^{ep}$ must be computed to preserve the quadratic rate of convergence of the Newton-Raphson method applied in the solution of the global equilibrium equations. In this scenario, $\boldsymbol{D}_{n+1}^{ep}$ can be computed by differentiating the system (5.1). Therefore, the consistent tangent modulus takes the form

$$\boldsymbol{D}_{n+1}^{ep} = \frac{d\boldsymbol{\sigma}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} = \boldsymbol{D}_{11,n+1} \tag{5.40}$$

in which $\boldsymbol{D}_{11,n+1}$ has dimension equal to the number of stress components and can be calculated from Equation (5.31) as follows:

$$\begin{bmatrix} \boldsymbol{D}_{11,n+1} & \boldsymbol{D}_{12,n+1} \\ \boldsymbol{D}_{21,n+1} & \boldsymbol{D}_{22,n+1} \end{bmatrix} = \boldsymbol{\Psi}_{n+1}(\boldsymbol{I} - \boldsymbol{E}_{n+1}\boldsymbol{\Lambda}_{n+1}(\boldsymbol{G}_{n+1})^{T}\boldsymbol{\Psi}_{n+1}) \tag{5.41}$$

5.3. Plane stress plasticity

The equations presented in Section 5.2 are applicable to three-dimensional problems. Moreover, plane strain and axisymmetric problems can be also analyzed using the same algorithm by considering the three normal stresses and only one shear stress. Nevertheless, to model plane stress problems, additional modifications must be made to the previous algorithm. Inspired by work of Aravas (1987), Equation (5.16) must be altered to obtain $\boldsymbol{r}_{aug,n+1}$ by adding a constraint on the out-of-plane stress $\sigma_{zz,n+1}$, which can be expressed as follows:

$$\boldsymbol{r}_{aug,n+1}\left(\boldsymbol{x}_{aug,n+1}\right) = \begin{bmatrix} \boldsymbol{r}_{1,n+1}\left(\boldsymbol{\sigma}_{n+1}, \boldsymbol{q}_{n+1}, \Delta\boldsymbol{\gamma}_{n+1}, \Delta\varepsilon_{zz,n+1}\right) \\ c^{q}\boldsymbol{r}_{2,n+1}(\boldsymbol{\sigma}_{n+1}, \boldsymbol{q}_{n+1}, \Delta\boldsymbol{\gamma}_{n+1}) \\ c^{\phi}\boldsymbol{r}_{3,n+1}(\boldsymbol{\sigma}_{n+1}, \boldsymbol{q}_{n+1}, \Delta\boldsymbol{\gamma}_{n+1}) \\ c^{\sigma}\sigma_{zz,n+1}(\boldsymbol{\sigma}_{n+1}) \end{bmatrix} \tag{5.42}$$

in which c^{σ} is a dimensional parameter and

$$\boldsymbol{x}_{aug,n+1} = \begin{bmatrix} \boldsymbol{\sigma}_{n+1} \\ \boldsymbol{q}_{n+1} \\ \Delta\boldsymbol{\gamma}_{n+1} \\ \Delta\varepsilon_{zz,n+1} \end{bmatrix} \tag{5.43}$$

Since the plane stress condition is directly enforced in the system of nonlinear equations, the presented procedure requires the use of $\boldsymbol{D}$ and $\boldsymbol{C}$ as the traditional matrices for the plane strain condition, as shown in the following equation:

$$\boldsymbol{D} = \boldsymbol{C}^{-1} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 \\ \nu & 1-\nu & \nu & 0 \\ \nu & \nu & 1-\nu & 0 \\ 0 & 0 & 0 & \dfrac{1-2\nu}{2} \end{bmatrix} \tag{5.44}$$

where E represents Young's modulus and ν represents Poisson's ratio. In this sense, $\boldsymbol{\sigma}_{n+1}$ consists of four stress components, where the out-of-plane stress will have a near-zero value upon convergence. In the context of an elastic predictor-plastic corrector algorithm, an initial guess $\Delta\varepsilon_{zz,n+1}^{(0)}$ is here required. Thus, it is assumed that there are no plastic increments, and $\Delta\varepsilon_{zz,n+1}^{(0)}$ can be computed considering linear elasticity, as given by

$$\Delta\varepsilon_{zz,n+1}^{(0)} = -\frac{\nu}{1-\nu}\left(\Delta\varepsilon_{xx,n+1} + \Delta\varepsilon_{yy,n+1}\right) \tag{5.45}$$

By utilizing Equation (5.45) to compute the stress tensor according to matrix (5.44), the resulting stresses are equivalent to those computed using matrix (5.46). Therefore, the state update scheme proposed in this work utilizes the plane strain state as basis to consider the plane stress state.

$$\boldsymbol{D} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \dfrac{1-\nu}{2} \end{bmatrix} \tag{5.46}$$

To calculate the novel Jacobian matrix $\boldsymbol{\mathcal{J}}_{aug,n+1}^{(k)}$ related to Equation (5.42), the matrix $\boldsymbol{\mathcal{J}}_{n+1}^{(k)}$ in Equation (5.22) can be used. Thus,

$$\boldsymbol{\mathcal{J}}_{aug,n+1}^{(k)} = \begin{bmatrix} \boldsymbol{\mathcal{J}}_{n+1}^{(k)} & -\boldsymbol{e}_{zz} \\ \boldsymbol{e}_{zz}^{T} & 0 \end{bmatrix} \tag{5.47}$$

with

$$\boldsymbol{e}_{zz} = [0 \quad 0 \quad 1 \quad 0 \quad 0 \quad ... \quad 0]^{T} \tag{5.48}$$

By applying the Banachiewicz-Schur form, the matrix $\boldsymbol{\mathcal{J}}_{aug,n+1}^{(k)}$ can be inverted analytically, as given by

$$\begin{aligned} &\left(\boldsymbol{\mathcal{J}}_{aug,n+1}^{(k)}\right)^{-1} \\ &= \frac{1}{\jmath_{33,n+1}^{(k)}} \begin{bmatrix} \left(\boldsymbol{\mathcal{J}}_{n+1}^{(k)}\right)^{-1}\left(\jmath_{33,n+1}^{(k)}\boldsymbol{I} - \boldsymbol{\mathcal{E}}_{zz}\left(\boldsymbol{\mathcal{J}}_{n+1}^{(k)}\right)^{-1}\right) & \boldsymbol{\jmath}_{i3,n+1}^{(k)} \\ -\boldsymbol{\jmath}_{3j,n+1}^{(k)} & 1 \end{bmatrix} \end{aligned} \tag{5.49}$$

where $\boldsymbol{\mathcal{E}}_{zz} = \boldsymbol{e}_{zz}(\boldsymbol{e}_{zz})^{T}$; the column vector $\boldsymbol{\jmath}_{i3,n+1}^{(k)}$ is the third column of $\left(\boldsymbol{\mathcal{J}}_{n+1}^{(k)}\right)^{-1}$; the row vector $\boldsymbol{\jmath}_{3j,n+1}^{(k)}$ is the third row of $\left(\boldsymbol{\mathcal{J}}_{n+1}^{(k)}\right)^{-1}$; and $\jmath_{33,n+1}^{(k)}$ is the element in the third row and third column of $\left(\boldsymbol{\mathcal{J}}_{n+1}^{(k)}\right)^{-1}$. Therefore, considering the plane stress condition, the descent direction takes the following form:

$$\begin{aligned} &\boldsymbol{d}_{aug,n+1}^{(k)} \\ &= \frac{1}{\jmath_{33,n+1}^{(k)}} \begin{bmatrix} \left(\jmath_{33,n+1}^{(k)}\boldsymbol{I} - \left(\boldsymbol{\mathcal{J}}_{n+1}^{(k)}\right)^{-1}\boldsymbol{\mathcal{E}}_{zz}\right)\boldsymbol{d}_{n+1}^{(k)} - c^{\sigma}\sigma_{zz,n+1}^{(k)}\boldsymbol{\jmath}_{i3,n+1}^{(k)} \\ \boldsymbol{\jmath}_{3j,n+1}^{(k)}\boldsymbol{r}_{n+1}^{(k)} - c^{\sigma}\sigma_{zz,n+1}^{(k)} \end{bmatrix} \end{aligned} \tag{5.50}$$

Since $\boldsymbol{\mathcal{E}}_{zz}$ has only one nonzero element (the third element in the main diagonal), the product $\left(\boldsymbol{\mathcal{J}}_{n+1}^{(k)}\right)^{-1}\boldsymbol{\mathcal{E}}_{zz}$ yields a matrix with only one nonzero column: the third column, which is identical to the third column of $\left(\boldsymbol{\mathcal{J}}_{n+1}^{(k)}\right)^{-1}$. To compute this product, according to Equation (5.31), one can simply select the pertinent elements from the products $\boldsymbol{\Psi}_{n+1}^{(k)}\left(\boldsymbol{I} - \boldsymbol{E}_{n+1}^{(k)}\boldsymbol{\Lambda}_{n+1}^{(k)}\left(\boldsymbol{G}_{n+1}^{(k)}\right)^{T}\boldsymbol{\Psi}_{n+1}^{(k)}\right)$ and $\boldsymbol{\Lambda}_{n+1}^{(k)}\left(\boldsymbol{G}_{n+1}^{(k)}\right)^{T}\boldsymbol{\Psi}_{n+1}^{(k)}$. An analogous process can be executed to define $\boldsymbol{\jmath}_{i3,n+1}^{(k)}$ and $\boldsymbol{\jmath}_{3j,n+1}^{(k)}$. In this scenario,

$$\boldsymbol{j}_{i3,n+1}^{(k)} = \begin{bmatrix} \boldsymbol{\psi}_{i3,n+1}^{(k)} - \boldsymbol{\Psi}_{n+1}^{(k)} \boldsymbol{E}_{n+1}^{(k)} \boldsymbol{\Lambda}_{n+1}^{(k)} \left(\boldsymbol{G}_{n+1}^{(k)} \right)^T \boldsymbol{\psi}_{i3,n+1}^{(k)} \\ \boldsymbol{\Lambda}_{n+1}^{(k)} \left(\boldsymbol{G}_{n+1}^{(k)} \right)^T \boldsymbol{\psi}_{i3,n+1}^{(k)} \end{bmatrix} \tag{5.51}$$

$$\boldsymbol{j}_{3j,n+1}^{(k)} = \begin{bmatrix} \boldsymbol{\psi}_{3j,n+1}^{(k)} - \boldsymbol{\psi}_{3j,n+1}^{(k)} \boldsymbol{E}_{n+1}^{(k)} \boldsymbol{\Lambda}_{n+1}^{(k)} \left(\boldsymbol{G}_{n+1}^{(k)} \right)^T \boldsymbol{\Psi}_{n+1}^{(k)} \\ \boldsymbol{\psi}_{3j,n+1}^{(k)} \boldsymbol{E}_{n+1}^{(k)} \boldsymbol{\Lambda}_{n+1}^{(k)} \end{bmatrix}^T \tag{5.52}$$

in which $\boldsymbol{\psi}_{i3,n+1}^{(k)}$ and $\boldsymbol{\psi}_{3j,n+1}^{(k)}$ are respectively the third column and third row of $\boldsymbol{\Psi}_{n+1}^{(k)}$. Hence, $\boldsymbol{d}_{aug,n+1}^{(k)}$ can be simplified by Equation (5.53). Importantly, $d_{zz,n+1}^{\boldsymbol{\sigma},\boldsymbol{q}\,(k)}$ is the third element of $\boldsymbol{d}_{n+1}^{\boldsymbol{\sigma},\boldsymbol{q}\,(k)}$ and $d_{n+1}^{\Delta\varepsilon_{zz}\,(k)}$ is the last element of $\boldsymbol{d}_{aug,n+1}^{(k)}$.

$$\boldsymbol{d}_{aug,n+1}^{(k)} = \begin{bmatrix} \begin{bmatrix} \boldsymbol{d}_{n+1}^{\boldsymbol{\sigma},\boldsymbol{q}\,(k)} \\ \boldsymbol{d}_{n+1}^{\Delta\boldsymbol{\gamma}\,(k)} \end{bmatrix} - \dfrac{c^{\sigma}\sigma_{zz,n+1}^{(k)} + d_{zz,n+1}^{\boldsymbol{\sigma},\boldsymbol{q}\,(k)}}{j_{33,n+1}^{(k)}} \boldsymbol{j}_{i3,n+1}^{(k)} \\ \dfrac{1}{j_{33,n+1}^{(k)}} \left(\boldsymbol{j}_{3j,n+1}^{(k)} \boldsymbol{r}_{n+1}^{(k)} - c^{\sigma}\sigma_{zz,n+1}^{(k)} \right) \end{bmatrix} \tag{5.53}$$

The consistent tangent modulus for the plane stress condition is also given by Equation (5.40), in which, according to Equation (5.49), $\boldsymbol{D}_{11,n+1}$ is computed by

$$\begin{bmatrix} \boldsymbol{D}_{11,n+1} & \boldsymbol{D}_{12,n+1} & \boldsymbol{D}_{13,n+1} \\ \boldsymbol{D}_{21,n+1} & \boldsymbol{D}_{22,n+1} & \boldsymbol{D}_{23,n+1} \\ \boldsymbol{D}_{31,n+1} & \boldsymbol{D}_{32,n+1} & \boldsymbol{D}_{33,n+1} \end{bmatrix} = \left(\boldsymbol{\mathcal{J}}_{n+1}^{(k)} \right)^{-1} \left(\boldsymbol{I} - \frac{1}{j_{33,n+1}^{(k)}} \boldsymbol{\mathcal{E}}_{zz} \left(\boldsymbol{\mathcal{J}}_{n+1}^{(k)} \right)^{-1} \right) \tag{5.54}$$

This expression can be simplified using the following:

$$\begin{bmatrix} \boldsymbol{D}_{11,n+1} & \boldsymbol{D}_{12,n+1} & \boldsymbol{D}_{13,n+1} \\ \boldsymbol{D}_{21,n+1} & \boldsymbol{D}_{22,n+1} & \boldsymbol{D}_{23,n+1} \\ \boldsymbol{D}_{31,n+1} & \boldsymbol{D}_{32,n+1} & \boldsymbol{D}_{33,n+1} \end{bmatrix} = \left(\boldsymbol{\mathcal{J}}_{n+1}^{(k)} \right)^{-1} - \frac{1}{j_{33,n+1}^{(k)}} \boldsymbol{j}_{i3,n+1}^{(k)} \boldsymbol{j}_{3j,n+1}^{(k)} \tag{5.55}$$

The Newton-Raphson-based scheme presented in this work is summarized in Algorithm 5, which takes into account different stress conditions, including the plane stress condition.

Algorithm 5 Newton-Raphson-based implicit integration scheme for multisurface plasticity considering both plane stress and plane strain conditions

Initialize: $k = 0$, k_{max}, $\Delta\boldsymbol{\varepsilon}_{n+1}$, $\boldsymbol{\varepsilon}_n^p$, $\boldsymbol{\sigma}_n$, $\boldsymbol{q}_n$, $\boldsymbol{D}$, $\boldsymbol{C}$, $f_i(\boldsymbol{\sigma}, \boldsymbol{q})$, $\boldsymbol{n}_i(\boldsymbol{\sigma}, \boldsymbol{q})$, $\boldsymbol{h}_i(\boldsymbol{\sigma}, \boldsymbol{q})$, N, tol_1, tol_2, tol_3, δ, c^γ, c^ϕ, and c^σ. In case of plane stress, initialize ν and tol_4, and compute $\boldsymbol{D}$, $\boldsymbol{C}$ according to the plane strain state

Procedure:

1: Elastic predictor:

$\boldsymbol{\sigma}_{n+1}^{(0)} = \boldsymbol{\sigma}_n + \boldsymbol{D}\Delta\boldsymbol{\varepsilon}_{n+1}$, $\boldsymbol{q}_{n+1}^{(0)} = \boldsymbol{q}_n$, $\Delta\boldsymbol{\gamma}_{n+1}^{(0)} = \boldsymbol{0}$, and $\boldsymbol{\varepsilon}_{n+1}^{p\,(0)} = \boldsymbol{\varepsilon}_n^p$

In case of plane stress, $\Delta\varepsilon_{zz,n+1}^{(0)} = -\frac{\nu}{1-\nu}\left(\Delta\varepsilon_{xx,n+1} + \Delta\varepsilon_{yy,n+1}\right)$

2: Check plastic condition:

Compute $f_{1,n+1}^{(0)}, f_{2,n+1}^{(0)}, \cdots, f_{N,n+1}^{(0)}$ according to the constitutive model

If $\max\left(f_{1,n+1}^{(0)}, f_{2,n+1}^{(0)}, \cdots, f_{N,n+1}^{(0)}\right) \leq tol_1$:

In case of plane stress, redefine $\boldsymbol{D}$ as the traditional 3×3 matrix in Equation (5.46)

Return $\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_{n+1}^{(0)}$, $\boldsymbol{q}_{n+1} = \boldsymbol{q}_{n+1}^{(0)}$, $\Delta\boldsymbol{\gamma}_{n+1} = \Delta\boldsymbol{\gamma}_{n+1}^{(0)}$, $\boldsymbol{\varepsilon}_{n+1}^p = \boldsymbol{\varepsilon}_{n+1}^{p\,(0)}$, and $\boldsymbol{D}_{n+1}^{ep} = \boldsymbol{D}$

End

3: Compute $\boldsymbol{N}_{n+1}^{(k)}$, $\boldsymbol{H}_{n+1}^{(k)}$, and $\boldsymbol{E}_{n+1}^{(k)}$ using Equations (5.29), (5.30), and (5.24), respectively

4: Compute $\partial_{\boldsymbol{\sigma}}\boldsymbol{f}_{n+1}^{(k)}$, $\partial_{\boldsymbol{q}}\boldsymbol{f}_{n+1}^{(k)}$, and $\boldsymbol{G}_{n+1}^{(k)}$ using Equations (5.27), (5.28), and (5.25), respectively. As $\boldsymbol{G}_{n+1}^{(k)}$ depends on $\partial_f\phi_{i,n+1}^{(k)}$, use Equations (5.4), (5.7), or (5.10)

5: Compute $\boldsymbol{\Psi}_{n+1}^{(k)}$, $\boldsymbol{\Gamma}_{n+1}^{(k)}$, and $\boldsymbol{\Lambda}_{n+1}^{(k)}$ using Equations (5.23), (5.26), and (5.32), respectively. As $\boldsymbol{\Gamma}_{n+1}^{(k)}$ depends on $\partial_{\Delta\gamma}\phi_{i,n+1}^{(k)}$, use Equations (5.5), (5.8), or (5.11)

6: Check the stopping criteria:

Compute $\boldsymbol{\phi}_{n+1}^{(k)}$ and $\boldsymbol{\rho}_{n+1}^{(k)}$ using Equations (5.19) and (5.35), respectively

If $\left\|\boldsymbol{\phi}_{n+1}^{(k)}\right\| \leq c^\phi tol_1$, $\left\|\boldsymbol{r}_{1,n+1}^{(k)}\right\| \leq tol_2$, and $\left\|\boldsymbol{r}_{2,n+1}^{(k)}\right\| \leq c^q tol_3$:

(In case of plane stress, also include $\left|\sigma_{zz,n+1}^{(k)}\right| \leq c^\sigma tol_4$)

Compute $\boldsymbol{D}_{n+1}^{ep}$ based on (5.40) and (5.41). In case of plane stress, use (5.55) instead of (5.41) and remove the third (zero) row and third (zero) column of $\boldsymbol{D}_{n+1}^{ep}$

Compute the plastic strains using Equation (5.39)

Return $\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_{n+1}^{(k)}$, $\boldsymbol{q}_{n+1} = \boldsymbol{q}_{n+1}^{(k)}$, $\Delta\boldsymbol{\gamma}_{n+1} = \Delta\boldsymbol{\gamma}_{n+1}^{(k)}$, $\boldsymbol{\varepsilon}_{n+1}^p = \boldsymbol{\varepsilon}_{n+1}^{p\,(k)}$, and $\boldsymbol{D}_{n+1}^{ep}$

End

If $k \leq k_{max}$:

Stop the algorithm since it fails to converge

End

7: Compute $\boldsymbol{d}_{n+1}^{\Delta\boldsymbol{\gamma}\,(k)}$ and $\boldsymbol{d}_{n+1}^{\boldsymbol{\sigma},\boldsymbol{q}\,(k)}$ using Equations (5.34) and (5.33), respectively

In case of plane stress, correct the directions using the following expressions based on (5.53):

$$\begin{bmatrix} \boldsymbol{d}_{n+1}^{\boldsymbol{\sigma},\boldsymbol{q}\,(k)} \\ \boldsymbol{d}_{n+1}^{\Delta\boldsymbol{\gamma}\,(k)} \end{bmatrix} \coloneqq \begin{bmatrix} \boldsymbol{d}_{n+1}^{\boldsymbol{\sigma},\boldsymbol{q}\,(k)} \\ \boldsymbol{d}_{n+1}^{\Delta\boldsymbol{\gamma}\,(k)} \end{bmatrix} - \frac{c^\sigma\sigma_{zz,n+1}^{(k)} + d_{zz,n+1}^{\boldsymbol{\sigma},\boldsymbol{q}\,(k)}}{j_{33,n+1}^{(k)}}\boldsymbol{j}_{i3,n+1}^{(k)}$$

$$d_{n+1}^{\Delta\varepsilon_{zz}\,(k)} \coloneqq \frac{1}{j_{33,n+1}^{(k)}}\left(\boldsymbol{j}_{3j,n+1}^{(k)}\boldsymbol{r}_{n+1}^{(k)} - c^\sigma\sigma_{zz,n+1}^{(k)}\right)$$

where $\left(\boldsymbol{J}_{n+1}^{(k)}\right)^{-1}$, $\boldsymbol{j}_{i3,n+1}^{(k)}$, $\boldsymbol{j}_{3j,n+1}^{(k)}$, and $j_{33,n+1}^{(k)}$ are the inverse of the Jacobian matrix and its parts and $\boldsymbol{r}_{n+1}^{(k)}$ is defined by Equation (5.16)

8: Perform the line search to compute $\alpha_{n+1}^{(k)}$

9: Update stresses, internal variables and plastic multiplier, using Equations (5.36) to (5.38). In case of plane stress, $\Delta\varepsilon_{zz,n+1}^{(k+1)} = \Delta\varepsilon_{zz,n+1}^{(k)} + \alpha_{n+1}^{(k)}d_{n+1}^{\Delta\varepsilon_{zz}\,(k)}$

10: Update counter variable:

$k \coloneqq k + 1$

Go to step 3)

6
Newton-Krylov-based implicit integration algorithm

The implicit substepping integration scheme proposed in this work demonstrates remarkable robustness, as it effectively handles severe hardening processes and singular points on yield surfaces. Nevertheless, it involves various components, such as a line search strategy and an adaptive substepping scheme. In addition, the algorithm requires derivatives of $f_{i,n+1}$, $\boldsymbol{n}_{i,n+1}$, and $\boldsymbol{h}_{i,n+1}$., which can pose challenges when dealing with complex constitutive formulations. Consequently, implementing multisurface plasticity models in a numerical simulator can be a cumbersome task. Numerical derivatives based on finite difference schemes offer a potential solution for computing these derivatives. In this context, this chapter aims to develop a Jacobian-free implicit integration algorithm that circumvents the need for some derivative computations.

In the scenario of implicit integration schemes for single and multisurface plasticity, the classical Newton-Raphson method is extensively applied to solve systems of nonlinear equations (Adhikary et al., 2017; Scalet and Auricchio, 2018; Pech, Lukacevic and Füssl, 2021). Nevertheless, other numerical methods for solving such systems are also applied in the literature: Broyden's method (Feenstra and de Borst, 1996), Levenberg-Marquardt method (Shterenlikht and Alexander, 2012), third-order Newton-like methods (Kiran, Li and Khandelwal, 2015), trust-region method (Shterenlikht and Alexander, 2012; Lester and Scherzinger, 2017; Zhou, Lu, Zhang, et al., 2022), and a hybrid Newton/Polak–Ribière conjugate gradient algorithm (Geng et al., 2021). Additionally, some solutions based on mathematical optimization are available in the literature: interior-point method (Krabbenhøft et al., 2007; Krabbenhøft, Lyamin and Sloan, 2007, 2008; Krabbenhøft and Lyamin, 2012; Bruno et al., 2020) and complementary theory (Zheng, Zhang and Wang, 2020; Zhao et al., 2022; Zheng and Chen, 2022).

The numerical solution of systems of nonlinear equations is a crucial and challenging research topic that has garnered significant attention in recent years.

Robust and efficient methods, often of an iterative nature, have been developed to tackle this problem. These methods are usually variants of well-known techniques, for instance, higher-order Newton-like methods (Grau-Sánchez, Grau and Noguera, 2011; Cordero et al., 2012; Sharma, Guha and Sharma, 2013; Sharma and Gupta, 2014; Esmaeili and Ahmadi, 2015; Sharma and Arora, 2017); Levenberg-Marquardt-like methods (Fan, 2012; Zhou, 2013; Chen, Du and Ma, 2017); Broyden-like methods (Ziani and Guyomarc'h, 2008; Fang, Ni and Zeng, 2018); and conjugate gradient-like methods (Zhang, Zhou and Li, 2006; Cheng, Xiao and Hu, 2009; Yuan and Zhang, 2015; Abubakar and Kumam, 2019; Zhang, 2020). Regarding this variety of methods, the author considers that investigating other analogous numerical algorithms can bring new alternatives for the state update procedure in elastoplasticity.

Newton-Krylov methods (Dembo, Eisenstat and Steihaug, 1982) are ingenious algorithms that combine Newton-type methods for linearizing systems of nonlinear equations and Krylov subspace methods to solve the system of linear equations at each nonlinear step. In this approach, the linearized system, defined by a Jacobian matrix, is solved using a Krylov subspace-based iterative method. These iterative schemes are commonly used to compute an inexact solution at each Newton step, reducing the computational cost of the Newton-Raphson method (Eisenstat and Walker, 1994). Newton-Krylov methods are widely employed in numerical simulations to solve large-scale systems of nonlinear equations, being also applied in the finite element context. These methods have been successfully employed in problems such as nonlinear elasticity (Barnafi, Pavarino and Scacchi, 2022), two-phase flow in porous media (Bergamaschi et al., 2012), and incompressible fluid flow (Bellavia and Berrone, 2007). Newton-Krylov methods have also been used in various other research areas, as reported by Knoll and Keyes (2004).

It is worth noting that in Krylov subspace methods, the operations associated with the Jacobian matrix involve exclusively matrix-vector products. These operations can be carried out using finite difference schemes, which do not require any information about the derivatives of the system of nonlinear equations (An, Wen and Feng, 2011; Kan et al., 2022). Consequently, it is possible to perform Newton steps without explicitly forming the elements of the Jacobian matrix. The

use of Jacobian-free Newton-Krylov algorithms has gained popularity due to their versatility and efficiency. These algorithms have been successfully applied to solve the global system of nonlinear equations in various finite element applications, including linear hardening plasticity (Hales et al., 2012), crystal plasticity (Chockalingam et al., 2013), geometric nonlinearity problems (Kan et al., 2022), multiscale modeling (Rahul and De, 2011, 2015b, 2015a), thermomechanical coupling problems (Hales et al., 2012), and simulation of nuclear processes (Gaston et al., 2009).

As mentioned earlier, one of the focuses of this work is to develop a Newton-Krylov-based algorithm that combines flexibility with competitive performance compared to other finite difference-based Newton-Raphson implementations. In opposition to the intricate implementation presented in Chapter 3, the algorithm proposed in this chapter does not require some derivative computations, simplifying the implementation of new multisurface plasticity models in a numerical simulator. From a practical perspective, the constitutive model information required to implement a forward Euler-based algorithm, a cutting plane algorithm, and a Newton-Krylov-based implicit algorithm is essentially the same. However, the last is unconditionally stable, making it a more robust and reliable option for numerical simulations. Therefore, this chapter explores the fundamentals of applying a Jacobian-free Newton-Krylov method to solve systems of nonlinear equations. Based on this review, it is defined a Newton-Krylov-based implicit integration algorithm, incorporating the features discussed in the previous chapter.

It is worth highlighting that the consistent tangent modulus depends on derivatives of $f_{i,n+1}$, $\boldsymbol{n}_{i,n+1}$, and $\boldsymbol{h}_{i,n+1}$, as presented in Chapter 3 and 5. Therefore, numerical differentiation is adopted to compute this modulus. Finite difference schemes are largely applied to calculate consistent tangent moduli and tangent stiffness matrices (Pérez-Foguet, Rodríguez-Ferran and Huerta, 2000a; Kim, Ryu and Cho, 2011; Kiran and Khandelwal, 2014; Tanaka et al., 2014; Scheunemann, Nigro and Schröder, 2021). In this scenario, a scheme previously applied by Seifert and Schmidt (2008) and Lu et al. (2023) is implemented in this work.

6.1. Jacobian-free Newton-Krylov method

6.1.1. Inexact Newton methods

Consider a generic system of nonlinear equations

$$\boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{0} \tag{6.1}$$

in which $\boldsymbol{x} \in \mathbb{R}^n$, $n \in \mathbb{Z}$, and $\boldsymbol{r} : \mathbb{R}^n \to \mathbb{R}^n$ is a vector function. Typically, the classical Newton-Raphson method is applied to solve this sort of system. The method is derived from a multivariate Taylor expansion of the function $\boldsymbol{r}(\boldsymbol{x})$ around the current point $\boldsymbol{x}^{(k)}$, that is

$$\boldsymbol{r}\left(\boldsymbol{x}^{(k+1)}\right) = \boldsymbol{r}\left(\boldsymbol{x}^{(k)}\right) + \frac{\partial \boldsymbol{r}\left(\boldsymbol{x}^{(k)}\right)}{\partial \boldsymbol{x}}\left(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\right) + \cdots \tag{6.2}$$

Neglecting the higher-order terms and making the right-hand side zero, the Newton-Raphson formula is obtained:

$$\boldsymbol{\mathcal{J}}^{(k)}\boldsymbol{d}^{(k)} = -\boldsymbol{r}\left(\boldsymbol{x}^{(k)}\right) \tag{6.3}$$

in which k defines the current iteration and

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{d}^{(k)} \tag{6.4}$$

$$\boldsymbol{\mathcal{J}}^{(k)} = \boldsymbol{\mathcal{J}}\left(\boldsymbol{x}^{(k)}\right) = \frac{\partial \boldsymbol{r}\left(\boldsymbol{x}^{(k)}\right)}{\partial \boldsymbol{x}} \tag{6.5}$$

It is important to highlight that the Newton-Raphson method is attractive because of its convergence characteristics, but it requires the exact Jacobian matrix in (6.5) to perform the calculations. Integrating the elastoplastic constitutive equations usually yields a computation of complex Jacobian matrices, a situation overcome by some authors through numerical differentiation (Pérez-Foguet, Rodríguez-Ferran and Huerta, 2000b; Choi and Yoon, 2019). In this scenario, inexact Newton methods (Dembo, Eisenstat and Steihaug, 1982) can be an attractive strategy to solve Equation (6.1).

In an inexact Newton method, iterative methods are commonly used to solve Equation (6.3) (Nocedal and Wright, 2006; Bergamaschi, Bru and Martínez, 2011; Bergamaschi et al., 2012; Barnafi, Pavarino and Scacchi, 2022). In this sense, Newton-Krylov methods are developed based on Krylov subspace iterative methods. The tolerance parameter of those iterative methods can be adjusted in each Newton step, producing inexact solutions of (6.3) that can accelerate the overall solution process (Eisenstat and Walker, 1996; Pernice and Walker, 1998). When an intermediate iteration value $\boldsymbol{x}^{(k)}$ is significantly far from the final solution of (6.1), the linearization process in Equation (6.2) may inaccurately approximate $\boldsymbol{r}(\boldsymbol{x})$ (Dembo, Eisenstat and Steihaug, 1982; Kan et al., 2022). Therefore, a tight tolerance is not always necessary, especially in the early iterations. Additionally, inexact Newton methods are valuable for solving large-scale problems, as calculating the exact solution can be computationally expensive (Dembo, Eisenstat and Steihaug, 1982; Eisenstat and Walker, 1994). In this scenario, an inexact Newton step must comply with the following criterion (Eisenstat and Walker, 1994):

$$\left\|\boldsymbol{r}\left(\boldsymbol{x}^{(k)}\right)+\boldsymbol{J}^{(k)}\boldsymbol{d}^{(k)}\right\| \leq \eta^{(k)}\left\|\boldsymbol{r}\left(\boldsymbol{x}^{(k)}\right)\right\| \tag{6.6}$$

where $\|\cdot\|$ represents the Euclidian norm and $\eta^{(k)} \in [0, 1)$ defines the forcing term.

Choosing an appropriate value for the forcing term in the inexact Newton method involves balancing the computational cost of solving the linear system with sufficient accuracy against the number of nonlinear iterations needed to achieve convergence (Kan et al., 2022). A smaller forcing term can lead to fewer nonlinear iterations, but it also requires a more accurate solution of the linear system, which can be computationally expensive. Conversely, a larger forcing term can reduce the cost of computing the linear system solution while it may involve a larger number of nonlinear iterations for convergence. Based on numerical experiments conducted during the development of the proposed scheme, the strategy proposed by Gomes-Ruggiero, Lopes and Toledo-Benavides (2008) was here implemented. The author proved the superlinear convergence of the Newton-Krylov method using this strategy. Additionally, practical constraints, called safeguards, were adopted to improve the convergence of the method. In this sense,

$$\eta_{unsafe}^{(k)} = \frac{\left(b^{(k)}\right)^2}{(a^{(k)})^2 + (b^{(k)})^2}\left(\frac{1}{k+1}\right)^{\chi}\frac{\left\|\boldsymbol{r}\left(\boldsymbol{x}^{(k)}\right)\right\|}{\left\|\boldsymbol{r}(\boldsymbol{x}^{(k-1)})\right\|} \tag{6.7}$$

where χ is a control parameter that ensures $\lim_{k\to\infty} \eta^{(k)} = 0$, that is $\chi > 1$, and

$$a^{(k)} = \log_{10}\left(\frac{\left\|\boldsymbol{r}\left(\boldsymbol{x}^{(k)}\right)\right\|}{\left\|\boldsymbol{r}(\boldsymbol{x}^{(k-1)})\right\|}\right) \tag{6.8}$$

$$b^{(k)} = \log_{10}\left(p^{(k)} - p^{(k-1)}\right) \tag{6.9}$$

where $p^{(k)}$ is the current total number of function evaluations, also counting the evaluations related to the finite difference scheme for approximating the Jacobian-vector products, which is discussed in Section 6.1.3.

The following practical safeguards are adopted based on Gomes-Ruggiero, Lopes and Toledo-Benavides (2008):

$$\eta_{safe}^{(k)} = \begin{cases} \min\left(\eta_{unsafe}^{(k)}, \eta_{max1}\right), & \text{if } a < 0 \text{ and } k \leq 3 \\ \min\left(\eta_{unsafe}^{(k)}, \eta_{max2}\right), & \text{if } a < 0 \text{ and } k > 3 \\ \eta_{max1}, & \text{if } a \geq 0 \end{cases} \tag{6.10}$$

in which η_{max1} and η_{max2} are limit values for $\eta^{(k)}$. Finally, the forcing term is computed using the strategy proposed by Pernice and Walker (1998).

$$\eta^{(k)} = \begin{cases} \eta_{safe}^{(k)}, & \text{if } \eta_{safe}^{(k)} \leq 2\tau \\ 0.8\tau\left\|\boldsymbol{r}\left(\boldsymbol{x}^{(k)}\right)\right\|, & \text{if } \eta_{safe}^{(k)} > 2\tau \end{cases} \tag{6.11}$$

in which τ is the termination tolerance value. This verification is important to improve the quality of the final solution of the nonlinear numerical solver at the last iteration. In order to initialize the Newton-Krylov steps, $\eta^{(0)} = 0.1$ is adopted, as suggested by Rahul and De (2011, 2015b, 2015a), Bellavia and Berrone (2007), and Gomes-Ruggiero, Lopes and Toledo-Benavides (2008). Additionally, $\eta_{max1} = 0.1$ and $\eta_{max2} = 0.01$ is adopted, according to Gomes-Ruggiero, Lopes and Toledo-Benavides (2008).

6.1.2. Krylov subspace methods

Krylov subspace methods are iterative projection algorithms to solve, in general, linear systems. These methods are mostly applied in large-scale problems. Formalized as iterative methods, they solve the system $\boldsymbol{Ax} = \boldsymbol{b}$ based on the j-order Krylov subspace, which is defined by the following linear subspace:

$$\mathcal{K}_j(\boldsymbol{A}, \boldsymbol{r}_0) = \text{span}\left(\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \boldsymbol{A}^2\boldsymbol{r}_0, \cdots, \boldsymbol{A}^{j-1}\boldsymbol{r}_0\right) \tag{6.12}$$

where $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$, and $\boldsymbol{x}_0$ is an initial guess of the solution and equals to a zero vector. In this sense, the linear span $\mathcal{K}_j$ is defined as the set of all finite linear combinations of the vectors in $\left\{\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \boldsymbol{A}^2\boldsymbol{r}_0, \cdots, \boldsymbol{A}^{j-1}\boldsymbol{r}_0\right\}$. As $\boldsymbol{x} \in \mathcal{K}_j$, Krylov iterative methods approximate the solution of $\boldsymbol{Ax} = \boldsymbol{b}$ with the following summation:

$$\boldsymbol{x}_j = \boldsymbol{x}_0 + \sum_{i=0}^{j-1} \gamma_i \boldsymbol{A}^i \boldsymbol{r}_0 \tag{6.13}$$

in which γ_i is the scalar to be solved to minimize the residual norm $\left\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_j\right\|$.

Since the bases $\left\{\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \boldsymbol{A}^2\boldsymbol{r}_0, \cdots, \boldsymbol{A}^{j-1}\boldsymbol{r}_0\right\}$ of the Krylov subspace $\mathcal{K}_j$ tend to be linearly dependent, various techniques have been adopted to redefine novel bases (Knoll and Keyes, 2004; Kan et al., 2022). In this study, the Generalized Minimal Residual (GMRES) method (Saad and Schultz, 1986), which is based on the modified Gram-Schmidt method, was utilized for computing the bases to solve the linear system (6.3). Kelley (1995) provided specific details about the implementation of the method. In addition, Kelley (2003) offered a MATLAB implementation of the GMRES method that is tailored to this study. In this implementation, a reorthogonalization strategy (Kelley, 1995) is not utilized in the GMRES method. Notably, the choice of this method in this study is based on its ability to handle linear systems with nonsymmetric matrices.

6.1.3. Jacobian-vector product computations by using finite difference schemes

The GMRES algorithm requires one matrix-vector product per iteration, representing a Jacobian-vector product in the case of systems of nonlinear equations. In the context of the Jacobian-free Newton-Krylov method, the linear system (6.3) can be solved by approximating that product using a forward finite difference scheme, that is:

$$\boldsymbol{\mathcal{J}}(\boldsymbol{x})\boldsymbol{v} \approx \frac{\boldsymbol{r}(\boldsymbol{x}+\epsilon\boldsymbol{v})-\boldsymbol{r}(\boldsymbol{x})}{\epsilon} \tag{6.14}$$

in which ϵ denotes a small perturbation. Knoll and Keyes (2004) presented a demonstration of Equation (6.14).

It is also possible to adopt different schemes to approximate Jacobian-vector products. Besides the forward scheme, Pernice and Walker (1998) and An, Wen and Feng (2011) adopted the central finite difference scheme to compute the Jacobian-vector products, as given by

$$\boldsymbol{\mathcal{J}}(\boldsymbol{x})\boldsymbol{v} \approx \frac{\boldsymbol{r}(\boldsymbol{x}+\epsilon\boldsymbol{v})-\boldsymbol{r}(\boldsymbol{x}-\epsilon\boldsymbol{v})}{2\epsilon} \tag{6.15}$$

Equations (6.14) and (6.15) suffer from ill-conditioning due to catastrophic cancellation, making it challenging to devise a stable and accurate strategy. To address this issue, Kan et al. (2022) explored the use of the complex-step finite difference scheme in a Jacobian-free Newton-Krylov algorithm. The complex-step method evaluates the function $\boldsymbol{r}(\boldsymbol{x})$ at a complex argument, allowing for an accurate computation of the derivative. Considering the imaginary unit $i = \sqrt{-1}$,

$$\boldsymbol{\mathcal{J}}(\boldsymbol{x})\boldsymbol{v} \approx \frac{Im\big(\boldsymbol{r}(\boldsymbol{x}+i\epsilon\boldsymbol{v})\big)}{\epsilon} \tag{6.16}$$

where $Im(\cdot)$ denotes an operation to extract the imaginary part of the complex number. To implement this approach, the code must be able to handle complex numbers. However, some nonsmooth functions, such as the minimum function, can cause numerical issues. Therefore, in this work, such functions are smoothed to ensure the stability and accuracy of the algorithm.

Regarding the perturbation ϵ, also known as finite difference step size, Knoll and Keyes (2004), An, Wen and Feng (2011) and Kan et al. (2022) presented the following vastly applied expressions for forward and central finite difference schemes, respectively:

$$\epsilon = \frac{\sqrt{(1 + \|\boldsymbol{x}\|)\epsilon_{mach}}}{\|\boldsymbol{v}\|} \tag{6.17}$$

$$\epsilon = \frac{\sqrt{(1 + \|\boldsymbol{x}\|)}}{\|\boldsymbol{v}\|}\sqrt[3]{\epsilon_{mach}} \tag{6.18}$$

in which ϵ_{mach} is the machine epsilon. As the finite difference step size ϵ is not a major issue for the complex-step scheme, $\epsilon = 10^{-25}$ was adopted for all the analyses conducted in this study.

6.1.4. Line search strategy

The Newton-Raphson method is a locally convergent algorithm, which means that it is guaranteed to converge when the initial guess of the method is sufficiently near the solution. Therefore, the line search and trust-region methods are two popular globalization strategies for the Newton-Raphson method (Dennis and Schnabel, 1996; Kelley, 2003), as well as for the Newton-Krylov method (Brown and Saad, 1994; Eisenstat and Walker, 1994; Bellavia and Berrone, 2007).

In this work, a backtracking line search is implemented, based on the work of Eisenstat and Walker (1994), due to its simplicity and robustness. This algorithm has been applied by other researchers such as Pernice and Walker (1998), Bellavia and Berrone (2007), and Shin, Darvishi and Kim (2010). In the backtracking algorithm, the update for $\boldsymbol{x}^{(k)}$ is determined by iteratively adjusting the step size (or step length) $\alpha^{(k)}$ to sufficiently decrease $\left\|\boldsymbol{r}\left(\boldsymbol{x}^{(k+1)}\right)\right\|$. The update expression is as follows:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha^{(k)}\boldsymbol{d}^{(k)} \tag{6.19}$$

where $\boldsymbol{d}^{(k)}$ is a search direction. In this process, it is usually adopted $0 < \alpha^{(k)} \leq 1$ (Knoll and Keyes, 2004). To compute the step size, it is considered the contraction factor $\rho_m^{(k)}$ and the following expression to update the step size in each iteration:

$$\bar{\alpha}_{m+1}^{(k)} = \rho_m^{(k)} \bar{\alpha}_m^{(k)} \tag{6.20}$$

where m denotes the current iteration of the backtracking algorithm, $0 < \rho_m^{(k)} < 1$, and $\bar{\alpha}_0^{(k)} = 1$. Since $0 < t < 1$, the procedure consists of verifying condition (6.21) until it is satisfied, while repeatedly incrementing m and applying Equation (6.20). When a value $\bar{\alpha}_m^{(k)}$ leads to Equation (6.21) being satisfied, set $\alpha^{(k)} = \bar{\alpha}_m^{(k)}$.

$$\left\| \boldsymbol{r}\left(\boldsymbol{x}^{(k)} + \bar{\alpha}_m^{(k)} \boldsymbol{d}^{(k)}\right) \right\| \leq \left(1 - t(1 - \bar{\eta}_m)\right) \left\| \boldsymbol{r}\left(\boldsymbol{x}^{(k)}\right) \right\| \tag{6.21}$$

in which $\bar{\eta}_0^{(k)} = \eta^{(k)}$ and

$$\bar{\eta}_{m+1}^{(k)} = 1 - \rho_m^{(k)} \left(1 - \bar{\eta}_m^{(k)}\right) \tag{6.22}$$

Usually, the scalar $\rho_m^{(k)}$ varies at each iteration, based on a minimization of a quadratic function that interpolates $\|\boldsymbol{r}(\boldsymbol{x})\|$ along the search direction (Pernice and Walker, 1998; Bellavia and Berrone, 2007). Nonetheless, in this work, maintaining $\rho_m^{(k)}$ constant during the simulation was sufficient to solve the systems of nonlinear equations. This strategy is also applied in other algorithms such as Newton-Raphson (Kelley, 1995, 2003) and Broyden's (Li and Fukushima, 2000) methods. Therefore, $\rho_m^{(k)} = 0.5$ was adopted in this work. Furthermore, t in Equation (6.21) is set to a small value, $t = 10^{-4}$ (Pernice and Walker, 1998; Bellavia and Berrone, 2007; Shin, Darvishi and Kim, 2010). Typically, a minimum value for the step size is adopted, as suggested by Dennis and Schnabel (1996), and is set to $\alpha_{min} = 0.1$ for this work.

6.1.5. Updating the preconditioners by Broyden's formula

Preconditioning procedures are necessary to improve the rate of convergence of Krylov subspace methods, reducing the condition number of the matrix

associated with linear systems. In the context of the solution of Equation (6.3), the preconditioner $\boldsymbol{\mathcal{P}}^{(k)}$ is typically an approximation of $\boldsymbol{\mathcal{J}}^{(k)}$. Therefore, it is expected that the product $\boldsymbol{\mathcal{J}}^{(k)}\left(\boldsymbol{\mathcal{P}}^{(k)}\right)^{-1} \approx \boldsymbol{I}$ has a condition number close to unity (Souläimani, Salah and Saad, 2002). In Newton-Krylov frameworks, the right preconditioning is traditionally applied. In this case, Equation (6.3) becomes

$$\left(\boldsymbol{\mathcal{J}}^{(k)}\left(\boldsymbol{\mathcal{P}}^{(k)}\right)^{-1}\right)\left(\boldsymbol{\mathcal{P}}^{(k)}\boldsymbol{d}^{(k)}\right) = -\boldsymbol{r}\left(\boldsymbol{x}^{(k)}\right) \tag{6.23}$$

The preconditioning strategy employed in this work is based on Souläimani, Salah and Saad (2002). In order to solve computational fluid dynamics problems, Souläimani, Salah and Saad (2002) investigated enhancements in the GMRES method applied in a Newton-Raphson framework. The strategy of interest here involves computing only the initial preconditioner $\boldsymbol{\mathcal{P}}^{(0)}$ for the GMRES algorithm in the first iteration, and then updating the preconditioner for subsequent iterations using Broyden's formula. Bergamaschi et al. (2006, 2012) also used Broyden's update to compute preconditioners for Krylov subspace methods within an inexact Newton approach. Bergamaschi, Bru and Martínez (2011) adopted a similar strategy; however, they used the BFGS method, which is more appropriate for symmetric positive definite Jacobian matrices.

Elastoplastic problems can produce nonsymmetric Jacobian matrices, such as in nonassociated plasticity. Therefore, Broyden's formula is here adopted. Broyden's method is a widely employed algorithm for solving systems of nonlinear equations. It is a quasi-Newton method that replaces the exact Jacobian calculation for an approximate matrix. Broyden's update is defined as follows:

$$\boldsymbol{\mathcal{P}}^{(k+1)} = \boldsymbol{\mathcal{P}}^{(k)} + \frac{\left(\boldsymbol{y}^{(k)} - \boldsymbol{\mathcal{P}}^{(k)}\boldsymbol{s}^{(k)}\right)\left(\boldsymbol{s}^{(k)}\right)^{T}}{\|\boldsymbol{s}^{(k)}\|^{2}} \tag{6.24}$$

with

$$\boldsymbol{s}^{(k)} = \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)} \tag{6.25}$$

$$\boldsymbol{y}^{(k)} = \boldsymbol{r}^{(k+1)} - \boldsymbol{r}^{(k)} \tag{6.26}$$

Manipulating the preconditioners using inverse matrices can be useful. Therefore, employing the Sherman-Morrison formula, the following expression is obtained and applied in this work:

$$\left(\boldsymbol{\mathcal{P}}^{(k+1)}\right)^{-1} = \left(\boldsymbol{\mathcal{P}}^{(k)}\right)^{-1} - \frac{\left(\boldsymbol{s}^{(k)} - \left(\boldsymbol{\mathcal{P}}^{(k)}\right)^{-1}\boldsymbol{y}^{(k)}\right)\left(\boldsymbol{s}^{(k)}\right)^{T}\left(\boldsymbol{\mathcal{P}}^{(k)}\right)^{-1}}{(\boldsymbol{s}^{(k)})^{T}(\boldsymbol{\mathcal{P}}^{(k)})^{-1}\boldsymbol{y}^{(k)}} \tag{6.27}$$

In the proposed implicit integration scheme, $\left(\boldsymbol{\mathcal{P}}^{(0)}\right)^{-1}$ can be computed analytically in a straightforward manner. In this scenario, the recurrence formula (6.27) is used to update the preconditioner in the subsequent iterations. Thus, for any preconditioner $\boldsymbol{\mathcal{P}}$, the Jacobian-vector products performed in this work are computed using the following expressions:

$$(\boldsymbol{\mathcal{J}}(\boldsymbol{x})\boldsymbol{\mathcal{P}}^{-1})\boldsymbol{v} = \boldsymbol{\mathcal{J}}(\boldsymbol{x})(\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{v}) \approx \frac{\boldsymbol{r}(\boldsymbol{x} + \epsilon\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{v}) - \boldsymbol{r}(\boldsymbol{x})}{\epsilon} \tag{6.28}$$

$$(\boldsymbol{\mathcal{J}}(\boldsymbol{x})\boldsymbol{\mathcal{P}}^{-1})\boldsymbol{v} = \boldsymbol{\mathcal{J}}(\boldsymbol{x})(\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{v}) \approx \frac{\boldsymbol{r}(\boldsymbol{x} + \epsilon\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{v}) - \boldsymbol{r}(\boldsymbol{x} - \epsilon\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{v})}{2\epsilon} \tag{6.29}$$

$$(\boldsymbol{\mathcal{J}}(\boldsymbol{x})\boldsymbol{\mathcal{P}}^{-1})\boldsymbol{v} = \boldsymbol{\mathcal{J}}(\boldsymbol{x})(\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{v}) \approx \frac{Im\left(\boldsymbol{r}(\boldsymbol{x} + i\epsilon\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{v})\right)}{\epsilon} \tag{6.30}$$

Considering a linear system $\boldsymbol{Ax} = \boldsymbol{b}$, Equation (6.23) states that $\boldsymbol{A} = \boldsymbol{\mathcal{J}}^{(k)}\left(\boldsymbol{\mathcal{P}}^{(k)}\right)^{-1}$ and $\boldsymbol{x} = \boldsymbol{\mathcal{P}}^{(k)}\boldsymbol{d}^{(k)}$. In this sense, when using (6.28), (6.29) or (6.30) during the GMRES execution, the final result calculated by this method must be multiplied by $\left(\boldsymbol{\mathcal{P}}^{(k)}\right)^{-1}$.

6.2. Proposed state-update algorithm for multisurface plasticity

6.2.1. Newton-Krylov-based state-update scheme

The proposed algorithm is based on the Jacobian-free Newton-Krylov method with backtracking line search, which results in an unconditionally stable algorithm. Typically, this method is used in large-scale problems. However, Shin, Darvishi and Kim (2010) showed that the Newton-Krylov method can also perform

satisfactorily for small systems of nonlinear equations, although it was outperformed by the higher-order Newton-like methods in most cases.

In this scenario, the Newton-Krylov method discussed in Section 6.1 is utilized to solve the systems of nonlinear equations defined by vector functions (5.16) and (5.42). Consequently, the proposed algorithm is capable of handling plane stress problems. Unlike Algorithm 5, the proposed scheme requires fewer interventions in its structure to handle plane stress problems since the Jacobian-vector products are numerically performed using Equations (6.28), (6.29) or (6.30). In addition, the use of complementary functions helps to avoid schemes that define active yield surfaces, allowing similar implementations to handle both single and multisurface plasticity.

The algorithm also follows the basic structure of elastic predictor-plastic corrector. Assuming there is no plastic evolution in the elastic predictor phase, $\Delta\boldsymbol{\gamma}_{n+1}^{(0)} = \mathbf{0}$ simplifies the analytical calculation of the inverse of $\boldsymbol{\mathcal{P}}_{n+1}^{(0)}$, which is required by the GMRES method. In the general case, Equation (5.31) is adopted to compute the inverse of $\boldsymbol{\mathcal{P}}_{n+1}^{(0)} = \boldsymbol{\mathcal{J}}_{n+1}^{(0)}$, whereas Equation (5.49) is applied for plane stress condition. Therefore, Equations (5.31) and (5.49) are computed for

$$\boldsymbol{\Psi}_{n+1}^{(0)} = \begin{bmatrix} \boldsymbol{D} & \mathbf{0} \\ \mathbf{0} & -c^{q}\boldsymbol{I} \end{bmatrix} \tag{6.31}$$

which do not need the computation of $\partial_{\boldsymbol{\sigma}}\boldsymbol{n}_{i,n+1}^{(k)}$, $\partial_{\boldsymbol{q}}\boldsymbol{n}_{i,n+1}^{(k)}$, $\partial_{\boldsymbol{\sigma}}\boldsymbol{h}_{i,n+1}^{(k)}$, and $\partial_{\boldsymbol{q}}\boldsymbol{h}_{i,n+1}^{(k)}$, simplifying the implementation of the constitutive model in a numerical simulator.

While the Newton-Raphson method is insensitive to scaling parameters (Dennis and Schnabel, 1996), the parameters c^{q}, c^{ϕ} and c^{σ} play an important role in the current formulation due to the GMRES method. During the formulation of the proposed scheme, tests revealed that the GMRES method was affected by the units of the vector functions, mainly due to the presence of multiple units within these functions. Therefore, modifying these parameters can potentially improve the convergence of the method.

Regarding the parameters required by the scheme, it is adopted $\chi = 2.5$ for Equation (6.7) and $\rho_{m}^{(k)} = \rho = 0.5$ for Equations (6.20) and (6.22). Observe that Equation (6.11) needs the definition of the termination tolerance τ. Therefore,

tolerances for subvectors (related to the flow rule, hardening/softening law, complementary functions, and out-of-plane stress) of the vector function are adopted. Then, the definition of τ is given as follows:

$$\tau = \sqrt{(tol_1)^2 + (tol_2)^2 + (tol_3)^2 + (tol_4)^2} \tag{6.32}$$

in which tol_1, tol_2, tol_3, and tol_4 are the user-defined tolerances, respectively, for these subvectors. Importantly, the last tolerance is only utilized when dealing with plane stress problems.

6.2.2. Consistent tangent modulus

To ensure the independence of the proposed scheme from the derivatives of $\boldsymbol{n}_i(\boldsymbol{\sigma}, \boldsymbol{q})$ and $\boldsymbol{h}_i(\boldsymbol{\sigma}, \boldsymbol{q})$, the consistent tangent modulus $\boldsymbol{D}_{n+1}^{ep}$ is here computed using numerical differentiation. In view of Equations (5.31), (5.40) and (5.41), $\boldsymbol{D}_{n+1}^{ep}$ can be directly defined from the inverse of $\boldsymbol{\mathcal{J}}_{n+1}^{(k)}$, which can be obtained using a finite difference scheme applied to either vector functions (5.16) or (5.42) (Seifert and Schmidt, 2008; Lu et al., 2023). Therefore, forward, central and complex-step finite difference schemes are respectively presented in Equations (6.33), (6.34) and (6.35), which are responsible for computing each column of $\boldsymbol{\mathcal{J}}_{n+1}^{(k)}$.

$$\frac{\partial \boldsymbol{r}}{\partial x_j} \approx \frac{\boldsymbol{r}(\boldsymbol{x} + h_j \boldsymbol{e}_j) - \boldsymbol{r}(\boldsymbol{x})}{h_j} \tag{6.33}$$

$$\frac{\partial \boldsymbol{r}}{\partial x_j} \approx \frac{\boldsymbol{r}(\boldsymbol{x} + h_j \boldsymbol{e}_j) - \boldsymbol{r}(\boldsymbol{x} - h_j \boldsymbol{e}_j)}{2h_j} \tag{6.34}$$

$$\frac{\partial \boldsymbol{r}}{\partial x_j} \approx \frac{Im\left(\boldsymbol{r}(\boldsymbol{x} + i h_j \boldsymbol{e}_j)\right)}{h_j} \tag{6.35}$$

in which $\boldsymbol{r}(\boldsymbol{x})$ represents the vector function (5.16) or (5.42), $\frac{\partial \boldsymbol{r}}{\partial x_j}$ represents the jth column of $\boldsymbol{\mathcal{J}}_{n+1}^{(k)}$, and $\boldsymbol{e}_j$ is the jth unit vector. Similar to the schemes to approximating Jacobian-vector products, the finite difference step size h_j can be computed using some pratical rules, as discussed by Dennis and Schnabel (1996). In a general approach, for Equations (6.33) or (6.34), h_j is defined as

$$h_j = \text{sgn}(x_j)\max(|x_j|, typx_j)\, h_r \tag{6.36}$$

where $typx_j$ represents a typical value of x_j, in which $typx_j = 1$ can often be set in a general context (Miehe, 1996). In addition, h_r is respectively defined by the forward and central finite difference schemes as follows (Dennis and Schnabel, 1996):

$$h_r = \sqrt{\varepsilon_{mach}} \tag{6.37}$$

$$h_r = \sqrt[3]{\varepsilon_{mach}} \tag{6.38}$$

Finally, the sign function is here defined as

$$\text{sgn}(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ -1, & \text{if } x < 0 \end{cases} \tag{6.39}$$

Similarly to the Jacobian-vector products, for the complex-step scheme, a step size of $h_j = 10^{-25}$ is used for all the analyses in this study.

The proposed Newton-Krylov-based scheme, which is able to consider the plane stress plasticity, is summarized in Algorithm 6. Notably, this algorithm requires fewer modifications specific to the plane stress condition compared to Algorithm 5, primarily due to step 7 in Algorithm 5.

Algorithm 6 Newton-Krylov-based implicit integration scheme for multisurface plasticity

Initialize: $k = 0$, k_{max}, $\Delta\boldsymbol{\varepsilon}_{n+1}$, $\boldsymbol{\varepsilon}_n^p$, $\boldsymbol{\sigma}_n$, $\boldsymbol{q}_n$, $\boldsymbol{D}$, $\boldsymbol{C}$, $f_i(\boldsymbol{\sigma},\boldsymbol{q})$, $\boldsymbol{n}_i(\boldsymbol{\sigma},\boldsymbol{q})$, $\boldsymbol{h}_i(\boldsymbol{\sigma},\boldsymbol{q})$, N, tol_1, tol_2, tol_3, δ, c^γ, c^ϕ, χ, η_{max1}, η_{max2}, ρ, α_{min}, and t. In case of plane stress, initialize ν and tol_4, and compute $\boldsymbol{D}$, $\boldsymbol{C}$ according to the plane strain state

Procedure:

1: Elastic predictor:
$\boldsymbol{\sigma}_{n+1}^{(0)} = \boldsymbol{\sigma}_n + \boldsymbol{D}\Delta\boldsymbol{\varepsilon}_{n+1}$, $\boldsymbol{q}_{n+1}^{(0)} = \boldsymbol{q}_n$, $\Delta\boldsymbol{\gamma}_{n+1}^{(0)} = \boldsymbol{0}$, and $\boldsymbol{\varepsilon}_{n+1}^{p\,(0)} = \boldsymbol{\varepsilon}_n^p$
In case of plane stress, $\Delta\varepsilon_{zz,n+1}^{(0)} = -\frac{\nu}{1-\nu}\left(\Delta\varepsilon_{xx,n+1} + \Delta\varepsilon_{yy,n+1}\right)$

2: Check plastic condition:
Compute $f_{1,n+1}^{(0)}, f_{2,n+1}^{(0)}, \cdots, f_{N,n+1}^{(0)}$ according to the constitutive model
If $\max\left(f_{1,n+1}^{(0)}, f_{2,n+1}^{(0)}, \cdots, f_{N,n+1}^{(0)}\right) \leq tol_1$:
In case of plane stress, redefine $\boldsymbol{D}$ as the traditional 3×3 matrix in Equation (5.46)
Return $\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_{n+1}^{(0)}$, $\boldsymbol{q}_{n+1} = \boldsymbol{q}_{n+1}^{(0)}$, $\Delta\boldsymbol{\gamma}_{n+1} = \Delta\boldsymbol{\gamma}_{n+1}^{(0)}$, $\boldsymbol{\varepsilon}_{n+1}^p = \boldsymbol{\varepsilon}_{n+1}^{p\,(0)}$, and $\boldsymbol{D}_{n+1}^{ep} = \boldsymbol{D}$
End

3: Define vector function $\boldsymbol{r}_{n+1}(\boldsymbol{x}_{n+1})$ according to Equation (5.16). In case of plane stress, use (5.42) instead of (5.16). As $\boldsymbol{r}_{3,n+1}$ depends on $\phi_{i,n+1}$, use Equations (5.3), (5.6), or (5.9)

4: Compute $\left(\boldsymbol{\mathcal{P}}_{n+1}^{(0)}\right)^{-1}$ using Equation (5.31). In case of plane stress, use (5.49) instead of (5.31)

5: Define the initial guess of the solution $\boldsymbol{x}_{n+1}^{(0)} = \left(\boldsymbol{\sigma}_{n+1}^{(0)}, \boldsymbol{q}_{n+1}^{(0)}, \Delta\boldsymbol{\gamma}_{n+1}^{(0)}\right)$. In case of plane stress, use $\boldsymbol{x}_{n+1}^{(0)} = \left(\boldsymbol{\sigma}_{n+1}^{(0)}, \boldsymbol{q}_{n+1}^{(0)}, \Delta\boldsymbol{\gamma}_{n+1}^{(0)}, \Delta\varepsilon_{zz,n+1}^{(0)}\right)$

6: Check the stopping criteria:
If $\left\|\boldsymbol{r}_{3,n+1}^{(k)}\right\| \leq c^\phi tol_1$, $\left\|\boldsymbol{r}_{1,n+1}^{(k)}\right\| \leq tol_2$, and $\left\|\boldsymbol{r}_{2,n+1}^{(k)}\right\| \leq c^q tol_3$:
(In case of plane stress, also include $\left|\boldsymbol{r}_{4,n+1}^{(k)}\right| \leq tol_4$)
Apply a finite difference scheme ((6.33), (6.34), or (6.35)) for approximating $\boldsymbol{\mathcal{J}}_{n+1}^{(k)}$
Find the inverse of $\boldsymbol{\mathcal{J}}_{n+1}^{(k)}$
Select the elements of the matrix $\boldsymbol{\mathcal{J}}_{n+1}^{(k)}$ related to the stress condition to define $\boldsymbol{D}_{n+1}^{ep}$
$\boldsymbol{\varepsilon}_{n+1}^p = \boldsymbol{\varepsilon}_n^p + \Delta\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{C}\left(\boldsymbol{\sigma}_{n+1}^{(k)} - \boldsymbol{\sigma}_n\right)$
Return $\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_{n+1}^{(k)}$, $\boldsymbol{q}_{n+1} = \boldsymbol{q}_{n+1}^{(k)}$, $\Delta\boldsymbol{\gamma}_{n+1} = \Delta\boldsymbol{\gamma}_{n+1}^{(k)}$, $\boldsymbol{\varepsilon}_{n+1}^p$, and $\boldsymbol{D}_{n+1}^{ep}$
End
If $k \leq k_{max}$:
Stop the algorithm since it fails to converge
End

7: Compute $\boldsymbol{d}_{n+1}^{(k)}$ such that $\left\|\boldsymbol{r}_{n+1}^{(k)} + \boldsymbol{\mathcal{J}}_{n+1}^{(k)}\boldsymbol{d}_{n+1}^{(k)}\right\| \leq \eta_{n+1}^{(k)}\left\|\boldsymbol{r}_{n+1}^{(k)}\right\|$ based on the GMRES method applying $\left(\boldsymbol{\mathcal{P}}_{n+1}^{(k)}\right)^{-1}$. Calculate Jacobian-vector products using Equations (6.28), (6.29), or (6.30)

8: Perform the line search:
Define $\bar{\eta}_{m=0}^{(k)} = \eta_{n+1}^{(k)}$, in which $\eta_{n+1}^{(0)} = \eta_{max1}$. While repeatedly incrementing m, compute $\alpha_{n+1}^{(k)} = \max\{1, \rho, \rho^2, \ldots, \rho^m\} \geq \alpha_{min}$ that satisfies:
$\left\|\boldsymbol{r}\left(\boldsymbol{x}_{n+1}^{(k)} + \alpha_{n+1}^{(k)}\boldsymbol{d}_{n+1}^{(k)}\right)\right\| \leq \left(1 - t(1 - \bar{\eta}_m)\right)\left\|\boldsymbol{r}\left(\boldsymbol{x}_{n+1}^{(k)}\right)\right\|$ and $\bar{\eta}_{m+1}^{(k)} = 1 - \rho\left(1 - \bar{\eta}_m^{(k)}\right)$

9: Update stresses, internal variables, and plastic multiplier:
$\boldsymbol{x}_{n+1}^{(k+1)} = \boldsymbol{x}_{n+1}^{(k)} + \alpha_{n+1}^{(k)}\boldsymbol{d}_{n+1}^{(k)}$

10: Update the forcing term $\eta_{n+1}^{(k)}$ using Equation (6.11) and $\tau = \sqrt{(tol_1)^2 + (tol_2)^2 + \cdots}$

11: Update $\left(\boldsymbol{\mathcal{P}}_{n+1}^{(k)}\right)^{-1}$ using Equation (6.27)

12: Update counter variable:
$k \coloneqq k + 1$
Go to step 6)

7
Applications of the Newton-Krylov-based implicit integration algorithm

In this chapter, a series of finite element analyses is conducted to evaluate the effectiveness of the Newton-Krylov-based implicit integration scheme, as described in Algorithm 6. The analyses encompass both single and multisurface plasticity models, considering the presence of hardening or softening behavior. These applications were developed in a nonlinear finite element solver implemented in the MATLAB environment specifically for this thesis.

7.1.
Plane strain versus plane stress

A long thick-walled cylinder subjected to internal pressure is here modeled with the finite element mesh presented in Figure 19. This problem is inspired by the work of de Souza Neto, Perić and Owen (2008). Due to the symmetry of the problem, a quarter of the cross-section is discretized while applying appropriate boundary conditions for this symmetric case. The mesh consists of 12 quadratic quadrilateral elements with four integration points. For this long cylinder simulation, it is assumed plane strain conditions. Nonetheless, analyses under plane stress conditions are carried out using the same mesh. In this sense, an internally pressurized ring with unit thickness is investigated. The proposed Newton-Krylov-based integration scheme is particularly valuable in this study as it effectively considers both plane strain and plane stress states.

Regarding the constitutive model, the Drucker-Prager criterion is adopted for all analyzes, considering both perfect plasticity and softening behavior. The yield function of this criterion is expressed as follows:

$$f(\boldsymbol{\sigma}, \boldsymbol{q}) = q(\boldsymbol{\sigma}) + p(\boldsymbol{\sigma}) \tan \phi - d(\boldsymbol{q}) \tag{7.1}$$

where ϕ is the friction angle and $d(\boldsymbol{q})$ evolves according to the hardening/softening variable α as given by

$$d(\boldsymbol{q}) = d_0 + H\alpha \tag{7.2}$$

with the initial material cohesion d_0 and the hardening/softening modulus H. Thus, $H = 0$ represents perfect plasticity. Moreover, it is considered $\boldsymbol{n}(\boldsymbol{\sigma}, \boldsymbol{q}) = \partial_{\boldsymbol{\sigma}} f$ and $\boldsymbol{h}(\boldsymbol{\sigma}, \boldsymbol{q}) = 1$. Then, the hardening/softening variable evolves according to $\dot{\alpha} = \gamma$. For computational purposes, it is assumed $J_2 \geq 10^{-16}$.

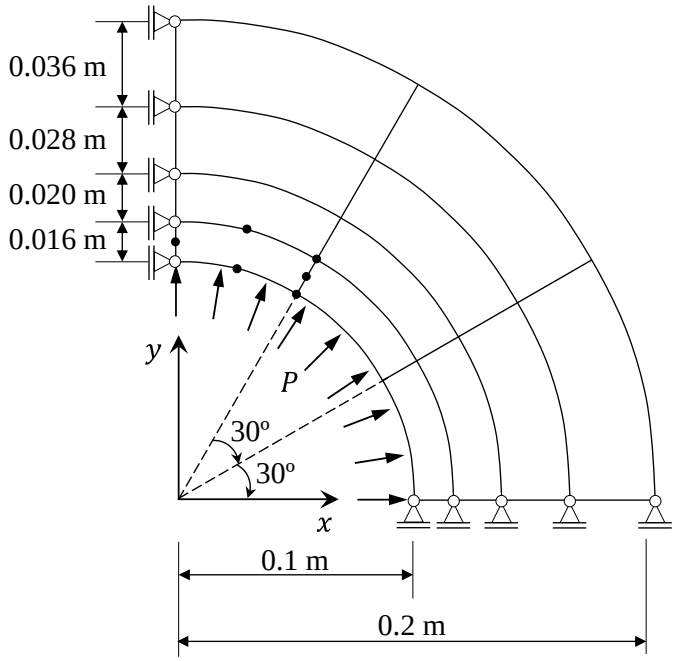

Figure 19 Internally pressurized thick-walled cylinder/ring: geometry, boundary conditions, and finite element mesh

The material properties for this problem are as follows: Young's modulus $E = 200$ GPa, Poisson's ratio $\nu = 0.3$, initial material cohesion $d_0 = 0.24$ GPa, and friction angle $\phi = 30°$. For this study, three different values for the hardening/softening modulus H are assumed: 0, -6 GPa, and -60 GPa. Notably, when $H = -60$ GPa, the material exhibits intense softening behavior, requiring a robust continuation method to solve the global system of nonlinear equations. In this sense, the variable displacement control proposed by Fujii et al. (1992) was here implemented. Leon et al. (2011) studied this control method, pointing its robustness in handling instability phenomena such as snap-through and snap-back. In contrast, other continuation methods, such as the spherical and cylindrical arc-

length methods, were unable to overcome the turning point of the equilibrium path for this case of severe softening under plane stress conditions.

The results obtained using the Newton-Krylov-based implicit integration scheme are presented in Figure 20. These results were obtained with the following scaling parameters: $c^q = c^\phi = c^\sigma = 1$. Moreover, the smooth FB complementary function in Equation (5.3) was adopted, considering $c^\gamma = 1$ and $\delta = 10^{-25}$. The consistent tangent modulus was computed using the complex-step finite difference scheme in Equation (6.35). For the numerical simulations, the tolerances adopted at the Gauss point and the global levels are equal to 10^{-6}.

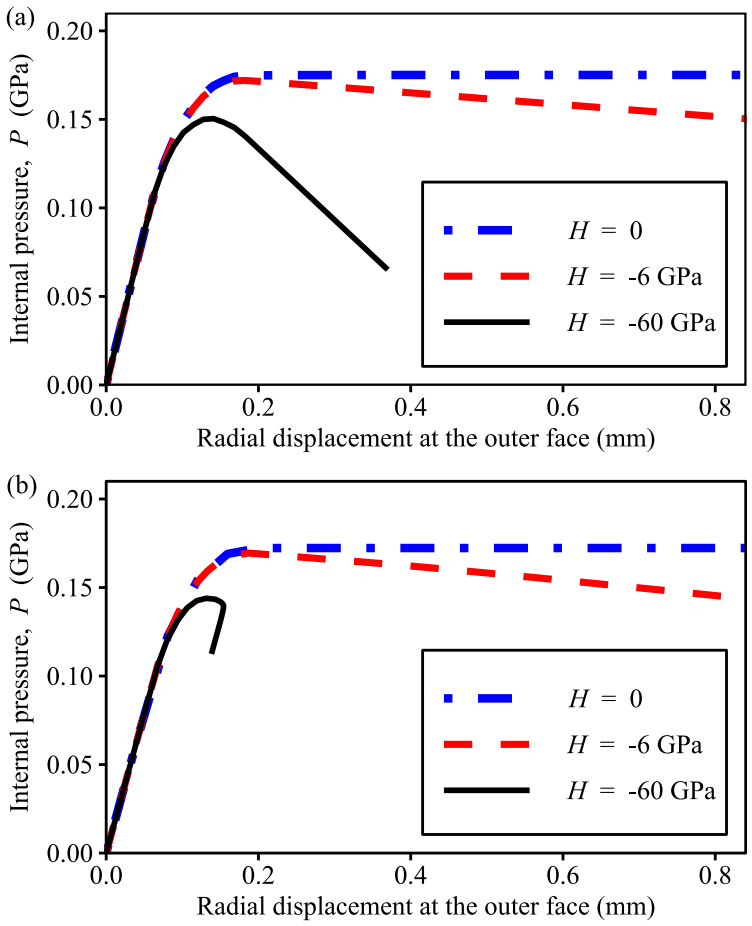

Figure 20 Pressure-displacement responses for the internally pressurized thick-walled cylinder/ring under (a) plane strain and (b) plane stress conditions

To validate the results in Figure 20, an alternative implementation of the state-update scheme was developed. This implementation is based on the implicit integration scheme for the Drucker-Prager criterion as presented by de Souza Neto, Perić and Owen (2008), and the outer Newton-Raphson scheme proposed by Dodds (1987) to enforce the plane stress constraint. In this scenario, the proposed Newton-

Krylov-based integration scheme and the alternative integration scheme produced the same results. This indicates that the Newton-Krylov-based scheme accurately computed the results, emphasizing its robustness in simulating softening behavior.

Under plane strain and plane stress conditions, the internally pressurized cylinder/ring presented similar responses for $H = 0$ and $H = -6$ GPa. Nevertheless, for the softening modulus $H = -60$ GPa, the internally pressurized ring under plane stress conditions exhibited a snap-back phenomenon. Figure 21 illustrates that only the finite elements near the inner face became plastified at the last analysis step. Elements further from the internal pressure P did not undergo plastification as the softening variable and, consequently, the plastic multiplier did not evolve. Hence, the internal pressure P is applied in severely degraded elements, leading to a global instability phenomenon.

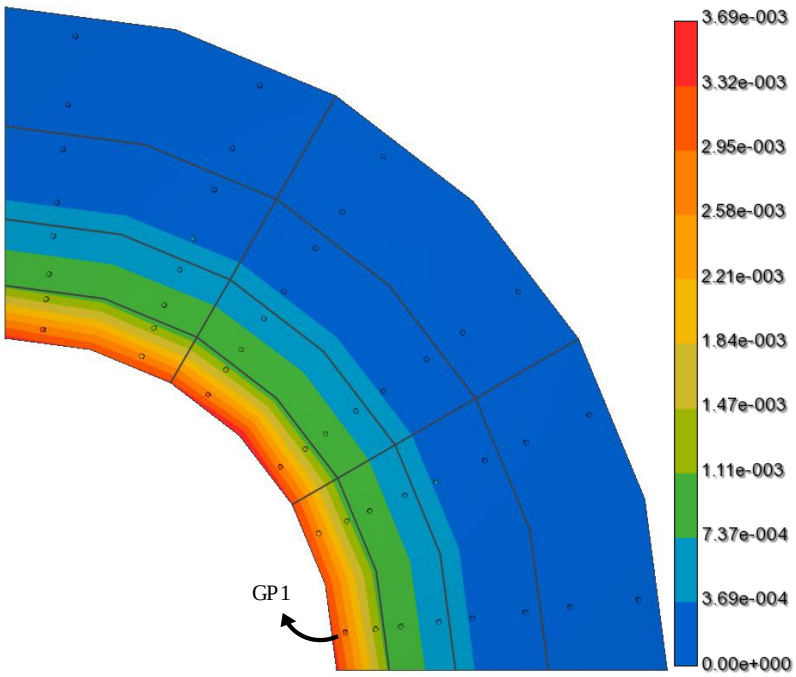

Figure 21 Spatial distribution of the softening variable α for the internally pressurized thick-walled ring at the last analysis step (plane stress conditions and $H = -60$ GPa)

Figure 22 shows the evolution of the softening variable at the first Gauss point of the model (indicated as GP1 in Figure 21). It is noteworthy that this variable consistently increases from the beginning of the plastification process until the end of the analysis, indicating that no artificial unloading is occurring.

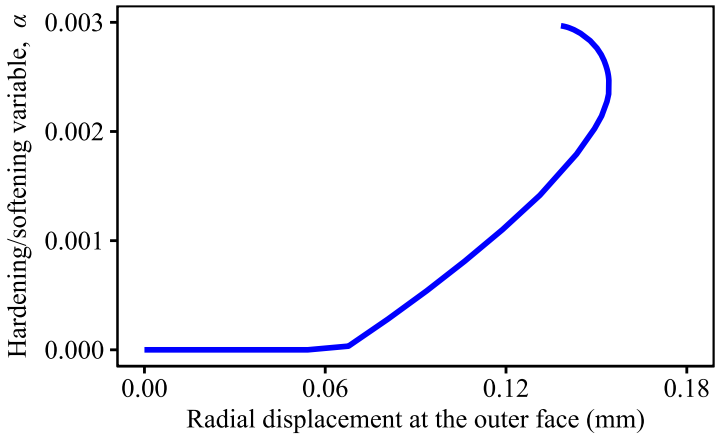

Figure 22 Evolution of the softening variable related to GP1 for the internally pressurized thick-walled ring (plane stress conditions and $H = -60$ GPa)

7.2. Computational time comparison

This section aims to investigate the computational efficiency of the proposed Newton-Krylov scheme in comparison with other finite difference-based Newton-Raphson schemes. The study involves evaluating three finite element problems. Firstly, the long internally pressurized cylinder described in Section 7.1 is utilized. Moreover, a wide bar subjected to pure bending (caused by two opposite nodal forces F) with a 90° V-shaped notch is considered (de Souza Neto, Perić and Owen, 2008). The finite element mesh of this bar is depicted in Figure 23. Lastly, a wide tapered cantilever loaded at its end is analyzed (de Souza Neto, Perić and Owen, 2008), with the finite element mesh shown in Figure 24 and a uniformly distributed shearing traction P. For both meshes, quadratic quadrilateral elements with four integration points were adopted.

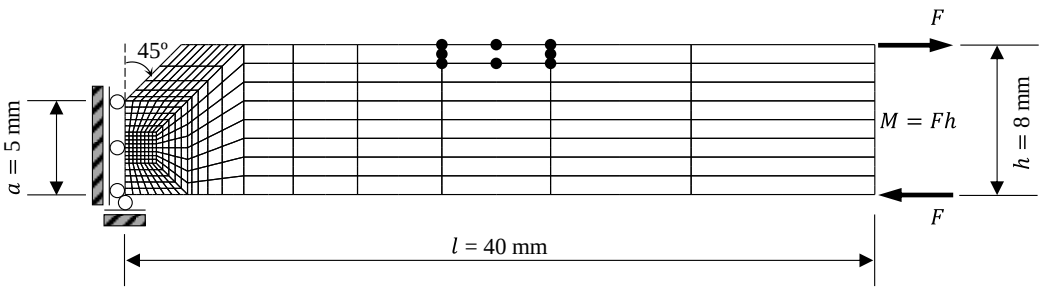

Figure 23 Bending of a V-notched bar: geometry, boundary conditions, and finite element mesh (adapted from de Souza Neto, Perić and Owen (2008))

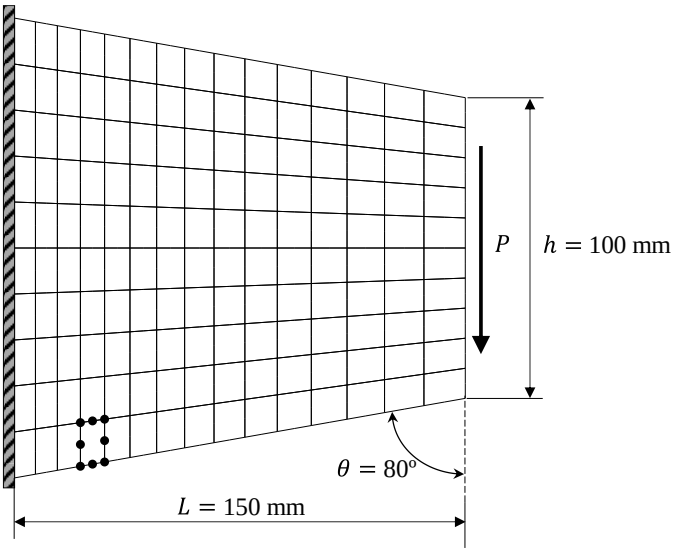

Figure 24 End-loaded tapered cantilever: geometry, boundary conditions, and finite element mesh (adapted from de Souza Neto, Perić and Owen (2008))

All analyses are conducted under plane strain conditions, as the cylinder, bar and cantilever are considered to be sufficiently wide. Furthermore, for all problems, the Tresca and Mohr-Coulomb criteria are assumed, following the yield function defined in Equation (4.1) and the parameters listed in Table 6. Parameter $\bar{\gamma}$, required for the Mohr-Coulomb criterion, is given by Equation (4.13). Additionally, the dilation angle ψ is considered equal to the friction angle ϕ.

Table 6 Shape and material parameters for the computational time comparison

Plastic model	**Shape parameter**				**Material parameter**	
	$\boldsymbol{\alpha}$	$\boldsymbol{\beta}$	$\boldsymbol{\gamma}$	$\boldsymbol{a}$	**Cohesion c (GPa)**	**Friction angle ϕ (°)**
Tresca	$\sec\left(\frac{\pi}{6}\right)$	0.9999	1	0	0.12	0
Mohr-Coulomb	$\sec\left(\frac{\pi}{6}(\bar{\gamma}+1)\right)$	0.99	$1-\bar{\gamma}$	$0.025c\cot\phi$	0.24	35

In this work, different finite difference-based Newton-Raphson schemes are investigated. A brief description of these schemes is shown in Table 7, detailing their characteristics, employed equations, and references in the literature when available. Importantly, scheme 2 utilizes analytical derivatives in the computations related to the updated stress. On the other hand, the consistent tangent modulus is computed by performing the state-update algorithm several times, perturbing each strain component to construct this modulus. Additionally, for this scheme, δ_{ij} is defined as the Kronecker delta.

Table 7 Description of the Newton-Raphson-based schemes

Newton-Raphson-based scheme	Description	Reference
Scheme 1	Calculate first and second derivatives of $f_{i,n+1}$ using the central finite difference scheme, computing the finite difference step size according to Pérez-Foguet, Rodríguez-Ferran and Huerta (2000a)	Choi and Yoon (2019) Pérez-Foguet, Rodríguez-Ferran and Huerta (2000a)
Scheme 2	Use (6.36) and (6.37) to compute $\left(\frac{d\boldsymbol{\sigma}_{n+1}}{d\boldsymbol{\varepsilon}}\right)_j = \frac{\boldsymbol{\sigma}(\boldsymbol{\varepsilon}_{n+1}+\boldsymbol{\epsilon})-\boldsymbol{\sigma}_{n+1}}{h_j}$ where $\epsilon_i = h_j\delta_{ij}$	Miehe (1996) Seifert and Schmidt (2008)
Scheme 3	Solve (6.3) using (6.5), (6.33), (6.36), and (6.37)	Seifert and Schmidt (2008)
Scheme 4	Substitute $\boldsymbol{r}(\boldsymbol{x})$ for $\boldsymbol{n}_{i,n+1}$ in (6.33) to compute $\partial_{\boldsymbol{\sigma}}\boldsymbol{n}_{i,n+1}$ by using (6.36) and (6.37)	Pérez-Foguet, Rodríguez-Ferran and Huerta (2000b)
Scheme 5	Solve (6.3) using (6.5) and (6.35)	-
Scheme 6	Substitute $\boldsymbol{r}(\boldsymbol{x})$ for $\boldsymbol{n}_{i,n+1}$ in (6.35) to compute $\partial_{\boldsymbol{\sigma}}\boldsymbol{n}_{i,n+1}$ by using $h_j = 10^{-25}$	-

Algorithm 6 is here examined by considering the complementary functions (5.3), (5.6) and (5.9). In another experiment, no complementary functions are assumed; employing Equation (5.2) leads to the traditional implicit integration algorithm. In this sense, a comprehensive study of the computational efficiency of these functions is developed. Additionally, various finite difference schemes listed in Table 8 to compute the Jacobian-vector products and the consistent tangent modulus are assessed. Regarding the scaling and dimensional parameters, the Newton-Krylov-based schemes considered the following: $c^q = c^\phi = c^\sigma = c^\gamma = 1$. Moreover, the smoothing parameters of the complementary functions in Equations (5.3), (5.6) and (5.9) were respectively assumed: $\delta = 10^{-25}$, $\delta = 0.003$, and $\delta = 10^{-8}$.

Table 8 Description of the Newton-Krylov-based schemes

Newton-Krylov-based scheme	Description
Scheme 1	Apply (6.28) and (6.33) in steps 7 and 6, respectively
Scheme 2	Apply (6.28) and (6.35) in steps 7 and 6, respectively
Scheme 3	Apply (6.30) and (6.33) in steps 7 and 6, respectively
Scheme 4	Apply (6.30) and (6.35) in steps 7 and 6, respectively

To obtain the equilibrium path of the studied finite element problems, two different continuation methods were applied: displacement control (Batoz and Dhatt, 1979) and cylindrical arc-length method (Crisfield, 1981). The former was used in the V-notched bar analysis, and the latter was applied in the internally pressurized cylinder and the tapered cantilever problems. To ensure a fair comparison, the same increments were prescribed for each finite element problem, regardless of the constitutive model. In addition, the tolerances adopted at the Gauss point and the global levels are both set to 10^{-6}. In this context, Figure 25 and Figure 26 display the numerical responses for the studied problems.

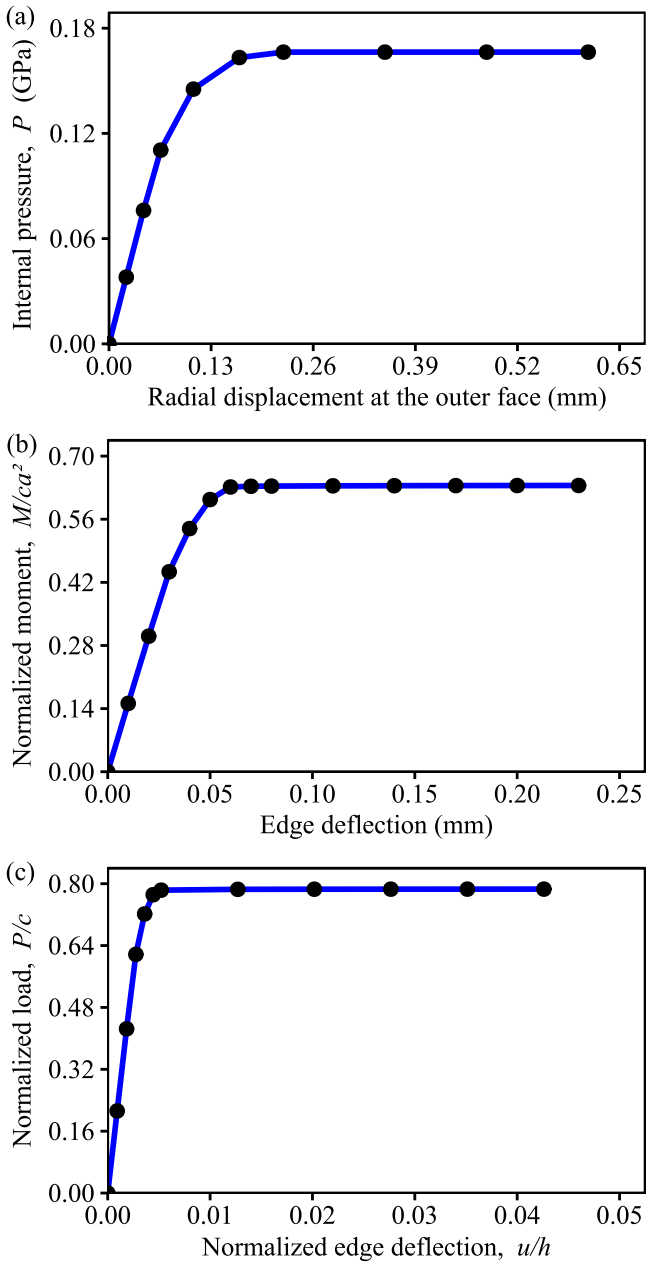

Figure 25 Finite element responses considering the Tresca criterion: (a) internally pressurized thick-walled cylinder, (b) bending of a V-notched bar, and (c) end-loaded tapered cantilever

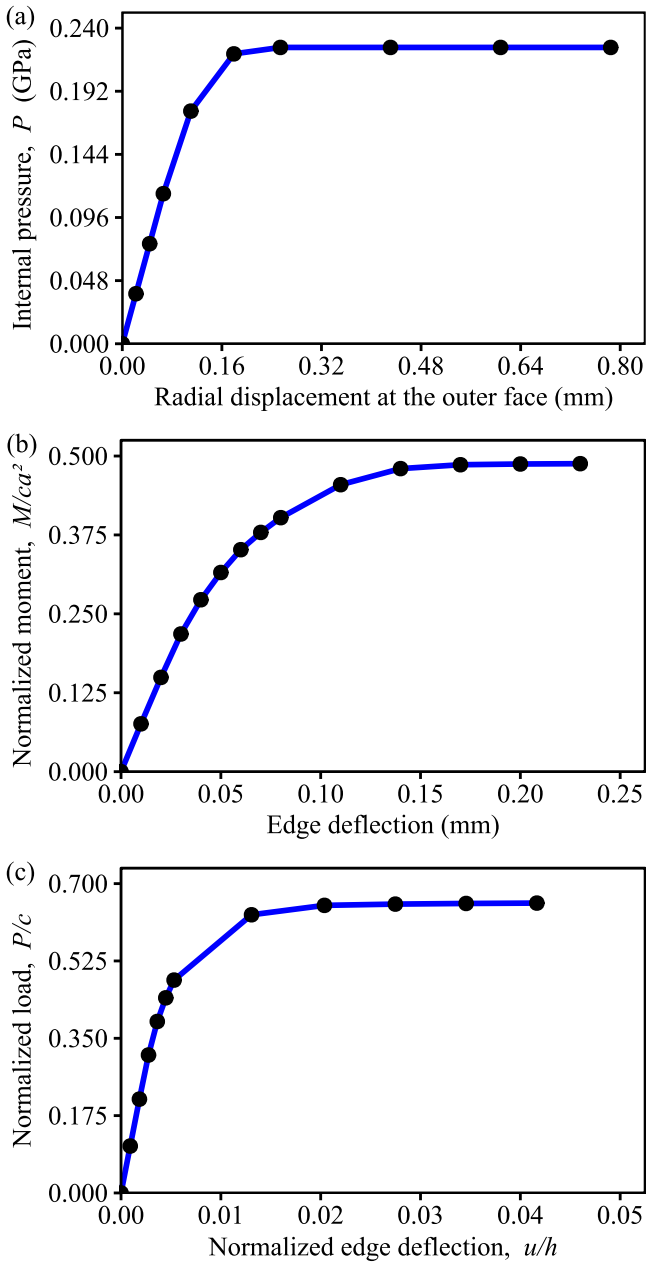

Figure 26 Finite element responses considering the Mohr-Coulomb criterion: (a) internally pressurized thick-walled cylinder, (b) bending of a V-notched bar, and (c) end-loaded tapered cantilever

Table 9 and Table 11 provide a summary of the relative computational times for the finite difference-based Newton-Raphson schemes, while Table 10 and Table 12 summarize the relative times for the Newton-Krylov-based schemes. Notably, these relative times are also computed by considering Algorithm 5, which employs analytical first- and second-order derivatives, a cubic interpolation-based line search scheme (Seifert and Schmidt, 2008), and the same scaling and dimensional parameters as the Newton-Krylov-based scheme.

Table 9 Relative computational time comparison amongst Newton-Raphson schemes for Tresca criterion

Function $\phi_{i,n+1}$	Problem	Algorithm 5	Scheme 1	2	3	4	5	6
Equation (5.2) (no complementary functions are assumed)	Internally pressurized thick-walled cylinder	1.000	4.276	2.759	1.960	**1.338**	2.601	1.739
	Bending of a V-notched bar	1.000	2.890	2.068	1.583	**1.196**	1.817	1.376
	End-loaded tapered cantilever	1.000	3.346	2.293	-	**1.254**	2.080	1.520
Equation (5.3) (smooth FB function)	Internally pressurized thick-walled cylinder	1.962	9.412	7.985	4.676	**2.580**	6.412	3.377
	Bending of a V-notched bar	1.469	5.639	4.621	2.982	**1.844**	3.853	2.284
	End-loaded tapered cantilever	1.679	7.578	5.942	3.911	**2.263**	5.166	2.899
Equation (5.6) (sigmoid-based function)	Internally pressurized thick-walled cylinder	3.153	11.947	8.928	6.022	**3.268**	8.191	4.339
	Bending of a V-notched bar	1.903	7.455	5.537	3.998	**2.416**	5.164	3.015
	End-loaded tapered cantilever	2.522	11.741	8.760	6.124	**3.375**	7.940	4.375
Equation (5.9) (smooth CHKS function)	Internally pressurized thick-walled cylinder	1.213	4.362	2.966	2.421	**1.366**	3.374	1.773
	Bending of a V-notched bar	0.998	2.856	2.207	2.151	**1.183**	2.149	1.388
	End-loaded tapered cantilever	1.026	3.326	2.533	-	**1.267**	2.579	1.532

To establish a baseline, the reference time is determined from a one-step implicit integration scheme with cubic interpolation-based line search. Thus, Algorithm 1 for single-surface plasticity and without substepping scheme is considered. As a result, the values presented in these tables specify how many times the computational time for a specific scheme is higher than the computational time related to Algorithm 1. The presented computational time is the average time of 20 executions of the finite element solver.

It is important to emphasize that certain analyses exhibited convergence issues for the prescribed increments. However, it does not necessarily imply that such schemes cannot solve the problem, being necessary to adjust the increments to achieve convergence. Regarding the Newton-Raphson schemes, it was observed that scheme 4 consistently outperformed the other finite difference-based schemes. In a general sense, schemes that incorporated the complex-step scheme tended to produce a more accurate consistent tangent modulus, but at the expense of higher computational costs. Therefore, a trade-off between accuracy and computational efficiency should be carefully considered when selecting the appropriate scheme for specific applications.

Table 10 Relative computational time comparison amongst Newton-Krylov-based implicit integration schemes for Tresca criterion

Function $\phi_{i,n+1}$	**Problem**	**Scheme**			
		1	**2**	**3**	**4**
Equation (5.2) (no complementary functions are assumed)	Internally pressurized thick-walled cylinder	**1.459**	1.753	1.633	1.852
	Bending of a V-notched bar	**1.167**	1.303	1.312	1.428
	End-loaded tapered cantilever	1.727	-	**1.445**	1.612
Equation (5.3) (smooth FB function)	Internally pressurized thick-walled cylinder	**1.926**	2.364	2.734	3.082
	Bending of a V-notched bar	**1.503**	1.728	1.997	2.155
	End-loaded tapered cantilever	**1.790**	2.099	2.459	2.743
Equation (5.6) (sigmoid-based function)	Internally pressurized thick-walled cylinder	**2.502**	2.908	3.527	3.999
	Bending of a V-notched bar	**2.033**	2.330	2.759	3.018
	End-loaded tapered cantilever	**2.715**	3.179	3.912	4.305
Equation (5.9) (smooth CHKS function)	Internally pressurized thick-walled cylinder	**1.688**	2.107	1.940	2.363
	Bending of a V-notched bar	**1.261**	1.530	1.438	1.622
	End-loaded tapered cantilever	1.731	2.661	**1.647**	1.909

The schemes that considered the smooth CHKS complementary function demonstrated promising results in terms of computational cost compared to other complementary functions. Nonetheless, it was noticed that using this function generally led to a slower scheme when compared to cases where no complementary functions were used. This observation is valuable as the smooth FB function and the sigmoid-based function are vastly employed in plasticity frameworks, and the CHKS complementary function can serve as a viable alternative for efficient simulations.

Table 11 Relative computational time comparison amongst Newton-Raphson schemes for Mohr-Coulomb criterion

Function $\phi_{i,n+1}$	**Problem**	**Algorithm 5**	**Scheme**					
			1	**2**	**3**	**4**	**5**	**6**
Equation (5.2) (no complementary functions are assumed)	Internally pressurized thick-walled cylinder	1.000	4.762	3.704	2.155	**1.431**	2.802	1.860
	Bending of a V-notched bar	1.000	3.195	2.519	1.517	**1.208**	1.871	1.414
	End-loaded tapered cantilever	1.000	4.039	2.852	1.838	**1.322**	2.392	1.674
Equation (5.3) (smooth FB function)	Internally pressurized thick-walled cylinder	1.675	7.931	6.905	4.113	**2.305**	5.621	2.902
	Bending of a V-notched bar	1.170	-	3.578	2.258	**1.458**	2.800	1.749
	End-loaded tapered cantilever	1.414	6.333	4.876	3.291	**1.875**	4.453	2.442
Equation (5.6) (sigmoid-based function)	Internally pressurized thick-walled cylinder	2.633	11.388	8.390	5.526	**2.939**	7.697	4.019
	Bending of a V-notched bar	1.466	5.662	4.559	2.853	**1.833**	3.774	2.335
	End-loaded tapered cantilever	1.545	7.157	5.418	3.690	**2.099**	4.945	2.695
Equation (5.9) (smooth CHKS function)	Internally pressurized thick-walled cylinder	1.081	4.805	4.080	2.537	**1.422**	3.630	1.890
	Bending of a V-notched bar	1.035	3.254	2.894	1.792	**1.236**	2.342	1.468
	End-loaded tapered cantilever	1.113	4.170	3.293	2.299	**1.367**	2.990	1.702

Table 12 Relative computational time comparison amongst Newton-Krylov-based implicit integration schemes for Mohr-Coulomb criterion

Function $\phi_{i,n+1}$	Problem	Scheme			
		1	**2**	**3**	**4**
Equation (5.2) (no complementary functions are assumed)	Internally pressurized thick-walled cylinder	**1.462**	1.727	1.993	2.175
	Bending of a V-notched bar	**1.206**	1.345	-	-
	End-loaded tapered cantilever	**1.253**	1.417	1.604	1.761
Equation (5.3) (smooth FB function)	Internally pressurized thick-walled cylinder	**1.788**	2.169	2.655	2.970
	Bending of a V-notched bar	**1.283**	1.591	1.957	1.845
	End-loaded tapered cantilever	**1.627**	1.883	2.435	2.723
Equation (5.6) (sigmoid-based function)	Internally pressurized thick-walled cylinder	**2.244**	2.652	3.439	3.785
	Bending of a V-notched bar	**1.706**	1.796	2.254	2.386
	End-loaded tapered cantilever	**1.706**	2.027	2.602	2.891
Equation (5.9) (smooth CHKS function)	Internally pressurized thick-walled cylinder	**1.643**	2.060	2.279	2.615
	Bending of a V-notched bar	**1.196**	1.503	-	-
	End-loaded tapered cantilever	**1.415**	1.643	1.939	2.187

The Newton-Krylov-based schemes demonstrated a noteworthy computational efficiency, with scheme 1 showing particularly promising results. Among the Newton-Krylov-based schemes, schemes 1 and 2 were the only ones able to outperform Algorithm 5 in four different analyses of the internally pressurized cylinder problem. The proposed Newton-Krylov-based algorithm proved to be faster than the finite difference-based Newton-Raphson schemes in most cases. However, the smooth CHKS function led to the fastest finite difference-based Newton-Raphson scheme (scheme 4) to be faster than the fastest Newton-Krylov-based scheme (scheme 1), except for the analysis of the bending of a V-

notched bar using the Tresca criterion. These observations highlight the computational advantages of the Newton-Krylov-based schemes, especially considering the effectiveness in various analyses of scheme 1 combined with the smooth CHKS function.

7.3. Implicit substepping scheme versus Newton-Krylov-based implicit integration scheme

The effectiveness of the proposed Newton-Krylov-based implicit integration scheme for multisurface plasticity under plane strain conditions is now evaluated. To this end, the Cap Model presented in Appendix C is utilized in the solution of two finite element problems: the bending of a V-notched bar and the end-loaded tapered cantilever. The geometry, boundary conditions, finite element mesh, continuation method, and tolerances of these problems were described in Section 7.2.

The material parameters assumed in Equations (C.1), (C.2), (C.3), and (C.6) for these problems are as follows: $E = 200$ GPa, $\nu = 0.2$, $\alpha = 0.35$ GPa, $\beta = 0.1$, $\lambda = 0.1$ GPa, $\theta = 0.7$, $R = 2$, $T = 0.08$ GPa, $\kappa_0 = -0.01$ GPa, $W = 0.05$, and $D = 0.1$ GPa^{-1}. As Equations (C.2) and (C.6) depend on the min function, one of its smooth versions is considered to compute $L(\kappa) = \min(\kappa, \kappa_0)$:

$$L(\kappa) = \frac{\kappa + \kappa_0}{2} - \frac{\sqrt{(\kappa - \kappa_0)^2 + \mu^2}}{2} \tag{7.3}$$

This equation is closely related to the smooth CHKS complementary function in (5.9). To ensure accurate numerical derivatives, it is assumed $\mu = 10^{-50}$. In this context, the consistent tangent modulus is computed using the complex-step finite difference scheme described in Equation (6.35), while Equation (7.3) plays a crucial role in ensuring its accuracy.

Regarding the Newton-Krylov-based implicit integration scheme, the scaling parameters $c^q = c^\phi = c^\sigma = 1$ are applied. Moreover, the smooth CHKS complementary function in Equation (5.9) is utilized with $c^\gamma = 10$ and $\delta = 10^{-8}$. The choice of $c^\gamma \neq 1$ was fundamental to improve the performance of the method

in terms of convergence. Lastly, the Jacobian-vector products are computed according to the complex-step finite difference scheme defined in Equation (6.30).

The numerical responses obtained using the proposed Newton-Krylov-based scheme are shown in Figure 27. Additionally, this scheme is compared with the implicit substepping algorithm proposed in Chapter 3, which utilizes an approach to identify active yield surfaces. Notably, the use of complementary functions satisfactorily replaces the process of identifying active surfaces and leads to accurate results. Note that the solution is not affected by the dimensional parameter $c^{\gamma} \neq 1$.

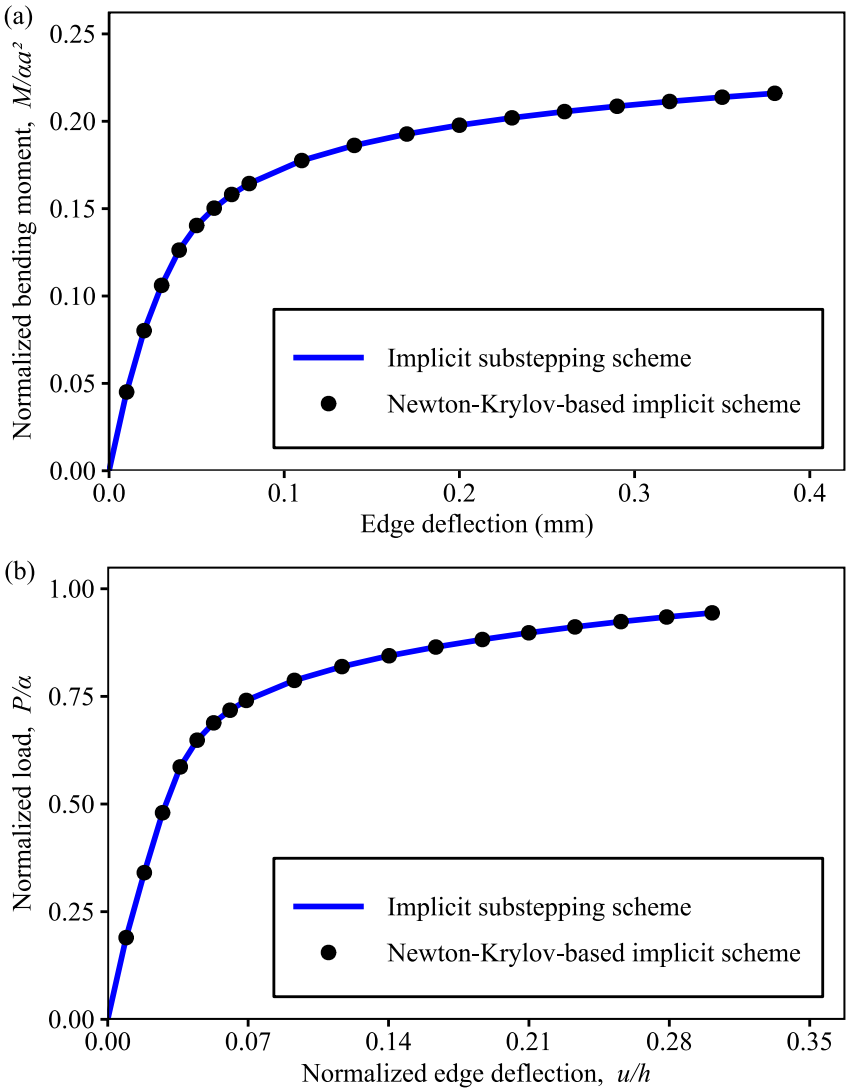

Figure 27 Finite element responses considering the Cap Model: (a) bending of a V-notched bar and (b) end-loaded tapered cantilever

8
Proposed coupled elastoplastic-damage modeling

The introduction of the damage concept has led to an increasing popularity of models involving elastic degradation and damage. These models are commonly used to describe the constitutive behavior of quasi-brittle materials, such as concrete, ceramics and rocks. In recent years, damage models have been widely accepted as an effective approach for describing material degradation. This approach offers versatility and simplicity, whereas being rigorously based on a fundamental constitutive theory.

The concept of damage was first formulated by Kachanov in 1958 (Kachanov, 1999) to model creep rupture. Since then, a significant number of works in the field of applied mechanics has been dedicated to formulating constitutive models that simulate the internal degradation of solids. After several decades of continuous development, significant progress has been made, and these theories have been merged into what is currently known as continuum damage mechanics.

Isotropic damage models, in which damage variables are formulated as scalar quantities for an isotropic representation of the medium degradation, are the most common constitutive model provided by the continuum damage mechanics. Several authors developed studies using this approach. Oliver et al. (1990) presented how to implement an state-update algorithm for isotropic damage, as well as a damage evolution law related to the fracture energy of the material. More recently, Kurumatani, Soma and Terada (2019) modeled reinforced concrete beams using an isotropic damage model. Experimental tests were used to validate the numerical model. Qi et al. (2020) proposed a damage constitutive model to simulate concrete structures. This model has material parameters that vary during the loading and damage process.

Jirásek (2011) described several pathological features related to strain-softening behavior in constitutive modeling, focusing on damage models. These features include: infinitely small softening regions; the potential occurrence of

snapbacks in the load-displacement response, independent of the size of the structure and the material ductility; and the total dissipated energy throughout the failure process is zero. According to Jirásek (2011), these features are associated to the loss of ellipticity of the governing differential equations, leading to ill-posed boundary value problems. Hence, there is no guarantee that the problem has a unique solution. In the finite element context, the ill-posed nature becomes evident through the pathological sensitivity of responses to finite element size.

In this scenario, several strategies have been developed in past years to mitigate the mesh dependency, such as viscous regularization (Simo and Ju, 1987a, 1987b; Faria, Oliver and Cervera, 1998; Wosatko et al., 2018), nonlocal models (Pijaudier-Cabot and Bažant, 1987; Bažant and Pijaudier-Cabot, 1989; de Vree, Brekelmans and van Gils, 1995; di Prisco and Mazars, 1996; De-Pouplana and Oñate, 2016; Pereira, Weerheijm and Sluys, 2017b), gradient-enhanced models (Peerlings et al., 1996; Jirásek, 2011; Wosatko et al., 2018), and micromorphic continuum theory (da Silva, Pitangueira and Penna, 2022; Reges, Pitangueira and Silva, 2023).

Bažant and Oh (1983) proposed a straightforward strategy to relate the finite element size and the crack band width, also known as characteristic length (Oliver, 1989). This length depends on factors such as the tangent stiffness pre- and post-peak, the ultimate tensile strength and the fracture energy, as described by Bažant and Oh (1983). The authors achieved the objectivity of the presented results for different finite element sizes. In this work, the crack band approach is adopted as a regularization technique to mitigate the mesh sensitivity. The estimation of the characteristic length is generally affected by the element size, element type, element shape, integration scheme, and mesh orientation (Jirásek and Bauer, 2012). However, the strategy presented by Kurumatani et al. (2016) is adopted, computing the characteristic length based on the area of each finite element, effectively reducing the mesh-size dependency. The mesh-bias dependency can be addressed by implementing crack tracking algorithms (Cervera et al., 2010).

In order to improve the capabilities of the constitutive modeling, coupling elastoplasticity and damage behavior is widely applied in the literature. Simo and Ju (1987a, 1987b) presented the theoretical and computational aspects of coupled elastoplastic-damage models, considering isotropic continuum damage modeling

and a cap plasticity model. Lee and Fenves (2001) presented a return mapping algorithm for a class of plastic-damage models according to both three-dimensional and plane stress conditions. Jason et al. (2006) studied a coupling technique of elastoplastic and damage models focusing on radial loading conditions for modeling a concrete column wrapped in a steel tube. Grassl et al. (2013) proposed a constitutive model based on the plasticity theory and damage mechanics in order to describe the failure of concrete structures. Mihai, Jefferson and Lyons (2016) presented a novel constitutive approach to simulate fiber reinforced concrete based on damage-plastic-contact modeling.

In this scenario, this work proposes a coupled elastoplastic-damage model, named Coupled Cracking-Crushing Elastoplastic-Damage (C3EPD) model, specifically designed to simulate concrete structures, with a focus on conventional concrete. This model captures the realistic behavior of concrete by incorporating distinct tensile and compressive damage evolution laws based on a bi-criterion approach. To regularize the softening behavior, the crack band width-based regularization technique is adopted for both tensile and compressive branches. Additionally, the multisurface plasticity formulation presented in Section 4.1, the CHGC model, is utilized to enhance the model's capability.

The literature related to realistically modeling the compressive behavior of concrete based on compressive damage criterion and compressive damage evolution law (without plasticity) is limited. Furthermore, to the best of the author's knowledge, the coupling of multisurface plasticity and a bi-criterion isotropic damage model has not been explored in other applications. Therefore, the proposed formulation aims to contribute to the field, with particular emphasis on presenting real-world applications of multisurface plasticity background and the involved computational aspects.

8.1.Continuum damage mechanics formulation

In the solid mechanics context, damage refers to the creation and growth of microcracks or microvoids, which represent discontinuities in a continuous medium in a large-scale problem. Thus, these degradations may be understood as the average effect of distributed microcracks or microvoids. In the engineering context, the

continuous mechanics introduces the concept of a Representative Volume Element (RVE), whose properties are denoted by homogenized variables (Lemaitre and Desmorat, 2005). Then, the damage discontinuities are irrelevant in relation to the dimension of the RVE. In this scenario, a damaged specimen may present volume defects such as microvoids, or surface defects such as microcracks. Therefore, the damage variable can be defined according to these volume and surface defects, producing distinct mathematical descriptions. When considering ductile materials, the damage variable D_v can be defined according to the volume density of microvoids (Lemaitre and Desmorat, 2005):

$$D_v = \frac{V_{voids}}{V_{RVE}} \tag{8.1}$$

where V_{voids} is the volume of microvoids and V_{RVE} is the volume of the RVE. In a general way, when microcracks and microvoids may exist, the surface density of microcracks S_D and intersections S of microvoids according to a cutting plane in the RVE physically define the damage variable (Lemaitre and Desmorat, 2005). The damage variable is computed considering the plane that maximizes this density, defined by a normal vector, as given in Figure 28.

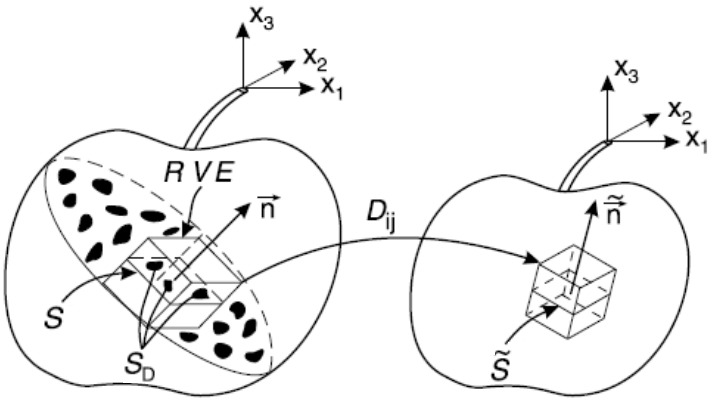

Figure 28 Physical damage and mathematical continuous damage (adapted from Lemaitre and Desmorat (2005))

Considering isotropic damage, which is adopted in this work, the scalar damage variable D does not depend on this normal vector. Therefore, the scalar damage variable here is given by

$$D = \frac{S_D}{S} \tag{8.2}$$

On the other hand, the effective stress is a tensor to which the material skeleton between microcracks or microvoids is subjected. In this scenario, the undamaged constitutive behavior describes the relationship between effective stress and total strain tensors. In the multiaxial case of isotropic damage, all the effective stress components act in the area without volume or surface discontinuities. Therefore, the effective stress tensor $\overline{\boldsymbol{\sigma}}$ can be defined as

$$\overline{\boldsymbol{\sigma}} = \frac{\boldsymbol{\sigma}}{1 - D} \tag{8.3}$$

where $\boldsymbol{\sigma}$ is the nominal stress tensor and D is a scalar variable that defines the damage level. Considering linear elasticity, the effective stress tensor is also given as

$$\overline{\boldsymbol{\sigma}} = \boldsymbol{D}: \boldsymbol{\varepsilon} = \boldsymbol{C}^{-1}: \boldsymbol{\varepsilon} \tag{8.4}$$

In order to describe the realistic concrete behavior, a bi-criterion damage model is adopted, as proposed by Mazars and Pijaudier-Cabot (1989). Therefore, two damage parameters are used to represent tensile and compressive damage variables. To simplify the mathematical formulation in this study, the symbol $\aleph$ is used to describe tensile stress states when $\aleph = t$ and compressive stress states when $\aleph = c$. In this context, the criteria that define damage growth take the following form:

$$\varphi_{\aleph}\left(\varepsilon_{\aleph}^{eq}, r_{\aleph}\right) = \varepsilon_{\aleph}^{eq} - r_{\aleph} \leq 0 \tag{8.5}$$

The equivalent strains $\varepsilon_{\aleph}^{eq}$ are scalar measures of the strain level. The internal variables $r_{\aleph}$ define the largest value of the referred measure, which is based on the deformation history of the material up to its current state. Therefore, the internal variables are given by

$$r_{\aleph}(T) = \max\left(r_{\aleph 0}, \max_{\tau \leq T} \varepsilon_{\aleph}^{eq}(\tau)\right) \tag{8.6}$$

in which T is the pseudo-time in the deformation history and $r_{\aleph 0}$ are the damage thresholds. These material parameters represent the initial value for the internal state variables. Hence, $r_{\aleph 0}$ indicate the equivalent strains at which the damage process begins. In general, $r_{\aleph 0}$ are defined as

$$r_{\aleph 0} = \frac{f_{\aleph 0}}{E} \tag{8.7}$$

where $f_{\aleph 0}$ are the uniaxial stresses at which the damage process is initialized and E is Young's modulus.

The evolution law of the damage variable D depends directly on the interval variables $r_{\aleph}$. Therefore, the damage variable D is defined by

$$D = g(r_t, r_c) \text{ such that } \begin{cases} g(r_t, r_c) = 0, & \text{if } r_t = r_{t0} \text{ and } r_c = r_{c0} \\ 0 < g(r_t, r_c) \leq 1, & \text{if } r_t > r_{t0} \text{ or } r_c > r_{c0} \end{cases} \tag{8.8}$$

As described by Jirásek (2011), the damage variable cannot decrease in order to adequately reproduce the loading-unloading-reloading process. For this reason, the Karush-Kuhn-Tucker conditions are introduced:

$$\varphi_{\aleph} \leq 0, \quad \dot{r}_{\aleph} \geq 0, \quad \dot{r}_{\aleph}\varphi_{\aleph} = 0 \tag{8.9}$$

Furthermore, the damage process must satisfy the consistency condition as described in Equation (8.10):

$$\dot{r}_{\aleph}\dot{\varphi}_{\aleph} = 0 \tag{8.10}$$

In Sections 8.2 and 8.3, the proposed model is described focusing on the definition of the equivalent strains $\varepsilon_{\aleph}^{eq}$ and the damage evolution law $g(r_t, r_c)$. When no associated with elastoplasticity, the following formulation is referred to as Cracking-Crushing Damage (CCD) model.

8.2. Proposed damage criteria

Two scalar measures $\varepsilon_{\aleph}^{eq}$ are applied in order to define the damage process. In general, these scalars are computed considering the strain tensor (Mazars, 1981; de Vree, Brekelmans and van Gils, 1995; Desmorat, Gatuingt and Ragueneau, 2007; De-Pouplana and Oñate, 2016; Pereira, Weerheijm and Sluys, 2017a; Moraes et al., 2020). However, the equivalent strains are computed in this work using the stress tensor divided by Young's modulus E, as applied by He et al. (2015, 2019). Through this approach, yield surfaces of plastic problems can be utilized as damage

criteria. Several works presented a good fit between their numerical results and experimental tests considering the criteria based on stress states (Cervera, Oliver and Faria, 1995; Cervera, Oliver and Manzoli, 1996; Faria, Oliver and Cervera, 1998; Wu, Li and Faria, 2006; Petracca et al., 2017; Saloustros et al., 2017).

In some models described in the literature, as a consequence of the mathematical definition of bi-criterion damage models, damage due to compressive stresses may emerge in tensile tests, and vice-versa (Wu, Li and Faria, 2006; Mazars, Hamon and Grange, 2015). Thus, Saloustros et al. (2017) and Petracca et al. (2017) suggested using the Heaviside function $H(\cdot)$ to inactivate the tensile criterion under compressive stress state and vice-versa.

This work adopts the criteria based on the yield surfaces proposed by Lubliner et al. (1989), combined with the propositions by He et al. (2015), Saloustros et al. (2017) and Petracca et al. (2017). Therefore, the equivalent strains are given by

$$\varepsilon_t^{eq} = \frac{H(\bar{\sigma}_{max})}{E}\left(\alpha_t \bar{I}_1 + \beta_t \sqrt{3\bar{J}_2} + \gamma_t \langle \bar{\sigma}_{max} \rangle\right) \tag{8.11}$$

$$\varepsilon_c^{eq} = \frac{H(-\bar{\sigma}_{min})}{(1-\alpha_c)E}\left(\alpha_c \bar{I}_1 + \beta_c \sqrt{3\bar{J}_2} - \gamma_c \langle -\bar{\sigma}_{max} \rangle\right) \tag{8.12}$$

in which $\bar{\sigma}_{max}$ and $\bar{\sigma}_{min}$ are respectively the maximum and minimum principal effective stresses; $\bar{I}_1$ represents the first invariant of the effective stress tensor; $\bar{J}_2$ represents the second invariant of the deviatoric effective stress tensor; and $\langle \cdot \rangle$ denotes the Macaulay brackets. Parameters $\alpha_{\aleph}$, $\beta_{\aleph}$ and γ_t can be expressed according to the strength of the material. In this context, f_{t0} and f_{cm} are the uniaxial tensile and compressive strength of the material, and f_{bt} and f_{bc} are the biaxial tensile and compressive strength of the material, respectively. Therefore, $\alpha_{\aleph}$, $\beta_{\aleph}$ and $\gamma_{\aleph}$ are defined as

$$\alpha_t = \frac{f_{t0}}{f_{bt}} - 1 \tag{8.13}$$

$$\beta_t = \frac{f_{t0}}{f_{tc}} + \alpha_t \tag{8.14}$$

$$\gamma_t = 1 - \alpha_t - \beta_t \tag{8.15}$$

$$\alpha_c = \frac{f_{bc} - f_{cm}}{2f_{bc} - f_{cm}} \tag{8.16}$$

$$\beta_c = 1 \tag{8.17}$$

$$\gamma_c = \frac{3(1-\rho)}{2\rho - 1} \tag{8.18}$$

As suggested by Lubliner et al. (1989), a typical value of the parameter ρ is $2/3$, also assumed in the analyses presented in this work. The parameter f_{tc} adjusts the tensile criterion shape. It is considered that $f_{tc} = 5f_{cm}$ is a representative value for this parameter.

8.3. Proposed damage evolution law

Mazars (1986) proposed an isotropic damage model that combines two scalar damage variables to consider tensile and compressive effects. For this purpose, the damage evolution law depended on the internal variable and the strain tensor. Some authors, such as Mazars, Berthaud and Ramtani (1990), Pijaudier-Cabot, Mazars and Pulikowski (1991), Tao and Phillips (2005) and Jason et al. (2006), adopted similar approaches. In a different strategy, Lee and Fenves (1998b) proposed a form of coupling the tensile and compressive damage variables using only the internal variables. This approach is adopted in this work and takes the following form:

$$g(r_t, r_c) = 1 - \big(1 - g_t(r_t)\big)\big(1 - g_c(r_c)\big) \tag{8.19}$$

This coupling approach was also applied by other authors such as Lee and Fenves (1998a), Alfarah, López-Almansa and Oller (2017), Chi et al. (2017), Pereira, Weerheijm and Sluys (2017b), and Poliotti and Bairán (2019). Therefore, based on the crack band approach, the tensile and compressive damage evolution laws are defined below. Appendix D details the derivation of the tensile and compressive damage evolution laws.

8.3.1. Uniaxial tensile response

In this work, the tensile and compressive damage evolution laws are derived from convenient stress-strain responses that agree with experimental observations

of the concrete behavior, as suggested by Alfarah, López-Almansa and Oller (2017). For this purpose, analytical expressions are previously defined for the uniaxial response of concrete after the beginning of the damage process. The tensile stress-strain responses adopted lead to the exponential damage evolution law presented by Kurumatani et al. (2016). According to this law, a sudden decrease in stress occurs before the elastic domain. Hence, the uniaxial stress f_{t0}, at which the damage process starts, equals the uniaxial tensile strength of the material, as presented in Figure 29.

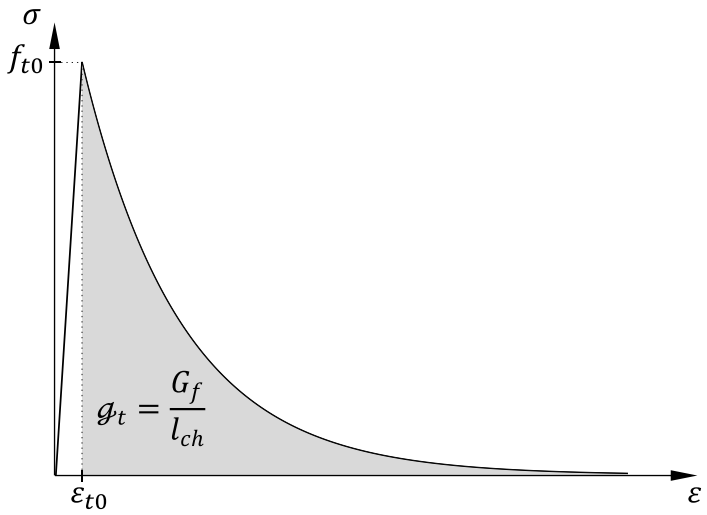

Figure 29 Typical uniaxial tensile response for a conventional concrete (the area of the shaded region is equal to $\mathcal{G}_t$)

In this sense, this stress-strain relation is defined by

$$\sigma = f_{t0}\exp\big(-a_t(\varepsilon - \varepsilon_{t0})\big), \text{ if } \varepsilon \geq \varepsilon_{t0} \tag{8.20}$$

in which σ and ε are uniaxial stress and strain, respectively; $\varepsilon_{t0} = r_{t0}$ indicates the strain at which the damage process begins; and a_t is a material parameter to be defined later. Therefore, the tensile damage evolution law related to the damage criterion (8.11) is given by

$$g_t(r_t) = 1 - \frac{r_{t0}}{r_t}\exp\big(-a_t(r_t - r_{t0})\big) \tag{8.21}$$

In order to obtain the parameter a_t, $\mathcal{G}_\aleph$ is defined as the energy per unit volume dissipated by damage during the degradation process. Furthermore, $\mathcal{G}_\aleph$ represents the finite area under the stress-strain curve in which damage is evolving.

Importantly, $\mathcal{G}_{\aleph}$ can be related to the fracture energy G_f and the crunching energy G_c, according to the following equation:

$$\mathcal{G}_t = \frac{G_f}{l_{ch}} \tag{8.22}$$

$$\mathcal{G}_c = \frac{G_c}{l_{ch}} \tag{8.23}$$

where l_{ch} is the characteristic length of the element. For quadrilateral finite elements, this length can be adopted as the square root of the element area (Kurumatani et al., 2016; He et al., 2019). In this sense, all elements with the same material properties comply with the same stress versus crack opening displacement response. Therefore, $\mathcal{G}_t$ takes the form

$$\mathcal{G}_t = \int_{\varepsilon_{t0}}^{\infty} \sigma d\varepsilon = \int_{\varepsilon_{t0}}^{\infty} f_{t0}\exp\big(-a_t(\varepsilon - \varepsilon_{t0})\big)d\varepsilon = \frac{f_{t0}}{a_t} \tag{8.24}$$

Based on Equations (8.22) and (8.24),

$$a_t = \frac{f_{t0} l_{ch}}{G_f} \tag{8.25}$$

Note that the stress-strain behavior is adapted according to the element size since the parameter a_t in Equation (8.25) changes for each finite element.

8.3.2. Uniaxial compressive response

To reproduce realistically the behavior of concrete, two different analytical expressions are adopted to define the compressive stress-strain curve. In this sense, the ascending branch is modeled by a quadratic Bézier curve, as suggested by Petracca et al. (2016, 2017), whereas the descending branch is modeled by an exponential curve based on the work of Winkler, Hofstetter and Lehar (2004).

Regarding the ascending branch, quadratic Bézier curves are parametric curves that require the definition of control points. Petracca et al. (2017) recommended obtaining control points based on experimental data. In the present

work, a set of prefixed control points is proposed. Considering f_{cm} as the uniaxial compressive strength of the concrete and r_{cm} as the strain that corresponds to the ultimate uniaxial compressive strength, the proposed set is given by:

$$P_0 = (r_{c0}, f_{c0}) \tag{8.26}$$

$$P_1 = (r_{ci}, f_{cm}) \tag{8.27}$$

$$P_2 = (r_{cm}, f_{cm}) \tag{8.28}$$

in which

$$r_{ci} = \frac{f_{cm}}{E} \tag{8.29}$$

These control points ensure that the complete stress-strain curve is smooth, providing its first-order derivative equals E at its beginning and zero at the peak stress. Concerning the descending branch, the adopted exponential curve provides a finite area under its curve, making it possible to compute $\mathscr{g}_c$ adequately. This expression is inspired by the work of Winkler, Hofstetter and Lehar (2004). The adopted stress-strain response for compression is shown in Figure 30.

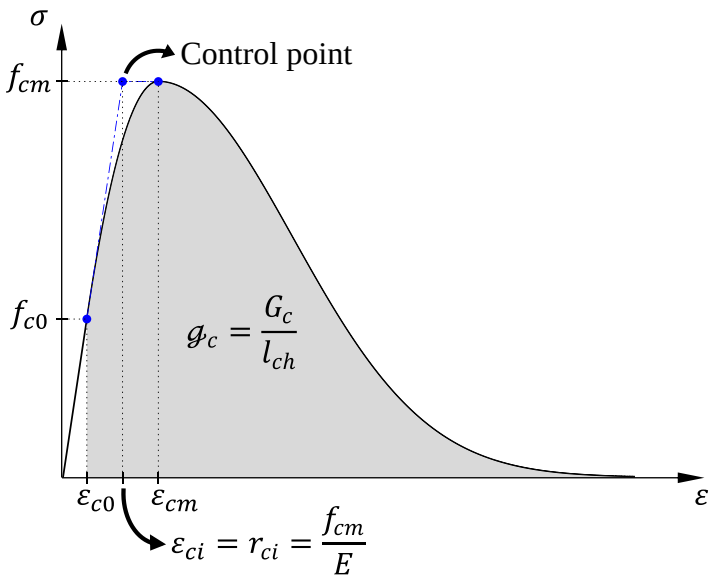

Figure 30 Typical uniaxial compressive response for a conventional concrete (the area of the shaded region is equal to $\mathscr{g}_c$)

Then, the compressive uniaxial response of concrete after the beginning of the nonlinear behavior is defined by

$$\sigma = \begin{cases} (1-t_c)^2 f_{c0} + (2-t_c) t_c f_{cm}, & \text{if } \varepsilon_{c0} \le \varepsilon < \varepsilon_{cm} \\ f_{cm} \exp(-d_c(\varepsilon - \varepsilon_{cm})^2), & \text{if } \varepsilon \ge \varepsilon_{cm} \end{cases} \tag{8.30}$$

where $\varepsilon_{c0} = r_{c0}$, $\varepsilon_{cm} = r_{cm}$, d_c is a material parameter to be defined later, and

$$t_c = \begin{cases} \dfrac{-b_c + \sqrt{(b_c)^2 - 4a_c(r_{c0} - r_c)}}{2a_c}, & \text{if } a_c \neq 0 \\ \dfrac{r_c - r_{c0}}{b_c}, & \text{if } a_c = 0 \end{cases} \tag{8.31}$$

$$a_c = r_{c0} - 2r_{ci} + r_{cm} \tag{8.32}$$

$$b_c = 2(r_{ci} - r_{c0}) \tag{8.33}$$

Therefore, considering $\varepsilon_c = r_c$, the compressive damage evolution law related to the damage criterion (8.12) is given by

$$g_c(r_c) = \begin{cases} 1 - \dfrac{r_{c0}}{r_c}(1-t_c)^2 - \dfrac{r_{ci}}{r_c}(2-t_c)t_c, & \text{if } r_{c0} \le r_c < r_{cm} \\ 1 - \dfrac{r_{ci}}{r_c}\exp(-d_c(r_c - r_{cm})^2), & \text{if } r_c \ge r_{cm} \end{cases} \tag{8.34}$$

In order to determine the parameter d_c, $\mathcal{G}_c$ is defined according to Equation (8.23). Based on Equation (8.30), $\mathcal{G}_c$ can be also computed by

$$\mathcal{G}_c = \int_{r_{c0}}^{\infty} \sigma d\varepsilon = \int_{r_{c0}}^{r_{cm}} \sigma d\varepsilon + \int_{r_{cm}}^{\infty} \sigma d\varepsilon = \mathcal{G}_{c1} + \mathcal{G}_{c2} \tag{8.35}$$

in which

$$\mathcal{G}_{c1} = -(f_{c0} + f_{cm})\frac{r_{c0}}{2} + (f_{c0} - f_{cm})\frac{r_{ci}}{3} + (f_{c0} + 5f_{cm})\frac{r_{cm}}{6} \tag{8.36}$$

$$\mathcal{G}_{c2} = \frac{f_{cm}}{2}\sqrt{\frac{\pi}{d_c}} \tag{8.37}$$

Based on Equations (8.23), (8.35), (8.36), and (8.37), the parameter d_c takes the form

$$d_c = \frac{\pi}{4}\left(\frac{f_{cm} l_{ch}}{G_c - g_{c1} l_{ch}}\right)^2 \tag{8.38}$$

Importantly, Equations (8.35) and (8.38) limit the characteristic length and, consequently, the size of the finite element. Then, the length l_{ch} must comply with the following expression:

$$l_{ch} < \frac{G_c}{g_{c1}} \tag{8.39}$$

Considering biaxial stress states, the damage criteria defined in Equations (8.5), (8.11), and (8.12) and the damage evolution law defined in Equations (8.8), (8.19), (8.21), and (8.34) lead to the numerical biaxial response presented in Figure 31, which agrees with the investigations by Kupfer, Hilsdorf and Rüsch (1969). The following parameters were adopted in this comparison: $E = 32$ GPa, $\nu = 0.20$, $f_{t0} = 2.9$ MPa, $f_{c0} = 0.33 f_c$, $f_{cm} = 32$ MPa, $f_{bc} = 1.16 f_{cm}$, $f_{bt} = 0.95 f_{t0}$, and $r_{cm} = 0.0022$. Notably, the proposed formulation effectively defines this biaxial strength envelope using parameters that are largely known or can be estimated.

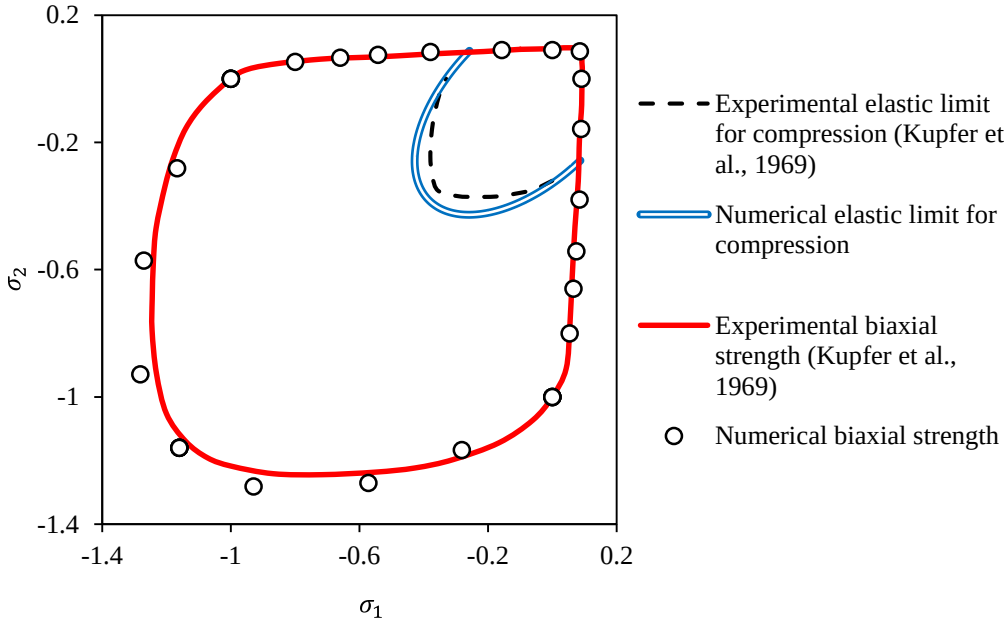

Figure 31 Comparison of numerical and experimental response of biaxial concrete behavior

8.4.Material tangent modulus

To define the tangent modulus, the constitutive relation in Equation (8.3) for the isotropic damage model has to be differentiated with respect to time, leading to

$$\dot{\boldsymbol{\sigma}} = (1 - D)\dot{\bar{\boldsymbol{\sigma}}} - \dot{D}\bar{\boldsymbol{\sigma}} \tag{8.40}$$

Thus, the tangent modulus can be obtained

$$\frac{d\boldsymbol{\sigma}}{d\boldsymbol{\varepsilon}} = (1 - D)\frac{d\overline{\boldsymbol{\sigma}}}{d\boldsymbol{\varepsilon}} - \overline{\boldsymbol{\sigma}} \otimes \frac{dD}{d\boldsymbol{\varepsilon}} \tag{8.41}$$

According to Equation (8.4), the effective tangent modulus for linear elasticity is defined as

$$\frac{d\overline{\boldsymbol{\sigma}}}{d\boldsymbol{\varepsilon}} = \boldsymbol{D} = \lambda \mathbf{1} \otimes \mathbf{1} + 2\mu \boldsymbol{I} \tag{8.42}$$

where λ and μ are the Lamé constants, $\mathbf{1}$ is the second-order identity tensor, and $\boldsymbol{I}$ is the fourth-order symmetric identity tensor.

The material tangent modulus depends on the tensile and compressive damage evolution laws. Jason et al. (2006) suggested that the plastic strains do not affect the damage evolution. Consequently, the damage variable exhibits a direct relation with the elastic strain. By deriving Equation (8.8) and using the chain rule, the following equation is defined:

$$\frac{dD}{d\boldsymbol{\varepsilon}} = \left(\frac{d\boldsymbol{\varepsilon}^e}{d\boldsymbol{\varepsilon}}\right)^T \left(\frac{dg}{dr_t}\frac{dr_t}{d\boldsymbol{\varepsilon}^e} + \frac{dg}{dr_c}\frac{dr_c}{d\boldsymbol{\varepsilon}^e}\right) \tag{8.43}$$

In a general context, the presented model can be coupled with several inelastic models. Hence, the elastic and total strain tensors are different. Thus, the following derivative is valid for inelastic models:

$$\frac{d\boldsymbol{\varepsilon}^e}{d\boldsymbol{\varepsilon}} = \frac{d\boldsymbol{\varepsilon}^e}{d\overline{\boldsymbol{\sigma}}}\frac{d\overline{\boldsymbol{\sigma}}}{d\boldsymbol{\varepsilon}} = \boldsymbol{C}\frac{d\overline{\boldsymbol{\sigma}}}{d\boldsymbol{\varepsilon}} \tag{8.44}$$

When linear elasticity is adopted, Equation (8.44) becomes

$$\frac{d\boldsymbol{\varepsilon}^e}{d\boldsymbol{\varepsilon}} = \boldsymbol{I} \tag{8.45}$$

To exploit the condition in (8.10) when $\dot{r}_\aleph \neq 0$, Equation (8.5) is differentiated with respect to time. Using the chain rule, the following differential equation is defined:

$$\dot{\varphi}_\aleph = \frac{\partial \varphi_\aleph}{\partial \boldsymbol{\varepsilon}^e}\dot{\boldsymbol{\varepsilon}}^e + \frac{\partial \varphi_\aleph}{\partial r_\aleph}\dot{r}_\aleph = 0 \tag{8.46}$$

which results in

$$\dot{r}_{\aleph} = -\frac{\frac{\partial \varphi_{\aleph}}{d\boldsymbol{\varepsilon}^{e}}}{\frac{\partial \varphi_{\aleph}}{dr_{\aleph}}}\dot{\boldsymbol{\varepsilon}}^{e} \tag{8.47}$$

In this scenario, Equation (8.47) defines the following expression:

$$\frac{dr_{\aleph}}{d\boldsymbol{\varepsilon}^{e}} = -\frac{\frac{\partial \varphi_{\aleph}}{\partial \boldsymbol{\varepsilon}^{e}}}{\frac{\partial \varphi_{\aleph}}{\partial r_{\aleph}}} \tag{8.48}$$

In the case of $\dot{r}_{\aleph} = 0$, the internal variable $r_{\aleph}$ does not evolve. Hence, Equation (8.49) is used instead of Equation (8.48):

$$\frac{dr_{\aleph}}{d\boldsymbol{\varepsilon}^{e}} = \mathbf{0} \tag{8.49}$$

8.5. Plane stress considerations

When considering linear elasticity, the effective stress tensor $\overline{\boldsymbol{\sigma}}$ can be computed using Equation (8.4) and the elastic modulus $\boldsymbol{D}$ in Equation (5.46). Thus, Equation (8.42) is invalid for plane stress problems. Using Equations (5.44) and (5.45), the effect of the out-of-plane strain is correctly accounted for in the computation of the effective stresses and, consequently, in the equivalent strains $\varepsilon_{\aleph}^{eq}$. Regarding the material tangent modulus, Equation (8.58) requires some attention during its computation to ensure the plane stress condition. Following the strategy described by de Souza Neto, Perić and Owen (2008) for plasticity applications, $\boldsymbol{D}_{11}$, $\boldsymbol{D}_{12}$, $\boldsymbol{D}_{21}$, and D_{22} are defined according to Equations (8.50) and (8.51).

$$\begin{bmatrix} \boldsymbol{D}_{11} & \boldsymbol{D}_{12} \\ \boldsymbol{D}_{21} & D_{22} \end{bmatrix} = (1-D)\frac{d\overline{\boldsymbol{\sigma}}}{d\boldsymbol{\varepsilon}} - \overline{\boldsymbol{\sigma}} \otimes \frac{dD}{d\boldsymbol{\varepsilon}} \tag{8.50}$$

$$\begin{bmatrix} d\sigma_{xx} \\ d\sigma_{yy} \\ d\sigma_{xy} \\ d\sigma_{zz} \end{bmatrix} = \begin{bmatrix} d\boldsymbol{\sigma} \\ d\sigma_{zz} \end{bmatrix} = \begin{bmatrix} \boldsymbol{D}_{11} & \boldsymbol{D}_{12} \\ \boldsymbol{D}_{21} & D_{22} \end{bmatrix} \begin{bmatrix} d\boldsymbol{\varepsilon} \\ d\varepsilon_{zz} \end{bmatrix} = \begin{bmatrix} \boldsymbol{D}_{11} & \boldsymbol{D}_{12} \\ \boldsymbol{D}_{21} & D_{22} \end{bmatrix} \begin{bmatrix} d\varepsilon_{xx} \\ d\varepsilon_{yy} \\ d\varepsilon_{xy} \\ d\varepsilon_{zz} \end{bmatrix} \tag{8.51}$$

Imposing $d\sigma_{zz} = 0$, $d\boldsymbol{\sigma}$ and $d\varepsilon_{zz}$ are computed according to

$$d\boldsymbol{\sigma} = \boldsymbol{D}_{11} d\boldsymbol{\varepsilon} + \boldsymbol{D}_{12} d\varepsilon_{zz} \tag{8.52}$$

$$d\varepsilon_{zz} = -\frac{1}{D_{22}} \boldsymbol{D}_{21} d\boldsymbol{\varepsilon} \tag{8.53}$$

Substituting (8.53) in (8.52), the material tangent modulus that agrees with the plane stress state is obtained, as presented in Equation (8.54). Note that this equation is closely related to (5.55) as they are computed according to the same definition (out-of-plane stress equals zero).

$$\frac{d\boldsymbol{\sigma}}{d\boldsymbol{\varepsilon}} = \boldsymbol{D}_{11} - \frac{1}{D_{22}} \boldsymbol{D}_{12} \boldsymbol{D}_{21} \tag{8.54}$$

8.6. Coupling elastoplasticity and damage

Numerous authors have explored different strategies for combining elastoplastic and damage models (Lemaitre, 1985; Lubliner et al., 1989; Faria, Oliver and Cervera, 1998; Jason et al., 2006; Feng, Ren and Li, 2018). This work adopted a rational and physically meaningful approach, which considers that elastoplasticity occurs exclusively in the intact material and not within microcracks or microvoids (de Borst et al., 2012). Hence, the effective stress tensor $\overline{\boldsymbol{\sigma}}$ is computed according to the plasticity theory. Considering multisurface plasticity, the plastically admissible states for the C3EPD model are defined as follows:

$$\overline{\mathbb{E}} = \left\{ (\overline{\boldsymbol{\sigma}}, \boldsymbol{q}) \;\middle|\; f_i(\overline{\boldsymbol{\sigma}}, \boldsymbol{q}) \leq 0, \;\; \forall i \in \{1,2,\dots,N\} \right\} \tag{8.55}$$

Moreover, the flow rule and the hardening law are redefined to be computed according to the effective stress tensor. In the end, the system of nonlinear equations to be solved in order to update the stress states takes the form

$$\begin{cases} \boldsymbol{C}(\overline{\boldsymbol{\sigma}}_{n+1} - \overline{\boldsymbol{\sigma}}_n) + \sum_{i=1}^{N} \Delta\gamma_{i,n+1} \boldsymbol{n}_i(\overline{\boldsymbol{\sigma}}_{n+1}, \boldsymbol{q}_{n+1}) = \Delta\boldsymbol{\varepsilon}_{n+1} \\ \boldsymbol{q}_{n+1} = \boldsymbol{q}_n + \sum_{i=1}^{N} \Delta\gamma_{i,n+1} \boldsymbol{h}_i(\overline{\boldsymbol{\sigma}}_{n+1}, \boldsymbol{q}_{n+1}) \\ \phi_{i,n+1} = 0, \;\; i \in \{1,2,\dots,N\} \end{cases} \tag{8.56}$$

The C3EPD model combines the damage constitutive equations described in Sections 8.2 and 8.3 with the elastoplastic formulation defined in Section 4.1. In this scenario, a typical integration algorithm for plasticity can be utilized in this coupling process, which starts with applying such algorithm in order to compute the effective stresses $\overline{\boldsymbol{\sigma}}$. Once the stresses $\overline{\boldsymbol{\sigma}}$ are determined, the equivalent strains $\varepsilon_{\aleph}^{eq}$ and damage variable D can be computed. Importantly, the damage variable does not evolve due to plastic strains (Jason et al., 2006). Therefore, according to Simo and Ju (1987b) and Jason et al. (2006), the state-update process of a coupled elastoplastic-damage model can be performed in two main steps: elastic predictor-plastic corrector and damage corrector. Other authors, such as Ju (1989a, 1989b), Wu, Li and Faria (2006), Häussler-Combe and Harting (2008), and Zhang and Li (2016) used the same strategy to couple elastoplasticity and damage.

Notably, the tensor $\frac{d\overline{\boldsymbol{\sigma}}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}}$ in Equations (8.41) and (8.50), represents the material tangent modulus for the adopted integration scheme. In this work, Algorithm 5 is adopted to compute the effective stresses and the other plastic variables. The consistent tangent modulus for elastoplasticity can be calculated using Equations (5.40) and (5.41). In the plane stress case, one could use Equation (5.55) rather than Equation (5.41). However, Equation (8.54) performs the same modification as Equation (5.55). In this work, Equations (5.40) and (5.41) are used to compute the consistent tangent modulus for elastoplasticity, also for the plane stress state, while Equation (8.54) yields the final material tangent tensor.

8.7. Computational aspects

The scheme, which also takes into account the plane stress condition, to update the states for the C3EPD model is summarized in Algorithm 7. Additional equations required to implement the model are presented in Appendix D. The proposed procedure incorporates certain computational aspects described in this section. These aspects were included to ensure a well-behaved algorithm, primarily in computing a material tangent modulus that accurately represents the derivative of stresses with respect to total strains. To evaluate the proposed procedure, finite difference schemes were adopted to verify the validity of the tangent modulus.

Attention is required to compute the principal stresses in Equations (8.11) and (8.12) due to accuracy issues related to the analytical computation of eigenvalues, presented in Appendix D. These expressions can exhibit numerical instability when two or three eigenvalues are nearly identical (Scherzinger and Dohrmann, 2008). In this scenario, several experiments were conducted based on the proposed damage model, including uniaxial, biaxial, and hydrostatic tests under tension and compression. Inconsistencies were observed in these tests when the invariants $\bar{J}_2$ and $\bar{J}_3$ approached zero. Hence, the algorithm was specifically modified to set theses invariants to zero in such situations. The tolerances $tol_1 = tol_2 = 10^{-10}$ are adopted to determine the proximity to zero of these invariants.

Moreover, the argument of the inverse sine function present in the analytic computation of the principal stresses may slightly exceed the domain of this function, deviating from the limit values of 1 and -1 due to floating-point operations. To address this, the proposed algorithm sets the limit values of the Lode angle $\bar{\theta}$ when such situation arises. These considerations are presented in Algorithm 8. It should be noted that in step 2 of this algorithm, a division by zero is possible ($\bar{\sigma}_{max}\ /\bar{t}_{max}$ and $\bar{\sigma}_{min}\ /\bar{t}_{min}$). These verifications are performed to check if any principal stress is approximately zero, based on the tolerance $tol_3 = 10^{-6}$. As is the case in many programming languages, assuming that this division results in ∞, no theoretical problems will arise when using this algorithm. Moreover, when computing the Heaviside function in Equations (8.11) and (8.12), it is adopted:

$$H(x) = \text{sign}\big(\max(x,0)\big) \tag{8.57}$$

Considering the matrix representation and the Voigt notation, the material tangent modulus in Equation (8.41) takes the following form:

$$\frac{d\boldsymbol{\sigma}}{d\boldsymbol{\varepsilon}} = (1-D)\frac{d\overline{\boldsymbol{\sigma}}}{d\boldsymbol{\varepsilon}} - \overline{\boldsymbol{\sigma}}\left(\frac{dD}{d\boldsymbol{\varepsilon}}\right)^T \tag{8.58}$$

Algorithm 7 State-update algorithm for the proposed coupled elastoplastic-damage model

Initialize: $\Delta\boldsymbol{\varepsilon}_{n+1}$, $\boldsymbol{\varepsilon}_n^p$, $\boldsymbol{\sigma}_n$, $\boldsymbol{q}_n$, D_n, $r_{t,n}$, $r_{c,n}$, $\boldsymbol{D}$, $\boldsymbol{C}$, $f_i(\boldsymbol{\sigma},\boldsymbol{q})$, $\boldsymbol{n}_i(\boldsymbol{\sigma},\boldsymbol{q})$, $\boldsymbol{h}_i(\boldsymbol{\sigma},\boldsymbol{q})$, f_{cm}, f_{t0}, f_{c0}, G_f, G_c, r_{t0}, r_{c0}, r_{ci}, r_{cm}, l_{ch}

Procedure:

1: Compute effective stresses $\overline{\boldsymbol{\sigma}}_{n+1}$:

In case of plane stress, the stress and strain tensors are defined with four components

$\overline{\boldsymbol{\sigma}}_n = \dfrac{\boldsymbol{\sigma}_n}{1-D_n}$

If $D_n = 0$:

Return $\boldsymbol{\sigma}_{n+1} = \boldsymbol{0}$, $\boldsymbol{q}_{n+1} = \boldsymbol{q}_n$, $\boldsymbol{\varepsilon}_{n+1}^p = \boldsymbol{\varepsilon}_n^p$, $D_{n+1} = D_n$, $r_{t,n+1} = r_{t,n}$, $r_{c,n+1} = r_{c,n}$, and $\frac{d\boldsymbol{\sigma}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} = \boldsymbol{0}$

End

Perform the Algorithm 1, Algorithm 5, or Algorithm 6 to compute $\overline{\boldsymbol{\sigma}}_{n+1}$, $\boldsymbol{q}_{n+1}$, $\boldsymbol{\varepsilon}_{n+1}^p$, and $\frac{d\overline{\boldsymbol{\sigma}}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} = \boldsymbol{D}_{n+1}^{ep}$ based on $\Delta\boldsymbol{\varepsilon}_{n+1}$, $\overline{\boldsymbol{\sigma}}_n$, $\boldsymbol{q}_n$, $\boldsymbol{\varepsilon}_n^p$ and the constitutive equations $f_i(\boldsymbol{\sigma},\boldsymbol{q})$, $\boldsymbol{n}_i(\boldsymbol{\sigma},\boldsymbol{q})$, and $\boldsymbol{h}_i(\boldsymbol{\sigma},\boldsymbol{q})$

2: Compute nominal stresses $\boldsymbol{\sigma}_{n+1}$:

Calculate $r_{t,n+1}$, $r_{c,n+1}$, $\frac{dr_{t,n+1}}{d\boldsymbol{\varepsilon}^e}$, and $\frac{dr_{c,n+1}}{d\boldsymbol{\varepsilon}^e}$ according to Algorithm 8

Calculate a_t, a_c, b_c, t_c, g_{c1}, and d_c using Equations (8.25), (8.32), (8.33), (8.31), (8.36), and (8.38). The proposed state-update scheme is valid for $g_{c1}l_{ch} < G_c$

Calculate the scalar damage variable D_{n+1} based on Equations (8.8), (8.19), (8.21), and (8.34)

Calculate $\frac{dg_{n+1}}{\partial r_t}$ and $\frac{dg_{n+1}}{\partial r_c}$ using Equations (D.16) to (D.20)

$\boldsymbol{\sigma}_{n+1} = (1-D_{n+1})\overline{\boldsymbol{\sigma}}_{n+1}$

3: Compute the material tangent modulus $\frac{d\boldsymbol{\sigma}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}}$:

Calculate $\frac{d\boldsymbol{\varepsilon}_{n+1}^e}{d\boldsymbol{\varepsilon}}$ using Equation (8.44)

Initialize the material tangent modulus using $\frac{d\boldsymbol{\sigma}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} = (1-D_{n+1})\frac{d\overline{\boldsymbol{\sigma}}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}}$ (secant modulus)

If $r_{t,n+1} > r_{t,n}$:

$\frac{d\boldsymbol{\sigma}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} \coloneqq \frac{d\boldsymbol{\sigma}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} - \frac{dg_{n+1}}{dr_t}\overline{\boldsymbol{\sigma}}_{n+1}\left(\frac{dr_{t,n+1}}{d\boldsymbol{\varepsilon}^e}\right)^T\frac{d\boldsymbol{\varepsilon}_{n+1}^e}{d\boldsymbol{\varepsilon}}$

End

If $r_{c,n+1} > r_{c,n}$:

$\frac{d\boldsymbol{\sigma}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} \coloneqq \frac{d\boldsymbol{\sigma}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} - \frac{dg_{n+1}}{dr_c}\overline{\boldsymbol{\sigma}}_{n+1}\left(\frac{dr_{c,n+1}}{d\boldsymbol{\varepsilon}^e}\right)^T\frac{d\boldsymbol{\varepsilon}_{n+1}^e}{d\boldsymbol{\varepsilon}}$

End

In case of plane stress, modify the material tangent modulus according to Equation (8.54)

Return $\boldsymbol{\sigma}_{n+1}$, $\boldsymbol{q}_{n+1}$, $\boldsymbol{\varepsilon}_{n+1}^p$, D_{n+1}, $r_{t,n+1}$, $r_{c,n+1}$, and $\frac{d\boldsymbol{\sigma}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}}$

Algorithm 8 Update the internal variables r_t and r_c based on the proposed damage criteria

Initialize: $\bar{\boldsymbol{\sigma}}_{n+1}$, $r_{t,n}$, $r_{c,n}$, $\boldsymbol{D}$, E, α_t, β_t, γ_t, α_c, β_c, γ_c, tol_1, tol_2, tol_3

Procedure:

1: Compute invariants $\bar{I}_1$, $\bar{J}_2$, and $\bar{J}_3$ and invariant gradients according to $\bar{\boldsymbol{\sigma}}_{n+1}$:

If $|\bar{J}_2/(\bar{I}_1)^2| < tol_1$ or $|\bar{J}_2| + |\bar{J}_3| = 0$:

$\bar{J}_2 = \bar{t}_{max} = \bar{t}_{min} = 0$

Elseif $|\bar{J}_3/(\bar{I}_1)^3| < tol_2$:

$\bar{t}_{max} = \sqrt{\bar{J}_2}$, $\bar{t}_{min} = -\bar{t}_{max}$, and $\frac{\partial \bar{\theta}}{\partial \bar{\boldsymbol{\sigma}}} = \mathbf{0}$

Else:

$\omega = -\frac{3\sqrt{3}\bar{J}_3}{2\bar{J}_2^{3/2}}$

If $\omega \geq 1$:

$\bar{t}_{max} = \frac{\sqrt{3\bar{J}_2}}{3}$, $\bar{t}_{min} = -\frac{2\sqrt{3\bar{J}_2}}{3}$, and $\frac{\partial \bar{\theta}}{\partial \bar{\boldsymbol{\sigma}}} = \mathbf{0}$

Elseif $\omega \leq -1$:

$\bar{t}_{max} = \frac{2\sqrt{3\bar{J}_2}}{3}$, $\bar{t}_{min} = -\frac{\sqrt{3\bar{J}_2}}{3}$, and $\frac{\partial \bar{\theta}}{\partial \bar{\boldsymbol{\sigma}}} = \mathbf{0}$

Else:

$\bar{\theta} = \frac{1}{3}\arcsin(\omega)$, $\frac{\partial \bar{J}_2}{\partial \bar{\boldsymbol{\sigma}}} = -\frac{tan3\bar{\theta}}{2\bar{J}_2}$, and $\frac{\partial \bar{J}_3}{\partial \bar{\boldsymbol{\sigma}}} = \frac{tan3\bar{\theta}}{3\bar{J}_3}$

Compute $\bar{t}_{max}$, $\bar{t}_{min}$ and $\frac{\partial \bar{\theta}}{\partial \bar{\boldsymbol{\sigma}}}$ according to Equations (D.4), (D.5) and (D.26)

End

End

2: Compute maximum and minimum principal stresses:

Calculate $\bar{p}$, $\bar{\sigma}_{max}$ and $\bar{\sigma}_{min}$ using Equations (D.3), (D.1) and (D.2)

If $|\bar{\sigma}_{max}\ /\bar{t}_{max}| < tol_3$:

$\bar{\sigma}_{max} = 0$

End

If $|\bar{\sigma}_{min}\ /\bar{t}_{min}| < tol_3$:

$\bar{\sigma}_{min} = 0$

End

3: Compute equivalent strains:

Calculate $\varepsilon_{t,n+1}^{eq}$ and $\varepsilon_{c,n+1}^{eq}$ using Equations (8.11) and (8.12)

$r_{t,n+1} = \max\left(r_{t,n}, \varepsilon_{t,n+1}^{eq}\right)$ and $r_{c,n+1} = \max\left(r_{c,n}, \varepsilon_{c,n+1}^{eq}\right)$

4: Compute derivatives $\frac{\partial r_t}{\partial \boldsymbol{\varepsilon}^e}$ and $\frac{\partial r_c}{\partial \boldsymbol{\varepsilon}^e}$:

If $|\bar{J}_2/(\bar{I}_1)^2| < tol_1$ or $|\bar{J}_2| + |\bar{J}_3| = 0$:

$\frac{\partial \varepsilon_t^{eq}}{\partial \bar{\boldsymbol{\sigma}}} = \frac{3\alpha_t + \gamma_t H(\bar{\sigma}_{max})}{3E}\frac{\partial \bar{I}_1}{\partial \bar{\boldsymbol{\sigma}}}$, $\frac{\partial \varepsilon_c^{eq}}{\partial \bar{\boldsymbol{\sigma}}} = \frac{3\alpha_c + \gamma_c H(-\bar{\sigma}_{max})}{3(1-\alpha_c)E}\frac{\partial \bar{I}_1}{\partial \bar{\boldsymbol{\sigma}}}$

Compute $\frac{\partial \varphi_t}{\partial \boldsymbol{\varepsilon}^e}$ and $\frac{\partial \varphi_c}{\partial \boldsymbol{\varepsilon}^e}$ using Equation (D.21)

Else:

Compute $\frac{\partial \bar{\sigma}_{max}}{\partial \bar{\boldsymbol{\sigma}}}$, $\frac{\partial \varphi_t}{\partial \boldsymbol{\varepsilon}^e}$ and $\frac{\partial \varphi_c}{\partial \boldsymbol{\varepsilon}^e}$ using Equations (D.25), (D.21), (D.23), and (D.24)

End

Return $r_{t,n+1}$, $r_{c,n+1}$, $\frac{\partial r_t}{\partial \boldsymbol{\varepsilon}^e} = \frac{\partial \varphi_t}{\partial \boldsymbol{\varepsilon}^e}$, and $\frac{\partial r_c}{\partial \boldsymbol{\varepsilon}^e} = \frac{\partial \varphi_c}{\partial \boldsymbol{\varepsilon}^e}$

8.8. Application of the CCD and C3EPD models

The proposed constitutive formulations are here confronted with experimental data to evaluate its capability of representation of concrete structures. These evaluations are conducted using the finite element method within the in-house framework GeMA (Mendes, Gattass and Roehl, 2016). Then, the CCD and C3EPD model were implemented in this framework based on Algorithm 7. For the implicit integration of multisurface plasticity, Algorithm 5 was applied. Importantly, the implementation adopted the golden section method as line search strategy, as detailed in Algorithm 3. All finite element models consider plane stress conditions and bilinear quadrilateral elements with four integration points. Furthermore, the cylindrical arc-length method (Crisfield, 1981) was utilized to obtain the global solution of the problem. The tolerance of Newton-Raphson global process is assumed 10^{-6}.

Unless otherwise specified, the following assumptions were made regarding the material parameters: Poisson's ratio is $\nu = 0.20$ (fib, 2013); the uniaxial compressive stress at which the damage process is initialized $f_{c0} = 0.4 f_{cm}$ (Alfarah, López-Almansa and Oller, 2017); and $r_{cm} = 0.002$, which was considered here a representative value according to the Model Code recommendations (fib, 2013). Moreover, the biaxial tensile and compressive strength of the material are respectively $f_{bt} = 1.05 f_{t0}$ (Jirásek and Bauer, 2012) and $f_{bc} = 1.16 f_{cm}$ (Kupfer, Hilsdorf and Rüsch, 1969). When the crunching energy G_c is not available, it is computed using the following expression recommended by Alfarah, López-Almansa and Oller (2017):

$$G_c = G_f \left(\frac{f_{cm}}{f_{tm}} \right)^2 \tag{8.59}$$

The applications discussed in Sections 8.8.3 and 8.8.4 focus on numerically simulating concrete structures using the C3EPD model. The parameters related to the damage formulation can be defined by basic experimental information (such as the compressive strength), empirical relations available in the literature and design guidelines for concrete structures. However, it is necessary to estimate some plastic parameters to match the numerical and experimental data. Therefore, the

methodology described in Appendix E is applied to solve the parameter identification problems under a feasible computational cost. This methodology is completely based on artificial intelligence methods, aiming to obtain accurate and efficient solutions.

In the parameter identification context, a gradient boosting machine is used to approximate the outcomes of the finite element simulations while a genetic algorithm minimizes the error between the numerical and experimental results. A Bayesian optimization algorithm is applied to compute the hyperparameters of the gradient boosting machine. To acquire data for the machine learning modeling process, various numerical models were simulated to generate load-displacement curves. Several of these simulations encountered convergence problems, resulting in their exclusion from the dataset. Despite that, the amount of data generated in this process was sufficient to produce accurate machine learning models.

8.8.1. Evaluation of the proposed damage criteria and evolution law

Before evaluating the coupled elastoplastic-damage model (C3EPD model), this section focuses solely on assessing the isotropic damage formulation presented in this work, the CCD model. Therefore, the effective stress tensor is computed according to Equation (8.4). In Section 8.3, the biaxial strength envelope of conventional concrete for the CCD model was validated with experimental data. The current study considers experimental stress-strain curves under biaxial stresses, considering different stress ratios σ_2/σ_1 from a series of experimental tests performed by Kupfer, Hilsdorf and Rüsch (1969).

In order to numerically simulate these biaxial tests, the adopted parameters are as follows: $E = 32$ GPa, $f_{t0} = 2.9$ MPa, $f_{c0} = 0.33 f_{cm}$, $f_{cm} = 32$ MPa, $f_{bt} = 0.95 f_{t0}$, $r_{cm} = 0.0022$, $G_f = 0.1$ kN/m, $G_c = 23$ kN/m, and $l_{ch} = 0.2$ m. Therefore, Figure 32 presents the comparison between numerical and experimental stress-strain responses under biaxial compression. Additionally, Figure 33 shows the comparison for the case of combined compression and tension. The predicted responses agree well with the experimental results, presenting satisfactory peak stresses under biaxial stresses. Importantly, a proper value for f_{tc} was fundamental

to reach the experimental peak stresses in Figure 33. Adjusting the tensile criterion shape using $f_{tc} = 5f_{cm}$ is here considered a good approach to match the numerical and experimental results.

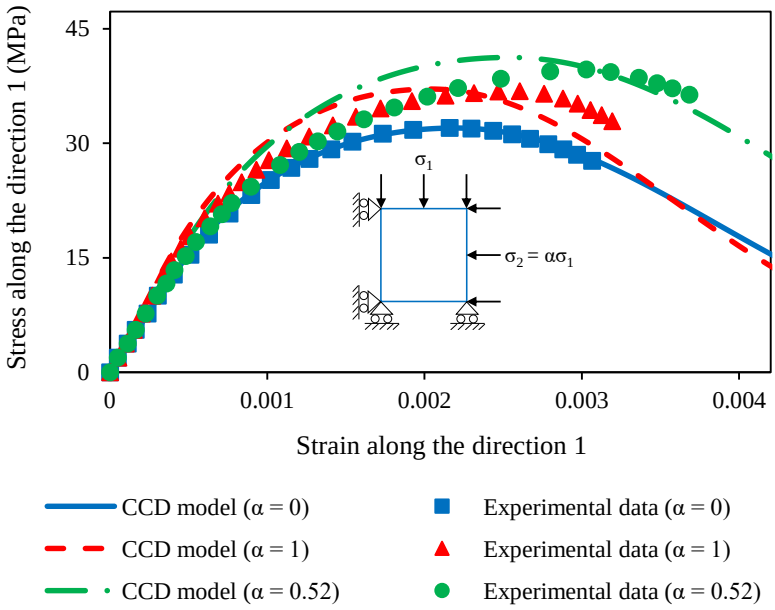

Figure 32 Stress-strain relation of conventional concrete under biaxial compression

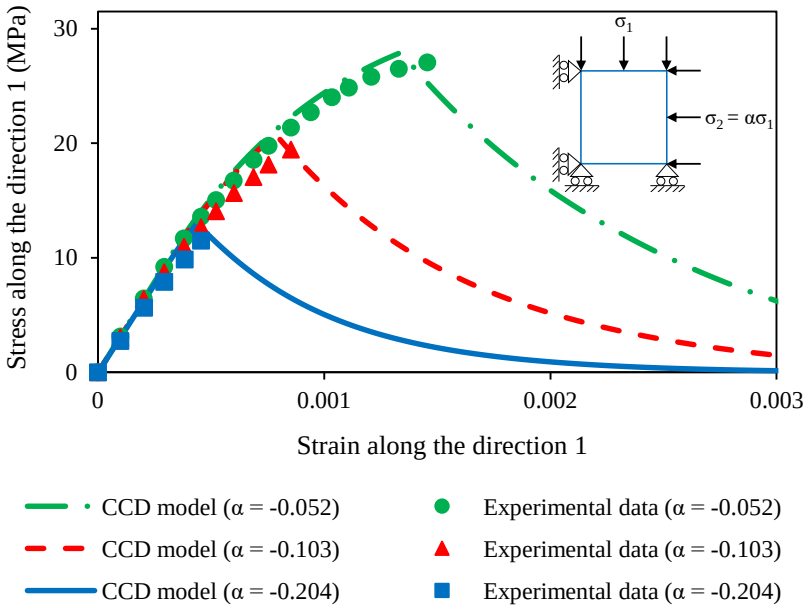

Figure 33 Stress-strain relation of conventional concrete under combined compression and tension

8.8.2. Hydrostatic and confined compression tests

The proposed coupled elastoplastic-damage model exhibits hardening behavior due to plasticity under specific conditions. This section aims to evaluate

this phenomenon through two types of experiments: hydrostatic and confined tests. The hydrostatic compression test is based on the numerical application developed by Hofstetter, Simo and Taylor (1993). The numerical confined compression test is conducted here using the same parameters to assess the influence of the damage formulation on the results.

Based on the works of Hofstetter, Simo and Taylor (1993) and Li and Crouch (2010), the subsequent simulations adopt the following parameters related to the damage formulation: $E = 27.691$ GPa, $f_{t0} = 2.5$ MPa, $f_{cm} = 10 f_{t0}$, $f_{bc} = 1.1 f_{cm}$, and $l_{ch} = 0.1$ m. The shape parameters of the plastic envelope are based on the Mohr-Coulomb criterion, as described in Table 1. The cohesion and friction angle are taken as $c = 250$ MPa and $\phi = 15°$, respectively. Moreover, the friction and dilation angles are assumed to be equal. Lastly, the cap surface with hardening behavior is defined according to the following parameters: $R = 0.35$, $\kappa_0 = -5$ MPa, $W = 0.42$, and $D = 0.000464$ MPa^{-1}.

Figure 34 illustrates the hydrostatic compression test results, comparing the numerical outcomes derived from the C3EPD model and the numerical results presented by Hofstetter, Simo and Taylor (1993) for the Cap Model. The solution for the C3EPD model exhibits a strong concordance with the reference solution, adequately capturing the expected hardening behavior of the cap surface. It is important to highlight that no damage evolution occurs in this application, as $\varepsilon_c^{eq} < 0$ according to Equation (8.12).

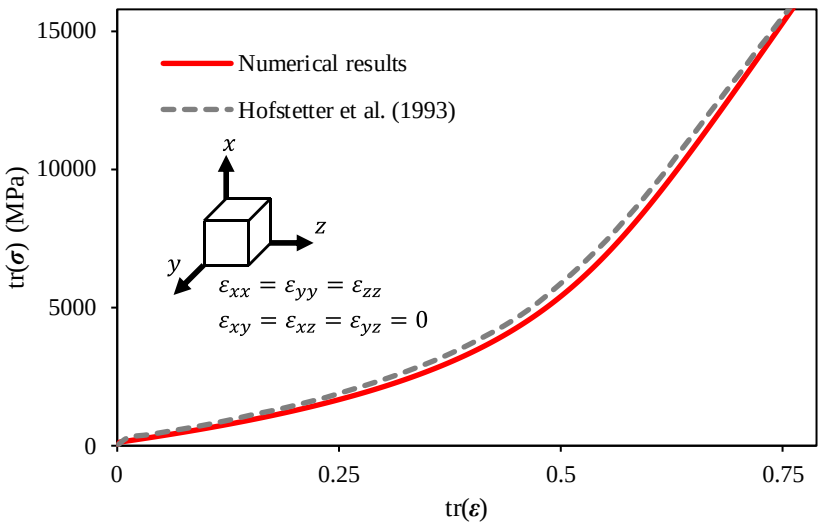

Figure 34 Hydrostatic behavior for numerical experiments using the C3EPD model (compression is assumed to be positive)

Figure 35 shows the numerical results for the confined compression test (oedometric test), employing both the CHGC and C3EPD models. This examination assesses the influence of the damage formulation within the framework of multisurface plasticity. Note that the plasticity results exhibit increasing stress throughout the test owing to cap hardening. Nonetheless, when considering the damage process, this increase is interrupted by stiffness degradation, leading to specimen collapse. Therefore, this application demonstrates the implications of coupling plasticity and damage for modeling engineering materials.

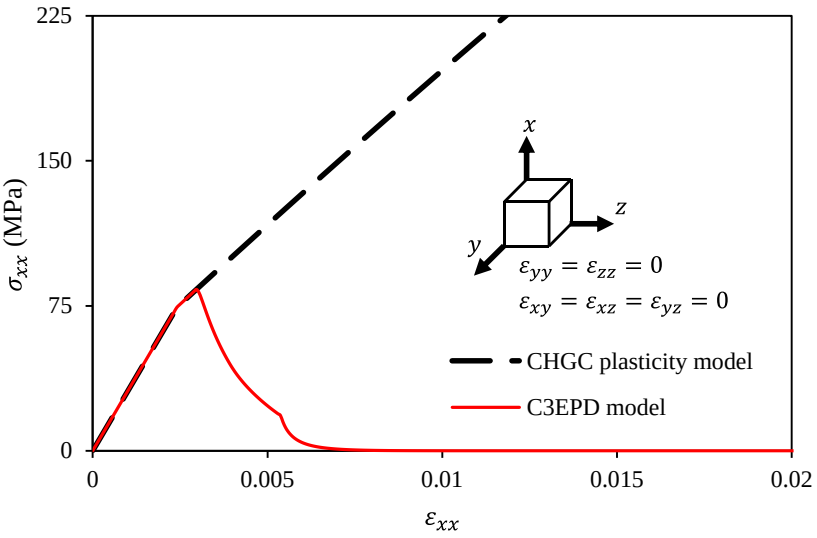

Figure 35 Confined numerical tests for the CHGC and C3EPD models (compression is assumed to be positive)

8.8.3. L-shaped panel

Winkler, Hofstetter and Niederwanger (2001) and Winkler, Hofstetter and Lehar (2004) presented the experimental program of an L-shaped concrete panel in order to assess elastoplastic models. This structure has become a well-known benchmark to validate the computational modeling of plain concrete under mode I fracture conditions. The boundary conditions and geometry of the problem are depicted in Figure 36. In addition, the thickness of the L-shaped panel is equal to 10 cm. Displacements are measured at the point load.

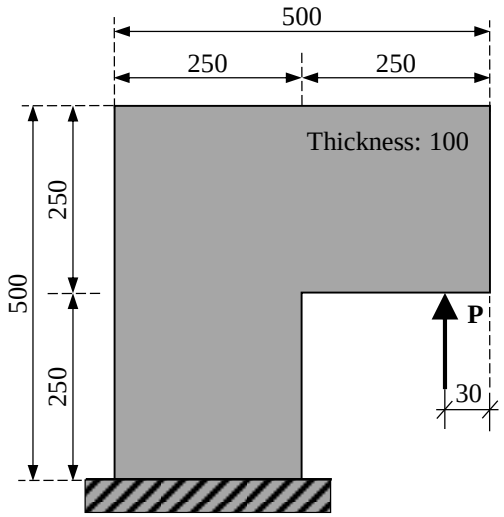

Figure 36 L-shaped panel: boundary conditions and geometry with all lengths in millimeters

For the present simulations, the following material parameters are adopted: $E = 25.85$ GPa, $\nu = 0.18$, $f_{t0} = 2.7$ MPa, $f_{cm} = 31$ MPa, $G_f = 0.065$ kN/m, and $G_c = 10.8$ kN/m. The assumed value for the fracture energy agrees with the works of Winkler, Hofstetter and Niederwanger (2001) and Winkler, Hofstetter and Lehar (2004). However, it differs from other predictions, such as Oliver et al. (2004), who computed $G_f = 0.09$ kN/m, and the Model Code recommendations (fib, 2013), which suggest $G_f = 0.073(f_{cm})^{0.18} = 0.135$ kN/m. Nevertheless, in this case study, $G_f = 0.065$ kN/m was adopted as it provided satisfactorily results.

Regarding the plastic parameters, the shape parameters are adopted according to the Mohr-Coulomb criterion, as considered in the previous application. Hence, it becomes necessary to identify the following parameters related to this criterion: cohesion c and friction angle ϕ. For simplicity, the friction and dilation angles are assumed to be equal. Additionally, the cap surface is here defined by the following parameters: $R = 2$, $\kappa_0 = -10$ MPa, $W = 0.5$, and $D = 0.001$ MPa^{-1}. The parameter identification task is conducted approximating the outcome of the numerical simulator using a machine learning model, specifically a gradient boosting machine. Then, the difference between the numerical and experimental data is minimized using a genetic algorithm. Solving the parameter identification problem, c was assumed to be 210 MPa and ϕ to be 15 degrees. Figure 37a illustrates the following load-displacement responses for the adopted values of c and ϕ: the numerical simulator, the gradient boosting machine, and the experimental data. The predicted responses by both the numerical and machine

learning models align with the experimental data, culminating in a successful parameter identification task.

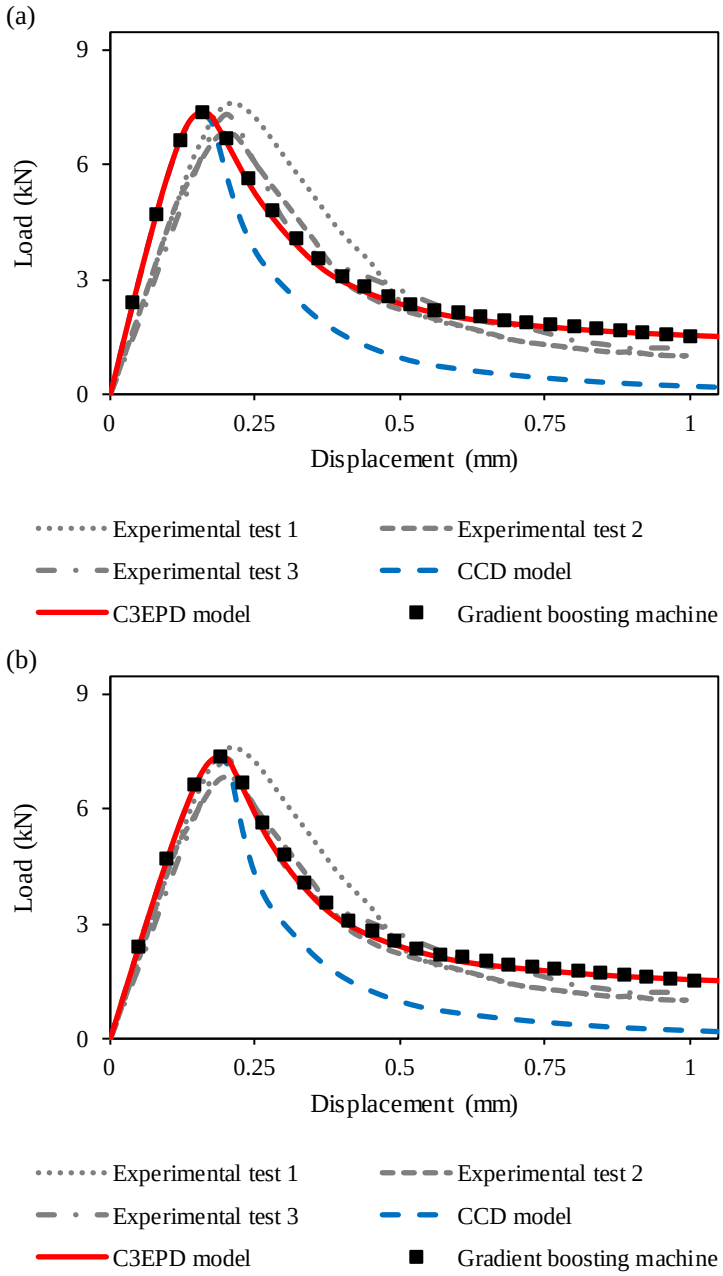

Figure 37 Load-displacement response for the L-shaped panel: (a) without and (b) with adjustment according to Kitzig and Häussler-Combe (2011)

In order to compare the quality of the results for the CCD and C3EPD models, both were utilized to compute the load-displacement response of the L-shaped panel. It is evident that both models successfully captured the initial branch of this response. Nevertheless, a notable difference is observed in the post-peak branch, where the CCD model exhibited a faster decrease than desired. This discrepancy is

effectively addressed by coupling elastoplasticity and damage, proving to be fundamental in accurately reproducing the descending branch of this load-displacement curve.

A correction procedure of the load-displacement response is applied in Figure 37b, according to Kitzig and Häussler-Combe (2011). As shown by these authors, a typical finite element modeling of this L-shaped panel tends to overestimate the initial branch of the mentioned response. A coupled elastoplastic-damage model cannot produce this additional stiffness since this model does not affect this branch. Therefore, the authors claimed that a rigid body rotation occurred in the panel due to some flexibility in the support. By using the proposed adjustment, a better match between the numerical and experimental peak load was achieved.

The damaged region resulting from the simulation based on the C3EPD model is depicted in Figure 38. As reported by Winkler, Hofstetter and Niederwanger (2001) and Winkler, Hofstetter and Lehar (2004), a fracture propagation with a slight inclination was observed. The proposed coupled elastoplastic-damage model satisfactorily captured this fracture shape.

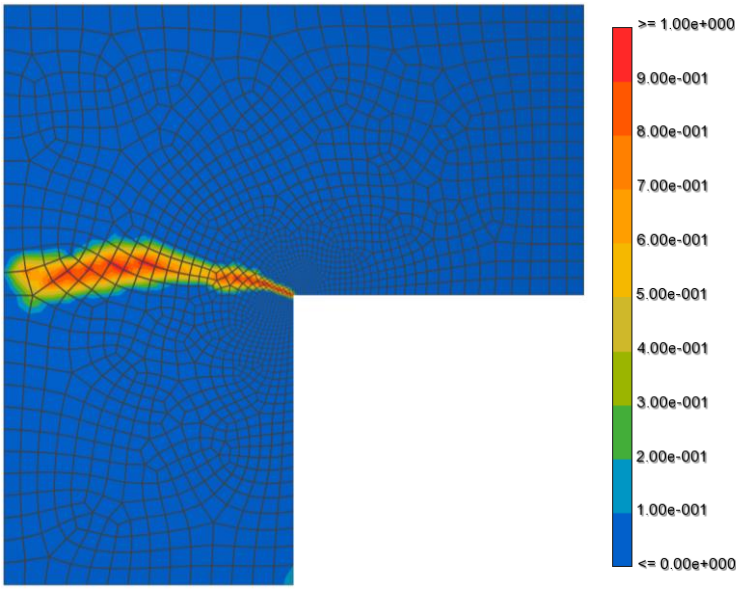

Figure 38 Spatial distribution of the scalar damage variable D for the L-shaped panel when the deflection reaches 1 mm at the point load

8.8.4. Four-point shear test

Mixed mode fracture propagation in structures is a critical aspect to assess the predictive capabilities of constitutive models for concrete. In this sense, the

experimental tests presented by Carpinteri *et al*. (1993) are simulated in this work. These authors developed an experimental mixed mode fracture program to study displacements in a plain concrete beam, detailing the Crack Mouth Sliding Displacement (CMSD) and the Crack Mouth Opening Displacement (CMOD). Figure 39 depicts the studied beam, in which the height and width of the beam are respectively 20 cm and 10 cm, while the size of the notch is $4 \times 0.5\ \text{cm}^2$. Concerning the load in the beam, force F2 corresponds to 10% of force F1. Displacements are measured at the respective point loads.

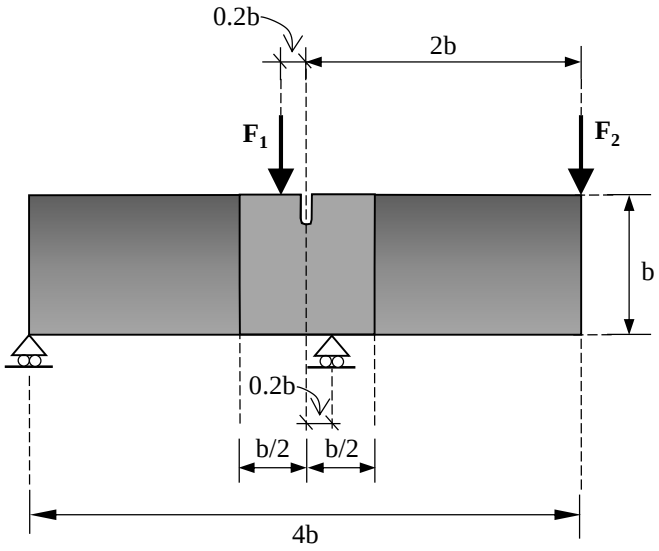

Figure 39 Four-point shear test: boundary conditions and geometry

The material parameters utilized in the numerical simulations are as follows: $E = 28$ GPa, $\nu = 0.1$, $f_{cm} = 10 f_{t0}$, $G_f = 0.145$ kN/m, $R = 2$, $\kappa_0 = -10$ MPa, $W = 0.5$, and $D = 0.001\ \text{MPa}^{-1}$. It should be highlighted that the assumed value for the fracture energy is similarly to the value reported in the Model Code recommendations (fib, 2013): $G_f = 0.138$ kN/m. As in the previous applications, the adopted shape parameters yield the Mohr-Coulomb criterion. Consequently, the parameters to be identified are: the uniaxial tensile stress f_{t0} at which the damage process is initialized; the cohesion c; and the friction angle ϕ. It is assumed that the friction and dilation angles are equal to each other. Based on the parameter identification procedure, the following parameters were estimated: $f_{t0} = 3.6$ MPa, $c = 270$ MPa and $\phi = 10°$. Notably, the computed uniaxial tensile strength reasonably agrees with other works in the literature (De-Pouplana and Oñate, 2016; Javanmardi and Maheri, 2019).

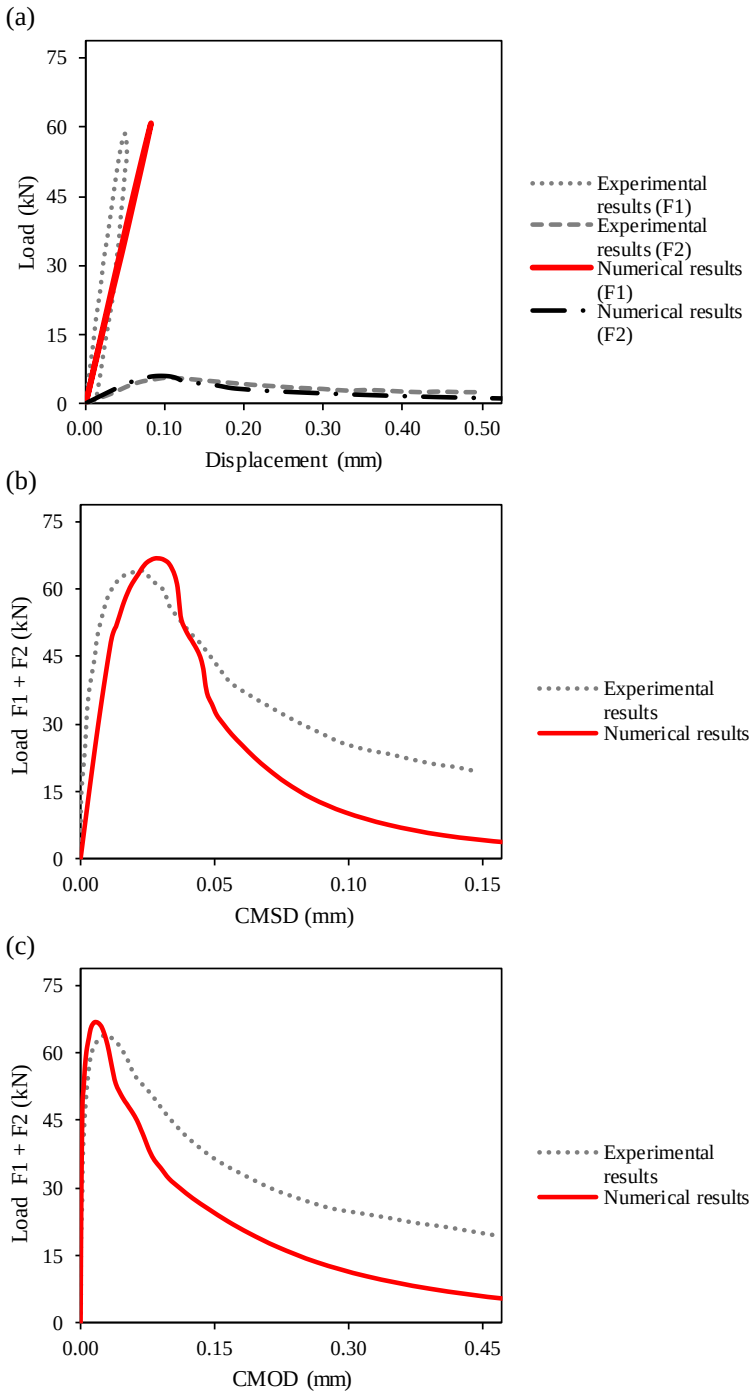

Figure 40 Load-displacement response for the four-point shear test considering the CCD model: (a) load-displacement, (b) load-CMSD, and (c) load-CMOD responses

To assess the impact of elastoplasticity in this modeling, the results for the CCD model are presented in Figure 40. Additionally, Figure 41 shows the response for the C3EPD model, which exhibits good agreement with the experimental data. Moreover, the machine learning prediction in Figure 41b confirms the accuracy of the gradient boosting machine since it coincides with the numerical data. When

comparing the CCD and C3EPD models, a significant difference is observed in the post-peak branch, highlighting the necessity of coupling elastoplasticity and damage.

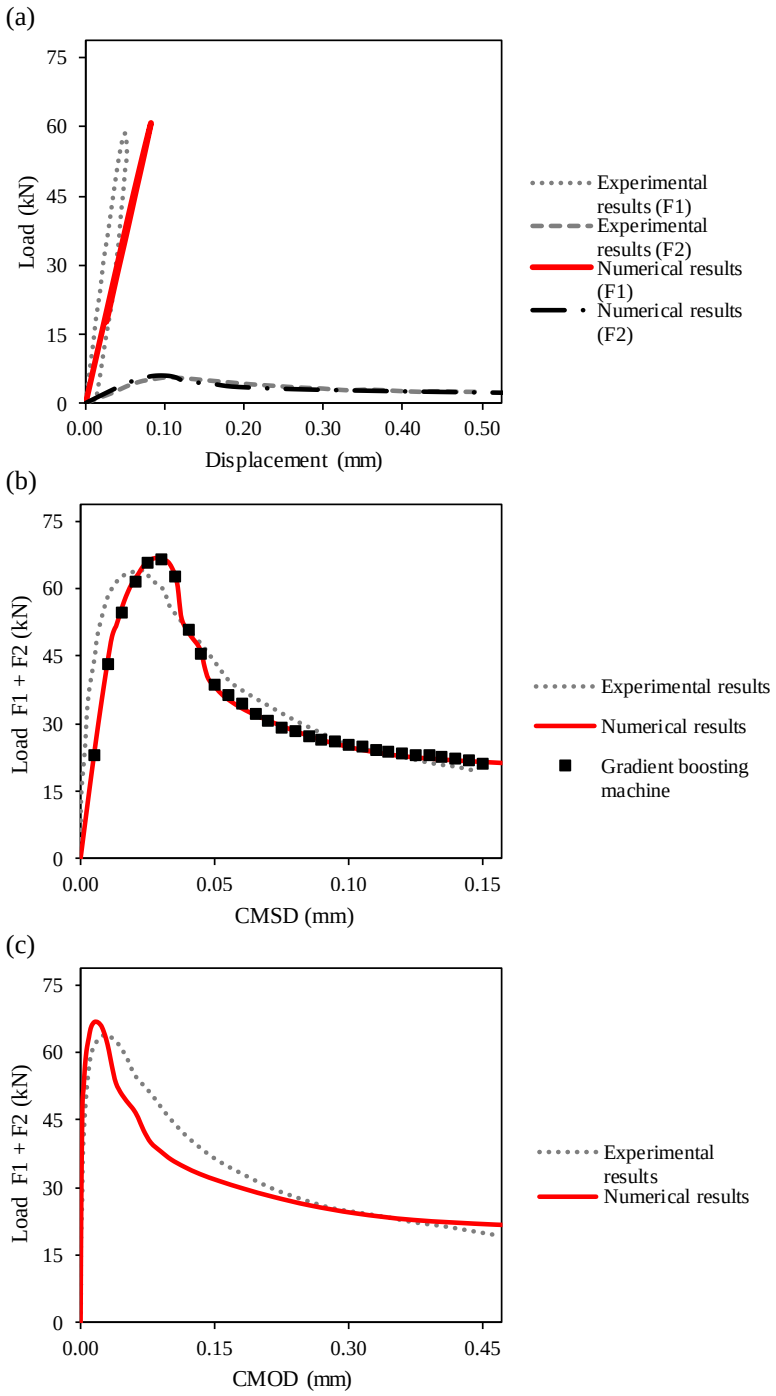

Figure 41 Load-displacement response for the four-point shear test considering the C3EPD model: (a) load-displacement, (b) load-CMSD, and (c) load-CMOD responses

Finally, the numerical fracture trajectory of the beam modeled using the C3EPD model is depicted in Figure 42. The numerical fracture trajectory is in

conformity with the experimental data provided by Carpinteri *et al*. (1993). In this scenario, the proposed coupled elastoplastic-damage model exhibits a worth capacity to define fracture shapes for conventional concrete structures.

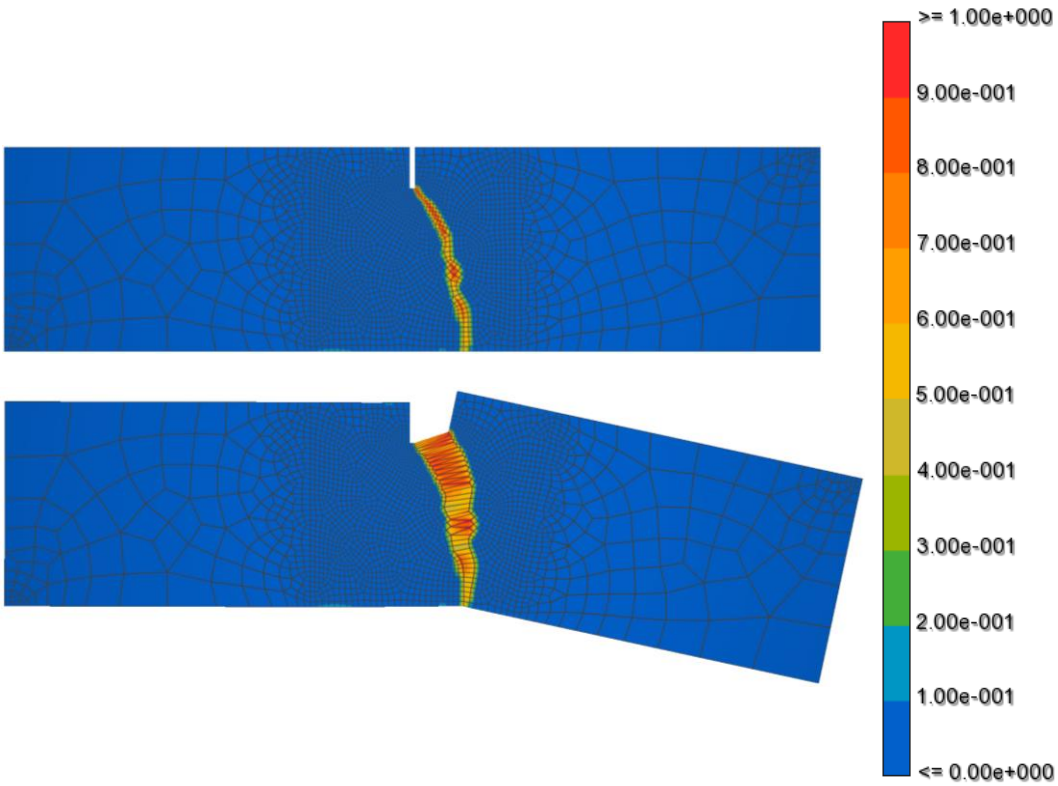

Figure 42 Spatial distribution of the scalar damage variable D for four-point shear test when the CMSD reaches 0.22 mm

9
Conclusions and suggestions for future research

9.1.
Conclusions

This work proposes novel algorithms to handle the challenges presented in modeling for complex plasticity applications, with a special focus on multisurface plasticity. This study explores and assesses state-update algorithms that are useful to simulate complex engineering materials. In this context, implicit integration algorithms are formulated aiming at robustness, comprehensiveness, and flexibility of the proposed schemes. Furthermore, this research develops coupled elastoplastic-damage modeling based on a novel constitutive model that also relies on multisurface plasticity.

Firstly, this study proposes a comprehensive implicit substepping integration scheme of single and multisurface plasticity models with nonassociated hardening/softening behavior. Line search and substepping techniques are applied to improve the convergence properties of the return mapping algorithm. The work proposes a line search strategy that combines the Armijo rule with the golden section method. Moreover, this work presents an adaptive substepping scheme and a corresponding consistent tangent modulus.

Local and global problems were studied to assess the convergence characteristics of the proposed algorithm. This work adopted the Drucker-Prager, Mohr-Coulomb and CHGC plasticity models. Some numerical examples show that the classical closest point projection method fails in some yield surface regions. The algorithm can oscillate between stress states and can enter an infinite iteration cycle. In contrast, the proposed algorithm with the Armijo rule and the golden section method adequately solved these problems. The presented state-update algorithm emerges as a way to integrate the Drucker-Prager constitutive equations since the golden section method allows the return mapping to the apex of the surface.

In addition, the finite element problems developed in this work showed how the simulation performance is improved using the line search and substepping techniques. The high curvature regions in the adopted plastic surfaces and the relatively large displacement increments provide challenging footing collapse problems since the classical closest point projection algorithm may fail in the return mapping process. According to the finite element analyses, when applying the CHGC model, the algorithm with the substepping scheme required roughly three times fewer global steps than the one without the substepping scheme. Due to the hardening behavior of the CHGC model, the substepping technique was essential to improve the convergence characteristics of the state-update algorithm and the performance of the nonlinear finite element solution. Additionally, the consistent tangent modulus formulated in this work assured the quadratic convergence of the global Newton-Raphson procedure, as shown by the global residual norms. Thus, the proposed state-update algorithm is a robust procedure to assist the simulation of complex elastoplastic models.

Besides the proposed implicit substepping scheme, this study introduces a novel Newton-Raphson-based implicit algorithm with two main characteristics: the incorporation of complementary functions to consider multisurface plasticity and the adaptation of plane strain plasticity to enforce plane stress conditions. As a result, the proposed scheme can simulate problems under different stress conditions. This algorithm provides a fundamental background for the development of a Jacobian-free implicit integration scheme.

In opposition to the complex implicit substepping scheme introduced here, a Newton-Krylov-based implicit integration algorithm was formulated in order to circumvent the need for some derivative calculations. This algorithm is remarkably flexible in handling both single and multisurface plasticity problems, as well as in considering plane stress, plane strain, axisymmetric, and three-dimensional stress conditions.

Some finite element problems were conducted to assess the effectiveness and efficiency of the Newton-Raphson-based and Newton-Krylov-based implicit integration schemes, considering both single and multisurface plasticity, as well as hardening and softening behavior. Notably, it was observed that the use of the smooth CHKS complementary function presented promising performance

compared to other complementary functions applied in other plasticity frameworks available in the literature. Moreover, the proposed Newton-Krylov-based implicit integration scheme was computationally faster compared to most cases when a finite difference-based Newton-Raphson scheme was applied. Consequently, the smooth CHKS function is highly recommended to other plasticity applications that involve multisurface plasticity or nonlinear elasticity.

Furthermore, this work proposes a coupled elastoplastic-damage model to simulate the multiaxial behavior of conventional concrete. By integrating the principles of continuum damage mechanics and plasticity theory, the damage criteria, damage evolution laws, and material tangent tensor were formulated and coupled with the novel multisurface plasticity model also developed in this work. To mitigate mesh dependency, a regularization technique based on the crack band approach is employed. To demonstrate the accuracy of the proposed model, several numerical problems were simulated using the finite element method and then compared with experimental tests. Additionally, when material parameters are not available, a parameter identification methodology was applied to estimate them accurately.

The proposed coupled elastoplastic-damage model demonstrates excellent agreement with experimental observations. The numerical predictions for load-displacement, load-CMSD, load-CMOD responses, and crack trajectories closely agree with the experimental data. Hence, the proposed model is certainly a promising approach for accurately predicting failure loads and crack propagations.

In this scenario, this research significantly advances the field of modeling complex constitutive models by addressing the challenges posed by such models with novel algorithms. This work provides an important background in formulating, implementing, and coupling in the multisurface plasticity context. These developments have practical applications in simulating the behavior of complex engineering materials and can serve as a valuable reference for future studies in the field.

9.2. Suggestions for future research

Examining different numerical methods in the context of state-update algorithms seems to be a promising research topic. Investigations may focus on exploring alternative numerical techniques for solving the nonlinear equations associated with the implicit integration of elastoplastic equations. Algorithms such as higher-order Newton-like methods, Broyden-like methods, conjugate gradient-like methods, and sequential quadratic programming may be potential tools in this scenario.

The proposed state-update algorithms are robust, comprehensive, and flexible; however, there is still room for their enhancement. An interesting aspect to be evaluated is the impact of the substepping scheme in the complementary function-based Newton-Raphson and Newton-Krylov-based formulations. Moreover, assessing the computational efficiency in the algorithm that applies the golden section method and inexact line search methods, such as quadratic and cubic interpolations, can be useful to better understand the efficiency of the proposed scheme. The line search strategy adopted in this work for the Newton-Krylov algorithm can be refined by using successive polynomial approximations to compute an optimum step size. Lastly, the Newton-Raphson-based schemes presented in this work can be further advanced by adopting the automatic differentiation, which relies heavily on the chain rule, to automatically compute the derivatives required in the implicit integration process.

The use of a suitable complementary function is essential for achieving an efficient implementation. Hence, it would be worthwhile to explore alternative functions for this purpose. Furthermore, expanding the study to evaluate the computational efficiency of different complementary functions within the state-update scheme is a valuable task, especially when considering other finite element models and constitutive formulations.

Assessing the applicability of the proposed strategies in various contexts can provide valuable insights. Therefore, evaluating these strategies in scenarios involving elastoplastic models with nonlinear elastic behavior, anisotropic plasticity, and crystal plasticity can yield further benefits and broaden the scope of their application.

In the context of the coupled elastoplastic-damage modeling proposed in this work, it would be beneficial to explore alternative regularization techniques that mitigate the mesh dependency issue. Techniques such as viscous regularization, nonlocal damage modeling, gradient-enhanced modeling, and micromorphic continuum theory could be assessed. Subsequent tests involving both coarse and fine meshes can support the evaluation of the effectiveness of these techniques. Furthermore, the proposed model can be enhanced by incorporating rate-dependent behavior, enabling accurate analysis under dynamic loading conditions for concrete structures. Investigating the model performance in dynamic scenarios can be a valuable topic for future research.

10
References

ABBO, A. J.; LYAMIN, A. V.; SLOAN, S. W.; HAMBLETON, J. P. A C2 continuous approximation to the Mohr–Coulomb yield surface. **International Journal of Solids and Structures**, v. 48, n. 21, p. 3001–3010, 2011. Available at: https://doi.org/10.1016/j.ijsolstr.2011.06.021

ABBO, A. J.; SLOAN, S. W. An Automatic Load Stepping Algorithm with Error Control. **International Journal for Numerical Methods in Engineering**, v. 39, n. 10, p. 1737–1759, 1996. Available at: https://doi.org/10.1002/(SICI)1097-0207(19960530)39:10<1737::AID-NME927>3.0.CO;2-5

ABREU, R. **Algoritmos para Integração e Calibração de Modelos Elastoplásticos com Múltiplas Superfícies de Plastificação**. 2019. 163 f. Dissertation (Master) - Departamento de Engenharia Civil e Ambiental, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2019.

ABREU, R.; MEJIA, C.; ROEHL, D. A comprehensive implicit substepping integration scheme for multisurface plasticity. **International Journal for Numerical Methods in Engineering**, v. 123, n. 1, p. 5–40, 2022a. Available at: https://doi.org/10.1002/nme.6826

ABREU, R.; MEJIA, C.; ROEHL, D. Inverse analysis of hydraulic fracturing tests based on artificial intelligence techniques. **International Journal for Numerical and Analytical Methods in Geomechanics**, v. 46, n. 13, p. 2582–2602, 2022b. Available at: https://doi.org/10.1002/nag.3419

ABREU, R.; MEJIA, C.; ROEHL, D.; PEREIRA, L. C. Estimating geomechanical parameters from hydraulic fracturing tests using a soft computing-based methodology. *In*: **Proceedings of the XLIV Ibero-Latin American Congress on Computational Methods in Engineering**. Porto, Portugal: 2023a.

ABREU, R.; MEJIA, C.; ROEHL, D.; PEREIRA, L. C. Parameter identification of minifrac numerical tests using a gradient boosting-based proxy model and genetic algorithm. **International Journal for Numerical and Analytical Methods in Geomechanics**, 2023b.

ABUBAKAR, A. B.; KUMAM, P. A descent Dai-Liao conjugate gradient method for nonlinear equations. **Numerical Algorithms**, v. 81, n. 1, p. 197–210, 2019. Available at: https://doi.org/10.1007/s11075-018-0541-z

ADHIKARY, D. P.; JAYASUNDARA, C. T.; PODGORNEY, R. K.; WILKINS, A. H. A robust return-map algorithm for general multisurface plasticity. **International Journal for Numerical Methods in Engineering**, v. 109, n. 2, p. 218–234, 2017. Available at: https://doi.org/10.1002/nme.5284

AKPAMA, H. K.; BETTAIEB, M. Ben; ABED-MERAIM, F. Numerical integration of rate-independent BCC single crystal plasticity models: comparative study of two classes of numerical algorithms. **International Journal for Numerical Methods in Engineering**, v. 108, n. 5, p. 363–422, 2016. Available at: https://doi.org/10.1002/nme.5215

ALFARAH, B.; LÓPEZ-ALMANSA, F.; OLLER, S. New methodology for calculating damage variables evolution in Plastic Damage Model for RC structures. **Engineering Structures**, v. 132, p. 70–86, 2017. Available at: https://doi.org/10.1016/j.engstruct.2016.11.022

AN, H.-B.; WEN, J.; FENG, T. On finite difference approximation of a matrix-vector product in the Jacobian-free Newton–Krylov method. **Journal of Computational and Applied Mathematics**, v. 236, n. 6, p. 1399–1409, 2011. Available at: https://doi.org/10.1016/j.cam.2011.09.003

ARAVAS, N. On the numerical integration of a class of pressure-dependent plasticity models. **International Journal for Numerical Methods in Engineering**, v. 24, n. 7, p. 1395–1416, 1987. Available at: https://doi.org/10.1002/nme.1620240713

AREIAS, P.; DIAS-DA-COSTA, D.; PIRES, E. B.; BARBOSA, J. I. A new semi-implicit formulation for multiple-surface flow rules in multiplicative plasticity. **Computational Mechanics**, v. 49, n. 5, p. 545–564, 2012. Available at: https://doi.org/10.1007/s00466-011-0659-5

AREIAS, P.; MSEKH, M. A.; RABCZUK, T. Damage and fracture algorithm using the screened Poisson equation and local remeshing. **Engineering Fracture Mechanics**, v. 158, p. 116–143, 2016. Available at: https://doi.org/10.1016/j.engfracmech.2015.10.042

AREIAS, P.; REINOSO, J.; CAMANHO, P.; RABCZUK, T. A constitutive-based element-by-element crack propagation algorithm with local mesh refinement. **Computational Mechanics**, v. 56, n. 2, p. 291–315, 2015. Available at: https://doi.org/10.1007/s00466-015-1172-z

ARMERO, F.; PÉREZ-FOGUET, A. On the formulation of closest-point projection algorithms in elastoplasticity-part I: The variational structure. **International Journal for Numerical Methods in Engineering**, v. 53, n. 2, p. 297–329, 2002. Available at: https://doi.org/10.1002/nme.278

ARMIJO, L. Minimization of functions having Lipschitz continuous first partial derivatives. **Pacific Journal of Mathematics**, v. 16, n. 1, p. 1–3, 1966. Available at: https://doi.org/10.2140/pjm.1966.16.1

ARORA, J. S. **Introduction to Optimum Design**. 3rd. ed. Waltham, USA: Elsevier, 2012.

AULIA, A.; JEONG, D.; SAAID, I. M.; KANIA, D.; SHUKER, M. T.; EL-KHATIB, N. A. A Random Forests-based sensitivity analysis framework for assisted history matching. **Journal of Petroleum Science and Engineering**, v. 181, p. 106237, 2019. Available at: https://doi.org/10.1016/j.petrol.2019.106237

BÄCK, T. Selective pressure in evolutionary algorithms: A characterization of selection mechanisms. *In*: **Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence**. Orlando, USA: IEEE, 1994. p. 57–62. Available at: https://doi.org/10.1109/ICEC.1994.350042

BAKSALARY, J. K.; STYAN, G. P. H. Generalized inverses of partitioned matrices in Banachiewicz–Schur form. **Linear Algebra and its Applications**, v. 354, n. 1–3, p. 41–47, 2002. Available at: https://doi.org/10.1016/S0024-3795(02)00334-8

BARNAFI, N. A.; PAVARINO, L. F.; SCACCHI, S. Parallel inexact Newton–Krylov and quasi-Newton solvers for nonlinear elasticity. **Computer Methods in Applied Mechanics and Engineering**, v. 400, p. 115557, 2022. Available at: https://doi.org/10.1016/j.cma.2022.115557

BATOZ, J.-L.; DHATT, G. Incremental displacement algorithms for nonlinear problems. **International Journal for Numerical Methods in Engineering**, v. 14, n. 8, p. 1262–1267, 1979. Available at: https://doi.org/10.1002/nme.1620140811

BAŽANT, Z. P.; OH, B. H. Crack band theory for fracture of concrete. **Matériaux et Constructions**, v. 16, n. 3, p. 155–177, 1983. Available at: https://doi.org/10.1007/BF02486267

BAŽANT, Z. P.; PIJAUDIER-CABOT, G. Measurement of Characteristic Length of Nonlocal Continuum. **Journal of Engineering Mechanics**, v. 115, n. 4, p. 755–767, 1989. Available at: https://doi.org/10.1061/(ASCE)0733-9399(1989)115:4(755)

BECKER, R. An alternative approach to integrating plasticity relations. **International Journal of**

Plasticity, v. 27, n. 8, p. 1224–1238, 2011. Available at: https://doi.org/10.1016/j.ijplas.2011.01.005

BELLAVIA, S.; BERRONE, S. Globalization strategies for Newton–Krylov methods for stabilized FEM discretization of Navier–Stokes equations. **Journal of Computational Physics**, v. 226, n. 2, p. 2317–2340, 2007. Available at: https://doi.org/10.1016/j.jcp.2007.07.021

BERGAMASCHI, L.; BRU, R.; MARTÍNEZ, A. Low-rank update of preconditioners for the inexact Newton method with SPD Jacobian. **Mathematical and Computer Modelling**, v. 54, n. 7–8, p. 1863–1873, 2011. Available at: https://doi.org/10.1016/j.mcm.2010.11.064

BERGAMASCHI, L.; BRU, R.; MARTÍNEZ, A.; PUTTI, M. Quasi-newton preconditioners for the inexact Newton method. **Electronic Transactions on Numerical Analysis**, v. 23, p. 76–87, 2006. Available at: http://eudml.org/doc/127449

BERGAMASCHI, L.; BRU, R.; MARTÍNEZ, A.; PUTTI, M. Quasi-Newton acceleration of ILU preconditioners for nonlinear two-phase flow equations in porous media. **Advances in Engineering Software**, v. 46, n. 1, p. 63–68, 2012. Available at: https://doi.org/10.1016/j.advengsoft.2010.10.011

BERGSTRA, J.; BARDENET, R.; BENGIO, Y.; KÉGL, B. Algorithms for Hyper-Parameter Optimization. *In*: (J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, K. Q. Weinberger, Org.). **25th Annual Conference on Neural Information Processing Systems 2011**. Granada, Spain: Curran Associates Inc., 2011. p. 2546–2554.

BERGSTRA, J.; BENGIO, Y. Random Search for Hyper-Parameter Optimization. **Journal of Machine Learning Research**, v. 13, p. 281–305, 2012. Available at: https://www.jmlr.org/papers/v13/bergstra12a.html

BERGSTRA, J.; YAMINS, D.; COX, D. D. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. *In*: (D. Sanjoy, D. McAllester, Org.). **Proceedings of the 30th International Conference on Machine Learning**. Atlanta, USA: 2013. p. 115–123.

BIĆANIĆ, N.; PEARCE, C. J. Computational aspects of a softening plasticity model for plain concrete. **Mechanics of Cohesive-frictional Materials**, v. 1, n. 1, p. 75–94, 1996. Available at: https://doi.org/10.1002/(SICI)1099-1484(199601)1:1<75::AID-CFM4>3.0.CO;2-I

BIKMUKHAMETOV, T.; JÄSCHKE, J. Oil Production Monitoring using Gradient Boosting Machine Learning Algorithm. **IFAC-PapersOnLine**, v. 52, n. 1, p. 514–519, 2019. Available at: https://doi.org/10.1016/j.ifacol.2019.06.114

BISHOP, C. M. **Pattern Recognition and Machine Learning**. New York, USA: Springer New York, 2006.

BREIMAN, L.; FRIEDMAN, J. H.; OLSHEN, R. A.; STONE, C. J. **Classification and Regression Trees**. Boca Raton, USA: Chapman & Hall/CRC, 1984.

BROWN, P. N.; SAAD, Y. Convergence Theory of Nonlinear Newton–Krylov Algorithms. **SIAM Journal on Optimization**, v. 4, n. 2, p. 297–330, 1994. Available at: https://doi.org/10.1137/0804017

BRUNO, H.; BARROS, G.; MENEZES, I. F. M.; MARTHA, L. F. Return-mapping algorithms for associative isotropic hardening plasticity using conic optimization. **Applied Mathematical Modelling**, v. 78, p. 724–748, 2020. Available at: https://doi.org/10.1016/j.apm.2019.10.006

CARPINTERI, A.; VALENTE, S.; FERRARA, G.; MELCHIORRL, G. Is mode II fracture energy a real material property? **Computers & Structures**, v. 48, n. 3, p. 397–413, 1993. Available at: https://doi.org/10.1016/0045-7949(93)90316-6

CERVERA, M.; OLIVER, J.; FARIA, R. Seismic evaluation of concrete dams via continuum

damage models. **Earthquake Engineering & Structural Dynamics**, v. 24, n. 9, p. 1225–1245, 1995. Available at: https://doi.org/10.1002/eqe.4290240905

CERVERA, M.; OLIVER, J.; MANZOLI, O. A rate-dependent isotropic damage model for the seismic analysis of concrete dams. **Earthquake Engineering & Structural Dynamics**, v. 25, n. 9, p. 987–1010, 1996. Available at: https://doi.org/10.1002/(SICI)1096-9845(199609)25:9<987::AID-EQE599>3.0.CO;2-X

CERVERA, M.; PELÀ, L.; CLEMENTE, R.; ROCA, P. A crack-tracking technique for localized damage in quasi-brittle materials. **Engineering Fracture Mechanics**, v. 77, n. 13, p. 2431–2450, 2010. Available at: https://doi.org/10.1016/j.engfracmech.2010.06.013

CHABOCHE, J. L.; CAILLETAUD, G. Integration methods for complex plastic constitutive equations. **Computer Methods in Applied Mechanics and Engineering**, v. 133, n. 1–2, p. 125–155, 1996. Available at: https://doi.org/10.1016/0045-7825(95)00957-4

CHATZIIOANNOU, K.; KARAMANOS, S. A.; HUANG, Y. An implicit numerical scheme for cyclic elastoplasticity and ratcheting under plane stress conditions. **Computers & Structures**, v. 249, p. 106509, 2021. Available at: https://doi.org/10.1016/j.compstruc.2021.106509

CHEN, C.; MANGASARIAN, O. L. Smoothing methods for convex inequalities and linear complementarity problems. **Mathematical Programming**, v. 71, n. 1, p. 51–69, 1995. Available at: https://doi.org/10.1007/BF01592244

CHEN, C.; MANGASARIAN, O. L. A class of smoothing functions for nonlinear and mixed complementarity problems. **Computational Optimization and Applications**, v. 5, n. 2, p. 97–138, 1996. Available at: https://doi.org/10.1007/BF00249052

CHEN, L.; DU, C.; MA, Y. The higher-order Levenberg–Marquardt method with Armijo type line search for nonlinear equations. **Optimization Methods and Software**, v. 32, n. 3, p. 516–533, 2017. Available at: https://doi.org/10.1080/10556788.2016.1225214

CHEN, T.; GUESTRIN, C. XGBoost: A Scalable Tree Boosting System. *In*: **Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. San Francisco, USA: Association for Computing Machinery, 2016. p. 785–794. Available at: https://doi.org/10.1145/2939672.2939785

CHEN, W.-F. **Limit analysis and soil plasticity**. Amsterdam, Netherlands: Elsevier Scientific Pub. Co, 1975.

CHEN, W.-F.; MIZUNO, E. **Nonlinear Analysis in Soil Mechanics: Theory and Implementation**. Amsterdam, Netherlands: Elsevier Science Publisher B. V., 1990.

CHENG, H.; TAN, P.-N.; GAO, J.; SCRIPPS, J. Multistep-Ahead Time Series Prediction. *In*: NG, W.-K.; KITSUREGAWA, M.; LI, J.; CHANG, K. (org.). **Advances in Knowledge Discovery and Data Mining**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 765–774. Available at: https://doi.org/10.1007/11731139_89

CHENG, W.; XIAO, Y.; HU, Q. J. A family of derivative-free conjugate gradient methods for large-scale nonlinear systems of equations. **Journal of Computational and Applied Mathematics**, v. 224, n. 1, p. 11–19, 2009. Available at: https://doi.org/10.1016/j.cam.2008.03.050

CHI, Y.; YU, M.; HUANG, L.; XU, L. Finite element modeling of steel-polypropylene hybrid fiber reinforced concrete using modified concrete damaged plasticity. **Engineering Structures**, v. 148, p. 23–35, 2017. Available at: https://doi.org/10.1016/j.engstruct.2017.06.039

CHOCKALINGAM, K.; TONKS, M. R.; HALES, J. D.; GASTON, D. R.; MILLETT, P. C.; ZHANG, L. Crystal plasticity with Jacobian-Free Newton–Krylov. **Computational Mechanics**, v. 51, n. 5, p. 617–627, 2013. Available at: https://doi.org/10.1007/s00466-012-0741-7

CHOI, H.; YOON, J. W. Stress integration-based on finite difference method and its application for anisotropic plasticity and distortional hardening under associated and non-associated flow rules. **Computer Methods in Applied Mechanics and Engineering**, v. 345, p. 123–160, 2019. Available at: https://doi.org/10.1016/j.cma.2018.10.031

CLAUSEN, J.; DAMKILDE, L.; ANDERSEN, L. Efficient return algorithms for associated plasticity with multiple yield planes. **International Journal for Numerical Methods in Engineering**, v. 66, n. 6, p. 1036–1059, 2006. Available at: https://doi.org/10.1002/nme.1595

CLAUSEN, J.; DAMKILDE, L.; ANDERSEN, L. An efficient return algorithm for non-associated plasticity with linear yield criteria in principal stress space. **Computers & Structures**, v. 85, n. 23–24, p. 1795–1807, 2007. Available at: https://doi.org/10.1016/j.compstruc.2007.04.002

CORDERO, A.; HUESO, J. L.; MARTÍNEZ, E.; TORREGROSA, J. R. Increasing the convergence order of an iterative method for nonlinear systems. **Applied Mathematics Letters**, v. 25, n. 12, p. 2369–2374, 2012. Available at: https://doi.org/10.1016/j.aml.2012.07.005

COX, A. D.; EASON, G.; HOPKINS, H. G. Axially symmetric plastic deformations in soils. **Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences**, v. 254, n. 1036, p. 1–45, 1961. Available at: https://doi.org/10.1098/rsta.1961.0011

CRISFIELD, M. A. A fast incremental/iterative solution procedure that handles "snap-through". **Computers & Structures**, v. 13, n. 1–3, p. 55–62, 1981. Available at: https://doi.org/10.1016/0045-7949(81)90108-5

CRISFIELD, M. A. **Non-linear Finite Element Analysis of Solids and Structures Volume 2: Advanced Topics**. Chichester, England: John Wiley & Sons, 1997.

CUI, S.; YIN, Y.; WANG, D.; LI, Z.; WANG, Y. A stacking-based ensemble learning method for earthquake casualty prediction. **Applied Soft Computing**, v. 101, p. 107038, 2021. Available at: https://doi.org/10.1016/j.asoc.2020.107038

DA SILVA, L. L.; PITANGUEIRA, R. L. da S.; PENNA, S. S. Multiscale numerical strategy for micromorphic description of quasi-brittle media from classical elastic damage models at the microscale. **Applied Mathematical Modelling**, v. 109, p. 52–76, 2022. Available at: https://doi.org/10.1016/j.apm.2022.03.036

DE-POUPLANA, I.; OÑATE, E. Combination of a non-local damage model for quasi-brittle materials with a mesh-adaptive finite element technique. **Finite Elements in Analysis and Design**, v. 112, p. 26–39, 2016. Available at: https://doi.org/10.1016/j.finel.2015.12.011

DE BORST, R. The zero-normal-stress condition in plane-stress and shell elastoplasticity. **Communications in Applied Numerical Methods**, v. 7, n. 1, p. 29–33, 1991. Available at: https://doi.org/10.1002/cnm.1630070105

DE BORST, R.; CRISFIELD, M. A.; REMMERS, J. J. C.; VERHOOSEL, C. V. **Non-Linear Finite Element Analysis of Solids and Structures**. Chichester, England: John Wiley & Sons Ltd, 2012.

DE SIMONE, M.; SOUZA, L. M. S.; ROEHL, D. Estimating DEM microparameters for uniaxial compression simulation with genetic programming. **International Journal of Rock Mechanics and Mining Sciences**, v. 118, p. 33–41, 2019. Available at: https://doi.org/10.1016/j.ijrmms.2019.03.024

DE SOUZA NETO, E. A.; PERIĆ, D.; OWEN, D. R. J. **Computational Methods for Plasticity**. Chichester, England: John Wiley & Sons, Ltd, 2008.

DE VREE, J. H. P.; BREKELMANS, W. A. M.; VAN GILS, M. A. J. Comparison of nonlocal approaches in continuum damage mechanics. **Computers & Structures**, v. 55, n. 4, p. 581–588, 1995. Available at: https://doi.org/10.1016/0045-7949(94)00501-S

DEMBO, R. S.; EISENSTAT, S. C.; STEIHAUG, T. Inexact Newton Methods. **SIAM Journal on Numerical Analysis**, v. 19, n. 2, p. 400–408, 1982. Available at: https://doi.org/10.1137/0719025

DENNIS, J. E.; SCHNABEL, R. B. **Numerical Methods for Unconstrained Optimization and Nonlinear Equations**. Philadelphia, USA: Society for Industrial and Applied Mathematics, 1996.

DESHPANDE, S.; WATSON, L. T.; SHU, J.; KAMKE, F. A.; RAMAKRISHNAN, N. Data driven surrogate-based optimization in the problem solving environment WBCSim. **Engineering with Computers**, v. 27, n. 3, p. 211–223, 2011. Available at: https://doi.org/10.1007/s00366-010-0192-8

DESMORAT, R.; GATUINGT, F.; RAGUENEAU, F. Nonlocal anisotropic damage model and related computational aspects for quasi-brittle materials. **Engineering Fracture Mechanics**, v. 74, n. 10, p. 1539–1560, 2007. Available at: https://doi.org/10.1016/j.engfracmech.2006.09.012

DI PRISCO, M.; MAZARS, J. "Crush-crack": a non-local damage model for concrete. **Mechanics of Cohesive-frictional Materials**, v. 1, n. 4, p. 321–347, 1996. Available at: https://doi.org/10.1002/(SICI)1099-1484(199610)1:4<321::AID-CFM17>3.0.CO;2-2

DIMAGGIO, F. L.; SANDLER, I. S. Material model for granular soils. **Journal of the Engineering Mechanics Division**, v. 97, p. 935–950, 1971.

DING, Y.; HUANG, W.; SHENG, D.; SLOAN, S. W. Numerical study on finite element implementation of hypoplastic models. **Computers and Geotechnics**, v. 68, p. 78–90, 2015. Available at: https://doi.org/10.1016/j.compgeo.2015.04.003

DODDS, R. H. Numerical techniques for plasticity computations in finite element analysis. **Computers & Structures**, v. 26, n. 5, p. 767–779, 1987. Available at: https://doi.org/10.1016/0045-7949(87)90026-5

DUTKO, M.; PERIĆ, D.; OWEN, D. R. J. Universal anisotropic yield criterion based on superquadric functional representation: Part 1. Algorithmic issues and accuracy analysis. **Computer Methods in Applied Mechanics and Engineering**, v. 109, n. 1–2, p. 73–93, 1993. Available at: https://doi.org/10.1016/0045-7825(93)90225-M

EISENSTAT, S. C.; WALKER, H. F. Globally Convergent Inexact Newton Methods. **SIAM Journal on Optimization**, v. 4, n. 2, p. 393–422, 1994. Available at: https://doi.org/10.1137/0804022

EISENSTAT, S. C.; WALKER, H. F. Choosing the Forcing Terms in an Inexact Newton Method. **SIAM Journal on Scientific Computing**, v. 17, n. 1, p. 16–32, 1996. Available at: https://doi.org/10.1137/0917003

ELSAS, J. H.; CASAPRIMA, N. A. G.; CARDOSO, P. H. S.; MENEZES, I. F. M. Bayesian optimization of riser configurations. **Ocean Engineering**, v. 236, p. 109402, 2021. Available at: https://doi.org/10.1016/j.oceaneng.2021.109402

ELSAYED, S. M.; SARKER, R. A.; ESSAM, D. L. A Comparative Study of Different Variants of Genetic Algorithms for Constrained Optimization. *In*: **8th International Conference, SEAL 2010, Kanpur, India, December 1-4, 2010, Proceedings**. v. LNCS 6457, p. 177–186. Available at: https://doi.org/10.1007/978-3-642-17298-4_18

ESMAEILI, H.; AHMADI, M. An efficient three-step method to solve system of nonlinear equations. **Applied Mathematics and Computation**, v. 266, p. 1093–1101, 2015. Available at: https://doi.org/10.1016/j.amc.2015.05.076

EVANS, M. J.; SWARTZ, T. B. **Approximating Integrals Via Monte Carlo and Deterministic Methods**. Oxford, England: Oxford University Press, 2000.

FAN, J. The modified Levenberg-Marquardt method for nonlinear equations with cubic convergence. **Mathematics of Computation**, v. 81, n. 277, p. 447–466, 2012. Available at: https://doi.org/10.1090/S0025-5718-2011-02496-8

FANG, X.; NI, Q.; ZENG, M. A modified quasi-Newton method for nonlinear equations. **Journal of Computational and Applied Mathematics**, v. 328, p. 44–58, 2018. Available at: https://doi.org/10.1016/j.cam.2017.06.024

FARIA, R.; OLIVER, J.; CERVERA, M. A strain-based plastic viscous-damage model for massive concrete structures. **International Journal of Solids and Structures**, v. 35, n. 14, p. 1533–1558, 1998. Available at: https://doi.org/10.1016/S0020-7683(97)00119-4

FEENSTRA, P. H.; DE BORST, R. A composite plasticity model for concrete. **International Journal of Solids and Structures**, v. 33, n. 5, p. 707–730, 1996. Available at: https://doi.org/10.1016/0020-7683(95)00060-N

FENG, D.-C.; REN, X.-D.; LI, J. Softened Damage-Plasticity Model for Analysis of Cracked Reinforced Concrete Structures. **Journal of Structural Engineering**, v. 144, n. 6, p. 1–15, 2018. Available at: https://doi.org/10.1061/(ASCE)ST.1943-541X.0002015

FIB. **fib Model Code for Concrete Structures 2010**. Weinheim, Germany: Wiley-VCH Verlag GmbH & Co. KGaA, 2013. Available at: https://doi.org/10.1002/9783433604090

FISCHER, A. A special newton-type optimization method. **Optimization**, v. 24, n. 3–4, p. 269–284, 1992. Available at: https://doi.org/10.1080/02331939208843795

FOSTER, C. D.; REGUEIRO, R. A.; FOSSUM, A. F.; BORJA, R. I. Implicit numerical integration of a three-invariant, isotropic/kinematic hardening cap plasticity model for geomaterials. **Computer Methods in Applied Mechanics and Engineering**, v. 194, n. 50–52, p. 5109–5138, 2005. Available at: https://doi.org/10.1016/j.cma.2005.01.001

FRANCHI, A.; GENNA, F.; PATERLINI, F. Research note on quasi-convexity of the yield function and its relation to Drucker's postulate. **International Journal of Plasticity**, v. 6, n. 3, p. 369–375, 1990. Available at: https://doi.org/10.1016/0749-6419(90)90008-3

FRAZIER, P. I. A Tutorial on Bayesian Optimization. **arXiv**: 1807.02811, 2018. Available at: http://arxiv.org/abs/1807.02811

FRIEDMAN, J. H. Greedy function approximation: A gradient boosting machine. **The Annals of Statistics**, v. 29, n. 5, p. 1189–1232, 2001. Available at: https://doi.org/10.1214/aos/1013203451

FUJII, F.; CHOONG, K. K.; GONG, S. X. Variable displacement control to overcome turning points of nonlinear elastic frames. **Computers & Structures**, v. 44, n. 1–2, p. 133–136, 1992. Available at: https://doi.org/10.1016/0045-7949(92)90232-O

FUSCHI, P.; DUTKO, M.; PERIĆ, D.; OWEN, D. R. J. On numerical integration of the five-parameter model for concrete. **Computers & Structures**, v. 53, n. 4, p. 825–838, 1994. Available at: https://doi.org/10.1016/0045-7949(94)90371-9

GASTON, D. R.; NEWMAN, C.; HANSEN, G.; LEBRUN-GRANDIÉ, D. MOOSE: A parallel computational framework for coupled systems of nonlinear equations. **Nuclear Engineering and Design**, v. 239, n. 10, p. 1768–1778, 2009. Available at: https://doi.org/10.1016/j.nucengdes.2009.05.021

GENG, D.; DAI, N.; GUO, P.; ZHOU, S.; DI, H. Implicit numerical integration of highly nonlinear plasticity models. **Computers and Geotechnics**, v. 132, p. 103961, 2021. Available at: https://doi.org/10.1016/j.compgeo.2020.103961

GODIO, M.; STEFANOU, I.; SAB, K.; SULEM, J. Multisurface plasticity for Cosserat materials: Plate element implementation and validation. **International Journal for Numerical Methods in**

Engineering, v. 108, n. 5, p. 456–484, 2016. Available at: https://doi.org/10.1002/nme.5219

GOLDBERG, D. E. Real-coded Genetic Algorithms, Virtual Alphabets, and Blocking. **Complex Systems**, v. 5, n. 2, p. 139–167, 1991.

GOMES-RUGGIERO, M. A.; LOPES, V. L. R.; TOLEDO-BENAVIDES, J. V. A globally convergent inexact Newton method with a new choice for the forcing term. **Annals of Operations Research**, v. 157, n. 1, p. 193–205, 2008. Available at: https://doi.org/10.1007/s10479-007-0196-y

GRANGE, S. An integration procedure based on multi-surface plasticity for plane stress plasticity: Theoretical formulation and application. **International Journal for Numerical Methods in Engineering**, v. 98, n. 4, p. 287–312, 2014. Available at: https://doi.org/10.1002/nme.4638

GRASSL, P.; XENOS, D.; NYSTRÖM, U.; REMPLING, R.; GYLLTOFT, K. CDPM2: A damage-plasticity approach to modelling the failure of concrete. **International Journal of Solids and Structures**, v. 50, n. 24, p. 3805–3816, 2013. Available at: https://doi.org/10.1016/j.ijsolstr.2013.07.008

GRATACOS, P.; MONTMITONNET, P.; CHENOT, J. L. An integration scheme for Prandtl-Reuss elastoplastic constitutive equations. **International Journal for Numerical Methods in Engineering**, v. 33, n. 5, p. 943–961, 1992. Available at: https://doi.org/10.1002/nme.1620330505

GRAU-SÁNCHEZ, M.; GRAU, Á.; NOGUERA, M. On the computational efficiency index and some iterative methods for solving systems of nonlinear equations. **Journal of Computational and Applied Mathematics**, v. 236, n. 6, p. 1259–1266, 2011. Available at: https://doi.org/10.1016/j.cam.2011.08.008

GU, J.; LIU, W.; ZHANG, K.; ZHAI, L.; ZHANG, Y.; CHEN, F. Reservoir production optimization based on surrograte model and differential evolution algorithm. **Journal of Petroleum Science and Engineering**, v. 205, p. 108879, 2021. Available at: https://doi.org/10.1016/j.petrol.2021.108879

HALES, J. D.; NOVASCONE, S. R.; WILLIAMSON, R. L.; GASTON, D. R.; TONKS, M. R. Solving nonlinear solid mechanics problems with the jacobian-free newton krylov method. **CMES - Computer Modeling in Engineering and Sciences**, v. 84, n. 2, p. 123–154, 2012. Available at: https://doi.org/10.3970/cmes.2012.084.123

HAMZAÇEBI, C.; AKAY, D.; KUTAY, F. Comparison of direct and iterative artificial neural network forecast approaches in multi-periodic time series forecasting. **Expert Systems with Applications**, v. 36, n. 2, p. 3839–3844, 2009. Available at: https://doi.org/10.1016/j.eswa.2008.02.042

HÄUSSLER-COMBE, U.; HARTIG, J. Formulation and numerical implementation of a constitutive law for concrete with strain-based damage and plasticity. **International Journal of Non-Linear Mechanics**, v. 43, n. 5, p. 399–415, 2008. Available at: https://doi.org/10.1016/j.ijnonlinmec.2008.01.005

HE, W.; WU, Y.-F.; XU, Y.; FU, T.-T. A thermodynamically consistent nonlocal damage model for concrete materials with unilateral effects. **Computer Methods in Applied Mechanics and Engineering**, v. 297, p. 371–391, 2015. Available at: https://doi.org/10.1016/j.cma.2015.09.010

HE, W.; XU, Y.; CHENG, Y.; JIA, P.-F.; FU, T.-T. Tension-Compression Damage Model with Consistent Crack Bandwidths for Concrete Materials. **Advances in Civil Engineering**, v. 2019, p. 1–16, 2019. Available at: https://doi.org/10.1155/2019/2810108

HEAD, T.; KUMAR, M.; NAHRSTAEDT, H.; LOUPPE, G.; SHCHERBATYI, I. **scikit-optimize/scikit-optimize**. Zenodo, 2021. Available at: https://doi.org/10.5281/zenodo.5565057

HERRERA, F.; LOZANO, M.; VERDEGAY, J. L. Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis. **Artificial Intelligence Review**, v. 12, n. 4, p. 265–319, 1998. Available at: https://doi.org/10.1023/A:1006504901164

HJIAJ, M.; FORTIN, J.; DE SAXCÉ, G. A complete stress update algorithm for the non-associated Drucker-Prager model including treatment of the apex. **International Journal of Engineering Science**, v. 41, n. 10, p. 1109–1143, 2003. Available at: https://doi.org/10.1016/S0020-7225(02)00376-2

HOFFMAN, M.; BROCHU, E.; DE FREITAS, N. Portfolio allocation for Bayesian optimization. *In*: (F. Cozman, A. Pfeffer, Org.). **Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence**. Barcelona, Spain: AUAI Press, 2011. p. 327–336.

HOFSTETTER, G.; SIMO, J. C.; TAYLOR, R. L. A modified cap model: Closest point solution algorithms. **Computers & Structures**, v. 46, n. 2, p. 203–214, 1993. Available at: https://doi.org/10.1016/0045-7949(93)90185-G

HOLLAND, J. H. **Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence**. Ann Arbor, USA: University of Michigan Press, 1975.

HUANG, J.; GRIFFITHS, D. V. Observations on Return Mapping Algorithms for Piecewise Linear Yield Criteria. **International Journal of Geomechanics**, v. 8, n. 4, p. 253–265, 2008. Available at: https://doi.org/10.1061/(ASCE)1532-3641(2008)8:4(253)

HUANG, J.; GRIFFITHS, D. V. Return Mapping Algorithms and Stress Predictors for Failure Analysis in Geomechanics. **Journal of Engineering Mechanics**, v. 135, n. 4, p. 276–284, 2009. Available at: https://doi.org/10.1061/(ASCE)0733-9399(2009)135:4(276)

JAKOBSEN, K. P.; LADE, P. V. Implementation algorithm for a single hardening constitutive model for frictional materials. **International Journal for Numerical and Analytical Methods in Geomechanics**, v. 26, n. 7, p. 661–681, 2002. Available at: https://doi.org/10.1002/nag.217

JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R. **An Introduction to Statistical Learning: with Applications in R**. New York, USA: Springer New York, 2013.

JASON, L.; HUERTA, A.; PIJAUDIER-CABOT, G.; GHAVAMIAN, S. An elastic plastic damage formulation for concrete: Application to elementary tests and comparison with an isotropic damage model. **Computer Methods in Applied Mechanics and Engineering**, v. 195, n. 52, p. 7077–7092, 2006. Available at: https://doi.org/10.1016/j.cma.2005.04.017

JAVANMARDI, M. R.; MAHERI, M. R. Extended finite element method and anisotropic damage plasticity for modelling crack propagation in concrete. **Finite Elements in Analysis and Design**, v. 165, p. 1–20, 2019. Available at: https://doi.org/10.1016/j.finel.2019.07.004

JIANG, Q.; SUN, Y.; YI, B.; LI, T.; XIONG, F. Inverse analysis for geomaterial parameter identification using Pareto multiobjective optimization. **International Journal for Numerical and Analytical Methods in Geomechanics**, v. 42, n. 14, p. 1698–1718, 2018. Available at: https://doi.org/10.1002/nag.2812

JIN, Y.-F.; YIN, Z.-Y.; SHEN, S.-L.; HICHER, P.-Y. Investigation into MOGA for identifying parameters of a critical-state-based sand model and parameters correlation by factor analysis. **Acta Geotechnica**, v. 11, n. 5, p. 1131–1145, 2016. Available at: https://doi.org/10.1007/s11440-015-0425-5

JIN, Y. A comprehensive survey of fitness approximation in evolutionary computation. **Soft Computing**, v. 9, n. 1, p. 3–12, 2005. Available at: https://doi.org/10.1007/s00500-003-0328-5

JIN, Y.; YIN, Z. Enhancement of backtracking search algorithm for identifying soil parameters. **International Journal for Numerical and Analytical Methods in Geomechanics**, v. 44, n. 9, p. 1239–1261, 2020. Available at: https://doi.org/10.1002/nag.3059

JING, C.; DONG, X.; CUI, W.; DONG, Z.; REN, L.; GU, X.; HE, Y.; PU, C.; LIU, X. Artificial

neural network–based time-domain interwell tracer testing for ultralow-permeability fractured reservoirs. **Journal of Petroleum Science and Engineering**, v. 195, p. 107558, 2020. Available at: https://doi.org/10.1016/j.petrol.2020.107558

JIRÁSEK, M. Damage and Smeared Crack Models. *In*: G., H.; G., M. (org.). **Numerical Modeling of Concrete Cracking**. Vienna, Austria: Springer Vienna, 2011. p. 1–49. Available at: https://doi.org/10.1007/978-3-7091-0897-0_1

JIRÁSEK, M.; BAUER, M. Numerical aspects of the crack band approach. **Computers & Structures**, v. 110–111, p. 60–78, 2012. Available at: https://doi.org/10.1016/j.compstruc.2012.06.006

JU, J. W. Energy-Based Coupled Elastoplastic Damage Models at Finite Strains. **Journal of Engineering Mechanics**, v. 115, n. 11, p. 2507–2525, 1989a. Available at: https://doi.org/10.1061/(ASCE)0733-9399(1989)115:11(2507)

JU, J. W. On energy-based coupled elastoplastic damage theories: Constitutive modeling and computational aspects. **International Journal of Solids and Structures**, v. 25, n. 7, p. 803–833, 1989b. Available at: https://doi.org/10.1016/0020-7683(89)90015-2

KACHANOV, L. M. Rupture time under creep conditions. **International Journal of Fracture**, v. 97, p. 11–18, 1999. Available at: https://doi.org/10.1023/A:1018671022008

KAELO, P.; ALI, M. M. Integrated crossover rules in real coded genetic algorithms. **European Journal of Operational Research**, v. 176, n. 1, p. 60–76, 2007. Available at: https://doi.org/10.1016/j.ejor.2005.07.025

KAN, Z.; SONG, N.; PENG, H.; CHEN, B. Extension of complex step finite difference method to Jacobian-free Newton–Krylov method. **Journal of Computational and Applied Mathematics**, v. 399, p. 113732, 2022. Available at: https://doi.org/10.1016/j.cam.2021.113732

KARAOULANIS, F. E. Implicit Numerical Integration of Nonsmooth Multisurface Yield Criteria in the Principal Stress Space. **Archives of Computational Methods in Engineering**, v. 20, n. 3, p. 263–308, 2013. Available at: https://doi.org/10.1007/s11831-013-9087-3

KELLEY, C. T. **Iterative Methods for Linear and Nonlinear Equations**. Philadelphia, USA: Society for Industrial and Applied Mathematics, 1995.

KELLEY, C. T. **Solving Nonlinear Equations with Newton's Method**. Philadelphia, USA: Society for Industrial and Applied Mathematics, 2003.

KIM, S.; RYU, J.; CHO, M. Numerically generated tangent stiffness matrices using the complex variable derivative method for nonlinear structural analysis. **Computer Methods in Applied Mechanics and Engineering**, v. 200, n. 1–4, p. 403–413, 2011. Available at: https://doi.org/10.1016/j.cma.2010.09.004

KIRAN, R.; KHANDELWAL, K. Complex step derivative approximation for numerical evaluation of tangent moduli. **Computers and Structures**, v. 140, p. 1–13, 2014. Available at: https://doi.org/10.1016/j.compstruc.2014.04.009

KIRAN, R.; LI, L.; KHANDELWAL, K. Performance of cubic convergent methods for implementing nonlinear constitutive models. **Computers & Structures**, v. 156, p. 83–100, 2015. Available at: https://doi.org/10.1016/j.compstruc.2015.04.011

KITZIG, M.; HÄUSSLER-COMBE, U. Modeling of plain concrete structures based on an anisotropic damage formulation. **Materials and Structures**, v. 44, n. 10, p. 1837–1853, 2011. Available at: https://doi.org/10.1617/s11527-011-9741-x

KNABE, T.; DATCHEVA, M.; LAHMER, T.; COTECCHIA, F.; SCHANZ, T. Identification of constitutive parameters of soil using an optimization strategy and statistical analysis. **Computers**

and Geotechnics, v. 49, p. 143–157, 2013. Available at: https://doi.org/10.1016/j.compgeo.2012.10.002

KNOLL, D. A.; KEYES, D. E. Jacobian-free Newton–Krylov methods: a survey of approaches and applications. **Journal of Computational Physics**, v. 193, n. 2, p. 357–397, 2004. Available at: https://doi.org/10.1016/j.jcp.2003.08.010

KOC, K.; EKMEKCIOĞLU, Ö.; GURGUN, A. P. Integrating feature engineering, genetic algorithm and tree-based machine learning methods to predict the post-accident disability status of construction workers. **Automation in Construction**, v. 131, p. 103896, 2021. Available at: https://doi.org/10.1016/j.autcon.2021.103896

KOHLER, M.; VELLASCO, M.; SILVA, E.; FIGUEIREDO, K. SimProxy Decision Support System: A Neural Network Proxy Applied to Reservoir and Surface Integrated Optimization. **IEEE Systems Journal**, v. 14, n. 4, p. 5111–5120, 2020. Available at: https://doi.org/10.1109/JSYST.2020.2968239

KOITER, W. T. Stress-strain relations, uniqueness and variational theorems for elastic-plastic materials with a singular yield surface. **Quarterly of Applied Mathematics**, v. 11, n. 3, p. 350–354, 1953. Available at: https://doi.org/10.1090/qam/59769

KRABBENHØFT, K.; LYAMIN, A. V. Computational Cam clay plasticity using second-order cone programming. **Computer Methods in Applied Mechanics and Engineering**, v. 209–212, p. 239–249, 2012. Available at: https://doi.org/10.1016/j.cma.2011.11.006

KRABBENHØFT, K.; LYAMIN, A. V.; SLOAN, S. W. Formulation and solution of some plasticity problems as conic programs. **International Journal of Solids and Structures**, v. 44, n. 5, p. 1533–1549, 2007. Available at: https://doi.org/10.1016/j.ijsolstr.2006.06.036

KRABBENHØFT, K.; LYAMIN, A. V.; SLOAN, S. W. Three-dimensional Mohr-Coulomb limit analysis using semidefinite programming. **Communications in Numerical Methods in Engineering**, v. 24, n. 11, p. 1107–1119, 2008. Available at: https://doi.org/10.1002/cnm.1018

KRABBENHØFT, K.; LYAMIN, A. V.; SLOAN, S. W.; WRIGGERS, P. An interior-point algorithm for elastoplasticity. **International Journal for Numerical Methods in Engineering**, v. 69, n. 3, p. 592–626, 2007. Available at: https://doi.org/10.1002/nme.1771

KUHN, M.; JOHNSON, K. **Applied Predictive Modeling**. New York, USA: Springer New York, 2013.

KUPFER, H.; HILSDORF, H. K.; RÜSCH, H. Behavior of Concrete Under Biaxial Stresses. **ACI Journal Proceedings**, v. 66, n. 8, p. 66–52, 1969. Available at: https://doi.org/10.14359/7388

KURUMATANI, M.; SOMA, Y.; TERADA, K. Simulations of cohesive fracture behavior of reinforced concrete by a fracture-mechanics-based damage model. **Engineering Fracture Mechanics**, v. 206, p. 392–407, 2019. Available at: https://doi.org/10.1016/j.engfracmech.2018.12.006

KURUMATANI, M.; TERADA, K.; KATO, J.; KYOYA, T.; KASHIYAMA, K. An isotropic damage model based on fracture mechanics for concrete. **Engineering Fracture Mechanics**, v. 155, p. 49–66, 2016. Available at: https://doi.org/10.1016/j.engfracmech.2016.01.020

LAGIOIA, R.; PANTEGHINI, A. On the existence of a unique class of yield and failure criteria comprising Tresca, von Mises, Drucker–Prager, Mohr–Coulomb, Galileo–Rankine, Matsuoka–Nakai and Lade–Duncan. **Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences**, v. 472, n. 2185, 2016. Available at: https://doi.org/10.1098/rspa.2015.0713

LE, T. M.; FATAHI, B.; KHABBAZ, H. Numerical optimisation to obtain elastic viscoplastic model parameters for soft clay. **International Journal of Plasticity**, v. 65, p. 1–21, 2015. Available at: https://doi.org/10.1016/j.ijplas.2014.08.008

LEE, J.; FENVES, G. L. A plastic-damage concrete model for earthquake analysis of dams. **Earthquake Engineering & Structural Dynamics**, v. 27, n. 9, p. 937–956, 1998a. Available at: https://doi.org/10.1002/(SICI)1096-9845(199809)27:9<937::AID-EQE764>3.0.CO;2-5

LEE, J.; FENVES, G. L. Plastic-Damage Model for Cyclic Loading of Concrete Structures. **Journal of Engineering Mechanics**, v. 124, n. 8, p. 892–900, 1998b. Available at: https://doi.org/10.1061/(ASCE)0733-9399(1998)124:8(892)

LEE, J.; FENVES, G. L. A return-mapping algorithm for plastic-damage models: 3-D and plane stress formulation. **International Journal for Numerical Methods in Engineering**, v. 50, n. 2, p. 487–506, 2001. Available at: https://doi.org/10.1002/1097-0207(20010120)50:2<487::AID-NME44>3.0.CO;2-N

LEMAITRE, J. Coupled elasto-plasticity and damage constitutive equations. **Computer Methods in Applied Mechanics and Engineering**, v. 51, n. 1–3, p. 31–49, 1985. Available at: https://doi.org/10.1016/0045-7825(85)90026-X

LEMAITRE, J.; DESMORAT, R. **Engineering Damage Mechanics**. Berlin/Heidelberg, Germany: Springer-Verlag, 2005.

LEON, S. E.; PAULINO, G. H.; PEREIRA, A.; MENEZES, I. F. M.; LAGES, E. N. A Unified Library of Nonlinear Solution Schemes. **Applied Mechanics Reviews**, v. 64, n. 4, 2011. Available at: https://doi.org/10.1115/1.4006992

LESTER, A. M.; SLOAN, S. W. A smooth hyperbolic approximation to the Generalised Classical yield function, including a true inner rounding of the Mohr-Coulomb deviatoric section. **Computers and Geotechnics**, v. 104, p. 331–357, 2018. Available at: https://doi.org/10.1016/j.compgeo.2017.12.002

LESTER, B. T.; SCHERZINGER, W. M. Trust-region based return mapping algorithm for implicit integration of elastic-plastic constitutive models. **International Journal for Numerical Methods in Engineering**, v. 112, n. 3, p. 257–282, 2017. Available at: https://doi.org/10.1002/nme.5515

LI, C.; LI, C.; ZHENG, H. Subspace tracking method for non-smooth yield surface. **Computers and Mathematics with Applications**, v. 90, p. 125–134, 2021. Available at: https://doi.org/10.1016/j.camwa.2021.03.012

LI, D.-H.; FUKUSHIMA, M. A derivative-free line search and global convergence of Broyden-like method for nonlinear equations. **Optimization Methods and Software**, v. 13, n. 3, p. 181–201, 2000. Available at: https://doi.org/10.1080/10556780008805782

LI, T.; CROUCH, R. A C2 plasticity model for structural concrete. **Computers & Structures**, v. 88, n. 23–24, p. 1322–1332, 2010. Available at: https://doi.org/10.1016/j.compstruc.2008.10.009

LIMA, M. J.; LEMOS, M.; PEREIRA, F.; COUTO, R.; ROEHL, D. Functional Requirements for Developing ERAS - A Portal to Support Collaborative Geomechanical Simulations. *In*: **BreSci - XII Brazilian e-Science Workshop - XXXVIII Congresso da Sociedade Brasileira de Computação**. Natal, Brazil: 2018.

LIU, J.; ZIO, E. SVM hyperparameters tuning for recursive multi-step-ahead prediction. **Neural Computing and Applications**, v. 28, n. 12, p. 3749–3763, 2017. Available at: https://doi.org/10.1007/s00521-016-2272-1

LLORET-CABOT, M.; SLOAN, S. W.; SHENG, D.; ABBO, A. J. Error behaviour in explicit integration algorithms with automatic substepping. **International Journal for Numerical Methods in Engineering**, v. 108, n. 9, p. 1030–1053, 2016. Available at: https://doi.org/10.1002/nme.5245

LOURENÇO, P. B.; DE BORST, R.; ROTS, J. G. A plane stress softening plasticity model for orthotropic materials. **International Journal for Numerical Methods in Engineering**, v. 40, n. 21,

p. 4033–4057, 1997. Available at: https://doi.org/10.1002/(SICI)1097-0207(19971115)40:21<4033::AID-NME248>3.0.CO;2-0

LOURENÇO, P. B.; ROTS, J. G. Multisurface Interface Model for Analysis of Masonry Structures. **Journal of Engineering Mechanics**, v. 123, n. 7, p. 660–668, 1997. Available at: https://doi.org/10.1061/(ASCE)0733-9399(1997)123:7(660)

LU, D.; ZHANG, Y.; ZHOU, X.; SU, C.; GAO, Z.; DU, X. A robust stress update algorithm for elastoplastic models without analytical derivation of the consistent tangent operator and loading/unloading estimation. **International Journal for Numerical and Analytical Methods in Geomechanics**, v. 47, n. 6, p. 1022–1050, 2023. Available at: https://doi.org/10.1002/nag.3503

LUBLINER, J.; OLIVER, J.; OLLER, S.; OÑATE, E. A plastic-damage model for concrete. **International Journal of Solids and Structures**, v. 25, n. 3, p. 299–326, 1989. Available at: https://doi.org/10.1016/0020-7683(89)90050-4

LUKACEVIC, M.; LEDERER, W.; FÜSSL, J. A microstructure-based multisurface failure criterion for the description of brittle and ductile failure mechanisms of clear-wood. **Engineering Fracture Mechanics**, v. 176, p. 83–99, 2017. Available at: https://doi.org/10.1016/j.engfracmech.2017.02.020

MA, C.; CHEN, W. Z. Three-dimensional elastoplastic damage concrete model by dissipation-based arc-length method. **Advances in Structural Engineering**, v. 19, n. 12, p. 1949–1962, 2016. Available at: https://doi.org/10.1177/1369433216649391

MACKENZIE-HELNWEIN, P.; EBERHARDSTEINER, J.; MANG, H. A. A multi-surface plasticity model for clear wood and its application to the finite element analysis of structural details. **Computational Mechanics**, v. 31, n. 1–2, p. 204–218, 2003. Available at: https://doi.org/10.1007/s00466-003-0423-6

MAZARS, J. Mechanical Damage and Fracture of Concrete Structures. *In*: **5th International Conference on Fracture**. Cannes, France: 1981. p. 1499–1506.

MAZARS, J. A description of micro- and macroscale damage of concrete structures. **Engineering Fracture Mechanics**, v. 25, n. 5–6, p. 729–737, 1986. Available at: https://doi.org/10.1016/0013-7944(86)90036-6

MAZARS, J.; BERTHAUD, Y.; RAMTANI, S. The unilateral behaviour of damaged concrete. **Engineering Fracture Mechanics**, v. 35, n. 4–5, p. 629–635, 1990. Available at: https://doi.org/10.1016/0013-7944(90)90145-7

MAZARS, J.; HAMON, F.; GRANGE, S. A new 3D damage model for concrete under monotonic, cyclic and dynamic loadings. **Materials and Structures**, v. 48, n. 11, p. 3779–3793, 2015. Available at: https://doi.org/10.1617/s11527-014-0439-8

MAZARS, J.; PIJAUDIER-CABOT, G. Continuum Damage Theory—Application to Concrete. **Journal of Engineering Mechanics**, v. 115, n. 2, p. 345–365, 1989. Available at: https://doi.org/10.1061/(ASCE)0733-9399(1989)115:2(345)

MCKAY, M. D.; BECKMAN, R. J.; CONOVER, W. J. Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. **Technometrics**, v. 21, n. 2, p. 239–245, 1979. Available at: https://doi.org/10.1080/00401706.1979.10489755

MENDES, C. A. T.; GATTASS, M.; ROEHL, D. The GeMA Framework - An Innovative Framework for the Development of Multiphysics and Multiscale Simulations. *In*: **VII European Congress on Computational Methods in Applied Sciences and Engineering**. Crete Island, Greece: 2016.

MENG, M.; ZHONG, R.; WEI, Z. Prediction of methane adsorption in shale: Classical models and machine learning based models. **Fuel**, v. 278, p. 118358, 2020. Available at:

https://doi.org/10.1016/j.fuel.2020.118358

MICHALEWICZ, Z. **Genetic Algorithms + Data Structures = Evolution Programs**. Berlin/Heidelberg, Germany: Springer Berlin Heidelberg, 1996.

MIEHE, C. Numerical computation of algorithmic (consistent) tangent moduli in large-strain computational inelasticity. **Computer Methods in Applied Mechanics and Engineering**, v. 134, n. 3–4, p. 223–240, 1996. Available at: https://doi.org/10.1016/0045-7825(96)01019-5

MIHAI, I. C.; JEFFERSON, A. D.; LYONS, P. A plastic-damage constitutive model for the finite element analysis of fibre reinforced concrete. **Engineering Fracture Mechanics**, v. 159, p. 35–62, 2016. Available at: https://doi.org/10.1016/j.engfracmech.2015.12.035

MILLER, B. L.; GOLDBERG, D. E. Genetic Algorithms, Tournament Selection, and the Effects of Noise. **Complex Systems**, v. 9, n. 3, p. 193–212, 1995. Available at: https://www.mitpressjournals.org/doi/abs/10.1162/evco.1996.4.2.113

MOKHADE, A. S.; KAKDE, O. G. Overview of Selection Schemes in Real-Coded Genetic Algorithms and Their Applications. **Journal of Industrial and Intelligent Information**, v. 2, n. 1, p. 71–77, 2014. Available at: https://doi.org/10.12720/jiii.2.1.71-77

MORAES, I.; ABREU, R.; MEJIA, C.; ROEHL, D. Numerical modeling of fracture propagation using continuum damage and cohesive crack models. *In*: **Proceedings of the XLI Ibero-Latin-American Congress on Computational Methods in Engineering**. Foz do Iguaçu, Brazil: 2020. p. 7.

MURPHY, K. P. **Machine Learning: A Probabilistic Perspective**. Cambridge, England: MIT Press, 2012.

NGUYEN, H.-P.; LIU, J.; ZIO, E. A long-term prediction approach based on long short-term memory neural networks with automatic parameter optimization by Tree-structured Parzen Estimator and applied to time-series data of NPP steam generators. **Applied Soft Computing**, v. 89, p. 106116, 2020. Available at: https://doi.org/10.1016/j.asoc.2020.106116

NIE, Y.; SHEIKH, A.; GRIFFITH, M.; VISINTIN, P. A damage-plasticity based interface model for simulating in-plane/out-of-plane response of masonry structural panels. **Computers & Structures**, v. 260, p. 106721, 2022. Available at: https://doi.org/10.1016/j.compstruc.2021.106721

NOCEDAL, J.; WRIGHT, S. J. **Numerical Optimization**. New York, USA: Springer New York, 2006.

OHNO, N.; TSUDA, M.; KAMEI, T. Elastoplastic implicit integration algorithm applicable to both plane stress and three-dimensional stress states. **Finite Elements in Analysis and Design**, v. 66, p. 1–11, 2013. Available at: https://doi.org/10.1016/j.finel.2012.11.001

OHNO, N.; TSUDA, M.; SUGIYAMA, H.; OKUMURA, D. Elastic–viscoplastic implicit integration algorithm applicable to both plane stress and three-dimensional stress states. **Finite Elements in Analysis and Design**, v. 109, p. 54–64, 2016. Available at: https://doi.org/10.1016/j.finel.2015.10.001

OLIVER, J. A consistent characteristic length for smeared cracking models. **International Journal for Numerical Methods in Engineering**, v. 28, n. 2, p. 461–474, 1989. Available at: https://doi.org/10.1002/nme.1620280214

OLIVER, J.; CERVERA, M.; OLLER, S.; LUBLINER, J. Isotropic Damage Models and Smeared Crack Analysis of Concrete. *In*: **II International Conference on Computer Aided Analysis and Design of Concrete Structures**. Zell am See, Austria: 1990. p. 945–957.

OLIVER, J.; HUESPE, A.; PULIDO, M. D. G.; BLANCO, S. Computational modeling of cracking of concrete in strong discontinuity settings. **Computers and Concrete**, v. 1, n. 1, p. 61–76, 2004.

Available at: https://doi.org/10.12989/cac.2004.1.1.061

ORTIZ, M.; POPOV, E. P. Accuracy and stability of integration algorithms for elastoplastic constitutive relations. **International Journal for Numerical Methods in Engineering**, v. 21, n. 9, p. 1561–1576, 1985. Available at: https://doi.org/10.1002/nme.1620210902

PANTEGHINI, A.; LAGIOIA, R. An approach for providing quasi-convexity to yield functions and a generalized implicit integration scheme for isotropic constitutive models based on 2 unknowns. **International Journal for Numerical and Analytical Methods in Geomechanics**, v. 42, n. 6, p. 829–855, 2018. Available at: https://doi.org/10.1002/nag.2767

PECH, S.; LUKACEVIC, M.; FÜSSL, J. A robust multisurface return-mapping algorithm and its implementation in Abaqus. **Finite Elements in Analysis and Design**, v. 190, p. 103531, 2021. Available at: https://doi.org/10.1016/j.finel.2021.103531

PEDREGOSA, F. *et al.* Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, v. 12, n. 85, p. 2825–2830, 2011. Available at: http://scikit-learn.sourceforge.net

PEERLINGS, R. H. J.; DE BORST, R.; BREKELMANS, W. A. M.; DE VREE, J. H. P. Gradient enhanced damage for quasi-brittle materials. **International Journal for Numerical Methods in Engineering**, v. 39, n. 19, p. 3391–3403, 1996. Available at: https://doi.org/10.1002/(SICI)1097-0207(19961015)39:19<3391::AID-NME7>3.0.CO;2-D

PEREIRA, L. F.; WEERHEIJM, J.; SLUYS, L. J. A new effective rate dependent damage model for dynamic tensile failure of concrete. **Engineering Fracture Mechanics**, v. 176, p. 281–299, 2017a. Available at: https://doi.org/10.1016/j.engfracmech.2017.03.048

PEREIRA, L. F.; WEERHEIJM, J.; SLUYS, L. J. A numerical study on crack branching in quasi-brittle materials with a new effective rate-dependent nonlocal damage model. **Engineering Fracture Mechanics**, v. 182, p. 689–707, 2017b. Available at: https://doi.org/10.1016/j.engfracmech.2017.06.019

PÉREZ-FOGUET, A.; ARMERO, F. On the formulation of closest-point projection algorithms in elastoplasticity-part II: Globally convergent schemes. **International Journal for Numerical Methods in Engineering**, v. 53, n. 2, p. 331–374, 2002. Available at: https://doi.org/10.1002/nme.279

PÉREZ-FOGUET, A.; RODRÍGUEZ-FERRAN, A.; HUERTA, A. Numerical differentiation for local and global tangent operators in computational plasticity. **Computer Methods in Applied Mechanics and Engineering**, v. 189, n. 1, p. 277–296, 2000a. Available at: https://doi.org/10.1016/S0045-7825(99)00296-0

PÉREZ-FOGUET, A.; RODRÍGUEZ-FERRAN, A.; HUERTA, A. Numerical differentiation for non-trivial consistent tangent matrices: an application to the MRS-Lade model. **International Journal for Numerical Methods in Engineering**, v. 48, n. 2, p. 159–184, 2000b. Available at: https://doi.org/10.1002/(SICI)1097-0207(20000520)48:2<159::AID-NME871>3.0.CO;2-Y

PÉREZ-FOGUET, A.; RODRÍGUEZ-FERRAN, A.; HUERTA, A. Consistent tangent matrices for substepping schemes. **Computer Methods in Applied Mechanics and Engineering**, v. 190, n. 35–36, p. 4627–4647, 2001. Available at: https://doi.org/10.1016/S0045-7825(00)00336-4

PERNICE, M.; WALKER, H. F. NITSOL: A Newton Iterative Solver for Nonlinear Systems. **SIAM Journal on Scientific Computing**, v. 19, n. 1, p. 302–318, 1998. Available at: https://doi.org/10.1137/S1064827596303843

PETRACCA, M.; PELÀ, L.; ROSSI, R.; OLLER, S.; CAMATA, G.; SPACONE, E. Regularization of first order computational homogenization for multiscale analysis of masonry structures. **Computational Mechanics**, v. 57, n. 2, p. 257–276, 2016. Available at: https://doi.org/10.1007/s00466-015-1230-6

PETRACCA, M.; PELÀ, L.; ROSSI, R.; ZAGHI, S.; CAMATA, G.; SPACONE, E. Micro-scale continuous and discrete numerical models for nonlinear analysis of masonry shear walls. **Construction and Building Materials**, v. 149, p. 296–314, 2017. Available at: https://doi.org/10.1016/j.conbuildmat.2017.05.130

PIJAUDIER-CABOT, G.; MAZARS, J.; PULIKOWSKI, J. Steel-Concrete Bond Analysis with Nonlocal Continuous Damage. **Journal of Structural Engineering**, v. 117, n. 3, p. 862–882, 1991. Available at: https://doi.org/10.1061/(ASCE)0733-9445(1991)117:3(862)

PIJAUDIER-CABOT, G.; BAŽANT, Z. P. Nonlocal Damage Theory. **Journal of Engineering Mechanics**, v. 113, n. 10, p. 1512–1533, 1987. Available at: https://doi.org/10.1061/(ASCE)0733-9399(1987)113:10(1512)

POLIOTTI, M.; BAIRÁN, J.-M. A new concrete plastic-damage model with an evolutive dilatancy parameter. **Engineering Structures**, v. 189, p. 541–549, 2019. Available at: https://doi.org/10.1016/j.engstruct.2019.03.086

PRESS, W. H.; TEUKOLSKY, S. A.; VETTERLING, W. T.; FLANNERY, B. P. **Numerical Recipes**. 3rd. ed. Cambridge, England: Cambridge University Press, 2007.

QI, H.; LI, T.; LIU, X.; ZHAO, L.; LIN, C.; FAN, S. A variable parameters damage model for concrete. **Engineering Fracture Mechanics**, v. 228, p. 106898, 2020. Available at: https://doi.org/10.1016/j.engfracmech.2020.106898

QUEVEDO, R.; FIRME, P.; ROEHL, D. Integration schemes with substepping algorithms for creep analysis in geomaterials. **International Journal for Numerical and Analytical Methods in Geomechanics**, v. 43, n. 7, p. 1467–1487, 2019. Available at: https://doi.org/10.1002/nag.2910

RAHUL; DE, S. An efficient block preconditioner for Jacobian-free global-local multiscale methods. **International Journal for Numerical Methods in Engineering**, v. 87, n. 7, p. 639–663, 2011. Available at: https://doi.org/10.1002/nme.3123

RAHUL; DE, S. Analysis of the Jacobian-free multiscale method (JFMM). **Computational Mechanics**, v. 56, n. 5, p. 769–783, 2015a. Available at: https://doi.org/10.1007/s00466-015-1200-z

RAHUL; DE, S. Multiscale modeling of polycrystalline materials with Jacobian-free multiscale method (JFMM). **Computational Mechanics**, v. 55, n. 4, p. 643–657, 2015b. Available at: https://doi.org/10.1007/s00466-015-1128-3

RAMM, E. Strategies for Tracing the Nonlinear Response Near Limit Points. *In*: WUNDERLICH, W.; E. STEIN; BATHE, K. J. (org.). **Nonlinear Finite Element Analysis in Structural Mechanics**. Berlin/Heidelberg, Germany: Springer Berlin Heidelberg, 1981. p. 63–89. Available at: https://doi.org/10.1007/978-3-642-81589-8_5

RAO, X.; ZHAO, H.; DENG, Q. Artificial-neural-network (ANN) based proxy model for performances forecast and inverse project design of water huff-n-puff technology. **Journal of Petroleum Science and Engineering**, v. 195, p. 107851, 2020. Available at: https://doi.org/10.1016/j.petrol.2020.107851

RASMUSSEN, C. E.; WILLIAMS, C. K. I. **Gaussian processes for machine learning**. Cambridge, England: MIT Press, 2006.

REGES, P. D. N.; PITANGUEIRA, R. L. S.; SILVA, L. L. Elastic degradation models for the micromorphic continuum. **International Journal of Non-Linear Mechanics**, v. 154, p. 104450, 2023. Available at: https://doi.org/10.1016/j.ijnonlinmec.2023.104450

REZAIEE-PAJAND, M.; NASIRAI, C. On the integration schemes for Drucker–Prager's elastoplastic models based on exponential maps. **International Journal for Numerical Methods in Engineering**, v. 74, n. 5, p. 799–826, 2008. Available at: https://doi.org/10.1002/nme.2178

RIAZI, S. H.; ZARGAR, G.; BAHARIMOGHADAM, M.; MOSLEMI, B.; SHARIFI DARANI, E. Fractured reservoir history matching improved based on artificial intelligent. **Petroleum**, v. 2, n. 4, p. 344–360, 2016. Available at: https://doi.org/10.1016/j.petlm.2016.09.001

SAAD, Y.; SCHULTZ, M. H. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. **SIAM Journal on Scientific and Statistical Computing**, v. 7, n. 3, p. 856–869, 1986. Available at: https://doi.org/10.1137/0907058

SALOUSTROS, S.; PELÀ, L.; CERVERA, M.; ROCA, P. Finite element modelling of internal and multiple localized cracks. **Computational Mechanics**, v. 59, n. 2, p. 299–316, 2017. Available at: https://doi.org/10.1007/s00466-016-1351-6

SANDLER, I. S. The Cap Model for Static and Dynamic Problems. *In*: **The 17th U.S. Symposium on Rock Mechanics (USRMS), 25-27 August, Snow Bird, Utah**. American Rock Mechanics Association, 1976.

SANDLER, I. S.; DIMAGGIO, F. L.; BALADI, G. Y. Generalized Cap Model for Geological Materials. **Journal of the Geotechnical Engineering Division**, p. 683–699, 1976.

SANDLER, I. S.; RUBIN, D. An algorithm and a modular subroutine for the CAP model. **International Journal for Numerical and Analytical Methods in Geomechanics**, v. 3, n. 2, p. 173–186, 1979. Available at: https://doi.org/10.1002/nag.1610030206

SANTANA, M. V. B.; KEO, P.; HJIAJ, M. Efficient and low memory strain-rate independent return mapping algorithm for general yield surfaces and stress states. **Advances in Engineering Software**, v. 164, p. 103067, 2022. Available at: https://doi.org/10.1016/j.advengsoft.2021.103067

SANTOS, E. S. R.; DE SOUZA NETO, E. A. Two-invariant modular implicit plasticity solver framework for geomechanical simulators. **International Journal for Numerical and Analytical Methods in Geomechanics**, v. 46, n. 18, p. 3310–3341, 2022. Available at: https://doi.org/10.1002/nag.3452

SARITAS, A.; FILIPPOU, F. C. Numerical integration of a class of 3d plastic-damage concrete models and condensation of 3d stress–strain relations for use in beam finite elements. **Engineering Structures**, v. 31, n. 10, p. 2327–2336, 2009. Available at: https://doi.org/10.1016/j.engstruct.2009.05.005

SAWYERR, B. A.; ADEWUMI, A. O.; ALI, M. M. Real-coded genetic algorithm with uniform random local search. **Applied Mathematics and Computation**, v. 228, p. 589–597, 2014. Available at: https://doi.org/10.1016/j.amc.2013.11.097

SCALET, G.; AURICCHIO, F. Computational Methods for Elastoplasticity: An Overview of Conventional and Less-Conventional Approaches. **Archives of Computational Methods in Engineering**, v. 25, n. 3, p. 545–589, 2018. Available at: https://doi.org/10.1007/s11831-016-9208-x

SCHERZINGER, W. M. A return mapping algorithm for isotropic and anisotropic plasticity models using a line search method. **Computer Methods in Applied Mechanics and Engineering**, v. 317, p. 526–553, 2017. Available at: https://doi.org/10.1016/j.cma.2016.11.026

SCHERZINGER, W. M.; DOHRMANN, C. R. A robust algorithm for finding the eigenvalues and eigenvectors of 3×3 symmetric matrices. **Computer Methods in Applied Mechanics and Engineering**, v. 197, n. 45–48, p. 4007–4015, 2008. Available at: https://doi.org/10.1016/j.cma.2008.03.031

SCHEUNEMANN, L.; NIGRO, P. S. B.; SCHRÖDER, J. Numerical treatment of small strain single crystal plasticity based on the infeasible primal-dual interior point method. **International Journal of Solids and Structures**, v. 232, p. 111149, 2021. Available at: https://doi.org/10.1016/j.ijsolstr.2021.111149

SCHMIDT-BALDASSARI, M. Numerical concepts for rate-independent single crystal plasticity. **Computer Methods in Applied Mechanics and Engineering**, v. 192, n. 11–12, p. 1261–1280, 2003. Available at: https://doi.org/10.1016/S0045-7825(02)00563-7

SCHWER, L. E.; MURRAY, Y. D. A three-invariant smooth cap model with mixed hardening. **International Journal for Numerical and Analytical Methods in Geomechanics**, v. 18, n. 10, p. 657–688, 1994. Available at: https://doi.org/10.1002/nag.1610181002

SEIFERT, T.; SCHMIDT, I. Line-search methods in general return mapping algorithms with application to porous plasticity. **International Journal for Numerical Methods in Engineering**, v. 73, n. 10, p. 1468–1495, 2008. Available at: https://doi.org/10.1002/nme.2131

SHARMA, J. R.; ARORA, H. Improved Newton-like methods for solving systems of nonlinear equations. **SeMA Journal**, v. 74, p. 147–163, 2017. Available at: https://doi.org/10.1007/s40324-016-0085-x

SHARMA, J. R.; GUHA, R. K.; SHARMA, R. An efficient fourth order weighted-Newton method for systems of nonlinear equations. **Numerical Algorithms**, v. 62, n. 2, p. 307–323, 2013. Available at: https://doi.org/10.1007/s11075-012-9585-7

SHARMA, J. R.; GUPTA, P. An efficient fifth order method for solving systems of nonlinear equations. **Computers and Mathematics with Applications**, v. 67, n. 3, p. 591–601, 2014. Available at: https://doi.org/10.1016/j.camwa.2013.12.004

SHI, R.; XU, X.; LI, J.; LI, Y. Prediction and analysis of train arrival delay based on XGBoost and Bayesian optimization. **Applied Soft Computing**, v. 109, p. 107538, 2021. Available at: https://doi.org/10.1016/j.asoc.2021.107538

SHIN, B. C.; DARVISHI, M. T.; KIM, C. H. A comparison of the Newton-Krylov method with high order Newton-like methods to solve nonlinear systems. **Applied Mathematics and Computation**, v. 217, n. 7, p. 3190–3198, 2010. Available at: https://doi.org/10.1016/j.amc.2010.08.051

SHTERENLIKHT, A.; ALEXANDER, N. A. Levenberg–Marquardt vs Powell's dogleg method for Gurson–Tvergaard–Needleman plasticity model. **Computer Methods in Applied Mechanics and Engineering**, v. 237–240, p. 1–9, 2012. Available at: https://doi.org/10.1016/j.cma.2012.04.018

SIMO, J. C.; HUGHES, T. J. R. **Computational Inelasticity**. New York, USA: Springer-Verlag, 1998.

SIMO, J. C.; JU, J.; PISTER, K. S.; TAYLOR, R. L. Assessment of Cap Model: Consistent Return Algorithms and Rate-Dependent Extension. **Journal of Engineering Mechanics**, v. 114, n. 2, p. 191–218, 1988. Available at: https://doi.org/10.1061/(ASCE)0733-9399(1988)114:2(191)

SIMO, J. C.; JU, J. W. Strain- and stress-based continuum damage models—I. Formulation. **International Journal of Solids and Structures**, v. 23, n. 7, p. 821–840, 1987a. Available at: https://doi.org/10.1016/0020-7683(87)90083-7

SIMO, J. C.; JU, J. W. Strain- and stress-based continuum damage models—II. Computational aspects. **International Journal of Solids and Structures**, v. 23, n. 7, p. 841–869, 1987b. Available at: https://doi.org/10.1016/0020-7683(87)90084-9

SIMO, J. C.; KENNEDY, J. G.; GOVINDJEE, S. Non-smooth multisurface plasticity and viscoplasticity. Loading/unloading conditions and numerical algorithms. **International Journal for Numerical Methods in Engineering**, v. 26, n. 10, p. 2161–2185, 1988. Available at: https://doi.org/10.1002/nme.1620261003

SIMO, J. C.; TAYLOR, R. L. Consistent tangent operators for rate-independent elastoplasticity. **Computer Methods in Applied Mechanics and Engineering**, v. 48, n. 1, p. 101–118, 1985.

Available at: https://doi.org/10.1016/0045-7825(85)90070-2

SIMO, J. C.; TAYLOR, R. L. A return mapping algorithm for plane stress elastoplasticity. **International Journal for Numerical Methods in Engineering**, v. 22, n. 3, p. 649–670, 1986. Available at: https://doi.org/10.1002/nme.1620220310

SLOAN, S. W. Substepping schemes for the numerical integration of elastoplastic stress-strain relations. **International Journal for Numerical Methods in Engineering**, v. 24, n. 5, p. 893–911, 1987. Available at: https://doi.org/10.1002/nme.1620240505

SLOAN, S. W.; ABBO, A. J.; SHENG, D. Refined explicit integration of elastoplastic models with automatic error control. **Engineering Computations**, v. 18, n. 1/2, p. 121–194, 2001. Available at: https://doi.org/10.1108/02644400110365842

SNOEK, J.; LAROCHELLE, H.; ADAMS, R. P. Practical Bayesian Optimization of Machine Learning Algorithms. **arXiv**: 1206.2944, 2012. Available at: http://arxiv.org/abs/1206.2944

SONG, K.; YAN, F.; DING, T.; GAO, L.; LU, S. A steel property optimization model based on the XGBoost algorithm and improved PSO. **Computational Materials Science**, v. 174, p. 109472, 2020. Available at: https://doi.org/10.1016/j.commatsci.2019.109472

SORJAMAA, A.; HAO, J.; REYHANI, N.; JI, Y.; LENDASSE, A. Methodology for long-term prediction of time series. **Neurocomputing**, v. 70, n. 16–18, p. 2861–2869, 2007. Available at: https://doi.org/10.1016/j.neucom.2006.06.015

SOULÄIMANI, A.; SALAH, N. Ben; SAAD, Y. Enhanced GMRES Acceleration Techniques for some CFD Problems. **International Journal of Computational Fluid Dynamics**, v. 16, n. 1, p. 1–20, 2002. Available at: https://doi.org/10.1080/10618560290003991

SULTANA, N.; HOSSAIN, S. M. Z.; ABUSAAD, M.; ALANBAR, N.; SENAN, Y.; RAZZAK, S. A. Prediction of biodiesel production from microalgal oil using Bayesian optimization algorithm-based machine learning approaches. **Fuel**, v. 309, p. 122184, 2022. Available at: https://doi.org/10.1016/j.fuel.2021.122184

SWOBODA, G.; ICHIKAWA, Y.; DONG, Q.; ZAKI, M. Back analysis of large geotechnical models. **International Journal for Numerical and Analytical Methods in Geomechanics**, v. 23, n. 13, p. 1455–1472, 1999. Available at: https://doi.org/10.1002/(SICI)1096-9853(199911)23:13<1455::AID-NAG33>3.0.CO;2-C

SZABÓ, L. A semi-analytical integration method for J2 flow theory of plasticity with linear isotropic hardening. **Computer Methods in Applied Mechanics and Engineering**, v. 198, n. 27–29, p. 2151–2166, 2009. Available at: https://doi.org/10.1016/j.cma.2009.02.007

TAIEB, S. Ben; BONTEMPI, G.; ATIYA, A. F.; SORJAMAA, A. A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. **Expert Systems with Applications**, v. 39, n. 8, p. 7067–7083, 2012. Available at: https://doi.org/10.1016/j.eswa.2012.01.039

TAIEB, S. Ben; HYNDMAN, R. J. Boosting multi-step autoregressive forecasts. *In*: (E. P. Xing, T. Jebara, Org.). **Proceedings of the 31st International Conference on Machine Learning**. Beijing, China: Proceedings of Machine Learning Research (PMLR), 2014. p. 109–117.

TANAKA, M.; FUJIKAWA, M.; BALZANI, D.; SCHRÖDER, J. Robust numerical calculation of tangent moduli at finite strains based on complex-step derivative approximation and its application to localization analysis. **Computer Methods in Applied Mechanics and Engineering**, v. 269, p. 454–470, 2014. Available at: https://doi.org/10.1016/j.cma.2013.11.005

TAO, X.; PHILLIPS, D. V. A simplified isotropic damage model for concrete under bi-axial stress states. **Cement and Concrete Composites**, v. 27, n. 6, p. 716–726, 2005. Available at: https://doi.org/10.1016/j.cemconcomp.2004.09.017

THORNTON, C.; HUTTER, F.; HOOS, H. H.; LEYTON-BROWN, K. Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms. **arXiv**: 1208.3719, 2012. Available at: http://arxiv.org/abs/1208.3719

TIKHONOV, A. N.; ARSENIN, V. Y. **Solutions of Ill-Posed Problems**. Washington, D.C., USA: V. H. Winston & Sons, 1977.

TRIPOPPOOM, S.; MA, X.; YONG, R.; WU, J.; YU, W.; SEPEHRNOORI, K.; MIAO, J.; LI, N. Assisted history matching in shale gas well using multiple-proxy-based Markov chain Monte Carlo algorithm: The comparison of K-nearest neighbors and neural networks as proxy model. **Fuel**, v. 262, p. 116563, 2020. Available at: https://doi.org/10.1016/j.fuel.2019.116563

TRIPOPPOOM, S.; WANG, X.; LIU, Z.; YU, W.; XIE, H.; SEPEHRNOORI, K.; MIAO, J. Characterizing hydraulic and natural fractures properties in shale oil well in Permian basin using assisted history matching. **Fuel**, v. 275, p. 117950, 2020. Available at: https://doi.org/10.1016/j.fuel.2020.117950

TRIPOPPOOM, S.; XIE, J.; YONG, R.; WU, J.; YU, W.; SEPEHRNOORI, K.; MIAO, J.; CHANG, C.; LI, N. Investigation of different production performances in shale gas wells using assisted history matching: Hydraulic fractures and reservoir characterization from production data. **Fuel**, v. 267, p. 117097, 2020. Available at: https://doi.org/10.1016/j.fuel.2020.117097

TRIPOPPOOM, S.; YU, W.; HUANG, H.; SEPEHRNOORI, K.; SONG, W.; DACHANUWATTANA, S. A practical and efficient iterative history matching workflow for shale gas well coupling multiple objective functions, multiple proxy-based MCMC and EDFM. **Journal of Petroleum Science and Engineering**, v. 176, p. 594–611, 2019. Available at: https://doi.org/10.1016/j.petrol.2019.01.080

VAN HOOF, J.; VANSCHOREN, J. Hyperboost: Hyperparameter Optimization by Gradient Boosting surrogate models. **arXiv**: 2101.02289, 2021. Available at: http://arxiv.org/abs/2101.02289

VASCONCELOS, T. de P.; DE SOUZA, D. A. R. M. A.; MATTOS, C. L. C.; GOMES, J. P. P. No-PASt-BO: Normalized Portfolio Allocation Strategy for Bayesian Optimization. *In*: **2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)**. Portland, USA: IEEE, 2019. p. 561–568. Available at: https://doi.org/10.1109/ICTAI.2019.00084

WANG, J.; SHEN, L.; BI, Y.; LEI, J. Modeling and optimization of a light-duty diesel engine at high altitude with a support vector machine and a genetic algorithm. **Fuel**, v. 285, p. 119137, 2021. Available at: https://doi.org/10.1016/j.fuel.2020.119137

WINKLER, B.; HOFSTETTER, G.; LEHAR, H. Application of a constitutive model for concrete to the analysis of a precast segmental tunnel lining. **International Journal for Numerical and Analytical Methods in Geomechanics**, v. 28, n. 78, p. 797–819, 2004. Available at: https://doi.org/10.1002/nag.362

WINKLER, B.; HOFSTETTER, G.; NIEDERWANGER, G. Experimental verification of a constitutive model for concrete cracking. **Proceedings of the Institution of Mechanical Engineers, Part L: Journal of Materials: Design and Applications**, v. 215, n. 2, p. 75–86, 2001. Available at: https://doi.org/10.1177/146442070121500202

WONG, T.-T.; LUK, W.-S.; HENG, P.-A. Sampling with Hammersley and Halton Points. **Journal of Graphics Tools**, v. 2, n. 2, p. 9–24, 1997. Available at: https://doi.org/10.1080/10867651.1997.10487471

WOSATKO, A.; GENIKOMSOU, A.; PAMIN, J.; POLAK, M. A.; WINNICKI, A. Examination of two regularized damage-plasticity models for concrete with regard to crack closing. **Engineering Fracture Mechanics**, v. 194, p. 190–211, 2018. Available at: https://doi.org/10.1016/j.engfracmech.2018.03.002

WU, J.; CHEN, S.; LIU, X. Efficient hyperparameter optimization through model-based reinforcement learning. **Neurocomputing**, v. 409, p. 381–393, 2020. Available at: https://doi.org/10.1016/j.neucom.2020.06.064

WU, J.; CHEN, X. Y.; ZHANG, H.; XIONG, L. D.; LEI, H.; DENG, S. H. Hyperparameter optimization for machine learning models based on Bayesian optimization. **Journal of Electronic Science and Technology**, v. 17, n. 1, p. 26–40, 2019. Available at: https://doi.org/10.11989/JEST.1674-862X.80904120

WU, J. Y.; LI, J.; FARIA, R. An energy release rate-based plastic-damage model for concrete. **International Journal of Solids and Structures**, v. 43, n. 3–4, p. 583–612, 2006. Available at: https://doi.org/10.1016/j.ijsolstr.2005.05.038

XAVIER, C. R.; SANTOS, E. P. dos; VIEIRA, V. da F.; SANTOS, R. W. dos. Genetic Algorithm for the History Matching Problem. **Procedia Computer Science**, v. 18, p. 946–955, 2013. Available at: https://doi.org/10.1016/j.procs.2013.05.260

XIA, Y.; LIU, C.; LI, Y.; LIU, N. A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. **Expert Systems with Applications**, v. 78, p. 225–241, 2017. Available at: https://doi.org/10.1016/j.eswa.2017.02.017

XIE, Z.; SOUSAMLI, M.; MESSALI, F.; ROTS, J. G. A sub-stepping iterative constitutive model for cyclic cracking-crushing-shearing in masonry interface elements. **Computers & Structures**, v. 257, p. 106654, 2021. Available at: https://doi.org/10.1016/j.compstruc.2021.106654

XUE, P.; JIANG, Y.; ZHOU, Z.; CHEN, X.; FANG, X.; LIU, J. Multi-step ahead forecasting of heat load in district heating systems using machine learning algorithms. **Energy**, v. 188, p. 116085, 2019. Available at: https://doi.org/10.1016/j.energy.2019.116085

YANG, L.; SHAMI, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. **Neurocomputing**, v. 415, p. 295–316, 2020. Available at: https://doi.org/10.1016/j.neucom.2020.07.061

YIN, Z.-Y.; JIN, Y.-F.; SHEN, J. S.; HICHER, P.-Y. Optimization techniques for identifying soil parameters in geotechnical engineering: Comparative study and enhancement. **International Journal for Numerical and Analytical Methods in Geomechanics**, v. 42, n. 1, p. 70–94, 2018. Available at: https://doi.org/10.1002/nag.2714

YIN, Z.-Y.; JIN, Y.-F.; SHEN, S.-L.; HUANG, H.-W. An efficient optimization method for identifying parameters of soft structured clay by an enhanced genetic algorithm and elastic–viscoplastic model. **Acta Geotechnica**, v. 12, n. 4, p. 849–867, 2017. Available at: https://doi.org/10.1007/s11440-016-0486-0

YUAN, G.; ZHANG, M. A three-terms Polak–Ribière–Polyak conjugate gradient algorithm for large-scale nonlinear equations. **Journal of Computational and Applied Mathematics**, v. 286, n. 2012, p. 186–195, 2015. Available at: https://doi.org/10.1016/j.cam.2015.03.014

ZHANG, J.; LI, J. Semi-Implicit Algorithm for Elastoplastic Damage Models Involving Energy Integration. **Advances in Materials Science and Engineering**, v. 2016, p. 1–9, 2016. Available at: https://doi.org/10.1155/2016/5289642

ZHANG, J.; SUN, Y.; SHANG, L.; FENG, Q.; GONG, L.; WU, K. A unified intelligent model for estimating the (gas + n-alkane) interfacial tension based on the eXtreme gradient boosting (XGBoost) trees. **Fuel**, v. 282, p. 118783, 2020. Available at: https://doi.org/10.1016/j.fuel.2020.118783

ZHANG, L. A derivative-free conjugate residual method using secant condition for general large-scale nonlinear equations. **Numerical Algorithms**, v. 83, n. 4, p. 1277–1293, 2020. Available at: https://doi.org/10.1007/s11075-019-00725-7

ZHANG, L.; LI, Z.; LAI, F.; LI, H.; ADENUTSI, C. D.; WANG, K.; YANG, S.; XU, W. Integrated optimization design for horizontal well placement and fracturing in tight oil reservoirs. **Journal of Petroleum Science and Engineering**, v. 178, p. 82–96, 2019. Available at: https://doi.org/10.1016/j.petrol.2019.03.006

ZHANG, L.; ZHOU, W.; LI, D.-H. A descent modified Polak–Ribière–Polyak conjugate gradient method and its global convergence. **IMA Journal of Numerical Analysis**, v. 26, n. 4, p. 629–640, 2006. Available at: https://doi.org/10.1093/imanum/drl016

ZHANG, S.; YIN, S.; WANG, F.; ZHAO, H. Characterization of In Situ Stress State and Joint Properties from Extended Leak-Off Tests in Fractured Reservoirs. **International Journal of Geomechanics**, v. 17, n. 3, p. 04016074, 2017. Available at: https://doi.org/10.1061/(ASCE)GM.1943-5622.0000757

ZHAO, R.; LI, C.; ZHOU, L.; ZHENG, H. A sequential linear complementarity problem for multisurface plasticity. **Applied Mathematical Modelling**, v. 103, p. 557–579, 2022. Available at: https://doi.org/10.1016/j.apm.2021.11.003

ZHENG, H.; CHEN, Q. Dimension extending technique for constitutive integration of plasticity with hardening–softening behaviors. **Computer Methods in Applied Mechanics and Engineering**, v. 394, p. 114833, 2022. Available at: https://doi.org/10.1016/j.cma.2022.114833

ZHENG, H.; ZHANG, T.; WANG, Q. The mixed complementarity problem arising from non-associative plasticity with non-smooth yield surfaces. **Computer Methods in Applied Mechanics and Engineering**, v. 361, 2020. Available at: https://doi.org/10.1016/j.cma.2019.112756

ZHOU, J.; QIU, Y.; ZHU, S.; ARMAGHANI, D. J.; KHANDELWAL, M.; MOHAMAD, E. T. Estimation of the TBM advance rate under hard rock conditions using XGBoost and Bayesian optimization. **Underground Space**, v. 6, n. 5, p. 506–515, 2021. Available at: https://doi.org/10.1016/j.undsp.2020.05.008

ZHOU, W. On the convergence of the modified Levenberg–Marquardt method with a nonmonotone second order Armijo type line search. **Journal of Computational and Applied Mathematics**, v. 239, n. 1, p. 152–161, 2013. Available at: https://doi.org/10.1016/j.cam.2012.09.025

ZHOU, X.; LU, D.; SU, C.; GAO, Z.; DU, X. An unconstrained stress updating algorithm with the line search method for elastoplastic soil models. **Computers and Geotechnics**, v. 143, p. 104592, 2022. Available at: https://doi.org/10.1016/j.compgeo.2021.104592

ZHOU, X.; LU, D.; ZHANG, Y.; DU, X.; RABCZUK, T. An open-source unconstrained stress updating algorithm for the modified Cam-clay model. **Computer Methods in Applied Mechanics and Engineering**, v. 390, p. 114356, 2022. Available at: https://doi.org/10.1016/j.cma.2021.114356

ZIANI, M.; GUYOMARC'H, F. An autoadaptative limited memory Broyden's method to solve systems of nonlinear equations. **Applied Mathematics and Computation**, v. 205, n. 1, p. 202–211, 2008. Available at: https://doi.org/10.1016/j.amc.2008.06.047

Appendix A
Auxiliary equations for elastoplastic model implementation

An elastoplastic model implementation requires some derivatives of yield functions with respect to stress tensor. The first and second derivatives are computed as follows:

$$\frac{\partial f_i}{\partial \boldsymbol{\sigma}} = \frac{\partial f_i}{\partial I_1}\frac{\partial I_1}{\partial \boldsymbol{\sigma}} + \frac{\partial f_i}{\partial J_2}\frac{\partial J_2}{\partial \boldsymbol{\sigma}} + \frac{\partial f_i}{\partial J_3}\frac{\partial J_3}{\partial \boldsymbol{\sigma}} \tag{A.1}$$

$$\begin{aligned}\frac{\partial^2 f_i}{\partial \boldsymbol{\sigma}^2} = {} & \frac{\partial^2 f_i}{\partial {I_1}^2}\frac{\partial I_1}{\partial \boldsymbol{\sigma}}\left(\frac{\partial I_1}{\partial \boldsymbol{\sigma}}\right)^T + \frac{\partial^2 f_i}{\partial {J_2}^2}\frac{\partial J_2}{\partial \boldsymbol{\sigma}}\left(\frac{\partial J_2}{\partial \boldsymbol{\sigma}}\right)^T + \frac{\partial^2 f_i}{\partial {J_3}^2}\frac{\partial J_3}{\partial \boldsymbol{\sigma}}\left(\frac{\partial J_3}{\partial \boldsymbol{\sigma}}\right)^T \\ & + \frac{\partial^2 f_i}{\partial I_1 \partial J_2}\left(\frac{\partial I_1}{\partial \boldsymbol{\sigma}}\left(\frac{\partial J_2}{\partial \boldsymbol{\sigma}}\right)^T + \frac{\partial J_2}{\partial \boldsymbol{\sigma}}\left(\frac{\partial I_1}{\partial \boldsymbol{\sigma}}\right)^T\right) \\ & + \frac{\partial^2 f_i}{\partial I_1 \partial J_3}\left(\frac{\partial I_1}{\partial \boldsymbol{\sigma}}\left(\frac{\partial J_3}{\partial \boldsymbol{\sigma}}\right)^T + \frac{\partial J_3}{\partial \boldsymbol{\sigma}}\left(\frac{\partial I_1}{\partial \boldsymbol{\sigma}}\right)^T\right) \\ & + \frac{\partial^2 f_i}{\partial J_2 \partial J_3}\left(\frac{\partial J_2}{\partial \boldsymbol{\sigma}}\left(\frac{\partial J_3}{\partial \boldsymbol{\sigma}}\right)^T + \frac{\partial J_3}{\partial \boldsymbol{\sigma}}\left(\frac{\partial J_2}{\partial \boldsymbol{\sigma}}\right)^T\right) + \frac{\partial f_i}{\partial J_2}\frac{\partial^2 J_2}{\partial \boldsymbol{\sigma}^2} \\ & + \frac{\partial f_i}{\partial J_3}\frac{\partial^2 J_3}{\partial \boldsymbol{\sigma}^2}\end{aligned} \tag{A.2}$$

where I_1 represents the first invariant of the stress tensor, J_2 represents the second invariant of the deviatoric stress tensor and J_3 represents the third invariant of the deviatoric stress tensor. Crisfield (1997) defined the invariant derivatives present in these equations. These invariants support the geometric representation of any stress state because they are coordinates to represent yield surfaces in stress space. Other representations are possible, such as the stress invariants: $p = I_1/3$, $J = \sqrt{J_2}$ and the Lode angle

$$-\frac{\pi}{6} \leq \theta = \frac{1}{3}\arcsin\left(\frac{-3\sqrt{3}J_3}{2J^3}\right) \leq \frac{\pi}{6} \tag{A.3}$$

Some constraints were adopted to compute these invariants and to avoid singularity in invariant derivatives. These constraints are defined in Equations (A.4) and (A.5):

$$J_2 \geq 10^{-6} \tag{A.4}$$

$$10^{-10} - 1 \leq \frac{-3\sqrt{3}J_3}{2J^3} \leq 1 - 10^{-10} \tag{A.5}$$

Considering the stress invariants J and θ, some invariant derivatives are defined as follow:

$$\frac{\partial J}{\partial J_2} = \frac{1}{2J} \tag{A.6}$$

$$\frac{\partial^2 J}{\partial {J_2}^2} = -\frac{1}{4J^3} \tag{A.7}$$

$$\frac{\partial \theta}{\partial J_2} = \begin{cases} -\dfrac{\tan 3\theta}{2J^2}, & \text{if } J_3 \neq 0 \\ 0, & \text{if } J_3 = 0 \end{cases} \tag{A.8}$$

$$\frac{\partial \theta}{\partial J_3} = \begin{cases} \dfrac{\tan 3\theta}{3J_3}, & \text{if } J_3 \neq 0 \\ 0, & \text{if } J_3 = 0 \end{cases} \tag{A.9}$$

$$\frac{\partial^2 \theta}{\partial {J_2}^2} = -2\left(3J^2\left(\frac{\partial \theta}{\partial J_2}\right)^2 + 5\left(\frac{\partial J}{\partial J_2}\right)^2\right)\frac{\partial \theta}{\partial J_2} \tag{A.10}$$

$$\frac{\partial^2 \theta}{\partial {J_3}^2} = 9J_3\left(\frac{\partial \theta}{\partial J_3}\right)^3 \tag{A.11}$$

$$\frac{\partial^2 \theta}{\partial J_2 \partial J_3} = -6\left(J^2\left(\frac{\partial \theta}{\partial J_2}\right)^2 + \left(\frac{\partial J}{\partial J_2}\right)^2\right)\frac{\partial \theta}{\partial J_3} \tag{A.12}$$

In this work, the model presented by Lester and Sloan (2018), also known as Hyperbolic Generalized Classical yield surface, is employed. This yield surface formulation is described in Appendix B. In order to define the deviatoric section of this yield surface, the parameter Π is defined as

$$\Pi = \alpha \cos\left(\frac{\pi}{6}(2-\gamma) - \frac{1}{3}\arccos(\beta \sin 3\theta)\right) \tag{A.13}$$

where α, β and γ are shape parameters presented by Lagioia and Panteghini (2016) and Lester and Sloan (2018). Each parameter value depends on the desired surface. Some important derivatives are shown below:

$$\frac{\partial \Pi}{\partial \theta} = \frac{\alpha\beta\cos 3\theta}{\sqrt{1-\beta^2\sin^2 3\theta}}\sin\left(\frac{1}{3}\arccos(\beta\sin 3\theta) - \frac{\pi}{6}(2-\gamma)\right) \tag{A.14}$$

$$\frac{\partial^2 \Pi}{\partial \theta^2} = \frac{\beta^2\cos 3\theta}{1-\beta^2\sin^2 3\theta}\left(3\sin 3\theta\frac{\partial \Pi}{\partial \theta} - \Pi\cos 3\theta\right) - 3\tan 3\theta\frac{\partial \Pi}{\partial \theta} \tag{A.15}$$

$$\frac{\partial \Pi}{\partial J_2} = \frac{\partial \Pi}{\partial \theta}\frac{\partial \theta}{\partial J_2} \tag{A.16}$$

$$\frac{\partial \Pi}{\partial J_3} = \frac{\partial \Pi}{\partial \theta}\frac{\partial \theta}{\partial J_3} \tag{A.17}$$

$$\frac{\partial^2 \Pi}{\partial {J_2}^2} = \frac{\partial^2 \Pi}{\partial \theta^2}\left(\frac{\partial \theta}{\partial J_2}\right)^2 + \frac{\partial \Pi}{\partial \theta}\frac{\partial^2 \theta}{\partial {J_2}^2} \tag{A.18}$$

$$\frac{\partial^2 \Pi}{\partial {J_3}^2} = \frac{\partial^2 \Pi}{\partial \theta^2}\left(\frac{\partial \theta}{\partial J_3}\right)^2 + \frac{\partial \Pi}{\partial \theta}\frac{\partial^2 \theta}{\partial {J_3}^2} \tag{A.19}$$

$$\frac{\partial^2 \Pi}{\partial J_2 \partial J_3} = \frac{\partial^2 \Pi}{\partial \theta^2}\frac{\partial \theta}{\partial J_2}\frac{\partial \theta}{\partial J_3} + \frac{\partial \Pi}{\partial \theta}\frac{\partial^2 \theta}{\partial J_2 \partial J_3} \tag{A.20}$$

Appendix B
Capped Hyperbolic Generalized Classical (CHGC) model formulation

This work presents a novel multisurface plasticity model that is used to assess the proposed algorithm. This constitutive model combines the Hyperbolic Generalized Classical yield surface, proposed by Lester and Sloan (2018), and a cap surface adapted from the works by Hofstetter, Simo and Taylor (1993) and Schwer and Murray (1994). The former surface is described in Equation (B.1):

$$f_1(\boldsymbol{\sigma}) = F + \sqrt{a^2 M^2 + J^2 \Pi^2} \tag{B.1}$$

where a is a user-defined parameter that defines the hyperbolic approximation and

$$F = Mp - K \tag{B.2}$$

$$M = \frac{6\sin\phi}{\sqrt{3}(3 - \sin\phi)} \tag{B.3}$$

$$K = \frac{6\cos\phi}{\sqrt{3}(3 - \sin\phi)} c \tag{B.4}$$

Parameters c and ϕ respectively represent the cohesion and friction angle. In order to define the nonassociated flow rule $\boldsymbol{n}_1(\boldsymbol{\sigma}, \boldsymbol{q})$, the friction angle must be replaced by the dilation angle ψ in Equation (B.1), and then Equation (A.1) must be computed. Following the traditional mechanic convention, tension is assumed to be positive in this formulation. The presented model can simulate several perfect elastoplastic criteria, as well as their rounded versions. These rounded models are defined according to parameters a and β (Lagioia and Panteghini, 2016; Lester and Sloan, 2018). They can eliminate "corners" (nonsmooth transitions) of this surface, generating regions of high curvature. The following derivatives are defined to support the computation of Equations (A.1) and (A.2):

$$\frac{\partial f_1}{\partial I_1} = \frac{1}{3} M \tag{B.5}$$

$$\frac{\partial f_1}{\partial J_2} = \frac{\partial f_1}{\partial J}\frac{\partial J}{\partial J_2} + \frac{\partial f_1}{\partial \Pi}\frac{\partial \Pi}{\partial J_2} \tag{B.6}$$

$$\frac{\partial f_1}{\partial J_3} = \frac{\partial f_1}{\partial \Pi}\frac{\partial \Pi}{\partial J_3} \tag{B.7}$$

$$\frac{\partial^2 f_1}{\partial {J_2}^2} = \frac{\partial^2 f_1}{\partial J^2}\left(\frac{\partial J}{\partial J_2}\right)^2 + \frac{\partial^2 f_1}{\partial \Pi^2}\left(\frac{\partial \Pi}{\partial J_2}\right)^2 + 2\frac{\partial^2 f_1}{\partial J \partial \Pi}\frac{\partial J}{\partial J_2}\frac{\partial \Pi}{\partial J_2} + \frac{\partial f_1}{\partial J}\frac{\partial^2 J}{\partial {J_2}^2} + \frac{\partial f_1}{\partial \Pi}\frac{\partial^2 \Pi}{\partial {J_2}^2} \tag{B.8}$$

$$\frac{\partial^2 f_1}{\partial {J_3}^2} = \frac{\partial^2 f_1}{\partial \Pi^2}\left(\frac{\partial \Pi}{\partial J_3}\right)^2 + \frac{\partial f_1}{\partial \Pi}\frac{\partial^2 \Pi}{\partial {J_3}^2} \tag{B.9}$$

$$\frac{\partial^2 f_1}{\partial J_2 \partial J_3} = \left(\frac{\partial^2 f_1}{\partial J \partial \Pi}\frac{\partial J}{\partial J_2} + \frac{\partial^2 f_1}{\partial \Pi^2}\frac{\partial \Pi}{\partial J_2}\right)\frac{\partial \Pi}{\partial J_3} + \frac{\partial f_1}{\partial \Pi}\frac{\partial^2 \Pi}{\partial J_2 \partial J_3} \tag{B.10}$$

$$\frac{\partial^2 f_1}{\partial {I_1}^2} = \frac{\partial^2 f_1}{\partial I_1 \partial J_2} = \frac{\partial^2 f_1}{\partial I_1 \partial J_3} = 0 \tag{B.11}$$

The invariant derivatives presented in Appendix A, as well as the derivatives of f_1 with respect to J and Π, are required to compute Equations (B.5) to (B.11). These derivatives are given as follows:

$$\frac{\partial f_1}{\partial J} = \zeta \Pi J \tag{B.12}$$

$$\frac{\partial f_1}{\partial \Pi} = \zeta J^2 \tag{B.13}$$

$$\frac{\partial^2 f_1}{\partial J^2} = \zeta \Pi (1 - \zeta^2 J^2) \tag{B.14}$$

$$\frac{\partial^2 f_1}{\partial \Pi^2} = \frac{\zeta J^2}{\Pi}(1 - \zeta^2 J^2) \tag{B.15}$$

$$\frac{\partial^2 f_1}{\partial J \partial \Pi} = \zeta J (2 - \zeta^2 J^2) \tag{B.16}$$

in which

$$\zeta = \frac{\Pi}{\sqrt{a^2 M^2 + J^2 \Pi^2}} \tag{B.17}$$

The proposed cap surface is described in Equation (B.18). There are distinct formulations of this yield surface in the literature. In this work, the form presented

by Hofstetter, Simo and Taylor (1993) is combined with Equation (B.1). Thus, the cap surface is given by

$$f_2(\boldsymbol{\sigma}, \boldsymbol{q}) = F_L + \sqrt{a^2 M^2 + J^2 \Pi^2 + \left(\frac{I_1 - L(\kappa)}{R}\right)^2}, \text{ for } X(\kappa) \leq I_1 \leq L(\kappa) \tag{B.18}$$

Considering

$$F_L = \frac{1}{3} M L(\kappa) - K \tag{B.19}$$

The parameter $L(\kappa)$ defines the center of the cap. Additionally, the parameter κ is the internal hardening variable which defines the tensor $\boldsymbol{q}$. The definition of $L(\kappa)$ depends on an initial internal variable value κ_0 that represents the initial position of the cap, according to the following equation:

$$L(\kappa) = \min(\kappa, \kappa_0) \tag{B.20}$$

The hardening behavior of this surface is described by a relation between volumetric plastic strain ε_v^p and the cap limit position $X(\kappa)$. Then, this hardening rule is defined by

$$\varepsilon_v^p = W\left(\exp\left(D\big(X(\kappa) - X(\kappa_0)\big)\right) - 1\right) \tag{B.21}$$

where W and D are material parameters, and

$$X(\kappa) = L(\kappa) - R F_L\big(L(\kappa)\big) \tag{B.22}$$

Figure 43 illustrates a three-dimensional representation of the CHGC model. Note that the proposed cap surface and the failure envelope presented by Lester and Sloan (2018) have high curvature regions.

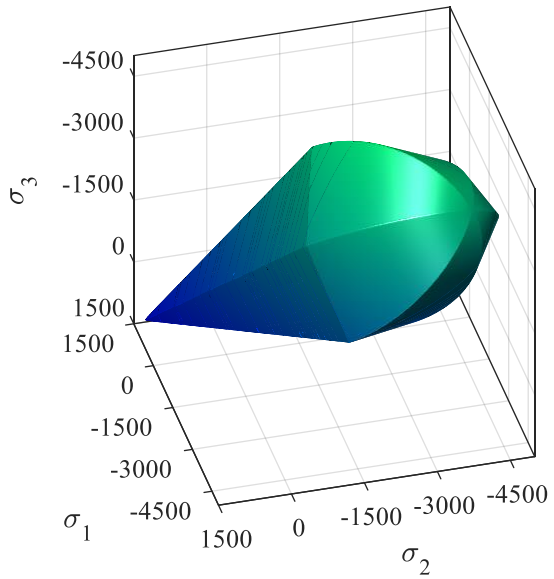

Figure 43 Capped Hyperbolic Generalized Classical (CHGC) surface in principal stress space

The following derivatives are presented to implement the proposed constitutive model:

$$\frac{\partial f_2}{\partial I_1} = \frac{I_1 - L(\kappa)}{R^2\sqrt{a^2M^2 + J^2\Pi^2 + \left(\frac{I_1 - L(\kappa)}{R}\right)^2}} \tag{B.23}$$

$$\frac{\partial f_2}{\partial J_2} = \frac{\partial f_2}{\partial J}\frac{\partial J}{\partial J_2} + \frac{\partial f_2}{\partial \Pi}\frac{\partial \Pi}{\partial J_2} \tag{B.24}$$

$$\frac{\partial f_2}{\partial J_3} = \frac{\partial f_2}{\partial \Pi}\frac{\partial \Pi}{\partial J_3} \tag{B.25}$$

$$\frac{\partial^2 f_2}{\partial {I_1}^2} = \frac{1}{\sqrt{a^2M^2 + J^2\Pi^2 + \left(\frac{I_1 - L(\kappa)}{R}\right)^2}}\left(\frac{1}{R^2} - \left(\frac{\partial f_2}{\partial I_1}\right)^2\right) \tag{B.26}$$

$$\frac{\partial^2 f_2}{\partial {J_2}^2} = \frac{\partial^2 f_2}{\partial J^2}\left(\frac{\partial J}{\partial J_2}\right)^2 + \frac{\partial^2 f_2}{\partial \Pi^2}\left(\frac{\partial \Pi}{\partial J_2}\right)^2 + 2\frac{\partial^2 f_2}{\partial J \partial \Pi}\frac{\partial J}{\partial J_2}\frac{\partial \Pi}{\partial J_2} + \frac{\partial f_2}{\partial J}\frac{\partial^2 J}{\partial {J_2}^2} + \frac{\partial f_2}{\partial \Pi}\frac{\partial^2 \Pi}{\partial {J_2}^2} \tag{B.27}$$

$$\frac{\partial^2 f_2}{\partial {J_3}^2} = \frac{\partial^2 f_2}{\partial \Pi^2}\left(\frac{\partial \Pi}{\partial J_3}\right)^2 + \frac{\partial f_2}{\partial \Pi}\frac{\partial^2 \Pi}{\partial {J_3}^2} \tag{B.28}$$

$$\frac{\partial^2 f_2}{\partial I_1 \partial J_2} = \frac{\partial^2 f_2}{\partial I_1 \partial J}\frac{\partial J}{\partial J_2} + \frac{\partial^2 f_2}{\partial I_1 \partial \Pi}\frac{\partial \Pi}{\partial J_2} \tag{B.29}$$

$$\frac{\partial^2 f_2}{\partial I_1 \partial J_3} = \frac{\partial^2 f_2}{\partial I_1 \partial \Pi}\frac{\partial \Pi}{\partial J_3} \tag{B.30}$$

$$\frac{\partial^2 f_2}{\partial J_2 \partial J_3} = \left(\frac{\partial^2 f_2}{\partial J \partial \Pi}\frac{\partial J}{\partial J_2} + \frac{\partial^2 f_2}{\partial \Pi^2}\frac{\partial \Pi}{\partial J_2}\right)\frac{\partial \Pi}{\partial J_3} + \frac{\partial f_2}{\partial \Pi}\frac{\partial^2 \Pi}{\partial J_2 \partial J_3} \tag{B.31}$$

Derivatives of f_2 with respect to J and Π are necessary to calculate Equations (B.23) to (B.31). Therefore, these derivatives are defined as follows:

$$\frac{\partial f_2}{\partial J} = \xi J \Pi \tag{B.32}$$

$$\frac{\partial f_2}{\partial \Pi} = \xi J^2 \tag{B.33}$$

$$\frac{\partial^2 f_2}{\partial J^2} = \xi \Pi (1 - \xi^2 J^2) \tag{B.34}$$

$$\frac{\partial^2 f_2}{\partial \Pi^2} = \frac{\xi J^2}{\Pi}(1 - \xi^2 J^2) \tag{B.35}$$

$$\frac{\partial^2 f_2}{\partial I_1 \partial J} = -\frac{J\Pi^2\left(I_1 - L(\kappa)\right)}{R^2\left(a^2M^2 + J^2\Pi^2 + \left(\frac{I_1 - L(\kappa)}{R}\right)^2\right)^{\frac{3}{2}}} \tag{B.36}$$

$$\frac{\partial^2 f_2}{\partial I_1 \partial \Pi} = -\frac{J^2\Pi\left(I_1 - L(\kappa)\right)}{R^2\left(a^2M^2 + J^2\Pi^2 + \left(\frac{I_1 - L(\kappa)}{R}\right)^2\right)^{\frac{3}{2}}} \tag{B.37}$$

$$\frac{\partial^2 f_2}{\partial J \partial \Pi} = \xi J (2 - \xi^2 J^2) \tag{B.38}$$

with

$$\xi = \frac{\Pi}{\sqrt{a^2M^2 + J^2\Pi^2 + \left(\frac{I_1 - L(\kappa)}{R}\right)^2}} \tag{B.39}$$

Also, derivatives of f_2 with respect to $\boldsymbol{q}$ are presented in Equations (B.40) and (B.41):

$$\frac{\partial f_2}{\partial \boldsymbol{q}} = \frac{1}{3}M - \frac{\partial f_2}{\partial I_1} \tag{B.40}$$

$$\frac{\partial^2 f_2}{\partial \boldsymbol{\sigma} \partial \boldsymbol{q}} = -\left(\frac{\partial^2 f_2}{\partial {I_1}^2}\frac{\partial I_1}{\partial \boldsymbol{\sigma}} + \frac{\partial^2 f_2}{\partial I_1 \partial J_2}\frac{\partial J_2}{\partial \boldsymbol{\sigma}} + \frac{\partial^2 f_2}{\partial I_1 \partial J_3}\frac{\partial J_3}{\partial \boldsymbol{\sigma}}\right) \tag{B.41}$$

Finally, the hardening behavior of the CHGC model is defined. As the yield function (B.1) simulates perfectly elastoplastic problems, $\boldsymbol{h}_1(\boldsymbol{\sigma}, \boldsymbol{q}) = 0$. Schwer and Murray (1994) clearly presented the definition of function $\boldsymbol{h}_2(\boldsymbol{\sigma}, \boldsymbol{q})$. Hofstetter, Simo and Taylor (1993) pointed out that the hardening law uniquely depends on f_2 providing a nonassociated hardening law and a nonsymmetric tangent modulus. This function and its derivatives are given by

$$\boldsymbol{h}_2 = 3\frac{\dfrac{\partial f_2}{\partial I_1}}{\dfrac{\partial \varepsilon_v^p}{\partial X}\dfrac{\partial X}{\partial \kappa}} \tag{B.42}$$

$$\frac{\partial \boldsymbol{h}_2}{\partial \boldsymbol{\sigma}} = 3\frac{\dfrac{\partial^2 f_2}{\partial {I_1}^2}\dfrac{\partial I_1}{\partial \boldsymbol{\sigma}} + \dfrac{\partial^2 f_2}{\partial I_1 \partial J_2}\dfrac{\partial J_2}{\partial \boldsymbol{\sigma}} + \dfrac{\partial^2 f_2}{\partial I_1 \partial J_3}\dfrac{\partial J_3}{\partial \boldsymbol{\sigma}}}{\dfrac{\partial \varepsilon_v^p}{\partial X}\dfrac{\partial X}{\partial \kappa}} \tag{B.43}$$

$$\frac{\partial \boldsymbol{h}_2}{\partial \boldsymbol{q}} = -3\frac{\dfrac{\partial^2 f_2}{\partial {I_1}^2}\dfrac{\partial \varepsilon_v^p}{\partial X}\dfrac{\partial X}{\partial \kappa} + \dfrac{\partial f_2}{\partial I_1}\dfrac{\partial^2 \varepsilon_v^p}{\partial \kappa^2}}{\left(\dfrac{\partial \varepsilon_v^p}{\partial X}\dfrac{\partial X}{\partial \kappa}\right)^2} \tag{B.44}$$

in which

$$\frac{\partial \varepsilon_v^p}{\partial X} = WD\exp\left(D\big(X(\kappa) - X(\kappa_0)\big)\right) \tag{B.45}$$

$$\frac{\partial^2 \varepsilon_v^p}{\partial X^2} = D\frac{\partial \varepsilon_v^p}{\partial X} \tag{B.46}$$

$$\frac{\partial^2 \varepsilon_v^p}{\partial \kappa^2} = \frac{\partial^2 \varepsilon_v^p}{\partial X^2}\left(\frac{\partial X}{\partial \kappa}\right)^2 \tag{B.47}$$

$$\frac{\partial X}{\partial \kappa} = 1 + \frac{1}{3}MR \tag{B.48}$$

Appendix C
Cap Model formulation

The Cap Model formulation adopted in this work is based on the studies of Hofstetter, Simo and Taylor (1993) and Schwer and Murray (1994). The three yield functions that define this nonsmooth model are given by

$$f_1(\boldsymbol{\sigma}) = \sqrt{2J_2} - F_e(I_1) \tag{C.1}$$

$$f_2(\boldsymbol{\sigma}, \boldsymbol{q}) = \sqrt{2J_2 + \left(\frac{I_1 - L(\kappa)}{R}\right)^2} - F_e\big(L(\kappa)\big), \text{ for } X(\kappa) \leq I_1 \leq L(\kappa) \tag{C.2}$$

$$f_3(\boldsymbol{\sigma}) = I_1 - T \tag{C.3}$$

considering that

$$F_e(I_1) = \alpha - \lambda \exp(\beta I_1) - \theta I_1 \tag{C.4}$$

$$L(\kappa) = \min(\kappa, \kappa_0) \tag{C.5}$$

The center of the cap surface is determined by the parameter $L(\kappa)$, while the internal hardening variable κ defines the tensor $\boldsymbol{q}$. The initial position of the center of the cap surface $L(\kappa_0)$ depends on the initial internal variable κ_0. Regarding the hardening evolution, the following volumetric plastic strain ε_v^p and the cap limit $X(\kappa)$, as shown in Figure 44, are defined:

$$\varepsilon_v^p = W\left(\exp\left(D\big(X(\kappa) - X(\kappa_0)\big)\right) - 1\right) \tag{C.6}$$

$$X(\kappa) = L(\kappa) - RF_e\big(L(\kappa)\big) \tag{C.7}$$

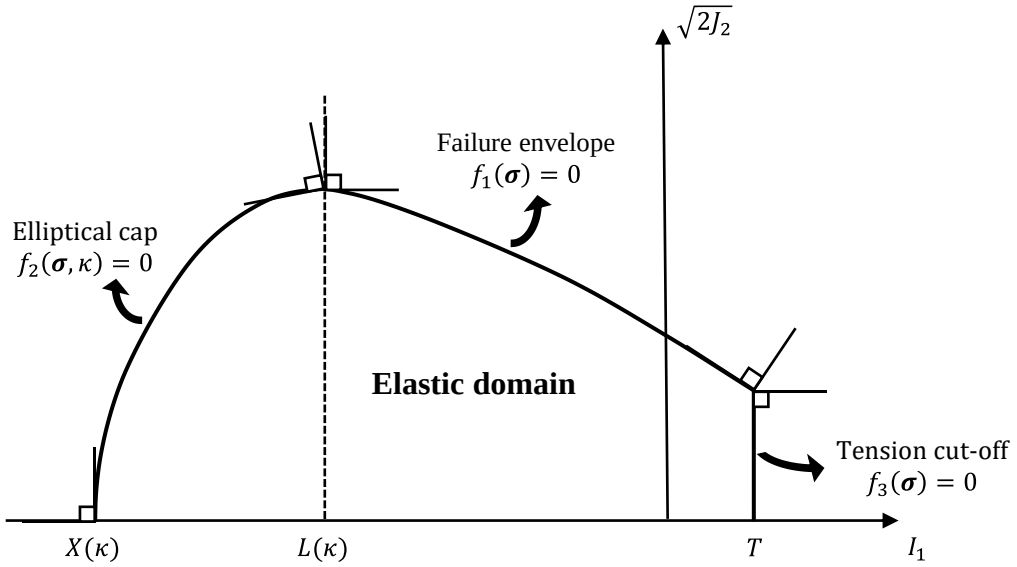

Figure 44 Cap Model in meridional plane

The following derivatives are required to compute Equations (A.1) and (A.2):

$$\frac{\partial f_1}{\partial I_1} = \lambda\beta\exp(\beta I_1) + \theta \tag{C.8}$$

$$\frac{\partial f_1}{\partial J_2} = \frac{1}{\sqrt{2J_2}} \tag{C.9}$$

$$\frac{\partial^2 f_1}{\partial {I_1}^2} = \beta\left(\frac{\partial f_1}{\partial I_1} - \theta\right) \tag{C.10}$$

$$\frac{\partial^2 f_1}{\partial {J_2}^2} = -\left(\frac{\partial f_1}{\partial J_2}\right)^3 \tag{C.11}$$

$$\frac{\partial^2 f_1}{\partial {J_3}^2} = \frac{\partial^2 f_1}{\partial I_1 \partial J_3} = \frac{\partial^2 f_1}{\partial J_2 \partial J_3} = 0 \tag{C.12}$$

$$\frac{\partial f_2}{\partial I_1} = \frac{I_1 - L(\kappa)}{R^2\sqrt{2J_2 + \left(\frac{I_1 - L(\kappa)}{R}\right)^2}} \tag{C.13}$$

$$\frac{\partial f_2}{\partial J_2} = \frac{1}{\sqrt{2J_2 + \left(\frac{I_1 - L(\kappa)}{R}\right)^2}} \tag{C.14}$$

$$\frac{\partial^2 f_2}{\partial {I_1}^2} = \left(\frac{1}{R^2} - \left(\frac{\partial f_2}{\partial I_1}\right)^2\right)\frac{\partial f_2}{\partial J_2} \tag{C.15}$$

$$\frac{\partial^2 f_2}{\partial {J_2}^2} = -\left(\frac{\partial f_2}{\partial J_2}\right)^3 \tag{C.16}$$

$$\frac{\partial^2 f_2}{\partial I_1 \partial J_2} = -\frac{\partial f_2}{\partial I_1}\left(\frac{\partial f_2}{\partial J_2}\right)^2 \tag{C.17}$$

$$\frac{\partial^2 f_2}{\partial {J_3}^2} = \frac{\partial^2 f_2}{\partial I_1 \partial J_3} = \frac{\partial^2 f_2}{\partial J_2 \partial J_3} = 0 \tag{C.18}$$

$$\frac{\partial f_3}{\partial I_1} = 1 \tag{C.19}$$

$$\frac{\partial f_3}{\partial J_2} = 0 \tag{C.20}$$

$$\frac{\partial^2 f_3}{\partial {I_1}^2} = \frac{\partial^2 f_3}{\partial {J_2}^2} = \frac{\partial^2 f_3}{\partial {J_3}^2} = \frac{\partial^2 f_3}{\partial I_1 \partial J_2} = \frac{\partial^2 f_3}{\partial I_1 \partial J_3} = \frac{\partial^2 f_3}{\partial J_2 \partial J_3} = 0 \tag{C.21}$$

Derivatives related to f_2 with respect to $\boldsymbol{q}$ are presented in Equations (C.22) and (C.23). On the other hand, derivatives related to f_1 and f_3 with respect to $\boldsymbol{q}$ are equal to zero.

$$\frac{\partial f_2}{\partial \boldsymbol{q}} = \left.\frac{\partial f_1}{\partial I_1}\right|_{I_1 = L} - \frac{\partial f_2}{\partial I_1} \tag{C.22}$$

$$\frac{\partial^2 f_2}{\partial \boldsymbol{\sigma} \partial \boldsymbol{q}} = -\left(\frac{\partial^2 f_2}{\partial {I_1}^2}\frac{\partial I_1}{\partial \boldsymbol{\sigma}} + \frac{\partial^2 f_2}{\partial I_1 \partial J_2}\frac{\partial J_2}{\partial \boldsymbol{\sigma}}\right) \tag{C.23}$$

In the end, the hardening behavior of the cap surface is defined by Equation (C.24) and its derivatives (C.25) and (C.26). Importantly, the other surfaces behave according to perfect elastoplasticity, $\boldsymbol{h}_1(\boldsymbol{\sigma}, \boldsymbol{q}) = \boldsymbol{h}_3(\boldsymbol{\sigma}, \boldsymbol{q}) = 0$.

$$\boldsymbol{h}_2 = 3\frac{\dfrac{\partial f_2}{\partial I_1}}{\dfrac{\partial \varepsilon_v^p}{\partial X}\dfrac{\partial X}{\partial \kappa}} \tag{C.24}$$

$$\frac{\partial \boldsymbol{h}_2}{\partial \boldsymbol{\sigma}} = 3\frac{\dfrac{\partial^2 f_1}{\partial {I_1}^2}\dfrac{\partial I_1}{\partial \boldsymbol{\sigma}} + \dfrac{\partial^2 f_1}{\partial I_1 \partial J_2}\dfrac{\partial J_2}{\partial \boldsymbol{\sigma}}}{\dfrac{\partial \varepsilon_v^p}{\partial X}\dfrac{\partial X}{\partial \kappa}} \tag{C.25}$$

$$\frac{\partial \boldsymbol{h}_2}{\partial \boldsymbol{q}} = -3\frac{\dfrac{\partial^2 f_2}{\partial {I_1}^2}\dfrac{\partial \varepsilon_v^p}{\partial X}\dfrac{\partial X}{\partial \kappa} + \dfrac{\partial f_2}{\partial I_1}\dfrac{\partial^2 \varepsilon_v^p}{\partial \kappa^2}}{\left(\dfrac{\partial \varepsilon_v^p}{\partial X}\dfrac{\partial X}{\partial \kappa}\right)^2} \tag{C.26}$$

where

$$\frac{\partial \varepsilon_v^p}{\partial X} = WD\exp\big(D(X(\kappa) - X_0)\big) \tag{C.27}$$

$$\frac{\partial^2 \varepsilon_v^p}{\partial X^2} = D\frac{\partial \varepsilon_v^p}{\partial X} \tag{C.28}$$

$$\frac{\partial^2 \varepsilon_v^p}{\partial \kappa^2} = \frac{\partial^2 \varepsilon_v^p}{\partial X^2}\left(\frac{\partial X}{\partial \kappa}\right)^2 + \frac{\partial \varepsilon_v^p}{\partial X}\frac{\partial^2 X}{\partial \kappa^2} \tag{C.29}$$

$$\frac{\partial X}{\partial \kappa} = 1 + R\frac{\partial f_1}{\partial I_1}\bigg|_{I_1 = L} \tag{C.30}$$

$$\frac{\partial^2 X}{\partial \kappa^2} = R\frac{\partial^2 f_1}{\partial {I_1}^2}\bigg|_{I_1 = L} \tag{C.31}$$

Appendix D
Auxiliary equations for Cracking-Crushing Damage (CCD) model definition

This appendix describes several auxiliary expressions for implementing the CCD model. Evidently, the same equations are used in the couple elastoplastic-damage modeling. In addition, further details about the derivation of the tensile and compressive damage evolution laws are presented.

D.1.
Principal stresses computation

In order to compute principal stresses, an analytical strategy is adopted. Since a stress tensor is a 3×3 real symmetric matrix, a closed-form expression of the three real roots of the characteristic equation of the matrix can be used. François Viète (1540-1603) derived a trigonometric solution for cubic equations with real roots that is adopted in this work. In this scenario, the maximum and minimum principal stresses are presented in Equations (D.1) and (D.2) using some stress invariants. Qi et al. (2020) presented a similar approach to compute maximum principal stresses and strains in their damage model.

$$\sigma_{max} = t_{max} + p \tag{D.1}$$

$$\sigma_{min} = t_{min} + p \tag{D.2}$$

where

$$p = \frac{I_1}{3} \tag{D.3}$$

$$t_{max} = \left(\cos\theta - \frac{1}{\sqrt{3}}\sin\theta\right)\sqrt{J_2} \tag{D.4}$$

$$t_{min} = -\left(\cos\theta + \frac{1}{\sqrt{3}}\sin\theta\right)\sqrt{J_2} \tag{D.5}$$

To compute the intermediate principal stress, let $\sigma_{int} = p - t_{max} - t_{min}$. Lode angle θ can be computed using the second and third invariants of the deviatoric effective stress tensor J_2 and J_3. This angle is defined as

$$-\frac{\pi}{6} \leq \theta = \frac{1}{3}\arcsin\left(\frac{-3\sqrt{3}J_3}{2J_2^{3/2}}\right) \leq \frac{\pi}{6} \tag{D.6}$$

D.2. Derivation of the tensile damage evolution law

Based on analytical expressions for stress-strain responses, damage evolution laws can be derived to ensure that a damage model comply this behavior. Therefore, the derivation of the tensile and compressive damage evolution laws is presented in this appendix. Firstly, for the tensile behavior, the stress-strain relation adopted in this work is given by

$$\sigma = \begin{cases} E\varepsilon, & \text{if } \varepsilon < \varepsilon_{t0} \\ f_{t0}exp\big(-a_t(\varepsilon - \varepsilon_{t0})\big), & \text{if } \varepsilon \geq \varepsilon_{t0} \end{cases} \tag{D.7}$$

where σ and ε are respectively uniaxial stress and strain; ε_{t0} denotes the strain at which the damage process begins; and a_t is a material parameter. Notably, Equation (8.11) yields $\varepsilon_t^{eq} = r_t = \varepsilon$ for a uniaxial tensile response without unloading. In addition, $g_c(r_c) = 0$ for a uniaxial tensile test. In this scenario, Equations (8.7) and (D.7) can be manipulated as follows for the case of $\varepsilon > \varepsilon_{t0}$:

$$\begin{aligned} \sigma = Er_{t0}\exp\big(-a_t(r_t - r_{t0})\big) &= \frac{r_{t0}}{r_t}\exp\big(-a_t(r_t - r_{t0})\big)E\varepsilon \\ &= \left(1 - \left(1 - \left(1 - \frac{r_{t0}}{r_t}\exp\big(-a_t(r_t - r_{t0})\big)\right)\right)\right)E\varepsilon \\ &= \Big(1 - \big(1 - \big(1 - g_t(r_t)\big)\big(1 - g_c(r_c)\big)\big)\Big)E\varepsilon \\ &= \big(1 - g(r_t, r_c)\big)E\varepsilon = (1 - D)E\varepsilon \end{aligned} \tag{D.8}$$

Comparing the equations in (D.8) and $g_c(r_c) = 0$, $g_t(r_t)$ takes the form presented in Equation (8.21).

D.3.
Derivation of the compressive damage evolution law

Regarding the compressive behavior, the uniaxial response of concrete is defined in this work according to

$$\sigma = \begin{cases} E\varepsilon, & \text{if } \varepsilon < \varepsilon_{c0} \\ (1-t_c)^2 f_{c0} + (2-t_c)t_c f_{cm}, & \text{if } \varepsilon_{c0} \le \varepsilon < \varepsilon_{cm} \\ f_{cm}\exp(-d_c(\varepsilon-\varepsilon_{cm})^2), & \text{if } \varepsilon \ge \varepsilon_{cm} \end{cases} \tag{D.9}$$

in which $\varepsilon \geq 0$, $\varepsilon_{c0} = r_{c0}$, $\varepsilon_{cm} = r_{cm}$, d_c is a material parameter, and

$$t_c = \begin{cases} \dfrac{-b_c + \sqrt{(b_c)^2 - 4a_c(\varepsilon_{c0}-\varepsilon)}}{2a_c}, & \text{if } a_c \neq 0 \\ \dfrac{\varepsilon - \varepsilon_{c0}}{b_c}, & \text{if } a_c = 0 \end{cases} \tag{D.10}$$

$$a_c = \varepsilon_{c0} - 2\varepsilon_{ci} + \varepsilon_{cm} \tag{D.11}$$

$$b_c = 2(\varepsilon_{ci} - \varepsilon_{c0}) \tag{D.12}$$

$$\varepsilon_{ci} = r_{ci} = \frac{f_{cm}}{E} \tag{D.13}$$

Importantly, Equation (8.12) leads to $\varepsilon_c^{eq} = r_c = \varepsilon$ for a uniaxial response without unloading. Moreover, $g_t(r_t) = 0$ for a uniaxial compressive test. Therefore, using Equations (8.7) and (D.13), the expression related to the nonlinear ascending branch in (D.9) can be manipulated as follows:

$$\begin{aligned} \sigma &= (1-t_c)^2 E r_{c0} + (2-t_c)t_c E r_{ci} \\ &= \left(\frac{r_{c0}}{r_c}(1-t_c)^2 + \frac{r_{ci}}{r_c}(2-t_c)t_c\right) E\varepsilon \\ &= \left(1 - \left(1 - \left(1 - \frac{r_{c0}}{r_c}(1-t_c)^2 - \frac{r_{ci}}{r_c}(2-t_c)t_c\right)\right)\right) E\varepsilon \\ &= \Big(1 - \big(1 - \big(1-g_t(r_t)\big)\big(1-g_c(r_c)\big)\big)\Big) E\varepsilon \\ &= \big(1 - g(r_t, r_c)\big) E\varepsilon = (1-D)E\varepsilon \end{aligned} \tag{D.14}$$

Considering the descending branch in (D.9) and Equation (D.13), the following manipulation is obtained:

$$\sigma = Er_{ci}\exp(-d_c(r_c - r_{cm})^2) = \frac{r_{ci}}{r_c}\exp(-d_c(r_c - r_{cm})^2)E\varepsilon$$
$$= \left(1 - \left(1 - \left(1 - \frac{r_{ci}}{r_c}\exp(-d_c(r_c - r_{cm})^2)\right)\right)\right)E\varepsilon$$
$$= \Big(1 - \big(1 - \big(1 - g_t(r_t)\big)\big(1 - g_c(r_c)\big)\big)\Big)E\varepsilon$$
$$= \big(1 - g(r_t, r_c)\big)E\varepsilon = (1 - D)E\varepsilon \tag{D.15}$$

Based on the equations in (D.14), (D.15) and $g_t(r_t) = 0$, $g_c(r_c)$ takes the form presented in Equation (8.34).

D.4. Derivatives related to material tangent modulus computation

To compute the material tangent modulus, several derivatives of the constitutive equations defined in this work are required. In this scenario, these derivatives are established in this appendix. Concerning Equation (8.43), the following expression can be obtained from Equation (8.19):

$$\frac{dg}{dr_ℵ} = \frac{dg}{dg_ℵ}\frac{dg_ℵ}{dr_ℵ} \tag{D.16}$$

Considering Equations (8.19), (8.21) and (8.34), the following derivatives are presented in order to define Equation (D.16):

$$\frac{dg}{dg_t} = 1 - g_c \tag{D.17}$$

$$\frac{dg}{dg_c} = 1 - g_t \tag{D.18}$$

$$\frac{dg_t}{dr_t} = (1 - g_t)\left(\frac{1}{r_t} + a_t\right) \tag{D.19}$$

$$\frac{dg_c}{dr_c} = \begin{cases} \dfrac{1}{r_c}\left(1 - g_c - \dfrac{2(r_{ci} - r_{c0})(1 - t_c)}{2a_c t_c + b_c}\right), & \text{if } r_{c0} \le r_c < r_{cm} \\ (1 - g_c)\left(\dfrac{1}{r_c} + 2d_c(r_c - r_{cm})\right), & \text{if } r_c \ge r_{cm} \end{cases} \tag{D.20}$$

To calculate Equation (8.48), the derivatives expressed in Equations (D.21) to (D.26) are shown:

$$\frac{\partial \varphi_{\aleph}}{\partial \boldsymbol{\varepsilon}^{e}} = \frac{\partial \varepsilon_{\aleph}^{eq}}{\partial \boldsymbol{\varepsilon}^{e}} = \left(\frac{d\overline{\boldsymbol{\sigma}}}{d\boldsymbol{\varepsilon}^{e}}\right)^{T} \frac{\partial \varepsilon_{\aleph}^{eq}}{\partial \overline{\boldsymbol{\sigma}}} = \boldsymbol{D}\frac{\partial \varepsilon_{\aleph}^{eq}}{\partial \overline{\boldsymbol{\sigma}}} \tag{D.21}$$

$$\frac{\partial \varphi_{\aleph}}{\partial r_{\aleph}} = -1 \tag{D.22}$$

$$\frac{\partial \varepsilon_{t}^{eq}}{\partial \overline{\boldsymbol{\sigma}}} = \frac{1}{E}\left(\alpha_t \frac{\partial \bar{I}_1}{\partial \overline{\boldsymbol{\sigma}}} + \frac{\beta_t}{2}\sqrt{\frac{3}{\bar{J}_2}}\frac{\partial \bar{J}_2}{\partial \overline{\boldsymbol{\sigma}}} + \gamma_t H(\bar{\sigma}_{max})\frac{\partial \bar{\sigma}_{max}}{\partial \overline{\boldsymbol{\sigma}}}\right) \tag{D.23}$$

$$\frac{\partial \varepsilon_{c}^{eq}}{\partial \overline{\boldsymbol{\sigma}}} = \frac{1}{(1-\alpha_c)E}\left(\alpha_c \frac{\partial \bar{I}_1}{\partial \overline{\boldsymbol{\sigma}}} + \frac{\beta_c}{2}\sqrt{\frac{3}{\bar{J}_2}}\frac{\partial \bar{J}_2}{\partial \overline{\boldsymbol{\sigma}}} + \gamma_c H(-\bar{\sigma}_{max})\frac{\partial \bar{\sigma}_{max}}{\partial \overline{\boldsymbol{\sigma}}}\right) \tag{D.24}$$

$$\frac{\partial \bar{\sigma}_{max}}{\partial \overline{\boldsymbol{\sigma}}} = \frac{1}{3}\frac{\partial \bar{I}_1}{\partial \overline{\boldsymbol{\sigma}}} + \frac{\bar{t}_{max}}{2\bar{J}_2}\frac{\partial \bar{J}_2}{\partial \overline{\boldsymbol{\sigma}}} + \frac{\bar{t}_{max} + 2\bar{t}_{min}}{\sqrt{3}}\frac{\partial \bar{\theta}}{\partial \overline{\boldsymbol{\sigma}}} \tag{D.25}$$

$$\frac{\partial \bar{\theta}}{\partial \overline{\boldsymbol{\sigma}}} = \frac{\partial \bar{\theta}}{\partial \bar{J}_2}\frac{\partial \bar{J}_2}{\partial \overline{\boldsymbol{\sigma}}} + \frac{\partial \bar{\theta}}{\partial \bar{J}_3}\frac{\partial \bar{J}_3}{\partial \overline{\boldsymbol{\sigma}}} \tag{D.26}$$

The invariant gradients $\frac{\partial \bar{I}_1}{\partial \overline{\boldsymbol{\sigma}}}$, $\frac{\partial \bar{J}_2}{\partial \overline{\boldsymbol{\sigma}}}$, $\frac{\partial \bar{J}_3}{\partial \overline{\boldsymbol{\sigma}}}$, $\frac{\partial \bar{\theta}}{\partial \bar{J}_2}$, and $\frac{\partial \bar{\theta}}{\partial \bar{J}_3}$ are the same used in plasticity frameworks and are presented in Appendix A.

Appendix E
Parameter identification procedure

Defining material parameters of numerical simulations from experimental tests demands an appropriate approach due to potential computational cost. Specifically, certain examples related to the C3EPD model require computation of parameters associated with elastoplasticity. Therefore, this appendix discusses a soft computing-based strategy to estimate this parameter using artificial intelligence techniques. The presented formulation is based on the work of Abreu, Mejia and Roehl (2022b) and Abreu et al. (2023b).

E.1.
Inverse analysis

The parameter identification of mathematical models from a set of observations is an inverse problem. This type of problem emerges in opposition to forward problems, in which their parameters are well-known, and the model response can be directly predicted. Inverse problems are typically ill-posed. Therefore, the uniqueness and existence of the solution are not guaranteed, and the problems may be unstable when subject to perturbations in the initial and boundary conditions (Tikhonov and Arsenin, 1977). In this scenario, inverse problems are cumbersome classical applications.

If the inversion process does not successfully define an acceptable match between observed and predicted data, the selected model may be an inadequate candidate for the problem, and consequently, a new model is required (Knabe et al., 2013). The trial-and-error technique is commonly applied in solving inverse problems. However, this process demands high human effort. The inverse analysis provides several approaches to cope with inverse problems in an automatic and rational scheme.

The inversion of model equations (inverse approach) is a classical strategy to solve inverse problems. Nevertheless, this procedure can be unfeasible for complex

and nonlinear models, mainly when closed-form solutions are unavailable (Knabe et al., 2013). Therefore, one can adopt optimization methods to overcome this issue. This approach aims to obtain a solution that minimizes an error function (direct approach). This function directly depends on the observed and the predicted data. Since this work targets obtaining material parameters for a nonlinear constitutive model, the proposed methodology adopts the direct approach using a genetic algorithm developed by Abreu (2019) and Abreu, Mejia and Roehl (2022b).

Based on Knabe et al. (2013), the parameter identification problem via direct approach includes the following stages: (1) define the observed data, usually acquired from an experimental test; (2) define a mathematical model, as well as its input data (such as finite element mesh and known parameters) and initial and boundary conditions; (3) define the problem domain $\boldsymbol{\Omega}$ (complex models may require nonlinear constraints) and/or an initial guess for the parameters (according to the adopted optimization algorithm); (4) define an appropriate objective function which measures an error between observed and predicted data; (5) execute an optimization method; and (6) evaluate the suitability of the mathematical model for the calculated parameters.

In this work, the inverse problems are formulated in terms of the following unconstrained optimization problem:

$$\min_{\boldsymbol{x}\in\boldsymbol{\Omega}} F(\boldsymbol{x}) \tag{E.1}$$

in which $\boldsymbol{x}$ denotes a set of parameters to be identified and $F(\boldsymbol{x})$ represents the objective function related to the inverse problem. In this scenario, the nonlinear least square objective function is adopted here, as described in the following equation:

$$F(\boldsymbol{x}) = \boldsymbol{r}(\boldsymbol{x})^T \boldsymbol{r}(\boldsymbol{x}) \tag{E.2}$$

where

$$\boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{y}_{obs} - \boldsymbol{y}_{pred}(\boldsymbol{x}) \tag{E.3}$$

in which $\boldsymbol{y}_{obs}$ represents the observed data and $\boldsymbol{y}_{pred}$ represents the model outcome (predicted data). This approach is widely applied in several curve-fitting problems.

Swoboda et al. (1999), Le, Fatahi and Khabbaz (2015) and Jiang et al. (2018) adopted this function in their inverse analysis problems.

Several engineering applications compute the predicted data based on finite element models. However, these mathematical models frequently have a high computational cost. An approximate solution can be a good alternative since it can reduce the computational time required in the optimization procedure. Therefore, $\boldsymbol{y}_{pred}(\boldsymbol{x})$ is computed in this work using a proxy model, considering a fitness approximation strategy (Jin, 2005).

For a generic parameter identification task, Figure 45 shows a flowchart of the inverse analysis process adopted in this work. The predicted data is defined by a previously built machine learning model, whereas the optimization method is a genetic algorithm. This work adopts the real-coded genetic algorithm (Abreu, 2019; Abreu, Mejia and Roehl, 2022b) presented in this appendix to identify material parameters, solving problem (E.1).

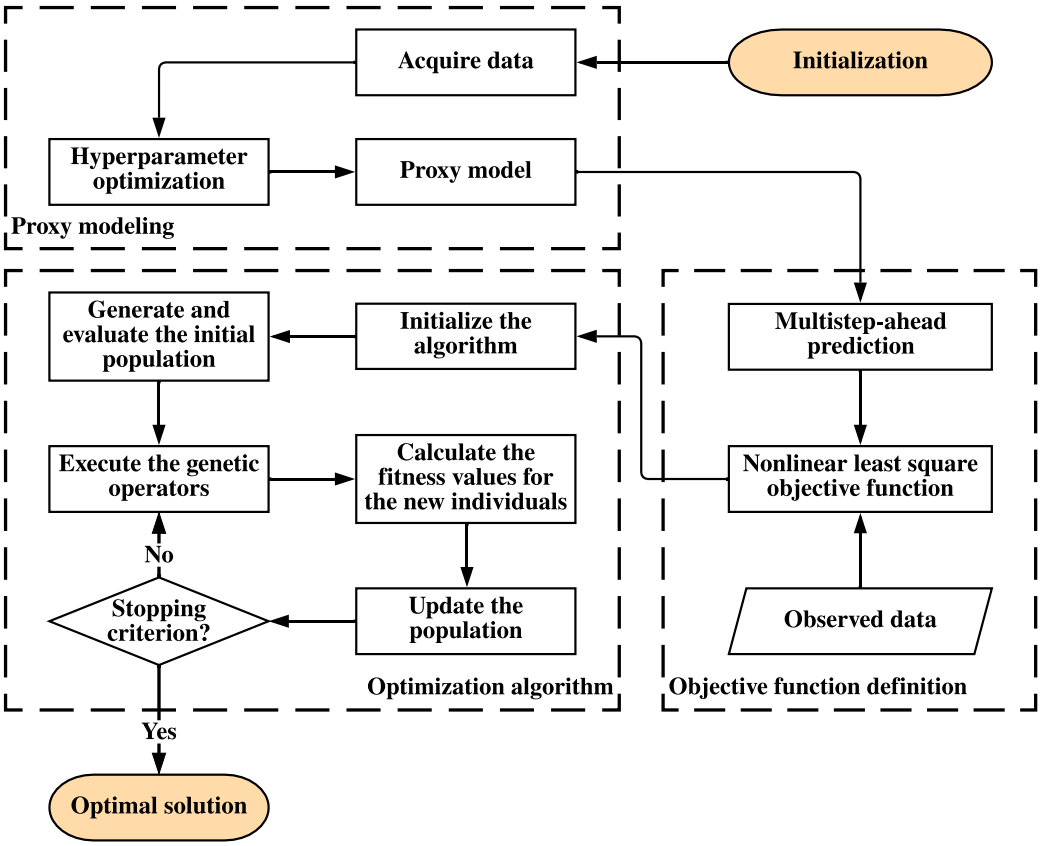

Figure 45 Flowchart of the adopted parameter identification procedure

The presented inverse analysis methodology was implemented in Minerva, a program written in Python and designed to solve parameter identification problems for user-defined mathematical models. Minerva is available on Eras Portal (https://eras.tecgraf.puc-rio.br), a cooperative environment developed by Tecgraf Institute/PUC-Rio to assist researchers who develop projects in geomechanics (Lima et al., 2018).

From the parameter identification perspective, artificial intelligence techniques are extensively employed. There are two key contexts for their use in such applications: proxy modeling (Zhang et al., 2017; de Simone, Souza and Roehl, 2019; Tripoppoom et al., 2019; Tripoppoom, Ma, et al., 2020; Tripoppoom, Wang, et al., 2020; Tripoppoom, Xie, et al., 2020) and optimization (Xavier et al., 2013; Yin et al., 2018; Jin and Yin, 2020). Machine learning algorithms and stochastic optimization are vastly applied in those tasks. Therefore, this work adopts the gradient boosting machine and a genetic algorithm, as described in the following sections.

E.2. Data acquisition

Since the machine learning modeling provides data-driven models to perform regression and classification tasks, its preliminary stage is based on data acquisition. Sample points of the original mathematical model must be generated to build a proxy model using machine learning concepts. A dataset of the input variables must be defined, having random sampling as a common approach (de Simone, Souza and Roehl, 2019; Rao, Zhao and Deng, 2020). Thus, the results of the mathematical model are collected. However, there are other valuable techniques to generate datasets. Design of experiment methods are widely employed in experimental (Wang et al., 2021; Sultana et al., 2022) and computational (Deshpande et al., 2011; Aulia et al., 2019) scenarios. On the other hand, quasi-random sequences quickly and uniformly produce sample points covering the domain of the problem. Low-discrepancy sequences, such as the Halton and the Hammersley, are quasi-random sampling methods commonly used in the quasi-Monte Carlo method (Evans and Swartz, 2000). In addition, the Halton sequence is helpful for incremental sampling (Wong, Luk and Heng, 1997).

Wong, Luk and Heng (1997) studied the Halton and the Hammersley sequences, which are deterministic formulae with roughly similar theoretical bases. These formulae generate uniformly distributed and apparently stochastic sampling patterns. Through experimental results, the authors showed that the Halton and Hammersley sequences produced good sampling patterns, especially the second one. On the other hand, Evans and Swartz (2000) highlighted that these sequences should not be used in high-dimensional applications. As a rule of thumb, ten is the limit number of dimensions (Head et al., 2021).

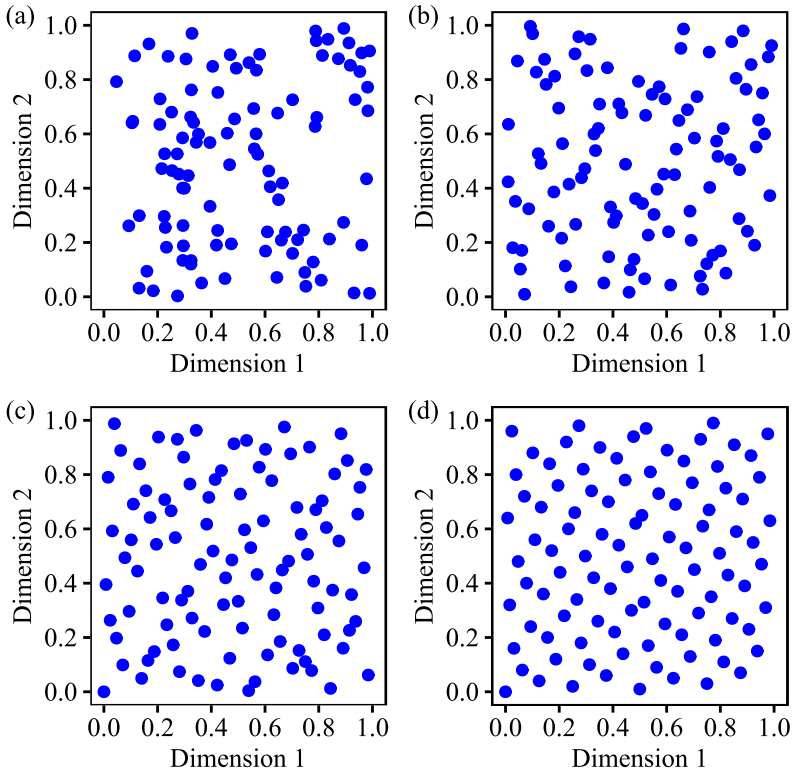

Figure 46 Sample points from four sampling methods: (a) random sampling, (b) Latin hypercube sampling, (c) Halton sequence, and (d) Hammersley sequence

Latin hypercube sampling (McKay, Beckman and Conover, 1979) is another well-known technique extensively adopted in proxy modeling (Riazi et al., 2016; Tripoppoom et al., 2019; Zhang et al., 2019; Kohler et al., 2020; Tripoppoom, Ma, et al., 2020; Tripoppoom, Wang, et al., 2020; Tripoppoom, Xie, et al., 2020). The method involves generating sample points considering a stratified domain which improves their distribution in the problem domain. Importantly, Latin hypercube sampling is a stochastic sampling method. Since the Halton and the Hammersley sequences produce similar sample points, the latter and the Latin hypercube sampling were utilized to respectively generate the training and test datasets

required in the machine learning modeling. Furthermore, the Halton sequence was applied in the sampling process related to the assumed hyperparameter optimization algorithm. Figure 46 compares the sampling methods mentioned above.

E.3. Gradient boosting machine

In regression analysis, a machine learning model establishes a mapping between input and output variables. In this sense, the gradient boosting machine is the selected method to build the proxy model of finite element simulations. In regression tasks, this method usually consists of regression trees, which provide predictions combining nested conditional statements (Kuhn and Johnson, 2013). Therefore, the input variable space is divided into non-overlapping regions called leaf nodes (James et al., 2013; Kuhn and Johnson, 2013). All data points belonging to a specific region have the same predicted output (leaf weight), usually the average outcome of the training data in the leaf node (James et al., 2013). These models can effectively deal with several types of input variables without any data scaling process (Kuhn and Johnson, 2013).

Hyperrectangular subspaces are widely adopted to define these leaf nodes for their simplicity and interpretability (James et al., 2013). The algorithm presented by Breiman et al. (1984) is extensively applied to split the input variable space into several subspaces. This process, also known as recursive partitioning, begins at the node representing the entire input variable space, the root node. This node is divided into two branches, considering the input variable and the split value that typically minimizes the sum of squared errors. Identifying the best input variable and split value can be a fast task, mainly when the number of input variables is not too large (James et al., 2013). Then, this splitting process is performed in each new subspace, transforming leaf nodes into internal nodes and increasing the tree depth (the longest path length from the root node to a leaf node). In practice, the process continues until the leaf nodes reach a certain number of sample points (Kuhn and Johnson, 2013).

In general, regression trees adequately predict the training data and produce poor predictions for the test dataset (James et al., 2013). In addition, regression trees have notable limitations: model instability (minor variations in the training data can

radically alter the tree structure) and suboptimal predictive performance (since the tree is defined by hyperrectangular subspaces that can yield substantial prediction error in relation to other machine learning models) (Kuhn and Johnson, 2013). The pruning approach (Breiman et al., 1984) is an attractive process to reduce the tree structure and, consequently, tree complexity. Nonetheless, the regression trees are not usually competitive with the best machine learning algorithms (James et al., 2013). In this context, the boosting approach is widely utilized since it combines multiple trees to achieve high prediction accuracy.

Boosting models work by sequentially adding predictive models, each utilizing information from previously trained ones (James et al., 2013). The general idea is to combine weak models to build a strong one (Kuhn and Johnson, 2013). Gradient boosted regression trees are built using regression trees (Chen and Guestrin, 2016). However, boosting is not restricted to tree-based models (James et al., 2013). This work develops the proxy modeling through XGBoost, an open-source library that provides a regularized gradient boosting system (Chen and Guestrin, 2016). XGBoost is a robust and efficient implementation extensively utilized in machine learning competitions (Chen and Guestrin, 2016; Xia et al., 2017; Song et al., 2020; Wu, Chen and Liu, 2020; Shi et al., 2021) and engineering applications (Song et al., 2020; Gu et al., 2021; Koc, Ekmekcioğlu and Gurgun, 2021; Zhou et al., 2021).

The presented mathematical description is based on Chen and Guestrin (2016). A gradient boosted regression tree f takes the following form:

$$f(\boldsymbol{x}_i) = \hat{y}_i(\boldsymbol{x}_i) = f_0(\boldsymbol{x}_i) + \eta \sum_{k=1}^{K} f_k(\boldsymbol{x}_i) \tag{E.4}$$

in which $\hat{y}_i$ denotes a predicted value for a set of input variables $\boldsymbol{x}_i$, K represents to the number of regression trees in the gradient boosting machine, f_0 corresponds an initial prediction (usually $f_0(\boldsymbol{x}_i) = 0.5$), f_k for $1 \leq k \leq K$ corresponds an independent regression tree, and η is the learning rate. The following loss function ℓ defines the optimization problem required to build the set of regression trees:

$$\ell(y_i, \hat{y}_i) = \frac{1}{2}(y_i - \hat{y}_i)^2 \tag{E.5}$$

where y_i is the output to be learned by the machine learning model. Thus, the objective function $\mathcal{L}(\boldsymbol{y}, \hat{\boldsymbol{y}})$ of the problem is defined according to Equation (E.6):

$$\mathcal{L}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{E.6}$$

in which n is the number of sample points in the training dataset and Ω is a regularization term that penalizes the complexity of the trees. This term is given by

$$\Omega(f) = \gamma T + \alpha \|w\|_1 + \frac{1}{2}\lambda \|w\|_2^2 \tag{E.7}$$

where $\|\cdot\|_1$ and $\|\cdot\|_2$ respectively denote the Manhattan and the Euclidian norms; T is the number of leaf nodes in the tree f; w represents the leaf weights of the tree f; and γ, α and λ are regularization parameters which are hyperparameters of the gradient boosting machine.

Considering conventional optimization methods, Equation (E.6) cannot be minimized since it includes functions as parameters. In this sense, the model is built in an additive process. Considering a number k of trees, a regression tree f_k needs to be added in the sequence of trees to minimize the following equation:

$$\mathcal{L}_k(\boldsymbol{y}, \hat{\boldsymbol{y}}_{k-1}) = \sum_{i=1}^{n} \ell\left(y_i, \hat{y}_{i,k}\right) + \sum_{i=1}^{k} \Omega(f_k) \tag{E.8}$$

in which

$$\hat{y}_{i,k} = \hat{y}_{i,k-1} + f_k(\boldsymbol{x}_i) \tag{E.9}$$

In this sense, f_k is greedily included, improving the model sequentially. The Newton-Raphson method is then applied to define the tree inserted in each iteration. According to this method, the descent direction $\boldsymbol{p}_k$ is given by Equation (E.10). Conceptually, this direction corrects the predecessor regression trees. Parameter η theoretically works as a step size of the optimization method, being typically constant and usually assuming a small value in order to prevent overfitting (Friedman, 2001).

$$\boldsymbol{p}_k = -\left(\frac{\partial^2 \mathcal{L}_k(\boldsymbol{y}, \widehat{\boldsymbol{y}}_{k-1})}{\partial(\widehat{\boldsymbol{y}}_{k-1})^2}\right)^{-1} \frac{\partial \mathcal{L}_k(\boldsymbol{y}, \widehat{\boldsymbol{y}}_{k-1})}{\partial \widehat{\boldsymbol{y}}_{k-1}} \tag{E.10}$$

Chen and Guestrin (2016) presented optimal strategies to train robust gradient boosting machines efficiently. These processes are related to additional hyperparameters that need to be optimized. In this sense, Table 13 describes all hyperparameters studied in this study, as well as their attributes and adopted domain.

Table 13 Description of the XGBoost hyperparameters, their domains and distributions

Hyperparameter	Symbol	Data type	Description	Upper bound	Lower bound	Distribution
n_estimators	K	Integer	Number of regression trees.	1500	50	Uniform
max_depth	-	Integer	Maximum tree depth.	25	3	Uniform
learning_rate	η	Real	Learning rate.	10^{-1}	10^{-3}	Log-uniform
min_child_weight	-	Real	Minimum sum of the second derivative of the loss function for the sample points in a node.	15	1	Uniform
subsample	-	Real	Ratio between the number of training sample points for each boosting iteration and the total number of training sample points.	1	0.5	Uniform
colsample_bytree	-	Real	Ratio between the number of input variables for constructing each tree and the total number of input variables.	1	0.5	Uniform
colsample_bylevel	-	Real	Ratio between the number of input variables for each level and the total number of input variables.	1	0.5	Uniform
gamma	γ	Real	Complexity parameter.	10^2	10^{-3}	Log-uniform
reg_alpha	α	Real	L_1 regularization term.	10^2	10^{-3}	Log-uniform
reg_lambda	λ	Real	L_2 regularization term.	10^2	10^{-3}	Log-uniform

E.4.
Multistep-ahead prediction

The proposed framework aims to identify parameters based on force-displacement curves computed according to the structural behavior. In practical terms, the rope of the displacements is comparable to the time. In this scenario, these curves are handled as time series. As this work models such curves based on a gradient boosting machine, this machine learning model must accurately predict time series for a long horizon. The trained models must consider an appropriate approach to perform time series prediction since this is a well-known challenging task (Sorjamaa et al., 2007). In this sense, predicting time series values involves calculating current values using previous data from the series (Sorjamaa et al., 2007; Taieb et al., 2012).

Time series modeling can be used to predict the next time instant (one-step-ahead prediction) or multiple time instants (multistep-ahead prediction) (Liu and Zio, 2017). In the multistep-ahead prediction context, Taieb et al. (2012) described several techniques to forecast time series that can be applied to any machine learning model. Among these approaches, two techniques can be highlighted: the direct and the recursive. On the one hand, the direct strategy predicts each time step using distinct models. Therefore, all models have the same inputs: the known current and the previous data. On the other hand, the recursive strategy involves using predicted values as model input in the following prediction since history data (moving window) is required.

The time series prediction faces difficulties with the accumulation of errors, accuracy deterioration, and increased uncertainty since predicted values are used as inputs (Cheng et al., 2006; Sorjamaa et al., 2007; Taieb et al., 2012). Nonetheless, the recursive strategy is successfully applied in real-world applications of time series prediction (Xue et al., 2019; Jing et al., 2020; Kohler et al., 2020; Gu et al., 2021; Abreu et al., 2023a). Sorjamaa et al. (2007) and Hamzaçebi, Akay and Kutay (2009) observed in their study that the direct strategy outperforms the recursive one. Nonetheless, Taieb and Hyndman (2014) presented some examples of time series in which the recursive approach outperforms the direct one. Xue et al. (2019) observed the benefits, in terms of performance, of using the recursive strategy compared to the direct one employed in heat load forecasting for district heating

systems. The combination of the XGBoost system and the recursive strategy was adopted by Gu et al. (2021) and Xue et al. (2019) to model time series. As modeling force-displacement curves involves long time series prediction, the recursive strategy is indicated since it avoids training a distinct model for each time step, reducing the modeling time (Xue et al., 2019).

To train the desired proxy model adopting the recursive strategy, the one-step-ahead prediction is used. Equation (E.11) defines how the model f should be trained. The presented strategy considers exogenous variables as model inputs (Sorjamaa et al., 2007). In this work, the material parameters to be estimated compose the major part of these exogenous variables.

$$\hat{p}_{t+1} = f(p_t, p_{t-1}, \dots, p_{t-h+1}, x_1, x_2, \dots, x_m) \tag{E.11}$$

in which $\hat{p}_i$ is a predicted force intensity, p_i is a known historical datum, t is the time step, h is the moving window size, and x_i is an exogenous variable of the set of m variables. After training f, the following value $\hat{p}_{t+2}$ is predicted according to Equation (E.12).

$$\hat{p}_{t+2} = f(\hat{p}_{t+1}, p_t, \dots, p_{t-h+2}, x_1, x_2, \dots, x_m) \tag{E.12}$$

The values $\hat{p}_{t+3}, \hat{p}_{t+4}, \dots, \hat{p}_{t+n}$ are calculated using the same scheme presented in Equation (E.12): the oldest input is removed and the last predicted value is inserted. In this work, the known values p_t to p_{t-h+1} for $t \leq 0$ are equal to the initial force intensity $p_o = 0$. Figure 47 describes the recursive strategy, relating the exogenous variables (uniaxial tensile stresses f_{t0} at which the damage process is initialized, cohesion c, and friction angle ϕ) and the known and the predicted time series values to each prediction.

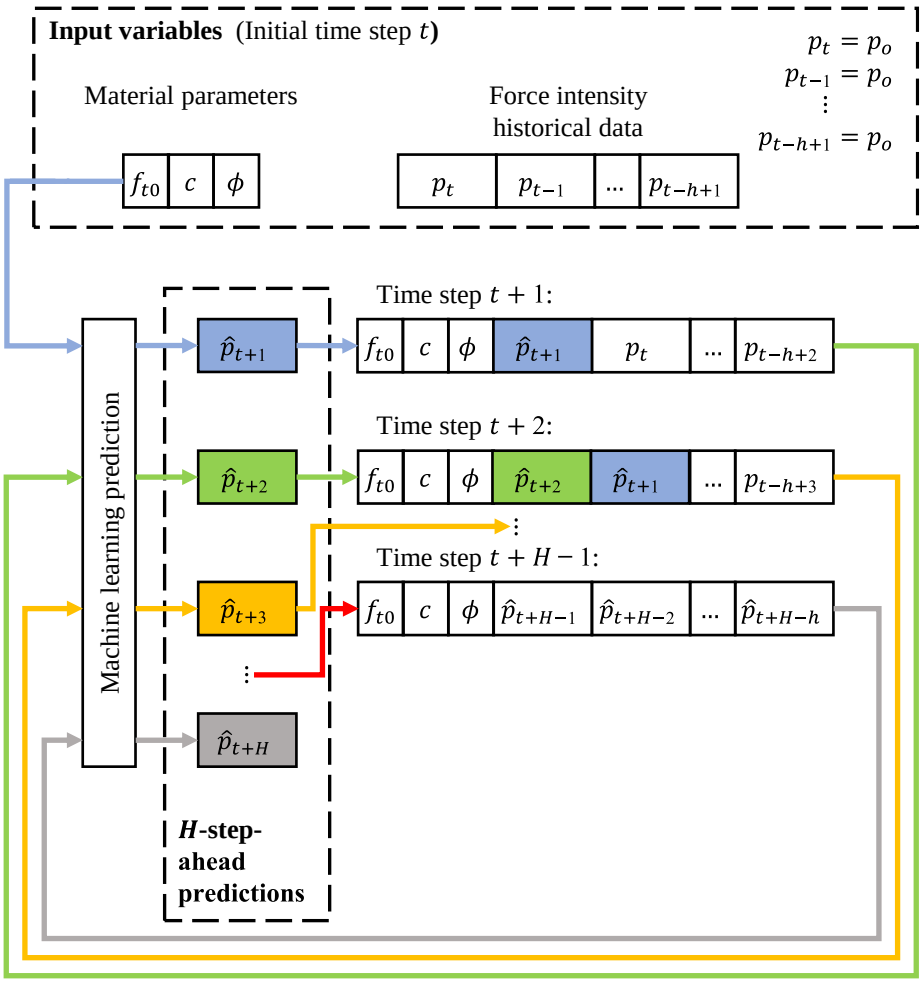

Figure 47 Schematic representation of the multistep-ahead prediction

E.5. Hyperparameter optimization

Machine learning algorithms are directly dependent on a set of parameters called hyperparameters. These are specified before the training stage, affecting how the algorithm defines the mapping between inputs and outputs (Wu et al., 2019). A proper definition of these hyperparameters can prevent the occurrence of overfitting. In this sense, grid (Meng, Zhong and Wei, 2020; Zhang et al., 2020; Wang et al., 2021) and random (Bikmukhametov and Jäschke, 2019) searches are vastly adopted to select proper hyperparameters in several machine learning applications. In general, the random search outperforms the grid one, mainly in high-dimensional problems (Bergstra and Bengio, 2012).

In the hyperparameter optimization scenario, Bayesian optimization is a robust approach that outperforms grid and random searches (Bergstra et al., 2011; Bergstra, Yamins and Cox, 2013; Nguyen, Liu and Zio, 2020). This procedure is

widely utilized to optimize objective functions that require high computational effort, tolerating noisy functions (Frazier, 2018). It is a derivative-free global method that deals with nonconvex and noncontinuous functions, as well as with black-box systems (Frazier, 2018; van Hoof and Vanschoren, 2021). This method appropriately solves optimization problems with less than 20 dimensions (Hoffman, Brochu and de Freitas, 2011; Frazier, 2018).

Bayesian optimization builds a probabilistic (proxy) model for the objective function and exploits this model to define the following points of function evaluation under uncertainty (Snoek, Larochelle and Adams, 2012; Frazier, 2018). In this work, the initial sampling that defines the first proxy model is carried out based on the Halton sequence. Thus, the proxy model is continually updated while new points are computed through the optimization algorithm. A predefined number of objective function evaluations is used as the stopping criterion. There are various mathematical strategies to build this proxy model, such as tree-structured Parzen estimator (Bergstra et al., 2011; Thornton et al., 2012; Bergstra, Yamins and Cox, 2013; Xia et al., 2017), random forest (Thornton et al., 2012), and gradient boosting machine (van Hoof and Vanschoren, 2021). This work assumes the well-known Gaussian process model to build that proxy model, producing an efficient hyperparameter optimization approach (Wu et al., 2019). Furthermore, this model is broadly used in continuous optimization problems (Yang and Shami, 2020). That is the case of the presented hyperparameter optimization task. Authors that integrate the XGBoost system and Bayesian optimization are Xia et al. (2017), Shi et al. (2021) and Zhou et al. (2021). Importantly, the scikit-optimize library (Head et al., 2021) is adopted to perform the Bayesian optimization. Furthermore, scikit-learn (Pedregosa et al., 2011) is a free library supporting some machine learning tasks, considering the Python environment.

The Gaussian process regression is a probabilistic machine learning method. Using previous knowledge, this algorithm provides predictions based on a multivariate normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ (Murphy, 2012). The procedure defines a prior distribution over functions used to characterize a posterior distribution based on known data (Murphy, 2012). That process is a Gaussian random function entirely characterized by its mean function $m(\boldsymbol{x})$ and covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$ (Rasmussen and Williams, 2006),

as presented in Equation (E.13). In addition, the covariance $k(\boldsymbol{x}, \boldsymbol{x}')$ between $\mathcal{F}(\boldsymbol{x})$ and $\mathcal{F}(\boldsymbol{x}')$ is typically written as a function of the input variables $\boldsymbol{x}$ and $\boldsymbol{x}'$ (Rasmussen and Williams, 2006). Any prior knowledge about $m(\boldsymbol{x})$ is usually unavailable and, by symmetry, $m(\boldsymbol{x}) = 0$ can be considered (Bishop, 2006).

$$\mathcal{F}(\boldsymbol{x}) \sim \mathcal{GP}\big(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')\big) \tag{E.13}$$

Considering training dataset $\boldsymbol{X}$ and test dataset (unseen data) $\boldsymbol{X}_*$of n and n_* sample points, respectively, the training outputs $\boldsymbol{y}$ and the test outputs $\boldsymbol{y}_*$ are possibly corrupted by noise (Murphy, 2012). The multivariate normal distribution of the training outputs $\boldsymbol{y}$ and the test outputs $\boldsymbol{y}_*$ has the following form (Rasmussen and Williams, 2006):

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{y}_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_* \end{bmatrix}, \begin{bmatrix} \boldsymbol{K} + \sigma_n^2 \boldsymbol{I} & \boldsymbol{K}_* \\ \boldsymbol{K}_*^T & \boldsymbol{K}_{**} \end{bmatrix} \right) \tag{E.14}$$

in which $\boldsymbol{\mu} = m(\boldsymbol{X})$, $\boldsymbol{\mu}_* = m(\boldsymbol{X}_*)$, $\boldsymbol{K} = k(\boldsymbol{X}, \boldsymbol{X})$ is a matrix of dimension $n \times n$, $\boldsymbol{K}_* = k(\boldsymbol{X}, \boldsymbol{X}_*)$ is a matrix of dimension $n \times n_*$, and $\boldsymbol{K}_{**} = k(\boldsymbol{X}_*, \boldsymbol{X}_*)$ is a matrix of dimension $n_* \times n_*$. It should be highlighted that $\sigma_n^2 \boldsymbol{I}$ represents the white noise covariance function. According to Rasmussen and Williams (2006), to obtain the posterior distribution over functions, it is required to constrain the prior distribution in (E.14) to the functions that agree with the training sample points. As shown by Murphy (2012), this operation can be performed by defining the following posterior conditional distribution:

$$\boldsymbol{y}_* | \boldsymbol{y} \sim \mathcal{N}(\overline{\boldsymbol{y}}_*, \boldsymbol{\Sigma}) \tag{E.15}$$

$$\overline{\boldsymbol{y}}_* = \boldsymbol{\mu}_* + \boldsymbol{K}_*^T (\boldsymbol{K} + \sigma_n^2 \boldsymbol{I})^{-1} (\boldsymbol{y} - \boldsymbol{\mu}) \tag{E.16}$$

$$\boldsymbol{\Sigma} = \boldsymbol{K}_{**} - \boldsymbol{K}_*^T (\boldsymbol{K} + \sigma_n^2 \boldsymbol{I})^{-1} \boldsymbol{K}_* \tag{E.17}$$

where $\boldsymbol{y}_* | \boldsymbol{y}$ represents the probability distribution of $\boldsymbol{y}_*$ when $\boldsymbol{y}$ assumes a specific known value. Conditional probability is a crucial concept in Bayes' theorem.

The outputs $\boldsymbol{y}_*$ can be sampled from the posterior distribution based on the mean vector (E.16) and covariance matrix (E.17). However, the mean vector $\overline{\boldsymbol{y}}_*$ can be used to predict $\boldsymbol{y}_*$ (Zhang et al., 2019). A typical assumption is $\boldsymbol{\mu} = \boldsymbol{\mu}_* = 0$ (Rasmussen and Williams, 2006; Wu et al., 2019) since it is simple and sufficient

(Bergstra et al., 2011). Rasmussen and Williams (2006) presented an efficient algorithm to make predictions based on Gaussian process regression.

Once defined the optimization procedure, the following phase is to define the objective function. The present work proposes to specify the objective function according to multistep-ahead prediction errors. A model trained to perform one-step-ahead tasks is assessed for multistep-ahead prediction. Then, the hyperparameters are selected to minimize k-fold cross-validation performance, the average Root Mean Squared Error (RMSE). Considering regression tasks in machine learning, both k-fold cross-validation (Thornton et al., 2012; Vasconcelos et al., 2019; Wu et al., 2019; Yang and Shami, 2020; Koc, Ekmekcioğlu and Gurgun, 2021; Zhou et al., 2021; Sultana et al., 2022) and RMSE (Riazi et al., 2016; Tripoppoom et al., 2019; Vasconcelos et al., 2019; Tripoppoom, Ma, et al., 2020; Zhou et al., 2021) are vastly used in hyperparameter tuning. It should be highlighted that each performance is computed using multiple multistep-ahead predictions, one for each set of material parameters, as each set is associated with a specific time series. Each series is kept complete in the folds created using cross-validation. The proposed process is described in Algorithm 9.

Algorithm 9 Objective function adopted by the hyperparameter optimization task

Initialize:

f: Machine learning model defined by a predetermined set of hyperparameters
$\mathcal{D}$: Full training dataset
k: Number of folds in the selected cross-validation method
$\mathcal{E}$: Metric to evaluate model performance

Procedure:

For i from 1 to k:
  Define the validation set $\mathcal{D}_i^{val}$ as fold i in $\mathcal{D}$
  Define the training set $\mathcal{D}_i^{train}$ as the remaining data in $\mathcal{D}$
  Define N as the number of time series in $\mathcal{D}_i^{val}$
  Train the model f on $\mathcal{D}_i^{train}$ with the one-step-ahead prediction
  For j from 1 to N:
    Predict outcomes $\hat{\boldsymbol{y}}_{i,j}$ on $\mathcal{D}_i^{val}$ using f and the recursive strategy
  End
  Based on $\mathcal{E}$, evaluate performance E_i using the outputs in $\mathcal{D}_i^{val}$ and $\hat{\boldsymbol{y}}_{i,j}, \forall j \in \{1,2,\dots,N\}$
End
Calculate mean $E = \frac{1}{k}\sum_{i=1}^{k} E_i$ (standard deviation can be useful in further analyses)
Return E

It is not recommended to minimize the objective function directly estimated by $\bar{\boldsymbol{y}}_*$, owing to insufficient information from the small number of sample points (Elsas et al., 2021). In this sense, an acquisition function typically designates new candidates for the global optimum. The acquisition function defines where to evaluate the objective function by balancing the trade-off between exploration (related to global search) and exploitation (related to local search) (Frazier, 2018; van Hoof and Vanschoren, 2021). Usually, the same point optimizes both the acquisition and the objective functions (Wu et al., 2019).

Common acquisition functions are Probability of Improvement (PI), Expected Improvement (EI), and Lower Confidence Bound (LCB). However, Hoffman, Brochu and de Freitas (2011) argue that choosing the proper acquisition function to deal with a specific problem is not trivial. Then, the authors proposed the GP-Hedge approach that adaptively manages an acquisition function set. Using the softmax function, this approach probabilistically selects one optimum among those computed by all acquisition functions in a predefined set. GP-Hedge satisfactorily solved the ten- and twenty-dimensional problems assessed by Hoffman, Brochu and de Freitas (2011).

In this scenario, the parameters associated with Bayesian hyperparameter optimization are presented in Table 14. As Snoek, Larochelle and Adams (2012) suggested, the twice-differentiable Matérn covariance function is adopted for the configuration established in Table 14. This work applies the limited-memory BFGS optimization algorithm. Figure 48 shows the adopted machine learning modeling process. After the hyperparameter tuning, if the final performance is not satisfactory, it is necessary to return to the hyperparameter set or data acquisition definitions.

Table 14 Bayesian hyperparameter optimization configuration adopted in this work

Gaussian process component	Symbol	Description	Value
	σ	Standard deviation	1
Matérn covariance function k	l	Length-scale parameter	1
	ν	Shape parameter for the Matérn family	2.5
White noise covariance function	σ_n	Standard deviation	1
PI and EI acquisition functions	ξ	Trade-off parameter	0.01
LCB acquisition function	κ	Variance weight parameter	1.96
Softmax function	η	Weight parameter	1

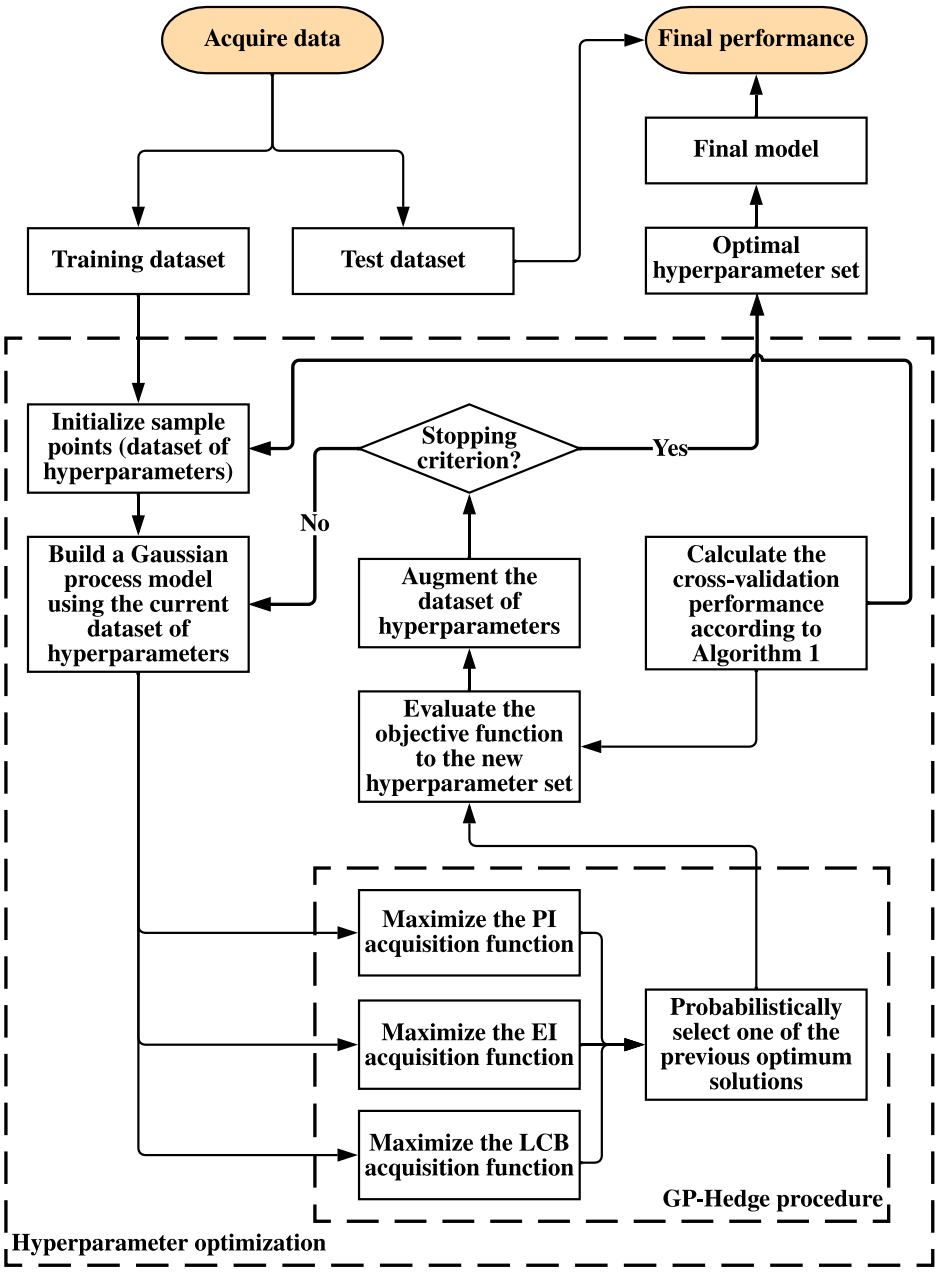

Figure 48 Flowchart of the proposed machine learning modeling

E.6. Genetic algorithm

The genetic algorithm (Holland, 1975) is a computational model of biological evolution that simulates Darwinian natural selection mechanisms. The algorithm is a metaheuristic global optimization and search procedure inspired by natural evolution. In this sense, an optimal solution of an objective function is computed by applying selection, crossover and mutation operators through successive

population generations. This section presents the algorithm implemented in the in-house Python framework Minerva, used to solve parameter identification problems.

The former implementations of genetic algorithms considered a binary codification to represent the candidate solutions. This procedure can adequately solve several sorts of problems. Nonetheless, it may have a high computational cost, especially for a high-dimensional continuous domain with great desired numerical precision (Goldberg, 1991; Herrera, Lozano and Verdegay, 1998). This work adopts a real-coded genetic algorithm that outperforms the binary-coded genetic algorithm in continuous optimization since it is more precise, more consistent, and faster in execution time (Michalewicz, 1996; Herrera, Lozano and Verdegay, 1998). The genetic algorithm presented below is based on the studies of Kaelo and Ali (2007), Sawyerr, Adewumi and Ali (2014), Jin et al. (2016), and Yin et al. (2017).

The algorithm is applied here to solve the optimization problem (E.1). The domain $\boldsymbol{\Omega}$ is defined as $l_i \leq x_i \leq u_i$, such that $l_i, u_i \in \mathbb{R}$ and $i \in \{1, 2, \dots, D\}$. Each vector of continuous components $\boldsymbol{x}$ is composed of D genes, representing each vector component. The solution $\boldsymbol{x}^* \in \boldsymbol{\Omega}$ of the problem (E.1) is defined according to $F(\boldsymbol{x}^*) \leq F(\boldsymbol{x}), \forall \boldsymbol{x} \in \boldsymbol{\Omega}$.

This technique starts with a set of candidate solutions. In the genetic algorithm context, this set is named population, whereas each solution in the population is named individual. This study assumes that the population in the initial iteration (generation) is produced using Latin hypercube sampling (McKay, Beckman and Conover, 1979) to obtain N random sample points in a stratified domain, as adopted by Jin et al. (2016) and Yin et al. (2018).

The fitness function guides the computational model toward the optimal solution. Accordingly, the fitness function value increases as the individual approximates the optimal solution. This function is defined according to the objective function. Based on the classical methodology that applies minimization methods to solve maximization problems, Equation (E.18) computes the fitness function $\mathcal{F}(\boldsymbol{x})$ to solve the problem (E.1). The adopted fitness function is adequate to deal with the selection mechanism presented in the next section. Additionally, this strategy is robust to handle positive and negative objective function values of any magnitude.

$$\mathcal{F}(\boldsymbol{x}) = -F(\boldsymbol{x}) \tag{E.18}$$

Figure 49 illustrates the population evolution in this genetic algorithm throughout the generations, in which the individuals' fitness is improved continually. As this algorithm is population-based, the global minimum (marked in green) is the individual with the highest fitness function (or the lowest objective function).

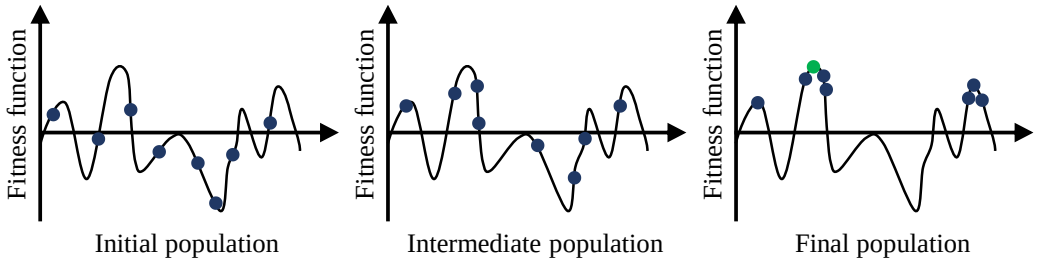

Figure 49 Schematic depiction of genetic algorithm

E.6.1. Genetic operators

This section describes the genetic operators applied in the algorithm. The selection operator is crucial to maintain populational diversity because it selects appropriate individuals to provide crossovers and mutations. According to Bäck (1994), selective pressure characterizes the emphasis on choosing the best individuals. The higher the selective pressure is, the more the best individuals are benefited. Nevertheless, high selective pressure provides a less diverse population (Mokhade and Kakde, 2014). This study adopts tournament selection, a process in which a set of individuals is randomly chosen from the population in this selection operator to compete with each other. The individual with the greatest fitness value wins the tournament and is placed in an intermediate population, also known as mating pool. Miller and Goldberg (1995) described this mechanism as simple to implement on parallel and nonparallel frameworks, robust in the presence of noise, and with adjustable selective pressure (altering the number of individuals in the tournament). The tournament size defines the number of selected individuals. Here, the tournament size is equal to two in order to preserve population diversity (Yin et al., 2017).

The arithmetic crossover linearly combines the genes x_i^a and x_i^b of two different individuals from the mating pool, also called parents, to generate new solutions, named offsprings (Sawyerr, Adewumi and Ali, 2014). Therefore, the genes $\bar{x}_i^a$ and $\bar{x}_i^b$ are defined as

$$\bar{x}_i^a = \alpha_i x_i^a + (1 - \alpha_i) x_i^b \tag{E.19}$$

$$\bar{x}_i^b = (1 - \alpha_i) x_i^a + \alpha_i x_i^b \tag{E.20}$$

in which α_i for each gene is a uniformly distributed random number generated between -1 and 1. The presented crossover mechanism can produce individuals outside the domain $\boldsymbol{\Omega}$. Thus, the presented implementation performs new attempts to generate other solutions inside the domain.

After the crossover execution, the mutation operator is performed to reduce the chance of computing local minima. This mechanism randomly modifies the genes of selected individuals to provide genetic diversity. This work applies the nonuniform mutation (Kaelo and Ali, 2007). This operator carries out a uniform search in domain $\boldsymbol{\Omega}$ in the initial generations, whereas it provides a local search in advanced generations (Kaelo and Ali, 2007). Thus, this mechanism considers the upper and lower bound u_i and l_i of each gene x_i to generate the mutated gene $\bar{x}_i$, according to

$$\bar{x}_i = \begin{cases} x_i + \xi(t, u_i - x_i), & \text{if } r_1 \leq 0.5 \\ x_i - \xi(t, x_i - l_i), & \text{if } r_1 > 0.5 \end{cases} \tag{E.21}$$

considering that

$$\xi(t, \chi) = \chi \left(1 - r_2^{\left(1 - \frac{t}{T}\right)^b} \right) \tag{E.22}$$

where r_1 and r_2 are uniformly distributed random numbers generated between 0 and 1; t is the generation number; and T is the maximum user-defined number of generations. In this study, it is adopted $b = 5$, according to Elsayed, Sarker and Essam (2010).

E.6.2.
Genetic algorithm implementation

Since genetic algorithms are computational population models, a set of N candidate solutions $\mathbf{P}_t = \{\boldsymbol{x}_t^1, \boldsymbol{x}_t^2, \dots, \boldsymbol{x}_t^N\}$ is maintained in each generation t. The initial population $\mathbf{P}_0$ is generated using Latin hypercube sampling in domain $\boldsymbol{\Omega}$ and the fitness values of all individuals of this population are calculated.

After defining the population, let $t \coloneqq t + 1$ and apply the tournament selection operator as described in the previous section to select a predefined number of individuals M individuals in $\mathbf{P}_{t-1}$ to compose the mating pool $\overline{\mathbf{P}}_t$, whereas $M \leq N$. As the adopted crossover operator produces two offsprings, M is an even number in this work. Moreover, the operator is allowed to select an individual multiple times. Next, the individuals in $\overline{\mathbf{P}}_t$ are paired and submitted to crossover and mutation mechanisms.

The crossover mechanism generates M offsprings to compose the auxiliary population $\widehat{\mathbf{P}}_t$. This operator is applied considering a probability p_c. Two offsprings are generated using Equations (E.19) and (E.20) if $p_c > p_1$, whereas p_1 is a distinct uniformly distributed random number generated between 0 and 1 for each crossover. These new individuals are placed in $\widehat{\mathbf{P}}_t$. If $p_c \leq p_1$, the selected parents do not perform the crossover and are copied to $\widehat{\mathbf{P}}_t$.

While p_2 is a distinct uniformly distributed random number generated between 0 and 1 for each gene of the individuals of $\widehat{\mathbf{P}}_t$, the mutation operator is applied if $p_m > p_2$. Equation (E.21) is employed to mutate the selected genes of these individuals. Thus, the individuals whose genes may or not have mutated are placed in the auxiliary population $\widetilde{\mathbf{P}}_t$. The fitness values of population $\widetilde{\mathbf{P}}_t$ can be now computed.

At the end of generation t, the individuals in $\widetilde{\mathbf{P}}_t$ are placed in the main population $\mathbf{P}_t$. Moreover, the elite, the best individuals of the population $\mathbf{P}_{t-1}$ in terms of the fitness function in Equation (E.18), are also placed in $\mathbf{P}_t$. Accordingly, $E = N - M$ represents the number of candidate solutions in the elite. This strategy is called elitism. A new generation is set up considering the procedures previously described unless the maximum number of generations T is reached. Figure 50 depicts a flowchart of the presented genetic algorithm.

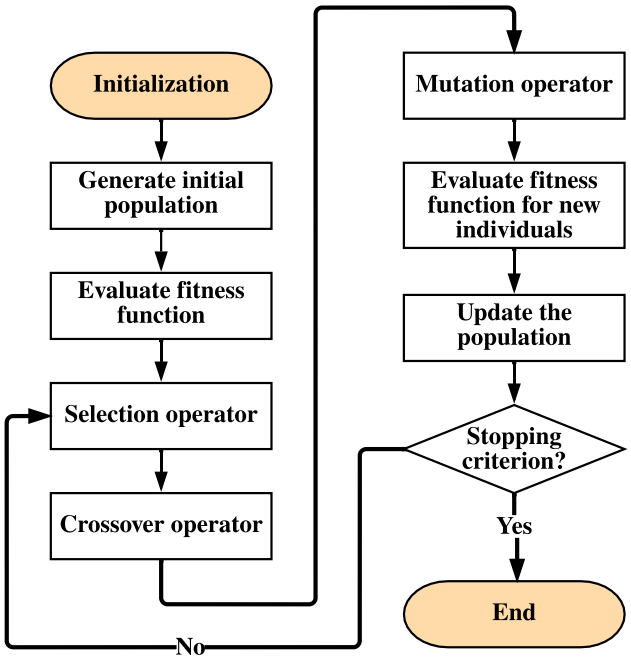

Figure 50 Flowchart of the presented genetic algorithm