Steady-State Algorithm with Structural Periodicity: Application to Computation of Railways' Ballast Plastic Strains

Thibault Badinier, Siegfried Maiolino, Habibou Maitournam

▶ **To cite this version:**

Thibault Badinier, Siegfried Maiolino, Habibou Maitournam. Steady-State Algorithm with Structural Periodicity: Application to Computation of Railways' Ballast Plastic Strains. Geotechnics, 2026, 6 (1), pp.29. ⟨10.3390/geotechnics6010029⟩. ⟨hal-05576255⟩

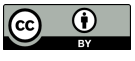
HAL Id: hal-05576255

https://hal.science/hal-05576255v1

Submitted on 1 Apr 2026

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

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



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

geotechnics

MDPI

Article

Steady-State Algorithm with Structural Periodicity: Application to Computation of Railways' Ballast Plastic Strains

Thibault Badinier [1,*], Siegfried Maiolino [2] and Habibou Maitournam [3]

[1] GERS-SRO, Université Gustave Eiffel, 77454 Marne-la-Vallée, France
[2] Département Laboratoire de Lyon, Cerema Centre-Est, Cerema, 69500 Bron, France
[3] IMSIA, ENSTA, EDF, CNRS, Institut Polytechnique de Paris, 91762 Palaiseau, France
* Correspondence: thibault.badinier@univ-eiffel.fr

Abstract

The geometry of ballasted railway tracks is crucial for ensuring railway safety and efficiency. This paper introduces the use of innovative steady-state algorithms designed to compute plastic strains in linear geotechnical structures like railway ballast layers, within Finite Element Methods (FEMs). Facing the specificities of moving loads, traditional step-by-step algorithms, while simple and adaptable, are computationally expensive and time-consuming. In contrast, the proposed steady-state algorithms leverage an Eulerian approach to describe the movement of loads significantly reducing computational time while maintaining accuracy. This paper proposes these algorithms as a methodological improvement and demonstrates the applicability and efficiency of the method for non-periodic structures, as well as for periodic structures, such as railway tracks with evenly spaced sleepers. This paper demonstrates the applicability and efficiency of theses algorithms through comparative studies with traditional methods on typical railway structures. The results show that the presented algorithm not only matches the accuracy of step-by-step methods but also drastically reduces computation time and data storage requirements. This advancement has practical applications for railway infrastructure managers, enabling more efficient and accurate predictions of track geometry evolution and preventing incidents through improved maintenance strategies.

Keywords: railway ballast; finite element methods; steady-state algorithm; elastic–plastic behaviour; structural periodicity

Check for updates

Academic Editor: Amin Chegenizadeh

Received: 4 January 2026
Revised: 17 February 2026
Accepted: 12 March 2026
Published: 20 March 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Railway tracks are infrastructures supporting train traffic. For example, in France the global network was 48,000 km long in 2026, supporting each day around 15,000 trains of goods and travelers (These numbers were taken from the official website www.sncf-reseau.com, in 17 February 2026). Therefore, they have a major social and economic impact in western countries.

Conventional ballasted railway tracks are composed of multiple elements: rails, sleepers, fasteners, ballast, rail pads, etc. The track's global mechanical behavior can be explained with the three main elements—rails, sleepers, and ballast—as shown in Figure 1 [1]. Rails ensure the support and guidance of the trains. Sleepers maintain rails and transmit load from rails to ballast. Ballast is a geo-material extracted from natural rocks. It ensures multiple roles, transmits and shares load from sleepers to the ground, maintains track geometry, absorbs part of the vibrations, etc.

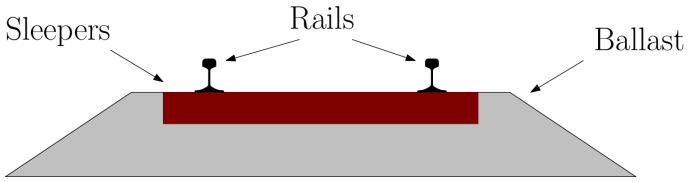


Figure 1. Ballasted railway track composition, simplified representation limited to rails, sleepers, and ballast.

Train transportation safety and efficiency is highly dependent on track geometry, and centimeters of differential deformations could lead to derailment. In order to prevent these incidents, the track geometry is monitored and maintained throughout its life cycle. In order to predict these deformations, numerical modeling can be conducted with Finite Element Methods (FEMs). These models are the most commonly used when it comes to modeling the general behavior of tracks [2]. These models allow the computation of track deformation, mainly due to the irreversible deformation of the ballast layer. Its elastic–plastic behavior widely contributes to track geometry evolution [3]. Therefore, understanding these behaviors and degradation mechanisms is a safety concern.

An accurate modeling process requires all specificities of the track geometry to be properly represented as a periodic structure [4,5] and mobile loading [6].

Some studies assume studying train solicitation like repeated loading on simplified models [7–9]. This method is efficient from a numerical point of view, as it simplifies the solicitation and reduces the model complexity. In tunneling applications, a similar philosophy is followed to represent progressive construction of the tunnel with "convergence–confinement" methods [10]. In doing so, transportation infrastructure allows high numbers of cyclic loading at reduced computational costs. However, it does not properly represent specificities of mobile loading, particularly stress history considering magnitude and stress orientation, even though stress history has a huge influence on plastic strain development [11,12]. An illustration of the importance of proper mobile loading representation is shown in Section 2.

Therefore, railway track structure behavior should be studied, considering the specificities of such structures and their loading. In the case of transportation infrastructure, we consider a linear structure with mobile loading solicitation. Linear structures are described as structures developed with large dimensions in one particular axis. Road, railways, and dams could be described as linear structures. In particular, we consider here s ballasted track that could be considered self-similar, i.e., any section of the track is equivalent regardless of its position in the structure. The second particularity of railway tracks is the moving of load solicitations. From the structure's point of view, these solicitations are non-permanent and repeated loading, characterized by loading phases: approaching, passing, and moving away. This moving load imposes stress magnitude and orientation variation that should be taken into account for proper structure behavior studies.

The most common strategies to represent load displacement effect on structure is to compute the structure behavior under several successive positions of the load. Although there are different strategies available for representing these load movements, they all rely on the Lagrangian study framework, i.e., from the point of view of the structure, and could be called step-by-step algorithms. These methods are widely used to study problems that involve a moving load such as tunneling (for example, Hasanpour and l. (2014) [13] or Bourgeois et al. (2025) [14]) or lorry loading on road pavement (for example, Valšková and Melcer (2018) [15] or Deng and al. (2019) [16]). In railway track studies, only step-by-step algorithms have been used to represent actual load movement on the model [17–20].

In order to study railway tracks' non-linear behaviour under moving loads, we present innovative methods called steady-state algorithms. Instead of using a Lagrangian approach, that decomposes the load movement in steps, these algorithms use an Eulerian framework.

The algorithm is based on the works of Nguyen and Rahimian (1981) [21] and Dang Van et al. (1985) [22] which describe the theoretical framework (see Section 2.3.1) and first applications of this method. It has been used for various problems involving moving loads, such as the impact of rolling on rail heads (Maïtournam , 1989 [23], Dang Van and Maïtournam, 1993 [24]), beading (Ouakka, 1993 [25]), interaction between rock and cutting tool (Geoffroy, 1996 [26]), automotive brake disc (Nguyen-Tajan et al., 2002 [27]), tunnelling (Corbetta, 1990 [28], Maïolino,2006 [29], Defay and al. (2020) [30]). Later adaptations of the steady state algorithm also allow studying different cases with non-constant load or non-constant speed within this Eulerian framework. The TRC algorithm (*Transitoire dans le Repère de Chargement*) developed by Nguyen-Tajan [27] allows such particularities.

Transportation infrastructure, in particular railway tracks, remains unstudied with such computation process, while structural periodicity of these structures is an additional difficulty. An adapted periodic-steady-state algorithm, as been developed to address such specific study cases.

Processing with steady state algorithms reduces the computations to one single computation step, in comparison with the multiple computation steps needed by step-by-step algorithms. The process results then in an efficient and time saving way to compute plastic strains in linear structure under moving loads. This paper aims to prove applicability and efficiency of those algorithms for studding the non-linear behavior of railways track infrastructure in comparison with the classical step-by-stem algorithm.

In the the following section, we describe a classic step-by-step algorithm, and then two versions of the steady state algorithm. The first one allows the computation of plastic strain in a linear, non-periodic structure. The second one is adapted to compute plastic strains in a periodic structure, such as railways tracks.

In subsequent sections, we expose the efficiency of these steady-state algorithms, comparing results computed by step-by-step and steady-state algorithms on typical railway structures. Two example models are presented, including geometry, load, and materials behaviors. Then, the results are compared in terms of plastic strain amplitude, computation time, and stored data.

The study presented in this paper addresses a critical need for transportation infrastructure management, such as railways, and the long-term numerical modeling of these infrastructures. It proposes a methodological improvement applicable to further research aiming to accurately predict ballasted track settlement. By introducing innovative steady-state algorithms in the finite element modeling process for computing plastic strains in ballasted railway tracks under moving loads, we drastically cut computational cost for such applications, while addressing the particularities of mobile loading. These methods are applicable to non-periodic and periodic structures, like ballasted railway tracks, and tremendously improve computational performance in both cases. This computational progress allows for the development of accessible long-term modeling and predictable infrastructure management. Although the application is demonstrated here on railways tracks, it also opens up the usability of these methods for various other geotechnical structures, contributing to improving their safety and maintenance strategies.

2. Computation Process for Moving-Load Solicitations

This section aims to describe the various computation process studied and compared in this paper.

2.1. No Movement Process

As introduced earlier, many studies on ballasted track behavior represent the loading cycle with a load of variable amplitude, or are precomputed load case, applied on a reduced

section of the structure in order to reduce computation complexity and time cost. We will call these processes "no movement processes". While efficient from a time-consumption point of view, especially for a high number of loading cycles, such a process hides the specificities of a moving load, such as a rolling train on a railway. Indeed, load position influences both stress magnitude and stress orientation all over the structure. Schematically, such a loading cycle is shown in Figure 2 and stress history is shown in Figure 3a. Therefore, no movement processes can mimic stress magnitude variation but will hide the variation in stress orientation, as illustrated in Figure 3b.

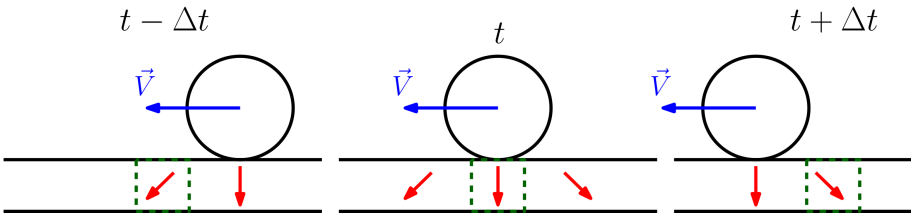
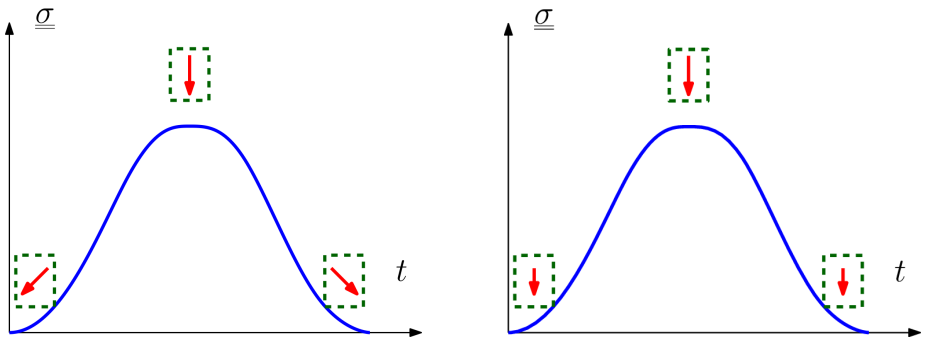


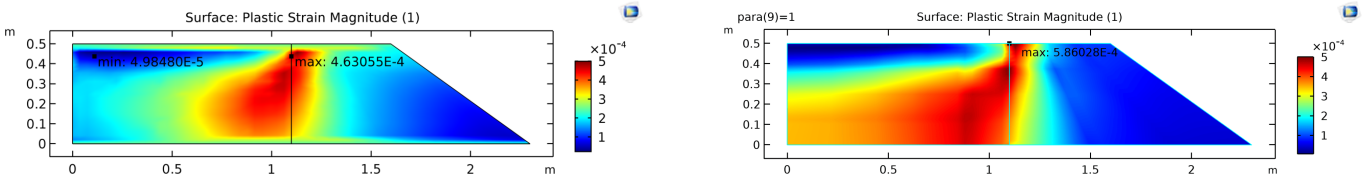
Figure 2. Schematic of loading process, exposing the principle of stress orientation variation during load movement.



(**a**) Moving loading (**b**) Non-moving loading

Figure 3. Schematic of stress histories, exposing the differences between them: (**a**) proper stress history with varying stress magnitude and orientation; (**b**) inaccurate stress history, representing variation in stress magnitude and constant orientation.

In order to illustrate the importance of considering stress rotation during loading cycles, we present in Figure 4 a plastic strain mapping computed with a moving load and a varying load (no movement process), with all other parameters being equal. These results clearly show the discrepancies between both approaches and demonstrate the importance of representing the complexity of the loading cycle.



(**a**) Moving load (**b**) Non-moving loading

Figure 4. Plastic strain obtained according to both type stress histories.

In order to obtain a correct numerical model of a moving-load influence, we can use step-by-step or steady-state processes, as explained in the following part.

For simplicity, in the rest of the paper, we suppose that the structure is an $\vec{x}$ axial structure and the load moves downstream of the structure, i.e., the load speed is $\vec{V} = -V\vec{x}$.

2.2. Step-by-Step Computation

As introduced previously, step-by-step algorithms describe the methods for studying the moving load problem by progressively displacing (or extending) the load on the structure while the structure remains unchanged. These methods rely on a Lagrangian framework, i.e., a study from the point of view of the structure.

Principles

Within this framework, the presented algorithm describes load movement using a set of successive load positions. During one loading cycle, the load starts for a position on one side of the model, and is then moved to the other side of the model, with multiple successive positions, as shown in Figure 5. In doing so, the central part of the model can be considered representative of the structure after the loading cycle as it undergoes the full loading cycle from approaching the load to the load moving away. Therefore, the size of the model and the initial and final load positions should be chosen in order to keep this central part out of the load's influence at the initial and final steps.

We define N computation steps corresponding to N successive positions of the load, describing its movement. For each step, the load position is defined by the abscissa X_n from X_1 down to X_N.

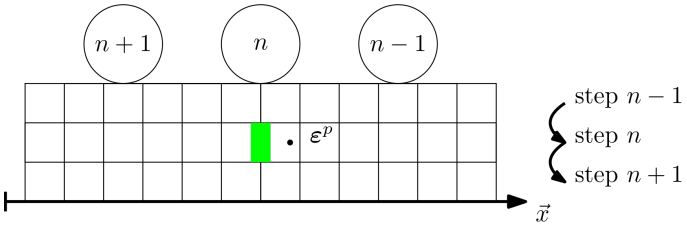


Figure 5. Schematic of step-by-step computation methods exposing successive load position and computation order.

At each step n, the plastic strain fields $\boldsymbol{\varepsilon}^p_{-,n}$ are computed over the entire structure based on the plastic strain fields resulting from the previous step $n-1$. Therefore, plastic strain variation is computed at each individual Gauss point according to Equation (1), where $\boldsymbol{\sigma}$ is the stress tensor, $g(\boldsymbol{\sigma})$ is the plastic potential function, and λ is the plastic multiplicative factor. The overall process uses a classical return mapping algorithm, as described in Appendix A.1.

$$\boldsymbol{\varepsilon}^p_{-,n} = \boldsymbol{\varepsilon}^p_{-,n-1} + \lambda \frac{\mathrm{d}g(\boldsymbol{\sigma})}{\mathrm{d}\boldsymbol{\sigma}} \tag{1}$$

The step-by-step method is very simple, but it needs to compute the plastic strain fields at each step. Moreover, improving the precision will lead to multiplying the number of computation steps, resulting in an increased computational time.

2.3. Steady-State Computation

Steady-state methods suppose switching the point of view. When studying the problems from the moving-load point of view, the structures are seen as flowing under the load, as shown in Figure 6. Any element (green rectangle) in front of the load represents the initial state of the structures. It will be seen as moving under the load, deforming while undergoing the load cycle. This ends at a final state at the back of the load, accounting for possible irreversible deformation. The simplest method supposes that the structure is continuous and invariant along the $\vec{x}$ axis. It also supposes that load speed $\vec{V}$ and load magnitude are constant. Similar processes can be used on curved structures, assuming a constant radius.

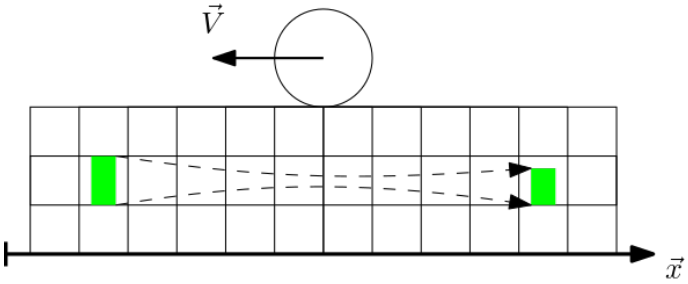


Figure 6. Schematic exposing the principle of a structure flowing under the load.

Because of the invariant structure assumption, a system composed of the load and its direct environment will be self-similar no matter the instant t during the loading cycle. Therefore, by knowing the load movement and speed ($-V\vec{x}$), we can determine a relation between the relative position of the load to a track element in the structure and time gap in the loading phases. In other words, while moving the load at speed V during Δt, the element at position X will be in the same relative position to the load, as well as in the same stress state as the element at position $X + V \cdot \Delta t$ before the movement. Therefore, during the loading cycle, any element will follow the same stress path, which could be read along a line in the structure. The stress state of an element in front of the load is representative of the beginning of the loading cycle, while the stress state of an element at the back of the load is representative of the end of the loading cycle. In the end, a single computation should be representative of the whole movement.

To study plastic strain evolution during a loading cycle, we will study the elements' state by virtually moving them along the structure, with a fixed load position in the middle of the structure.

A loading cycle is described as follows. At the beginning of the cycle, at time $t = 0$, the studied element is considered to be located at one end of the structure, in front of the load. During the cycle, this process virtually moves the studied element from the front to the back of the load. At the end of the loading cycle, the studied element is considered to be located at the other end of the structure, at the back of the load. It should be noted that the size of the structure should be carefully designed in order to keep both ends of the structure out of the load influences. This could be achieved with usual preliminary computation, ensuring no deformation at both model ends. For example, in the models presented in the following part, the structure is set to 15 m length.

2.3.1. Theoretical Framework

Assuming a tensorial quantity $\boldsymbol{A}(t, \vec{X})$, depending on time t and particle position $\vec{X}$, its time derivative $\dot{\boldsymbol{A}}$ will be written as (2). Assuming that particle position $\vec{X}$ is time-dependent, the quantity $\frac{\mathrm{d}\vec{X}}{\mathrm{d}t}$ is equal to the particle speed $\vec{V}$.

$$\dot{\boldsymbol{A}} = \frac{\partial \boldsymbol{A}}{\partial t} + \frac{\mathrm{d}\vec{X}}{\mathrm{d}t} \cdot \frac{\partial \boldsymbol{A}}{\partial \vec{X}} = \frac{\partial \boldsymbol{A}}{\partial t} + \vec{V} \cdot \vec{grad}\boldsymbol{A} \tag{2}$$

In the particular case described here (Section 2), we studied the structure from the point of view of a moving load, assuming structure self-similarity and constant speed. While load speed is described as $\vec{V} = -V \cdot \vec{x}$, the relative speed of the particle is then $V \cdot \vec{x}$. Also, $\vec{grad}\boldsymbol{A}$ can be reduced to $\frac{\partial \boldsymbol{A}}{\partial x}$. Finally, the self-similarity assumption induces an invariable environment from the load's point of view; therefore, $\frac{\partial \boldsymbol{A}}{\partial t}$ could be eliminated from the equation. In the end, the time derivative $\dot{\boldsymbol{A}}$ equation could be reduced to Equation (3), in the described frame of study.

$$\dot{\boldsymbol{A}} = V \frac{\partial \boldsymbol{A}}{\partial x} \tag{3}$$

In a steady-state algorithm, we study plastic strain evolution, usually described using the classical flow rules, Equation (4), which could be rewritten as Equation (5) with this framework:

$$\dot{\boldsymbol{\varepsilon}}^p = \lambda \frac{\mathrm{d}g(\boldsymbol{\sigma})}{\mathrm{d}\boldsymbol{\sigma}} \tag{4}$$

with $\lambda \cdot f(\boldsymbol{\sigma}) = 0$, $\lambda \cdot \dot{f}(\boldsymbol{\sigma}) = 0$, $\lambda \geq 0$ and $f(\boldsymbol{\sigma}) \leq 0$.

$$\frac{\partial \boldsymbol{\varepsilon}^p}{\partial x} = \Lambda \frac{\mathrm{d}g(\boldsymbol{\sigma})}{\mathrm{d}\boldsymbol{\sigma}} \tag{5}$$

with $\Lambda \geq 0$ if $f(\boldsymbol{\sigma}) = 0$, and $\Lambda = 0$ otherwise. $f(\boldsymbol{\sigma})$ is the stress function describing the elastic criterion.

2.3.2. Steady-State Computation Principle

Like every computational method, this one uses the discretization of stress history. The steady-state assumption induces that the stress history during the loading cycle can be described by following the stress state along lines parallel to the structure axis. To do so, the mesh is conveniently built with quadrilateral elements extruded along the structure axis, as presented in Figure 7. Therefore, the Gauss points are lined up in the structure, forming parallel lines where Gauss points can be numbered from 1 to L by increasing the abscissa.

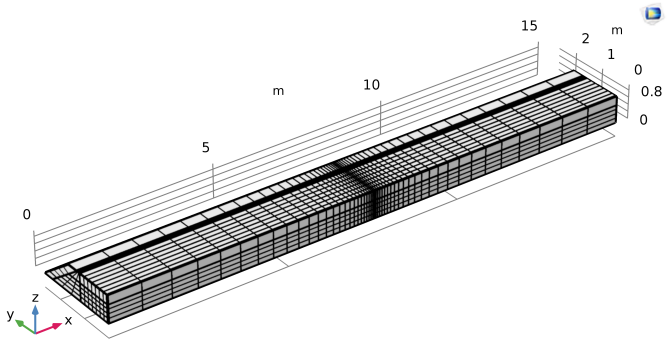


Figure 7. Example of adequate model meshing with a lined element parallel to the structure axis.

Since we assume continuous yielding during the movement, we describe plastic strain evolution by following the Gauss point lines. In doing so, the plastic strain state on a Gauss point $n-1$ represents the past situation of the plastic strain at point n. Therefore, the computational process is based on the transfer of plastic strain from Gauss point $n-1$ to the next n, as described in Equation (6) and shown in Figure 8.

$$\boldsymbol{\varepsilon}_n^p = \boldsymbol{\varepsilon}_{n-1}^p + \Lambda \frac{\mathrm{d}g(\boldsymbol{\sigma})}{\mathrm{d}\boldsymbol{\sigma}} \tag{6}$$

With $\Lambda \geq 0$ if $f(\boldsymbol{\sigma}) = 0$ or $\Lambda = 0$ otherwise.

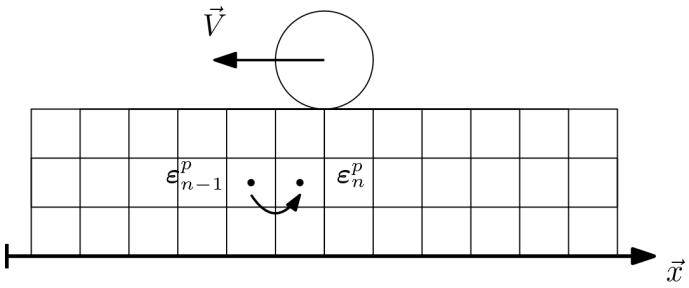


Figure 8. Schematic of steady-state plastic strain computation process, showing computation order.

Plastic strain computation is then conducted on each independent integration line. When the iterative process of the algorithm is over, the resulting state describes an ongoing loading cycle. In front of the load, the structure state corresponds to the beginning of the cycle. At the back of the load, it corresponds to the structure state at the end of the cycle. Following the evolution of the stress–strain state along the structure with an increasing abscissa allows us to describe the temporal evolution of the structure during the loading–unloading cycle.

The entire computation process can be summarized in the detailed algorithm presented in Appendix A.2. This computation algorithm, being non-standard because of the particular integration process, has slightly higher complexity compared to classical step-by-step algorithms. In exchange, the computation of the entire loading cycle only requires the determination of one unique acceptable global plastic strain state, reducing the overall computation time.

2.4. Adaptation for Periodic Structure

The basic steady-state algorithm assumption requires an invariable structure along the $\vec{x}$ axis, which limits the variety of structures applicable with the steady-state algorithm. Indeed, railway tracks exhibit a periodic structure due to rail–sleeper construction. Nevertheless, it is possible to use modified algorithms to compute such periodic structures. We call this version the periodic-steady-state algorithm.

Periodic Steady-State Computation Principle

While the self-similarity assumption is invalid in periodic structures, we suppose a structure composed of a P periodic section. We consider that it is Δl long and corresponds to N Gauss points along the integration lines.

During a loading cycle, points n and $n+1$ will not have the same loading history due to their different structural positions within a periodic section. However, points n and $n+N$ will have the same loading history and same behavior with temporal shift. Therefore, the computation of the plastic strain at point $n+N$ can use the current strain state at point n.

As a drawback, during the computational process, the jump from Gauss point n to Gauss point $n+N$ is equivalent to the load moving by Δl for a time step of $\Delta t = \frac{\Delta l}{V}$. Therefore, the loading gap can be huge, and the stress path discretization may be insufficient to accurately describe plastic strain evolution. In order to enrich this time decomposition of the movement, we split Δt time steps with T smaller steps δt. We then adopt an intermediate approach between the step-by-step and steady-state computations. This sub-decomposition of δt corresponds to a sub-decomposition of the movement with the T position of the load along a period Δl.

As with the previous process, the plastic strain computation requires the immediate previous plastic strain state at each Gauss point and for each computation step. In the case of the computation of plastic strain at Gauss point n and time $t \neq 1$, the immediate previous plastic strain state is taken from point n at time $t-1$, and in the case of the computation of

plastic strain at Gauss point n and time $t = 1$, the immediate previous plastic strain state is taken from point $n - N$ at time $t = T$ (see Equation (7)). From a practical point of view, the computational process transfers plastic strain from point n at time t (n,t) to point n at time $t + 1$ (n,$t + 1$), if $t \neq T$. Also, it transfers (n,T) to the corresponding point, $n + N$, in the next section of the structure at time $t = 1$ ($n + N$,1) (see Figure 9).

$$\varepsilon^p_{n,t} = \varepsilon^p_{n,t-1} + \Lambda \frac{\mathrm{d}g(\sigma)}{\mathrm{d}\sigma} \quad \text{if } t \neq 1 \tag{7a}$$

$$\varepsilon^p_{n,t} = \varepsilon^p_{n-N,T} + \Lambda \frac{\mathrm{d}g(\sigma)}{\mathrm{d}\sigma} \quad \text{if } t = 1 \tag{7b}$$

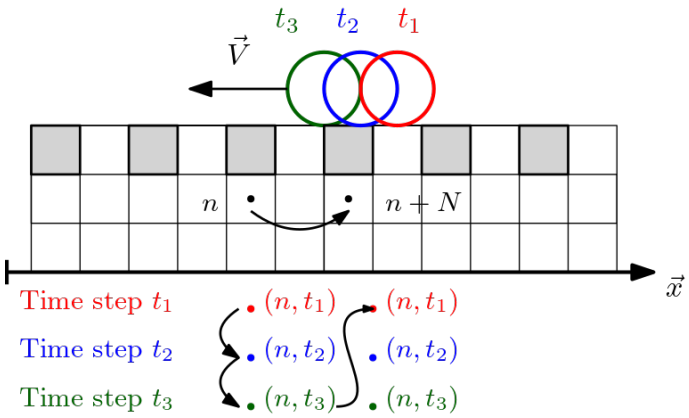


Figure 9. Schematics of periodic steady-state plastic computation process, showing the successive position principle and computation order.

Plastic strains are computed according to this process on each Gauss point line independently. Ultimately, this results in T different states that correspond to the ongoing loading cycle of the structure for each particular load position. Following the plastic strain along the structure provides an idea of the structure evolution during the loading cycle, but it provides a poor discretization. A complete description of the loading cycle can be obtained by combining the T results.

This computation algorithm is more complex than both of the previously presented algorithms because of its particular integration process. In turn, it allows us to model periodic structures in opposition to the simple steady-state computation. In contrast with the step-by-step algorithm, the computation of the entire loading cycle only needs T acceptable plastic strain states to be determined, which results in a reduced computation time.

The detailed computation algorithm is described in Appendix A.3.

2.5. Multiple Loading Cycles

When computing multiple loading cycles, all algorithms use the same principle. A plastic strain state in a representative section of the structure is identified when the cycle computation ends. This state is then repeated in the entire structure as if the whole structure had undergone the same and full loading cycle.

For a step-by-step algorithm, the process describes a load movement that approaches, passes, and moves away from the central elements. Therefore, any central slice undergoes the full loading cycle and should be representative.

For a steady-state algorithm, the process moves the plastic strain state the from front side to the back side of the load. Therefore, a structure slice at the back end of the structure is in a plastic strain state representative of a completed loading cycle. For the following loading cycle, this representative strain state will be used as the initial plastic strain state of the steady-state algorithm, as illustrated in Figure 10.

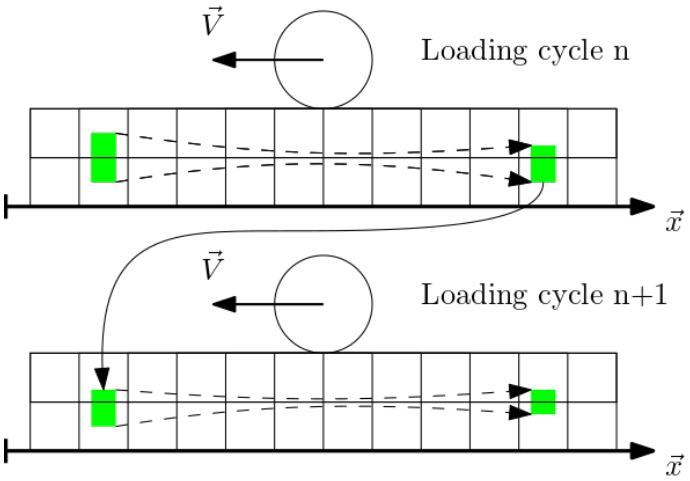


Figure 10. Schematics showing process principles for multiple loading cycles.

3. Example Model

In order to demonstrate the steady-state algorithm's efficiency, we compare the plastic strain state resulting from the step-by-step and steady-state algorithms, as well as the computation time.

Based on a 3D model and the elastic–plastic behavior described in the following sections, we compute the final stress–strain state in two structures after one loading cycle, including plastic strain. The first model is an approximate non-periodic structure and the second is a detailed periodic structure presenting a detailed railway track including rails and sleepers.

3.1. Railway Track Model

The studied models are two 3D models of a railway track. We used the FEM (Finite Element Method) software COMSOL Multiphysics for the model settings and FEM elastic resolution. All plastic strain computation algorithms were programmed in a Matlab script interacting with the COMSOL software. Details of the Comsol-Matlab interaction are given in Appendix A.1. All computations were conducted in same the software and under the same hardware conditions.

While the step-by-step algorithm could be built with preexisting functions mostly in the FEM software, steady-state algorithms require specific computation processes, which were built externally with Matlab. For the sake of the comparison in the following sections, we used the same external computational principle with Matlab for the step-by-step and steady-state algorithms.

3.1.1. Three-Dimensional Models

Three-dimensional models were built directly within the COMSOL Multyphysics model builder. The model represents half of a linear track because of the symmetric structure assumption. The 3D models were built with an extruded transversal 2D shape, as shown in Figure 11. The track model uses a 50 cm ballast sublayer and single-block sleepers with a height of 25 cm height and length of 1.1 m (2.2 m for the full structure). On the side of the sleepers, ballast is arranged as a 50 cm shoulder followed by a slope with a 2/3 ratio. A steel beam can be added (6 cm $\times$ 12.28 cm), with no contact with the ballast, in order to represent a UIC60 rail with the same flexural inertia.

This 2D model is extruded to build a 3D structure with or without distributed sleepers, depending on the need. In this case, sleepers are 25 cm wide and planted every 60 cm along the structure. The spaces between the sleepers are filled with ballast, which forms the ballast top layer. The final structures represent a 15 m track (with 25 sleepers). Due to our

assumption of a half model, a symmetric constraint is used at the middle of the structure. The faces below and end faces are planar-constrained. The resulting 3D model is presented in Figure 12.

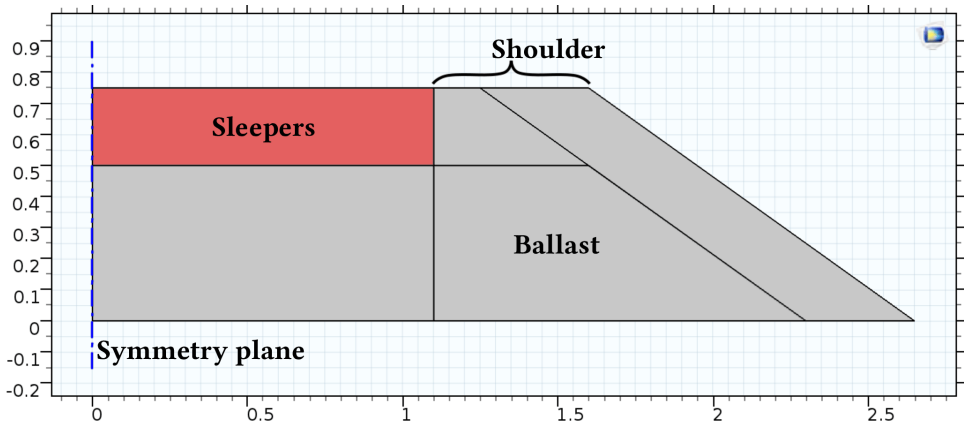
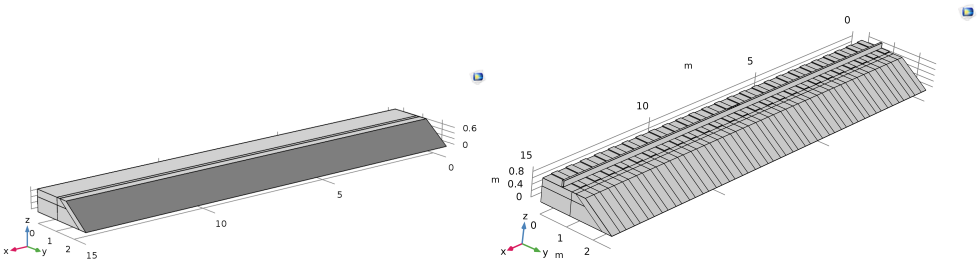


Figure 11. The 2D transversal shape representing the model construction.



(**a**) Non-periodic model (**b**) Periodic model

Figure 12. The 3D model resulting from 2D shape extrusion to form a (**a**) non-periodic model and (**b**) periodic model.

Because of the computation methods, we used a structured quadrilateral mesh, as described previously. Mesh elements have to be parallel to the structure axis to ensure the alignment of the Gauss point. Also, for periodic structures, the mesh has to be regular all along the structure. For non-periodic structures, the mesh's elements can be denser near the load and larger at the edge of the model.

The load is equal to half of one axle's maximum load, i.e., 10 t on one symmetrical half-track. In the case of the periodic model, the load is located on top of the rail. Then, the rail's flexibility shared the load between sleepers. In the case with the simplified non-periodic model, the load is imposed directly on ballast layers according to a precomputed shearing pattern [31]. In both cases, the load abscissa X_L describes the loaded zone. Self weight also applies on the entire structure.

The step-by-step process for the non-periodic structure is described with 57 successive load positions. The same algorithm with the periodic structure uses 73 load positions. The non-periodic steady-state algorithm uses only one load position in the middle of the structure. The periodic steady-state algorithm uses six intermediate load positions.

3.1.2. Material Behaviors

We studied the railway structure with a continuous material assumption. The ballast is considered to have elastic–plastic behavior, described with linear elasticity, elastic–plastic criteria, and a plastic flow rule.

We assume infinitesimal strain, supposing an additive decomposition of the global strain tensor $\boldsymbol{\varepsilon}$ between the elastic strain $\boldsymbol{\varepsilon}^e$ and the plastic strain $\boldsymbol{\varepsilon}^p$ (Equation (8)).

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^p \tag{8}$$

The elastic strain tensor is linearly linked with the stress tensor $\boldsymbol{\sigma}$ by Hooke's equation, using Young's modulus E and Poisson's ratio ν. Elastic strains are recoverable, i.e., in an unloaded situation, its tensor's values return to zeros.

The plastic strains are, conversely, irrecoverable, i.e., the tensor keeps its value after the unloading of the structure. Its evolution depend on the stress history, and is described using an elastic-plastic criterion and a flow rule.

For the purpose of this example, we used a simple Drucker–Prager criterion [32]. It is defined with the stress function of Equation (9a), with mean stress σ_m and shear stress amplitude $\sqrt{J_2}$. The Drucker–Prager parameters are α and H, linked to the classic Mohr–Coulomb parameters. According to previous works [33], we chose the compression-fitted parameters (Equation (9b)) in order to limit the modeling discrepancies.

$$f(\boldsymbol{\sigma}) = 3\alpha(\sigma_m - H) + \sqrt{J_2} \tag{9a}$$

$$\alpha = \frac{2\sin\phi}{\sqrt{3}(3-\sin\phi)} \text{ and } H = \frac{C}{\tan\phi} \tag{9b}$$

For simplification, we assumed an associated flow rule, i.e., the criterion yield function $f(\boldsymbol{\sigma})$ and the plastic potential function $g(\boldsymbol{\sigma})$ are equal.

All other material behaviors are described with linear elastic behavior.

It should be noted that a perfectly chosen plastic Drucker–Prager model is considered an oversimplification of the material behavior. This model choice was made in order to focus the discussion on algorithm efficiency. However, the presented developments are perfectly compatible with more complex material behavior, including hardening and material degradation.

3.1.3. Material Parameters

Because we focus on ballast settlement, only the ballast under-layer is considered with elastic–plastic behavior. The top ballast layer is considered with elastic behavior.

The rail's steel and sleeper's concrete proprieties are taken from the COMSOL Multiphysics material library. The steel is modeled with $E = 205$ GPa, $\nu = 0.28$, and density $\rho = 7.85$ t.m^{-3}. The concrete is modeled with $E = 25$ GPa, $\nu = 0.2$, and $\rho = 2.3$ t.m^{-3}.

The ballast's parameters are taken from Profillidis's study [31] i.e., $E = 110$ MPa, $\nu = 0.2$, and $\rho = 1.8$ t.m^{-3}. Equivalent parameters can be found in many other studies.

We use the Mohr–Coulomb parameters, with ϕ taken to be 40° according to Suiker's studies [34] and Bernard et al.'s tests [35]. The parameter C is supposedly null due to the cohesion-less behavior of the ballast. But we take $C = 5$ Pa, negligible, to avoid numerical issues during the computation process.

4. Results and Comparison

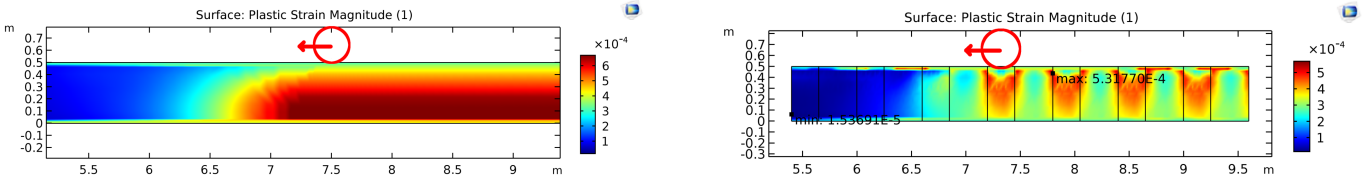
In this section, we present results comparatively. Numerical results are presented in the following figures to show the magnitude of the plastic strain, i.e., : $\epsilon^p = \sqrt{\boldsymbol{\varepsilon}^p : \boldsymbol{\varepsilon}^p}$. Other measures, such as displacement, would lead to the same comparison results and conclusion.

4.1. Periodic vs. Non-Periodic Structure

Firstly, in Figure 13, we present steady-state results for both periodic and non-periodic structures. The plastic strain magnitude is plotted on a longitudinal slice in the center of each structure. These representations are helpful for understanding the steady-state algorithm. In this representation, the load is located in the middle of the track and it moves from the right side to the left side. On the left of the load, we observe the plastic strain state to be representative of the structure before the loading cycle. On the right, we observe the plastic strain state resulting from the loading cycle.

In this representation, we can follow the evolution of the structure during the loading cycle by reading the plastic strain state from left to right.

It should be noted that, due to the simple model assumption, the plastic strain amplitudes are not representative of real settlement.



(**a**) Non-periodic structure (**b**) Periodic structure

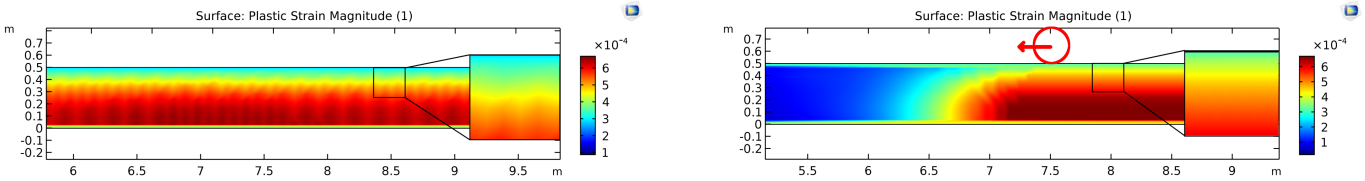
Figure 13. Longitudinal representation of plastic strain state resulting from steady-state algorithms on both the (**a**) non-periodic and (**b**) periodic model.

With the steady-state algorithm, we can focus on a representative section of the structure at the back end of the model. According to the computation process, the stress–strain state computed in this section corresponds to the result of a full loading–unloading cycle. In contrast, with the step-by-step algorithm, we focus on a central section, which, at the final step, has gone through a full loading cycle. Following this, the results are compared in a pertinent way.

Comparing results with those of two structures allows us to observe particularities and interesting periodic structures. This results in the localization of the plastic strain under sleepers with a particular U-shaped core.

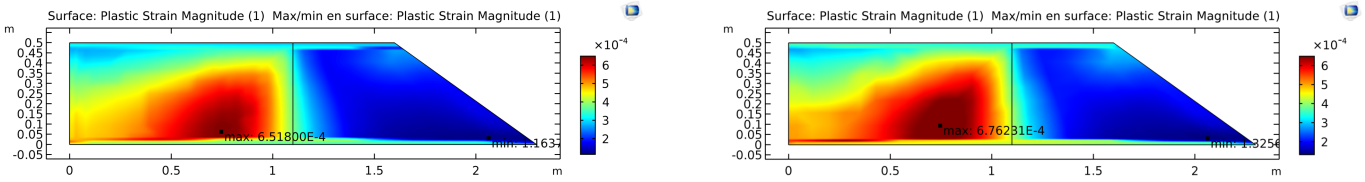
4.1.1. Steady-State vs. Step-by-Step Algorithms on Non-Periodic Structure

For the non-periodic structure, we compare a simple steady-state algorithm with a step-by-step algorithm. The results from comparable sections are shown in Figure 14 for longitudinal sections and and Figure 15 for transversal sections.



(**a**) Step-by-step results (**b**) Steady-state results

Figure 14. Longitudinal representation of plastic strain state in non-periodic structure for both (**a**) step-by-step and (**b**) steady-state methods.



(**a**) Step-by-step results (**b**) Steady-state results

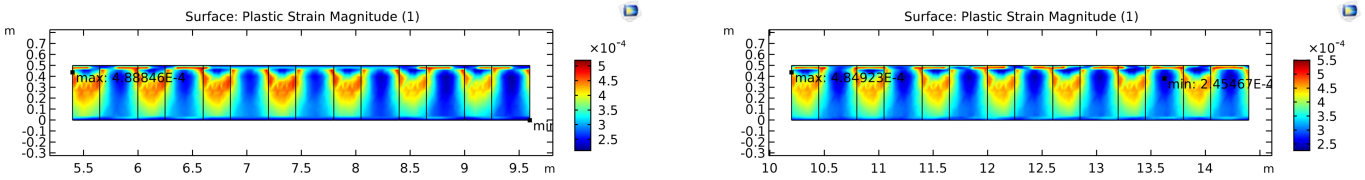
Figure 15. Transversal representation of plastic strain state in non-periodic structure for both (**a**) step-by-step and (**b**) steady-state methods.

For both comparisons, we can observe similar results. The plastic strain zones have a similar shape and the same amplitudes. For example, the maximal plastic strain magnitude plotted in Figure 15 is $\epsilon^p = 6.52 \times 10^{-4}$ and $\epsilon^p = 6.76 \times 10^{-4}$ for the step-by-step and steady-state algorithm, respectively.

However, we observe a particular result with the step-by-step process. It presents irregularities that do not appear with the steady-state algorithm. We can explain these irregularities by the artificial jump of the load described by the computational process. This results in a default continuous loading with the step-by-step algorithm. Such defaults are, by construction, absent from the steady-state algorithm.

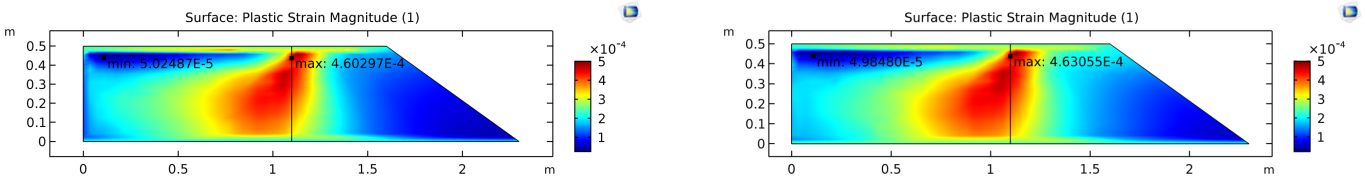
4.1.2. Steady-State vs. Step-by-Step Algorithms on Periodic Structure

For the periodic structure, we compare the periodic steady-state algorithm and step-by-step algorithm. The results are plotted and compared on a longitudinal slice in Figure 16 and on a transversal slice in Figure 17.



(**a**) Step-by-step results (**b**) Steady-state results

Figure 16. Longitudinal representation of plastic strain state in periodic structure for both (**a**) step-by-step and (**b**) steady-state methods.



(**a**) Step-by-step results (**b**) Steady-state results

Figure 17. Transversal representation of plastic strain state in periodic structure for both (**a**) step-by-step and (**b**) steady-state methods.

Again, in both cases, the comparison yields similar results in terms of plastic strain shape and amplitude. The periodic pattern induced by the sleepers is the same as in the longitudinal results. In transversal results, the maximal plastic strain magnitudes are similar, with $\epsilon^p = 4.60 \times 10^{-4}$ and $\epsilon^p = 4.63 \times 10^{-4}$ for the step-by-step and steady-state algorithm, respectively.

In order to numerically evaluate the discrepancies between the results of both algorithms in this second case, we compare the plastic strain tensor, plotting the value of the plastic strain discrepancies, defined in Equation (10), on a periodic element.

$$\delta\varepsilon_p = \frac{|\boldsymbol{\varepsilon}_1^p - \boldsymbol{\varepsilon}_2^p|}{|\boldsymbol{\varepsilon}_1^p|} \tag{10}$$

Figure 18 shows these results in one periodic section of the track. We can observe that numerical artifacts on the edges of the model show important discrepancies. However, beyond these artifacts, the discrepancies are under 1% in the major part of the models. Overall, the average discrepancy is only 3.2%, which is a comparable result between both algorithms.

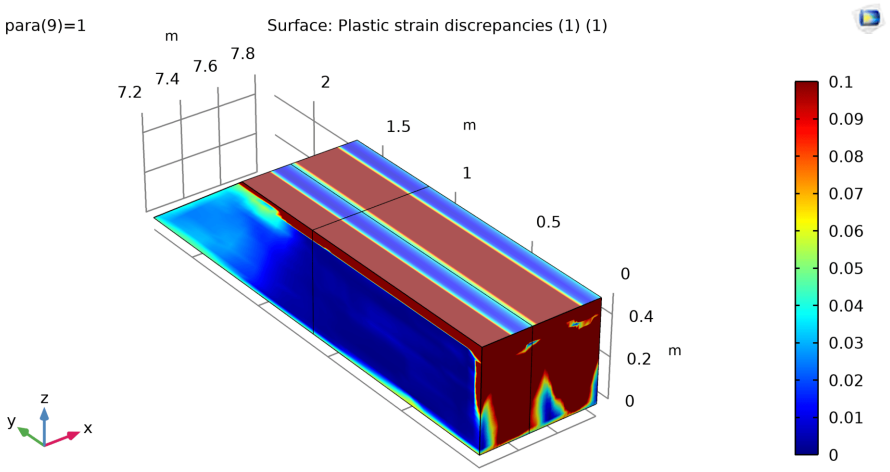


Figure 18. Discrepancies between plastic strain from both algorithm in a periodic section.

5. Discussion

The comparisons presented in the previous section are meaningful and highlight the accuracy of the steady-state methods compared to classical step-by-step approaches.

Given equal results, the efficiency of steady-state algorithms can be established by comparing the amount of stored data and the computation times.

Comparing the step-by-step and stead-state algorithms on a non-periodic structure gives meaningful results. The step-by-step algorithm yields 57 files, for a total of 106.6 Mb. The total amount of computation time needed is 395 ks. For the same model and with the same computer resources, the steady-state algorithm yields only one file of 1.9 Mb and a total computation time of 5.7 ks, equivalent to 1.5% of the computation time needed by the step-by-step algorithm.

These results are logical according to the construction of each algorithm. The step-by-step algorithm consists of the complete determination of 57 different plastic strain states corresponding to the 57 load positions. For the steady-state algorithm, only one plastic strain state is determined. Despite the particular computation process of the steady-state algorithm, it almost corresponds to a single-step computation.

Comparing the step-by-step algorithm to the steady-state algorithm on periodic structure lead to a similar conclusion.

For this structure, the step-by-step algorithm uses 73 steps. This results in 73 saved files for a total of 288 Mb stored. The 73 computation steps are conducted with an average of 23 iterations each and a total computation time of 316 ks. In contrast, the periodic steady-sate algorithm simultaneously computes six steps in 36 iterations. This results in six saved files for a total of 22.7 Mb stored and a total computation time of 41.4 ks.

These numbers also show an impressive decrease in the computation time, reducing it to 13% of its initial value. The computational time consumption is again almost linear to the number of computational step. However, for a periodic structure, the complexity of the computation process leads to an increase in computation time per step ratio. Such a decrease was not observed with the non-periodic structure.

In any case, both comparisons are meaningful. In both cases, step-by-step algorithms are much more expensive in terms of computation time than steady-state algorithms. With reasonable simulation parameters, steady-state algorithms are computationally quicker than step-by-step methods and allow storage space saving.

It should be noted that the presented steady-state algorithm and periodic steady-sate algorithm rely on restrictive assumption. The range of applicable structure is limited to self-similar, or periodical self-similar structures. Also, the presented versions of these algorithms limit the loading case to a mobile load at constant speed and constant magnitude. While the variety of cases studies is limited, it is perfectly suitable for the study of linear transportation infrastructure like railway tracks.

6. Conclusions

In conclusion, the numerical study of linear infrastructure requires particular consideration. The plastic strain computation is highly dependent on the stress–strain path description. Moreover, the mobile loading of such a linear structure induces a particular stress path, including variation in both stress magnitude and orientation. These particularities cannot be correctly modeled with a static load of variable amplitude. Consequently, the numerical modeling of railway tracks and mobile loads requires a specific process.

In this study, we present two types of computational algorithms that allow the representation of a moving load. The first one is a step-by-step algorithm. It describes the temporal evolution of the load position. Therefore, stress amplitude and orientation naturally evolve to properly describe the loading cycle. The other algorithms are steady-state algorithms. They aim to observe the loading cycle from the load's point of view. The structures are then viewed as flowing under the load. The steady-state algorithms also use the self-similitude of the linear structure. Therefore, a particular computation process allows for artificial movement of the structural state under the load in order to compute the final plastic strain state within a single step. An adaptation for the periodic structures of the steady-state algorithm is also presented.

To demonstrate the efficiency of the steady-state algorithm, two elastic–plastic models of a railway are described: one periodic and one non-periodic. Each model represents half of a symmetrical longitudinal railway structure. A ten-tonnes load was used to represent half of the axle load. For this example, the ballast was modeled with a simple Drucker–Prager elastic–plastic model.

The comparison of steady-state and step-by-step results provides convincing results. For both periodic and non-periodic structures, the results are similar in terms of topology (plastic strain zones) and magnitude. In both cases, transversal and longitudinal plastic strain representations show identical results within numerical tolerance. In terms of efficiency, each algorithm demonstrates performance corresponding to its construction. The number of computational steps is the key parameter needed to save on computational time. For a non-periodic structure, computation time is directly proportional to the number of computation steps. The steady-state algorithm is then equivalent to a single computation step. For the periodic structure, the steady-state algorithm needs a computation time nearly proportional to the number of computation steps compared to the step-by-step algorithm.

Steady-state algorithms also reduce the amount of stored data. For non-periodic structure, as the single final result is enough to describe a full loading cycle by following the integration

lines. The same information would be obtained with the step-by-step algorithm, but would require the results of all computation steps and proper data processing. This multiplies the amount of stored data and increases the complexity of interpreting the results.

Finally, the steady-state algorithms are extremely efficient in comparison to a classical algorithm. Without the degradation of results, these algorithms reduce the computation time by up to 98% and generally scale linearly with the number of computational steps.

The results presented in this paper open new methodological possibilities for the study of the long-term behavior of transportation infrastructure, particularly railways tracks. Even when cutting computational costs, accurate behavior prediction under high number of loading cycles would still be possible within a reasonable amount of time. Further work should use the proposed improvements with advanced behavior models, taking into account material hardening, complex plastic phenomena, or material degradation, in order to obtain insight into ballasted track degradation.

Author Contributions: Conceptualization, T.B., S.M. and H.M.; methodology, T.B. and S.M.; software, T.B.; validation, T.B. and S.M.; original draft preparation, T.B.; supervision, H.M.; project administration, S.M. All authors have read and agreed to the published version of the manuscript.

Funding: Work presented in this paper was developed as part of Thibault Badinier's PhD thesis. The PhD was funded by the French Ministry of ecological transition.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author due to privacy.

Conflicts of Interest: The authors declare no conflicts of interest.

Nomenclature

α	Friction parameters for Drucker–Prager criterion
C	Cohesion of Mohr–Coulomb
E	Young's modulus
$\boldsymbol{\varepsilon}$, $\boldsymbol{\varepsilon}^p$, and $\boldsymbol{\varepsilon}^e$	Strain tensor, plastic strain tensor, and elastic strain tensor
ϵ^p	Plastic strain magnitude
$\boldsymbol{\varepsilon}^p_n$	Plastic strain tensor at step n
$\boldsymbol{\varepsilon}^p_{i,t}$	Plastic strain tensor at position i and at time t
$\boldsymbol{\varepsilon}^p_{-,t}$	Overall plastic strain field at time t
ϕ	Internal friction angle
$f(\boldsymbol{\sigma})$	Elastic limit stress function
$g(\boldsymbol{\sigma})$	Plastic potential function
H	Cohesion pressure
L	Number of Gauss points along a line
λ or Λ	Plastic strain multiplicative factor
N	Number of computational steps
ν	Poisson's ratio
P	Number of periodic sections
ρ	Material density
$\boldsymbol{\sigma}$	Stress tensor
T	Number of time steps
t, Δt	Time and time increment
$\vec{V}$	Displacement velocity vector
$\vec{x}$	Main axis of the structure and the displacement
X_n	Load coordinate along the $\vec{x}$ axis at step n

Appendix A. Algorithm Description

Appendix A.1. Step-by-Step Algorithm

The step-by-step algorithm aims to describe a number of successive stress–strain states. Each step uses an iterative process and return mapping process for plastic strain computation. The entire algorithm considers each step one after another.

At the first time step $t = 0$ ($n = 1$), the load is located at abscissa X_1. The initial plastic strain state $\boldsymbol{\varepsilon}^p_{-,1} = \boldsymbol{\varepsilon}^p_{-,Init}$ is taken as null on the overall structure, in the case of the first loading. Otherwise, the initial plastic strain state is described using the representative state at the end of the previous cycle.

At any other time step n, the structure starts with an initial global and acceptable $\boldsymbol{\varepsilon}^p_{-,n-1}$ plastic strain state resulting from the computation of the previous step $n-1$. The load is moved at abscissa X_n.

The algorithm begins an iterative process, with iteration $i = 1$. The process initially assumes the plastic strain state is identical to the plastic strain state resulting from the previous step: $\boldsymbol{\varepsilon}^p_{-,n} = \boldsymbol{\varepsilon}^p_{-,n-1}$.

At iteration i, the supposed global plastic strain state $\boldsymbol{\varepsilon}^p_{-,n}$ is used to compute the stress–strain state according to the Finite Element Method and Equation (A1).

$$\boldsymbol{K} \cdot \boldsymbol{u} = \boldsymbol{F} \tag{A1a}$$

$$\boldsymbol{\sigma} = \boldsymbol{L} \cdot \boldsymbol{\varepsilon}^e \tag{A1b}$$

with $\boldsymbol{K}$ as the FEM rigidity matrix, $\boldsymbol{F}$ as the generalized force tensor, $\boldsymbol{u}$ as the generalized displacement vector, $\boldsymbol{\sigma}$ as the stress tensor, $\boldsymbol{\varepsilon}^e$ as the elastic strain tensor, and $\boldsymbol{L}$ as the materials' rigidity matrix.

Plastic strain computation is conducted on each independent integration point using the return mapping process describe by Simo and Hughes (1998) [12]. If stress $\boldsymbol{\sigma}$ is acceptable, i.e., $f(\boldsymbol{\sigma}) < 0$, the behavior remains elastic, and the plastic strain does not evolve for the next iteration (Equation (A2a)). Otherwise, stress $\boldsymbol{\sigma}$ is not acceptable. The plastic strain state is recomputed from the initial plastic strain state (A2b).

$$\boldsymbol{\varepsilon}^p_n = \boldsymbol{\varepsilon}^p_n \tag{A2a}$$

$$\boldsymbol{\varepsilon}^p_n = \boldsymbol{\varepsilon}^p_{n-1} + \lambda \frac{\mathrm{d}g(\boldsymbol{\sigma})}{\mathrm{d}\boldsymbol{\sigma}} \tag{A2b}$$

If the stress state is globally acceptable, the global plastic strain state $\boldsymbol{\varepsilon}^p_n$ corresponds to the step's final plastic strain state. Then, the algorithm moves to the next loading state.

After the last step, the loading–unloading process of the central element is considered to have concluded. The representative plastic strain state is described with the final stress–strain state of the central section. It can be reproduced throughout the entire structure to define the global plastic strain state after the loading cycle.

This entire computational process can be summarized with the algorithm shown in Figure A1.

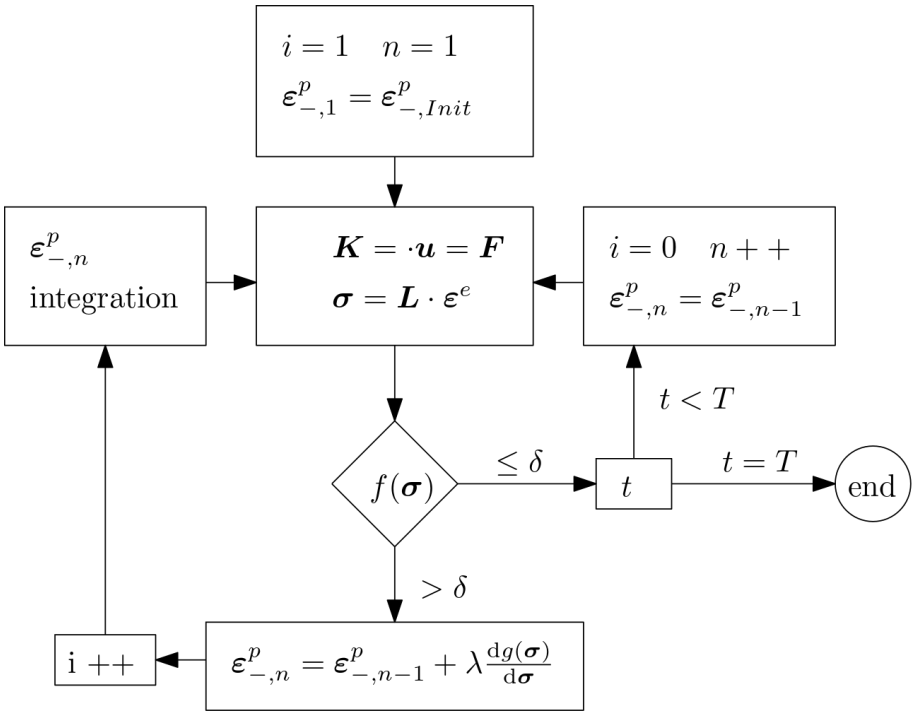


Figure A1. Step-by-step algorithm, detailed computation process.

Appendix A.2. Steady-State Algorithm

For the first loading cycle, the structure is set with zero plastic strain as the initial plastic strain state $\boldsymbol{\varepsilon}^p_{Init}$. Otherwise, the initial plastic strain state is described using a representative state at the end of the previous cycles. Then, the algorithm begins the iterative process.

At the beginning of an iteration i, the previously computed plastic strain state is supposed to be acceptable and the stress–strain state is computed according to the FEM process and Equation (A1).

By construction, the Gauss points are lined up in the structure and numbered from 1 to L on each line by increasing the abscissa. The plastic strain computation is conducted on each independent integration line.

If the stress σ is acceptable on the entire line, the plastic strain does not evolve for the next iteration (Equation (A3)). Otherwise, if the stress σ is not acceptable at some point on the line, the plastic strain state is recomputed on the entire line. The plastic strain state at the first point is described as the initial plastic strain. Then, the computational process computes plastic strain at point n by transferring the plastic strain from Gauss point $n-1$ and computing the plastic strain increment according to Equation (A4).

$$\boldsymbol{\varepsilon}^p_- = \boldsymbol{\varepsilon}^p_- \tag{A3}$$

$$\boldsymbol{\varepsilon}^p_1 = \boldsymbol{\varepsilon}^p_I \tag{A4a}$$

$$\boldsymbol{\varepsilon}^p_n = \boldsymbol{\varepsilon}^p_{n-1} + \Lambda \frac{\mathrm{d}g(\sigma)}{\mathrm{d}\sigma} \tag{A4b}$$

If the stress state is globally acceptable throughout the structure, the computation process is complete and the global plastic strain state $\boldsymbol{\varepsilon}^p_-$ is validated for the current loading cycle.

The entire computation process can be summarized with the algorithm shown in Figure A2.

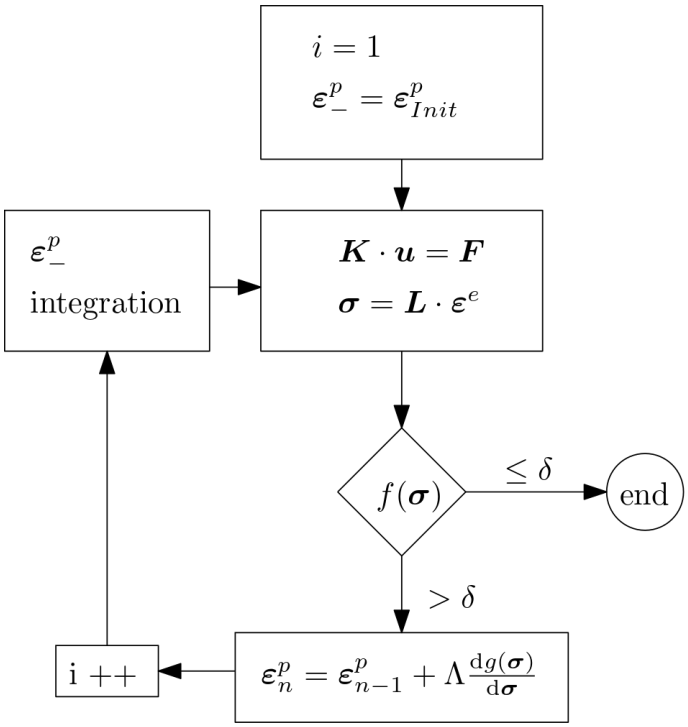


Figure A2. Steady-state algorithm, detailed computation process.

The validated global plastic strain state allows us to follow the plastic strain evolution during the loading cycle by simply measuring it along an integration line. The plastic strain state computed for a representative section at the back of the load and near the end of the structure is considered representative of the final global plastic strain state after the loading cycle.

Appendix A.3. Periodic Steady-State Algorithm

For the periodic steady-state method, we consider a central load on T different and consecutive positions covering one periodic section of the structure.

For the first loading cycle, the structure is set with zero plastic strain as the initial plastic strain state ε^p_{Init}. Otherwise, the initial plastic strain state is described using the representative state at the end of the previous cycles. The initial plastic strain state is set the same for the T loading step computations :$\varepsilon^p_{-,t} = \boldsymbol{\varepsilon}^p_{Init}$. Then, the algorithm begins the iterative process.

At iteration i, the T global plastic strain states $\boldsymbol{\varepsilon}^p_{-,t}$ are assumed to be acceptable. T global stress–strain states are then computed according to the FEM process and Equation (A1).

The plastic strain computation is conducted on each independent integration line. If stress σ is acceptable on the entire line, for each time t, the plastic strain does not evolve for the next iteration (Equation (A5)).

$$\boldsymbol{\varepsilon}^p_{-,t} = \boldsymbol{\varepsilon}^p_{-,t} \tag{A5}$$

If stress σ is not acceptable at some point on the line and/or at some time t, the plastic strain state is recomputed on the entire line. The first periodic element state at the first computational time is set equal to the initial plastic strain state. Then, the computational process transfers the plastic strain from point n at time t (n,t) to point n at time $t+1$ (n,$t+1$) if $t \neq T$. Otherwise, it transfers (n,T) to the corresponding point, $n+N$, in the next element

of the structure at time $t = 1$ ($n + N$,1) according to Equation (A6). The T plastic strain states are then computed simultaneously.

$$\boldsymbol{\varepsilon}^{p}_{n,t} = \boldsymbol{\varepsilon}^{p}_{n,t-1} + \Lambda \frac{\mathrm{d}g}{\mathrm{d}\boldsymbol{\sigma}} \quad \text{if } t \neq 1 \tag{A6a}$$

$$\boldsymbol{\varepsilon}^{p}_{n,t} = \boldsymbol{\varepsilon}^{p}_{n-N,T} + \Lambda \frac{\mathrm{d}g}{\mathrm{d}\boldsymbol{\sigma}} \quad \text{if } t = 1 \tag{A6b}$$

If the stresses are globally acceptable all over the structure on every time t steps, the T global plastic strain states $\boldsymbol{\varepsilon}^{p}_{-,t}$ are validated.

The entire computation process can be summarized with the algorithm in Figure A3.

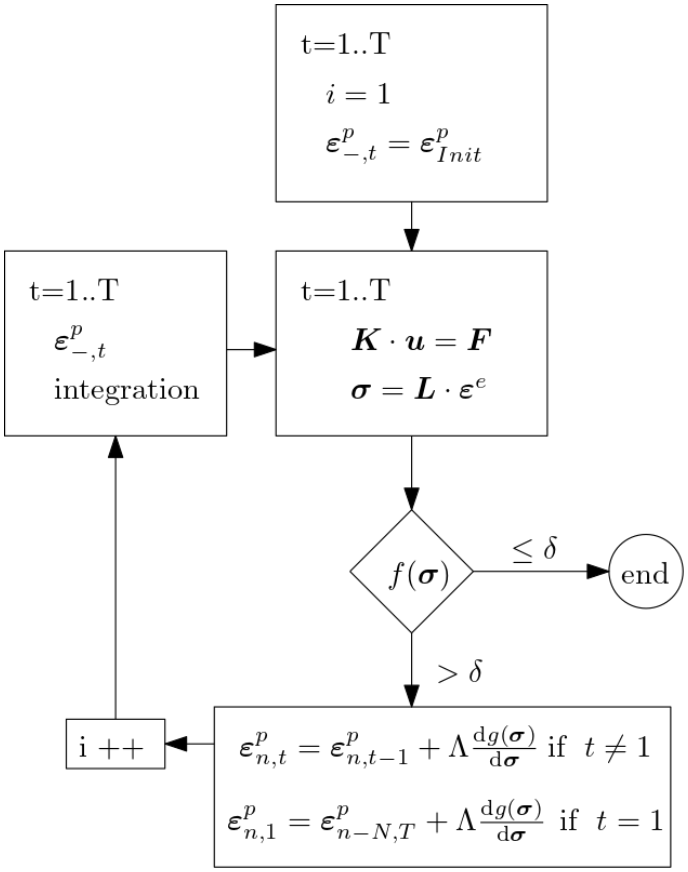


Figure A3. Periodic steady-state algorithm, detailed computation process.

The T validated plastic strain states obtained through this process allow us to follow the plastic strain state evolution during loading by composing measurements along a line at the different time steps.

The final plastic strain state in a representative periodic section at the back of the load near the end of the structure and at time T is considered representative of the global plastic strain state resulting from the loading cycle.

Appendix A.4. Matlab–Comsol Co-Processing

The step-by-step process was achieved within the classical FEM software interface by constructing successive computation, activating and deactivating the load at different positions. However, the steady-state algorithms require a specific integration process, computing plastic strain at Gauss points in a specific order. Therefore, the iteration and return mapping processes were reproduced and adapted with the help of a Matlab (version 2018a) script and COMSOL (version 5.3a) to Maltlab LiveLink capabilities.

The co-processing proceeded as follows:

- The COMSOL model, parameters, and loading were set within the COMSOL interface.
- The initial plastic strain state was set in Matlab in a file containing Gauss point coordinates and tensor information.
- The iteration process was started.
- The plastic strain was loaded as inelastic, imposed strains in the COMSOL model, and the FEM computation was processed by the COMSOL routine.
- The global strain was extracted from the FEM computation result in the Matlab script at each Gauss point.
- The stress state at each Gauss point was computed according to the global strain tensor and the known plastic strain. The stress at a Gauss point was sorted along the integration line by the Matlab script.
- The stress state was verified according to the plastic criterion by the Matlab script, possibly according to the integration line, as explained earlier.
- If needed, the plastic strain was recomputed according to the appropriate process described earlier within the Matlab script.
- The resulting plastic strain state was saved by MATLAB in a new external file.
- The global plastic strain state could then be reloaded into COMSOL for another iteration.
- If the global plastic strain state was acceptable, the iteration process was considered over.
- The computation process could then move to another loading step.

References

1. Esveld, C. *Modern Railway Track*; MRT-Productions: Zaltbommel, The Netherlands, 2001; Volume 385.
2. Ngamkhanong, C.; Kaewunruen, S.; Baniotopoulos, C. A review on modelling and monitoring of railway ballast. *Struct. Monit. Maint.* **2017**, *4*, 195.
3. Alabbasi, Y.; Hussein, M. Geomechanical modelling of railroad ballast: A review. *Arch. Comput. Methods Eng.* **2021**, *28*, 815–839. [CrossRef]
4. Lassoued, R.; Guettiche, A. Mechanical behaviour of railway track. *Phys. Procedia* **2011**, *21*, 166–173. [CrossRef]
5. García-Palacios, J.; Samartín, A.; Melis, M. Analysis of the railway track as a spatially periodic structure. *Proc. Inst. Mech. Eng. Part F J. Rail Rapid Transit* **2012**, *226*, 113–123. [CrossRef]
6. Merwin, J.E.; Johnson, K.L. An analysis of plastic deformation in rolling contact. *Proc. Inst. Mech. Eng.* **1963**, *177*, 676–690. [CrossRef]
7. Suiker, A.S.; de Borst, R. A numerical model for the cyclic deterioration of railway tracks. *Int. J. Numer. Methods Eng.* **2003**, *57*, 441–470. [CrossRef]
8. Gallego, I.; Muñoz, J.; Sánchez-Cambronero, S.; Rivas, A. Recommendations for numerical rail substructure modeling considering nonlinear elastic behavior. *J. Transp. Eng.* **2013**, *139*, 848–858. [CrossRef]
9. Jiang, Y.; Nimbalkar, S. Finite element modeling of ballasted rail track capturing effects of geosynthetic inclusions. *Front. Built Environ.* **2019**, *5*, 69. [CrossRef]
10. Gesta, P.; Kerisel, J.; Londe, P.; Louis, C.; Panet, M. Stability of tunnels by the "convergence-confinement" method. *Construction* **1979**, *11*, 17–24.
11. Halphen, B.; Salençon, J. *Elasto-Plasticité*; Presses de l'École nationale des ponts et chaussées: Champs-sur-Marne, France, 1987.
12. Simo, J.C.; Hughes, T.J. *Computational Inelasticity*; Springer: Berlin/Heidelberg, Germany, 1998.
13. Hasanpour, R.; Rostami, J.; Ünver, B. 3D finite difference model for simulation of double shield TBM tunneling in squeezing grounds. *Tunn. Undergr. Space Technol.* **2014**, 40, 109–126. [CrossRef]
14. Bourgeois, E.; Mohamad, W.; Szymkiewicz, F.; Le Kouby, A.; Kreziak, C. Three-dimensional finite element analysis of TBM influence on existing deep foundations: Application to the TULIP project. *Geomech. Geoengin.* **2025**, *20*, 376–389. [CrossRef]
15. Valsšková, V.; Melcer, J. Some possibilities of modeling of moving load on concrete pavements. *J. Meas. Eng.* **2018**, *6*, 203–209. [CrossRef]
16. Deng, Y.; Luo, X.; Gu, F.; Zhang, Y.; Lytton, R.L. 3D simulation of deflection basin of pavements under high-speed moving loads. *Constr. Build. Mater.* **2019**, *226*, 868–878. [CrossRef]
17. Paixão, A.; Varandas, J. N.; Fortunato, E.; Calçada, R. Non-linear behaviour of geomaterials in railway tracks under different loading conditions. *Procedia Eng.* **2016**, *143*, 1128–1135. [CrossRef]

18. Jing, G.; Wang, J.; Wang, H.; Siahkouhi, M. Numerical investigation of the behavior of stone ballast mixed by steel slag in ballasted railway track. *Constr. Build. Mater.* **2020**, *262*, 120015. [CrossRef]
19. Varandas, J.N.; Paixão, A.; Fortunato, E.; Coelho, B.Z.; Hölscher, P. Long-term deformation of railway tracks considering train-track interaction and non-linear resilient behaviour of aggregates—A 3D FEM implementation. *Comput. Geotech.* **2020**, *126*, 103712. [CrossRef]
20. Mezeh, R.; Mroueh, H.; Hosseingholian, M.; Sadek, M. Fully-coupled numerical model for ballasted track analysis–Field measurements and predictions. *Transp. Geotech.* **2021**, *27*, 100483. [CrossRef]
21. Nguyen, Q.S.; Rahimian, M. Mouvement permanent d'une fissure en milieu élasto-plastique. *J. Mécanique Appliquée* **1981**, *5*, 95–120.
22. Van, K.D.; Inglebert, G.; Proix, J. Sur un nouvel algorithme de calcul de structure élastoplastique en régime stationnaire. In Proceedings of the 3ème Colloque:" Tendances Actuelles en Calcul de Structures", Bastia, France, 6–8 november 1985.
23. Maitournam, H. Résolution Numérique des Problèmes Élastoplastiques Stationnaires. Ph.D. Thesis, ENPC, Paris, France, 1989.
24. Dang Van, K.; Maitournam, M. Steady-state flow in classical elastoplasticity: Applications to repeated rolling and sliding contact. *J. Mech. Phys. Solids* **1993**, *41*, 1691–1710. [CrossRef]
25. Ouakka, A. Approches Stationnaires en Mécanique non Linéaire. Ph.D. Thesis, ENPC, Marne-la-vallée, France, 1993.
26. Geoffroy, H. Etude de L'interaction Roche/Outil de Forage: Influence de L'usure sur les Paramètres de Coupe. Ph.D. Thesis, Ecole Polytechnique, Palaiseau, France, 1996.
27. Nguyen-Tajan, T.M.L. Modélisation Thermomécanique des Disques de Frein par une Approche Eulérienne. Ph.D. Thesis, Ecole Polytechnique, Palaiseau, France, 2002.
28. Corbetta, F. Nouvelles Méthodes D'étude des Tunnels Profonds: Calculs Analytiques et Numériques. Ph.D. Thesis, ENMP, Paris, France, 1990.
29. Maiolino, S. Fonction de Charge Générale en Géomécanique: Application aux Travaux Souterrains. Ph.D. Thesis, Ecole Polytechnique, Palaiseau, France, 2006.
30. Defay, A.; Maïolino, S.; Maïtournam, H.; Subrin, D. Numerical simulation of excavation in rock masses with the Hoek-Brown criterion. In Proceedings of the ISRM International Symposium, EUROCK, physical event not held, 14–19 June 2020; ISRM–EUROCK.
31. Profillidis, V.; Humbert, P. Étude en élastoplasticité par la méthode des éléments finis du comportement de la voie ferrée et de sa fondation. *Bull. Liaison Lab. Ponts Chaussées* **1986**, *141*, 13–19.
32. Drucker, D.C.; Prager, W. Soil mechanics and plastic analysis or limit design. *Q. Appl. Math.* **1952**, *10*, 157–165. [CrossRef]
33. Badinier, T.; Maïolino, S. Limiting Discrepancies in Substitution of Mohr-Coulomb with Fast Computation Smooth Criteria: Application to Ballast Layer. *Int. J. Railw. Technol.* **2018**, *7*, 41–66. [CrossRef]
34. Suiker, A.S.; Selig, E.T.; Frenkel, R. Static and cyclic triaxial testing of ballast and subballast. *J. Geotech. Geoenviron. Eng.* **2005**, *131*, 771–782. [CrossRef]
35. Bernard, A.; Peyras, L.; Royet, P. L'essai de cisaillement à la grande boîte de Casagrande: un banc expérimental pour évaluer les propriétés des sols grossiers et pour d'autres applications en géomécanique. *Rev. Fr. Géotech.* **2016**, *146*, 4. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.