Defining Creation in 1D Dislocation Dynamics

A study in theory and simulation of the extension of 1D discrete dislocation dynamics models with creation

Moraal, Jan

Award date:
2022

Eindhoven University of Technology
Department of Mathematics & Computer Science

Defining creation in 1D dislocation dynamics

A study in theory and simulation of the extension of 1D discrete dislocation dynamics models with creation

J. Moraal

Supervisor:
Prof. dr. M.A. Peletier

Eindhoven, May 2022

Abstract

We consider an interacting particle system with singular interaction, describing the dynamics of finite numbers of (charged) dislocations in one-dimensional space, including annihilation. In this thesis, we formalise several possible methods for introducing creation into the formal description of such a system. This results in three definitions of solution concepts to these systems, for one of which we prove uniqueness and other properties. Moreover, we describe a framework in which long-range effects of different creation methods can be compared, and draw conclusions on the effect of our developed creation methods. In addition, we describe basic numerical simulation of the developed system, which is non-trivial (even with automatic ODE-solvers) due to the intricacies of creation.

Contents

1 Introduction

The study of dislocations in lattice structures gives important insights into the behaviour and material properties of metals and other materials under stress. Dislocations are defects in regular lattice structures, such as the atoms in a crystal or metal. We specifically consider edge dislocations, the phenomenon where a plane of atoms is not continued in the way that is regular for the lattice. Figure 1.1 visualises an example:

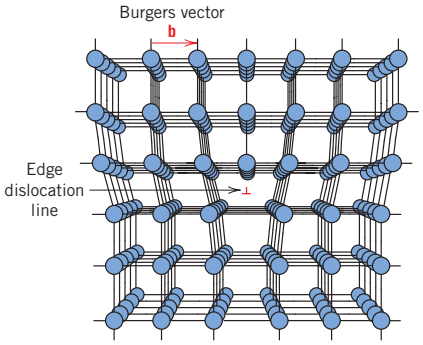


Figure 1.1: Example of an edge dislocation in a cubic lattice. [9]

In essence, dislocations can be seen as charged particles, attracting or repelling each other depending on the orientation of the dislocation; in particular, we model the defect between atoms, not the atoms themselves, as particles. When forces such as shear stress act upon the material, dislocations can move through the material as shown in Figure 1.2:

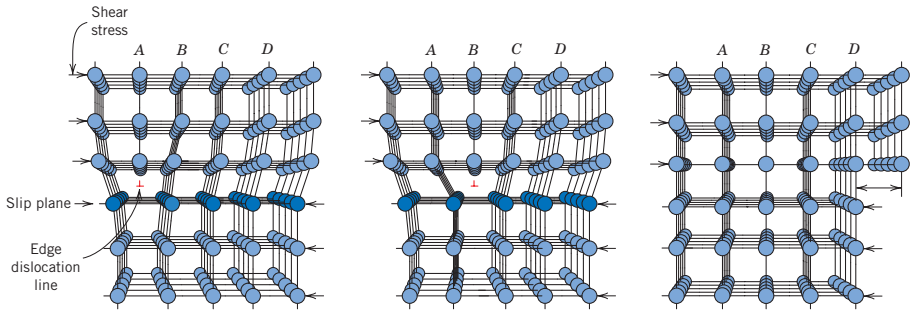


Figure 1.2: Slip-confined dislocation movement in cubic lattice. [9]

When two dislocations with half-planes extending on opposite sides (i.e. with opposite charges) meet, the half-planes connect and both dislocations disappear; this is known as annihilation. Under external forces, new dislocations can also appear; this is called creation or nucleation. Disregarding the boundary of the material, dislocations can only be created in pairs of opposite charge, which we refer to as dipoles.[1]

Considering edge dislocations extending throughout the material reduces the space from three to two spatial dimensions. Even more, dislocations often move within a cross-section of the material, on so-called slip planes; with this in mind, we restrict ourselves to studying the movement of dislocations

[1] In principle, several pairs could be created at a single point in time and space, as long as the net charge of all created dislocations is 0. We however restrict ourselves to the creation of one pair at a time.

in a single dimension. This movement is governed by the following (singular) ODE, where x_i is the position, and b_i the charge of dislocation i:

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = \sum_{j \neq i} \frac{b_i b_j}{x_i - x_j}.$$

1.1 Motivation

Evolutions of such dislocations have been studied extensively in various ways and on different scales. Currently however, only few mathematical models for dislocations include creation of new dislocations, whereas in reality this is an important aspect and defining for material properties; thus, most existing models fall short of physical representation. The studies that do cover creation usually do not give a precise and fundamental mathematical description of the process.

The main motivation for this thesis is that covering the aforementioned gap would be of great value for the understanding of dislocation behaviour and for physical accuracy. We aim to build a mathematical foundation beneath the numerous studies involving the creation of dislocations. In doing so, we conclude that it is possible to add creation to a mathematical model of dislocation motion in a mathematically precise manner, although the details require a lot of care. Moreover, we show that solutions to the resulting evolution equation are unique, and propose methods to simulate such systems.

There are ample theoretical studies of dislocation evolution in lattice structures, the setting that we also work in. Research is conducted on many different levels, ranging from experimental to purely theoretical. We will focus on the latter. Hull and Bacon provide a comprehensive overview of dislocation theory in [2]. Including creation and annihilation in studies of dislocation dynamics is not new, and neither is analysis of fundamental questions such as well-posedness of dislocation dynamics models. However, the combination of both is rarely found. In [1], Cleveringa, Van Der Giessen and Needleman study creation via Frank-Read sources, only more from an engineering point of view and with emphasis on numerical simulation; the choice of the used process is only briefly argued. In 2004, Yefimov, Groma and Van der Giessen continue the study, developing a two-dimensional continuum plasticity theory with creation.

Hudson studies dislocations on a more fundamental mathematical level. In [5] he discusses dislocations on a triangular lattice, and proves existence of a geometrically necessary dislocation. Later, in [7] he gives a well-posed formulation of Discrete Dislocation Dynamics (DDD) in 3D, with dislocation configurations represented on larger scale, as integral currents. The study gives an explicit expression of the Peach-Koehler force, but creation is not included.

On the side of continuum dislocation dynamics (CDD), Monavari and Zaiser study dislocation dynamics in terms of field variables (similar to densities), and in [8] describe both creation and annihilation of dislocations. On a more microscopic scale, in [6] Van Meurs studies discrete-to-continuum limits of gradient flows in interacting dislocations, both in 1D and 2D, and describes and analyses numerical simulation with regularised interaction potentials. Recently, together with Peletier and Pozar, in [10] he also formally defines a continuous dislocation dynamics model with annihilation as an interacting particle system in 1D, and proves well-definedness and properties of such a system. From there on, they study the system's many-particle limit.

The main novelty in our work is defining creation on the most fundamental level, considering specific dislocations and single creation events, and working towards a well-posed description of a

DDD system with creation. The system of interest throughout this thesis is the following:

$$\begin{cases} \dfrac{\mathrm{d}x_i}{\mathrm{d}t} = \displaystyle\sum_{j \neq i} \frac{b_i b_j}{x_i - x_j} \quad \text{for } t \in (0, T), i = 1, ..., n \\ \text{with annihilation upon collision} \\ \text{with creation via Frank-Read sources} \end{cases} \tag{PC_n}$$

Here the novelty will be in describing a creation process, as the system without creation has been studied extensively in other work (e.g. [10]).

The outline of our work is as follows. We develop a model for dislocation dynamics with creation of new dislocations in several phases. In Section 2 we first construct a general framework for the definition of solutions to (PC_n), leaving the exact details of creation methods open. Then in Section 3 we study specific creation processes separately, to fit into the previously built framework. Having properly defined different creation methods, in Section 4 we analyse and compare the different systems with creation and their solutions. We conclude our theoretical analysis by stating properties of one of the formal definitions of solutions to (PC_n). In Section 6 we then discuss the numerical simulation of such systems, and further analyse and compare their behaviour. Finally, we draw conclusions and reflect on our work.

2 The Solution Concept

In this section, we develop a general framework for a solution definition to PC_n. We first discuss the definition of a solution to (PC_n) but without creation; this will serve as a basis for our definition incorporating creation. Next, we study how the definition should be adapted to facilitate the incorporation of creation.

2.1 Solutions without creation

The same system as (PC_n) but then without creation is formally defined in [10], Section 2. Van Meurs et al. additionally scale the ODE by a factor $1/n$, but leaving out this term does not influence the proposed theory[1]. We state and briefly discuss this result here, as it serves as a basis for our further work. The system of interest may be given as follows:

$$\begin{cases} \dfrac{\mathrm{d}x_i}{\mathrm{d}t} = \displaystyle\sum_{j\neq i} \frac{b_i b_j}{x_i - x_j} & \text{for } t \in (0,T), i = 1, ..., n \\ \text{with annihilation upon collision.} \end{cases} \tag{P_n}$$

As state-space, we take the following subset of the cartesian product of all positions and charges:

$$\mathcal{Z}_n \;:=\; \left\{(x,b) \in \mathbb{R}^n \times \{-1,0,1\}^n \text{ s.t } \text{ if } i > j \text{ and } b_i b_j \neq 0, \text{ then } x_i > x_j\right\}$$

A solution to (P_n) is then a pair of n-dimensional vector functions of the form $(\mathbf{x}, \mathbf{b}) : [0,T] \to \mathcal{Z}_n$. The condition imposed here ensures ordering of charged dislocations (i.e. dislocations i with $b_i \neq 0$). Note that if $b_i = 0$, by the ODE in (P_n), $x_i' = 0$ and there is no interaction anymore between dislocation i and the rest of the system; hence, setting a dislocation's charge to 0 is equivalent to removing it from the system, and ordering is not relevant anymore.[2] We adhere to this convention of 'removing' dislocations from the system by setting their charge to 0, instead of re-defining the state space with a different number of dislocations.

Having defined the the state space, we arrive at the formal definition of a solution:

Definition 2.1.1 (Solution to (P_n))**.** *Let $n \geq 2$ and $(\mathbf{x}^0, \mathbf{b}^0) \in \mathcal{Z}_n$. Then $(\mathbf{x}, \mathbf{b}) : [0,T] \to \mathcal{Z}_n$ is a solution of (P_n) if there exists a finite subset $S \subset (0,T]$ such that*

(i) (Regularity) $\mathbf{x} \in C([0,T]) \cap C^1([0,T] \setminus S)$, and $b_1, \ldots, b_n : [0,T] \to \{-1,0,1\}$ are right-continuous;

(ii) (Initial condition) $(\mathbf{x}(0), \mathbf{b}(0)) = (\mathbf{x}^0, \mathbf{b}^0)$;

(iii) (Annihilation) Each b_i has at most one discontinuity[3]. If b_i is discontinuous at $t \in [0,T]$, then $t \in S$, $\lim_{s\uparrow t} |b_i(s)| = 1$ and $b_i(t) = 0$ (b is right-continuous). Moreover, for all $(\tau, y) \in S \times \mathbb{R}$,

$$\sum_{i\,:\,x_i(\tau)=y} \big((b_i(\tau+) - b_i(\tau-)\big) \;=\; 0.^4$$

[1] For details, see Appendix A.2.1

[2] This is an important modelling choice, and is specific to this setting with annihilation. Some studies leave dislocations in the same position with their original charges.

[3] Note that b_i is a function of time for all $i = 1, ..., n$.

[4] Here we use the convention of writing $f(t^*-) := \lim_{t\uparrow t^*} f(t)$ and $f(t^*+) := \lim_{t\downarrow t^*} f(t)$ for the left and right limits respectively.

(iv) (ODE for $\mathbf{x}$*) On* $(0,T)\backslash S$*,* $\mathbf{x}$ *satisfies the ODE in* (P_n)*.*

The choice of the set S is free, as long as it is finite and contains all annihilation times from point (iii). A reasonable choice would be to let S be exactly the set of annihilation times; this is referred to as the *minimal choice* for S.

Note the importance of the requirement that $(\mathbf{x},\mathbf{b}) \in \mathcal{Z}_n$; it is the only condition ensuring that dislocation charges are indeed set to 0 if two dislocations of opposite charge end up in the same position (i.e. upon collision). Condition (iii) ensures that the total charge before and after an annihilation is equal; in particular, equally many dislocations of both charges are needed to annihilate.

One of the main results achieved by Van Meurs et al. in [10, Theorem 2.4] is the well-posedness of (P_n), along with a range of properties.

Both as illustration and to serve future analysis, we study the above system more closely in a minimal example of two dislocations. This allows for explicit solutions up to a collision event via standard ODE theory, and will be useful for comparison later on, when introducing creation.

Example 2.1.2. Let $x_1^0, x_2^0 \in \mathbb{R}$ such that $x_1^0 < x_2^0$, and let $b_1, b_2 \in \{-1,1\}$. We solve the following system:

$$\begin{cases} \frac{\mathrm{d}x_1}{\mathrm{d}t} &= \frac{b_1 b_2}{x_1(t)-x_2(t)} \\ \frac{\mathrm{d}x_2}{\mathrm{d}t} &= \frac{b_1 b_2}{x_2(t)-x_1(t)} \\ x_1(0) = x_1^0,\ x_2(0) = x_2^0 \end{cases} \tag{2.1}$$

We do so by re-parametrising; the only quantity of importance is the difference between x_1 and x_2, so we define $r(t) := x_2(t) - x_1(t)$ and set $r_0 := x_1^0 - x_2^0$. We obtain the following system:

$$\begin{cases} \dfrac{\mathrm{d}r}{\mathrm{d}t} = \dfrac{2b_1 b_2}{r(t)} \\ r(0) = r_0 \end{cases} \tag{2.2}$$

This separable ODE may be solved as follows:

$$\begin{aligned} & \frac{\mathrm{d}r}{\mathrm{d}t} = \frac{2b_1 b_2}{r(t)} \\ \implies \quad & \int_0^t r(\tau)\mathrm{d}r(\tau) = 2\int_0^t b_1 b_2 \mathrm{d}\tau \\ \implies \quad & \frac{1}{2}r(t)^2 = 2b_1 b_2 \cdot t + c,\ c \in \mathbb{R} \\ \implies \quad & r(t) = \sqrt{2c + 4b_1 b_2 \cdot t} \end{aligned}$$

Taking into account the initial condition $r(0) = r_0 > 0$ (as $x_2 > x_1$), it follows that $c = r_0^2/2$. We now discern two cases:

$b_1 = b_2$: The derived solution becomes $r(t) = \sqrt{r_0^2 - 4t}$. The two dislocations attract each other, and r is decreasing in time. In particular, with the given initial distance, we find that $t = r_0^2/4$ is the first zero of r, implying that the dislocations collide after $r_0^2/4$ time. The ODE in 2.2 is no longer well-defined for $r = 0$, so this solution holds on $[0, r_0^2/4)$ (but may be extended continuously to include the right endpoint).

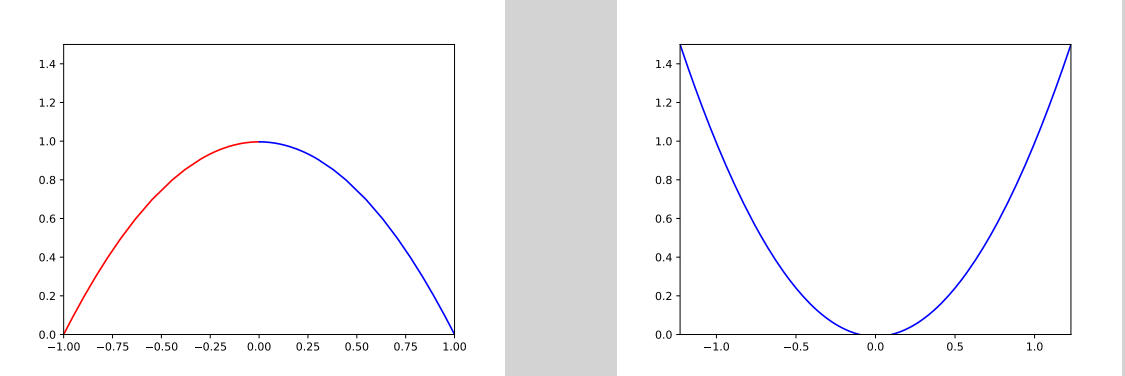
$b_1 = -b_2$: We now have $r(t) = \sqrt{r_0^2 + 4t}$, so x_1 and x_2 repel each other and diverge to $-\infty$ and $+\infty$ respectively.

Note that $\frac{dx_1}{dt} + \frac{dx_2}{dt} = 0$, so $x_1 + x_2$ is constant. Thus the dislocation trajectories are spatially symmetric, and the following formulas can be used to recover x_1 and x_2:

$$x_1(t) = \frac{x_1^0 + x_2^0}{2} - \frac{1}{2}r(t), \quad x_2(t) = \frac{x_1^0 + x_2^0}{2} + \frac{1}{2}r(t) \tag{2.3}$$

Intuitively, this follows immediately from (3.1); each action on one dislocation has an equally large but opposite effect on the other (Newton's third law); since there are only two dislocations, their movement is equal but opposite, and the sum (and average) of their positions remains constant.

The two possibilities analysed above are visualised in Figure 2.1, plotting the individual trajectories x_1 and x_2.



(a) $\mathbf{x}^0 = (-1, 1),\ \mathbf{b}^0 = (1, -1)$ **(b)** $\mathbf{x}^0 = (-0.1, 0.1),\ \mathbf{b}^0 = (-1, -1)$

Figure 2.1: Solution trajectories to 2.1 (or (P_2)) for differing initial locations and charges.

2.2 Creation Procedure

Before we go further into the definition and properties of creation, we need some notion of what a creation event is and what requirements it should meet. We base our principles on the theory in the work of Cleveringa et al., e.g. in [1][5] Here a pair of dislocations of opposite charge (a dipole) is created at a pre-defined source location if the force experienced in that source point exceeds some critical force magnitude during a certain period of time. The orientation of the dipole then follows from the direction (or sign, in 1D) of the force.

The relevant force here is the *Peach-Koehler force*; similar to the right-hand side of the ODE in (P_n), it represents the total force experienced in a certain position at a certain time; this however can be evaluated in *any* point in space, not only the position of a dislocation. We may define it as follows:

Definition 2.2.1 (Peach-Koehler force)**.** *Let* $(\mathbf{x}, \mathbf{b}) : [0, T] \to \mathcal{Z}_n$ *represent the solution to* (P_n) *for some given initial condition. We then define* $\mathrm{PK} : \mathbb{R} \times [0, T] \to \mathbb{R}$ *as follows:*

$$\text{For any } t \in [0, T], \text{ and all } s \in \mathbb{R} \text{ s.t } \forall i : s \neq x_i(t),\ \mathrm{PK}(s, t) := \sum_{i=1}^{n} \frac{b_i(t)}{s - x_i(t)}$$

[5] Although the authors study the two-dimensional case in this paper, the same principles still apply in our setting.

The dependence on time in the right-hand side may not always be made explicit, especially when we only consider a separate state of the system instead of the evolution over time.

With this definition we can reformulate the creation rule from [1] more precisely:

> If at a source with position s the magnitude of the Peach-Koehler force exceeds a given threshold F_{nuc}, i.e. $|\operatorname{PK}(s,t)| \geq F_{nuc}$ for a time-interval of length t_{nuc}, a dislocation dipole (x_i, x_{i+1}) is introduced with $b_i = \operatorname{sgn}(\operatorname{PK}(s,t))$, $b_{i+1} = -b_i$.

We add one additional constraint to this description. Even though not explicitly mentioned in the above, we require that the Peach-Koehler force magnitude stays above the force threshold *with the same sign*; i.e., if $\operatorname{PK}(s,\cdot)$ jumps between F_{nuc} and $-F_{nuc}$, this should not be considered staying above the threshold. Such discontinuities may exist, both because PK as given in Definition 2.2.1 may have singularities and because the creation of new dislocations may cause it to change instantaneously, thus resulting in a discontinuity, depending on the creation process.

In [1], if the above condition is met, the dipole is introduced with a predetermined distance between the two dislocations. This distance is chosen such a way that the forces between the new dipole are in balance with the forces acting on the dipole from the rest of the system, and are based on experimentally found material properties. However, this is just one of many possible ways to introduce dislocations. We deviate from Cleveringa et al. by not fixing a predetermined distance, and additionally study two different creation procedures. This analysis is postponed to Section 3 and further.

2.3 Formal definition of (PC_n)

We now formally define what a solution to (PC_n) is, using a similar construction as for solutions to (P_n) in Definition 2.1.1. However, the precise description of a solution with a creation procedure requires some notion of a solution to add that procedure to. Hence, we first introduce a solution concept *without* the actual description of how a creation event takes place. For now we consider this to be a fixed (but unknown) procedure, define what a solution would be given such a procedure, and later discuss the details to fit inside this framework.

Before we can introduce such a definition, we need to modify the state space and requirements on the b_i, and introduce a new concept involved in the creation process. Firstly, we alter our state-space to not require ordering of charged dislocations:[6] This leads to the definition of a new state space:

$$\mathcal{Z}_n^c := \left\{(\mathbf{x},\mathbf{b}) \in \mathbb{R}^n \times \{-1,0,1\}^n \text{ s.t } \text{ if } i \neq j \text{ and } b_i b_j \neq 0, \text{ then } x_i \neq x_j\right\}.$$

Moreover, a fundamental difference to (P_n) is that now, dislocations can change charge (or 'b_i can jump') more than once: a dislocation that was created can also annihilate again. To characterise this in terms of the dislocation's charge, we define the following class of functions:

Definition 2.3.1 (Charge function)**.** *Let $b : [0,\infty) \to \{-1,0,1\}$ satisfy the following conditions:*

(i) If $|b(0)| = 1$, then there exists at most one $t^* > 0$ such that $\lim_{t\uparrow t^*} b(t) \neq \lim_{t\downarrow t^*} b(t)$ ("b jumps at most once"). If such t^* exists, then b is right-continuous in t^* and $b(t^*)$ equals 0, i.e., $\lim_{t\downarrow t^*} b(t) = b(t^*) = 0$;

[6] Creation of new dislocations does not make it impossible to retain ordering, but the mild convenience this might bring does not weigh up to the effort it takes. Moreover, requiring ordering of charged dislocations is not feasible in numerical simulations, when the future evolution of the system is usually unknown.

(ii) If $|b(0)| = 0$, then there exist at most two distinct $t^* > 0$ such that $\lim_{t\uparrow t^*} b(t) \neq \lim_{t\downarrow t^*} b(t)$. At the first discontinuity, if it exists, b must be left-continuous If a second discontinuity exists, then this satisfies the same criteria as in point (i).

We then call b a charge function*, and denote the collection of all such functions by $\mathcal{B}$*

This enables us to formulate a solution concept for (PC_n), closely following Definition 2.1.1:

Definition 2.3.2 (Solution to (PC_n))**.** *Let $T > 0$, $n \geq 2$ and $(\mathbf{x}^0, \mathbf{b}^0) \in \mathcal{Z}_n$. Furthermore, let $t_{nuc}, F_{nuc} > 0$, let $C \subset \mathbb{R}$ be a finite set, and set $k := 2 \cdot \lfloor T/t_{nuc} \rfloor$. Enumerating all elements $s_i \in C$, additionally define $x^0_{n+2(i\cdot k+j)+\ell} := s_i$ for $i = 0, \dots, |C| - 1$, $j = 1, ..., k$ and $\ell = 0, 1$ and $b^0_{n+m} = 0$ for $m = 1, \dots, |C| \cdot k$.*

Then, setting $N := n + |C| \cdot k$, the functions $(\mathbf{x}, \mathbf{b}) : [0, T] \to (\mathcal{Z}^c_N)$ form a solution of (PC_n) if there exist finite subsets $S, \mathcal{T} \subset (0, T]$ such that

(i) (Regularity) $\mathbf{x} \in C([0, T]) \cap C^1([0, T] \setminus S)$, and $b_1, \dots, b_N : [0, T] \to \{-1, 0, 1\}$ are charge functions ($b_i \in \mathcal{B}$);

(ii) (Initial condition) $(\mathbf{x}(0), \mathbf{b}(0)) = (\mathbf{x}^0, \mathbf{b}^0)$;

(iii) (Annihilation) If b_i jumps at $t > 0$ and $\lim_{s\uparrow t} \left|b_i(s)\right| = 1$, then $t \in S$ and we call t an 'annihilation time'. Moreover, for all $(\tau, y) \in S \times \mathbb{R}$,

$$\sum_{i\,:\,x_i(\tau)=y} \left((b_i(\tau+) - b_i(\tau-)\right) \;=\; 0$$

(iv) (Creation event) If b_i jumps at $t \in [0, T]$ and $b_i(t-) = 0$, then there is exactly one other $j \in \mathbb{N}$ (i.e. $i \neq j$) such that $x_i(t) = x_j(t)$ and $b_i(t+) = -b_j(t+)$. Then and only then, we say that a creation event occurs *at $(x_i(t), t)$*

(v) (Creation moment) Creation occurs at (s, t), where $t \in [0, T]$, if and only if $s \in C$ and the following conditions both hold:

- *for all $\tau \in [t - t_{nuc}, t]$ it holds that $\mathrm{PK}(s, \tau) > F_{nuc}$, or for all $\tau \in [t - t_{nuc}, t]$ it holds that $\mathrm{PK}(s, \tau) < -F_{nuc}$, and in both cases, no creation event occurred at s in $(t - t_{nuc}, t)$;*[7] *and*
- *there is no $t^* \in (t - t_{nuc}, t)$ for which the previous condition holds.*

Moreover, we then have $t \in \mathcal{T}_s$, and the union of $\mathcal{T}_s$ over all sources $s \in C$ is contained in $\mathcal{T}$

(vi) (Creation procedure) [To be added depending on precise description]

(vii) (ODE for $\mathbf{x}$) On $(0, T)\setminus(S \cup \mathcal{T})$, $\mathbf{x}$ satisfies the ODE in (PC_n).

Apart from the omission of point (vi), some choices in Definition 2.3.2 are non-trivial and require motivation. We argue these point by point below.

In condition (iii), right-continuity of b and the fact that $b_i(t) = 0$ both now follow from b being a charge function, contrary to the explicit requirements posed in Definition 2.1.1. Condition (iv) can be seen as the equivalent of annihilation condition (iii), only then for creation. A difference to annihilation is that at a creation event, only one dipole is introduced, whereas annihilation could also happen with

[7] Note that this requirement is automatically not satisfied if $t - t_{nuc} < 0$, because $\mathrm{PK}(s, \tau)$ is not defined for $\tau < 0$.

more than two dislocations. Hence, the requirement that the net charge of all dislocations is 0 for annihilation is replaced by the condition that exactly two dislocations with opposite charges are created. Furthermore, we need left continuity instead of right, as was the case for annihilation, because $|b(\cdot)|$ being lower semi continuous is a requirement for b being a charge function.

The creation condition (v) encodes the requirement formulated in Section 2.2. Intuitively, it requires that at a given source, a creation occurs at time t if and only if it is the first time since a previous creation (or since $t = 0$) at which the force threshold is met during a long enough time-interval. In fact, the two points can be combined into a single condition; if $\mathrm{PK} > 0$, to ensure t meets the given requirements we may require the following:

$$t - t_{nuc} \;=\; \sup\{\tau \leq t \;:\; \mathrm{PK}(s,\tau) < F_{nuc} \text{ and no creation occurred at } s \text{ in } (t - t_{nuc}, t)\}.$$

This ensures that $t - t_{nuc}$ is the last time at which either the threshold was *not* met or at which creation occurred at s, implying that after $t - t_{nuc}$, $|\,\mathrm{PK}\,|$ was above the threshold and no creation occurred. A similar expression is still needed for $\mathrm{PK} < 0$. For readability, we prefer the formulation as given in Definition 2.3.2.

Finally, this requirement in part explains the choice for defining dislocations with zero charge at $t = 0$. If $|\,\mathrm{PK}(s,t)| > F_{nuc}$ for all $t \in [0,T]$, creations occur at regular intervals, namely at times $n \cdot t_{nuc}$ for $n = 1, ..., \lfloor T/t_{nuc} \rfloor$; more creations cannot occur at a given source within the specified time. This provides us with a bound on the total number of creation events that can occur in the time-interval $[0,T]$ at any source; taking the minimal choice for $\mathcal{T}_s$, i.e. only the times at which creation occurs at s, we have $|\mathcal{T}_s| \leq \lfloor T/t_{nuc} \rfloor$. Consequently, given a finite set C of sources, the total number of creation events in a given system during the period of time $[0,T]$ is bounded from above by $|C| \cdot \lfloor T/t_{nuc} \rfloor$.

To account for all dislocation pairs that may be created, at each source $s \in C$, we define $k := 2 \cdot \lfloor T/t_{nuc} \rfloor$ dislocations, each with location s and initial charge 0. This is what the (somewhat cumbersome) definition of initial data (x_i, b_i) for $i = n+1, \dots n+2 \cdot |C| \cdot k$ encodes. Then at each creation event, a dipole can be introduced by setting the charge of one dislocation to -1 and another to 1. This way, often far too many dislocations are initialised, since the threshold may only be met in a fraction of cases. In a theoretical setting where the full solution is known it would also be possible to only introduce exactly as many dislocations as will eventually be created, but in practice this is usually infeasible. Another possibility is to define an alternative state-space from the union over state-spaces of varying dimension

$$\bigcup_{k=1}^{N} \left\{(\mathbf{x}, \mathbf{b}) \in \mathbb{R}^k \times \{-1, 0, 1\}^k \;:\; \forall i, j = 1, \dots, k, x_i \neq x_j\right\}.$$

This would also allow for the removal of dislocations from the system, instead of setting their charge to 0. We choose not to pursue this approach, and keep the number of dislocations fixed.

3 Creation Procedures

The main achievement in this section is defining the creation of a dipole from a single source such that it overcomes the singular interaction and does not immediately annihilate again. In the following subsections, we consider two approaches: either temporarily adjusting the interaction between the newly created dislocations as in Section 3.1, or introducing the dipole at some non-zero distance apart as in Section 3.2. Both approaches require slight modifications to the solution concept given in Definition 2.3.2. Only important differences and motivation for certain choices are given here, the reader may find the full definitions in Appendix A.2.3.

One conformity between the approaches is the principles we apply to choose parameters of the creation process: these should be chosen in such a way that forces within the created dipole are balanced by the force exerted by the remainder of dislocations in the system, as Cleveringa et al. also mention in [1]. We stress here that the force between dislocations should only be adjusted within the created dipole; interactions with any other dislocation, whether already present in the system or created at the same time but from a different source, should not be altered.

Our approach for choosing suitable creation process parameters is to first consider a system with a single creation and a constant external force, replacing the influence of any other dislocations. A value for the creation parameter leading to an equilibrium in such a simplified setting also seems a reasonable choice for systems involving more dislocations: this way, it is both possible that the dipole re-collides or stays apart, depending on the other dislocations (i.e. the creation procedure does not have a clear tendency toward either option). We therefore choose to make parameters creation-specific; if a creation event is to occur, the corresponding parameter is chosen according to the force at the source at that moment.

It may seem wrong to simply adjust a system of equations in order to obtain desired behaviour; however, creation events do happen in reality, which indicates the necessity to change any system not accounting for creation. We argue that it is better to alter a system such as (P_n) in seemingly arbitrary and non-physical ways than to leave out an observed effect completely.

3.1 γ-creation

We consider an adapted version of (PC_n), where we temporarily alter the interactions between the newly created dislocations within a dipole. All other interactions should be as normally described by (PC_n). To incorporate this, we introduce an additional time-dependent multiplicative factor $\gamma_{i,j} : [0,\infty) \to \mathbb{R}$ for each pair of dislocations, adjusting the interaction force within a newly created dipole:

$$\begin{cases} \dfrac{\mathrm{d}x_i}{\mathrm{d}t} = \displaystyle\sum_{j\neq i} \frac{b_i b_j \gamma_{i,j}}{x_i - x_j} \quad \text{for } t \in (0,T), i = 1, ..., n \\ \text{with annihilation upon collision} \\ \text{with } \gamma\text{-creation} \end{cases} \tag{PC_n-γ}$$

As default, one should see $\gamma_{i,j} \equiv 1$; only if i, j are created as a dipole, it is given by a different function yet to be specified for a short period of time. There are many possible choices, two of which we study below. The length of the time-interval during which the exception holds is henceforth denoted t_{exc}; outside of these time-intervals, γ should equal 1. The precise value of t_{exc} depends on the definition of γ, and is derived in the following subsections.

To derive reasonable choices for t_{exc} and to analyse creation procedures in $(PC_n$-$\gamma)$, we first revisit the example of a simplified system, serving as a framework for further analysis.

Example 3.1.1. Assume a single creation event to occur at (s, t) for $t = 0$, and replace the influence of any other dislocations by a constant external force F. The dislocations should be in the same position at $t = 0$, which causes a singularity in their interaction. Hence, instead of requiring $x_1(0) = x_2(0) = 0$, take limits $t \downarrow 0$ for the initial condition. This leads to the following system of ODEs:

$$\begin{cases} \dfrac{\mathrm{d}x_1}{\mathrm{d}t} = \dfrac{-1}{x_1(t) - x_2(t)} \cdot \gamma(t) + b_1 F \\ \dfrac{\mathrm{d}x_2}{\mathrm{d}t} = \dfrac{-1}{x_2(t) - x_1(t)} \cdot \gamma(t) + b_2 F \\ \lim\limits_{t \downarrow 0} x_1(t) = \lim\limits_{t \downarrow 0} x_2(t) = s \end{cases} \quad \text{for } t \in (0, t_{exc}) \tag{3.1}$$

Similar to Example 2.1.2, we define $r(t) := x_1(t) - x_2(t)$ and rewrite the ODE as (3.2) below. Depending on the sign of b_1 and b_2, we obtain an additional term of either $+2F$ or $-2F$. For now we choose the former, with $b_1 = -1, b_2 = 1$; we will later see that this is without loss of generality. This yields the following ODE:

$$\begin{cases} \dfrac{\mathrm{d}r}{\mathrm{d}t} = \dfrac{-2}{r(t)} \cdot \gamma(t) + 2F & \text{for } t \in (0, t_{exc}) \\ \lim_{t \downarrow 0} r(t) = 0 \end{cases} \tag{3.2}$$

Note that again $x_1 + x_2$ is constant, since $b_1 = -b_2$ so the force terms cancel out. This implies that the dislocation trajectories are again spatially symmetric, and (2.3) can still be used to recover x_1 and x_2.

The goal is to choose t_{exc} in such a way that creation of a dipole with γ as force exception function leads to an equilibrium solution of (PC_n-γ) once the exception time t_{exc} has passed. We have $\gamma(t) = 1$ for $t \geq t_{exc}$, so setting $\frac{\mathrm{d}r}{\mathrm{d}t} = 0$ for $t \geq t_{exc}$, we find the following equilibrium:

$$\forall t \geq t_{exc} : \ \frac{\mathrm{d}r}{\mathrm{d}t} = 0 \implies \frac{-2}{r(t)} \cdot \gamma(t) + 2F = 0 \implies r(t) = \frac{1}{F}.$$

Thus, for a given γ and F, we should choose t_{exc} such that $r(t_{exc}) = 1/F$. Unless $\gamma \equiv 0$, the ODE (3.2) has no general explicit solution for r, so we continue our analysis in different fashion depending on the given expression for γ.

3.1.1 Linear γ

It is reasonable to let multiplication of the dipole interaction by γ result in a gradual transition from some adjusted interaction to the 'true' interaction as given in (PC_n). However, as long as γ is positive, the dislocations attract each other, meaning the dislocations would never move apart. Hence, a sensible choice of γ would for example be a linear (in time) transition from some repulsive force, e.g. the opposite of the interaction described by (P_n), to the interaction as given. For this transition to take place during the exception time interval $(0, t_{exc})$, we define $\gamma : [0, \infty) \to \mathbb{R}$ as

$$\gamma(t) := \begin{cases} 2 \cdot \dfrac{t}{t_{exc}} - 1 & \text{for } t \in [0, t_{exc}) \\ 1 & \text{for } t \geq t_{exc} \end{cases} \tag{3.3}$$

Choosing a suitable t_{exc} is not trivial, since (3.1) no longer has a closed-form solution for this γ.

Although we do not have an explicit expression, we do have an existence and uniqueness result for t_{exc} given an external force:

Lemma 3.1.2. *There exists a unique $t_{exc} > 0$ such that the creation of a dipole with linear γ, given by* (3.3)*, as adaptive force yields an equilibrium solution to* (3.1)*.*

We postpone the (rather elaborate) proof to Appendix A.2.2, and here only describe the resulting creation process. Choosing the t_{exc} described by Lemma 3.1.2 even for a system in which there are other dislocations, we define the creation procedure for linear γ:

Definition 3.1.3 (Linear γ-creation)**.** *If $t^* \in \mathcal{T}_s$ for some $s \in C$, then $\lim_{t \downarrow t^*} b_i = \operatorname{sgn}(\mathrm{PK}(s, t^*))$ and $\lim_{t \downarrow t^*} b_{i+1} = -\operatorname{sgn}(\mathrm{PK}(s, t^*))$ for index $i = n + 2(m \cdot k + j)$ corresponding to the j^{th} creation at $s_m = s$.*[1] *Furthermore, $\gamma_{i,i+1} = \gamma_{i+1,i} = 2t/t_{exc} - 1$ for $t \in (t^*, t^* + t_{exc}]$, where t_{exc} is the unique $t > 0$ such that in* (3.1)*, for $F = \mathrm{PK}(s, t^*)$, it holds that $x_1'(t_{exc}) = x_2'(t_{exc}) = 0$. In all other cases, $\gamma_{j,k}(t) = 1$.*

The requirements on b_i, b_{i+1}, defining the charge the dislocations are assigned, are taken from the description in Section 2.2. Formally it does not matter whether $b_i(t^*+) = 1$ or -1 as long as $b_i(t^*+) = -b_{i+1}(t^*+)$; the dislocations are created in the exact same position, so there is no ordering. However, for consistency and with numerical applications in mind, we choose to adhere to the sense of orientation.

Note that the requirement that $\gamma_{j,k}(t) = 1$ means that the interaction is only adapted within the newly created pair of dislocations; in particular, if another creation event takes place within the force exception time interval $(t^*, t^* + t_{exc}]$, the interaction from dipole to dipole is as usual and only the interaction within the dipoles is temporarily adapted. Moreover, note that with this definition, we have $\lim_{t \downarrow t^*} x_i(t) = \lim_{t \downarrow t^*} x_j(t) = s$ if dipole i, j is created at time t^* from source s.

3.1.2 Zero-γ

Another possibility is setting $\gamma(t) = 0$ for $t \in [0, \varepsilon)$ for some $\varepsilon > 0$. The external force then moves x_1 and x_2 apart. A benefit of this approach is that the interaction in (3.1) is no longer singular, which means we can truly start with the initial condition $x_1(0) = x_2(0) = 0$. For $t \geq \varepsilon$, we could either let γ transition gradually to 1 as above, or we could set $t_{exc} = \varepsilon$ and use the usual interaction for $t \geq t_{exc}$. For now, we analyse the latter, as the former has similarities to the analysis for linear γ and introduces another degree of freedom. Thus, for $t_{exc} > 0$ we define γ as follows:

$$\gamma : [0, \infty) \to \mathbb{R}, \quad \gamma(t) := \mathbb{1}_{\{t \geq t_{exc}\}}.$$

With this exception in mind, we define the creation process where interactions within a created dipole are temporarily set to 0:

Definition 3.1.4 (Zero γ-creation)**.** *If $t^* \in \mathcal{T}_s$ for some $s \in C$, then $\lim_{t \downarrow t^*} b_i = \operatorname{sgn}(\mathrm{PK}(s, t^*))$ and $\lim_{t \downarrow t^*} b_{i+1} = -\operatorname{sgn}(\mathrm{PK}(s, t^*))$ for index $i = n + 2(m \cdot k + j)$ corresponding to the j^{th} creation at $s_m = s$. Furthermore, $\gamma_{i,i+1} = \gamma_{i+1,i} = 0$ for $t \in (t^*, t^* + t_{exc}]$, where $t_{exc} = 1/(2\,\mathrm{PK}(s, t^*))^2$. In all other cases, $\gamma_{j,k}(t) = 1$.*

The time-derivatives of the dislocation trajectories x_i, x_{i+1} are discontinuous at t_{exc} because $\gamma_{i,i+1}$ jumps there, and thus the trajectories are not smooth. Hence, in the full solution definition for this creation process, next to creation and annihilation times we also exclude the end of every exception

[1] Recall that $k := 2 \cdot \lfloor T/t_{nuc} \rfloor$, where $t_{nuc} > 0$ is a fixed system constant.

period in the smoothness condition (A.2.5, condition (i)) by requiring $t^* + t_{exc}$ to be in a finite set $E \subset \mathbb{R}$ for each creation event.

Another differences to linear γ-creation is the fact that we have an explicit formula for t_{exc}. This choice is motivated by the following result:

Lemma 3.1.5. *Given $F > 0$, setting $t_{exc} = 1/(2\,\mathrm{PK}(s,t^*))^2$ and $\gamma(t) = \mathbb{1}_{\{t \geq t_{exc}\}}$ for $t \geq 0$ yields an equilibrium solution to* (3.1).

Proof. Consider two dislocations of opposite charge with positions $x_1(0) = x_2(0) = 0$ at time 0, and set $r(t) := x_2(t) - x_1(t)$. Furthermore let $F > 0$, $t_{exc} = 1/(2\,\mathrm{PK}(s,t^*))^2$ and $\gamma(t) = \mathbb{1}_{\{t \geq t_{exc}\}}$. Since $\gamma(0) = 0$ we may take $r(0) = 0$ as initial condition, and from (3.2) obtain the following ODE:

$$\begin{cases} \frac{\mathrm{d}r}{\mathrm{d}t} = \frac{-2}{r(t)} \cdot \gamma(t) + 2F & \text{for } t > 0 \\ r(0) = 0. \end{cases} \tag{3.4}$$

Since $\gamma(t) = 0$ for $0 \leq t < t_{exc}$, the first term drops out and we are left with a constant right-hand side:

$$\begin{cases} \frac{\mathrm{d}r}{\mathrm{d}t} = 2F \\ r(0) = 0 \end{cases} \implies \begin{cases} r(t) = 2F \cdot t + C,\ C \in \mathbb{R} \\ r(0) = 0 \end{cases} \implies r(t) = 2F \cdot t \quad \text{for} \quad 0 \leq t < t_{exc}.$$

Now by substitution of our choice of t_{exc}, we find $r(t_{exc}) = 2F \cdot t_{exc} = 2F \cdot \frac{1}{(2F)^2} = \frac{1}{2F}$. For $t \geq t_{exc}$ we have $\gamma(t) = 1$, so

$$\frac{\mathrm{d}r}{\mathrm{d}t} = \frac{-1}{r(t)} + 2F = \frac{-1}{1/2F} + 2F = 0 \ \text{ for } \ t \geq t_{exc}.$$

Thus, choosing $t_{exc} = 1/(2F)^2$ leads to an equilibrium solution for (3.6). Recovering trajectories x_1, x_2 from r using (2.3), this also yields an equilibrium solution for (3.1), which proves the result. □

3.2 Distance creation

We now consider the creation of a dipole at some nucleation distance $L_{nuc} > 0$. A benefit of this procedure is that the creation itself has no singular interaction, and that interactive forces between dislocations do not need to be adjusted in some artificial way. This approach is not uncommon in literature, especially in numerical simulation but also in theoretical studies, as for example in [1]. Distance creation is defined as follows:

Definition 3.2.1 (Nucleation distance creation)**.** *If $t^* \in \mathcal{T}_s$ for some $s \in C$, then $\lim_{t \downarrow t^*} b_i(t) = \mathrm{sgn}(\mathrm{PK}(s,t^*))$ and $\lim_{t \downarrow t^*} b_{i+1}(t) = -\mathrm{sgn}(\mathrm{PK}(s,t^*))$ for the index $i = n + 2(m \cdot k + j)$ corresponding to the j^{th} creation at $s_m = s$. Furthermore for $L_{nuc} = 1/\mathrm{PK}(s,t^*)$, we have $x_i(t^*+) = s - L_{nuc}/2$ and $x_{i+1}(t^*+) = s + L_{nuc}/2$.*

The system (PC_n) remains unchanged, and there is only a minor change to the solution concept as given in Definition 2.3.2. We relax the regularity condition, so that dislocation trajectories are no longer required to be continuous at creation times. We do so to facilitate the creation of dislocations at any chosen distance around a source, without having to define all dislocations that are yet to be created at their specific position. Aside from these changes, the solution concept from Definition 2.3.2 is still valid. The adapted conditions may be found in Definition A.2.6.

The choice of L_{nuc} in Definition 3.2.1 is again a consequence of an equilibrium result:

Lemma 3.2.2. *Given $F > 0$, setting $L_{nuc} = 1/F$ yields an equilibrium solution to the following system of ODEs:*

$$\begin{cases} \dfrac{\mathrm{d}x_1}{\mathrm{d}t} &= \dfrac{-1}{x_1(t) - x_2(t)} + b_1 F \\ \dfrac{\mathrm{d}x_2}{\mathrm{d}t} &= \dfrac{-1}{x_2(t) - x_1(t)} + b_2 F \\ x_1(0) = -\dfrac{1}{2}L_{nuc},\ x_2(0) = \dfrac{1}{2}L_{nuc} \end{cases} \quad \text{for } t \in (0, t_{exc}) \tag{3.5}$$

Proof. Let $F > 0$ be given, and set $L_{nuc} := 1/F$. Similar to before, we can reparametrise (again assuming for now that $b_1 = -1, b_2 = 1$) as follows:

$$\begin{cases} \dfrac{\mathrm{d}r}{\mathrm{d}t} = \dfrac{-2}{r(t)} + 2F & \text{for } t \in (0, t_{exc}) \\ r(0) = L_{nuc} \end{cases} \tag{3.6}$$

Then substitution of $L_{nuc} = 1/F$ yields

$$\left.\frac{\mathrm{d}r}{\mathrm{d}t}\right|_{t=0} = \frac{-2}{L_{nuc}} + 2F = \frac{-2}{1/F} + 2F = 0.$$

Since $r'(0) = 0$, we find stationary solution $r \equiv 1/2F$, and with that stationary solutions $x_1 \equiv -\frac{1}{2}L_{nuc},\ x_2(0) \equiv \frac{1}{2}L_{nuc}$.

Again, note that the assumptions $b_1 = -1, b_2 = 1$ and $F > 0$ were both without loss of generality; if $F < 0$, then the dipole is created with charges $b_1 = 1, b_2 = -1$ and (3.6) has an equilibrium at $r = -1/2F > 0$. □

This creation method is the only one also found in literature, and is used by Cleveringa et al. in [1]. There, L_{nuc} is fixed for the entire system, and is given by $Eb/(4\pi(1-\nu^2)\tau_{nuc})$ where E, b and ν are system constants and $\tau_{nuc}b$ is the Peach-Koehler threshold (which we denote by F_{nuc}). Because we choose L_{nuc} per creation, it is possible that the Peach-Koehler force at a creation event is much larger, and hence that the nucleation distance is several orders of magnitude smaller than in [1].

Note that in Definition 3.2.1 the interaction between dislocations is not adapted, contrary to γ-creation. Also, the way in which dislocations are placed at sources in Definition 2.3.2 and A.2.6, this creation procedure results in discontinuities in the trajectories of created dislocations. Knowing the entire evolution of the system we could artificially 'move' the trajectories to the correct starting position belonging to a certain creation at a later point: if creation of some pair $i, i+1$ happens at time t^* and position x^* with distance L_{nuc} we could define $x_i = x^* - (t/t^*) \cdot L_{nuc}/2$ and $x_i = x^* + (t/t^*) \cdot L_{nuc}/2$ for $t \leq t^*$.

However, this is rather cumbersome, and one may question the value continuity of x_i if $b_i = 0$. With that in mind, we choose to initialise all dislocations with initial charge 0 at an arbitrary location (the construction from Definition 2.1.1 for example), and modify the solution definition to only require continuity of x_i if $b_i \neq 0$. Even more, dislocation positions can only change if their charge is non-zero, leading to the adapted regularity condition $\mathbf{x} \in C([0,T]\setminus\mathcal{T})$ in Definition A.2.6. In practical situations (e.g. numerical simulation) this is clearly preferable, since usually information on future states of the system is not available.

4 Analysis of creation processes

4.1 Long-range effects

We now compare the long-range influence of different creation processes. Deriving an exact representation is not always possible, so in this section we resort to multipole expansion. The idea behind multipole expansion is to view the long-range effect of a set of dislocations as if it were caused by a single object. Here, 'multipole' refers to the way this object is decomposed; similar to the approximation of a function via Taylor series, a set of charged particles (dislocations) can be approximated by adding terms for total charge, dipole moment (orientation), quadrupole moment and so on. Continuing this expansion leads to an exact representation of all dislocations and charges, but more often the series is truncated to obtain an estimate. We use this to obtain an approximation of certain influences of different creation procedures on the rest of the system, from expressions that are easier to analyse than exact representations. The main result, showing that such an approach is reasonable, is the following:

Lemma 4.1.1 (Multipole approximation of Peach-Koehler force)**.** *Let* $\mathbf{x}, \mathbf{b} \in \mathcal{Z}_N^c$ *be the state of* (PC_n) *at some time* $t \geq 0$*. Then for* $|R| > \|\mathbf{x}\|_\infty$*, the following approximation holds:*

$$\mathrm{PK}(R,t) \;=\; \sum_{j=0}^{\ell} \frac{B_j}{R^{j+1}} + O\left(\frac{\|\mathbf{x}\|_\infty^{\ell+1}}{|R|^{\ell+2}}\right) \;\; as \;\; \frac{\|\mathbf{x}\|_\infty}{R} \to 0, \tag{4.1}$$

where $B_k(t) := \sum_{i=1}^n b_i(t)x_i(t)^k$*. We denote the series expansion up to order* ℓ *without the error term by* PK_ℓ*.*

This expansion may be interpreted as follows. The first term can be considered to represent all dislocations by a *monopole*; it is simply the sum over all charges, $B_0 := \sum_{i=1}^n b_i$. The second term represents the *dipole* influence, giving a sense of orientation: $B_1 := \sum_{i=1}^n b_i x_i$. Generalising this notion, the k^{th} term is also called a 2^k-pole. We now prove Lemma 4.1.1.

Proof. Take $(\mathbf{x}, \mathbf{b}) \in \mathcal{Z}_n^c$ and $\ell \in \mathbb{N}$, and let $R \in \mathbb{R}$ such that $|R| > \|\mathbf{x}\|_\infty$. Now because $|x_i| \leq \|\mathbf{x}\|_\infty < R$, we have $\|\mathbf{x}\|_\infty / R < 1$ and thus $|x_i/R| < 1$ for all $i = 1, ..., n$. Thus, the harmonic series $\sum_{j=0}^\infty (x_i/R)^j$ converges to $1/(1 - x_i/R)$ (this can also be derived by Taylor expansion). Even more, the series is absolutely convergent, which means we may interchange the order of summation (e.g. by Fubini's theorem) in $(*)$:

$$\begin{aligned}
\mathrm{PK}(R) \;&=\; \sum_{i=1}^n \frac{b_i}{R - x_i} \;=\; \frac{1}{R}\sum_{i=1}^n \frac{b_i}{1 - \frac{x_i}{R}} \\
&=\; \frac{1}{R}\sum_{i=1}^n b_i \sum_{j=0}^\infty \left(\frac{x_i}{R}\right)^j \\
&\overset{(*)}{=}\; \frac{1}{R}\sum_{j=0}^\infty \sum_{i=1}^n b_i \left(\frac{x_i}{R}\right)^j \\
&=\; \sum_{j=0}^\infty \frac{1}{R^{j+1}} \sum_{i=1}^n b_i x_i^j \\
&=\; \sum_{j=0}^\infty \frac{B_j}{R^{j+1}}.
\end{aligned}$$

Here the final step is substitution of B_k, proving the expression for the series expansion. Next we derive the error term. Truncating the final expression after $\ell + 1$ terms, we obtain the following error:

$$\begin{aligned}
\left|\mathrm{PK}(R) - \mathrm{PK}_\ell(R)\right| &= \left|\sum_{j=0}^{\infty} \frac{B_j}{R^{j+1}} - \sum_{j=0}^{\ell} \frac{B_j}{R^{j+1}}\right| \\
&= \left|\sum_{j=\ell+1}^{\infty} \frac{B_j}{R^{j+1}}\right| \\
&\leq \sum_{j=\ell+1}^{\infty} \left|\frac{B_j}{R^{j+1}}\right| \\
&= \sum_{j=\ell+1}^{\infty} \left|\frac{\sum_{i=1}^{n} b_i x_i^j}{R^{j+1}}\right| \\
&\leq \sum_{j=\ell+1}^{\infty} \frac{\sum_{i=1}^{n} \|\mathbf{x}\|_\infty^j}{|R|^{j+1}} \qquad (|b_i| \leq 1 \text{ for all } i) \\
&= \sum_{j=\ell+1}^{\infty} n \cdot \frac{\|\mathbf{x}\|_\infty^j}{|R|^{j+1}} \\
&= n \cdot \frac{\|\mathbf{x}\|_\infty^{\ell+1}}{|R|^{\ell+2}} \cdot \sum_{j=0}^{\infty} \frac{\|\mathbf{x}\|_\infty^j}{|R|^j} \\
&= n \cdot \frac{\|\mathbf{x}\|_\infty^{\ell+1}}{|R|^{\ell+2}} \cdot \frac{1}{1 - \frac{\|\mathbf{x}\|_\infty}{R}}
\end{aligned}$$

Here we considering the number of dislocations n fixed (note that this is also valid in a system with creation)[1]. Additionally bounding $\|\mathbf{x}\|_\infty / R$ arbitrarily, e.g. taking $R > 2\|\mathbf{x}\|_\infty$, we may omit the final term $1/(1 - \|\mathbf{x}\|_\infty / R) < 2$ in O-notation. Thus, we conclude that the magnitude of the error term is indeed $O(\|\mathbf{x}\|_\infty^{\ell+1} / R^{\ell+2})$. □

Note that there are settings where the error term can become arbitrarily large; e.g. taking $x_n = 1/n^{2-\varepsilon}$ for arbitrary $\varepsilon \in (0, 1)$ and $R_n = 1/n$, then for all $n > 1$, we indeed have $0 < x_n < R_n$ and $x_n/R_n = 1/n^{1-\varepsilon} \to 0$ as $n \to \infty$. However, taking $\ell = 0$, at the same time we have $x_n^1/R_n^2 = n^2/n^{2-\varepsilon} = n^\varepsilon \to \infty$ as $n \to \infty$. In general however, we are interested in situations where $\|x\|_\infty$ is bounded away from 0. Further expanding the series, i.e. taking larger ℓ, resolves the issue as well.

We now use this expansion method to analyse the long-range effect of different creation procedures. An important observation is that B_0 is constant throughout the evolution of the system, due to the fact that both annihilation and creation (i.e. the only moments at which b_i may jump) require that the left and right limits of the sum of charges are equal. Thus, when comparing the Peach-Koehler force at different points in time, the first term of the expansion (4.1) cancels out.

We now state and prove another observation that will make further analysis significantly easier.

Lemma 4.1.2. *Let* $(\mathbf{x}, \mathbf{b})$ *be a solution to* (PC_n) *or* $(PC_n\text{-}\gamma)$ *on* $[0, T]$ *for some* $T > 0$*, and take* $R \in \mathbb{R}$*. Then for all* $t \in [0, T] \setminus \mathcal{T}$ *(*t *is no creation time), if* $|R| > \|\mathbf{x}(t)\|_\infty$*, then* $\mathrm{PK}(R, \cdot)$ *is continuous in* t*.*

[1] Only when comparing different system sizes, n should not be omitted, and the error term becomes $O(n \cdot \|\mathbf{x}\|_\infty^\ell / R^{\ell+1})$.

The idea is that from a position outside the system, i.e. at $R \in \mathbb{R}$ with $R \geq \|\mathbf{x}\|_\infty$, left and right limits of the Peach-Koehler force at a certain time are always equal, as long as dislocations are not created at that time; comparing limits, PK does not 'see' dislocations that were already in the system, regardless of whether they annihilate or continue to exist. The only disturbance can be caused by a creation event. We now prove the statement:

Proof. Let $(\mathbf{x}, \mathbf{b})$, t and R be given as in the statement of the lemma, in particular such that no creation event takes place at time t (i.e. none of the b_i jumps from 0 to ± 1). Furthermore, let I be the set of all dislocations that annihilate at time t, and let $J = \{1, \dots, n\} \setminus I$ be all other dislocations. This implies that b_i are continuous in t for $i \in J$, since no creation or annihilation occurs at time t for dislocations in J. For $i \in I$, all b_i are right-continuous because they are charge functions.

Furthermore, by definition of solutions to (PC_n), all x_i are continuous, and because $|R| > \|\mathbf{x}(t)\|_\infty$, for $i = 1, \dots, n$ it holds that $R - x_i \neq 0$. This implies that

$$\begin{aligned}
\mathrm{PK}(R, t+) - \mathrm{PK}(R, t-) &= \sum_{i=1}^{n} \frac{b_i(t+)}{R - x_i(t+)} - \sum_{i=1}^{n} \frac{b_i(t-)}{R - x_i(t-)} \\
&= \sum_{i \in J} \left[\frac{b_i(t+)}{R - x_i(t+)} - \frac{b_i(t-)}{R - x_i(t-)} \right] + \sum_{i \in I} \left[\frac{b_i(t+)}{R - x_i(t+)} - \frac{b_i(t-)}{R - x_i(t-)} \right] \\
&= 0 + \sum_{i \in I} \left[\frac{b_i(t+)}{R - x_i(t+)} - \frac{b_i(t-)}{R - x_i(t-)} \right] \quad \text{(By continuity of } x_i \text{ and } b_i \text{ for } i \in J\text{)} \\
&= \sum_{i \in I} \frac{b_i(t+) - b_i(t-)}{R - x_i(t)} \quad \text{(By continuity of } x_i\text{)} \\
&= 0 \quad \text{(By annihilation rule).}
\end{aligned}$$

Finally, by right continuity of all b_i and x_i, we have $\mathrm{PK}(R, t) = \mathrm{PK}(R, t+)$, proving continuity at t. □

A similar result holds for the time-derivative of the Peach-Koehler force, which is given by the following expression for all t at which no b_i jumps:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} \mathrm{PK}(R, t) &= \frac{\mathrm{d}}{\mathrm{d}t} \sum_{i=1}^{n} \frac{b_i(t)}{R - x_i(t)} \\
&= \sum_{i=1}^{n} \left[\frac{\frac{\mathrm{d}}{\mathrm{d}t} b_i(t)}{R - x_i(t)} + b_i(t) \frac{\mathrm{d}}{\mathrm{d}t} \frac{1}{R - x_i(t)} \right] \\
&= \sum_{i=1}^{n} \frac{b_i(t)\, x_i'(t)}{(R - x_i(t))^2}.
\end{aligned}$$

In the final term, we may substitute the right-hand side of either (PC_n) or $(PC_n\text{-}\gamma)$, depending on the chosen creation process. The following result however, on time-differentiability of the Peach-Koehler force, holds in both cases:

Lemma 4.1.3. *Consider the same setting as Lemma 4.1.2, and additionally assume no annihilation occurs at time t. In case of γ-creation, also assume that $\gamma_{i,j}$ are continuous in t for all $i, j = 1, \dots, n$. Then the time-derivative of the Peach-Koehler force, $\frac{\mathrm{d}}{\mathrm{d}\tau} \mathrm{PK}(R, \tau)$, exists at time t.*

Proof. Again let $(\mathbf{x}, \mathbf{b})$, t and R be given as in the statement of Lemma 4.1.2, recall that no creation and no annihilation event takes place at time t. Thus, all b_i are continuous in t.

Furthermore, by definition of solutions to (PC_n) and $(PC_n\text{-}\gamma)$, all x_i are continuous, and because $|R| > \|\mathbf{x}(t)\|_\infty$, for $i = 1, \dots, n$ it holds that $R - x_i \neq 0$. For solutions to (P_n), this implies that

$$\begin{aligned}
\lim_{\tau\downarrow t} \frac{\mathrm{d}}{\mathrm{d}\tau} \mathrm{PK}(R,\tau) - \lim_{\tau\uparrow t} \frac{\mathrm{d}}{\mathrm{d}\tau} \mathrm{PK}(R,\tau) &= \lim_{\tau\downarrow t} \sum_{i=1}^{n} \frac{b_i(t)\, x_i'(t)}{(R - x_i(t))^2} - \lim_{\tau\uparrow t} \sum_{i=1}^{n} \frac{b_i(t)\, x_i'(t)}{(R - x_i(t))^2} \\
&= \lim_{\tau\downarrow t} \sum_{i=1}^{n} \frac{b_i(\tau) \sum_{j\neq i} \frac{b_i(\tau) b_j(\tau)}{x_i(\tau) - x_j(\tau)}}{(R - x_i(\tau))^2} - \lim_{\tau\uparrow t} \sum_{i=1}^{n} \frac{b_i(\tau) \sum_{j\neq i} \frac{b_i(\tau) b_j(\tau)}{x_i(\tau) - x_j(\tau)}}{(R - x_i(\tau))^2} \\
&= \lim_{\tau\downarrow t} \sum_{i=1}^{n} \frac{\sum_{j\neq i} \frac{b_j(\tau)}{x_i(t) - x_j(t)}}{(R - x_i(t))^2} - \lim_{\tau\uparrow t} \sum_{i=1}^{n} \frac{\sum_{j\neq i} \frac{b_j(\tau)}{x_i(t) - x_j(t)}}{(R - x_i(t))^2} \quad (x_i \text{ continuous and } b_i^2 = 1) \\
&= 0.
\end{aligned}$$

For solutions to $(PC_n\text{-}\gamma)$, by the assumption that $\gamma_{i,j}$ are continuous at t, the same derivation holds. □

Note that the continuity requirement for γ only excludes the end of exception time periods in zero-γ creation; creation moments were already ruled out, and everywhere else, both linear and zero-γ are continuous.

At annihilation moments, the Peach-Koehler force is not continuous; even more, if some pair i, j annihilates at time t, the right limit of the time-derivative is infinite. This can be seen in the derivation above, as we would keep non-zero terms with $x_i(t) - x_j(t)$ in the denominator, while $\lim_{\tau\downarrow t} x_i(t) - x_j(t) = 0$.

Still, the above results imply that comparing limits of the Peach-Koehler force before and after creation, we may consider only the contributions from the newly created dislocations, instead of considering the entire system. When comparing the time-derivative of the Peach-Koehler force we may do the same, only with the additional assumption that no annihilation event occurs at the same time. This we will exploit in the next sections.

4.2 On γ-creation

To compare the Peach-Koehler force itself just before and just after creation, we do not need to do multipole expansion since we know that with γ-creation, all dislocation trajectories remain continuous. This gives us the following continuity result, which is slightly stronger than Lemma 4.1.2:

Lemma 4.2.1. *Let* $(\mathbf{x}, \mathbf{b})$ *be a solution to* $(PC_n\text{-}\gamma)$ *on* $[0, T]$ *for some* $T > 0$*, and* $R \in \mathbb{R}$*. Assume that the creation of a dipole* $+, -$ *occurs at* (s, t)*, and that* $|R| > \|\mathbf{x}(t)\|_\infty$*. Then* $\mathrm{PK}(R, \cdot)$ *is continuous in* t*.*

Proof. The derivation is as follows, and hinges on the fact that dipoles are created in the same location, i.e. $x_+(t) = x_-(t)$:

$$
\begin{aligned}
\lim_{\tau \downarrow t} \mathrm{PK}(R,\tau) - \mathrm{PK}(R,t) &= \lim_{\tau \downarrow t} \left[\sum_{i=1}^{n+2} \frac{b_i(\tau)}{R - x_i(\tau)} \right] - \sum_{i=1}^{n+2} \frac{b_i(t)}{R - x_i(t)} \\
&= \lim_{\tau \downarrow t} \left[\frac{b_+(\tau)}{R - x_+(\tau)} + \frac{b_-(\tau)}{R - x_-(\tau)} \right] - \frac{b_+(t)}{R - x_+(t)} - \frac{b_-(t)}{R - x_-(t)} \\
&= \frac{b_+(t) - b_+(t)}{R - x_+(t)} + \frac{b_-(t) - b_-(t)}{R - x_-(t)} \quad \text{(By continuity of } x_i\text{)} \\
&= \frac{b_+(t)}{R - x_+(t)} + \frac{b_-(t)}{R - x_-(t)} \\
&= \frac{1}{R - s} - \frac{1}{R - s} \\
&= 0.
\end{aligned}
$$

□

Next we consider the time-derivative of PK with γ-creation. We cannot take the derivative in creation time t itself, since b_+ and b_- are discontinuous there. To abbreviate notation, we again use $t-$ and $t+$ for the left and right limits respectively. Moreover, as a consequence of Lemmas 4.1.2 and 4.1.3, we only need to consider the contributions of the created dislocations. To simplify notation, we assume a single dipole is created at time $t \geq 0$. Denote the trajectories of the created dislocations by x_+ and x_-, with respective charges $b_+(t+) = 1$, $b_-(t+) = -1$, and $b_+(\tau) = b_-(\tau) = 0$ for $\tau \leq t$. Considering the first-order expansion of the Peach-Koehler force, we then obtain the following:

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\tau} \mathrm{PK}_1(R,\tau)\Big|_{t+} - \frac{\mathrm{d}}{\mathrm{d}\tau} \mathrm{PK}_1(R,\tau)\Big|_{t-} &= \frac{B_1'(t+) - B_1'(t-)}{R^2} = \frac{b_+(t+)x_+'(t+) + b_-(t+)x_-'(t+)}{R^2} \\
&= \frac{1}{R^2} \left(\sum_{j=1}^{n} \left[b_+ \cdot \frac{b_+ b_j \gamma_{+,j}}{x_+ - x_j} + b_- \cdot \frac{b_- b_j \gamma_{-,j}}{x_- - x_j} \right] + b_+ \cdot \frac{b_+ b_- \gamma_{+,-}}{x_+ - x_-} + b_- \cdot \frac{b_- b_+ \gamma_{-,+}}{x_- - x_+} \right) \Bigg|_{t+} \\
&\overset{(*)}{=} \frac{1}{R^2} \left(\sum_{j=1}^{n} \left[\frac{b_j \gamma_{+,j}}{x_+ - x_j} + \frac{b_j \gamma_{-,j}}{x_- - x_j} \right] + \frac{(b_- - b_+)\gamma_{+,-}}{x_+ - x_-} \right) \Bigg|_{t+}
\end{aligned} \tag{4.2}
$$

Here at $(*)$ we use the fact that for both of our γ-creation methods, we have $\gamma_{i,j} = \gamma_{j,i}$ for all i, j, and that $b_+ b_-^2 = b_+$ and $b_- b_+^2 = b_-$ (this also holds if $b_\pm = 0$, as both jump simultaneously). We use this approximation to estimate the influence of γ-creation processes on the time-derivative of the Peach-Koehler force at R.

4.2.1 Linear γ

To further evaluate the expression above for linear γ, we now use several properties of solutions according to Definition A.2.4: first that $x_+(t+) = x_-(t+) = s$ by our assumption that creation occurs at (s,t), second that $\gamma_{+,-}(t+) = \gamma_{-,+}(t+) = -1$, and $\gamma_{i,j}(t+) = 1$ for all other pairs i, j, and finally that all other charges and trajectories are right-continuous[2]. Substituting these in Equation (4.2), we obtain the following:

[2] All trajectories and all b_i for which no annihilation occurs at t are even continuous, but since we take the right limit we do not need this.

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\tau}\mathrm{PK}_1(R,\tau)\Big|_{t+} - \frac{\mathrm{d}}{\mathrm{d}\tau}\mathrm{PK}_1(R,\tau)\Big|_{t-} &= \ldots = \frac{1}{R^2}\left(\sum_{j=1}^{n}\left[\frac{b_j\gamma_{+,j}}{x_+ - x_j} + \frac{b_j\gamma_{-,j}}{x_- - x_j}\right] + \frac{(b_- - b_+)\gamma_{+,-}}{x_+ - x_-}\right)\Bigg|_{t+} \\
&= \frac{1}{R^2}\left(\sum_{j=1}^{n}\left[\frac{b_j(t+)}{x_+(t+) - x_j(t+)} + \frac{b_j(t+)}{x_-(t+) - x_j(t+)}\right] + \frac{2b_-(t+)\gamma_{+,-}(t+)}{x_+(t+) - x_-(t+)}\right) \\
&\overset{(*)}{=} \frac{1}{R^2}\left(\sum_{j=1}^{n}\frac{2b_j(t)}{s - x_j(t)} + \frac{4}{x_+(t+) - x_-(t+)}\right) \\
&= \frac{2}{R^2}\mathrm{PK}(s,t) + \infty = \infty.
\end{aligned}$$

Here at $(*)$ in the final term, the limits in the numerator and denominator may be evaluated separately, as the numerator evaluates to a (finite) non-zero real number whereas the denominator evaluates to 0.

The fact that this limit does not exist should be no surprise: similar to how the time-derivative of trajectories of annihilating dislocations becomes arbitrarily large before annihilation, creating dislocations with a singular interaction makes their time-derivative infinitely large.

At the end of the exception time however, i.e. at $t + t_{exc}$, $\lim_{\tau\uparrow t+t_{exc}}\gamma(\tau) = \gamma(t + t_{exc}) = 1$ by definition of linear γ. Thus, by Lemma 4.1.3, $\frac{\mathrm{d}}{\mathrm{d}\tau}\mathrm{PK}(R,\tau)$ exists and is continuous in $\tau = t$.

4.2.2 Zero-γ

We conduct a similar analysis, but now for $\gamma(\tau) = \mathbb{1}_{\{\tau\geq t+t_{exc}\}}$. Suppose creation of dipole $+,-$ occurs at (s,t). Now there are two moments of interest: not only t, but also $t + t_{exc}$, because γ jumps there. Recall from the definition of zero-γ creation (3.1.4) that we have $x_+(t+) = x_-(t+) = s$, $b_+(t+) = -b_-(t+)$ and $\gamma_{i,j}(t+) = 0$ if $i,j = +,-$ and 1 else. Again we use Equation (4.2) to estimate the time-derivative of PK at creation time:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\tau}\mathrm{PK}_1(R,\tau)\Big|_{t+} - \frac{\mathrm{d}}{\mathrm{d}\tau}\mathrm{PK}_1(R,\tau)\Big|_{t-} &= \ldots = \frac{1}{R^2}\left(\sum_{j=1}^{n}\left[\frac{b_j\gamma_{+,j}}{x_+ - x_j} + \frac{b_j\gamma_{-,j}}{x_- - x_j}\right] + \frac{(b_- - b_+)\gamma_{+,-}}{x_+ - x_-}\right)\Bigg|_{t+} \\
&= \frac{1}{R^2}\left(\sum_{j=1}^{n}\left[\frac{b_j\cdot 1}{s - x_j} + \frac{b_j\cdot 1}{s - x_j}\right] + \frac{-2\cdot 0}{x_+ - x_-}\right)\Bigg|_{t+} \\
&= \frac{1}{R^2}\left(\sum_{j=1}^{n}\frac{2b_j(t)}{s - x_j(t)}\right) \\
&= \frac{2}{R^2}\mathrm{PK}(s,t).
\end{aligned}$$

Thus, the Peach-Koehler force at R is non-smooth at creation moments, as its time-derivative has a discontinuity with magnitude of order $2\,\mathrm{PK}(s,t)/R^2$. Moreover, because the force threshold of F_{nuc} must be reached for creation to occur, we have $|\mathrm{PK}(s,t)| \geq F_{nuc}$, so the size of the discontinuity is bounded from below by F_{nuc}/R^2. However, the discontinuity is finite, in contrast to linear γ-creation.

We also compare limits at the end of the exception time following the creation, since $\gamma_{+,-}$ is discontinuous there. To abbreviate notation, we assume that creation occurred at $t = 0$ (without loss of

generality, because the system is invariant to translation of time). Furthermore, we assume that the newly created dislocations have not yet re-collided (i.e. we assume $x_+(t_{exc}) \neq x_-(t_{exc})$), since otherwise they do not influence the system anymore. Using Lemma 4.1.3 we then obtain the following:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\,\mathrm{PK}_1(R,t)\Big|_{t_{exc}+} - \frac{\mathrm{d}}{\mathrm{d}t}\,\mathrm{PK}_1(R,t)\Big|_{t_{exc}-} &= \frac{1}{R^2}\lim_{\tau\downarrow t_{exc}}\frac{(b_- - b_+)\gamma_{+,-}}{x_+ - x_-} - \frac{1}{R^2}\lim_{\tau\uparrow t_{exc}}\frac{(b_- - b_+)\gamma_{+,-}}{x_+ - x_-} \\
&= \frac{1}{R^2}\frac{-2\gamma_{+,-}(t_{exc}+)}{x_+(t_{exc}) - x_-(t_{exc})} - \frac{1}{R^2}\frac{-2\gamma_{+,-}(t_{exc}-)}{x_+(t_{exc}) - x_-(t_{exc})} \\
&= \frac{1}{R^2}\frac{-2\cdot 0}{x_+(t_{exc}) - x_-(t_{exc})} - \frac{1}{R^2}\frac{-2\cdot 1}{x_+(t_{exc}) - x_-(t_{exc})} \\
&= \frac{1}{R^2}\frac{2}{x_+(t_{exc}) - x_-(t_{exc})}.
\end{aligned}$$

This implies that the zero-γ creation procedure also gives rise to a discontinuity in the time-derivative of the Peach-Koehler force at the end of the exception time. The size of this discontinuity is inversely proportional to the distance between the created dislocations. Estimating the distance between the dislocations at time t_{exc} after their creation by the derivation after Definition 3.1.4, we would obtain $\left|x_+(t_{exc}) - x_-(t_{exc})\right| \approx 1/2\,\mathrm{PK}(s,t)$ (recall that t_{exc} is chosen such that in an otherwise empty system, $\left|x_+(t_{exc}) - x_-(t_{exc})\right| = 1/2\,\mathrm{PK}(s,t)$, so that zero-$\gamma$ creation yields an equilibrium solution). This would imply that the magnitude of the discontinuity scales approximately linearly with the Peach-Koehler force at the creation moment, namely as $4\,\mathrm{PK}(s,t)/R^2$. However, this is only a rough estimate and should not be taken too seriously without further support.

4.3 On distance creation

Suppose a dipole $+,-$ with distance L is created at time t around s. Furthermore, assume the Peach-Koehler force has positive sign[3]; then $x_+(t) = s + L/2$, $x_-(t) = s - L/2$ and $b_+(t+) = 1, b_-(t+) = -1$. The Peach-Koehler force is now discontinuous; to get an estimate, we consider the second-order multipole expansion according to 4.1.1:

$$\begin{aligned}
\lim_{\tau\downarrow t}\mathrm{PK}_2(R,\tau) - \mathrm{PK}_2(R,t) &= \lim_{\tau\downarrow t}\left[\frac{B_1(\tau) - B_1(t)}{R^2} + \frac{B_2(\tau) - B_2(t)}{R^3}\right] \\
&= \lim_{\tau\downarrow t}\left[\frac{b_+(\tau)x_+(\tau) + b_-(\tau)x_-(\tau)}{R^2} + \frac{b_+(\tau)x_+(\tau)^2 + b_-(\tau)x_-(\tau)^2}{R^3}\right] \\
&= \frac{(s+\frac{L}{2}) - (s-\frac{L}{2})}{R^2} + \frac{(s+\frac{L}{2})^2 - (s-\frac{L}{2})^2}{R^3} \\
&= \frac{L}{R^2} + \frac{2sL}{R^3} \\
&= \frac{L(1+2\frac{s}{R})}{R^2}.
\end{aligned}$$

Contrary to γ-creation, the location of the source at which creation occurs now affects the result. As we are only interested in the influence of the creation event and not the (arbitrary) choice of $s \in \mathbb{R}$, we consider a system where $s = 0$, thus removing one degree of freedom. This can always be

[3] If the sign of the Peach-Koehler force were negative, the same derivation and conclusions would hold, only with opposite sign.

achieved, by translating a given solution so that a creation event occurs at position 0. With $s = 0$, PK has a discontinuity with magnitude of order L/R^2. By Definition 3.2.1, $L = 1/\operatorname{PK}(s,t)$, and $|\operatorname{PK}(s,t)| \geq F_{nuc}$ since otherwise creation does not occur; thus, the size of the discontinuity is bounded by $1/(R^2 \cdot F_{nuc})$.

In fact, an exact expression can also be derived in a similar way without using multipole expansion; however, the resulting expression is less clear, and since discontinuities for γ-creation are only approximated, we choose to compare to the more unambiguous estimate found above. We refer the interested reader to Appendix A.2.4.

4.4 Summary

A concise summary of the results from this section is presented in Table 4.1, where we denote $\operatorname{PK}(s,t)$ by F. This is by no means precise or rigorous, and only serves to get an overview of the long-range effects on $\operatorname{PK}(R,\cdot)$ and its time-derivative. A careful conclusion would be that (PC_n-γ) with zero-γ yields the most well-behaved Peach-Koehler force, and that distance creation is not preferred from an analytic point of view.

	Linear γ	Zero-γ	Distance
$\operatorname{PK}(R,t)$	Continuous	Continuous	$1/FR^2$
$\frac{\mathrm{d}}{\mathrm{d}t}\operatorname{PK}(R,t)$ at t	∞	$2F/R^2$	non-existent
$\frac{\mathrm{d}}{\mathrm{d}t}\operatorname{PK}(R,t)$ at $t + t_{exc}$	Continuous	$4F/R^2$	(n.a.)

Table 4.1: Approximate discontinuities of $\operatorname{PK}(R,\cdot)$ and its derivative for different creation methods, where $F = \operatorname{PK}(s,t)$.

5 Properties of (PC_n)

5.1 Existence & Uniqueness

The natural question to ask next is whether the definitions of solutions to (PC_n) and $(PC_n\text{-}\gamma)$ facilitate existence and/or uniqueness of solutions. We start off by considering zero-γ creation, in part due to its similarities to the original system without creation (because γ only jumps between 0 and 1), and since the Peach-Koehler force enjoys continuity, as opposed to distance creation, and is more well-behaved than for linear γ-creation. For the full solution concept, see Definition A.2.5.

From [10, Theorem 2.4], we know that for a given initial datum, there exists a unique (up to relabelling) solution to (P_n), and hence also to (PC_n) up to the first creation time. Similarly, in between collision times, as long as the evolution is governed by the ODE from (P_n), uniqueness is ensured. Now, we extend this to $(PC_n\text{-}\gamma)$ with γ-creation. We first prove that if a solution to $(PC_n\text{-}\gamma)$ exists, it must be unique (part of the proof inspired by [10][Theorem 2.4]). To this end, denote by S_n the set of all permutations of $\{1, \dots, n\}$.

Theorem 5.1.1 (Uniqueness of solutions to $(PC_n\text{-}\gamma)$)**.** *Let $T > 0$, $n \geq 2$ and $(\mathbf{x}^0, \mathbf{b}^0) \in \mathcal{Z}_n^c$. Furthermore, let $t_{nuc}, F_{nuc} > 0$, let $C \subset \mathbb{R}$ be a finite set, and set $k := 2 \cdot \lfloor T/t_{nuc} \rfloor$. For every $s \in C$, define $2k$ dislocations with $b_i = 0$ and $x_i = s$ according to Definition A.2.5. If $(PC_n\text{-}\gamma)$ has a solution according to Definition A.2.5, then this solution is unique over S_n.*

Proof. Write $N := n + |C| \cdot k$ and let $(\mathbf{x}, \mathbf{b})$, $(\tilde{\mathbf{x}}, \tilde{\mathbf{b}}) \in \mathcal{Z}_N^c$ be two solutions to $(PC_n\text{-}\gamma)$ with the same parameters t_{nuc}, F_{nuc} and source set $C \subset \mathbb{R}$. We denote the corresponding *minimal* sets of creation times by $\mathcal{T}, \tilde{\mathcal{T}}$ and annihilation times by $S, \tilde{S}$; by minimality, these are exactly the sets of times at which a b_i jumps from 0 to ± 1 or ± 1 to 0, respectively, for $i = 1, \dots, n + k$. Furthermore, denote the Peach-Koehler force for the two solutions by $\mathrm{PK}(s,t)$ and $\widetilde{\mathrm{PK}}(s,t)$ respectively.

We additionally define an extended Peach-Koehler force, PK_0. For a given $t \in [0,T]$, set $\mathrm{PK}_0(s,t) := \mathrm{PK}(s,t)$ if (s,t) is in the domain of PK, and $\mathrm{PK}_0(s,t) = 0$ for all other s (i.e. those for which there exists an i s.t. $x_i(t) = s$). This extended function is defined on all of $\mathbb{R}$, and in fact does not change the occurrence of creation compared to PK itself: PK_0 only differs from PK if it is evaluated in $s \in \mathbb{R}$ for which there exists an i such that $x_i(t) = s$. Should a creation event occur in position $x_i(t) = s$, then one of the created dislocations would immediately annihilate with dislocation i. This is equivalent to no creation occurring at all. Moreover, since PK itself is not defined in (s,t), it also makes no difference that the creation threshold is not met by PK_0.

By standard ODE theory, the two solutions must agree up until the first creation or annihilation event. Therefore, $\min\{\tau \in \mathcal{T} \cup S\} = \min\{\tau \in \tilde{\mathcal{T}} \cup \tilde{S}\}$. More precisely, denoting this first creation/annihilation moment by t_0, for $t \in [0, t_0)$ we have $\mathbf{x}(t) = \tilde{\mathbf{x}}(t)$ and $\mathbf{b}(t) = \tilde{\mathbf{b}}(t)$. Recall that the evolutions of initial data $(\mathbf{x}^0, \mathbf{b}^0)$, $(\tilde{\mathbf{x}}^0, \tilde{\mathbf{b}}^0)$ are governed by

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = \frac{1}{n} \sum_{j \neq i} \frac{b_i b_j \gamma_{i,j}}{x_i - x_j}.$$

We consider three (disjunct and exhaustive) cases:

$t_0 \in S$ and $t_0 \notin \mathcal{T}$: No creation has occurred yet, so we have $\gamma_{i,j}|_{[0,t_0)} \equiv 1$ for all i, j. Thus, up to time t_0, we could equivalently consider (P_n). Then from [10, Theorem 2.4] we know that at t_0, the ODEs for $\mathbf{x}$ and $\tilde{\mathbf{x}}$ are identical up to relabelling, so that again by standard ODE theory, the solutions $(\mathbf{x}, \mathbf{b})$, $(\tilde{\mathbf{x}}, \tilde{\mathbf{b}})$ are equal on some time-interval $[t_0, t_0 + \varepsilon)$ not containing the next creation or annihilation event.

$t_0 \in \mathcal{T}$ and $t_0 \notin S$: We again have $\gamma_{i,j}|_{[0,t_0)} \equiv 1$ for all i, j, and since $t_0 \notin S$ (and hence $t_0 \notin \tilde{S}$), the ODE is non-singular. Again, solutions $(\mathbf{x}, \mathbf{b})$, $(\tilde{\mathbf{x}}, \tilde{\mathbf{b}})$ are equal on $[0, t_0)$, and thus

$$\mathrm{PK}(s,t) = \sum_{i=1}^{n} \frac{b_i(t)}{s - x_i(t)} = \sum_{i=1}^{n} \frac{\tilde{b}_i(t)}{s - \tilde{x}_i(t)} = \widetilde{\mathrm{PK}}(s,t) \text{ for all } t \in [0, t_0).$$

Therefore, $t_0 \in \mathcal{T} \iff t_0 \in \tilde{\mathcal{T}}$. Analogous to Lemma 4.2.1, it holds that $\mathrm{PK}_0(s,t)$ is continuous in t, the only difference being that we evaluate the Peach-Koehler force at a point inside the system, not at $R \geq \|\mathbf{x}\|_\infty$; however, the same proof remains valid. Thus, $\mathrm{PK}_0(s,t_0) = \widetilde{\mathrm{PK}_0}(s,t_0)$. It follows that dipoles are introduced in both solutions at (s,t_0), and moreover both have equal $t_{exc} = \tilde{t}_{exc} > 0$ since this is fully determined by the magnitude of the Peach-Koehler force at source location and creation time, both for zero-γ and linear γ. Thus, both ODEs are again restarted identically.

$t_0 \in \mathcal{T}$ and $t_0 \in S$: Creation must occur in $(\mathbf{x}, \mathbf{b})$ at some (s,t). It then follows by continuity of $\mathbf{x}, \tilde{\mathbf{x}}$ and standard ODE theory that creation also occurs in $(\tilde{\mathbf{x}}, \tilde{\mathbf{b}})$ at (s,t), and that $\mathrm{PK}_0(s,t) = \widetilde{\mathrm{PK}_0}(s,t)$. Again, analogous to Lemma 4.2.1, it holds that $\mathrm{PK}_0(s,t)$ is continuous in t.

At the same time, no creation has yet occurred and we still have $\gamma_{i,j}|_{[0,t_0)} \equiv 1$ for all i, j. Thus, all arguments from the first case still hold. In particular, even if annihilation occurs at the same position as creation, and defining $I' := \{i \in \mathbb{N} : x_i(t_0) = x\}$, by the creation and annihilation conditions in A.2.6, we still have that

$$\sum_{i \in I} \tilde{b}_i(t_0+) = \sum_{i \in I} \tilde{b}_i(t_0-) = \sum_{i \in I} b_i(t_0-) = \sum_{i \in I} b_i(t_0+).$$

Moreover, because $\mathrm{PK}_0(s,t) = \widetilde{\mathrm{PK}_0}(s,t)$, creation in both solutions occurs with the same creation parameter $t_{exc} = 1/(2\,\mathrm{PK}(s,t))^2 > 0$.

Therefore again, it follows that at t_0, the ODEs for $\mathbf{x}$ and $\tilde{\mathbf{x}}$ are identical up to relabelling, and the solutions $(\mathbf{x}, \mathbf{b})$, $(\tilde{\mathbf{x}}, \tilde{\mathbf{b}})$ are equal on some time-interval $[t_0, t_0 + \varepsilon)$ not containing the next creation or annihilation event.

We may repeat these arguments over all other creation and annihilation times, with the possible adaption that there may be $t \in S \cup \mathcal{T}$ such that $\gamma_{i,j}(t) = 0$ for some dislocation pair i, j that was created at $t^* < t$. This however does not change the validity of the arguments above. This way, we find $S = \tilde{S}$, $\mathcal{T} = \tilde{\mathcal{T}}$ and $(\mathbf{x}, \mathbf{b}) = (\tilde{\mathbf{x}}, \tilde{\mathbf{b}})$, all over permutations S_{n+k} of indices. This proves uniqueness of any given solution to (PC_n-γ). □

5.2 Further Properties

For (P_n), Van Meurs et al. state properties such as translation and scale invariance. Some of these still hold, but the addition of creation to the system renders others invalid; hence we also explicitly mention properties that do not hold for (PC_n).

Lemma 5.2.1 (Properties of solutions to (PC_n-γ))**.** *Suppose* $(\mathbf{x}, \mathbf{b})$ *is a solution to* (PC_n-γ) *on* $[0, T]$ *for some* $T > 0$*,* $C \subset \mathbb{R}$ *and* $F_{nuc}, t_{nuc} > 0$*. Then*

(i) for any $\lambda \in \mathbb{R}$*, the translated trajectories* $(\mathbf{x} + \lambda \cdot \mathbf{1}, \mathbf{b})$ *(where* $\mathbf{1} = (1, \dots, 1) \in \mathbb{R}^n$*) form a solution to* ($PC_n$-$\gamma$) *with source set* $C + \lambda := \{c + \lambda | c \in C\}$*;*

(ii) the scaled trajectories and charges $t \mapsto (\alpha \mathbf{x}(t/\alpha^2)), \mathbf{b}(t/\alpha^2)$ *form a solution to* (PC_n-γ) *with source set* $\alpha \cdot C := \{\alpha \cdot c | c \in C\}$*, creation force threshold* F_{nuc}/α *and time threshold* t_{nuc}/α^2 *for all* $\alpha > 0$*;*

(iii) $B_0(t)$ *is constant on* $[0, T]$*, i.e., net charge is conserved;*

(iv) the first moment of dislocation trajectories, $M_1(t) := \sum x_i(t)$*, is conserved.*[1]

Proof. Let $(\mathbf{x}, \mathbf{b})$ be a solution to (PC_n-γ) on $[0, T]$ for some $T > 0$, with given set of sources C, and $F_{nuc}, t_{nuc} > 0$. For S and $\mathcal{T}$ from Definition A.2.5, take the minimal sets containing only annihilation and creation times respectively.

(i) Translation invariance on $[0, T] \setminus (S \cup \mathcal{T})$ follows from the definition of the ODE in (PC_n-γ): for any $\lambda \in \mathbb{R}$ and any i, we have

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = \frac{1}{n} \sum_{j \neq i} \frac{b_i b_j \gamma_{i,j}}{x_i - x_j} \frac{1}{n} = \sum_{j \neq i} \frac{b_i b_j \gamma_{i,j}}{x_i + \lambda - x_j - \lambda},$$

so replacing x_i by $x_i + \lambda$ does not change the evolution between creation and annihilation moments. Moreover, if i, j annihilate at (x, t), their translated counterparts must annihilate at $(x + \lambda, t)$, so S remains unchanged. Now assume a creation event occurs in $(\mathbf{x}, \mathbf{b})$ at (s, t^*) for some $s \in C$. This implies that $s + \lambda \in C + \lambda$. Denote the Peach-Koehler force for solution $(\mathbf{x}, \mathbf{b})$ by PK, and the Peach-Koehler force for $(\mathbf{x} + \lambda \cdot \mathbf{1}, \mathbf{b})$ by PK_λ. We consider PK_λ at an arbitrary time and position:

$$\mathrm{PK}_\lambda(y + \lambda, t) = \sum_{i=1}^{n} \frac{b_i(t)}{y + \lambda - (x_i(t) + \lambda)} = \sum_{i=1}^{n} \frac{b_i(t)}{y - x_i(t)} = \mathrm{PK}(y, t).$$

Thus, the creation conditions from A.2.5 are met in the translated solution at $(s + \lambda, t)$ if and only if creation occurs in the original solution at (s, t).

(ii) The fact that $t \mapsto (\alpha \mathbf{x}(t/\alpha^2), \mathbf{b}(t/\alpha^2)$ is a solution to the ODE in (PC_n-γ) follows from standard ODE scaling theory. By the same theory, setting $S_\alpha := \{t/\alpha^2 | t \in S\}$ gives the set of all annihilation times.

We now show that creation events in both solutions also correspond. Assume a creation event occurs in $(\mathbf{x}, \mathbf{b})$ at (s, t^*) for some $s \in C$; then $\alpha \cdot s \in \alpha \cdot C$. Denote the Peach-Koehler force for solution $(\mathbf{x}, \mathbf{b})$ by PK, and the Peach-Koehler force for $(\alpha \mathbf{x}(t/\alpha^2), \mathbf{b}(t/\alpha^2))$ by PK_α. We consider PK_α at an arbitrary time and position:

$$\mathrm{PK}_\alpha(\alpha \cdot y, t/\alpha^2) = \sum_{i=1}^{n} \frac{b_i(t/\alpha^2)}{\alpha \cdot y - \alpha \cdot x_i(t/\alpha^2)} = \frac{1}{\alpha} \sum_{i=1}^{n} \frac{b_i(t/\alpha^2)}{y - x_i(t/\alpha^2)} = \frac{1}{\alpha} \mathrm{PK}(y, t/\alpha^2).$$

Thus, $|\mathrm{PK}_\alpha(\alpha \cdot s, t/\alpha^2)| \geq F_{nuc}/\alpha$ if and only if $|\mathrm{PK}(s, t/\alpha^2)| \geq F_{nuc}$.

Furthermore, writing $\tau = t/\alpha^2$, we have $|\mathrm{PK}_\alpha(\alpha \cdot s, \tau)| \geq F_{nuc}/\alpha$ during time-interval $\tau \in [t^*/\alpha^2 - t_{nuc}/\alpha^2, t^*/\alpha^2]$ if and only if $|\mathrm{PK}(s, t/\alpha^2)| \geq F_{nuc}$ for $t/\alpha^2 \in [t^*/\alpha^2 - t_{nuc}/\alpha^2, t^*/\alpha^2]$, i.e. for $t \in [t^* - t_{nuc}, t^*]$.

Therefore, condition (v) with force and time threshold F_{nuc}/α and t_{nuc}/α^2 is met by the scaled solution at $(a \cdot s, t^*/\alpha^2)$ if and only if creation occurs in $(\mathbf{x}, \mathbf{b})$ at (s, t^*), which proves the statement.

[1] Note that this includes dislocations with zero charge. The first moment of charged dislocations is not conserved, as creation or annihilation of two dislocations at position x results in a jump in M_1 of size $2x$.

(iii) Conditions (iii) and (iv) in Definition A.2.5 ensure that at all times at which b_i can jump, net charge is conserved (as also remarked in Section 4.1); hence, B_0 must be constant on $[0, t]$.

(iv) Conservation of the first moment of dislocation trajectories follows from the symmetry in the ODE in (PC_n-γ); for any i, j, we have $b_i b_j \gamma_{i,j} = b_j b_i \gamma_{j,i}$, and $x_i - x_j = -(x_j - x_i)$. Thus, defining $c_{ij}(t) := (b_i(t) \cdot b_j(t) \cdot \gamma_{i,j}(t))/(x_i(t) - x_j(t))$, we have $c_{ij}(t) = -c_{ji}(t)$ for all $t \in [0, T]$. Now taking the time-derivative of M_1, by linearity of differentiation we obtain

$$M_1'(t) = \sum_{i=1}^{n} \frac{\mathrm{d}x_i}{\mathrm{d}t} = \sum_{i=1}^{n} \sum_{j \neq i} c_{ij} = \sum_{i,j\,:\,i<j} c_{ij} + \sum_{i,j\,:\,i>j} c_{ij} = \sum_{i,j\,:\,i<j} \left[c_{ij} + c_{ji}\right] = 0.$$

Note that this implies conservation of the first moment only of charged dislocations between creation and annihilation events. The conservation of the first moment of all particles over all times then follows from the fact that positions of dislocations with zero charge (i.e. before creation or after annihilation) do not change. Thus at creation and annihilation moments, the first moment over all dislocations (also with charge 0) is conserved as well. □

5.3 Influence of creation on total annihilation time

If the configuration of (P_n) is such that all dislocations annihilate in finite time, we conjecture that the creation of new dipoles can only speed up the annihilation of all dislocations (if dipoles are oriented according to the description below Definition 2.2.1). For this, we define the *total annihilation time*:

Definition 5.3.1 (Total annihilation time)**.** *Let* $(\mathbf{x}, \mathbf{b})$ *be a solution to* (PC_n)*. Then the* total annihilation time*, denoted* t_{ann} *is the smallest time at which all dislocations have charge 0:*

$$t_{ann} := \inf\{t > 0 \ : \ b_i(t) = 0 \ \textit{for } i = 1, ..., N\}.$$

If this infimum does not exist, $t_{ann} = \infty$*. If* $(\mathbf{x}, \mathbf{b})$ *is only defined on time-interval* $[0, T]$*,* $t_{ann} = \infty$ *if there are* i *such that* $b_i(T) \neq 0$*.*

We now formulate our conjecture more precisely:

Conjecture 5.3.2. *Let* $(\mathbf{x}^0, \mathbf{b}^0) \in \mathcal{Z}_n$ *be an initial configuration such that* (P_n) *has a unique solution on* $[0, T]$ *and all dislocations annihilate in finite time* $t_{ann} \leq T$*. Then we conjecture that starting* (PC_n) *with the same initial configuration of charged dislocations and a single source at an arbitrary position, all initial dislocations annihilate in some time* $\tilde{t}_{ann}$ *with* $\tilde{t}_{ann} \leq t_{ann}$

More preliminary, we first see whether a single 'artificial' creation can slow down the total annihilation time:

Conjecture 5.3.3. *Let* $(\mathbf{x}^0, \mathbf{b}^0) \in \mathcal{Z}_n$ *be an initial configuration such that* (P_n) *has a unique solution on* $[0, T]$ *and all dislocations annihilate in finite time* $t_{ann} \leq T$*. Recall that* $\mathbf{x} = (x_1, ..., x_n)$ *is an ordered* n*-dimensional vector. Now we add two dislocations,* x_α *and* x_β*, with opposite charge (a dipole) to the initial configuration in such a way that* $x_i < x_\alpha < x_\beta < x_{i+1}$ *and* $b_i = -b_\alpha = b_\beta = -b_{i+1}$ *for some* $1 < i < n$*, to obtain a new initial configuration* $(\tilde{\mathbf{x}}^0, \tilde{\mathbf{b}}^0) \in \mathcal{Z}_{n+2}$ *We then conjecture the following: following* (P_{n+2})*, all dislocations annihilate in some time* $\tilde{t}_{ann}$ *with* $\tilde{t}_{ann} \leq t_{ann}$*.*

This conjecture is not true, or at least not without further restrictions. This can be seen from the following (not completely rigorous, but convincing) counterexample:

Example 5.3.4. Consider the case $n = 4$, with $\mathbf{x}^0 = (-2, -1, 1, 2)$ and $\mathbf{b}^0 = (-, +, -, +)$. Here the left and right dislocation pairs each annihilate in approximately $(2-1)^2/4 = 1/4$ units of time. This can be seen as follows: for each of the dislocation pairs separately, we have an explicit solution, as derived in Example 2.1.2; from this we know that dislocations 1 and 2 on their own would annihilate in $(x_2 - x_1)^2/4 = 1/4$ time. Combining the two dislocation pairs in one system affects this time, but only slightly, as the distance between the pairs is significantly larger than the distance between dislocations within a pair.

Next, we consider the introduction of additional dislocations at $x = -1 + \delta$ and $x = 1 - \delta$ for some $\delta < 1/2$, with respective charges $-$ and $+$. This yields $\tilde{\mathbf{x}}^0 = (-2, -1, -1 + \delta, 1 - \delta, 1, 2)$ and $\tilde{\mathbf{b}}^0 = (-, +, -, +, -, +)$. Then by choosing δ small enough, we can make the dislocation pairs x_2, x_3 and x_4, x_5 annihilate arbitrarily quickly, leaving x_1 and x_6 with distance $r^* = 4 - \varepsilon$ apart for some ε that can be made arbitrarily small by appropriately choosing δ. There are now only two dislocations with opposite charge in the system, meaning they annihilate in $(r^*)^2/4 \approx 4$ time (similar to the reasoning above). This suggests that $t_{ann}(\mathbf{x}^0, \mathbf{b}^0) < t_{ann}(\tilde{\mathbf{x}^0}, \tilde{\mathbf{b}}^0)$, thus disproving Conjecture 5.3.3.

Numerical approximations of the solution trajectories to both cases are visualised in Figures 5.1 and 5.2 (note that in the latter, the initial configuration indeed consists of six dislocations; the annihilation already occurs at small t already).

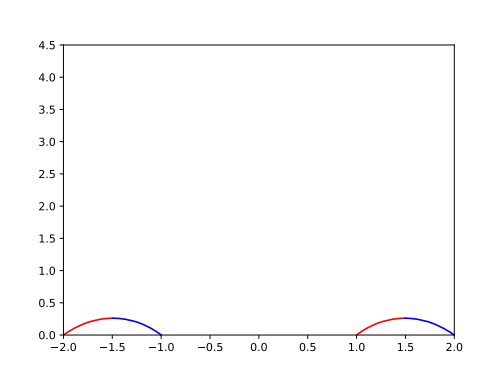


Figure 5.1: Solution trajectories for $\mathbf{x}^0 = (-2, -1, 1, 2)$ and $\mathbf{b}^0 = (-, +, -, +)$.

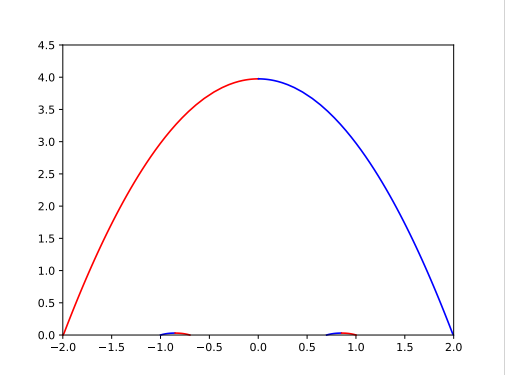


Figure 5.2: Solution trajectories for $\tilde{\mathbf{x}}^0 = (-2, -1, -1 + \delta, 1 - \delta, 1, 2)$ and $\tilde{\mathbf{b}}^0 = (-, +, -, +, -, +)$ with $\delta = 0.15$

However, this counterexample to Conjecture 5.3.3 does not necessarily mean that Conjecture 5.3.2 is not true; the given example does not depend on an actual creation procedure, and the dipole is added to the four-dislocation system with a large distance between the introduced dislocations. It may be that this situation cannot arise 'naturally', when following the creation processes described in Section 3.

6 Simulation

Numerical simulation of (PC_n) serves as a further tool to compare creation methods to each other as well as to existing research, and enables us to study systems with many dislocations, that often cannot be solved analytically. However, implementing such a singular system with annihilation and creation is far from trivial. Hence, in the following sections, we first describe the general implementation of (PC_n), and discuss choices made in the process. We then turn to the implementation of the specific creation processes, and finally analyse simulation results. Much of the work in this section is on a rather basic level, and should mostly be seen as an exploration of the limits and possibilities of simulating with dislocation and annihilation. Our main goal is not to provide exact and reliable results, but rather to provide a starting point for further development of similar simulations and numerical studies.

The code used in the following section is added in Appendix A.3. For executable scripts, we refer to `https://github.com/jmoraal/MasterThesis`. Scripts were written in Python 3.8.3.

6.1 Regularisation

One of the difficulties in simulating (PC_n) is that the right-hand side of the ODE is singular. As computers cannot handle arbitrarily large (or small) numbers, we need to make adaptions so that the system can still be simulated in a reasonable way. There are two main adaptions: introducing a collision threshold, and using a regularised interaction function, replacing the right-hand side of the ODE in (PC_n).

To avoid having to compute the ODE near the singularity in the interaction given by (PC_n) and ensuing numerical inaccuracies, when simulating annihilation we introduce a so-called *collision threshold*. This is a predefined distance, preferably small, where we consider two dislocations to have the same position if their distance is less than that threshold. This is frequently used in literature, as for example by Cleveringa et al. in [1], and its value may be based on known material properties. When considering annihilation, introducing a collision threshold is also be necessary if the dislocation interaction is not singular; two dislocations may never obtain the exact same position (possibly depending on the integration method). For a non-singular interaction however, the choice of the threshold is less critical, as a bounded interaction decreases the risk of overshooting. In any setting however, choosing a too large collision threshold may result in the occurrence of annihilation in instances where it should not take place. Choosing the collision threshold is therefore a delicate matter, and should be studied more extensively.

Another solution is to adjust (PC_n), regularising the interaction force to obtain a non-singular ODE. In [6], Van Meurs studies the following regularised system for two different bounded and continuous regularised forces, where $\delta > 0$ is a regularisation parameter

$$\begin{cases} \dfrac{\mathrm{d}x_i}{\mathrm{d}t} = b_i \displaystyle\sum_{j \neq i} b_j \cdot \phi_\delta(x_i - x_j) & \text{for } t > 0 \\ x_i(0) = x_i^0 \end{cases} \qquad \text{for } i = 1, ..., n. \tag{6.1}$$

The interaction force Van Meurs concludes to be most suitable is $\phi_\delta(r) := r/(r^2 + \delta^2)$, visualised in Figure 6.1:

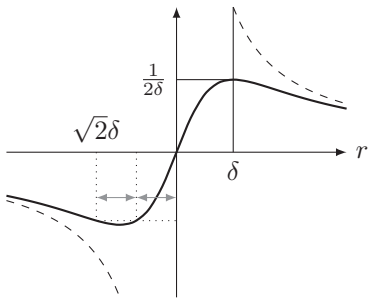


Figure 6.1: Illustration of true force $1/r$ (dashed) and regularised force $r/(r^2 + \delta^2)$ (solid). [6]

For $\delta \to 0$, this function converges pointwise to $1/r$, while it is continuous and bounded (and even differentiable). This makes it particularly suitable for theoretical analysis. However, dislocations at distance δ from each other exert a stronger force on each other than dislocations closer by, since ϕ_δ has a global maximum and minimum at $r = \delta$ and $-\delta$ respectively. Thus, fixing any $\delta > 0$, solutions to (6.1) may be very different from the non-regularised version. In particular, it is possible that a pair of dislocations would annihilate with regular interaction, but does not with regularised interaction. This

Hence we also consider a regularised interaction force that is not increasing around 0. For $\delta > 0$, we define $\varphi_\delta(r) := \operatorname{sgn}(r) \max\{|1/r|, 1/\delta\}$ for $t \neq 0$, and $\varphi_\delta(0) = 0$.

Finally, when using a sufficiently large collision threshold and an adaptive time-step (becoming smaller when dislocations are closer), it is also possible to simulate to reasonable accuracy without regularisation.

6.2 Implementation

In general, automatic ODE-solvers are the preferred method to solve systems of differential equations. In this case however, we cannot use them directly for two reasons. Firstly, because the interaction is singular, usually causing errors or even termination of such algorithms; this can be circumvented in various ways, for example by regularisation or a collision threshold. The second difficulty, combining creation methods with automatic solvers, is more of a challenge. For this reason, and to gain better insight in the subtleties of simulating a system such as (PC_n), we develop a basic 'manual' integrator, which makes it easier to include custom features such as annihilation and creation. This is far from state-of-the-art from an ODE-solving point of view, but works reasonably well and is much clearer than the black box-like automatic solvers.

We implement our simulation in Python, combining a functional and object-oriented approach. We heavily rely on the `numpy` module for fast computations with large arrays, but use no other packages. Especially large computations are done using `numpy`, which functions as an interface to C++ and thus enables fast operations on large arrays; speed-up compared to regular implementation in python is not rarely by a factor 100, as described e.g. in [4]. However, the module is less adequate for objects with custom features, such as creation events. We define these as classes, which causes a significant overhead computation time in Python and is therefore slow. However, relative to the total amount of computation, the number of operations is small enough (compared to e.g. the distance computation for all dislocations in line 17) that the effect on runtime is not restrictive. Our simulation algorithm is described by the following pseudocode; the corresponding file is `1DDislocationsCreation.py`, and may be found in Appendix A.3.1.

Algorithm 1 Dislocation Simulation with Creation and Annihilation

1: **function** DISLOC-SIM($n, m \in \mathbb{N},\ \delta, t_{exc}, T > 0,\ \gamma : \mathbb{R}^+ \to \mathbb{R}$)
2: Initialise $\mathbf{x} \in [0,1]^n$ randomly ▷ *Dislocation positions*
3: Initialise $\mathbf{b} \in \{-1,1\}^n$ randomly s.t. $\lfloor n/2 \rfloor$ indices have $b_i = 1$ ▷ *Dislocation charges*
4: Initialise $\mathbf{s} \in [0, L]^m$ evenly spaced on $[0,1]$ ▷ *Source locations*
5: $t \leftarrow 0$
6: **while** $t < T$ **do**
7: **for** all sources $s \in S$ **do**: ▷ *Creation procedure*
8: Compute Peach-Koehler forces $PK(s,t)$
9: **if** $|PK[s]| > F_{nuc}$ **then** Increment above-threshold time counter for s by dt
10: **else**: Reset counter to 0
11:
12: **if** Counter at s reached t_{nuc} **then** ▷ *Iff PK at s was above F_{nuc} long enough*
13: Create dipole at s (with or without distance L_{nuc});
14: Assign charge $(-1,1)$ or $(1,-1)$ depending on the sign of the PK force
15: Set counter for time since creation for each new dislocation
16:
17: Compute pairwise difference D between all dislocation positions $\mathbf{x}$ ▷ *Main interactions*
18: $L \leftarrow$ all pairs i, j with $i < j$, $D_{i,j}$ smaller than collision threshold and $b_i = -b_j \neq 0$
19: **for** all pairs (i,j) in L **do**: ▷ *Annihilation*
20: **if** i and j have not annihilated yet **then**
21: Set $b_i = b_j = 0$
22: Compute (regularised) pairwise interaction between all dislocations from D, $\mathbf{b}$ and δ
23: **if** γ-creations occurred less than t_{exc} ago **then** ▷ *Force exception*
24: Multiply interactions between new dislocations by γ(*time since creation*)
25: Increment all time-since-creation counters by dt
26: **if** time-since-creation $> t_{exc}$ **then** no longer see dislocation as recently created
27: Sum all interactions to obtain *update* per dislocation
28: Pick dt such that $dt \cdot \left\| \mathrm{d}^2\mathbf{x} \right\|$ is small
29: $\mathbf{x} \leftarrow \mathbf{x} + updates \cdot dt$
30: Save $\mathbf{x}$ to *trajectories* and dt to *timesteps*
31: $t \leftarrow t + dt$
32: **return** *trajectories*, *timesteps*

Here in line 28, $\mathrm{d}^2\mathbf{x}$ denotes the component-wise second time-derivative of $\mathbf{x}$. $\|\cdot\|$ may represent any vector norm, since $\mathbf{x}(t) \in \mathbb{R}^n$ so all norms are equivalent. We use the supremum-norm, because it is most important that time-steps are small if one particular derivative is rapidly changing (as e.g. at an annihilation event).

The way annihilation is implemented in lines 19-21 may seem restrictive, but is in fact consistent with the annihilation rule in (PC_n). Although collisions with more than two dislocations are not implemented as such, iterating over all dislocation pairs with opposite charge has the same effect. In [10], Van Meurs et al. show that multiple-dislocation collisions can only occur when dislocations (ordered by position) have alternating sign. Furthermore, if the number of colliding dislocations is odd, then the net charge of all dislocations involved in the annihilation must be $+1$ or -1. Then all dislocations except one, with charge $+1$ or -1 respectively, annihilate. Because solutions are invariant under

relabelling of dislocations, it does not matter which dislocation this is.

6.2.1 Linear γ

It seems that with linear γ creation, for each $F > 0$, there exists a unique t_{exc} (and with that a specific γ) such that creating a dipole according to (3.1) leads to a stationary solution of that equation. We can deduce these from the zeros of R_{crit}: using a root-finding algorithm, in `ODEPhaseplot.py` (Appendix A.3.3) we determine the corresponding force for a range of exception times[1]. This results in the following (one-to-one) curve:

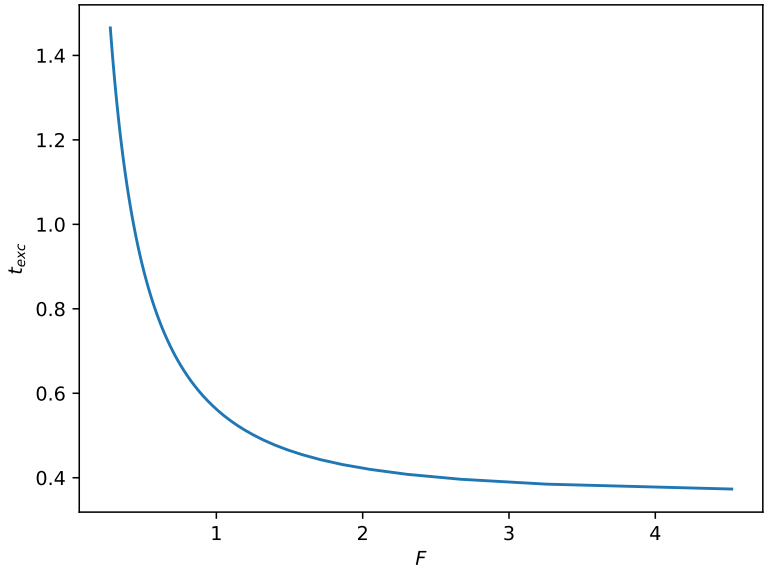


Figure 6.2: Numerical approximations of the zeros of (A.3) for varying values of t_{exc} and linear γ.

We recognise similarity to a polynomial in $1/(t_{exc} - d)$ in the data, i.e. with x and y-axis interchanged in Figure 6.2. The following third-order polynomial is fit to numerically computed data-points using a least-squares method:

$$f(x) = \frac{a}{x-d} + \frac{b}{(x-d)^2} + \frac{c}{(x-d)^3} \tag{6.2}$$

We can now use f to find an approximation of the zero of $R_{crit}(\cdot, t_{exc})$ in implementation. To determine the t_{exc} corresponding to a given force, we invert (6.2). Denoting $f^{-1}(x)$ by y, this yields the following on intervals on which f is bijective:

$$\begin{aligned} f(x) &= \frac{a}{x-d} + \frac{b}{(x-d)^2} + \frac{c}{(x-d)^3} \\ \implies \quad x &= \frac{a}{y-d} + \frac{b}{(y-d)^2} + \frac{c}{(y-d)^3} \\ \implies \quad x(y-d)^3 &= a(y-d)^2 + b(y-d) + c \end{aligned}$$

[1] Although finding the exception time to a given force would be more intuitive, this was easier from the side of implementation.

Thus, if λ is the root of the polynomial $-x\lambda^3 + a\lambda^2 + b\lambda + c$, then we have $f^{-1}(x) = \lambda + d$. Uniqueness of such a root is ensured in regions where f is bijective, which is the case here.

To obtain an explicit expression, we numerically searched roots for 100 values of t_{exc} between 0.37 and 1.5, and removed 5 outliers resulting from numerical inaccuracies. We fit the data to (6.2) using the least-squares method. This results in the fitted function (recall that the fit is for F as a function of t_{exc}, i.e. with axes in Figure 6.3 interchanged):

$$f_{\text{fit}}(t) = \frac{0.328}{t - 0.311} - \frac{0.0234}{(t - 0.311)^2} + \frac{0.00128}{(t - 0.311)^3}.$$

Both the data and f_{fit} are visualised in Figure 6.3:

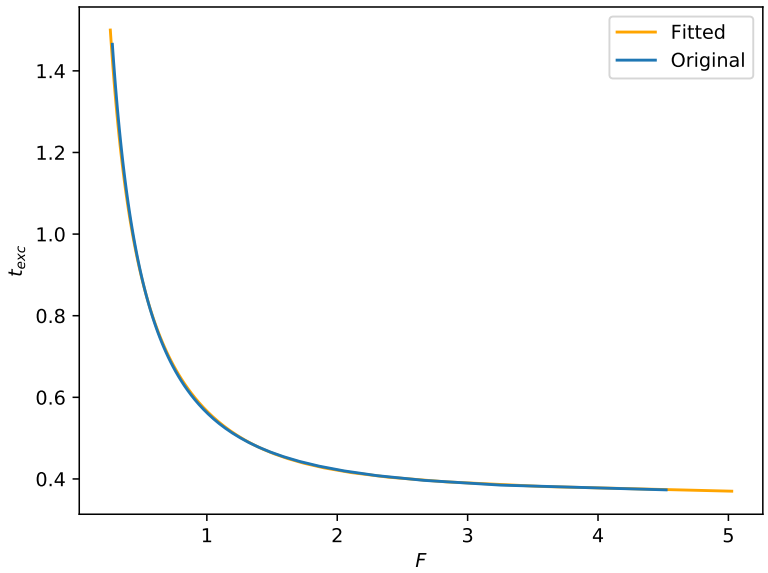


Figure 6.3: Numerical approximations of the zeros of (A.3) with linear γ for varying values of t_{exc} (blue), and fitted function (orange)

We then implement the choice of t_{exc} depending on F using Cardano's explicit formula for roots of third-degree polynomials, avoiding much slower (and possibly error-prone numerical root-finding algorithms.

Note here that there is an asymptote in t above 0; regardless the choice of F, the corresponding R_{crit} such that $R_{crit}(F, t_{exc}) = 0$ is greater than approximately 0.311. Whether this is a fundamental system constant with more meaning remains to be seen.

6.2.2 Automatic ODE-solver with annihilation

When only interested in annihilation and not creation of dislocations, then automatic ODE-solvers can in fact be used. Given an ODE such as $\mathrm{d}y/\mathrm{d}t = f(t, y)$, most well-known ODE solvers in Python[2] take in f, a time-domain and an initial value, and return the approximated value of the solution at a range of points in time. There is no access to intermediate results, so changes in parameters cannot simply be conditioned on the state of the approximated ODE solution outside the solver, as we did in

[2]We use `solve_ivp` from the package `scipy.integrate`.

Algorithm 1. However, if f calls on some global parameter, then every time f is evaluated by the solver, it can change this global parameter. This change then remains in place for the next iteration of the solver, thus effectively changing the function parameters.

In our example, this entails including an annihilation condition in the function describing the ODE, $\mathrm{d}\mathbf{x}/\mathrm{d}t$. The only things required to implement the annihilation condition are the positions and charges of the dislocations at the current timestep. Positions are available through the solver itself, and charges can be passed as a global variable. An example of such an implementation for (P_n) may be found in the script `1DautoSolver.py`, or Appendix A.3.2.

It is possible that creation can also be added in a similar way. However, this is more complicated since implementing the creation thresholds requires knowledge of more than one state of the system. Further investigating this possibility falls beyond the scope of our work.

6.3 Results

As mentioned at the beginning of this section, obtaining numerical results is not the main goal in this work. Still, there are some outcomes we would like to remark upon.

6.3.1 Survival rate

First, for all three creation methods, we estimate how many of the created pairs of dipoles eventually do not re-collide. To do so, for each of the creation procedures we run the simulation described in Algorithm 1 100 times for different initial values, each with 20 initial dislocations and 20 sources. The dislocation positions are initiated with uniformly random positions on $[0, 1]$, and random charges such that the net charge is 0. The source locations are evenly spaced on the interval $[0, 1]$. Comparing the creation and annihilation events, for every iteration we compute the fraction of created dipoles that do not re-collide again out of all dipoles that were created; we call this the survival rate. The mean, standard deviation and the resulting 95-% confidence interval for the mean are given in Table 6.1:

	Linear γ	Zero-γ	Distance
Sample mean	0.5317	0.5039	0.69
Standard deviation	0.0330	0.0132	0.11
95-% CI for mean	$[0.525, 0.538]$	$[0.501, 0.506]$	$[0.6684, 0.7116]$

Table 6.1: Statistics of the survival rate over 100 runs with 20 initial dislocations and 20 sources.

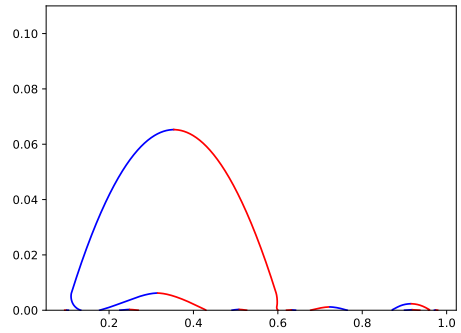
We would expect half of all dipoles to re-collide, and the other half to survive, depending on the influence of other dislocations; we therefore expect the survival rate to be 0.5. This follows from our definition of all three creation procedures in Section 3, in such a way that they form equilibrium solutions in an otherwise empty system with a constant force. From the results in Table 6.1 we see that both γ-creation methods perform close to this expectation. Zero-γ yields a survival rate slightly closer to 0.5 than linear γ; this may be a consequence of the fact that we have an explicit formula for choosing t_{exc} for the former, whereas determining the former took several steps of numerical approximation.

The deviation of the mean survival rate for distance creation may be caused by the use of a collision threshold. If the Peach-Koehler force at a creation event is large, then by the parameter choice $L_{nuc} = 1/F$ a dipole is created at a small distance. If this is smaller than the collision threshold, the dipole may immediately re-annihilate. Avoiding this may be subject to further research.

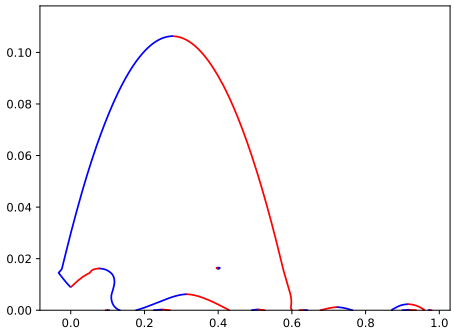
6.3.2 Examples of solution trajectories

To illustrate the effect of different creation procedures, we simulate the evolution from a specific initial value from previous section (i.e $n = 20$, $|C| = 20$) with the three different creation methods and

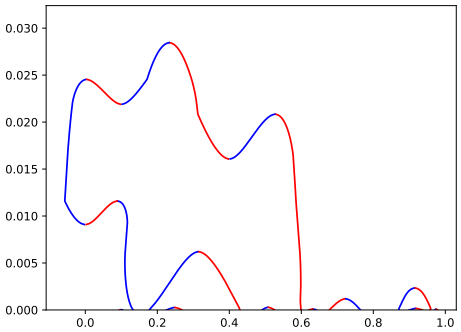
without creation, until all dislocations have annihilated. We take $F_{nuc} = 5, t_{nuc} = 0.01$ and a collision threshold of $3 \cdot 10^{-3}$.



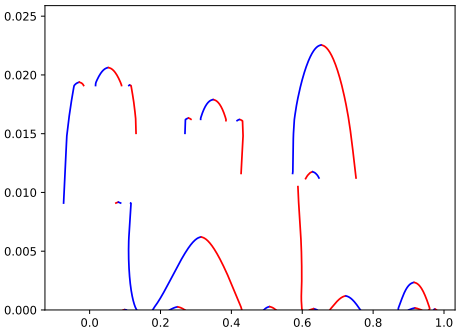
(a) No creation



(b) Zero-γ



(c) Linear γ



(d) Distance creation

Figure 6.4: Simulated trajectories in (t, x)-plane for identical initial conditions without creation and with three different creation processes (with different time-scales).

Each of the plots 6.4b-6.4d shows characteristic features of the different creation methods. In Figure 6.4b, after the creation event at position and time $(0, 0.09)$, the created dipole moves apart at a near-linear rate due to γ temporarily being set to 0. At time $t \approx 0.015$, the exception time is over and interaction returns to normal; this causes a clear discontinuity in the time-derivative, making the trajectory non-smooth. Also note that the total annihilation time is greater than without creation, shown in Figure 6.4a. This implies that, if our implementation is accurate enough, Conjecture 5.3.2 is not true for zero-γ creation. For linear γ in Figure 6.4c, the trajectories are smooth aside from annihilation times. The creation of dipoles occurs with singular derivative, causing dislocations to move apart as $\sqrt{r}$. Note that the behaviour of a dipole immediately after creation is similar to behaviour closely before annihilation, only with inverted time. Distance creation results in discontinuous trajectories. The Peach-Koehler force threshold in this simulation is set to 5, which means that discontinuities may be as large as $L_{nuc} = 1/F_{nuc} = 0.2$; this explains the distances between created dislocations.

The simulation is still inaccurate on some points; e.g. in Figure 6.4b, more creation events should occur at the source with position 0.4. The fact that these do not show may be a consequence of a too large collision threshold, in combination with the code structure where annihilation removes created dislocations before they are plotted.

7 Conclusions & Discussion

Our main goal was to extend existing theory of dislocation dynamics in a single dimension with the creation of new dipoles. Having developed solution definitions for three different ways of creating dipoles and proving uniqueness for one of these, we have made a first step towards a consistent theory including creation. We may conclude that it is possible to introduce new dipoles, although the details are subtle. In addition, we developed a framework to compare the long-range influence of different creation processes, and described methods to numerically simulate the developed systems.

Comparing the three creation procedures described in Section 3, we may conclude that the two γ-creation procedures have significant advantages over distance creation. Creation with linear γ and zero-γ both have benefits and drawbacks; the former has a singular force at creation moments, the latter introduces discontinuities in the ODE in $(PC_n\text{-}\gamma)$. On numerical implementation, zero-γ results in a creation survival rate slightly closer to the expected 50%. However, this may also be due to numerical inaccuracies in the (rather complicated) parameter choice for linear γ, which may be improved further.

Overall, much of our work covers early stages of the development of a new formal system to adequately represent dislocation dynamics with annihilation and creation. The novelty of this report is more in the construction of definitions and a theoretical framework to conduct further analysis in, than in the analysis of properties of (PC_n) itself. Hence there is a wide range of open questions and conjectures, to be studied in further research.

Within our line of work, the next main objective is to prove well-posedness of (PC_n), following uniqueness of solutions in Theorem 5.1.1. Further properties of solutions similar to those stated for (P_n) by Van Meurs et al. in [10, Theorem 2.4] may be derived for (PC_n) as well. Comparing these properties for different creation methods might give better insight into the difference of system behaviour for various choices. Also, other creation methods than the ones we treat in this report could be of interest. For example, taking a combination of linear and zero-γ such as $\gamma(t) = \mathbb{1}_{\{t\geq t_1\}} \cdot (t-t_1)/t_2$ with $t_1 + t_2 = t_{exc}$ in γ-creation may combine the benefits of linear and zero-γ. A more basic question, also relevant to numerical simulation, is how to choose system parameters such as F_{nuc} and t_{nuc}. Ideally, they can be expressed as fundamental system constants, in which case they can be compared to values found experimentally. Concerning the simulation, both running time and accuracy can be improved to conduct more thorough statistical analysis. Concrete suggestions are either implementing a more advanced integration scheme with an improved adaptive timestep or rewriting the ODE such that an automatic solver may be used, and speeding up computation by removing annihilated dislocations from computations. Methods of finding suitable simulation parameters would be valuable too.

Extending the scope, another relevant question is what happens when taking the many-particle limit $n \to \infty$ and the many-source limit $|C| \to \infty$ (it would be reasonable to choose n and C in the same regime), as well as formulating the system as Hamilton-Jacobi equation. The results thereof may pose an important criterion on which to compare different creation processes. Both the Peach-Koehler force and multipole expansion can be extended to this setting as in Appendix A.2.6. How to generalise creation procedures to a continuous setting is less trivial. Other possible extensions of the model we describe are to consider a higher-dimensional domain, adding random noise to the dislocation movement, or considering a periodic domain to approximate behaviour in an infinite medium with dislocations throughout.

In conclusion, we have developed a workable model for both theory and implementation of a 1D dislocation dynamics system with creation and annihilation, which to our knowledge did not yet exist. Besides value in its own right, with this work we hope to provide a foundation that can be built upon in further studies, ultimately contributing to the understanding material properties.

Bibliography

[1] H. H.M. Cleveringa, E. Van Der Giessen and A. Needleman. 'Discrete dislocation analysis of bending'. In: *International journal of plasticity* 15.8 (1999), pp. 837–868. ISSN: 07496419. DOI: 10.1016/S0749-6419(99)00013-3.

[2] Derek Hull and David J Bacon. *Introduction to Dislocations*. 4th. Butterworth-Heinemann, 2001. ISBN: 0750646810.

[3] S. Yefimov, I. Groma and E. Van der Giessen. 'A comparison of a statistical-mechanics based plasticity model with discrete dislocation plasticity calculations'. In: *Journal of the Mechanics and Physics of Solids* 52.2 (2004), pp. 279–300. ISSN: 00225096. DOI: 10.1016/S0022-5096(03)00094-2.

[4] Stéfan Van Der Walt, S. Chris Colbert and Gaël Varoquaux. 'The NumPy array: A structure for efficient numerical computation'. In: *Computing in Science and Engineering* 13.2 (2011), pp. 22–30. ISSN: 15219615. DOI: 10.1109/MCSE.2011.37.

[5] Thomas Hudson and Christoph Ortner. 'Existence and Stability of a Screw Dislocation under Anti-Plane Deformation'. In: *Archive for Rational Mechanics and Analysis* 213.3 (2014), pp. 887–929. ISSN: 14320673. DOI: 10.1007/s00205-014-0746-9.

[6] Patrick Van Meurs. 'Discrete-to-Continuum Limits of Interacting Dislocations'. PhD thesis. Eindhoven University of Technology, 2015, p. 234. ISBN: 9789038639048.

[7] Thomas Hudson. 'An existence result for Discrete Dislocation Dynamics in three dimensions'. In: (2018), pp. 1–31. URL: http://arxiv.org/abs/1806.00304.

[8] Mehran Monavari and Michael Zaiser. 'Annihilation and sources in continuum dislocation dynamics'. In: *Materials Theory* 2.1 (2018). DOI: 10.1186/s41313-018-0010-z.

[9] William D. Callister and David G. Rethwisch. *Materials science and engineering: An introduction*. 7th ed. Vol. 94. John Wiley & Sons, 2020, pp. 266–267. ISBN: 9780471736967. DOI: 10.1016/0025-5416(87)90343-0.

[10] Patrick Van Meurs, Mark Peletier and Norbert Požár. 'Discrete-to-continuum convergence of charged particles in 1D with annihilation'. In: (2021).

A Appendices

A.1 Notation

We use $f(t-) := \lim_{\tau \uparrow t} f(\tau)$ and $f(t+) := \lim_{\tau \downarrow t} f(\tau)$ for the left and right limits of f in t respectively.

Furthermore, throughout our work we adhere to the following notations:

Name	Definition/Type	Description	Reference
$\mathcal{B}$	$\subset \mathcal{B}(\mathbb{R}^+)$	Collection of charge functions	2.3.1
B_k	$\sum_{i=1}^n b_i(t) x_i(t)^k$	2^k-pole term in multipole expansion	4.1.1
$\mathbf{b}$	$(b_1, \dots, b_n) \in \mathbb{R}^n$	Dislocation charges (Burgers vectors)	
C	$\subset \mathbb{R}$	Finite set of source locations	2.3.2
F_{nuc}	$\in \mathbb{R}^+$	Creation force threshold (fixed)	
γ	$\mathbb{R}^+ \to \mathbb{R}$	Force exception function in γ-creation	3.1.3 & 3.1.4
k	$2 \cdot \lfloor T/t_{nuc} \rfloor$	Maximum number of creations at a source[1]	2.3.2
L_{nuc}	$\in \mathbb{R}^+$	Nucleation distance (for distance creation)	3.2.1
n	$\in \mathbb{N}$	(Initial) number of dislocations	
N	$n + \lvert C \rvert \cdot k$	Total nr. of dislocations in (PC_n) (system size)	
$\mathrm{PK}(s,t)$	$\sum_{i=1}^n \frac{b_i(t)}{s - x_i(t)}$	Peach-Koehler force	2.2.1
$\mathrm{PK}_k(s,t)$	$\sum_{j=0}^{\ell} \frac{B_j}{R^{j+1}}$	Multipole approximation of PK	4.1.1
$r(t)$	$x_2(t) - x_1(t)$	Difference of positions in 2-particle system	2.1.2 & 3.1.1
S	$\subset \mathbb{R}^+$	Finite set containing annihilation times	2.3.2
$\mathcal{T}$	$\subset \mathbb{R}^+$	Finite set containing creation times	2.3.2
T	$\in \mathbb{R}^+$	Right boundary of solution time-domain	
t_{nuc}	$\in \mathbb{R}^+$	Creation time threshold (fixed)	
t_{exc}	$\in \mathbb{R}^+$	Exception time (for γ-creation)	3.1.3 & 3.1.4
$\mathbf{x}$	$(x_1, \dots, x_n) \in \mathbb{R}^n$	Dislocation positions	
$\mathcal{Z}_n$	$\left\{ \begin{array}{l} (\mathbf{x}, \mathbf{b}) \in \mathbb{R}^n \times \{-1, 0, 1\} \text{ s.t} \\ (i > j \wedge b_i b_j \neq 0) \implies x_i > x_j \end{array} \right\}$	State space of (P_n)	
$\mathcal{Z}_n^c$	$\left\{ \begin{array}{l} (\mathbf{x}, \mathbf{b}) \in \mathbb{R}^n \times \{-1, 0, 1\} \text{ s.t} \\ (i \neq j \wedge b_i b_j \neq 0) \implies x_i \neq x_j \end{array} \right\}$	State space of (PC_n)	

Table A.1: Notations and their description.

A.2 Postponed details and proofs

A.2.1 ODE time-scaling

As mentioned in Section 2.1, [10] treats a slightly different system of ODEs than we do, multiplying the interaction by a factor $1/n$:

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = \frac{1}{n}\sum_{j\neq i}\frac{b_i b_j}{x_i - x_j} \quad \text{for } t \in (0,T), i = 1, ..., n \tag{A.1}$$

The reason for this is that one of the goals of that paper is to study the many-particle limit $n \to \infty$ with mean-field analysis. For us however, an additional factor depending on the size of the system makes it difficult to compare systems with differing numbers of particles. Hence, the systems we defined do not include this factor. We now show that the results derived in [10] still apply.

The idea is that a non-zero constant only changes the speed at which the system evolves, not the interaction behaviour itself. This can be seen at hand of the following example: suppose

$$\frac{\mathrm{d}x}{\mathrm{d}t} \;=\; \frac{1}{n}f(x(t),t).$$

We now rewrite by substituting $t = n \cdot \tilde{t}$ (and hence $\frac{\mathrm{d}\tilde{t}}{\mathrm{d}t} = 1/n$):

$$\frac{1}{n}f(x(t),t)) = \frac{\mathrm{d}x}{\mathrm{d}t} = \frac{\mathrm{d}x}{\mathrm{d}\tilde{t}}\frac{\mathrm{d}\tilde{t}}{\mathrm{d}t} = \frac{1}{n}\frac{\mathrm{d}x}{\mathrm{d}\tilde{t}} \implies \frac{\mathrm{d}x}{\mathrm{d}\tilde{t}} = f(x(n\cdot\tilde{t}), n\cdot\tilde{t})$$

which corresponds to scaling time in the original equation by n. In [10] it is also shown that from a solution to A.1 one may obtain another solution by scale invariance. This ensures that our analysis may be placed in the same framework as in [10].

A.2.2 Existence of a unique t_{exc} for linear γ

We continue from the end of Example 3.1.1, where we derived the following initial value problem:

$$\begin{cases} \dfrac{\mathrm{d}r}{\mathrm{d}t} = \dfrac{-1}{r(t)}\cdot\gamma(t) + F & \text{for } t \in (0, t_{exc}) \\ r(0) = 0 \end{cases}$$

To further analyse the ODE, we reparametrise by setting $R := \frac{1}{2}r^2$. Note that this does not limit us, since we are only interested in the situation where $r \geq 0$; if we had $r(t) < 0$ for some t, we could relabel the dislocations and obtain non-negative r, since the sign of r cannot change (i.e. dislocations cannot cross paths). This yields $r = \sqrt{2R}$. Then using the chain rule, we can rewrite (3.2) as follows:

$$\begin{aligned} & \frac{\mathrm{d}r}{\mathrm{d}t} = \frac{-1}{r(t)}\cdot\gamma(t) + F \\ \iff \quad & r\frac{\mathrm{d}r}{\mathrm{d}t} = -\gamma(t) + Fr \\ \iff \quad & \frac{\mathrm{d}R}{\mathrm{d}t} = -\gamma(t) + F\sqrt{2R} \end{aligned}$$

Furthermore, $R = 0$ if and only if $r = 0$, which yields the boundary condition $R(0) = 0$. Together, we find a new ODE system:

$$\begin{cases} \dfrac{\mathrm{d}R}{\mathrm{d}t} = -\gamma(t) + F\sqrt{2R} \quad \text{for } t \in (0, t_{exc}) \\ R(0) = 0 \end{cases} \tag{A.2}$$

Note that neither the assumption $b_1 = -1, b_2 = 1$ nor the requirement $F > 0$ are restrictive in the argument above. $F = 0$ will never lead to a creation event since no positive threshold is met, and if $F < 0$, then the sign of the dipole is reverted, yielding $b_1 = 1, b_2 = -1$ and equilibrium of 3.6 at $r = -1/2F$. We now consider (A.2) for fixed F and t_{exc}, but for different initial values:

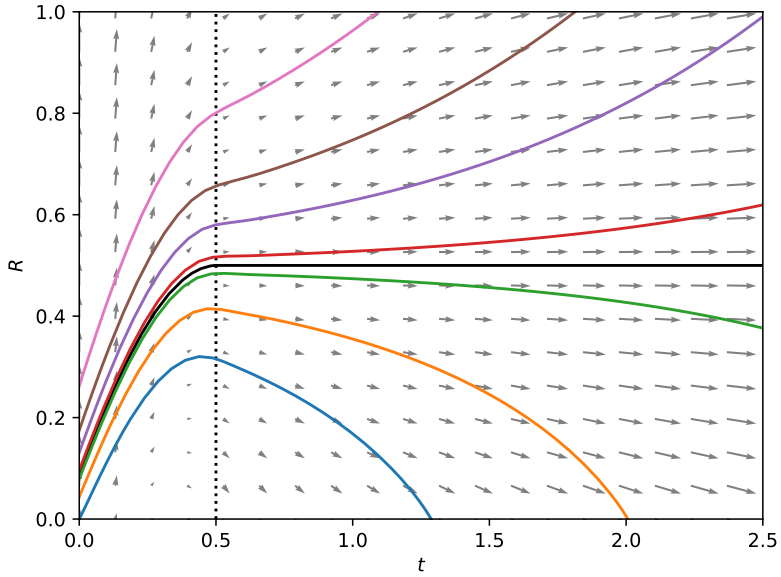


Figure A.1: Plot of (numerical) solutions to (A.2) for different initial values, for linear γ with $t_{exc} = 1/2$ (dotted line) and $F = 1$.

After t_{exc}, a stationary solution exists for $R \equiv 1/2F^2$. We also see that there is a 'critical' initial value, for $R(0) \approx 0.2$ with the given settings. Starting with smaller R leads to the trajectory returning to $R = 0$ (collision of dislocations), whereas starting at a higher value leaves R diverging.

This critical value may be found by solving the ODE from (A.2) backwards in time with final condition $R(t_{exc}) = \frac{1}{2F^2}$, and retrieving $R(0)$. Following this idea, we define the following function (where the brackets denote the solution to the given ODE):

$$R_{crit}(F, t_{exc}) \;:=\; \left\{ \begin{cases} \dfrac{\mathrm{d}R}{\mathrm{d}t} = -\gamma(t) + F\sqrt{2R} \\ R(t_{exc}) = \frac{1}{2F^2} \end{cases} \right\}\Bigg|_{t=0} \tag{A.3}$$

Well-posedness of this boundary-value problem and hence well-definedness of the function is assumed for now on some domain D. This will be discussed later. We do already restrict the domain of the function to exclude $F = 0$ and $t_{exc} = 0$, since these may cause singularities in the initial condition or the definition of γ respectively.

The definition in (A.3) can be difficult to analyse and is tedious to implement numerically, since many ODE solvers only approximate solutions forward in time, i.e., with a given condition for $t = 0$. Hence, it can be useful to invert time by the substitution $\tilde{t} = t_{exc} - t$. Thus reformulate this procedure equivalently as a usual boundary value problem with $\tilde{R}(0) = \frac{1}{2F^2}$. Note that this substitution changes the sign, which yields the following:

$$\begin{cases} \dfrac{\mathrm{d}\tilde{R}}{\mathrm{d}t} = \gamma(t_{exc} - t) - F\sqrt{2\tilde{R}} \quad \text{for } \tilde{t} \in (0, t_{exc}) \\ \tilde{R}(0) = \frac{1}{2F^2} \end{cases} \tag{A.4}$$

Now the critical initial value is given by $\tilde{R}(t_{exc})$. With this alternative formulation of the ODE, R_{crit} can also be written as follows:

$$R_{crit}(F, t_{exc}) \;=\; \left\{ \begin{cases} \dfrac{\mathrm{d}\tilde{R}}{\mathrm{d}t} = \gamma(t_{exc} - t) - F\sqrt{2\tilde{R}} \\ \tilde{R}(0) = \frac{1}{2F^2} \end{cases} \right\} \Bigg|_{t=t_{exc}} \tag{A.5}$$

To give an impression of the behaviour of the function, we plot $R_{crit}(\cdot, t_{exc})$ for varying values of t_{exc}:

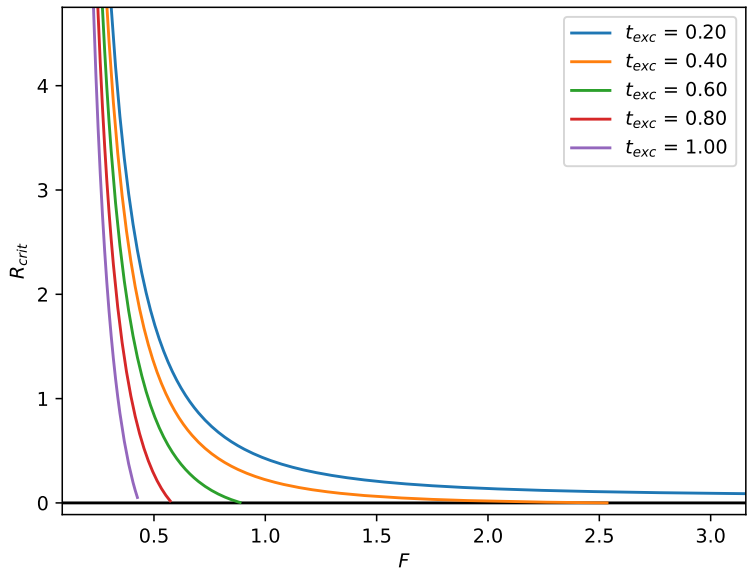


Figure A.2: Graphs of (numerical approximations of) (A.3) for varying values of t_{exc} and linear γ.

We find the following properties:

Lemma A.2.1 (Properties of R_{crit})**.** *For the function R_{crit} as given in* (A.3) *with linear γ as in* (3.3)*, the following properties hold:*

(i) (Well-definedness) R_{crit} is well-defined on a non-empty and connected subset of the open first quadrant of the (F, t_{exc})-plane, $(0, \infty)\times(0, \infty)$, and takes non-negative values. More specifically, if R_{crit} is well-defined for some $\hat{F}$ and $\hat{t}_{exc}$, it is also well-defined for F, t_{exc} such that $0 < F < \hat{F}$ and $0 < t_{exc} < \hat{t}_{exc}$.

(ii) (Continuity) R_{crit} is continuous for all $t_{exc} > 0$ and $F > 0$ for which it is well-defined

(iii) (Monotonicity in F) $R_{crit}(\cdot, t_{exc})$ is monotonically decreasing in F for all $t_{exc} > 0$

(iv) (Monotonicity in t_{exc}) $R_{crit}(F, \cdot)$ is monotonically decreasing in t_{exc} for all $F > 0$.

Proof. Let γ be given as in Equation (3.3), for an arbitrary $t_{exc} > 0$ unless specified otherwise.

(i) By standard ODE theory, the ODE in (A.5) is well-posed for any $F > 0$ and $t_{exc} > 0$ on the domain $[0, T]$ on which $\tilde{R}(t) \geq 0$, proving non-negativity of R_{crit}. Thus, as long as $\tilde{R}(t) \geq 0$ for all $t \in [0, t_{exc}]$, R_{crit} is well-defined.

We now show there exist F, t_{exc} such that $R_{crit}(F, t_{exc})$ is well-defined, by fixing some F, showing the ODE R must stay positive on some time-interval, and then choosing t_{exc} such that it falls within this interval.

Let $F < 1/\sqrt{2}$. Because $\gamma(t_{exc} - t)$ is continuous for $t \in [0, t_{exc}]$, $r \mapsto 2F\sqrt{2r}$ is continuous for $r \geq 0$ and $\tilde{R}(0) > 1$ (because of our choice of F), there must be an $\varepsilon > 0$ such that $\tilde{R} \geq 1$ for all $t \in [0, \varepsilon]$

Now let $t \in [0, \varepsilon]$. Then we have $\sqrt{\tilde{R}(t)} \leq \tilde{R}(t)$, and by definition we have $\gamma(t) \geq -1$. Estimating the ODE from (A.4), we obtain

$$\frac{\mathrm{d}\tilde{R}}{\mathrm{d}t} = \gamma(t_{exc} - t) - F\sqrt{2\tilde{R}} \geq -1 - F\sqrt{2\tilde{R}} \geq -1 - F\sqrt{2} \cdot \tilde{R},\ t \in [0, \varepsilon].$$

We obtain a linear first-order ODE $\mathrm{d}\bar{R}/\mathrm{d}t = -1 - \sqrt{2}F\bar{R}$, which we solve using standard methods (e.g. method of integrating factor) to obtain

$$\bar{R}(t) = c \cdot e^{-F\sqrt{2}\cdot t} - \frac{1}{F\sqrt{2}}, \quad c \in \mathbb{R}.$$

With the initial condition $\bar{R}(0) = 1/(2F^2)$ from (A.4), we obtain $c = 1/(F\sqrt{2}) + 1/(2F^2)$. Together with the fact that $t \leq t_{exc}$, this implies

$$\bar{R}(t) = \frac{1}{F\sqrt{2}} \cdot \left(\left(1 + \frac{1}{F\sqrt{2}} \right) e^{-F\sqrt{2}\cdot t} - 1 \right) \geq \frac{1}{F\sqrt{2}} \cdot \left(\left(1 + \frac{1}{F\sqrt{2}} \right) e^{-F\sqrt{2}\cdot t_{exc}} - 1 \right).$$

Choosing $t_{exc} \leq \ln 1/2 + 1/(2F\sqrt{2})/F\sqrt{2}$ we have $\left(1 + \frac{1}{F\sqrt{2}} \right) e^{-F\sqrt{2}\cdot t_{exc}} \geq 2$, which implies $\bar{R} \geq 1$ on $[0, t_{exc}]$, and hence $\bar{R}(t_{exc}) \geq 0$. Thus, R_{crit} is well-defined for arguments F, t_{exc}.

The initial condition is decreasing in F and the expression above is decreasing in t_{exc}. Thus, taking $\tilde{F}, \tilde{t}_{exc}$ such that $0 < \tilde{F} < F$ and $0 < \tilde{t}_{exc} < t_{exc}$ leaves the argument above valid, implying that $R_{crit}(\tilde{F}, \tilde{t}_{exc})$ is well-defined too.

(ii) Continuity of R_{crit} in both t_{exc} and F follows from the well-posedness of the ODE in (A.5), as this implies continuous dependence on the data.

(iii) *(Monotonicity in F)* Suppose $F_2 > F_1 > 0$, and let $t_{exc} > 0$ be fixed such that $R_{crit}(F_1, t_{exc})$ and $R_{crit}(F_1, t_{exc})$ exist. Furthermore, let R_1, R_2 be the solutions to the ODE given in (A.5) for F being F_1 and F_2 respectively. The idea of the proof is that solution trajectories cannot cross

paths, and thus preserve some monotonicity property; we now formalise this. By construction of R_{crit}, we have

$$\tilde{R}_1(0) = \frac{1}{2F_1^2} > \frac{1}{2F_2^2} = \tilde{R}_1(0).$$

We argue by contradiction; assume $\tilde{R}_1(t_{exc}) \leq \tilde{R}_2(t_{exc})$, i.e. $R_{crit}(F_1, t_{exc}) \leq R_{crit}(F_2, t_{exc})$. Then there must be a largest $t_0 \in (0, t_{exc})$ such that $\tilde{R}_1(t_0) = \tilde{R}_2(t_0)$ (e.g. by the intermediate value theorem; $R_1 - R_2$ changes sign). Then it follows that

$$\left(\tilde{R}^1\right)'(t_0) = \gamma(t_0) - F_1\sqrt{2\tilde{R}^1(t_0)} > \gamma(t_0) - F_2\sqrt{2\tilde{R}^1(t_0)} = \left(\tilde{R}^2\right)'(t_0)$$

This implies that for some $\varepsilon > 0$ we have $\tilde{R}_1(t_0 + \varepsilon) > \tilde{R}_2(t_0 + \varepsilon)$. Because t_0 is the largest value for t such that $\tilde{R}_1(t_0) = \tilde{R}_2(t_0)$ and $t_0 < t_{exc}$, it follows that $\tilde{R}_1(t_{exc}) > \tilde{R}_2(t_{exc})$; but this contradicts our assumption, thus proving our claim.

(iv) Now let $T_2 > T_1 > 0$ denote different values for t_{exc}, and fix $F > 0$ such that $R_{crit}(F, t_{exc})$ and $R_{crit}(F, t_{exc})$ exist (we now write T_i instead of $t_{exc\ i}$ for clarity). Now denote by R_1 and R_2 the respective solutions to (A.3) (i.e. not the time-inverted version). First off, note that contrary to the previous point, γ depends on the value of t_{exc}; we hence denote

$$\gamma_1(t) := \begin{cases} 2 \cdot \dfrac{t}{T_1} - 1 & \text{for } t \in [0, T_1) \\ 1 & \text{for } t \geq T_1 \end{cases}, \qquad \gamma_2(t) := \begin{cases} 2 \cdot \dfrac{t}{T_2} - 1 & \text{for } t \in [0, T_2) \\ 1 & \text{for } t \geq T_2 \end{cases}$$

Note that $T_1 < T_2$ implies that $\gamma_1(t) > \gamma_2(t)$ for $0 < t < T_2$, and that $\gamma_1(t) = \gamma_2(t)$ for $t = 0$ and $t \geq T_2$. We cannot directly compare the solution trajectories R_1 and R_2, since they are solutions to the ODE system in (A.3) with different functions γ. Instead, we scale the time in both systems by T_i, resulting in the following for $i = 1, 2$:

$$\begin{cases} \dfrac{\mathrm{d}R}{\mathrm{d}t} = -\gamma_i(t) + F\sqrt{2R} & \text{for } t \in [0, T_i] \\ R(T_i) = \frac{1}{2F^2} \end{cases}$$

$$\xrightarrow{\tau = t/T_i} \quad \begin{cases} \dfrac{\mathrm{d}R(T_i\tau(t))}{\mathrm{d}\tau}\dfrac{\mathrm{d}\tau}{\mathrm{d}t} = -\gamma_i(T_i\tau(t)) + F\sqrt{2R(T_i\tau(t))} & \text{for } t \in [0, T_i] \\ R(T_i) = \frac{1}{2F^2} \end{cases}$$

$$\xrightarrow{\bar{R}(\tau) = R(T_i \cdot \tau)} \quad \begin{cases} \dfrac{1}{T_i}\dfrac{\mathrm{d}\bar{R}(\tau)}{\mathrm{d}\tau} = -\gamma(\tau) + F\sqrt{2\bar{R}(\tau)} & \text{for } \tau \in [0, 1] \\ \bar{R}(1) = \frac{1}{2F^2} \end{cases}$$

Here $\bar{\gamma}$ represents (3.3) with $t_{exc} = 1$, but the exact description does not matter for our cause; what is more important is that $\bar{R}_1$ and $\bar{R}_2$ now share the same time-span and both contain $\bar{\gamma}$ instead of two separate functions.

Also transforming R_1 and R_2, it may be verified that $\bar{R}_1(\tau) = R_1(T_1 \cdot \tau)$ and $\bar{R}_2(\tau) = R_2(T_2 \cdot \tau)$ are solutions to the new ODE systems. It follows that we still have $R_{crit}(F, T_i) = R_i(0) = \bar{R}_i(0)$,

and of course $\bar{R}_1(1) = \bar{R}_2(1) = 1/2F^2$ because of the boundary condition. At any given $\tau \in (0,1)$, if $\bar{R}_1(\tau) \leq \bar{R}_2(\tau)$ it follows that

$$\frac{1}{T_1}\bar{R}_1'(\tau) = -\gamma(\tau) + F\sqrt{2\bar{R}_1(\tau)} \leq -\gamma(\tau) + F\sqrt{2\bar{R}_2(\tau)} = \frac{1}{T_2}\bar{R}_2'(\tau)$$

Recall that $T_1 < T_2$, implying $T_1/T_2 < 1$. This yields

$$\bar{R}_1'(\tau) \leq \frac{T_1}{T_2}\bar{R}_2'(\tau) < \bar{R}_2'(\tau).$$

Now as $\bar{R}_1(1) = \bar{R}_2(1)$, solving backwards in time we can conclude that $\bar{R}_1(1-s) > \bar{R}_2(1-s)$ for all $0 < s \leq 1$, implying that

$$R_{crit}(F, T_1) = \bar{R}_1(0) > \bar{R}_2(0) = R_{crit}(F, T_2)$$

which concludes the proof. □

Next to the proven properties, we conjecture further behaviour of R_{crit}. Proving or disproving these falls beyond the scope of our work and is left for further research.

Conjecture A.2.2 (Further properties of R_{crit})**.** *For the function R_{crit} as given in* (A.3) *with linear γ as in* (3.3)*, we conjecture the following properties:*

(i) (Limiting behaviour in F) $R_{crit}(\cdot, t_{exc})$ has a singularity at $F = 0$ for all $t_{exc} > 0$; to be more precise,

$$\lim_{F \to 0} R_{crit}(F, t_{exc}) = \infty \quad \forall t_{exc} > 0 \tag{A.6}$$

On the other hand, it seems that there exists a constant $c > 0$ such that for any $t_{exc} > c$, there exists a force $F_0 > 0$ such that $R_{crit}(F_0, t_{exc}) = 0$. If $t_{exc} < c$, then $R_{crit}(\cdot, t_{exc})$ has no zeroes.

(ii) (Limiting behaviour in t_{exc}) For $t_{exc} \to 0$, $R_{crit}(\cdot, t_{exc})$ converges pointwise to a continuous function supported on $\mathbb{R}^+$. For $t_{exc} \to \infty$, the domain of $R_{crit}(\cdot, t_{exc})$ narrows towards 0, i.e.,

$$\forall \varepsilon > 0 \; \exists t_{exc} \text{ s.t. } \operatorname{supp}(R_{crit}(\cdot, t_{exc})) \subset (0, \varepsilon).$$

If Conjecture A.2.2-(*ii*) is true, this implies that there is a lower bound $c > 0$ on all exception times t_{exc} in linear γ-creation as given in 3.1.3. Whether this is indeed the case and whether this c is a fundamental constant for the linear γ creation process remains to be seen in further research.

With the results proven in Lemma A.2.1, we can prove Lemma 3.1.2, which we restate below:

Lemma A.2.3. *For any given $F > 0$ exists a unique $t_{exc} > 0$ such that the creation of a dipole with linear γ, given by* (3.3)*, as adaptive force yields an equilibrium solution to 3.1.*

Proof. Let $F > 0$ be given.

Uniqueness: Suppose T_1 and T_2 are such that $R_{crit}(F, T_1) = R_{crit}(F, T_2) = 0$. Then from monotonicity of $R_{crit}(F, \cdot)$, proven as property (iv) of Lemma A.2.1, it follows that $T_1 = T_2$.

Existence: We have already established that there are pairs $F, t_{exc} > 0$ such that R_{crit} is positive, i.e., such that in (A.4), $\tilde{R}(t_{exc}) \geq 0$. If we now also show that there are pairs such that $\tilde{R}(t) = 0$ for $t \leq t_{exc}$, then by continuous dependence on initial data (also a 'shooting argument'), then for every F there must be a $\tilde{t}_{exc}$ such that $R_{crit}(F, \tilde{t}_{exc}) = 0$.

We follow a similar approach as in the proof of well-definedness of R_{crit}, in property (i) of Lemma A.2.1, only the other way around; we now show there exist F, t_{exc} such that in (A.4), $\tilde{R}(t) = 0$ for some $t < t_{exc}$.

Take $F > 1/\sqrt{2}$ and arbitrary $t_{exc} > 0$. Then $\tilde{R}(0) < 1$, and there exists an $\varepsilon > 0$ such that $\tilde{R}(t) < 1$ for $t \in [0, \varepsilon]$. For $t \in [0, \varepsilon]$, it follows that $\sqrt{\tilde{R}(t)} > \tilde{R}$. From (A.4) we then have

$$\frac{\mathrm{d}\tilde{R}}{\mathrm{d}t} = \gamma(t_{exc} - t) - F\sqrt{2\tilde{R}} \;=\; 1 - 2 \cdot \frac{t}{t_{exc}} - F\sqrt{2\tilde{R}} \;\leq\; 1 - 2 \cdot \frac{t}{t_{exc}} - F\sqrt{2} \cdot \tilde{R},\; t \in [0, \varepsilon].$$

Similar to the proof of (i) in Lemma A.2.1, we consider the following linear first-order ODE (only now, simply estimating $\gamma < 1$ does not give a sharp enough bound), writing $K := F\sqrt{2}$ for brevity:

$$\begin{cases} \dfrac{\mathrm{d}\bar{R}}{\mathrm{d}t} = 1 - 2 \cdot \dfrac{t}{t_{exc}} - K \cdot \bar{R} & \text{for } t \in [0, t_{exc}] \\ \bar{R}(0) = \frac{1}{2F^2}. \end{cases}$$

We recognise the form $\mathrm{d}y/\mathrm{d}t + p(t)y = q(t)$ with $p(t) \equiv K$ and $q(t) = \gamma(t_{exc} - t) = 1 - 2t/t_{exc}$. The integrating factor method provides a solution of the following form:

$$\frac{\mathrm{d}y}{\mathrm{d}t} + p(t)y = q(t) \quad \Longrightarrow \quad y(t) = e^{-\mu(t)} \int_0^t e^{\mu(\tau)} q(\tau) \mathrm{d}\tau, \;\text{ where }\; \mu(t) := \int_0^t p(\tau) \mathrm{d}\tau.$$

We have $\mu(t) = K \cdot t$, so (integrating by parts at $(*)$ we obtain

$$\begin{aligned} \bar{R}(t) \;&=\; e^{-\mu(t)} \int_0^t e^{\mu(\tau)} q(\tau) \mathrm{d}\tau \\ &=\; e^{-K \cdot t} \int_0^t e^{K \cdot \tau} \left(1 - 2\frac{\tau}{t_{exc}}\right) \mathrm{d}\tau \\ &\overset{(*)}{=}\; e^{-K \cdot t} \left(\left[\frac{e^{K \cdot \tau}}{K} - 2 \cdot \frac{\tau \cdot e^{K \cdot \tau}}{K \cdot t_{exc}} \right]_{\tau=0}^{\tau=t} + 2 \int_0^t \frac{e^{K \cdot \tau}}{t_{exc}} \mathrm{d}\tau \right) \\ &=\; e^{-K \cdot t} \left(\frac{e^{K \cdot t}}{K} - 2 \cdot \frac{t \cdot e^{K \cdot t}}{K \cdot t_{exc}} + 2\frac{e^{K \cdot t}}{t_{exc}} + c \right) \\ &=\; \frac{1}{K} \left(1 - \frac{2t + 2}{t_{exc}}\right) + c \cdot e^{-K \cdot t} \end{aligned}$$

where

$$c = \bar{R}(0) - \frac{1}{K} + \frac{2}{K \cdot t_{exc}} = \frac{1}{K^2} - \frac{1}{K} + \frac{2}{K \cdot t_{exc}} = \frac{1}{K} \left(\frac{1}{K} + \frac{2}{t_{exc}} - 1 \right).$$

By our choice of F, we have $1/K = 1/F\sqrt{2} < 1$. Now choosing t_{exc} small enough, we have $c > 0$, so that $\bar{R}$ is decreasing in t. Furthermore, the linear term $(1 - (2t + 2)/t_{exc})/K$ is decreasing in t. Taking t_{exc} small enough there must be some $t < t_{exc}$ such that $\bar{R}(t) = 0$.

By the above, this implies that there must be a $t_{exc} > 0$ such that $R_{crit}(F, t_{exc}) = 0$, proving existence. □

A.2.3 Full solution definitions

Linear γ

We consider the following system:

$$\begin{cases} \dfrac{\mathrm{d}x_i}{\mathrm{d}t} = \displaystyle\sum_{j\neq i} \frac{b_i b_j \gamma_{i,j}}{x_i - x_j} \quad \text{for } t \in (0,T), i = 1, ..., n \\ \text{with annihilation upon collision} \\ \text{with } \gamma\text{-creation} \end{cases} \tag{PC_n-γ}$$

Then a complete and formal definition of a solution to (PC_n-γ) is given by the following:

Definition A.2.4 (Solution to linear (PC_n-γ))**.** *Let $T > 0$, $n \geq 2$ and $(\mathbf{x}^0, \mathbf{b}^0) \in \mathcal{Z}_n$. Furthermore, let $t_{nuc}, F_{nuc} > 0$, let $C \subset \mathbb{R}$ be a finite set, and set $k := 2 \cdot \lfloor T/t_{nuc} \rfloor$. Enumerating all elements $s_i \in C$, additionally define $x^0_{n+2(i\cdot k+j)+\ell} := s_i$ for $i = 0, \dots, |C| - 1$, $j = 1, ..., k$ and $\ell = 0, 1$, and $b^0_{n+m} = 0$ for $m = 1, \dots, |C| \cdot k$.*

Then, setting $N := n + |C| \cdot k$, the functions $(\mathbf{x}, \mathbf{b}, \gamma) : [0,T] \to (\mathcal{Z}^c_N) \times \mathbb{R}^{N\times N}$ form a solution of (PC_n-γ) with linear γ-creation if there exist finite subsets $S, \mathcal{T} \subset (0,T]$ such that

(i) (Regularity) $\mathbf{x} \in C([0,T]) \cap C^1([0,T] \setminus (S \cup \mathcal{T}))$, $b_1, \dots, b_n : [0,T] \to \{-1, 0, 1\}$ are charge functions ($b_i \in \mathcal{B}$) and γ_{ij} is bounded for all $i, j = 1, \dots, N$;

(ii) (Initial condition) $(\mathbf{x}(0), \mathbf{b}(0)) = (\mathbf{x}^0, \mathbf{b}^0)$;

(iii) (Annihilation) If b_i jumps at $t > 0$ and $\lim_{s\uparrow t} \left|b_i(s)\right| = 1$, then $t \in S$. Moreover, for all $(\tau, y) \in S \times \mathbb{R}$,

$$\sum_{i\,:\,x_i(\tau)=y} \Big((b_i(\tau+) - b_i(\tau-) \Big) \;=\; 0$$

(iv) (Creation event) If b_i jumps at $t \in [0,T]$ and $b_i(t) = 0$, then there is exactly one other $j \in \mathbb{N}$ (i.e. $i \neq j$) such that $x_i(t) = x_j(t)$ and $b_i(t+) = -b_j(t+)$

(v) (Creation moment) Creation at $s \in C$ occurs at time $t \in [0,T]$ if and only if one of the following condition holds:

- *Either*

$$t - t_{nuc} \;=\; \sup\{\tau \leq t \;:\; PK(s,\tau) < F_{nuc} \text{ and no creation at } s \text{ in } (t - t_{nuc}, t)\}$$

- *Or*

$$t - t_{nuc} \;=\; \sup\{\tau \leq t \;:\; PK(s,\tau) > -F_{nuc} \text{ and no creation at } s \text{ in } (t - t_{nuc}, t)\}$$

Moreover, we then have $t \in \mathcal{T}_s$, and the union of $\mathcal{T}_s$ over all sources $s \in C$ is contained in $\mathcal{T}$

(vi) (Creation procedure) If $t^* \in \mathcal{T}_s$ for some $s \in C$, then $\lim_{t\downarrow t^*} b_i = \operatorname{sgn}(\mathrm{PK}(s, t^*))$ and $\lim_{t\downarrow t^*} b_{i+1} = -\operatorname{sgn}(\mathrm{PK}(s, t^*))$ for index $i = n + 2(m \cdot k + j)$ corresponding to the j^{th} creation at $s_m = s$.[2] Furthermore, $\gamma_{i,i+1} = 2t/t_{exc} - 1$ for $t \in (t^*, t^* + t_{exc}]$, where t_{exc} is the unique $t > 0$ such that such that in (3.1), $\frac{\mathrm{d}x_1}{\mathrm{d}t} = \frac{\mathrm{d}x_2}{\mathrm{d}t} = 0$. In all other cases, $\gamma_{j,k}(t) = 1$

[2] Recall that $k := 2 \cdot \lfloor T/t_{nuc} \rfloor$, where $t_{nuc} > 0$ is a fixed system constant.

(vii) (ODE for $\mathbf{x}$*) On* $(0,T)\backslash(S\cup\mathcal{T})$, $\mathbf{x}$ *satisfies the ODE in* (PC_n).

In Condition (vi), the requirement that in (3.1), $\frac{\mathrm{d}x_1}{\mathrm{d}t} = \frac{\mathrm{d}x_2}{\mathrm{d}t} = 0$ can be replaced by the equivalent requirement that $R_{crit}(\mathrm{PK}(s,t^*),t_{exc}) = 0$, following the derivation of (A.3).

Zero-γ

Again consider $(PC_n\text{-}\gamma)$, but now with zero-γ:

Definition A.2.5 (Solution to zero-$(PC_n\text{-}\gamma)$)**.** *Let* $T > 0$, $n \geq 2$ *and* $(\mathbf{x}^0,\mathbf{b}^0) \in \mathcal{Z}_n$. *Furthermore, let* $t_{nuc}, F_{nuc} > 0$, *let* $C \subset \mathbb{R}$ *be a finite set, and set* $k := 2\cdot\lfloor T/t_{nuc}\rfloor$. *Enumerating all elements* $s_i \in C$, *additionally define* $x^0_{n+2(i\cdot k+j)+\ell} := s_i$ *for* $i = 0,\dots,|C|-1$, $j = 1,...,k$ *and* $\ell = 0,1$, *and* $b^0_{n+m} = 0$ *for* $m = 1,\dots,|C|\cdot k$.

Then, setting $N := n+|C|\cdot k$, *the functions* $(\mathbf{x},\mathbf{b},\gamma) : [0,T] \to (\mathcal{Z}^c_N)\times\mathbb{R}^{N\times N}$ *form a solution of* $(PC_n\text{-}\gamma)$ *with zero-*γ *creation if there exist finite subsets* $S,\mathcal{T},E \subset (0,T]$ *such that*

(i) (Regularity) $\mathbf{x} \in C([0,T]) \cap C^1([0,T]\setminus(S\cup\mathcal{T}\cup E))$, $b_1,\dots,b_n : [0,T] \to \{-1,0,1\}$ *are charge functions (*$b_i \in \mathcal{B}$*) and* γ_{ij} *is bounded for all* $i,j = 1,\dots,N$*;*

(ii) (Initial condition) $(\mathbf{x}(0),\mathbf{b}(0)) = (\mathbf{x}^0,\mathbf{b}^0)$*;*

(iii) (Annihilation) If b_i *jumps at* $t > 0$ *and* $\lim_{s\uparrow t}\left|b_i(s)\right| = 1$, *then* $t \in S$. *Moreover, for all* $(\tau,y) \in S\times\mathbb{R}$,

$$\sum_{i\,:\,x_i(\tau)=y}\Big((b_i(\tau+)-b_i(\tau-)\Big) \;=\; 0$$

(iv) (Creation event) If b_i *jumps at* $t \in [0,T]$ *and* $b_i(t) = 0$, *then there is exactly one other* $j \in \mathbb{N}$ *(i.e.* $i \neq j$*) such that* $x_i(t) = x_j(t)$ *and* $b_i(t+) = -b_j(t+)$

(v) (Creation moment) Creation at $s \in C$ *occurs at time* $t \in [0,T]$ *if and only if one of the following condition holds:*

- *Either*

$$t - t_{nuc} \;=\; \sup\{\tau \leq t \;:\; PK(s,\tau) < F_{nuc} \text{ and no creation at } s \text{ in } (t-t_{nuc},t)\}$$

- *Or*

$$t - t_{nuc} \;=\; \sup\{\tau \leq t \;:\; PK(s,\tau) > -F_{nuc} \text{ and no creation at } s \text{ in } (t-t_{nuc},t)\}$$

Moreover, we then have $t \in \mathcal{T}_s$, *and the union of* $\mathcal{T}_s$ *over all sources* $s \in C$ *is contained in* $\mathcal{T}$

(vi) (Creation procedure) If $t^* \in \mathcal{T}_s$ *for some* $s \in C$, *then* $\lim_{t\downarrow t^*} b_i = \mathrm{sgn}(\mathrm{PK}(s,t^*))$ *and* $\lim_{t\downarrow t^*} b_{i+1} = -\mathrm{sgn}(\mathrm{PK}(s,t^*))$ *for index* $i = n+2(m\cdot k+j)$ *corresponding to the* j^{th} *creation at* $s_m = s$. *Furthermore,* $\gamma_{i,i+1} = \gamma_{i+1,i} = 0$ *for* $t \in (t^*,t^*+t_{exc}]$, *where* $t_{exc} = 1/(2\,\mathrm{PK}(s,t^*))^2$. *In all other cases,* $\gamma_{j,k}(t) = 1$. *Finally,* $t^*+t_{exc} \in E$

(vii) (ODE for $\mathbf{x}$*) On* $(0,T)\backslash(S\cup\mathcal{T})$, $\mathbf{x}$ *satisfies the ODE in* (PC_n).

Distance creation

Now consider regular system, i.e. without γ:

$$\begin{cases} \dfrac{\mathrm{d}x_i}{\mathrm{d}t} = \displaystyle\sum_{j\neq i} \frac{b_i b_j}{x_i - x_j} \quad \text{for } t \in (0,T), i = 1, ..., n \\ \text{with annihilation upon collision} \\ \text{with distance creation} \end{cases} \tag{PC_n-d}$$

Definition A.2.6 (Solution to PC_n-d)**.** *Let $T > 0$, $n \geq 2$ and $(\mathbf{x}^0, \mathbf{b}^0) \in \mathcal{Z}_n$. Furthermore, let $t_{nuc}, F_{nuc} > 0$, let $C \subset \mathbb{R}$ be a finite set, and set $k := 2 \cdot \lfloor T/t_{nuc} \rfloor$. Enumerating all elements $s_i \in C$, additionally define $x_{n+2(i\cdot k+j)+\ell} := s_i$ for $i = 0, \dots, |C| - 1$, $j = 1, ..., k$ and $\ell = 0, 1$ and $b_{n+m} = 0$ for $m = 1, \dots, |C| \cdot k$.*

Then, setting $N := n + |C| \cdot k$, the functions $(\mathbf{x}, \mathbf{b}) : [0,T] \to (\mathcal{Z}_N^c)$ form a solution of (PC_n) if there exist finite subsets $S, \mathcal{T} \subset (0,T]$ such that

(i) (Regularity) $\mathbf{x} \in C([0,T] \setminus \mathcal{T}) \cap C^1([0,T] \setminus (S \cup \mathcal{T}))$, and $b_1, \dots, b_n : [0,T] \to \{-1, 0, 1\}$ are charge functions ($b_i \in \mathcal{B}$);

(ii) (Initial condition) $(\mathbf{x}(0), \mathbf{b}(0)) = (\mathbf{x}^0, \mathbf{b}^0)$;

(iii) (Annihilation) $b_i(t) = 0$ (b is right-continuous). If b_i jumps at $t > 0$ and $\lim_{s\uparrow t} \left|b_i(s)\right| = 1$, then $t \in S$. Moreover, for all $(\tau, y) \in S \times \mathbb{R}$,

$$\sum_{i\,:\,x_i(\tau)=y} \Big((b_i(\tau+) - b_i(\tau-) \Big) \;=\; 0$$

(iv) (Creation event) If b_i jumps at $t \in [0,T]$ and $b_i(t) = 0$, then there is exactly one other $j \in \mathbb{N}$ (i.e. $i \neq j$) such that $x_i(t) = x_j(t)$ and $b_i(t+) = -b_j(t+)$

(v) (Creation moment) Creation at $s \in C$ occurs at time $t \in [0,T]$ if and only if one of the following condition holds:

- *Either*

$$t - t_{nuc} \;=\; \sup\{\tau \leq t \;:\; PK(s,\tau) < F_{nuc} \text{ and no creation at } s \text{ in } (t - t_{nuc}, t)\}$$

- *Or*

$$t - t_{nuc} \;=\; \sup\{\tau \leq t \;:\; PK(s,\tau) > -F_{nuc} \text{ and no creation at } s \text{ in } (t - t_{nuc}, t)\}$$

Moreover, we then have $t \in \mathcal{T}_s$, and the union of $\mathcal{T}_s$ over all sources $s \in C$ is contained in $\mathcal{T}$

(vi) (Creation procedure) If $t^* \in \mathcal{T}_s$ for some $s \in C$, then $\lim_{t\downarrow t^*} b_i = \operatorname{sgn}(\mathrm{PK}(s,t^*))$ and $\lim_{t\downarrow t^*} b_{i+1} = -\operatorname{sgn}(\mathrm{PK}(s,t^*))$ for the index $i = n + 2(m \cdot k + j)$ corresponding to the j^{th} creation at $s_m = s$. Furthermore, setting $L_{nuc} := 1/2\,\mathrm{PK}(s,t^*)$, we have $x_i(t^*+) = s - L_{nuc}/2$ and $x_{i+1}(t^*+) = s + L_{nuc}/2$.

(vii) (ODE for $\mathbf{x}$) On $(0,T)\setminus(S \cup \mathcal{T})$, $\mathbf{x}$ satisfies the ODE in (PC_n).

A.2.4 Discontinuity of the Peach-Koehler force for distance creation

Suppose a dipole $+,-$ with distance L is created at time t around s. Furthermore, assume the Peach-Koehler force has positive sign. Then similar to the proof of Lemma 4.2.1, we obtain

$$\begin{aligned}
\lim_{\tau \downarrow t} \mathrm{PK}(R,\tau) - \mathrm{PK}(R,t) &= \lim_{\tau \downarrow t} \left[\sum_{i=1}^{n+2} \frac{b_i(\tau)}{R - x_i(\tau)} \right] - \sum_{i=1}^{n+2} \frac{b_i(t)}{R - x_i(t)} \\
&= \lim_{\tau \downarrow t} \left[\frac{b_+(\tau)}{R - x_+(\tau)} + \frac{b_-(\tau)}{R - x_-(\tau)} \right] - \frac{b_+(t)}{R - x_+(t)} - \frac{b_-(t)}{R - x_-(t)} \\
&= \frac{b_+(t)}{R - x_+(t)} + \frac{b_-(t)}{R - x_-(t)} \\
&= \frac{1}{R - s - L/2} - \frac{1}{R - s + L/2} \\
&= \frac{(R - s + L/2) - (R - s - L/2)}{(R - s - L/2)(R - s + L/2)} \\
&= \frac{L}{(R-s)^2 - L^2/4}.
\end{aligned}$$

Here from Definition 3.1.4, one may substitute $L = 1/\mathrm{PK}(s,t)$.

A.2.5 Uniqueness of solutions with zero-γ up to first annihilation

In the proof of Theorem 5.1.1, the case $t_0 \in S$ and $t_0 \notin \mathcal{T}$ may be continued as follows:

Proof. Continued. By standard ODE theory, as the system has no discontinuities or singularities on $[0, t_0)$, solutions $(\mathbf{x}, \mathbf{b})$ and $(\tilde{\mathbf{x}}, \tilde{\mathbf{b}})$ are equal on that interval. Because $t_0 \in S$, we know there is an annihilation point (x, t_0) of $\mathbf{x}$. Furthermore, by continuity we have $\mathbf{x}(t_0) = \tilde{\mathbf{x}}(t_0)$, so (x, t_0) is also an annihilation point for $\tilde{\mathbf{x}}$, so $t_0 \in \tilde{S}$.

Now let I be the set of indices of all (charged) colliding particles, i.e.

$$I \ := \ \{i \in \mathbb{N} \,:\, x_i(t_0) = x,\ b_i(t_0-) \neq 0\}.$$

Then by the annihilation condition (point (iii) in Definition A.2.5), we have that

$$\sum_{i \in I} \tilde{b}_i(t_0) = \sum_{i \in I} \tilde{b}_i(t_0-) = \sum_{i \in I} b_i(t_0-) = \sum_{i \in I} b_i(t_0),$$

so the total charge after annihilation is equal for both solutions. Moreover, there is at most one $i \in I$ such that $b_i(t_0) \neq 0$, since $(\mathbf{x}, \mathbf{b})$, $(\tilde{\mathbf{x}}, \tilde{\mathbf{b}}) \in \mathcal{Z}_n^c$ and this implies that for all $j \neq i$, we have $x_i(t_0) = x_j(t_0) \implies b_i(t_0) b_j(t_0) = 0$. Therefore, the multiset (unordered tuple) of $b_i(t_0)$ and $\tilde{b}_i(t_0)$ are equal, i.e., there exists a permutation $\sigma \in S_n$ such that for all $i \in I$, $b_i(t_0) = \tilde{b}_{\sigma(i)}(t_0)$.

Therefore at t_0, the ODEs for $\mathbf{x}$ and $\tilde{\mathbf{x}}$ are identical up to relabelling, so that again by standard ODE theory, the solutions $(\mathbf{x}, \mathbf{b})$, $(\tilde{\mathbf{x}}, \tilde{\mathbf{b}})$ are equal on some time-interval $[t_0, t_0 + \varepsilon)$ not containing the next creation or annihilation event.

Proof continues below Theorem 5.1.1. □

A.2.6 Continuous versions of PK and multipole expansion

We can also define a continuous version of the Peach-Koehler force. Let $\rho : \mathbb{R} \times [0, \infty) \to \mathbb{R}^+$ be such that for $t > 0$, $\rho(\cdot, t)$ is charge density, i.e. $\int \rho(x,t) \mathrm{d}x = 1$. Then Definition 2.2.1 can be adapted to

$$\mathrm{PK}(x,t) \ := \ \int \frac{\rho(\tau)}{x - \tau} \mathrm{d}\tau$$

Note that for the continuous version of the Peach-Koehler force, there is also a form of multipole expansion; in this setting, we assume that $\rho(x) = 0$ for $|x| > R$ for some $r > 0$ with $r < R$. Then

$$\begin{aligned}
\mathrm{PK}(R) &= \int_{\mathbb{R}} \frac{\rho(x)}{R - x} \mathrm{d}x \\
&= \frac{1}{R} \int_{\mathbb{R}} \frac{\rho(x)}{1 - \frac{x}{R}} \mathrm{d}x \\
&= \frac{1}{R} \int_{\mathbb{R}} \rho(x) \sum_{j=0}^{\infty} \left(\frac{x}{R} \right)^j \mathrm{d}x \\
&= \sum_{j=0}^{\infty} \frac{1}{R^{j+1}} \int_{\mathbb{R}} \rho(x) x^j \mathrm{d}x
\end{aligned}$$

Here the final integral is exactly the j^{th} moment of the distribution with density ρ.

A.3 Simulation code

A.3.1 Manual solver with annihilation and creation

This concerns the file `1DDislocationsCreation.py`. Plots from Section 6.3.2 can be recreated by uncommenting the indicated lines (i.e. setting `N,M = 20,20` and `creaProc = 'lin'`, `'zero'` or `'dist'`.

```
import numpy as np
import matplotlib.pyplot as plt
import time as timer
from scipy.optimize import root

#%% INITIALISATION

### Simulation settings
simTime = 0.12          # Total simulation time
dt = 0.001               # Timestep for discretisation (or maximum, if adaptive)
minTimestep = 1e-7       # Minimum size of timestep (for adaptive)
adaptiveTime = True      # Whether to use adaptive timestep in integrator
reg = 'cutoff'           # Regularisation; either 'eps', 'V1', 'cutoff' or 'none'
eps = 0.01               # Regularisation parameter (for all three methods)
randomness = False       # Whether to add random noise to dislocation positions
sigma = 0.01             # Standard dev. of noise (volatility)
withAnnihilation = True # Whether dislocations disappear from system after annihilation
collTres = 3e-3          # Collision threshold
stress = 0               # External force (also called 'F')
withCreation = False      # Whether to include creation
creaProc = 'zero'        # Creation procedure; either 'lin', 'zero' or 'dist'
Fnuc = 5                # Threshold for magnitude of Peach-Koehler force
tnuc = 0.01             # Threshold for duration of PK force magnitude before creation
domain = (0,1)           # Interval where initial dislocations and sources are placed


# # For Figure 6.4a, uncomment following lines:
# withCreation = False     # Whether to include creation
# N, M = 20, 20 # Number of initial dislocations and sources respectively

# # For Figure 6.4b, uncomment following lines:
# withCreation = True      # Whether to include creation
# N, M = 20, 20
# creaProc = 'lin'

# # For Figure 6.4c, uncomment following lines:
# withCreation = True      # Whether to include creation
# N, M = 20, 20
# creaProc = 'zero'

# # For Figure 6.4d, uncomment following lines:
# withCreation = True      # Whether to include creation
# N, M = 20, 20
# creaProc = 'dist'


def setExample(N):
    """
    Initialises positions and charges of given nr of dislocations
```

```
    Parameters
    ----------
    N : int
        number of dislocations to initialise. 0,1,2,5 gives fixed examples,
        other N give randomised settings with equally many +/- dislocations.

    Returns
    -------
    None (sets global variables).

    """
    global initialPositions, initialCharges, initialNrParticles

    if N == 0: ### Empty example; requires some adaptions in code below
        # (e.g. turn off 'no dislocations left' break)
        initialPositions = np.array([np.nan, np.nan])
        initialCharges = np.array([1, -1])


    elif N == 1: ### Example 1:
        initialPositions = np.array([0.5])
        initialCharges = np.array([1])


    elif N == 2: ### Example 2:
        initialPositions = np.array([0.3, 0.75])
        initialCharges = np.array([1, -1])


    elif N == 5: ### Example 3:
        initialPositions = np.array([0.02, 0.2, 0.8, 0.85, 1])
        initialCharges = np.array([-1, -1, 1, 1, -1]) # Particle charges


    elif N == -1: # Additional comparison case (randomly generated but fixed):
        initialPositions = np.array([0.00727305, 0.04039581, 0.25157344, 0.2757077, 0.28350536,
                                     0.36315111, 0.60467167, 0.68111491, 0.72468363, 0.7442808 ])
        initialCharges = np.array([-1.,  1., -1., -1.,  1.,  1.,  1., -1.,  1., -1.])

    elif N == 20: # Additional comparison case (randomly generated but fixed):
        initialPositions = np.array([0.90041661, 0.09512205, 0.93625452, 0.67799578, 0.49170662,
            0.1327828 , 0.17790777, 0.76411685, 0.97124885, 0.22572291,
            0.4294073 , 0.87120555, 0.60016304, 0.97865076, 0.52582236,
            0.64176168, 0.10342922, 0.26874082, 0.6207242 , 0.95723599])
        initialCharges = np.array([-1.,  1.,  1.,  1., -1., -1., -1.,
                                   -1., -1., -1.,  1., -1.,  1.,  1.,
                                    1., -1., -1.,  1.,  1.,  1.])

    else: ### Given nr of particles, and option
        initialPositions = np.random.uniform(size = N)
        initialCharges = np.ones(N)
        neg = np.random.choice(range(N),N//2, replace=False) #pick floor(N/2) indices at random
        initialCharges[neg] = -1 # Set half of charges to -1, rest remains 1
        # note: different from completely random charges, which would be the following:
        # initialCharges = np.random.choice((-1,1),nrParticles)
```

```
    # Dependent paramaters (deducable from the ones defined above):
    initialNrParticles = len(initialPositions)



def setSources(M):
    """
    Initialises positions and charges of given nr of sources

    Parameters
    ----------
    M : int
        number of sources to initialise (except -1). -1,1 gives fixed examples,
        -1 serving as fixed reference case with 5 non-evenly distributed sources.
        Other values give N sources evenly distributed on interval [0,1]

    Returns
    -------
    None (sets global variables).

    """
    global nrSources, sourceLocs, nrBackgrSrc, backgrSrc
    # Initialise sourceLocs for creation:

    if M == -1:
        # sourceLocs = np.array([0.21, 0.3, 0.45, 0.75, 0.8]) # Irregularly spaced sources
        sourceLocs = np.array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ])

    elif M == 1:
        sourceLocs = np.array([0.55])


    else: # Evenly distribute given nr of sources
        sourceLocs = np.linspace(0, 1, M)


    nrSources = len(sourceLocs)


setExample(N)
if withCreation:
    setSources(M)




# %% DEFINITIONS
def pairwiseDistance(x1, x2 = None):
    """ Compute distances between given positions

    Computes distances between all atoms, optionally for
    closest copies taking boundaries into account.

    INPUT: np array of coordinates (single or multiple dimensions)
    OUTPUT: array of difference vectors and one of distances
            between all coordinates
    """
```

```
    if x2 is None:
        x2 = x1

    diff = x1 - x2[:,np.newaxis] # Difference vectors

    dist = np.abs(diff) # Compute length of difference vectors

    return diff, dist



def interaction(diff,dist,b, regularisation = 'eps'):
    """
    Compute array of pairwise particle interactions for given
    array of particle coordinates and charges

    Parameters
    ----------
    diff : 2D numpy array (nxn)
        pairwise differences between all particle positions.
    dist : 2D numpy array (nxn)
        pairwise distances between all particle positions.
    b : 1D numpy array (n)
        charges of dislocations.
    regularisation : 'eps', 'V1', 'cutoff' or 'none', optional
        interaction regularisation method. The default is 'eps'.

    Returns
    -------
    interactions : 2D numpy array (nxn)
        representing force pairs of particles exert on each other.

    """
    # Set distance from particle to itself to (arbitrary) non-zero value to
    # avoid div by 0; arbitrary, since this term cancels out anyway
    np.fill_diagonal(dist, 1)
    chargeArray = b * b[:,np.newaxis] # Create matrix (b_i b_j)_ij

    if regularisation == 'eps':
        distCorrected = (dist**2 + eps**2)
    else:
        distCorrected = dist**2 # Square to normalise difference vector

        # interactions = -(1/diff) * chargeArray # Only in 1D; else need diff/dist

    # Calculate matrix b_i b_j / (x_i - x_j):
    interactions = -(diff / distCorrected) * chargeArray
    interactions = np.nan_to_num(interactions) # Set NaNs to 0

    if regularisation == 'V1':
        interactions[dist < eps] = diff/eps**2

    if regularisation == 'cutoff': # Set values outside [-c,c] to  closest boundary
        interactions = np.clip(interactions, -1/eps, 1/eps)

    return interactions
```

```
def PeachKoehler(sourceLocs, x, b, stress, regularisation = 'eps'):
    """Computes Peach-Koehler force for each source, possibly for regularised interaction

    (Sources are typically fixed equidistant grid points) """

    x = np.nan_to_num(x) #sets NaNs to 0
    diff, dist = pairwiseDistance(x, x2 = sourceLocs)

    if regularisation == 'eps':
        distCorrected = (dist**2 + eps**2)
    else:
        distCorrected = dist**2

    interactions = (diff / distCorrected) * b[np.newaxis,:] + stress
    # (Stress according to final expression in meeting notes of 211213)
    interactions = np.nan_to_num(interactions) # Set NaNs to 0

    f = np.sum(interactions, axis = 1) # Per source, sum contributions over all dislocs

    return f



def texcToForce(t):
    """ Given force exception time, computes corresponding force yielding equilibrium"""
    params = [0.283465, -0.013909, 0.000511, 0.325376]
    a,b,c,d = params

    return a/(t-d) + b/(t-d)**2 + c/(t-d)**3



def poly3(x, a,b,c,d):
    """ Computes third-degree polynomial in x with coefficients a,b,c,d """
    return a * x**3 + b* x**2 + c * x + d


def forceTotexcSlow(F):
    """ Given force, computes corresponding exception time yielding equilibrium
        using root-finding algorithm. """
    params = [0.283465, -0.013909, 0.000511, 0.325376]
    a,b,c,d = params

    sol = root(poly3, 1/F, args=(-F,a,b,c))

    return sol.x + d



def forceTotexcFast(PK):
    """ Given force, computes corresponding exception time yielding equilibrium
        using explicit formula for root of a 3rd-degree polynomial. """
    params = [0.283465, -0.013909, 0.000511, 0.325376]
    a,b,c,d = params
```

```
    x = np.abs(PK)

    A = -2* a**3 - 9 *a* b* x - 27* c* x**2
    B = -a**2 - 3* b* x
    C = (3 *x * (A + np.sqrt(A**2 + 4* B**3))**(1/3))

    rt = (2**(1/3) * B)/C - C/(3* 2**(1/3) *x) + (a + 3* d* x)/(3* x)

    return rt


class Source:
    """ Represents a creation source and tracks creation thresholds.

    Attributes:
        pos: float,
            position of source
        tAboveThreshold: float (positive)
            tracks how long threshold is reached

    Methods:
        updateThresTime: updates threshold time depending on
                         whether force threshold is reached
    """

    def __init__(self, loc):
        """ Initialises source at position loc and time 0 (via Source(loc)) """

        self.loc = loc
        self.tAboveThreshold = 0


    def updateThresTime(self, PK, dt):
        """Given force and timestep, either increments threshold or sets to 0 """
        if (np.abs(PK) > Fnuc):
            self.tAboveThreshold += dt
            if self.tAboveThreshold >= tnuc:
                self.tAboveThreshold = 0
                return True
        else:
            self.tAboveThreshold = 0

        return False



class Creation:
    """ Single creation event (and exception time in case of gamma-creation)

    Attributes:
        loc: float
            location at which creation occurred
        PKAtCrea: float
            Peach-Koehler force at creation moment and location
        creaTime: float (positive)
            time at which creation occurred
```

```
        idx: integer
            index in position array of first created dislocation (from pair)
        inProgress: boolean
            indicating whether creation process is still in progress,
            i.e. whether force exception should still hold
        """

    def __init__(self, loc, PK, t, idx):
        """ Initialisation of creation event at given parameters"""

        self.loc = loc
        self.PKAtCrea = PK
        self.creaTime = t
        self.idx = idx
        self.inProgress = True


    def createDipole(self):
        """ Creation of dislocations and definition of creation parameters
            according to set creation procedure. """

        if creaProc == 'lin':
            self.texc = forceTotexcFast(self.PKAtCrea) # Choose fnct giving texc here

            locs = np.array([self.loc - 0.5*collTres, self.loc + 0.5*collTres])

        elif creaProc == 'zero':
            self.texc = 1/(2*np.abs(self.PKAtCrea))**2
            locs = np.array([self.loc - 0.5*collTres, self.loc + 0.5*collTres])

        elif creaProc == 'dist':
            self.Lnuc = 1/(np.abs(self.PKAtCrea))
            locs = np.array([self.loc - 0.5*self.Lnuc, self.loc + 0.5*self.Lnuc])

        charges = np.array([1,-1])*np.sign(self.PKAtCrea)

        return locs, charges


    def forceAdjustment(self, t):
        """ Compute factor with which interaction between created pair is
            multiplied to adjust forces """

        if creaProc == 'lin':
            forceFact = (2*(t - self.creaTime)/self.texc - 1)

        elif creaProc == 'zero':
            forceFact = 0

        return forceFact


    def exceptionCheck(self, t):
        """ Check whether exception still applies to created pair """

        if (creaProc == 'lin') or (creaProc == 'zero'):
            if t > self.creaTime + self.texc:
```

```
                self.inProgress = False
        elif creaProc == 'dist':
            self.inProgress = False


class Annihilation:
    """ Describes annihilation event

    Attributes:
        annTime: float (positive)
            time at which annihilation event occurred
        dislocs: tuple of integers
            indices of dislocations involved in annihilation/collision
    """
    # Note: currently, this does not need to be a class. However, extensions
    #       with greater functionality may be desirable

    def __init__(self, annTime, dislocs):
        self.annTime = annTime
        self.dislocs = dislocs



# %% SIMULATION

### Precomputation
t = 0
stepSizes = []
times = [0]
maxUpdate = 0 # To store for computing adaptive timestep
maxTimestep = dt

trajectories = initialPositions[None,:] # Change shape into (1,len)
x = np.copy(initialPositions)
b = np.copy(initialCharges)
# ('copy' to create new array; otherwise, is just a
# reference and changes if initialCharges changes)

annihilations = [] # To keep track of annihilation events (only for later analysis)

if withCreation: # Initialise sources and store classes in list
    sources = [Source(x) for x in sourceLocs] # Initialise source classes and store in list
    creations = []  # List of all creations (for analysis afterwards)
    currentCreations = [] # List to keep track of creations for which exception holds

simStartTime = timer.time() # To measure simulation computation time

### Simulation loop
while t < simTime:
    # Creation:
    if withCreation: # Check creation moment condition:
        PK = PeachKoehler(sourceLocs, x, b, stress)

        locs = []
        charges = []

        for i,src in enumerate(sources):
```

```
        thresReached = src.updateThresTime(PK[i], dt)
        # (Sources updated, boolean indicates whether threshold is reached)
        if thresReached: # If threshold is reached, initiate creation procedure:
            # Initialise Creation:
            newCrea = Creation(src.loc, PK[i], t, len(x) + 2*len(creations))
            creations.append(newCrea) # Append to  list of all creations
            currentCreations.append(newCrea) # Append to list of current creations
            # Obtain corresponding dipole locations and charges:
            locs, charges = Creation.createDipole(newCrea)
            x = np.append(x, locs) # Add new dipole to position vector
            b = np.append(b, charges) # Add new dipole to charge vector

            # Store charges of new For correct plot colours
            initialCharges = np.append(initialCharges, charges)
            # extend _entire_ position array (over all timesteps) with NaNs.
            trajectories = np.append(trajectories,
                                     np.zeros((len(trajectories), len(locs)))*np.nan,
                                     axis = 1) #


    # Check whether creations still have exception:
    for crea in currentCreations:
        crea.exceptionCheck(t)

    # Remove Creations that have no force exception (anymore) from list:
    currentCreations = [crea for crea in currentCreations if crea.inProgress]


# Compute main forces/interaction:
diff, dist = pairwiseDistance(x)


if withAnnihilation: # Make dislocs annihilate when necessary
                     #(before computing interactions, s.t. annihilated
                     # dislocs indeed do not influence the system anymore)
    tempDist = np.nan_to_num(dist, nan = 1000) + 1000*np.tril(np.ones((len(x),len(x))))
    # Set nans and non-above-diagonal entries to arbitrary large number, so
    # that effectively all entries on and below diagonal are disregarded below

    chargeArray = b * b[:,np.newaxis] # Create matrix (b_i b_j)_ij
    collPart1, collPart2 = np.where((tempDist < collTres) * (chargeArray == -1))
    # Identifies pairs with opposite charge closer than collTres together
    #('*' works as and-operator for 0/1 booleans). Format: ([parts A], [parts B]).

    # Go through list sequentially, annihilating not-yet-annihilated dislocs.
    for i in range(len(collPart1)):
        i1 = collPart1[i]
        i2 = collPart2[i]
        if b[i1] != 0 and b[i2] != 0:
            b[i1] = 0
            b[i2] = 0
            x[i1] = np.nan
            x[i2] = np.nan


interactions = interaction(diff,dist,b, regularisation = reg)
```

```
    # Adjust forces between newly created dislocations
    # (keeping track of time since each creation separately)
    if withCreation:
        for crea in currentCreations: # only contains pairs with force exception
            j = crea.idx
            interactions[j : j + 2, j : j + 2] *= crea.forceAdjustment(t)


    ## Main update step:
    updates = np.nansum(interactions,axis = 1)  # Treats NaNs as zero


    if adaptiveTime:
        # rudimentary adaptive timestep; always between (minTimestep, dt)
        # dt = np.clip(0.001/np.max(np.abs(updates)), minTimestep, maxTimestep)

        # better: also always between (minTimestep, dt)
        newMaxUpdate = np.max(np.abs(updates)) # For computing adaptive timestep
        approxD2 = (newMaxUpdate - maxUpdate)/dt # Estimate 2nd derivative
        dt = np.clip(1/approxD2, minTimestep, maxTimestep)
        stepSizes.append(dt)
        maxUpdate = newMaxUpdate # Interchange to store for next iteration



    x_new = x + updates * dt
    # (Alternative file available with built-in ODE-solver, without creation)

    if randomness:
        random = sigma * np.random.normal(size = len(x)) # Random part of update
        x_new += random * np.sqrt(dt)


    # Store all positions for visualisation:
    trajectories = np.append(trajectories, x_new[None,:], axis = 0)


    x = x_new
    t += dt
    times.append(t)


    # Stop simulation if all dislocations have annihilated:
    # (in this case PK=0, so no creations can occur anymore either)
    if np.isnan(x_new).all():
        print('No dislocations left')
        break # Breaks from outer 'while'-loop


    if((10*t/simTime) % 1 < 2*dt):
        print(f"{t:.5f} / {simTime}")


#Compute and print copmutation time for simulation
duration = timer.time() - simStartTime
print("Simulation duration was ", int(duration/3600), 'hours, ',
```

```
                                int((duration%3600)/60), " minutes and ",
                                int(duration%60), "seconds")


# %% VISUALISATION

#1D plot:
def plot1D(bInitial, trajectories, t, log = False):
    global pos, x_temp
    plt.clf() # Clears current figure

    trajectories = np.ndarray.squeeze(trajectories)
    colorDict = {1:'red', 0:'grey', -1:'blue'}

    nrPoints, nrTrajectories = np.shape(trajectories)

    y = t
    plt.ylim((0,1.1*t[-1]))

    if log: # Plot with time on log-scale
        t[0] = t[1]/2 # So that first timestep is clearly visible in plot.
                      # Not quite truthful, but also not quite wrong.
        plt.yscale('log')
        plt.ylim((t[0],t[-1]))


    for i in range(nrTrajectories):
        x_current = trajectories[:,i]
        y_current = y

        plt.plot(x_current, y_current, c = colorDict.get(bInitial[i]))
        # Set colour of created dislocations according to charge they
        # eventually get (not 0, which they begin with)



def printSummary():
    global nrRecollided, annDislocs, creaDislocs, overlap
    if withCreation:
        print("Total nr of Creations: ", len(creations))
    print("Total nr of Annihilations: ", len(annihilations))
    if t < simTime:
        print("Total annihilation time: ", t)
    else:
        print("Total annihilation time: not reached")

    if withCreation:
        if len(creations) > 0:
            # See how many created dipoles recollided:
            annDislocs = np.zeros((len(annihilations),2))
            creaDislocs = np.zeros((len(creations),2))
            nrRecollided = 0
            for i,ann in enumerate(annihilations):
                annDislocs[i,:] = np.sort(ann.dislocs) #sort on lower index first to ease comparison below

            for i,crea in enumerate(creations):
                creaDislocs[i,:] = np.array([crea.idx, crea.idx + 1])
```

```
            # Create list of pairs that were created, but also annihilated
            overlap = [pair for pair in creaDislocs if pair in annDislocs]

            nrRecollided = len(overlap)
            fracSurvived = 1-nrRecollided/len(creations)

            print(len(creations), " creations, ",
                  len(creations) - nrRecollided, " survived. (i.e. ",
                  100*fracSurvived, "%)")

    if withCreation:
        return len(annihilations), len(creations), nrRecollided, fracSurvived
    else:
        return len(annihilations)


plot1D(initialCharges, trajectories, times, log = False)
printSummary()
```

A.3.2 Automatic solver with annihilation

This concerns the file `1DautoSolver.py`, and is referred to in Section 6.2.2 as the implementation using a black-box automatic ODE solver.

```
import numpy as np
import matplotlib.pyplot as plt
import time as timer
from scipy.integrate import solve_ivp


# %% Simulation settings
simTime = 0.1          # Total simulation time
reg = 'none'          # Regularisation technique; for now either 'eps' or 'cutoff'
eps = 0.01         # To avoid singular force making computation instable.
boxLength = 1        # Domain width for initial coniguration
annihilation = True # Simulate with/without annihilation upon collision
collTres = 1e-4 # Collision threshold



def setExample(N, boxLen = 1):
    '''
    Sets initial configuration (location and charge) as global parameters
    for 3 minimal examples (N = 0,1,2) or any given number of dislocations (N > 2)

    '''
    global boxLength, initialPositions, b, creations, domain, initialNrParticles

    boxLength = boxLen

    if N == 0: ### Example 0:
        initialPositions = np.array([-0.1, 0.1])
        b = np.array([1, 1])
    elif N == 1: ### Example 1:
        initialPositions = np.array([0.21, 0.7, 0.8])
        b = np.array([-1, 1, 1])
    elif N == 2: ### Example 2:
        initialPositions = np.array([0.02, 0.2, 0.8, 0.85, 1])
        b = np.array([-1, -1, 1, 1, -1]) # Particle charges
    elif N == 4:
        initialPositions = np.array([-2,-1,1,2])
        b = np.array([-1, 1, -1, 1])
    elif N == 6:
        initialPositions = np.array([-2,-1,-0.7, 0.7,1,2])
        b = np.array([-1, 1,-1, 1, -1, 1])
    else: ### Example 3: Arbitrary number of particles
        initialPositions = np.random.uniform(size = N, low = 0, high = boxLength)
        #charges:
        b = np.ones(N)
        neg = np.random.choice(range(N),N//2, replace=False)
        b[neg] = -1

    # Dependent paramaters (deducable from the ones defined above):
    initialNrParticles = len(initialPositions)
    domain = (0,boxLength)


setExample(50, boxLen = boxLength)
```

```
# To plot:
bInitial = np.copy(b)
annTimes = np.zeros(len(b))


# %% FUNCTIONS

def pairwiseDistance(x1, PBCs = True, x2 = None):
    """ Compute distances between all given coordinates,
    optionally with PBCs and optionally to a second given set of coordinates

    INPUT: array of coordinates (single or multiple dimensions)
           (opt) boolean PBCs, whether to use periodic boundary conditions
           (opt) second coordinates array, computes pairwise distance between
    OUTPUT: array of difference vectors and one of distances
            between all coordinates
    """

    if x2 is None:
        x2 = x1

    diff = x1 - x2[:,np.newaxis] # Difference vectors
    if PBCs: # Calculate difference vector to closest copy of particle
        diff = diff - np.floor(0.5 + diff/boxLength)*boxLength

    dist = np.abs(diff) # Length of difference vectors (in 1D)

    return diff, dist



def f(t,x, regularisation = 'eps', PBCs = False):
    """ Compute discrete gradient flow for given array of coordinates
    according to Equation (9.1) from vMeurs15

    uses global parameters b (charges), eps, and
    updates global parameters b and annTimes

    t is dummy argument used by ODE solver"""
    global b2
    diff, dist = pairwiseDistance(x, PBCs = PBCs)
    # Set distance from particle to itself to (arbitrary) non-zero value to
    # avoid div by 0; arbitrary, since this term cancels out anyway
    np.fill_diagonal(dist, 1)
    chargeArray = b * b[:,np.newaxis] # Create matrix b_i b_j

    if regularisation == 'eps':
        distCorrected = (dist**2 + eps**2)
    else:
        distCorrected = dist**2


    interactions = -(diff / distCorrected) * chargeArray #len(b) is nr of particles
    interactions = np.nan_to_num(interactions) # Set NaNs to 0

    if regularisation == 'V1':
```

```
        interactions[dist < eps] = diff/eps**2

    if regularisation == 'cutoff': # Set all values outside [-c,c] to closest boundary
        interactions = np.clip(interactions, -1/eps, 1/eps)


    updates = np.nansum(interactions,axis = 1)

    if annihilation:
        collPart1, collPart2 = np.where((dist < collTres) * (chargeArray == -1))
        # Identifies pairs with opposite charge closer than collTres together
        #('*' works as 'and' for 0/1 booleans). Format: ([parts A], [parts B]).

        bOld = np.copy(b) #to compare below, seeing whether annihilation happened

        # Go through list sequentially, annihilating not-yet-annihilated dislocs.
        for i in range(len(collPart1)):
            i1 = collPart1[i]
            i2 = collPart2[i]
            if b[i1] != 0 and b[i2] != 0 and i1 != i2:
                b[i1] = 0
                b[i2] = 0

        annTimes[bOld != b] = t #save for plotting


    return updates



# %% SIMULATION

simStartTime = timer.time() # To measure simulation computation time

### Simulation:
sol = solve_ivp(f, [0, simTime], initialPositions, method = 'BDF', max_step = 0.01)
# (BDF is best for problems that may be stiff)

duration = timer.time() - simStartTime
print("Simulation duration was ", int(duration/3600), 'hours, ',
                                  int((duration%3600)/60), " minutes and ",
                                  int(duration%60), "seconds")


# %% VISUALISATION:

def plotODESol(solution, charges, log = False, annihilationTimes = None):
    ''' Given a solution from solve_ivp, plots trajectories according to given charges.
    Optional:
     - plot on log-scale if log = True
     - stop plotting if all dislocations annihilated
    '''

    plt.clf()
    t = sol.t
    x = np.transpose(sol.y)
```

```
    colorDict = {1:'red', 0:'grey', -1:'blue'}
    nrParticles = len(x[0])
    plt.ylim((0,t[-1]))
    plt.xlim((np.min(x[np.isfinite(x)]),np.max(x[np.isfinite(x)])))
    if log:
        t[0] = t[1]/2 #So that first timestep is clearly visible (avoid log(0))
        plt.yscale('log')
        plt.ylim((1e-6,t[-1]))


    for i in range(nrParticles) :
        if annihilationTimes is None:
            t_current = t
            x_current = x[:,i]
        else: # Only plot trajectories up to annihilation
            t_current = t[t < annTimes[i]]
            x_current = x[:len(t_current),i]

        x_new = x_current
        t_new = t_current

        plt.plot(x_new, t_new, c = colorDict.get(charges[i]))

    plt.show()

if annihilation:
    plotODESol(sol, bInitial, log = False, annihilationTimes=annTimes)
else:
    plotODESol(sol, bInitial, log = False)
```

A.3.3 Analysis of R_{crit}

This concerns the file `ODEPhaseplot.py`. The figures in this report concerning the analysis of Linear γ can be reproduced by executing the corresponding (commented) lines in the final section of the code.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp
from scipy.optimize import fsolve
from scipy.optimize import root
from scipy.optimize import curve_fit


F = 1
texc = 0.5
rmin = 0
rmax = 1.0
tmax = 2.5


#%% (Adapted) Creation ODE and 'creation exception function' gamma:
def gamma(t, texc = texc):
    return min(2*t/texc - 1, 1)
    # if t < texc:
    #     return 2*t/texc - 1
    # else:
    #     return 1


def f(t, R, gamma = gamma, F = F, texc = texc):
    # np.abs(R) yields non-physical negative solutions for R (idea of starting
    # at negative distance); prevents errors. This is accounted for later on
    return -gamma(t, texc = texc) + F*np.sqrt(2*np.abs(R))



#%% Plots:

### Plot inherent boundaries and system constants
def plotFeatures():

    plt.clf()
    plt.vlines(texc, rmin, 10*rmax, linestyles = 'dotted') #vertical line at t = texc

    #"level set" of points with derivative 0
    def separation(t):
        return 0.5*((gamma(t))/F)**2

    ts = np.linspace(0.5*texc, tmax, 100)
    ys = np.zeros(100)

    for i in range(100):
        ys[i] = separation(ts[i])



def fprime(t, R, texc = texc):
```

```
    """ plots the 'critical solution', above which R
    diverges and below which R goes to 0: """
    return -f(texc - t, R, texc = texc)


def critical(F, texc, plot = False):
    """ given F, texc, finds initial condition yielding equilibrium solution """
    global critR, thres

    thres = np.reshape(1/(2*F**2), (1,))
    # (Reshape is technicality, to account both for F given as 'c' or '[c]')

    boundSol2 = solve_ivp(fprime, [0, texc], thres,
                          method = 'BDF', max_step = 5e-2)
    critR = boundSol2.y[0][-1]

    if plot:
        boundSol1 = solve_ivp(f, [texc, tmax], [np.squeeze(thres)],
                              method = 'BDF', max_step = 5e-2)
        plt.plot(boundSol1.t,np.squeeze(boundSol1.y), color = 'black')
        # 'invert' ODE because we solve backwards to t=0:
        plt.plot(texc - boundSol2.t,np.squeeze(boundSol2.y), color = 'black')

        print(f"Critical initial condition: R = {critR}" )

    return critR



def plotSols():
    """Print range of solution trajectories """
    for R0 in np.array([0, 0.5, 0.9, 1.1, 1.5, 2, 3])*critR:
        sol = solve_ivp(f, [0,tmax], [R0], method = 'BDF', max_step = 5e-2)
        #set maximum timestep, else solver 'gets arrogant' and inaccurate
        plt.plot(sol.t,np.squeeze(sol.y))



def plotPhaseVF():
    """ Plot phase-space vector field"""
    rlim = max(rmax, 2.5*thres)
    ts = np.linspace(0, tmax, 20)
    rs = np.linspace(0, rlim, 20)
    T, R = np.meshgrid(ts, rs)


    u, v = np.zeros(T.shape), np.zeros(R.shape)
    NI, NJ = T.shape

    for i in range(NI):
        for j in range(NJ):
            der = f(T[i, j],R[i, j]) #derivative value according to ODE
            u[i,j] = T[i,j]
            v[i,j] = der
```

```
    plt.quiver(T, R, u, v, alpha = 0.5) # in form (x coords, y coords, x dir, y dir)

    plt.xlabel('$t$')
    plt.ylabel('$R$')
    plt.ylim([rmin, rmax])
    plt.xlim([0,tmax])


def plotAll():
    """ Execute all plot-functions above """
    plotFeatures()
    critical(F, texc, plot = True)
    plotSols()
    plotPhaseVF()



#%% More critical-value analysis: critical initial value for R, varying F and texc


def initialCrit(Fmin, Fmax, N, tmin, tmax, M, plot = True):
    """ For range of forces and times, plots IC yielding equilibrium solution """
    plt.clf() # Clears current figure

    Fs = np.linspace(Fmin, Fmax, N)
    ts = np.linspace(tmin, tmax, M)
    crits = np.zeros((M,N))
    for j in range(M):
        texc = ts[j]
        for i in range(N):
            crits[j,i] = critical(Fs[i], texc)

        Rs = crits[j,:]
        Rs[Rs < 0] = np.nan #remove negative values (so solver works more easily)
        plt.plot(Fs, Rs, label = "$t_{exc}$ =" + f" {ts[j]:.2f}")
        # (only round (:.2f) in case linspace is coarse)

    plt.hlines(0, 0, Fmax)
    plt.xlabel('$F$')
    plt.ylabel('$R_{crit}$')
    plt.xlim([Fmin, Fmax])

    plt.legend()

    # describe(crits)


def plotRcrit():
    Fmin = 0.2
    Fmax = 10
    N = 100

    tmin = 0.2
    tmax = 1
    M = 5
    initialCrit(Fmin, Fmax, N, tmin, tmax, M) # Critical R for varying F, texc
```

```
#%% One level deeper: for each texc, pick F s.t. Rcrit = 0. (bad computation time...)


def FzeroSolver(ts, method):
    """ For each texc, attempts to find F s.t. Rcrit = 0 """

    M = len(ts)
    Fzeros = np.zeros(M)
    reachedSol = np.zeros(M, dtype = int)

    for j in range(M):
        texc = ts[j]
        if method == 'fsolve':
            x, infodict, ier, msg = fsolve(critical, 1/texc,
                                           full_output = True, args = (texc,))
            Fzeros[j] = x

            reachedSol[j] = (ier == 1)

        if method == 'root':
            sol = root(critical, 1/texc, method = 'broyden1', args = texc)
            Fzeros[j] = sol.x
            reachedSol[j] = sol.success # Store whether root-finder was succesful

        print(f"Step {j} out of {M}, succesful: {reachedSol[j]}")


    plt.clf()
    tsProper = ts[reachedSol == 1]
    FzerosProper = Fzeros[reachedSol == 1]
    plt.plot(tsProper, FzerosProper)

    plt.xlabel('$t_{exc}$')
    plt.ylabel('$F$')

    return tsProper, FzerosProper



def plotRcritZeros():
    tmin = 0.35
    tmax = 1.5
    M = 100
    ts = np.linspace(tmin, tmax, M) #evenly distributed
    method = 'root'  #very slow, but seems accurate
    # method = 'fsolve' # fast but error-prone

    # For given t, find F s.t. Rcrit = 0:
    tsProper, FzerosProper = FzeroSolver(ts, method)

    return tsProper, FzerosProper


def modelFunc(x, d,e,f,k):
    """ Function to fit to FzeroSolver """
```

```
    return d/(x-k) + e/(x-k)**2 + f/(x-k)**3


def fitFunction(F,t):
    """ Fit and plot approximation of FzeroSolver """

    # Fit data to modelFunc:
    params, _ = curve_fit(modelFunc, t, F, p0 = (1,1,1,0.3)) # p0 is first guess


    print("Optimal parameters: ", params)
    a,b,c,d = params
    print(f"Optimal function: f(x) = {a}/(x-{d}) + {b}/(x-{d})**2 + {c}/(x-{d})**3")
    print(f"Rounded: f(x) = {a:.3f}/(x-{d:.3f}) \
          + {b:.3f}/(x-{d:.3f})**2 + {c:.3f}/(x-{d:.3f})**3")

    # Plot original and fitted function:
    tSol = np.linspace(0.37, 1.5, 1000)
    FSol = modelFunc(tSol, a,b,c,d)

    plt.clf()
    plt.plot(FSol, tSol, label = "Fitted", color = 'orange', zorder = 0)
    plt.plot(F, t, label="Original", zorder = 1)
    plt.xlabel('$F$')
    plt.ylabel('$t_{exc}$')
    plt.legend()



#%% Executables (reproducing report):

## Plot phase-space and range of solutions:
plotAll() # Phase-space, trajectories & vector field


## Plot Rcrit:
# plotRcrit()


## Find and plot zeros of Rcrit:
# t, F = plotRcritZeros()


## Fit solution to curve of given form:  [execute below arrays]

# Solving is slow, so may take following data instead of plotRcritZeros():

F = np.array([4.52249233,  3.25458218,  2.66708927,  2.30421261,
        2.05243906,  1.86080379,  1.71050999,  1.58773009,  1.48501998,
        1.39732073,  1.32130856,  1.25497266,  1.19522794,  1.14207018,
        1.09396342,  1.05035073,  1.01065947,  0.97400351,  0.94019942,
        0.90895822,  0.87986711,  0.8529585 ,  0.82745218,  0.80383456,
        0.78152787,  0.76045845,  0.74060595,  0.72184348,  0.70408308,
        0.68725822,  0.67118846,  0.65595032,  0.64140218,  0.62749767,
        0.61426562,  0.60162896,  0.5894524 ,  0.5778138 ,  0.56663221,
        0.55590293,  0.54561666,  0.53566809,  0.52610706,  0.51692251,
        0.50805281,  0.49946465,  0.49118272,  0.48316996,  0.47545857,
```

```
        0.46797684,  0.46075236,  0.45371297,  0.4469083 ,  0.44031849,
        0.43393784,  0.42771814,  0.42166954,  0.4158048 ,  0.41011023,
        0.40458888,  0.39918341,  0.39394864,  0.38883055,  0.38386148,
        0.37903058,  0.37429789,  0.36969419,  0.36520798,  0.36084195,
        0.35656104,  0.35238137,  0.34830612,  0.34432858,  0.34045044,
        0.33664585,  0.33292527,  0.32929026,  0.32575529,  0.32227046,
        0.31886116,  0.31553138,  0.31227159,  0.30909019,  0.30595748,
        0.30288957,  0.29988414,  0.29694334,  0.29406616,  0.29123311,
        0.28845638,  0.28573354,  0.28307541,  0.28045133,  0.27787487,
        0.27534974])

t = np.array([0.37323232, 0.38484848, 0.39646465, 0.40808081,
        0.41969697, 0.43131313, 0.44292929, 0.45454545, 0.46616162,
        0.47777778, 0.48939394, 0.5010101 , 0.51262626, 0.52424242,
        0.53585859, 0.54747475, 0.55909091, 0.57070707, 0.58232323,
        0.59393939, 0.60555556, 0.61717172, 0.62878788, 0.64040404,
        0.6520202 , 0.66363636, 0.67525253, 0.68686869, 0.69848485,
        0.71010101, 0.72171717, 0.73333333, 0.74494949, 0.75656566,
        0.76818182, 0.77979798, 0.79141414, 0.8030303 , 0.81464646,
        0.82626263, 0.83787879, 0.84949495, 0.86111111, 0.87272727,
        0.88434343, 0.8959596 , 0.90757576, 0.91919192, 0.93080808,
        0.94242424, 0.9540404 , 0.96565657, 0.97727273, 0.98888889,
        1.00050505, 1.01212121, 1.02373737, 1.03535354, 1.0469697 ,
        1.05858586, 1.07020202, 1.08181818, 1.09343434, 1.10505051,
        1.11666667, 1.12828283, 1.13989899, 1.15151515, 1.16313131,
        1.17474747, 1.18636364, 1.1979798 , 1.20959596, 1.22121212,
        1.23282828, 1.24444444, 1.25606061, 1.26767677, 1.27929293,
        1.29090909, 1.30252525, 1.31414141, 1.32575758, 1.33737374,
        1.3489899 , 1.36060606, 1.37222222, 1.38383838, 1.39545455,
        1.40707071, 1.41868687, 1.43030303, 1.44191919, 1.45353535,
        1.46515152])


# fitFunction(F,t)
```

An incomplete list of points of improvement for this thesis:

- Report - general
  - analysis of simulation is very brief
  - Plots in report lack labels on axes
  - Plots are not well-chosen, as they show not too varying behaviour and some need too much analysis/explanation
  - Requirement that $\forall s \in C : |R| > |s|$ is not emphasised in multipole analysis
  - Row labels in Table 4.1 are bad
- Definition
  - initialisation of particles that are never created is redundant, and explicit indexing not preferred
- Theoretical results
  - Repeating argument in 5.1.1 is non-rigorous
  - What if annihilation happens at same moment creation occurs in Theorem 5.1.1?
- Simulation
  - Forward-Euler may be replaced by any (better) forward integration scheme
  - choice of timestep is not yet great
  - Simulation loop order is now creation-annihilation-store positions, meaning created and immediately annihilating dislocations are not plotted
  - simulation not sufficiently tested for errors
  - Counting of surviving dipoles has suspicious results; survival rate may be wrong...
  - There is no particle ordering in simulation. This could solve many issues
  - black-box automatic ODE solver with adapting global variables is 'nasty'
  - Restarting system and removing non-charged dislocations could speed up simulation
  - More general remark: we actually deviate from the exact definition by not initialising all N dislocations. Instead, we are closer to the alternative state-space mentioned at the end of Section 2.3, except that we do not remove dislocations upon annihilation (yet)

Fixed mistakes:

- in Table 4.1, results of columns 'zero' and 'linear' were swapped
- Code recreating plots can now be run more easily
- there was still an erroneous $1/n$ in (PC_n-γ), Section3.1
- 'with creation' in (PC_n) was too vague, now specified to Frank-Read (but is this better? Still have to describe this!)

- In Table 4.1, mixed F_{nuc} and PK.
- Main goal ((PC_n)) was imprecise, stating 'with creation'; now specifically mentions Frank-Read sources
- Particles/dislocations were used interchangeably; maybe not wrong, but confusing

Possible questions:

- How do F_{nuc}, t_{nuc}, system size (both domain and number of particles) relate?
- Why is $|b|$ lsc, and not continuous in a different way? First, because otherwise at a creation and/or annihilation moment, one might have $x_i = x_j$ and $b_i \neq 0, b_j \neq 0$; this is not in $\mathcal{Z}_n$ nor $\mathcal{Z}_n^c$. This, in turn, is because otherwise there is no necessity