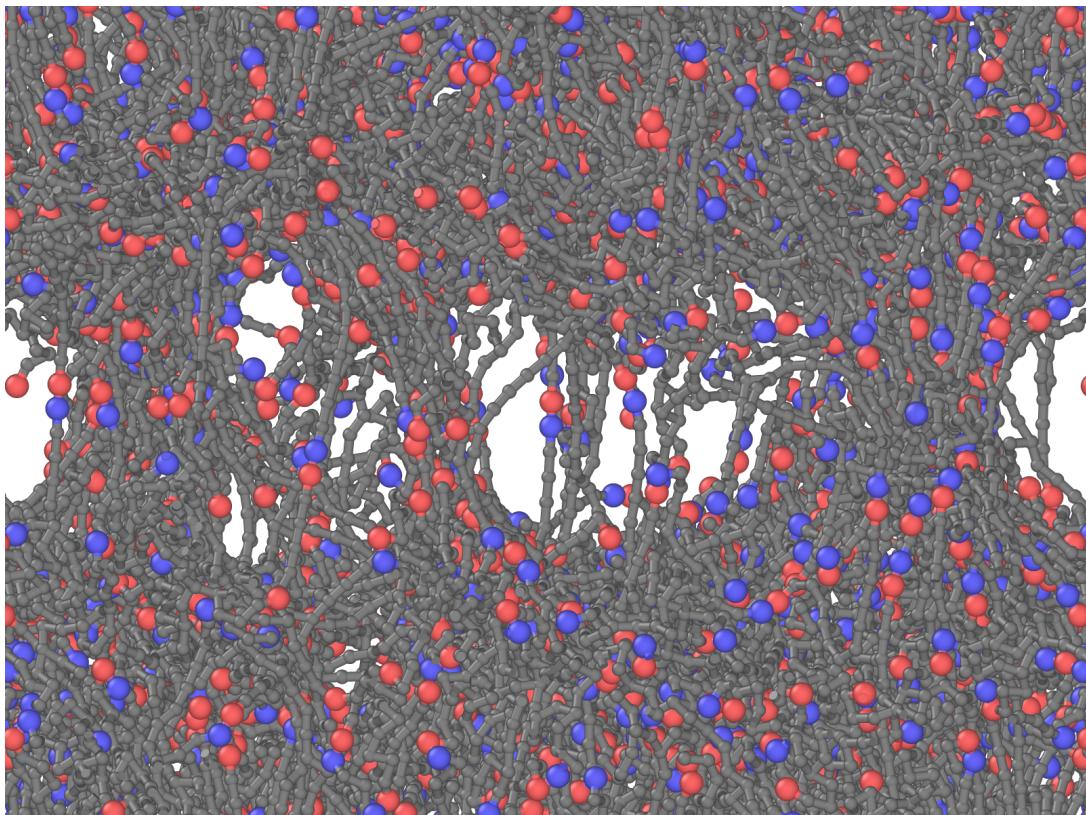


Self-Healing of Comb Polymer Vitrimer



Author

F.M. Sherry (1310380)

Supervisors

Dr. W.G. Ellenbroek & Dr. B. Baumeier

Departments of Applied Physics & Mathematics and Computer Science

Eindhoven University of Technology

Friday 7th July, 2023

Abstract

To combat the many challenges to sustainability posed by a ‘take-make-waste’ society, the European Union wants to transition to a ‘circular economy’, in which products are re-used as much as possible. The usable lifetimes of the materials these products are made of must therefore be extended. Due to their omnipresence, improving the lifespan of polymers is of particular importance; we can achieve this by making the polymers capable of autonomous self-healing, so that they can heal damage incurred during operation without any external interference. This promising idea has been held back by the poor mechanical properties of typical self-healing polymers. Fortunately, in 2019, Ciarella and Ellenbroek developed a vitrimer network based on star polymer that was capable of self-healing, while remaining rigid on timescales around three orders of magnitude greater than the healing time [1]. In this work, we seek to improve upon this result by using comb instead of star polymers, hypothesising that the networks will heal on similar timescales, which being stable for much longer due to the entanglement of the backbones.

We first develop a simple theoretical model to predict the typical bond swap times in the network, and test this using Molecular Dynamics simulations; our model appears to extrapolate well outside of the data it was fitted on. The simulations additionally show that the swap times cannot be described by an exponential or Weibull distribution.

Using these typical swap times, we derive an expression for the evolution of the number of bonds around damage and the equilibrium shear modulus. These models are again tested using Molecular Dynamics simulations. We see that the vitrimer networks based on combs indeed heal at similar speeds to the one based on stars, confirming the first part of our hypothesis.

Finally, we measure the stress relaxation of a comb vitrimer network using Molecular Dynamics simulations to find the timescale on which the network is rigid. Unfortunately, we believe we have encountered an issue in the implementation of the dynamic bonds in simulations, causing us to observe a nonphysical stress relaxation; we therefore have been unable to test the second part of our hypothesis.

In short, vitrimers made of combs are promising candidates for a self-healing polymer; their long-term stability remains to be investigated in future research.

Contents

1	Introduction	3
2	Theory	5
2.1	Bond swaps	5
2.1.1	Empirical Lifetime Distributions	6
2.2	Linear Viscoelasticity	12
2.2.1	Affine Network Model	12
2.2.2	Stress Relaxation in Vitrimers	14
2.3	Polymer Morphology	15
3	Simulation Protocol	16
3.1	Potentials and Parameters	17
3.1.1	Integrators and Thermostats	20
3.2	Network Generation and Evolution	21
3.2.1	Simulation Parameters	21
3.2.2	Network Generation	22
3.2.3	Network Evolution	23
3.3	RevCross GPU Issues	23
4	Mathematical Model	24
4.1	Swap Time	24
4.1.1	Interassociation Time	27
4.1.2	Model Fitting	31
4.1.3	Simulations	32
4.1.4	Conclusions	37
4.2	Healing Swaps	37
4.2.1	Rate Equations	38
4.2.2	Equilibrium Shear Modulus	40
4.2.3	Simulations	43
4.2.4	Conclusions	48
4.3	Vitrimer Relaxation	48
5	Discussion	51
5.1	Conclusions and Limitations	51
5.2	Outlook	52
A	Supplementary Proofs	59
A.1	Tails of Improper Riemann Integrals	59
A.2	Chain Length Distributions	59
A.3	Exponential Splitting	63

Chapter 1

Introduction

The current ‘take-make-waste’ society poses myriad challenges, such as the loss of biodiversity and vast greenhouse gas emissions [2]. To combat these issues, the European Union (EU) has proposed a transition to a ‘circular economy’. A central aspect of a circular economy is the waste hierarchy, shown in Figure 1.1, which defines how we should preferably deal with the 2.2 billion tones of wastes we generated each year [3]. At the top of the hierarchy is prevention, which means that we should re-use products as much as possible [4]. It is therefore clear that the success of the circular economy rests on the existence of materials that allow for re-use.

In this context, polymers deserve particular attention due to their ubiquity and the environmental impact of the techniques commonly employed to deal with them at their end-of-life, such as landfilling and incineration [5, 6]; there is consequently a lot to be gained by extending the lifetime of polymer materials. There are numerous ways in which the longevity of polymer could be increased. One approach, which has been frequently studied in the last decade (e.g. [7, 8, 9, 10, 11]), is to improve the reprocessibility of polymers, so that they can be made into new products more easily. However, we can do even better by preventing the need for reprocessing in the first place, for instance by making the polymers capable of healing damage they incur during their life. Ideally, this healing happens autonomously, with as little outside intervention as possible [9]. *Self-healing* polymers are already in use for instance in the automotive industry [12], but have been held back by numerous deficiencies. For instance, they often require some trigger such as temperature to activate the healing [13], and they are typically weak and prone to damage [14], mitigating the advantages of damage recovery.

To understand these weaknesses, we must look to the past. We have historically been able to divide polymer materials into two categories, depending on how their constituent polymers are bound together [15]. On the one hand we have *thermosets*, which have been cured irreversibly into an polymer network [16]. In this process permanent crosslinks are formed, which tend to make thermosetting materials tough and resistant to solvents. Thermosets are also almost impossible to recycle [17, p. 965], however, since it is difficult to destroy the crosslinks without destroying the bonds within the polymers. They moreover have no mechanism to recover their mechanical strength after sustaining damage. On the other hand we have *thermoplastics*, which consist of polymers that are bound by relatively weak, thermoreversible intermolecular forces [18], such as van der Waals forces. These materials become soft when heated, and can therefore be molded and recycled more easily [19]. Regrettably, they also tend to be less resilient [20, p. 45]. Consequently, neither traditional thermoset nor thermoplastics materials are well-suited to self-healing.



Figure 1.1: The European waste hierarchy, showing waste management strategies sorted by preference from top to bottom. Taken from the EU [Waste Framework Directive](#).

Fortunately, recent years have seen the development of Covalent Adaptive Networks (CANs), a new family of polymer materials, which are bound together by strong forces and nevertheless are able to alter their topology [21], and therefore ‘CAN’ intrinsically self-heal [22]. In particular, Leibler et al. investigated polymer networks with strong yet exchangeable covalent crosslinks in 2011. They called these materials *vitrimers*¹ for their glasslike properties [17]. The rate at which bonds are exchanged increases as the temperature goes up, and, as a consequence, vitrimers tend to be hard like thermosets at low temperatures, but malleable like thermoplastics at high temperatures. Vitrimers moreover are resistant to solvents [17, p. 966] and creep [10] like thermosets. In short, vitrimers promise to combine the best properties of both thermosets and thermoplastics [23, p. 72].

The dynamic nature of vitrimers moreover makes them capable of self-healing: in 2019, Ciarella and Ellenbroek found in a simulation study that a vitrimer network consisting of star polymers can self-heal while displaying thermoset-like mechanical properties on timescales around a thousand times as large as the typical healing time [1]. Star polymer vitrimers are very tunable: by varying the distribution of reactive groups at the ends of the stars, we can exercise great influence on the stress relaxation [24], for instance. Their usability is limited, however, by the similarity of the mechanisms that effect the self-healing and those that bring about the stress relaxation: to heal, the network must be able to rearrange itself around the damage; to relax, the stars must be able to move. One must therefore choose between autonomous healing and long-term stability [1].

This dilemma naturally raises the following question:

How can we design a vitrimer network that is capable of healing autonomously without giving up long-term rigidity?

We believe that the solution may be found by considering different building blocks than stars. In particular, we hypothesise that comb polymers with long backbones and many branches can give rise to a vitrimer network that heals autonomously on similar timescales as the network investigated in [1], while relaxing much more slowly. To support this hypothesis, we consider the following arguments. We can create a network made of combs with a similar density of reactive groups, which should heal at a comparable rate, since this density appears to determine how fast the network can recover around damage [1]. Additionally, the backbones of the combs in this network will become entangled, so that their motion is limited; the time it takes for them to become unentangled and relax will scale with the length of the backbone [25, Eq. (9.8)], giving us a way to prolong the stability of the network.

In this project we will test the validity of our hypothesis by developing simple theoretical self-healing models and simulating damaged vitrimer networks. We start off in Chapter 2 by discussing the bond exchange mechanism which defines vitrimers. We additionally go over the statistical theory of lifetime distributions and the basics of linear viscoelasticity. Thereafter, in Chapter 3, we summarise how we simulate vitrimer networks. We present our results in Chapter 4; this chapter is divided into three sections. In Section 4.1, we create a model to predict the typical swap times in the network depending on the shape of the building blocks. We subsequently perform simulations to find the entire swap time distributions and verify our mathematical model. Next, in Section 4.2, we derive and solve rate equations that describe the evolution of the number of bonds across damage in the network, making use of the typical swap times found in Section 4.1. We can then predict the mechanical response of the material as a function of healing time using the theory of linear viscoelasticity; we then validate this model with simulations. This will allow us to compare the healing time of a vitrimer network made of combs with the star polymer based one considered in [1], testing the first part of our hypothesis. Finally, we attempt in simulations to measure the time it takes for a comb polymer vitrimer network to lose its rigidity, which would allow us to test the second part of our hypothesis. We conclude in Chapter 5 by summarising our findings and conclusions and highlighting some interesting avenues for future research.

¹Derived from ‘vitrum’, Latin for glass.

Chapter 2

Theory

In this chapter, we will review some polymer theory, to serve as a basis for our mathematical model in Chapter 4. We will start with an introduction to bond swaps, the defining feature of vitrimers. We will also go over some tools from the theory of sequential and nonparametric statistics with which we may investigate and characterise the distribution of the swap times observed in simulations. Thereafter, we will review the basics of linear viscoelasticity, since we will be using concepts from that theory to quantify the mechanical stability of our materials. Finally, we will briefly discuss the morphologies of the building blocks of our vitrimer networks.

2.1 Bond swaps

Vitrimers are defined by the interesting behaviour of their crosslinks. For a concrete vitrimeric example, Leibler et al. in their seminal paper [17] considered carboxy ($-\text{C}(=\text{O})\text{OH}$) and hydroxy ($-\text{OH}$) moieties, which bond to form an ester group ($-\text{C}(=\text{O})\text{O}-$), with a water molecule as by-product. It turns out that such an ester bond is exchangeable; the corresponding bond swap, – called transesterification [17] – is depicted in Figure 2.1.

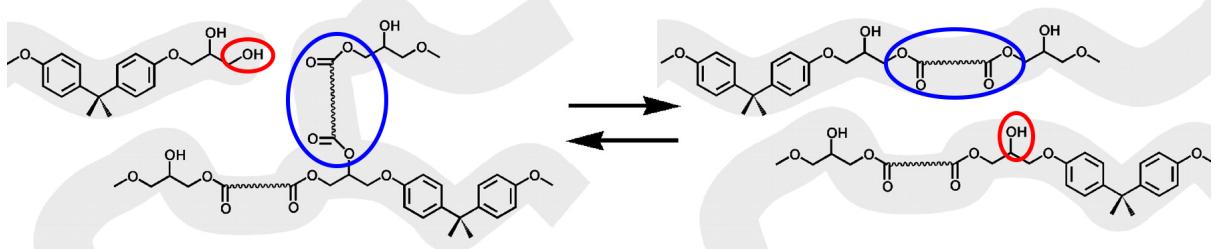


Figure 2.1: Transesterification, an example of a bond swap mechanism, depicted schematically. The target ester bond and the swapping alkoxy moieties are circled in blue and red, respectively. Adapted from [17].

Other chemistries give rise to vitrimers too; we will hence consider generic vitrimer networks with two distinct types of reactive moieties, which we will call (type-)A₁ and (type-)A₂. By construction, A₁ can form a bond with A₂, but A₁ cannot form a bond with A₁ and A₂ cannot form a bond with A₂.

In Figure 2.2, we have schematically depicted one of our generic bond swaps, with the red spheres representing A₁ moieties and the blue spheres representing A₂ moieties.¹ We start (a) with a bond between moiety 1 and 2, of type-A₂ and -A₁, respectively. Subsequently (b), moiety 3, of type-A₁, joins in, forming an intermediate state. This is called the *association* of the moieties [17]. Finally (c), moiety 2 separates from the complex, leaving behind a new bond.

There is a notable asymmetry in the transesterification reaction: the associating group must be an hydroxy moiety [27, Eq. (2)]. While our vitrimer networks are more general, they will also be asymmetric in this sense: the associating moiety must be of type-A₁.

¹We keep this convention throughout the figures in this report.

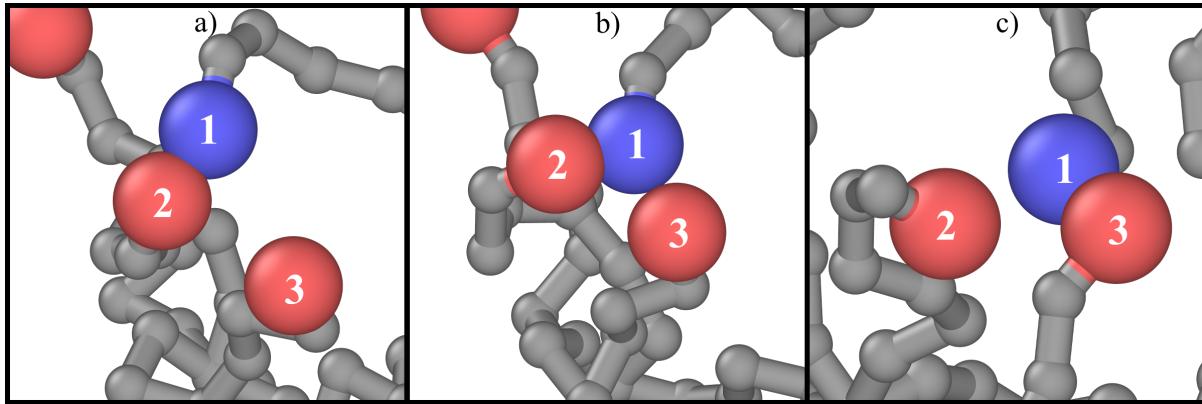


Figure 2.2: Illustration of a generic bond swap, where the blue sphere represents a A_2 moiety and the red spheres represent A_1 moieties. Generated using OVITO [26].

The exchangeable bonds are very strong: the chance that a crosslink will break due to thermal fluctuations is assumed to be negligible [28, p. 3]. In particular, this means that the number of crosslinks cannot decrease. If a given vitrimer network contains $N_{A_1}, N_{A_2} \in \mathbb{N}$ moieties of type- A_1 and - A_2 , the maximum number of bonds would be $\min\{N_{A_1}, N_{A_2}\}$. It follows that we must have more moieties of type- A_1 than type- A_2 for the network to be dynamic. We therefore assume that $N_{A_1} > N_{A_2}$.

2.1.1 Empirical Lifetime Distributions

Since vitrimers derive their unique properties from the bond swap mechanism, it makes sense to ask how this mechanism behaves statistically. A particularly interesting quantity to investigate for this is the *swap time* T_{sw} , which we will discuss in greater detail in Section 4.1. Since the swap times are determined by the complex interactions between polymers involving innumerable particles on a microscopic scale, a deterministic model for the swap times is not feasible, and consequently it makes sense to treat them as being random. Then, swap times are typical examples of lifetime data, since they are nonnegative, real random numbers [29, Def. 1.1], and for such data it is natural to look at the so-called *survival function* [29, Eq. (1.2)]

$$S(t) := \mathbb{P}(T \geq t), \quad (2.1.1)$$

which may be interpreted as the probability that some random timer T ‘survives’ until at least time t .

It is possible to estimate the survival function using observations. In practice, however, data is often imperfect due to censoring, so that a naive estimator would be biased. Unfortunately, this will also be the case for the swap times that we will measure in Section 4.1. We will consequently now introduce the necessary machinery to deal with censoring. Censoring comes in many shapes and sizes:

- Left censoring: when we know only an upper bound for the value of a random variable;
- Right censoring: when we know only a lower bound for the value of a random variable;
- Interval censoring: when we know a lower and upper bound for the value, but not the precise value, of a random variable.

For us, only right and interval censoring are relevant. Interval censoring is rather difficult to deal with analytically²; we can mitigate these issues by choosing our intervals sufficiently small, and simply ignoring the interval censoring. In other words:

Assumption 2.1.1. Observations are either right censored or exact, but not left or interval censored.

²Methods exist (e.g. [30]), but they scale poorly with the number of observations. We will be dealing with much more data (roughly 10^5 points) than in typical medical research, making these methods impractical.

Observations (censored or not) will be denoted T_i , with corresponding censoring indicator³

$$\Delta_i := 1 - \mathbf{1}\{T_i \text{ censored}\}. \quad (2.1.2)$$

With any censored observation we do not know what the ‘true’ value is; some information has been lost. However, we can effectively maximise the amount of gathered information by including the censoring indicator Δ_i in our estimator, since we know that the ‘true’ value is at least equal to the observed value if that observation was right censored. This is precisely the idea behind the *Kaplan-Meier (KM)* estimator.

Definition 2.1.2 (Kaplan-Meier Estimator). Let $\{T_i\}_{i=1}^{N_{\text{obs}}}$ be independent and identically distributed (i.i.d.) observed lifetime data, with corresponding censoring indicators $\{\Delta_i\}_{i=1}^{N_{\text{obs}}}$, where the censoring times are independent of the true lifetimes. Then, define $\{T'_i\}_{i=1}^n$ to be the unique ordered observed times, i.e. $\{T_i\}_{i=1}^{N_{\text{obs}}} = \{T'_i\}_{i=1}^n$ and $T'_1 < \dots < T'_n$. Then, define $\{d_i\}_{i=1}^n$ and $\{r_i\}_{i=1}^n$ by

$$d_i = \sum_{k=1}^{N_{\text{obs}}} \Delta_k \mathbf{1}\{T_k = T'_i\}, \text{ and } r_i = \sum_{k=1}^{N_{\text{obs}}} \mathbf{1}\{T_k \geq T'_i\},$$

respectively. The Kaplan-Meier (KM) estimate of the survival function of the distribution underlying the lifetime data is given by [31, Eq. (2b)]

$$\hat{S}_{\text{KM}}(t) = \prod_{i:T'_i \leq t} \left(1 - \frac{d_i}{r_i}\right). \quad (2.1.3)$$

E.L. Kaplan developed the KM estimator particularly for use in medical studies [31], but it may be applied to any lifetime data. Note that d_i is equal to the total number of uncensored observations at time T'_i , while r_i is the number of timers that last until at least T'_i . Consequently, if there is a censored observation at T'_n , $\hat{S}_{\text{KM}}(t) > 0$ for all $t \geq 0$. We can also construct approximate confidence intervals for the KM estimator, since we can estimate its variance:

Definition 2.1.3 (Greenwood’s Formula). Let \hat{S}_{KM} be a KM estimate, and $\{d_i\}_{i=1}^n$ and $\{r_i\}_{i=1}^n$ as in Definition 2.1.2. Then, we may estimate the variance of \hat{S}_{KM} using Greenwood’s formula [31, Eq. (6d)] [32, Eq. (4)]:

$$\hat{\mathbb{V}}(\hat{S}_{\text{KM}}(t)) := \hat{S}_{\text{KM}}(t)^2 \sum_{i:T'_i \leq t} \frac{d_i}{r_i(r_i - d_i)}. \quad (2.1.4)$$

Using Greenwood’s formula, we can estimate the standard deviation on the KM estimate at each time. If this standard deviation is small compared to the value of the survival function, we can be confident about our KM estimate⁴; if the standard deviation is large, we need to be careful inferring distributional properties from the empirical survival curves.

In theory, all information about a lifetime distribution can be extracted from the corresponding survival function. It is, however, quite difficult to read *local* information about the distribution (e.g. where the mode is) from a survival curve. Such local information could be helpful when trying to identify relevant physical processes that contribute to the lifetime. For this reason, we will also look at the Probability Density Function (PDF), defined as the continuous function f satisfying [33, Def. 4.1(1)]

$$\mathbb{P}(T \leq t) =: \int_0^t f(u) du, t \geq 0. \quad (2.1.5)$$

It can easily be seen that⁵

$$f(t) = -\frac{d}{dt} S(t).$$

³Note the confusing convention that the censoring indicator is unity when the observation is uncensored.

⁴assuming the bias is small too, which should be the case if the amount of censoring is limited.

⁵e.g. by the Fundamental Theorem of Calculus [34, Theorem 6.4.4].

The PDF $f(t)$ gives a measure of the likelihood of a lifetime being around t , in the sense that

$$\mathbb{P}(T \in [t - \Delta t, t + \Delta t]) \approx f(t) \cdot 2\Delta t.$$

Consequently, a peak in the PDF will indicate that we will commonly observe a lifetime value around that peak; if a PDF displays multiple peaks, this suggests that there are multiple distinct processes – each with their own typical time scales – which together determine the lifetime.

Unfortunately, we cannot estimate the true PDF by taking the derivative of the empirical survival function since it is discontinuous at every observation but constant in between observations. We can deal with this issue by mollifying the empirical survival function. To that end, let K be a kernel, so that $\int_{-\infty}^{\infty} K(t)dt = 1$, and observe that

$$\begin{aligned} f(t) &\stackrel{(I)}{\approx} (K * f)(t) := \int_{-\infty}^{\infty} K(t-x) \underbrace{f(x)dx}_{-\text{d}S(x)} \stackrel{(II)}{\approx} - \int_{-\infty}^{\infty} K(t-x) d\hat{S}_{\text{KM}}(x) \\ &= - \int_{-\infty}^{\infty} K(t-x) \frac{d\hat{S}_{\text{KM}}}{dx}(x) dx = \int_{-\infty}^{\infty} K(t-x) \sum_{i=1}^n \delta(x - T'_i) \Delta\hat{S}_{\text{KM}}(x) dx \quad (2.1.6) \\ &= \sum_{i=1}^n \int_{-\infty}^{\infty} K(t-x) \delta(x - T'_i) \Delta\hat{S}_{\text{KM}}(x) dx = \sum_{i=1}^n K(t - T'_i) \Delta\hat{S}_{\text{KM}}(T'_i). \end{aligned}$$

Using this idea, we define the following density estimator, which has frequently been used to investigate the distribution of censored lifetime data (e.g. [35, 36, 37]):

Definition 2.1.4 (Kaplan-Meier Kernel Density Estimator). Let $\{T'_i\}_{i=1}^n$ be the unique ordered observed times of some lifetime data and \hat{S}_{KM} the corresponding KM estimate of the survival function. Next, define

$$\Delta\hat{S}_{\text{KM}}(t) = \lim_{x \uparrow t} \hat{S}_{\text{KM}}(x) - \lim_{x \downarrow t} \hat{S}_{\text{KM}}(t),$$

which gives the size of the jump discontinuities of \hat{S}_{KM} . Finally, let K be a kernel, so that $\int_{-\infty}^{\infty} K(t)dt$. Then, we define the Kernel Density Estimator (KDE) by

$$\hat{f}_{\text{KM}}(t) := \sum_{i=1}^n K(t - T'_i) \Delta\hat{S}_{\text{KM}}(T'_i), \quad (2.1.7)$$

for $t \in \mathbb{R}$.

Notably, if we are dealing with uncensored data so that we may replace the \hat{S}_{KM} estimate of the survival function by the standard naive estimate, Equation (2.1.7) reduces to the commonly used Rosenblatt-Parzen KDE [38, 39]

$$\hat{f}(t) := \sum_{i=1}^{N_{\text{obs}}} K(t - T_i).$$

Definition 2.1.4 allows for quite general kernels K , but the choice of the kernel is in fact rather important. It is evident from Equation (2.1.7) that the estimate \hat{f}_{KM} inherits its smoothness from K . Hence, if there is good reason to believe that the true density is not analytic, it makes sense not to use an analytic kernel. Additionally, convolving with the kernel will broaden the support, as

$$\text{supp}(K * f) \subset \overline{\text{supp } K + \text{supp } f},$$

so that $K * f$ will typically *not* be the PDF of lifetime data. It might therefore be advantageous to use a kernel with small, or even strictly positive [37], support. We will, however, use Gaussian kernels which are analytic and have unbounded support⁶, because this will allow us to make use of the convolution tools implemented in the popular package SciPy [40].

Finally, we need to take care when choosing how wide the kernel is, as the two approximations in Equation (2.1.6) will vary differently with the width of the kernel. In essence, we can decrease the

⁶In practice we will convolve with discretised Gaussian kernels that have compact support and certainly are not analytic.

approximation error (I) – manifesting itself as bias – by using a narrow kernel, since then $K * f$ will obviously be closer to f , while we can decrease the estimation error (II) – corresponding to the variance – by using a broad kernel, since then the estimator will be less sensitive to small perturbations in the observations. There will consequently be some optimum kernel width [41]. It is possible to select a good kernel width in practice using Leave-One-Out Cross-Validation (LOOCV) to minimise an estimate of the total error [41]; we will forego this and choose a kernel width by eye.

The number of surviving timers decreases over time, which can make the PDF somewhat difficult to interpret. Consider for instance a situation in which timers run out at a uniform rate 1 until some time t^* , i.e. in every interval in $[0, t^*]$ of fixed width the number of events is roughly constant, after which the timers run out at twice that rate. Evidently, there is some interesting additional process going on beyond t^* . However, if t^* is large, there will be few timers left anyway, so that the density of events will be small. It therefore makes sense to also look at the instantaneous rate at which timers run out, i.e.

$$\lambda(t) := \lim_{h \downarrow 0} \frac{1}{h} \mathbb{P}(T \leq t + h \mid T \geq t) = \lim_{h \downarrow 0} \frac{\mathbb{P}(t \leq T \leq t + h)}{h \mathbb{P}(T \geq t)} = \frac{f(t)}{S(t)}. \quad (2.1.8)$$

λ is called the *hazard function* [29, Def. 1.5]. Figure 2.3 shows the (a) survival functions, (b) PDFs, and (c) hazard functions for our example for a small and a large t^* . For the small t^* , we can easily spot the change in the PDF; for the large t^* this is not so easy. Another nice property of the hazard function

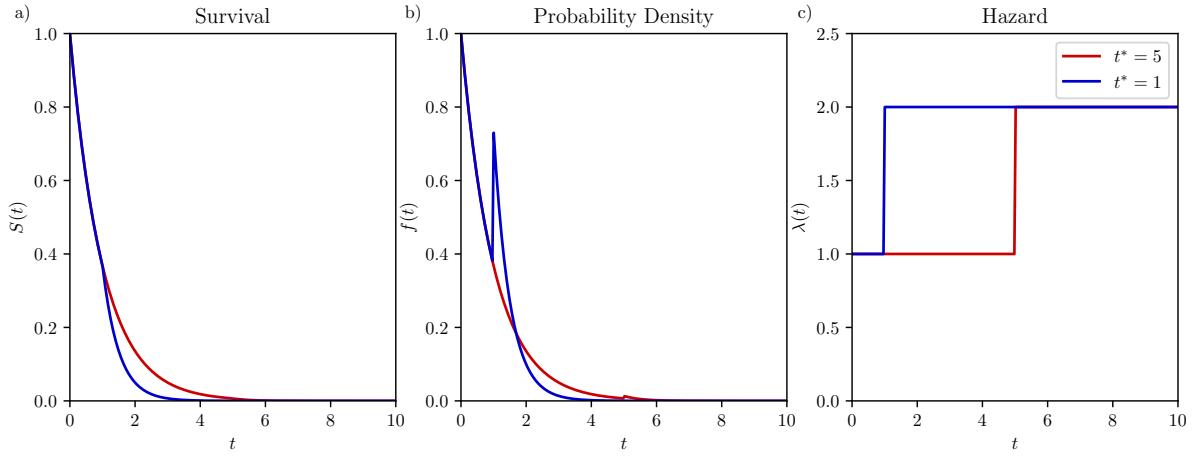


Figure 2.3: Example showing how the hazard function can make it easier to identify when the processes underlying the life of a timer change at long timescales.

is that it immediately reveals whether the lifetime is exponentially distributed, since the exponential distribution has constant hazard: if $T \sim \text{Exp}(\tau)$, then

$$\lambda(t) = \frac{\frac{1}{\tau} \exp(-\frac{t}{\tau})}{\exp(-\frac{t}{\tau})} = \frac{1}{\tau}. \quad (2.1.9)$$

The exponential distribution is the prototypical lifetime distribution, with numerous desirable attributes. For instance, the exponential distribution is the only continuous lifetime distribution that is memoryless, see Appendix A.3, and the counting process with exponential interarrival times is the well-known Poisson process [42, Thm. 6.8(10)]. It is therefore quite valuable to be able to assess to what degree it is reasonable to model the lifetimes with an exponential distribution. We can also easily tell from the hazard function whether the lifetime data has a Weibull distribution, which is a common extension of the exponential distribution. The Weibull(α, β) distribution is characterised by the survival function [33, Def. 4.4(9)]

$$S(t) := \exp\left(-\left(\frac{t}{\alpha}\right)^\beta\right), \text{ so that } f(t) = -\frac{d}{dt}S(t) = \frac{\beta}{\alpha}\left(\frac{t}{\alpha}\right)^{\beta-1} \exp\left(-\left(\frac{t}{\alpha}\right)^\beta\right).$$

Note that $\text{Weibull}(\tau, 1) = \text{Exp}(\tau)$, so that the Weibull distribution indeed generalises the exponential distribution.

$$\lambda(t) = \frac{\frac{\beta}{\alpha}\left(\frac{t}{\alpha}\right)^{\beta-1} \exp\left(-\left(\frac{t}{\alpha}\right)^\beta\right)}{\exp\left(-\left(\frac{t}{\alpha}\right)^\beta\right)} = \frac{\beta}{\alpha}\left(\frac{t}{\alpha}\right)^{\beta-1}. \quad (2.1.10)$$

As we would expect, we recover Equation (2.1.9)⁷ when $\beta = 1$.

We could compute an estimate of the hazard function by dividing an estimate for the PDF by an estimate for the survival function. Such an estimate will perform poorly, however, since the KM estimate is not smooth. To get a better estimator, we first rewrite the hazard function in terms of the survival function only:

Lemma 2.1.5 (Hazard as a Function of Survival). *Let T be a lifetime random variable with survival function S and hazard function λ . Then for all $t \in \text{supp } S$ it holds that*

$$\lambda(t) = -\frac{d}{dt} \log(S(t)). \quad (2.1.11)$$

Proof. Let f be the PDF of T , so that by Equation (2.1.8) on $\text{supp } S$ it holds that

$$\lambda(t) = \frac{f(t)}{S(t)} = -\frac{S'(t)}{S(t)}.$$

By applying the chain rule, we can find that

$$-\frac{d}{dt} \log(S(t)) = -\frac{1}{S(t)} \frac{d}{dt} S(t) = -\frac{S'(t)}{S(t)},$$

proving the result. \square

Note that $f(t) = -\frac{d}{dt} S(t)$: this suggests that we can reuse the idea for the KDE, simply replacing $S(t)$ by $\log(S(t))$.

Definition 2.1.6 (Kaplan-Meier Kernel Hazard Estimator). Let $\{T'_i\}_{i=1}^n$ be the unique ordered observed times of some lifetime data and \hat{S}_{KM} the corresponding KM estimate of the survival function. Next, define

$$\Delta \log(\hat{S}_{\text{KM}}(t)) = \lim_{x \uparrow t} \log(\hat{S}_{\text{KM}}(x)) - \lim_{x \downarrow t} \log(\hat{S}_{\text{KM}}(t)),$$

which gives the size of the jump discontinuities of $\log(\hat{S}_{\text{KM}})$. Finally, let K be a kernel, so that $\int_{-\infty}^{\infty} K(t) dt$. Then, we define the Kernel Hazard Estimator (KHE) by

$$\hat{\lambda}_{\text{KM}}(t) := \sum_{i=1}^n K(t - T'_i) \Delta \log(\hat{S}_{\text{KM}}(T'_i)), \quad (2.1.12)$$

for $t \in \mathbb{R}$.

Typically, a slightly different KHE is used, which makes use of the so-called Nelson-Aalen (NA) estimator of the cumulative hazard function (e.g. [35]), which can be derived from the KM with a first order Taylor series approximation.⁸ For the sake of simplicity, we only make use of the KM estimator in this project; fortunately, the two estimators are asymptotically equivalent [43].⁹ Notably, the previously discussed issues of the KDE carry over to the KHE; we will again use Gaussian kernels with hand-picked width for the KHE.

In short, local information about distributions can be very valuable, and we have ways to estimate it from data. Unfortunately, local information is intrinsically much more difficult to estimate than global information. Consider, for instance, that the value of \hat{S}_{KM} at any point is depends on all data points, while the value of \hat{f}_{KM} or $\hat{\lambda}_{\text{KM}}$ at a point is determined by only those data points that are sufficiently close. The hazard function is particularly difficult to estimate in the right tail, because the logarithm

⁷now with mean α .

⁸ $-\log(\hat{S}_{\text{KM}}(t)) = -\sum_{i:T'_i \leq t} \log\left(1 - \frac{d_i}{r_i}\right) \approx \sum_{i:T'_i \leq t} \frac{d_i}{r_i} =: \hat{\Lambda}_{\text{NA}}(t)$ [43, Eq. (2)].

⁹The NA estimator is more commonly used as it is less sensitive to noise in the tail of the distribution [43]; since we have a large amount of data, this is not problematic in our case.

explodes as its argument tends to zero; $\hat{\lambda}_{\text{KM}}$ will therefore be sensitive to small variations on long timescales. We therefore must take care when drawing conclusions from \hat{f}_{KM} and $\hat{\lambda}_{\text{KM}}$.

While having (an estimate of) the entire survival, density, and hazard function is nice, it would be convenient to be able to boil it down to a single number such as the mean. It turns out to be possible to compute the mean from the survival function: if we do this with our KM estimate, we can get an estimate for the expected swap time that takes censoring into account.

Theorem 2.1.7 (Mean of Lifetime Data). *Let $T \geq 0$ continuous with density f and survival function S , with $\mathbb{E}[T] < \infty$. Then,*

$$\mathbb{E}[T] = \int_0^\infty S(t)dt. \quad (2.1.13)$$

Proof. Note that $\frac{dS}{dt}(t) = -f(t)$ for $t \geq 0$. We can then use integration by parts to see that

$$\int_0^\infty S(t)dt = [tS(t)]_0^\infty - \int_0^\infty t dS(t) = [tS(t)]_0^\infty + \underbrace{\int_0^\infty tf(t)dt}_{=\mathbb{E}[T]}.$$

Clearly $tS(t)|_{t=0} = 0 \cdot 1 = 0$. Now, consider that for all $t \geq 0$ it holds that

$$0 \leq tS(t) = t \int_t^\infty f(u)du = \int_t^\infty tf(u)du \leq \int_t^\infty uf(u)du.$$

Note that $\int_0^\infty uf(u)du = \mathbb{E}[T] < \infty$; since this is a converging improper Riemann integral, we know by Lemma A.1.1 that the tail tends to zero. It follows, by applying the Squeeze Theorem [44, Thm. 2.2.6], that

$$\lim_{t \rightarrow \infty} tS(t) = 0,$$

as

$$0 \leq tS(t) \leq \int_t^\infty uf(u)du, \text{ and } \lim_{t \rightarrow \infty} 0 = 0 = \lim_{t \rightarrow \infty} \int_t^\infty uf(u)du.$$

We may then conclude that

$$\int_0^\infty S(t)dt = [tS(t)]_0^\infty + \mathbb{E}[T] = 0 - 0 + \mathbb{E}[T] = \mathbb{E}[T],$$

as required. \square

Hence, it seemingly makes sense to estimate the mean of the lifetime distribution using the KM estimate of the survival function.

Definition 2.1.8 (Kaplan-Meier Estimator of Mean). Let \hat{S}_{KM} be the KM estimate of the survival function of some lifetime data. Then, we define the KM estimate of the mean as

$$\hat{\mu}_{\text{KM}} := \int_0^\infty \hat{S}_{\text{KM}}(t)dt. \quad (2.1.14)$$

We must be careful, however: as noted before, the KM estimate fails to go to zero when the last observation is censored, so that the area under the empirical survival curve will be infinite. To remedy this, we can truncate the survival function, so that we may compare the expected lifetime up to some given time [45].

Definition 2.1.9 (Truncated Kaplan-Meier Estimator and Restricted Mean Lifetime). Let \hat{S}_{KM} be the KM estimate of the survival function of some lifetime data. Furthermore, let $t^* > 0$. Then, we define the (t^* -) truncated KM estimate of the survival function as

$$\hat{S}_{\text{KM,truncated}}(t; t^*) = \begin{cases} \hat{S}_{\text{KM}}(t), & t \leq t^*, \\ 0, & t > t^*, \end{cases} \quad (2.1.15)$$

and we define the (t^* -) restricted mean lifetime as [45]

$$\hat{\mu}_{\text{KM},\text{restricted}}(t^*) = \int_0^\infty \hat{S}_{\text{KM},\text{truncated}}(t, t^*) dt = \int_0^{t^*} \hat{S}_{\text{KM}}(t) dt. \quad (2.1.16)$$

Since $\hat{S}_{\text{KM},\text{truncated}}(\cdot; t^*)$ is a bounded function with bounded support $\text{supp } \hat{S}_{\text{KM},\text{truncated}}(\cdot; t^*) \subset [0, t^*]$, it holds that $\hat{\mu}_{\text{KM},\text{restricted}}(t^*) \leq t^* < \infty$ for any finite t^* . Moreover,

$$\lim_{t^* \rightarrow \infty} \hat{\mu}_{\text{KM},\text{restricted}}(t^*) = \hat{\mu}_{\text{KM}}.$$

It is important that the point of truncation t^* is chosen a priori [46]; in our case it makes sense to set t^* to be the length of a simulation. Fortunately, since we gather our data from simulations,¹⁰ we can gather data over such long timescales that truncating \hat{S}_{KM} is not problematic. For legibility, we will suppress the fact that we truncate the KM estimator in our notation from now on.

We have implemented all of the statistical tools in this subsection in `analysers.py` in [Hoomd Polymer Tools](#).

2.2 Linear Viscoelasticity

We can assess the quality of self-healing over time by investigating how the rigidity of the vitrimer material improves as it is allowed to heal for longer. To quantify the rigidity, we look to the theory of linear viscoelasticity. By measuring how the stress in the material relaxes after a step strain, it is i.a. possible to assess to what degree the material is solid or liquid [47, Fig. 7.22]. To apply a shear *strain* γ to a block of material with height h , we must deform it with a perpendicular displacement $\Delta x = \gamma h$, so that [47, Eq. (7.97)]

$$\gamma := \frac{h}{\Delta x}. \quad (2.2.1)$$

Upon applying this strain, the stress σ in the material will jump up, after which it will start to decay over time. By dividing the stress by the strain, we obtain the so-called *stress relaxation* [47, Eq. (7.98)] or *shear modulus* [48, Eq. (6.9)]:

$$G(t) := \frac{\sigma(t)}{\gamma}. \quad (2.2.2)$$

This is a sensible thing to do, since for sufficiently small strains the stress will depend linearly on the strain, and so G will not depend on γ [47, p. 282]. As the material relaxes, G will decay towards the equilibrium shear modulus [47, Eq. (7.105)]:

$$G_\infty = \lim_{t \rightarrow \infty} G(t). \quad (2.2.3)$$

Using a so-called Maxwell model, which represents a viscoelastic material as a damper and spring – for the viscous and elastic response, respectively – in series, we can moreover predict that this decay will be exponential for viscoelastic liquids [47, Eq. (7.111)]. The equilibrium shear modulus is a measure for the rigidity of the material, and we can use it to classify our material: if $G_\infty = 0$, we say it is a (viscoelastic) liquid, while we say it is a (viscoelastic) solid if $G_\infty > 0$. Of course, there are many other relevant mechanical properties beside rigidity, such as solvent resistance [49]; we have limited our scope to just the equilibrium shear modulus since this will allow us to compare the self-healing of a comb polymer vitrimer network with that of the star polymer network investigated by Ciarella and Ellenbroek [1].

2.2.1 Affine Network Model

The elastic properties of a polymer material will depend in part on what its network looks like on a microscopic scale. Figure 2.4 schematically depicts such a network. The black dots represent nodes in the network. Each node has a number of dangling arms with reactive groups at their ends, represented by red and blue dots. Red and blue dots can bond together, forming so-called ‘crosslinks’.

¹⁰in contrast with data gathered from clinical trials, for which the tools discussed in this subsection were developed.

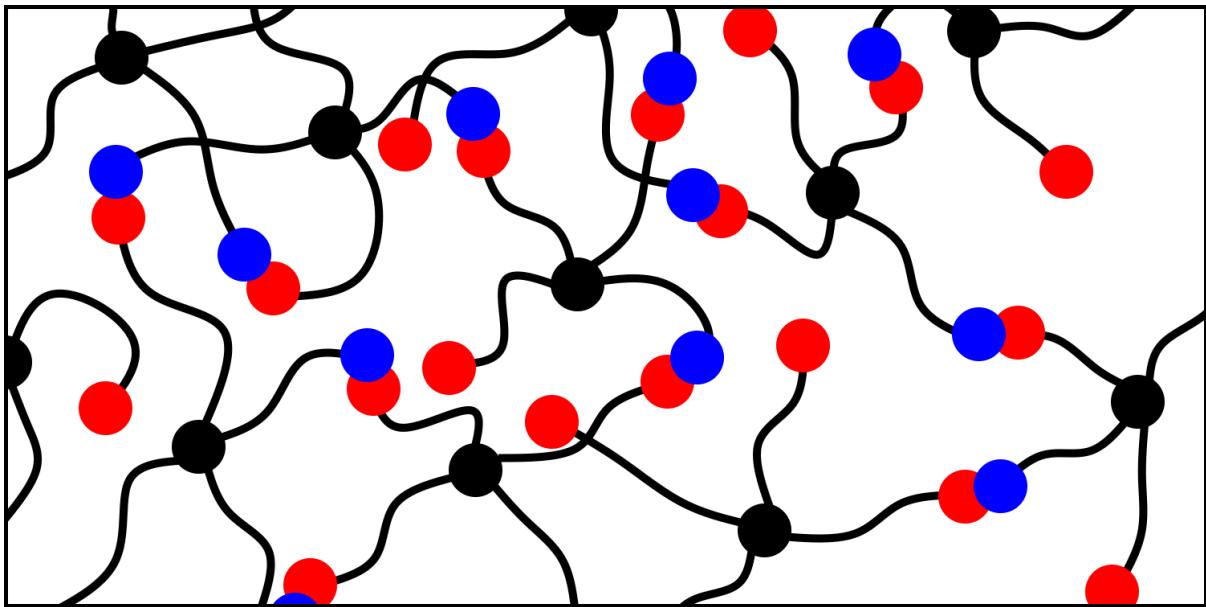


Figure 2.4: Schematic depiction of a polymer network. Black dots represent nodes in the network, blue and red dots are reactive particles that can form crosslinks.

Many simple models have been developed to describe how polymer networks will behave when stressed. We will apply the affine network model, which means that we assume that deformations of the macroscopic material are achieved by equal and uniform relative deformations on the microscopic scale. In other words, if we apply an affine transformation¹¹ to the bulk, the position of each node undergoes the same affine transformation. In the affine network model it is possible to predict the shear modulus [47, Sec. 7.2]: the shear modulus will then be given by [47, Eq. (7.31)]

$$G = \nu k_B T \propto \nu, \quad (2.2.4)$$

where ν is the number density of the chains connecting nodes in the polymer network, k_B is the Boltzmann constant, and T is the temperature. Since we can choose the building blocks of the vitrimer network, we can control ν . We will be able to use Equation (2.2.4) to relate the number of bonds that have formed across damage in our network to the shear modulus, and thereby to the healing quality.

There are some additional implicit assumptions in the affine network model. For one, the forces in the network are derived by differentiating the free energy [47, Eq. (7.28)]. The entropy term herein is valid for an ideal chain that is sufficiently long that its end-to-end length may be well-modelled as being normally distributed [50, Eq. (2.92)]. The chains in the network we will be working with are typically only about eight particles long, however, so this assumption may not be reasonable. In Appendix A.2 we therefore computed the exact density of the chain length assuming that the length of each bond is uniformly distributed Figure 2.5; even for such small chains the normal density does not look to bad. Finally, the network is assumed to be entanglement free, whereas we will be investigating comb polymer vitrimer networks with long backbones precisely because these backbones can become entangled.

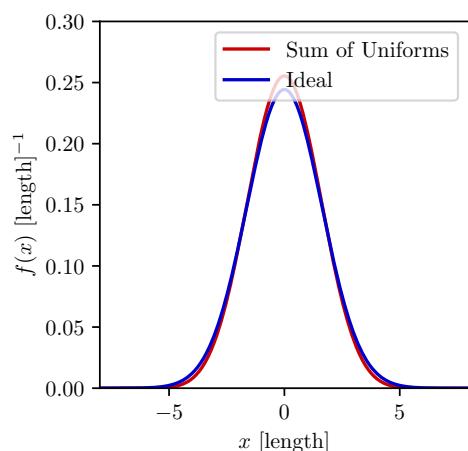


Figure 2.5: Eight particle chain length density for uniformly distributed links compared to corresponding ideal chain length density.

¹¹i.e. $\mathbf{x} \mapsto \mathbf{Ax} + \mathbf{b}$ for some matrix \mathbf{A} and vector \mathbf{b} .

2.2.2 Stress Relaxation in Vitrimers

Stress in a material is relaxed through myriad processes, each occurring at different timescales. For instance, in crosslinked networks stress can be relaxed by chain rearrangements [23, p. 115]. Vitrimers have an additional way to relax stress: bond exchanges. Figure 2.6 [24, Fig. 2] shows a typical example of how the shear modulus of a vitrimer network evolves over time after a step strain has been applied. Here, $\beta\Delta E$ is the energy associated with a swap: when $\beta\Delta E > 0$, exchanges are inhibited; in our simulations it always holds that $\beta\Delta E = 0$. When swaps are possible, we see two relaxations [24, Fig. 2]:

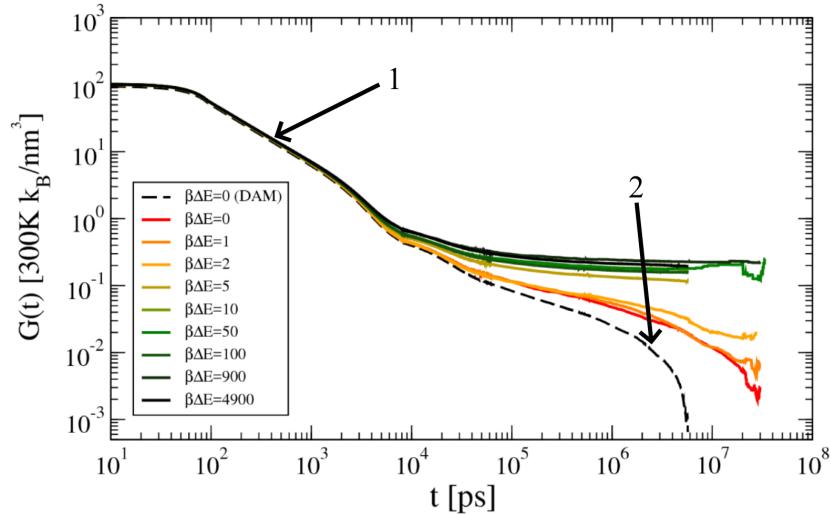


Figure 2.6: Typical example of evolution of stress relaxation modulus after a step strain has been applied. For $\beta\Delta E > 0$, exchanges are inhibited. Adapted from [24, Fig. 2].

1. For $t \in (10^2 \text{ ps}, 10^4 \text{ ps})$, we see the decay towards the elastic plateau, which we would also observe in a permanently crosslinked network;
2. For $t > 10^6 \text{ ps}$, we see the decay towards 0 due to the bond exchanges.

Hence, vitrimer networks are typically solid on short timescales and liquid on very long timescales [51, p. 17]. Notably, for small times the modulus does not depend on the exchange energy; for these timescales we can reasonably assume the network is fixed when probing the shear modulus.

The dashed curve in Figure 2.6 corresponds to Defect-Allowing Mixture (DAM) samples, while the solid curves correspond to Defect-Free Mixture (DFM) samples; their building blocks are shown in Figure 2.7. The mixtures have been designed such that the total fraction of A_1 moieties is the same in both types of mixtures, namely roughly 63 %. It is evident from Figure 2.6 that DAM samples become liquid on the order of ten times faster than DFM samples [24]. We are interested in designing a material that in

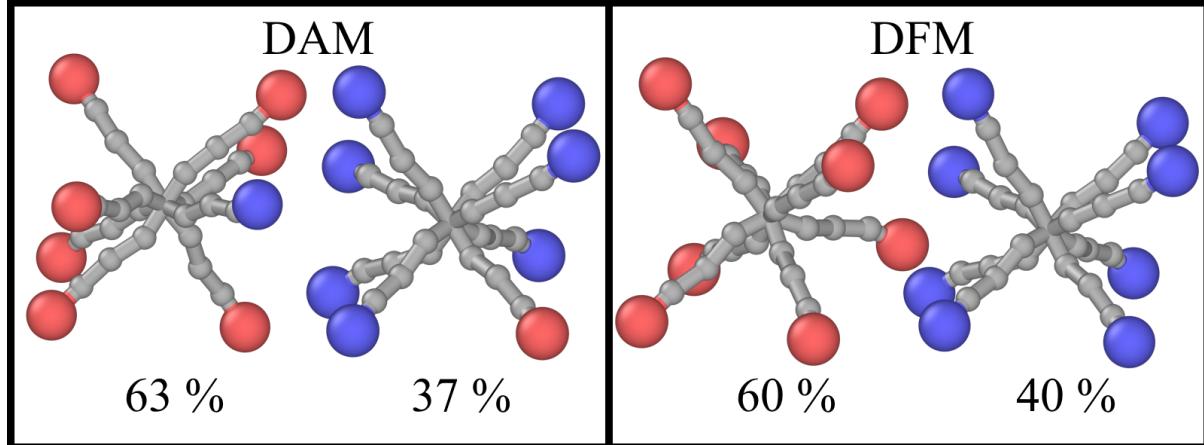


Figure 2.7: Comparison of the building blocks of DAM and DFM. Generated with OVITO [26]. addition to being capable of self-healing is also mechanically strong: the material should not become liquid (on practical timescales). In 2019, Ciarella and Ellenbroek observed that the relaxation of their

vitrimers based on star polymers takes roughly three orders of magnitudes longer than the self-healing [1]; ideally, we would make this gap larger. Despite this, in this report we have exclusively considered comb polymer equivalents of DAM samples, in order to remain consistent with Ciarella and Ellenbroek.

2.3 Polymer Morphology

It is almost self-evident that the morphology of the building blocks of the vitrimer network will be of importance to the mechanical properties of the network. In particular, we believe that comb polymers can improve the separation between the healing time and timescales on which the network behaves like a liquid compared to the previously studied [1] star polymers, since the backbones of the combs can become entangled and may thereby delay stress relaxation.

We are able to encode the shapes of these combs and stars using just a few parameters. Figure 2.8 shows a star polymer with eight arms, each consisting of four beads, and a comb polymer with sixteen arms, each consisting of four beads, separated by three beads on the backbone. From now on, we will denote the number of arms by N_{arms} , the number of beads in each arm by n_{arm} , and (for combs) the number of beads separating each arm by n_{sep} ; collectively, we will call them ‘shape parameters’.

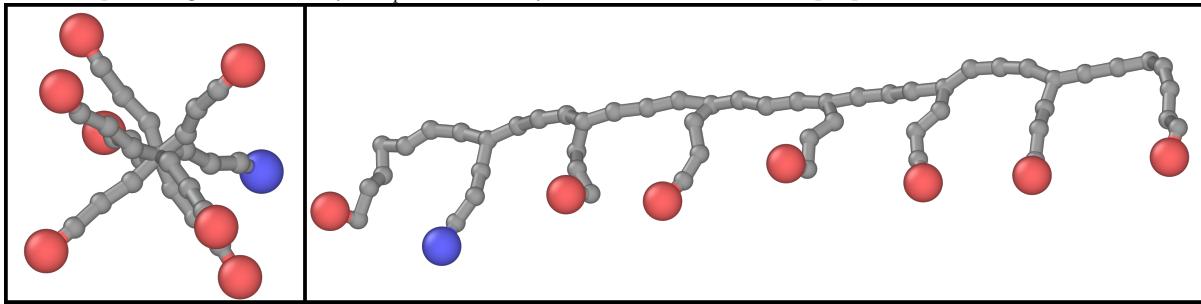


Figure 2.8: Star polymer with $N_{\text{arms}} = 8$ and $n_{\text{arm}} = 4$ and comb polymer with $N_{\text{arms}} = 16$, $n_{\text{arm}} = 4$, and $n_{\text{sep}} = 3$. Generated with OVITO [26].

We will work with stars with $N_{\text{arms}} = 8$ and $n_{\text{arm}} = 4$, so that we may compare our results to those of Ciarella and Ellenbroek [1], as well as combs with a various shape parameters.

Chapter 3

Simulation Protocol

In this chapter, we will discuss how we can model vitrimer networks using simulations. There are, in essence, two classes of approaches, with some overlap, to simulate molecular systems: Monte Carlo (MC) and Molecular Dynamics (MD). MC methods sample (indirectly) from the distribution of the system. Note that this implicitly assumes the system of interest to be in equilibrium, for else the distribution function would be ill-defined¹. This allows one to calculate estimates for ensemble averaged properties by employing the law of large numbers.

Conversely, MD methods integrate Newton's equations of motion. As a consequence, MD can be used to investigate out-of-equilibrium systems². In equilibrium, assuming the system is moreover ergodic, time averages and ensemble averages coincide, so that we may estimate ensemble averaged properties by measuring them over time and taking the sample average. At the intersection of these two techniques are hybrid MD-MC methods, which use MD to model the motion of particles in the system and use MC to model events such as bond swaps (e.g. [49, 54]). It would be beneficial if we could use a fully MD approach, however, for instance due to the various discontinuities introduced by the MC events [55]. Fortunately, such an approach turns out to be possible; the exact method will depend on the desired 'resolution' of the simulation.

We will now briefly go over the basics of simulation 'resolution'. On the one hand there are *atomistic* approaches, in which all the atoms, or at least all sufficiently small groups, are modelled. Furthermore, experimental data is often used to derive the interactions between these groups, and simulation outcomes may be compared with physical experimental results [56]. On the other hand we have *coarse-grained* approaches, which, as the name would suggest, are much more coarse: we model larger groups, say on the order of Kuhn segments [56], and the potentials are typically quite simple. These differences mean that an atomistic approach will allow one to investigate properties of polymer networks quantitatively, at the cost of being significantly more computationally expensive, whereas a coarse-grained approach will be cheaper but are unable to resolve the details of specific molecules. Fortunately, however, it is often still possible to examine generic properties using coarse-grained simulations both qualitatively and quantitatively [23, Sec. 3.1].

There have been atomistic MD investigations into vitrimers (e.g. [57, 58, 59]). We will, however, make use of coarse-grained MD simulations, since we are interested in the behaviour of generic vitrimers and would like to observe events that occur on vastly different timescales, such as the relaxation to the elastic plateau and to liquid of vitrimers (see Section 2.2).³

Sciortino developed a way to model vitrimers with exclusively MD [55]. This approach is very appealing, and has been used frequently since its conception in 2017 (e.g. [28, 60, 11]). We will therefore make use of it too.

¹There are MC methods to model out-of-equilibrium processes, but typical MC methods – such as the Metropolis algorithm [52] – do not model the actual dynamics of the system; there is therefore no a priori reason to believe that the approach to stationarity of such a MC method will be similar to the approach of equilibrium of an out-of-equilibrium system.

²with the appropriate choice of thermostat [53, p. 156].

³For reference, [58] simulated roughly three swap times, whereas we expect transition of the vitrimer network to liquid to certainly take on the order of one hundred swap times, but hopefully longer, see Section 4.3.

We will start in Section 3.1 by discussing the interaction potentials used in our MD simulations and how we choose their parameters. In Subsection 3.1.1, we will discuss how MD integrates Newton's equations to model molecular systems. We will then explain how we can generate a reasonably realistic initial state for our simulations and how we evolve these systems in Section 3.2. Finally, in Section 3.3, we will investigate some issues we encountered while performing our experiments with HOOMD-blue.

3.1 Potentials and Parameters

For our coarse-grained MD simulations, we use HOOMD-blue⁴, a free particle simulation tool with a Python interface, developed by the Glotzer group at the University of Michigan [61]. Ciarella and Ellenbroek have previously investigated the self-healing of star polymer based vitrimers [1]; since we want to be able to compare some results, we will use the same potentials and parameters as in [1].

HOOMD-blue defines its own, self-consistent set of units, with base units [energy], [length], and [mass]. From these base unit we can derive the dimensions of all the relevant quantities, such as time, velocity, and force, denoted [time], [velocity], and [force], respectively. Notably, we do not set the temperature directly; instead, we fix $k_B T = 1$ [energy]. Because we are interested in the *autonomous* self-healing of vitrimer materials, which by definition should not require an external trigger to activate, we keep this temperature constant.

As noted before, MD works by integrating Newton's equations. The interactions between particles, such as covalent bonds or van der Waals forces, are modelled using potentials.

We can model covalent bonds between particles with a harmonic potential:

$$V_{\text{harm}}(r_{ij}) = \frac{1}{2}k(r_{ij} - r_0)^2, \quad (3.1.1)$$

where $k > 0$ is the spring constant and $r_0 > 0$ is the rest length. A natural choice for the rest length is $r_0 := 1$ [length].

It is sensible to model the covalent bonds with a harmonic potential: the true potential will be analytic, and in equilibrium we expect to be in a local minimum of the potential; analytic functions are well-approximated by parabolas around their minima⁵. More complex models for covalent bonds exist, such as Finitely Extensible Nonlinear Elastic (FENE) bonds, which limits the degree to which the bonds can be stretched [23, Ch. 3].

Since covalent bonds tend to be strong, the variance in the distance between bonded particles due to thermal fluctuations should be quite limited. Defining the displacement from the rest length as $\Delta r = r - r_0$, we can find that the typical displacement caused by thermal fluctuations will satisfy

$$\frac{1}{2}k\Delta r^2 \sim k_B T, \text{ so that } \Delta r \sim \sqrt{\frac{k_B T}{k}}.$$

Choosing $k = 10^3$ [energy] · [length]⁻², we find $\Delta r \sim 3 \times 10^{-2}$ [length].

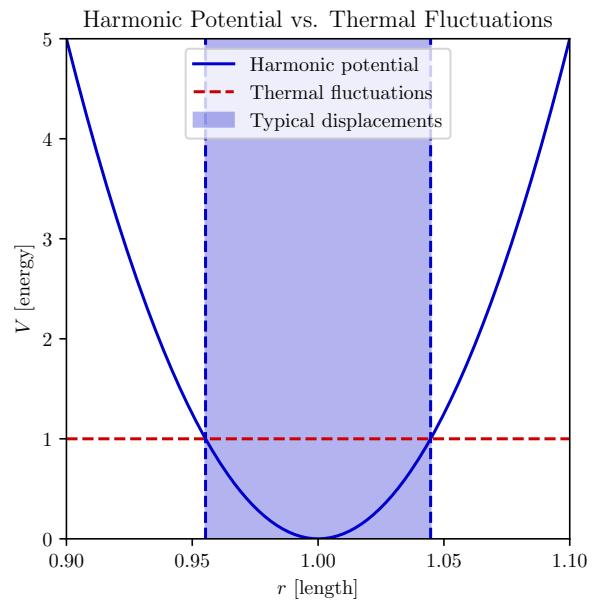


Figure 3.1: Plot of harmonic potential $V_{\text{harm}}(r; k, r_0)$ (Equation (3.1.1)) against r with $k = 10^3$ [energy] · [length]⁻² and $r_0 = 1$ [length]. The dashed green line represents the thermal fluctuations on the order of $k_B T = 1$ [energy]; the red area will typically be explored due to such thermal fluctuations.

⁴github: <https://github.com/glotzerlab/hoomd-blue>.

⁵Around the minimum, the approximation error will scale with the third power of the displacement.

Particles that are not covalently bonded still interact: at short range they strongly repel each other, such that they do not overlap, while at longer ranges they attract via van der Waals forces [23, Ch. 3]. These interactions are frequently modelled using a Lennard-Jones (LJ) potential:

$$V_{\text{LJ}}(r_{ij}) = 4\varepsilon_{\text{LJ}} \left(\left(\frac{\sigma_{\text{LJ}}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{\text{LJ}}}{r_{ij}} \right)^6 \right), \quad (3.1.2)$$

where r_{ij} is the distance between the pair of particles i and j , $\varepsilon_{\text{WCA}} > 0$ is the depth of the LJ potential well, and $\sigma_{\text{WCA}} > 0$ is the LJ length. We choose $\sigma_{\text{WCA}} := r_0 := 1$ [length], in accordance with [1]. Figure 3.2 shows the LJ potential and the corresponding force in red.

The attractive $\mathcal{O}(-r^{-6})$ term in Equation (3.1.2) can be theoretically justified: it has been derived from quantum mechanical considerations that the van der Waals potential energy scales in this way [62]. Conversely, the repulsive $\mathcal{O}(r^{-12})$ term was chosen mostly for computational convenience, as it can be computed from the $\mathcal{O}(-r^{-6})$ term by squaring. In practice, however, using a LJ potential in this form would be intractable, since it would have to be computed for every pair of particles.

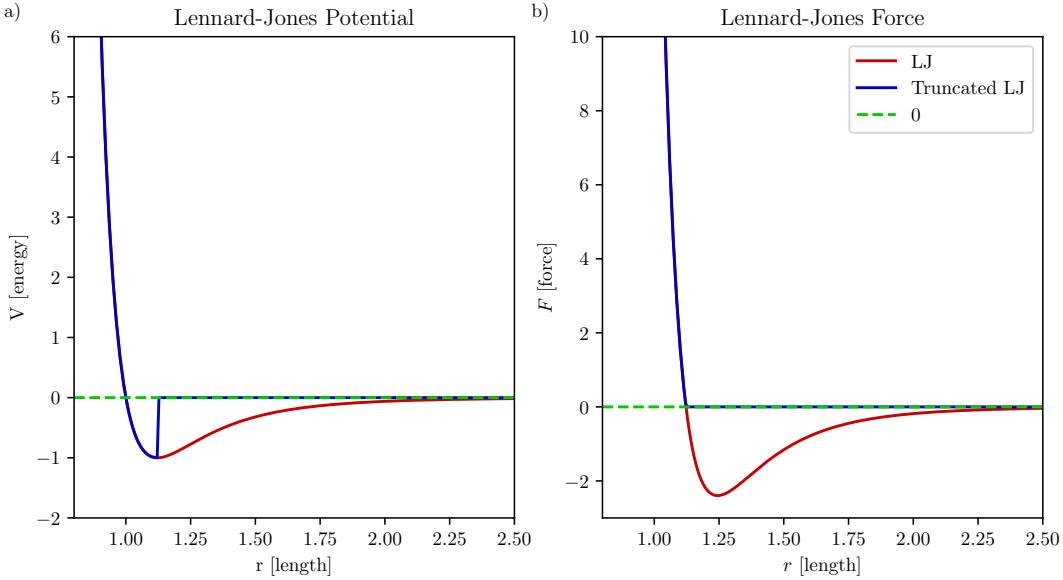


Figure 3.2: Plot of (truncated) LJ potential (Equations (3.1.2) and (3.1.3)) with $\varepsilon_{\text{WCA}} = 1$ [energy], $\sigma_{\text{WCA}} = 1$ [length], and $r_{\text{cut}} = 2^{\frac{1}{6}}\sigma_{\text{WCA}}$, and the corresponding force.

Consequently, the LJ potential is often truncated. In particular, the LJ has frequently been truncated at its minimum [23, Ch. 3]; it is straightforward to confirm that the LJ potential is minimised for $r_{\text{cut}} = r_{\text{min}} := 2^{\frac{1}{6}}\sigma_{\text{WCA}} \approx 1.12$ [length]. We can then write the resulting potential as:

$$V_{\text{LJ}}(r_{ij}) = \begin{cases} 4\varepsilon_{\text{LJ}} \left(\left(\frac{\sigma_{\text{LJ}}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{\text{LJ}}}{r_{ij}} \right)^6 \right), & 0 < r_{ij} \leq r_{\text{cut}}, \\ 0, & r_{ij} > r_{\text{cut}}; \end{cases} \quad (3.1.3)$$

this potential and the corresponding force are shown in blue in Figure 3.2. Clearly, this potential is purely repulsive and scales with distance r according to $\mathcal{O}(r^{-12})$; it is physically quite different from the untruncated LJ potential. It is additionally discontinuous $r = r_{\text{cut}}$, and so we will shift it to make it continuous. The resulting potential is called the Weeks-Chandler-Andersen (WCA) potential [23, 1], and is given by

$$V_{\text{WCA}}(r_{ij}) = \begin{cases} 4\varepsilon_{\text{WCA}} \left(\left(\frac{\sigma_{\text{WCA}}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{\text{WCA}}}{r_{ij}} \right)^6 + \frac{1}{4} \right), & 0 < r_{ij} \leq r_{\text{cut}}, \\ 0, & r_{ij} > r_{\text{cut}}. \end{cases} \quad (3.1.4)$$

Note that since only the force (see Figure 3.2 b)), i.e. the additive inverse of the derivative of the potential, is relevant for the dynamics, the LJ potential truncated at r_{min} and WCA potential should lead to similar

dynamics.⁶

Since the WCA potential is purely repulsive and scales with distance r according to $\mathcal{O}(r^{-12})$, it will be very sharp for any $\varepsilon_{\text{WCA}} > 0$, ensuring that particles cannot overlap. In our simulations, we will set $\varepsilon_{\text{WCA}} = 1$ [energy].

With these potentials alone we cannot yet model vitrimers: we still need to include the associative bond swaps. This can be done using a hybrid MD-MC approach, in which most of the dynamics is modelled using MD while the bond swapping is determined by MC sampling (e.g. [54, 49]). Another method, which does not require any MC, would be to keep track of the distance between reactive groups and add in bonds when the groups are sufficiently close (e.g. [59]). However, in 2017, Sciortino showed that vitrimer bonds can also be modelled using three-body potentials made by combining a pair of two-body potentials, which should be more computationally efficient [55].

For HOOMD-blue, Ciarella and Ellenbroek have implemented the three-body potential Reversible Crosslinks (RevCross) specifically for this purpose [28]. It should be noted that this implementation is symmetric in the sense that the associating particle can be of either type-A₁ or -A₂, in contrast with the asymmetric nature of physical bond swap mechanisms such as transesterification in which only moieties of one of the two types can associate (see Section 2.1). This is not a problem in practice if N_{A_1}/N_{A_2} is sufficiently large. RevCross makes use of the so-called Generalised Lennard-Jones (GLJ) potential⁷:

$$V_{\text{GLJ}}(r_{ij}; \varepsilon, \sigma, n, r_{\text{cut}}) = \begin{cases} 4\varepsilon \left(\left(\frac{\sigma}{r_{ij}} \right)^{2n} - \left(\frac{\sigma}{r_{ij}} \right)^n \right), & 0 < r_{ij} \leq r_{\text{cut}}, \\ 0, & r_{ij} > r_{\text{cut}}. \end{cases} \quad (3.1.5)$$

Notably, for the GLJ potential, the minimum is at $r = r_{\min} := 2^{\frac{1}{n}}\sigma$. A pair of A₁ and A₂ moieties, labelled i and j , respectively, will have a potential

$$V_{\text{RC, pair}}(r_{ij}) = V_{\text{GLJ}}(r_{ij})|_{n=10} = \begin{cases} 4\varepsilon \left(\left(\frac{\sigma}{r_{ij}} \right)^{20} - \left(\frac{\sigma}{r_{ij}} \right)^{10} \right), & 0 < r_{ij} \leq r_{\text{cut}}, \\ 0, & r_{ij} > r_{\text{cut}}. \end{cases} \quad (3.1.6)$$

We choose $n = 10$ and $\varepsilon = 10^2$ [energy], so that the bond between a pair behaves like a covalent bond [28]. While the RevCross potential is relatively efficient, care must be taken when choosing a cutoff distance so that it remains physically realistic yet computationally tractable; we choose $r_{\text{cut, RC}} := 2\sigma = 2$ [length], since we observed in tests that at this cutoff distance the maximum number of bonds is consistently achieved and maintained.

To construct the three-body interaction potential, we first define the following two-body potential [28]:

$$V_{2b}(r_{ij}) = \begin{cases} 1, & r_{ij} \leq 2^{\frac{1}{10}}\sigma, \\ -\frac{V_{\text{RC, pair}}(r_{ij}; \varepsilon, \sigma, r_{\text{cut}})}{\varepsilon}, & r_{ij} > 2^{\frac{1}{10}}\sigma, \end{cases} \quad (3.1.7)$$

or, equivalently⁸

$$V_{2b}(r_{ij}) = \begin{cases} 1, & 0 < r_{ij} \leq 2^{\frac{1}{10}}\sigma, \\ -4 \left(\left(\frac{\sigma}{r_{ij}} \right)^{20} - \left(\frac{\sigma}{r_{ij}} \right)^{10} \right), & 2^{\frac{1}{10}}\sigma < r_{ij} \leq r_{\text{cut}}, \\ 0, & r_{ij} > r_{\text{cut}}. \end{cases}$$

The three-body potential is then finally constructed as [28]

$$V_{\text{RC, triplet}}(r_{ij}, r_{ik}) = \lambda \varepsilon V_{2b}(r_{ij}) V_{2b}(r_{ik}). \quad (3.1.8)$$

The λ in Equation (3.1.8) sets the energy barrier for a swap event: we will choose $\lambda = 1$ such that there is no energy barrier [55, 28].

⁶Around the cutoff, we can imagine that the LJ potential will appear attractive if derivatives are calculated using finite differences

⁷A GLJ potential is used for convenience, as it is well-known and implemented in MD tools; unlike the LJ potential, it does not model repulsive and attractive forces between general particles.

⁸Note that the ε drops out.

3.1.1 Integrators and Thermostats

We have now alluded a few times to the fact that MD models molecular dynamics by integrating Newton's equations. In this subsection, we will explain what we mean by this, and discuss the integrators that we will use for our simulations.

We will start by looking at modelling isolated systems, which are described by a microcanonical Number, Volume, Energy (*NVE*) ensemble. We call it an *NVE* ensemble because the number of particles N , system volume V , and total system energy E are fixed [63, Sec. 13.9]. We denote the position, velocity, and acceleration of particle i with \mathbf{r}_i , \mathbf{v}_i , and \mathbf{a}_i , respectively, and the vectors of all particle positions, velocities, and accelerations \mathbf{r} , \mathbf{v} , and \mathbf{a} , so that

$$\mathbf{r} = (\mathbf{r}_1, \dots, \mathbf{r}_N), \mathbf{v} = (\mathbf{v}_1, \dots, \mathbf{v}_N), \text{ and } \mathbf{a} = (\mathbf{a}_1, \dots, \mathbf{a}_N).$$

If we assume that all the forces acting on the particles in the system are conservative, then there exists a potential energy function $U(\mathbf{r})$ such that [64, Eq. (4.13)]⁹

$$\mathbf{F} = -\nabla U(\mathbf{r}). \quad (3.1.9)$$

In our case, we choose U to be the sum of the potentials discussed above. Then Newton's second law tells us that [64, Eq. (1.19), Eq. (1.17)]⁹

$$\mathbf{F} = \mathbf{p} = m\mathbf{a}. \quad (3.1.10)$$

Consequently, we can find that the dynamics of the system are governed by the following set of differential equations:

$$\ddot{\mathbf{r}} = -\frac{1}{m} \nabla U(\mathbf{r}), \quad (3.1.11)$$

furnished with appropriate initial conditions $\mathbf{r}(0) = \mathbf{r}_0$ and $\mathbf{v}(0) = \mathbf{v}_0$. Hence, in theory, we could solve System (3.1.11) to find how our system evolves over time; in practice, however, doing this analytically is not feasible. Instead, MD solves System (3.1.11) numerically. This solving is called *integrating*, and the solver is therefore called the *integrator*. In HOOMD-blue, *NVE* ensembles are integrated with the Velocity-Verlet algorithm¹⁰. The Velocity-Verlet algorithm solves for both the positions \mathbf{r} and the velocities \mathbf{v} , by iterating the system [65]

$$\begin{cases} \mathbf{r}_{n+1} = \mathbf{r}_n + \Delta t \mathbf{v}_n + \frac{\Delta t^2}{2m} \mathbf{F}_n, \\ \mathbf{v}_{n+1} = \mathbf{v}_n + \frac{\Delta t}{2m} (\mathbf{F}_{n+1} + \mathbf{F}_n), \end{cases} \quad (3.1.12)$$

where $\mathbf{c}_k = \mathbf{c}(k\Delta t)$ is the k th iteration of quantity \mathbf{c} . Note that we also must compute $\mathbf{F}_n = -\frac{1}{m} \nabla U(\mathbf{r}_n)$ at each timestep. The Velocity-Verlet algorithm is a so-called symplectic integrator [66]; by definition, symplectic integrators conserve phase-space volume, which in essence means that the trajectories found by numerical integration stay close to the true trajectories even on long timescales [67]. As a consequence, symplectic integrators typically, and Velocity-Verlet in particular [66], approximately conserve the energy of the system [68]. Conversely, other, nonsymplectic integration schemes such as Forward Euler and Runge-Kutta will not conserve the energy, and can hence give very unphysical trajectories. In Figure 3.3, we show a nice example by Hairer, Lubich and Wanner of this difference between symplectic and nonsymplectic integrators. The Kepler problem describes e.g. the motion of a planet around the sun. We expect a closed, elliptic trajectory, which we indeed see in the exact solution. While the numerical solution generated using a symplectic integrator does not fall exactly on an ellipse, it stays close to the true trajectory. On the other hand, the nonsymplectic integrator yields a trajectory that moves increasingly far away from the true solution [69, Fig. 1.5].

It therefore makes a lot of sense to use a symplectic integrator such as Velocity-Verlet when modelling an *NVE* system. However, it is not realistic to assume that the systems we want to model, i.e. polymeric materials, are isolated. It would be more sensible to model the system as being closed, so that the number of particles N , the volume V , and the temperature T are fixed [63, Sec. 13.9]. In other words,

⁹In our simulations, the mass of all particles will be fixed and equal, so that $\dot{\mathbf{p}} = m\dot{\mathbf{v}} = \mathbf{F}$.

¹⁰See documentation. Our particles are isotropic, so we do not have any rotational degrees of freedom to integrate.

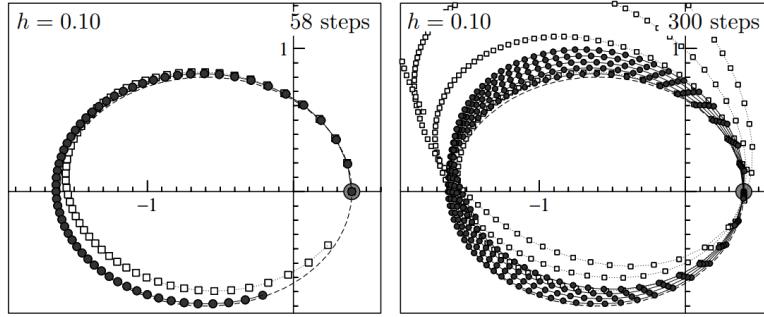


Figure 3.3: Solutions to the Kepler problem. Dashed lines represent the exact solution, black dots the numerical solution found using a symplectic integrator, and white boxes the numerical solution found using a nonsymplectic integrator. Taken from [69, Fig. 1.5].

we should model our system with a canonical Number, Volume, Temperature (*NVT*) ensemble. To control the temperature of the system, a so-called *thermostat* is employed. In a physical system, the thermostat corresponds to an external bath that is sufficiently large to keep its temperature constant when exchanging heat with the system of interest. In numerical simulation, there are various approaches to thermostat the system [70]: HOOMD-blue uses a Nose-Hoover thermostat¹¹. In short, the Nose-Hoover thermostat works by extending the Lagrangian of the system with by adding an additional virtual particle to the system [71, Subsec. 6.1.2].

For [1] – with which we would like to compare results – Langevin dynamics was used, which effectively models the particles as floating in a solvent, so that they experience random forces [70]. This was done to suppress fluctuations on short timescales, making it possible to extract linear response properties from the decay of correlations in equilibrium via the fluctuation-dissipation theorem. Notably, Langevin dynamics does not simulate an *NVT* ensemble. Since we are modelling polymer melts with no solvent, we find simulating a *NVT* ensemble with the Nose-Hoover thermostat more appealing; to circumvent the issues with short timescale fluctuations, we will measure the stress relaxation directly by applying a strain.

3.2 Network Generation and Evolution

In this section, we will delve a little deeper into how we generate and evolve polymer networks. We will start by tabulating the most important simulation parameters in Subsection 3.2.1. Then, in Subsection 3.2.2, we will discuss how to create – in a computationally efficient way – an initial system configuration that is somewhat realistic.

3.2.1 Simulation Parameters

We would like to be able to compare our results to those of Ciarella and Ellenbroek [1]; we therefore will choose our parameters as similar to those in [1] as possible. These parameter choices have been summarised in Table 3.1. Most of these parameters have been explained in Section 3.1; we will now go over the new ones.

Table 3.1: Simulation parameters to evolve the vitrimer networks.

parameter	value	parameter	value	parameter	value
$k_B T$	1 [energy]	σ_{WCA}	1 [length]	ε_{RC}	10^2 [energy]
φ	0.29	ε_{WCA}	1 [energy]	$r_{cut,RC}$	2 [length]
r_0	1 [length]	$r_{cut,WCA}$	$2^{\frac{1}{6}}$ [length]	dt	10^{-3} [time]
k	10^3 [energy] · [length] ⁻²	σ_{RC}	1 [length]	τ_{NVT}	10^{-1} [time]

- φ fixes the volume fraction of the network, where the volume of each particle is taken to be

$$\frac{4\pi}{3} \left(\frac{\sigma_{WCA}}{2} \right)^3 = \frac{\pi}{6} [\text{length}]^3,$$

¹¹See documentation. This is true in HOOMD-blue 3; in HOOMD-blue 4 the user may choose a different thermostat, such as the Bussi-Donadio-Parrinello thermostat, see documentation.

i.e. , we say the particles are spherical with radius $\sigma_{\text{WCA}}/2 = \sigma_{\text{RC}}/2$. It is important that the mixture is sufficiently dense that free dangling arms can find bonds, but not so high that the system becomes glassy, at which stage the motion of arms would be inhibited by surrounding particles [1].

- The integrators use a timestep $dt = 10^{-3}$ [time]. This seems to be sufficiently small that the evolution of each of our configurations is stable.
- τ_{NVT} is the coupling constant of the Nosé-Hoover *NVT* integrator. In simulations using the Nosé-Hoover thermostat, the temperature will fluctuate; the coupling constant sets the timescale on which the temperature will tend to the target value that we impose. We have $\tau_{NVT} = 100dt$ as this is the value recommended by HOOMD-blue¹².

We use a variety of mixtures of polymer building blocks to make our vitrimer networks. All of our samples are Defect-Allowing Mixture (DAM) (see Section 2.2), as in [1]. This means that we have two species of building block, with either roughly 1/8 of the reactive particles will be type-A₁ and about 7/8 will be type-A₂, or the other way around. For each mixture we choose the fraction of each species such that in total 60 % of the reactive particles are A₁ moieties, like in [1].

3.2.2 Network Generation

We have created Hoomd Polymer Tools¹³, a collection of Python scripts which reduce the amount of boilerplate code needed to initialise and run simulations. Hoomd Polymer Tools also includes many functions for analysing e.g. connectedness of a polymer network or the swap time distributions. Here, we will sketch the procedures we use to simulate vitrimers.

The first step of any of our MD experiments will be to generate an initial vitrimer network. This generated network should have a realistic configuration when we start measuring its properties; since we want to simulate a polymer melt, we expect roughly ideal, i.e. random walk-like, chain conformations [72, Sec. 4.5.2]. However, if we naively generate polymers with random walks, we will end up with overlapping beads, where the distance between beads will be smaller than σ . As can be seen in Figure 3.2, LJ-like potentials are very steep for $r < \sigma$, and consequently overlapping beads experience extremely large forces, which can lead to numerical instabilities.

It is possible to start with a very sparse configuration of stretched polymers, such that we may be certain that there are no overlapping beads. We must subsequently compress to the desired volume fraction. Then, however, we must simulate a long time to get realistic, entangled chain conformations. This is problematic since the entire premise of our hypothesis is that comb polymers would be suitable building blocks for self-healing vitrimer networks because the entanglement makes them more rigid.

Therefore, we will generate the initial conformations using random walks. With a self-avoiding random walk, the overlapping can be avoided, though we cannot fully control the length of polymers since they might get terminated prematurely [23, Sec. 3.4]. We instead initialise with a non-self-avoiding random walk, leading to a network that likely contains overlapping beads. We subsequently relax the system; this can be done in HOOMD-blue by integrating with `DisplacementCapped`¹⁴, which adapts a Velocity-Verlet *NVE* ensemble integrator by limiting the maximum displacement in each timestep. This relaxation is very quick: we need only 10^4 integration steps of size 10^{-3} [time].

With this process we generate a dense, (possibly) entangled polymer network. However, it is important to additionally introduce the RevCross potential, so that bonds can be formed. For this, we use a Nosé-Hoover *NVT* integrator¹², which we will also use for the network evolution. Ideally, at the end of the equilibration, the maximum number of bonds N_{A_2} will have formed. With the polymer building blocks we considered, this seems to occur within 9×10^4 steps of size 10^{-3} [time].

HOOMD-blue works with a periodic simulation box. It is important to make this box sufficiently large to minimise periodic effects; if the box were too small, one of the building blocks could form a reversible

¹²See documentation.

¹³Hoomd Polymer Tools on GitLab.

¹⁴See documentation.

bond with itself, for instance. To model vitrimer networks with damage, we include LJ-walls along two opposing boundaries of the simulation box during the entire generation process, which will prevent the formation of RevCross bonds across the corresponding boundaries. To investigate the self-healing, we stack two damaged samples and remove the LJ-walls between them. The resulting system will then have two healing interfaces: one at the top and bottom of the simulation cell and one in the centre. Conversely, if we are interested in equilibrium properties of the material, we use a single undamaged sample, which was generated without any LJ-walls.

3.2.3 Network Evolution

After the network generation, we would like to evolve the system in a physically realistic way. In a real life situation we could reasonably control the number of particles N and the temperature T . In chemistry it is typically assumed that also the pressure P is held constant **citation**. Since we are dealing with (viscoelastic) solids, however, it makes sense to instead fix the volume V **citation**. Hence, we will assume that the states of our system may be described by the canonical NVT ensemble, and we will also use a Nosé-Hoover NVT integrator during the network evolution process.

3.3 RevCross GPU Issues

Unfortunately, we encountered issues when simulating vitrimer networks using the RevCross potential on GPUs: we observed very different dynamics than in comparable simulations run on CPUs. We additionally believe that the behaviour observed on GPUs is extremely nonphysical. For instance, on the GPU we found that the typical swap times do not depend on the concentration of vitrimer particles, contradicting the results found using CPUs (discussed in 4.1.3). The GPU swap times also appear to be exponentially distributed, whereas the CPU swap times are quite clearly not.

The issue occurs in both NVE and NVT simulations. Moreover, we observe different trajectories each time a simulation is run, even though the methods should be deterministic and hence should yield the same results given the same input.

We did not have enough time to find the root cause of the issue, but we have some ideas. The fact that the swap times are exponentially distributed, implies that swaps occur at a constant rate. This is consistent with the behaviour we would get if at each timestep every particle has a small chance of being switched with another particle.

Consequently, we do not trust physical results generated on GPUs making use of RevCross. However, as we will discuss in Subsection 4.3, we do not think it is feasible to measure the relaxation of a comb vitrimer network to liquid on CPUs, since the GPUs in the TU/e hpc cluster can execute simulations on the order of one hundred times faster than the CPUs therein. We therefore think that it would be valuable to fix the implementation of RevCross on the GPU.

Chapter 4

Mathematical Model

In this chapter, we will develop and test mathematical models in order to help answer how we can design a polymer material that is capable of healing while being mechanically strong. Many simple models have been developed to describe polymers. The Rouse model, for instance, describes the behaviour of unentangled chains in polymer melts [73, p. 340], while the reptation model was created to model the dynamics of entangled chains [74, Sec. 8.3]. In 2013, Stukalin et al. came up with a simple model for the self-healing of *dissociative* Covalent Adaptive Networks (CANs) [22]. However, we have not found any model for the self-healing of vitrimers in the literature.

We will first study the self-healing properties of comb and star polymer vitrimer networks. As noted in Subsection 2.2.1, the affine network model predicts that the number of vitrimer bonds across the healing interface could be a good proxy for the mechanical strength of the healed wound. For this reason, we will develop a healing model which describes how the number of interfacial bonds evolves over time in Section 4.2. Since the swap rate will clearly play a role in how fast interfacial bonds can be formed, we will first dedicate Section 4.1 to the typical swap time; for this, we can reuse some ideas from [22]. Not all swaps are created equal: we will therefore look at doing and undoing swaps in Subsection 4.1.1. We will round off Section 4.1 by performing and analysing simulations to empirically determine swap time distributions in Subsection 4.1.3.

In Subsection 4.2.1, we will put our newfound knowledge of bond swaps to use in a model that predicts the number of bonds across the healing interface. We will thereafter link the evolution of the number of interfacial bonds to the equilibrium shear modulus using the affine network model in Subsection 4.2.2. These healing models will then be tested using simulations in Subsection 4.2.3.

We will conclude by turning our attention to relaxation of stress in an undamaged comb vitrimer, to test our hypothesis that vitrimers made of combs have better long term mechanical properties than ones made of stars due to the entanglement of the stymieing the bond swap-mediated stress relaxation. Ciarella, Sciortino and Ellenbroek have measured the relaxation of a star polymer vitrimer network [24], providing a benchmark timescale on which the network should not behave like a liquid.

4.1 Swap Time

We commence our investigation of swap times by considering a single minority moiety, which we assume starts off in a bonded state. Consequently, as bonds may not simply break due to thermal fluctuations (see Section 2.1), at any positive time it will either be bonded to a single majority moiety, or will be associating with two majority moieties. If there are sufficiently many unbonded majority moieties, so $N_{A_1} \gg N_{A_2}$, and the free dangling arms are sufficiently long, then at some stage such a free dangling arm will encounter the bond containing our target minority moiety, and so there will be two majority moieties associating with the target. After some time, which we call the *association time* T_{ass} , one of the majority partners will leave the association. We call the time between two subsequent associations the *interassociation time* T_{int} . This process of alternating association and interassociation periods goes on indefinitely: we therefore define the sets of random variables $\{T_{\text{ass},i}\}_{i=1}^{\infty}$ and $\{T_{\text{int},i}\}_{i=1}^{\infty}$, where $T_{\text{ass},i}$ and $T_{\text{int},i}$ are the i th association and interassociation time of the target minority moiety, respectively.

Suppose now that the partner just before and just after an association is the same. Then, effectively no swap has occurred: we call this a *failed swap*. If instead a different majority moiety has bonded to the target, we say that a *proper swap has* occurred; we are interested in the distribution of the time between proper swaps.

Definition 4.1.1 (Swap Time). Let $\{T_{\text{ass},i}\}_{i=1}^{\infty}$ and $\{T_{\text{int},i}\}_{i=1}^{\infty}$ be the association and interassociation times, respectively, of a minority moiety, and let N be the (random) number of associations until the first successful swap has occurred. Then, the *swap time* is defined as

$$T_{\text{sw}} := \sum_{i=1}^N (T_{\text{ass},i} + T_{\text{int},i}). \quad (4.1.1)$$

Correspondingly, the *typical swap time* is defined as

$$\tau_{\text{sw}} := \mathbb{E}[T_{\text{sw}}], \quad (4.1.2)$$

which is well-defined, though not necessarily finite, since T_{sw} is positive.

Definition 4.1.1 gives only the first swap time, but it is obvious that it may be adapted to find later swap times too.

Specifying the entire distribution of T_{sw} would be rather tricky. Indeed, previous simulation studies have concluded from the decay of the bond autocorrelation function, which corresponds to the fraction of bonds that have survived to a certain time,¹ have concluded that the swap time is not exponentially distributed: Wu et al., who employed a hybrid MD-MC approach, found a stretched exponential fit [54], which corresponds to a Weibull lifetime distribution; Rovigatti et al., who modelled the vitrimer bonds with a three-body potential similar to RevCross, found conversely that the swap times are not described by a Weibull distribution [75].

We will consequently focus on describing just the mean τ_{sw} in terms of a limited number of parameters that we can directly link to the physical shape of the building blocks of our vitrimer networks, which will involve evaluating the expectation in Equation (4.1.2); we will rely on simulations for the entire swap time distributions. To proceed, we will need to think about how the random variables that make up T_{sw} depend on each other. For this, the following assumption is very convenient:

Assumption 4.1.2. N , $\{T_{\text{ass},i}\}_{i=1}^{\infty}$, and $\{T_{\text{int},i}\}_{i=1}^{\infty}$ are independent.

To see if this is a reasonable assumption, we define a further random variable:

$$P_i := \text{partner after } i\text{th association.}$$

Without loss of generality we may say that $P_0 = 0$. A proper swap occurs when the partner changes after an association. Clearly then it must hold that

$$N = \min\{n \in \mathbb{N} \mid P_i \neq 0\}.$$

A logical consequence hereof is that $\mathbf{1}\{P_N \neq 0\} = 1$ and that $P_{N-1} = 0$. We therefore now investigate how the (conditional) indicator $\mathbf{1}\{P_i \neq 0\} \mid P_{i-1} = 0$ for $i \in \mathbb{N}$ might depend on $\{T_{\text{ass},i}\}_{i=1}^{\infty}$ and $\{T_{\text{int},i}\}_{i=1}^{\infty}$.

At the end of an association event, one of the majority moiety particles is randomly selected to remain as partner. It is conceivable that the selection depends on the length of the association. For instance, if the association is short, that might mean that there was not enough time to allow the system to locally reconfigure to make it favourable for the new particle to remain, so that it is more likely to be ejected. However, such a short association could also be caused by the new particle entering in such a way that

¹In theory, this corresponds to the survival function. However, it is calculated in a different way than the KM estimate. Since only the first swap of each bond contributes to the function, we believe it is less efficient than the KM estimate, which can use every swap of every bond observed during the simulation.

the old partner leaves quickly. Since we cannot determine any a priori preference for either possibilities, it makes sense to simply assume that $\mathbf{1}\{P_i \neq 0\} | P_{i-1} = 0$ is independent of $T_{\text{ass},i}$.

The selection of the new partner occurs fully during the association; there is therefore no reason to believe that the length of the interassociation will influence or be influenced by which partner is chosen. In other words, it seems sensible to assume that $\mathbf{1}\{P_i \neq 0\} | P_{i-1} = 0$ is independent of $T_{\text{int},i}$. There is similarly no evidence to suggest that previous and future interassociations and associations will be correlated to the current selection process, justifying the assumption that $\mathbf{1}\{P_i \neq 0\} | P_{i-1} = 0$ is independent of $T_{\text{ass},j}$ and $T_{\text{int},j}$ for $j \neq i$.

Since N is fully determined $\{\mathbf{1}\{P_i \neq 0\} | P_{i-1} = 0\}_{i=1}^{\infty}$ it then seems reasonable to model N as being independent of $\{T_{\text{ass},i}\}_{i=1}^{\infty}$ and $\{T_{\text{int},i}\}_{i=1}^{\infty}$. We furthermore have no reason to believe that length of associations will influence the time between associations or vice versa. In conclusion, we believe that Assumption 4.1.2 is reasonable.

We will next investigate the distributions of N , $\{T_{\text{ass},i}\}_{i=1}^{\infty}$, and $\{T_{\text{int},i}\}_{i=1}^{\infty}$ so that we may compute Equation (4.1.2). Since $\mathbf{1}\{P_i \neq 0\} | P_{i-1} = 0$ is a binary random variable, there exists for each $i \in \mathbb{N}$ some $p_i \in (0, 1)$ such that

$$\mathbf{1}\{P_i \neq 0\} | P_{i-1} = 0 \sim \text{Bern}(p_i),$$

where p_i does not depend on $T_{\text{ass},i}$ or $T_{\text{int},i}$. We once again assume that our system evolves in a temporally homogeneous way, so that there exists a $p \in (0, 1)^2$ such that $p_i = p$ for $i \in \mathbb{N}$. In other words:

Assumption 4.1.3. There is a $p \in (0, 1)$ such that

$$\mathbf{1}\{P_i \neq 0\} | P_{i-1} = 0 \sim \text{Bern}(p).$$

Assumption 4.1.3 allows us to specify the distribution of N :

Lemma 4.1.4 (Distribution of N). $N \sim \text{Geom}(p)$, and consequently $\mathbb{E}[N] = \frac{1}{p}$.

Proof. We have defined N as the number of associations until a new partner bonds. We therefore can interpret N as counting a number of trials (associations) until the first success (new bond partner). Assumption 4.1.3 now tells us that each trial independently either succeeds with some probability $p > 0$ or fails with probability $1 - p$. It then follows that $N \sim \text{Geom}(p)$ [76, Def. 3.5(5)]. We may subsequently compute that

$$\begin{aligned} \nu := \mathbb{E}[N] &:= \sum_{k=1}^{\infty} kp(1-p)^{k-1} = -p \sum_{k=1}^{\infty} \frac{\partial(1-p)^k}{\partial p} \stackrel{(I)}{=} -p \frac{\partial}{\partial p} \sum_{k=1}^{\infty} (1-p)^k \\ &= -p \frac{\partial}{\partial p} \sum_{k=0}^{\infty} (1-p)^k \stackrel{(II)}{=} -p \frac{\partial}{\partial p} \frac{1}{1-(1-p)} = -p \cdot -\frac{1}{p^2} = \frac{1}{p}, \end{aligned}$$

where we could exchange summation and differentiation in (I) since this is a power series [77, Thm. 8.5.15], and in (II) we identified the geometric series [78, Thm. 7.1.14]. \square

With the following assumption, we can greatly reduce the number of relevant random variables:

Assumption 4.1.5. There are random variables T_{ass} and T_{int} such that $T_{\text{ass},i} \sim T_{\text{ass}}$ and $T_{\text{int},i} \sim T_{\text{int}}$ for all $i \in \mathbb{N}$. In other words, each $T_{\text{ass},i}$ and each $T_{\text{int},i}$ is identically distributed.

As a consequence of Assumption 4.1.5, we can see that

$$\mathbb{E}[T_{\text{ass},i}] = \mathbb{E}[T_{\text{ass}}] =: \tau_{\text{ass}}, \text{ and } \mathbb{E}[T_{\text{int},i}] = \mathbb{E}[T_{\text{int}}] =: \tau_{\text{int}}.$$

²It would make sense, especially after a long association, that each majority moiety is equally likely to get ejected, so that $p = \frac{1}{2}$; for the moment, however, we leave p as an unknown parameter in our model.

In essence, Assumption 4.1.5 states that the times related to the swap process do not change as the system evolves.³

The above results now allow us to describe the typical swap time in terms of just three parameters.

Theorem 4.1.6 (Typical Swap Time). *The typical swap time defined in Definition 4.1.1 is given by*

$$\tau_{\text{sw}} := \mathbb{E}[T_{\text{sw}}] = \frac{\tau_{\text{ass}} + \tau_{\text{int}}}{p}. \quad (4.1.3)$$

Proof. Starting with the definition of the typical swap time from Equation (4.1.2), we can apply the law of total expectation [33, Thm. 4.6(5)] to find that

$$\tau_{\text{sw}} := \mathbb{E}\left[\sum_{i=1}^N (T_{\text{ass},i} + T_{\text{int},i})\right] = \mathbb{E}\left[\mathbb{E}\left[\sum_{i=1}^N (T_{\text{ass},i} + T_{\text{int},i}) \mid N\right]\right].$$

Then, for any $n \in \mathbb{N}$ we can use the linearity of the expectation to see that

$$\mathbb{E}\left[\sum_{i=1}^N (T_{\text{ass},i} + T_{\text{int},i}) \mid N = n\right] = \mathbb{E}\left[\sum_{i=1}^n (T_{\text{ass},i} + T_{\text{int},i}) \mid N = n\right] = \sum_{i=1}^n \mathbb{E}[(T_{\text{ass},i} + T_{\text{int},i}) \mid N = n].$$

It then follows from Assumption 4.1.5 that

$$\sum_{i=1}^n \mathbb{E}[T_{\text{ass},i} + T_{\text{int},i} \mid N = n] = \sum_{i=1}^n \mathbb{E}[T_{\text{ass},i} + T_{\text{int},i}] = n(\tau_{\text{ass}} + \tau_{\text{int}}).$$

We can then apply Lemma 4.1.4 to find that

$$\tau_{\text{sw}} = \mathbb{E}[N(\mathbb{E}[T_{\text{ass},i}] + \mathbb{E}[T_{\text{int},i}])] = \mathbb{E}[N](\tau_{\text{ass}} + \tau_{\text{int}}) = \frac{\tau_{\text{ass}} + \tau_{\text{int}}}{p}.$$

We may therefore conclude that the typical swap time is given by

$$\tau_{\text{sw}} = \frac{\tau_{\text{ass}} + \tau_{\text{int}}}{p},$$

as required. □

Therefore, we are left with three parameters on which the typical swap time τ_{sw} depends: the proper swap probability p , the typical association time τ_{ass} , and the typical time between associations τ_{int} . We have no reason to believe that the association depends in any way on the shapes of the building blocks of the vitrimers. Hence, p and τ_{ass} will essentially be fit parameters in our model. Conversely, it would make sense if the time between associations τ_{int} depended on the length and concentration of the free dangling arms; we will therefore investigate τ_{int} in greater detail in the next subsection.

4.1.1 Interassociation Time

Consider again a minority moiety which has just undergone a swap. There are now exactly two possibilities:

1. Before a new majority moiety has a chance to bond with the minority moiety, the departing majority moiety returns and rebinds. In this case we say an *undoing* swap has occurred;
2. A new majority moiety binds with the minority moiety before the departing majority moiety can rebind to it. In this case we say a *doing* swap has occurred.

³If the system were in equilibrium, this would certainly be true, since then the system is homogeneous in time. Note, however, that self-healing is clearly not an equilibrium process, and certain aspects of the system – such as the number of vitrimer bonds at the damage – will likely vary over time. It therefore stands to reason that the distributions of $T_{\text{ass},i}$ and $T_{\text{int},i}$ do in fact vary with i .

Unfortunately, the start and finish of an association are ill-defined, which means that doing and undoing swaps are also ill-defined. We will therefore define swaps in our model in the same way as we do in our simulations: we will say that an association is occurring when there are two majority moieties within the cutoff distance $r_{\text{cut,RC}} = 2\sigma = 2$ [length] to the minority moiety. We furthermore say that a swap has occurred when the partner changes.⁴ Finally, from this swap definition we can easily identify when undoing and doing occur. Note that a swap is always either doing or undoing, so that for $t \geq 0$

$$\mathbb{E}[T_{\text{int}}] = \mathbb{E}[T_{\text{int}} \mid \text{doing}] \mathbb{P}(\text{doing}) + \mathbb{E}[T_{\text{int}} \mid \text{undoing}] \mathbb{P}(\text{undoing}) \quad (4.1.4)$$

by the Law of Total Expectation [33, Thm. 4.6(5)]. We will therefore now investigate doing and undoing swaps separately.

We will first examine undoing swaps. Intuitively, if the particle is going to reassociate, it will do so while it is still close to the bond. In that case, it will explore a neighbourhood with radius roughly σ , so that this neighbourhood will have a volume V such that $V^{\frac{1}{3}} \sim \sigma$. Notably, this volume will not depend on the shape of the building blocks of the network. It would be convenient to be able to use an existing polymer model to determine how long it takes for the departing moiety to explore its neighbourhood.

Assumption 4.1.7. The arms of the polymer building blocks of the network are sufficiently long enough to be well-modelled by the Rouse model.

Using the Rouse model, we can find that the time it takes for the majority moiety to explore this neighbourhood will be on the order of τ_0 – the Kuhn monomer relaxation time – since the volume to be explored is roughly the size of a single monomer [73, Eq. (8.44)], so that

$$\mathbb{E}[T_{\text{int}} \mid \text{undoing}] = \alpha_{\text{undoing}} \tau_0, \quad (4.1.5)$$

for some proportionality constant $\alpha_{\text{undoing}} > 0$. While the time it takes for the majority moiety to explore its neighbourhood should not depend on the shape parameters, the probability that it encounters its old bond certainly will: if it encounters a different bond first, or if its previous bond is already associating with a different majority moiety, the undoing swap will be prevented. Note that an undoing swap is only possible if the previous swap was a doing swap. Hence, using the Law of Total Probability [79, Lem. 1.4(4)] we can find that

$$\begin{aligned} \mathbb{P}(\text{undoing}) &= \mathbb{P}(\text{undoing} \mid \text{previous doing}) \underbrace{\mathbb{P}(\text{previous doing})}_{=\mathbb{P}(\text{doing})} \\ &\quad + \underbrace{\mathbb{P}(\text{undoing} \mid \text{previous undoing})}_{=0} \mathbb{P}(\text{previous undoing}) \\ &= \mathbb{P}(\text{undoing} \mid \text{previous doing}) \mathbb{P}(\text{doing}). \end{aligned}$$

Rewriting this, noting that $\mathbb{P}(\text{doing}) = 1 - \mathbb{P}(\text{undoing})$, we see that

$$\mathbb{P}(\text{undoing}) = \frac{\mathbb{P}(\text{undoing} \mid \text{previous doing})}{1 + \mathbb{P}(\text{undoing} \mid \text{previous doing})}. \quad (4.1.6)$$

Determining $\mathbb{P}(\text{undoing} \mid \text{previous doing})$ – the probability that a different moiety is able to associate with the bond before the departing moiety return – turns out to be rather difficult; with the following assumption we will fortunately be able to neglect this contribution entirely.

Assumption 4.1.8. The time it takes for a departing moiety to explore its neighbourhood is much smaller than the time it typically takes for another majority moiety to reach the vitrimer bond.

Under Assumption 4.1.8, undoing swaps are only averted by the departing moiety finding a different bond. Clearly, as the number of vitrimer bonds in the neighbourhood grows, the chance that the majority moiety will first find a different bond will increase. We can make this idea more concrete with the following definitions:

⁴While this seems like the obvious definition, with this definition we fail to distinguish between failed swaps where an association has occurred but the same partner remains bonded, and cases where the association was actually never really over.

Definition 4.1.9 (Vitrimer Concentrations). We define ρ_{total} as the concentration, i.e. the total number over the total volume, of particles, ρ_{vit} as the concentration of vitrimer particles, ρ_{bonds} as the concentration of vitrimer bonds, and ρ_{reactive} as the concentration of free reactive vitrimer moieties.

We can hence see that the neighbourhood explored by the majority moiety contains roughly $1 + V \cdot \rho_{\text{bonds}}$ vitrimer bonds.⁵ Then, the majority moiety will rebind with its previous bond if it encounters that bond first *and* that bond is not yet associating with a new majority moiety, so that the probability that the majority moiety first encounters the bond it just left will be roughly $(1 + V \rho_{\text{bonds}})^{-1}$. Hence, we can conclude that

$$\begin{aligned} p_{\text{undoing}} &:= \mathbb{P}(\text{undoing}) = \frac{\mathbb{P}(\text{undoing} \mid \text{previous doing})}{1 + \mathbb{P}(\text{undoing} \mid \text{previous doing})} \\ &= \frac{(1 + V \rho_{\text{bonds}})^{-1}}{1 + (1 + V \rho_{\text{bonds}})^{-1}} = \frac{1}{2 + V \rho_{\text{bonds}}}. \end{aligned} \quad (4.1.7)$$

Since the neighbourhood we defined is rather vague, it makes sense to fit Equation (4.1.7) for V . Notably $p_{\text{undoing}} \in [0, \frac{1}{2}]$ and

$$\lim_{\rho_{\text{bonds}} \rightarrow 0} p_{\text{undoing}} = \frac{1}{2}, \quad \text{and} \quad \lim_{\rho_{\text{bonds}} \rightarrow \infty} p_{\text{undoing}} = 0,$$

which are sensible limits: if the concentration of bonds is sufficiently low, a departing moiety will never be able to find a different bond to associate with; if the concentration is sufficiently high, at the end of its association the majority moiety will be so close to other vitrimer bonds that it will always associate with another bond first. With the next lemma, we will be able to compute ρ_{bonds} in terms of the building block shape parameters.

Lemma 4.1.10 (Vitrimer Concentrations). *Let σ be the diameter, and φ be the volume fraction of particles in a vitrimer network. Suppose that the vitrimer network is made of comb polymers with N_{arms} arms, each containing n_{arm} particles, separated by n_{sep} particles on the backbone (see Section 2.3). Then, the concentration of vitrimer particles is given by*

$$\rho_{\text{vit}} = \frac{6\varphi}{\pi\sigma^3} \frac{N_{\text{arms}}}{N_{\text{arms}}(n_{\text{arm}} + n_{\text{sep}} + 1) - n_{\text{sep}}}. \quad (4.1.8)$$

Conversely, if the vitrimer network is made of star polymers with N_{arms} arms, each containing n_{arm} particles. Then, the concentration of vitrimer particles is given by

$$\rho_{\text{vit}} = \frac{6\varphi}{\pi\sigma^3} \frac{N_{\text{arms}}}{N_{\text{arms}}(n_{\text{arm}} + 1)}. \quad (4.1.9)$$

Proof. Consider the concentration of particles ρ_{total} is given by the ratio of the volume fraction φ over the volume of a particle V_p . Since our particles are spherical and have diameter σ , we can hence find that

$$\rho_{\text{total}} = \frac{\varphi}{V_p} = \varphi \left(\frac{4\pi}{3} \left(\frac{\sigma}{2} \right)^3 \right)^{-1} = \frac{6\varphi}{\pi\sigma^3}.$$

In a comb vitrimer network, each building block has N_{arms} vitrimer particles. There are $N_{\text{arms}} \cdot n_{\text{arm}}$ particles in the arms of each comb; the backbone consists of $(N_{\text{arms}} - 1) \cdot n_{\text{sep}} + 1$ particles. Hence, the fraction of vitrimer particles on each building block is given by

$$f_{\text{vit}} = \frac{N_{\text{arms}}}{N_{\text{arms}} \cdot n_{\text{arm}} + (N_{\text{arms}} - 1) \cdot (n_{\text{sep}} + 1) + 1} = \frac{N_{\text{arms}}}{N_{\text{arms}}(n_{\text{arm}} + n_{\text{sep}} + 1) - n_{\text{sep}}}.$$

Therefore we have shown that indeed the concentration of vitrimer particles in a comb vitrimer network is given by

$$\rho_{\text{vit}} = \rho_{\text{total}} \cdot f_{\text{vit}} = \frac{6\varphi}{\pi\sigma^3} \frac{N_{\text{arms}}}{N_{\text{arms}}(n_{\text{arm}} + n_{\text{sep}} + 1) - n_{\text{sep}}}.$$

⁵The constant 1 comes from the fact that the neighbourhood certainly contains the bond that the majority moiety just departed from.

In a star vitrimer network, each building block has N_{arms} vitrimer particles. There are $N_{\text{arms}} \cdot n_{\text{arm}} + 1$ particles in each star, so that the fraction of vitrimer particles on each building block is given by

$$f_{\text{vit}} = \frac{N_{\text{arms}}}{N_{\text{arms}}n_{\text{arm}} + 1},$$

as required. \square

As the backbone grows, the number of arms on each comb increases; in the limit we see that

$$\begin{aligned} \lim_{N_{\text{arms}} \rightarrow \infty} \rho_{\text{vit}} &= \lim_{N_{\text{arms}} \rightarrow \infty} \frac{6\varphi}{\pi\sigma^3} \frac{N_{\text{arms}}}{N_{\text{arms}}(n_{\text{arm}} + n_{\text{sep}} + 1) - n_{\text{sep}}} \\ &= \lim_{N_{\text{arms}} \rightarrow \infty} \frac{6\varphi}{\pi\sigma^3} \frac{1}{n_{\text{arm}} + n_{\text{sep}} + 1 - \frac{n_{\text{sep}}}{N_{\text{arms}}}} = \frac{6\varphi}{\pi\sigma^3} \frac{1}{n_{\text{arm}} + n_{\text{sep}} + 1}. \end{aligned}$$

Since we assume all the minority moieties are bonded, we know that there are exactly as many bonds as there are minority moieties. If we now moreover assume that at any given time the number of associations that are ongoing is negligible, we can conclude that the number of free reactive particles is the number of majority moieties minus the number of minority moieties. Hence, the concentration of vitrimer bonds is given by

$$\rho_{\text{bonds}} = \rho_{A_2} = \rho_{\text{vit}} \cdot f_{A_2},$$

while the concentration of free reactive moieties is given by

$$\rho_{\text{reactive}} = \rho_{A_1} - \rho_{A_2} = \rho_{\text{vit}} \cdot (f_{A_1} - f_{A_2}) = \rho_{\text{vit}} \cdot (1 - 2f_{A_2}),$$

where f_{A_1} and f_{A_2} are the fraction of vitrimer particles that are type- A_1 and - A_2 , respectively. By construction, all the network we will consider have $f_{A_1} \approx 0.6$ and $f_{A_2} \approx 0.4$. In short, we have found that

$$p_{\text{undoing}} = \frac{1}{1 + V\rho_{\text{bonds}}} = \frac{1}{1 + V\rho_{\text{vit}}f_{A_2}}. \quad (4.1.10)$$

To derive this result, we neglected the possibility that other majority moiety could associate with the bond before the departing moiety can return. Hence, Equation (4.1.10) will overestimate the true probability of an undoing swap. Moreover, the error won't be uniform: as the concentration of free reactive particles increases, the probability that another moiety will beat the departing particle to the bond will become larger.

Next, we turn our attention to doing swaps. We can directly find the probability that the swap will be undoing from Equation (4.1.10):

$$\mathbb{P}(\text{doing}) = 1 - \mathbb{P}(\text{undoing}) = 1 - \frac{1}{2 + V\rho_{\text{vit}}f_{A_2}} = \frac{1 + V\rho_{\text{vit}}f_{A_2}}{2 + V\rho_{\text{vit}}f_{A_2}}; \quad (4.1.11)$$

the estimate in Equation (4.1.11) will evidently suffer the same issues as the estimate for p_{undoing} . To find the interassociation time for doing swaps, we look to the literature. In [22], Stukalin et al. also investigate a self-healing polymer network, which consists of many free dangling arms fixed on one end to a stationary background network. The ends of such free dangling arms can bind together through dissociative, reversible bonds. One of their goals is to determine the *typical renormalised bond lifetime* τ_{renm} , which requires calculating how long it takes on average for a pair of reactive particles to encounter each other and how many bond events it takes on average for one of the reactive particles to bond with a different particle than before. This appears to be completely analogous to the interassociation time in the case of a doing swap. In particular, we know that the distance of the target bond to a free majority moiety is on the order of $r := \rho_{\text{reactive}}^{-\frac{1}{3}}$. Such a majority moiety will move by subdiffusive Rouse motion, since it is attached to a chain of particles [22]; the time it takes for a majority moiety to explore a ball with radius r will then be roughly $\tau_0(\frac{r}{\sigma})^4$ [22, Eq. (15)]. We can therefore find that

$$\mathbb{E}[T_{\text{int}} \mid \text{doing}] = \alpha_{\text{doing}} \tau_0 \left(\frac{r}{\sigma} \right)^4 = \alpha_{\text{doing}} \tau_0 (\sigma^3 \rho_{\text{reactive}})^{-\frac{4}{3}}, \quad (4.1.12)$$

for some constant α_{doing} . Filling in our expressions for the conditional interassociations times, Equations (4.1.12) and (4.1.5), and for the doing and undoing swap probability, Equations (4.1.11) and (4.1.10), into Equation (4.1.4), we can hence conclude that

$$\tau_{\text{int}} = \mathbb{E}[T_{\text{int}}] = \frac{\tau_0}{2 + V\rho_{\text{bonds}}} \left(\alpha_{\text{doing}} (\sigma^3 \rho_{\text{reactive}})^{-\frac{4}{3}} (1 + V\rho_{\text{bonds}}) + \alpha_{\text{undoing}} \right). \quad (4.1.13)$$

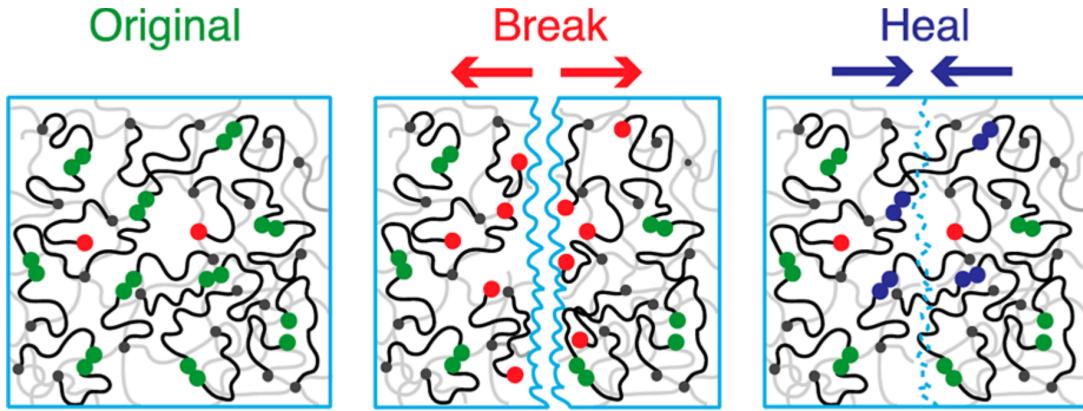


Figure 4.1: Model by Stukalin et al. for the healing of dissociative CANs. Taken from [22].

Recall that we assumed that the departing moiety explores its neighbourhood faster than a different majority moiety can find the target bond (Assumption 4.1.8). Hence, it should hold that

$$\mathbb{E}[T_{\text{int}} \mid \text{doing}] \gg \mathbb{E}[T_{\text{int}} \mid \text{undoing}], \text{ which is true when} \\ \sigma^{-3} \gg \rho_{\text{reactive}}. \quad (4.1.14)$$

This is a sensible requirement: there will be a significant chance that another majority moiety will beat the departing moiety to the bond if its neighbourhood contains any other majority moiety. Furthermore, it is straightforward to check whether this requirement is satisfied.

4.1.2 Model Fitting

We can now substitute our expression for the typical interassociation time, Equation (4.1.14), into Equation (4.1.3) to see that

$$\tau_{\text{sw}} = \frac{\tau_{\text{ass}}}{p} + \frac{\tau_0}{(2 + V\rho_{\text{bonds}})p} \left(\alpha_{\text{doing}} (\sigma^3 \rho_{\text{reactive}})^{-\frac{4}{3}} (1 + V\rho_{\text{bonds}}) + \alpha_{\text{undoing}} \right). \quad (4.1.15)$$

Equation (4.1.15) contains six parameters for which we have not theoretically derived a value: the typical association time τ_{ass} , the proper swap probability p , the self-diffusion time τ_0 ⁶, the neighbourhood volume V , and the proportionality constants α_{doing} and α_{undoing} . The values moreover would not be directly identifiable from a fit of the data using Equation (4.1.15). In this subsection, we will discuss how we can extract the greatest possible amount of information from (doing and undoing) swap time data.

We start by noting that we have other equations which we can fit to derive (functions of) the parameters. By fitting the fraction of swaps that are undoing with 4.1.10 we can determine V . Then, noting that

$$\mathbb{E}[T_{\text{sw}} \mid \text{doing}] = \frac{\tau_{\text{ass}}}{p} + \frac{1}{p} \mathbb{E}[T_{\text{int}} \mid \text{doing}] = \frac{\tau_{\text{ass}}}{p} + \frac{\alpha_{\text{doing}} \tau_0}{p} (\sigma^3 \rho_{\text{reactive}})^{-\frac{4}{3}}, \quad (4.1.16)$$

we can see that we may recover

$$\frac{\tau_{\text{ass}}}{p} \text{ and } \frac{\alpha_{\text{doing}} \tau_0}{p}$$

by fitting the average doing swap times. We can similarly find that

$$\mathbb{E}[T_{\text{sw}} \mid \text{undoing}] = \frac{\tau_{\text{ass}}}{p} + \frac{1}{p} \mathbb{E}[T_{\text{int}} \mid \text{undoing}] = \frac{\tau_{\text{ass}}}{p} + \frac{\alpha_{\text{undoing}} \tau_0}{p}; \quad (4.1.17)$$

by fitting the average undoing swap times, we can hence determine

$$\frac{\tau_{\text{ass}}}{p} + \frac{\alpha_{\text{undoing}} \tau_0}{p}, \text{ and so also } \frac{\alpha_{\text{undoing}} \tau_0}{p}.$$

The total swap time itself is perhaps not very interesting when studying self-healing: a material with a higher swap rate may take longer to heal if a greater fraction of its swaps undo the previous swap. A

⁶It should be possible to measure the self-diffusion time τ_0 in simulations, but we did not get around to this.

more useful concept would be *effective swap time*, which we could define as the time it takes for moiety to undergo one more doing than undoing swap. We may compute the mean effective swap time as follows: we define the *effectiveness* e of a doing and an undoing swap to be $+1$ and -1 respectively, so that an effective swap has occurred when the sum of effectiveness is 1 . Using the Law of Total Expectation [33, Thm. 4.6(5)], we can see that the expected effectiveness of a single swap is

$$\mathbb{E}[e] = \mathbb{E}[e \mid \text{doing}] \mathbb{P}(\text{doing}) + \mathbb{E}[e \mid \text{undoing}] \mathbb{P}(\text{undoing}) = 1 - p_{\text{undoing}} - p_{\text{undoing}} = 1 - 2p_{\text{undoing}} > 0.$$

It therefore takes on average $\frac{1}{\mathbb{E}[e]} = \frac{1}{1-2p_{\text{undoing}}}$ swaps to get an effective swap, of which $\frac{1-p_{\text{undoing}}}{1-2p_{\text{undoing}}}$ will be doing swaps and $\frac{p_{\text{undoing}}}{1-2p_{\text{undoing}}}$ will be undoing swaps.⁷ Therefore, we can define the expected effective swap time as

$$\mathbb{E}[\tau_{\text{sw,eff}}] = \frac{1-p_{\text{undoing}}}{1-2p_{\text{undoing}}} \mathbb{E}[T_{\text{sw}} \mid \text{doing}] + \frac{p_{\text{undoing}}}{1-2p_{\text{undoing}}} \mathbb{E}[T_{\text{sw}} \mid \text{undoing}]. \quad (4.1.18)$$

4.1.3 Simulations

In this subsection we will discuss the MD simulations that we have executed and analysed to test our swap time model. For this, we used Hoomd Polymer Tools, a collection of scripts which may be found on [GitLab](#):

1. The MD simulations were performed using `generating_data/measure_swap_time.py`;
2. From the resulting simulation trajectories we determined when swaps occurred using `processing_data/swap_finder.py`;
3. The empirical swap time distributions were finally calculated using `analysing_data/empirical_swap_time_distributions.py`.

Figure 4.2 shows the survival functions and densities of the swap time distributions. The legend gives the shape (star or comb) of the building blocks of the vitrimer network, followed by the number of arms N_{arms} , the number of particles in each arm n_{arm} , and, for combs, the number of particles between each arm on the backbone n_{sep} . The configurations have been sorted according to their concentration of vitrimer particles ρ_{vit} . The arrows likewise indicate the direction of increasing ρ_{vit} . The (estimates of) the survival functions were computed using a KM estimator, while the densities were estimated using KDEs⁸; for more details, see Subsection 2.1.1. We furthermore computed estimates of the error on the empirical survival curves using Greenwood's formula (Equation (2.1.4)). Since the estimated standard deviations were all less than 10^{-3} , it would not be possible to see the error bars; they have therefore been omitted.

For the most part, the survival curves move up monotonically when decreasing the concentration of reactive groups. This corresponds to the reduction of the short-time peaks and the fattening of the tails of the density curves in Figure 4.2 b) as the concentration of reactive groups decreases. There is one exception, however: vitrimer networks made of combs with $N_{\text{arms}} = 13$, $n_{\text{arm}} = 4$, and $n_{\text{sep}} = 4$ have significantly longer swap times than other vitrimer networks with a similar concentration of reactive groups (e.g. Comb: 16, 5, 3). We were not able to find a satisfactory explanation for this anomaly.

As we noted in Subsection 4.1.1, we can classify swaps as being either *undoing*, if they undo the previous swap, or otherwise as *doing* swaps. However, we have in essence a four parameter model to fit on just 33 data points. It would therefore be questionable to conclude the model is reliable based solely on it fitting the data on which it was fitted well. To deal with this issue, we have added an additional configuration: a vitrimer network made of comb polymers with $N_{\text{arms}} = 6$, $n_{\text{arm}} = 10$, and $n_{\text{sep}} = 10$. In this network, the concentration of vitrimer particles is significantly lower than in all other networks we considered. We can therefore investigate how well our model extrapolates beyond its observations.

Figure 4.3 a) shows how the fraction of undoing swaps p_{undoing} depends on the concentration of vitrimer particles ρ_{vit} . The solid data points were used to fit Equation (4.1.10); the unfilled data points correspond

⁷this derivation is somewhat shaky, because we are ignoring the dependence between the type of subsequent swaps.

⁸All KDEs shown in this subsection were computed with Gaussian kernels with standard deviation 1 [time] – chosen by eye.

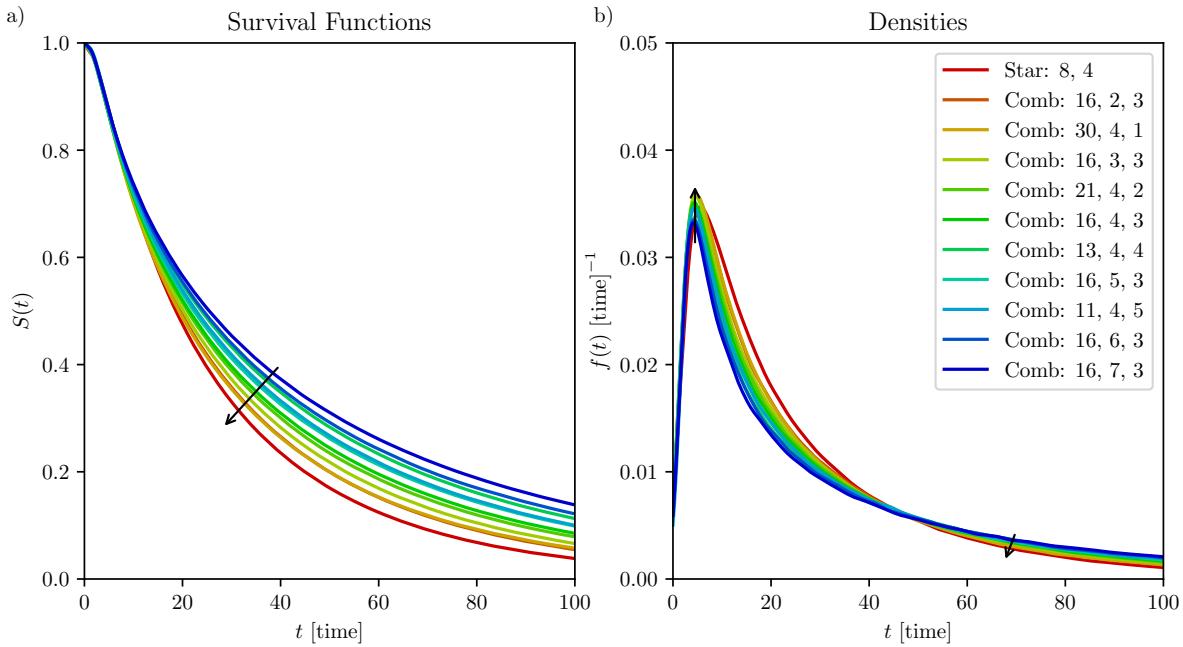


Figure 4.2: KM estimates of survival curves and KDEs for swap times in vitrimer networks with various building blocks, sorted by concentration of reactive groups.

to the network made of comb polymers with $N_{\text{arms}} = 6$, $n_{\text{arm}} = 10$, and $n_{\text{sep}} = 10$. It was found that $V = 93.3 \text{ [length]}^3$, so that $V^{\frac{1}{3}} = 4.5 \text{ [length]} \sim \sigma$, as expected. The model fits the training data fairly well, and extrapolates decently: on the ‘unseen’ data point it is off by about 7.8×10^{-3} , or 2.5 %. The main outlier is the point with the highest concentration of vitrimer bonds, which corresponds to the network made of star polymers with $N_{\text{arms}} = 8$, $n_{\text{arm}} = 4$. This error might be caused by the geometry of the star polymers: it would not be surprising if the way in which bonds are distributed around a majority moiety is different in stars than in combs.

In Figure 4.3 b), the KM estimates of the mean swap times – computed according to Definition 2.1.8 – are depicted using red circles; the green squares and blue triangles represent the KM estimates of the mean swap times conditioned on the swaps being doing and undoing, respectively; the blue stars correspond to the effective swap times, calculated using Equation (4.1.18). Once again, the models were fitted on the filled data points. By fitting the doing swap times with Equation (4.1.16), we found that

$$\frac{\tau_{\text{ass}}}{p} = 18.6 \text{ [time]} \text{ and } \frac{\alpha_{\text{doing}} \tau_0}{p} = 8.6 \times 10^{-2} \text{ [time].}$$

In our simulations we additionally found that the proper swap probability is $p \approx 1$ for all configurations, so that $\tau_{\text{ass}} \sim 10^1 \text{ [time]}$, which does not appear unreasonable. The green curve passes nicely through the data it was fitted on. It underestimates the doing swap time of the ‘unseen’ configuration by 8.6 %, which is not too bad.

On the other hand, the fit for the undoing swap times – computed using Equation (4.1.17) – underestimates the ‘unseen’ data point by 37 %. We can see that the model is oversimplified: the fitted undoing swap times appear to decrease slightly with increasing concentration, whereas we expected them to remain constant. This could again be related to Assumption 4.1.8. To see this, consider a departing moiety that will return to the bond it left. The longer it takes to do so, the greater the chance will be that a different majority moiety will associate with the bond first, preventing an undoing swap. If ρ_{reactive} increases, more undoing swaps will be prevented, and long undoing swaps are particularly likely to be prevented. Hence, if Assumption 4.1.8 does not hold, the mean undoing swap time will decrease as we increase ρ_{vit} . From the fit, and the previously found value of $\frac{\tau_{\text{ass}}}{p}$, we could derive that

$$\frac{\alpha_{\text{undoing}} \tau_0}{p} = -2.9 \text{ [time].}$$

This does not make sense, since $\alpha_{\text{undoing}}, \tau_0, p \geq 0$. We found two explanations for this observation. One possibility is that the association time τ_{ass} and/or the proper swap probability p are not the same for

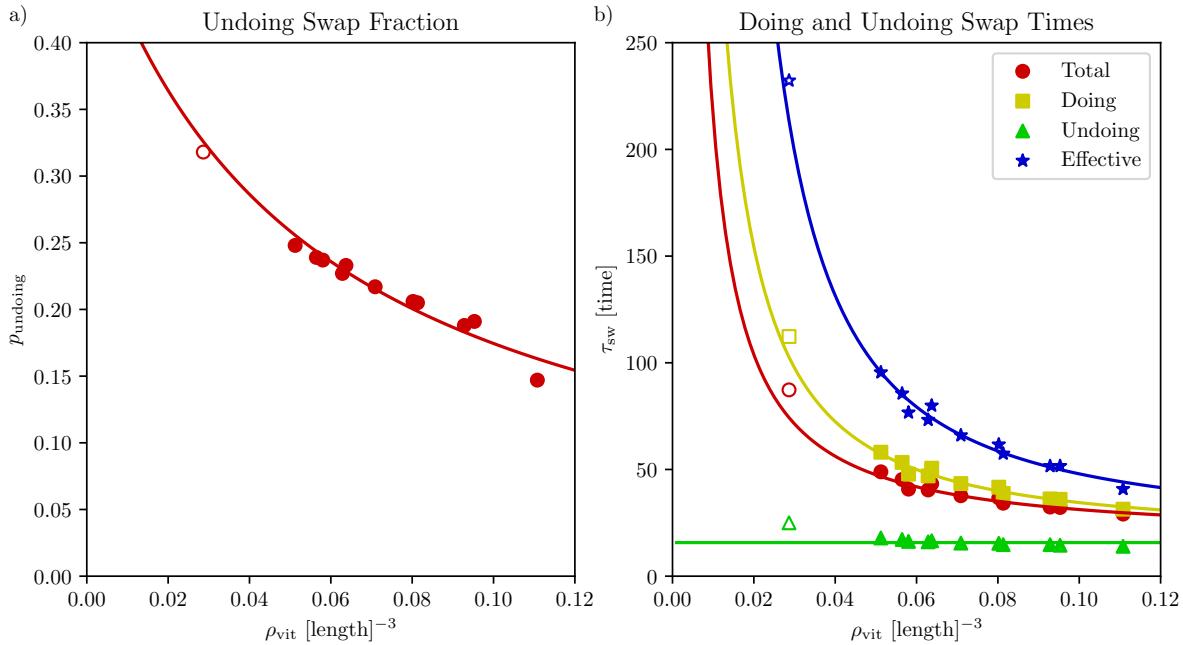


Figure 4.3: The dependence of the mean swap times and the fraction of undoing swaps on the concentration of reactive groups. Filled markers correspond to data points which were used for fitting, while unfilled data points serve to investigate how well the models extrapolate.

doing and undoing swaps; we could not find any a priori reason why either of these would depend on the type of swap. It is also possible that we overestimated $\frac{\tau_{\text{ass}}}{p}$ when fitting Equation (4.1.16). Indeed, we found in our simulations that 99.8 % of the associations lasted less than 5×10^{-2} [time].⁹ This would notably imply that there must be an additional process that is independent of ρ_{vit} that happens for doing swaps but not for undoing swaps; otherwise we would expect the doing and undoing swap time to both tend to $\frac{\tau_{\text{ass}}}{p}$ as $\rho_{\text{vit}} \rightarrow \infty$, which does not appear to be the case.

We subsequently fitted the total swap times by combining our other fits according to the Law of Total Expectation [33, Thm. 4.6(5)]:

$$\begin{aligned} \tau_{\text{sw}} &= \mathbb{E}[T_{\text{sw}}] = \mathbb{E}[T_{\text{sw}} \mid \text{doing}] \mathbb{P}(\text{doing}) + \mathbb{E}[T_{\text{sw}} \mid \text{undoing}] \mathbb{P}(\text{undoing}) \\ &= \mathbb{E}[T_{\text{sw}} \mid \text{doing}] (1 - p_{\text{undoing}}) + \mathbb{E}[T_{\text{sw}} \mid \text{undoing}] p_{\text{undoing}}. \end{aligned}$$

On the fitted data, the model performs rather well, since the errors in the fits for p_{undoing} and the undoing swap times somewhat cancel each other out. On the other hand, since our fit for the undoing swap times performs so poorly, the combined fit underestimates the swap time of the ‘unseen’ configuration by 15 %. We similarly fitted the effective swap times by combining our fits using Equation (4.1.18); the error in the prediction was only 8.2 %. This is again caused by the fact that the fit overestimates p_{undoing} , while underestimating the doing and undoing swap times.

Figure 4.3 shows only point estimates; to get a better idea of how the distributions of doing and undoing swap times differ locally, we look at their KDEs in Figure 4.4. From Figure 4.4 a) we can see how the increase in typical doing swap time with decreasing vitrimer particle concentration is effected through the changing of the doing swap time PDFs: the mode shifts by about 1 [time] to the right, and, more significantly, probability mass moves from the mode towards the tail.

We hypothesise that the shift in mode is caused by the fact that the bonds that are initially nearest to the departing moiety will be further away when ρ_{vit} is low. As a simple check, we will consider the configuration with the highest concentration of vitrimer particles, the network made of stars with $N_{\text{arms}} = 8$ and $n_{\text{arm}} = 4$, and the configuration with the lowest concentration of vitrimer particles, the network made of combs with $N_{\text{arms}} = 16$, $n_{\text{arm}} = 7$, and $n_{\text{sep}} = 3$. The fraction of their vitrimer concentrations is 4.64×10^{-1} , which implies that the fraction of the distances of a departing moiety to its

⁹the time resolution of the analysed trajectories, corresponding to 50 simulation timesteps.

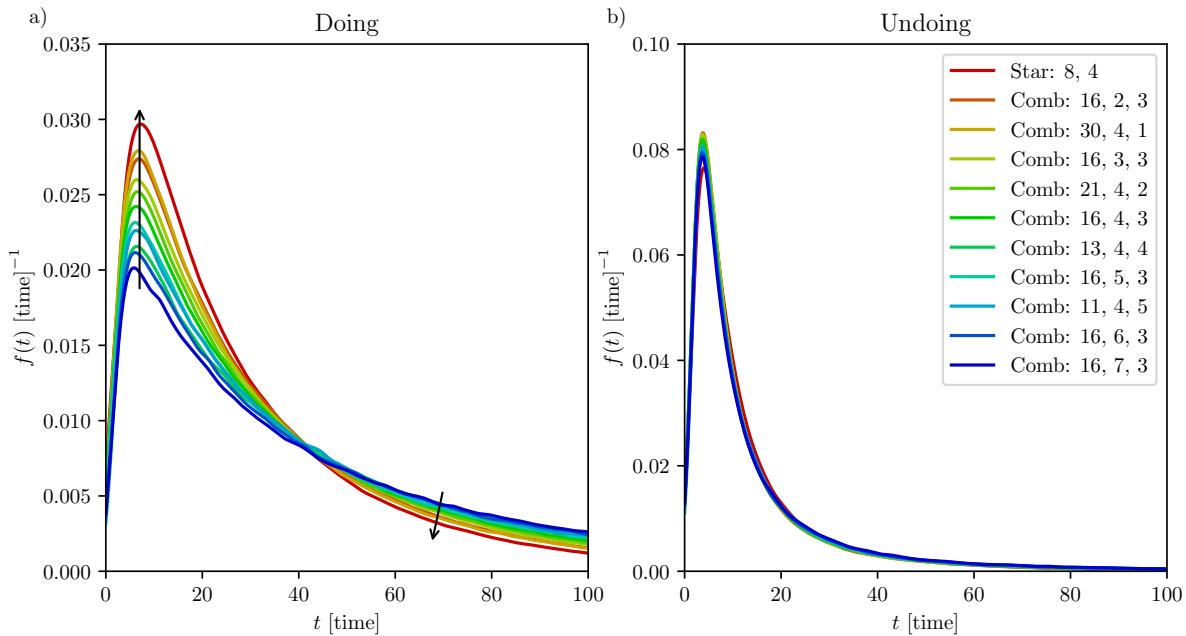


Figure 4.4: KDEs for doing and undoing swap times in vitrimer networks with various building blocks, sorted by concentration of reactive groups.

closest neighbouring vitrimer bonds will be 1.29. For the comb network, the mode occurs at 7.20 [time], while for the star network the mode occurs at 5.85 [time]; the fraction of their modes is therefore 1.23. Hence, our hypothesis seems reasonable. We can similarly attempt to explain why probability mass moves to the tail as ρ_{vit} decreases. We can imagine that a departing moiety might ‘miss’ its nearest neighbouring vitrimer bonds, so that it has to find a bond further away to associate with. The probability that this will happen will clearly be greater if the concentration of bonds is lower, since then there will be fewer close vitrimer bonds. Using the same comb and star polymer vitrimer networks as before, we can see that any fixed neighbourhood will contain 2.16 times as many bonds in the network made of stars than in the network made of combs, while the density at the mode is only 1.47 times as high. Consequently, we have only found part of the explanation, and there must be some other process that contributes significantly to shifting the probability mass from the mode to the tail.

We next look at the distributions of undoing swap times. In Subsection 4.1.1 we postulated that the undoing swap times would not depend on the concentration of vitrimer particles, and in Figure 4.3 we saw that this is mostly true for the mean of the distribution. Figure 4.4 b) shows that in fact the entire distribution coincide, since the KDEs of the undoing swap times practically lie on top of each other.

To finish off our study of swap times, we will now investigate whether the swap times may be described by a well-known distribution, such as the Exponential distribution. Since for the most common lifetime distributions the hazard function is very simple, it makes sense to look at the KHEs of the swap time distributions. We must bear in mind, however, that estimating the hazard function, especially in the tail, is hard.

We have plotted the KHEs of the doing and undoing swap times in Figure 4.5 a). The dashed curves are the ‘fits’, given by the hazard functions of exponential random variables with the same means. Since the fits are so poor, it is immediately evident that neither the doing nor the undoing swap times are well-described by an exponential distribution. In particular, whereas the hazard function of an exponential random variable is constant and so remains positive at $t = 0$, the true hazard functions go to zero at small times. This corresponds physically to the fact that a majority moiety must travel some nonzero distance at a finite¹⁰ speed.

It is therefore not surprising that the total swap times do not follow an exponential distribution either, see Figure 4.5 b). The green dashed line, which we call the naive fit, is again the hazard function of

¹⁰even bounded.

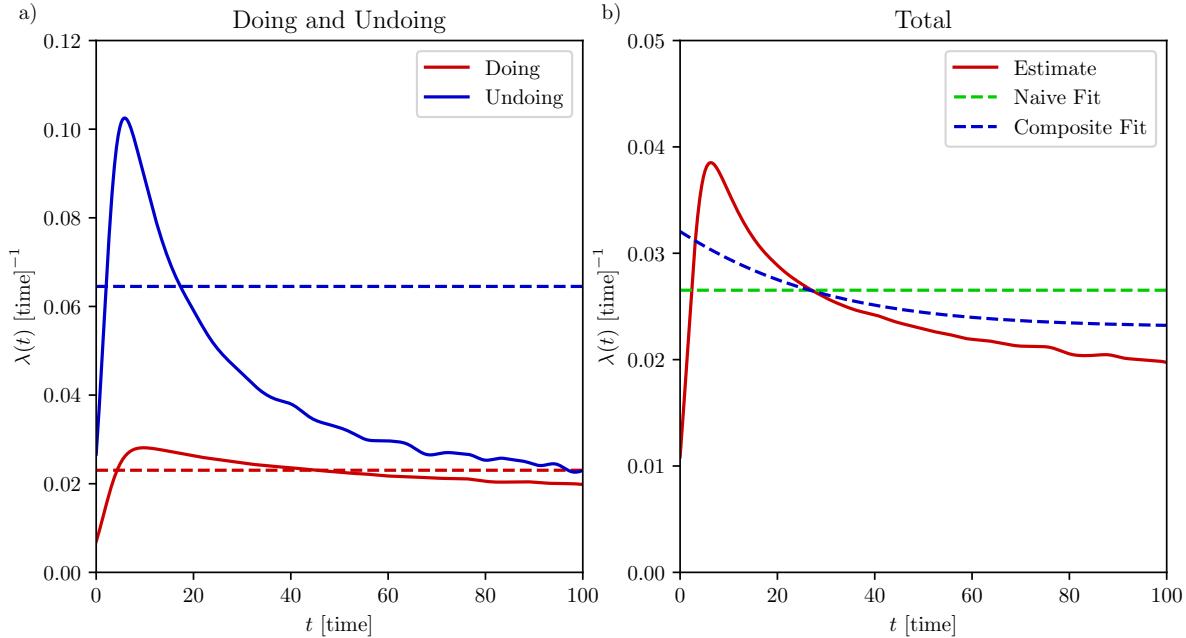


Figure 4.5: KHEs for total, doing, and undoing swap times, drawn as solid curves, with as fits the hazard function of an exponential random variable with the corresponding mean, drawn as dashed curves. Composite fit is the hazard function of the p_{undoing} -weighted sum of the doing and undoing exponential random variables.

an exponential random variable, this time with mean τ_{sw} : the observed hazard function is completely different. The fact that the swap times are not exponentially distributed has been observed before, both in pure MD simulations similar to our own [75] and in hybrid MD-MC simulations [54]. It is worth noting that Rovigatti et al. investigated a binary system of vitrimer particles at a volume fraction of 0.38 [75], while Wu et al. studied a mixture of star and linear polymers at a volume fraction of 0.58 [54]: the nonexponential nature of the swap time distribution therefore seems to be a rather general property of vitrimer networks, perhaps because a bond exchange involves two successive processes – an association and an interassociation. The blue dashed curve, which we call the composite fit, was computed using

$$\lambda(t) = \frac{f(t)}{S(t)} = \frac{f_{\text{doing}}(t)(1 - p_{\text{undoing}}) + f_{\text{undoing}}(t)p_{\text{undoing}}}{S_{\text{doing}}(t)(1 - p_{\text{undoing}}) + S_{\text{undoing}}(t)p_{\text{undoing}}}.$$

The composite fit performs slightly better than the naive fit, but still clearly misses important features of the distribution. It is moreover clear from Figure 4.5 that Weibull distributions would not be appropriate either: in that case we would expect the hazard functions to follow power laws – see Equation (2.1.10) – which the true hazard functions do not. We can see this e.g. since the KHEs are not monotone functions of t . We have been unable to find a family of distributions that can appropriately describe our swap times.

The fact that we see that the swap times do not follow a Weibull distribution contradicts the findings of Wu et al., who found that the bond autocorrelation function – which may be interpreted as the survival function of the swap times – is well-fitted by a stretched exponential [54]. This contradiction could be explained by the fact that we have simulated rather different vitrimer networks: our networks are made of combs or stars with arms consisting of roughly four particles, while the networks investigated by Wu et al. consist of linear chains that are five particles long and stars with four arms each containing a single particle; it is evident that the dynamics of the reactive moieties, and therefore also the swap times, will differ in the two systems. It is also conceivable, however, that the discrepancy can be explained by the differences in implementation of the bond exchanges. In particular, the association of reactive moieties is determined with a MC algorithm in [54], while we model associations with a three-body potentials. Finally, the systems we have simulated have a volume fraction of 0.29, while the volume fraction of those considered by Wu et al. was twice as high at 0.58;¹¹ our systems are far away from the glass

¹¹Density $\rho = 1.1/\sigma^3$ [54], volume per spherical particle $4\pi(\sigma/2)^3/2$, so volume fraction $1.1 \cdot 4\pi/24 \approx 0.58$.

transition, while their systems are near to the glass transition [54]. The super cooled nature of their vitrimer networks could inhibit certain bond swap mechanisms which make the swap time distributions more complex at lower densities.

4.1.4 Conclusions

In short, by classifying swaps as being either doing or undoing, we were able to predict that the swap times would grow as the concentration of vitrimer particles decreases. We found simple models to calculate the typical doing and undoing swap times, as well as the probability that a swap will be undoing. We saw that the model for the mean doing swap times extrapolates decently beyond data that it was fitted on. The model for the undoing swap probability performed reasonably well, although there was a sizable error residual with the network made of star polymers, which seems to suggest that the building block morphology plays an important additional role beyond just determining the concentration of vitrimer particles. Conversely, the model for the undoing swap times clearly fails to reproduce key features of the dependence on the concentration of vitrimer particles, as evidenced by its poor extrapolation properties, perhaps in part due to Assumption 4.1.8. Since the undoing swap time is an important factor in the total and effective swap times, we believe further development of an undoing swap time model would be valuable. By looking at the KDEs of the doing swap times, we were able to come up with physical mechanisms by which the typical doing swap time increases with decreasing concentration of vitrimer particles. Finally, by analysing the KHEs of the swap time distributions we found that neither of the two standard families of lifetime distributions, the exponential and Weibull distributions, can describe the doing, undoing, or total swap times, in agreement with [75], but contradicting [54]; the latter conflict may be due e.g. to differences in how the bond exchanges are modelled. It might be possible to partition the swaps into even more classes, such that the distribution of each of these classes is comparatively simple.

4.2 Healing Swaps

We will now develop a model to study the swaps that contribute to the healing of damage in a vitrimer network. We will consider a section of damage with finite area A . In Figure 4.6, we show a schematic depiction of the volume around such damage. In white, labelled ‘interface’, we have the *healing interface*. On each side of the healing interface there is vitrimeric material, which we assume extends indefinitely in each direction. In light grey, labelled ‘bulk’, we have that part of the material that is sufficiently close to the healing interface that it may contribute to the healing through bond swaps; we then assume that there is a pool of N_0 minority moieties that could form bonds across the healing interface. Finally, we denote the number of bonds across the healing interface at time $t > 0$ by $N_{\text{int}}(t)$, and the number of bonds in the bulk by $N_{\text{bulk}}(t)$. Notably it therefore holds that $N_0 = N_{\text{int}}(t) + N_{\text{bulk}}(t)$ for $t > 0$.

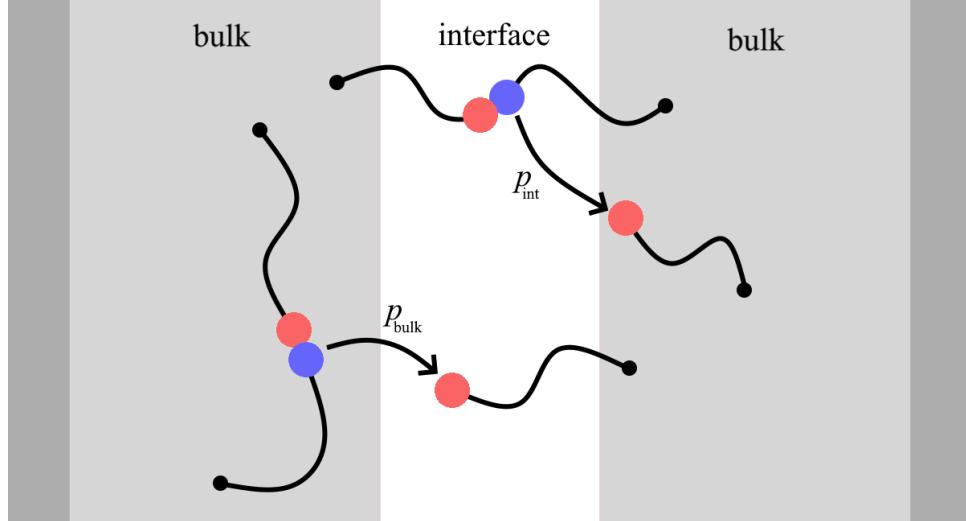


Figure 4.6: Schematic depiction of the healing interface, and how bond swaps can contribute to the healing.

We will study the number of bonds across the healing interface $N_{\text{int}}(t)$ using rate equations. To make such an analysis feasible, we make the following assumption.

Assumption 4.2.1. The swap times of different moieties do not depend on each other.

Assumption 4.2.1 allows us to now consider the behaviour of single minority moiety. It makes sense to quantify the evolution of the network using the effective swap time, Equation (4.1.18). Hence, our investigation will involve effective swaps; to reduce notational clutter, we will from now on simply call these swaps.

Suppose that such a minority moiety in the bulk undergoes a bond swap. After this swap there are then two possibilities: either the moiety is still part of a bond in the bulk, or it is now part of a bond across the healing interface. Similarly, a minority moiety in a bond across the healing interface can swap to another bond across the healing interface, or a bond in the bulk. To simplify our analysis, we will now assume that all bonds behave in an identical and time-invariant way:

Assumption 4.2.2. A bond across the healing interface gets swapped to a bond within the bulk with probability $p_{\text{int}} \in (0, 1)$, while a bond within the bulk gets swapped to a bond across the healing interface with probability $p_{\text{bulk}} \in (0, 1)$.

It is then easily seen that a bond across the healing interface gets swapped to another bond across the healing interface with probability $1 - p_{\text{int}}$ and a bond within the bulk gets swapped to another bond within the bulk with probability $1 - p_{\text{bulk}}$. There are no a priori obvious candidates for the values of p_{int} and p_{bulk} .

4.2.1 Rate Equations

To define rate equations, we obviously need to know reaction rates. If we are dealing with processes that are characterised by exponentially distributed times, it makes sense to speak of reaction rates, since the hazard functions will be constant. However, if the times are not exponentially distributed, the hazard functions will not be constant, and so the reaction rates will be ill-defined. We will therefore make the following additional assumption:

Assumption 4.2.3. The swap times are exponentially distributed with mean $\tau_{\text{sw,eff}}$.

We have good reason to doubt Assumption 4.2.3: we know in equilibrium that the swap times are definitely not exponentially distributed, see Subsection 4.1.3. It is, however, a necessary assumption to be able to properly define the rate equations. It may be possible to analyse how the number of interfacial bonds changes over time without needing Assumption 4.2.3 if we use a different approach. For instance, we could perform a simple MC experiment, in which we keep track of the state of each bond, and randomly sample swap times from any distribution we would like. It would be interesting to analyse how strongly the evolution of the expected number of bonds across the healing interface depends on the shape of the swap time distribution. We will, however, content ourselves with solving rate equations set up under Assumption 4.2.3.

Lemma 4.2.4 (Rate Equation for Bonds Across Healing Interface). *The evolution of the number of bonds across the interface N_{int} over time is governed by the rate equation*

$$\frac{dN_{\text{int}}}{dt}(t) = \frac{p_{\text{bulk}}}{\tau_{\text{sw,eff}}} N_0 - \frac{p_{\text{bulk}} + p_{\text{int}}}{\tau_{\text{sw,eff}}} N_{\text{int}}(t), \quad (4.2.1)$$

with the initial condition

$$N_{\text{int}}(0) = 0. \quad (4.2.2)$$

Proof. Equation (4.2.2) defines the natural initial condition since there will be no bonds across the healing interface at time $t = 0$.

There are N_0 minority moieties that can participate in bond swaps. Since each moiety does so independently at rate $\tau_{\text{sw,eff}}^{-1}$, in total swaps occur at rate $N_0 \tau_{\text{sw,eff}}^{-1}$. We now fix time t , and consider what the next swap will be. Since all of the bonds have i.i.d. exponential swap times, we can apply the

‘exponential splitting property’ – Theorem A.3.3 – to see that the swap involves a minority moiety in the healing interface with probability (w.p.) $\frac{N_{\text{int}}(t)}{N_0}$. Such a swap will decrease the number of bonds across the healing interface w.p. p_{int} . It clearly follows that the next swap involves a minority moiety in the bulk w.p. $\frac{N_{\text{bulk}}(t)}{N_0}$, and increases the number of bonds across the healing interface w.p. p_{bulk} . We can therefore compose the following simple rate equation:

$$\begin{aligned}\frac{dN_{\text{int}}}{dt}(t) &= \frac{N_0}{\tau_{\text{sw,eff}}} \left(-1 \cdot \frac{N_{\text{int}}(t)}{N_0} \cdot p_{\text{int}} + 1 \cdot \frac{N_{\text{bulk}}(t)}{N_0} \cdot p_{\text{bulk}} \right) = \frac{p_{\text{bulk}}}{\tau_{\text{sw,eff}}} N_{\text{bulk}}(t) - \frac{p_{\text{int}}}{\tau_{\text{sw,eff}}} N_{\text{int}}(t) \\ &= \frac{p_{\text{bulk}}}{\tau_{\text{sw,eff}}} (N_0 - N_{\text{int}}(t)) - \frac{p_{\text{int}}}{\tau_{\text{sw,eff}}} N_{\text{int}}(t) = \frac{p_{\text{bulk}}}{\tau_{\text{sw,eff}}} N_0 - \frac{p_{\text{bulk}} + p_{\text{int}}}{\tau_{\text{sw,eff}}} N_{\text{int}}(t),\end{aligned}$$

which gives us exactly Equation (4.2.1). \square

We finally extract the time evolution of N_{int} from Rate Equation (4.2.1) in the following theorem.

Theorem 4.2.5 (Evolution of Number of Bonds Across Healing Interface). *The number of bonds across the healing interface at time $t > 0$ is given by*

$$N_{\text{int}}(t) = \frac{p_{\text{int}}}{p_{\text{bulk}} + p_{\text{int}}} N_0 \left(1 - \exp \left(-(p_{\text{bulk}} + p_{\text{int}}) \frac{t}{\tau_{\text{sw,eff}}} \right) \right). \quad (4.2.3)$$

Proof. According to Lemma 4.2.4, the number of bonds across the healing interface is governed by

$$\frac{dN_{\text{int}}}{dt}(t) = \frac{p_{\text{bulk}}}{\tau_{\text{sw,eff}}} N_0 - \frac{p_{\text{bulk}} + p_{\text{int}}}{\tau_{\text{sw,eff}}} N_{\text{int}}(t),$$

with the initial condition

$$N_{\text{int}}(0) = 0.$$

We can easily solve Rate Equation (4.2.1) as follows:

$$\int_0^t dt = \int_{N_{\text{int}}(0)}^{N_{\text{int}}(t)} \frac{dN_{\text{int}}(s)}{\frac{p_{\text{bulk}}}{\tau_{\text{sw,eff}}} N_0 - \frac{p_{\text{bulk}} + p_{\text{int}}}{\tau_{\text{sw,eff}}} N_{\text{int}}(s)} = - \frac{\tau_{\text{sw,eff}}}{p_{\text{bulk}} + p_{\text{int}}} \int_{N_{\text{int}}(0)}^{N_{\text{int}}(t)} \frac{dN_{\text{int}}(s)}{N_{\text{int}}(s) - \frac{p_{\text{int}}}{p_{\text{bulk}} + p_{\text{int}}} N_0}.$$

Substituting $u = N_{\text{int}}(t) - \frac{p_{\text{int}}}{p_{\text{bulk}} + p_{\text{int}}} N_0$, noting that $N_{\text{int}}(0) = 0$, we find that

$$\begin{aligned}-(p_{\text{bulk}} + p_{\text{int}}) \frac{t}{\tau_{\text{sw,eff}}} &= \int_{N_{\text{int}}(0)}^{N_{\text{int}}(t)} \frac{dN_{\text{int}}(s)}{N_{\text{int}}(s) - \frac{p_{\text{int}}}{p_{\text{bulk}} + p_{\text{int}}} N_0} = \int_{-\frac{p_{\text{int}}}{p_{\text{bulk}} + p_{\text{int}}} N_0}^{N_{\text{int}}(t) - \frac{p_{\text{int}}}{p_{\text{bulk}} + p_{\text{int}}} N_0} \frac{du}{u} \\ &= [\log|u|]_{-\frac{p_{\text{int}}}{p_{\text{bulk}} + p_{\text{int}}} N_0}^{N_{\text{int}}(t) - \frac{p_{\text{int}}}{p_{\text{bulk}} + p_{\text{int}}} N_0} = \log \left(1 - \frac{p_{\text{bulk}} + p_{\text{int}}}{p_{\text{int}}} \frac{N_{\text{int}}(t)}{N_0} \right).\end{aligned}$$

Taking the exponential on both sides and rewriting gives us

$$N_{\text{int}}(t) = \frac{p_{\text{int}}}{p_{\text{bulk}} + p_{\text{int}}} N_0 \left(1 - \exp \left(-(p_{\text{bulk}} + p_{\text{int}}) \frac{t}{\tau_{\text{sw,eff}}} \right) \right),$$

as required. \square

It is not hard to derive from Equation (4.2.3) that N_{int} tends to a horizontal asymptote:

Corollary 4.2.6 (Asymptotic Number of Bonds Across Healing Interface). *The number of bonds across the healing interface over time tends to*

$$N_{\text{int}}(\infty) := \lim_{t \rightarrow \infty} N_{\text{int}}(t) = \frac{p_{\text{int}}}{p_{\text{bulk}} + p_{\text{int}}} N_0. \quad (4.2.4)$$

Theorem 4.2.5 tells us how N_{int} will evolve over time. If we define the typical healing time τ_h to be the time it takes until a certain fraction (say $1 - 1/e$) of $N_{\text{int}}(\infty)$ is reached, then we may identify from Equation (4.2.3) that

$$\tau_h = \frac{\tau_{\text{sw,eff}}}{p_{\text{bulk}} + p_{\text{int}}}. \quad (4.2.5)$$

Hence, we can speed up self-healing by increasing the swap rate; from Section 4.1 we know that we can do this by increasing the concentration of vitrimer particles.

Equation (4.2.3) contains two fit parameters:

$$\frac{p_{\text{int}}}{p_{\text{bulk}} + p_{\text{int}}} N_0 \text{ and } \frac{\tau_{\text{sw,eff}}}{p_{\text{bulk}} + p_{\text{int}}}. \quad (4.2.6)$$

We have already determined $\tau_{\text{sw,eff}}$ independently in Section 4.1. On the other hand, N_0 , p_{int} , and p_{bulk} are somewhat ill-defined. The most concrete definition of N_0 would be the number of minority moiety that can form bonds across the healing interface. Clearly, a minority moiety within the ‘bulk’ that is far away from the healing interface is less likely swap into an interfacial bond than a minority moiety that is closer to the healing interface. Hence, we should view p_{bulk} and p_{int} as being averaged over the volume defined by N_0 . Determining N_0 a priori in an exact way seems impossible: we can either look a posteriori at which minority moieties have been part of interfacial bonds, or estimate the size of the ‘bulk’ – the volume which contains minority moieties that are sufficiently close to the healing interface. Here, we will estimate N_0 using the second approach.

First, note that a minority moiety can be a part of an interfacial bond if it can be reached by a majority moiety from the other side of the healing interface. Since we assume the backbones of the vitrimer network are stationary, this is the case if the spherical volume traced by the minority moiety overlapping with that of some majority moiety across the healing interface. The radius R of the volume will evidently depend on the bond length σ and the number of bonds in the chain n_{arm} . It is well-established in polymer theory that an ideal chain with $n_{\text{arm}} \gg 1$ will have length $R \sim \sqrt{n_{\text{arm}}}\sigma$ [50, Eq. (2.168)]. However, we are dealing with chains with $n_{\text{arm}} = \mathcal{O}(1)$; there will therefore be a nonnegligible amount of probability mass beyond $\sqrt{n_{\text{arm}}}\sigma$ in the chain length distribution. Hence, we will choose $R = n_{\text{arm}}\sigma$, so that a minority moiety belongs to the ‘bulk’ if it is at a distance of at most $2R = 2n_{\text{arm}}\sigma$ to the healing interface. We can then find that the volume of the ‘bulk’ is $4An_{\text{arm}}\sigma$, from which it directly follows that

$$N_0 = V_{\text{bulk}} \cdot \rho_{\text{bonds}} = 4An_{\text{arm}}\sigma\rho_{\text{vit}}f_{\text{A}_2}. \quad (4.2.7)$$

In theory we can now extract p_{int} and p_{bulk} from the fit parameters equation (4.2.6), since

$$\begin{aligned} p_{\text{int}} &= \frac{1}{N_0} \frac{p_{\text{int}}}{p_{\text{bulk}} + p_{\text{int}}} N_0 \cdot \tau_{\text{sw,eff}} \frac{p_{\text{bulk}} + p_{\text{int}}}{\tau_{\text{sw,eff}}}, \text{ and} \\ p_{\text{bulk}} &= \tau_{\text{sw,eff}} \frac{p_{\text{bulk}} + p_{\text{int}}}{\tau_{\text{sw,eff}}} - p_{\text{int}}, \end{aligned}$$

and we know all the terms on the right hand side of these expressions.

4.2.2 Equilibrium Shear Modulus

Up to now, we have attempted to assess the healing of a vitrimer material by counting the number of bonds across the healing interface. It is, however, not directly clear how the mechanical strength of the material depends on this number of bonds. Moreover, we cannot yet choose some benchmark number above which we can say that the material is sufficiently healed. As noted in Section 2.2, mechanical strength can be quantified using the equilibrium value of the shear modulus. Hence, we will now develop a simple model for the shear modulus of our system in terms of the number of bonds across the healing interface. Our model will contain two healing interfaces, to align with the simulations we will perform in Subsection 4.2.3, in which our sample vitrimer will be in a periodic box with two healing interfaces in the unit cell.

Figure 4.7 a) schematically depicts a unit cell in our simulations, with in light grey the healing interfaces and in dark grey the bulk. We assume that the healing interfaces and bulk are homogeneous; each has a well-defined equilibrium shear modulus, G_{int} and G_{bulk} , respectively. As a consequence, the total unit cell will have an effective equilibrium shear modulus $G_{\text{eff}} \in [G_{\text{int}}, G_{\text{bulk}}]$. Additionally, we assume that

the healing interfaces each have a thickness of 2δ , while the entire unit cell has a thickness of h . Finally, we freeze the network topology: the associative bonds are replaced by normal covalent bonds. This is reasonable as long as we look at sufficiently short timescales, since stress relaxation by bond exchanges is much slower than the relaxation towards the elastic plateau, and indeed much slower than healing [1]. To simplify the analysis, we will now work with the mechanically equivalent unit cell shown in Figure 4.7 b); we have fused the healing interfaces and bulks, so that the unit cell consists of now two contiguous, homogeneous blocks. Finally, Figure 4.7 c) shows how this equivalent unit cell will have deformed once

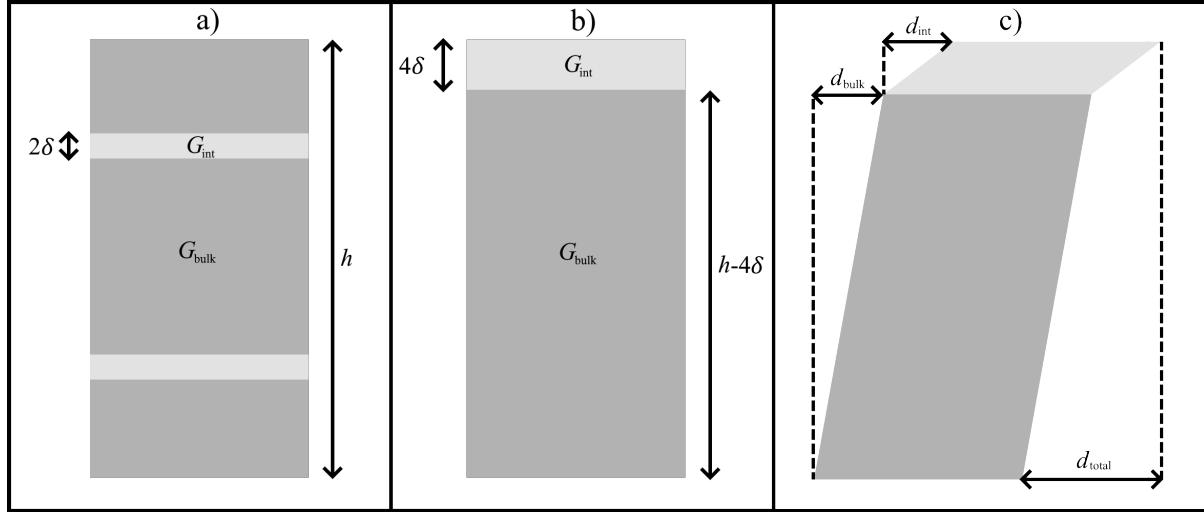


Figure 4.7: (a) Schematic depiction of unit cell, consisting of bulk areas (dark) with equilibrium shear modulus G_{bulk} , and interfacial areas (light) with equilibrium shear modulus G_{int} . (b) Mechanically equivalent cell, where the areas of bulk and interface have been combined into two areas, respectively. (c) Upon shearing, the bulk and interface areas deform with different strains.

it has reached the elastic plateau after a step strain $\gamma_{\text{total}} = \frac{d_{\text{total}}}{h}$ has been applied. By assumption, the system is in (mechanical) equilibrium in the elastic plateau, so that in particular the stress is uniform in the unit cell [80, p. 7]. Recall from Equation (2.2.2) that by definition it holds that $\gamma = \frac{G}{\sigma}$. Consequently, when the entire unit cell is sheared with strain γ_{total} , the bulk and the healing interface will shear with strains

$$\gamma_{\text{bulk}} = \frac{\sigma}{G_{\text{bulk}}} \neq \gamma_{\text{total}} \quad \text{and} \quad \gamma_{\text{int}} = \frac{\sigma}{G_{\text{int}}} \neq \gamma_{\text{total}},$$

respectively. It can then easily be seen that

$$G_{\text{bulk}}\gamma_{\text{bulk}} = G_{\text{int}}\gamma_{\text{int}},$$

so that

$$\gamma_{\text{bulk}} = \frac{G_{\text{int}}}{G_{\text{bulk}}}\gamma_{\text{int}}. \quad (4.2.8)$$

Now note that the total deformation is the sum of the deformations of the constituent parts, i.e.

$$d_{\text{total}} = d_{\text{bulk}} + d_{\text{int}}.$$

We can also relate the deformations to the strains through Equation (2.2.1):

$$d_{\text{total}} = \gamma_{\text{total}}h, \quad d_{\text{bulk}} = (h - 4\delta)\gamma_{\text{bulk}}, \quad \text{and} \quad d_{\text{int}} = 4\delta\gamma_{\text{int}},$$

which implies that

$$\gamma_{\text{total}}h = (h - 4\delta)\gamma_{\text{bulk}} + 4\delta\gamma_{\text{int}}.$$

We can then apply Equation (4.2.8) to see that

$$\gamma_{\text{total}}h = \left((h - 4\delta)\frac{G_{\text{int}}}{G_{\text{bulk}}} + 4\delta \right) \gamma_{\text{int}}.$$

It finally follows that

$$\gamma_{\text{int}} = \frac{h}{(h - 4\delta)\frac{G_{\text{int}}}{G_{\text{bulk}}} + 4\delta} \gamma_{\text{total}}. \quad (4.2.9)$$

As a sanity check, we now look at what happens to the strain of the healing interface as its shear modulus is varied. When no healing has occurred, there will be no bonds across the healing interface, and we expect $G_{\text{int}} = 0$. We then predict a strain

$$\lim_{G_{\text{int}} \rightarrow 0} \gamma_{\text{int}} = \lim_{G_{\text{int}} \rightarrow 0} \frac{h}{(h - 4\delta) \frac{G_{\text{int}}}{G_{\text{bulk}}} + 4\delta} \gamma_{\text{total}} = \frac{h}{4\delta} \gamma_{\text{total}}.$$

This makes sense, since it implies that $d_{\text{int}} = d_{\text{total}}$ and $d_{\text{bulk}} = 0$: the entire deformation occurs in the healing interface. On the other hand, when the material is fully healed, the healing interface will be indistinguishable from the bulk, so we expect $G_{\text{int}} = G_{\text{bulk}}$. We then predict a strain

$$\lim_{G_{\text{int}} \rightarrow G_{\text{bulk}}} \gamma_{\text{int}} = \lim_{G_{\text{int}} \rightarrow G_{\text{bulk}}} \frac{h}{(h - 4\delta) \frac{G_{\text{int}}}{G_{\text{bulk}}} + 4\delta} \gamma_{\text{total}} = \frac{h}{h} \gamma_{\text{total}} = \gamma_{\text{total}},$$

which makes sense as now both the healing interface and the bulk strain equally.

It is easier to measure G_{eff} in simulations: we therefore fill in Equation (4.2.9) in Equation (2.2.1) to find

$$\sigma = G_{\text{int}} \gamma_{\text{int}} = \frac{hG_{\text{bulk}}}{(h - 4\delta) + 4\delta \frac{G_{\text{bulk}}}{G_{\text{int}}}} \gamma_{\text{total}},$$

so that the effective equilibrium shear modulus of the entire unit cell is

$$G_{\text{eff}} = \frac{\sigma}{\gamma_{\text{total}}} = \frac{hG_{\text{bulk}}}{h - 4\delta + 4\delta \frac{G_{\text{bulk}}}{G_{\text{int}}}}. \quad (4.2.10)$$

Here too we perform some sanity checks. When no healing has occurred, and $G_{\text{int}} = 0$, we see that

$$\lim_{G_{\text{int}} \rightarrow 0} G_{\text{eff}} = \lim_{G_{\text{int}} \rightarrow 0} \frac{hG_{\text{bulk}}}{h - 4\delta + 4\delta \frac{G_{\text{bulk}}}{G_{\text{int}}}} = 0,$$

so that the bulk pieces can freely flow across each other. Conversely, when the material is fully healed, and $G_{\text{int}} = G_{\text{bulk}}$, we see that

$$\lim_{G_{\text{int}} \rightarrow G_{\text{bulk}}} G_{\text{eff}} = \lim_{G_{\text{int}} \rightarrow G_{\text{bulk}}} \frac{hG_{\text{bulk}}}{h - 4\delta + 4\delta \frac{G_{\text{bulk}}}{G_{\text{int}}}} = \frac{hG_{\text{bulk}}}{h} = G_{\text{bulk}},$$

as expected. It now seems reasonable to use

$$G_{\text{eff}} \geq \alpha G_{\text{bulk}}, \quad (4.2.11)$$

as a mechanical self-healing criterion, where $\alpha \in (0, 1)$: we can then say that the macroscopic material has recovered a fraction of α of its original stiffness. Hence, we can keep track of G_{eff} as a function of the amount of time the system has been allowed to self-healing to find the mechanical healing time.

One advantage of Equation (4.2.11) is that it is comparatively straightforward to verify whether the threshold has been reached: we can measure G_{eff} after some fixed healing time by shearing the samples in directions parallel to the healing interfaces, while we can find G_{bulk} by shearing in the perpendicular direction. It is additionally very much in line with the approach used by Ciarella and Ellenbroek, which uses compares G_{xz} and G_{yz} to G_{xy} after various healing times [1, Fig. 4.3].

This definition of mechanical recovery does have some downsides, however. First, the influence of G_{int} on G_{eff} depends on the size of the unit cell: if we move the healing interfaces further apart, G_{eff} will become closer to G_{bulk} . Hence, we have to keep the system size constant to be able to compare different networks fairly. This issue could be circumvented by slightly adapting the mechanical self-healing criterion to

$$G_{\text{int}} \geq \alpha G_{\text{bulk}}.$$

However, we cannot measure G_{int} directly in simulations, and would instead have to infer it from G_{eff} using

$$G_{\text{int}} = \frac{4\delta G_{\text{bulk}}}{hG_{\text{bulk}} - (h - 4\delta)G_{\text{eff}}},$$

which is rather sensitive to δ . Conversely, Equation (4.2.11) does not rely on our simple model to the same degree. Another potential shortcoming of Equation (4.2.11) is that it classifies samples as being healed in a manner that is relative to the initial mechanical properties; in practical applications it perhaps makes more sense to use an absolute threshold, to penalise configurations with poor mechanical properties, such as viscoelastic liquids, which have $G_{\text{bulk}} = 0$ by definition. Notably, such an absolute threshold would necessarily depend on the application, and could be checked in exactly the same way as our relative threshold.

While the procedure to determine the mechanical healing time described above is straightforward, it is rather computationally expensive. To track the mechanical recovery over time we must:

1. heal the sample;
2. take snapshots during the healing;
3. shear the snapshots and study their stress response.

However, during the self-healing process we can already keep observe the evolution of the number of interfacial bonds, which should give a good idea of the mechanical properties of the sample: recall that in the affine network model the equilibrium shear modulus is proportional to the number density of chains connecting nodes in the network, see Equation (2.2.4). Hence, we expect that $G_{\text{int}} \propto N_{\text{int}}$. Since $G_{\text{int}} \rightarrow G_{\text{bulk}}$ for long healing times, and $N_{\text{int}}(t) \rightarrow N_{\text{int}}(\infty)$, we can therefore conclude that

$$G_{\text{int}}(t) = \frac{N_{\text{int}}(t)}{N_{\text{int}}(\infty)} G_{\text{bulk}}, \quad (4.2.12)$$

so that

$$G_{\text{eff}}(t) = \frac{hG_{\text{bulk}}}{h - 4\delta + 4\delta \frac{G_{\text{bulk}}}{N_{\text{int}}(\infty)}} = \frac{hN_{\text{int}}(t)G_{\text{bulk}}}{(h - 4\delta)N_{\text{int}}(t) + 4\delta N_{\text{int}}(\infty)}. \quad (4.2.13)$$

Consequently

$$G_{\text{eff}}(t) > \alpha G_{\text{bulk}} \iff \frac{hN_{\text{int}}(t)}{(h - 4\delta)N_{\text{int}}(t) + 4\delta N_{\text{int}}(\infty)} > \alpha. \quad (4.2.14)$$

Hence, we are able to skip the second and – more significantly – third step when tracking the mechanical recovery. This makes analysing the self-healing much cheaper, so that we may experiment with a greater variety of vitrimer network building blocks with the same computational budget.

4.2.3 Simulations

We will now discuss the MD simulations that we have executed and analysed to test our healing model. The scripts we used for this may again be found on [Hoomd Polymer Tools](#):

1. The self-healing simulations were performed using `generating_data/heal_sample.py`;
2. The evolution of the number of interfacial bonds was tracked using `processing_data/bonds_between_samples.py`;
3. The equilibrium shear moduli were found by shearing the partially healed samples in `processing_data/measure_stress_relaxation.py`, and analysing the resulting trajectories in `analysing_data/compute_shear_moduli.py`;
4. The shear profiles were finally computed using `analysing_data/shear_throughout_sample.py`.

We will first turn our attention to how the number of bonds across the healing interface changes over time; Figure 4.8 a) shows this evolution eleven distinct vitrimer networks. Each curve is accompanied by a filled region corresponding to approximate 95 % confidence intervals determined using 45 independent samples.¹² At first sight, the evolution appears to follow some exponential decay, as we anticipated in Subsection 4.2.1. It is quite evident from Figure 4.8 a) that $t_{\text{heal}} = 300$ [time] is not enough time for the samples to fully heal; we therefore cannot simply read off $N_{\text{int}}(\infty)$ from the curves. Our model,

¹²We assume the noise is roughly normally distributed; then the standard deviation of the mean may be estimated with the standard error. The shaded regions have a width of two standard errors on each side of their corresponding curve.

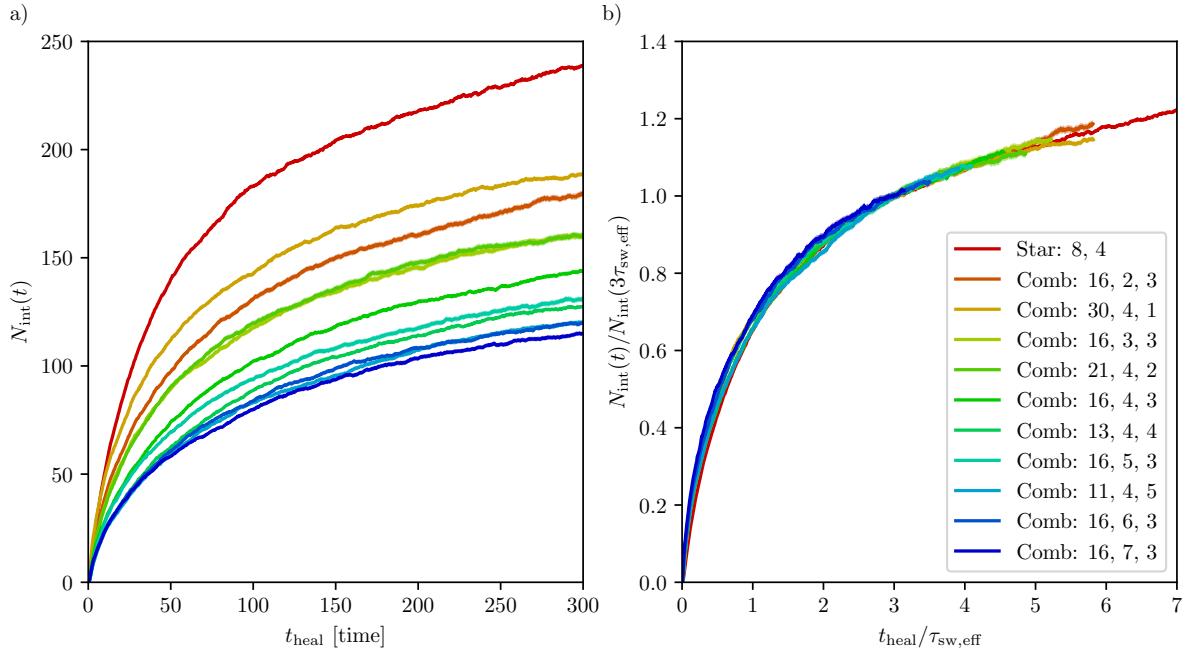


Figure 4.8: Evolution of the number of interfacial bonds over time for vitrimer networks made of various polymer building blocks. In b), the x-axis has been normalised for each sample with the effective swap times $\tau_{\text{sw,eff}}$ measured in Subsection 4.1.3, while the y-axis has been normalised by the number of interfacial bonds at fixed time $3\tau_{\text{sw,eff}}$.

described in Equation (4.2.3), predicts that the form of this evolution does not depend on the shape of the building blocks of the network, since

$$\frac{N_{\text{int}}(t)}{N_0} = \frac{p_{\text{int}}}{p_{\text{bulk}} + p_{\text{int}}} \left(1 - \exp \left(-(p_{\text{bulk}} + p_{\text{int}}) \frac{t}{\tau_{\text{sw,eff}}} \right) \right),$$

in which only the number of sufficiently close vitrimer bonds N_0 and the effective swap time $\tau_{\text{sw,eff}}$ depend on the polymer shapes. In Figure 4.8 b) we therefore normalise the time axis by the $\tau_{\text{sw,eff}}$ – determined in Subsection 4.1.3 – for each sample. Since we do not have values for $N_{\text{int}}(\infty)$ or N_0 , we normalise the y-axis by the number of interfacial bonds at $t_{\text{heal}} = 3\tau_{\text{sw,eff}}$; if the curves are truly similar all these normalisations are equivalent up to scalar multiplication. Figure 4.8 b) indeed shows that the form of the evolution does not depend on the shape of polymer building blocks.

It would be interesting to know whether our expression for N_0 , Equation (4.2.7), is reasonable. As we alluded to previously, in theory $N_{\text{int}}(3\tau_{\text{sw,eff}})$ gives us N_0 up to a scalar multiple. Hence, we can fit our expression on $N_{\text{int}}(3\tau_{\text{sw,eff}})$. In our simulations, n_{arm} is the only relevant parameter that is varied; Figure 4.9 shows how $N_{\text{int}}(3\tau_{\text{sw,eff}})$ depends on n_{arm} . We can clearly see that our model is too simple, even for configurations with the same

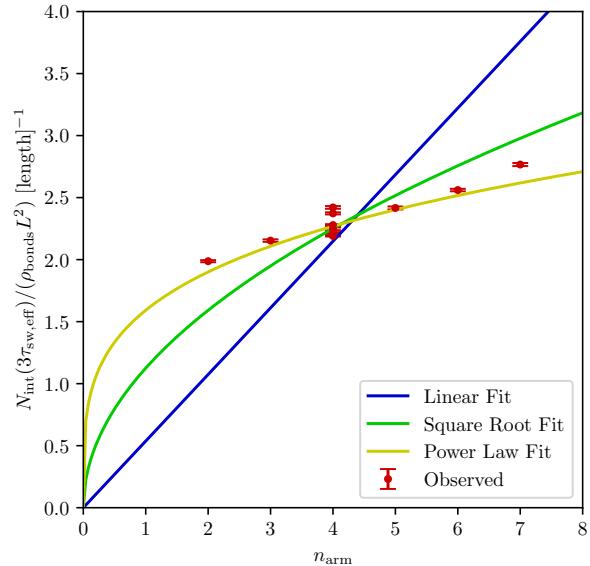


Figure 4.9: Normalisation $N_{\text{int}}(3\tau_{\text{sw,eff}})/(\rho_{\text{bonds}} L^2)$ as a function of the number of particles in the arms n_{arm} of the building blocks of the vitrimer network. Data fitted with $a n_{\text{arm}}$ (Linear), $a \sqrt{n_{\text{arm}}}$ (Square Root), and $a n_{\text{arm}}^\beta$ (Power Law).

number of particles in each arm the values do not coincide. We moreover see that the linear fit predicted by Equation (4.2.7) is terrible. We have therefore also fitted the curve $a\sqrt{n_{\text{arm}}}$, which we would expect in long chains ($n_{\text{arm}} \gg 1$); this fit performs only slightly better than the linear fit, and moreover does not make physical sense. Finally, we fitted the power law an_{arm}^β . Clearly, our data cannot be described by such a power law either.

We can hence conclude that some of our modelling assumptions were too naive. To derive Equation (4.2.7), we assumed that the attachment points of the arms are stationary. Indeed, if the backbones of the comb polymers are entangled we expect them to remain roughly fixed in space. However, a segment between two points of entanglement will be able to move around.

As a consequence, the arms will be able to reach farther than we predicted. Since of the lengths of these segments will depend i.a. on the number of particles separating two arms n_{sep} , it is not so strange that configurations with the same value for n_{arm} can have ‘bulks’ of different sizes. We moreover assumed that p_{bulk} and p_{int} do not depend on the shape parameters. This is clearly not true however: we already noted in Subsection 4.2.1 that in essence p_{bulk} and p_{int} are averaged over the ‘bulk’. It might be possible to improve the model by working with the chain length distribution of the arms instead a single point estimate. In this way, minority moiety which are further away will give a smaller contribution.

We next fit our self-healing model described Equation (4.2.3) by in Figure 4.10. It turns out that the evolution does not follow some simple exponential decay: for both the comb vitrimer network a) and the star vitrimer network the fit of the healing model, represented as a dashed blue curve, underestimates the observed number of interfacial bonds, shown as the solid red curve, at short and long timescales, while overestimating it at intermediate timescales. The model additionally underestimates the number of bonds across the healing interface in equilibrium: by fitting we find that $N_{\text{int}}(\infty) = 137$ for the comb network and $N_{\text{int}}(\infty) = 225$, while at $t = 300$ [time] we on average count 145 and 240 bonds across the healing interface, respectively.

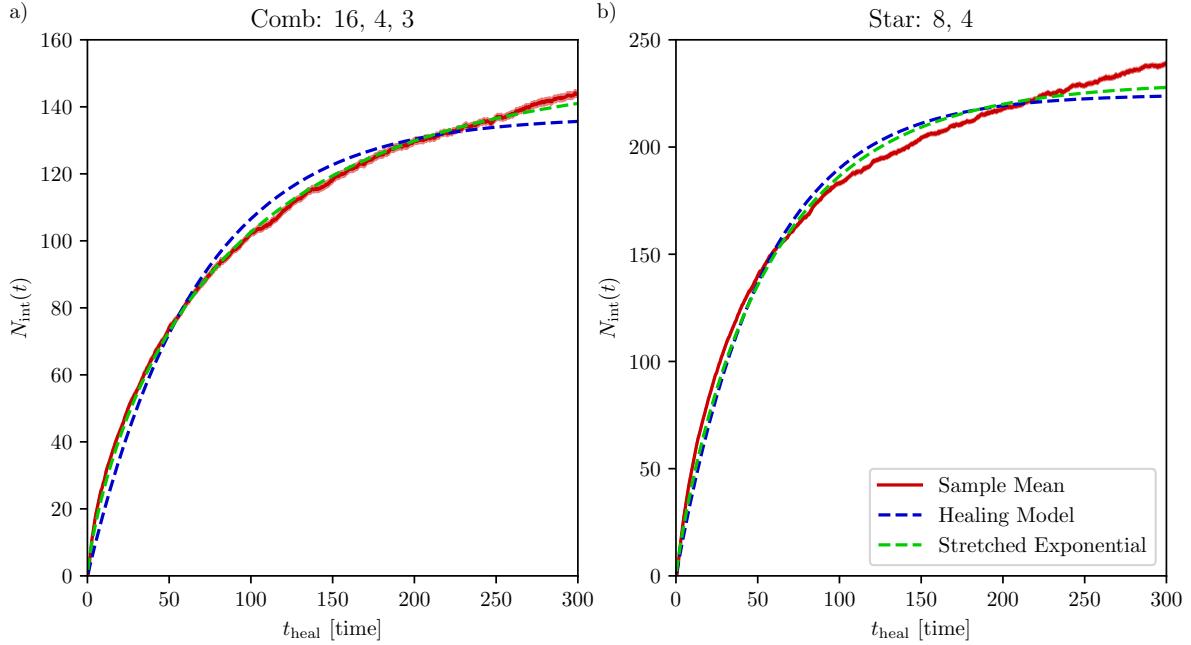


Figure 4.10: Evolution of the number of interfacial bonds over time for vitrimer networks, fitted with our self-healing model – Equation (4.2.3) – and a stretched exponential generalisation thereof, given by Equation (4.2.15).

To mitigate these issues, we consider the most straightforward generalisation of our healing model: we replace the exponential in Equation (4.2.3) with a stretched exponential:

$$N_{\text{int}}(t) = \frac{p_{\text{int}}}{p_{\text{bulk}} + p_{\text{int}}} N_0 \left(1 - \exp \left(- \left((p_{\text{bulk}} + p_{\text{int}}) \frac{t}{\tau_{\text{sw,eff}}} \right)^\beta \right) \right). \quad (4.2.15)$$

It is important to note that stretched exponentials are usually not as easy to justify theoretically as an exponential fit, although they are frequently used when fitting polymer relaxation processes [48,

Eq. (6.97)]. Unfortunately, the stretched exponential fit suffers from the same issues as the exponential fit, underestimating the number of bonds across the healing interface on short and long timescales but overestimating them on intermediate timescales.

We can get a better understanding of why the fitting fails by investigating the healing speed, i.e. the derivative of the number of interfacial bonds. On average, we expect $N_{\text{int}}(t)$ to be a strictly increasing function of t . Since we moreover know that

$$N_{\text{int}}(0)/N_{\text{int}}(\infty) = 0 \text{ and } \lim_{t \rightarrow \infty} N_{\text{int}}(t)/N_{\text{int}}(\infty) = 1,$$

we can interpret $S(t) := 1 - N_{\text{int}}(t)/N_{\text{int}}(\infty)$ as a survival function. As a consequence, we can apply the lifetime distribution machinery introduced in Subsection 2.1.1 to this problem, too. In particular, we can compute the corresponding PDF f up to a multiplicative constant of $N_{\text{int}}(\infty)$, since we know that

$$f(t) = -\frac{d}{dt}S(t) = -\frac{d}{dt}\left(1 - \frac{N_{\text{int}}(t)}{N_{\text{int}}(\infty)}\right) \implies N_{\text{int}}(\infty)f(t) = -\frac{d}{dt}N_{\text{int}}(t).$$

We can estimate the last term using a KDE as discussed in Definition 2.1.4: Figure 4.11 shows the estimates of $N_{\text{int}}(\infty)f$ for all of our vitrimer networks. We have less data now than when we computed the swap time KDEs in Subsection 4.1.3, and so we convolve with Gaussians kernels with $\sigma = 15$ [time].¹³ As a consequence, our estimates will be more biased now, especially around $t_{\text{heal}} = 0$ [time] and $t_{\text{heal}} = 300$ [time], since for the computation of the convolution the data is extended outside of its support with 0. Moreover, the densities are not normalised, so we cannot directly compare the densities for different network configurations. Consequently, we have to be careful drawing any conclusions from Figure 4.11. The densities appear to contain a kink around $t_{\text{heal}} = 100$ [time]: before and after this kink the densities seem to decay exponentially, but the decay rate is high before the kink. This could be explained the healing being mediated by two distinct – potentially exponentially decaying – processes, where one has a typical timescale smaller than 100 [time], and the other has a typical timescale much larger than 100 [time].

It would be much easier to read off the presence or absence of such processes from plots of the hazard function. Unfortunately, since we do not know $N_{\text{int}}(\infty)$, we are unable to determine the hazard function, as

$$\lambda(t) = \frac{f(t)}{S(t)} = \frac{\frac{d}{dt}\frac{N_{\text{int}}(t)}{N_{\text{int}}(\infty)}}{1 - \frac{N_{\text{int}}(t)}{N_{\text{int}}(\infty)}} = \frac{\frac{d}{dt}N_{\text{int}}(t)}{N_{\text{int}}(\infty) - N_{\text{int}}(t)}.$$

However, from the fact that Equation (4.2.3), which we can identify with the exponential distribution, deviates significantly from the data, we may conclude that the true hazard function is not a constant. Similarly, since Equation (4.2.15) does not fit our observations, we know the hazard function does not follow a power law, as the stretched exponential corresponds to a Weibull distribution.

Finally, we will attempt to determine the typical mechanical healing times by looking at the equilibrium shear moduli, as described in Subsection 4.2.2. We took numerous snapshots during the healing process. Each of these snapshots was subjected to shears $\gamma = 0.25$ in the xy -, xz -, and yz -directions, and allowed to relax to the elastic plateau,¹⁴ after which we determined the average value of the corresponding component of the stress tensor.

¹³We again chose the kernel width by eye.

¹⁴Different samples relaxed at different speeds: the most healed samples relaxed after roughly 10^3 [time], while the least healed samples took 9×10^3 [time].

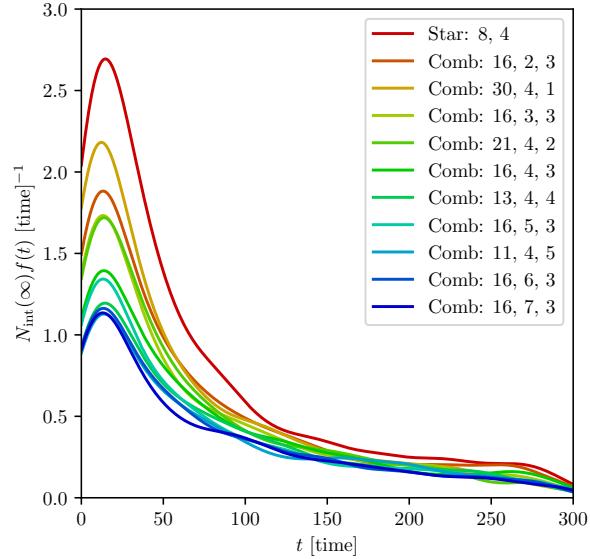


Figure 4.11: Estimates of $N_{\text{int}}(\infty)f$ as a function of healing time computed using KDEs with Gaussian kernels with $\sigma = 15$ [time] for vitrimer networks made of various polymer building blocks.

We start off by checking whether our model of the unit cell, depicted schematically in Figure 4.7 is reasonable. To do so, we measure how particles throughout the sample have moved compared to their initial position once the sample has reached mechanical equilibrium. In this way, we can get an idea of the ‘shear profile’, as seen in Figure 4.7 c). If our model is realistic, we expect that after short healing times the sample will shear more extremely at the healing interfaces than in the bulk. Moreover, we expect the shear within the bulk and within the healing interfaces to be uniform. Around a single healing interface centred at $z = 0$ [length] we can state this mathematically using the following fit:

$$\Delta x = \begin{cases} -\gamma_{\text{int}}\delta + \gamma_{\text{bulk}}(z + \delta), & z < -\delta, \\ \gamma_{\text{int}}z, & -\delta \leq z \leq \delta, \\ \gamma_{\text{int}}\delta + \gamma_{\text{bulk}}(z - \delta), & z > \delta. \end{cases} \quad (4.2.16)$$

For a single sample, we have plotted the shear profiles after various amounts of healing time – fitted with Equation (4.2.16) – in Figure 4.12. To smooth the profiles, we organised the 9.9×10^4 particles by their z -position, and applied a Gaussian filter with $\sigma = 5 \times 10^2$ particles.¹⁵

Since we do not expect the width of the healing interface δ to depend on the amount of time that the sample has healed for, we fitted all profiles simultaneously for δ ; in this way we found $\delta = 0.53$ [length], which seems sensible since it is on the order of a single particle diameter. The observed profiles are well-described by Equation (4.2.16); we therefore believe our model is appropriate. Next, we will see whether the model we derived from this model, can accurately describe the dependence of the equilibrium shear modulus on the number of interfacial bonds.

In Figure 4.13 we have plotted the equilibrium shear modulus averaged over 45 samples after various amounts of healing time; the error bars correspond to two standard errors of the mean. For the vitrimer network made of combs, we can see in Figure 4.13 a) that for $t_{\text{heal}} \geq 200$ [time] the equilibrium shear modulus in the directions parallel to the healing interfaces significantly exceeds the modulus in the perpendicular direction. This may be an artefact caused by the samples having insufficient time to relax to the elastic plateau. Notably, we do not see this in the star polymer network shown in Figure 4.13 b).

We have fitted Equation (4.2.13) for δ , $N_{\text{int}}(\infty)$, and G_{bulk} ; the fitted curves were subsequently computed by filling in the mean number of interfacial bonds shown in Figure 4.10. This fitting problem is somewhat ill-posed, in the sense that many wildly different combinations of parameters will lead to roughly the same fit: there therefore is no unique solution. By determining some of these parameters independently and filling their values in, we could remove this degeneracy. For instance, $N_{\text{int}}(\infty)$ can be found independently by healing the samples sufficiently long that the number of interfacial bonds plateaus, and G_{bulk} can be estimated from the equilibrium shear moduli in the direction perpendicular to the healing interface. Additionally, Equation (4.2.13) does not seem to be able to properly fit the shear moduli observed after long healing times. This may in part be due to the affine network assumption, which neglects entanglements, as noted in Subsection 2.2.1: we expect that there has not been enough time for entanglements to be formed across the healing interface.

We have computed the equilibrium shear modulus of the bulk by averaging the shear moduli in the direction perpendicular to the healing interface, and (arbitrarily) chosen as mechanical healing threshold $0.9G_{\text{bulk}}$. We find that the comb vitrimer network passes this threshold at $\tau_h = 22.8$ [time], while the

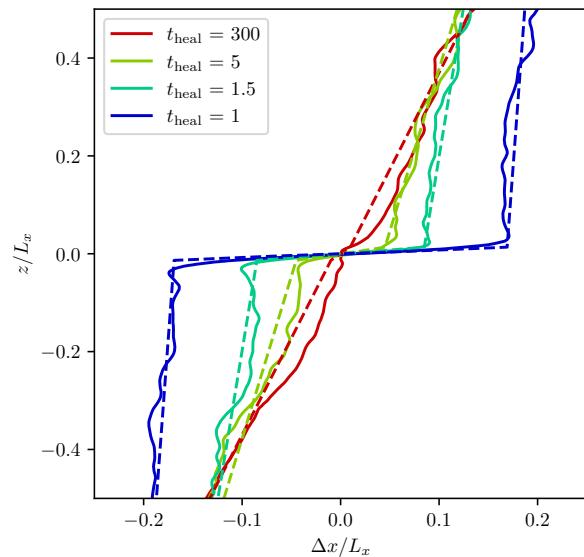


Figure 4.12: Smoothed shear profiles fitted with Equation (4.2.16) for samples that have healed for varying amounts of time. Δx is the displacement in the x -direction in the elastic plateau compared to the initial position and $L_x \approx 45$ [length] is the width of the unit cell in the x -direction.

¹⁵The fit was performed on the raw, unsmoothed data.

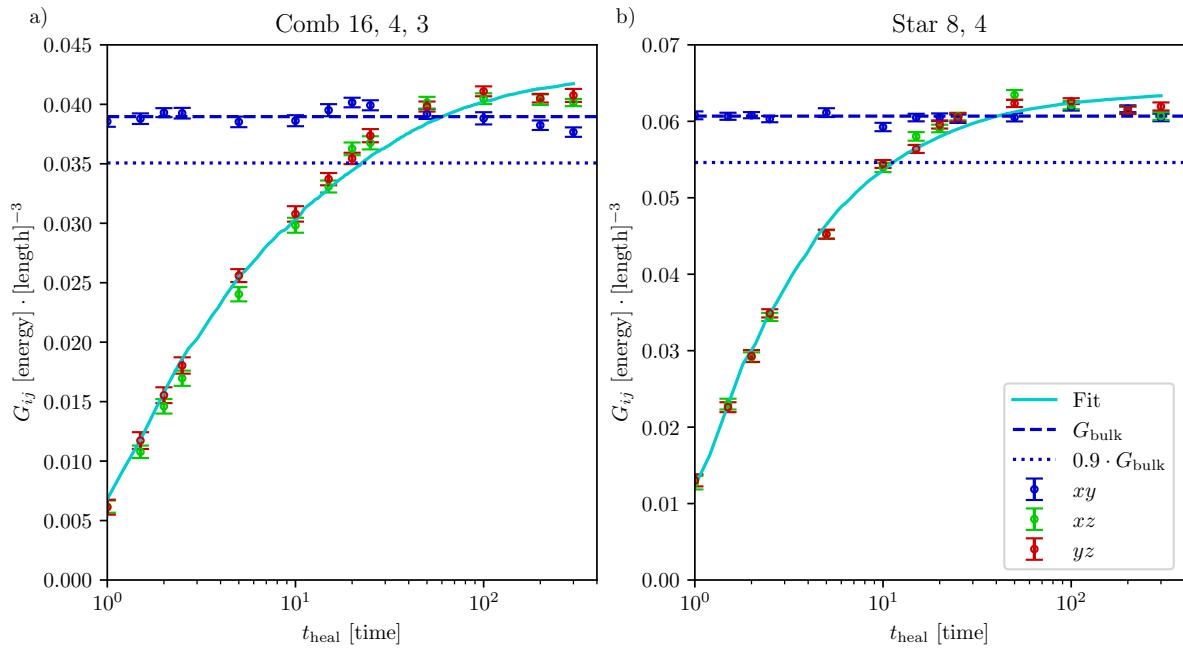


Figure 4.13: Equilibrium shear moduli in the xy -, xz -, and yz -directions as a function of healing time. Fits were computed using Equation (4.2.13). The blue dashed line represents the shear modulus of the bulk computed by averaging the shear moduli in the xy -direction, while the dotted line is a mechanical healing threshold.

star network vitrimer does so at $\tau_h = 11.4$ [time]. These healing times are much shorter than the effective swap times of 60.7 [time] and 40.8 [time], respectively.

Despite the limitations of our model, we see that there is decent agreement between the fits and the observed shear moduli, which suggests that we can deduce mechanical healing directly from the number of interfacial bonds.

4.2.4 Conclusions

In conclusion, we have seen that the evolution of the number of interfacial bonds is not quite exponential, in contradiction with our model (Equation (4.2.3)). However, we do see that the curves collapse when we normalise the number of bonds by some factor that should proportional to the number of bonds in equilibrium, and normalise the times by the mean effective swap times for each network. Consequently, we can still use the effective swap time as a typical timescale for the recovery of the number of interfacial bonds.

We have also predicted the evolution of the equilibrium shear modulus as a function of the evolution of the number of bonds across the healing interface using the affine network model. On short times scales, our prediction, Equation (4.2.13), performs well. Conversely, on long timescales, Equation (4.2.13) does not fit the data. Fortunately, most of the mechanical healing occurs on short timescales. We can therefore use our model to find the typical healing time. For the networks we have investigated, the typical mechanical healing time is less than the effective swap time, which is incredibly fast.

4.3 Vitrimer Relaxation

We have now seen how comb polymer vitrimer networks heal. In particular, we have seen that the self-healing happens on timescales on the order of the swap time or smaller. Hence, comb vitrimer networks indeed seem suitable candidates for autonomously healing materials. However, a practical self-healing material must have additional mechanical properties. For instance, water clearly would not be an appropriate self-healing material, even though it will almost instantly autonomously recover if you ‘damage’ it. It therefore seems sensible to require that the material behaves like a solid on

relevant timescales. Fortunately, on sufficiently short timescales vitrimers behave like viscoelastic solids, as evidenced by their creep resistance [10], only relaxing to a liquid when each minority moiety has undergone many swaps (see Subsection 2.2.2).

In 2019, Ciarella and Ellenbroek observed that the self-healing in a vitrimer network made of star polymers happens about three orders of magnitude faster than the relaxation to liquid [1]; we posited that this can be improved by using a vitrimer made of comb polymers, since the entanglement of the backbones will frustrate the stress relaxation. Indeed, in a melt of linear chains the relaxation will certainly take longer than the reptation time, which – at fixed entanglement length – scales with the cube of the chain length [25, Eq. (9.8)]. In a comb polymer melt, the relaxation will take even longer if the arms are sufficiently long [81]. It stands to reason that a vitrimer network made of comb polymers will relax even more slowly, since the existence of crosslinks between arms will inhibit the reptation of the backbones.

In this section, we will discuss the experiments we have performed to test this hypothesis. Since we need to measure the evolution of the stress tensors on both short and long time scales, which is computationally expensive, we only considered the networks made of comb polymers with $N_{\text{arms}} = 16$, $n_{\text{arm}} = 4$, and $n_{\text{sep}} = 3$.

First, we look at the stress relaxation when we fix the network topology by replacing the exchangeable bonds with simple harmonic bonds; on short timescales we should observe the same relaxation for networks with frozen and dynamic topologies. For this, we used the stress tensors we found in Subsection 4.2.3 for the xy shears, since we are now interested in the properties of the undamaged material. In total, we simulated 495 of such shears. The evolution of the shear modulus is shown in Figure 4.14; the shaded region corresponds again to two standard errors in the mean. As discussed in Subsection 2.2.1, we expect an exponential decay towards a plateau, which could be described as

$$G(t) = G_\infty + a \cdot \exp\left(-\frac{t - \tau_{\text{shift}}}{\tau_{\text{scale}}}\right), \quad (4.3.1)$$

for some $G_\infty, a, \tau_{\text{shift}}, \tau_{\text{scale}} > 0$. In fact, we clearly do not observe exponential decay; this discrepancy may be caused by the fact that shear is only approximately a step, occurring over 1 [time]: some of the stress can already be relaxed before the entire shear has been applied.

We subsequently looked at the stress relaxation of the dynamic vitrimer network. Ciarella, Sciortino and Ellenbroek found that it takes about 10^6 [time] for a vitrimer network made of stars to become liquid [24]; since we expect our network based on combs to relax more slowly, we should consequently certainly simulate until 10^6 [time], which corresponds to 10^9 timesteps. To make such simulations feasible, we had to reduce the size of the unit cell from 9.9×10^4 to just 2×10^4 particles; even with this reduction, a single simulation takes roughly two days to complete on a GPU.¹⁶ As a consequence, the noise on the stress tensor will be larger: we observed standard deviations across independent samples around 5×10^{-1} [energy] · [length]⁻³. We performed 105 independent stress relaxation simulations on such systems. We therefore expect a standard error in the mean of 5×10^{-2} [energy] · [length]⁻³, which is on the order of the elastic plateau shear modulus we measured before: this does not bode well for our ability extract a signal from the noise. By contrast, in Figure 4.14 the standard error is typically

¹⁶We estimate that it would take about a year on a single CPU core.

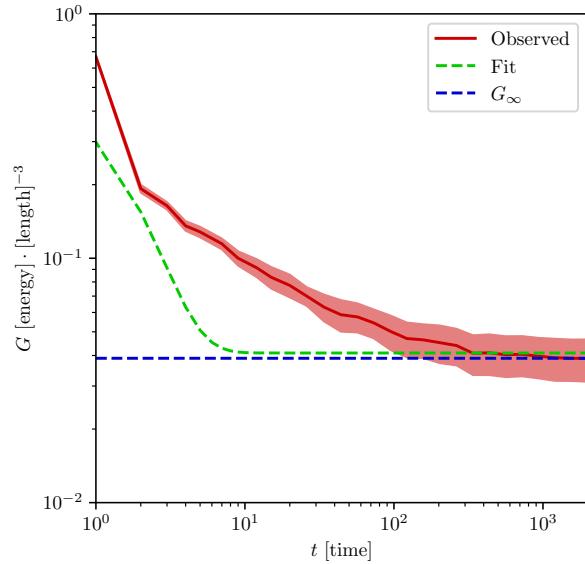


Figure 4.14: Evolution of the shear modulus in a comb polymer network, with fixed topology, after the application of a $\gamma = 0.25$ step shear. The dashed green curve has been fitted using Equation (4.3.1), which describes the expected decay to the elastic plateau; the dashed blue line is the equilibrium shear modulus.

10^{-2} [energy] · [length] $^{-3}$, a factor 5 lower, which makes sense because there we had roughly 25 times as much data.¹⁷.

The blue curve Figure 4.15 shows the evolution of the shear modulus; the noise is quite extreme. There are numerous approaches that can be taken to improve the chances of extracting the signal from the noise. The most obvious approach is to perform more simulations; to reduce the noise to the same level as we saw in the stress relaxation of the frozen vitrimer networks we would need about 400 more simulations.¹⁸ It would perhaps be more sensible to instead try to filter out the noise. The noise appears to have a very high frequency compared to the timescales that are relevant for stress relaxation; hence a low pass filter such as a Gaussian filter might help. However, if we use a Gaussian kernel, we will not be able to resolve the signal on short timescales. Hence, it might be interesting to use a kernel with a time dependent width.

Perhaps a more significant issue with the observed stress relaxation is that it decays to 0 in about the same amount of time that it takes for the networks with fixed topology to relax

to the elastic plateau, whereas it should take much longer. The fact that we see only a single relaxation process implies that the building blocks of the network are not actually crosslinked. The only explanation we could find for this is that there is an issue with RevCross on the GPU. Indeed, we already noted in Section 3.3 that different swap times are observed on the GPU than on the CPU; the swap times found using the GPU moreover did not appear to depend on e.g. the concentration of reactive particles, which does not make physical sense.

In conclusion, with the currently gathered simulation data we are unable to determine how long it takes for a comb vitrimer network to relax into liquid.

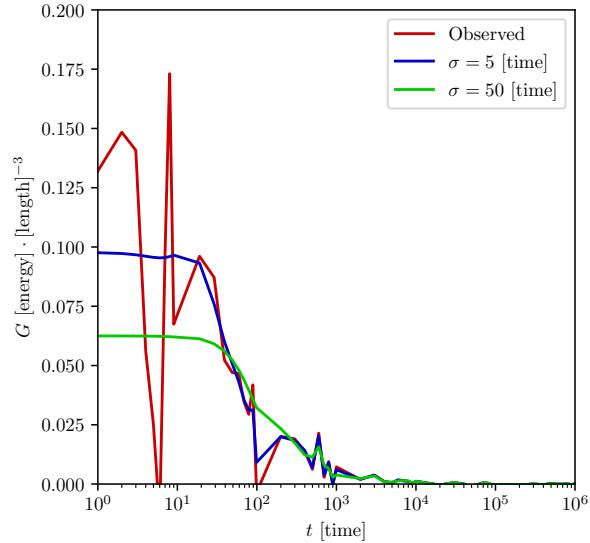


Figure 4.15: Evolution of the shear modulus in a dynamic comb vitrimer network after the application of a $\gamma = 0.25$ step shear. The blue and green curves were computed by convolving with a Gaussian kernel with the corresponding standard deviation.

¹⁷Namely about 5 times as many samples which were each about 5 times as large

¹⁸Using all GPUs in the hpc cluster of the TU/e Physics department continuously this would take about half a year.

Chapter 5

Discussion

In this work, we have strived to find materials with good mechanical properties that can self-heal. Previous research indicates that CANs are suitable candidates: the covalent nature of their bonds makes them rigid, while their ability to reconfigure their topology allows them to heal damage [22]. In particular, in 2019, Ciarella and Ellenbroek found that vitrimer networks made of star polymers can mechanically heal roughly three orders of magnitude sooner than they start to flow like a liquid [1]. We have therefore investigated how the self-healing properties of vitrimer networks made of comb polymers compare with those of networks consisting of star polymers, since we hypothesised that the possibility for entanglement of the backbones of the comb polymers would increase the timescales on which the material remains rigid. We have also looked into the influence of i.a. the length of the arms of the comb polymers on the self-healing.

For our investigation, we have first developed and tested a simple model to describe the dependence of typical swap times on the shape of the building blocks of the network. We have subsequently studied the evolution over time of the number of vitrimer bonds across some damage, making use of the previously found swap times. This is interesting since we can predict the mechanical rigidity of the network using the number of such bonds; we may therefore map the growth of this microscopic quantity onto the recovery of mechanical properties on the macroscopic scale. Finally, we have attempted to measure how the stress in the network is relaxed after the application of a step shear, in order to identify on what timescale the comb polymer vitrimer becomes liquid.

In this chapter, we will first summarise the main conclusions we can draw from our mathematical models and simulations, as well as the limitations of our work, in Section 5.1. We will thereafter conclude this thesis by going over some of the most promising avenues for future research into self-healing vitrimers materials in Section 5.2.

5.1 Conclusions and Limitations

Our first result is a simple polymer model that describes how the typical swap times in a comb or star vitrimer network depend on the shape of the combs or stars. Central to deriving this model is the classification of swaps into two categories: ‘undoing’ if they undo the previous swap and ‘doing’ otherwise. The model predicts that the typical swap time depends on the shape only via the concentration of reactive particles, see Equation (4.1.15). We have verified this model using simulations. From these simulations, we have additionally computed local estimates of the entire swap time distributions, namely Kernel Density Estimators and Kernel Hazard Estimators; we have been unable to find similar results in the literature. We should note that the model has been derived assuming the system is in equilibrium, and the simulations have been performed in equilibrium. However, we believe that it is possible to identify the key mechanisms that determine the swap times by investigating e.g. the corresponding PDF or hazard function. We believe that understanding these mechanisms will allow us to make predictions about typical swap times even in out-of-equilibrium systems by then considering how these mechanisms change when the system is not in equilibrium. Say we want to estimate the typical swap time in a network that is being stretched, for example. Then, we can imagine that the probability that a swap is

undoing is lower than in an equilibrium system, since the neighbourhoods of minority moieties changes over time. By adjusting the corresponding parameter in the model, we may be able to apply it even in this out-of-equilibrium situation.

We have also derived a model based on rate equations to describe how the number of vitrimer bonds across damage evolves over time: we predicted an exponential decay towards a plateau, see Equation (4.2.4). In simulations, however, we have observed a decay that cannot be described by either a simple or stretched exponential. Careful consideration of the derivative of the evolution suggests that there are two independent processes with different timescales going on. Although the exponential nature of the model is not reflected in the data, we do see that the curves describing the evolution for vitrimer networks made with various types of comb polymers collapse when we normalise the number of bonds and divide the times by the mean effective swap times for each network. Consequently, the effective swap time is a good proxy for the time it takes for a certain fraction of the equilibrium number of bonds to be reached.

We have furthermore predicted the evolution of the equilibrium shear modulus of such vitrimer networks from the evolution of the number of bonds across the damage using the affine network model. For this prediction, we assumed that we can decompose the shear of the entire network into a simple shear of the healing interfaces and one of the bulk, so that the strain in the healing interfaces and bulk is constant but different from each other; by looking at the equilibrium shear profiles in simulations we have confirmed that this is indeed the case. Our prediction for the equilibrium shear modulus – described by Equation (4.2.13) – performs well on short timescales. Conversely, at long timescales, Equation (4.2.13) does not fit the data. Fortunately, most of the mechanical healing occurs on short timescales. Hence, we can use our model to find the typical healing time. For the networks we have investigated, the typical mechanical healing time is less than the effective swap time, which is incredibly fast. These results imply that we do not need to perform expensive stress relaxation experiments to determine the self-healing times, which means we can explore more network topologies with the same computational budget.

Finally, we have tried to look how long it takes for a comb vitrimer network to become liquid. We have done this by applying step shears and measuring the evolution of the stress tensor. Since this process is computationally expensive, our data contains a lot of noise. More importantly, we have seen in the simulations that the network liquefies in about the same amount of time that a corresponding comb polymer network with fixed crosslinks takes to reach the elastic plateau: the absence of two relaxation processes, which are well-established in vitrimers [24], suggests that there are issues in the software implementation of the reversible crosslinks for simulations run on GPUs. As a consequence, we have not been able to determine on what timescales the network becomes liquid, and so we are unable to fully compare the performance of vitrimer networks made of combs to those made of stars at the moment.

5.2 Outlook

This work has raised numerous questions that are, in our opinion, worth investigating further. For instance, we believe that it should be possible to improve the swap time model, since the hazard functions indicate that we can divide swaps into more fine categories than just ‘doing’ and ‘undoing’. Such an improvement would be valuable, since our results indicate that the effective swap time is closely related to the mechanical healing time. By using a non-Gaussian kernel, e.g. one with positive support [37], with a width chosen in a data-driven way, e.g. using LOOCV, one could reduce the bias and variance of the hazard estimators, making it easier to identify the timescales of mechanisms that determine the bond exchanges.

The self-diffusion time of a single particle is a fit parameter in our swap time model. It should be possible to measure self-diffusion time independently in simulations; we unfortunately did not get around to doing this in our project. We believe such an independent measurement would be valuable, since filling in the value of the self-diffusion time would alleviate the degeneracy and the identifiability issues of our model.

A particularly straightforward follow-up research would involve healing the samples for more time, so that we can measure the number of bonds across the healing interface in equilibrium. This would reduce the degeneracy in Equation (4.2.13), so that we can better predict the equilibrium shear modulus from

the number of interfacial bonds. Additionally, it would make it possible to estimate the hazard function of the evolution, which might make it easier to find why our model for the number of interfacial bonds does not fit the data.

In this study, we considered networks where the fraction of vitrimer moieties of type-A₂ was fixed at $f_{A_2} = 0.4$. We think it would be interesting to investigate the influence of f_{A_2} on the self-healing properties. We expect f_{A_2} to affect both the healing rate and the mechanical properties of the vitrimer network. For instance, we derived that the swap time, and so also the healing time, depends on f_{A_2} via concentration of vitrimer bonds and the concentration of free vitrimer particles, see Equation (4.1.13). Additionally, since f_{A_2} is proportional to the number of crosslinks in the network, we expect that the rubber plateau to rise and that the network will take longer to relax to liquid with increasing f_{A_2} .

Finally, to check our hypothesis it will be necessary to measure the stress relaxation of a comb vitrimer network again. Since we believe the issues in our results are caused by an issue in the implementation of the reversible crosslinks on the GPU, we see two options: simulations can be performed on the CPU, or the software issue can be patched. The latter option would have our preference, since the relaxation simulations are very expensive, and so would take extremely long on CPUs. It would additionally open up the option to use the GPU for other purposes, such as determining swap time distributions.

Bibliography

- [1] Simone Ciarella and Wouter G. Ellenbroek. ‘Swap-driven self-adhesion and healing of vitrimers’. In: *Coatings* 9.2 (2019), pp. 1–12. ISSN: 20796412. DOI: [10.3390/COATINGS9020114](https://doi.org/10.3390/COATINGS9020114).
- [2] Directorate General for Communication. *Circular economy: definition, importance and benefits*. May 2023. URL: <https://www.europarl.europa.eu/news/en/headlines/economy/20151201ST005603/circular-economy-definition-importance-and-benefits>.
- [3] Directorate General for Communication. *Waste management in the EU : infographic with facts and figures*. June 2023. URL: <https://www.europarl.europa.eu/news/en/headlines/society/20180328ST000751/waste-management-in-the-eu-infographic-with-facts-and-figures>.
- [4] Directorate General for Communication. *How the EU wants to achieve a circular economy by 2050*. Jan. 2023. URL: <https://www.europarl.europa.eu/news/en/headlines/society/20210128ST096607/how-the-eu-wants-to-achieve-a-circular-economy-by-2050#:~:text=How%20the%20EU%20wants%20to%20achieve%20a%20circular,circular%20...%204%20Waste%20management%20and%20shipment>.
- [5] Directorate General for Communication. *Plastic waste and recycling in the EU: facts and figures*. Jan. 2023. URL: <https://www.europarl.europa.eu/news/en/headlines/society/20181212ST021610/plastic-waste-and-recycling-in-the-eu-facts-and-figures>.
- [6] Xiuhong Li, Yujie Peng, Yichen Deng, Fangping Ye, Chupeng Zhang, Xinyu Hu, Yong Liu and Daode Zhang. ‘Recycling and Reutilizing Polymer Waste via Electrospun Micro/Nanofibers: A Review’. In: *Nanomaterials* 12.10 (2022), pp. 1–18. ISSN: 20794991. DOI: [10.3390/nano12101663](https://doi.org/10.3390/nano12101663).
- [7] Lucie Imbernon and Sophie Norvez. ‘From landfilling to vitrimer chemistry in rubber life cycle’. In: *European Polymer Journal* 82 (2016), pp. 347–376. ISSN: 00143057. DOI: [10.1016/j.eurpolymj.2016.03.016](https://doi.org/10.1016/j.eurpolymj.2016.03.016).
- [8] Shi Yin, Rabin Tuladhar, Feng Shi, Robert A. Shanks, Mark Combe and Tony Collister. ‘Mechanical reprocessing of polyolefin waste: A review’. In: *Polymer Engineering & Science* 55.12 (Dec. 2015), pp. 2899–2909. ISSN: 00323888. DOI: [10.1002/pen.24182](https://doi.org/10.1002/pen.24182). URL: <https://onlinelibrary.wiley.com/doi/10.1002/pen.24182>.
- [9] Kai Yu, Philip Taynton, Wei Zhang, Martin L. Dunn and H. Jerry Qi. ‘Reprocessing and recycling of thermosetting polymers based on bond exchange reactions’. In: *RSC Advances* 4.20 (2014), pp. 10108–10117. ISSN: 20462069. DOI: [10.1039/c3ra47438k](https://doi.org/10.1039/c3ra47438k).
- [10] Wim Denissen, Johan M. Winne and Filip E. Du Prez. ‘Vitrimers: Permanent organic networks with glass-like fluidity’. In: *Chemical Science* 7.1 (2016), pp. 30–38. ISSN: 20416539. DOI: [10.1039/c5sc02223a](https://doi.org/10.1039/c5sc02223a).
- [11] Hengheng Zhao, Xuefeng Wei, Yue Fang, Ke Gao, Tongkui Yue, Liqun Zhang, Venkat Ganesan, Fanlong Meng and Jun Liu. ‘Molecular Dynamics Simulation of the Structural, Mechanical, and Reprocessing Properties of Vitrimers Based on a Dynamic Covalent Polymer Network’. In: *Macromolecules* 55.4 (2022), pp. 1091–1103. ISSN: 15205835. DOI: [10.1021/acs.macromol.1c02034](https://doi.org/10.1021/acs.macromol.1c02034).
- [12] Tingting Song, Bojun Jiang, Yida Li, Zhuohan Ji, Hongling Zhou, Dawei Jiang, Ilwoo Seok, Vignesh Murugadoss, Nan Wen and Henry Colorado. ‘Self-healing Materials: A Review of Recent Developments’. In: *ES Materials and Manufacturing* 14 (2021), pp. 1–19. ISSN: 2578062X. DOI: [10.30919/esmm5f465](https://doi.org/10.30919/esmm5f465).
- [13] By Martin D Hager, Peter Greil, Christoph Leyens, Sybrand Van Der Zwaag and Ulrich S Schubert. ‘Self-Healing Materials’. In: *Advanced Materials* 22.47 (2010), pp. 5424–5430. ISSN: 0935-9648. DOI: [10.1002/adma.201003036](https://doi.org/10.1002/adma.201003036).

- [14] Jiheong Kang, Donghee Son, Ging Ji Nathan Wang, Yuxin Liu, Jeffrey Lopez, Yeongin Kim, Jin Young Oh, Toru Katsumata, Jaewan Mun, Yeongjun Lee, Lihua Jin, Jeffrey B.H. Tok and Zhenan Bao. ‘Tough and Water-Insensitive Self-Healing Elastomer for Robust Electronic Skin’. In: *Advanced Materials* 30.13 (2018), pp. 1–8. ISSN: 15214095. DOI: [10.1002/adma.201706846](https://doi.org/10.1002/adma.201706846).
- [15] L. Hollaway. ‘Polymeric materials’. In: *Polymer Composites for Civil and Structural Engineering*. 1st ed. Springer Science+Business Media Dordrecht, 1993. Chap. 2.7, pp. 23–27. ISBN: 978-94-010-4946-7. DOI: [10.1007/978-94-011-2136-1](https://doi.org/10.1007/978-94-011-2136-1).
- [16] K. Horie, M. Barón, R. B. Fox, J. He, M. Hess, J. Kahovec, T. Kitayama, P. Kubisa, E. Maréchal, W. Mormann, R. F.T. Stepto, D. Tabak, J. Vohlídal, E. S. Wilks and W. J. Work. ‘Definitions of terms relating to reactions of polymers and to functional polymeric materials: (iupac recommendations 2003)’. In: *Pure Applied Chemistry* 76.4 (2004), pp. 889–906. DOI: [10.1351/pac200476040889](https://doi.org/10.1351/pac200476040889).
- [17] Ludwik Leibler, Damien Montarnal, Mathieu Capelot and François Tournilhac. ‘Silica-like malleable materials from permanent organic networks’. In: *Science* 334.6058 (2011), pp. 965–968. ISSN: 10959203. DOI: [10.1126/science.1212648](https://doi.org/10.1126/science.1212648).
- [18] J. Alemán, A. V. Chadwick, J. He, M. Hess, K. Horie, R. G. Jones, P. Kratochvíl, I. Meisel, I. Mita, G. Moad, S. Penczek and R. F.T. Stepto. ‘Definitions of terms relating to the structure and processing of sols, gels, networks, and inorganic-organic hybrid materials (IUPAC recommendations 2007)’. In: *Pure and Applied Chemistry* 79.10 (2007), pp. 1801–1829. ISSN: 00334545. DOI: [10.1351/pac200779101801](https://doi.org/10.1351/pac200779101801).
- [19] A. R. Rennie. ‘Thermoplastics and Thermosets’. In: *Mechanical Properties and Testing of Polymers*. Ed. by G. M. Swallowe. Springer Science+Business Media Dordrecht, 1999. Chap. 53, p. 248. ISBN: 9789048140244. DOI: [10.1007/978-94-015-9231-4](https://doi.org/10.1007/978-94-015-9231-4).
- [20] Robyn H. Pritchard, Anna Lena Redmann, Zhiqiang Pei, Yan Ji and Eugene M. Terentjev. ‘Vitrification and plastic flow in transient elastomer networks’. In: *Polymer* 95 (2016), pp. 45–51. ISSN: 00323861. DOI: [10.1016/j.polymer.2016.04.060](https://doi.org/10.1016/j.polymer.2016.04.060).
- [21] Christopher J. Kloxin, Timothy F. Scott, Brian J. Adzima and Christopher N. Bowman. ‘Covalent adaptable networks (CANs): A unique paradigm in cross-linked polymers’. In: *Macromolecules* 43.6 (2010), pp. 2643–2653. ISSN: 00249297. DOI: [10.1021/ma902596s](https://doi.org/10.1021/ma902596s).
- [22] Evgeny B. Stukalin, Li Heng Cai, N. Arun Kumar, Ludwik Leibler and Michael Rubinstein. ‘Self-healing of unentangled polymer networks with reversible bonds’. In: *Macromolecules* 46.18 (2013), pp. 7525–7541. ISSN: 00249297. DOI: [10.1021/ma401111n](https://doi.org/10.1021/ma401111n).
- [23] Chiara Raffaelli, Anwesha Bose, Cyril H. M. P. Vrusch, Simone Ciarella, Theodoros Davris, Nicholas B. Tito, Alexey V. Lyulin, Wouter G. Ellenbroek and Cornelis Storm. ‘Rheology, Rupture, Reinforcement and Reversibility: Computational Approaches for Dynamic Network Materials’. In: *Self-Healing and Self-Recovering Hydrogels*. Ed. by Costantino Creton and Oguz Okay. Cham, Switzerland: Springer Nature Switzerland, 2020. Chap. 3, pp. 64–113.
- [24] Simone Ciarella, Francesco Sciortino and Wouter G. Ellenbroek. ‘Dynamics of Vitrimers: Defects as a Highway to Stress Relaxation’. In: *Physical Review Letters* 121.5 (2018), p. 58003. ISSN: 10797114. DOI: [10.1103/PhysRevLett.121.058003](https://doi.org/10.1103/PhysRevLett.121.058003).
- [25] Michael Rubinstein and Ralph H. Colby. ‘Entangled Polymer Dynamics’. In: *Polymer Physics*. Oxford University Press, 2003. Chap. 9, pp. 361–421. ISBN: 978-0-19-852059-7.
- [26] Alexander Stukowski. ‘Visualization and analysis of atomistic simulation data with OVITO—the Open Visualization Tool’. In: *Modelling and Simulation in Materials Science and Engineering* 18.1 (2010). ISSN: 09650393. DOI: [10.1088/0965-0393/18/1/015012](https://doi.org/10.1088/0965-0393/18/1/015012).
- [27] Junzo Otera. ‘Transesterification’. In: *Chemical Reviews* 93.4 (1993), pp. 1449–1470. ISSN: 0009-2665/93/0793-1449.
- [28] Simone Ciarella and Wouter G. Ellenbroek. ‘Associative bond swaps in molecular dynamics’. In: *SciPost Physics* 12.4 (2022). ISSN: 25424653. DOI: [10.21468/SCIPOSTPHYS.12.4.128](https://doi.org/10.21468/SCIPOSTPHYS.12.4.128).
- [29] Peter J. Smith. ‘Survival Distributions’. In: *Analysis of Failure and Survival Data*. 1st ed. New York City, USA: CRC Press, 2002. Chap. 1, pp. 1–18. ISBN: 9781315273150.
- [30] Richard Peto. ‘Experimental Survival Curves for Interval-Censored Data’. In: *Journal of the Royal Statistical Society* 22.1 (1973), pp. 86–91.
- [31] Paul Meier E.L. Kaplan. ‘Nonparametric Estimation from Incomplete Observations’. In: *American statistical Association* 53.282 (1958), pp. 457–481. DOI: [10.1080/01621459.1958.10501452](https://doi.org/10.1080/01621459.1958.10501452).
- [32] Alan B. Cantor. ‘Projecting the standard error of the Kaplan-Meier estimator’. In: *Statistics in Medicine* 20.14 (2001), pp. 2091–2097. ISSN: 02776715. DOI: [10.1002/sim.856](https://doi.org/10.1002/sim.856).
- [33] Geoffrey Grimmett and David Stirzaker. ‘Continuous Random Variables’. In: *Probability and Random Processes*. 3rd ed. Oxford, UK: Oxford University Press, 2001. Chap. 4, pp. 89–147.

- [34] Witold A.J. Kosmala. ‘Integration’. In: *A Friendly Introduction to Analysis*. 2nd ed. Upper Saddle River, NJ, USA: Pearson, 2004. Chap. 6, pp. 240–293. ISBN: 0-13-045796-5.
- [35] Sabine Diehl and Winfried Stute. ‘Kernel density and hazard function estimation in the presence of censoring’. In: *Journal of Multivariate Analysis* 25.2 (1988), pp. 299–310. ISSN: 10957243. DOI: [10.1016/0047-259X\(88\)90053-X](https://doi.org/10.1016/0047-259X(88)90053-X).
- [36] Jan Mielniczuk. ‘Some Asymptotic Properties of Kernel Estimators of a Density Function in Case of Censored Data’. In: *The Annals of Statistics* 14.2 (1986), pp. 766–773. URL: <http://www.jstor.org/stable/2241251>.
- [37] K. B. Kulasekera and W. J. Padgett. ‘Bayes bandwidth selection in kernel density estimation with censored data’. In: *Journal of Nonparametric Statistics* 18.2 (2006), pp. 129–143. ISSN: 10485252. DOI: [10.1080/10485250600556744](https://doi.org/10.1080/10485250600556744).
- [38] Murray Rosenblatt. ‘Remarks on Some Nonparametric Estimates of a Density Function’. In: *The Annals of Mathematical Statistics* 27.3 (1956), pp. 832–837. DOI: [10.1214/aoms/1177728190](https://doi.org/10.1214/aoms/1177728190).
- [39] Emanuel Parzen. ‘On Estimation of Probability Density Functions and Mode’. In: *The Annals of Mathematical Statistics* 33.3 (1962), pp. 1065–1076. DOI: [10.1214/aoms/1177704472](https://doi.org/10.1214/aoms/1177704472).
- [40] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Matthew van der Walt Stéfan J.Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, Ilhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt and SciPy 1.0 Contributors. ‘SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python’. In: *Nature Methods* 17.3 (2020), pp. 261–272. DOI: [10.1038/s41592-019-0686-2](https://doi.org/10.1038/s41592-019-0686-2).
- [41] Alexandre B Tsybakov. ‘Nonparametric estimators’. In: *Introduction to Nonparametric Estimation*. Paris, France: Springer, 2009. Chap. 1, pp. 1–76. ISBN: 978-0-387-79051-0. DOI: [10.1007/978-0-387-79052-7](https://doi.org/10.1007/978-0-387-79052-7).
- [42] Geoffrey Grimmett and David Stirzaker. ‘Markov Chains’. In: *Probability and Random Processes*. 3rd ed. Oxford, UK: Oxford University Press, 2001. Chap. 6, pp. 213–304.
- [43] Edsel A. Peña and Vijay K. Rohatgi. ‘Small sample and efficiency results for the Nelson-Aalen estimator’. In: *Journal of Statistical Planning and Inference* 37.2 (1993), pp. 193–202. ISSN: 03783758. DOI: [10.1016/0378-3758\(93\)90088-N](https://doi.org/10.1016/0378-3758(93)90088-N).
- [44] Witold A.J. Kosmala. ‘Sequences’. In: *A Friendly Introduction to Analysis*. 2nd ed. Upper Saddle River, NJ, USA: Pearson, 2004. Chap. 2, pp. 65–116. ISBN: 0-13-045796-5.
- [45] Theodore Karrison. ‘Restricted mean life with adjustment for covariates’. In: *Journal of the American Statistical Association* 82.400 (1987), pp. 1169–1176. ISSN: 01972456. DOI: [10.1080/0197-2456\(86\)90108-X](https://doi.org/10.1080/0197-2456(86)90108-X).
- [46] Sarwar I. Mozumder, Mark J. Rutherford and Paul C. Lambert. ‘Estimating restricted mean survival time and expected life-years lost in the presence of competing risks within flexible parametric survival models’. In: *BMC Medical Research Methodology* 21.1 (2021), pp. 1–20. ISSN: 14712288. DOI: [10.1186/s12874-021-01213-0](https://doi.org/10.1186/s12874-021-01213-0).
- [47] Michael Rubinstein and Ralph H. Colby. ‘Networks and Gels’. In: *Polymer Physics*. Oxford University Press, 2003. Chap. 7, pp. 253–305. ISBN: 978-0-19-852059-7.
- [48] Gert Strobl. ‘Mechanical and Dielectric Response’. In: *The Physics of Polymers: Concepts for Understanding Their Structures and Behavior*. 3rd ed. Springer, 2007. Chap. 6, pp. 223–286. ISBN: 9783540252788. DOI: [10.1007/978-3-540-68411-4](https://doi.org/10.1007/978-3-540-68411-4).
- [49] Alessandro Pergo and Fardin Khabaz. ‘Volumetric and Rheological Properties of Vitrimers: A Hybrid Molecular Dynamics and Monte Carlo Simulation Study’. In: *Macromolecules* 53.19 (2020), pp. 8406–8416. ISSN: 15205835. DOI: [10.1021/acs.macromol.0c01423](https://doi.org/10.1021/acs.macromol.0c01423).
- [50] Michael Rubinstein and Ralph H. Colby. ‘Ideal Chains’. In: *Polymer Physics*. Oxford University Press, 2003. Chap. 2, pp. 49–96. ISBN: 978-0-19-852059-7.
- [51] Simone Ciarella. ‘Relaxation Pathways for Soft Materials’. PhD thesis. Eindhoven University of Technology, 2021, pp. 1–218. ISBN: 978-90-386-5205-4.
- [52] Daan Frenkel and Berend Smit. ‘Monte Carlo Simulations’. In: *Understanding Molecular Simulation*. 2nd ed. 2002. Chap. 3, pp. 23–61. DOI: [10.1016/B978-012267351-1/50005-5](https://doi.org/10.1016/B978-012267351-1/50005-5).
- [53] Denis J. Evans and Gary Morriss. ‘Computer simulation algorithms’. In: *Statistical Mechanics of Nonequilibrium Liquids*. 2nd ed. Cambridge University Press, 2009. Chap. 6, pp. 119–166. ISBN: 9780511535307. DOI: [10.1017/cbo9780511535307.008](https://doi.org/10.1017/cbo9780511535307.008).

- [54] Jian Bo Wu, Shu Jia Li, Hong Liu, Hu Jun Qian and Zhong Yuan Lu. ‘Dynamics and reaction kinetics of coarse-grained bulk vitrimers: A molecular dynamics study’. In: *Physical Chemistry Chemical Physics* 21.24 (2019), pp. 13267–13276. ISSN: 14639076. DOI: [10.1039/c9cp01766f](https://doi.org/10.1039/c9cp01766f).
- [55] Francesco Sciortino. ‘Three-body potential for simulating bond swaps in molecular dynamics’. In: *The European Physical Journal E* 40.1 (2017), pp. 3–6. ISSN: 1292895X. DOI: [10.1140/epje/i2017-11496-5](https://doi.org/10.1140/epje/i2017-11496-5).
- [56] Jean Louis Barrat, Jörg Baschnagel and Alexey Lyulin. ‘Molecular dynamics simulations of glassy polymers’. In: *Soft Matter* 6.15 (2010), pp. 3430–3446. ISSN: 17446848. DOI: [10.1039/b927044b](https://doi.org/10.1039/b927044b).
- [57] Chanwook Park, Geonwoo Kim, Jiwon Jung, Balaji Krishnakumar, Sravendra Rana and Gun Jin Yun. ‘Enhanced self-healing performance of graphene oxide/vitrimer nanocomposites: A molecular dynamics simulations study’. In: *Polymer* 206.122862 (2020), pp. 1–11. ISSN: 00323861. DOI: [10.1016/j.polymer.2020.122862](https://doi.org/10.1016/j.polymer.2020.122862).
- [58] Hua Yang, Kai Yu, Xiaoming Mu, Xinghua Shi, Yujie Wei, Yafang Guo and H. Jerry Qi. ‘A molecular dynamics study of bond exchange reactions in covalent adaptable networks’. In: *Soft Matter* 11.31 (2015), pp. 6305–6317. ISSN: 17446848. DOI: [10.1039/c5sm00942a](https://doi.org/10.1039/c5sm00942a).
- [59] Yaguang Sun, Hua Yang, Kai Yu, Yafang Guo and Jianmin Qu. ‘A molecular dynamics study of decomposition of covalent adaptable networks in organic solvent’. In: *Polymer* 180.April (2019), p. 121702. ISSN: 00323861. DOI: [10.1016/j.polymer.2019.121702](https://doi.org/10.1016/j.polymer.2019.121702). URL: <https://doi.org/10.1016/j.polymer.2019.121702>.
- [60] Lorenzo Rovigatti, John Russo and Flavio Romano. ‘How to Simulate Patchy Particles’. In: *The European Physical Journal E* 41.59 (2018), pp. 1–12. DOI: [10.1140/epje/i2018-11667-x](https://doi.org/10.1140/epje/i2018-11667-x).
- [61] Joshua A. Anderson, Jens Glaser and Sharon C. Glotzer. ‘HOOMD-blue: A Python package for high-performance molecular dynamics and hard particle Monte Carlo simulations’. In: *Computational Materials Science* 173.109363 (2020), pp. 1–6. ISSN: 09270256. DOI: [10.1016/j.commatsci.2019.109363](https://doi.org/10.1016/j.commatsci.2019.109363).
- [62] John Edward Lennard-Jones. ‘Cohesion’. In: *THE PROCEEDINGS OF THE PHYSICAL SOCIETY* 43.461 (1931), pp. 461–482.
- [63] Ashley H. Carter. ‘Classical and Quantum Statistics’. In: *Classical and Statistical Thermodynamics*. Pearson, 2001. Chap. 13, pp. 233–260.
- [64] John Robert Taylor. *Classical Mechanics*. Springer New York, 2005, pp. 1–786. ISBN: 1-891389-22-X. URL: http://link.springer.com/10.1007/0-387-25112-X_2.
- [65] William C. Swope, Hans C. Andersen, Peter H. Berens and Kent R. Wilson. ‘A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters’. In: *The Journal of Chemical Physics* 76 (1982), pp. 637–649. DOI: [10.1063/1.442716](https://doi.org/10.1063/1.442716).
- [66] Glenn J Martyna and Mark E Tuckerman. ‘Symplectic reversible integrators : Predictor – corrector methods’. In: *The Journal of Chemical Physics* 102.20 (1995), pp. 8071–8077. DOI: [10.1063/1.469006](https://doi.org/10.1063/1.469006).
- [67] Ronald D Ruth. ‘A Canonical Integration Technique’. In: *IEEE Transactions on Nuclear Science* 30.4 (1983), pp. 2669–2671. DOI: [10.1109/TSNS.1983.4332919](https://doi.org/10.1109/TSNS.1983.4332919).
- [68] Hiqmet Kamberaj, R. J. Low and M. P. Neal. ‘Time reversible and symplectic integrators for molecular dynamics simulations of rigid molecules’. In: *Journal of Chemical Physics* 122.22 (2005). ISSN: 00219606. DOI: [10.1063/1.1906216](https://doi.org/10.1063/1.1906216).
- [69] Ernst Hairer, Christian Lubich and Gerhard Wanner. ‘Geometric numerical integration illustrated by the Störmer-Verlet method’. In: *Acta Numerica* 12 (2003), pp. 399–450. ISSN: 09624929. DOI: [10.1017/S0962492902000144](https://doi.org/10.1017/S0962492902000144).
- [70] Hiqmet Kamberaj. *Molecular Dynamics Simulations in Statistical Physics: Theory and Applications*. Springer, 2020, pp. 1–463. ISBN: 9783030357016. URL: <http://link.springer.com/10.1007/978-3-030-35702-3>.
- [71] Daan Frenkel and Berend Smit. ‘Molecular Dynamics in Various Ensembles’. In: *Understanding Molecular Simulation*. 2nd ed. 2002. Chap. 6, pp. 139–163. DOI: [10.1016/b978-0-12267351-1/50008-0](https://doi.org/10.1016/b978-0-12267351-1/50008-0).
- [72] Michael Rubinstein and Ralph H. Colby. *Polymer Physics*. Oxford University Press, 2003. ISBN: 978-0-19-852059-7.
- [73] Michael Rubinstein and Ralph H. Colby. ‘Unentangled Polymer Dynamics’. In: *Polymer Physics*. Oxford University Press, 2003. Chap. 8, pp. 309–360. ISBN: 978-0-19-852059-7.
- [74] Gert Strobl. ‘Microscopic Dynamics’. In: *The Physics of Polymers: Concepts for Understanding Their Structures and Behavior*. 3rd ed. Springer, 2007. Chap. 8, pp. 313–356. ISBN: 9783540252788. DOI: [10.1007/978-3-540-68411-4](https://doi.org/10.1007/978-3-540-68411-4).

- [75] Lorenzo Rovigatti, Giovanni Nava, Tommaso Bellini and Francesco Sciortino. ‘Self-Dynamics and Collective Swap-Driven Dynamics in a Particle Model for Vitrimers’. In: *Macromolecules* 51.3 (2018), pp. 1232–1241. ISSN: 15205835. DOI: [10.1021/acs.macromol.7b02186](https://doi.org/10.1021/acs.macromol.7b02186).
- [76] Geoffrey Grimmett and David Stirzaker. ‘Discrete Random Variables’. In: *Probability and Random Processes*. 3rd ed. Oxford, UK: Oxford University Press, 2001. Chap. 3, pp. 46–88.
- [77] Witold A.J. Kosmala. ‘Sequences and Series of Functions’. In: *A Friendly Introduction to Analysis*. 2nd ed. Upper Saddle River, NJ, USA: Pearson, 2004. Chap. 8, pp. 334–382. ISBN: 0-13-045796-5.
- [78] Witold A.J. Kosmala. ‘Infinite Series’. In: *A Friendly Introduction to Analysis*. 2nd ed. Upper Saddle River, NJ, USA: Pearson, 2004. Chap. 7, pp. 294–333. ISBN: 0-13-045796-5.
- [79] Geoffrey Grimmett and David Stirzaker. ‘Events and Their Probabilities’. In: *Probability and Random Processes*. 3rd ed. Oxford, UK: Oxford University Press, 2001. Chap. 1, pp. 1–25.
- [80] Ashley H. Carter. ‘The Nature of Thermodynamics’. In: *Classical and Statistical Thermodynamics*. Pearson, 2001. Chap. 1, pp. 1–18. ISBN: 9780137792085.
- [81] Sidath Wijesinghe, Dvora Perahia, Ting Ge, K. Michael Salerno and Gary S. Grest. ‘Stress Relaxation of Comb Polymer Melts’. In: *Tribology Letters* 69.2 (2021), pp. 1–8. ISSN: 15732711. DOI: [10.1007/s11249-021-01432-y](https://doi.org/10.1007/s11249-021-01432-y). URL: <https://doi.org/10.1007/s11249-021-01432-y>.
- [82] Witold A.J. Kosmala. ‘Iterated Integrals’. In: *A Friendly Introduction to Analysis*. 2nd ed. Upper Saddle River, NJ, USA: Pearson, 2004. Chap. 11, pp. 480–531.
- [83] Michael Renardy and Robert C Rogers. ‘An Introduction to Partial Differential Equations’. In: *An Introduction to Partial Differential Equations*. Ed. by J. E. Marsden, L. Sirovich and S. S. Antman. 2nd ed. Springer, 2004. Chap. 8, pp. 228–282. ISBN: 0387004440. DOI: [10.1007/b97427](https://doi.org/10.1007/b97427).
- [84] Witold A.J. Kosmala. ‘Introduction’. In: *A Friendly Introduction to Analysis*. 2nd ed. Upper Saddle River, NJ, USA: Pearson, 2004. Chap. 1, pp. 1–64. ISBN: 0-13-045796-5.

Appendix A

Supplementary Proofs

A.1 Tails of Improper Riemann Integrals

The following lemma will allow us to prove the correctness of Formula 2.1.13 for computing the mean from the survival function.

Lemma A.1.1 (Tails of Improper Riemann Integrals). *Let $f : [a, \infty) \rightarrow \mathbb{R}$ be improperly Riemann integrable, i.e. f is Riemann integrable on $[a, b]$ for $b \in \mathbb{R}_{>0}$ and $\lim_{b \rightarrow \infty} \int_a^b f(x)dx = A < \infty$ [34, Def. 6.5.1]. Then, the tail of the improper integral tends to zero, i.e.*

$$\lim_{t \rightarrow \infty} \int_t^\infty f(x)dx = 0. \quad (\text{A.1.1})$$

Proof. We start by noting that when f is Riemann integrable on $[a, b]$, f is also integrable on $[a, t]$ and $[t, b]$ for $a < t < b$: $\int_t^b f(x)dx$ is well-defined for such t and b , and is given by [34, Thm. 6.3.3]

$$\int_t^b f(x)dx = \int_a^b f(x)dx - \int_a^t f(x)dx.$$

It then follows that

$$\int_t^\infty f(x)dx = \lim_{b \rightarrow \infty} \int_t^b f(x)dx = \lim_{b \rightarrow \infty} \int_a^b f(x)dx - \int_a^t f(x)dx = A - \int_a^t f(x)dx,$$

and from this we may find that

$$\lim_{t \rightarrow \infty} \int_t^\infty f(x)dx = A - \lim_{t \rightarrow \infty} \int_a^t f(x)dx = A - A = 0,$$

as required. □

A.2 Chain Length Distributions

In this appendix, we derive an expression for the exact PDF of the length of a chain of particles each separated by a uniformly distributed distance. To compute this PDF, we will need to convolve functions. The following lemma shows that the convolution will be well-behaved.

Lemma A.2.1 (Convolutions on L^p -Spaces). *Let $f \in L^1(\mathbb{R})$, with $f \geq 0$ almost everywhere (a.e.) and $\|f\|_1 = 1$, and let $g \in L^p(\mathbb{R})$ for some $p \in [1, \infty)$. Then $f * g \in L^p(\mathbb{R})$ and*

$$\|f * g\|_p \leq \|f\|_1 \|g\|_p \leq \|g\|_p. \quad (\text{A.2.1})$$

Proof. Let $f \in L^1(\mathbb{R})$, with $f \geq 0$ a.e. and $\|f\|_1 = 1$, and let $g \in L^p(\mathbb{R})$ for some $p \in [1, \infty)$. Define $p' \in (1, \infty]$ such that $\frac{1}{p} + \frac{1}{p'} = 1$. Clearly it then holds that $f = f^{\frac{1}{p}} f^{\frac{1}{p'}}$. Furthermore, we may find that

$$\int_{\mathbb{R}} \int_{\mathbb{R}} f(x-y) |g(y)|^p dx dy = \int_{\mathbb{R}} \underbrace{\int_{\mathbb{R}} f(x-y) dx}_{=\|f\|_1=1} |g(y)|^p dy = \int_{\mathbb{R}} |g(y)|^p dy = \|g\|_p^p < \infty.$$

Then, Fubini's Theorem [82, Thm. 11.2.2] implies that $y \mapsto f(x-y)|g(y)|^p \in L^{\mathbb{R}}(1)$, or, equivalently, that $y \mapsto f(x-y)^{\frac{1}{p}} * |g(y)| \in L^{\mathbb{R}}(p)$. Next, we can apply Hölder's inequality to see that

$$\begin{aligned} \int_{\mathbb{R}} f(x-y) |g(y)| dy &= \int_{\mathbb{R}} \underbrace{f(x-y)^{\frac{1}{p}} |g(y)|}_{\in L^p(\mathbb{R})} \underbrace{f(x-y)^{\frac{1}{p'}} dy}_{\in L^{p'}(\mathbb{R})} \\ &\leq \left(\int_{\mathbb{R}} f(x-y) |g(y)|^p dy \right)^{\frac{1}{p}} \underbrace{\left(\int_{\mathbb{R}} f(x-y) dy \right)}_{\|f\|_1=1}^{\frac{1}{p'}} \\ &= \left(\int_{\mathbb{R}} f(x-y) |g(y)|^p dy \right)^{\frac{1}{p}} = \|y \mapsto f(x-y)|g(y)|^p\|_1^{\frac{1}{p}} < \infty, \end{aligned}$$

Hence, $f * g$ is a well-defined function. We can now confirm that

$$\begin{aligned} \|f * g\|_p^p &= \int_{\mathbb{R}} \left| \int_{\mathbb{R}} f(x-y) g(y) dy \right|^p dx \leq \int_{\mathbb{R}} \left| \left(\int_{\mathbb{R}} f(x-y) |g(y)|^p dy \right)^{\frac{1}{p}} \right|^p dx \\ &= \int_{\mathbb{R}} \int_{\mathbb{R}} f(x-y) |g(y)|^p dx = \|g\|_p^p. \end{aligned}$$

We may finally conclude that $f * g \in L^{\mathbb{R}}(p)$ and $\|f * g\|_p \leq \|f\|_1 \|g\|_p \leq \|g\|_p$, as required. \square

We are now ready to discuss the main result of this appendix:

Theorem A.2.2 (Density of Sum of Uniform Random Variables). *Let $X_i \sim \text{Uni}[0, 1]$ i.i.d. for $i = 1, \dots, n$. Then the density of $\sum_{i=1}^n X_i$ is given by*

$$f_{\sum_{i=1}^n X_i}(x) = \frac{1}{(n-1)!} \sum_{k=0}^{n-1} (-1)^k \binom{n}{k} (x-k)^{n-1} \mathbf{1}\{x \in [k, n]\}. \quad (\text{A.2.2})$$

Proof. We will prove Equation (A.2.2) using induction.

base Consider that $\sum_{i=1}^1 X_i = X_1$, so that

$$f_{\sum_{i=1}^1 X_i}(x) = f_X(x) = \mathbf{1}\{x \in [0, 1]\},$$

the density of a random variable uniformly distributed on $[0, 1]$. We next note that

$$\begin{aligned} \frac{1}{(n-1)!} \sum_{k=0}^{n-1} (-1)^k \binom{n}{k} (x-k)^{n-1} \mathbf{1}\{x \in [k, n]\} \Big|_{n=1} &= \frac{1}{0!} \sum_{k=0}^0 (-1)^k \binom{1}{k} (x-k)^0 \mathbf{1}\{x \in [k, 1]\} \\ &= \mathbf{1}\{x \in [0, 1]\}, \end{aligned}$$

so that indeed Equation (A.2.2) holds for $n = 1$.

step Suppose that there is some $N \in \mathbb{N}$ such that Equation (A.2.2) holds for $n = 1, \dots, N$. Then, consider that by the Law of Total Probability

$$\begin{aligned} f_{\sum_{i=1}^{N+1} X_i}(x) &= \frac{d}{dx} \mathbb{P}\left(\sum_{i=1}^{N+1} X_i \leq x\right) = \frac{d}{dx} \int_{-\infty}^{\infty} \mathbb{P}\left(\sum_{i=1}^{N+1} X_i \leq x \mid X_{N+1} = y\right) f_X(y) dy \\ &= \frac{d}{dx} \int_{-\infty}^{\infty} \mathbb{P}\left(\sum_{i=1}^N X_i \leq x - y\right) f_X(y) dy \stackrel{(I)}{=} \int_{-\infty}^{\infty} \frac{d}{dx} \mathbb{P}\left(\sum_{i=1}^N X_i \leq x - y\right) f_X(y) dy \\ &= \int_{-\infty}^{\infty} f_{\sum_{i=1}^N X_i}(x-y) f_X(y) dy = (f_{\sum_{i=1}^N X_i} * f_X)(x). \end{aligned}$$

Note that $\mathbb{P}\left(\sum_{i=1}^N X_i \leq \cdot\right) \in L^\infty(\mathbb{R})$ and $f_X \in L^1(\mathbb{R})$, so that the integral on the left of (I) is convergent by Hölder's inequality, while $f_{\sum_{i=1}^N X_i} \in L^1(\mathbb{R})$, so that the convolution on the right (I) is convergent by Lemma A.2.1. Since differentiation $\frac{d}{dx}$ is a closed operator¹, it then follows that we are able to exchange integration and differentiation in (I)².

By assumption we know that

$$f_{\sum_{i=1}^N X_i}(x) = \frac{1}{(N-1)!} \sum_{k=0}^{N-1} (-1)^k \binom{N}{k} (x-k)^{N-1} \mathbf{1}\{x \in [k, N]\},$$

and $f_X(x) = \mathbf{1}\{x \in [0, 1]\}$, so that

$$\begin{aligned} f_{\sum_{i=1}^{N+1} X_i}(x) &= (f_{\sum_{i=1}^N X_i} * f_X)(x) = \int_{-\infty}^{\infty} f_{\sum_{i=1}^N X_i}(x-y) f_X(y) dy \\ &= \frac{1}{(N-1)!} \sum_{k=0}^{N-1} (-1)^k \binom{N}{k} \int_{-\infty}^{\infty} (x-y-k)^{N-1} \mathbf{1}\{x-y \in [k, N]\} \mathbf{1}\{y \in [0, 1]\} dy \\ &= \frac{1}{(N-1)!} \sum_{k=0}^{N-1} (-1)^k \binom{N}{k} \int_0^1 (x-y-k)^{N-1} \mathbf{1}\{x-y \in [k, N]\} dy. \end{aligned}$$

We next consider the above integral for a fixed $k \in \{0, \dots, N\}$. We substitute $u = x - y$, to see that

$$\begin{aligned} \int_0^1 (x-y-k)^{N-1} \mathbf{1}\{x-y \in [k, N]\} dy &= - \int_x^{x-1} (u-k)^{N-1} \mathbf{1}\{u \in [k, N]\} du \\ &= \int_{x-1}^x (u-k)^{N-1} \mathbf{1}\{u \in [k, N]\} du \\ &= \int_{\max\{x-1, k\}}^{\min\{x, N\}} (u-k)^{N-1} du \\ &= \frac{1}{N} \begin{cases} 0, & x < k, \\ (x-k)^N, & k \leq x < k+1 \\ (x-k)^N - (x-(k+1))^N, & k+1 \leq x < N \\ (N-k)^N - (x-(k+1))^N, & N \leq x \leq N+1 \\ 0, & x > N+1. \end{cases} \end{aligned}$$

Now, let $k' \in \{0, \dots, N-1\}$, and $x \in [k', k'+1)$. Then we may find that

$$f_{\sum_{i=1}^{N+1} X_i}(x) = \frac{1}{N!} \left((-1)^{k'} \binom{N}{k'} (x-k')^N + \sum_{k=0}^{k'-1} (-1)^k \binom{N}{k} ((x-k)^N - (x-(k+1))^N) \right).$$

¹on the Sobolev space H^1 [83, Ex. 8.29].

²as a consequence of [83, Lem. 8.26], since the integral may be seen as the limit of a sequence.

Note that

$$\begin{aligned}
\sum_{k=0}^{k'-1} (-1)^k \binom{N}{k} ((x-k)^N - (x-(k+1))^N) &= \sum_{k=0}^{k'-1} (-1)^k \binom{N}{k} (x-k)^N \\
&\quad - \sum_{k=0}^{k'-1} (-1)^k \binom{N}{k} (x-(k+1))^N \\
&= \sum_{k=0}^{k'-1} (-1)^k \binom{N}{k} (x-k)^N \\
&\quad + \sum_{k=1}^{k'} (-1)^k \binom{N}{k-1} (x-k)^N \\
&= x^N + \sum_{k=1}^{k'-1} (-1)^k \left(\binom{N}{k} + \binom{N}{k-1} \right) (x-k)^N \\
&\quad + (-1)^{k'} \binom{N}{k'-1} (k'-k)^N,
\end{aligned}$$

and

$$\begin{aligned}
\binom{N}{k-1} &= \frac{N!}{(k-1)!(N-k+1)!} = \frac{kN!}{k!(N-k)!(N-k+1)} = \binom{N}{k} \frac{k}{N-k+1}, \\
\binom{N}{k-1} + \binom{N}{k} &= \binom{N}{k} \frac{N+1}{N-k+1} = \binom{N+1}{k},
\end{aligned}$$

so that

$$\begin{aligned}
\sum_{k=0}^{k'-1} (-1)^k \binom{N}{k} ((x-k)^N - (x-(k+1))^N) &= x^N + \sum_{k=1}^{k'-1} (-1)^k \binom{N}{k} \frac{N+1}{N-k+1} (x-k)^N \\
&\quad + (-1)^{k'} \binom{N}{k'-1} (k'-k)^N \\
&= \sum_{k=0}^{k'-1} (-1)^k \binom{N+1}{k} (x-k)^N \\
&\quad + (-1)^{k'} \binom{N}{k'-1} (k'-k)^N.
\end{aligned}$$

Similarly,

$$(-1)^{k'} \binom{N}{k'} (x-k')^N + (-1)^{k'} \binom{N}{k'-1} (k'-k)^N = (-1)^{k'} \binom{N+1}{k'} (x-k')^N$$

We can plug this into our expression for the density to find that

$$\begin{aligned}
f_{\sum_{i=1}^{N+1} X_i}(x) &= \frac{1}{N!} \sum_{k=0}^{k'} (-1)^k \binom{N+1}{k} (x-k)^N \\
&= \frac{1}{N!} \sum_{k=0}^N (-1)^k \binom{N+1}{k} (x-k)^N \mathbf{1}\{x \in [k, N+1]\},
\end{aligned}$$

so that Equation (A.2.2) holds also for $n = N+1$.

We may finally conclude by induction that Equation (A.2.2) holds for $n \in \mathbb{N}$, as required. \square

To find the density of a sum of $\text{Uni}[-1, 1]$, we can use the following approach. Let $X_i \sim \text{Uni}[-1, 1]$ i.i.d. for $i = 1, \dots, n$, and define $\tilde{X}_i = \frac{X_i+1}{2}$, so that $\tilde{X}_i \sim \text{Uni}[0, 1]$ i.i.d. for $i = 1, \dots, n$. We can use

Theorem A.2.2 to compute the density $f_{\sum_{i=1}^n \tilde{X}_i}$; from this we may find the density of $\sum_{i=1}^n X_i$ as

$$\begin{aligned} f_{\sum_{i=1}^n X_i}(x) &= \frac{d}{dx} \mathbb{P}\left(\sum_{i=1}^n X_i \leq x\right) = \frac{d}{dx} \mathbb{P}\left(\sum_{i=1}^n (2\tilde{X}_i - 1) \leq x\right) \\ &= \frac{d}{dx} \mathbb{P}\left(\sum_{i=1}^n \tilde{X}_i \leq \frac{x+n}{2}\right) = \frac{1}{2} f_{\sum_{i=1}^n \tilde{X}_i}\left(\frac{x+n}{2}\right). \end{aligned}$$

A.3 Exponential Splitting

In Subsection 4.2.1, we want to setup rate equations to describe how the number of vitrimer bonds across the healing interface changes over time. For this, we need to know how to properly combine the rates of a number of different species. Hence, in this appendix we will discuss a nice property of the exponential distribution, which we will call the ‘exponential splitting property’, since it is somewhat related to the splitting of Poisson processes.

The splitting property is made possible by the memorylessness of the exponential distribution. We will therefore now discuss what the memoryless property is, and prove that the exponential distribution is indeed the only possible distribution for a positive continuous random variable with the memoryless property.

Definition A.3.1 (Memoryless Continuous Distribution). Let T be a continuous random variable. Then, we say that T is *memoryless* if for all $t \in \text{supp } T$ and $s > 0$ with $t + s \in \text{supp } T$ we have that [33, p. 140]

$$\mathbb{P}(T > t + s \mid T > t) = \mathbb{P}(T > s). \quad (\text{A.3.1})$$

It is not so hard to see that a distribution with the memoryless property will have a constant hazard function, as

$$\mathbb{P}(T > t + s \mid T > t) = \mathbb{P}(T > s) \implies \underbrace{\lim_{s \downarrow 0} \frac{1}{s} \mathbb{P}(T \leq t + s \mid T > t)}_{=\lambda(t)} = \underbrace{\lim_{s \downarrow 0} \frac{1}{s} \mathbb{P}(T \leq s)}_{=f(0)},$$

by the definition of the hazard function (Equation (2.1.8)). It turns out that the converse is also true: every distribution with constant hazard function is memoryless. Moreover, we can show that the exponential distribution is the only memoryless lifetime distribution.

Lemma A.3.2 (Exponential Only Memoryless Positive Continuous Distribution). *Let T be a continuous random variable with the memoryless property and $\text{supp } T = (0, \infty)$. Then T must be exponentially distributed, i.e. there exists a $\tau > 0$ such that $T \sim \text{Exp}(\tau)$.*

Proof. Let T be a continuous random variable with the memoryless property and $\text{supp } T = (0, \infty)$. Furthermore, let $t \in (0, \infty)$ and $s > 0$. We can now apply the definition of conditional probabilities [79, Def. 1.4.(1)] to see that

$$\mathbb{P}(T > t + s \mid T > t) := \frac{\mathbb{P}(T > t + s \cap T > t)}{\mathbb{P}(T > t)}.$$

Then, Definition A.3.1 tells us that

$$\frac{\mathbb{P}(T > t + s \cap T > t)}{\mathbb{P}(T > t)} = \mathbb{P}(T > s), \text{ so that } \mathbb{P}(T > t + s \cap T > t) = \mathbb{P}(T > t + s) \mathbb{P}(T > t).$$

Now, choose $\frac{p}{q} \in \mathbb{Q}_{>0}$, where without loss of generality $p, q \in \mathbb{N}$. Then, we may apply the result above p times to find

$$\mathbb{P}\left(T > \frac{p}{q}\right) = \mathbb{P}\left(T > \sum_{i=1}^p \frac{1}{q}\right) = \prod_{i=1}^p \mathbb{P}\left(T > \frac{1}{q}\right) = \mathbb{P}\left(T > \frac{1}{q}\right)^p,$$

and note furthermore,

$$\mathbb{P}(T > 1) = \mathbb{P}\left(T > \sum_{i=1}^q \frac{1}{q}\right) = \mathbb{P}\left(T > \frac{1}{q}\right)^q, \text{ so that } \mathbb{P}\left(T > \frac{1}{q}\right) = \mathbb{P}(T > 1)^{\frac{1}{q}}.$$

Consequently, it holds that

$$\mathbb{P}\left(T > \frac{p}{q}\right) = \left(\mathbb{P}(T > 1)^{\frac{1}{q}}\right)^p = \mathbb{P}(T > 1)^{\frac{p}{q}}. \quad (\text{A.3.2})$$

Since T is a continuous random variable by assumption, we know that $\mathbb{P}(T > x)$ is a continuous function of x on $(0, \infty)$. As moreover Equation (A.3.2) holds for any arbitrary $\frac{p}{q} \in \mathbb{Q}_{>0}$, and $\mathbb{Q}_{>0}$ is dense in $\mathbb{R}_{>0} = (0, \infty)$ [84, p. 46], we may conclude that in fact Equation (A.3.2) holds on all of $(0, \infty)$, i.e.

$$\mathbb{P}(T > x) = \mathbb{P}(T > 1)^x \text{ for all } x \in (0, \infty).$$

Now, consider that necessarily $\mathbb{P}(T > 1) > 0$, as $\text{supp } T = (0, \infty)$, so that

$$\mathbb{P}(T > 1)^x = \exp(\log(\mathbb{P}(T > 1))x).$$

Hence, it holds that the Cumulative Distribution Function (CDF) of T is given by

$$F_T(x) := \mathbb{P}(T \leq x) = 1 - \mathbb{P}(T > x) = 1 - \exp(\log(\mathbb{P}(T > 1))x) \text{ for all } x \in (0, \infty).$$

We can now identify that F_T is the CDF of an exponential random variable with parameter $\tau = -\log(\mathbb{P}(T > 1))^{-1} \in (0, \infty)$ [33, p. 95], as required. \square

We can now finally get to the main result of this appendix: the exponential splitting property.

Theorem A.3.3 (Exponential Splitting). *Let $N \in \mathbb{N}$, let $\{T_i\}_{i=1}^N$ i.i.d. exponentially distributed. Suppose we have $\{t_i\}_{i=1}^N$ such that we know that $T_i \geq t_i$ for $i = 1, \dots, N$. Then,*

$$\mathbb{P}\left(\arg \min_{i=1, \dots, N} \{T_i - t_i\}_{i=1}^N = 1 \mid T_i \geq t_i, i = 1, \dots, N\right) = \frac{1}{N}. \quad (\text{A.3.3})$$

Proof. Fix $i \in \{1, \dots, N\}$. Then, we can see for $t > 0$ that

$$\mathbb{P}(T_i - t_i < t \mid T_i \geq t_i) = \mathbb{P}(T_i < t + t_i \mid T_i \geq t_i).$$

By Lemma A.3.2 we know that the exponential distribution is memoryless, so that

$$\mathbb{P}(T_i < t + t_i \mid T_i \geq t_i) = \mathbb{P}(T_i < t).$$

Hence, if we define $U_i := T_i - t_i \mid T_i \geq t_i$ for $i = 1, \dots, N$, we can see that

$$\mathbb{P}(U_i < t) = \mathbb{P}(T_i < t).$$

Consequently, we may find that

$$\mathbb{P}\left(\arg \min_{i=1, \dots, N} \{T_i - t_i\}_{i=1}^N = 1 \mid T_i \geq t_i, i = 1, \dots, N\right) = \mathbb{P}\left(\arg \min_{i=1, \dots, N} \{U_i\}_{i=1}^N = 1\right).$$

Since $\{U_i\}_{i=1}^N$ i.i.d., we can conclude that each i is equally likely to minimise U_i . Since $\{U_i\}_{i=1}^N$ contains N i.i.d. copies, we can then conclude that

$$\mathbb{P}\left(\arg \min_{i=1, \dots, N} \{U_i\}_{i=1}^N = 1\right) = \frac{1}{N},$$

as required. \square

We may apply Theorem A.3.3 in the following way: we fix some time $t > 0$. Then, we know that we have N_0 vitrimer bonds, which have each last swapped some known time ago. The time between two successive swaps of a bond is i.i.d. for all bonds. Hence, we can conclude that each bond is equally likely to be the next to swap after time t ; each bond has a probability of N_0 of being the ‘chosen one’.

Glossary

doing property that swap does not undo the previous swap, so that the new partner is not the same as the one before last.. 24, 27, 28, 30–37, 51, 52

healing interface interface between two pieces of vitrimeric material across which healing can take place. 23, 24, 37–43, 45–48, 52, 63

HOOMD-blue particle simulation tool [61]. 17, 19–22

memoryless property that distribution of random variable has no memory, see Definition A.3.1. 63, 64

moiety part of a molecule. 5, 6, 14, 19, 22, 24–40, 45, 49, 52, 53

OVITO Open VIualization TOol, software for visualising molecular simulations [26]. 6, 14, 15

self-healing property of material that can autonomously recover mechanical properties after damage. 1, 3, 4, 12, 14, 15, 17, 23, 24, 27, 30, 31, 40, 42, 43, 45, 48, 49, 51–53

thermoplastic polymer material held together by weak intramolecular forces. 3, 4

thermoset polymer material held together by irreversible crosslinks. 3, 4

undoing property that swap undoes the previous swap, so that the new partner is the same as the one before last.. 24, 27, 28, 30–37, 51, 52

vitrimer polymer material with exchangeable crosslinks. 1, 4–6, 12–17, 19, 21–25, 27–30, 32–37, 40, 43–53, 63, 64

Acronyms

NVE Number, Volume, Energy. 20, 22, 23

NVT Number, Volume, Temperature. 21–23

a.e. almost everywhere. 59, 60

CAN Covalent Adaptive Network. 4, 24, 31, 51

CDF Cumulative Distribution Function. 64

DAM Defect-Allowing Mixture. 14, 15, 22

DFM Defect-Free Mixture. 14

EU European Union. 1, 3

FENE Finitely Extensible Nonlinear Elastic. 17

GLJ Generalised Lennard-Jones. 19

i.i.d. independent and identically distributed. 7, 38, 60, 62, 64

KDE Kernel Density Estimator. 8, 10, 32–35, 37, 46, 51

KHE Kernel Hazard Estimator. 10, 35–37, 51

KM Kaplan-Meier. 7, 8, 10–12, 25, 32, 33

LJ Lennard-Jones. 18, 19, 22, 23

LOOCV Leave-One-Out Cross-Validation. 9, 52

MC Monte Carlo. 16, 19, 25, 36, 38

MD Molecular Dynamics. 1, 16, 17, 19, 20, 22, 25, 32, 36, 43

NA Nelson-Aalen. 10

PDF Probability Density Function. 7–10, 34, 46, 51, 59

RevCross Reversible Crosslinks. 19, 22, 23, 25, 50

w.p. with probability. 39

WCA Weeks-Chandler-Andersen. 18, 19